Analysis of the Brun Gcd Algorithm

Valérie Berthé
IRIF (CNRS and Université Paris Diderot), France
berthe@liafa.univ-paris-diderot.fr

Loïck Lhote
GREYC (CNRS, ENSICAEN, and Université of Caen), France
loick.lhote@ensicaen.fr

Brigitte Vallée
GREYC (CNRS, ENSICAEN, and Université of Caen), France
brigitte.vallee@unicaen.fr

ABSTRACT

We introduce and study a multiple gcd algorithm that is a natural extension of the usual Euclid algorithm, and coincides with it for two entries; it performs Euclidean divisions, between the largest entry and the second largest entry, and then re-orderings. This is the discrete version of a multidimensional continued fraction algorithm due to Brun. We perform the average-case analysis of this algorithm, and prove that the mean number of steps is linear with respect to the size of the entry. The method relies on dynamical analysis, and is based on the study of the underlying Brun dynamical system. The dominant constant of the analysis is related to the entropy of the system. We also compare this algorithm to another extension of the Euclid algorithm, proposed by Knuth, and already analyzed by the authors.

Keywords

GCD algorithms; Brun continued fractions; average-case analysis; analytic combinatorics; Dirichlet generating functions; dynamical analysis; transfer operators; Tauberian theorems

Acknowledgments. Thanks to the ANR DynA3S (ANR-13-BS02-0003) and AleaEnAmSud AmSud-STIC Projects.

1. INTRODUCTION.

General Context. We study a multiple gcd algorithm that is a natural extension of the usual Euclid algorithm for $(d+1)$ integers, and coincides with it for $d = 1$. This is a discrete version of a multidimensional continued fraction algorithm, that is itself based on a dynamical system, the Brun dynamical system, described for instance in [6, 12]. The Brun continued fraction algorithm admits various descriptions and appears under various names; it is closely related to the Podsypanin modified Jacobi–Perron algorithm, it is also called the d-dimensional Gauss transformation or the ordered Jacobi–Perron algorithm [7].

This dynamical system belongs to the class of multidimensional unimodular continued fraction algorithms, described in [12], which produce simultaneous diophantine approximations of a real vector. Then, the literature, for instance in [9, 5], mainly focuses on the convergence of these approximations, closely related to the Lyapunov exponents of the underlying dynamical systems. These algorithms have also important applications to discrete geometry [3].

With each multidimensional continued fraction algorithm, a gcd algorithm may be of course associated. However, the analysis of such a class of gcd algorithms has not been yet considered. This is our general project, and we begin by the BrunGcd Algorithm, as it is one of the most "natural" algorithms of this class, and shares many important properties with the Euclid dynamical system based on the Gauss map.

Main results. We perform the probabilistic analysis of the BrunGcd algorithm. We first focus on the total number of steps and prove that it is on average linear in the size of the entries. The dominant constant equals $(d+1)/\mathcal{E}_d$ where $\mathcal{E}_d$ is the entropy of the Brun dynamical system. This entropy is not precisely studied in the literature, but there exists a conjecture in [7] that states that $\mathcal{E}_d$ is $\Theta(1)$ for $d \to \infty$. However, we show that $\mathcal{E}_d$ is $\sim \log d$ for $d \to \infty$, so that the dominant constant for the number of steps grows as $d/\log d$.

We then compare the BrunGcd with another multiple gcd algorithm, the PlainGcd algorithm, described in Knuth's book [8], which deals with the classical one-dimensional Euclid dynamical system. The authors have already analyzed this algorithm in [4], and prove that the mean number of steps is also linear in the size of the entries. However, the dominant constant is independent of the dimension d and equals $2/\mathcal{E}_1$, where $\mathcal{E}_1$ is the entropy of the Euclid dynamical system. We conclude that the PlainGcd algorithm is much more efficient than the BrunGcd algorithm, in particular for large d.

We finally explain the inefficiency of the BrunGcd algorithm when d is large: almost all divisions deal with a quotient equal to 1. Then the main operation performed is not a division but... a plain subtraction. This is reinforced by the comparison with the subtractive version of the algorithm, whose number of steps is proven to be also of linear complexity. This exhibits two strong differences with the classical Gcd algorithm (case $d = 1$).

Methods. We use here the methods of *dynamical analysis* such as developed in [2, 10, 13]: a gcd algorithm is viewed as a dynamical system, with each iterative step being a linear fractional transformation. Costs of interest are then described with Dirichlet generating functions that are algebraically related to transfer operators of the system. The main analytical property of these series is the existence of a dominant pole, which is itself closely related to the existence

ISSAC '16, July 19–22, 2016, Waterloo, ON, Canada.
© 2016 ACM. ISBN 978-1-4503-4380-0/16/07…$15.00
DOI: http://dx.doi.org/10.1145/2930889.2930899

of a spectral gap for the corresponding transfer operators. The asymptotic extraction of coefficients is then achieved by means of Tauberian theorems.

Plan of the paper. We first introduce the algorithm in Section 2 and state the main results. Then, we study in Section 3 the underlying dynamical system, closely related to the Brun dynamical system, and we provide a characterization of its rational trajectories. The following two sections perform the dynamical analysis of the algorithm, with its two steps, the combinatorial step (Section 4) and the analytic step (Section 5). The paper ends with open problems.

2. THE BRUN GCD ALGORITHM.

We describe the `BrunGcd` algorithm, state the main complexity results, and compare it with the `PlainGcd` algorithm.

2.1 General description.

The algorithm `BrunGcd`(d) computes the gcd of $(d+1)$ positive integers. It deals with the input set $\Omega_{(d)}$ which gathers the *ordered* $(d+1)$-uples $\boldsymbol{u}$ formed with *positive* and *distinct* integer numbers

$$\Omega_{(d)} := \{\boldsymbol{u} = (u_0, u_1, \dots, u_d) \mid u_0 > u_1 > u_2 > \dots > u_d > 0\}\,.$$

During the execution of the algorithm, some components "disappear" and the algorithm deals with the *disjoint union*

$$\Gamma_{(d)} = \bigoplus_{\ell=0}^{d-1} \Omega_{(d-\ell)}\,.$$

The algorithm `BrunGcd`(d) performs a sequence of steps, and each step deals with the pair (u_0, u_1) (that contains the two largest entries of $\boldsymbol{u}$) and the list $\mathtt{End}\,\boldsymbol{u}$ which gathers all the components of $\boldsymbol{u}$ except u_0; it divides the first component u_0 by the second component u_1, and creates a remainder v_0

$$v_0 := u_0 - m u_1, \quad m := \left\lfloor \frac{u_0}{u_1} \right\rfloor\,.$$

Then, the procedure $\mathtt{InsDis}\,(v_0, \mathtt{End}\,\boldsymbol{u})$ inserts $v_0 \geq 0$ at a suitable position inside the list $\mathtt{End}\,\boldsymbol{u}$, so that the result remains an ordered uple of distinct positive values: the second component u_1 becomes the largest one, and there are three possible cases for the insertion (or not) of v_0:

(G) (Generic case) if v_0 is not present in the list $\mathtt{End}\,\boldsymbol{u}$, this is a usual insertion;

(Z) (Zero case) if $v_0 = 0$, we *do not* insert v_0;

(E) (Equality case) if $v_0 \neq 0$ is already present in the list $\mathtt{End}\,\boldsymbol{u}$ at position i, we *do not* insert v_0.

In each of the cases (Z) or (E), we do not insert v_0, but we memorize the potential insertion position (in case (E), we would have inserted v_0 "in front of" u_i). Finally, each *step* of the algorithm `BrunGcd`(d) is described by the map $U_{(d)} : \Gamma_{(d)} \to \Gamma_{(d)}$ which associates with $\boldsymbol{u}$

$$U_{(d)}(\boldsymbol{u}) = \mathtt{InsDis}\,(u_0 \bmod u_1, \mathtt{End}\,\boldsymbol{u})\,. \qquad (1)$$

The algorithm `BrunGcd`(d) described in the following figure decomposes into d *phases*, labelled from $\ell = 0$ to $\ell = d-1$. During each phase, a component is "lost", and the ℓ-th phase, denoted by $\mathtt{BrunGcd}_{(d,\ell)}$, transforms an element of $\Omega_{(d-\ell)}$ into an element of $\Omega_{(d-\ell-1)}$. The phase ends as soon as it meets case (Z), or[1] case (E), where it looses a component. The algorithm stops at the end of the $(d-1)$-th phase with an element of $\Omega_{(0)}$ which equals the gcd.

```
BrunGcd$(d)$
Input : $\boldsymbol{u} \in \Omega_{(d)}$
Ouput: $\boldsymbol{u} \in \mathbb{N}$
For $\ell = 0$ to $d-1$ do $\boldsymbol{u} := \mathtt{Gcd}_{(d,\ell)}(\boldsymbol{u})$;

BrunGcd$_{(d,\ell)}$
Input : $\boldsymbol{u} \in \Omega_{(d-\ell)}$
Repeat $\boldsymbol{u} := U_{(d)}(\boldsymbol{u})$ until $u \in \Omega_{(d-\ell-1)}$.
```

2.2 Worst-case behavior.

As will be shown in the long version paper, the worst-case of the `BrunGcd` algorithm arises when the quotients are the smallest possible (all equal to 1, except the last one, equal to 2), and the insertion positions the largest possible. Then, the best worst-case bound involves an algebraic number τ_d which extends to general dimensions the inverse of the Golden ratio.

PROPOSITION 1. *For any fixed positive integer d, consider the smallest real root $\tau_d \in]0, 1[$ of the polynomial $z^{d+1} + z - 1$, and, for any integer N, the set of inputs*

$$\Omega_{(d,N)} := \{\boldsymbol{u} \in \Omega_{(d)} \mid u_0 \leq N\}\,. \qquad (2)$$

Then the maximum number $Q_{(d,N)}$ of steps of the `BrunGcd` *Algorithm on $\Omega_{(d,N)}$ satisfies*

$$Q_{(d,N)} \sim \frac{1}{|\log \tau_d|} \log N \qquad (N \to \infty).$$

When $d \to \infty$, the real τ_d satisfies $1/|\log \tau_d| \sim (d+1)/\log d$.

2.3 Probabilistic behavior.

We now describe the precise probabilistic behavior of the algorithm `BrunGcd`(d) on the set $\Omega_{(d,N)}$ defined in (2).

THEOREM 1. *When the algorithm* `BrunGcd` *acts on the set $\Omega_{(d,N)}$ endowed with the uniform distribution, the following holds when d is fixed and N tends to ∞:*

(a) *The total number L_d of steps and the number M_d of steps performed during the first phase satisfies*

$$\mathbb{E}_N[L_d] \sim \mathbb{E}_N[M_d] \sim \frac{d+1}{\mathcal{E}_d} \cdot \log N$$

and involves the entropy $\mathcal{E}_d$ of the underlying Brun dynamical system in dimension d.
When $d \to \infty$, the entropy $\mathcal{E}_d$ satisfies $\mathcal{E}_d \sim \log d$.
The number R_d of steps performed during the remainder of the execution (after the first phase) has a mean value that is asymptotic to a constant r_d.

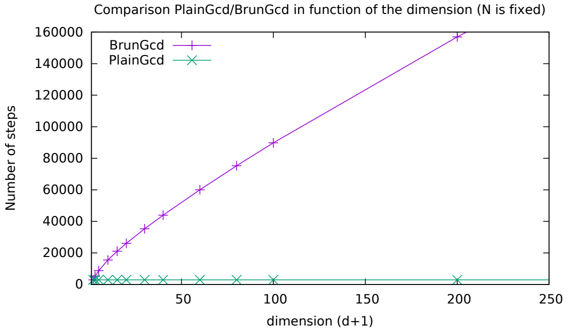
(b) *Let O_d be the number of quotients equal to 1 during the first phase. The ratio between the means $\mathbb{E}_N[O_d]$ and $\mathbb{E}_N[M_d]$ is asymptotic to a constant $\rho_d < 1$, that tends to 1 for $d \to \infty$ with a speed $O(2^{-d/\log d})$.*
Let Σ_d be the number of steps of the subtractive version of `BrunGcd` *during the first phase. The ratio between $\mathbb{E}_N[\Sigma_d]$ and $\mathbb{E}_N[M_d]$ is asymptotic to a constant σ_d[2].*

[1] The $(d-1)$-th phase always ends with the case (Z).

[2] We conjecture the asymptotics $\sigma_d \sim \zeta(d/\log d)$ for $d \to \infty$.

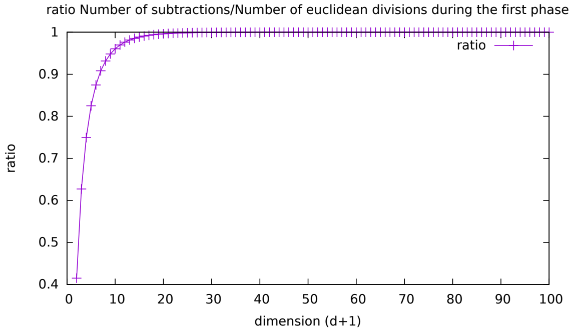
In (a), the total number of steps L_d is proven to remain on average linear in the size $\log N$. Moreover, (a) exhibits a strong difference between the first phase, where most of the work is done, and the remainder of the execution, where the total number of steps R_d is on average asymptotically constant. The PlainGcd algorithm exhibits exactly the same phenomena, as it is shown in [4].

The following figure compares the number of steps of the BrunGcd and the PlainGcd algorithms, as a function of dimension d, when the binary size is fixed to $\log_2 N = 5000$. The complexity of BrunGcd algorithm appears to be sublinear with respect to d (see Section 5), whereas the complexity of the PlainGcd algorithm appears to be independent of d.



The two dominant constants, the ratio $(d+1)/\mathcal{E}_d$ which involves the entropy and intervenes in the average case, and the ratio $1/|\log \tau_d|$ which arises in the worst-case, both behave as $d/\log d$ for $d \to \infty$. This indicates the same behavior for the algorithm in the average-case and in the worst-case. As the worst-case is reached when the quotients are all equal to 1, this seems to indicate that the BrunGcd Algorithm deals with quotients which are very often equal to 1.

This is indeed the case, described in (b), and also illustrated in the following figure, that exhibits the proportion of quotients equal to 1 during the first phase as a function of the dimension d. This proportion tends quickly to 1: when $d = 16$, more than 99% of the Euclidean divisions are in fact subtractions and for $d = 50$, the proportion is 99.99%.



3. THE UNDERLYING DYNAMICAL SYSTEM.

We first describe a continuous extension of the algorithm, relate it with the Brun dynamical system, and provide an exact characterization of the trajectories that are related to the execution of the algorithm.

3.1 Continuous extension of the Gcd algorithm.

We now extend the BrunGcd(d) algorithm into a continuous process. We use the projection π defined on $\Gamma_{(d)} \setminus \{0\}$ as

$$\pi(\boldsymbol{u}) = \frac{1}{u_0}\texttt{End}\,\boldsymbol{u}\,,$$

and the closure of the image $\pi(\Gamma_d)$ is exactly the disjoint union $\mathcal{I}_{(d)}$ of simplexes $\mathcal{J}_{(d-\ell)}$, defined as

$$\mathcal{J}_{(d-\ell)} := \{\boldsymbol{x} = (x_1, \ldots, x_{d-\ell}) \mid 1 \geq x_1 \geq \ldots \geq x_{d-\ell} \geq 0\}\,,$$

and is written as $\mathcal{I}_{(d)} := \bigoplus_{\ell=0}^{d-1} \mathcal{J}_{(d-\ell)}$. Now, the map $V_{(d)} : \mathcal{I}_{(d)} \to \mathcal{I}_{(d)}$ is defined on each $\mathcal{J}_{(d-\ell)}$ by $V_{(d)}(\mathbf{0}^{\boldsymbol{d}-\ell}) = \mathbf{0}^{\boldsymbol{d}-\ell}$,

$$\text{and} \quad V_{(d)}(\boldsymbol{x}) = \texttt{InsDis}\left(\left\{\frac{1}{x_1}\right\}, \frac{1}{x_1}\texttt{End}\,\boldsymbol{x}\right) \quad \text{for } \boldsymbol{x} \neq \mathbf{0}^{\boldsymbol{d}-\ell}\,,$$

where $\texttt{InsDis}(y_0, \boldsymbol{y})$ is now extended to $\mathcal{I}_{(d)}$. The map $V_{(d)}$ provides the extension of the conjugate with the projection π of the map $U_{(d)}$ defined in (1) and used in the BrunGcd(d) algorithm. Indeed, the equality $V_{(d)} \circ \pi(\boldsymbol{u}) = \pi \circ U_{(d)}(\boldsymbol{u})$ leads to the definition of $V_{(d)}$ on the set $\pi(\Gamma_d)$ which is further extended to $\mathcal{I}_{(d)}$ "by continuity".

The dynamical system $(V_d, \mathcal{I}_{(d)})$ is closely related to the usual Brun dynamical system defined on the simplex $\mathcal{J}_{(d)}$ with the transformation $T_{(d)}$ defined by $T_{(d)}(\mathbf{0}^{\boldsymbol{d}}) = \mathbf{0}^{\boldsymbol{d}}$, and

$$T_{(d)}(\boldsymbol{x}) = \texttt{Ins}\left(\left\{\frac{1}{x_1}\right\}, \frac{1}{x_1}\texttt{End}\,\boldsymbol{x}\right) \quad \text{for } \boldsymbol{x} \neq \mathbf{0}^{\boldsymbol{d}}\,, \qquad (3)$$

where now the map $\texttt{Ins}(y_0, \boldsymbol{y})$ is the usual insertion "in front of": it performs as InsDis, *without* removing zeroes and equal components, and the cases (Z) and (E) may be gathered into a unique case (ZE). There are now only two cases:

(G) if y_0 is not present in the list $\boldsymbol{y}$, this is an usual insertion;

(ZE) if y_0 is already present in the list $\boldsymbol{y}$, we insert y_0 *in front of the block* of components equal to y_0.

It is clear that $V_{(d)}(\boldsymbol{x}) = T_{(d)}(\boldsymbol{x})$ except in case (ZE). Then, the two dynamical systems coincide "almost everywhere" and the trajectories which do not meet case (ZE) will be the same for the two systems. But we are mainly interested in rational trajectories, and the rational trajectories may meet case (ZE) and differ in the two systems. For instance, the two trajectories of the input $\pi(4, 3, 2, 1) = (3/4, 2/4, 1/4)$ are (the inserted component is in bold):

for $V_{(d)}$: $(3/4, 2/4, 1/4) \to (2/3, 1/3) \to (1/2, 1/3) \to$
$(1/3) \to (0)$

for $T_{(d)}$: $(3/4, 2/4, 1/4) \to (2/3, \mathbf{1/3},\ 1/3) \to$
$(\mathbf{1/2}, 1/2, 1/2) \to (1, 1, \mathbf{0}) \to (1, \mathbf{0}, 0) \to (\mathbf{0}, 0, 0)\,.$

However, we will prove in Section 3.3 that a rational trajectory under $V_{(d)}$ –which exactly describes an execution of the BrunGcd(d) algorithm– mainly decomposes into rational trajectories under $T_{(d-\ell)}$. Indeed, except possibly at the end of each phase, it uses Ins and not InsDis. This is why the BrunGcd algorithm will use, except at the end of each phase, the Brun dynamical system, which we now describe.

3.2 The Brun dynamical system.

The pair $(T_{(d)}, \mathcal{J}_{(d)})$ defines a dynamical system denoted as $\mathcal{D}_{(d)}$. For any $\boldsymbol{x} \in \mathcal{J}_{(d)}$, the map $T_{(d)}$ defined in (3) uses *digit* (m, j) formed with a quotient $m(\boldsymbol{x}) \geq 1$ and an insertion index $j(\boldsymbol{x}) \in [1..d]$. The set of digits is thus $\mathcal{A}_{(d)} := \mathbb{N}^* \times [1..d]$. We associate with (m, j) the subset

$$\mathcal{K}_{(d,m,j)} := \{\boldsymbol{x} \in \mathcal{J}_{(d)} \mid m(\boldsymbol{x}) = m, \quad j(\boldsymbol{x}) = j\}.$$

When (m, j) varies in $\mathcal{A}_{(d)}$, the subsets $\mathcal{K}_{(d,m,j)}$ form a topological partition of $\mathcal{J}_{(d)}$, and the restriction $T_{(d,m,j)}$ of $T_{(d)}$ to $\mathcal{K}_{(d,m,j)}$ is a bijection from $\mathcal{K}_{(d,m,j)}$ onto $\mathcal{J}_{(d)}$, written as

$$T_{(d,m,j)}(x_1, x_2, \ldots, x_d) =$$

$$\left(\frac{x_2}{x_1}, \ldots, \frac{x_{j-1}}{x_1}, \frac{1}{x_1} - m, \frac{x_{j+1}}{x_1}, \ldots, \frac{x_d}{x_1}\right).$$

Its inverse is a bijection from $\mathcal{J}_{(d)}$ onto $\mathcal{K}_{(d,m,j)}$ written as

$$h_{(d,m,j)}(y_1, \ldots, y_d) =$$

$$\left(\frac{1}{m + y_j}, \frac{y_1}{m + y_j}, \ldots, \frac{y_{j-1}}{m + y_j}, \frac{y_{j+1}}{m + y_j}, \ldots, \frac{y_d}{m + y_j}\right). \quad (4)$$

Any inverse branch of the map $T_{(d)}$ is called an *elementary inverse branch* (or an *inverse branch of depth one*) and the set of the elementary inverse branches is then

$$\mathcal{H}_{(d)} := \left\{h_{(d,m,j)} \mid (m, j) \in \mathcal{A}_{(d)}\right\}, \quad (5)$$

whereas the inverse branches of the map $T^k_{(d)}$ are said to be of *depth* k and belong to the set $\mathcal{H}^k_{(d)}$. We thus define

$$\mathcal{H}^\star_{(d-\ell)} := \bigoplus_{k \geq 0} \mathcal{H}^k_{(d-\ell)}.$$

3.3 Return to the Gcd Algorithm

On an input $\boldsymbol{u} \in \Omega_{(d)}$, the execution of the $\texttt{BrunGcd}(d)$ algorithm is described by the trajectory of $\boldsymbol{u}$ under the map $U_{(d)}$ which ends at $\gcd(\boldsymbol{u})$. It proves useful to consider one more step, and now, the trajectory of $\boldsymbol{u}$ under the map $U_{(d)}$ ends at 0. It gives rise to the rational trajectory of the vector $\boldsymbol{x} := \pi(\boldsymbol{u})$ under $V_{(d)}$, that also ends at 0. This trajectory uses at each step a branch of the map $V_{(d)}$. And we wish to precisely describe the set $\mathcal{B}_{(d)}$ of compositions of inverse branches of the map $V_{(d)}$ which are possibly used by such particular trajectories. Then the equality $\pi(\boldsymbol{u}) = h(0)$ (for $h \in \mathcal{B}_{(d)}$) gives rise to a bijection between $\pi(\Omega_{(d)})$ and $\mathcal{B}_{(d)}$. Moreover, there is also a bijection between $\pi(\Omega_{(d)})$ and the set of coprime inputs

$$\underline{\Omega}_{(d)} := \{u \in \Omega_{(d)} \mid \gcd(\boldsymbol{u}) = 1\}, \quad (6)$$

and thus a bijection between $\mathcal{B}_{(d)}$ and $\underline{\Omega}_{(d)}$.

We now describe $\mathcal{B}_{(d)}$, and first focus, for each $\ell \in [0..d-1]$, on the set $\mathcal{P}_{(d-\ell)}$ used by the part of the trajectory associated with the ℓ-th phase, when the iterates of $\boldsymbol{u}$ under $U_{(d)}$ belong to $\Omega_{(d-\ell)}$. Then, the iterates of $\boldsymbol{x}$ under $V_{(d)}$ belong to the simplex $\mathcal{J}_{(d-\ell)}$. We study in a separate way

(a) the steps of the ℓ-phase, which are not the last one,

(b) and the last step of the ℓ-phase.

Consider first (a). In this case, such a step does not loose a component, and only involves a step of type (G). Then, the trajectory uses branches of the map $T_{(d-\ell)}$, and each step uses possibly any inverse branch in the set $\mathcal{H}_{(d-\ell)}$ defined in (5). Hence, the set of the inverse branches used during the ℓ-phase, except in the final step, is the set $\mathcal{H}^\star_{(d-\ell)}$.

Consider (b). Now, a component is lost since the insertion is not done, and there are two possible cases.

Case (Z). The quotient $1/x_1$ is equal to an integer $m \geq 2$, the position of potential insertion is $j = d - \ell$. The equality $V_{(d)}(\boldsymbol{x}) = m \cdot \texttt{End}\,\boldsymbol{x}$ holds and the inverse branch associates with the $(d - \ell - 1)$-uple $\boldsymbol{y}$ the $(d - \ell)$-uple

$$z_{(d-\ell,m)}(\boldsymbol{y}) = \frac{1}{m}(1, \boldsymbol{y}). \quad (7)$$

The set $\mathcal{Z}_{(d-\ell)}$ of inverse branches used in case (Z) is thus

$$\mathcal{Z}_{(d-\ell)} = \{z_{(d-\ell,m)} \mid m \geq 2\}.$$

Case (E). An equality of the form $(1/x_1) - m = x_i/x_1$ holds with a quotient $m \geq 1$ and a potential insertion position[3] $j = i - 1 < d - \ell$. Then, Case (E) cannot occur for $\ell = d - 1$. The equality $V_{(d)}(\boldsymbol{x}) = (\texttt{End}\,\boldsymbol{x})/(1 - x_i)$ holds and the inverse branch associates with the $(d - \ell - 1)$-uple $\boldsymbol{y}$ the $(d - \ell)$-uple

$$s_{(d-\ell,m,j)}(\boldsymbol{y}) = \frac{1}{m + y_j}(1, \boldsymbol{y}). \quad (8)$$

The set of inverse branches $\mathcal{S}_{(d-\ell)}$ used in case (E) is thus the empty set for $\ell = d - 1$ and

$$\mathcal{S}_{(d-\ell)} = \{s_{(d-\ell,m,j)} \mid m \geq 1, \, j < d - \ell\} \quad (\text{for } \ell < d - 1).$$

In summary, the set of possible inverse branches used during the last step of the ℓ-th phase is $\mathcal{F}_{(d-\ell)} = \mathcal{S}_{(d-\ell)} \cup \mathcal{Z}_{(d-\ell)}$.

Finally, we have proven the following.

PROPOSITION 2. *The* $\texttt{BrunGcd}(d)$ *algorithm builds a bijection between the set* $\underline{\Omega}_{(d)}$ *of coprime inputs of* $\Omega_{(d)}$ *and the set* $\mathcal{B}_{(d)}$ *of inverse branches possibly used by the rational trajectories of the shift* $V_{(d)}$*. This set is written as*

$$\mathcal{B}_{(d)} = \mathcal{P}_{(d)} \circ \mathcal{P}_{(d-1)} \circ \ldots \circ \mathcal{P}_{(1)} = \mathcal{P}_{(d)} \circ \mathcal{B}_{(d-1)}$$

and involves the sets $\mathcal{P}_{(d-\ell)}$ *of inverse branches used by the* $\texttt{BrunGcd}_{(d,\ell)}$*, characterized as*

$$\mathcal{P}_{(d-\ell)} = \mathcal{H}^\star_{(d-\ell)} \circ \mathcal{F}_{(d-\ell)}.$$

3.4 Why InsDis rather than Ins ?

We have decided to deal with the set $\Gamma_{(d)}$. We are then led to use $\texttt{InsDis}$ to stay inside $\Gamma_{(d)}$. But there is a natural question: why not dealing with the set $\Gamma_{(=,d)}$ with possible blocks of equal non-zero components? Then, we could use $\texttt{Ins}$ and stay inside $\Gamma_{(=,d)}$. This defines another algorithm, the $\texttt{BrunGcd}_=(d)$ algorithm, whose continuous extension directly leads to the Brun dynamical system. Even though the whole path seems more natural, we have to memorize the position of each block of equal components, and this leads to a quite involved analogue set $\mathcal{B}_{(=,d)}$ that describes the rational trajectories of the $\texttt{BrunGcd}_=(d)$ algorithm.

4. DYNAMICAL ANALYSIS (I).

We now begin the analysis of the algorithm, and introduce the main objects: the class of costs of interest, the (Dirichlet) generating functions, the generating operators. The main result of this section (Theorem 2) relates generating functions and generating operators.

4.1 Additive costs

We consider here costs that are said to be additive. One begins with a nonnegative *elementary* cost c defined on each inverse branch in $\mathcal{B}_{(d)}$ of depth one. Such a cost is then extended in an additive way on $\mathcal{B}_{(d)}$, namely

$$c(h_1 \circ h_2 \circ \cdots \circ h_p) := \textstyle\sum_{i=1}^p c(h_i).$$

[3] We recall that we would have inserted "in front of".

Now, a cost C defined on $\Omega_{(d)}$ is said to be *additive* if it is associated with such a cost c, and satisfies

$$C(\boldsymbol{u}) := c(h) \qquad \text{when } \pi(\boldsymbol{u}) = h(0).$$

Then $C(\boldsymbol{u})$ equals the total cost of c of the trajectory of $\pi(\boldsymbol{u})$ and satisfies $C(\boldsymbol{u}) = C(\lambda \boldsymbol{u})$ for any integer $\lambda \neq 0$. There are three main additive costs of interest here.

(i) The total number $C_{d,\ell}$ of steps during the ℓ-th phase (except the final step), associated with the characteristic function c of the set $\mathcal{H}_{(d-\ell)}$; with the family $C_{d,\ell}$, we return to the number of steps mentioned in Theorem 1, as the following relations hold: $L_d = M_d + R_d$

$$M_d = 1 + C_{d,0}, \qquad R_d = (d-1) + \sum_{\ell=1}^{d-1} C_{d,\ell}\,. \quad (9)$$

(ii) The number O_d of steps of the first phase with a quotient equal to 1, associated with the characteristic function c of the subset $\mathcal{H}^{(1)}_{(d)}$ with quotients equal to 1.

(iii) The number Σ_d of steps of the subtractive algorithm during the first phase, associated with the cost c which associates with an inverse branch h of the first phase its quotient m.

The associated elementary costs deal with a specific phase[4]: the ℓ-phase for $C_{d,\ell}$ and the first phase $\ell = 0$ for M_d, O_d, R_d. The cost C is said to *be concentrated* on this phase. inverse,

4.2 Dirichlet generating functions.

The (basic) Dirichlet generating function $S_{(d)}(s)$, of the input set $\Omega_{(d)}$ relative to the length $\|\boldsymbol{u}\| := u_0$, is defined as

$$S_{(d)}(s) := \sum_{\boldsymbol{u}\in\Omega_d} \frac{1}{||\boldsymbol{u}||^s}\,. \quad (10)$$

In the same vein, with a cost $C : \Omega_{(d)} \to \mathbb{N}$, we associate the cumulative generating function of the cost

$$\widehat{S}_{(d,C)}(s) := \sum_{\boldsymbol{u}\in\Omega_d} \frac{C(\boldsymbol{u})}{||\boldsymbol{u}||^s} = \sum_{n\geq 1} n^{-s} \sum_{\|\boldsymbol{u}\|=n} C(\boldsymbol{u})\,. \quad (11)$$

When $\Omega_{(d,N)}$ is endowed with the uniform distribution, the mean $\mathbb{E}_N[C]$ of cost C on $\Omega_{(d,N)}$ is expressed with the coefficients of the cumulative generating function, as

$$\mathbb{E}_N[C] = \frac{1}{\Phi_N\left[S_{(d)}\right]} \Phi_N\left[\widehat{S}_{(d,C)}\right]\,, \quad (12)$$

where $\Phi_N[f]$ is defined as

$$\Phi_N[f] = \sum_{n\leq N} a_n \quad \text{when} \quad f(s) = \sum_{n\geq 1} \frac{a_n}{n^s}. \quad (13)$$

Here, singularity analysis uses the Delange Tauberian Theorem (stated in Section 5) which relates the asymptotics of the coefficients of a generating function to the nature and the position of its dominant singularity. Then, we need an alternative expression of the series which highlights the singularities. Such an expression exists for the series $S_{(d)}(s)$,

$$S_{(d)}(s) = \frac{1}{d!} \sum_{n\geq d+1} \frac{1}{n^s}(n-1)(n-2)\dots(n-d).$$

As its "dominant" series is $(1/d!)\zeta(s-d)$, the series $S_{(d)}(s)$ has a dominant simple pole at $s = d+1$ with residue $1/d!$. However, such a "dominant" behavior is not known for the other Dirichlet series. The remaining of the paper is devoted to this task and is based on the dynamical analysis approach where generating functions are expressed with generating transfer operators. For this aim, we introduce an intermediate tool, the Dirichlet (bivariate) generating function,

$$S_{(d,C)}(s,w) := \sum_{\boldsymbol{u}\in\Omega_d} \frac{w^{C(\boldsymbol{u})}}{||\boldsymbol{u}||^s}\,. \quad (14)$$

whose derivative is related to the cumulative generating function: the relation $\widehat{S}_{(d,C)} = \Delta[S_{(d,C)}]$ holds and involves the functional Δ defined as

$$\Delta[A](s) := \frac{\partial}{\partial w} A(s,w)\Big|_{w=1}\,. \quad (15)$$

As we wish to use the bijection described in Proposition 2, we also introduce the three underlined series that are the exact counterparts of series defined in (10), (11), (14), now with respect to the set $\underline{\Omega}_{(d)}$ of coprime inputs. As C is an additive cost, the non-underlined series are related to their underlined conterparts as

$$\frac{S_{(d)}(s)}{\underline{S}_{(d)}(s)} = \frac{S_{(d,C)}(s,w)}{\underline{S}_{(d,C)}(s,w)} = \frac{\widehat{S}_{(d,C)}(s)}{\underline{\widehat{S}}_{(d,C)}(s)} = \zeta(s)\,. \quad (16)$$

4.3 Generating operators

Let $\mathcal{G}$ be a set of inverse branches; we say that h is a *factor* of $\mathcal{G}$ if h is a factor of an element of $\mathcal{G}$, and we denote this situation as $h \propto \mathcal{G}$. In the same vein, a set $\mathcal{H}$ is said to be a factor of $\mathcal{G}$ if each element of $\mathcal{H}$ is a factor of $\mathcal{G}$, and this will be denoted as $\mathcal{H} \propto \mathcal{G}$. We will consider in the sequel factors of $\mathcal{B}_{(d)}$. The following easy result is central in this work.

LEMMA 1. *Any inverse branch $h \propto \mathcal{B}_{(d)}$ is a linear fractional transformation (LFT), and any factor of $\mathcal{P}_{(d-\ell)}$ can be written as* $\quad h = \dfrac{1}{D[h]}\Big(N_1[h], N_2[h], \dots, N_{d-\ell}[h]\Big)$,

where the denominator $D[h]$ and the numerators $N_i[h]$ are co-prime affine functions. When $h \in \mathcal{H}_{(d-\ell)}$, its Jacobian $J[h]$ is related to the denominator $D[h]$ as

$$J[h] = |D[h]|^{-(d-\ell+1)}\,. \quad (17)$$

The proof (by recurrence on the depth of h) begins with the expression of the branches of depth 1 given in (4),(7),(8).

Transfer operators are central tools for studying probabilistic properties of trajectories in dynamical systems; see e.g. [2, 10, 13]. Here, we adapt these tools to our needs, strongly use the fact that inverse branches are LFT, and also consider an additive cost c. It proves useful to deal with two parameters s and w, and we define the *generating operator* $\mathbf{G}_{s,w,[c],\langle h\rangle}$ of the inverse branch $h \propto \mathcal{B}_{(d)}$ as

$$\mathbf{G}_{s,w,[c],\langle h\rangle}[f](\boldsymbol{x}) := |D[h](\boldsymbol{x})|^{-s}\, w^{c(h)} f\circ h(\boldsymbol{x})\,.$$

Then, the operator may be extended for a subset $\mathcal{G} \propto \mathcal{B}_{(d)}$,

via the equality $\quad \mathbf{G}_{s,w,[c],\langle\mathcal{G}\rangle} := \sum_{h\in\mathcal{G}} \mathbf{G}_{s,w,[c],\langle h\rangle}\,,$

and the following relation is valid for disjoint factors of $\mathcal{B}_{(d)}$:

[4] except with its final step

$$\mathbf{G}_{s,w,[c],\langle \mathcal{G}_1+\mathcal{G}_2\rangle} = \mathbf{G}_{s,w,[c],\langle \mathcal{G}_1\rangle} + \mathbf{G}_{s,w,[c],\langle \mathcal{G}_2\rangle} \ .$$

If now $\mathcal{G}_1$ and $\mathcal{G}_2$ may be composed and satisfy $(\mathcal{G}_1 \circ \mathcal{G}_2) \propto \mathcal{B}_{(d)}$, multiplicative properties of the denominator, together with additive properties of the cost, entail the equality

$$\mathbf{G}_{s,w,[c],\langle \mathcal{G}_1\circ\mathcal{G}_2\rangle} = \mathbf{G}_{s,w,[c],\langle \mathcal{G}_2\rangle} \circ \mathbf{G}_{s,w,[c],\langle \mathcal{G}_1\rangle} \ .$$

Then, there exists a "dynamical" dictionary similar to the analytic combinatorics dictionary. In particular, when $\mathcal{G}^\star \propto \mathcal{B}_{(d)}$, the generating operator of $\mathcal{G}^\star$ is the quasi-inverse

$$\mathbf{G}_{s,w,[c],\langle \mathcal{G}^\star\rangle} = \left(\mathbf{I} - \mathbf{G}_{s,w,[c],\langle \mathcal{G}\rangle}\right)^{-1} .$$

When $\mathcal{G}$ coincides with the set $\mathcal{H}_{(d-\ell)} \propto \mathcal{B}_{(d)}$ of inverse branches of the Brun dynamical system $\mathcal{D}_{(d-\ell)}$, then the generating operator $\mathbf{G}_{s,w,[c],\langle \mathcal{H}_{(d-\ell)}\rangle}$ is closely related to the (usual) weighted transfer operator $\mathbf{H}_{s,w,[c],(d-\ell)}$ of the Brun dynamical system $\mathcal{D}_{(d-\ell)}$ defined by

$$\mathbf{H}_{s,w,[c],(d-\ell)}[f](\boldsymbol{x}) := \sum_{h\in\mathcal{H}_{(d-\ell)}} J[h](\boldsymbol{x})^s \, w^{c(h)} f \circ h(\boldsymbol{x}) \, , \quad (18)$$

and, with the scale change of (17), we will obtain (19).

With Proposition 2, this yields the following characterization of the generating operator of the set $\mathcal{B}_{(d)}$ used by the $\mathtt{BrunGcd}(d)$ algorithm.

Proposition 3. *Let c be an additive cost on the set $\mathcal{B}_{(d)}$.*
(a) The generating operator $\mathbf{B}_{s,w,[c],(d)}$ of the set $\mathcal{B}_{(d)}$ used by the $\mathtt{BrunGcd}(d)$ algorithm decomposes as

$$\mathbf{B}_{s,w,[c],(d)} = \mathbf{P}_{s,w,[c],(1)} \circ \mathbf{P}_{s,w,[c],(2)} \circ \cdots \circ \mathbf{P}_{s,w,[c],(d)} \, ,$$

and involves the sequence of generating operators $\mathbf{P}_{s,w,[c],(d-\ell)}$ of the set $\mathcal{P}_{(d-\ell)}$ used by the $\mathtt{BrunGcd}_{(d,\ell)}$ algorithm. It then satisfies the recurrence relation

$$\mathbf{B}_{s,w,[c],(d)} = \mathbf{B}_{s,w,[c],(d-1)} \circ \mathbf{P}_{s,w,[c],(d)} .$$

(b) Each operator $\mathbf{P}_{s,w,[c],(d-\ell)}$ involves the quasi-inverse of the generating operator $\mathbf{G}_{s,w,[c],(d-\ell)}$ of the set $\mathcal{H}_{(d-\ell)}$ and the generating operator $\mathbf{F}_{s,w,[c],(d-\ell)}$ associated with the final set $\mathcal{F}_{(d-\ell)}$ of the ℓ-th phase, and is equal to

$$\mathbf{P}_{s,w,[c],(d-\ell)} = \mathbf{F}_{s,w,[c],(d-\ell)} \circ \left(I - \mathbf{G}_{s,w,[c],(d-\ell)}\right)^{-1} .$$

(c) The generating operator $\mathbf{G}_{s,w,[c],(d-\ell)}$ of the set $\mathcal{H}_{(d-\ell)}$ is closely related to the (weighted) transfer operator $\mathbf{H}_{s,w,[c],(d-\ell)}$ of the dynamical system $\mathcal{D}_{(d-\ell)}$ defined in (18) via the change of scale $s \mapsto s/(d-\ell+1)$, namely

$$\mathbf{G}_{s,w,[c],(d-\ell)} = \mathbf{H}_{s/(d-\ell+1),w,[c],(d-\ell)} \, . \quad (19)$$

Remark. It proves also useful to deal with the plain (unweighted) operator $\mathbf{G}_{s,\langle\mathcal{G}\rangle}$, and the cumulative operator $\widehat{\mathbf{G}}_{s,\langle\mathcal{G}\rangle}$ defined in an analogue way as for series as

$$\mathbf{G}_{s,\langle\mathcal{G}\rangle} = \sum_{h\in\mathcal{G}} \mathbf{G}_{s,\langle h\rangle} \qquad \widehat{\mathbf{G}}_{s,\langle\mathcal{G}\rangle} = \sum_{h\in\mathcal{G}} c(h)\mathbf{G}_{s,\langle h\rangle}, \quad (20)$$

$$\text{with } \mathbf{G}_{s,\langle h\rangle}[f](\boldsymbol{x}) := |D[h](\boldsymbol{x})|^{-s} \, f \circ h(\boldsymbol{x}) \, .$$

These two operators are related to the weighted operator relative to any cost c via the equality

$$\mathbf{G}_{s,\langle\mathcal{G}\rangle} = \mathbf{G}_{s,1,[c],\langle\mathcal{G}\rangle}, \qquad \widehat{\mathbf{G}}_{s,[c],\langle\mathcal{G}\rangle} = \Delta[w \mapsto \mathbf{G}_{s,w,[c],\langle\mathcal{G}\rangle}] \, .$$

4.4 Useful expressions for generating functions.

The following result is one of the key ingredients of the paper. It is typical in dynamical analysis as it relates generating functions and generating operators.

Theorem 2. *Consider the $\mathtt{BrunGcd}(d)$ algorithm acting on the set $\Omega_{(d)}$, together with an additive cost C, related to some elementary cost c. Then, the three following relations hold between*

(i) the three generating functions relative to cost C, defined in (10),(11),(14),

(ii) the three analogous generating operators of the set $\mathcal{B}_d$, i.e., the bivariate one defined in Proposition 3 and the other two defined in (20)

$$S_{(d,C)}(s,w) = \zeta(s) \cdot \mathbf{B}_{s,w,[c],(d)}[1](0) \, ,$$
$$S_{(d)}(s) = \zeta(s) \cdot \mathbf{B}_{s,1,[c],(d)}[1](0) \, ,$$
$$\widehat{S}_{(d,C)}(s) = \zeta(s) \cdot \widehat{\mathbf{B}}_{s,[c],(d)}[1](0) \, .$$

The proof is indeed quite short. Let $h \in \mathcal{B}_{(d)}$. The equalitiy

$$\pi(\boldsymbol{u}) = \left(\frac{u_1}{u_0}, \frac{u_2}{u_0}, \ldots, \frac{u_d}{u_0}\right) = h(0)$$

together with $\gcd(\boldsymbol{u}) = 1$ proves that the denominator of the LFT h satisfies $|D[h](0)| = u_0$. Moreover, as C is an additive cost associated with cost c, the equality $C(\boldsymbol{u}) = c(h)$ holds. Now, the bijection between $\underline{\Omega}_{(d)}$ and the set $\mathcal{B}_{(d)}$, together with Eq. (16), entail the relations

$$S_{(d,C)}(s,w) = \zeta(s) \cdot \underline{S}_{(d,C)}(s) = \zeta(s) \cdot \sum_{\boldsymbol{u}\in\underline{\Omega}_{(d)}} \frac{w^{C(\boldsymbol{u})}}{u_0^s}$$
$$= \zeta(s) \cdot \sum_{h\in\mathcal{B}_{(d)}} w^{c(h)} \, |D[h](0)|^{-s} = \zeta(s) \cdot \mathbf{B}_{s,w,[c],d}[1](0) \, .$$

4.5 Derivatives of quasi-inverses.

When the cost C is concentrated on the phase ℓ, the functional Δ defined in (15) is applied to the quasi-inverse $(I - \mathbf{G}_{s,w,[c],(k)})^{-1}$ with $k = d - \ell$. This produces two (plain) quasi-inverses and a middle operator (which depends on the cost c). Now, via (19), we return to the operator $\mathbf{H}$ defined in (18), with the scale-change $t = s/(k+1)$, and obtain

$$\Delta[w \mapsto (I - \mathbf{G}_{s,w,[c],(k)})^{-1}]$$
$$= (I - \mathbf{H}_{t,(k)})^{-1} \circ \ \mathbf{H}^{(c)}_{t,(k)} \circ \ (I - \mathbf{H}_{t,(k)})^{-1} \, ,$$

where the "middle" operator is a weighted operator

$$\mathbf{H}^{(c)}_{t,(k)}[f] := \sum_{h\in\mathcal{H}_{(k)}} c(h) \cdot J[h]^t \cdot f \circ h \, .$$

The (plain) quasi-inverses will play a central role in the analysis, whereas the middle operators[5] and the end-of-phase operators will only play a secondary role. We will now focus on the quasi-inverses of the transfer operators and denote

$$\mathbf{Q}_{(k)}(t) := (I - \mathbf{H}_{t,(k)})^{-1}. \quad (21)$$

The plain Dirichlet series involves one quasi-inverse for each phase, whereas the cumulative Dirichlet series relative to a cost C concentrated on the ℓ-th phase involves two quasi-inverses for the ℓ-th phase, and only one for each other phase. This is now precisely stated:

Proposition 4. *Associate with $\ell \in [0..d-1]$ the scale change $t_\ell : s \mapsto s/(d-\ell+1)$. Then the following holds:*

[5] except for $d = 1$

(a) *The plain Dirichlet series $S_{(d)}(s)$ involves a unique quasi-inverse $\mathbf{Q}_{(d-\ell)}(t_\ell)$ for each phase of index $\ell \in [0..d-1]$.*

(b) *If C is ℓ-concentrated, the cumulative series $\widehat{S}_{(d,C)}$ of cost C involves one quasi-inverse $\mathbf{Q}_{(d-k)}(t_{d-k})$ for each phase of index $k \neq \ell$ and a "double" quasi-inverse $\mathbf{Q}^2_{(d-\ell)}(t_{d-\ell})$ for the ℓ-th phase.*

This ends the first part of our analysis, of a combinatorial nature. We now begin the second part of our analysis, of an analytic nature.

5. DYNAMICAL ANALYSIS (II)

We now return to the proof of Theorem 1. With (12) and (13), we know that the mean values of interest are related to coefficients of Dirichlet series, plain or cumulative ones. With the Delange Tauberian theorem, stated in Section 5.3, we know how to relate the asymptotic behavior of coefficients with singularities of the Dirichlet series. We then have now to study the Dirichlet series, from an analytic point of view, and discover their singularities. With Proposition 4, their singularities are brought by quasi-inverses. This is why we begin by studying the quasi-inverse in Section 5.1, and then apply this study to the Dirichlet series in Section 5.2.

5.1 Properties of quasi-inverses.

Consider an index $k \in [1..d]$. The transfer operator $\mathbf{H}_{t,(k)}$ acts on the space $\mathcal{C}^1(\mathcal{J}_{(k)})$. It is *quasi-compact* and the dominant part of its spectrum is discrete. Furthermore, as the operator $\mathbf{H}_{t,(k)}$ is the density transformer of the system for $t = 1$, the value 1 belongs to the spectrum, and as the system is ergodic [12], this is its unique dominant eigenvalue. The dominant eigenfunction is explicitly known (see [1])

$$\psi_k(\boldsymbol{x}) = \sum_{\sigma \in \mathfrak{S}_k} \prod_{i=1}^{k} \frac{1}{1 + x_{\sigma(1)} + x_{\sigma(2)} + \ldots + x_{\sigma(i)}}; \qquad (22)$$

and involves the set $\mathfrak{S}_k$ of permutations on $[1..k]$. The quasi-compacity of the operator together with perturbation theory entails the existence of a spectral decomposition

$$\mathbf{H}_{t,(k)} = \lambda_{(k)}(t)\mathbf{A}_{t,(k)} + \mathbf{K}_{t,(k)}\,,$$

on a neighborhood of $t = 1$ that involves the dominant eigenvalue $\lambda_{(k)}(t)$, the projection $\mathbf{A}_{t,(k)}$ on the dominant eigenspace, and a "remainder" operator $\mathbf{K}_{s,(k)}$ whose spectral radius is strictly smaller than $|\lambda_{(k)}(t)|$. Then, the relation $\mathbf{A}_{t,(k)} \circ \mathbf{K}_{t,(k)} = \mathbf{K}_{t,(k)} \circ \mathbf{A}_{t,(k)} = 0$ leads to the spectral decomposition for the quasi-inverse

$$\mathbf{Q}_{(k)}(t) = \frac{\lambda_{(k)}(t)}{1 - \lambda_{(k)}(t)} \mathbf{A}_{t,(k)} + (\mathbf{I} - \mathbf{K}_{t,(k)})^{-1}\,.$$

Thus, the quasi-inverse has a pole at $t = 1$, with a residue which involves in particular the entropy $\mathcal{E}_k = -\lambda'_{(k)}(1)$. Furthermore, as the inverse branches are LFT's, there is an *aperiodicity* property that entails that $t \mapsto \mathbf{Q}_{(k)}(t)$ is analytic on the punctured vertical line $\{\Re t = 1,\ t \neq 1\}$.

Letting now $k := d - \ell$, and using for each phase of index ℓ the scale change $t_\ell = s/(d-\ell+1)$, we describe the behavior of the quasi inverse $s \mapsto \mathbf{Q}_{(d-\ell)}(t_\ell)$.

Proposition 5. *Consider for $\ell \in [0..d-1]$ the scale change $s \mapsto s/(d - \ell + 1)$. Then, the quasi-inverse $s \mapsto \mathbf{Q}_{(d-\ell)}(t_\ell)$ relative to the ℓ-th phase is analytic on the punctured half-plane $\{\Re s \geq d - \ell + 1, s \neq d - \ell + 1\}$ and admits a simple pole at $s = d - \ell + 1$, with a residue $\mathbf{T}_{(d-\ell)}$ defined as*

$$\mathbf{T}_{(d-\ell)}[f] = \frac{d - \ell + 1}{\mathcal{E}_{d-\ell}}\, I_{d-\ell}[f]\, \frac{\psi_{d-\ell}}{\kappa_{d-\ell}}\,, \qquad (23)$$

which involves the entropy $\mathcal{E}_k$, the integral I_k on the simplex $\mathcal{J}_{(k)}$ and the constant $\kappa_k := I[\psi_k]$.

5.2 Analytic properties of Dirichlet series.

We now return to the number of steps that the `BrunGcd` algorithm performs during its various phases. With Theorem 2, Propositions 4, 5, and Eq. (9), we describe the analytic properties of the Dirichlet series of interest.

Proposition 6. *Consider the* `BrunGcd` *algorithm when acting on $\Omega_{(d)}$, together with the total number L_d of steps, the number M_d of steps in the first phase, the number O_d of quotients equal to 1 during the first phase, the number Σ_d of subtractive steps during the first phase and the number R_d of steps in the following phases. Associate the six Dirichlet generating functions of interest, namely, the plain series and the five cumulative series relative with number of steps $L_d, M_d, O_d, \Sigma_d, R_d$. The following holds:*

(a) *The six series are analytic in the punctured half-plane $\{\Re s \geq d + 1, s \neq d + 1\}$.*

(b) *The plain series and the cumulative series relative to R_d admit a simple pole at $s = d + 1$. Their residues are denoted as T_d and $r_d T_d$, with $T_d = \mathbf{T}_{(d)}[1](0)$.*

(c) *The cumulative series relative to M_d, O_d, Σ_d and L_d admit a pole of order 2 at $s = d+1$, and their dominant constants* `Dom` *are expressed with the residue T_d and three other constants a_d, ρ_d, σ_d. One has:*

$$\texttt{Dom}[L_d] = \texttt{Dom}[M_d] = a_d\, T_d\,,$$
$$\texttt{Dom}[O_d] = a_d\, \rho_d\, T_d, \quad \texttt{Dom}[\Sigma_d] = a_d\, \sigma_d\, T_d\,.$$

The constants a_d, ρ_d and σ_d involve the entropy $\mathcal{E}_d$ and the constant $\kappa_d(y)$, defined in (24),

$$a_d = \frac{d+1}{\mathcal{E}_d}, \;\; \rho_d = 1 - \frac{\kappa_d(1/2)}{\kappa_d}, \;\; \sigma_d = \frac{1}{\kappa_d} \sum_{m \geq 1} \kappa_d\left(\frac{1}{m}\right)\,.$$

5.3 Extraction of coefficients.

It remains to relate the asymptotics of the coefficients of the Dirichlet series and their dominant singularities, and this is done via the Delange Tauberian Theorem.

Theorem 3 (Delange). *For $\sigma > 0$, consider a Dirichlet series $S(s)$ with non-negative coefficients which converges for $\Re s > \sigma$. Assume that the following holds:*
(i) $S(s)$ is analytic on $\Re s = \sigma, s \neq \sigma$;
(ii) near σ, $S(s)$ satisfies $S(s) = A(s)/(s - \sigma)^{\gamma+1}$ where A is analytic in σ, $A(\sigma) \neq 0$ and $\gamma \geq 0$.
Then as $N \to \infty$, the following asymptotics holds

$$\Phi_N[S] \sim \frac{A(\sigma)}{\sigma\Gamma(\gamma+1)} N^\sigma (\log N)^\gamma, \qquad \textit{for } \Phi_N \textit{ defined in } (13).$$

With Proposition 6, all the Dirichlet series of interest satisfy the hypotheses of the Delange Tauberian Theorem with $\sigma = d + 1$, the series of (b) with $\gamma = 0$ and the series of (c)

with $\gamma = 1$. With Eq. (12), the mean value of any cost C of interest equals a ratio whose numerator and denominator can be estimated with the Delange Tauberian Theorem. This concludes the proof of Theorem 1, and the main constants involved are constants $r_d, a_d, \rho_d, \sigma_d$. We do not know much about the asymptotics of r_d, but we now focus on the asymptotics of the other constants a_d, ρ_d, σ_d for $d \to \infty$.

5.4 The main constants of the analysis.

Except for $d = 1$ and $d = 2$, the values of the volume constant κ_d and the entropy $\mathcal{E}_d$ are not known, as well as their dependency with respect to dimension d. We deal with the simplex $y\mathcal{J}_{(d)}$, for $y \in [0, 1]$, and introduce the function

$$\kappa_d(y) := \int_{y\mathcal{J}_{(d)}} \psi_d(\boldsymbol{x})d\boldsymbol{x} = \int_{[0,y]^d} \varphi_d(\boldsymbol{x})d\boldsymbol{x} \qquad (24)$$

$$\text{with} \quad \varphi_d(\boldsymbol{x}) = \frac{1}{1+x_1} \cdots \frac{1}{1+x_1+x_2+\cdots+x_d}.$$

Note that $\kappa_d = \kappa_d(1)$. The ratio $\kappa_d(y)/\kappa_d$ is central, because it coincides with the limit probability that a quotient m of the Brun system in dimension d be larger than $1/y$.

The following result provides two new (and important) estimates: the first one contradicts a conjecture of Hardcastle and Khanin [7] which states that the entropy is asymptotically constant. The second one describes (for $d \to \infty$) the limit distribution of the quotients in the Brun system.

Proposition 7. *The constant $\kappa_d(y)$ and the entropy $\mathcal{E}_d$ are expressed as one-dimensional integrals:*

$$\kappa_d(y) = \frac{1}{d!}\int_0^\infty e^{-u}\beta(uy)^d du, \quad \beta(u) := \int_0^u \frac{1-e^{-v}}{v}\,dv,$$

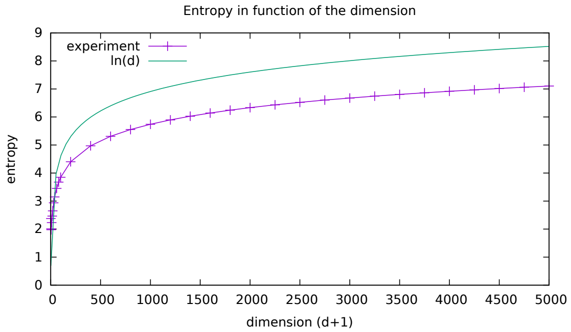
$$\mathcal{E}_d = (d+1)\int_0^1 \frac{1}{y}\frac{\kappa_d(y)}{\kappa_d}dy = \frac{d+1}{d!\,\kappa_d}\int_0^\infty \frac{e^{-u}}{u}\beta(u)^d du.$$

The following estimates hold for $d \to \infty$ and $y \in]0, 1]$ fixed:

$$\mathcal{E}_d \sim \log d, \qquad \frac{\kappa_d(y)}{\kappa_d} = y^{(1-\epsilon)d/\log d} \quad \textit{for } d > d_\epsilon\,. \qquad (25)$$

The complete proof will be given in the long version paper. We describe here the main ideas. The integral expression for $\kappa_d(y)$ is obtained via the Laplace transform. The expression of the entropy is obtained via the Rohlin formula. As each integral expression involves a large power of the function β, a method of Laplace type asymptotically compares such an integral to the maximum value of the integrand.

The following figure compares the theoretical curve $\mathcal{E}_d = \log d$ with experimental values of $\mathcal{E}_d$ which are obtained via the number of steps during the first phase.



6. CONCLUSION AND OPEN PROBLEMS

We have precisely used the Brun underlying dynamical system to describe the probabilistic behaviour of the `BrunGcd` algorithm. We have studied the asymptotics (for $d \to \infty$) of the main constants that intervene in the analysis and also confirmed with experiments the main results of the analysis. We conclude that the `BrunGcd` algorithm is not efficient. This is probably the case for all the gcd algorithms which are based on multidimensional continued fraction algorithms.

Nevertheless, it would be interesting to study other costs such as the bit-complexity as in [10] or perform a distributional analysis; in this case, we need to extend to higher dimensions the approach that was conducted in [2] for $d = 1$.

We also wish to use our good knowledge of the Brun rational trajectories obtained through dynamical analysis, in order to study simultaneous approximations associated with rational vectors, which corresponds to the real algorithmic situation.

Finally, it would be interesting to analyze the extended gcd algorithm based on the LLL algorithm and described in [11], even if its underlying system is quite complex to deal with.

7. REFERENCES

[1] P. Arnoux and A. Nogueira. Mesures de Gauss pour des algorithmes de fractions continues multidimensionnelles. *Ann. Sci. Ecole Norm. Sup.*, 26:645–664, 1993.

[2] V. Baladi and B. Vallée. Euclidean algorithms are Gaussian. *Journal Number Theory*, 110:331–386, 2006.

[3] V. Berthé, J. Bourdon, T. Jolivet, and A. Siegel. Generating discrete planes with substitutions. In *Combinatorics on words*, volume 8079 of *LNCS*, pages 58–70. Springer, 2013.

[4] V. Berthé, L. Lhote, and B. Vallée. Probabilistic analyses of the plain multiple gcd algorithm. *Journal of Symbolic Computation*, 74:425–474, 2016.

[5] A. Broise-Alamichel and Y. Guivarc'h. Exposants caractéristiques de l'algorithme de Jacobi-Perron et de la transformation associée. *Annales de l'Institut Fourier*, 51(3):565–686, 2001.

[6] V. Brun. Algorithmes euclidiens pour trois et quatre nombres. In *13e congrès des mathématiciens scandinaves, Helsinki 1957*, pages 45–64.

[7] D. M. Hardcastle and K. Khanin. On almost everywhere strong convergence of multi-dimensional continued fraction algorithms. *Ergodic Theory Dynam. Systems*, 20(6):1711–1733, 2000.

[8] D. E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, 3rd edition, 1998.

[9] J. Lagarias. The quality of the diophantine approximations found by the Jacobi-Perron algorithm, and related algorithms. *Mh. Math*, 115:299–328, 1993.

[10] L. Lhote and B. Vallée. Gaussian laws for the main parameters of the Euclid algorithms. *Algorithmica*, 50(4):497–554, 2008.

[11] B. S. Majewski and G. Havas. A solution to the extended GCD problem. In *ISSAC95*, pages 248–253.

[12] F. Schweiger. *Multidimensional continued fractions.* Oxford University Press, 2000.

[13] B. Vallée. Euclidean dynamics. *Discrete and Continuous Dynamical Systems*, 1(15):281–352, 2006.