Analysis of the Brun Gcd Algorithm

V. Berthé, L. Lhote, B. Vallée

DYNA3S, Novembre 2016

Brun gcd algorithm

- A multiple gcd algorithm that is a natural extension of the usual Euclid algorithm for $(d+1)$ integers.
- It coincides with it for two entries.
- It performs Euclidean divisions, between the largest entry and the second largest entry.
- This is the discrete version of a multidimensional continued fraction algorithm due to Brun ('57).

Also called Podsypanin modified Jacobi–Perron algorithm, d-dimensional Gauss transformation, ordered Jacobi–Perron algorithm, etc.

and also an algorithm for efficient exponentiation with precomputation [de Rooij]

Outline

- We perform the worst-case and the average-case analysis of this algorithm for the number of steps.
- We prove that the worst-case and the mean number of steps are 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 average-case analysis is related to the entropy of the system.
- We provide asymptotic estimates for the Brun entropy.
- We also compare this algorithm to Knuth's extension of the Euclid algorithm.

Euclid algorithm and continued fractions

- We start with two (coprime) integers
- One divides the largest by the smallest
- Euclid's algorithm yields the digits of the continued fraction expansion of their quotient
- Euclid's algorithm becomes in its continuous version the Gauss transformation

$$T: [0,1] \to [0,1], \ x \mapsto \{1/x\}$$

- Rational trajectories behave like generic trajectories for the Gauss transformation (methods from Dynamical Analysis [Baladi-Vallée])
- Our strategy: consider the generalizations of Euclid's algorithm issued from multidimensional continued fraction algorithms endowed with a "good" dynamical system (Brun, Jacobi-Perron, Selmer etc.)

Brun algorithm

We divide the largest entry by the second largest entry and reorder.

$$(74, 37, 13, 5, 3) \mapsto (37, 13, 5, 3) \mapsto (13, 11, 5, 3) \mapsto (11, 5, 3, 2) \mapsto$$

$$(5, 3, 2, 1) \mapsto (3, 2, 1) \mapsto (2, 1) \mapsto (1)$$

Brun algorithm

We divide the largest entry by the second largest entry and reorder.

$$(74, 37, 13, 5, 3) \mapsto (37, 13, 5, 3) \mapsto (13, 11, 5, 3) \mapsto (11, 5, 3, 2) \mapsto$$

$$(5, 3, 2, 1) \mapsto (3, 2, 1) \mapsto (2, 1) \mapsto (1)$$

Start with $(u_0, u_1, \ldots, u_d)$ with $u_0 > u_1 > u_2 > \ldots > u_d > 0$

- In each step, the first component u_0 is divided by the second component u_1, and creates a remainder v_0

$$v_0 := u_0 - mu_1 \quad \text{Remainder} \qquad m := \left[\frac{u_0}{u_1}\right] \quad \text{Partial quotient}$$

- The second component u_1 becomes the largest one.
- There are different cases for the insertion (or not) of v_0.

The algorithm $\texttt{BrunGcd}(d)$

$$u_0 > u_1 > u_2 > \ldots > u_d > 0$$

We divide the largest entry u_0 by the second largest entry u_1 and we reorder.

$$v_0 := u_0 - \left[\frac{u_0}{u_1}\right] u_1$$

- (G) (Generic case) if v_0 is not present in $(u_1, \cdots, u_d)$, we perform 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 (at position i, say), we do not insert v_0.

Phases of the algorithm

$$\Omega_{(k)} = \{\mathbf{u} = (u_0, u_1, \ldots, u_k) \mid u_0 > u_1 > u_2 > \ldots > u_k > 0\}.$$

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

- The algorithm $\texttt{BrunGcd}(d)$ decomposes into d phases, labelled from $\ell = 0$ to $\ell = d-1$. During each phase, a component is "lost", and the ℓ-th phase transforms an element of $\Omega_{(d-\ell)}$ into an element of $\Omega_{(d-\ell-1)}$.
- The phase ends as soon as it looses a component:
  - if $v_0 = 0$;
  - or else, if $v_0 \neq 0$ is already present in $(u_1, \cdots, u_k)$.
- The algorithm stops at the end of the $(d-1)$-th phase with an element of $\Omega_{(0)}$ which equals the gcd.

The algorithm $\texttt{BrunGcd}(d)$

We divide the largest entry by the second largest entry and reorder.

The algorithm $\texttt{BrunGcd}(d)$ computes the gcd of $(d+1)$ positive integers. It deals with the input set

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

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

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

Results

Maximum number of steps

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.

Maximum number of steps

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.

Theorem [Lam-Shallit-Vanstone] The maximum number $Q_{(d,N)}$ of steps of the BrunGcd Algorithm on the set

$$\Omega_{(d,N)} := \{\mathbf{u} = (u_0, u_1, \ldots, u_d) \mid N \geq u_0 > u_1 > u_2 > \ldots > u_d > 0\}$$

satisfies

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

Let $\tau_d \in]0, 1[$ be the smallest real root of $X^{d+1} + X - 1$

$$1/|\log \tau_d| \sim \frac{(d+1)}{\log d} \qquad (d \to \infty)$$

Mean number of steps

The algorithm `BrunGcd` acts on the set

$$\Omega_{(d,N)} = \{(u_0, u_1, \ldots, u_d) \mid N \geq u_0 > u_1 > u_2 > \ldots > u_d > 0\}$$

endowed with the uniform distribution

- The total number of steps L_d is on average linear in the size $\log N$ of the entries

Mean number of steps

The algorithm `BrunGcd` acts on the set

$$\Omega_{(d,N)} = \{(u_0, u_1, \ldots, u_d) \mid N \geq u_0 > u_1 > u_2 > \ldots > u_d > 0\}$$

endowed with the uniform distribution

- The total number of steps L_d is on average linear in the size $\log N$ of the entries

  Theorem Here d is fixed, N tends to ∞. One has

$$\mathbb{E}_N[L_d] \sim \frac{d+1}{\mathcal{E}_d} \cdot \log N \qquad (N \to \infty)$$

$\mathcal{E}_d$: entropy of the Brun dynamical system

Mean number of steps

The algorithm `BrunGcd` acts on the set

$$\Omega_{(d,N)} = \{(u_0, u_1, \ldots, u_d) \mid N \geq u_0 > u_1 > u_2 > \ldots > u_d > 0\}$$

endowed with the uniform distribution

- The total number of steps L_d is on average linear in the size $\log N$ of the entries
- Number of steps performed during the first phase: M_d

  Theorem $\mathbb{E}_N[L_d] \sim \mathbb{E}_N[M_d] \sim \frac{d+1}{\mathcal{E}_d} \cdot \log N \qquad (N \to \infty)$

- Number of steps performed after the first phase: R_d

  Theorem $\mathbb{E}_N[R_d] \sim r_d \qquad (N \to \infty)$

- One has a strong difference between the first phase, where most of the work is done, and the remainder of the execution, where R_d is on average asymptotically constant

Comparison between the worst and the average case

- Both dominant constants behave as $d/\log d$ for $d \to \infty$

$$\mathbb{E}_N[L_d] \sim \frac{d+1}{\mathcal{E}_d} \cdot \log N \qquad Q_{(d,N)} \sim \frac{1}{|\log \tau_d|} \cdot \log N \qquad (N \to \infty)$$

$$1/|\log \tau_d| \sim \frac{(d+1)}{\log d} \qquad \mathcal{E}_d \sim \log d \qquad (d \to \infty)$$

- 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.

On the quotients equal to 1

- Number of steps performed during the first phase: M_d
- Number of quotients equal to 1 during the first phase: O_d

Theorem

$$\frac{\mathbb{E}_N[O_d]}{\mathbb{E}_N[M_d]} \sim \rho_d \qquad (N \to \infty)$$

$$\rho_d = 1 + O(2^{-d/\log d}) \qquad (d \to \infty)$$

- Number of steps of the subtractive version of `BrunGcd` during the first phase: Σ_d

Theorem

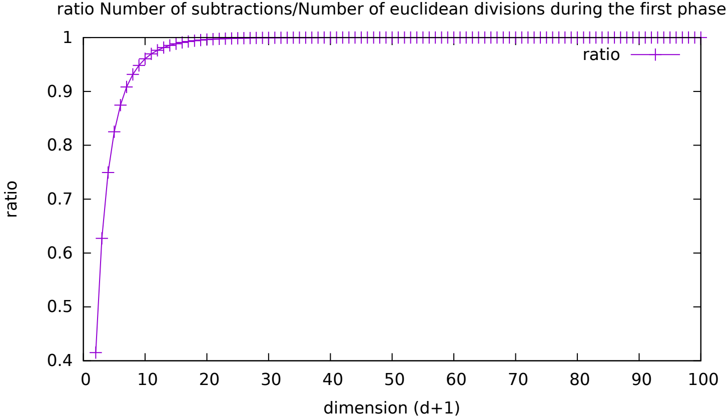
$$\frac{\mathbb{E}_N[\Sigma_d]}{\mathbb{E}_N[M_d]} \sim \sigma_d \qquad (d \to \infty)$$

$$1 \le \sigma_d \le 2 + (\log d)^{-1/2}$$

On the proportion of quotients equal to 1

The following figure 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
- for $d = 50$, the proportion is 99.99%.



On the constants

The constants $\mathcal{E}_d, \rho_d, \sigma_d, r_d$ are dynamical constants

They are defined via the dynamical system underlying the `BrunGcd` algorithm.

It is defined on the simplex

$$\mathcal{J}_{(d)} = \{\mathbf{x} = (x_1, \ldots, x_d \mid 1 \geq x_1 \geq \ldots \geq x_d \geq 0\}$$

and admits an invariant measure defined on $\mathcal{J}_{(d)}$

$$\Psi_d(x) = \sum_{\sigma \in \mathfrak{S}_d} \prod_{i=1}^{d} \frac{1}{1 + x_{\sigma(1)} + x_{\sigma(2)} + \ldots + x_{\sigma(i)}}$$

Consider the measure ν_d associated with Ψ_d, and the function

$$\mu_d : [0,1] \to [0,1], \ y \mapsto \nu_d(y\mathcal{J}_{(d)})$$

$$\mathcal{E}_d = (d+1) \int_0^1 \mu_d(y) \frac{dy}{y}, \qquad \rho_d = 1 - \mu_d\left(\frac{1}{2}\right), \qquad \sigma_d = \sum_{m \geq 1} \mu_d\left(\frac{1}{m}\right)$$

On the number of steps

Gauss map and continued fractions

$T_G \colon [0,1] \to [0,1],\ x \mapsto \{1/x\},\ \text{if } x \neq 0,\ \text{and } T_G(0) = 0$

$$x = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}} \qquad a_n = \left[\frac{1}{T^{n-1}(x)}\right],\ n \geq 1$$

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = x \begin{bmatrix} 0 & 1 \\ 1 & \left[\frac{1}{x}\right] \end{bmatrix} \begin{bmatrix} T(x) \\ 1 \end{bmatrix} = \theta(x) \begin{bmatrix} 0 & 1 \\ 1 & a_1(x) \end{bmatrix} \begin{bmatrix} T(x) \\ 1 \end{bmatrix}$$

$$A_n(x) = A(x)A(T(x))\ldots A(T^{n-1}(x)) \quad \theta_n(x) = \theta(x)\ldots\theta(T^{n-1}(x))$$

$$A_n(x) = \begin{bmatrix} p_{n-1} & p_n \\ q_{n-1} & q_n \end{bmatrix} \theta_n(x) = |q_n x - p_n| \begin{bmatrix} x \\ 1 \end{bmatrix} = \theta_n(x) A_n(x) \begin{bmatrix} T^n(x) \\ 1 \end{bmatrix}$$

Gauss map and continued fractions

$T_G\colon [0,1] \to [0,1],\ x \mapsto \{1/x\},$ if $x \neq 0,$ and $T_G(0) = 0$

$$x = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cdots}} \qquad a_n = \left[\frac{1}{T^{n-1}(x)}\right],\ n \geq 1$$

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = x \begin{bmatrix} 0 & 1 \\ 1 & \left[\frac{1}{x}\right] \end{bmatrix} \begin{bmatrix} T(x) \\ 1 \end{bmatrix} = \theta(x) \begin{bmatrix} 0 & 1 \\ 1 & a_1(x) \end{bmatrix} \begin{bmatrix} T(x) \\ 1 \end{bmatrix}$$

$$A_n(x) = A(x)A(T(x))\ldots A(T^{n-1}(x)) \quad \theta_n(x) = \theta(x)\ldots\theta(T^{n-1}(x))$$

$$A_n(x) = \begin{bmatrix} p_{n-1} & p_n \\ q_{n-1} & q_n \end{bmatrix} \theta_n(x) = |q_n x - p_n| \begin{bmatrix} x \\ 1 \end{bmatrix} = \theta_n(x)A_n(x) \begin{bmatrix} T^n(x) \\ 1 \end{bmatrix}$$

Thm For a.e. x, $\lim \frac{1}{n} \log q_n = \frac{\pi^2}{12 \log 2} = 1.18 \cdots = \lambda_1$ first Lyapunov exponent

first Lyapunov exponent $=$ "log largest eigenvalue" $\rightsquigarrow$ size of the matrices/convergents

$A_n(x) \sim q_n(x) \sim e^{\lambda_1 n} \rightsquigarrow$ Number of steps $=$ size/ log eigenvalue$=$ $\log N/\lambda_1$

Lyapunov exponents and continued fractions

Let $X \subset [0,1]^{d-1}$

A d-dimensional continued fraction map over X is given by measurable maps

$$T\colon X \to X,\ A\colon X \to GL(d,\mathbb{Z}),\ \theta\colon X \to \mathbb{R}_+$$

that satisfy the following: for a.e. $x \in X$, one has

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = \theta(x)A(x)\begin{bmatrix} T(x) \\ 1 \end{bmatrix}$$

Let

$$A_n(x) = A(x)A(T(x))\dots A(T^{n-1}(x)),\ \theta_n(x) = \theta(x)\dots\theta(T^{n-1}(x))$$

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = \theta_n(x)A_n(x)\begin{bmatrix} T^n(x) \\ 1 \end{bmatrix}$$

First Lyapunov exponent λ_1 = log eigenvalue $\rightsquigarrow$ size of the matrices $A_n(x) = e^{\lambda_1 n} \rightsquigarrow$ Number of steps $= \log N/\lambda_1$

Number of steps $\ell(u, v)$

$\ell(u, v)$: number of steps in Euclid algorithm $0 < v < u$

- Worst case

$$\ell(u, v) = O(\log v) \qquad (\leq 5 \log_{10} v, \text{ Lamé } 1844)$$

Reynaud 1821 [$\ell(u, v) < v/2$], see Shallit's survey

Number of steps $\ell(u, v)$

$\ell(u, v)$: number of steps in Euclid algorithm $0 < v < u$

- Worst case

$$\ell(u, v) = O(\log v) \qquad (\leq 5 \log_{10} v, \text{ Lamé } 1844)$$

- Mean case $\qquad 0 < v < u \leq N \qquad \gcd(u, v) = 1$

$$\mathbb{E}_N(\ell) \sim \frac{12 \log 2}{\pi^2} \cdot \log N$$

[see Knuth, Vol. 2]

Number of steps $\ell(u, v)$

$\ell(u, v)$: number of steps in Euclid algorithm $0 < v < u$

- Worst case

$$\ell(u, v) = O(\log v) \qquad (\leq 5 \log_{10} v, \text{ Lamé } 1844)$$

- Mean case $0 < v < u \leq N \qquad \gcd(u, v) = 1$

$$\frac{12 \log 2}{\pi^2} \cdot \log N + \eta + O(N^{-\gamma})$$

η Porter's constant

asymptotically normal distribution

[Heilbronn'69,Dixon'70,Porter'75,Hensley'94,Baladi-Vallée'05...]

Distributional dynamical analysis

$\gcd(u_0, u_1) = 1 \quad N \geq u_0 > u_1 > \cdots \quad u_{k-1} = a_k u_k + u_{k+1}$

Cost of moderate growth $c(a) = O(\log a)$

- Number of steps in Euclid algorithm $c \equiv 1$
- Number of occurrences of a quotient $c = \mathbf{1}_a$
- Binary length of a quotient $c(a) = \log_2(a)$

Distributional dynamical analysis

$\gcd(u_0, u_1) = 1 \quad N \geq u_0 > u_1 > \cdots \quad u_{k-1} = a_k u_k + u_{k+1}$

Cost of moderate growth $c(a) = O(\log a)$

- Number of steps in Euclid algorithm $c \equiv 1$
- Number of occurrences of a quotient $c = \mathbf{1}_a$
- Binary length of a quotient $c(a) = \log_2(a)$

Theorem [Baladi-Vallée'05]

$$\mathbb{E}_N[\text{Cost}] = \frac{12 \log 2}{\pi^2} \cdot \widehat{\mu}(\text{Cost}) \cdot \log N + O(1)$$

The distribution is asymptotically Gaussian (CLT)

Discrete framework-Euclid algorithm

Ergodic theorem

We are given a dynamical system $(X, T, \mathcal{B}, \mu)$

$$T: X \to X$$

- Average time values: one particle over the long term Ergodic theory
- Average space values: all particles at a particular instant, average over all possible sets Dynamical analysis of algorithms

$$\mu(B) = \mu(T^{-1}B) \quad T\text{-invariance}$$

$$T^{-1}B = B \implies \mu(B) = 0 \text{ or } 1 \quad \text{ergodicity}$$

Ergodic theorem space mean= average mean

$$\frac{1}{N} \sum_{0 \le n \le N} f(T^n)x = \int f d\mu \quad \text{a.e. } x$$

Ergodic theorem

Theorem [Baladi-Vallée'05]

$$\mathbb{E}_N[\mathsf{Cost}] = \frac{12\log 2}{\pi^2}\cdot\widehat{\mu}(\mathsf{Cost})\cdot\log N + O(1)$$

Ergodic theorem

Theorem [Baladi-Vallée'05]

$$\mathbb{E}_N[\mathrm{Cost}] = \frac{12\log 2}{\pi^2} \cdot \widehat{\mu}(\mathrm{Cost}) \cdot \log N + O(1)$$

$$\mathbb{E}_N[c] = \frac{\text{dimension}}{\text{entropy}} \cdot \widehat{\mu}(c) \cdot \log N + O(1)$$

$$\widehat{\mu}(c) = \int_0^1 c([1/x]) \cdot \frac{1}{\log 2} \frac{1}{1+x} dx$$

Continuous framework-truncated trajectories

Cost of truncated trajectories

Cost of moderate growth

$c(a_i) = O(\log a_i)$ for a_i partial quotient

$$x = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

Cost of truncated trajectories

Cost of moderate growth

$$c(a_i) = O(\log a_i) \text{ for } a_i \text{ partial quotient}$$

Cost of a truncated trajectory

$$C_n(x) = \sum_{i=1}^{n} c(a_i(x)) \qquad a_i = \left[\frac{1}{T^{i-1}(x)}\right]$$

According to the ergodic theorem, for a.e. $x \in [0,1]$

$$C_n(x)/n \to \widehat{\mu}(x)$$

$$\widehat{\mu}(C) = \int_0^1 c\left([1/x]\right) \cdot \frac{1}{\log 2} \frac{1}{1+x} \cdot dx$$

$$\mathbb{E}_N[C] = \frac{2}{\pi^2/(6\log 2)} \cdot \widehat{\mu}(C) \cdot \log N$$

Multidimensional Euclid's algorithms and continued fractions

- Jacobi-Perron We subtract the first one to the two other ones with $u_0 \geq u_1, u_2 \geq 0$

$$(u_0, u_1, u_2) \mapsto \left(u_2, u_0 - \left[\frac{u_0}{u_2}\right] u_2, u_1 - \left[\frac{u_1}{u_2}\right] u_2\right)$$

- Brun We subtract the second largest entry and we reorder. If $u_0 \geq u_1 \geq u_2 \geq 0$

$$(u_0, u_1, u_2) \mapsto (u_0 - u_1, u_1, u_2)$$

- Poincaré We subtract the previous entry and we reorder

$$(u_0, u_1, u_2) \mapsto (u_0 - u_1, u_1 - u_2, u_2)$$

- Selmer We subtract the smallest to the largest and we reorder

$$(u_0, u_1, u_2) \mapsto (u_0 - u_2, u_1, u_2)$$

- Fully subtractive We subtract the smallest one to the other ones and we reorder

$$(u_0, u_1, u_2) \mapsto (u_0 - u_2, u_1 - u_2, u_2)$$

Number of steps for the Euclid algorithm

Consider

$$\Omega_m := \{(u_1, u_2) \in \mathbb{N}^2,\ 0 \le u_1, u_2 \le m\}$$

endowed with the uniform distribution

- Theorem The mean value $\mathbb{E}_m[L]$ of the number of steps satisfies

$$\mathbb{E}_m[L] \sim \frac{2}{\pi^2/(6\log 2)} \log m = \frac{1}{\lambda_1} \log m$$

λ_1 is the first Lyapunov exponent of the Gauss map

$\pi^2/(6\log 2)$ is the entropy

[Heilbronn'69,Dixon'70,Hensley'94,Baladi-Vallée'03...]

Number of steps for a generalized Euclid algorithm

Consider parameters $(u_1, \cdots, u_d)$ with $0 \leq u_1, \cdots, u_d \leq m$

To be expected

$$\mathbb{E}_m[L] \sim \frac{\text{dimension}}{\text{Entropy}} \times \log m = \frac{1}{\text{first Lyapounov exponent}} \times \log m$$

The first Lyapounov exponent governs the growth of the denominators of the convergents q_n

Comparison of gcd algorithms

We consider three Euclid algorithms for polynomials in $\mathbb{F}_q[X]$

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

- One chooses one specific component. This is
  - the first component for the Jacobi-Perron algorithm
  - the second largest component for the Brun algorithm
  - and the smallest component for the Fully Subtractive algorithm
- Each algorithm divides the other two components by this specific component, and replaces these components by their remainders in the division by the specific component.
- After having performed these divisions, this specific component becomes the largest one, and it is thus placed at the third position.

The algorithm stops when there remains only one non-zero component. This is the gcd.

Costs

Theorem [B.-Nakada-Natsui-Vallée]

$$\Omega_m := \{R = (R_1, R_2, R_3) \mid m = \deg R_3 > \max(\deg R_1, \deg R_2)\}$$

- Number of steps

$$\frac{3}{\text{Entropy}} \cdot m$$

- Bit-complexity

Quadratic m^2 Brun $<$ Jacobi-Perron$<$ Fully Subtractive

- Fine bit-complexity (non-zero terms)
  We find the same value for the three algorithms!

$$\frac{3(q-1)}{2q} \cdot m^2$$

On Knuth gcd algorithm

Knuth gcd algorithm

Consider the input $(u_0, u_1, ..., u_d)$

- $v_0 := u_0$
- For $k \in [1..d]$, one successively computes

$$v_k := \gcd(u_k, v_{k-1}) = \gcd(u_0, u_1, \ldots, u_k)$$

The total gcd $v_d := \gcd(u_0, u_1, \ldots, u_d)$ is obtained after d phases

One performs a sequence of d gcd computations
on two entries

Knuth gcd algorithm

Consider the input $(u_0, u_1, ..., u_d)$

- $v_0 := u_0$
- For $k \in [1..d]$, one successively computes

$$v_k := \gcd(u_k, v_{k-1}) = \gcd(u_0, u_1, \ldots, u_k)$$

The total gcd $v_d := \gcd(u_0, u_1, \ldots, u_d)$ is obtained after d phases

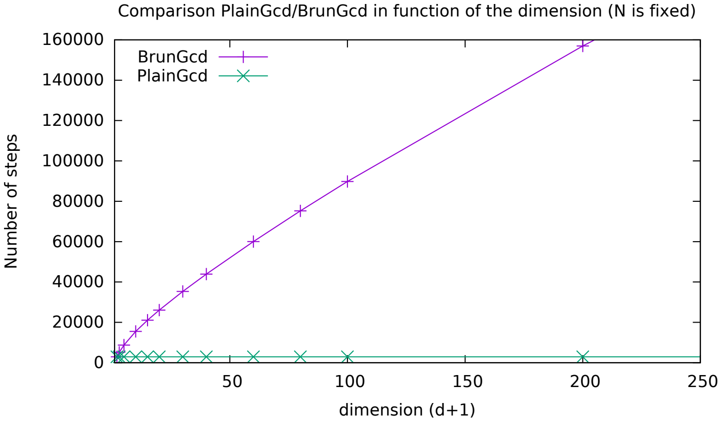
One performs a sequence of d gcd computations
on two entries

The same formal scheme can be applied to

- positive integers
- polynomials with coefficients in $\mathbb{F}_q$

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.
- The complexity of the `PlainGcd` algorithm appears to be independent of d.



Number of steps for Knuth gcd algorithm

A different notion of size

$$\Omega'_{(d,N)} := \{(u_0, \cdots, u_d) \mid u_0 u_1 \dots u_d \leq N\}$$

The expectation of the number of steps L_d during the first phase is linear with respect to the size N and satisfies

$$\mathbb{E}_N[L_d] \sim \frac{6 \log 2}{\pi^2} \cdot \frac{\log N}{(d+1)}$$

First phase linear on average
For the other phases $k \geq 2$ constant in average
Almost all the calculation is done during the first phase
Analogous results for formal power series with coefficients in a finite field
Average-case and distributional analysis

[B.-Creusefond-Lhote-Vallée], ISSAC 13

Comparison of gcd algorithms

- Brun algorithm for $d+1$ integers

$$\text{Number of steps} \quad \mathbb{E}_N[L] \sim \frac{d+1}{\mathcal{E}_d^B} \cdot \log N$$

$$\text{Entropy} \quad \mathcal{E}_d^B \sim \log d$$

- Knuth algorithm

$$\text{Number of steps} \quad \mathbb{E}_N[L] \sim \frac{1}{\mathcal{E}_2^K} \cdot \frac{\log N}{(d+1)}$$

$$\text{Entropy } \mathcal{E}_2^K = \pi^2/(6\log 2)$$

☹ For Brun algorithm, $\log N$ is the size of the maximal input, whereas for Knuth algorithm, $\log N$ is the cumulative size

Method

Method

- A bijection between the set of entries and the sets of quotients together with possible insertion places and gcd's.

$$\text{Inputs} \sim \text{quotients} \times \text{possible insertion places} \times \text{gcd}$$

- Expression of associated Dirichlet series in terms of transfer operators of the dynamical system which highlight the singularities
- This proves in particular that the first phase dominates (dominant singularity)
- We use a Delange type theorem

Brun dynamical system

A continuous extension of the algorithm that provides an exact characterization of the trajectories that are related to the execution of the algorithm. It acts on the simplex $\mathcal{J}_{(d)} \subset \mathbb{R}^d$

$$\mathcal{J}_{(d)} := \{\mathbf{x} = (x_1, \ldots, x_d) \mid 1 \geq x_1 \geq \ldots \geq x_d \geq 0\}$$

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

The algorithm $\mathtt{BrunSD}(d)$ The map $\mathtt{Ins}(y_0, \mathbf{y})$ is the insertion "in front of", with two cases:

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

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

We use here the existence of an ergodic absolutely continuous invariant measure, and contraction properties of Brun Dynamical system [Broise]

Transfer operators and Gauss map $T\colon x \mapsto \{1/x\}$

Perron-Frobenius operator Think of f as a density function

$$P[f](x) = \sum_{y\,:\,T(y)=x} \frac{1}{|T'(y)|} f(y) = \sum_{k\geq 1} \left(\frac{1}{k+x}\right)^2 f\left(\frac{1}{k+x}\right)$$

Let $\mathcal{H}$ stand for the set of inverse branches of the Gauss map

$$P[f](x) = \sum_{h\in\mathcal{H}} h'(x) \cdot f \circ h(x)$$

Transfer operators and Gauss map $T\colon x \mapsto \{1/x\}$

Perron-Frobenius operator Think of f as a density function

$$P[f](x) = \sum_{y\,:\,T(y)=x} \frac{1}{|T'(y)|} f(y) = \sum_{k\geq 1} \left(\frac{1}{k+x}\right)^2 f\left(\frac{1}{k+x}\right)$$

Let $\mathcal{H}$ stand for the set of inverse branches of the Gauss map

$$P[f](x) = \sum_{h\in\mathcal{H}} h'(x) \cdot f \circ h(x)$$

Ruelle operator

$$P_s[f](x) = \sum_{h\in\mathcal{H}} h'(x)^s \cdot f \circ h(x) \qquad s \in \mathbb{C}$$

Dirichlet series

Take $x = 0,\ f = 1 \ \rightsquigarrow_{\mathcal{H}^*} \ (\mathrm{Id} - P_s)^{-1} \ \rightsquigarrow \ \sum_{\ell\geq 1} 1/\ell^{2s}$ ☺

Transfer operators and Gauss map $T\colon x \mapsto \{1/x\}$

Perron-Frobenius operator Think of f as a density function

$$P[f](x) = \sum_{y\,:\,T(y)=x} \frac{1}{|T'(y)|} f(y) = \sum_{k\geq 1} \left(\frac{1}{k+x}\right)^2 f\left(\frac{1}{k+x}\right)$$

Let $\mathcal{H}$ stand for the set of inverse branches of the Gauss map

$$P[f](x) = \sum_{h\in\mathcal{H}} h'(x) \cdot f\circ h(x)$$

Ruelle operator

$$P_s[f](x) = \sum_{h\in\mathcal{H}} h'(x)^s \cdot f\circ h(x) \qquad s\in\mathbb{C}$$

Dirichlet series

Take $x=0$, $f=1$ $\rightsquigarrow_{\mathcal{H}^*}$ $(\mathrm{Id}-P_s)^{-1}$ $\rightsquigarrow$ $\sum_{\ell\geq 1} 1/\ell^{2s}$ ☺

Involving additive costs

$$P_{s,w}[f](x) = \sum_{h\in\mathcal{H}} h'(x)^s \cdot w^{c(h)} \cdot f\circ h(x)$$

Transfer operators and Brun algorithm

Each step of the algorithm is a linear fractional transformation

Let h be an inverse branch and $J[h]$ its Jacobian

$$P_s[f](x) = \sum_{h \in \mathcal{H}} J[h](x)^s \cdot f \circ h(x)$$

$$T(\mathbf{x}) = \mathtt{Ins}\left(\left\{\frac{1}{x_1}\right\}, \left(\frac{x_1}{x_2}, \ldots, \frac{x_d}{x_1}\right)\right)$$

$$m(x) = \left[\frac{1}{x_1}\right], \qquad j(x) = \mathtt{Pos}\left(\left\{\frac{1}{x_1}\right\}, \left(\frac{x_2}{x_1}, \ldots, \frac{x_d}{x_1}\right)\right)$$

Inverse branch

$$h_{(m,j)}(y_1, y_2, \ldots, y_d) = \left(\tfrac{1}{m+y_j}, \tfrac{y_1}{m+y_j}, \ldots, \tfrac{y_{j-1}}{m+y_j}, \tfrac{y_{j+1}}{m+y_j}, \ldots, \tfrac{y_d}{m+y_j}\right)$$

Jacobian $\quad J[h_{(m,j)}](y) = \frac{1}{(m+y_j)^{d+1}} \quad \rightsquigarrow_{\mathcal{H}^*} \quad J[h](0) = \frac{1}{u_0^{d+1}}$ ☺

Generating functions and transfer operators

$$\mathbf{u} = (u_0, u_1, \cdots, u_d), \quad u_0 > u_1 > \cdots > u_d > 0, \quad ||\mathbf{u}|| := u_0$$

Dirichlet series

$$\sum_{\mathbf{u}} \frac{C(\mathbf{u})}{||\mathbf{u}||^s} = \sum_{n \geq 1} n^{-s} \sum_{||\mathbf{u}||=n} C(\mathbf{u})$$

We then introduce a further indeterminate w

$$\sum_{\mathbf{u}} \frac{w^{C(\mathbf{u})}}{||\mathbf{u}||^s}$$

The derivative w.r.t. w at $w = 1$ yields cumulative generating functions

Generating functions and transfer operators

Generating function $\sum_{\mathbf{u}} \frac{w^{C(\mathbf{u})}}{||\mathbf{u}||^s}$

Operator $P_{s,w}[f](x) = \sum_{h \in \mathcal{H}} J[h](x)^s \cdot w^{c(h)} \cdot f \circ h(x)$

Jacobian $J[h](0) = \frac{1}{||\mathbf{u}||^{d+1}}$

For the number of steps C, take $x = 0$, $f = 1$, $c = 1$, and $\frac{\partial}{\partial w}|_{w=1}$

$$\sum_{\mathbf{u}} \frac{C(\mathbf{u})}{||\mathbf{u}||^s} \rightsquigarrow_{h \in \mathcal{H}^*} (\mathrm{Id} - P_{s,w})^{-1}[1](0) \rightsquigarrow_{\text{Perron-Frobenius}} \frac{1}{1 - \lambda_s}$$

Singularity for s such that $\lambda_s = 1$ with λ_s dominant eigenvalue of the operator P_s (cf. invariant measure)

Branches and inverse branches

For any $\mathbf{x} \in \mathcal{J}_{(d)}$, the map $T_{(d)}$ uses a digit

$$(m, j) \in \mathcal{A}_{(d)} := \mathbb{N}^* \times [1..d]$$

with a quotient $m(\mathbf{x}) \geq 1$ and an insertion index $j(\mathbf{x}) \in [1..d]$.
Let $\mathcal{K}_{(d,m,j)} := \{\mathbf{x} \in \mathcal{J}_{(d)} \mid m(\mathbf{x}) = m, \quad j(\mathbf{x}) = j\}$
When (m, j) varies in $\mathcal{A}_{(d)}$
– the subsets $\mathcal{K}_{(d,m,j)}$ form a topological partition of $\mathcal{J}_{(d)}$
– 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$\mathcal{J}_{(d)}$

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

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

$$h_{(d,m,j)}(y_1, \ldots, y_d) = \left(\frac{1}{m+y_j}, \frac{y_1}{m+y_j}, \ldots, \frac{y_{j-1}}{m+y_j}, \frac{y_{j+1}}{m+y_j}, \ldots, \frac{y_d}{m+y_j}\right)$$

The Brun Perron–Frobenius operator

$$\mathbf{H}_{(d)}[f](\mathbf{x}) = \sum_{h \in \mathcal{H}_{(d)}} |J[h](\mathbf{x})| \, f \circ h(\mathbf{x})$$

A convenient functional space is $C^1(\mathcal{J}_{(d)}), || \cdot ||_1)$

$$||f||_1 = \sup_{\mathbf{x} \in \mathcal{J}_{(d)}} |f(\mathbf{x})| + \sup_{\mathbf{x} \in \mathcal{J}_{(d)}} ||\mathbf{D} f(\mathbf{x})||$$

$\mathbf{D}f(\mathbf{x})$= the differential of f at $\mathbf{x}$ and $|| \cdot ||$= a norm on $\mathbb{R}^d$

$\mathbf{H}_{(d)}$ acts on $\left(C^1(\mathcal{J}_{(d)}), \| \cdot \|_1\right)$ and is quasi-compact: the "upper" part of its spectrum is formed with isolated eigenvalues of finite multiplicity. The quasi-compacity is due to:

- A contraction ratio

$$\tau_d := \limsup_{n \to \infty} \sup_{h \in \mathcal{H}^n_{(d)}} \sup_{\mathbf{x} \in \mathcal{J}_{(d)}} ||\mathbf{D} h(\mathbf{x})||^{1/n} < 1$$

  τ_d is the smallest real root of $z^{d+1} + z - 1 = 0$

- A distortion constant

$$\exists L > 0, \quad ||\mathbf{D} J[h](\mathbf{x})|| \leq L \, |J[h](\mathbf{x})|, \qquad \forall h \in \mathcal{H}^\star_{(d)}, \forall \mathbf{x} \in \mathcal{J}_{(d)}$$

Spectral properties of $\mathbf{H}_{(d)}$ acting on $C^1(\mathcal{J}_{(d)})$

- $\lambda = 1$ is the unique simple dominant eigenvalue of maximum modulus, isolated from the remainder of the spectrum by a spectral gap
- The dominant eigenfunction is explicit

$$\psi_d(\mathbf{x}) = \sum_{\sigma \in \mathfrak{S}_d} \prod_{i=1}^{k} \frac{1}{1 + x_{\sigma(1)} + x_{\sigma(2)} + \ldots + x_{\sigma(i)}}$$

- Except for small d, there is no explicit expression known for the integral

$$\kappa_d := \int_{\mathcal{J}_{(d)}} \psi_d(\mathbf{x})\, d\mathbf{x}$$

The invariant density Ψ_d and the invariant measure ν_d are not explicit.

Conclusion and future work

- We have 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.
- We conclude that the `BrunGcd` algorithm is less efficient than the Knuth gcd algorithm.
- This is probably the case for all the gcd algorithms which are based on multidimensional continued fraction algorithms.
- We plan to study other costs such as the bit-complexity or to perform a distributional analysis $\rightsquigarrow$ More needs for the properties of dynamical systems.
- We plan to study finite and periodic trajectories.
- We want to conduct a systematic comparison of continued fraction algorithms with respect to Lyapunov exponents.
- We plan to analyze the extended gcd algorithm based on the LLL algorithm, even if its underlying system is quite complex to deal with.