Dynamical analysis of euclidean algorithms

Benoît Daireaux

GREYC, Université de Caen
Joint works with Loïck Lohte, Véronique Maume-Deschamps et Brigitte Vallée

LORIA, Nancy

march 23rd 2006

Outline of the talk

1. Introduction
2. Dynamical analysis of euclidean algorithms
3. Application to accelerated algorithms
4. Analyse de l'algorithme LSB
5. Conclusion

Gcd computation

Integer gcd computation:

- gave rise to the eldest known algorithm...
- most complex of basic arithmetic operations
- intensively used in many areas: cryptography, computer algebra, rational computations...

Analyses of the algorithms:

- worst-case analyses not very usefull
- average-case analysis $\sim$ theoretical colement to experiments, helps to understand the mecanisms of the algorithms

Many families of algorithms

"Basics algorithms" (= sequence of divisions)

- MSB algorithms(Most Significant Bits)
  Euclid and its variants: Centered, α-euclideans, By-Excess...
- LSB algorithms (Least Significant Bits)
- Mixed algorithms
  Binary, Plus-Minus et generalisations

Accelerated algorithms (= simulation of the divisions on a truncated part of the integers)

- Lehmer-Euclide, Knuth-Schönhage, recursive LSB

State of the art

		Average case analysis	
		Digit cost	Bit comp.
MSB	Euclid	[Dix70], [Hei69], [Va]	[AV00], [Val00],
	Variants	[YK75],[Rie78], [Val03]	[Val00]
	α-euclideans	[BDV02]	[BDV02]
LSB		[DMDV05]	[DMDV05]
Mixed	Binary	[Bre76], [Va98a]	[Va98a]
Accelerated	Lehmer	[DV04]	[DV04]
	Knuth-Schönhage	[DLMV06]	[DLMV06]

		Distributional analysis	
		Digit cost	Bit Comp.
MSB	Euclid	[Hen94], [BV04],	[Lho05],
	Variants	[BV04]	
	α-euclideans		
LSB			
Mixed	Binary		
Accelerated	Lehmer		

Introduction

Dynamical analysis of euclidean algorithms

General principle

An example: Euclid algorithm

Application to accelerated algorithms

Analyse de l'algorithme LSB

Conclusion

1 Introduction
- Integer gcd algorithms
- State of the art

2 Dynamical analysis of euclidean algorithms
- General principle
- An example: Euclid algorithm

3 Application to accelerated algorithms
- The Knuth-Schönhage algorithm
- Interrupted algorithms

4 Analyse de l'algorithme LSB
- Extension continue
- Produits de matrices aléatoires

5 Conclusion

Dynamical analysis: general principle

A dynamical analysis has 3 steps:

1. Modelization into a dynamical system:
   Extension of divisions to a continuous world
2. Study of the continuous model
   Statistical properties, evolution of densities, operators
3. Return to the discrete model

Introduction

Dynamical analysis of euclidean algorithms

General principle

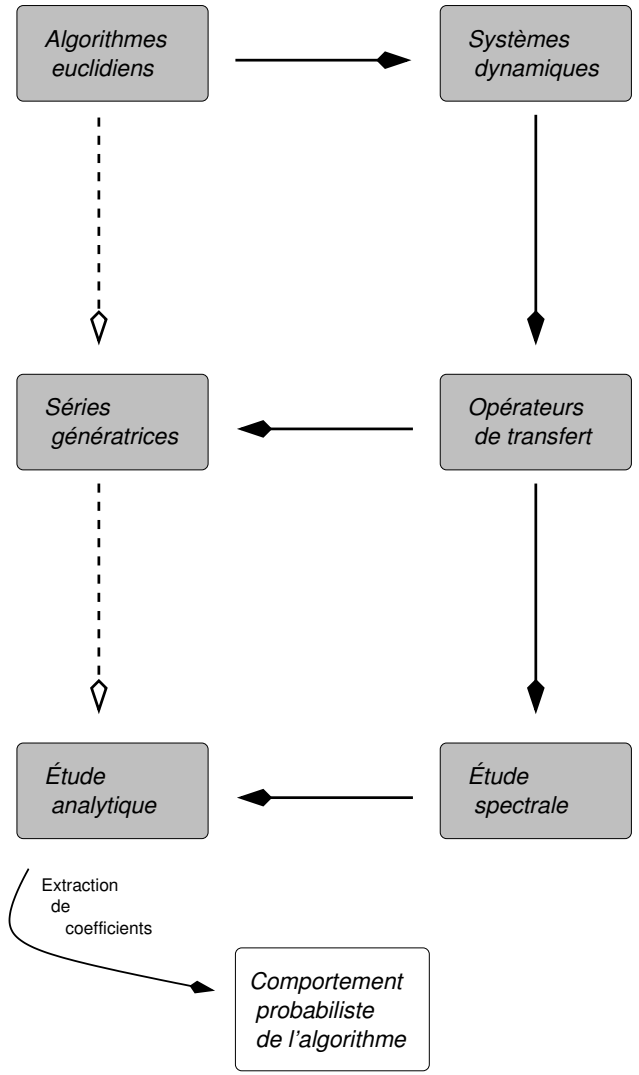
An example: Euclid algorithm

Application to accelerated algorithms

Analyse de l'algorithme LSB

Conclusion

Dynamical analysis: general principle



Euclid algorithm

Let (u, v) be an input of the algorithm, $u \geq v$.

- The algorithm performs the sequence of divisions

$$u_0 = u_1 q_1 + u_2, \quad u_1 = u_2 q_2 + u_3, \ldots u_{p-1} = u_p q_p + 0,$$

$$q_{i+1} = \left\lfloor \frac{u_i}{u_{i+1}} \right\rfloor$$

- With $\mathcal{Q} = \begin{pmatrix} 0 & 1 \\ 1 & q \end{pmatrix}$ the algorithm computes the sequence

$$\mathcal{M}_i := \mathcal{Q}_1 \cdot \mathcal{Q}_2 \cdot \mathcal{Q}_3 \cdots \mathcal{Q}_i$$

- in particular one has

$$\begin{pmatrix} u_1 \\ u_0 \end{pmatrix} = \mathcal{M}_i \cdot \begin{pmatrix} u_{i+1} \\ u_i \end{pmatrix}$$

Number of iterations of the Euclid algorithm

Study of the number of iterations $P(u, v)$ on the sets

$$\Omega := \{(u, v), \quad u > v \geq 0, \quad \text{pgcd}(u, v) = 1\},$$

$$\Omega_N := \{(u, v) \in \Omega, \quad \ell_2(u) = N\}$$

Number of iterations of the Euclid algorithm

Study of the number of iterations $P(u,v)$ on the sets

$$\Omega := \{(u,v), \quad u > v \geq 0, \quad \text{pgcd}(u,v) = 1\},$$

$$\Omega_N := \{(u,v) \in \Omega, \quad \ell_2(u) = N\}$$

Generating functions:

- $F(s) = \sum_{(u,v)\in\Omega} \frac{P(u,v)}{u^s}$

Number of iterations of the Euclid algorithm

Study of the number of iterations $P(u,v)$ on the sets

$$\Omega := \{(u,v), \quad u > v \geq 0, \quad \text{pgcd}(u,v) = 1\},$$

$$\Omega_N := \{(u,v) \in \Omega, \quad \ell_2(u) = N\}$$

Generating functions:

- $F(s) = \sum_{n \geq 1} \frac{f_n}{n^s} \qquad f_n = \sum_{\substack{(u,v) \in \Omega \\ u=n}} P(u,v)$

Number of iterations of the Euclid algorithm

Study of the number of iterations $P(u,v)$ on the sets

$$\Omega := \{(u,v), \quad u > v \geq 0, \quad \text{pgcd}(u,v) = 1\},$$
$$\Omega_N := \{(u,v) \in \Omega, \quad \ell_2(u) = N\}$$

Generating functions:

- $F(s) = \sum_{n \geq 1} \frac{f_n}{n^s} \qquad f_n = \sum_{\substack{(u,v)\in\Omega \\ u=n}} P(u,v)$

Average number of iterations on Ω_N :

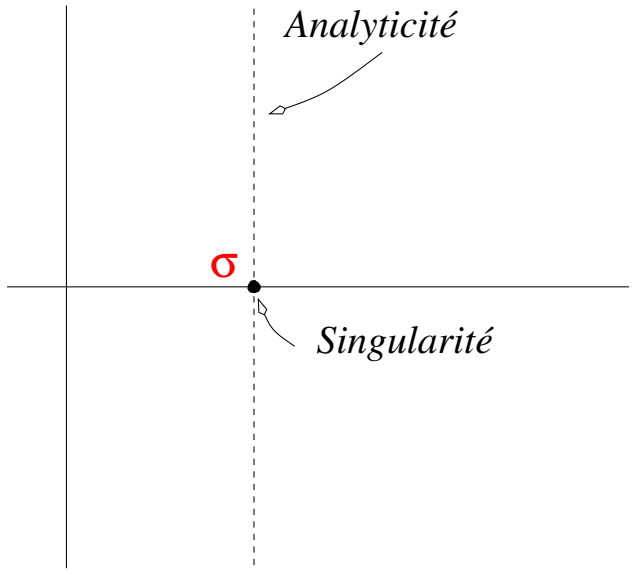
- $\mathbb{E}_N[P] = \frac{1}{|\Omega_N|} \sum_{k=2^{N-1}}^{2^N} f_k$

Théorème Taubérien

Théorème *Soit $F(s)$ une série de Dirichlet à coefficients positifs telle que $F(s)$ converge pour $\Re(s) > \sigma > 0$. Si*
(i) $F(s)$ est analytique pour $\Re(s) = \sigma, s \neq \sigma$, et
(ii) pour $\gamma \geq 0$, $F(s)$ s'écrit

$$F(s) = \frac{A(s)}{(s-\sigma)^{\gamma+1}} + C(s),$$

où $A(s)$ et $C(s)$ sont analytiques en $s = \sigma$ et $A(\sigma) \neq 0$,



Théorème Taubérien

Théorème *Soit $F(s)$ une série de Dirichlet à coefficients positifs telle que $F(s)$ converge pour $\Re(s) > \sigma > 0$. Si*
(i) $F(s)$ est analytique pour $\Re(s) = \sigma, s \neq \sigma$, et
(ii) pour $\gamma \geq 0$, $F(s)$ s'écrit

$$F(s) = \frac{A(s)}{(s-\sigma)^{\gamma+1}} + C(s),$$

où $A(s)$ et $C(s)$ sont analytiques en $s = \sigma$ et $A(\sigma) \neq 0$, Alors,

$$\sum_{n=2^{N-1}}^{2^N} f_n = K_{\gamma,\sigma} \cdot 2^{\sigma N} \cdot N^{\gamma} \cdot [1 + \epsilon(N)],$$

$$K_{\gamma,\sigma} = \frac{A(\sigma)}{\sigma\Gamma(\gamma+1)}(1-2^{-\sigma})(2\log 2)^{\gamma}, \quad \lim_{N\to\infty} \epsilon(N) = 0.$$

Euclid algorithm and continued fractions

- An execution of the algorithm: $u_0 = u, \quad u_1 = v$

$$u_0 = u_1 q_1 + u_2, \ u_1 = u_2 q_2 + u_3, \ldots, \ u_{p-1} = u_p q_p + 0$$

- Corresponds to the CFE of v/u :

$$\frac{v}{u} = \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \cfrac{1}{q_p}}}}$$

- $\frac{v}{u} = h_{q_1} \circ \cdots \circ h_{q_p}(0), \qquad h_q(x) = \frac{1}{q + x}$

Associated dynamical system

- The Gauss map $T(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$ extends to $\mathbb{R}$ the euclidean division:

$$u = vq + r \Longrightarrow \frac{r}{v} = \frac{u}{v} - q = T\left(\frac{v}{u}\right)$$

- $S = ([0,1], T)$ is the dynamical system corresponding to the algorithm
  - $T(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$
  - Partition: $\left[\frac{1}{q+1}, \frac{1}{q}\right]$ $(T_q(x) = 1/x - q)$
  - $\mathcal{H}=$ set of inverse branches:

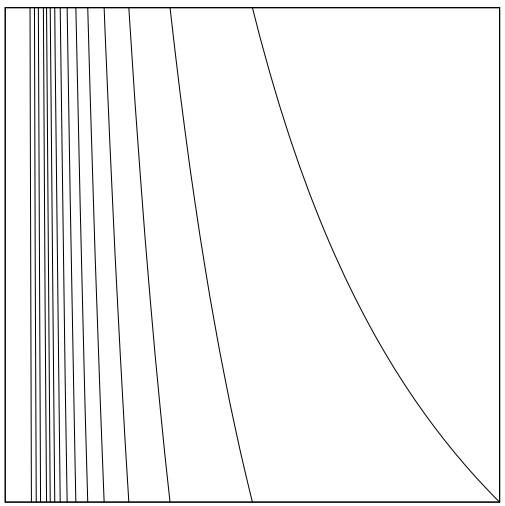
$$\mathcal{H} := \left\{ h_q; \quad h_q(x) = \frac{1}{q+x}, \quad q \geq 1 \right\}$$

Associated dynamical system

- The Gauss map $T(x) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$ extends to $\mathbb{R}$ the euclidean division:

$$u = vq + r \Longrightarrow \frac{r}{v} = \frac{u}{v} - q = T\left(\frac{v}{u}\right)$$

- $S = ([0,1], T)$ is the dynamical system corresponding to the algorithm



The Perron-Frobenius operator

Describes the evolution of densities with times

- let f be a density function on $[0,1]$
- let $f_0 = f, f_1, f_2 \ldots$ be the sequence of densities, then

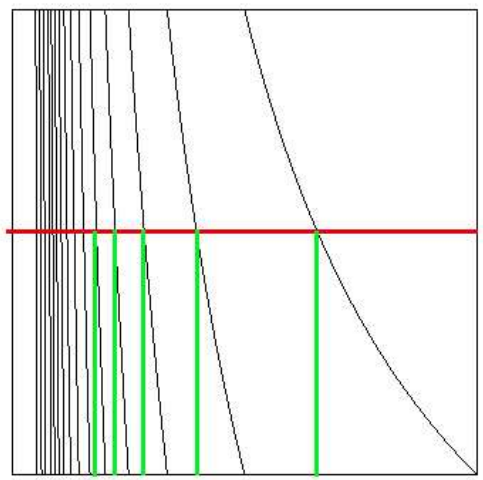
$$f_{i+1} = \mathbf{H}[f_i]$$

The Perron-Frobenius operator

Describes the evolution of densities with times

- let f be a density function on $[0, 1]$
- let $f_0 = f, f_1, f_2 \ldots$ be the sequence of densities, then

$$f_{i+1} = \mathbf{H}[f_i]$$



The Perron-Frobenius operator

Describes the evolution of densities with times

- let f be a density function on $[0,1]$
- let $f_0 = f, f_1, f_2 \ldots$ be the sequence of densities, then

$$f_{i+1} = \mathbf{H}[f_i]$$

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

The Perron-Frobenius operator

Describes the evolution of densities with times

- let f be a density function on $[0,1]$
- let $f_0 = f, f_1, f_2 \ldots$ be the sequence of densities, then

$$f_{i+1} = \mathbf{H}[f_i]$$

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

For Euclid:

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

The transfer operator

Main tool to rely discrete and continuous models

- built from **H** by adding a complex s :

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

The transfer operator

Main tool to rely discrete and continuous models

- built from $\mathbf{H}$ by adding a complex s :

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

Powers of the operator:

- multiplicative properties of derivatives:

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

Generating properties

Bijection between $\Omega^{[p]}$ et $\mathcal{H}^p$

- let $(u, v) \in \Omega^{[p]} := \{(u, v) \in \Omega, P(u, v) = p\}$
- sequence of quotient $q_1, q_2, \ldots, q_p$
- $\dfrac{v}{u} = h_{q_1} \circ h_{q_2} \circ \cdots \circ h_{q_p}(0) = h(0), \quad h \in \mathcal{H}^p$

Generating properties

Bijection between $\Omega^{[p]}$ et $\mathcal{H}^p$

- let $(u, v) \in \Omega^{[p]} := \{(u, v) \in \Omega, P(u, v) = p\}$
- sequence of quotient $q_1, q_2, \ldots, q_p$
- $\dfrac{v}{u} = h_{q_1} \circ h_{q_2} \circ \cdots \circ h_{q_p}(0) = h(0), \quad h \in \mathcal{H}^p$

Generation of denominator :

- $h(x) = \dfrac{ax + b}{cx + d}, \qquad |h'(0)| = \dfrac{|\det \ h|}{(D[h](0))^2}$
- $\det \ h = \pm 1, \quad \text{pgcd}\ (u, v) = 1 \Longrightarrow |h'(0)|^s = \dfrac{1}{u^{2s}}$

Relations operators GF

- generating function: $F(s) = \sum_{(u,v)\in\Omega} \frac{P(u,v)}{u^s}$
- bijection between $\Omega^{[p]}$ et $\mathcal{H}^p$

$$\begin{aligned}\sum_{p\geq 1} p\ \mathbf{H}_s^p[1](0) &= \sum_{p\geq 1} p \sum_{h\in\mathcal{H}^p} |h'(0)|^s \\ &= \sum_{p\geq 1} \sum_{(u,v)\in\Omega^{[p]}} \frac{P(u,v)}{u^{2s}} \\ &= F(2s)\end{aligned}$$

The GF is expressed with the quasi-inverse $(I-\mathbf{H}_s)^{-1}$

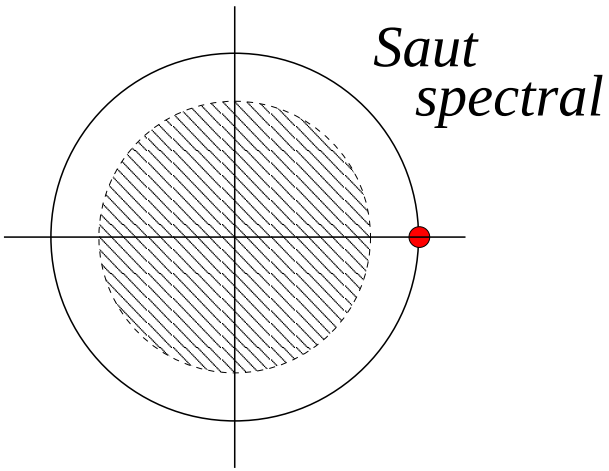
$$(I-\mathbf{H}_s)^{-1}\circ\mathbf{H}_s\circ(I-\mathbf{H}_s)^{-1}[1](0) = F(2s)$$

Study of functions replaced by study of operators

Spectral study of operators

Find a functional space such that:

- around $s = 1$ there exists a unique dominant eigenvalue, simple, $\lambda(s)$



Spectral study of operators

Find a functional space such that:

- around $s = 1$ there exists a unique dominant eigenvalue, simple, $\lambda(s)$

$$\Rightarrow \mathbf{H}_s^n[f](x) = \lambda^n(s)\mathbf{P}_s[f](x) + \mathbf{N}_s^n[f](x)$$

$$\Rightarrow (I - \mathbf{H}_s)^{-1}[f](x) \sim \frac{1}{s-1}\frac{-1}{\lambda'(1)}\psi(x)\int_0^1 f(t)dt$$

Spectral study of operators

Find a functional space such that:

- around $s = 1$ there exists a unique dominant eigenvalue, simple, $\lambda(s)$

$$\Rightarrow (I - \mathbf{H}_s)^{-1}[f](x) \sim \frac{1}{s-1}\frac{-1}{\lambda'(1)}\psi(x)\int_0^1 f(t)dt$$

- Spectral radius of the operator < 1 for $\Re(s) = 1, s \neq 1$

$$\Rightarrow s \mapsto (I - \mathbf{H}_s)^{-1} \text{ analytique sur la droite}$$

Spectral study of operators

Find a functional space such that:

- around $s = 1$ there exists a unique dominant eigenvalue, simple, $\lambda(s)$

$$\Rightarrow (I - \mathbf{H}_s)^{-1}[f](x) \sim \frac{1}{s-1}\frac{-1}{\lambda'(1)}\psi(x)\int_0^1 f(t)dt$$

- Spectral radius of the operator < 1 for $\Re(s) = 1, s \neq 1$

$$\Rightarrow s \mapsto (I - \mathbf{H}_s)^{-1} \text{ analytique sur la droite}$$

Choice of the functional space:

- for Euclid, one can choose $C^1([0,1])$

Conclusion

Spectral conditions are satisfied:

- double quasi-inverse $\longrightarrow$ double pôle for the function $F(s)$:

$$F(s) \sim \left(\frac{1}{s-1}\right)^2 \left(\frac{-1}{\lambda'(1)}\right)^2$$

- linear average number of iterations
- derivative $\lambda'(1)$ expressed in terms of entropie of the system :

$$-\lambda'(1) = h(S)$$

Theorem *On the set Ω_N, the average number of iterations of Euclid algorithm is asymptotically linear,*

$$\mathbb{E}_N[P] \sim \frac{2\log 2}{h(S)} \cdot N = \frac{12\log^2 2}{\pi^2} \cdot N$$

Introduction

Dynamical analysis of euclidean algorithms

Application to accelerated algorithms

The Knuth-Schönhage algorithm

Interrupted algorithms

Analyse de l'algorithme LSB

Conclusion

1 Introduction
- Integer gcd algorithms
- State of the art

2 Dynamical analysis of euclidean algorithms
- General principle
- An example: Euclid algorithm

3 Application to accelerated algorithms
- The Knuth-Schönhage algorithm
- Interrupted algorithms

4 Analyse de l'algorithme LSB
- Extension continue
- Produits de matrices aléatoires

5 Conclusion

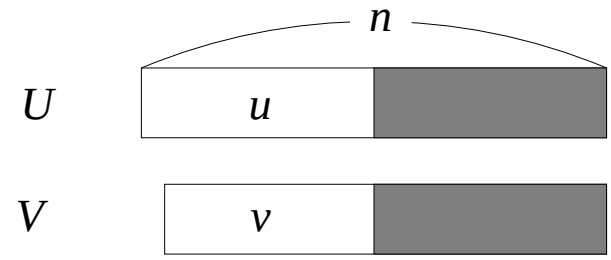
Acceleration of the algorithm

Computation of the sequence $\mathcal{Q}_1 \cdot \mathcal{Q}_2 \cdot \mathcal{Q}_3 \cdots \mathcal{Q}_p$

- Euclid : euclidean divisions: quadratic bit-complexity
- Lehmer[38] : only the first digits of u and v are necessary to compute q_1
  $\longrightarrow$ a first acceleration, still quadratic
- Knuth[71], Schönhage [71] :
  Lehmer + Divide and Conquer + Rapid Multiplication (FFT)
  $= O(n \log^2 n \log\log n)$ algorithm

The $\mathcal{H}G$ function

Property [Jebelean]



- *Soient* (U, V), $\ell(U) = n$ $(u, v) = T_{\lceil n/2 \rceil}(U, V)$
- *Let* $q_1, q_2 \ldots$ *and* $Q_1, Q_2, \ldots$ *be the sequences of quotients associated to the pairs* (u, v) *and* (U, V)
- *Let* $u_0, u_1, u_2 \ldots$ *be the sequence of remainders of Euclid on input* (a, b)
- *Let* u_k *be the last remainder such that* $\ell(u_k) > \ell(u_0)/2$.

Then the sequence of quotients q_i *and* Q_i *are the same until* $i = k - 2$.

The $\mathcal{HG}$ function

Definition

- *Let (u, v) with $u > v$ and $\ell(u) = n$.*
- *Let $u_0, u_1, u_2 \ldots$ the sequence of remainders associated to (u, v)*
- *Let u_k the last remainder such that $\ell(u_k) > n/2$*

Then the $\mathcal{HG}$ function returns $(u_k, u_{k+1}, \mathcal{M}_k) = \mathcal{HG}(u, v)$ with

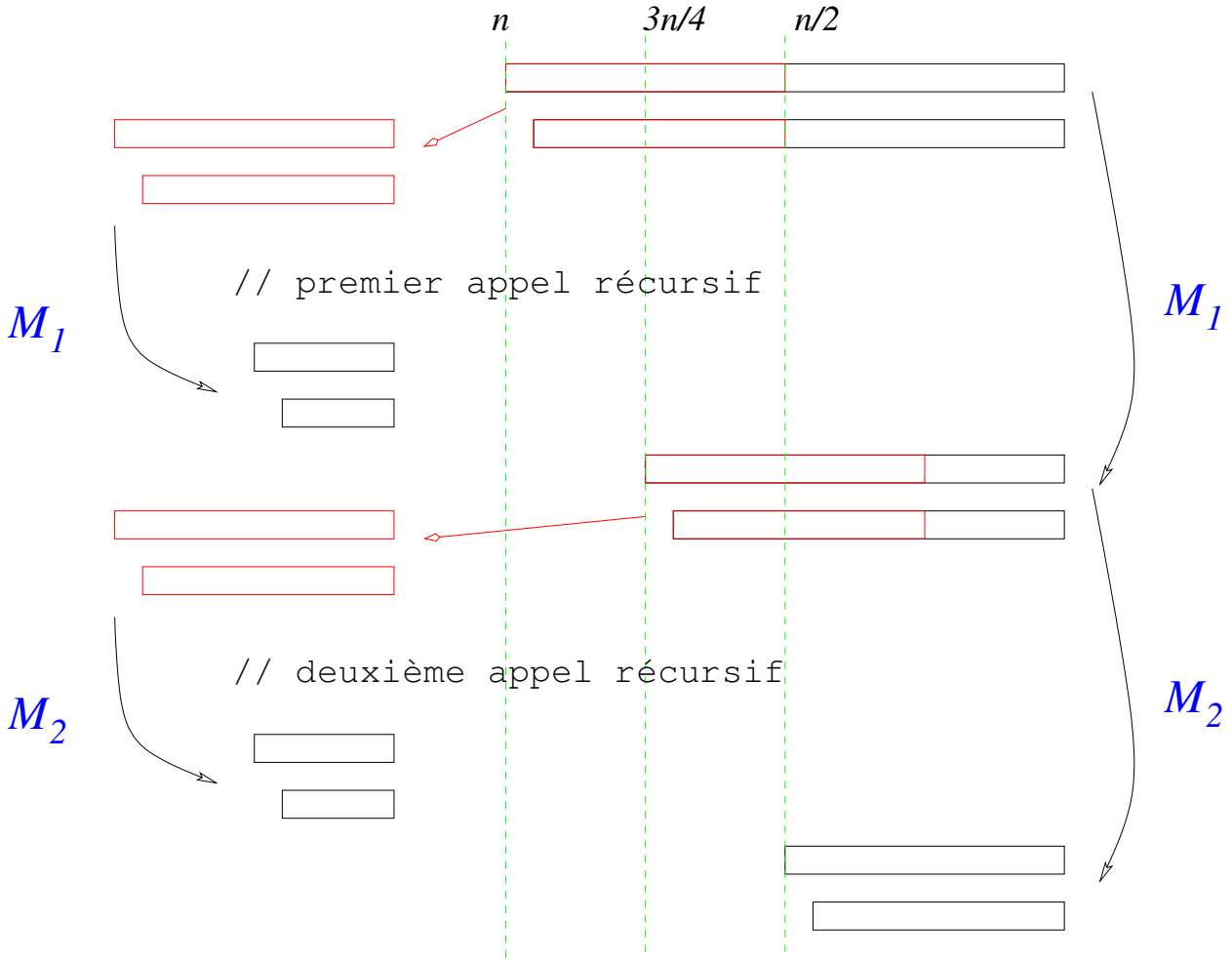
$$\mathcal{M}_k = \mathcal{Q}_1 \cdot \mathcal{Q}_2 \cdot \mathcal{Q}_3 \cdots \mathcal{Q}_{k-2}$$

Computation of $\mathcal{HG}$

- Interrupted euclid algorithms
- Knuth-Schönhage: sub-quadratic complexity

Knuth-Schönhage algorithm

Recursive computation of $\mathcal{H}G$



The algorithm $\mathcal{HG}$

Algorithm $\mathcal{HG}(U, V)$

1 $n := \ell(U)$

2 If $n \leq S$ then return $\widehat{\mathcal{E}}_{1/2}(U, V)$

3 $m := \lfloor n/2 \rfloor$; $k = n - m$

4 $(u, v) := T_m(U, V)$

5 $(c, d, \mathcal{M}_1) := \mathcal{HG}(u, v)$

6 $(C; D) := \mathcal{M}_1^{-1}(U; V)$

7 $\mathtt{Adjust}_1(C, D, \mathcal{M}_1)$

8 $(c, d) := T_k(C, D)$

9 $(e, f, M_2) := \mathcal{HG}(c, d)$

10 $(E; F) := \mathcal{M}_2^{-1}(C; D)$

11 $\mathtt{Adjust}_2(E, F, \mathcal{M}_2)$

12 Return $(E, F, \mathcal{M} := \mathcal{M}_1 \cdot \mathcal{M}_2)$

Comments

```
1    n := ℓ(U)
2    If n ≤ S then return Ê_{1/2}(U, V)
```

- If $\ell(U)$ is smaller than a treshold S, use euclid to compute $\mathcal{HG}(U, V, S)$.
- In practice: S fixed, computed experimentaly

Comments

```
1   n := ℓ(U)
2   If n ≤ S then return Ê_{1/2}(U, V)
3   m := ⌊n/2⌋; k = n − m
4   (u, v) := T_m(U, V)
5   (c, d, M_1) := HG(u, v, S)
```

- First recursive call on integers of size $n/2$
- Matrix $\mathcal{M}_1$ of size $n/4$

Comments

1	$n := \ell(U)$
2	`If` $n \leq S$ `then return` $\widehat{\mathcal{E}}_{1/2}(U, V)$
3	$m := \lfloor n/2 \rfloor; k = n - m$
4	$(u, v) := T_m(U, V)$
5	$(c, d, \mathcal{M}_1) := \mathcal{HG}(u, v, S)$
6	$(C; D) := \mathcal{M}_1^{-1}(U; V)$
7	$\mathtt{Adjust}_1(C, D, \mathcal{M}_1)$

- Multiplication matrix $\times$ vector.
- The pair (C, D) is of size $3n/4$.
- Function $\mathtt{Adjust}_1$: one euclidean division to ensure $\ell(C) < 3n/4$.

Comments

```
1   n := ℓ(U)
2   If n ≤ S then return Ê_{1/2}(U, V)
3   m := ⌊n/2⌋; k = n − m
4   (u, v) := T_m(U, V)
5   (c, d, M_1) := HG(u, v, S)
6   (C; D) := M_1^{-1}(U; V)
7   Adjust_1(C, D, M_1)
8   (c, d) := T_k(C, D)
9   (e, f, M_2) := HG(c, d, S)
```

- Second recursive call on integers of size $n/2$.

Comments

1	$n := \ell(U)$
2	`If` $n \leq S$ `then return` $\widehat{\mathcal{E}}_{1/2}(U, V)$
3	$m := \lfloor n/2 \rfloor; k = n - m$
4	$(u, v) := T_m(U, V)$
5	$(c, d, \mathcal{M}_1) := \mathcal{HG}(u, v, S)$
6	$(C; D) := \mathcal{M}_1^{-1}(U; V)$
7	$\mathtt{Adjust}_1(C, D, \mathcal{M}_1)$
8	$(c, d) := T_k(C, D)$
9	$(e, f, M_2) := \mathcal{HG}(c, d, S)$
10	$(E; F) := \mathcal{M}_2^{-1}(C; D)$
11	$\mathtt{Adjust}_2(E, F, \mathcal{M}_2)$

- Multiplication matrix $\times$ vector.
- The pair (C, D) is of size $n/2$.
- Function $\mathtt{Adjust}_2$: to ensure that Jebelean criterion is satisfied, eventually undo some divisions

Comments

1	$n := \ell(U)$
2	`If` $n \leq S$ `then return` $\widehat{\mathcal{E}}_{1/2}(U, V)$
3	$m := \lfloor n/2 \rfloor; k = n - m$
4	$(u, v) := T_m(U, V)$
5	$(c, d, \mathcal{M}_1) := \mathcal{HG}(u, v, S)$
6	$(C; D) := \mathcal{M}_1^{-1}(U; V)$
7	$\mathtt{Adjust}_1(C, D, \mathcal{M}_1)$
8	$(c, d) := T_k(C, D)$
9	$(e, f, M_2) := \mathcal{HG}(c, d, S)$
10	$(E; F) := \mathcal{M}_2^{-1}(C; D)$
11	$\mathtt{Adjust}_2(E, F, \mathcal{M}_2)$
12	`Return` $(E, F, \mathcal{M} := \mathcal{M}_1 \cdot \mathcal{M}_2)$

- Last multiplication between two matrices of size $n/2$

Complexity of the algorithm

Divide and conquer equation

$$B(n) = 2B(\frac{n}{2}) + 28\mu(\frac{n}{4}) + O(n)$$

where $\mu(n)$ is th cot of the division of two n bits integers

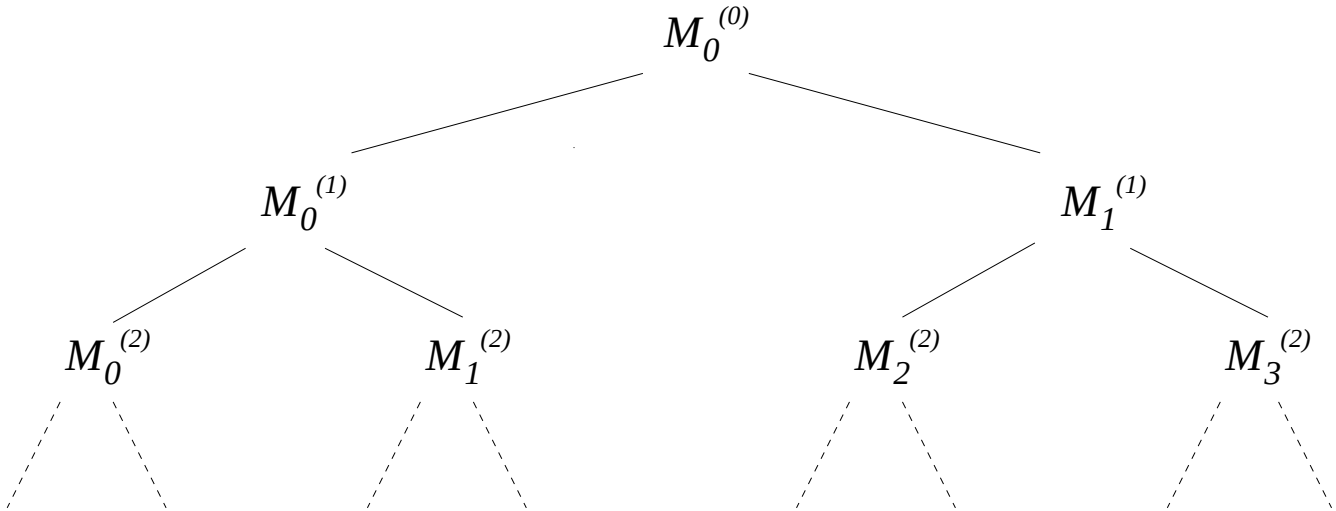
- Rapid multiplication $\mu(n) = O(n \log n \log\log n)$:

$$B(n) = O(n \log^2 n \log\log n)$$

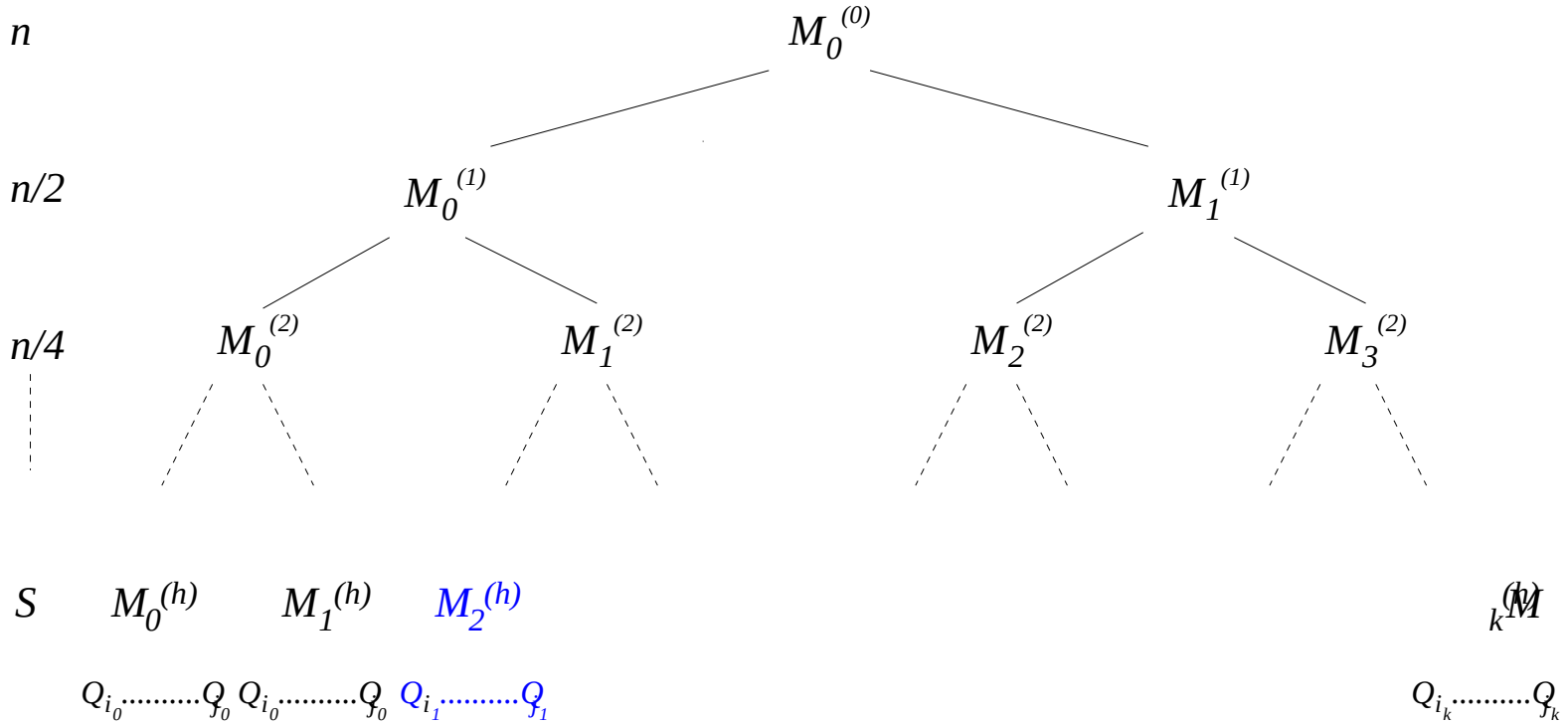
- Multiplications in $\mu(n) = O(n^\alpha)$, $1 < \alpha < 2$ (Karatsuba $\alpha \sim 1.6$, Toom-Cook $\alpha = 1.465$)

$$B(n) = O(n^\alpha)$$

An execution of the algorithm



An execution of the algorithm



Leaves of the tree: interrupted algorithms on integers of size S

Modification of the algorithm

Modification of the treshold S: one can stop the recursion sooner without loosing the complexity :

- one has to balance the costs of leaves and nodes

Treshlod depending on th multiplication:

- $\mu(n) = O(n^\alpha)$: treshold S of the form

$$S(n) = n^{\alpha-1}$$

- $\mu(n) = O(n \log n \log\log n)$: treshold S of the form

$$S(n) = \log n \log\log n$$

The $\mathcal{HG}_\alpha$

Algorithm $\mathcal{HG}(U, V)$
  $n := \ell(U)$
  $S := S_M(n)$
  Return $\mathcal{HG}(U, V, S)$

Algorithm $\mathcal{HG}(U, V, S)$

1 $n := \ell(U)$
2 If $n \leq S$ then return $\widehat{\mathcal{E}}_{1/2}(U, V)$
3 $m := \lfloor n/2 \rfloor; k = n - m$
4 $(u, v) := T_m(U, V)$
5 $(c, d, \mathcal{M}_1) := \mathcal{HG}(u, v, S)$
6 $(C; D) := \mathcal{M}_1^{-1}(U, V)$
7 $\mathtt{Adjust}_1(C, D, \mathcal{M}_1)$
8 $(c, d) := T_k(C, D)$
9 $(e, f, M_2) := \mathcal{HG}(c, d, S)$
10 $(E; F) := \mathcal{M}_2^{-1}(C; D)$
11 $\mathtt{Adjust}_2(E, F, \mathcal{M}_2)$
12 Return $(E, F, \mathcal{M} := \mathcal{M}_1 \cdot \mathcal{M}_2)$

The algorithm $\widehat{\mathcal{E}}_{[\gamma,\delta]}$

- (u, v) une entrée de l'algorithme d'Euclide, $\ell(u) = n$.
- $u_0, u_1, u_2 \ldots$ la suite de restes associée
- $Q_1, Q_2, Q_3 \ldots$ la suite de quotients correspondante

L'algorithme $\widehat{\mathcal{E}}_{[\gamma,\delta]}$ renvoie la matrice

$$\mathcal{M} = Q_i \cdots Q_{j-2}$$

où i et j sont les premiers indices tels que :

- $\ell(u_i) < n - \gamma \cdot n$
- $\ell(u_j) < \ell(u_i) - \delta \cdot n$

Algorithme $\mathcal{HG}$ et algorithmes interrompus

Chaque matrice apparaissant au cours de l'algorithme s'exprime en terme d'algorithme d'Euclide interrompu à partir de l'entrée initiale (U, V).

Proposition

- *Soit (U, V) une entrée de l'algorithme $\mathcal{HG}$*
- *Soit $\mathcal{M}_j^{(i)}$ la matrice renvoyée au jème noeud du niveau i de l'arbre des appels récursifs.*
- *Cette matrice est également renvoyée par l'algorithme $\widehat{\mathcal{E}}_{[\gamma,\delta]}$ sur entrée (U, V) avec*

$$\gamma := j/2^{i+1}, \quad \delta := 1/2^{i+1}.$$

Aux feuilles...

Les paramètres $\gamma(n)$, $\delta(n)$ aux feuilles de l'arbre dépendent de la multiplication utilisée :

- Multiplication en n^α :

$$\delta = n^{\alpha-2}$$

- Multiplication en $n \log n \log\log n$

$$\delta \sim \frac{1}{n}$$

Comportement des algorithmes interrompus

Théorème *Soient $\delta(n), \gamma(n)$ des paramètres rationnels avec un dénominateur D qui vérifie $D = n^{1/2}/b(n)$, avec $\lim_n b(n) = +\infty$. Alors*

- $\mathbb{P}_n \left[\left| P_{[\gamma,\delta]} - \lfloor \delta P \rfloor \right| > (b(n) \cdot n)^{1/2} \right] = O(2^{-b(n)})$,
- $\mathbb{P}_n \left[\left| \ell^{\beta}_{[\gamma,\delta]} - (\delta n)^{\beta} \right| > (b(n) \cdot n)^{1/2} \cdot (\delta n)^{\beta-1} \right] = O(2^{-b(n)})$

Remarques :
- preuve qui fait appel à des résultats plus 'fins' que d'habitude : utilisation de la formule de Perron, étude des séries dans une bande verticale $|\Re(s) - 1| \leq \alpha$
- dans notre cadre, ce théorème s'applique pour une multiplication de la forme $\mu(n) = \Theta(n^{3/2+r})$

Pour l'algorithme $\mathcal{HG}$

Du théorème précédent et du lien algorithme $\mathcal{HG}$ algorithmes interrompus on déduit

Théorème

Soit le $j$ème noeud du $i$ème niveau de l'arbre des appels récursif de l'algorithme $\mathcal{HG}$. Si le niveau i satisfait $i < [(1/2) - r] \lg n$ pour $r > 0$, alors la taille $\ell_j^{(i)}$ de la matrice $\mathcal{M}_j^{(i)}$ vérifie pour tout $\beta > 1$

$$\mathbb{P}_n \left[\left| (\ell_j^{(i)})^\beta - \left(\frac{1}{2^{i+1}} n \right)^\beta \right| > n^{(1/2)+r} \cdot (\frac{1}{2^{i+1}} n)^{\beta-1} \right] = O(2^{-n^r}),$$

Retour vers $\mathcal{HG}$ I

Soit l'ensemble Ω_n des entrées de l'algorithme $\mathcal{HG}$. On peut séparer Ω_n en deux sous-ensembles

- un ensemble "ordinaire" : toutes les tailles $\ell_j^{(i)}$ vérifient

$$\left[\left|(\ell_j^{(i)})^\beta - \left(\frac{1}{2^{i+1}}n\right)^\beta\right| \leq \left(\frac{1}{2^{i+1}}n\right)^\beta \cdot \varepsilon_i\right], \quad \varepsilon_i = n^{r-1/2}2^{i+1}$$

- le complementaire

Le théorème précédent implique que la probabilité de l'ensemble "exceptionnel" est $O(2^{-n^{r/2}})$.
Cet ensemble est donc négligeable pour l'analyse en moyenne.

Retour vers $\mathcal{HG}$ II

Soit U, V une entrée de l'algorithme appartenant à l'ensemble ordinaire.

- on a $\left[\left|(\ell_j^{(i)})^\beta - \left(\frac{1}{2^{i+1}}n\right)^\beta\right| \leq \left(\frac{1}{2^{i+1}}n\right)^\beta \cdot \varepsilon_i\right]$ avec $\varepsilon_i = n^{r-1/2}2^{i+1}$
- si la multiplication satisfait $A_1 \cdot n^\alpha \leq \mu(n) \leq A_2 \cdot n^\alpha$ alors on a l'encadrement

$$28A_1 \sum_{i=0}^{h-1} 2^i \left(\frac{1}{2^{i+1}}n\right)^\alpha \cdot [1-\varepsilon_i] \leq K_\alpha(A,B)$$

$$K_\alpha(A,B) \leq 28A_2 \sum_{i=0}^{h-1} 2^i \left(\frac{1}{2^{i+1}}n\right)^\alpha [1+\varepsilon_i]$$

- on en déduit

$$A_1\,[1+O(n^{-r})] \leq \frac{\mathrm{E}_n[K_\alpha]}{7n^\alpha} \cdot \frac{2^{\alpha-1}-1}{2^{1-\alpha}} \leq A_2\,[1+O(n^{-r})].$$

1 Introduction
- Integer gcd algorithms
- State of the art

2 Dynamical analysis of euclidean algorithms
- General principle
- An example: Euclid algorithm

3 Application to accelerated algorithms
- The Knuth-Schönhage algorithm
- Interrupted algorithms

4 Analyse de l'algorithme LSB
- Extension continue
- Produits de matrices aléatoires

5 Conclusion

Principe de la division

Analogue pour les entiers à la division de séries formelles:

Utilisation de la valuation 2-adique :

$$\nu(u) = \max\{k, 2^k | u\}, \qquad |u|_2 = 2^{-\nu(u)}$$

Euclide (=MSB)	LSB
Entrée: $v > u$ Division: $v = uq + r$ $u > r \geq 0$	Entrée: $\lvert v\rvert_2 > \lvert u\rvert_2$ Division: $v = uq + r$ $\lvert u\rvert_2 > \lvert r\rvert_2 > 0$
$29 = 2 \times 12 + 5$	$29 = \frac{-1}{4} \times 12 + 32$

Le but est de faire apparaître des 0 à la droite des entiers.

L'algorithme LSB

La division LSB

- de la forme $u = vq + r, \quad q = \dfrac{a}{2^k}, \quad k \geq 1, \quad a$ impair,
$$|a| < 2^k, \quad (v', r') = 2^k (v, r)$$
- forme matricielle $\begin{pmatrix} v \\ u \end{pmatrix} = \begin{pmatrix} 0 & 2^k \\ 2^k & a \end{pmatrix} \cdot \begin{pmatrix} r' \\ v' \end{pmatrix}$

L'algorithme LSB :

- une suite de divisions

$$\begin{pmatrix} v \\ u \end{pmatrix} = \begin{pmatrix} 0 & 2^{k_1} \\ 2^{k_1} & a_1 \end{pmatrix} \cdots\cdots \begin{pmatrix} 0 & 2^{k_p} \\ 2^{k_p} & a_p \end{pmatrix} \cdot \begin{pmatrix} 0 \\ d \end{pmatrix}$$

avec $\mathrm{pgcd}(u, v) = d$.

i	u_i [base 2]	u_i[base 10]	$a_i/2^{k_i}$
0	1000110010100001	72001	
1	11110101100000010100 0	2011176	-3 / 8
2	11001001101101010000	826192	1 / 2
3	11000011000101000000 0	1598080	1 / 8
4	10011000111100000000	626432	-1 / 2
5	11101001010100000000 0	1911296	-1 / 2
6	11000001001000000000 0	1582080	1 / 2
7	10001000110000000000 0	1120256	-1 / 2
8	10000101100000000000 0	2142208	1 / 2
9	11000000000000000	49152	1 / 4
10	10000010000000000000 0	2129920	-1 / 2
11	10001000000000000000	1114112	1 / 2
12	11000000000000000000	1572864	-5 / 8
13	10000000000000000000	2097152	3 / 4

$$\mathbb{E}_N[P] \sim \frac{1}{2-\gamma_0} \cdot N$$

Extension continue

Extension réelle de $u = vq + r$:

- chaque quotient q définit les applications

$$T_q(x) = \frac{1}{x} - q, \qquad h_q(x) = \frac{1}{q + x}$$

- les rationnels $\frac{v}{u}$ n'appartiennent pas à un intervalle borné
- on préfère travailler sur un ensemble compact que sur $\mathbb{R}$

"Bonne" extension :

- la droite projective réelle, isomorphe au tore

$$J := \left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$$

  muni de la topologie projective
- conjugaison de T_q, h_q via l'application tangente :

$$\ell_q(x) = \arctan \circ h_q \circ \tan(x) = \arctan\left(\frac{1}{q + \tan x}\right)$$

Propriétés génératrices

Soit une entrée de l'algorithme

- $\begin{pmatrix} v \\ u \end{pmatrix} = M_{q_1} \cdots M_{q_p} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} := M_q \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Alors :

- $\frac{v}{u} = \arctan \ell_{q_1} \circ \cdots \circ \ell_{q_p}(0) := \arctan \ell_q(0)$
- $|\det M_q|^{-1} \cdot |\ell_q'(0)| = \frac{||(0,1)||^2}{||M_q(0,1)||^2} = \frac{1}{u^2+v^2}$

Système de fonctions itérées

Soit le système suivant :

- $\mathcal{L} := \left\{ \ell_q,\ \ell_q(y) = \arctan\left(\frac{1}{q+\tan y}\right),\quad q = \frac{a}{2^k} \right\}$
- ℓ_q est choisie avec prob. $\delta_q := |\det M_q|^{-1} = 2^{-2k}$

Système de fonctions itérées

Soit le système suivant :

- $\mathcal{L} := \left\{ \ell_q,\ \ell_q(y) = \arctan\left(\frac{1}{q+\tan y}\right),\quad q = \frac{a}{2^k} \right\}$
- ℓ_q est choisie avec prob. $\delta_q := |\det M_q|^{-1} = 2^{-2k}$

L'opérateur associé à ce système est

$$\sum_{\ell_q \in \mathcal{L}} \delta_q \cdot |\ell_q'(x)| \cdot f \circ \ell_q(x) = \mathbf{H}[f](x)$$

Système de fonctions itérées

Soit le système suivant :

- $\mathcal{L} := \left\{ \ell_q,\ \ell_q(y) = \arctan\left(\frac{1}{q+\tan y}\right),\quad q = \frac{a}{2^k} \right\}$
- ℓ_q est choisie avec prob. $\delta_q := |\det M_q|^{-1} = 2^{-2k}$

L'opérateur associé à ce système est

$$\sum_{\ell_q \in \mathcal{L}} \delta_q \cdot |\ell_q'(x)| \cdot f \circ \ell_q(x) = \mathbf{H}[f](x)$$

L'opérateur de transfert est

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

Système de fonctions itérées

Soit le système suivant :

- $\mathcal{L} := \left\{ \ell_q,\ \ell_q(y) = \arctan\left(\frac{1}{q+\tan y}\right),\quad q = \frac{a}{2^k} \right\}$
- ℓ_q est choisie avec prob. $\delta_q := |\det M_q|^{-1} = 2^{-2k}$

L'opérateur associé à ce système est

$$\sum_{\ell_q \in \mathcal{L}} \delta_q \cdot |\ell_q'(x)| \cdot f \circ \ell_q(x) = \mathbf{H}[f](x)$$

L'opérateur de transfert est

$$\sum_{\ell_q \in \mathcal{L}^n} {\delta_q}^s \cdot |\ell_q'(x)|^s \cdot f \circ \ell_q(x) = {\mathbf{H}_s}^n[f](x)$$

Séries génératrices et opérateurs

Coût quotient :

- soit un coût c tq $\sum_q \delta_q c(q) < \infty$
- série génératrice : $F_C(s) := \sum_{(u,v)\in\Omega} \frac{C(u,v)}{(u^2+v^2)^s}$

Opérateur $\mathbf{H}_s^{[c]}$:

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

Lien SG opérateurs

$$F_C(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{[c]} \circ (I - \mathbf{H}_s)^{-1}[1](0)$$

Séries génératrices et opérateurs

Complexité en bits

- coût de $u = vq + r$: $\ell_2(v) \times [k(q) + s(q)]$
- série génératrice $F_B(s)$

Dérivation : $\Delta := \dfrac{1}{\log 2} \dfrac{-d}{ds}$

- $\Delta \mathbf{H}_s[f](x) = \sum_{\ell_q \in \mathcal{L}} \delta_q^s \cdot |\ell_q'(x)|^s \cdot \log_2(\delta_q \cdot |\ell_q'(x)|) \cdot f \circ \ell_q(x)$

Lien SG opérateurs $F_B(s) =$

$$(I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{[k+s]} \circ (I - \mathbf{H}_s)^{-1} \circ \Delta \mathbf{H}_s \circ (I - \mathbf{H}_s)^{-1}[1](0)$$

Produits de matrices aléatoires

Soit l'ensemble de matrices :

$$\mathcal{N} := \left\{ N_q = \begin{pmatrix} 0 & 1 \\ 1 & q \end{pmatrix} ; q = \frac{a}{2^k} ; k \geq 1, a \text{ impair} \in]-2^k, 2^k[\right\},$$

- N_q est choisie avec prob. $\delta_q = 2^{-2k}$
- exposant de Lyapunov :

$$\gamma = \lim_{n \to \infty} \frac{1}{n} \mathbb{E} \left[\log ||N_1 \cdot N_2 \cdot \ldots \cdot N_n|| \right]$$

- Exposant de Lyapunov binaire : $\gamma_0 := \gamma / \log 2$

Opérateur associé

Opérateur associé à cet ensemble :

$$\mathbf{H}_{t,s}[f](x) = \sum_{\ell_q \in \mathcal{L}} {\delta_q}^t \cdot |\ell_q'(x)|^s \cdot f \circ \ell_q(x), \qquad (\mathbf{H}_s = \mathbf{H}_{s,s})$$

Résultats classiques autour de $(t, s) = (1, 0)$ ([Fur63,GR85,LP82,BL85])

- opérateur quasi-compact sur l'espace $\mathbf{H}$ des fonctions Hölder
- saut spectral + perturbation : $\lambda(s, t)$ *vp* dominante isolée
- exposant de Lyapunov existe, donné par

$$\lambda_s'(1, 0) = \gamma$$

Retour vers $(t, s) = (1, 1)$

Relations entre $\mathbf{H}_{1,0}$ et $\mathbf{H}_{1,1}$:

- (relation intégrale) F une primitive de f:

$$\int_a^x \mathbf{H}_{1,1}^p[f](t)dt = \mathbf{H}_{1,0}^p[F](x) - \mathbf{H}_{1,0}^p[F](a)$$

Retour vers $(t, s) = (1, 1)$

Relations entre $\mathbf{H}_{1,0}$ et $\mathbf{H}_{1,1}$:

- (relation intégrale) F une primitive de f:

$$\int_a^x \mathbf{H}_{1,1}^p[f](t)dt = \mathbf{H}_{1,0}^p[F](x) - \mathbf{H}_{1,0}^p[F](a)$$

- (relation de dualité) $\mathbf{H}_{1-s,t}$= dual de $\mathbf{H}_{s,t}$, d'où

$$\lambda(1 - s, t) = \lambda(s, t)$$

Finalement,

Les opérateurs satisfont les conditions requises

Application du théorème taubérien :

- pôle double pour $F_C(s)$

$$F_C(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{[c]} \circ (I - \mathbf{H}_s)^{-1}[1](0)$$

- comportement linéaire pour un coût C

Finalement,

Les opérateurs satisfont les conditions requises

Application du théorème taubérien :

- pôle double pour $F_C(s)$

$$F_C(s) = (I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{[c]} \circ (I - \mathbf{H}_s)^{-1}[1](0)$$

- comportement linéaire pour un coût C
- pôle triple pour $F_B(s) =$

$$(I - \mathbf{H}_s)^{-1} \circ \mathbf{H}_s^{[k+s]} \circ (I - \mathbf{H}_s)^{-1} \circ \Delta\mathbf{H}_s \circ (I - \mathbf{H}_s)^{-1}[1](0)$$

- comportement quadratique pour la complexité en bits B

Constantes

Constantes issues de l'application $s \mapsto \lambda(s,s)$:

- $\lambda(1-s,t) = \lambda(s,t)$

$$\lambda'_s(s,s)|_{s=1} = \lambda'_t(1,1) - \lambda'_s(1,1) = 2\log 2 - \gamma = \frac{1}{\log 2}(2-\gamma_0)$$

Autres constantes :

- décomposition spectrale en $s=(1,1)$ fait apparaître l'intégrale

$$\int_J H_{1,1}^{[c]}[f](t)dt = \mu(c)$$

Conclusion :

$$\mathbb{E}_N[C] \sim \frac{1}{2-\gamma_0} \cdot \mu(c) \cdot N$$

$$\mathbb{E}_N[B] \sim \frac{1}{2-\gamma_0} \cdot \mu(k+s) \cdot \frac{N^2}{2}$$

Conclusion

Analyse dynamique :

- méthode robuste puisqu'elle permet de traiter en profondeur plusieurs types de division
- permet l'étude précise d'algorithmes à la structure complexe
- lien étroit avec la théorie des systèmes dynamiques

Perspectives

- analyse en distribution de LSB
- analyse de LSB diviser pour régner
- passage à des dimensions supérieures (réductions des réseaux)