DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS
Volume **15**, Number **1**, May **2006**

Website: http://aimSciences.org

pp. **281–352**

EUCLIDEAN DYNAMICS

Brigitte Vallée

GREYC, UMR CNRS 6072, University of Caen
Batiment Sciences 3, Campus II
F-14032 Caen, France

Abstract. We study a general class of Euclidean algorithms which compute the greatest common divisor [gcd], and we perform probabilistic analyses of their main parameters. We view an algorithm as a dynamical system restricted to rational inputs, and combine tools imported from dynamics, such as transfer operators, with various tools of analytic combinatorics: generating functions, Dirichlet series, Tauberian theorems, Perron's formula and quasi-powers theorems. Such dynamical analyses can be used to perform the average-case analysis of algorithms, but also (dynamical) analysis in distribution.

1. **Introduction.** Computing the Greatest Common Divisor [Gcd] –on integers or polynomials– is a central problem in computer algebra, and all the gcd algorithms are based on main principles due to Euclid. See for instance [113] or [110] for a description of use of Gcd Algorithms in Computer Algebra. According to Knuth [56], "we might call Euclid's method the granddaddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day." Indeed, Euclid's algorithm is currently a basic building block of computer algebra systems and multiprecision arithmetic libraries, and, in many such applications, most of the time is spent in computing gcd's. However, the Euclidean algorithms have not yet been completely analyzed, and it is the purpose of this paper to provide such an analysis.

1.1. **Various divisions.** All the basic Gcd algorithms can be described as a sequence of divisions, and the Gcd Algorithms mainly depend on what kind of division is used.

On polynomials, there exist two possible divisions: the first one is directed by leading monomials (i.e., monomials of highest degree) and deals with decreasing degrees. The second one is directed by monomials of lowest degree and deals with increasing valuations. In fact, the probabilistic behaviours of the two associated gcd algorithms are similar, since the execution of the first algorithm on the pair (u, v) coincides with the execution of the second algorithm on the mirror pair $(\overline{u}, \overline{v})$ formed with the mirrors $\overline{u}, \overline{v}$ of u, v.

On integer numbers, there exist many different divisions: on a pair (u, v) of integers, a division performs a decomposition of the form $v = m \cdot u + r$, with a quotient m and a remainder r. Here, all the integers are written in base 2, and we work with their Most Significant Bits (MSB's) [i.e., the bits on the left of the binary expansion] or with their Least Significant Bits (LSB's) [i.e., the bits on the

2000 *Mathematics Subject Classification.* Primary: 68Q25, 68W40, 37E05, 37C30, 11M41; Secondary : 47A.

Key words and phrases. Analysis of algorithms, dynamical systems of the interval, transfer operators, Dirichlet series .

right of the binary expansion]. The choice of pair (m, r) can be directed by the MSB's of the integers u, v or by their LSB's; for instance, the usual division, which is directed by the MSB's, aims to combine v with a multiple of u, of the form $m \cdot u$ in order to create zeroes on the MSB's [i.e., on the left]: then, the remainder r has a smaller absolute value than u. On the contrary, the LSB division is directed by the Least Significant Bits: it aims to combine v with a multiple of u, of the form $m' \cdot u$ in order to create zeroes on the LSB's [i.e., on the right]; then, the remainder r has more zeroes on the right than u: the 2–adic absolute value of r is smaller than the 2–adic absolute value of u. Here, what we call a "direction" or a "decision" is related to the choice of the pair (m, r). However, after this choice, all the computations [multiplication of u by m, subtraction $r := v - m \cdot u$ are the usual ones, and are performed, as usual, from the right to the left. The carry propagates also from the right to the left. This explains that all these algorithms do not have the same behaviour, because of the carry propagation, which may play a different rôle in these various divisions. In particular, the mirror property of polynomials is lost for integers.

These integer divisions, and thus the Euclidean algorithms based on these divisions, can be gathered into four groups, or four types:

The MSB Group [or Type 1] contains all the divisions which choose the quotient according to the MSB's : it is the analogue, for the numbers, of the decreasing–degree algorithms for polynomials. It contains of course the (Standard) Euclid algorithm, but also its possible variants, according to the position of remainder r [Centered division, By-Excess division, α–division, as described in [19]], or the parity of quotient m [Odd division, Even division]. Finally the Subtractive Algorithm does not perform any divisions, only subtractions. [See Figure 1].

It is also interesting to consider divisions which choose the quotient according to the LSB's. The LSB Group [or Type 4], is the integer analogue to increasing–valuation gcd algorithm for polynomials. Such a gcd algorithm is described in [98] for instance. In fact, there are two LSB divisions, the Plain LSB division and the Centered LSB division, according to the position of the quotient [non centered or centered].

There also exist two mixed groups which operate a sort of transition between these two extremal groups; the mixed divisions are directed by both MSB's and LSB's, in order to create zeroes both on the right and on the left. However, the dominant rôle can be played by the MSB's or LSB's.

For some divisions, the decision is mostly made by the MSB's, and the LSB's play only an auxilliary rôle; these divisions form the MLSB Group [or Type 2] which contain the so–called pseudo–Euclidean Algorithms, introduced by Shallit [88] and Vallée [106], [107]. Roughly speaking, a pseudo–division is just a MSB division where powers of two are removed from the remainder, after the division: This means that the pair (m, r) is chosen according to the MSB's, and, after this, there is a binary shift a on r directed by the LSB's which creates an odd pseudo-remainder s which satisfies $r := 2^a \cdot s$. Pseudo–divisions give rise to algorithms which only deal with odd numbers, and they are well–adapted to computing the Jacobi symbol [52] [61], for instance [the Quadratic Reciprocity law being only true for a pair of odd integers].

For the Binary division of Stein [96] described in [56] and the Plus-Minus division, of Brent and Kung [15], the main decision is made by the LSB's; the MSB's play

only an auxilliary rôle, and only decide when the exchange has to be done: these two algorithms form the LMSB Group [or Type 3]

Finally, polynomial divisions form their own type, [Type 0], which contains the two divisions previously described.

1.2. **A general framework for gcd algorithms.** Any gcd algorithm performs a sequence of steps. Each step is formed with a division, (possible) binary shifts (uniquely for numbers), and (possible) sign changings. The total operation performed in each step is called a division step. Such a step is followed by an exchange. We will see in the following that the probabilistic behaviour of a gcd algorithm heavily depends on the division-step which will be used. For the moment, in this Section, we describe the general framework for all the gcd algorithms which are studied in this paper.

For all types, each division–step can be written as

$$u = 2^a \cdot u', \qquad v = m \cdot u' + \epsilon \cdot 2^b \cdot r'.$$

It performs (before the next division) a first binary shift equal to some integer $a \geq 0$ on divisor u, then the division itself, which produces a remainder shifted by a shift equal to some integer $b \geq 0$. Remark that the shift b is produced by the division itself. This remainder has also a sign $\epsilon = \pm 1$. Here u, v, m, u', r' are integers[1]. The division uses a "digit" $d = (m, \epsilon, a, b)$, and changes the old pair (u, v) into the new pair (r', u') and can be written as a matrix transformation

$$\begin{pmatrix} u \\ v \end{pmatrix} = M_{[d]} \begin{pmatrix} r' \\ u' \end{pmatrix}, \qquad M_{[d]} := \begin{pmatrix} 0 & 2^a \\ \epsilon\, 2^b & m \end{pmatrix}. \tag{1.1}$$

For Types 0 and 1, there are no binary shifts to be used, and the two exponents a and b equal 0. For Type 2, the shift a is possibly not zero, while b equals 0. For Type 3, a equals 0, while the shift b is always non zero. Finally, for Type 4, the two exponents a and b are equal and non zero.

Instead of "integer" pairs (u, v), (r', u'), we consider "rationals" [the old rational $x = u/v$, and the new rational $y = r'/u'$], and we wish to describe the relation induced by the division on x, y: For each digit $d = (m, \epsilon, a, b)$, there exists a linear fractional transformation (LFT) $h_{[d]}$, associated to the matrix $M_{[d]}$ of Eqn (1.1) for which

$$x = h_{[d]}(y) \qquad \text{with} \quad h_{[d]}(y) = \frac{2^a}{m + \epsilon\, 2^b y}\ .$$

Remark that the absolute value $|\det h_{[d]}|$ of the determinant of the LFT $h_{[d]}$ is equal to 2^{a+b} and thus involves the total number $a + b$ of binary shifts that are used in the division-step.

Any execution of a gcd algorithm can be described as follows. On the input pair $(u, v) = (u_1, u_0)$, it will be of the form

$$\left\{ \begin{array}{lllllllll} u_1 & := & 2^{-a_1} u_1, & u_0 & = & m_1\, u_1 & + & \epsilon_1\, 2^{b_1}\, u_2, \\ u_2 & := & 2^{-a_2} u_2, & u_1 & = & m_2\, u_2 & + & \epsilon_2\, 2^{b_2}\, u_3, \\ & & \cdots, & & & \cdots & & \\ u_i & := & 2^{-a_i} u_i, & u_{i-1} & = & m_i\, u_i & + & \epsilon_i\, 2^{b_i} u_{i+1} \\ & & \cdots, & & & \cdots & & \end{array} \right\}, \tag{1.2}$$

[1] "integer" with quote has here a generic meaning; it denotes integer (numbers) or polynomials while "rationals" denote rational numbers or rational fractions.

and uses the sequence of digits $d_i := (m_i, \epsilon_i, a_i, b_i)$. It stops at the p–th iteration with $u_{p+1} = \eta \cdot u_p$. Then $\gcd(u, v) = u_p$. Very often, the final value η equals 0, but, in some cases, η may be equal to 1.

On an input (u, v) whose gcd is d, the execution (1.2) creates a matrix product of the form

$$\begin{pmatrix} u \\ v \end{pmatrix} = M_1 \cdot M_2 \cdot \ldots \cdot M_p \begin{pmatrix} \eta d \\ d \end{pmatrix} = M \begin{pmatrix} \eta d \\ d \end{pmatrix} \qquad \text{with} \quad M_i := M_{[d_i]}, \tag{1.3}$$

and also a continued fraction expansion (CFE) of the form

$$\frac{u}{v} = \cfrac{e_0}{m_1 + \cfrac{e_1}{m_2 + \cfrac{e_2}{\ddots + \cfrac{e_{p-1}}{m_p + e_p\,\eta}}}} = h_1 \circ h_2 \circ \ldots h_p(\eta) = h(\eta). \tag{1.4}$$

Here, the LFT's h_i are defined as $h_i := h_{[d_i]}$ and the numerators e_i which appear in the CFE are defined as

$$e_i := \epsilon_i \cdot 2^{b_i + a_{i+1}}, \qquad \text{with} \quad \epsilon_0 = 1, \quad b_0 = a_{p+1} = 0. \tag{1.5}$$

Remark that h in (1.4) is a LFT, and M in (1.3) is an integer matrix of the form

$$h(x) = \frac{\alpha x + \beta}{\gamma x + \delta}, \qquad M = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \qquad \text{with } \alpha, \beta, \gamma, \delta \text{ coprime integers.}$$

When the algorithm performs p iterations, it gives rise to a continued fraction of depth p [the depth of a continued fraction equals its number of levels].

For each algorithm, we define a pair (u, v) to be valid if the rational u/v satisfies the same conditions as the ouput rational r'/u' created by the division. We choose as the set of inputs of the algorithm, the set of valid pairs. In this way, the first step of the algorithm will resemble any other step of the algorithm. Quite often, each particular algorithm defines a precise subset $\mathcal{H}$ of LFT's which it uses at each iteration [except perhaps for the first and the final steps where it may use some slightly different sets $\mathcal{J}$ and $\mathcal{F}$]. See Figure 1 for examples of such a situation for the MSB Class [Type 1]. Then, Relation (1.4) [summarized by $u/v = h(\eta)$] defines a bijection between the set Ω of valid coprime inputs (u, v) of the algorithm and the set of LFT's $\mathcal{I} \cdot \mathcal{H}^\star \cdot \mathcal{F}$ where $\mathcal{H}^\star$ denotes the semi-group $\cup_{k \geq 0} \mathcal{H}^k$.

For Algorithms of Type 2, the valid inputs are restricted to be odd. Since, for this type, the shift b is zero, there is a relation between parity of the quotient m_i at step i and the exponent a_{i+1} at the beginning of the $i+1$-th step: if m_i is odd, then the remainder is even, and thus a_{i+1} satisfies $a_{i+1} \geq 1$; if m_i is even, then the remainder is odd, and thus a_{i+1} equals 0. We then consider two states: the 0–state, which means "the previous quotient of (v, u) is even" (or equivalently the previous remainder is odd), i.e., the present shift a equals 0; the 1–state, which means "the previous quotient of (v, u) is odd" (or equivalently the previous remainder is even), i.e., the present shift a satisfies $a \geq 1$. Then, the processus is of a Markovian type, and it uses four different sets $\mathcal{H}_{<i|j>}$, where $\mathcal{H}_{<i|j>}$ brings rationals from state i to state j. The initial state is always the 0–state and the final state is always the 1–state. See Figure 14 for instances of such a situation.

1.3. **Main parameters.** The main parameters which describe the execution of the algorithm on the input (u, v), namely the digits and the continuants, can be read on the continued fraction of u/v. The m_i's are called the quotients, the triples $d_i = (m_i, \epsilon_i, a_i, b_i)$ are the digits. The continuants are defined when one "splits"

Alg., X. η	*Division*	*Set of LFT's*	*Conditions on $\mathcal{J}$ or $\mathcal{F}$.*
(*G*) [0, 1], 0	$v = mu + r$ $0 \le r < u$	$\mathcal{G} = \{\frac{1}{m+x},\ m \ge 1\}$	$\mathcal{F} = \mathcal{G} \bigcap \{m \ge 2\}$
(*M*) [0, 1], 1	$v = mu - r$ $0 \le r < u$	$\mathcal{M} = \{\frac{1}{m-x},\ m \ge 2\}$	$\mathcal{F} = \mathcal{M} \bigcap \{m \ge 3\}$
(*K*) $[0, 1/2]$, 0	$v = mu + \epsilon r$ $\epsilon = \pm 1$, $(m, \epsilon) \ge (2, +1)$ $0 \le r < \frac{u}{2}$	$\mathcal{K} = \{\frac{1}{m+\epsilon x}, \epsilon = \pm 1,$ $(m, \epsilon) \ge (2, +1)\}$	$\mathcal{F} = \mathcal{K} \bigcap \{\epsilon = 1\}$
(*E*) [0, 1], 1	$v = mu + \epsilon r$ m even, $\epsilon = \pm 1$, $0 < r < u$	$\mathcal{E} = \{\frac{1}{m+\epsilon x}, m$ even, $\epsilon = \pm 1\}$	$\mathcal{F} = \mathcal{E} \bigcap \{\epsilon = 1\}$
(*O*) [0, 1], 0	$v = mu + \epsilon r$ m odd, $\epsilon = \pm 1$, $0 \le r < u$	$\mathcal{O} = \{\frac{1}{m+\epsilon x}, m$ odd, $\epsilon = \pm 1,$ $(m, \epsilon) \ge (1, 1)\}$	$\mathcal{F} = \mathcal{O} \bigcap \{m \ge 3, \epsilon = 1\}$
(*T*) [0, 1], 0	$v = u + r$	$\mathcal{T} = \{q = \frac{1}{1+x}, p = \frac{x}{1+x}\}$	Finishes with pq

FIGURE 1. The six Euclidean algorithms of the MSB Class.

the CFE (1.4) of u/v at depth i, or when one splits the matrix product (1.3) at step i; one obtains two CFE's defining a rational number [or two matrix products defining an integer vector]. The first one defines the beginning rational p_i/q_i, [with coprimes p_i, q_i]. The pair $Q_i := (p_i, q_i)$ is called the i-th beginning continuant. Pair (p_i, q_i) is defined, from (1.4) or (1.3), as

$$\frac{p_i}{q_i} := \cfrac{e_0}{m_1 + \cfrac{e_1}{m_2 + \cfrac{e_2}{m_3 + \cfrac{e_3}{\ddots + \cfrac{e_{i-1}}{m_i}}}}} = h_1 \circ h_2 \circ \ldots \circ h_i(0), \tag{1.6}$$

or, with the matrix point of view,

$$Q_i = \begin{pmatrix} p_i \\ q_i \end{pmatrix} = M_1 \cdot M_2 \cdot \ldots \cdot M_i \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.7}$$

Rationals p_i/q_i are often called convergents, and they are useful for approximating the rational u/v. The beginning continuants are closely related to Bezout sequences that appear in the Extended Euclidean Algorithms which compute not only the gcd but also the Bezout coefficients.

The second CFE defines the ending rational t_i/v_i with coprimes t_i, v_i, The i-th ending continuant is the pair $V_i := (t_i, v_i)$. Pair (t_i, v_i) is defined, from (1.4) or (1.3), as

$$\frac{t_i}{v_i} := \cfrac{2^{a_{i+1}}}{m_{i+1} + \cfrac{e_{i+1}}{m_{i+2} + \cfrac{e_{i+2}}{m_{i+3} + \cfrac{e_{i+3}}{\ddots + \cfrac{e_{p-1}}{m_p + \epsilon_p 2^{b_p} f}}}}} = h_{i+1} \circ h_2 \circ \ldots \circ h_p(\eta). \tag{1.8}$$

or, with the matrix point of view,

$$V_i = \begin{pmatrix} t_i \\ v_i \end{pmatrix} = M_{i+1} \cdot M_{i+2} \cdot \ldots \cdot M_p \begin{pmatrix} \eta \\ 1 \end{pmatrix} \tag{1.9}$$

The ending continuants are closely related to the remainder pairs $U_i := (u_{i+1}, u_i)$ that appear in the execution (1.2) of the Euclidean algorithms on input (u, v), via the relation $U_i = \gcd(u, v)\ V_i$.

1.4. **Probabilistic analyses.** Here, we are interested in the probabilistic behaviour of each algorithm: it is mainly described by the probabilistic behaviour of digits and continuants, which play here the rôle of observables. As it is usual in complexity studies, we consider the set formed of the valid inputs (u, v) of a gcd algorithm and we first define both an absolute value and a size on it.

In fact, we consider two sets Ω and $\widetilde{\Omega}$: the second is formed with all the valid inputs of the algorithm [the notion of valid inputs is defined in Section 1.2], while the first one only contains the coprime valid inputs. We mainly deal with the set Ω. It may seem strange –at least from an algorithmic point of view– to study sets of inputs for which the answer of the algorithm is trivial! However, we shall prove that this (trivial) set is in a sense generic, and it will be easy to transfer the results on Ω to the (more natural) set $\widetilde{\Omega}$ [see Section 1.6 for a first explanation of this fact].

We first define the absolute value of an "integer" u. In the polynomial case [Type 0], the absolute value of a non zero polynomial u of $\mathbb{F}_q[Z]$ is $|u| := q^{\deg u}$, while its size $\ell(u)$ is $\ell(u) := 1 + \deg u$. In the integer case [all the other types], the absolute value is the usual (archimedean) one, and the size of a (non-zero) integer u is its binary length, equal to $\ell(u) := 1 + \lfloor \log_2 |u| \rfloor$. In both cases, the size $\ell(u)$ is $\Theta(\log |u|)$.

We wish to define the absolute value of a pair (u, v) as a function of the absolute values of u and v. There are two main cases, according to the type [Type 4, or all the other types].

Case (a). For all the algorithms [except Type 4], the absolute value of remainder r' is less than the absolute value of u'. This means that any valid input (u, v) satisfies $|u| \leq |v|$, and it is adequate to choose as the absolute value of the input pair (u, v) the maximum of the absolute values $|u|$ and $|v|$, which equals $|v|$. Finally, the absolute value $|(u, v)|$ of the pair (u, v) and its size $L(u, v)$ are

$$|(u, v)| := \max(|u|, |v|) = |v|, \qquad L(u, v) := \ell(|(u, v)|) = \ell(|v|), \tag{1.10}$$

Moreover, using the equality $u/v = h(\eta)$ —where η is the final rational value of the algorithm, and h is the LFT built by the algorithm [see (1.4)] — entails the important relation, valid for any coprime input pair (u, v),

$$|(u, v)| = |v| = D[h](\eta), \qquad \text{where } \eta \text{ is the final value of the algorithm,} \tag{1.11}$$

which involves the denominator function $D[h]$ of the LFT h, defined by

$$D[h](x) := |\gamma x + \delta|, \qquad \text{for} \quad h(x) := \frac{\alpha x + \beta}{\gamma x + \delta} \qquad \text{with} \quad \alpha, \beta, \gamma, \delta \text{ coprime integers.}$$

Case (b). For the LSB Group [Type 4], it is no longer true that the absolute value of remainder r' is less than the absolute value of u'. The set of rationals u/v related to valid inputs (u, v) is no longer a subset of a compact interval of $\mathbb{R}$. Then, it is convenient to deal with the Euclidean norm and to choose as the absolute value

$|(u,v)|$ of the input pair (u,v) the Euclidean norm $(u^2+v^2)^{1/2}$. Then, the absolute value $|(u,v)|$ of the pair (u,v) and its size $L(u,v)$ are

$$|(u,v)| := (u^2+v^2)^{1/2}, \qquad L(u,v) := \ell(|(u,v)|) = \frac{1}{2}\ell(u^2+v^2). \tag{1.12}$$

Moreover, for any coprime input pair (u,v), the final pair of the LSB algorithm is $(0,1)$, and using the equality $(u,v) = M(0,1)$ entails the relation

$$(u^2+v^2) = ||(u,v)||^2 = ||M(0,1)||^2, \tag{1.13}$$

where M is the matrix built by the algorithm [see (1.3)].

Finally, for all types, the sets

$$\Omega_N := \{(u,v) \in \Omega; \quad L(u,v) = N\}, \quad \widetilde{\Omega}_N := \left\{(u,v) \in \widetilde{\Omega}; \quad L(u,v) = N\right\} \tag{1.14}$$

gathers valid inputs of size N and are endowed with uniform probabilities denoted by $\mathbb{P}_N, \widetilde{\mathbb{P}}_N$. We denote by $\mathbb{E}_N, \widetilde{\mathbb{E}}_N$ the associate expectations.

1.5. **Main parameters.** We wish to analyze the probabilistic behavior of the main observables (as digits or continuants) on the set Ω_N, when the size N of the input (u,v) becomes large. We then (easily) come back to $\widetilde{\Omega}_N$. Sometimes, for distributional analysis, we work on the set

$$\Omega_N^+ := \bigcup_{M \le N} \Omega_M, \qquad \widetilde{\Omega}_N^+ := \bigcup_{M \le N} \widetilde{\Omega}_M.$$

The complexity analysis of each algorithm first aims to study the number of iterations that are performed during the execution (1.2). More generally, we wish to study three kinds of parameters: digit-costs, continuants and bit–complexity.

Digit-costs. These are general additive parameters which (only) depend on the sequence of the digits $d_i = (m_i, \epsilon_i, a_i, b_i)$. More precisely, we consider the set $\mathcal{D}$ of the possible digits, and we define a cost $c(d)$ relative to each digit d. Since the LFT $h_{[d]}$ only depends on digit d, this cost can also be defined on $\mathcal{H}$, via $c(h) := c(d)$ for $h = h_{[d]}$. We can extend this cost to the semi-group $\mathcal{H}^*$ in an additive way, and deal with "additive" costs where the total cost is the sum of elementary costs of each step. We then attach to the execution (1.2) of the gcd algorithm on the input (u,v) the total cost $C(u,v)$ defined by

$$C(u,v) := \sum_{i=1}^{p} c(d_i) = c(h) \qquad \text{if} \qquad \frac{u}{v} = h(\eta). \tag{1.15}$$

We consider a large class of digit-costs c, and in particular the so–called costs of moderate growth where digit-cost c is $O(\ell)$ where ℓ is the size [of a digit]. This class contains in particular some particular parameters which are are of great algorithmic interest. For instance, if $c = 1$, then C is the number of iterations. If c is the characteristic function of some particular quotient m_0, then C is the number of occurrences of this particular quotient during the execution of the algorithm. If c is the size $\ell(d)$ of the digit d, then C is the length of the binary encoding of the continued fraction.

Continuants. Second, we wish to describe the evolution of the size $L(Q_i)$, $L(V_i)$ of the beginning or ending continuants, together with the size $L(U_i)$ of the remainder pairs U_i. These parameters are central in the study of the extended gcd algorithms, and also in the so–called interrupted gcd algorithms which stop as soon as the size

$L(U_i)$ of the i-th remainder becomes smaller than a given proportion of the input size $L(U_0)$.

Bit-complexities. Third, the bit–complexity is the most possible precise complexity measure of a gcd–algorithm; It is is defined as the sum of the binary costs of each division-step, and the binary cost of a division of the form $v = mu + \epsilon r$ is equal to $\ell(m) \cdot \ell(u)$. Then, the bit-complexity B of a gcd algorithm, is defined as

$$B(u, v) := \sum_{i=1}^{p} \ell(d_i) \cdot L(U_i),$$

and involves parameters of both types. This is the same situation for the bit–complexity of the extended gcd algorithm

$$\widehat{B}(u, v) := B(u, v) + \overline{B}(u, v) \qquad \text{with} \quad \overline{B}(u, v) := \sum_{i=1}^{p} \ell(d_i) \cdot L(Q_i),$$

which involves all the various kinds of parameters. Remark that such costs, even if they are written as a sum of elementary costs, are no longer additive, since [for instance] the term $\ell(d_i) \cdot L(Q_i)$ involves the i-th continuant Q_i which depends on all the previous steps.

We wish to provide probabilistic analyses for all these parameters: the first study aims to exhibit their average–case behaviour [expectation, and more generally, moments of higher order], while the second study, much more precise, aims to describe their asymptotic distribution, when the size N of the inputs becomes large.

1.6. **The main steps of a "dynamical analysis".** Until 95, the methods employed in Euclidean Analysis are rather disparate, and their applicability to new situations is somewhat unclear. See Section 9 for an historical account. Then, during the last ten years, the Caen Group designed a unifying framework for the analysis of Euclidean Algorithms that additionally provided new results, for the average–case analysis as well for distributional analysis. All the analyses which will be described here are instances of this methodology, the so–called dynamical analysis, where one proceeds in three main steps: First, the (discrete) algorithm is extended into a continuous process, which can be defined in terms of a dynamical system, where executions of the gcd algorithm are then described by particular trajectories [i.e., trajectories of "rational" points]. Second, the main parameters of the algorithm are extended and studied in this continuous framework: the study of particular trajectories is replaced by the study of generic trajectories. Finally, one operates a transfer "from continuous to discrete", and proves that the probabilistic behaviour of gcd algorithms [related to "rational" trajectories] is quite similar to the behaviour of their continuous counterparts [related to generic trajectories].

1.7. **The Dirichlet moment generating functions.** Our main tool is, as it is generally the case in analysis of algorithms, generating functions. The crucial rôle of generating functions in the analysis of data structures and algorithms is well described in books by Flajolet and Sedgewick [33, 34].

We consider a general parameter R defined on $\Omega, \widetilde{\Omega}$, and we wish to study its distribution on Ω_N, when endowed with the uniform probability. Our final probabilistic tool [for distributional analyses] is the sequence of moment generating functions $\mathbb{E}_N[\exp(wR)]$,

$$\mathbb{E}_N[\exp(wR)] = \frac{\Phi_w(N)}{\Phi_0(N)}, \qquad \text{with} \quad \Phi_w(N) := \sum_{(u,v)\in\Omega_N} \exp[wR(u,v)]. \tag{1.16}$$

If we restrict ourselves to average-case analysis, we aim studying all the moments of order k, namely

$$\mathbb{E}_N[R^k] = \frac{1}{\Phi_0(N)} \cdot \frac{\partial^k}{\partial w^k} \Phi_w(N)|_{w=0} = \frac{1}{\Phi_0(N)} \cdot \sum_{(u,v)\in\Omega_N} R^k(u,v).$$

We first consider the whole set Ω of inputs and our strategy consists in encapsulating all the moment generating functions $\mathbb{E}_N[\exp(wR)]$ in a unique Dirichlet series $S_R(s,w)$,

$$S_R(s,w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^{2s}} \exp[wR(u,v)], \tag{1.17}$$

which deals with the absolute values of inputs (u,v) defined in (1.10) or in (1.12) according to the type [Type 4, or not]. Remark that our main object of interest, the moment generating functions $\mathbb{E}_N[\exp(wR)]$ can be easily recovered with coefficients of series $S_R(s,w)$: if $\phi_w(n)$ denotes the cumulative value of cost $\exp[wR]$ on pairs (u,v) whose absolute value equals n,

$$\phi_w(n) := \sum_{(u,v);|(u,v)|=n} \exp[wR(u,v)],$$

then the series $S_R(s,w)$ is of the form

$$S_R(s,w) := \sum_{n\geq 1} \frac{\phi_w(n)}{n^{2s}}, \qquad \text{with} \qquad \sum_{n;\ell(n)=N} \phi_w(n) = \Phi_w(N). \tag{1.18}$$

[Here, ℓ is the size [defined in Section 1.4] and $\Phi_w(N)$ is defined in (1.16)]. Then, Equations (1.16) and (1.18) show that the moment generating function $\mathbb{E}_N[\exp(wR)]$ is a ratio, where numerators and denominators are sums of coefficients of the Dirichlet series $S_R(s,w)$.

The series $S_R(s,w)$ is a generating function, of Dirichlet type with respect to the variable s. It is important to notice that, in the polynomial case, it is also a power series with respect to $z = q^{-2s}$, denoted by $T_R(z,w)$,

$$T_R(z,w) := \sum_{(u,v)\in\Omega} z^{\deg v} \exp[wR(u,v)]. \tag{1.19}$$

Parameters s or z "mark" the size, while the parameter w "marks" the cost R.

We can adopt the same strategy for each moment of order k, and we obtain a Dirichlet Series $S_R^{[k]}(s)$ with respect to the unique variable s, which is the k-th derivative of $w \mapsto S_R(s,w)$ at $w=0$,

$$S_R^{[k]}(s) := \frac{\partial^k}{\partial w^k} S_R(s,w)|_{w=0} = \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^{2s}} R^k(u,v). \tag{1.20}$$

As previously, these series are of Dirichlet type with respect to the variable s; moreover, they are power series with respect to $z = q^{-2s}$ in the polynomial case,

$$T_R^{[k]}(z) := \sum_{(u,v)\in\Omega} z^{\deg v}\, R^k(u,v). \tag{1.21}$$

And, in the same vein as above, the moment $\mathbb{E}_N[R^k]$ of order k can be easily recovered with coefficients of series $S_R^{[k]}(s)$: if $\phi^{[k]}(n)$ denotes the cumulative value of cost R^k relative to pairs (u,v) whose absolute value equals n,

$$\phi^{[k]}(n) := \sum_{(u,v);|(u,v)|=n} R^k(u,v),$$

then the series $S_R^{[k]}(s)$ is of the form

$$S_R^{[k]}(s) := \sum_{n\geq 1} \frac{\phi^{[k]}(n)}{n^{2s}}, \qquad \text{with} \quad \mathbb{E}_N[R^k] = \frac{\Phi^{[k]}(N)}{\Phi^{[0]}(N)}, \quad \Phi^{[k]}(N) := \sum_{n;\ell(n)=N} \phi^{[k]}(n).$$

Then, the previous equations show that each moment $\mathbb{E}_N[R^k]$ of order k is a ratio, where numerators and denominators are sums of coefficients of the Dirichlet series $S_R^{[k]}(s)$.

We look for alternative expressions for the bivariate generating functions $S_R(s,w)$. [Note that these expressions can be easily transfered to the univariate generating functions $S_R^{[k]}(s)$]. From these alternative expressions, the position and the nature of the (dominant) singularity of $S_R(s,w)$ become apparent. These informations will be transfered to informations about the asymptotics of coefficients, and provide distributional analyses [if we can work with the bivariate generating functions $S_R(s,w)$] or only average-case analysis [if it is only possible to deal with univariate functions $S_R^{[k]}(s)$].

Since the natural set of inputs is the set $\widetilde{\Omega}$ defined in (1.14), it would be convenient to directly deal with the tilde series $\widetilde{S}_R(s,w)$ and $\widetilde{S}_R^{[k]}(s)$ relative to set $\widetilde{\Omega}$, and deal with the tilded objects in the same way as with the untilded objects. However, it is easier to work with untilded versions, and, for the particular costs to be studied [additive costs C associated to digit costs c, or continuant lengths], there exists a simple relation between $S_R(s,w)$ and its tilde version. In the case when $R = C$ or $R = \log(|Q_i|), \log(|V_i|)$, one has $R(du,dv) = R(u,v)$, which entails the equality

$$\widetilde{S}_R(s,w) = \zeta(2s)\cdot S_R(s,w) \tag{1.22}$$

where $\zeta(s)$ is the Zeta series of possible gcd's. In the case when $R = \log(U_i)$, the relation $R(du,dv) = \log d + R(u,v)$ entails the equality

$$\widetilde{S}_R(s,w) = \zeta(2s-2w)\cdot S_R(s,w). \tag{1.23}$$

Taking the derivatives with respect to w in (1.22) or in (1.23) provides relations between the series $S_R^{[k]}(s)$ and its tilde version. With well–known properties of the zeta function, it will be easy to transfer properties of the untilded generating functions to tilded generating functions. This is why it will be sufficient to study coprime inputs, as we already said in Section 1.4.

Plan of the paper. The plan of the paper will follow the main steps of Dynamical Analysis, as described in Section 1.5. Section 2 describes the main notions on dynamical systems which will be used here, namely random dynamical systems, transfer operators, induction method, etc... Section 3 builds the dynamical

systems which extend the gcd algorithms. Then, in Section 4, we describe the behaviour of the main parameters along generic [truncated] trajectories, with the help of transfer operators. After that, Section 5 returns to the discrete case and gcd algorithms themselves. We introduce our main tools, generating functions together with new extensions of transfer operators, which play the rôle of generating operators. We establish a fundamental relation between these two objects. Then, we perform average-case analysis [Section 6] and finally distributional analysis [Section 7]. Section 8 describes the general framework of functional analysis which is needed. Section 9 provides historical and bibliographic references, describes related works and states numerous open problems.

2. **Dynamical Systems.** As we already mention it in Section 1.6, the first step aims to find a continuous counterpart to the gcd algorithms, and dynamical systems [or iterated functions systems] will provide such continuous extensions. In the sequel of the paper, the data size plays an important rôle in generating functions, and we have to build dynamical systems for which transfer operators may be used for generating data sizes.

2.1. **Various dynamical systems.** We recall some facts about various kinds of dynamical systems, and we define our main notations.

Plain complete dynamical systems. Recall that a (plain) dynamical system is a pair formed by a compact set X and a mapping $V : X \to X$ for which there exists a (finite or denumerable) set $\mathcal{D}$, (whose elements are called digits), and a topological partition $\{X_d\}_{d\in\mathcal{D}}$ of the set X in subsets X_d such that the restriction of V to each element X_d of the partition is C^2 and invertible.

Here, we are led to so–called *complete* dynamical systems, where the restriction $V|_{X_d} : X_d \to X$ is surjective. A special rôle is played by the set $\mathcal{H}$ of branches of the inverse function V^{-1} of V that are also naturally numbered by the index set $\mathcal{D}$: we denote by $h_{[d]}$ the inverse of the restriction $V|_{X_d}$, so that X_d is exactly the image $h_{[d]}(X)$. Since V is a mapping $V : X \to X$, the mapping V can be iterated, and the study of dynamical systems aims describing iterations of mapping V. The set $\mathcal{H}^k$ is the set of the inverse branches of the iterate V^k; its elements are of the form $h_{[d_1]} \circ h_{[d_2]} \circ \cdots \circ h_{[d_k]}$ and are called the inverse branches of depth k. The set $\mathcal{H}^\star := \cup_k \mathcal{H}^k$ is the semi-group generated by $\mathcal{H}$.

Given an initial point x in X, the sequence $\mathcal{V}(x) := (x, Vx, V^2x, \ldots)$ of iterates of x under the action of V forms the trajectory of the initial point x. The map $\sigma : X \to \mathcal{D}$ whose restriction to X_d is constant and equal to d is useful to encode the trajectory $\mathcal{V}(x)$ with an (infinite) sequence of digits,

$$\mathcal{D}(x) := (d_1(x), d_2(x), \ldots, d_i(x), \ldots) \qquad \text{with} \quad d_i(x) = \sigma(V^{i-1}x). \tag{2.1}$$

Markovian dynamical systems. We have already observed that most algorithms of Type 2 are Markovian with two states. They will give rise to Markovian dynamical systems (with two states), which we now describe. In this case, X is the union of two sets $X_{<0>}$ and $X_{<1>}$; when one is in $X_{<j>}$, one is in state j. Each $X_{<j>}$ has a topological partition $\{X_{<j>,d}\}_{d\in\mathcal{D}_{<j>}}$, and, for $j = 0, 1$, there exists a mapping $V_{<j>} : X_{<j>} \to X$. Each set $\mathcal{D}_{<j>}$ of digits admits a partition into two subsets $\mathcal{D}_{<i|j>}$, and, for any $d \in \mathcal{D}_{<i|j>}$, the restriction $V_{<j>}|_{X_{<j>,d}}$ is a surjection onto $X_{<i>}$. The set of inverse branches of $V_{<j>}$ is denoted by $\mathcal{H}_{<j>}$, and there are four sets $\mathcal{H}_{<i|j>}$, each of them is the set of inverse branches of the restriction of $V_{<j>}$ to the union $\cup_{d\in\mathcal{D}_{<i|j>}} X_{<j>,d}$. See Figure 14 for instances of such a situation.

Random dynamical systems and iterated functions systems. Here, we mainly deal with random dynamical systems. Consider a compact set X, a sequence of mappings $V_{(1)}, V_{(2)}, \ldots, V_{(k)}, \ldots$, and a probability π defined on $\mathbb{N}^+$. At each step of the process, one chooses the dynamical system $(X, V_{(k)})$ with probability π_k. Now, the trajectory $\mathcal{V}(x)$ and the encoding $\mathcal{D}(x)$ become random variables, since they depend on the random choices that are made at each step of the process. Finally, at each step of the process, there are two different choices: first, one chooses the mapping $V_{(k)}$ according to probability π_k, then the position of x with respect to the topological partition of $V_{(k)}$ determines the branch of the mapping $V_{(k)}$ which will be used in the process.

A dynamical system with only one branch is not a priori very interesting, because it describes a process which always performs the same operation. However, a random dynamical system where each dynamical system $(X, V_{(k)})$ has only one branch may be interesting, since there remains one random choice in this process, namely the choice related to probability π: such a system is called a system of iterated functions [(IFS) in short].

Finally, we will use in the sequel three kinds of dynamical processes: Two kinds of processes, where there is only one source of randomness for choosing the operation of the algorithm (and thus the inverse branch used),

— (deterministic) dynamical systems (possibly markovian), where the randomness is only due to the position of $x \in X$ [governed by the most significant bits]. This situation arises for Types 0 and 1.

— or systems of iterated functions, where the randomness is only due to the probability π (governed by the least significant bits). This situation arises for Type 4.

The third kind of process, which is the random dynamical system (possibly markovian), contains two sources of randomness, due to the position $x \in X$ and probability π. This happens for mixed types [Types 2 and 3].

2.2. **Euclidean algorithms and dynamical systems.** The main (and simple) idea aims to relate each algorithm to a dynamical system (X, V), such that the execution of the algorithm on the input (u, v) is closely related to the trajectory $V(u/v)$. Then the set X must contain the set of valid "rationals", i.e., the set of all "rationals" u/v related to valid input pairs (u, v), the topology which endows X must be "compatible" with the main choices of the algorithm, and the set formed with the "rationals" of X must be dense in X for this topology. In this way, it is possible to get extensions for each LFT used for building continued fractions. And the set of the inverse branches of V must coincide with the set of such extensions of LFT's.

We recall that our final tools are generating functions, defined in Section 1.7, [bivariate as $S_R(s, w)$ or univariate as $S_R^{[k]}(s)$] for which we aim to obtain alternative expressions. Our first task is to generate, with these continuous extensions, the inputs, and more precisely the absolute value of the input sizes. Then, we would like the topology on X to be compatible with the notion of size.

For the first two types, the topology on X is clearly compatible with the notion of size: Type 0 [polynomial case] is totally ultrametric [both, topology on X and size are ultrametric], while Type 1 [MSB class] is totally archimedean. For these first two Types, we are then led to deterministic dynamical systems.

This compatibility no longer holds for the other types, and there is a conflict between the size and the topology: the size remains archimedean, whereas the topology is no more totally archimedean. The topology of Type 4 is a priori totally dyadic, and for mixed types [Types 2 and 3], the extension should be a double extension, both archimedean (for dealing with operations on most significant bits) and dyadic (for dealing with operations on least significant bits). For all these types [Types 2, 3, 4], we need a double extension, both archimedean and dyadic, where both input sizes and operations performed by the algorithms can be easily generated and extended. Finally, we shall decide to mainly work with real extensions, well–adapted for generating input sizes, and the dyadic topology only translates in a probabilistic way: we shall deal with random dynamical systems, where the partition of X is defined by the most significant bits, whereas probability π is related to the dyadic topology (and the least significant bits). Since Type 4 is not governed at all by the MSB's, there is only one branch for each dynamical system, and we are led to a system of iterated functions.

2.3. **The transfer operator.** The main study in dynamical systems concerns itself with the interplay between properties of the transformation V and properties of trajectories [or encoded trajectories defined in (2.1)] under iteration of the transformation. The behaviour of typical trajectories of dynamical systems is more easily explained by examining the flow of densities. In each case, there exists on X a Haar measure (normalized), and the set X is endowed with some initial distribution relative to some density $f = f_0$.

The time evolution governed by the map V modifies the density, and the successive densities $f_0, f_1, f_2, \ldots, f_n, \ldots$ describe the global evolution of the system at time $t = 0, 1, 2, \ldots$. For each inverse branch h, the component operator $\mathbf{H}_{[h]}$ defined as

$$\mathbf{H}_{[h]}[f](x) = |h'(x)| \cdot f \circ h(x). \tag{2.2}$$

expresses the part of the new density which is brought when one uses the branch h. (Here $|.|$ denotes the absolute value on X, i.e. the ultrametric absolute value [Type 0] or the archimedean absolute value [for all other types]). Then, if the dynamical system is generic, the operator

$$\mathbf{H} := \sum_{h \in \mathcal{H}} \mathbf{H}_{[h]} \tag{2.3}$$

is the density transformer, or the Perron-Frobenius operator which expresses the new density f_1 as a function of the old density f_0 via the relation $f_1 = \mathbf{H}[f_0]$.

It proves convenient to add a (complex) parameter s, and introduce the component operator

$$\mathbf{H}_{s,[h]}[f](x) = |h'(x)|^s \cdot f \circ h(x). \tag{2.4}$$

The transfer operator $\mathbf{H}_s$, defined as

$$\mathbf{H}_s := \sum_{h \in \mathcal{H}} \mathbf{H}_{s,[h]} \tag{2.5}$$

can be viewed as an extension of the density transformer, since the equality $\mathbf{H}_1 = \mathbf{H}$ holds. It admits the following general form

$$\mathbf{H}_s[f](x) := \sum_{h \in \mathcal{H}} |h'(x)|^s \cdot f \circ h(x). \tag{2.6}$$

For Markovian dynamical systems with two states 0 and 1, the set $\mathcal{H}_{<i>}$ denotes the set of inverse branches used in the state i, and the set $\mathcal{H}_{<i|j>}$ "brings" the system

from state j to state i. If $\mathbf{H}_{s,<i|j>}$ [resp $\mathbf{H}_{s,<i>}$], denotes the transfer operator associated to set $\mathcal{H}_{<i|j>}$ [resp. $\mathcal{H}_{<i>}$] the transfer operator $\mathbf{H}_s$ relative to this Markovian system is a "matrix operator",

$$\mathbf{H}_s = \begin{pmatrix} \mathbf{H}_{s,<0|0>} & \mathbf{H}_{s,<0|1>} \\ \mathbf{H}_{s,<1|0>} & \mathbf{H}_{s,<1|1>} \end{pmatrix} \qquad \text{with} \quad \mathbf{H}_{s,<i|j>} := \sum_{h\in\mathcal{H}_{<i|j>}} \mathbf{H}_{s,[h]}, \tag{2.7}$$

that operates on pairs $f = (f_{<0>}, f_{<1>})$ of functions. In the same vein as previously, the transfer operator is an extension of the density transformer $\mathbf{H}$, defined as equal to $\mathbf{H}_1$.

Until now, we have only defined the transfer operator in the case of deterministic dynamical systems. In the case of a random dynamical defined by a compact set X, a sequence of mappings $V_{(k)}$ and a probability π, the formula (2.3) extends to

$$\mathbf{H} := \sum_{k\geq 1} \pi_k \cdot \mathbf{H}_{(k)}$$

where $\mathbf{H}_{(k)}$ is the density transformer relative to the dynamical system $(X, V_{(k)})$. Then, for a random dynamical system, the transfer operator will be defined as

$$\mathbf{H}_s[f](x) := \sum_{k\geq 1} \pi_k^s \sum_{h\in\mathcal{H}_{(k)}} |h'(x)|^s \cdot f\circ h(x).$$

It is easy to get a formula of the same vein in the case of a random Markov dynamical system,

These transfer operators are a central tool for studying dynamical systems.

2.4. **Transfer operator and generation of inputs.** Dynamical analysis also uses the transfer operator; however, it uses it (in a non classical way) for generating absolute values of the inputs which appear in a central way in the generating functions [see Section 1.7]. Transfer operators can be viewed as "generating operators" since they generate themselves ... generating functions.

Remember that Section 1.4 explained why there are two main cases for defining absolute values of input pairs (u, v): case (a), relative to all types except Type 4, where it is defined in (1.10), and case (b), relative to Type 4, where it is defined in (1.12).

(a) For all types except Type 4, the absolute value $|(u, v)| = |v|$ of an input of Ω is closely related to the denominator function $D[h](x)$ taken at the final value $x := \eta$ [see (1.11)]. Since all the inverse branches are LFTs, the denominator function is itself related to the derivative, via the relation

$$\frac{1}{D[h](x)^{2s}} = \delta_h^s \cdot |h'(x)|^s \qquad \text{with} \quad \delta_h := \frac{1}{|\det h|},$$

so that the general term of the Dirichlet series $S_R(s, 0)$ defined in Equation 1.18 can be written as

$$\frac{1}{|(u, v)|^{2s}} = \delta_h^s \cdot |h'(\eta)|^s, \qquad \text{with} \quad \delta_h := \frac{1}{|\det h|}. \tag{2.8}$$

We are then led to deal with (possibly random) dynamical systems where the set of branches h with $|\det h| = D$ is chosen with probability $1/D$. Then, the component operator $\mathbf{H}_{s,[h]}$ of such a random dynamical system is

$$\mathbf{H}_{s,[h]}[f](x) = \delta_h^s \cdot |h'(x)|^s \cdot f\circ h(x) = |D[h](x)|^{-2s} \cdot f\circ h(x), \tag{2.9}$$

and the transfer operator $\mathbf{H}_s$ defined in (2.5) or in (2.7) can be used for generating input data.

(b) For Type 4, the absolute value $|(u,v)|$ of an input of Ω equals the Euclidean norm $||(u,v)||$, which coincides with the Euclidean norm of the vector $M(0,1)$ where M is the matrix built by the execution of the algorithm [remember that the final pair is (0, 1)]. [see Eqn (1.13)]. More generally, we aim to generate the ratios

$$\frac{||(a,b)||^2}{||M(a,b)||^2}. \tag{2.10}$$

The LFT g associated to matrix M defines a bijection of the projective line. Denote by θ an angle of the torus $J := \mathbb{R}/\pi\mathbb{Z}$ which can be identified with the interval $]-\pi/2, +\pi/2[$ (where the two points $-\pi/2$ and $+\pi/2$ are the same), and by Ψ the "tangent" map. Consider the map $\underline{g} : J \to J$ which is conjugate of g by Ψ and defined as $\underline{g} := \Psi^{-1} \circ g \circ \Psi$. For a vector (a,b) parallel to $(y,1)$ with $y = a/b = \tan\theta$, one has

$$\underline{g}(\theta) = \arctan[g(y)] = \arctan\left(\frac{\alpha y + \beta}{\gamma y + \delta}\right),$$

and the derivative $\underline{g}'(\theta)$ satisfies, with $D := |\det g|$,

$$\underline{g}'(\theta) = D \cdot \frac{1+y^2}{(\alpha y+\beta)^2 + (\gamma y+\delta)^2} = D \cdot \frac{||(a,b)||^2}{||M(a,b)||^2}.$$

Now, for an input (u,v) which equals $M(0,1)$, one has

$$\frac{1}{|(u,v)|^2} = \frac{||(0,1)||^2}{||M(0,1)||^2},$$

and finally

$$\frac{1}{|(u,v)|^{2s}} = \delta_g^s \cdot |\underline{g}'(0)|^s \qquad \text{with} \quad \delta_g := \frac{1}{|\det g|}. \tag{2.11}$$

We are then led to deal with a dynamical process where branches $h := \underline{g}$ are the conjugates of the LFT's g produced by the Euclidean algorithm with the tangent map. This will be a random dynamical process where each branch $h := \underline{g}$ relative to $|\det g| = D$ is chosen with probability $1/D$. Then, with $\delta_{\underline{g}} := \delta_g = 1/|\det g|$, the component operator $\mathbf{H}_{s,[h]}$ relative to such a random process is

$$\mathbf{H}_{s,[h]}[f](x) = \delta_h^s \cdot |h'(x)|^s \cdot f \circ h(x) \tag{2.12}$$

and the transfer operator $\mathbf{H}_s$ defined in (2.5) can be used for generating input data.

In both cases (a) and (b), the Dirichlet series $S_R(s,0)$ can be generated with the help of a transfer operator related to a [possibly] random system, as we see from Equations (2.8) or (2.11). However, there is an important difference between these two random processes. The first one [case (a), for all types except Type 4] is a (possibly random) dynamical system, while the second one [case (b), only for Type 4] is a system of iterated functions. In case (a), the set of LFT's h with $|\det h| = D$ is the set of inverse branches of a dynamical system, and this dynamical system is chosen with probability $1/D$. Then, if the set X is endowed with some density f, the set $h(X)$ is a proper subset of the set X, and the probability that an inverse branch h [of any depth] is chosen is

$$\mathbb{P}[h \text{ is chosen}] = \frac{1}{|\det h|} \int_{h(X)} f(u)du = \delta_h \int_{h(X)} f(u)du. \tag{2.13}$$

$(P1)$ *Uniform contraction of inverse branches.*

$$\rho := \left(\max\{|h'(x)|; h \in \mathcal{H}^n, x \in X\}\right)^{1/n} < 1.$$

$(P2)$ *Bounded distortion.*

$$\exists K > 0, \forall h \in \mathcal{H}, \forall x \in X, \quad |h''(x)| \leq K \cdot |h'(x)|.$$

$(P3)$ *Convergence on the left of* $s = 1$.

$$\exists \sigma_0 < 1, \forall \sigma > \sigma_0, \quad \sum_{h \in \mathcal{H}} \delta_h^\sigma \cdot \sup_{x \in X} |h'(x)|^\sigma < \infty$$

FIGURE 2. Good properties of the set $\mathcal{H}$ of inverse branches

In case (b), each LFT g with a given determinant D [or its conjugate $h := \underline{g}$] is chosen with probability $1/D$. Each function h is a bijection $h : X \to X$, with $X := J$, and finally, if the set X is endowed with some density f, the probability that a function h [of any depth] is chosen is

$$\mathbb{P}[h \text{ is chosen}] = \frac{1}{|\det h|} = \delta_h \int_{h(X)} f(u)du. \tag{2.14}$$

Then, the formulae (2.13) or (2.14) are of the same vein, and define a common framework of use.

2.5. **Three main classes.** It is well-known that the long time behaviour of trajectories is closely related to expansion properties of mapping V, or contracting properties of the set $\mathcal{H}$ of inverse branches of V. More precisely, we introduce three important properties of set $\mathcal{H}$ in Figure 2.

For dynamical systems of Types 3 and 4, the rôle of LSB's is too important to give rise to smooth archimedean dynamical systems. This explains why Property $(P1)$ does not hold. There are four algorithms of types 3 or 4 : the Binary Algorithm, the Plus-Minus Algorithm and the two versions of the LSB Algorithm. We do not succeed at all analyzing the Plus-Minus Algorithm: we have built a dynamical system, but we do not know how to deal with it. For the other three algorithms which form the so–called Difficult Class, the situation is different: we obtain some important results, even if we do not answer all the main questions concerning the analysis of these algorithms.

Dynamical Systems of Type 0, 1, and 2 are easier to analyze (at least a priori), and they form the Easy Class; however, the behaviour of the set $\mathcal{H}$ of inverse branches [with respect to Properties $(P1, P2, P3)$] depends on the algorithms. This gives rise to a partition of Types 0, 1, 2 into two main classes. The first class, which is called the Good Class gathers all the Euclidean Dynamical Systems which satisfy $(P1, P2, P3)$, and all the associated gcd algorithms of the Class will be "fast". The Second class, called the Bad Class, gathers the other algorithms, that are in fact almost contracting, since the only problem for $(P1)$ is due to the existence of an indifferent point....The gcd algorithms of the Bad Class will be "slow". The good dynamical systems give rise to chaotic behaviour of trajectories, while, for the slow ones, the trajectories present an intermittency phenomenon. [See Figure 3]

2.6. **The Good Class versus the Bad Class. Induced dynamical systems.** For dynamical systems of the Good Class, the density transformer $\mathbf{H}$ acts on $\mathcal{C}^1(X)$

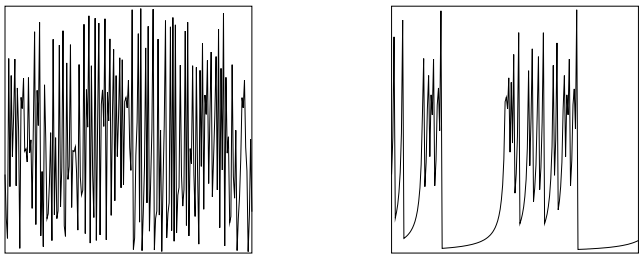

FIGURE 3. Chaotic Orbit [Good Class], Intermittent Orbit [Bad Class].

functions; furthermore, it has a unique dominant eigenvalue $\lambda = 1$, with an associated dominant eigenfunction ψ which is also an invariant function. Moreover, there is a spectral gap, and the rest of the spectrum lies in a disk of radius < 1. Furthermore, with the condition $\int_X \psi(t)dt = 1$, [where dt is the Haar measure on X], the function ψ is unique too.

Two constants will play a crucial rôle in the sequel. The entropy α of such a system is well–defined, with the help of Rohlin's formula which can be written here as

$$\alpha = -\sum_{h\in\mathcal{H}} \delta_h \log \delta_h - \int_X \log|V'(t)|\psi(t)dt. \tag{2.15}$$

The average $\mu(c)$ of a digit-cost c of moderate growth with respect to measure $\psi(t)dt$,

$$\mu(c) = \sum_{h\in\mathcal{H}} \delta_h \cdot c(h) \cdot \int_{h(X)} \psi(t)dt \tag{2.16}$$

plays also an important rôle in the paper.

The situation is quite different for the algorithms of the Slow Class: The invariant function ψ is singular at the point η which is the stopping value for the algorithm. The reason is that this point η is indifferent under the action of V: it is a fixed point for V with a derivative whose absolute value is equal to 1 (i. e. $V(\eta) = \eta, |V'(\eta)| = 1$). See Figure 10. Then, the typical trajectory admits a quite different form, since, when it arrives near this point, it passes many times near it. As this point η is (weakly) repulsive, the rational trajectories that eventually finish at this point η, will attain it after a long time. Then it is (at least intuitively) clear that the relative Euclidean algorithms will be "slow".

However, the induced dynamical system which "forgets" all the sub-trajectories that stay near the indifferent point η admits typical trajectories that again exhibit a chaotic behaviour. The notion of induction was introduced by Bowen [13]. Beginning with a dynamical system (X, V), the induced system $(\widetilde{X}, \widetilde{V})$ is defined in a more formal way as follows: If p denotes the "bad" inverse branch that contains the indifferent point η and $\mathcal{Q}$ denotes the set $\mathcal{H} \setminus p$ of good inverse branches, the interval $\widetilde{X}$ is defined as

$$\widetilde{X} := X \setminus p(X) = \bigcup_{h\in\mathcal{Q}} h(X).$$

The induced shift mapping $\widetilde{V} : X \to X$ is then defined from the first iterate of V that returns into $\widetilde{X}$: if $n(x)$ is the smallest integer $k \geq 0$ such that $V^k(x) \in \widetilde{X}$, then we let $\widetilde{V}(x) := V^{n(x)+1}(x)$. This means that $\widetilde{V}(x)$ equals $V(x)$ for $x \in \widetilde{X}$ while, for

$x \notin \widetilde{X}$, $\widetilde{V}(x)$ equals $V(y)$ where y is the first iterate of x that belongs to $\widetilde{X}$. Then the trajectory $(x, \widetilde{V}x, \widetilde{V}^2x, \ldots)$ is exactly the trajectory $(x, Vx, V^2x, \ldots)$ which forgets all the sub-trajectories that stay near the indifferent point η, whereas the set $\widetilde{\mathcal{H}}$ of inverse branches is exactly the set $\widetilde{\mathcal{H}} = p^\star\mathcal{Q}$ where one groups a sequence of bad LFTs with a good one. Now, the following result will be very important for the sequel:
For any algorithm of the Slow class, the set $\widetilde{\mathcal{H}}$ satisfies properties $(P1, P2, P3)$ of Figure 2, and the induced system is good.

Consider the operators $\mathbf{Q}_s, \mathbf{P}_s$ relative to the sets $\{p\}, \mathcal{Q}$,

$$\mathbf{Q}_s := \sum_{h\in\mathcal{Q}} \mathbf{H}_{s,[h]}, \qquad \mathbf{P}_s := \mathbf{H}_{s,[p]}$$

which satisfy $\mathbf{P}_s + \mathbf{Q}_s = \mathbf{H}_s$, and consider, as in [82], the transfer operator $\widetilde{\mathbf{H}}_s$ of this induced dynamical system. It involves the operators $\mathbf{Q}_s, \mathbf{P}_s$ under the form

$$\widetilde{\mathbf{H}}_s = \sum_{k\geq 0} \mathbf{Q}_s \circ \mathbf{P}_s^k = \mathbf{Q}_s \circ (I - \mathbf{P}_s)^{-1}, \qquad \text{so that} \qquad \widetilde{\mathbf{H}}_s \circ (I - \mathbf{P}_s) = \mathbf{Q}_s. \quad (2.17)$$

Since the operator $\mathbf{H}$ admits an invariant density ψ which satisfies

$$\mathbf{H}[\psi] = \mathbf{P}[\psi] + \mathbf{Q}[\psi] = \psi,$$

the function $g := \mathbf{Q}[\psi] = (I - \mathbf{P})[\psi]$ satisfies

$$\widetilde{\mathbf{H}}[g] = \widetilde{\mathbf{H}} \circ (I - \mathbf{P})[\psi] = \mathbf{Q}[\psi] = g,$$

so that $(I - \mathbf{P})[\psi]$ is an invariant function for $\widetilde{\mathbf{H}}$. Furthermore, the semigroup $\mathcal{H}^\star$ can be written as $\widetilde{\mathcal{H}}^\star \cdot p^\star$. However, the rational trajectories stop as soon as they reach the stopping value η. Since η is a fixed point for the bad branch p, the rational trajectories only use the LFT's which do not finish with p, and they use exactly the semi-group $\widetilde{\mathcal{H}}^\star$. Consequently, we may replace in our study the quasi-inverse $(I - \mathbf{H}_s)^{-1}[1][\eta)$ by its induced form $(I - \widetilde{\mathbf{H}}_s)^{-1}[1][\eta)$, and this helps a lot since the tilde operator, relative to set $\widetilde{\mathcal{H}}$ has now good properties.

2.7. **Transfer operators viewed as generating operators.** Finally, in each case, the transfer operator is written as a sum of operators $\mathbf{H}_{s,[h]}$ taken over some set, $\mathcal{H}$ in (2.5), sets $\mathcal{H}_{i|j}$ for (2.7). In each case, the n-th iterate of the operator has exactly the same expression as the operator itself, except that the sum is now taken over the n–th power of the initial set, namely $\mathcal{H}^n$ in (2.5), sets $\mathcal{H}_{i|j}^{[n]}$ for (2.7) [which bring data from state i to state j in n steps]. This fact is due to the multiplicative properties of derivatives, denominators, matrices, probabilities, determinants.... In a very general sense, the n-th iterate of the transfer operator describes the data after n iterations. Then, the evolution of data during all the possible executions of the algorithm, correspond to the semi–group $\mathcal{H}^\star$ for generic algorithms. For Markovian algorithms, it is related to the four sets $\mathcal{H}_{i|j}^{[\star]}$ [which bring data from state i to state j in an arbitrary (but finite) number of steps]. We are led to work with the quasi-inverse of the transfer operators. Each of these quasi–inverses generates all the possible iterations, and, in a quite general framework, the operator

$$(I - \mathbf{H}_s)^{-1}[1](\eta) \qquad [\eta \text{ is the final value of the algorithm}]$$

will generate all the input sizes. This is why these quasi-inverse will play a central rôle in our study.

In the following, we use extensions of the various transfer operators introduced in (2.5, 2.7) [see Figure 4]. When we wish to study a parameter R relative to

	Name	Definition of component operator
Plain	$\mathbf{H}_s$	$\delta_h^s \cdot \lvert h'(x)\rvert^s \cdot f \circ h(x)$
	$\widehat{\mathbf{H}}_s$	$\delta_h^s \cdot \lvert h'(x)\rvert^s \cdot F(h(x), y)$
	$\underline{\mathbf{H}}_s$	$\delta_h^s \cdot \lvert h'(x)\rvert^s \cdot F(h(x), h(y))$
Weighted	$\mathbf{H}_{s,w,(c)}$	$\delta_h^s \cdot \exp[wc(h)] \cdot \lvert h'(x)\rvert^s \cdot f \circ h(x)$
	$\widehat{\mathbf{H}}_{s,w,(c)}$	$\delta_h^s \cdot \exp[wc(h)] \cdot \lvert h'(x)\rvert^s \cdot F(h(x), y)$
	$\underline{\mathbf{H}}_{s,w}$	$\delta_h^{s-w} \cdot \lvert h'(x)\rvert^s \cdot \lvert h'(y)\rvert^{-w} \cdot F(h(x), h(y))$
Various Derivatives	$\mathbf{H}_s^{(c)} := (\partial/\partial w)\, \mathbf{H}_{s,w}\lvert_{w=0}$	$\delta_h^s \cdot c(h) \cdot \lvert h'(x)\rvert^s \cdot f \circ h(x)$
	$\widehat{\mathbf{H}}_s^{(c)} := (\partial/\partial w)\, \widehat{\mathbf{H}}_{s,w}\lvert_{w=0}$	$\delta_h^s \cdot c(h) \cdot \lvert h'(x)\rvert^s \cdot f \circ h(x)$
	$\Delta\mathbf{H}_s := (d/ds)\mathbf{H}_s$	$\delta_h^s \cdot (\log \lvert h'(x)\rvert + \log \delta_h) \cdot \lvert h'(x)\rvert^s \cdot f \circ h(x)$
	$\Delta\underline{\mathbf{H}}_s := (\partial/\partial w)\, \underline{\mathbf{H}}_{s,w}\lvert_{w=0}$	$-\delta_h^s \cdot (\log \delta_h + \log \lvert h'(y)\rvert) \cdot \lvert h'(x)\rvert^s \cdot F(h(x), h(y))$

FIGURE 4. Definition of operators via their component operators. [remark that f denotes a function of one variable, and F a function of two variables]

	Total costs	Beginning continuants	Ending continuants	Bit-Complexity
Real Trajectories (Section 4)	$\mathbf{H}_{1,w,(c)}$	$\underline{\mathbf{H}}_{1,w}$	————	————
Rational Trajectories (average) (Sections 5 and 6)	$\mathbf{H}_s$, $\mathbf{H}_s^{(c)}$	$\widehat{\mathbf{H}}_s, \underline{\mathbf{H}}_{s,0}$	$\mathbf{H}_s$	$\mathbf{H}_s$, $\mathbf{H}_s^{(\ell)}$ $\widehat{\mathbf{H}}_s^{(\ell)}, \underline{\mathbf{H}}_{s,0}$
Rational Trajectories (distribution) (Section 7)	$\mathbf{H}_{s,w,(c)}$	$\widehat{\mathbf{H}}_s, \underline{\mathbf{H}}_{s,w}$	$\mathbf{H}_{s-w}, \mathbf{H}_s$	————

FIGURE 5. Main transfer operators used in the analyses.

a Euclidean Algorithm, we add an extra parameter [called w] inside a transfer operator $\mathbf{H}_s$ in order to obtain a weighted transfer operator of the form $\mathbf{H}_{s,w,(c)}$ [which operates on functions of one variable]. For continuants, we will also deal with an underlined version $\underline{\mathbf{H}}_{s,w}$ and, also sometimes, with an "hat" version $\widehat{\mathbf{H}}_{s,w}$ of the transfer operator. Note that these last two versions of the transfer operator operate on functions of two variables, even if the "hat" version does not modify the second variable y.

2.8. **The nine Theorems.** In this paper, we state nine Theorems, numbered from 1 to 9 [cf Figure 6]. The first two Theorems [Theorems 1 and 2], stated in Section 4, describe the behaviour of generic truncated trajectories, from the digit–point of view [Theorem 1] or from the continuant point of view [Theorem 2]. Then Sections 5 and 6 perform the average–case analysis of Euclidean Algorithms, and then study the average behaviour of the particular rational trajectories. Here, four Theorems are stated [Theorems 3, 4, 5, 6]. Theorems 3 and 4 are relative to the average–case analysis of additive costs, whereas Theorem 5 provides an average case analysis of the continuants [ending or beginning] at a given fraction of the depth. Theorem 6 is devoted to the average-case analysis of the bit–complexity cost. Finally, Section 7 performs distributional analysis of additive costs [Theorem 7] and continuants [Theorem 8], and finally states results on the distributional analysis of the bit–complexity cost [Theorem 9].

Figure 6 summarizes the subject of the nine theorems. Figure 7 describes the constants which intervene in the dominant terms of the various expectations and

	Additive costs	Beginning continuants	Ending continuants	Bit-Complexity
Real Trajectories (Section 4)	Thm 1	Thm 2	————-	———
Rational Trajectories (average) (Sections 5 and 6)	Thms 3 and 4	Thm 5	Thm 5	Thm 6
Rational Trajectories (distribution) (Section 7)	Thm 7	Thm 8	———	Thm 9

FIGURE 6. The subject of the nine theorems

Constants for the expectations

$$\mu(c) = \Lambda'_w(1,0), \qquad \alpha = |\Lambda'_s(1,0)|.$$

Alternative forms for these constants are given in (2.15) and (2.16).

Theorem 1	$\mu(c)$
Theorem 2	$\alpha/2$
Theorem 3 and 7	$\widehat{\mu} := (2\log 2)/\alpha$
Theorem 4, 6, 7, 9	$\widehat{\mu}(c) = \widehat{\mu} \cdot \mu(c)$
Theorem 5 and 8	δ

Constants for the variance

$$\gamma := \Lambda''_{s^2}(1,0), \qquad \rho(c) = \Lambda''_{w^2}(1,0), \qquad \chi(c) = \Lambda''_{sw}(1,0).$$

No alternative form is known for these constants.

Theorem 1	$\rho(c)$
Theorem 2	$\gamma/4$
Theorem 7	$\widehat{\rho} := 2\log 2 \cdot (\gamma/\alpha^3)$
Theorems 7 and 9	$\widehat{\rho}(c) := \mu^2(c) \cdot \widehat{\rho}^2 + \widehat{\mu} \cdot \rho^2(c) + \widehat{\mu}^2 \mu(c) \cdot \chi(c)$
Theorem 8	$\delta(1-\delta) \cdot (\gamma/\alpha)$

FIGURE 7. Constants in the dominant terms of the expectation and variance. $\Lambda(s,w)$ is the pressure, i.e., $\Lambda(s,w) := \log \lambda(s,w)$, where $\lambda(s,w)$ is the dominant eigenvalue of the transfer operator $\mathbf{H}_{s,w}$

variances studied in these theorems. It exhibits the crucial rôle played by the pressure function $\Lambda(s,w)$ [which is the logarithm of the dominant eigenvalue $\lambda(s,w)$ of the transfer operator $\mathbf{H}_{s,w}$], since all the constants of the paper involve its first five derivatives (two derivatives of order 1 and three derivatives of order two).

UDE	Unique Dominant Eigenvalue at $(1,0)$
SG	Spectral Gap at $(1,0)$
$An_w(1,0)$	The map $w \mapsto \mathbf{H}_{1,w}$ is analytic at $w=0$
$An_s(s,0)$	The map $s \mapsto \mathbf{H}_{s,0}$ is analytic for $\Re s > \sigma_0$ with $\sigma_0 < 1$
$An_{s,w}(s,0)$	The map $(s,w) \mapsto \mathbf{H}_{s,w}$ is analytic for $\Re s > \sigma_0$ and w near 0, with $\sigma_0 < 1$
SM	Strict Maximum on the Vertical Line attained at $s=\sigma$
	$\forall \sigma(\text{real}), \forall t \neq 0, R(\sigma+it,0) < R(\sigma,0)$
US	Uniform Estimates on Vertical Strips :
	$\forall \xi > 0, \exists M > 0,\ t_0 > 0,$ such that $\forall n, \forall(\sigma,\nu)(\text{real})$ near $(1,0)$
	$\forall s = \sigma + it, w = \nu + i\tau, |t| > t_0, \quad ||\mathbf{H}^n_{s,w}||_{1,t} \leq M \cdot \gamma^n \cdot |t|^\xi$
SLC_w	Strict log convexity of $w \mapsto \log \lambda(1,w)$ at 0
SLC	Strict log convexity of $w \mapsto \log \lambda(1+qw, rw)$ at 0 for any $(q,r) \neq (0,0)$

FIGURE 8. Main analytical properties of the operator $\mathbf{H}_{s,w}$, its dominant eigenvalue $\lambda(s,w)$ and its spectral radius $R(s,w)$ (on a convenient functional space). Note that $||.||_{1,t}$ is some norm (depending on the imaginary part t of s) described in Section 7.

	Average-case	Distribution
Truncated	*UDE* + *SG*	*UDE* + *SG*
real trajectories		+ $An_w(1,0)$ + SLC_w
Rational	*UDE* + *SG*	*UDE* + *SG*
trajectories	+ $An_s(1,0)$ + *SM*	+ $An_{s,w}(1,0)$ + *US* + *SLC*

FIGURE 9. Properties of the transfer operator useful for analyzing trajectories.

Sections 4, 5, 6, 7 will describe, for each analysis, the convenient operator: it depends on the parameter to be studied, but also on the whole framework: continuous or discrete, average-case analysis or distributional analysis. Figures 4 and 5 summarize the operators used and the context of their use.

Figures 8 and 9 describe the main properties of the operator which are needed for the various analyses performed in the paper. Section 8, and particularly Figure 21 of Section 8, will provide the convenient functional spaces $\mathcal{F}$ [depending on the algorithms] where such properties will be proven to hold.

3. **First Step: Building the Euclidean dynamical Systems.** We are now ready for performing the first step of dynamical analysis: we aim to build, for each Euclidean Algorithm, a dynamical system which extends the operations of the algorithms, and whose transfer operator can be used as a generating operator. We describe here, in this Section, the characteristics of the dynamical system relative to each Euclidean Algorithm. Even if these systems share important properties –for instance, all their branches are LFT's–, their main characteristics may vary in a significant way. The class of Euclidean systems contains in fact quite different systems, whose analysis does not seem [a priori] to be performed in a common framework.

3.1. **The Euclidean dynamical systems for Type 0.** As we already mentioned it in Section 1.1, there are two divisions on polynomials; the first one is directed

by leading monomials (i.e., monomials of highest degree) and deals with decreasing degrees. The second one is directed by monomials of lowest degree and deals with increasing valuations.

The first gcd algorithm on polynomials with coefficients in the finite field $\mathbb{F}_q$ with q elements is based on the Eucldean division (with decreasing degrees) on a pair (u, v) of polynomials with $\deg v > \deg u$,

$$v = m \cdot u + r, \qquad \text{with } r = 0 \text{ or } \deg r < \deg u.$$

The analogue of the ring $\mathbb{Z}$ is the ring $\mathbb{F}_q[Z]$, and the field $\mathbb{F}_q(Z)$ plays the same rôle as the field $\mathbb{Q}$ of rational numbers. We work on the completion of $\mathbb{F}_q[Z]$ with respect to the (ultrametric) absolute value $|.|$ defined as $|u| := q^{\deg u}$: this is the field of Laurent formal power series $\mathbb{F}_q((1/Z))$ where each element f has a Hensel expansion

$$f = \sum_{n \geq n_0} f_n \, (1/Z)^n, \quad \text{with } f_n \in \mathbb{F}_q \text{ and } n_0 \in \mathbb{Z}. \tag{3.1}$$

This expansion is parallel to the binary expansion of a real [replace Z by 2]. From the Hensel expansion (3.1), it is possible to define the function integer part, denoted by $\lfloor . \rfloor$, and the function fractional part, denoted with $\{.\}$, with

$$\lfloor f \rfloor := \sum_{n=n_0}^{0} f_n \, (1/Z)^n \qquad \{f\} := \sum_{n \geq 1} f_n \, (1/Z)^n.$$

The convenient set X is the unit open ball $\mathcal{X}_q$ of $\mathbb{F}_q((1/Z))$, which is also the set of elements with zero integer part, and the shift $V : \mathcal{X}_q \to \mathcal{X}_q$ is defined by

$$V(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor = \left\{ \frac{1}{x} \right\}. \qquad \text{for } x \neq 0, \qquad V(0) = 0.$$

The set $\mathcal{D}$ of digits is

$$\mathcal{D} := \{m \in \mathbb{F}_q[Z]; \ \ |m| > 1\} = \{m \in \mathbb{F}_q[Z]; \ \ \deg m > 0\}.$$

This dynamical systems is precisely described for instance in [11].

The second gcd algorithm on polynomials is based on the division (with increasing valuations) on a pair (u, v) of polynomials with $\operatorname{val} u > \operatorname{val} v = 0$. If a denotes the valuation of u, the division step can be written as

$$u := Z^a \cdot u', \qquad v = m \cdot u + r, \qquad \text{with } r = 0 \text{ or } \quad \operatorname{val} r > \operatorname{val} u, \qquad r := Z^a \cdot r',$$

and the new pair is (u', r'). This division on (u, v) gives rise exactly to the Euclidean division on the mirror pair $(\overline{v}, \overline{u})$ defined by

$$\overline{v}(Z) := Z^n \cdot v(\frac{1}{Z}), \qquad \overline{u}(Z) := Z^n \cdot u(\frac{1}{Z}) \qquad \text{with} \quad n := \max(\deg v, \deg u)$$

These two divisions are "conjugated" via the mirror application, and their associated dynamical systems have exactly the same properties. We shall see that the situation is quite different for integers.

The Euclidean dynamical system possesses, in the polynomial case, very particular properties. The density transformer, and its extension the transfer operator, mainly deals with the absolute values $|h'(x)|$ of derivatives of inverse branches h, for $x \in \mathcal{X}_q$. And, in the polynomial case, thanks to the ultrametric topology on $\mathcal{X}_q$, this absolute value $|h'_{[m]}(x)|$ is constant on $\mathcal{X}_q$ and equals to $|m|^{-2}$. Then, when

applied to uniform density $f_0 = 1$, the transfer operator $\mathbf{H}_s$ transforms it into a constant function

$$\mathbf{H}_s[1] = \sum_{m\in\mathcal{D}} \frac{1}{|m|^{2s}} = \sum_{m\in\mathcal{D}} \frac{1}{q^{2s\deg m}}.$$

This Dirichlet series is in fact a power series in $z = q^{-2s}$. It coincides with the (usual) generating function $D(z)$ of the set $\mathcal{D}$, which gathers all the non constant polynomials of $\mathbb{F}_q(Z)$,

$$\mathbf{H}_s[1] = D(z) = \sum_{m\in\mathcal{D}} z^{\deg m} = (q-1)\sum_{n\geq 1} q^n z^n = \frac{q(q-1)z}{1-qz} = \frac{q-1}{q^{2s-1}-1}.$$

The entropy of the dynamical system is equal to $1-(1/q)$.

Of course, it is easier to deal with power series than with Dirichlet series. This is why the polynomial case is easier than the number case [as it is usual in the mathematical life, since there are no carries for polynomials ...].

3.2. **The Euclidean dynamical systems for Type 1.** The main features of MSB-gcd's are described in Figure 1. All the continuous extensions of the MSB-gcd's lead to dynamical systems where X is a real compact interval endowed with the usual absolute value [see Figure 10]. For surveys on dynamical systems on a real interval, see [17], [22].

For Type 1, there are many different possible generalizations for the integer part function which can be defined from the binary expansion of a real number. Such possible generalizations are described in Figure 3. All the mappings V [except for the Subtractive algorithm] are of the form

$$V(x) := \frac{1}{x} - A\left(\frac{1}{x}\right),$$

for some function A [See Figure 11], which generalizes the integer part function. The Subtractive Dynamical system is defined on the unit interval X; it has two branches

$$V(x) = \frac{x}{1-x}, \qquad \text{or} \quad V(x) = \frac{1-x}{x},$$

according as $(1-x)/x$ belongs to X or not, and its two inverse branches are

$$p(x) = \frac{1}{1+x}, \qquad q(x) = \frac{x}{1+x}. \tag{3.2}$$

See Figure 12 for a description of Subtractive Algorithm.

Transfer operators relative to some algorithms of Type 1 are very classical. For instance, the transfer operator associated to the Standard Euclidean Algorithm was introduced by Gauss himself; it can be written as

$$\mathbf{H}_s[f](x) := \sum_{m\geq 1} \left(\frac{1}{m+x}\right)^{2s} \cdot f\left(\frac{1}{m+x}\right)$$

For the subtractive algorithm, one has:

$$\mathbf{H}_s[f](x) := \left(\frac{1}{1+x}\right)^{2s} \cdot \left[f\left(\frac{1}{1+x}\right) + f\left(\frac{x}{1+x}\right)\right]$$

There are three algorithms of Type 1 [the Classical Algorithm (G), the Centered Algorithm (K), and the Odd Algorithm (O)] which belong to the Good Class (they correspond to the left column of Figure 10), and three algorithms of Type 1 [the

Even Algorithm (E), the By-Excess Algorithm (M), and the Subtractive Algorithm (T)] which belong to the Bad Class (they correspond to the right column of Figure 10). As we already explained in Section 2.6, in the case of algorithms of the Bad Class, we consider the induced dynamical systems. Figure 13 describes the main properties of the induced Dynamical systems $(\widetilde{M}), (\widetilde{E}), (\widetilde{T})$. Note that system $(\widetilde{T})$ is exactly the Classical Euclidean System (G).

For all these systems, the invariant density is explicit, and given in Figures 11 and 12. Then, with Rohlin's Formula [see Equation (2.15)], the entropy is also explicit (when it exists), and this is the same for the entropy of the induced dynamical systems [cf Figures 11,12,13].

3.3. **The Euclidean dynamical systems for Types 2 and 3: generalities.** How to obtain extensions for gcd's of the mixed groups? In fact, these gcd algorithms work partly with LSB's, but the size remains archimedean. Then, (as we already announced it), we choose to mainly work with real extensions, where the influence of LSB's leads to a probabilistic framework.

We consider first the behaviour of a step of the gcd algorithm when the total binary shift $a + b$ [see Equation (1.1)] equals k, and we extend it to real numbers of X: this defines a dynamical systems $(X, V_{(k)})$, where all the reals behave as if they use a total shift equal to k. In this case, the set of the inverse branches of $(X, V_{(k)})$ is denoted by $\mathcal{H}_{(k)}$, and the absolute value of the determinant of any LFT of $\mathcal{H}_{(k)}$ equals 2^k. For a further extension, we need to extend the total shift on real numbers of X. We have already explained (in Section 2.4) why it is convenient to extend it in a probabilistic way, into a random variable on X, according to the law $\mathbb{P}[k = d] = 2^{-d}$ (for $d \geq 1$).

Finally, the whole process gives rise to a probabilistic dynamical system $(X, \breve{V})$ defined as follows. At each time t, one chooses first the random binary shift k according to the previous law $\mathbb{P}[k = d] = 2^{-d}$ (for $d \geq 1$), then we choose $V_{(k)}$ as the mapping for this step, and, for any $x \in X$, the next point is $\breve{V}(x) := V_{(k)}(x)$. The set $\breve{\mathcal{H}}$ of all possible inverse branches is then the (disjoint) union of sets $\mathcal{H}_{(k)}$ for $k \geq 1$. In such a way, we extend a deterministic gcd algorithm into a random algorithm, and we can define (random) continued fraction for real numbers. Remark that now rational numbers themselves have random continued fraction expansions, and, amongst all these expansions, one can find the finite continued fraction of the rational.

Now, we make precise this probabilistic extension for each type [first Type 2, then Type 3].

3.4. **The Euclidean dynamical systems for Type 2.** [106] [107]. Consider four Euclidean divisions of Type 1[2], namely the standard division (G), the centered division (K), the Odd Division (O) and the By Excess division (M). Each of them can be slightly modified in the following way: the Euclidean division is performed as usual, and, after the division [or, more exactly, before the following division], the powers of two are removed from the remainder, in order to obtain an odd value for

[2] for the remaining two divisions, the situation is different, since the pseudo-version of the Subtractive Algorithm will be the Binary Algorithm, which belongs to Type 3, and the pseudo-version of the Even division coincides with the Even Division.

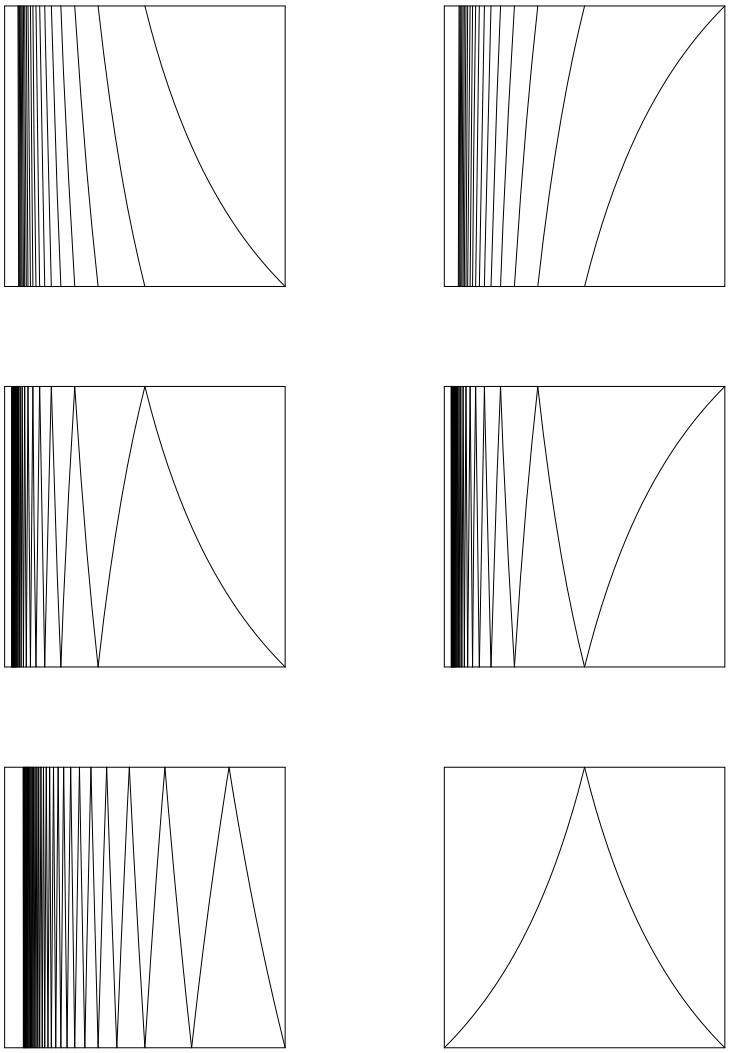

FIGURE 10. The six Euclidean Dynamical systems of the MSB Group [Type 1] ; on the left, Standard (G), Odd (O), Centered(K); on the right, By-Excess (M), Even (E), Subtractive (T). Note the presence of an indifferent point for each system on the right column: $\eta = 1$ for (M) and (E) – $\eta = 0$ for (T).

u'. Each pseudo-division step can be written as

$$u := 2^a \cdot u', \qquad v = m \cdot u' + \epsilon r, \qquad r' := r,$$

and the LFT h relative to this division is

$$h(x) := \frac{2^a}{m + \epsilon x}.$$

Alg.	*Function $A(x)$*	*Invariant function ψ*	*Entropy*	*Bad LFT p*	*Class.*
(G)	Integer part of x	$\frac{1}{\log 2}\frac{1}{1+x}$	$\frac{\pi^2}{6\log 2}$	——-	Good
(M)	Smallest integer at least equal to x	$\frac{1}{1-x}$	Not defined.	$\frac{1}{2-x}$	Bad
(K)	Nearest integer to x	$\frac{1}{\log\phi}[\frac{1}{\phi+x}+\frac{1}{\phi^2-x}]$.	$\frac{\pi^2}{6\log\phi}$	——-	Good
(E)	Even integer nearest to x	$\frac{1}{1-x}+\frac{1}{1+x}$	Not defined	$\frac{1}{2-x}$	Bad
(O)	Odd integer nearest to x	$\frac{1}{\phi-1+x}+\frac{1}{\phi^2-x}$	$\frac{\pi^2}{9\log\phi}$	——-	Good

FIGURE 11. The first five Euclidean dynamical systems [Type 1]. Here, the shift mapping V and the encoding mapping σ are defined from function A, with the formula $V(x) := |\frac{1}{x} - A(\frac{1}{x})|$, $\sigma(x) := A(\frac{1}{x})$.

Alg.	*Function $V(x)$*	ψ	*Entropy*	*Bad LFT p*	*Class.*
(T)	$V(x) := \begin{cases} \frac{x}{1-x} & \text{for } 0 \le x \le 1/2; \\ \frac{1-x}{x} & \text{for } 1/2 \le x \le 1 \end{cases}$	$\frac{1}{x}$	Not defined	$\frac{x}{1+x}$	Bad.

FIGURE 12. The dynamical system relative to the Subtractive Algorithm.

Induced Alg.	*Invariant function $\widetilde{\psi}$*	*Entropy $h(\widetilde{\mathcal{H}})$*
$(\widetilde{M})$	$\frac{1}{\log 2}\frac{1}{2-x}$	$\frac{\pi^2}{3\log 2}$
$(\widetilde{E})$	$\frac{1}{\log 3}[\frac{1}{3-x}+\frac{1}{1+x}]$	$\frac{\pi^2}{2\log 3}$
$(\widetilde{T})$= (G)	$\frac{1}{\log 2}\frac{1}{1+x}$	$\frac{\pi^2}{6\log 2}$

FIGURE 13. The induced dynamical systems relative to the three "bad" systems of Type 1.

The pseudo-Euclidean Algorithms are described in Figure 15. We recall that pseudo-Euclidean algorithms deal with odd inputs, and create a binary shift b equal to 0 [see Section 1]. Then, they are of Markovian type, since there is a relation between parity of the quotient m_i at step i and the exponent a_{i+1} at the beginning of the $i+1$-th step: if m_i is odd, then the remainder is even, and thus a_{i+1} satisfies $a_{i+1} \ge 1$; if m_i is even, then the remainder is odd, and thus a_{i+1} equals 0. We then consider two states: the 0–state, which means "the previous quotient of (v,u) is even" (or equivalently the previous remainder is odd), i.e., the present shift a equals 0; the 1–state, which means "the previous quotient of (v,u) is odd" (or equivalently the previous remainder is even), i.e., the present shift a satisfies $a \ge 1$. Then, the processus uses four different sets $\mathcal{H}_{<i|j>}$, where $\mathcal{H}_{<i|j>}$ brings rationals

Alg., X, η	*Division*	*Set of LFT's*	*Conditions*
($\breve{O}$) $[0,\,1]$, 0	$v = mu + \epsilon 2^k\, s$ m odd, $\epsilon = \pm 1$, s odd $k \geq 1, 0 \leq 2^k s < u$	$\breve{\mathcal{O}}_{(k)} = \{\dfrac{2^k}{m+\epsilon x}, \epsilon = \pm 1, m \text{ odd},$ $(m, \epsilon) \geq (2^k, +1)\}$	$\mathcal{J} = \mathcal{O}$
($\breve{G}$) $[0,\,1]$, 0	$v = mu + 2^k s$ $s = 0$ or s odd, $k \geq 0$ $0 \leq 2^k s < u$	$\breve{\mathcal{G}}_{<0>} = \mathcal{G},\ \breve{\mathcal{G}}_{<1>} = \bigcup_{k\geq 1} \breve{\mathcal{G}}_{<1>,(k)}$ $\breve{\mathcal{G}}_{<1>,(k)} = \{\dfrac{2^k}{m+x}, m \geq 2^k\}$ $\breve{\mathcal{G}}_{<i\|j>} = \breve{\mathcal{G}}_{<j>} \cap \{m \equiv i \bmod 2\}$	$\mathcal{I} = \breve{\mathcal{G}}_{<0>}$ $\mathcal{F} = \breve{\mathcal{G}}_{<1>}$
($\breve{M}$) $[0,\,1]$, 1	$v = mu - 2^k s$ s odd, $k \geq 0$ $0 \leq 2^k s < u$	$\breve{\mathcal{M}}_{<0>} = \mathcal{M},\ \breve{\mathcal{M}}_{<1>} = \bigcup_{k\geq 1} \breve{\mathcal{M}}_{<1>,(k)}$ $\breve{\mathcal{M}}_{<1>,(k)} = \{\dfrac{2^k}{m-x}, m > 2^k\}$ $\breve{\mathcal{M}}_{<i\|j>} = \breve{\mathcal{M}}_{<j>} \cap \{m \equiv i \bmod 2\}$	$\mathcal{I} = \breve{\mathcal{M}}_{<0>}$ $\mathcal{F} = \breve{\mathcal{M}}_{<1>}$
($\breve{K}$) $[0, 1/2]$, 0	$v = mu + \epsilon 2^k\, s$ $s = 0$ or s odd, $k \geq 0$ $0 \leq 2^k s < \frac{u}{2}$	$\breve{\mathcal{K}}_{<0>} = \mathcal{K},\ \breve{\mathcal{K}}_{<1>} = \bigcup_{k\geq 1} \breve{\mathcal{K}}_{<1>,(k)}$ $\breve{\mathcal{K}}_{<1>,(k)} = \{\dfrac{2^k}{m+\epsilon x},$ $\epsilon = \pm 1, (m, \epsilon) \geq (2^{k+1}, +1)\}$ $\breve{\mathcal{K}}_{<i\|j>} = \breve{\mathcal{K}}_{<j>} \cap \{m \equiv i \bmod 2\}$	$\mathcal{I} = \breve{\mathcal{K}}_{<0>}$ $\mathcal{F} = \breve{\mathcal{K}}_{<1>}$

FIGURE 14. The four pseudo-Euclidean algorithms.

from state i to state j. The initial state is always the 0–state and the final state is always the 1–state.
In State 1, the mapping of a pseudo–Euclidean Algorithm is defined from the mapping V of the related Euclidean Algorithm with $V_{(k)}(x) := V(x/2^k)$, so that the set $\mathcal{H}_{(k)}$ of the inverse branches of $V_{(k)}$ is

$$\mathcal{H}_{(k)} = \{g; \quad g(x) := 2^k h(x) \quad \text{with } h \in \mathcal{H} \text{ such that } \ 2^k h(X) \subset X\ \},$$

and its elements are of determinant 2^k.

There are three algorithms of Type 2 which belong to the Good Class, and these algorithms are the pseudo-versions of the "good" algorithms of Type 1, namely $(\breve{G}), (\breve{K}), (\breve{O})$. On the otherside, the algorithm $(\breve{M})$ belongs to the Bad Class. Remind that the pseudo-version of the Even Algorithm coincides with its plain version [because all the remainders are always odd]. We will see that the pseudo-version of the Subtractive Algorithm is the Binary Algorithm, which we now describe.

3.5. **The Euclidean dynamical systems for Type 3.** For the LMSB Group which contains two divisions, the Binary division or the Plus-Minus Division, the main decision is taken by the LSB's.

The *Binary algorithm* operates on pairs of odd integers that belong to the set $\widetilde{\Omega} := \{(u, v),\ u, v \,\text{odd},\ 0 < u \leq v\}$. The binary division creates zeroes on the left of v, and builds a quotient m such that $r := v - m \cdot u$ is strictly smaller than u: it is directed both by the least significant bits (for creating zeroes on the left) and by the most significant bits (for deciding the moment of the exchange). The Binary Algorithm is described in Figure 16.

The Binary division corresponds to each inner loop and is performed on associated rationals as follows: For a pair (u, v) formed with odd integers, the rational $1/x = v/u \in [0, 1]$ has a 2-adic norm equal to 1: It belongs to the set U_2 of 2-adic

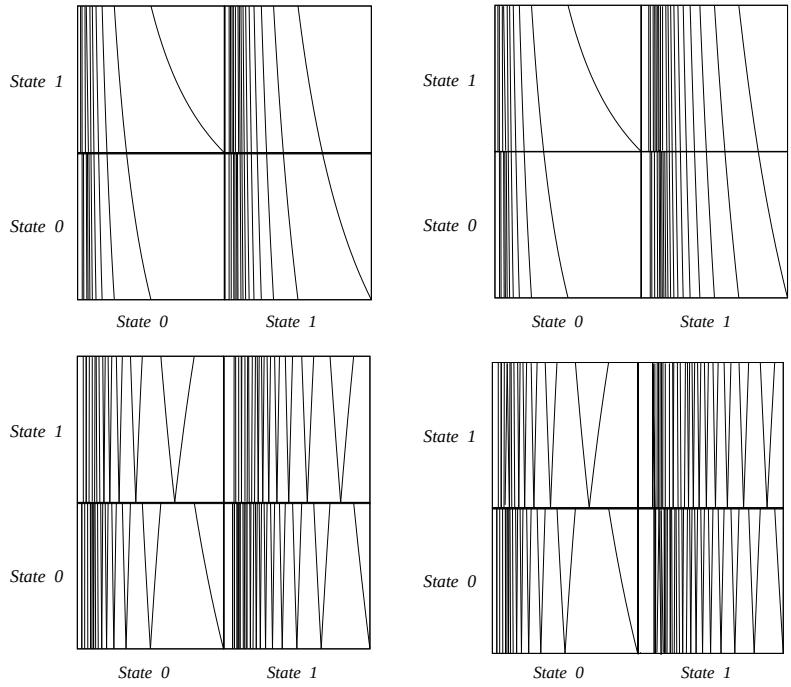

FIGURE 15. Two examples of Euclidean Dynamical Systems relative to Type 2. Above, Algorithm $\breve{G}$ [Pseudo-Standard] with $k = 1$ and $k = 2$. Below, Algorithm $\breve{K}$ [Pseudo-Centered] with $k = 1$ and $k = 2$.

Input: $(u, v) \in \widetilde{\Omega} := \{(u, v),\ u, v \,\text{odd},\ 0 < u \leq v\}$;
While $u \neq v$ do
  While $u < v$ **do**
    $b := \text{val}_2\,(v-u)$; $v := (v-u)/2^b$;
  Exchange u and v;
Output: u (or v).

FIGURE 16. Description of the Binary Algorithm

units. Then the 2-adic valuation b of $z := (1/x) - 1$ satisfies $b \geq 1$ and $y := 2^{-b}z$ again belongs to U_2 and is positive. There are now two cases: if $y \leq 1$ then $V_{(b)}(x) := y$, else $V_{(b)}(x) := 1/y$. Finally, the dynamical system $([0, 1], V_{(b)})$ has two branches

$$V_{(b)}(x) = 2^b \frac{x}{1-x} \quad \text{if} \quad x \leq \frac{1}{2^b+1}, \qquad V_{(b)}(x) = \frac{1}{2^b}\frac{1-x}{x} \quad \text{if} \quad x \geq \frac{1}{2^b+1},$$

and its inverse branches can be defined by means of inverse branches of the Subtractive Algorithm

$$p_{(b)}(x) = p\left(\frac{x}{2^b}\right) = \frac{x}{x+2^b}, \qquad q_{(b)}(x) = q\left(2^b x\right) = \frac{1}{1+2^b x}, \tag{3.3}$$

so that the Binary Algorithm is a pseudo-version of the Subtractive Algorithm; however the mixing between the binary shift and branches p, q of the Subtractive System is more involved than above [See Figure 17]. The density transformer $\mathbf{H}$ of the Binary Euclidean System, defined as

$$\mathbf{H}[f](x) := \sum_{b\geq 1} \left(\frac{1}{1+2^b x}\right)^2 f\left(\frac{1}{1+2^b x}\right) + \sum_{b\geq 1} \left(\frac{1}{x+2^b}\right)^2 f\left(\frac{x}{x+2^b}\right), \qquad (3.4)$$

was introduced by Brent [14] in his study of the binary gcd, but it is not easy to deal with. This is why the "induced version" of the transfer operator has been further introduced by Vallée [105].

Between two exchanges, there is a sequence of internal loops (composed with subtractions and binary shifts) that can be written as

$$v = u + 2^{b_1} v_1, \qquad v_1 = u + 2^{b_2} v_2, \qquad v_2 = u + 2^{b_3} v_3, \qquad \dots \qquad v_{\ell-1} = u + 2^{b_\ell} v_\ell,$$

with $v_\ell < u$. Then, we let $r' := v_\ell$ and we exchange u and r'. Formally, this sequence can also be written as

$$u/v = p_{(b_1)} \circ p_{(b_2)} \circ \dots \circ p_{(b_{\ell-1})} \circ q_{(b_\ell)}(v_\ell/u),$$

[where $p_{(b)}$ and $q_{(b)}$ are inverse branches of shift $V_{(b)}$ defined in (3.3)]. Note that we use here the operation of induction described in Section 2.6, with respect to the "bad" inverse branch p : we form sequences of branches p followed with the "good" inverse branch q. An elementary step of this induced process is summarized by the relation $v = mu + 2^k v_\ell$ where m is an odd integer equal to

$$m = 1 + 2^{b_1} + 2^{b_1+b_2} + 2^{b_1+b_2+b_3} + \dots + 2^{b_1+b_2+b_3+\dots+b_{\ell-1}},$$

while the exponent k is equal to $k = b_1 + b_2 + b_3 + \dots + b_{\ell-1} + b_\ell$. If $x = x_0$ denotes the rational u/v at the beginning of an internal loop, the global result of an elementary step of this induced process is the rational $x_1 = v_\ell/u$, defined by

$$x_0 = h(x_1) \qquad \text{with} \quad h(x) := \frac{1}{m+2^k x}, \qquad \text{with } m \text{ odd, } 1 \leq m < 2^k, \text{ and } k \geq 1. \qquad (3.5)$$

The transfer operator relative to the induced process is then

$$\widetilde{\mathbf{H}}[f](x) := \sum_{k\geq 1} \sum_{\substack{m \text{ odd}, \\ 1\leq m<2^k}} \left(\frac{1}{m+2^k x}\right)^2 f\left(\frac{1}{m+2^k x}\right) \qquad (3.6)$$

The *Plus-Minus division* is similar; it is just a refinement of the Binary Division. For any $x \in U_2$, amongst

$$V_-(x) := \frac{1}{x} - 1, \qquad V_+(x) := \frac{1}{x} + 1,$$

there is exactly a unique $\epsilon = \pm 1$ for which $V_\epsilon(x)$ has a valuation b that satisfies $b \geq 2$, and we choose this $V_\epsilon(x)$ as our y. For any (ϵ, b), with $\epsilon = \pm 1$ and $b \geq 2$, we build a dynamical system $([0,1], V_{[\epsilon,b]})$ which has two branches of the form

$$V_{(\epsilon,b)}(x) = 2^b \frac{x}{1-\epsilon x} \quad \text{if} \quad x \leq \frac{1}{2^b - \epsilon}, \qquad V_{(\epsilon,b)}(x) = \frac{1}{2^b} \frac{1-\epsilon x}{x} \quad \text{if} \quad x \geq \frac{1}{2^b - \epsilon}.$$

This modelisation appeared for the first time in [23].

Remark that the image of the unit interval by the branch $V_{(-,b)}$ equals $[2^{1-b}, 1]$, so that the Dynamical System is no longer complete…: this is a main difference between the two algorithms of Type 3. [See Figure 17 below]. As previously, the

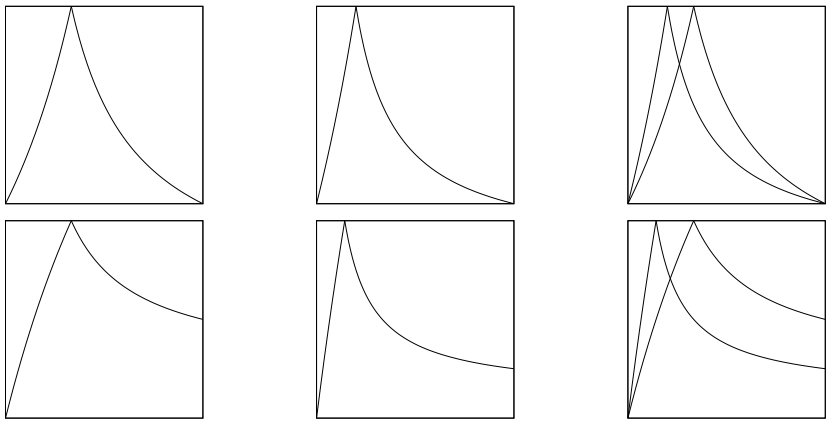

FIGURE 17. The Euclidean Dynamical Systems relative to Type 3. Above, the Binary Algorithm, with $b = 1$, $b = 2$, then $b = 1$ and $b = 2$ on the same figure. Below, the Plus-Minus Algorithm, with $\epsilon = -1, b = 2$, $\epsilon = -1, b = 3$, then $b = 2$ and $b = 3$ on the same figure.

induced system can be built. Between two exchanges, there is a sequence of internal loops (composed with subtractions, additions and binary shifts) that can be written as

$$v = \epsilon_1\, u + 2^{b_1} v_1, \qquad v_1 = \epsilon_2\, u + 2^{b_2}\, v_2, \qquad \dots \qquad v_{\ell-1} = \epsilon_\ell\, u + 2^{b_\ell}\, v_\ell,$$

with $v_\ell < u$. Then, we let $r' := v_\ell$ and we exchange u and r'. An elementary step of this induced process is summarized by the relation $v = mu + 2^k v_\ell$ where m is an odd integer equal to

$$m = \epsilon_1 + \epsilon_2\, 2^{b_1} + \epsilon_3\, 2^{b_1+b_2} + \ldots + \epsilon_\ell\, 2^{b_1+b_2+b_3+\ldots+b_{\ell-1}},$$

while the exponent k is equal to $k = b_1 + b_2 + b_3 + \ldots + b_{\ell-1} + b_\ell$. If $x = x_0$ denotes the rational u/v at the beginning of an internal loop, the global result of an elementary step of this induced process is the rational $x_1 = v_\ell/u$, defined by

$$x_0 = h(x_1) \qquad \text{with} \quad h(x) := \frac{1}{m + 2^k x}, \tag{3.7}$$

where the possible values of "digits" (m, k) are

$$\mathcal{D} = \{d = (m, k),\ k \geq 2 \text{ and } m \text{ odd }, |m| \leq m(k)\},$$

where $m(k)$ equals $(2^k - 1)/3$ if k is even and $(2^k - 5)/3$ if k is odd.

3.6. **The Euclidean dynamical systems for Type 4.** The LSB algorithms operate on pairs of integers that belong to the set $\widetilde{\Omega} := \{(u, v), v \text{ odd}, u \text{ even}\}$. If a is the number of zeroes on the left of u [i.e., $a := \mathrm{Val}_2(u)$], then $u' := 2^{-a} \cdot u$ is odd. The LSB division creates zeroes on the left of v, and builds a quotient m such that $r := v - m \cdot u'$ has a number b of zeroes on the left strictly greater than a. Then r and u are both multiples of 2^a, and the relations $u' := 2^{-a} \cdot u, r' := 2^{-a} \cdot r$ defines an integer pair (u', r') which will be an element of $\widetilde{\Omega}$ and the new pair for the next step. It is easy to see that this (plain) quotient m satisfies the three conditions : m odd, $m \geq 1$, $m < 2^{a+1}$. It is of course possible to center the quotient m, so that

Input: $(u, v) \in \Omega$;
Repeat
$a := \mathrm{Val}_2(u)$; $u := 2^{-a} \cdot u$; $b := 0; m := 0$
While $b \leq a$ **do**
$k := \mathrm{Val}_2(v - u)$; $v := (v - u)/2^k$;
$m := m + 2^b$; $b := b + k$;
(*) **If** $m > 2^a$, **then** $[m := m - 2^{a+1}$; $v := v + 2^{a+1} \cdot u$;]
Exchange u and v;
until $u = 0$.
Output: v .

FIGURE 18. Description of the LSB Algorithms. The line (∗) is only executed for the centered LSB algorithm.

this (centered) quotient m satisfies the three conditions: m odd, $|m| < 2^a$ [see Line (*) in Figure 18].

The algorithms are then completely governed by the least significant bits. Figure 18 describes two gcd algorithms; each algorithm is relative to a LSB division, the plain LSB division or the centered LSB division. The centered LSB algorithm was introduced in [98], because the plain LSB division does not always terminate. It loops forever for instance on integer pairs of the form $(u, v) = (-2v, v)$. However, we shall prove in the sequel that the average number of iterations is finite.

The (plain) LSB algorithm uses the set $\mathcal{M}$ of matrices,

$$\mathcal{M} := \left\{ \begin{pmatrix} 0 & 2^a \\ 2^a & m \end{pmatrix}; \qquad a \geq 1, m \text{ odd }, |m| < 2^a \right\}, \tag{3.8}$$

while the (centered) LSB algorithm uses the set $\check{\mathcal{M}}$ of matrices,

$$\check{\mathcal{M}} := \left\{ \begin{pmatrix} 0 & 2^a \\ 2^a & m \end{pmatrix}; \qquad a \geq 1, \, m \text{ odd }, 1 \leq m < 2^{a+1} \right\}. \tag{3.9}$$

Each algorithm uses a set $\mathcal{G}$ [resp $\check{\mathcal{G}}$] of LFT's of the form $g(x) = 2^a/(m + 2^a x)$, where the odd m satisfies the plain condition $1 \leq m < 2^{a+1}$ [for $\mathcal{G}$] or the centered condition $|m| < 2^a$ [for $\check{\mathcal{G}}$]. As we already explained in Section 2.4, we are then led to work with a system $\mathcal{H}$ [resp. $\check{\mathcal{H}}$] of iterated functions, defined as

$$\mathcal{L} := \{h := \underline{g}; \quad g \in \mathcal{G}\}, \qquad \check{\mathcal{H}} := \{h := \underline{g}; \quad g \in \check{\mathcal{G}}\}$$

where the underline is the conjugaison with the Tangent map, and $\mathcal{G}$ [resp. $\check{\mathcal{G}}$] is the set of LFT's relative to set $\mathcal{M}$ [resp. $\check{\mathcal{M}}$] of matrices used by the LSB algorithm. Then, each inverse branch h of the form

$$h(\theta) = \arctan\left(\frac{2^a}{m + 2^a \tan\theta}\right)$$

is chosen with probability 2^{-2a}. Finally, the transfer operator which will be used in the analysis is

$$\mathbf{H}[f](\theta) := \sum_{k \geq 1} \sum_{\substack{m \text{ odd} \\ |m| < 2^k}} \frac{1 + \tan^2\theta}{2^{2k} + (m + 2^k \tan\theta)^2} \cdot f\left(\arctan\left(\frac{2^k}{m + 2^k \tan\theta}\right)\right), \tag{3.10}$$

in the case of the LSB centered algorithm, for instance.

Group	*Type*	*Generic or Markovian*	*Pure or Mixed*	*Good, Bad or Difficult*	*Name*
Po	0	Ge	Pure [U]	Good	Standard [PoG]
MSB	1	Ge	Pure [A]	Good	Standard [G]
				Good	Odd [O]
				Good	Centered [K]
				Bad	By-Excess [M]
				Bad	Even [E]
				Bad	Subtractive [T]
MLSB	2	Ma	Mixed	Good	Pseudo- Standard [$\breve{G}$]
				Good	Pseudo-Odd [$\breve{O}$]
				Good	Pseudo-Centered [$\breve{K}$]
				Bad	Pseudo-By-Excess [$\breve{M}$]
LMSB	3	Ge	Mixed	Difficult	Binary [B]
				Not studied	Plus-Minus [PM]
LSB	4	Ge	Mixed	Difficult	Plain LSB [L]
				Difficult	Centered LSB [$\check{L}$]

FIGURE 19. Main features of the Euclidean Dynamical Systems. A means Archimedean, U means ultrametric.

Class	Proven Theorems
Good	All
Bad	3,6
Difficult	1,3,4,5,6

Fast Class	Good Class $\bigcup$ Difficult Class.
Slow Class	Bad Class
Easy Class	Good Class $\bigcup$ Bad Class

FIGURE 20. Proven Theorems. Various Classes

Here, the entropy α of this system of iterative functions is related to a Lyapounov exponent β. Consider the set $\mathcal{M}$ of matrices defined in (3.8) or in (3.9), where a matrix M relative to a valuation a is chosen with probability $\delta_M := 2^{-2a}$, and remind the definition of the Lyapounov exponent β,

$$\beta := \frac{1}{n} \lim_{n\to\infty} \mathbb{E}\left[\log ||M_1 \cdot M_2 \cdot \ldots \cdot M_n||\right],$$

[when each matrix M is independently drawn in $\mathcal{M}$ with probability δ_M]. Then $\alpha = 8\log 2 - 2\beta$ [see [25] for a proof].

3.7. **Final Classification of all the Euclidean Dynamical systems.** Figure 19 provides a list of all the algorithms which are studied in this paper, with their main characteristics: their type (defined by a number between 0 and 4), if they are generic or markovian, if they are pure or mixed, easy or difficult, fast or slow,

etc. Figure 20 shows what Theorems are valid for each Class [see Section 2.8 for a description of the main nine theorems of this paper].

4. **Main results: Generic truncated trajectories.** Here, we describe the second step of Dynamical Analysis, where generic trajectories are studied. Under some precise conditions, and for a set of algorithms which will be described further, we shall prove that our main parameters, the total cost C_n, or the size of continuants ℓ_n follow asymptotically a Gaussian law. We provide precise central limit theorems for variables C_n, ℓ_n, with an optimal speed of convergence. Furthermore, we give explicit formulae for the mean and the variance. Analog results can be found in [17] or [22]. However, the general framework is not exactly the same.

4.1. **Generic truncated trajectories versus rational trajectories.** We recall that our final goal is to study the execution of a gcd algorithm. These executions correspond to rational trajectories $(x, Vx, V^2x, ...)$ of the Dynamical System which meet the final value η. If $P(x)$ is the first index i for which $V^i(x) = \eta$, the algorithm stops at the $P(x)$ step, and produces a stopping trajectory $\underline{\mathcal{V}}(x) := (x, Vx, V^2x, \dots V^{P(x)}x)$. We first replace this study [rational trajectories] by a slightly different one, which will be easier. We recall that such a transform constitutes the Second Step in a Dynamical Analysis.

We consider for $x \in X$ the trajectory $\mathcal{V}(x) = (x, Vx, V^2x, ...)$, we choose some integer n, which will be fixed in the whole Section, and we truncate [ourselves] the trajectory at the n–th step: we obtained a truncated trajectory $\mathcal{V}_n(x) := (x, Vx, V^2x, \dots V^{n-1}x)$ which depends both on n (which will be fixed) and x (which will be chosen at random in X): more precisely, we fix a "smooth" probability measure on X, we denote by $\mathbb{E}[\cdot]$ the corresponding expectation, and we wish to study the probabilistic behaviour of these truncated trajectories. We consider that X is endowed with some probability absolutely continuous with respect to the Haar measure on X. For mixed Types, we recall that the Dynamical System is probabilistic, so that the trajectories also depend on random choices.

Such a truncated trajectory can be encoded as

$$(d_1(x), d_2(x), \dots d_n(x)), \qquad \text{with} \quad d_i(x) = \sigma(V^{i-1}x),$$

and it uses the inverse branch $h := h_{[d_1]} \circ h_{[d_2]} \circ \dots \circ h_{[d_n]}$ of depth n. The main observables, which are now the total cost C_n and the n-th continuant q_n, are random variables defined on X, which we now describe.

The total cost C_n is defined as

$$C_n(x) := \sum_{i=1}^{n} c(d_i(x)), \qquad \text{with} \quad d_i(x) = \sigma(V^{i-1}x).$$

Since the digit–cost only depends on the digit d, it depends only on the inverse branch $h = h_{[d]}$ which is used. Then, the cost c is also defined on the set $\mathcal{H}$ of inverse branches, and it can be extended to a cost on the semi-group $\mathcal{H}^\star$ by additivity. The total cost can be alternatively defined as

$$C_n(x) := \sum_{i=1}^{n} c(h_{[d_i]}) = c(h_{[d_1]} \circ h_{[d_2]} \circ \dots \circ h_{[d_n]}), \tag{4.1}$$

and we are interested in the asymptotic distribution of cost C_n (for $n \to \infty$).

The truncated trajectory $\mathcal{V}_n(x)$ builds a continued fraction of depth n: It is the truncation of the continued fraction (of infinite depth) which corresponds to the

complete trajectory $\mathcal{V}(x)$. It is defined by the LFT $h_{[d_1]} \circ h_{[d_2]} \circ \ldots \circ h_{[d_n]}$ and the rational

$$\frac{p_n}{q_n}(x) := h_{[d_1]} \circ h_{[d_2]} \circ \ldots \circ h_{[d_n]}(0)$$

is called the n–th convergent of x. The pair $Q_n := (p_n, q_n)$, defined as

$$Q_n = \begin{pmatrix} p_n \\ q_n \end{pmatrix} = M_{[d_1]} \cdot M_{[d_2]} \cdot \ldots \cdot M_{[d_n]} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

is a random variable which is called the continuant of order n, and we wish to study its size $\ell(|Q_n|)$, where ℓ is the size, and the absolute value is defined in (1.10, 1.12). In fact, in the number case, we replace the binary size by the plain logarithm, and we define in all the cases

$$\ell_n := \log_*(|Q_n|),$$

where the logarithm $\log_*$ is $\log_q$ in polynomial case, and $\log_2$ in the number case. It is of great (algorithmic) interest to study the asymptotic distribution of random variable ℓ_n.

4.2. **The Quasi-Powers Theorem.** How to obtain an asymptotic Gaussian law? We consider a sequence of random variables R_n defined on the same probabilistic space, and we wish to prove that the limit law of variable R_n is the Gaussian law. To establish such an asymptotic gaussian law, it is standard to use the moment generating function, i.e., the expectation $\mathbb{E}[\exp(wR_n)]$, for w complex. In probabilistic i.i.d. situations, $\mathbb{E}[\exp(wR_n)]$ is the n-th power of some expectation. In our setting, and for each parameter $R_n = C_n$ or $R_n = \ell_n$, a quasi-powers approximation (with uniform remainder term) will be be obtained for $\mathbb{E}[\exp(wR_n)]$. In this case, the following Quasi-Powers Theorems of Hwang [47, 48, 49] will be used and this will lead to the asymptotic gaussian law. This theorem is a compact and versatile statement, which encapsulates the consequences of the Lévy continuity theorem and the Berry-Esseen inequality.

Theorem A [Quasi-Powers Theorem.] *Assume that the moment generating functions $\mathbb{E}[\exp(wR_n)]$ for a sequence of functions R_n are analytic in a complex neighborhood $\mathcal{W}$ of $w = 0$, and satisfy*

$$\mathbb{E}[\exp(wR_n] = \exp[\beta_n U(w) + V(w)] \left(1 + O(\kappa_n^{-1})\right) , \tag{4.2}$$

with β_n, $\kappa_n \to \infty$ as $n \to \infty$, $U(w)$, $V(w)$ analytic on $\mathcal{W}$ and the O–term uniform in $\mathcal{W}$. Then, the mean and the variance satisfy

$$\mathbb{E}[R_n] = U'(0) \cdot \beta_n + V'(0) + O(\kappa_n^{-1}) , \qquad \mathbb{V}[R_n] = U''(0) \cdot \beta_n + V''(0) + O(\kappa_n^{-1}) .$$

Furthermore, if $U''(0) \neq 0$, the distribution of R_n is asymptotically Gaussian, with speed of convergence $O(\kappa_n^{-1} + \beta_n^{-1/2})$,

$$\mathbb{P}_\nu \left[x \;\middle|\; \frac{R_n(x) - U'(0)n}{\sqrt{U''(0)n}} \leq Y \right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y} e^{-y^2/2}\, dy + O(\kappa_n^{-1} + \beta_n^{-1/2}) .$$

4.3. **Relating moment generating functions to n–th powers of transfer operators.** We now provide an alternative expression of moment generating functions of our main parameters which relate them to some extensions of the density transformer.

Digit-costs. We relate this expectation $\mathbb{E}[\exp(wC_n)]$ to the n–th iterate of the operator $\mathbf{H}_{1,w,(c)}$ defined in Figure 4. This weighted transfer operator $\mathbf{H}_{1,w,(c)}$ will be our main tool. It is defined as a perturbation of the density transformer $\mathbf{H}_1$: we multiply the component operator $\mathbf{H}_{1,[h]}$ by the factor $\exp[wc(h)]$. We then obtain a weighted component operator

$$\mathbf{H}_{1,w,[h],(c)} = \delta_h \cdot \exp[wc(h)] \cdot |h'(x)| \cdot f \circ h(x)$$

and the "total" weighted transfer operator is defined as in (2.5) or in (2.7),

$$\mathbf{H}_{1,w,(c)}[f](x) = \sum_{h\in\mathcal{H}} \delta_h \cdot \exp[wc(h)] \cdot |h'(x)| \cdot f \circ h(x) \tag{4.3}$$

for generic types. For Markovian systems [Type 2], we perform this transformation in each component of each coefficient of the matrix in (2.7). Then, additive properties of costs and multiplicative properties of derivatives [or denominators] entail a nice formula for the n-th iterate of the operator $\mathbf{H}_{1,w,(c)}$, for instance,

$$\mathbf{H}^n_{1,w,(c)}[f](x) = \sum_{h\in\mathcal{H}^n} \delta_h \exp[wc(h)] \cdot |h'(x)| \cdot f \circ h(x)\,, \tag{4.4}$$

for generic types.

We now relate the expectation $\mathbb{E}[\exp(wC_n)]$ to the n–th iterate of the operator $\mathbf{H}_{1,w,(c)}$. Remark that the cost $C_n(x)$ is equal to $c(h)$ on the "fundamental" set $h(X)$. A fundamental set is just the image $h(X)$ of X by an element $h \in \mathcal{H}^\star$; When the depth of h equals n, such a set is exactly the subset of $x \in X$ for which the first n digits of x are fixed and equal to digits of the rational $h(\eta)$. Remind that a LFT $h \in \mathcal{H}$ is chosen with a probability equal to

$$\mathbb{P}[h \text{ is chosen}] = \delta_h \cdot \int_{h(X)} f(u)du = \delta_h \int_X |h'(x)| f \circ h(x)dx.$$

Then,

$$\mathbb{E}[\exp(wC_n)] = \sum_{h\in\mathcal{H}^n} \exp[wc(h)] \cdot \mathbb{P}[h \text{ is chosen}] = \int_X \mathbf{H}^n_{1,w,(c)}[f](x)\,dx \tag{4.5}$$

For Markovian types, instead of $\mathbf{H}^n_{1,w,(c)}[f](u)$, we have a matrix relation, of the form

$$\begin{pmatrix}1 & 1\end{pmatrix} \mathbf{H}^n_{1,w,(c)} \begin{pmatrix} f^{[0]} \\ f^{[1]} \end{pmatrix}(u).$$

Continuants. The moment generating function $\mathbb{E}[\exp(2w\ell_n)]$ of ℓ_n can be expressed in terms of the n–th iterate of the operator $\underline{\mathbf{H}}_{1,w}$. This operator is defined by its component operators $\underline{\mathbf{H}}_{1,w,[h]}$ which operate on functions F of two real variables x and y as follows:

$$\underline{\mathbf{H}}_{1,w,[h]}[F](x,y) = \delta_h^{1-w} \cdot |h'(x)| \cdot |h'(y)|^{-w} \cdot F(h(x),h(y))\,, \tag{4.6}$$

Then, the total transfer operator is defined as in (2.5) [for generic types] or as in (2.7) for Markovian types. Remark that, on the diagonal $x = y$, one has

$$\underline{\mathbf{H}}_{1,w}[F](x,x) = \sum_{h\in\mathcal{H}} \delta_h^{1-w} \cdot |h'(x)|^{1-w} \cdot F(h(x),h(x)).$$

Then, the operator $\underline{\mathbf{H}}_{1,w}$ is related to the (plain) transfer operator $\mathbf{H}_s$ defined in (2.5) or in (2.7), in the sense that $\underline{\mathbf{H}}_{1,w}$ can be viewed as an "extension" of $\mathbf{H}_{1-w}$ to functions of two variables. It can be also viewed as an "extended" perturbation of $\mathbf{H}_1$.

In all the cases, the relation $|Q_n^{-2}(x)| = \delta_h \cdot |h'(0)|$ [which relates the point x and the inverse branch of depth n which is relative to the beginning CFE of x] is the key for the alternative expression for the moment generating function of $\mathbb{E}[\exp(2w\ell_n)]$ as a function of $\underline{\mathbf{H}}_{1,w}^n$,

$$\mathbb{E}[\exp(2w\ell_n)] = \sum_{h\in\mathcal{H}^n} \delta_h^{-w} \cdot |h'(0)|^{-w} \cdot \mathbb{P}[h \text{ is chosen}] = \int_X \underline{\mathbf{H}}_{1,w}^n[F](u,0)du, \quad (4.7)$$

with $F(x,y) := f(x)$, where f is the density chosen on X.

Finally, Relations (4.7) and (4.5) show that the moment generating functions of interest are closely related to powers of transfer operators $\mathbf{H}_{1,w,(c)}$ or $\underline{\mathbf{H}}_{1,w}$.

4.4. **Functional analysis.** We first describe a set of conditions on a general operator $\mathbf{G}_w$, and we will prove that these conditions are sufficient to entail a quasi-power behaviour of the n–th power of this transfer operator $\mathbf{G}_w$, and thus a quasi-power behaviour for related moment generating functions.

Conditions (A). *There exists a functional space $\mathcal{F}$ where the following is true:*
[$\mathrm{An}_w(1,0)$]. *The operator $\mathbf{G}_w$ acts on $\mathcal{F}$ when w is (complex) near 0 and the map $w \mapsto \mathbf{G}_w$ is analytic at $w = 0$.*
[UDE and SG]. *The operator $\mathbf{G}_0 : \mathcal{F} \to \mathcal{F}$ has a unique dominant eigenvalue $\lambda = 1$, and a spectral gap: the rest of the spectrum lies in a disk whose radius is < 1.*
[SLC_w]. *The pressure map $\Lambda_G(w) := \log \lambda_G(w)$ has a second derivative which is not zero at $w = 0$.*

Theorem AA. *Suppose that Conditions (A) hold for an operator $\mathbf{G}_w$. Then the hypotheses of Theorem A [Quasi-Powers Theorem] are fulfilled for all the variables R_n whose moment generating function is "closely" related to the n-th power $\mathbf{G}_w^n$, applied to some function F of $\mathcal{F}$.*

Proof. Elementary perturbation theory [53] implies that $\mathbf{G}_w$ inherits the dominant eigenvalue property and the spectral gap when w is near 0. This proves that the n–th iterate $\mathbf{G}_w^n$ of the operator behaves as a uniform quasi–power where the main term involves the n–th power of the dominant eigenvalue $\lambda_G(w)$ of the operator $\mathbf{G}_w$, so that $U(w) = \log \lambda_G(w))$ and $\beta_n = n$. Finally, the condition $U''(0) \neq 0$ is exactly Condition [SLC_w]. □

4.5. **Generic truncated trajectories: Study of digit–costs.** When the cost c is of moderate growth [i.e., $c(d)$ is $O(\ell(d)$], and the gcd algorithm is any element of the Fast Class, then Conditions (A) are fullfilled for the weighted transfer operator $\mathbf{G}_w := \mathbf{H}_{1,w,(c)}$ defined in Figure 4. These facts will be proven in Section 8, where the convenient functional space $\mathcal{F}$ will be described.

In both cases, relation (4.5) and Theorem (AA) entail that the Quasi-Powers Theorem can be applied. This leads to an asymptotic Gaussian Law for the total cost C_n. We provide a quite precise central limit theorem for variable C_n, with an optimal speed of convergence. Furthermore, we give explicit formulae for the mean and the variance.

Theorem 1. *Consider any gcd algorithm which belongs to the Fast Class, together with a non constant digit cost of moderate growth. Consider any probability $\mathbb{P}$ on*

X with a smooth density $f \in \mathcal{F}$ with respect to the Haar measure on X. Then, there are $\mu(c)$ and $\rho(c)$ so that for any n, and any $Y \in \mathbb{R}$

$$\mathbb{P}\left[x \;\middle|\; \frac{C_n(x) - \mu(c)n}{\rho(c)\sqrt{n}} \leq Y\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y} e^{-y^2/2}\, dy + O\left(\frac{1}{\sqrt{n}}\right).$$

Furthermore, if r_1 is the subdominant spectral radius of the density transformer $\mathbf{H}$, for any θ which satisfies $r_1 < \theta < 1$, one has :

$$\mathbb{E}[C_n] = \mu(c) \cdot n + \mu_1(c) + O(\theta^n), \quad \mathbb{V}[C_n] = \rho^2(c) \cdot n + \rho_1(c) + O(\theta^n).$$

Moreover, the constants $\mu(c)$ and $\rho^2(c)$ admit expressions which involve the pressure function $\Lambda(1, w) := \log \lambda(1, w)$ relative to the weighted transfer operator $\mathbf{H}_{1,w,(c)}$ defined in (4.3),

$$\mu(c) = \Lambda'_w(1, 0) = \lambda'_w(1, 0), \qquad \rho^2(c) = \Lambda''_{w^2}(1, 0) = \lambda''_{w^2}(1, 0) - \lambda'^2_w(1, 0).$$

Finally, $\mu(c)$ involves the invariant density ψ,

$$\mu(c) = \sum_{h \in \langle} \delta_h \cdot c(h) \int_{h(X)} \psi(t)\, dt\,.$$

4.6. **Generic trajectories for the Good Class: Study of continuants.** For the good Euclidean dynamical systems, then Conditions (A) are fullfilled for the transfer operator $\mathbf{G}_w := \underline{\mathbf{H}}_{1,w}$ defined in Figure 4. This will be proven in Section 8. Then, relation (4.7) and Theorem (AA) entail that the Quasi-Powers Theorem can be applied. This leads to an asymptotic Gaussian Law for the size of continuants ℓ_n. We provide a quite precise central limit theorem for variable ℓ_n, with an optimal speed of convergence. Furthermore, we give explicit formulae for the mean and the variance.

Remark that, for algorithms of the Difficult Class, the transfer operator $\mathbf{G}_w := \underline{\mathbf{H}}_{1,w}$ does not seem to satisfy Conditions (A) [see Section 8]. Finally, we have proven:

Theorem 2. *Consider any gcd algorithm of the Good Class. Denote by ℓ_n the logarithm of the n–th continuant Q_n (in the number case) or $\deg Q_n$ (in the polynomial case). Consider any probability $\mathbb{P}$ on X with a smooth density f. Then, there are β and γ so that for any n, and any $Y \in \mathbb{R}$*

$$\mathbb{P}\left[x \;\middle|\; \frac{2\ell_n(x) - \alpha n}{\gamma\sqrt{n}} \leq Y\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{Y} e^{-y^2/2}\, dy + O\left(\frac{1}{\sqrt{n}}\right).$$

Furthermore, if r_1 is the subdominant spectral radius of the density transformer $\mathbf{H}$, for any θ which satisfies $r_1 < \theta < 1$, one has :

$$\mathbb{E}[\ell_n] = \frac{\alpha}{2} \cdot n + \alpha_1 + O(\theta^n), \quad \mathbb{V}[C_n] = \frac{\gamma^2}{4} \cdot n + \gamma_1 + O(\theta^n).$$

Moreover, the constants α and γ admit expressions which involve derivatives of $w \mapsto \Lambda(1 - w)$ at $w = 0$, where $\Lambda(s) := \log \lambda(s)$ is the pressure function of the weighted transfer operator $\mathbf{H}_s$ defined in (2.5) or in (2.7),

$$\alpha = |\Lambda'(1)| = -\lambda'(1), \qquad \gamma^2 = \Lambda''(1) = \lambda''(1) - \lambda'^2(1)$$

We recall that α is the entropy of the dynamical system and can be expressed with the invariant function ψ,

$$\alpha = -\sum_{h\in\mathcal{H}} \delta_h \log \delta_h - \int_X \log|V'(t)|\psi(t)dt.$$

5. **Back to the Euclidean Algorithms.** With this Section, we begin the third step of Dynamical Analysis. As it is often the case, the discrete problem is more difficult than its continuous counterpart.

An execution of a Euclidean algorithm on the input (u, v) formed with two integers u, v such that $u/v = x$ gives rise to a rational trajectory $\underline{\mathcal{V}}(x)$ which stops at the $P(x)$–step when it meets η, and, we work with the stopping trajectory $\underline{\mathcal{V}}(x)$. When we worked previously [in Section 4] with (generic) truncated trajectories, the truncation degree n was chosen by us, and we are led to study n-th powers of transfer operators, with respect to the reference parameter n. For a rational trajectory $\underline{\mathcal{V}}(x)$ corresponding to an input (u, v), the truncation degree is in a sense automatic [it is no longer chosen], and the reference parameter is now the size $L(u, v)$ of the input (u, v). On inputs on size N, the reference probability measure $\mathbb{P}_N$ is now the uniform discrete measure on the (finite) sets Ω_N, $\widetilde{\Omega}_N$ of inputs of size N defined in (1.14).

We then proceed in two steps: First, we work with all possible inputs of Ω, which build all the rational trajectories [these which meet η]. Since we have to consider LFT's with any depth, we wil be led to study quasi–powers of transfer operator [This first step will be performed in this Section]. Second, later on, in Sections 6 and 7, we restrict ourselves to a given size N, and we have to "extract" these particular inputs.

5.1. **Parameters of interest.** We now describe the new parameters of interest. First, the total cost of the trajectory, relative to a digit–cost c, is

$$C(u, v) := \sum_{i=1}^{P(u,v)} c(d_i(\frac{u}{v}))\,. \tag{5.1}$$

Second, we are also interested in the behaviour of the remainder pair $U_i := (u_{i+1}, u_i)$. More precisely, we wish to study the Interrupted Algorithm which stops as soon as the absolute value of the remainder pair U_i becomes smaller than some power of the absolute value $|U_0|$ of the input pair, and we are led to describe the behaviour of the continuant at a fraction of the depth $P(u, v)$ [the depth P is the depth of the continued fraction or the number of iterations of the algorithm]. We consider a parameter $\delta \in [0, 1]$, and we define the remainder at the fraction δ as a random variable $U_{[\delta]}$ on $\Omega, \widetilde{\Omega}$. More generally, we wish to study the behaviour of (beginning and ending) continuants at a fraction of the depth. The definition of beginning and ending continuants are given in Section 1. 3 [See Equations (1.6, 1.8)]. It is then natural to let, for $X \in \{U, V, Q\}$,

$$X_{[\delta]}(u, v) = X_{\lfloor \delta P(u,v)\rfloor}(u, v), \qquad LX_{[\delta]}(u, v) := \log_*(|X_{[\delta]}(u, v)|) \tag{5.2}$$

where, as previously, $\log_*$ is $\log_2$ in the number case or $\log_q$ for polynomials in $\mathbb{F}_q[Z]$.

Third, the bit–complexities of the (plain) algorithm (which only computes the gcd) or of the extended algorithm (which also computes the Bezout Coefficients) on an input (u, v), defined as

$$B(u,v) := \sum_{i=1}^{P(u,v)} \ell(d_i) \cdot L(U_i), \qquad \overline{B}(u,v) := \sum_{i=1}^{P(u,v)} \ell(d_i) \cdot L(Q_i) \tag{5.3}$$

involve parameters of various types.

Remark. On a coprime input (u, v), the Euclid algorithm "writes" the result $u/v = h(0)$. We remark that the last step of a Euclid algorithm is particular [see Figure 1 for some instances]. As an example, the standard Euclidean algorithm cannot use the quotient $m = 1$ in its last step. Then, the LFT used by the algorithm belongs to $\mathcal{H}^\star \times \mathcal{F}$ where $\mathcal{F}$ is the set of LFT's used in the last step. In fact, in this case, any rational admits two CFE's, the first one, built by the Euclid Algorithm, is the proper one: its sequence of digits ends with an element of $\mathcal{F}$. The second one is the improper one and does not end with an element of $\mathcal{F}$: it cannot be produced by the Algorithm, but this is a possible CFE for the rational u/v. We consider here these two CFE's which generate together the whole set $\mathcal{H}^\star$, and we do not study exactly costs defined as in (5.1,5.2,5.3) but their "smoothed version" which takes into account the two possible CFE's: it is the average between the cost on the proper extension and the cost on the improper one. For all the costs R which are studied here, whose growth is at most moderate, it is clear that $\widetilde{R}_n - R_n$ is $O(1)$. It is then sufficient to study the smoothed version, and we shall denote by S_R the Dirichlet series relative to this smoothed version $\widetilde{R}$.

5.2. **Relating Dirichlet series to quasi-inverses of transfer operators: Digit–cost.** The fundamental fact is the existence, for all the types, of a simple relation between $S_C(2s, w)$ and the quasi-inverse $(I - \mathbf{H}_{s,w,(c)})^{-1}$ of a weighted transfer operator $\mathbf{H}_{s,w,(c)}$ (which depends on two parameters s and w).

For all types, the transfer operator $\mathbf{H}_{s,w,(c)}$ is defined via its components operators [see Figure 4]

$$\mathbf{H}_{s,w,[h],(c)}[f](x) = \delta_h^s \cdot \exp[wc(h)] \cdot |h'(x)|^s \cdot f \circ h(x) \ . \tag{5.4}$$

For any generic type, the operator $\mathbf{H}_{s,w,(c)}$ is the sum of all its component operators, while, for Markovian Type 2, the extension is made, as previously, on all the components of the coefficients of the matrix defining $\mathbf{H}_s$ in (2.7). Note that this operator is a simultaneous extension of the three operators $\mathbf{H}_1, \mathbf{H}_{1,w,(c)}, \mathbf{H}_s$ defined previously. Remark also, as previously, that the n-th iterate of this operator has a nice form; for instance, for generic types, one has

$$\mathbf{H}^n_{s,w,(c)}[f](x) = \sum_{h \in \mathcal{H}^n} \delta_h^s \cdot \exp[wc(h)] \cdot |h'(x)|^s \cdot f \circ h(x) \ . \tag{5.5}$$

Let us explain why such a relation exists between $S_C(s, w)$ and the quasi–inverse of $\mathbf{H}_{s,w}$. We recall that, for any $(u, v) \in \Omega$, there exists a unique $h \in \mathcal{H}^\star$ for which $u/v = h(\eta)$ [η is the stopping value of the gcd algorithm][3]. This entails a bijection

[3]we suppose here (for simplicity) that the initial set $\mathcal{J}$ and the final set $\mathcal{F}$ coincide with the total set $\mathcal{H}$. This has been justified at the end of Section 5.1.

between the set of coprime valid inputs Ω and the semi group $\mathcal{H}^\star$. Then, the sum which defines $S_C(s,w)$ can be completely expressed by means of LFT $h \in \mathcal{H}^\star$,

$$S_C(s,w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^{2s}} \exp[wC(u,v)] = \sum_{h\in\mathcal{H}^\star} \delta_h^s \cdot |h'(\eta)|^s \cdot \exp[wc(h)]$$

$$= \sum_{k\geq 0} \sum_{h\in\mathcal{H}^k} \delta_h^s \cdot |h'(\eta)|^s \cdot \exp[wc(h)] = \sum_{k\geq 0} \mathbf{H}^k_{s,w,(c)}[1](\eta).$$

Here, we used the alternative expression of $|(u,v)|^{-2s}$ given in (2.8, 2.11) and the fact that $C(u,v)$ equals $c(h)$, due to Equations (5.1, 4.1). Finally,

$$S_C(s,w) = (I - \mathbf{H}_{s,w,(c)})^{-1}[1](\eta). \tag{5.6}$$

An analog relation holds in Markovian cases.

When derivating with respect to w Equation (5.6), one relates the generating function $S_C^{[1]}(s)$ defined in (1.20) to the quasi-inverse $(I - \mathbf{H}_s)^{-1}$ together with its weighted version $\mathbf{H}_s^{(c)}$ defined in Figure 4 via the relation

$$S_C^{[1]}(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{(c)} \circ (I - \mathbf{H}_s)^{-1}[1](\eta). \tag{5.7}$$

For the second moment, with a supplementary derivation, one gets

$$S_C^{[2]}(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{(c^2)} \circ (I - \mathbf{H}_s)^{-1}[1](0) + \tag{5.8}$$

$$+2(I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{(c)} \circ (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{(c)} \circ (I - \mathbf{H}_s)^{-1}[1](0).$$

We observe that all these Dirichlet series have the same structure: quasi–inverses of the form $(I - \mathbf{H}_s)^{-1}$, and between these quasi-inverses, operators which involve various weights, i.e., operators of the form $\mathbf{H}_s^{(c^i)}$. Let us consider the weighting operator $W_{(c)}$ which operates on transfer operators and weights with cost $c(h)$ each component of a transfer operator. It satisfies

$$W_{(c)}\mathbf{H}_s = \mathbf{H}_s^{(c)} = \frac{\partial}{\partial w}\mathbf{H}_{s,w}|_{w=0}, \qquad W^i_{(c)}\mathbf{H}_s = W_{(c^i)}\mathbf{H}_s = \mathbf{H}_s^{(c^i)} = \frac{\partial^i}{\partial w^i}\mathbf{H}_{s,w}|_{w=0}.$$

Furthermore, we adopt shorthand notations where we omit the quasi-inverses, the zeta function, the function 1, and the point η: we only take into account the operators "between" the quasi inverses. For instance, equations (5.7, 5.8) can be re-written as [here, we omit the index (c) in the operator W]

$$S_C^{[1]} = [W], \qquad S_C^{[2]} = [W^2] + 2[W,W].$$

Any series $S_C^{[k]}(s)$ can be expressed with involved expression which contains iterates of the weighting operator W. For costs of moderate growth, we will see that the "dominant" term of the series $S^{[k]}(s)$ is

$$S_C^{[k]} \asymp [W,W,W,\ldots,W] \quad (k \text{ times}) \tag{5.9}$$

while, for costs of large growth, the "dominant term" will be

$$S_C^{[k]}(s) \asymp [W^k] \tag{5.10}$$

5.3. **Relating Dirichlet series to quasi-inverses of transfer operators: Continuants at a fraction of the depth.** We study the parameters $LX_{[\delta]}$ for $X \in \{U, V, Q\}$. As previously, we consider the Dirichlet moment generating function series $S_{LX_{[\delta]}}(s, w)$ of $LX_{[\delta]}$, defined as

$$S_{LX_{[\delta]}}(s, w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u, v)|^{2s}} |X_{\lfloor \delta P(u,v)\rfloor}|^{2w}.$$

Consider an input (u, v) of Ω on which the algorithm performs p iterations. There exists a unique LFT h of depth p such that $u/v = h(\eta)$. One can decompose h in two LFT's g and r of depth $\lfloor \delta p\rfloor$ and $p - \lfloor \delta p\rfloor$ such that $h = g \circ r$. In this case, one has

$$\delta_{g\circ r} \cdot |(g \circ r)'(\eta)| = |U_0|^{-2}, \qquad \delta_r \cdot |r'(\eta)| = |U_i|^{-2}, \qquad \delta_g \cdot |g'(0)| = |Q_i|^{-2}.$$

Ending continuants [$X = U$ or $X = V$]. Remark that the two variables U and V coincide on the set Ω of valid coprime inputs. The general term of the series $S_{LU^{[\delta]}}(s, w)$ decomposes as

$$\frac{|U_i|^{2w}}{|U_0|^{2s}} = \delta_r^{-w} \cdot |r'(\eta)|^{-w} \cdot \delta_{g\circ r}^s |(g \circ r)'(\eta)|^s = [\delta_r^{s-w} \cdot |r'(\eta)|^{s-w}] \cdot [\delta_g^s \cdot |g'(r(\eta))|^s].$$

Now, when (u, v) varies in the set of coprime inputs of Ω with a given height p, we obtain

$$\sum_{\substack{(u,v)\in\Omega \\ P(u,v)=p}} \frac{|U_i|^{2w}}{|U_0|^{2s}} = \mathbf{H}_{s-w}^{p-i} \circ \mathbf{H}_s^i[1](\eta), \tag{5.11}$$

and finally, with all depths

$$S_{LU_{[\delta]}}(s, w) = \left(\sum_{p\geq 0} \mathbf{H}_{s-w}^{p-\lfloor \delta p\rfloor} \circ \mathbf{H}_s^{\lfloor \delta p\rfloor}[1](0) \right).$$

Now, if δ is a rational of the form $\delta = c/(c + d)$, then

$$S_{LU[\delta]}(s, w) = \sum_{j=0}^{c+d-1} \mathbf{H}_{s-w}^{j-\lfloor \delta j\rfloor} \circ \left(\sum_{k\geq 0} \mathbf{H}_{s-w}^{dk} \circ \mathbf{H}_s^{ck} \right) \circ \mathbf{H}_s^{\lfloor \delta j\rfloor}[1](0). \tag{5.12}$$

The central part of the previous formula defines the so–called pseudo–quasi-inverse $\mathbb{H}_{s,w}$, namely

$$\mathbb{H}_{s,w} := \sum_{k\geq 0} \mathbf{H}_{s-w}^{dk} \circ \mathbf{H}_s^{ck}. \tag{5.13}$$

Of course, since $\mathbf{H}_s$ and $\mathbf{H}_{s,w}$ do not commute, this is not a "true" quasi-inverse. However, we study this operator when w is near to 0, and we can hope that the properties of $\mathbb{H}_{s,w}$ will be close to properties of a true quasi-inverse. We see that this is true in Section 7.

Beginning continuants [$X = Q$]. The general term of the series $S_{LQ_{[\delta]}}(s, w)$ decomposes as

$$\frac{|Q_i|^{2w}}{|U_0|^{2s}} = \delta_g^{-w} \cdot |g'(0)|^{-w} \cdot \delta_{g\circ r}^s |(g \circ r)'(\eta)|^s = [\delta_g^{s-w} \cdot |g'(0)|^{-w} \cdot |g'(r(\eta))|^s] \cdot [\delta_r^s \cdot |r'(\eta)|^s].$$

Now, when (u,v) varies in the set of all inputs of Ω with a given height p, we obtain

$$\sum_{\substack{(u,v)\in\Omega\\ P(u,v)=p}} \frac{|Q_i|^{2w}}{|U_0|^{2s}} = \widehat{\mathbf{H}}_s^{p-i} \circ \underline{\mathbf{H}}_{s,w}^{i}[1](\eta,0), \tag{5.14}$$

where the (new) operator $\widehat{\mathbf{H}}_s$ is defined via its components $\widehat{\mathbf{H}}_{s,[h]}$ in Figure 4. We consider, as previously, an integer i of the form $i = \lfloor \delta P \rfloor$, with a rational $\delta \in [0,1]$ of the form $\delta = c/(c+d)$. Thus, $P = (c+d)k + j$, with $j < c+d$, and a summation over p gives

$$S_{LQ_{[\delta]}}(s,w) = \sum_{j=0}^{c+d-1} \widehat{\mathbf{H}}_s^{j-\lfloor \delta j\rfloor} \circ \left(\sum_{k\geq 0} \widehat{\mathbf{H}}_s^{dk} \circ \underline{\mathbf{H}}_{s,w}^{ck} \right) \circ \underline{\mathbf{H}}_{s,w}^{\lfloor \delta j\rfloor}[1](\eta,0). \tag{5.15}$$

In the same vein as previously, for the ending coontinuants, there appears a pseudo-quasi-inverse $\underline{\widehat{\mathbb{H}}}_{s,w}$, defined as

$$\underline{\widehat{\mathbb{H}}}_{s,w} := \sum_{k\geq 0} \widehat{\mathbf{H}}_{s-w}^{dk} \circ \underline{\mathbf{H}}_s^{ck},$$

which can be expected to be close to a true quasi-inverse.

5.4. **Relating Dirichlet series to quasi-inverses of transfer operators: Bit–complexity.** The bit-complexity is more involved to study ... and was studied for the first time in [2], then in [108]. Remember that it is defined via costs $B, \overline{B}$ [See (5.3)]. Here, we only obtain an alternative expression for the Dirichlet expectation series

$$S_R^{[i]}(s) := \sum_{(u,v)\in\Omega} \frac{R(u,v)}{|(u,v)|^{2s}}$$

when R is equal to B or $\overline{B}$ and $i = 1, 2$.

We first deal with the elementary costs

$$\ell(m_i)\cdot |U_i|^w, \quad \ell(m_i)\cdot |Q_i|^w,$$

for some (small) w and fixed index i with $1 \leq i \leq p$. The corresponding Dirichlet generating functions are obtained by an easy modification of series involved in (5.11) or (5.14), namely

$$\sum_{p\geq i} \mathbf{H}_{s-w}^{p-i} \circ \mathbf{H}_s^{(\ell)} \circ \mathbf{H}_s^{i-1}[1](\eta), \quad \sum_{p\geq i} \widehat{\mathbf{H}}_s^{p-i-1} \circ \widehat{\mathbf{H}}_s^{(\ell)} \circ \underline{\mathbf{H}}_{s,w}^{i}[1](\eta,0),$$

where the operators $\mathbf{H}_s^{(\ell)}, \widehat{\mathbf{H}}_s^{(\ell)}$ are defined as in Figure 4, with an associated digit-cost $c := \ell$, the binary length of an integer. Now, the generating Dirichlet series of B and $\overline{B}$ are just obtained with taking the sum over all the indices i between 1 and p, and taking the derivative with respect to w (at $w = 0$). We obtain, after the first step [i.e., taking the sum over indices i]

$$(I - \mathbf{H}_{s-w})^{-1} \circ \mathbf{H}_s^{(\ell)} \circ (I - \mathbf{H}_s)^{-1}[1](\eta), \quad (I - \widehat{\mathbf{H}}_s)^{-1} \circ \widehat{\mathbf{H}}_s^{(\ell)} \circ (I - \underline{\mathbf{H}}_{s,w})^{-1}[1](\eta,0),$$

and, after the second step,

$$S_B^{[1]}(s) = (I - \mathbf{H}_s)^{-1} \circ \Delta\mathbf{H}_s \circ (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{(\ell)} \circ (I - \mathbf{H}_s)^{-1}[1](\eta) \tag{5.16}$$

$$S_{\overline{B}}^{[1]}(s) = (I - \widehat{\mathbf{H}}_s)^{-1} \circ \widehat{\mathbf{H}}_s^{(\ell)} \circ (I - \underline{\mathbf{H}}_s)^{-1} \circ \Delta\underline{\mathbf{H}}_s \circ (I - \underline{\mathbf{H}}_s)^{-1}[1](\eta,0). \tag{5.17}$$

Let us consider the derivation operator (with respect to s), denoted by Δ, which operates on transfer operators and use the similar shorthand notations as in Section 5.2, with omitting the index ℓ in the weighting operator W. Then, we have, for instance, $S_B^{[1]} = [\Delta, W]$.

For the moment of order 2, we first deal with the elementary costs

$$\ell(m_i) \cdot \ell(m_j) \cdot |U_i|^w \cdot |U_j|^t, \quad \ell(m_i) \cdot \ell(m_j) \cdot |Q_i|^w \cdot |Q_j|^t,$$

for some (small) w, t and fixed index i, j with $1 \leq i, j \leq p$, and it is easy to obtain an alternative expression for the corresponding Dirichlet series, at least for the ending continuants. If we are only interested by the "dominant" terms, i.e., the terms which bring at least four quasi–inverses, we have

$$\frac{1}{2} S_B^{[2]} \asymp 2[\Delta, \Delta, W, W] + [\Delta, W, \Delta, W] + [\Delta^2, W, W] + [\Delta, \Delta W, W] + [\Delta, \Delta, W^2].$$

6. **Average–case analysis of Euclidean Algorithms.** Here we perform an average–case analysis, and we wish to obtain the asymptotic behaviour of the mean $\mathbb{E}_N[R]$ of our main parameters, and more generally some hints on the moments $\mathbb{E}_N[R^k]$ of order k. Our strategy is as follows. We have obtained alternative expressions of $S_R(s, w)$ [from which we can obtain expressions for $S_R^{[k]}(s)$ by taking derivatives with respect to w at $w = 0$], or directly expressions for $S_R^{[k]}(s)$. The series $S_R^{[k]}(s)$ are Dirichlet generating series, and they gather all the input data $(u, v) \in \Omega$ marked with their size. We now "extract" coefficients of these Dirichlet series in order to obtain probabilistic behaviours on Ω_N.

6.1. **Particularities of the polynomial case.** In this case, both size and topology are ultrametric. The series $S_R(s, w)$ is a power series with respect to w and $z = q^{-2s}$ [denoted by $T_R(z, w)$], and the series $S_R^{[k]}(s)$ is a power series with respect to $z = q^{-2s}$, denoted by $T_R^{[k]}(z)$,

$$T_R(z, w) := \sum_{(u,v)\in\Omega} z^{L(u,v)} \exp[wR(u, v)], \qquad T_R^{[k]}(z) = \sum_{(u,v)\in\Omega} z^{L(u,v)} R^k(u, v).$$

If we denote by $[z^n]A(z)$ the coefficient of z^n inside the power series $A(z)$, the moment $\mathbb{E}_N[R^k]$ of order k, or the moment generating function $\mathbb{E}_N[wR]$ are defined by

$$\mathbb{E}_N[\exp(wR)] = \frac{[z^N]T_R(z, w)}{[z^N]T_R(z, 0)}, \qquad \mathbb{E}_N[R^k] = \frac{[z^N]T_R^{[k]}(z)}{[z^N]T_R(z, 0)},$$

and are easily extracted by methods of analytic combinatorics [33][34]. This study of Euclidean Algorithms on polynomials can be entirely done with classical generating functions, at least when a uniform probability is chosen on Ω_N. This is due to the fact that, for Type 0, both the topology and the size are ultrametric. Since all the analyses deal with power series, the extraction is simpler and can be done more precisely than in our general framework, where we deal with Dirichlet series. See [38, 57] for instance.

6.2. **Coming back to the number case.** In the number case, the series $S_R^{[k]}(s)$ are Dirichlet series which can be written as

$$S_R^{[k]}(s) = \sum_{(u,v)\in\Omega} \frac{R^k(u,v)}{|(u,v)|^{2s}} = \sum_{n\geq 1} \frac{R_n^{[k]}}{n^s} \qquad \text{with} \quad R_n^{[k]} = \sum_{\substack{(u,v)\in\Omega,\\ |(u,v)|^2=n}} R^k(u,v),$$

We wish to evaluate

$$\mathbb{E}_N[R^k] = \frac{\sum_{n=2^{2N-1}}^{2^{2N}} \phi^{[k]}(n)}{\sum_{n=2^{2N-1}}^{2^{2N}} \phi^{[0]}(n)}, \tag{6.1}$$

where, as in Section 1.7, $\phi^{[k]}(n)$ are the coefficients of the Dirichlet series

$$S_R^{[k]}(s) = \sum_{(u,v)\in\Omega} \frac{R^k(u,v)}{|(u,v)|^{2s}}.$$

Remark that, in all the cases, the series $S_R^{[0]}(s)$ are related to ζ functions associated to valid inputs of the algorithm.

It is then sufficient to obtain an asymptotic behaviour for the sum of coefficients of a Dirichlet series. This is why the following Tauberian Theorem [27] [99] will be an essential tool in this Section.

Theorem B. [Tauberian Theorem]. [Delange] *Let $F(s)$ be a Dirichlet series with non negative coefficients such that $F(s)$ converges for $\Re(s) > \sigma > 0$. Assume that*
(i) $F(s)$ is analytic on $\Re(s) = \sigma, s \neq \sigma$, and
(ii) for some $\gamma \geq 0$, one has $F(s) = A(s)(s-\sigma)^{-\gamma-1} + C(s)$, where A, C are analytic at σ, with $A(\sigma) \neq 0$.
Then, as $K \to \infty$,

$$\sum_{n\leq K} a_n = \frac{A(\sigma)}{\sigma\Gamma(\gamma+1)}\, K^\sigma \log^\gamma K \quad [1+\epsilon(K)\,], \quad \epsilon(K)\to 0.$$

We will see that such a Theorem will be easy to use in our framework. In particular, the sums which appear in the numerator and denominator of $\mathbb{E}_N[R^k]$ [see (6.1)] can be easily evaluated with this Theorem, via the estimates

$$\sum_{n=2^{2N-1}}^{2^{2N}} a_n = C(\sigma,\gamma)\cdot 2^{2N\sigma}\, N^\gamma \cdot\, [1+\epsilon(N)\,], \qquad \epsilon(N)\to 0,$$

$$\text{with} \quad C(\sigma,\gamma) := \frac{A(\sigma)}{\sigma\Gamma(\gamma+1)}\,(1-2^{-\sigma})\,(2\log 2)^\gamma.$$

However, the Tauberian Theorem does not provide any remainder term, and this is why it will be no longer useful for distributional analyses. This is also a main difference with the study of the polynomial case, where use of power series easily allows to obtain remainder terms.

6.3. **Functional analysis.** We first describe a set of conditions, and we will prove that these conditions are sufficient to entail that hypotheses of Tauberian Theorem are valid.

Conditions (B)**.** *There exists a functional space $\mathcal{F}$ where some operator $\mathbf{G}_s$ satisfies the following:*

$[\mathrm{An}_s(s,0)]$. *The operator $\mathbf{G}_s$ acts on $\mathcal{F}$ when s satisfies $\Re s > \sigma$ with $\sigma < 1$ and the map $s \mapsto \mathbf{G}_s$ is analytic on $\Re s > \sigma$.*

[UDE and SG]. *The density transformer* $\mathbf{G} := \mathbf{G}_1$ *has a unique dominant eigenvalue* $\lambda = 1$, *and a spectral gap: the rest of the spectrum lies in a disk of radius* < 1.
[SM]. *The spectral radius* $R(s)$ *of* $\mathbf{G}_s$ *is strictly less than 1 on* $\Re s = 1$, *except at* $s = 1$.

Theorem BB. *Suppose that Conditions* (B) *hold for a transfer operator* $\mathbf{G}_s$. *Denote by* $\mathcal{A}$ *the set of operators* A *that act on transfer operators, and for which the operator* $A\mathbf{G}$ *acts on the Banach space* $L^1(I)$. *Then the hypotheses of Theorem B [Tauberian Theorem] are fulfilled for all the Dirichlet Series* $F(s)$ *denoted by an expression* $[A_1, A_2, \dots A_k]$ *where each* A_i *belongs to* $\mathcal{A}$. *More precisely,* $F(s)$ *has a pôle of order* $k+1$ *at* $s = 1$, *and it admits, at* $s = 1$, *an expansion of the form*

$$F(s) = \frac{a_0}{(s-1)^{k+1}} + \frac{a_1}{(s-1)^k} + \dots .$$

Moreover, the "dominant" coefficient a_0 *can be expressed as*

$$a_0 = \frac{1}{\log 2} \cdot \frac{1}{\lambda'(1)^{k+1}} \cdot \prod_{i=1}^{k} I[A_i \mathbf{G}] \qquad \textit{where} \quad I[\mathbf{H}] := \int_I \mathbf{H}[\psi](t)dt \tag{6.2}$$

involves the dominant eigenfunction ψ *of the operator* $\mathbf{G}$ *and* $\lambda'(1)$ *is the derivative of the dominant eigenvalue* $\lambda(s)$ *at* $s = 1$.

Then, the dominant constant a_0 depends only on the subset $\{A_1, A_2, \dots A_k\}$ and does not depend on the order of the sequence $(A_1, A_2, \dots, A_k)$. Moreover, if $\mathbf{G}_{s,w}$ is a weighted transfer operator relative to a dynamical system (X, V) and a digit-cost of moderate growth, the integrals $I[A\mathbf{G}]$, for $A := W_{[c]}$ or $A := \Delta$, admit alternative expressions

$$I[\Delta\mathbf{G}] = \int_X \Delta\mathbf{G}[\psi](t)dt = \lambda'_s(1,0), \quad I[W_{[c]}\mathbf{G}] = \sum_{h\in\mathcal{H}} c(h) \cdot \int_{h(X)} \psi(t)dt = \lambda'_w(1,0),$$

which involve the derivatives of the dominant eigenvalue $\lambda(s, w)$ at $(1, 0)$. [We have already considered such expressions in Equations (2.15, 2.16)].

Proof. Elementary perturbation theory [53] implies that $\mathbf{G}_s$ inherits the dominant eigenvalue property and the spectral gap when s is (complex) near 1, so that $\mathbf{G}_s$ decomposes as $\mathbf{G}_s = \lambda(s)\cdot\mathbf{P}_s + \mathbf{N}_s$, where $\lambda(s)$ is the dominant eigenvalue of $\mathbf{G}_s$, $\mathbf{P}_s$ is the projection on the dominant eigensubspace, and $\mathbf{N}_s$ is the operator relative to the remainder of the spectrum. The previous decomposition extends to any power $\mathbf{G}_s^n = \lambda(s)^n \cdot \mathbf{P}_s + \mathbf{N}_s^n$, and to the quasi–inverse

$$(I - \mathbf{G}_s)^{-1} = \lambda(s)\frac{\mathbf{P}_s}{1-\lambda(s)} + (I - \mathbf{N}_s)^{-1}.$$

Since $\mathbf{G}_1$ has a dominant eigenvalue equal to 1, its dominant eigenfunction is the invariant density denoted by ψ. Finally, for any strictly positive function f, one has

$$(I - \mathbf{G}_s)^{-1}[f](x) \sim \frac{-1}{\lambda'(1)}\psi(x) \cdot \int_X f(t)dt \qquad \text{when } s \to 1.$$

Finally, hypothesis (ii) of Tauberian Theorem is satisfied at $s = 1$ for any Dirichlet series of the form $[A_1, A_2, \dots A_k]$ with $\gamma = k$, and the dominant constant a_0 has the described form.

On the other side, Condition *SM* proves that hypothesis (i) of Tauberian Theorem is satisfied. □

6.4. **Number of steps.** [105, 106, 107, 108]. For algorithms of the Fast Class [we recall that the Fast Class is the union of the Good Class and the Difficult Class], the operator $\mathbf{H}_s$ satisfies Conditions (B). [See Section 8]. Then, Relation (5.7) applied to the particular case $c = 1$,

$$S^{[1]}(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s \circ (I - \mathbf{H}_s)^{-1}[1](\eta)$$

entails that Tauberian Theorem can be applied at $s = 1$ with an exponent $\gamma = 1$ for the expectation $\mathbb{E}_N[P]$. More generally, Relation (5.9) entails that Tauberian Theorem can be applied at $s = 1$ and $\gamma = k$ for the moment $\mathbb{E}_N[P^k]$ of order k. Theorem (BB) proves that the moment of order k is asymptotic to the k-th power of the mean value. In particular, the variance is of order $o(N)$.

For algorithms of the Bad Class, the transfer operator $\widetilde{\mathbf{H}}_s$ of the induced dynamical system [see Section 2.6] satisfies Conditions (B) [see Section 8 for a proof]. The operator $\mathbf{H}_s$ can be viewed as the operator $\widetilde{\mathbf{H}}_s^{(c_0)}$ relative to the cost $c_0(d) = d$. Since c_0 is a cost of large growth, $\mathbf{H}_s$ brings a supplementary pôle of order 1 at $s = 1$. Then, Relation (5.7) applied to this particular case

$$S^{[1]}(s) = (I - \widetilde{\mathbf{H}}_s)^{-1} \circ \widetilde{\mathbf{H}}_s^{(c_0)} \circ (I - \widetilde{\mathbf{H}}_s)^{-1}[1](\eta)$$

entails that Tauberian Theorem can be applied at $s = 1$ with an exponent $\gamma = 2$ for the expectation $\mathbb{E}_N[P]$. Now, for the moment $\mathbb{E}_N[P^k]$ of order k, the operator $\widetilde{\mathbf{H}}_s^{(c_0^k)}$ has a dominant pôle at $s = k$. Then, Relation (5.10) shows that the moment $\mathbb{E}_N[P^k]$ of order k is of exponential order.

Theorem 3. [Number of steps.] *For any algorithm of the Fast Class, the expectation $\mathbb{E}_N[P]$ of the number P of steps on the valid inputs of size N is asymptotically linear with respect to size N,*

$$\mathbb{E}_N[P] \sim \widehat{\mu} \cdot N, \qquad \text{where} \quad \widehat{\mu} := \frac{2 \log 2}{\alpha}$$

where α is the entropy of the dynamical system. The standard deviation is $o(N)$, and, consequently the random variable P satisfies the concentration of distribution property.
Any Euclidean Algorithm associated to a dynamical system of the Bad Class performs an average number of steps on valid rationals of size N that is quadratic with respect to size N,

$$\mathbb{E}_N[P] \sim \frac{\log^2 2}{\overline{\zeta}(2)} N^2,$$

where $\overline{\zeta}(s)$ is the Zeta function relative to valid inputs. For any integer $k \geq 2$, the k–th moment of total number of steps is of exponential type

$$\mathbb{E}_N[P^k] = \Theta[2^{N(k-1)}].$$

In particular the standard deviation is $\Theta(2^{N/2})$.

6.5. **Digit–costs of moderate growth.** [108] For algorithms of the Fast Class, the operator $\mathbf{H}_s$ satisfies Conditions (B). Moreover, for a cost c of moderate growth, the operator $\mathbf{H}_s^{(c)}$ defined in Figure 4 is analytic on $\Re s \geq 1$.[see Section 8]

Theorem 4. [Total cost.] *Consider any algorithm of the Fast Class, together with a cost c of moderate growth. The expectation $\mathbb{E}_N[C]$ of the total cost C on*

the valid inputs of sire N is asymptotically linear with respect to size N,

$$\mathbb{E}_N[C] \sim \widehat{\mu}(c) \cdot N, \qquad \textit{with} \quad \widehat{\mu}(c) := \frac{2\log 2}{\alpha} \cdot \mu(c), \qquad \mathbb{E}_N[C^k] \sim (\mathbb{E}_N[C])^k$$

where α is the entropy of the dynamical system, and $\mu(c)$ the average value of cost c with respect to the density defined by the invariant function ψ [see (2.15, 2.16)]. The standard deviation is $o(N)$, and, consequently the random variable expressing the total cost C satisfies the concentration of distribution property.

It is possible to obtain a more general theorem for costs of large growth and/or algorithms of Bad Class. (See [108]).

Remark. Comparing Theorems 1, 3 and 4 shows that, for any algorithm of the Fast class, and any cost c of moderate growth, one has $\widehat{\mu}(c) = \widehat{\mu} \cdot \mu(c)$, so that

$$\mathbb{E}_N[C] \sim \mu(c) \cdot \mathbb{E}_N[P], \qquad \frac{\mathbb{E}_N[C]}{\mathbb{E}_N[P]} \sim \frac{\mathbb{E}[C_n]}{n}. \tag{6.3}$$

This proves that executions of Euclidean Algorithms [which correspond to rational trajectories] in the Fast Class behave on average similarly to the way truncated real trajectories behave on average.

6.6. **Continuants at a fraction of the depth.** Here, our starting point are Relations (5.12, 5.15). For $X \in \{U, V, Q\}$, the Dirichlet series $S^{[1]}_{LX_{[\delta]}}(s)$ is obtained by taking the derivative (with respect to w) of $S_{LX_{[\delta]}}(s, w)$ at $w = 0$. For $X \in \{U, V\}$, and for Algorithms of the Fast Class, it has a dominant pole at $s = 1$ of order 2. Then, Tauberian Theorem can be applied at $s = 1$ with $\gamma = 1$. The series $S^{[k]}_{LU_{[\delta]}}(s)$ relative to moments of order k has a dominant pole at $s = 1$ of order $k+1$. This analysis appears under a different form in [24].

For $X = Q$, the situation is more involved. For algorithms of the Good Class, the operator $\underline{\mathbf{H}}_s$ has good dominant spectral properties, closely related to these of the operator $\mathbf{H}_s$; this is due, in particular, to the property $(P2)$ of Figure 2, which was called "Bounded Distortion Property" . Then, the analyses for the beginning and ending continuants are similar. For the algorithms of the Difficult Class, the behaviour of the operator $\underline{\mathbf{H}}_s$ may be quite involved, and the behaviour of the beginning continuants is more difficult to study. It is not the same a priori as for the ending continuants.

Theorem 5. *For any algorithm of the Fast Class, the expectation of the size of the (ending) continuant at a fraction δ on the set of valid inputs with size N is asymptotically of the form*

$$\mathbb{E}_N[LU_{[\delta]}] \sim (1 - \delta) \cdot N.$$

The standard deviation is $o(N)$. Consequently the random variable $LU_{[\delta]}$ satisfies the concentration of distribution property.

For any algorithm of the Good Class, the expectation of the size of the (beginning) continuant at a fraction δ on the set of valid inputs with size N is asymptotically of the form

$$\mathbb{E}_N[LQ_{[\delta]}] \sim \delta \cdot N.$$

The standard deviation is $o(N)$. Consequently the random variable $\ell_{[\delta]}$ satisfies the concentration of distribution property.

Remark. Comparing Theorems 2, 3 and 5 shows that, for any algorithm of the Good Class, the behaviour of beginning continuants is the same [on average] for rationals and for reals, one has

$$\mathbb{E}_N[LQ_{[\delta]}] \sim \frac{\alpha}{2} \cdot \mathbb{E}_N[\delta P], \qquad \frac{\mathbb{E}_N[LQ_{[\delta]}]}{\mathbb{E}_N[\delta P]} \sim \frac{\mathbb{E}[\ell_n]}{n}. \tag{6.4}$$

6.7. **Bit–complexity.** Here, our starting point are Relations (5.16, 5.17). For algorithms of the Fast Class, the operator $\mathbf{H}_s$ satisfies Conditions (B). Moreover, since the size–cost ℓ is of moderate growth, the operator $\mathbf{H}_s^{(\ell)}$ defined in Figure 4 is analytic on $\Re s \geq 1$. Then Tauberian Theorem can be applied to Dirichlet series $S_B^{[1]}(s)$, at $s = 1$ and $\gamma = 2$. This analysis can be found in [2], [108]. Theorem (BB) also proves that the moment of order two is asymptotic to the square of the mean. For the Bad Class, the induced system satisfies Condition (B), while the operator $\mathbf{H}_s^{(\ell)}$ brings a supplementary pole at $s = 1$. See [108] for more details.

Theorem 6. *For any algorithm of the Fast Class, the average bit-complexity of the algorithm on the set of valid inputs with size N is asymptotically of quadratic order*

$$\mathbb{E}_N[B] \sim \frac{\log 2}{\alpha} \cdot \mu(\ell) \cdot N^2, \qquad \mathbb{E}_N[B^2] \sim (\mathbb{E}_N[B])^2 .$$

Here α is the entropy of the dynamical system relative to the algorithm and $\mu(\ell)$ denotes the average value of digit–size ℓ defined in (2.16). The standard deviation is $o(N^4)$. Consequently the random variables B satisfy the concentration of distribution property.
For any algorithm of the Good Class, the same is true for the variable $\overline{B}$.
For any Algorithm of the Bad Class, the average bit-complexity on the set of valid inputs with size N is asymptotically of order three

$$\mathbb{E}_N[B] \sim \mathbb{E}_N[\overline{B}] = \Theta(N^3)$$

For any integer $k \geq 2$, the k–th moment of bit–complexity is of exponential type.

7. **Main results: Distribution Analysis of Euclidean Algorithms of the Good Class.** The previous dynamical approach exhibits two main facts for algorithms of the Good Class.

(i) First, the various parameters of interest –additive costs, size of continuants at a fraction of the depth, bit complexity– satisfy the concentration property, i.e., $\mathbb{E}_N[R^k] \sim (\mathbb{E}_N[R])^k$. Then, we already know that the variance $\mathbb{V}_N[R]$ is $o(\mathbb{E}_N[R^2])$. However, since Tauberian Theorems do not provide remainder terms, we did not obtain a precise asymptotic behaviour for the variance. Now, we are interested in obtaining such a result for all our parameters of interest.

(ii) Second, truncated real trajectories exhibit Gaussian behaviour [Section 4] and executions of Euclidean Algorithms [which correspond to rational trajectories] in the Fast Class behave on average similarly to the way truncated real trajectories behave on average. [See remarks at the end of Sections 6.5 and 6.6, together with Equations (6.3,6.4)]. It is then natural to ask whether this analogy extends to distributions: Is it true that the distribution of the total cost $C(u, v)$ on an input (u, v) of size N is asymptotically Gaussian (when N tends to ∞)? Is it true that the size $LU_{[\delta]}$ of the ending continuant at a fraction of the depth of a input (u, v)

of size N is asymptotically Gaussian (when N tends to ∞)? Is it true that the bit-complexity follows an asymptotic gaussian law? How to compare the distribution of some parameter R on truncated real trajectories and on rational trajectories?

We will provide a precise answer to all these questions for all the algorithms of the Good Class and our three parameters of interest, the total cost relative to a digit–cost of moderate growth, the size of the ending continuant U at a fraction δ of the depth, the bit–complexity. These results appeared in [7][5] for the total cost C and Good algorithms of Type 1. They are generalized here to all the Algorithms of the Good Class [even those of Type 2]. Results about continuants and bit–complexity can be found in [69].

Note that the distributional analysis in polynomial case is relatively easy to deal with: Since we work with powers series, Cauchy Formula is used on compact domains instead of [not compact] strips, and uniformity estimates are easier to obtain [see [57]].

Our method is not the same for the three parameters. For $R = C$ and $R = LU_{[\delta]}$, we use a dynamical approach. In the study of bit-complexity, we adopt an indirect approach, where we apply the previous results.

We first work in Ω_N^+ ,$\widetilde{\Omega}_N^+$ defined as

$$\Omega_N^+ := \bigcup_{M\le N} \Omega_M, \qquad \widetilde{\Omega}_N^+ := \bigcup_{M\le N} \widetilde{\Omega}_M,$$

$$\Omega_N^+ := \{(u,v) \in \Omega \ \ ; L(u,v) \le N\}, \qquad \widetilde{\Omega}_N^+ := \{(u,v) \in \widetilde{\Omega} \ \ ; L(u,v) \le N\}.$$

For the first two parameters, we perform a distributional analysis, and our strategy consists to mainly use the two expressions that relate the Dirichlet moment generating series $S_C(s,w)$ or $S_{LU_{[\delta]}}(s,w)$ to the operators $\mathbf{H}_{s,w}, \mathbf{H}_{s-w}$ [See (5.6, 5.12]. and we recall that, for a general cost R, coefficients of series $S_R(s,w)$ are closely related to the moment generating functions $\mathbb{E}_N^+[\exp(wR)]$

$$\mathbb{E}_N^+[\exp(wR)] := \frac{\Phi_w^+(N)}{\Phi_0^+(N)}\,, \tag{7.1}$$

where $\Phi_w^+(N)$ is the cumulative value of $\exp(wR)$ on Ω_N^+, i.e., $\Phi_0^+(N) = |\Omega_N^+|$, and

$$\Phi_w^+(N) := \sum_{(u,v)\in\Omega_N^+} \exp[wR(u,v)] = \sum_{\substack{(u,v)\in\Omega\\ L(u,v)\le N}} \exp[wR(u,v)]. \tag{7.2}$$

If we succeed in expressing these MGF's as uniform quasi-powers, we can apply the Quasi-Powers Theorem [our Theorem A of Section 4] and obtain a limit gaussian law.

If we only wish to obtain a precise information on $\mathbb{E}_N^+[R^k]$, we recall that

$$\mathbb{E}_N^+[R^k] := \frac{\Phi_{[k]}^+(N)}{\Phi_{[0]}^+(N)}\,, \tag{7.3}$$

where $\Phi_{[k]}^+(N)$ is the cumulative value of R^k on Ω_N^+,

$$\Phi_{[k]}^+(N) := \sum_{(u,v)\in\Omega_N^+} R^k(u,v) = \sum_{\substack{(u,v)\in\Omega\\ L(u,v)\le N}} R^k(u,v)\,, \quad \Phi_{[0]}^+(N) = |\Omega_N^+|\,. \tag{7.4}$$

7.1. **Perron's Formula.** Tauberian theorems are now insufficient for extracting coefficients from a Dirichlet series, since they do not provide remainder terms. We need a more precise "extractor" of coefficients, and the Perron formula [of order 2] is well-suited to this purpose. The Perron Formula of order two (see [31]) is valid for a Dirichlet series $F(s) = \sum_{n\geq 1} a_n n^{-s}$ and a vertical line $\Re s = D > 0$ inside the domain of convergence of F.

$$\Psi(T) := \sum_{n\leq T} a_n (T-n) = \frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} F(s) \frac{T^{s+1}}{s(s+1)}\, ds\,,$$

Distributional analysis. Applying it to the Dirichlet series $S(s,w) = \sum_n \phi_w(n) \cdot n^{-s}$, we find

$$\Psi_w(T) := \sum_{n\leq T} \phi_w(n) \cdot (T-n) = \frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} S(2s,w) \frac{T^{2s+1}}{s(s+1)}\, ds\,. \tag{7.5}$$

Thus, Perron's formula gives us information on $\Psi_w(2^N)$, which is just a Cesaro sum of the $\Phi^+_w(Q)$ [defined in (7.2)]

$$\Psi_w(2^N) = \sum_{Q<N} \sum_{n;\ell(n)\leq Q} \phi_w(n) = \sum_{Q<N} \Phi^+_w(Q)\,.$$

We shall explain later (in Section 7.5) how to transfer information from Ψ_w to Φ^+_w.

Study of the moments of higher order. We apply the Perron Formula to Dirichlet series $S^{[k]}(s) = \sum_n \phi^{[k]}(n) \cdot n^{-s}$, and we find

$$\Psi_{[k]}(T) := \sum_{n\leq T} c^{(k)}_n (T-n) = \frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} S^{(k)}(s) \frac{T^{2s+1}}{s(s+1)}\, ds\,. \tag{7.6}$$

Thus, Perron's formula gives us information on $\Psi_{[k]}(2^N)$, which is just a Cesaro sum of the $\Phi^+_{[k]}(Q)$ [defined in (7.4)]

$$\Psi_{[k]}(2^N) = \sum_{Q<N} \Phi^+_{[k]}(Q)\,.$$

7.2. ***US* Properties.** We first discuss the choice of D. For all the parameters of interest $[R = C, R = LU_{[\delta]}, R = B]$, the Dirichlet series $S^{[k]}_R(s)$ have a singularity at $s = 1$, which is in fact a pôle. In the same vein, for $R = C$ or $R = LU_{[\delta]}$, the Dirichlet series $S_R(s,w)$ has a singularity at $s = \sigma_R(w)$, which is in fact a pôle:

– In the case of a digit cost of moderate cost, there is a unique value $\sigma(w) = \sigma_C(w)$ of s near 1 for which the dominant eigenvalue $\lambda(s,w)$ of $\mathbf{H}_{s,w}$ equals 1.

– In the case of ending continuants, there is a unique value $\sigma_{LU_{[\delta]}}(w)$ of s near 1 for which the dominant eigenvalue $\lambda(s)$ of $\mathbf{H}_s$ satisfies $\lambda(s-w)^d \lambda(s)^c = 1$.

These assertions are clear consequences of the Implicit Function Theorem [see Property 7 of Section 8.5] which defines an analytic function $\sigma_R(w)$ near 0 which satisfies $\sigma_R(0) = 1$.

It is next natural to modify the integration contour $\Re s = D$ into a contour containing $\sigma_R(w)$ as a unique pole of $S_R(2s,w)$. and it is thus useful to know that the following Properties *US* [Uniform Estimates on Strips] hold. In fact, we consider two properties *US*, the first one $US(s)$ is adapted for univariate Dirichlet series, whereas the second property $US(s,w)$ is convenient for bivariate series $S(s,w)$.

Property *US*(s) There is $\alpha > 0$ for which the following is true:

(*i*) $S(s)$ admits $s=1$ as a unique pole in the strip $|\Re s-1|\leq\alpha$.
(*ii*) On the left vertical line $\Re s=1-\alpha$, the Dirichlet series $S(s))$ is $O(|\Im s|^{\xi})$, with a small ξ,

Property *US*(s,w) There is $\alpha>0$ and a neighborhood $\mathcal{W}$ of 0 for which the following is true:

(*i*) for all $w\in\mathcal{W}$, $S_R(s,w)$ admits $s=\sigma_R(w)$ as a unique pole in the strip $|\Re s-1|\leq\alpha$.
(*ii*) On the left vertical line $\Re s=1-\alpha$, the Dirichlet series $S_R(s,w)$ is $O(|\Im s|^{\xi})$, with a small ξ, and a uniform O-term (with respect to w).

With the *US*(s) Property, it is possible to control the integral (7.6) on the left vertical line $\Re s=1-\alpha$, and spectral properties of $\mathbf{H}_s$ inherited from $\mathbf{H}_1$ give the desired expansion for $\Psi_{[k]}(2^N)$.

With *US*(s,w) Property, it is possible to control the integral (7.5) on the left vertical line $\Re s=1-\alpha$, and spectral properties of $\mathbf{H}_{s,w}$ or $\mathbf{H}_{s-w}$ inherited from $\mathbf{H}_1$ give the desired uniform quasi-power expansion for $\Psi_w(2^N)$.

It is important to remark that the first assertion of Condition $US(s)$ does not always hold, even in simpler cases, when the series $S(s)$ is just the quasi inverse $(I-\mathbf{H}_s)^{-1}[1](\eta)$. Consider now the case of a dynamical system of the unit interval $[0,1]$ with two affine complete branches of respective slopes p and q $[p+q=1]$. There are two cases:

(a) if $\log p/\log q\in\mathbb{Q}$, then there are infinitely many poles of $(I-\mathbf{H}_s)^{-1}$ on the line $\Re s=1$, and this set of poles is of the form $1+i\mathbb{Z}a$ for some non zero real number a.

(b) if $\log p/\log q\notin\mathbb{Q}$, there is a unique pole of $(I-\mathbf{H}_s)^{-1}$ on the line $\Re s=1$, at $s=1$. However, there is an accumulation of poles on the left of the line $\Re s=1$.

Then Condition *US* (s) is never satisfied for Complete Dynamical Systems with two affine branches. On the otherhand, the second assertion of Condition $US(s)$ is often very difficult to obtain; such a property for the Riemann ζ function is closely related to the Prime Numbers Theorem (see [31]).

7.3. ***UNI* Conditions.** If we wish Conditions *US* to be true, we have then to exclude dynamical systems with affine branches, or systems which are "like" dynamical systems with affine branches. This is why Dolgopyat introduces a "distance" Δ between two inverse branches h et k of same depth,

$$\Delta(h,k)=\inf_{x\in\mathcal{I}}|\Psi'_{h,k}(x)|,\qquad\text{with}\quad\Psi_{h,k}(x)=\log\frac{|h'(x)|}{|k'(x)|}\tag{7.7}$$

and he asks this distance not to behave as in dynamical systems which are $\mathcal{C}^2$ conjugated to a system with affine branches. What kind of properties does this distance $\Delta(h,k)$ satisfy in the case of a system which is $\mathcal{C}^2$ conjugated to a system with affine branches? In this case, there exists $f>0$ of class $\mathcal{C}^1$ such that, for any n, and for any $h\in\mathcal{H}^n$, there is a constant $d(h)$ for which $|h'(x)|f\circ h(x)=d(h)f(x)$ for any $x\in X$. Then, taking the logarithm, differentiating, and putting $\widehat{f}:=\log f$, we get

$$\Psi'_{h,k}(x)=h'(x)\cdot\widehat{f}'\circ h(x)-k'(x)\cdot\widehat{f}'\circ k(x).$$

Then, there exists $A>0$, for which $\Delta(h,k)$ satisfies

$$\Delta(h,k)\leq A\rho^n,\qquad\forall n,\forall h\in\mathcal{H}^n,\forall k\in\mathcal{H}^n,$$

where ρ is the contraction ratio defined in Figure 8 which satisfies $\rho < 1$ for all the algorithms of the Good Class [see Property $(P1)$ of Section 2.5].

Conditions UNI [introduced by Dolgopyat] express that the behaviour of the distance $\Delta(h,k)$ must be not the same as for systems (X,V) which are $\mathcal{C}^2$ conjugated to systems with affine branches: the inverse branches of the Dynamical system are required to have not all "the same form", [i.e., their derivatives must be "not too often too close" (with respect to Δ)]. There are two different conditions UNI, the Weak Condition UNI, and the strong Condition UNI.

Weak Condition *UNI. There exists some $\eta > 0$ and an integer n_0 such that, for any integer $n \geq n_0$, there are two elements $h, k \in \mathcal{H}^n$ for which $\Delta(h,k) \geq \eta$.*

Strong Condition *UNI. For any $a, 0 < a < 1$, the probability that two inverse branches h, k of $\mathcal{H}^n$ satisfy $\Delta(h,k) \leq \rho^{an}$ is $O(\rho^{an})$, with a uniform O–term (with respect to n, a).*

More precisely, for h in $\mathcal{H}^n$, and $\eta > 0$, we denote by $J(h,\eta)$ the union of the fundamental intervals $k(\mathcal{I})$, where $k \in \mathcal{H}^n$ ranges over the Δ-ball of center h and radius η,

$$J(h,\eta) := \bigcup_{k\in\mathcal{H}^n, \Delta(h,k)\leq\eta} k(\mathcal{I}),$$

Condition *UNI* expresses that, for any $a, 0 < a < 1$, the Lebesgue measure of $J(h,\rho^{an})$ (for $h \in \mathcal{H}^n$) is $O(\rho^{an})$, with a uniform O–term (with respect to h, n, a).

All the Dynamical Systems of the Good Class satisfy Condition *UNI*: Good Algorithms of Type 1 satisfy the Strong Condition *UNI* [This is due to the good properties of their "dual" systems. See [7] for a proof]. It is also easy to see that Good Algorithms of Type 2 satisfy the Weak Condition *UNI*.

7.4. ***UNI* Conditions imply *US* Property.** First, we have previously seen that a system which satisfies Condition *UNI* is not $\mathcal{C}^2$ conjugated to a dynamical system with affine branches. In fact, Dolgopyat [29] proved that the *UNI* Condition is sufficient to also imply that the quasi-inverse of the transfer operator satisfies $US(s)$ –at least, in the case of one–variable transfer operators $\mathbf{H}_s$ which are related to dynamical systems with a finite number of branches. Then, Baladi and Vallée have adapted and generalized Dolgopyat's result. In fact, there are two different results. The first result [7, 5] shows that the Strong Condition *UNI* implies the *US*(s,w) Property for the quasi–inverse of the transfer operator, even for dynamical systems with an infinite number of branches, and for weighted transfer operators relative to costs of moderate growth. The second result [6], whose proof is more involved and less natural, shows that the Weak Condition *UNI* implies the *US*(s) Property for the quasi–inverse of the transfer operator, even for dynamical systems with an infinite number of branches. It can be generalized in order to prove that Weak Condition *UNI* also implies the *US*(s,w) Condition for weighted transfer operators relative to costs of moderate growth. These various proofs are closely based on estimates on the following type which generalize analogue bounds due to Dolgopyat:

Theorem. [Dolgopyat-type estimates]. *Consider a Good Euclidean Dynamical System, with contraction ratio ρ, and let $\mathbf{H}_s$, $\mathbf{H}_{s,w}$ be its transfer operator and its weighted transfer operator [relative to a cost of moderate growth]. For any $\xi > 0$, there is a real neighborhood $\Sigma_1 \times W_1$ of $(1,0)$, and there are two constants $M > 0$*

and $\gamma < 1$, such that, for all $n \geq 1$, for all $s = \sigma + it$, $w = \nu + i\tau$ with $(\sigma, \nu) \in \Sigma_1 \times W_1$ and $|t| \geq 1/\rho^2$,

$$||\mathbf{H}^n_{s-w}||_{1,t} \leq M \cdot |t|^\xi \cdot \gamma^n, \qquad ||\mathbf{H}^n_{s,w}||_{1,t} \leq M \cdot |t|^\xi \cdot \gamma^n. \tag{7.8}$$

[Here the norm $||.||_{1,t}$ is defined by $||f||_{1,t} := \sup|f| + (1/t)\sup|f'|$.

A work in progress (by Lhote and Vallée) seems to prove that (7.8) also holds for the underlined operator $\underline{\mathbf{H}}_{s,w}$ (in the case of the Good Class), with a norm $||\cdot||_{1,t}$ adapted to functions F of two variables.

7.5. **Applying Perron's Formula [distributional analysis].** Perron's Formula (7.5) combined with fundamental relations (5.6, 5.12, 5.15), together with Property *US*(s, w) will provide the following estimate for the Cesaro sum Ψ_w of Φ^+_w, as $T \to \infty$,

$$\Psi_w(T) := \sum_{n \leq T} c_n(w)(T-n) = \frac{E_R(w)}{\sigma_R(w)(2\sigma_R(w)+1)}\, T^{2\sigma_R(w)+1}\left[1+O(T^{-2\tau})\right], \tag{7.9}$$

where $E_R(w)$ is the residue of $S_R(s, w)$ at the pole $s = \sigma_R(w)$, τ is some positive constant, and the O–term is uniform on $\mathcal{W}$ when $T \to \infty$. Note that σ_R and E_R are analytic on a complex neighborhood of $w = 0$.

7.6. **Asymptotic normality and Quasi-Power estimates.** It does not seem easy to transfer the information (7.9) on $\Psi_w(T)$ to estimates on $\Phi^+_w(T)$; we proceed in three steps to prove asymptotic normality of parameter R for $R = C$ or $R = LX_{[\delta]}$:

First Step. We introduced a smoothed model: we associate to function $\epsilon(T) = T^{-2\tau}$ the probabilistic models $(\overline{\Omega}^+_N(\epsilon), \overline{\mathbb{P}}^+_N(\epsilon))$ as follows: For any integer N, set $\overline{\Omega}^+_N(\epsilon) = \Omega^+_N$; next, choose uniformly an integer Q between $N - \lfloor N\epsilon(N)\rfloor$ and N, and draw uniformly an element (u, v) of Ω^+_Q. Slightly abusing language, we refer to the function R in the model $(\overline{\Omega}^+_N(\epsilon), \overline{\mathbb{P}}^+_N(\epsilon))$ as the "smoothed parameter". Now, we appeal to a classical result that is often used in number theory contexts, and we then deduce from (7.9) the following quasi-power estimates for the moment generating function of the "smoothed" version of the parameter R,

$$\frac{\overline{\mathbb{E}}^+_N[\exp(wR)]}{[1+O(2^{-\tau N})]} = \exp\left(2\log 2[\sigma_R(w) - \sigma_R(0)]N + \log\frac{E_R(w)}{E_R(0)\sigma_R(w)}\right), \tag{7.10}$$

where the O-term is uniform in w.

Second Step. Now, we are again in the framework of Theorem A of Section 4 [the Quasi-Powers Theorem] , and we get that the smoothed version of cost R follows an asymptotic Gaussian distribution, with a speed of convergence in $O(1/\sqrt{N})$, together with precise informations about the asymptotic behavior of the expectation $\overline{\mathbb{E}}^+_N[R]$ and the variance $\overline{\mathbb{V}}^+_N[R]$. The function $U(w)$ of the Quasi-Powers Theorem [See Section 3] is

$$U(w) = 2\log 2[\sigma_R(w) - \sigma_R(0)]$$

and the constants of the expectation and variance involve the first two derivatives of U which can be computed with the Implicit function Theorem and the definition of σ_R. [See the beginning of Section 6.2].

Third Step. We prove that the distributions of R on $\overline{\Omega}_N^+(\epsilon)$ and on Ω_N^+ are $\epsilon(N)$–close, i.e.,

$$|\,\mathbb{P}_N^+(u,v) - \overline{\mathbb{P}}_N^+(u,v)| = O(\epsilon(N))$$

so that the distribution of R on Ω_N is also asymptotically Gaussian, with a speed of convergence in $O(1/\sqrt{N})$. The closeness of distributions, together with the worst–case polynomial complexity of the algorithms of Good Class also provides precise information about the asymptotic behavior of the expectation $\mathbb{E}_N^+[R]$ and the variance $\mathbb{V}_N^+[R]$. Finally, it is possible to return in $(\Omega_N, \mathbb{P}_N)$.

7.7. **Distributional analysis of total cost.** The first result of this Section proves that, for any algorithm of the Good Class, and any digit–cost of moderate growth, the total cost on Ω_N follows an asymptotic Gaussian, with an optimal speed of convergence. [See [7] or [5]]

Theorem 7. [Central Limit Theorem for total cost.] *For a Euclidean algorithm of the Good Class, and any cost c of moderate growth,*
(a) The distribution of the total cost C on Ω_N is asymptotically Gaussian, with speed of convergence $O(1/\sqrt{N})$, i.e., there exist two constants $\widehat{\mu}(c) > 0$ and $\widehat{\rho}(c) > 0$ such that, for any N, and any $y \in \mathbb{R}$

$$\mathbb{P}_N\left[(u,v); \frac{C(u,v) - \widehat{\mu}(c)N}{\widehat{\rho}(c)\sqrt{N}} \leq y\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-x^2/2}\, dx + O\left(\frac{1}{\sqrt{N}}\right).$$

(b) The mean and the variance satisfy

$$\mathbb{E}_N[C] = \widehat{\mu}(c)N + \widehat{\mu}_1(c) + O(2^{-N\tau}), \qquad \mathbb{V}_N[C] = \widehat{\rho}^2(c)N + \widehat{\rho}_1(c) + O(2^{-N\tau}),$$

where τ is a strictly positive constant that does not depend on cost c.
(c) Let $\Lambda(s)$ denote the pressure function. In the special case $c \equiv 1$, denoting $\widehat{\mu} := \widehat{\mu}(1)$, $\widehat{\rho}^2 := \widehat{\rho}^2(1)$, we have

$$\widehat{\mu} = \frac{2\log 2}{|\Lambda'(1)|} = \frac{2\log 2}{\alpha} > 0, \quad \widehat{\rho}^2 = 2\log 2 \frac{|\Lambda''(1)|}{|\Lambda'(1)^3|} > 0\,.$$

In the general case,

$$\widehat{\mu}(c) = \widehat{\mu} \cdot \mu(c)\,, \qquad \widehat{\rho}^2(c) = \mu^2(c) \cdot \widehat{\rho}^2 + \widehat{\mu} \cdot \rho^2(c) + \widehat{\mu}^2 \cdot \mu(c) \cdot \chi(c) > 0\,,$$

where $\mu(c) > 0$ and $\rho^2(c) \geq 0$ are given in Theorem 1, and $\chi(c) = \Lambda''_{sw}(1,0)$.
Claims (a), (b), and (c) also hold for $\widetilde{\mathbb{P}}_N$ on $\widetilde{\Omega}_N$.

7.8. **Distributional analysis of continuants.** The second result of this Section proves that, on Ω_N^+, for any algorithm of the Good Class, the sizes of ending continuants at a fraction of the depth follow an asymptotic Gaussian law, with an optimal speed of convergence. This is a result obtained in [69].

Theorem 8. [Central Limit Theorem for size of ending continuants at a fraction of the depth.] *For a Euclidean algorithm of the Good Class, and for any rational $\delta \in]0,1[$,*
(a) The distribution of $LU_{[\delta]}$ on Ω_N is asymptotically Gaussian, with speed of convergence $O(1/\sqrt{N})$: there exist two constants $\mu_{[\delta]}$ and $\rho_{\delta]}$, such that, for any N, and any $y \in \mathbb{R}$

$$\mathbb{P}_N\left[(u,v); \frac{LU_{[\delta]}(u,v) - \mu_{[\delta]}N}{\rho_{[\delta]}\sqrt{N}} \leq y\right] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-x^2/2}\, dx + O\left(\frac{1}{\sqrt{N}}\right).$$

(*b*) *The mean and the variance satisfy*

$$\mathbb{E}_N[LU_{[\delta]}] = \mu_{[\delta]} \cdot N + \nu_{[\delta]} + O(2^{-N\tau}), \qquad \mathbb{V}_N[LU_{[\delta]}] = \rho_{[\delta]} \cdot N + \eta_{[\delta]} + O(2^{-N\tau}),$$

where where τ *is a strictly positive constant that depends on rational* δ.

(*c*) *One has:*

$$\mu_{[\delta]} = 1 - \delta, \qquad \rho_{[\delta]} = \delta(1-\delta)\frac{|\Lambda''(1)|}{|\Lambda'(1)|}$$

Claims (a), (b), and (c) also hold for $\widetilde{\mathbb{P}}_N$ *on* $\widetilde{\Omega}_N$.

7.9. **Distributional analysis of bit complexity.** The third result proves that the total bit–complexity of the Extended Algorithm also follows an asymptotic Gaussian law. However, this result, described in [69] is not directly obtained with Dynamical Methods: it is a consequence of Theorem 7 applied to size–cost ℓ, and the speed of convergence is probably not optimal.

The main idea is the decomposition of the extended bit-complexity $\widehat{B}$ as

$$\widehat{B}(u,v) = L(u,v) \cdot C_0(u,v) + Y(u,v),$$

where L is the size defined in Section 1.4, C_0 is the total cost relative to the digit -cost ℓ, and Y is a "remainder" cost where $\mathbb{E}_N[Y] = o(N^2), \mathbb{V}_N[Y] = o(N^3)$. Then, since the variable $L \cdot C_0$ follows an asymptotic gaussian law, it is the same for the variable $\widehat{B}$, with

$$\mathbb{E}_N[\widehat{B}] = \widehat{\mu}(\ell) \cdot N^2 + o(N^2), \qquad \mathbb{V}_N[\widehat{B}] = \widehat{\rho}(\ell) \cdot N^3 + o(N^3).$$

However, this method does not provide an optimal speed of convergence, and, at this moment, we do not know how to obtain a speed of convergence of order $O(N^{-1/2})$.

Theorem 9. [Central Limit Theorem for bit–complexity.] *For a Euclidean algorithm of the Good Class, the distribution of total bit complexity* $\widehat{B} := B + \overline{B}$ *of the Extended Algorithm on* Ω_N *is asymptotically Gaussian, with speed of convergence* $O(1/\ N^{1/3})$*: For any* N*, and any* $y \in \mathbb{R}$*, one has*

$$\mathbb{P}_N\left[(u,v); \frac{\widehat{B}(u,v) - \widehat{\mu}(\ell)\cdot N^2}{\widehat{\rho}(\ell)\cdot N^{3/2}} \le y\right] = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{y} e^{-x^2/2}\,dx + O\left(\frac{1}{N^{1/3}}\right),$$

where $\widehat{\mu}(\ell)$ *and* $\widehat{\rho}(\ell)$ *are the constants of Theorem 7 relative to the size–cost* ℓ.

7.10. **Subdominant constants "à la Porter".** The same tools [Perron's formula, *UNI* Conditions] applied to $\Psi_{[k]}(T)$ defined in (7.6) give access to an asymptotic expansion for $\mathbb{E}_N[R^k]$, for all our parameters of interest. In particular, they give access to subdominant constants of the expectations and variances, and relate them to spectral objects of the transfer operator. For instance, the constant $\widehat{\mu}_1(1)$ of Theorem 7 is the Porter constant, and our study provides an alternative expression of this constant, as a function of spectral objects of the transfer operator [67]

8. **Functional Analysis.** Here, we explain how to obtain functional spaces $\mathcal{F}$ where the main properties of transfer operators described in Figure 8 may hold. We recall that the properties needed for the analyses depend both on the kind of analysis [rational trajectories or real trajectories, average-case analysis or distributional analysis, see Figure 9]. All the transfer operators which are used in our analyses are summarized in Figure 5, and Figure 4 recalls the definition of the component operator in each case.

Any density transformer $\mathbf{H}$ acts on the Banach space $\mathcal{L}^1(X)$: this is due to the inequality

$$\int_X |\mathbf{H}[f](t)|dt \leq \int_0^1 |f(t)|dt.$$

But the space $\mathcal{L}^1(X)$ seems too large to be used here, since the spectrum of $\mathbf{H}$ when acting on $\mathcal{L}^1(X)$ is very often continuous: the essential property "Spectral Gap" does not hold. The main difficulty is to find a convenient functional space where all the Properties of Figure 8 hold. This functional space must be sufficiently "large" so that $\mathbf{H}$ acts on it, and sufficiently "small" so that there is a spectral gap. Property *UDE* is true under quite general "positivity" hypotheses (for instance, theorems due to Krasnoselski [58], or cone methods as described in Baladi's book [4]), and very often true for systems with complete branches [as soon as they are topologically mixing]. In contrast, Property *SG* is both central and not so easy to obtain in a quite general framework. [See [65] for a nice introduction to Spectral Theory].

There are two classes of operators for which it is easy to prove that spectrum exhibits a spectral gap : the compact operators, whose spectrum is discrete (except an accumulation point at zero) or the quasi-compact operators. We recall now this notion: For an operator $\mathbf{L}$, denote by $R(\mathbf{L})$ its spectral radius, i.e., the supremum of moduli $|\lambda|$ when λ is an element of $\mathrm{Sp}(\mathbf{L})$, and by $R_e(\mathbf{L})$ its essential spectral radius, i.e., the smallest positive number r such that any eigenvalue λ of $\mathrm{Sp}(\mathbf{L})$ with modulus $|\lambda| > r$ is an isolated eigenvalue of finite multiplicity. For compact operators, the essential radius equals 0. An operator $\mathbf{L}$ is quasi-compact if the strict inequality $R_e(\mathbf{L}) < R(\mathbf{L})$ holds. Then, except for the part of the spectrum inside the closed disk of radius $R_e(\mathbf{L})$, the operator behaves just like a compact operator (in the sense that its spectrum consists of isolated eigenvalues of finite multiplicity). In order to prove that Property *SG* holds, we have to exhibit a functional space where the density transformer acts and is compact or quasi–compact.

8.1. **Compacity and analytic spaces.** There exist results due to Schwartz [87], Shapiro and Taylor [92], Shapiro [91] which exhibit sufficient conditions under which transfer operators are compact on a functional space of analytic functions defined on some disk $\mathcal{M}$. The operator $\mathbf{H}_s$ is the sum of the component operators $\mathbf{H}_{s,[h]}$, and each component operator $\mathbf{H}_{s,[h]}$ is a so–called composition operator of the form $f \mapsto g \cdot f \circ h$, where the inverse branch h is an analytic function which maps $\mathcal{M}$ inside $\mathcal{M}$, and g an analytic function defined on $\mathcal{M}$. It is then possible to relate the properties of the operator and the position of the image $h(\mathcal{M})$ with respect to the boundary of $\mathcal{M}$. More precisely, there are two distinct situations, according as the image $h(\mathcal{M})$ lies strictly inside $\mathcal{M}$ or not.

First Situation. *The image $h(\overline{\mathcal{M}})$ lies strictly inside $\mathcal{M}$.*
More precisely, suppose that there exists an open disk $\mathcal{M}$ such that
(i) every LFT $h \in \mathcal{H}$ has an analytic continuation on $\mathcal{M}$, and maps the closure $\overline{\mathcal{M}}$ of disk $\mathcal{M}$ inside $\mathcal{M}$;
(ii) For each $h \in \mathcal{H}$, there exists $\delta(h) < 1$ for which the analytic continuation of the function $|h'|$, denoted by $\widetilde{h}$, satisfies $0 < |\widetilde{h}(z)| \leq \delta(h)$ for all $z \in \mathcal{M}$
(iii) the series $\sum_{h\in\mathcal{H}} \delta(h)^{s/2} \cdot [\det h]^{-s/2}$ converges on the plane $\Re(s) > \alpha$ for some $\alpha < 1$.
Then, the convenient functional space will be the space $A_\infty(\mathcal{M})$ of all functions f that are holomorphic in the domain $\mathcal{M}$ and are continuous on the closure $\overline{\mathcal{M}}$.

Endowed with the sup-norm,

$$||f|| = \sup \{|f(u)|;\ u \in \mathcal{M}\},$$

$A_\infty(\mathcal{M})$ is a Banach space. Under previous conditions, the transfer operator operator $\mathbf{H}_s$ acts on $\mathcal{A}_\infty(\mathcal{M})$ for $\Re(s) > \sigma$ and is compact. It is moreover nuclear of order 0 (in the sense of Grothendieck [40], [41]). Property (i) is essential here, and it is necessary that the closure $\overline{\mathcal{M}}$ of disk $\mathcal{M}$ is mapped inside $\mathcal{M}$.

Mayer [72, 73, 71] deeply studied the transfer operator $\mathbf{H}_s$ relative to the Classical Euclidean Dynamical System [often called the Ruelle-Mayer operator] and proved that $\mathbf{H}_s$ satisfies properties $(i), (ii), (iii)$ for some disk $\mathcal{M}$. He deduced the nuclearity of such an operator on $A_\infty(\mathcal{M})$. These properties were generalized to any good algorithm of Type 1 in [107]. For algorithms of the Bad Class, it will be the same for the transfer operator $\widetilde{\mathbf{H}}_s$ relative to the induced system : see [106], [107]. Then, for all the algorithms of the Easy Class, it is possible to work with compact operators on sets $A_\infty(\mathcal{M})$ of analytical functions, for a convenient disk $\mathcal{M}$.

Second Situation. *The image $h(\mathcal{M})$ lies inside $\mathcal{M}$, but the frontier of $h(\mathcal{M})$ may "touch" the frontier of $\mathcal{M}$.*

More precisely, suppose that there exists an open disk $\mathcal{M}$ such that every LFT $h \in \mathcal{H}$ has an analytic continuation on $\mathcal{M}$, and maps the disk $\mathcal{M}$ inside $\mathcal{M}$.

Then the convenient space will be the Hardy space $\mathcal{H}^2(\mathcal{M})$. We consider a disk $\mathcal{M}$ whose frontier is denoted by δ. We then consider the space of functions defined in $\mathcal{M}$, that are analytic inside $\mathcal{M}$ and such that the quantity

$$||f||_2^2 := \frac{1}{2\pi\rho} \int_\delta |f(z)|^2\ |dz|$$

is finite. This space is classically denoted by $\mathcal{H}^2(\mathcal{M})$ and is called the Hardy space of order two associated to the disk $\mathcal{M}$. The quantity $||f||_2$ defined above is a norm which endows $\mathcal{H}^2(\mathcal{M})$ with a structure of Banach space, even more of Hilbert space. Each component operator $\mathbf{H}_{s,[h]}$ is a composition operator which acts on $\mathcal{H}^2(\mathcal{M})$ and is compact as soon as h maps the disk $\mathcal{M}$ inside $\mathcal{M}$. As previously, it is moreover nuclear of order 0 (in the sense of Grothendieck [40], [41]. When the series which defines $\mathbf{H}_s$ as a sum of component operators $\mathbf{H}_{s,[h]}$ is convergent in $\mathcal{H}^2(\mathcal{M})$, the same properties hold for the operator $\mathbf{H}_s$.

This framework is well adapted for the induced version $\widetilde{\mathbf{H}}$ of the Binary Algorithm, which is defined in (3.6), see [105].

8.2. **Quasi–compacity.** The following theorem, due to Hennion [43] is a generalisation of previous theorems due to Ionescu-Tulcea and Marinescu, or Lasota-Yorke. It gives sufficient conditions that entail that an operator is quasi–compact. It deals with some Banach space $\mathcal{F}$ endowed with two norms, a weak norm $|.|$ and a strong norm $||.||$, for which the unit ball of $(\mathcal{F}, ||.||)$ is precompact in $(\mathcal{F}, |.|)$.

Theorem [Hennion, Ionescu-Tulcea and Marinescu, Lasota-Yorke]. *Suppose that the Banach space $\mathcal{F}$ is endowed with two norms $|.|$ and $||.||$, and the unit ball of $(\mathcal{F}, ||.||)$ is precompact in $(\mathcal{F}, |.|)$. Let $\mathbf{L}$ be a bounded operator on $(\mathcal{F}, ||.||)$. Assume that there exist two sequences $\{r_n\}$ and $\{t_n\}$ of positive numbers such that, for all $n \geq 1$, one has*

$$||\mathbf{L}^n[f]|| \leq r_n \cdot ||f|| + t_n \cdot |f|. \tag{8.1}$$

Then, the essential spectral radius of the operator $\mathbf{L}$ on $(\mathcal{F}, ||.||)$ satisfies

$$R_e(\mathbf{L}) \leq r := \lim_{n\to\infty} \inf (r_n)^{1/n}.$$

If, moreover, the spectral radius $R(\mathbf{L})$ in $(\mathcal{F}, ||.||)$ satisfies $R(\mathbf{L}) > r$, then the operator $\mathbf{L}$ is quasi-compact on $(\mathcal{F}, ||.||)$.

This general Theorem has many applications in our framework.

(*i*) *Good Class.* For algorithms of the Good Class, one chooses $\mathcal{F} := \mathcal{C}^1(X)$, the weak norm is the sup-norm $||f||_0 := \sup|f(t)|$, while the strong norm is the norm $||f||_1 := \sup|f(t)| + \sup|f'(t)|$. Then, the density transformer satisfies the hypotheses of Hennion's Theorem. Note that Properties $(P1), (P2)$ in Figure 2 are essential here.

(*ii*) *Bad Class.* The previous statement is also true for the transfer operator relative to the induced systems associated to the Bad Class, since the induced set $\widetilde{\mathcal{H}}$ fulfills Properties $(P1), (P2)$ [see Section 2.6].

(*iii*) *Type 4.* For this type, one uses various functional spaces, with various applications of Hennion's Theorem. We work both in the space $\mathbb{H}_\alpha(J)$ of α–Hölder functions [the strong norm is the α–Hölder norm, and the weak norm is the $\mathcal{L}^1$ norm] and in the space $\mathcal{C}^0(J)$) of continuous functions [the strong norm is now the norm sup, and the weak norm is the $\mathcal{L}^1$ norm].

8.3. **Study of the underlined operator.** Recall that the underlined operators act on functions F of two variables, and their components are of the form

$$\underline{\mathbf{H}}_{s,w,[h]}[F](x,y) := \delta_h^{s-w} \cdot |h'(x)|^s \cdot |h'(y)|^{-w} \cdot F(h(x), h(y)).$$

On the diagonal $x = y$, the function $F(x,x)$ is denoted by $f(x)$ and the equality $\underline{\mathbf{H}}_{s,w,[h]}[F](x,x) = \mathbf{H}_{s-w}[f](x)$ holds. When the Bounded Distortion Property holds, it is –at least intuitively– clear that the behaviour of the underlined operator $\underline{\mathbf{H}}_{s,w}$ is "close" to the plain operator $\mathbf{H}_{s-w}$.

Consequently, there are two main cases. The Easy Class, for which Property $(P2)$ of Figure 2 holds, or the Difficult Class, where it does not hold.

Good Class. One chooses $\mathcal{F} := \mathcal{C}^1(X \times X)$, the weak norm is the sup-norm $||F||_0 := \sup|F(x,y)|$, while the strong norm is the norm $||F||_1 := \sup|F(x,y)| + \sup|DF(x,y)|$, where $DF(x,y)$ denotes the differential of F at (x,y). Then, the transfer operator $\underline{\mathbf{H}}_{s,w}$ satisfies the hypotheses of Hennion's Theorem, and is proven to be quasi–compact.

Bad Class. The previous statement is also true for the transfer operator relative to the induced systems associated to the Bad Class.

Difficult Class. For Type 3 or Type 4, neither Property $(P1)$ nor Property $(P2)$ hold, and we do not succeed to apply Hennion's theorem: We do not know how to obtain a functional space $\mathcal{F}$ where the underlined operators are proven to be quasi–compact.

Easy Class. It is also possible to work inside the Banach space $A_\infty(\mathcal{M} \times \mathcal{M})$, for a convenient disk $\mathcal{M}$ which contains the real interval X of the relative dynamical system; in this case, the underlined transfer operators [the underlined transfer operators for the Good Class or the "induced" underlined transfer operator for the Bad Class] have very nice supplementary properties, for instance compacity and nuclearity.

Type	Plain Operator	Underlined Operator
Types 0,1, 2	$\mathcal{C}^1(X)$ $A_\infty(\mathcal{M})$	$\mathcal{C}^1(X \times X)$ $A_\infty(\mathcal{M} \times \mathcal{M})$
Binary	$\mathcal{H}^2(\mathcal{M})$	———
LSB	$\mathcal{C}^1(J), \mathcal{C}^0(J)$	———

FIGURE 21. Functional spaces associated to each operator.

8.4. **Choice of the functional Space.** We summarize the possible choices.

Transfer operators on functions of one variable.

Type 1 and 2. As we saw it, there are two possible choices

(i) the Banach space $A_\infty(\mathcal{M})$, for a convenient disk $\mathcal{M}$ which contains the real interval X of the relative dynamical system; in this case, the transfer operators [the plain transfer operators for the Good Class or the "induced" transfer operator for the Bad Class] have very nice supplementary properties, for instance compacity and nuclearity.

(ii) the Banach space $\mathcal{C}^1(X)$, which is easier to use. For instance, results à la Dolgopyat are proven to hold in this space, and not in the previous space.

Type 3. For the induced Binary System, we choose as $\mathcal{F}$ the Hardy space $\mathcal{H}^2(\mathcal{M})$ relative to a disk $\mathcal{M}$ of diameter $[0, \rho]$ with $1 < \rho < 2$. We did not succeed in finding a convenient functional space for the Plus-Minus Algorithm.

Type 4. We choose as $\mathcal{F}$ both spaces $\mathcal{C}^1(J), \mathcal{C}^0(J)$.

Transfer operators on functions of two variables [only for the Easy Class].

As previously there are two possible choices

(i) the Banach space $A_\infty(\mathcal{M} \times \mathcal{M})$, for a convenient disk $\mathcal{M}$ which contains the real interval X of the relative dynamical system; in this case, the transfer operators [the plain transfer operators for the Good Class or the "induced" transfer operator for the Bad Class] have very nice supplementary properties, for instance compacity and nuclearity.

(ii) the Banach space $\mathcal{C}^1(X \times X)$, which is easier to use.

8.5. **General spectral properties of transfer operators.** We now describe the main spectral properties of the transfer operator $\mathbf{H}_{s,w}$ [plain, with a hat, or underlined] on this convenient functional space $\mathcal{F}$. All these properties are easy consequences of three central properties for the density transformer $\mathbf{H}$:

Property *UDE*: Unique Dominant Eigenvalue

Property *SG*: Spectral Gap

Properties $An_?$: various analyticity properties.

Consider a Euclidean dynamical system (X, V) of any type and the functional space $\mathcal{F}$ previously defined in Figure 21. Consider any of the three transfer operators: the operators $\mathbf{H}_{s,w,(c)}$ or $\widehat{\mathbf{H}}_{s,w,(c)}$ associated to a cost of moderate growth, or an underlined transfer operator $\underline{\mathbf{H}}_{s,w}$.

In the following, the operator $\mathbf{G}_{s,w}$ denotes:

– for the Good Class, any of the three previous operators.

– for the Bad Class, any of the induced versions of the three operators.

– for the Difficult Class, only the operators $\mathbf{H}_{s,w}$ or $\widehat{\mathbf{H}}_{s,w}$ associated to a cost of moderate growth.

In any cases, there exist an interval $\Sigma_0 :=]\sigma_0, +\infty[$ with $\sigma_0 < 1$ and a real neighborhood W_0 of $w = 0$ for which the following eight properties are true when $(\sigma = \Re s, \nu = \Re w)$ belongs to $\Sigma_0 \times W_0$:

(1) [Quasi-compactness.] The operator $\mathbf{G}_{s,w}$ acts boundedly on $\mathcal{F}$, and $\mathbf{G}_{s,w}$ is (uniformly) quasi-compact for real (s, w).

(2) [Unique dominant eigenvalue.] For real $(\sigma, \nu) \in \Sigma_0 \times W_0$, $\mathbf{G}_{\sigma,\nu}$ has a unique eigenvalue $\lambda(\sigma, \nu)$ of maximal modulus, which is real and simple, the *dominant eigenvalue*. The associated eigenfunction $f_{\sigma,\nu}$ is strictly positive, and the associated eigenvector $\widehat{\mu}_{\sigma,\nu}$ of the adjoint operator $\mathbf{G}^*_{\sigma,\nu}$ is a positive Radon measure. With the normalization conditions, $\widehat{\mu}_{\sigma,\nu}[1] = 1$ and $\widehat{\mu}_{\sigma,\nu}[f_{\sigma,\nu}] = 1$, the measure $\mu_{\sigma,\nu} := f_{\sigma,\nu}\widehat{\mu}_{\sigma,\nu}$ is a probability measure. In particular, $\widehat{\mu}_1$ is the Haar measure, with $\lambda(1) = 1$, and $f_{1,0} = \psi$ the invariant density of the dynamical system..

(3) [Spectral gap.] For real parameters $(\sigma, \nu) \in \Sigma_0 \times W_0$, there is a spectral gap, i.e., the subdominant spectral radius $r_{\sigma,\nu} \geq R_e(\sigma, \nu)$ defined by $r_{\sigma,\nu} := \sup\{|\lambda|; \lambda \in \mathrm{Sp}(\mathbf{G}_{\sigma,\nu}), \lambda \neq \lambda(\sigma, \nu)\}$, satisfies $r_{\sigma,\nu} < \lambda(\sigma, \nu)$.

(4) [Analyticity in compact sets.] The operator $\mathbf{G}_{s,w}$ depends analytically on (s, w). Thus, $\lambda(\sigma, \nu)^{\pm 1}$, $f^{\pm 1}_{\sigma,\nu}$, and $f'_{\sigma,\nu}$ depend analytically on (σ, ν).

(5) [Analyticity in a neighborhood of $(1, 0)$.] If (s, w) is complex near $(1, 0)$ then $\lambda(s, w)^{\pm 1}$, $f^{\pm 1}_{s,w}$, and $f'_{s,w}$ are well-defined and analytic; moreover, for any θ, with $r_1 < \theta < 1$, one has $r_{1,w}/|\lambda(1, w)| \leq \theta$.

(6) [Derivatives of the pressure.] The first derivatives of the dominant eigenvalue function $\lambda(\sigma, \nu)$ of the plain operator at $(1, 0)$ satisfy the following: $\lambda'(1) = \lambda'_s(1, 0)$ is the opposite of the entropy of the dynamical system (V, dx), and, in the case of a cost c, $\lambda'_w(1, 0)$ is the average of the cost:

$$\lambda'(1) = -\alpha = -\sum_{h\in\mathcal{H}} \delta_h \log \delta_h - \int_X \log |V'(t)|\psi(t)dt$$

$$\lambda'_w(1, 0) = \mu(c) = \sum_{h\in\mathcal{H}} \delta_h \cdot c(h) \int_{h(X)} \psi(t)\, dt.$$

(7) [Function $w \mapsto \sigma(w)$.] There is a complex neighborhood $\mathcal{W}$ of 0 and a unique function $\sigma : \mathcal{W} \to \mathbb{C}$ such that $\lambda(\sigma(w), w) = 1$, this function is analytic, and $\sigma(0) = 1$.

(8) Furthermore, for the Good Class, there exist relations between the dominant eigenvalues of the three operators $\widehat{\mathbf{H}}_{s,w}$, $\underline{\mathbf{H}}_{s,w}$ and $\mathbf{H}_{s,w}$, namely

$$\widehat{\lambda}(s, w) = \lambda(s, w), \qquad \underline{\lambda}(s, w) = \lambda(s - w, 0).$$

8.6. **Properties *SM, SLC, UNI, US.*** These properties are not automatic consequences of the three main properties *UDE, SG, An.* As we already said, some of these properties do not hold for dynamical systems with affine branches. However, as soon as a dynamical system is not $\mathcal{C}^2$ conjugated with a dynamical system with affine branches, Property *SM* holds. Property *SLC* also holds in this case provided the cost c is not constant (see for instance Lemma 7 of [7]).

(9) [Property *SM*]. For any $t \neq 0$, the spectral radius $R(\sigma + it, 0)$ satisfies $R(\sigma + it, 0) < R(\sigma, 0)$.

(10) [Property *SLC*]. For any $(q, r) \neq (0, 0)$, the second derivative of the pressure function $w \mapsto \log \lambda(1 + qw, rw)$ is not zero at $w = 0$.

For Euclidean dynamical systems, all the branches are LFT's, and it is easy to prove that these systems are not $\mathcal{C}^2$ conjugated to dynamical systems with affine branches. Then, these two properties *SM* and *SLC* hold for all our Euclidean dynamical systems. On the other hand, Condition *UNI* [Strong Condition *UNI*, or only Weak Condition *UNI*], when it is fulfilled, entails that all the three properties *SM, SLC, US* hold.

8.7. **Conclusion of the Functional Analysis Study.** Finally, this Section proves the following facts

— Theorem 1 holds for all Systems of the Fast Class.
— Theorem 2 holds for all systems of the Good Class.
— Theorems 3, 6 hold for all the systems.
— Theorems 4, 5 hold for the Fast Class.
— Theorems 7, 8, 9 hold for all the Good Class.

9. **Historical Notes, Possible extensions and Open Problems.** Here, we provide a complete description of related works, and state some open problems.

9.1. **Euclidean Algorithms and Worst-case Analysis.** For a description of Euclidean Algorithms, see Knuth's and Shallit's vivid accounts [56, 89].

Classical Euclidean Algorithms. Euclid's Algorithm based on the usual division was discovered as early as 300BC, and is "the grandfather of all the algorithms", as Knuth says. Euclidean Algorithm was analysed first in the worst case in 1733 by de Lagny, The Centered algorithm (K) has been considered by Rieger [83]. The Even Algorithm is introduced by Eisenstein [32]. The Even and Odd algorithms are described in [94, 95].

Pseudo–Euclidean Algorithms. Two of these Algorithms, the Pseudo-Classical ($\breve{G}$), the Pseudo-Centered ($\breve{K}$) have been studied by Shallit [88] who limited himself to a worst-case analysis and wrote "Determining the average behaviour for these algorithms seems quite hard." Then, Vallée introduces a general formalism for what she called pseudo-euclidean algorithms [106], [107].

Binary, Plus-Minus Algorithms. The Binary algorithm of Stein [96] is described for instance in Knuth [56] while the Plus-Minus algorithm is due to Brent and Kung [15].

LSB Algorithm. Finally, Stehlé and Zimmermann proposed to consider divisions that are totally directed by the LSB's, which then lead to the integer analogue to increasing–degree gcd algorithm for polynomials. Such an algorithm is described in [98] for instance, where the authors also provide a worst–case analysis of this algorithm.

Lehmer-Euclid algorithm, Interrupted Algorithm. The Lehmer-Euclid algorithm is an improvement of the Euclid algorithm when applied for large integers. It was introduced by Lehmer [62] and first analyzed in the worst–case by Sorenson [97]. It uses what Daireaux et Vallée have called the Interrupted Euclidean algorithm [24]. This interrupted algorithm depends on some parameter $\alpha \in [0, 1]$, and, when running with an input (u, v), it performs the same steps as the usual Euclidean algorithm, but it stops as soon as the current integer is smaller than v^α.

9.2. **Dynamical Euclidean systems and transfer operators.** We summarize here the main works about the continuous Euclidean framework.

Dynamical systems. See for instance [70] for a readable treatment of dynamical systems of intervals, and [9] for a general overview on dynamical systems. The Classical Euclidean system was first studied by Gauss himself. The density transformer, also known as the Perron-Frobenius operator, was introduced early in the study of continued fractions (see for instance Lévy [63], Khinchin [54], Kuzmin [59], Wirsing [111] and Babenko [3]). It was more recently deeply studied by Mayer, in a sequence of papers [72, 73, 71, 74, 75, 76]. The Centered system was studied by Rieger [83, 84], the Even system by Schweiger, Bauer, Kraaicamp and Lopes [94, 8, 55], the Odd System by Schweiger [94]. The dynamical system for polynomials was described in [11].

Transfer operators. See the book of V. Baladi [4] for a general overview on transfer operators. The density transformer is a special case of a transfer operator, and the general notion of transfer operators was introduced by Ruelle, in connection with his thermodynamic formalism (see for instance [85, 86]). Then Mayer has applied such operators to the classic continued fraction transformation.

After works of Chernov [21], Dolgopyat [29] was interested in the decay of correlations for hyperbolic flows satisfying some uniform nonintegrability condition (*UNI*). Later on, Pollicott and Sharp used Dolgopyat's bounds together with Perron's formula to find error terms in asymptotic estimates for geodesic flows on surfaces of variable negative curvature; see e.g. [81], where only univariate Dirichlet series with positive cofficients appear.

Truncated trajectories and metrical properties of continued fractions.. Central Limit Theorem for costs [stated here as Theorem 1] is quite well-known. See for instance [22] or [17] for interval maps, and [1] for a more abstract framework and references to the pioneering paper of Nagaev [78]. The situation is less clear for continuants: There are previous works due to Philipp [80] which have been generalized by Vallée [101]. These results are extended to our general framework for the first time in the present paper. A survey for metrical properties of continued fractions is [50].

9.3. **Euclidean Analysis.** Inside the Euclidean framework, most of dynamical studies concern the continuous point of view [metric properties of continued fraction expansions for instance], and not the discrete analysis of gcd algorithms. On the otherhand, most of the analyses of Euclidean Algorithms do not adopt a dynamical point of view. The methods used till the early 1980's are quite various, and they range from combinatorial (de Lagny, Heilbronn) to probabilistic (Dixon).

Average-case analysis. The standard Euclidean Algorithm was analysed first in the average-case around 1969 independently by Heilbronn [42] and Dixon [28]. The centered algorithm was studied by Rieger [83]. Brent [14] has analysed the Binary algorithm under some heuristic hypotheses. Brent's work deals with the operator **H** relative to the plain [i.e., not induced] Binary Euclidean System defined in (3.4), and he conjectures that there exists a limit density for the algorithm. Then, he makes a (essential) heuristic hypothesis: the rationals have a "typical" behaviour inside the reals, so that the restriction to the rationals of the limit (real) density is also the limit (rational) density. Then, under the conjecture and the heuristic hypothesis, he obtains the average–case analysis of the Binary Algorithm (B). The Subtractive algorithm (T) was studied by Knuth and Yao [112]. Results on the average-case analysis of the polynomial gcd can be found in [38, 56].

Distributional analysis. Concerning the standard Euclidean algorithm and the number of steps (i.e., the constant cost $c \equiv 1$), Hensley [44] has obtained a Central Limit Theorem, and a Local Limit Theorem with speed of convergence $O((\log N)^{-1/24})$. Hensley has used a transfer operator $\mathbf{H}_{s,0}$, to obtain distributional results on rational trajectories upon aproximating discrete measures on rationals by continuous measures. In particular, his approach avoids parameters s of large imaginary parts.

9.4. **Dynamical Euclidean Analysis.** We now cite the main works of our group which are closely related to the present survey. Note that the Caen group introduced dynamical methods in another algorithmic domain, the Information Theory Domain. See [104] for an instance of such a work.

Average-case dynamical analysis. A precise description of Euclidean analyses can be found in the following papers. Paper [35] is itself a survey paper where transfer operators are used for analysing the Euclid Algorithm together some of its generalization on higher dimensions. Paper [101] introduces for the first time the underlined operators [for analyzing algorithms], and uses them for obtaining Theorem 2 for continuants in the case of the Classical Euclidean Algorithm. More general transfer operator with two variables are introduced in [104]. Papers [106, 107] provide an unifying point of view on Algorithms of Type 1 and 2. The analysis of the Binary Algorithm can be found in [105], while Daireaux explicits the Plus-Minus Dynamical System in [23]. A recent work [25] done by Daireaux, Maume-Deschamps, and Vallée provides the average-case analysis of algorithms of Type 4. The strange title of the paper is due to the fact that executions of such algorithms can be viewed as a race between a (dyadic) hare, and a (Lyapounov) tortoise.

Papers [2, 108] introduce the main parameters: digits, continuants, bit-complexities, and gives a panorama for their analyses. Paper [24] deeply studies (in the average case) the particular parameter "continuant at a fraction of the depth".

Distributional dynamical analysis. To the best of our knowledge, the general framework described in Section 7, due to Baladi and Vallée [7] provides the first instance of a dynamical distributional analysis. The authors apply and extend powerful tools due to Dolgopyat to dynamical systems with infinitely many branches, with two different points of view: they consider the Strong *UNI* Condition, in [5, 7] or the Weak *UNI* Condition in [6]. In this way, they improve Hensley's result [44] while extending it to a large class of cost functionals and to several algorithms and obtaining an optimal speed of convergence. The last two results of Section 7 are due to Lhote and Vallée [69, 67].

9.5. **Open Problems.** This survey shows that there yet exist many problems to solve in this area.

(*i*) What happens for the distribution of the algorithms of the Difficult Class, namely the Binary Algorithm or the LSB Algorithms? The functional space used for the average–case analysis of the Binary Algorithm [a Hardy Space] is not so easy to deal with, and, at the moment, we do not succeed to extend Dolgopyat methods to such a space.

(*i*) Does Theorem 8 hold for beginning continuants? This is related to a possible extension of the result of Section 7.4 to underlined operators $\underline{\mathbf{H}}_{s,w}$.

(*iii*) On another register, the extension to "large" costs or Bad Class is likely to lead us to the realm of stable laws: see for instance Gouezel's work [39] for occurrences of these laws in continued fraction related matters.

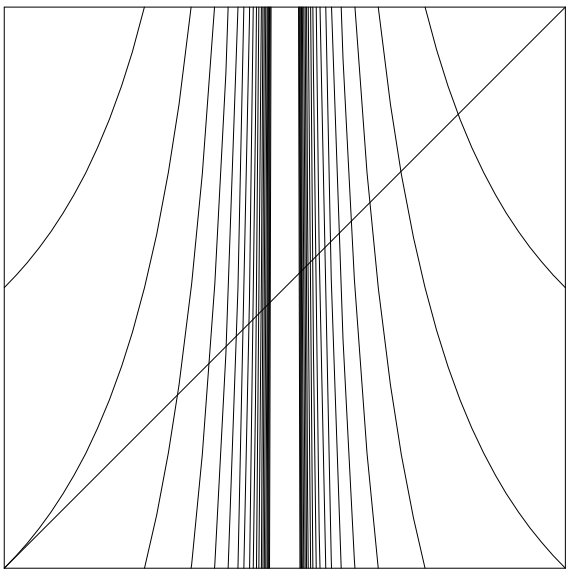

FIGURE 22. Japanese dynamical systems

(*iv*) There exist fast gcd algorithms [93] which are based on a *Divide and Conquer* method, and use the principles of the Lehmer method [61]. For a readable treatment of such algorithms, see the book of Yap [113], for instance. Dynamical Analysis methods probably apply to this class of algorithms (work in progress of Daireaux, Lhote, Maume-Deschamps and Vallée).

9.6. **α-Euclidean Algorithms.** Three Algorithms of the MSB Class are defined by the position of the remainder : the standard division relative to a remainder in $[0, 1[$, the centered division, with a remainder in $[-1/2, +1/2[$, or the by-excess division with a remainder in $[-1, 0[$. With respect to the classification previously described, the first two algorithms (i.e., standard and centered) belong to the Good Class, while the third one, by excess, belongs to the Bad Class. It is thus quite natural to study a "generic" Euclidean algorithm, called the α–Euclidean algorithm, where the remainder has to belong to some interval $[\alpha - 1, \alpha[$, with $\alpha \in [0, 1]$. When the parameter α varies in $[0, 1]$, this gives rise to a whole class of Euclidean algorithms. There are now natural questions to ask: Are there other values than 0 of parameter α for which the algorithm belongs to the Bad Class? How do the number of iterations and the bit–complexity evolve with respect to α? What is the best algorithm in the whole class? Paper [19] provides some answers to these questions.

All the dynamical systems $\mathcal{S}_\alpha$ relative to α appear in a quite natural manner. First, we draw the set of all the maps F_i defined on $[-1, +1] \setminus \{0\}$ by

$$F_i(x) = \left|\frac{1}{x}\right| - i$$

for any integer $i \geq 1$. Then, we only "keep" the window $\mathcal{I}_\alpha \times \mathcal{I}_\alpha = [\alpha - 1, \alpha] \times [\alpha - 1, \alpha]$, and we obtain the representation of the dynamical system $\mathcal{S}_\alpha$ [see Figure 22].

There is a main difference with the Dynamical Systems described here, which are "complete" –in the sense that all the branches are surjective–, and for which the

transfer operator is then proven to act on $\mathcal{C}^1([0,1])$ Here, the involved dynamical system $\mathcal{S}_\alpha$ is no longer "complete" –in the sense that there exist some branches that are not surjective–. Generally speaking, it is not even Markovian, and the convenient functional space $\mathcal{F}$ is the set BV of functions defined on $[0,1]$ with bounded variation; If we choose as a weak norm the $\mathcal{L}^1$-norm and as a strong norm the norm $||f||_{BV}$, Hennions's Theorem (see Section 8.2) can be applied, and the transfer operator $\mathbf{H}_s$ is quasi-compact on BV.
When parameter α belongs to $[1/2,1]$, this dynamical system $\mathcal{S}_\alpha$ has been first extensively studied by Ito, Tanaka and Nakada [51, 79]. This is why the α–Euclidean algorithms are often nicknamed as "Japanese algorithms". Later, Moussa, Cassa, Marmi [77] provided an extension of these results to the range $\alpha \in [\sqrt{2}-1, 1/2]$.

Paper [19] proves the following:

(i) For any parameter $\alpha \neq 0$, all the algorithms $\mathcal{E}_\alpha$ belong to the Fast Class, and the analogues of Theorem 3, 4, 5, 6 are true for all these Algorithms. The mean values involve the entropy $h(\alpha)$ of the dynamical system, together with a constant $\mathbb{E}_\alpha[\ell]$ related to the digit– size.

(ii) There is a central range $\alpha \in [\sqrt{2}-1, \phi-1]$ where the average number of iterations of the Euclidean Algorithms is optimal and independent of α. It corresponds to the number of iterations of the Centered Algorithm (K).

9.7. **Generalizations of the Euclid Algorithm in higher dimensions.** There are many various points of view for a generalization of the Euclidean algorithms.

The lattice reduction problem. [26][109]. The lattice reduction problem consists in finding a short basis of a lattice of Euclidean space given an initially skew basis. This reduction problem is well–known to be central to many areas of approximation and optimization with deep consequences in computational number theory, cryptography, and symbolic computation.

In dimension $d = 1$, lattice reduction may be viewed as a mere avatar of the Euclidean GCD algorithm and of continued fraction expansions. Lattice reduction *per se* really started with Gauss who gave an algorithm that solves the problem exactly using what resembles a lifting of the Euclidean algorithm to 2–dimensional lattices. In recent times, an important discovery was made by Lenstra, Lenstra and Lovász in 1982 [64]; their algorithm, called the LLL algorithm, is able to find reduced bases in all dimensions $d \geq 3$. The LLL algorithm itself proceeds by stages based on the Gaussian algorithm as the main reduction step.

Paper [26] provides a detailed analysis of the Gaussian algorithm, both in the average case and in probability. Like its one–dimensional counterpart, the algorithm is known to be of worst–case logarithmic complexity, a result due to Lagarias [60], with best possible bounds being provided by Vallée [100]. The analysis provided in [26] is another instance of a dynamical analysis; it deals with the transfer operator $\mathbf{H}_s$ relative to the Classical Euclidean Algorithm with the special value $s = 2$, and all the results can be expressed with the spectral objects of $\mathbf{H}_2$. The probabilistic behaviour of the Gaussian algorithm turns out to be appreciably different of the Euclidean Algorithm:

(i) The average–case complexity of the Gaussian algorithm (measured in the number of iterations performed) is asymptotically constant, and thus essentially independent of the size of the input vectors. The expectation μ is equal to $(I - \mathbf{H}_2)^{-1}[1](0)$ and $\mu \sim 1.35113\,15744\ldots$

(*ii*) The distribution of the number of iterations is closely approximated by a geometric law whose ratio is $\lambda(2) \sim 0.1994$.

(*iii*) The dynamics of the algorithm is governed by a (conditional) limit measure that constitutes the analogue of the limit measure first observed by Gauss for continued fractions. This measure has a density which is closely related to the dominant eigenfunction of $\mathbf{H}_2$.

These results have been recently extended to study additive costs (see [109]). On average, the Gaussian algorithm is thus of complexity $O(1)$, which is of an order different from the worst-case. The case of dimension $d = 2$ therefore departs significantly from its 1–dimensional analogue, and it would be of interest to determine to which extent such a phenomenon propagates to higher dimensions.

Our analytic knowledge of the LLL algorithm in higher dimensions is of course less advanced, but Daudé and Vallée [30] already succeeded in proving that the LLL algorithm, when applied to d–dimensional lattices, has an average–case complexity that is bounded from above by a constant K_d, where $K_d = O(d^2 \log d)$. The present work thus fits as a component of a more global enterprise whose aim is to understand theoretically why the LLL algorithm performs in practice much better than worst–case bounds predict, and to quantify precisely the probabilistic behaviour of lattice reduction in higher dimensions.

Comparison of rationals. [103] [36] How to compare two rationals numbers a/b and c/d [with a, b, c, d four integers of length N], if we do not wish to work with integers of length $2N$? One uses a continued fraction expansion algorithm applied simultaneously to the two numbers and stopped as soon as a discrepancy of CFE digits is encountered. This algorithm can be used to compute the sign of the determinant $(ad - bc)$ without computing the determinant itself, and it is crucial in geometric algorithms, for instance. How many steps are necessary (on average) to compare two rationals? What is the (asymptotic) distribution of the numbers of steps when N becomes large? The two answers are the same as in the previous paragraph:

(*i*) the average number of iterations is asymptotically constant, and the constant is equal to the previous constant μ.

(*ii*) The distribution of the number of iterations is closely approximated by a geometric law whose ratio is $\lambda(2) \sim 0.1994$.

In [109], Vera provides an explanation of the similarity between the two algorithms (the Gauss Algorithm and the Comparison Algorithm). This similarity is well–described by the geometry of the domains $[L \geq k]$ (where L is the number of iterations of the algorithm) [see Figure 23]. For the Gauss Algorithm, the domain $[L_G \geq k]$ is built by disks whereas the domain $[L_C \geq k]$ relative to the comparison algorithm is built by squares. However, both disks and squares are themselves built on the fundamental intervals $h(I)$ of the Euclidean Algorithm. And the constant $\lambda(2)$ can be read on the figure: it is approximatively the ratio between the measure of two successive domains $[L = k]$ and $[L = k + 1]$.

*Sorting rational numbers.*The sign problem leads to the more general question of sorting n numbers $x_1, x_2, \ldots, x_n$. The principle is to determine the first CFE–digit of each number, group the numbers according to their first CFE–digit, and then sort recursively the subgroups based on the second CFE–digit. The data structure which is underlying this algorithm is a digital tree (often called a trie). The analysis of such algorithm answers the following question: How many digits in total must

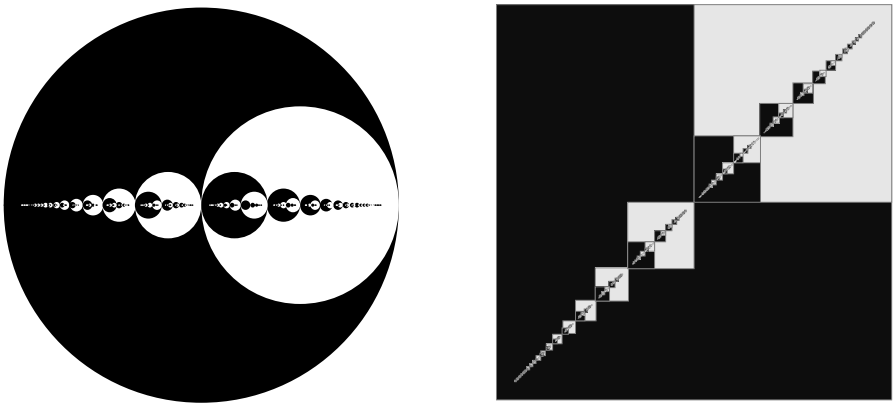

FIGURE 23. The domains $[L = k]$ are drawn alternatively in black and white for rank $k \leq 4$ [on the left, for the Gauss Algorithm, and on the right for the Comparison Algorithm].

be determined in order to sort n real numbers? In symbolic dynamical terms, it describes the way trajectories of n random points under the shift V evolve by sharing some common digits before branching off from each other. And the answer is : the expected number $P(n)$ of quotients that are necessary to sort n numbers satisfy

$$P(n) = \frac{1}{\alpha} \cdot n \log n + O(n), \qquad n \to \infty,$$

where α is the entropy of the CFE dynamical system and the constant which appears in the term $O(n)$ is a variant of the Porter constant [see Section 7. 10].

Other generalizations. ([95, 16, 10, 18]). The Jacobi–Perron algorithm, or its variants, were introduced to generalize the continued fraction algorithm, from the point of view of the construction of a rational approximation of a vector of $\mathbb{R}^d$. The continued fraction algorithm (in one dimensional–case) provides all the best rational approximations of a given real x, and many different algorithms were built for generalizing this property in higher dimensions. Any multidimensional continued fraction algorithm produces, for a given irrational vector $(x_1, x_2, \ldots, x_d)$ of $[0,1]^d \setminus \mathbb{Q}^d$ a sequence of irrational vectors $(p_1(n)/q(n), p_2(n)/q(n), \ldots, p_d(n)/q(n))$ which converges to $(x_1, x_2, \ldots, x_d)$. Very often, the algorithms are only convergent in the weak sense, i.e.

$$\lim_{n\to\infty} \left\| (x_1, x_2, \ldots, x_d) - \left(\frac{p_1(n)}{q(n)}, \frac{p_2(n)}{q(n)}, \ldots, \frac{p_d(n)}{q(n)} \right) \right\| = 0,$$

but some of them are strongly convergent, i.e.

$$\lim_{n\to\infty} ||q(n)(x_1, x_2, \ldots, x_d) - (p_1(n), p_2(n), \ldots, p_d(n))|| = 0.$$

These convergence properties are closely related with the Lyapounov exponents (see [12]) of the set of matrices used by the algorithm. In particular, if the second Lyapounov exponent is strictly negative, then the algorithm is strongly convergent. As in the lattice reduction problem, the two dimensional–case is particular and easier to deal with, and the Jacobi-Perron algorithm is proven to be strongly convergent for $d = 2$.

9.8. **Study of Constrained Continued Fractions.** ([102] [20]). What can be said about rationals or reals whose continued fraction expansion obeys some constraints on its digits? In a quite general setting, such sets of reals have zero measure, and it is thus interesting to study their Hausdorff Dimension. For instance, the reals whose all digits in continued fraction expansion are at most M are deeply studied since they are badly approximable by rationals, and intervene in many contexts of number theory (see [90]). Hensley precisely studied this set E_M, its Hausdorff dimension t_M, and exhibits the asymptotic behaviour of $|t_M - 1|$ when the constraint bound M tends to ∞ [45]. These results have been generalized in [102] where "periodic" constraints are considered.

It is also of great interest to consider more general constraints which only deal with all the digit prefix averages of continued fraction expansions. Paper [20] considers some digit–cost c [a weight] and studies the set F_M of reals for which the weighted average of each prefix (in the continued fraction expansion) has to be bounded by M. This setting can be translated in terms of random walks where each step performed depends on the present digit, and walks under study are constrained to be always under a line of slope M. Paper [20] first provides a characterization of the Hausdorff dimension s_M, in terms of the dominant eigenvalue of the weighted transfer operator $\mathbf{H}_{s,w}$ relative to the Euclidean dynamical system, and cost c. Then, it exhibits the behaviour of $|s_M - 1|$ when the bound M becomes large. The (dynamical) analysis involves, as in previous works, the weighted transfer operator $\mathbf{H}_{s,w}$. However, the costs to be considered here are of large growth [for instance the cost $c_0(d) := d$ is of great interest and already considered in [46]], and the weighted transfer operator is no longer analytic near the reference point $(s, w) = (1, 0)$. This is why the analysis is quite involved.

When $c_0(d) = d$, this analysis is closely related to the Subtractive Algorithm. For performing the precise analysis of the Subtractive algorithm (T), one needs precise information on the set of *rational* numbers whose digits in the continued fraction expansion have an average less than M. This discrete problem is more difficult to solve than the continuous problem solved in [20]. In the case of "fast Euclidean Algorithms", relative to costs of moderate growth, the weighted transfer operator $\mathbf{H}_{s,w}$ is analytic at the reference point $(s, w) = (1, 0)$. Then, we have seen in the present paper how Tauberian Theorems or Perron's Formula allow a transfer "from continuous to discrete". Here, it does not seem possible to use directly these tools, due to the non–analyticity of $\mathbf{H}_{s,w}$ at $(1, 0)$. This is the same problem which we meet when we study the Bad Class.

9.9. **Computation of Constants related to the analysis.** ([35] [36] [26] [68] [66] [37]). It is of theoretical and practical interest to make precise the status of constants which appear in the probabilistic analysis of Euclidean Algorithms (see Finch's book [37] for a nice introduction to classical constants). The constants which appear in the expectations (see Figure 7 are explicit as soon as the invariant density is known: this is the case for all the algorithms of Type 1, but this is no longer true for other types. How to obtain in this case an approximation for constants α and $\mu(c)$? For any type, are the constants $\gamma, \rho(c)$ of Figure 7 which intervene in the asymptotic expression of variances [in Theorems 1, 2, 7 and 8] polynomial–time computable? What is the status of constants $\mu, \lambda(2)$ which intervene in the analysis of Gauss Algorithm ? How to compute the Hausdorff dimensions s_M, t_m of Section 8.7 ? In the case of the Euclidean Dynamical System, the authors of [26] introduce a method for computing (a finite part of) the spectrum of transfer

operators. They consider a sequence of matrices $\mathcal{M}_n$ which are "truncations" of the transfer operator, and they "approximate" the dominant part of the spectrum of the transfer operator. They observe that the sequence formed with the dominant eigenvalue λ_n of $\mathcal{M}_n$ seems to converge to the dominant eigenvalue λ of the transfer operator (when the truncation degree n tends to ∞), with exponential speed. They conjecture the following: *There exist n_0, K, θ such that, for any $n \geq n_0$, one has $|\lambda_n - \lambda| \leq K\theta^n$.* Lhote [66, 68] proves that the conjecture is true in a very general framework, as soon as the transfer operator is relative to a Dynamical System which is "strongly contracting". He also proves that the constant θ is closely related to the contraction ratio of the Dynamical System. It is then exactly computable, and he proves in this way that the dominant eigenvalue λ (that is unique and isolated in this framework) is polynomial–time computable as soon as the truncated matrix $\mathcal{M}_n$ is computable in polynomial–time (in n). However, if one wishes to obtain proven digits for λ, explicit values of K and n_0 must be exhibited. This does not seem possible for general Dynamical Systems but Lhote proves that it is the case when the transfer operator is normal on a convenient functional space. This property actually holds for the Classical Euclidean Dynamical System, as Mayer showed it in his works.

Acknowledgements. This survey summarizes the activity of the Caen Group during the last ten years. I wish to thank all the persons who participated to this group [in the alphabetic order]: Ali Akhavi, Jérémie Bourdon, Julien Clément, Benoît Daireaux, Hervé Daudé, Charlie Lemée, Loick Lhote, Damien Stehlé, Antonio Vera. On the otherhand, this new framework which mixed two domains —analytic combinatorics and dynamical systems— could not have been elaborated without the parallel influence and help of specialists in each of these domains: many thanks to Viviane Baladi, Philippe Flajolet, Véronique Maume, Bernard Schmitt.

References

[1] Aaronson, J., and Denker, M. Local limit theorems for partial sums of stationary sequences generated by Gibbs-Markov maps, *Stoch. Dyn.* 1 (2001) pp 193–237

[2] Akhavi, A., Vallée, B. Average bit–complexity of Euclidean Algorithms, *Proceedings of ICALP'2000, Lecture Notes in Computer Science* 1853, pp 373-387, Springer.

[3] Babenko, K. I. On a problem of Gauss. *Soviet Mathematical Doklady 19*, 1 (1978), pp 136–140.

[4] Baladi, V. *Positive Transfer operators and decay of correlations,* Advanced Series in non linear dynamics, World Scientific, 2000.

[5] Baladi, V., and Vallée, B. Distributional analyses of Euclidean algorithms, *Proceedings of Alenex–ANALCO'04*, pp 170–184.

[6] Baladi, V., and Vallée, B. Exponential Decay of Correlations for surface semi-flows without finite Markov partitions, to appear in the *Proceedings of the American Mathematical Society*, 2004.

[7] Baladi, V., and Vallée, B. Euclidean Algorithms are Gaussian, to appear in *Journal of Number Theory*.

[8] Bauer, M. and Lopes, A. A billiard in the hyperbolic plane with decay of correlations of type n^{-2}, *Discrete and continuous dynamical systems* 3 (1997) pp 107-116.

[9] Bedford, T., Keane, M., and Series, C., Eds. *Ergodic Theory, Symbolic Dynamics and Hyperbolic Spaces*, Oxford University Press, 1991.

[10] Bernstein, L. The Jacobi-Perron algorithm, its theory and application, Lecture Notes in Mathematics 207, Springer 1971.

[11] Berthé, V. and Nakada, H. On Continued Fraction Expansions in Positive Characteristic: Equivalence Relations and some metric properties. *Expositiones Mathematicae* 18 (2000) pp 257–284.

[12] Bougerol, P., and Lacroix, J. *Products of Random Matrices with Applications to Schrodinger Operators*, Progress in Probability and Statistics, Birkhauser (1985)
[13] Bowen, R. Invariant measures for Markov maps of the interval, *Commun. Math. Phys.* 69 (1979) pp 1–17.
[14] Brent, R.P. Analysis of the binary Euclidean algorithm, *Algorithms and Complexity, New directions and recent results,* ed. by J.F. Traub, Academic Press 1976, pp 321–355
[15] Brent, R. P. and Kung, H.T. A systolic VLSI array for integer GCD computation, *ARITH-7, Proceedings of the Seventh Symposium on Computer Arithmetic (edited by K. Hwang)*, IEEE CS Press, 1985, pp 118–125.
[16] Brentjes, A.J. *Multidimensional continued fraction algorithms.* Mathematical centre tracts 145, Mathematisch Centrum, Amsterdam, 1981
[17] Broise, A. Transformations dilatantes de l'intervalle et théorèmes limites, *Astérisque* 238, pp 5–109, Société Mathématique de France, 1996.
[18] Broise, A. Fractions continues multidimensionnelles et lois stables. Bulletin de la Société Mathématique de France, 124 no. 1 (1996), p. 97-139
[19] Bourdon, J. Daireaux, and B. Vallée, B. Dynamical analysis of α-Euclidean Algorithms, *Journal of Algorithms* 44 (2002) pp 246–285.
[20] Cesaratto, E., and Vallée, B. Hausdorff dimension of reals with bounded weighted averages, *Proceedings of Colloquium on Mathematics and Computer Science: Algorithms, Trees, Combinatorics and Probability, M. Drmota et al., ed.,* pp 473–490, Birkhauser Verlag, Trends in Mathematics, 2004.
[21] Chernov, N. *Markov approximations and decay of correlations for Anosov flows,* Ann. of Math. (2) 147 (1998) 269–324.
[22] Collet, P. Some ergodic properties of maps of the interval, *Dynamical systems, Proceedings of the first UNESCO CIMPA School on Dynamical and Disordered Systems* (Temuco, Chile, 1991), Hermann, 1996.
[23] Daireaux, B. Master Thesis, Université de Caen, 2001.
[24] Daireaux, B., and Vallée, B. Dynamical Analysis of the Parametrized Lehmer-Euclid Algorithm, *Combinatorics, Probability, Computing*, pp 499–536 (2004).
[25] Daireaux, B., Maume-Deschamps, V., and Vallée, B. The Lyapounov Tortoise and the Dyadic Hare, to appear in Discrete Mathematics and Theoretical Computer Science (2005).
[26] Daudé, H., Flajolet, P., and Vallée, B. An average-case analysis of the Gaussian algorithm for lattice reduction, *Combinatorics, Probability and Computing* (1997) 6 pp 397–433
[27] Delange, H. Généralisation du Théorème d'Ikehara, *Ann. Sc. ENS*, (1954) 71, pp 213–242.
[28] Dixon, J. D. The number of steps in the Euclidean algorithm, *Journal of Number Theory 2* (1970), pp 414–422.
[29] Dolgopyat, D. On decay of correlations in Anosov flows, *Ann. of Math.* 147 (1998) pp 357–390.
[30] Daudé, H. and Vallée, B. An upper bound on the average number of iterations of the LLL algorithm, *Theoretical Computer Science* 123 (1994) pp 95-115.
[31] Ellison, W. and Ellison, F. *Prime Numbers,* Hermann, Paris, 1985.
[32] Eisenstein, G. Einfacher Algorithmus zur Bestimmung der Werthes von $(\frac{a}{b})$, *J. für die Reine und Angew. Math.* 27 (1944) pp 317-318.
[33] Flajolet, P. Analytic analysis of algorithms, In *Proceedings of the 19th International Colloquium "Automata, Languages and Programming"*, Vienna, July 1992, W. Kuich, editor, Lecture Notes in Computer Science 623, pp 186–210
[34] Flajolet, P. and Sedgewick, R. Analytic Combinatorics, Book in preparation (1999), see also INRIA Research Reports 1888, 2026, 2376, 2956.
[35] Flajolet, P., and Vallée, B. Continued fraction Algorithms, Functional operators and Structure constants, *Theoretical Computer Science* 194 (1998), pp 1–34.
[36] Flajolet, P., and Vallée, B.Continued Fractions, Comparison Algorithms, and Fine Structure Constants, *Constructive, Experimental et Non-Linear Analysis*, Michel Thera, Editor, Proceedings of Canadian Mathematical Society, Vol 27 (2000), pp 53-82
[37] Finch, S. R. *Mathematical Constants*, Cambridge University Press, 2003.
[38] Friesen, C., and Hensley, D. The statistics of continued fractions for polynomials over a finite field, *Proceedings of the American Mathematical Society*, 124, (1996) 9, pp 2661–2673,
[39] Gouezel, S. Central limit theorem and stable laws for intermittent maps, *Prob. Theory and Related Fields* 128 pp 82–122 (2004)

[40] Grothendieck, A. Produits tensoriels topologiques et espaces nucléaires, *Mem. Am. Math. Soc.* 16 (1955)
[41] Grothendieck, A. La théorie de Fredholm, *Bull. Soc. Math. France* 84 pp 319-384.
[42] Heilbronn, H. On the average length of a class of continued fractions, *Number Theory and Analysis*, ed. by P. Turan, New-York, Plenum, 1969, pp 87-96.
[43] Hennion H. Sur un théorème spectral et son application aux noyaux lipschitziens, *Proc. Amer. Math. Soc.* 118 (1993) pp 627–634
[44] Hensley, D. The number of steps in the Euclidean algorithm, *Journal of Number Theory* 49, 2 (1994), pp 142–182.
[45] Hensley, D. Continued Fraction Cantor sets, Hausdorff dimension, and functional analysis, *Journal of Number Theory* 40 (1992) pp 336-358.
[46] Hensley, D. The statistics of the continued fraction digit sum, *Pacific Journal of Mathematics*, Vol. 192, No2, 2000.
[47] Hwang, H.-K. *Théorèmes limite pour les structures combinatoires et les fonctions arithmétiques,* PhD thesis, Ecole Polytechnique, Dec. 1994.
[48] Hwang, H.-K. Large deviations for combinatorial distributions: I. Central limit theorems, *The Annals of Applied Probability* 6 (1996) pp 297–319.
[49] Hwang, H.-K. On convergence rates in the central limit theorems for combinatorial structures, *European Journal of Combinatorics* 19 (1998) pp 329–343.
[50] Iosifescu, M. and Kraaicamp, C. *Metrical Theory of Continued Fractions.* (2002)
[51] Ito, S. and Tanaka, S. On a family of continued fraction transformations and their ergodic properties *Tokyo J. Math* 4 (1981) pp 153–175.
[52] Jacobi, C.G.J. Uber die Kreistheilung und ihre Anwendung auf die Zalhentheorie, *J. für die Reine und Angew. Math.* 30 (1846) pp 166–182.
[53] Kato, T. *Perturbation Theory for Linear Operators*, Springer-Verlag, 1980.
[54] Khinchin, A. I. *Continued Fractions*. University of Chicago Press, Chicago, 1964. A translation of the Russian original published in 1935.
[55] Kraaikamp, C. and Lopes, A. The Theta group and the continued fraction expansion with even partial quotients. preprint, 1995
[56] Knuth, D.E. The art of Computer programming, Volume 2, 3rd edition, Addison Wesley, Reading, Massachussets, 1998.
[57] Knopfmacher, J. and Knopfmacher, A. The exact length of the Euclidean algorithm in $F_q[X]$, *Mathematika*, 35, (1988), pp 297-304
[58] Krasnoselsky, M. *Positive solutions of operator equations*, P. Noordhoff, Groningen, 1964.
[59] Kuzmin, R. O. Sur un problème de Gauss, *Atti del Congresso Internationale dei Matematici* 6 (Bologna, 1928) pp 83-89.
[60] Lagarias, J. C. *Worst–case complexity bounds for algorithms in the theory of integral quadratic forms*, *Journal of Algorithms 1*, 2 (1980), pp 142–186.
[61] Lebesgue V. A. Sur le symbole (a/b) et quelques unes de ses applications, *J. Math. Pures Appl.* 12 pp 497–517
[62] Lehmer, D. H. Euclid's algorithm for large numbers. *Am. Math. Mon. (1938) 45 pp 227–233.*
[63] Lévy, P. Sur les lois de probabilité dont dépendent les quotients complets et incomplets d'une fraction continue. *Bull. Soc. Math. France* 57 (1929) pp 178-194
[64] Lenstra, A. K., Lenstra, H. W., and Lovász, L. *Factoring polynomials with rational coefficients*, *Mathematische Annalen 261* (1982), pp 513–534.
[65] Lorch, E. R. *Spectral Theory*, Oxford University Press, New York, 1962.
[66] Lhote, L. Master Thesis, Université de Caen, 2002
[67] Lhote, L. PhD Thesis, Université de Caen, (in preparation).
[68] Lhote, L. Computation of a Class of Continued Fraction Constants *Proceedings of Alenex–ANALCO'04*, pp 199–210
[69] Lhote, L., and Vallée, B. Sharp estimates for the main parameters of the Euclid Algorithm. submitted
[70] Lasota, A. and Mackey, M. *Chaos, Fractals and Noise; Stochastic Aspects of Dynamics,* Applied Mathematical Science 97, Springer (1994)
[71] Mayer, D. H. On a ζ function related to the continued fraction transformation, *Bulletin de la Société Mathématique de France 104* (1976), pp 195–203.

[72] Mayer, D. H. Continued fractions and related transformations, In *Ergodic Theory, Symbolic Dynamics and Hyperbolic Spaces*, T. Bedford, M. Keane, and C. Series, Eds. Oxford University Press, 1991, pp. 175–222.
[73] Mayer, D. H. Spectral properties of certain composition operators arising in statistical mechanics, *Commun. Math. Phys.* pp 68, 1-8 (1979)
[74] Mayer, D. H. *On composition Operators on Banach spaces of Holomorphic Functions*, *Journal of functional analysis 35* pp 191-206 (1980)
[75] Mayer, D. H. *On the thermodynamic formalism for the Gauss Map*, *Commun. Math. Phys.* 130, pp 311-333 (1990)
[76] Mayer, D., and Roepstorff, G. On the relaxation time of Gauss's continued fraction map. I. The Hilbert space approach, Journal of Statistical Physics 47, 1/2 (Apr. 1987), pp 149–171. II. The Banach space approach (transfer operator approach), *Journal of Statistical Physics* 50, 1/2 (Jan. 1988), pp 331–344.
[77] Moussa, P., Cassa, A. and Marmi, S., Continued Fractions and Brjuno functions, *Journal of Computational and Applied Mathematics* 105 (1999) pp 403–415.
[78] Nagaev, S.V. Some limit theorems for stationary Markov chains, *Theor. Probab. Appl.* 2 (1957) pp 378–406.
[79] Nakada, H. Metrical Theory for a Class of Continued Fraction Transformations and Their Natural Extensions, *Tokyo J. Math.*, 4 (2) (1981) pp. 399–426.
[80] Philipp, W. Some metrical results in Number Theory II, *Duke Math.* J. 38 (1970) pp 447-488. Errata p 788.
[81] Pollicott, M., and Sharp, R. Exponential error terms for growth functions on negatively curved surfaces, *Amer. J. Math.* 120 (1998) pp 1019–1042.
[82] Prellberg, T. and Slawny, J. Maps of intervals with Indifferent fixed points: Thermodynamic formalism and Phase transitions. *Journal of Statistical Physics* 66 (1992) pp 503-514
[83] Rieger, G. J. Über die mittlere Schrittanzahl bei Divisionalgorithmen, *Math. Nachr.* (1978) pp 157–180.
[84] Rieger, G. J., Über die Schrittanzahl beim Algorithmus von Harris und dem nach nächsten Ganzen, *Archiv der Mathematik* 34 (1980), pp 421–427.
[85] Ruelle, D. *Thermodynamic formalism,* Addison Wesley (1978)
[86] Ruelle, D. *Dynamical Zeta Functions for Piecewise Monotone Maps of the Interval*, vol. 4 of *CRM Monograph Series*, American Mathematical Society, Providence, 1994.
[87] Schwartz, H. Composition operators in $\mathcal{H}^p$, Ph.D. Thesis, Univ. of Toledo.
[88] Shallit, J. On the worst–case of the three algorithmss for computing the Jacobi symbol, *Journal of Symbolic Computation* 10 (1990) pp 593–610.
[89] Shallit, J. Origins of the analysis of the Euclidean Algorithm, *Historia Mathematica* 21 (1994) pp 401-419
[90] Shallit, J. Real numbers with bounded partial quotients. A survey. *L'Enseignement Mathématique*, t. 38, pp 151-187, 1992.
[91] Shapiro, J. *Composition operators and classical function theory*, Universitext: Tracts in Mathematics, Springer-Verlag, 1993.
[92] Shapiro, J. and Taylor, P.D. Compact, nuclear, and Hilbert–Schmidt composition operators on $\mathcal{H}^2$, *Indiana Univ. Math. J.* (1973) 23, pp 471-496
[93] Schonhage, A. Schnelle Berechnung von Kettenbruchentwicklungen, Acta Informatica pp 139–144 (1971)
[94] Schweiger, F. Continued fractions with odd and even partial quotients. Mathematisches Institut der Universitat Salzburg, Arbeitsbericht 2/1982
[95] Schweiger, F. *Multidimensional Continued Fractions,* Oxford University Press, (2000)
[96] Stein, J. Computational Problems Associated with Racah Algebra, *Journal of Computational Physics* 1 (1967) pp 397–405.
[97] Sorenson, J. An analysis of Lehmer's Euclidean GCD Algorithm, *Proceedings of ISSAC 1995, pp 254–258*
[98] Stehlé, D. and Zimmermann, P. A Binary Recursive Gcd Algorithm, *Proceedings of ANTS'04*, Lecture Notes in Computer Science.
[99] Tenenbaum, G. *Introduction à la théorie analytique des nombres*, vol. 13. Institut Élie Cartan, Nancy, France, 1990.
[100] Vallée, B. Gauss' algorithm revisited, *Journal of Algorithms 12* (1991), pp 556–572.
[101] Vallée, B. Opérateurs de Ruelle-Mayer généralisés et analyse des algorithmes d'Euclide et de Gauss, *Acta Arithmetica* 81.2 (1997) pp 101–144.

[102] Vallée, B. Fractions continues à contraintes périodiques, *Journal of Number Theory* 72 (1998) pp 183–235.
[103] Vallée, B. Algorithms for computing signs of 2×2 determinants: dynamics and average-case algorithms, *Proceedings of the 8 th Annual European Symposium on Algorithms,* ESA'97, pp 486–499, LNCS 1284, Springer Verlag.
[104] Vallée, B. Dynamical Sources in Information Theory: Fundamental intervals and Word Prefixes, *Algorithmica* (2001), vol 29 (1/2) pp 262–306
[105] Vallée, B. Dynamics of the Binary Euclidean Algorithm: Functional Analysis and Operators., *Algorithmica* (1998) vol 22 (4) pp 660–685.
[106] Vallée, B. A Unifying Framework for the analysis of a class of Euclidean Algorithms., *Proceedings of LATIN'00,* Lecture Notes in Computer Science 1776, pp 343–354
[107] Vallée, B. Dynamical Analysis of a Class of Euclidean Algorithms, *Theoretical Computer Science,* vol 297/1-3 (2003) pp 447–486
[108] Vallée, B. Digits and Continuants in Euclidean Algorithms. Ergodic Versus Tauberian Theorems, *Journal de Théorie des Nombres de Bordeaux* 12 (2000) pp 531-570.
[109] Vera, A. Master Thesis (2005)
[110] Von Zur Gathen, J. and Gerhard, J. Modern Computer Algebra, Cambridge University Press (1999)
[111] Wirsing, E. On the theorem of Gauss–Kusmin–Lévy and a Frobenius–type theorem for function spaces. *Acta Arithmetica* 24 (1974) pp 507–528.
[112] Yao, A.C., and Knuth, D.E. Analysis of the subtractive algorithm for greatest common divisors. *Proc. Nat. Acad. Sc. USA* 72 (1975) pp 4720-4722.
[113] Yap, C.K. *Fundamental Problems in Algorithmic Algebra*, Princeton University Press (1996)

Received December 2004; revised October 2005.

E-mail address: brigitte.vallee@info.unicaen.fr