ANALYSIS of EUCLIDEAN ALGORITHMS

An Arithmetical Instance of Dynamical Analysis

Dynamical Analysis :=

Analysis of Algorithms + Dynamical Systems

Brigitte VALLÉE (CNRS and Université de Caen, France)

Results obtained with :

Ali AKHAVI, Viviane BALADI, Jérémie BOURDON,
Eda CESARATTO, Julien CLÉMENT, Benoît DAIREAUX,
Philippe FLAJOLET, Loïck LHOTE, Véronique MAUME.

Plan of the Talk

I– The Euclid Algorithm, and the underlying dynamical system

II– The other Euclidean Algorithms

III– Probabilistic –and dynamical– analysis of algorithms

IV– Euclidean algorithms : the underlying dynamical systems

V– Dynamical analysis of Euclidean algorithms

Plan of the Talk

The (standard) Euclid Algorithm: the grand father of all the algorithms.

On the input (u, v), it computes the gcd of u and v,
together with the Continued Fraction Expansion of u/v.

The (standard) Euclid Algorithm: the grand father of all the algorithms.

On the input (u, v), it computes the gcd of u and v,
together with the Continued Fraction Expansion of u/v.

$$u_0 := v;\ u_1 := u; u_0 \geq u_1$$

$$\left\{\begin{array}{ccccccc} u_0 & = & m_1 u_1 & + & u_2 & & 0 < u_2 < u_1 \\ u_1 & = & m_2 u_2 & + & u_3 & & 0 < u_3 < u_2 \\ \dots & = & \dots & + & & & \\ u_{p-2} & = & m_{p-1} u_{p-1} & + & u_p & & 0 < u_p < u_{p-1} \\ u_{p-1} & = & m_p u_p & + & 0 & & u_{p+1} = 0 \end{array}\right\}$$

u_p is the gcd of u and v, the m_i's are the digits. p is the depth.

The (standard) Euclid Algorithm: the grand father of all the algorithms.

On the input (u, v), it computes the gcd of u and v,
together with the Continued Fraction Expansion of u/v.

$$u_0 := v;\; u_1 := u; u_0 \geq u_1$$

$$\left\{\begin{array}{ccccc} u_0 & = & m_1 u_1 & + & u_2 & & 0 < u_2 < u_1 \\ u_1 & = & m_2 u_2 & + & u_3 & & 0 < u_3 < u_2 \\ \dots & = & \dots & + & & & \\ u_{p-2} & = & m_{p-1} u_{p-1} & + & u_p & & 0 < u_p < u_{p-1} \\ u_{p-1} & = & m_p u_p & + & 0 & & u_{p+1} = 0 \end{array}\right\}$$

u_p is the gcd of u and v, the m_i's are the digits. p is the depth.

CFE of $\frac{u}{v}$:
$$\frac{u}{v} = \cfrac{1}{m_1 + \cfrac{1}{m_2 + \cfrac{1}{\ddots + \cfrac{1}{m_p}}}},$$

The underlying Euclidean dynamical system (I).

The trace of the execution of the Euclid Algorithm on (u_1, u_0) is:

$$(u_1, u_0) \to (u_2, u_1) \to (u_3, u_2) \to \ldots \to (u_{p-1}, u_p) \to (u_{p+1}, u_p) = (0, u_p)$$

The underlying Euclidean dynamical system (I).

The trace of the execution of the Euclid Algorithm on (u_1, u_0) is:

$$(u_1, u_0) \to (u_2, u_1) \to (u_3, u_2) \to \ldots \to (u_{p-1}, u_p) \to (u_{p+1}, u_p) = (0, u_p)$$

Replace the integer pair (u_i, u_{i-1}) by the rational $x_i := \dfrac{u_i}{u_{i-1}}$.

The division $u_{i-1} = m_i u_i + u_{i+1}$ is then written as

$$x_{i+1} = \frac{1}{x_i} - \left\lfloor \frac{1}{x_i} \right\rfloor \qquad \text{or} \qquad x_{i+1} = T(x_i), \qquad \text{where}$$

$$T : [0,1] \longrightarrow [0,1], \quad T(x) := \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor \quad \text{for } \ x \neq 0, \quad T(0) = 0$$

The underlying Euclidean dynamical system (I).

The trace of the execution of the Euclid Algorithm on (u_1, u_0) is:

$$(u_1, u_0) \to (u_2, u_1) \to (u_3, u_2) \to \ldots \to (u_{p-1}, u_p) \to (u_{p+1}, u_p) = (0, u_p)$$

Replace the integer pair (u_i, u_{i-1}) by the rational $x_i := \dfrac{u_i}{u_{i-1}}$.

The division $u_{i-1} = m_i u_i + u_{i+1}$ is then written as

$$x_{i+1} = \frac{1}{x_i} - \left\lfloor \frac{1}{x_i} \right\rfloor \qquad \text{or} \qquad x_{i+1} = T(x_i), \qquad \text{where}$$

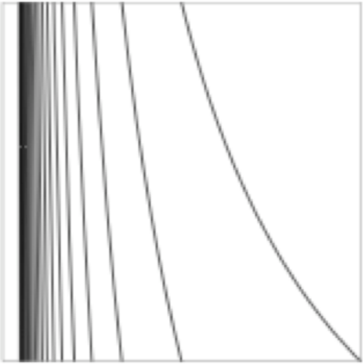
$$T : [0,1] \longrightarrow [0,1], \quad T(x) := \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor \quad \text{for } \ x \neq 0, \quad T(0) = 0$$

An execution of the Euclidean Algorithm $(x, T(x), T^2(x), \ldots, 0)$

$=$ A rational trajectory of the Dynamical System $([0,1], T)$

$=$ a trajectory that reaches 0.

The dynamical system is a continuous extension of the algorithm.



$$T(x) := \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$$

$$T_{[m]} : \left]\frac{1}{m+1}, \frac{1}{m}\right[\longrightarrow]0,1[,$$

$$T_{[m]}(x) := \frac{1}{x} - m$$

$$h_{[m]} :]0,1[\longrightarrow \left]\frac{1}{m+1}, \frac{1}{m}\right[$$

$$h_{[m]}(x) := \frac{1}{m+x}$$

The Euclidean dynamical system (II).

A dynamical system with a denumerable system of branches $(T_{[m]})_{m\geq 1}$,

$$T_{[m]} :]\frac{1}{m+1}, \frac{1}{m}[\longrightarrow]0,1[, \qquad T_{[m]}(x) := \frac{1}{x} - m$$

The set $\mathcal{H}$ of the inverse branches of T is

$$\mathcal{H} := \{\ h_{[m]} :]0,1[\longrightarrow]\frac{1}{m+1}, \frac{1}{m}[; \qquad h_{[m]}(x) := \frac{1}{m+x} \}$$

The Euclidean dynamical system (II).

A dynamical system with a denumerable system of branches $(T_{[m]})_{m\geq 1}$,

$$T_{[m]} :]\frac{1}{m+1}, \frac{1}{m}[\longrightarrow]0,1[, \qquad T_{[m]}(x) := \frac{1}{x} - m$$

The set $\mathcal{H}$ of the inverse branches of T is

$$\mathcal{H} := \{\ h_{[m]} :]0,1[\longrightarrow]\frac{1}{m+1}, \frac{1}{m}[; \qquad h_{[m]}(x) := \frac{1}{m+x}\}$$

The set $\mathcal{H}$ builds one step of the CF's.

The set $\mathcal{H}^n$ of the inverse branches of T^n builds CF's of depth n.

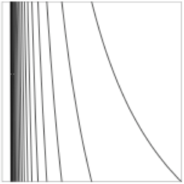
The set $\mathcal{H}^\star := \bigcup \mathcal{H}^n$ builds all the (finite) CF's.

$$\frac{u}{v} = \cfrac{1}{m_1 + \cfrac{1}{m_2 + \cfrac{1}{\ddots + \cfrac{1}{m_p}}}} = h_{[m_1]} \circ h_{[m_2]} \circ \ldots \circ h_{[m_p]}(0)$$

The Euclidean dynamical system (III).

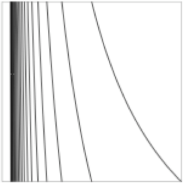
Density Transformer:

For a density f on $[0,1]$, $\mathbf{H}[f]$ is the density on $[0,1]$ after one iteration of the shift



$$\mathbf{H}[f](x) = \sum_{h \in \mathcal{H}} |h'(x)| \, f \circ h(x) = \sum_{m \in \mathbb{N}} \frac{1}{(m+x)^2} f(\frac{1}{m+x}).$$

The Euclidean dynamical system (III).



Density Transformer:

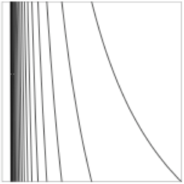
For a density f on $[0,1]$, $\mathbf{H}[f]$ is the density on $[0,1]$ after one iteration of the shift

$$\mathbf{H}[f](x) = \sum_{h \in \mathcal{H}} |h'(x)|\, f \circ h(x) = \sum_{m \in \mathbb{N}} \frac{1}{(m+x)^2} f(\frac{1}{m+x}).$$

Transfer operator (Ruelle):

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

The Euclidean dynamical system (III).



Density Transformer:

For a density f on $[0,1]$, $\mathbf{H}[f]$ is the density on $[0,1]$ after one iteration of the shift

$$\mathbf{H}[f](x) = \sum_{h \in \mathcal{H}} |h'(x)|\, f \circ h(x) = \sum_{m \in \mathbb{N}} \frac{1}{(m+x)^2} f\left(\frac{1}{m+x}\right).$$

Transfer operator (Ruelle):

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

The k-th iterate satisfies:

$$\mathbf{H}_s^k[f](x) = \sum_{h \in \mathcal{H}^k} |h'(x)|^s\, f \circ h(x)$$

Plan of the Talk

I– The Euclid Algorithm, and the underlying dynamical system

II– The other Euclidean Algorithms

III– Probabilistic –and dynamical– analysis of algorithms

IV– Euclidean algorithms : the underlying dynamical systems

V– Dynamical analysis of Euclidean algorithms

Many variants of the Euclid Algorithm.

A Euclidean algorithm:=

any algorithm which performs a sequence of divisions $v = mu + r$.

There are various possible types of Euclidean divisions

– MSB divisions [directed by the Most Significant Bits]

shorten integers on the left,

and provide a remainder r smaller than u,

(w.r.t the usual absolute value), i.e. with more zeroes on the left.

– LSB divisions [directed by the Least Significant Bits]

shorten integers on the right,

and provide a remainder r smaller than u

(w.r.t the dyadic absolute value), i.e. with more zeroes on the right.

– Mixed divisions

shorten integers both on the right and on the left,

with new zeroes both on the right and on the left.

Instances of MSB Algorithms.

– Variants according to the position of remainder r,

By Default:	$v = mu + r$	with	$0 \leq r < u$
By Excess:	$v = mu - r$	with	$0 \leq r < u$
Centered:	$v = mu + \epsilon r$	with	$\epsilon = \pm 1, \quad 0 \leq r \leq u/2$

Instances of MSB Algorithms.

– Variants according to the position of remainder r,

By Default:	$v = mu + r$	with	$0 \le r < u$
By Excess:	$v = mu - r$	with	$0 \le r < u$
Centered:	$v = mu + \epsilon r$	with	$\epsilon = \pm 1, \quad 0 \le r \le u/2$

– Subtractive Algorithm :

A division with quotient m can be replaced by m subtractions

`While` $v \ge u$ `do` $v := v - u$

An instance of a Mixed Algorithm.

The Subtractive Algorithm,

where the zeroes on the right are removed from the remainder defines the Binary Algorithm.

Subtractive Gcd Algorithm.	Binary Gcd Algorithm.
Input. u, v; $v \geq u$	**Input.** u, v odd; $v \geq u$
`While` $(u \neq v)$ `do`	`While` $(u \neq v)$ `do`
`While` $v > u$ `do`	`While` $v > u$ `do`
	$k := \nu_2(v-u);$
$v := v - u$	$v := \dfrac{v-u}{2^k};$
Exchange u and v.	Exchange u and v.
Output. u (or v).	**Output.** u (or v).

The 2-adic valuation ν_2 counts the number of zeroes on the right

An instance of a LSB Algorithm.

On a pair (u, v) with v odd and u even,

with $\nu_2(u) = k$, of the form $u := 2^k\, u'$

the LSB division writes $$v = a \cdot u' + 2^k \cdot r',$$

with $\nu_2(r') > \nu_2(u') = 0$ and $\gcd(u, v) = \gcd(r', u')$.

The pair (u', r') will be the new pair for the next step.

An execution of the LSB Algorithm:

the Tortoise and the Hare

0	10001100101000001
1	11110101100000101000
2	1100100110110101 0000
3	11000011000101 0000000
4	100110001111 00000000
5	11101001010 1000000000
6	1100000100 10000000000
7	100010001 100000000000
8	100000101 1000000000000
9	11 00000000000000
10	100000 1000000000000000
11	1000 10000000000000000
12	11 00000000000000000000
13	1000000000000000000000000

Three main outputs for any Euclidean Algorithm

– the $\gcd(u, v)$ itself

Essential in exact rational computations,

for keeping rational numbers under their irreducible forms

60% of the computation time in some symbolic computations

Three main outputs for any Euclidean Algorithm

– the $\gcd(u, v)$ itself

Essential in exact rational computations,

for keeping rational numbers under their irreducible forms

60% of the computation time in some symbolic computations

– the Continued Fraction Expansion CFE (u/v)

Often used directly in computation over rationals.

Three main outputs for any Euclidean Algorithm

– the $\gcd(u, v)$ itself

Essential in exact rational computations,

for keeping rational numbers under their irreducible forms

60% of the computation time in some symbolic computations

– the Continued Fraction Expansion CFE (u/v)

Often used directly in computation over rationals.

– the modular inverse $u^{-1} \mod v$, when $\gcd(u, v) = 1$.

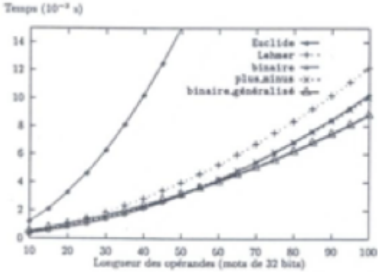
Extensively used in cryptography

A basic algorithm ... Perhaps the fifth main operation?

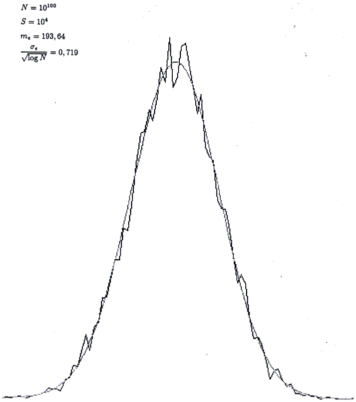
Main algorithmic questions.

- Analyse the behaviour of these various Euclidean algorithms
- Compare them with respect to various costs

and particularly the bit–complexity.



Experimental comparison
of bit–complexities.



A gaussian law
for the number of steps?

Comparison for five algorithms on the input $(2011176, 72001)$

Evolution of the remainders

Standard	Centered	By-Excess	Binary	LSB
67149	4852	4852	44849	51637
4852	779	779	1697	12485
4073	178	601	1697	2447
779	67	423	125	3733
178	23	245	125	1545
67	2	67	9	547
44	1	23	9	523
23	–	2	5	3
19	–	1	1	65
4	–	–	–	17
3	–	–	–	3
1	–	–	–	1

Comparison for five algorithms on the input $(2011176, 72001)$

Evolution of the remainders

Standard	Centered	By-Excess	Binary	LSB
67149	4852	4852	44849	51637
4852	779	779	1697	12485
4073	178	601	1697	2447
779	67	423	125	3733
178	23	245	125	1545
67	2	67	9	547
44	1	23	9	523
23	–	2	5	3
19	–	1	1	65
4	–	–	–	17
3	–	–	–	3
1	–	–	–	1

Explain the behaviour of algorithms

For instance, an execution of the LSB Algorithm : the Tortoise and the Hare

0	10001100101000001
1	111101011000000101000
2	11001001101101010000
3	110000110001010000000
4	10011000111100000000
5	111010010101000000000
6	110000010010000000000
7	100010001100000000000
8	1000001011000000000000
9	1100000000000000
10	1000001000000000000000
11	100010000000000000000
12	110000000000000000000
13	10000000000000000000000

Plan of the Talk

I– The Euclid Algorithm, and the underlying dynamical system

II– The other Euclidean Algorithms

III– Probabilistic –and dynamical– analysis of algorithms

IV– Euclidean algorithms : the underlying dynamical systems

V– Dynamical analysis of Euclidean algorithms

Probabilistic Analysis of Algorithms

An algorithm with a set of inputs Ω,

and a parameter (or a cost) C defined on Ω which describes

– the execution of the algorithm (number of iterations, bit–complexity)

– or the geometry of the output

(here: the continued fraction)

Probabilistic Analysis of Algorithms

An algorithm with a set of inputs Ω,

and a parameter (or a cost) C defined on Ω which describes

– the execution of the algorithm (number of iterations, bit–complexity)

– or the geometry of the output

(here: the continued fraction)

– Gather the inputs wrt to their sizes (here, their number of bits)

$$\Omega_n := \{(u, v) \in \Omega, \quad \text{size}(u, v) = n\}.$$

– Consider a distribution on Ω_n (for instance the uniform distribution),

– Study the cost C on Ω_n in a probabilistic way:

– Estimate the mean value of $C_n := C_{|\Omega_n}$, its variance, its distribution...

in an asymptotic way (for $n \to \infty$)

The main costs of interest for Euclidean Algorithms

– The additive costs, which depend on the digits

$$C(u,v) := \sum_{i=1}^{p} c(m_i)$$

if $c = 1$, then $C :=$ the number of iterations

if $c = \mathbf{1}_{m_0}$, then $C :=$ the number of digits equal to m_0

if $c = \ell$ (the binary length), then $C :=$ the length of the CFE

The main costs of interest for Euclidean Algorithms

– The additive costs, which depend on the digits

$$C(u,v) := \sum_{i=1}^{p} c(m_i)$$

if $c = 1$, then $C :=$ the number of iterations
if $c = \mathbf{1}_{m_0}$, then $C :=$ the number of digits equal to m_0
if $c = \ell$ (the binary length), then $C :=$ the length of the CFE

– The bit complexity (not an additive cost)

$$C(u,v) := \sum_{i=1}^{p} \ell(u_i)\,\ell(m_i)$$

The results (I)
Previous results

– mostly in the average-case,

– only for the number of iterations, and specific to particular algorithms...

– well–described in Knuth's book (Tome II)

The results (I)
Previous results

– mostly in the average-case,

– only for the number of iterations, and specific to particular algorithms...

– well–described in Knuth's book (Tome II)

Heilbronn, Dixon, Rieger (70): Standard and Centered Alg.

Yao and Knuth (75): Subtractive Alg.

Brent (78): Binary Alg (partly heuristic),

Hensley (94) : A distributional study for the Standard Alg.

Stehlé and Zimmermann (05) : LSB Alg (experiments)

The new results

With Dynamical Analysis method, our group [1995 $\rightarrow$ now] obtains

The new results

With Dynamical Analysis method, our group [1995 $\rightarrow$ now] obtains

– a complete classification into two classes,

  – the Fast Class $=\{$Standard, Centered, Binary, LSB$\}$,

  – the Slow Class $=$ $\{$By-Excess, Subtractive$\}$.

The new results

With Dynamical Analysis method, our group [1995 $\to$ now] obtains

– a complete classification into two classes,

  – the Fast Class $=${Standard, Centered, Binary, LSB},

  – the Slow Class $=$ {By-Excess, Subtractive}.

– an average-case analysis of a broad class of costs,

  – all the additive costs

  – and also the bit–complexity.

The new results

With Dynamical Analysis method, our group [1995 $\to$ now] obtains

– a complete classification into two classes,

  – the Fast Class ={Standard, Centered, Binary, LSB},

  – the Slow Class = {By-Excess, Subtractive}.

– an average-case analysis of a broad class of costs,

  – all the additive costs

  – and also the bit–complexity.

– a distributional analysis of a subclass of the Fast Class,

the Good Class = {Standard, Centered}.

Asymptotic gaussian laws hold for:

– P, and additive costs of moderate growth,

– the remainder size $\log u_i$ for $i \sim \delta P$,

– the bit-complexity of the extended Alg.

Here, focus on average-case results $(n := \text{input size})$

Here, focus on average-case results ($n :=$ input size)

– For the Fast Class $=\{$Standard, Centered, Binary, LSB $\}$,

– the mean values of costs P, C are linear wrt n,

– the mean bit-complexity is quadratic.

$$\mathbb{E}_n[P] \sim \frac{2\log 2}{h(\mathcal{S})} n, \qquad \mathbb{E}_n[C] \sim \frac{2\log 2}{h(\mathcal{S})} \mu[c]\, n, \qquad \mathbb{E}_n[B] \sim \frac{\log 2}{h(\mathcal{S})} \mu[\ell]\, n^2.$$

$h(\mathcal{S})$ is the entropy of the system, $\mu[c]$ the mean value of step–cost c.

– Moreover, these costs are concentrated: $\mathbb{E}_n[C^k] \sim \mathbb{E}_n[C]^k$

Here, focus on average-case results ($n :=$ input size)

– For the Fast Class ={Standard, Centered, Binary, LSB } ,

– the mean values of costs P, C are linear wrt n,

– the mean bit-complexity is quadratic.

$$\mathbb{E}_n[P] \sim \frac{2\log 2}{h(\mathcal{S})} n, \qquad \mathbb{E}_n[C] \sim \frac{2\log 2}{h(\mathcal{S})} \mu[c]\, n, \qquad \mathbb{E}_n[B] \sim \frac{\log 2}{h(\mathcal{S})} \mu[\ell]\, n^2.$$

$h(\mathcal{S})$ is the entropy of the system, $\mu[c]$ the mean value of step–cost c.

– Moreover, these costs are concentrated: $\qquad \mathbb{E}_n[C^k] \sim \mathbb{E}_n[C]^k$

– For the Slow Class = {By-Excess, Subtractive},

– the mean values of costs P, C are quadratic,

– the mean bit-complexity of B is cubic,

– the moments of order $k \geq 2$ are exponential: $\quad \mathbb{E}_n[C^k] = \Theta(2^{n(k-1)})$.

The main constant $h(\mathcal{S})$ is the entropy of the Dynamical System.

A well-defined mathematical object, computable.

The main constant $h(\mathcal{S})$ is the entropy of the Dynamical System.

A well-defined mathematical object, computable.

– Related to classical constants for the first two algs

$$h(\mathcal{S}) = \frac{\pi^2}{6\log 2} \sim 2.37 \quad \text{[Standard]}, \qquad h(\mathcal{S}) = \frac{\pi^2}{6\log\phi} \sim 3.41 \quad \text{[Centered]}.$$

The main constant $h(\mathcal{S})$ is the entropy of the Dynamical System.

A well-defined mathematical object, computable.

– Related to classical constants for the first two algs

$$h(\mathcal{S}) = \frac{\pi^2}{6\log 2} \sim 2.37 \ \text{[Standard]}, \quad h(\mathcal{S}) = \frac{\pi^2}{6\log\phi} \sim 3.41 \ \text{[Centered]}.$$

– For the LSB alg, $h(\mathcal{S}) = 4 - 2\gamma \sim 3.91$ involves the Lyapounov exponent γ of the set of random matrices, where

$$N_{a,k} = \frac{1}{2^k}\begin{pmatrix} 0 & 2^k \\ 2^k & a \end{pmatrix} \text{ with } k \geq 1, a \text{ odd}, |a| < 2^k \text{ is taken with prob. } 2^{-2k},$$

The main constant $h(\mathcal{S})$ is the entropy of the Dynamical System.

A well-defined mathematical object, computable.

– Related to classical constants for the first two algs

$$h(\mathcal{S}) = \frac{\pi^2}{6\log 2} \sim 2.37 \ \ \text{[Standard]}, \qquad h(\mathcal{S}) = \frac{\pi^2}{6\log \phi} \sim 3.41 \ \ \text{[Centered]}.$$

– For the LSB alg, $h(\mathcal{S}) = 4 - 2\gamma \sim 3.91$ involves the Lyapounov exponent γ of the set of random matrices, where

$$N_{a,k} = \frac{1}{2^k}\begin{pmatrix} 0 & 2^k \\ 2^k & a \end{pmatrix} \text{ with } k \geq 1, a \text{ odd}, |a| < 2^k \text{ is taken with prob. } 2^{-2k},$$

– For the Binary alg, $h(\mathcal{S}) = \pi^2 f(1) \sim 3.6$ involves the value $f(1)$ of the unique density which satisfies the functional equation

$$f(x) = \sum_{k\geq 1} \sum_{\substack{a \text{ odd} \\ 1 \leq a < 2^k}} \left(\frac{1}{2^k x + a}\right)^2 f\left(\frac{1}{2^k x + a}\right)$$

Precise comparisons between the four Fast Algorithms

Algs	Nb of iterations	Bit-complexity
Standard	$0.584\,n$	$1.242\,n^2$
Centered	$0.406\,n$	$1.126\,n^2$
(Ind.) Binary	$0.381\,n$	$0.720\,n^2$
LSB	$0.511\,n$	$1.115\,n^2$

Main principles of Dynamical Analysis :=

Analysis of Algorithms + Dynamical Systems

1– Interaction between the discrete world and the continuous world.

Three steps.

(a) The discrete algorithm is extended into a continuous process.....

(b) .... which is studied – more easily, using all the analytic tools.

1– Interaction between the discrete world and the continuous world.

Three steps.

(a) The discrete algorithm is extended into a continuous process.....

(b) .... which is studied – more easily, using all the analytic tools.

(c) Returning to the discrete algorithm,

with various principles of transfer from continuous to discrete.

The discrete data are of zero measure amongst the continuous data.

Main tools for probabilistic analysis of algorithms
2– Generating functions ?

A classical tool : Generating functions of various types

$$A(z) := \sum_{n\geq 0} a_n z^n, \qquad \hat{A}(z) := \sum_{n\geq 0} a_n \frac{z^n}{n!}, \qquad \widetilde{A}(s) := \sum_{n\geq 1} \frac{a_n}{n^s}$$

Directly used when the distribution of data does not change too much
during the execution of the algorithm
(for instance: the Euclid Algorithm on polynomials)

Main tools for probabilistic analysis of algorithms
2– Generating functions ?

A classical tool : Generating functions of various types

$$A(z) := \sum_{n\geq 0} a_n z^n, \qquad \hat{A}(z) := \sum_{n\geq 0} a_n \frac{z^n}{n!}, \qquad \widetilde{A}(s) := \sum_{n\geq 1} \frac{a_n}{n^s}$$

Directly used when the distribution of data does not change too much during the execution of the algorithm
(for instance: the Euclid Algorithm on polynomials)

Here, this is not the case .... due to the existence of the carries

The study of the dynamical system underlying the algorithm explains how the distribution of data evolves during the execution of the algorithm.

It also describes the behaviour of the generating functions of costs...

Main tools for probabilistic analysis of algorithms
3- Dynamical Analysis –main principles.

Input.- A discrete algorithm.

Step 1.- Extend the discrete algorithm into a continuous process,

i.e. a dynamical system. (X, V) X compact, $V : X \to X$,

where the discrete alg. gives rise to particular trajectories.

Main tools for probabilistic analysis of algorithms
3- Dynamical Analysis –main principles.

Input.- A discrete algorithm.

Step 1.- Extend the discrete algorithm into a continuous process,

i.e. a dynamical system. (X, V) X compact, $V : X \to X$,

where the discrete alg. gives rise to particular trajectories.

Step 2.- Study this dynamical system, via its generic trajectories.

A main tool: the transfer operator.

Main tools for probabilistic analysis of algorithms
3- Dynamical Analysis –main principles.

Input.- A discrete algorithm.

Step 1.- Extend the discrete algorithm into a continuous process,
i.e. a dynamical system. (X, V) X compact, $V : X \to X$,
where the discrete alg. gives rise to particular trajectories.

Step 2.- Study this dynamical system, via its generic trajectories.
A main tool: the transfer operator.

Step 3.- Coming back to the algorithm: we need proving that
"the discrete trajectories behaves like the generic trajectories".
Use the transfer operator as a generating operator,
which generates itself ..... the generating functions

Output.- Probabilistic analysis of the Algorithm.

Dynamical analysis of a Euclidean Algorithm.

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$$\Downarrow$$

Arithmetic properties of the division

$$\Downarrow$$

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$\Downarrow$

Arithmetic properties of the division

$\Downarrow$

Geometric properties of the branches

$\Downarrow$

Spectral properties of the transfer operator

$\Downarrow$

Analytical properties of the Quasi-Inverse of the transfer operator

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$\Downarrow$

Arithmetic properties of the division

$\Downarrow$

Geometric properties of the branches

$\Downarrow$

Spectral properties of the transfer operator

$\Downarrow$

Analytical properties of the Quasi-Inverse of the transfer operator

$\Downarrow$

Analytical properties of the generating function

$\Downarrow$

Probabilistic analysis of the Euclidean Algorithm

Plan of the Talk

I– The Euclid Algorithm, and the underlying dynamical system

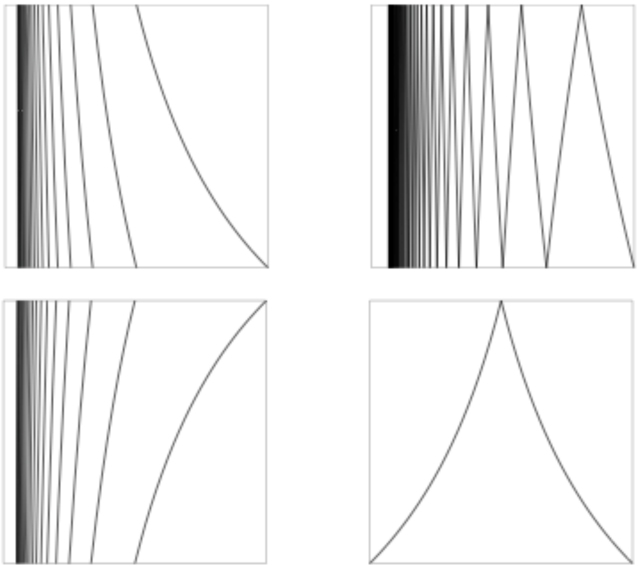
II– The other Euclidean Algorithms

III– Probabilistic –and dynamical– analysis of algorithms

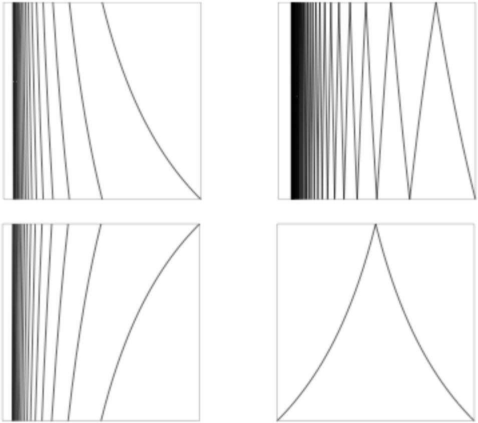
IV– Euclidean algorithms : the underlying dynamical systems

V– Dynamical analysis of Euclidean algorithms

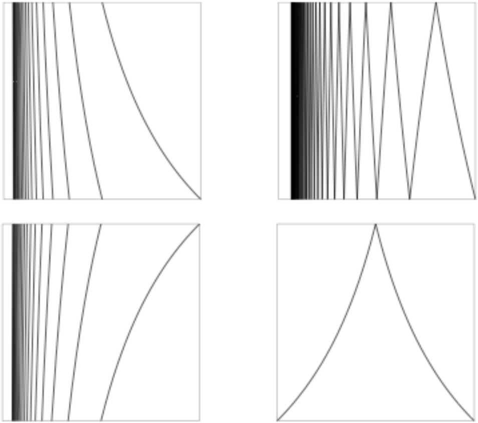
Four Euclidean dynamical systems (related to MSB divisions)



Four Euclidean dynamical systems (related to MSB divisions)

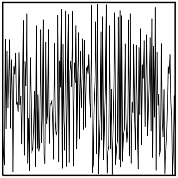


Four Euclidean dynamical systems (related to MSB divisions)

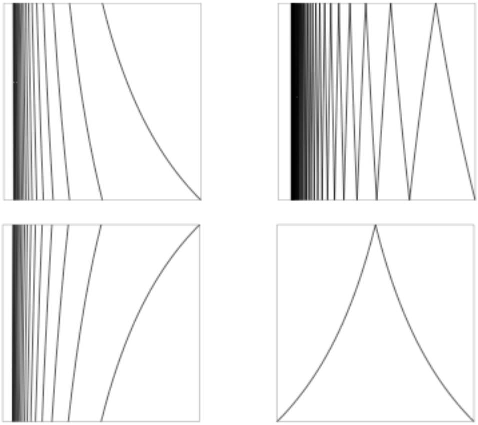


Two different classes

Fast Class

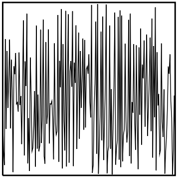


Four Euclidean dynamical systems (related to MSB divisions)

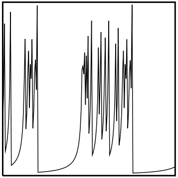


Two different classes

Fast Class



Slow Class



Dynamical Systems relative to MSB Algorithms.

Key Property : Expansiveness of branches of the shift T

$$|T'(x)| \geq A > 1 \text{ for all } x \text{ in } \mathcal{I}$$

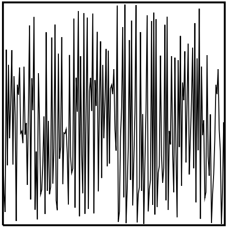
When true, this implies a chaotic behaviour for trajectories.

The associated algos are Fast and belong to the Good Class

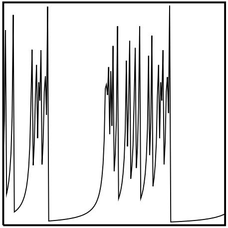
When this condition is violated at only one indifferent point,

this leads to intermittency phenomena.

The associated algos are Slow.



Chaotic Orbit [Fast Class],



Intermittent Orbit [SlowClass].

Induction Method

For a DS (I, T) with a "slow" branch relative to a slow interval J, contract each part of the trajectory which belongs to J into one step.

This (often) transforms the slow DS (I, T) into a fast one (I, S):

While $x \in J$ do $x := T(x)$;
$S(x) := T(x)$;

The Induced DS of the Subtractive Alg $=$ the DS of the Standard Alg.

Two other Euclidean dynamical systems, related to mixed or LSB divisions: the Binary Algorithm and the LSB Algorithm.

These algorithms use the 2–adic valuation ν .... only defined on rationals.

The 2–adic valuation ν is extended to a real random variable ν with

$$\Pr[\nu = k] = 1/2^k \qquad \text{for} \quad k \geq 1.$$

This gives rise to probabilistic dynamic systems.

(I) The DS relative to the Binary Algorithm

Two other Euclidean dynamical systems, related to mixed or LSB divisions: the Binary Algorithm and the LSB Algorithm.

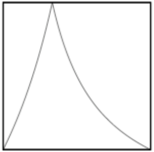
These algorithms use the 2–adic valuation ν .... only defined on rationals.

The 2–adic valuation ν is extended to a real random variable ν with

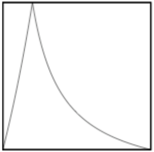
$$\Pr[\nu = k] = 1/2^k \qquad \text{for} \quad k \geq 1.$$

This gives rise to probabilistic dynamic systems.

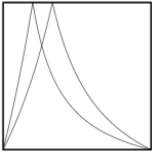
(I) The DS relative to the Binary Algorithm



$k = 1$



$k = 2$



$k = 1$ and $k = 2$

Two other Euclidean dynamical systems, related to mixed or LSB divisions: the Binary Algorithm and the LSB Algorithm.

These algorithms use the 2–adic valuation ν .... only defined on rationals.

The 2–adic valuation ν is extended to a real random variable ν with

$$\Pr[\nu = k] = 1/2^k \qquad \text{for} \quad k \geq 1.$$

This gives rise to probabilistic dynamic systems.

(II) The DS relative to the LSB Algorithm

Two other Euclidean dynamical systems, related to mixed or LSB divisions:
the Binary Algorithm and the LSB Algorithm.

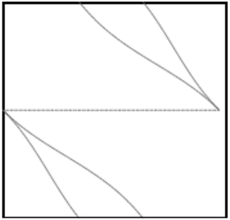
These algorithms use the 2–adic valuation ν .... only defined on rationals.

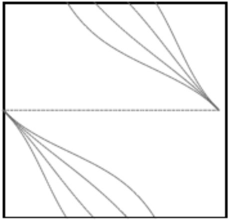
The 2–adic valuation ν is extended to a real random variable ν with

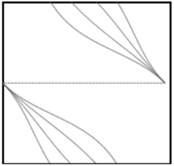
$$\Pr[\nu = k] = 1/2^k \qquad \text{for} \quad k \geq 1.$$

This gives rise to probabilistic dynamic systems.

(II) The DS relative to the LSB Algorithm







In all the cases (probabilistic or deterministic),
the density transformer $\mathbf{H}$ expresses the new density f_1 as a function of the old density f_0, as $f_1 = \mathbf{H}[f_0]$.
It involves the set $\mathcal{H}$

$$\mathbf{H}[f](x) := \sum_{h\in\mathcal{H}} \delta_h \cdot |h'(x)| \cdot f\circ h(x) \qquad \text{(here, } \delta_h = \Pr[h]\text{)}$$

With a cost $c : \mathcal{H} \to \mathbf{R}^+$, and two parameters (s, w), it gives rise to the bivariate transfer operator

$$\mathbf{H}_{s,w}[f](x) := \sum_{h\in\mathcal{H}} {\delta_h}^s \cdot \exp[wc(h)] \cdot |h'(x)|^s \cdot f\circ h(x)$$

and the weighted transfer operator

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

Plan of the Talk

I– The Euclid Algorithm, and the underlying dynamical system

II– The other Euclidean Algorithms

III– Probabilistic –and dynamical– analysis of algorithms

IV– Euclidean algorithms : the underlying dynamical systems

V– Dynamical analysis of Euclidean algorithms

The Dirichlet series of cost C.

If Ω is the whole set of inputs, the Dirichlet generating function of C

$$S_C(s) = \sum_{(u,v)\in\Omega} \frac{C(u,v)}{|(u,v)|^{2s}} = \sum_{m\geq 1} \frac{c_m}{m^{2s}} \qquad \text{with} \quad c_m := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)|=m}} C(u,v)$$

The Dirichlet series of cost C.

If Ω is the whole set of inputs, the Dirichlet generating function of C

$$S_C(s) = \sum_{(u,v)\in\Omega} \frac{C(u,v)}{|(u,v)|^{2s}} = \sum_{m\geq 1} \frac{c_m}{m^{2s}} \qquad \text{with} \quad c_m := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)|=m}} C(u,v)$$

is used for expressing the mean value $\mathbb{E}_n[C]$ of C on Ω_n,

with $\Omega_n := \{(u,v) \in \Omega; \quad \ell\,(|(u,v)|) = n\}$

The Dirichlet series of cost C.

If Ω is the whole set of inputs, the Dirichlet generating function of C

$$S_C(s) = \sum_{(u,v)\in\Omega} \frac{C(u,v)}{|(u,v)|^{2s}} = \sum_{m\geq 1} \frac{c_m}{m^{2s}} \qquad \text{with} \quad c_m := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)|=m}} C(u,v)$$

is used for expressing the mean value $\mathbb{E}_n[C]$ of C on Ω_n,

with $\Omega_n := \{(u,v) \in \Omega; \quad \ell\left(|(u,v)|\right) = n\}$

The mean value $\mathbb{E}_n[C]$ is expressed with coefficients of $S_C(s)$ as

$$\mathbb{E}_n[C] = \frac{1}{|\Omega_n|} \sum_{m|\ell(m)=n} c_m.$$

The mixed series of cost C

Now, two parameters s and w: s marks the size, and w marks the cost,

The mixed series of cost C

Now, two parameters s and w: s marks the size, and w marks the cost,

$$S_C(s,w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^s} \exp[wC(u,v)] = \sum_{m\geq 1} \frac{c_m(w)}{m^s}$$

with $\;c_m(w) := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)=m}} \exp[wC(u,v)]$

The mixed series of cost C

Now, two parameters s and w: s marks the size, and w marks the cost,

$$S_C(s,w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^s} \exp[wC(u,v)] = \sum_{m\geq 1} \frac{c_m(w)}{m^s}$$

$$\text{with} \quad c_m(w) := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)=m}} \exp[wC(u,v)]$$

The moment generating function $\mathbb{E}_n[\exp(wC)]$ of C on Ω_n

with $\Omega_n := \{(u,v) \in \Omega; \quad \ell\left(|(u,v)|\right) = n\}$

is expressed with coefficients of $S_C(s,w)$

The mixed series of cost C

Now, two parameters s and w: s marks the size, and w marks the cost,

$$S_C(s,w) := \sum_{(u,v)\in\Omega} \frac{1}{|(u,v)|^s} \exp[wC(u,v)] = \sum_{m\geq 1} \frac{c_m(w)}{m^s}$$

$$\text{with} \quad c_m(w) := \sum_{\substack{(u,v)\in\Omega \\ |(u,v)=m}} \exp[wC(u,v)]$$

The moment generating function $\mathbb{E}_n[\exp(wC)]$ of C on Ω_n

$$\text{with } \Omega_n := \{(u,v)\in\Omega; \quad \ell\left(|(u,v)|\right) = n\}$$

is expressed with coefficients of $S_C(s,w)$

$$\mathbb{E}_n[\exp(wC)] = \frac{1}{|\Omega_n|} \sum_{m|\ell(m)=n} c_m(w) \qquad \text{with} \quad |\Omega_n| = \sum_{m|\ell(m)=n} c_m(0)$$

For the asymptotics of $\mathbb{E}_n[C]$ or $E_n[\exp(wC)]$,

For the asymptotics of $\mathbb{E}_n[C]$ or $E_n[\exp(wC)]$,

we need a precise knowledge

about the position and the nature of singularities of $S_C(s)$ or $S_C(S, w)$.

For the asymptotics of $\mathbb{E}_n[C]$ or $E_n[\exp(wC)]$,

we need a precise knowledge

about the position and the nature of singularities of $S_C(s)$ or $S_C(S, w)$.

There exist alternative expressions for $S_C(s)$, or $S_C(s, w)$

from which the position and the nature of singularities become apparent.

For the asymptotics of $\mathbb{E}_n[C]$ or $E_n[\exp(wC)]$,

we need a precise knowledge
about the position and the nature of singularities of $S_C(s)$ or $S_C(S, w)$.

There exist alternative expressions for $S_C(s)$, or $S_C(s, w)$
from which the position and the nature of singularities become apparent.

These alternative expressions will involve the (various) transfer operators.

Relations between the generating functions and the transfer operators (I).

Relations between the generating functions and the transfer operators (I).

A Euclid Algorithm builds a bijection between Ω and $\mathcal{H}^\star$:

$$(u, v) \mapsto h \quad \text{with} \quad \frac{u}{v} = h(0).$$

Then, due to the fact that branches are LFT's of determinant 1,

$$\frac{1}{v} = |h'(0)|^{1/2}, \qquad C(u, v) = c(h).$$

Relations between the generating functions and the transfer operators (I).

A Euclid Algorithm builds a bijection between Ω and $\mathcal{H}^\star$:

$$(u,v) \mapsto h \quad \text{with} \quad \frac{u}{v} = h(0).$$

Then, due to the fact that branches are LFT's of determinant 1,

$$\frac{1}{v} = |h'(0)|^{1/2}, \qquad C(u,v) = c(h).$$

Then: $$S_C(2s,w) := \sum_{(u,v)\in\Omega} \frac{1}{v^{2s}} \exp[wC(u,v)] = \sum_{h\in\mathcal{H}^\star} |h'(0)|^s \exp[wc(h)],$$

admits an alternative expression

with the quasi inverse $(I - \mathbf{H}_{s,w})^{-1}$ of the transfer operator $\mathbf{H}_{s,w}$,

$$S_C(2s,w) = (I - \mathbf{H}_{s,w})^{-1}[1](0)$$

Relations between the generating functions and the transfer operators (I).

A Euclid Algorithm builds a bijection between Ω and $\mathcal{H}^\star$:

$$(u,v) \mapsto h \quad \text{with} \quad \frac{u}{v} = h(0).$$

Then, due to the fact that branches are LFT's of determinant 1,

$$\frac{1}{v} = |h'(0)|^{1/2}, \qquad C(u,v) = c(h).$$

Then: $$S_C(2s,w) := \sum_{(u,v)\in\Omega} \frac{1}{v^{2s}} \exp[wC(u,v)] = \sum_{h\in\mathcal{H}^\star} |h'(0)|^s \exp[wc(h)],$$

admits an alternative expression

with the quasi inverse $(I - \mathbf{H}_{s,w})^{-1}$ of the transfer operator $\mathbf{H}_{s,w}$,

$$S_C(2s,w) = (I - \mathbf{H}_{s,w})^{-1}[1](0)$$

Remind : $$\mathbf{H}_{s,w}[f](x) := \sum_{h\in\mathcal{H}} |h'(x)|^s \cdot \exp[wc(h)] \cdot f \circ h(x)$$

Relation between the transfer operator and the Dirichlet series.

Since: $$S_C(2s) := \sum_{(u,v)\in\Omega} \frac{C(u,v)}{|(u,v)|^{2s}} = \frac{\partial}{\partial w} S_C(2s,w)\Big|_{w=0}$$

there is a relation

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

between $S_C(s)$ and two transfer operators:

Relation between the transfer operator and the Dirichlet series.

Since: $$S_C(2s) := \sum_{(u,v)\in\Omega} \frac{C(u,v)}{|(u,v)|^{2s}} = \frac{\partial}{\partial w} S_C(2s,w)\Big|_{w=0}$$

there is a relation

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

between $S_C(s)$ and two transfer operators:

the weighted one

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

and the quasi-inverse $(I - \mathbf{H}_s)^{-1}$ of the plain transfer operator $\mathbf{H}_s$,

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

In both cases,

In both cases,

singularities of $s \mapsto (I - \mathbf{H}_s)^{-1}$ or $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$

In both cases,

singularities of $s \mapsto (I - \mathbf{H}_s)^{-1}$ or $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$

are related to spectral properties of $\mathbf{H}_s$ or $\mathbf{H}_{s,w}$

In both cases,

singularities of $s \mapsto (I - \mathbf{H}_s)^{-1}$ or $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$

are related to spectral properties of $\mathbf{H}_s$ or $\mathbf{H}_{s,w}$

..... on a convenient functional space ..

In both cases,

singularities of $s \mapsto (I - \mathbf{H}_s)^{-1}$ or $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$

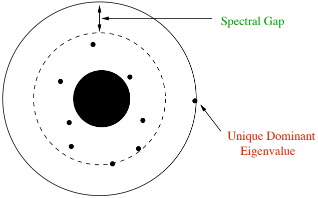
are related to spectral properties of $\mathbf{H}_s$ or $\mathbf{H}_{s,w}$

..... on a convenient functional space ..

.... which depends on the dynamical system (and thus the algorithm)...

Average-case analysis:
Expected spectral properties of $\mathbf{H}_s$

(i) *UDE* and *SG* for s near 1:
UDE – Unique dominant eigenvalue $\lambda(s, w)$ with $\lambda(1, 0) = 1$
SG – Existence of a spectral gap
(ii) Aperiodicity: On the line $\Re s = 1, s \neq 1$, the spectral radius of $\mathbf{H}_s$ is < 1



On which functional space?

The answer depends on the Dynamical System,

and thus on the algorithm....

The functional spaces where the triple $UDE + SG + \text{Aperiodicity}$ holds.

Algs	Geometry of branches	Convenient Functional space
Good Class (Standard, Centered)	Contracting	$\mathcal{C}^1(\mathcal{I})$
Binary	Not contracting	The Hardy space $\mathcal{H}(\mathcal{D})$
LSB	Contracting on average	Various spaces: $\mathcal{C}^0(J), \mathcal{C}^1(J)$ Hölder $\mathbb{H}_\alpha(J)$
Slow Class (Subtractive, By-Excess)	An indifferent point	Induction $+\ \mathcal{C}^1(\mathcal{I})$

In each case, the aperiodicity holds since the branches have not "all the same form".

The triple $UDE + SG + \mathrm{Aperiodicity}$ entails good properties for $(I - \mathbf{H}_s)^{-1}$, sufficient for applying Tauberian Theorems to $S_C(s)$.

$s = 1$ is the only pole
on the line $\Re s = 1$

Expansion near the pole $s = 1$

$$(I - \mathbf{H}_s)^{-1} \sim \frac{a}{s-1}$$

s=1

Half–plane of convergence $\Re s > 1$

No hypothesis needed
on the half–plane $\Re s < 1$.

Second direction: a distribution study.

Uniform Extraction of coefficients via the Perron Formula

For $F(s,w) := \sum_{m\geq 1} \frac{a_m(w)}{m^s}$, $\qquad \sum_{m\leq N} \sum_{q\leq m} a_q = \frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} F(s,w) \frac{N^{s+1}}{s(s+1)} ds$

. . . A first step for estimating $\mathbb{E}_N[\exp(wC)]$ . . . uniformly in w.

Perron's formula relates the MGF $\mathbb{E}_N[\exp(wC)]$ to

$$\frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} S(2s,w) \frac{N^{2s+1}}{s(2s+1)} ds$$

$$= \frac{1}{2i\pi} \int_{D-i\infty}^{D+i\infty} (I - \mathbf{H}_{s,w})^{-1}[1](0) \frac{N^{2s+1}}{s(2s+1)} ds$$

What can be expected on $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$
for dealing "uniformly" with the Perron Formula?

Dynamical analysis of a Euclidean Algorithm.

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$\Downarrow$

Arithmetic properties of the division

$\Downarrow$

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$\Downarrow$

Arithmetic properties of the division

$\Downarrow$

Geometric properties of the branches

$\Downarrow$

Spectral properties of the transfer operator

$\Downarrow$

Analytical properties of the Quasi-Inverse of the transfer operator

Dynamical analysis of a Euclidean Algorithm.

A Euclidean Algorithm

$\Downarrow$

Arithmetic properties of the division

$\Downarrow$

Geometric properties of the branches

$\Downarrow$

Spectral properties of the transfer operator

$\Downarrow$

Analytical properties of the Quasi-Inverse of the transfer operator

$\Downarrow$

Analytical properties of the generating function

$\Downarrow$

Probabilistic analysis of the Euclidean Algorithm

Here, we have used the transfer operator $\mathbf{H}_s$ of the underlying DS and studied it for complex numbers s with $\Re s \geq 1$.

Three instances of possible extensions.

– Distributional analysis of the Euclidean algorithms

– Analysis of Fast variants of the Euclidean Algorithms

Use the same transfer operator $\mathbf{H}_s$, with its behaviour for $\Re s < 1$

A vertical strip free of poles with polynomial growth for $(I - \mathbf{H}_s)^{-1}$

– Study of the Gauss algorithm (for reducing lattices) for $n = 2$

Use of an extension of the transfer operator $\mathbf{H}_s$,

which operates on functions of two variables, for $s \sim 2$

A central tool for reducing lattices in general dimensions n

Extension I

Distributional Study.

Property $US(s, w)$: Uniformity on Vertical Strips

There exist $\alpha > 0, \beta > 0$ such that,

on the vertical strip $\mathcal{S} := \{s; |\Re(s) - 1| < \alpha\}$,

and uniformly when $w \in \mathcal{W} := \{w; |\Re w] < \beta\}$,

Property $US(s,w)$: Uniformity on Vertical Strips

There exist $\alpha > 0, \beta > 0$ such that,

on the vertical strip $\mathcal{S} := \{s; |\Re(s) - 1| < \alpha\}$,

and uniformly when $w \in \mathcal{W} := \{w; |\Re w] < \beta\}$,

(i) [Strong aperiodicity] $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$ has a unique pole inside $\mathcal{S}$;
it is located at $s = \sigma(w)$ defined by $\lambda(\sigma(w), w) = 1$.

Property $US(s, w)$: Uniformity on Vertical Strips

There exist $\alpha > 0, \beta > 0$ such that,

on the vertical strip $\mathcal{S} := \{s; |\Re(s) - 1| < \alpha\}$,

and uniformly when $w \in \mathcal{W} := \{w; |\Re w] < \beta\}$,

(i) [Strong aperiodicity] $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$ has a unique pole inside $\mathcal{S}$;
it is located at $s = \sigma(w)$ defined by $\lambda(\sigma(w), w) = 1$.

(ii) [Uniform polynomial estimates] For any $\gamma > 0$, there exists $\xi > 0$ s.t,

$$(I - \mathbf{H}_{s,w})^{-1}[1] = O(|\Im s|^{\xi}) \qquad \forall s \in \mathcal{S}, \quad |t| > \gamma, \quad w \in \mathcal{W}$$

Property $US(s,w)$: Uniformity on Vertical Strips

There exist $\alpha > 0, \beta > 0$ such that,

on the vertical strip $\mathcal{S} := \{s; |\Re(s) - 1| < \alpha\}$,

and uniformly when $w \in \mathcal{W} := \{w; |\Re w] < \beta\}$,

(i) [Strong aperiodicity] $s \mapsto (I - \mathbf{H}_{s,w})^{-1}$ has a unique pole inside $\mathcal{S}$;
it is located at $s = \sigma(w)$ defined by $\lambda(\sigma(w), w) = 1$.

(ii) [Uniform polynomial estimates] For any $\gamma > 0$, there exists $\xi > 0$ s.t,

$$(I - \mathbf{H}_{s,w})^{-1}[1] = O(|\Im s|^{\xi}) \qquad \forall s \in \mathcal{S}, \quad |t| > \gamma, \quad w \in \mathcal{W}$$

With the Property *US*,

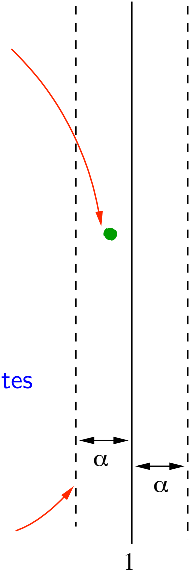
it is easy to deform the contour of the Perron Formula

and use Cauchy's Theorem . . .

Near $w = 0$, the function σ is defined by $\lambda(\sigma(w), w) = 1$

$s = \sigma(w)$ is the only pole

on the strip $|\Re s - 1| \leq \alpha$



Expansion

near the pole $s = \sigma(w)$

$$(I - \mathbf{H}_{s,w})^{-1} \sim \frac{a}{s - \sigma(w)}$$

Half–plane of

convergence $\Re s > \sigma(w)$

Uniform polynomial estimates

needed on the left domain

$1 - \alpha \leq \Re s \leq 1, |\Im s| \geq \gamma$.

Property $US(s)$ is not always true

Item (i) is always false for Dynamical Systems with affine branches.

Example: Location of poles of $(I - \mathbf{H}_s)^{-1}$ near $\Re s = 1$

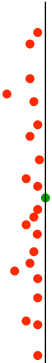
in the case of affine branches of slopes $1/p$ and $1/q$ with $p + q = 1$.

Two main cases

If $\quad \dfrac{\log p}{\log q} \in \mathbf{Q}$

Regularly spaced poles

on $\Re s = 1$



If $\quad \dfrac{\log p}{\log q} \notin \mathbf{Q}$

Only one pole at $s = 1$

on $\Re s = 1$

but accumulation of poles

on the left of $\Re s = 1$

Three main facts.

(a) There exist various conditions, (introduced by Dolgopyat),

the Conditions *UNI* that express that

"the dynamical system is quite different from a system

with piecewise affine branches"

(b) For a good Dynamical system

[complete, strongly expansive, with bounded distortion],

Conditions *UNI* imply the Uniform Property $US(s, w)$.

(c) Conditions *UNI* are true in the Euclid context.

Dolgopyat (98) proves the Item (b) but

- only for Dynamical Systems with a finite number of branches
- He considers only the $US(s)$ Property

Baladi-Vallée adapt his arguments to generalize this result:

For a Dynamical System
with a denumerable number of branches (possibly infinite),
Conditions UNI [Strong or Weak] imply $US(s, w)$.

Precisions about the UNI Conditions

Distance Δ. $\quad \Delta(h,k) := \inf_{x\in\mathcal{I}} \Psi'_{h,k}(x), \quad$ with $\quad \Psi_{h,k}(x) := \log \frac{|h'(x)|}{|k'(x)|}$

Contraction ratio ρ. $\quad \rho := \limsup \left(\{\max |h'(x)|; h \in \mathcal{H}^n, x \in \mathcal{I}\}\right)^{1/n}.$

Probability $\Pr_n$ on $\mathcal{H}^n \times \mathcal{H}^n$. $\quad \Pr_n(h,k) := |h(\mathcal{I})| \cdot |k(\mathcal{I})|$

For a system $\mathcal{C}^2$–conjugated with a piecewise-affine system :
For any $\hat\rho$ with $\rho < \hat\rho < 1$, for any n, $\quad \Pr_n[\Delta < \hat\rho^n] = 1$

Strong Condition UNI.

For any $\hat\rho$ with $\rho < \hat\rho < 1$, for any n, $\quad \Pr_n[\Delta < \hat\rho^n] \ll \hat\rho^n$

Weak Condition UNI.

$\exists D > 0, \exists n_0 \geq 1$, $\forall n \geq n_0$, $\quad \Pr_n[\Delta \leq D] < 1.$

Extension II

Mean bit–complexity of fast variants of the Euclid Algorithm

Mean bit–complexity of fast variants of the Euclid Algorithm (I)

Main principles of Fast Euclid Algorithms:

– Based on a Divide and Conquer principle with two recursive calls.

– Study "slices" of the original Euclid Algorithm

– begin when the data has already lost δn bits.

– end when the data has lost γn additional bits.

– Replace large divisions by small divisions and large multiplications.

– Use fast multiplication algorithms (based on the FFT)

of complexity $n \log n \, a(n)$

with $a(n) = \log \log n$ [Schönhage Strassen]

now $a(n) = 2^{O(\log^\star n)}$ [Fürer, 2007]

with $\log^\star n =$ the smallest integer k for which $\log^{(k)} n < 1$

We obtain the mean bit-complexity of (variants of) these algorithms

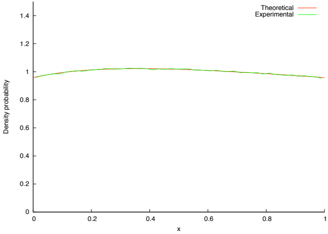
$$\Theta(n(\log n)^2 a(n))$$

with a precise estimate of the hidden constants

Analysis based on the answer to the question:

What is the distribution of the data

when they have already lost a fraction δ of its bits?



Unexpected occurrence

of a particular density ψ

$$\psi(x) = \frac{12}{\pi^2} \sum_{m \geq 1} \frac{\log(m+x)}{(m+x)(m+x+1)}$$

distinct of the Gauss density

$$\varphi(x) = \frac{1}{\log 2} \frac{1}{1+x}$$

Extension III

Probabilistic analysis of the Gauss Algorithm

The general problem of lattice reduction

A lattice of $\mathbb{R}^n$ = a discrete additive subgroup of $\mathbb{R}^n$.

A lattice $\mathcal{L}$ possesses a basis $B := (b_1, b_2, \ldots, b_p)$ with $p \leq n$,

$$\mathcal{L} := \{x \in \mathbb{R}^n; \quad x = \sum_{i=1}^{b} x_i b_i, \qquad x_i \in \mathbb{Z}\}$$

... and in fact, an infinite number of bases....

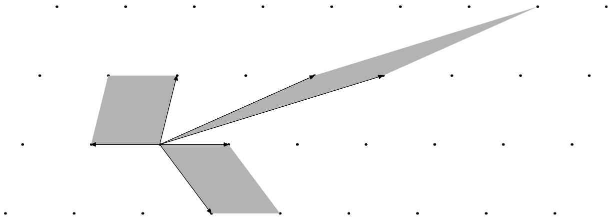
If now $\mathbb{R}^n$ is endowed with its (canonical) Euclidean structure, there exist bases (called reduced) with good Euclidean properties: their vectors are short enough and almost orthogonal.

Lattice reduction Problem : From a lattice $\mathcal{L}$ given by a basis B, construct from B a reduced basis $\hat{B}$ of $\mathcal{L}$.

Many applications of this problem in various domains:

number theory, arithmetics, discrete geometry..... and cryptology.

Lattice reduction algorithms in the two dimensional case.



Lattice Reduction in two dimensions.

Up to an isometry, the lattice $\mathcal{L}$ is a subset of $\mathbb{R}^2$ or..... $\mathbb{C}$.

To a pair $(u, v) \in \mathbb{C}^2$, with $u \neq 0$, we associate a unique $z \in \mathbb{C}$:

$$z := \frac{v}{u} = \frac{(u \cdot v)}{|u|^2} + i\frac{\det(u, v)}{|u|^2}.$$

Up to a similarity, the lattice $\mathcal{L}(u, v)$ becomes $\mathcal{L}(1, z) =: L(z)$.

Bad bases (u, v) correspond to complex z with small $|\Im z|$.

Three main facts in two dimensions.

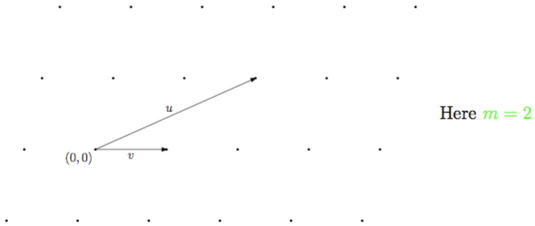
– The existence of an optimal basis = a minimal basis

– A characterization of an optimal basis.

– An efficient algorithm which finds it = The Gauss Algorithm.

The Gauss algorithm is an extension of the Euclid algorithm.
It performs integer translations – seen as “vectorial” divisions–

$$u = qv + r \quad \text{with} \quad q = \left\lfloor \Re\left(\frac{u}{v}\right) \right\rceil = \left\lfloor \frac{u \cdot v}{|v|^2} \right\rceil, \quad \left|\Re\left(\frac{r}{v}\right)\right| \leq \frac{1}{2}$$



Here $q = 2$

The Gauss algorithm is an extension of the Euclid algorithm.

It performs integer translations – seen as "vectorial" divisions–, and exchanges.

Euclid's algorithm	Gauss' algorithm
Division between real numbers $u = qv + r$ with $q = \left\lfloor \frac{u}{v} \right\rceil$ and $\left\| \frac{r}{v} \right\| \le \frac{1}{2}$	Division between complex vectors $u = qv + r$ with $q = \left\lfloor \Re\left(\frac{u}{v}\right) \right\rceil$ and $\left\| \Re\left(\frac{r}{v}\right) \right\| \le \frac{1}{2}$
Division + exchange $(v,u) \to (r,v)$ "read" on $x = v/u$ $T(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rceil$	Division + exchange $(v,u) \to (r,v)$ "read" on $z = v/u$ $T(z) = \frac{1}{z} - \left\lfloor \Re\left(\frac{1}{z}\right) \right\rceil$
Stopping condition: $x = 0$	Stopping condition: $z \in \mathcal{F}$

Analysis of the Gauss Algorithm: Instance of a Dynamical Analysis.

The analysis of the Euclid Algorithm uses the transfer operator

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

where $\mathcal{H}$ is the set of the inverse branches of (I, T)

The analysis of the Gauss Algorithm uses the transfer operator

$$\underline{\mathbf{H}}_s[F](x, y) := \sum_{h \in \mathcal{H}} |h'(x)|^{s/2} |h'(y)|^{s/2} \cdot F(h(x), h(y))$$

which acts on functions of two variables and extends $\mathbf{H}_s$, since

$$\underline{\mathbf{H}}_s[F](x, x) = \mathbf{H}_s[f](x), \qquad \text{with} \qquad f(x) := F(x, x)$$

The dynamics of the EUCLID Algorithm is described with $s = 1$.

The (uniform) dynamics of the GAUSS Algorithm is described with $s = 2$.

The (general) dynamics of the GAUSS algorithm is described with $s = 2 + r$.

When $r \to -1$, the GAUSS Algorithm tends to the EUCLID Algorithm.

THE END....