M2AN. MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
- MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE

JEAN-MICHEL MULLER

Une méthodologie du calcul hardware des fonctions élémentaires

M2AN. Mathematical modelling and numerical analysis - Modélisation mathématique et analyse numérique, tome 20, n° 4 (1986), p. 667-695

<http://www.numdam.org/item?id=M2AN_1986__20_4_667_0>

MATHEMATICAL MODELLING AND NUMERICAL ANALYSIS
MODÉLISATION MATHÉMATIQUE ET ANALYSE NUMÉRIQUE

(Vol 20, n° 4, 1986, p 667 à 695)

UNE MÉTHODOLOGIE DU CALCUL HARDWARE DES FONCTIONS ÉLÉMENTAIRES (*)

par Jean-Michel MULLER (1)

Communiqué par F ROBERT

Resumé — *Dans cet article de synthèse, nous présentons en détail la notion de « base discrète », qui nous permet d'élaborer des algorithmes de calcul des fonctions mathématiques usuelles se prêtant particulièrement bien à une réalisation* câblee *Nous exhibons ainsi toute une classe d'algorithmes simples et efficaces, qui inclut des methodes bien connues aussi bien que de nouvelles méthodes*

Abstract — *In this survey paper, we examine thoroughly the notion of « discrete basis », which enables us to build some efficient* hardware *algorithms for computing the most usual mathematical functions We then present a class of algorithms including some well-known methods, and some new ones*

I. INTRODUCTION

Nous nous proposons ici d'étudier certains algorithmes de calcul des fonctions élémentaires au moyen d'un outil mathématique nouveau : les *bases discrètes.* Certains algorithmes de ce style sont anciens : il y a 300 ans, Briggs avait déjà mis au point des algorithmes de calcul de l'exponentielle et du logarithme qui, en base 10, coincident avec les exemples de la partie III.

Nous cherchons à élaborer des algorithmes *hardware*, qui par conséquent ne peuvent se baser que sur des primitives simples (contrairement à ce qui se passe lorsqu'on écrit des programmes en langage évolué, où l'on peut se permettre l'appel de fonctions aussi complexes que la division flottante). Dans l'état actuel des connaissances en matériel, il paraît donc logique de n'utiliser que les deux primitives suivantes :

— L'addition/soustraction.

— La multiplication *par une puissance de la base de l'arithmétique du système*, qui se réduit à un décalage des chiffres si l'on travaille en virgule fixe, et à une addition/soustraction à l'exposant en virgule flottante. Par abus de langage,

(*) Reçu en septembre 1985
(1) Laboratoire TIM3, Institut IMAG, BP 68, 38402 Saint-Martin d'Hères Cedex, France

M² AN Modelisation mathematique et Analyse numerique 0399-0516/86/04/667/29/$ 4 90
Mathematical Modelling and Numerical Analysis

on qualifiera cette opération de « décalage » même si l'on travaille sur un système à virgule flottante.

Il est alors évident que les algorithmes classiques utilisés en machine pour le calcul des fonctions standard, et qui utilisent essentiellement des approximations polynomiales ou rationnelles, deviennent caducs. Il convient donc d'exhiber une classe d'algorithmes n'utilisant que des additions et des décalages.

La première tentative importante dans cette voie date de 1958, et est due à J. Volder : c'est l'algorithme CORDIC (COordinate Rotation on a DIgital Computer), qui permet d'effectuer des multiplications, des divisions et de calculer les principales fonctions trigonométriques. Ensuite, en 1971, J. Walther montra que CORDIC pouvait être étendu au calcul des fonctions hyperboliques.

L'idée de base de CORDIC, pour calculer une fonction f sur un système travaillant en base 2 est d'approximer l'argument t par la somme :

$$t \sim d_0 e_0 + d_1 e_1 + \cdots + d_n e_n, \qquad d_i = \pm 1 .$$

où les e_i sont des constantes précalculées, choisies telles que $f(d_0 e_0 + \cdots + d_{n+1} e_{n+1})$ puisse être calculé à partir de $f(d_0 e_0 + \cdots + d_n e_n)$ en n'utilisant que des additions et des décalages. Notre but est d'étendre cette idée à des bases différentes de 2 et à d'autres valeurs de d_i. Il convient donc d'étudier le problème sous deux aspects :

— La recherche des suites (e_n) telles que tout élément x d'un domaine donné puisse s'écrire sous la forme :

$$x = d_0 e_0 + \cdots + d_n e_n + \cdots$$

où les d_i seront compris entre 0 et un entier naturel p (on se ramène au cas de CORDIC, où l'on désire des d_i négatifs par plusieurs soustractions de la quantité $\sum_{i=0}^{\infty} e_i$). C'est ce qui sera fait dans la partie II de cette étude.

— La connaissance d'algorithmes permettant de calculer les termes d_i correspondant à un x donné, et l'utilisation de ces algorithmes au calcul de certaines fonctions, ce qui fera l'objet de la partie III, où l'on présentera deux exemples d'application de cette étude : le calcul du logarithme et de l'exponentielle.

II. BASES DISCRÈTES

II.A. Bases discrètes additives

II.A.1. *Premières définitions*

Nous noterons S le cône des suites réelles, strictement positives, décroissantes et sommables $\left((e_n) \in S \Rightarrow \sum_{n=0}^{\infty} e_n < +\infty\right)$.

DÉFINITION 1 : *Soit* $E = (e_n)$ *un élément de* S, *on appellera* Ensemble généré d'ordre p de E *l'ensemble* :

$$G_p(E) = \left\{ \sum_{n=0}^{\infty} d_n e_n / d_n \in \{ 0, 1, ..., p \} \right\}.$$

Soit $x \in G_p(E)$, si x peut s'écrire $\sum_{n=0}^{\infty} d_n e_n$ on dira que la suite (d_n) est un *système de coordonnées* (non nécessairement unique) de x sur E.

PROPRIÉTÉ 1 : *Pour tout élément* E *de* S, $G_p(E)$ *a la puissance du continu, et n'a pas de points isolés.*

Démonstration : a) $G_p(E)$ *n'a pas de points isolés.*

Posons $E = (e_n)$.

Soit $x \in G_p(E)$, $x = \sum_{n=0}^{\infty} d_n e_n$.

Montrons que pour tout $\varepsilon > 0$, il existe $y \in G_p(E)$, $y \neq x$, tel que $| x - y | \leqslant \varepsilon$.

E est sommable, donc $\lim_{n \to \infty} e_n = 0$, et par conséquent, pour tout $\varepsilon > 0$, il existe N tel que :

$$n \geqslant N \Rightarrow 0 \leqslant e_n \leqslant \varepsilon / p .$$

Soit y défini par $y = \sum_{n=0}^{\infty} d'_n e_n$, où :

$$\begin{cases} d'_n = d_n \text{ si } n \neq N . \\ - d'_N = 0 \text{ si } d_N \neq 0 , \quad d'_N = 1 \text{ sinon} . \end{cases}$$

Nous avons $| y - x | = | d'_N - d_N | e_n$

soit $\qquad 0 < | y - x | \leqslant p e_n$

soit $\qquad 0 < | y - x | \leqslant \varepsilon .$

Ce qu'il fallait démontrer.

b) $G_p(E)$ *a la puissance du continu.*

Étant donnée l'inclusion évidente $G_q(E) \subset G_p(E)$ pour tout $q \leqslant p$, il suffit de montrer que $G_1(E)$ a la puissance du continu. Pour ceci, remarquons que si $E' = (e'_n)$ est une suite extraite de $E = (e_n)$, alors $G_1(E') \in G_1(E)$.

Construisons une suite (e'_n) comme suit :

$e'_0 = 0$.

$\lim_{n \to \infty} e_n = 0$ donc il existe N tel que $e_N < e_0/2$.

Posons $e'_1 = e_N$.

De même, il existe $M > N$ tel que $e_M < e'_1/2$, et nous choisirons $e'_2 = e_M$.

En itérant ce processus, nous construisons une suite $E' = (e'_n)$ extraite de (e_n) telle que pour tout n,

$$e'_{n+1} < e'_n/2 \,. \tag{1}$$

Montrons que $G_1(E')$ a la puissance du continu.

Pour ceci, établissons que l'application de $\{ 0, 1 \}^{\mathbb{N}}$ vers $G_1(E')$ qui à (d_n) fait correspondre $\sum_{n=0}^{\infty} d_n e'_n$ est injective.

Soit $x = \sum_{n=0}^{\infty} d_n e'_n$. Supposons qu'il existe (d'_n) différente de (d_n) et telle que $x = \sum_{n=0}^{\infty} d'_n e'_n$.

En notant k le plus petit entier vérifiant $d_k \neq d'_k$, il vient :

$$(d_k - d'_k) e'_k = - \sum_{n=k+1}^{\infty} (d_n - d'_n) e'_n \,.$$

Soit :

$$e'_k = \left| \sum_{n=k+1}^{\infty} (d_n - d'_n) e'_n \right| \leqslant \sum_{n=k+1}^{\infty} \left| (d_n - d'_n) e'_n \right|$$

$$\leqslant \sum_{n=k+1}^{\infty} e'_n \,. \tag{2}$$

Or, (1) implique que pour tout $n > k$, $e'_n < 2^{-(n-k)} e'_k$.

Par conséquent $\sum_{n=k+1}^{\infty} e'_n < e'_k \sum_{n=k+1}^{\infty} 2^{-(n-k)} = e'_k$.

Ce qui contredit (2).

D'où le résultat.

PROPRIÉTÉ 2 (unicité d'écriture) : *Si pour tout entier naturel n,* $e_n > p \sum_{k=n+1}^{\infty} e_k$, *alors tout élément de* $G_p(E)$ *admet un système de coordonnées unique sur E.*

Démonstration : Si x s'écrit $x = \sum_{\iota=0}^{\infty} d_\iota e_\iota = \sum_{\iota=0}^{\infty} d'_\iota e_\iota(d_\iota) \neq (d'_\iota)$, alors, en notant r le plus petit entier tel que $d_r \neq d'_r$, il vient :

$$(d_r - d'_r) e_r = \sum_{\iota=r+1}^{\infty} (d'_\iota - d_\iota) e_\iota .$$

Et par conséquent :

$$e_r \leqslant |(d_r - d'_r) e_r| \leqslant \sum_{\iota=r+1}^{\infty} |d_\iota - d'_\iota| e_\iota$$
$$\leqslant p \sum_{\iota=r+1}^{\infty} e_\iota .$$

Ce qui contredit l'hypothèse.

Remarque : Si l'on a seulement $e_n \geqslant p \sum_{\iota=n+1}^{\infty} e_\iota$, alors, dans la démonstration précédente, il ne peut y avoir égalité entre $\sum_{\iota=0}^{\infty} d_\iota e_\iota$ et $\sum_{\iota=0}^{\infty} d'_\iota e_\iota$ que si :

$$d_r - d'_r = 1 \quad \text{et} \quad \forall n > r, \quad d_n = 0 \quad \text{et} \quad d'_n = p$$

ou :

$$d_r - d'_r = -1 \quad \text{et} \quad \forall n > r, \quad d_n = p \quad \text{et} \quad d'_n = 0 .$$

On en déduit de plus que dans ce cas, un élément de $G_p(E)$ admet au plus deux systèmes de coordonnées sur E. Ceci est à rapprocher de la convention 0.99999999... = 1 de l'écriture décimale des nombres.

II.A.2. *Bases discrètes additives : Définition et propriétés*

DÉFINITION 2 : *Soit E un élément de S. On dira que E est une* base discrète (additive) d'ordre p si $G_p(E)$ *est un intervalle. C'est alors l'intervalle* $\left[0, p \sum_{n=0}^{\infty} e_n\right]$.

THÉORÈME 1 : $E = (e_n)$ *est une base discrète d'ordre p si et seulement si pour tout entier n,*

$$e_n \leqslant p \sum_{k=n+1}^{\infty} e_k , \qquad \text{(A)}$$

Démonstration :

a) *La condition est nécessaire.*

Supposons qu'il existe $n \in \mathbb{N}$ tel que $e_n > p \sum_{k=n+1}^{\infty} e_k = r_n$, et posons $I =]r_n, e_n[$. Montrons que l'intersection de I avec $G_p(E)$ est vide.

Soit $a \in G_p(E)$, $a = \sum_{i=0}^{\infty} d_i e_i$.

1) S'il existe $k \leqslant n$ tel que $d_k \neq 0$, alors $a \geqslant d_k e_k \geqslant e_k \geqslant e_n$ puisque la suite (e_i) décroît.

2) Si pour tout $k \leqslant n$, $d_k = 0$, alors $a \leqslant p \sum_{k=n+1}^{\infty} e_k$. Par conséquent, dans tous les cas, $a \notin I$.

b) *La condition est suffisante.*

Pour établir ceci, montrons que si (A) est vérifiée, alors pour tout $a \in I = \left[0, p \sum_{n=0}^{\infty} e_n\right]$, la suite (d_n) définie comme suit nous assure : $\sum_{n=0}^{\infty} d_n e_n = a$.

$$\begin{cases} a_0 = 0 \\ d_i = \text{Max}\,\{ j \in \{0, 1, \ldots, p\} / a_i + j e_i \leqslant a \} \\ a_{i+1} = a_i + d_i e_i . \end{cases}$$

Il est clair que notre problème revient à démontrer que a est la limite de la suite a_n.

Dans ce but, établissons par récurrence le résultat suivant :

$$| a_n - a | \leqslant p \sum_{k=n}^{\infty} e_k . \qquad \text{(B)}$$

— Cette propriété est vraie si $n = 0$, puisque $a \in I$.

— Supposons qu'elle soit vraie pour un entier $n \geqslant 0$, et évaluons la quantité $| a_{n+1} - a |$.

1) Si $d_n < p$, alors, puisque d_n est égal à

$$\text{Max}\,\{ j \in \{0, \ldots, p\} / a_n + d_n e_n \leqslant a \},$$

nous avons :

$$a_{n+1} = a_n + d_n e_n \leqslant a \leqslant a_n + (d_n + 1) e_n$$

donc, par conséquent : $a_{n+1} \leqslant a < a_{n+1} + e_n$ donc, en utilisant (A),

$$| a_{n+1} - a | \leqslant p \sum_{k=n+1}^{\infty} e_k .$$

2) Si $d_n = p$, alors $a_{n+1} = a_n + pe_n \leqslant a$, donc $| a_{n+1} - a | = | a - a_n | - pe_n \leqslant p \sum_{k=n}^{\infty} e_k - pe_n$ (d'après l'hypothèse de récurrence)

$$\leqslant p \sum_{k=n+1}^{\infty} e_k .$$

Ce qu'il fallait démontrer.

Dans de nombreux cas, la condition (A) est difficile à vérifier directement. Nous donnons ci-après un théorème permettant de prouver plus simplement que certaines suites de S sont des bases discrètes.

THÉORÈME 2 : *Si $E = (e_n) \in S$ est une base discrète d'ordre p, si f est une fonction à variable réelle vérifiant* :

1) *f' est continue sur* $I = \left[0, p \sum_{n=0}^{\infty} e_n \right]$.

2) $f(0) = 0$.

3) *f est concave et strictement croissante sur I. Alors $f(E) = (f(e_n))$ est une base discrète d'ordre p.*

Démonstration :

a) *$(f(e_n))$ est une suite décroissante de réels strictement positifs.*

Ceci provient de la croissance stricte de f. Pour tout entier naturel n, $e_n \geqslant e_{n+1} > 0$, donc $f(e_n) \geqslant f(e_{n+1}) > f(0) = 0$.

b) $\sum_{n=0}^{\infty} f(e_n) < + \infty$.

Pour ceci, il suffit d'établir que pour tout entier naturel n :

$$f(e_n) \leqslant e_n f'(0) .$$

En effet, f étant concave et f' étant continue sur I, f' est donc décroissante sur I. Par conséquent :

$$f(e_n) = \int_0^{e_n} f'(x)\, dx \leqslant f'(0) \int_0^{e_n} dx = e_n f'(0) .$$

c) $(f(e_n))$ *vérifie* (A).

La concavité de f sur I implique que pour tout $(a_1, a_2, ..., a_n) \in I^n (a_i \geqslant 0)$ tel que $p(a_1 + a_2 + \cdots + a_n)$ est inclus dans I, nous avons :

$$f[p(a_1 + \cdots + a_n)] \leqslant p[f(a_1) + \cdots + f(a_n)] .$$

Par conséquent, f étant continue, si la série $p \sum_{i=0}^{\infty} a_i$ converge vers un élément de $I (a_i \in I)$, alors :

$$f\left(p \sum_{i=0}^{\infty} a_i\right) \leqslant p \sum_{i=0}^{\infty} f(a_i) .$$

Or, pour tout entier naturel n, nous avons $e_n \leqslant p \sum_{i=n+1}^{\infty} e_i$, par conséquent,

$$f(e_n) \leqslant f\left(p \sum_{i=n+1}^{\infty} e_i\right)$$

$$\leqslant p \sum_{i=n+1}^{\infty} f(e_i) .$$

Donc $(f(e_n))$ est une base discrète d'ordre p.

Exemples.

a) *Suites géométriques.*

Le théorème 1 nous permet de constater que la suite (e_n) définie par $e_n = Ka^{-n}$ est une base discrète d'ordre p si et seulement si $1 < a \leqslant p + 1$.

De telles bases, étudiées par A. Renyi ([47], [48]), constituent une première généralisation de la notion usuelle de bases de numération ($a \in \mathbb{N}$).

On peut, par exemple écrire « en base π » 1/2 sous la forme « 0.11211202... », ce qui signifie qu'il existe une suite (d_n) (non nécessairement unique), telle que $1/2 = \sum_{n=0}^{\infty} d_n \pi^{-n}$, et vérifiant $d_0 = 0, d_1 = 1, d_2 = 1, d_3 = 2$, etc...

Nous étudions ultérieurement des algorithmes permettant de calculer les termes d_i.

PROPRIÉTÉ 3 : (*Caractère* Fractal *de* $G_p(E)$). *Si E est une suite géométrique de raison strictement inférieure à* 1 *alors toute intersection non vide de $G_p(E)$ avec un intervalle ouvert contient un sous-ensemble homothétique à $G_p(E)$.*

En effet, posons $e_n = ca^{-n}$ soit $I =]\alpha, \beta[$ un intervalle ouvert d'intersection non vide avec $G_p(E)$, soit x un élément de cette intersection, définissons la quantité $\varepsilon = \min \{ x - \alpha, \beta - x \}$.

La suite $\sum_{k=n}^{\infty} e_k$ tend vers zéro lorsque n tend vers l'infini, par conséquent, il existe N tel que :

$$n \geqslant N \Rightarrow p \sum_{k=n}^{\infty} e_k < \varepsilon .$$

Si x s'écrit $\sum_{n=0}^{\infty} d_n e_n$, avec $d_n \in \{ 0, ..., p \}$, alors si l'on considère l'ensemble :

$$H_p^N(E) = \left\{ \sum_{n=0}^{\infty} d'_n e_n / d'_n = d_n \quad \text{si} \quad n \leqslant N, d'_n \in \{ 0, ..., p \} \text{ sinon} \right\}$$

on a l'inclusion évidente : $H_p^N(E) \in I$. Or on a manifestement $H_p^N(E) = f(G_p(E))$ où f est l'application :

$$y \to a^{-N} y + x_N$$

où

$$x_N = \sum_{i=0}^{N} d_i e_i$$

ce qu'il fallait démontrer.

PROPRIÉTÉ 4 . (*Séparation de $G_p(E)$ en $p+1$ branches translatées*). *Il existe un ensemble A_0 tel que $G_p(E) = A_0 \cup A_1 \cup \cdots \cup A_p$, où $A_i = A_0 + ie_0$.*

La démonstration est triviale si l'on pose $A_i = \left\{ \sum_{n=0}^{\infty} d_n e_n / d_0 = i \right\}$.

Les figures 1 à 3 donnent des exemples d'ensembles $G_p(E)$ pour E géométrique.

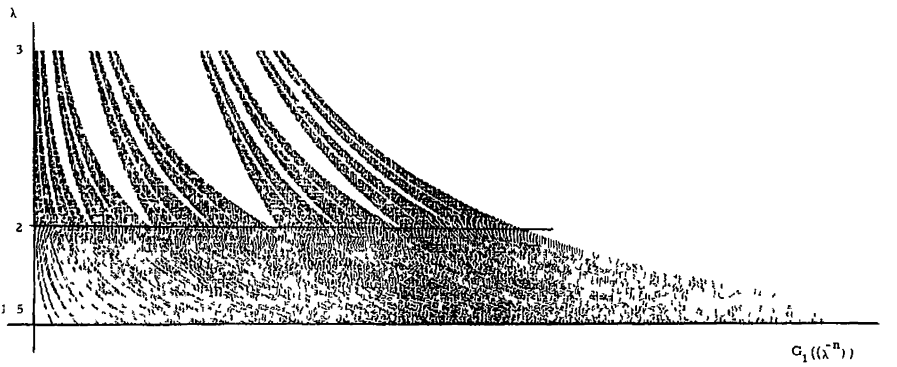

Fig 1 - Ensembles $G_1((\lambda^{-n}))$ pour differentes valeurs du paramètre λ On voit clairement que la valeur $\lambda = 2$ apparaît comme un point critique

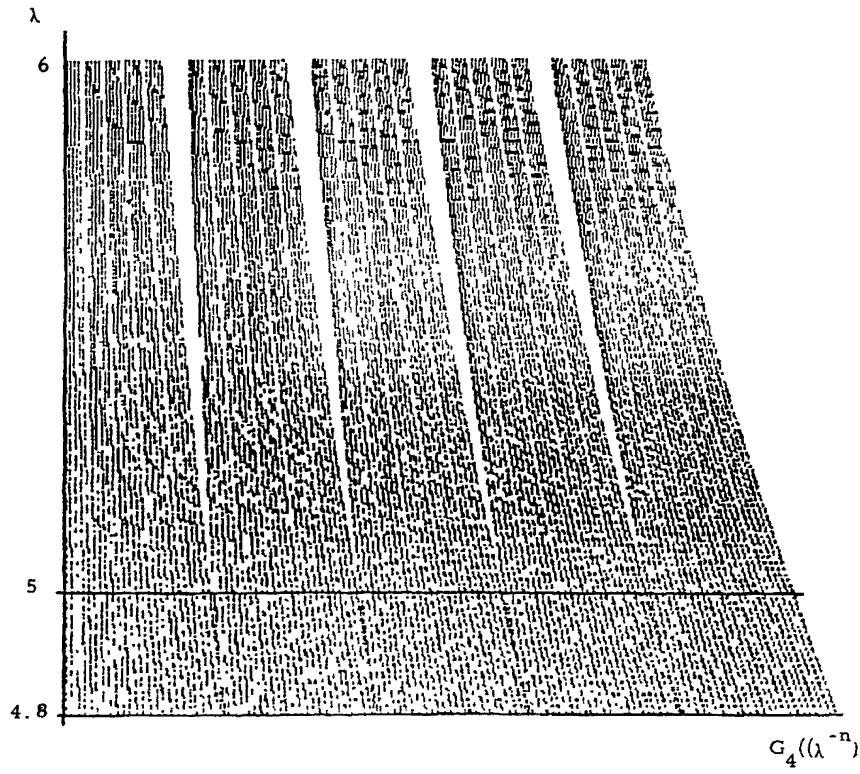

Fig. 2 - Tracé des ensembles $G_4((\lambda^{-n}))$, λ variant continûment entre 4.8 et 6.

Suites asymptotiquement géométriques.

Le théorème 2 nous prouve que les suites :

$$e_n = \text{Arctg}\,(a^{-n})$$
$$e'_n = \text{Log}\,(1 + a^{-n})\,.$$

Sont des bases discrètes d'ordre p si et seulement si $1 < a \leqslant p + 1$.

Par exemple, en « base $(\text{Log}\,(1 + \pi^{-n}))$ », Log (2) peut s'écrire « 1.0000... », et 2 peut s'écrire « 2.2012202... ».

On peut qualifier ces suites d'« asymptotiquement géométriques » puisqu'elles sont toutes deux équivalentes à a^{-n}.

Les figures 4 et 5 donnent des exemples d'ensembles $G_p(E)$ pour E asymptotiquement géométrique. On constatera la similitude (prévisible) avec le cas où E est géométrique. Les ensembles obtenus dans ces exemples sont soit des

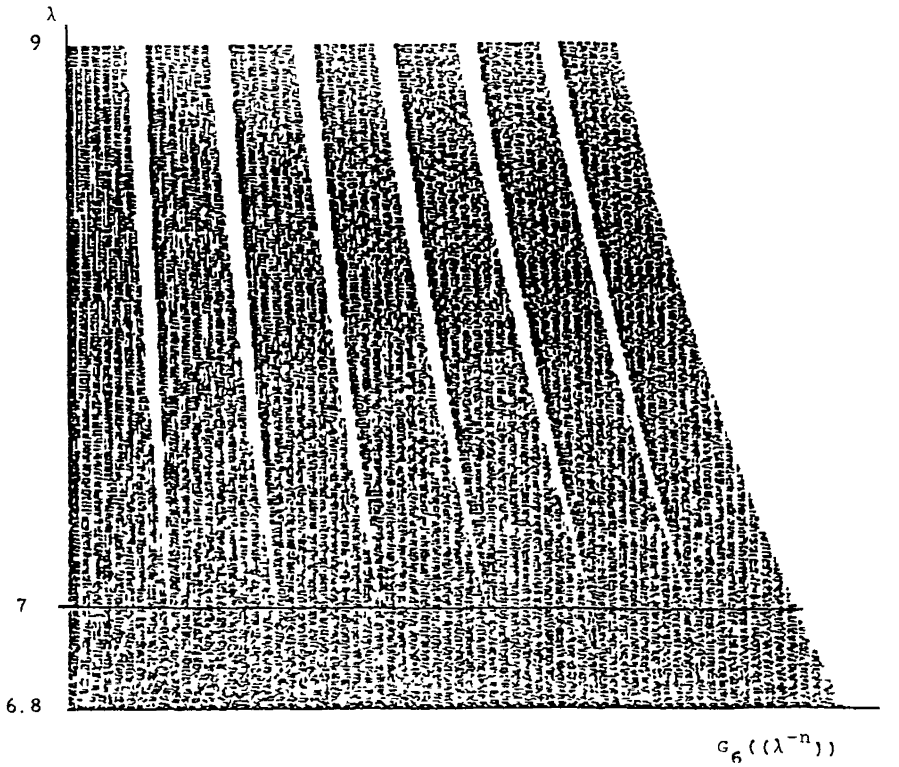

Fig. 3 - Tracé des ensembles $G_6((\lambda^{-n}))$, λ variant continûment entre 6.8 et 9

intervalles, soit, comme nous le verrons ultérieurement, des *ensembles parfaits* ([31]).

On a donc caractérisé les éléments E de S tels que $G_p(E)$ est un intervalle. On est alors en droit de chercher à quelles conditions $G_p(E)$ *contient* un intervalle I, car nous verrons que cette propriété pourrait nous permettre de calculer des fonctions sur I. Le théorème suivant répond partiellement à cette interrogation.

THÉORÈME 3 : *Si la suite* $E = (e_k)$ *vérifie* :

$$\forall n \in \mathbb{N}, \quad e_n > p \sum_{k=n+1}^{\infty} e_k \tag{C}$$

Alors $G_p(E)$ *ne contient aucun intervalle.*

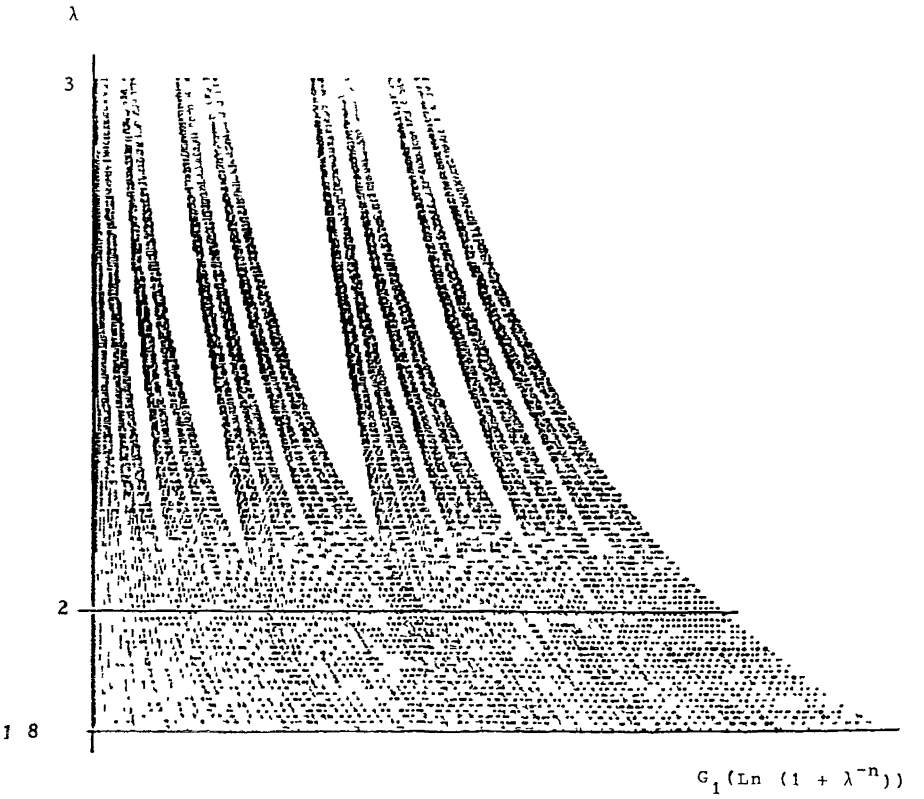

Fig 4. - Tracé des ensembles $G_1(\mathrm{Ln}(1+\lambda^{-n}))$, λ variant continûment de 1.8 à 3

Démonstration : Soit a un élément quelconque de $G_p(E)$.

Soit ε un réel strictement positif.

Montrons qu'il existe un réel b n'appartenant pas à $G_p(E)$ tel que $|b-a| < \varepsilon$.

Posons $a = \sum_{n=0}^{\infty} d_n e_n$, $d_n \in \{0, 1, ..., p\}$.

La série (e_n) étant convergente, nous avons $\lim_{n\to\infty} e_n = 0$, par conséquent il existe $n \in \mathbb{N}$ tel que pour tout $n > N$, on ait $pe_n < \varepsilon$.

Posons $c = \sum_{n=0}^{\infty} d'_n e_n$, où

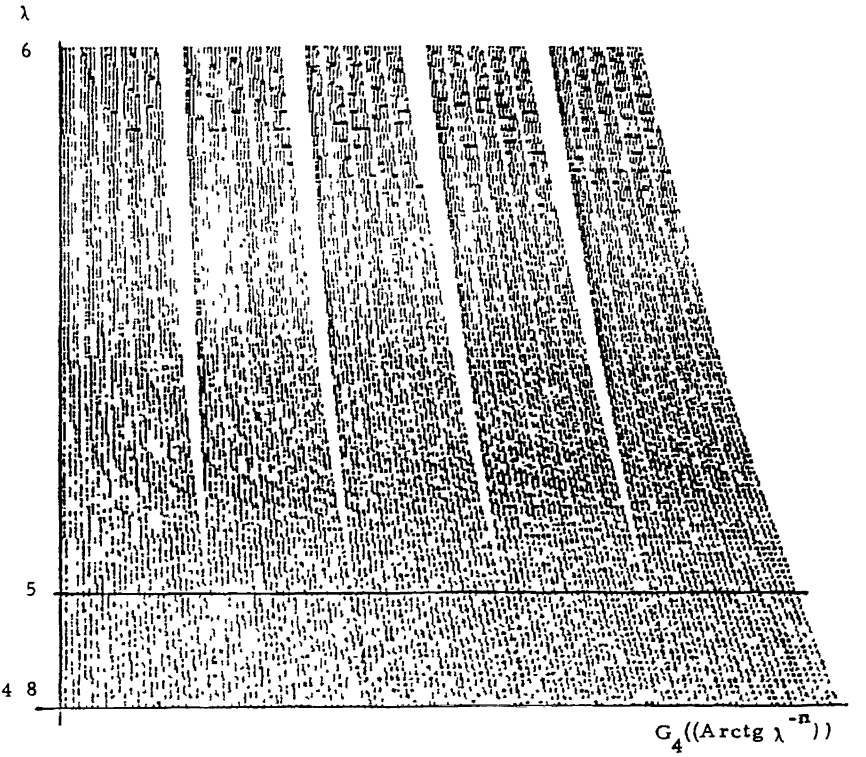

Fig 5. - Ensembles $G_4(\text{Arctg}(\lambda^{-n}))$ tracés pour λ variant continûment entre 4 8 et 6

$$\begin{cases} d'_n = d_n & \text{si} \quad n < N \\ d'_N = 0 \\ d'_n = p & \text{si} \quad n > N\,. \end{cases}$$

et $d = \sum_{n=0}^{\infty} d''_n\, e_n$, où

$$\begin{cases} d''_n = d_n & \text{si} \quad n < N \\ d''_N = 1 \\ d''_n = 0 & \text{si} \quad n > N\,. \end{cases}$$

(C) implique $c < d$. Soit b un élément de l'intervalle $]c, d[$.

Supposons que b puisse s'écrire $\sum_{n=0}^{\infty} d_n''' e_n$, avec $d_n''' e \{ 0, 1, ..., p \}$. Soit alors r le plus petit entier naturel tel que d_r''' est différent de d_r.

1) Si $r < N$, alors :

— si $d_r''' < d_r$ alors

$$c - b = \sum_{n=0}^{\infty} (d_n' - d_n''') e_n$$

$$= (d_r - d_r''') e_r + \sum_{n=r+1}^{\infty} (d_n' - d_n''') e_n$$

$$> 0 \quad \text{puisque} \quad e_r > p \sum_{n=r+1}^{\infty} e_n .$$

— si $d_r''' > d_r$ alors on montre de la même façon que $b - d > 0$.

2) Si $r \geqslant N$ alors :

— si $d_N''' = 0$ alors $b \leqslant c$

— si $d_N''' \geqslant 1$ alors $b \geqslant d$ par construction.

D'où contradiction, ce qui démontre le résultat énoncé.

Lorsque (C) est vérifiée, l'ensemble $G_p(E)$ vérifie d'autres propriétés intéressantes, comme le montrent les résultats suivants :

PROPRIÉTÉ 5 : *Si $E \in S$ vérifie (C) alors $G_p(E)$ est* fermé.

Démonstration : Soit (x_n) une suite d'éléments de $G_p(E)$ convergeant dans $\mathbb{R}$. Montrons que sa limite appartient a $G_p(E)$.

Posons $x_n = \sum_{i=0}^{\infty} d_i^n e_i$. Nous allons exhiber une suite (d_i), $i \in \{ 0, 1, ..., p \}$ telle que pour tout entier naturel i, il existe $N \in \mathbb{N}$ tel que si n est supérieur ou égal à N, alors $d_i^n = d_i$. (Ceci n'est qu'une façon d'exprimer une « convergence » de la suite des coordonnées des x_n.)

En effet, supposons :

$$\exists j \in \mathbb{N}, \quad \forall N, \quad \exists p \geqslant N, \quad \exists n \geqslant N, \quad d_j^n \neq d_j^p .$$

Nous avons :

$$| x_n - x_p | = \left| \sum_{i=0}^{\infty} (d_i^n - d_i^p) e_i \right| .$$

Soit $\varepsilon_j = \min_{i \leqslant j} \left\{ e_i - p \sum_{k=i+1}^{\infty} e_k \right\}$. D'après (C), $\varepsilon_j > 0$.

Soit r le plus petit entier naturel tel que $d_r^p \neq d_r^n$, il vient :

$$\begin{aligned} | x_n - x_p | &= \left| (d_r^n - d_r^p)\, e_r - \sum_{k=r+1}^{\infty} (d_k^p - d_k^n)\, e_k \right| \\ &\geqslant \left| e_r - \left| \sum_{k=r+1}^{\infty} (d_k^n - d_k^p)\, e_k \right| \right| \\ &\geqslant \left| e_r - p \sum_{k=r+1}^{\infty} e_k \right| \geqslant \varepsilon_j . \end{aligned}$$

On en déduit donc :

$$\exists \varepsilon = \varepsilon_j , \quad \forall N \in \mathbb{N} , \quad \exists n \geqslant N , \quad \exists p \geqslant N \quad | x_n - x_p | \geqslant \varepsilon .$$

Donc la suite (x_n) n'est pas une suite de Cauchy, ce qui est évidemment en contradiction avec le fait qu'elle soit convergente. Nous avons par conséquent :

$$\forall j \in \mathbb{N} , \quad \exists N_j \in \mathbb{N} , \quad \forall n \geqslant N_j , \quad \forall p \geqslant N_j , \quad d_j^n = d_j^p .$$

Soit la suite (d_j) définie par $d_j = d_j^{N_j}$, et soit x le réel égal à $\sum_{j=0}^{\infty} d_j\, e_j$. Il nous reste à montrer que x est la limite de la suite (x_n). Pour ceci, posons $M_j = \operatorname{Sup}_{k \leqslant j} N_k$.

Il vient :

$$\forall j \in \mathbb{N} , \quad \exists M_j \in \mathbb{N} , \quad \forall n \geqslant M_j , \quad \forall i \leqslant j , \quad d_i^n = d_i .$$

D'où :

$$\forall j \in \mathbb{N} , \quad \exists M_j \in \mathbb{N} , \quad \forall n \geqslant M_j , \quad | x - x_n | \leqslant p \sum_{i=j+1}^{\infty} e_i .$$

Or, puisque la série (e_i) est convergente, nous avons :

$$\lim_{j \to \infty} \sum_{i=j+1}^{\infty} e_i = 0 .$$

Donc x est bien la limite de la suite (x_n), donc, puisque x est élément de $G_p(E)$ par construction, la propriété 5 est démontrée.

PROPRIÉTÉ 6 : *Si $E \in S$ est une suite asymptotiquement géométrique de la forme (a^{-n}), avec $a > p + 1$, et vérifiant* (C), *alors la mesure de Lebesgue de $G_p(E)$ est nulle.*

Avant de démontrer cette propriété, énonçons le lemme suivant :

LEMME 1 : *Si l'on note*

$$B_{N,p}^{(d_0,d_1,\ \ ,d_N)}(E) = \left[\sum_{i=0}^{N} d_i e_i, \sum_{i=0}^{N} d_i e_i + p \sum_{i=N+1}^{\infty} e_i \right]$$

$$\Omega_p(E) = \bigcap_{N \in \mathbb{N}} \left[\bigcup_{(d_0,\ \ ,d_N) \in \{0,1,\ \ ,p\}^{N+1}} (B_{N,p}^{(d_0,d_1,\ \ ,d_N)}(E) \right]$$

Alors :

1) $\forall E \in S,\ G_p(E) \subset \Omega_p(E)$.

2) *Si* $E \in S$ *vérifie* (C), *alors,* $G_p(E) = \Omega_p(E)$, *et tous les intervalles* $B_{N,p}^{D}(E)$, *où* $D = (d_0, ..., d_N) \in \{ 0, 1, ..., p \}^{N+1}$ *sont disjoints.*

N.B. Pour démontrer la proposition 6, 1) suffit.

Démonstration du lemme :

— 1) est trivialement vraie.

— Pour prouver 2), nous attirons l'attention du lecteur sur le fait que la disjonction des $B_{N,p}^{D}(E)$ est une conséquence immédiate de (C), il nous suffit donc de montrer que $\Omega_p(E)$ est inclus dans $G_p(E)$. Pour ceci, considérons un élément x de $\Omega_p(E)$ et montrons qu'il appartient à $G_p(E)$:

Soit $N \in \mathbb{N}$, puisque les $B_{N,p}^{D}$ sont disjoints, il existe un unique $(d_0^x, d_1^x, ..., d_N^x)$ tel que $x \in B_{N,p}^{(d_0^x, d_1^x,\ \ ,d_N^x)}(E)$.

Considérons alors $y = \sum_{n=0}^{\infty} d_n^x e_n$. Il est clair que pour tout $N \in \mathbb{N}$ il existe $D \in \{ 0, 1, ..., p \}^N$ tel que x et y appartiennent au même $B_{N,p}^{D}(E)$. Or, la longueur de $B_{N,p}^{D}(E)$ décroît vers zéro avec N. Par conséquent, on a nécessairement $x = y$.

Démonstration de la propriété 6 : Posons :

$$A_{N,p}(E) = \bigcup_{(d_0,\ \ ,d_N) \in \{0,\ \ ,p\}^{N+1}} [B_{N,p}^{(d_0,d_1,\ \ ,d_N)}(E)] .$$

$$\left(\text{Nous avons alors } \Omega_p(E) = \bigcap_{N \in \mathbb{N}} A_{N,p}(E) \right).$$

Or, la mesure de $B_{N,p}^{D}(E)$ ne dépend pas de D, et vaut $P \sum_{n=N+1}^{\infty} e_n$, par conséquent la mesure de $A_{N,p}(E)$ est majorée par :

$$\sum_{(d_0,\ \ ,d_N) \in \{0,\ \ ,P\}} \left(p \sum_{k=N+1}^{\infty} e_k \right) = (p+1)^{(N+1)} p \sum_{n=N+1}^{\infty} e_n .$$

Mais ce dernier terme est équivalent, lorsque N tend vers l'infini, à :

$$((p + 1)/a)^N \, p(p + 1)/(a - 1) .$$

Or, cette quantité tend vers zéro lorsque N tend vers l'infini.

Donc la mesure de Lebesgue de $G_p(E)$ est nulle.

PROPRIÉTÉ 7 : *Si $E \in S$ est une suite géométrique de raison a, $a > p + 1$, alors la dimension de Hausdorff de $G_p(E)$ est égale à* $\text{Log}\,(p + 1)/\text{Log}\,(a)$.

Exemple : On retrouve comme cas particulier de ce résultat l'ensemble triadique de Cantor, dont la dimension de Hausdorff est bien connue et vaut Log (2)/Log (3).

Démonstration : Considérons la *mesure de Hausdorff dans la dimension d*, définie comme suit :

Soit G un sous-ensemble compact de $\mathbb{R}$, considérons un recouvrement Δ^ρ de G par des intervalles ouverts $\Delta_i^\rho (i \in \mathbb{N})$, de diamètre inférieur ou égal à ρ. Le terme :

$$H_\rho^d(G) = \inf_{\Delta^\rho} \left(\sum_{i=0}^{\infty} \text{Diam}\,(\Delta_i^\rho)^d \right).$$

Admet une limite (éventuellement infinie) quand ρ tend vers zéro par valeur supérieure, cette limite, notée $H(G)$, sera appelée *mesure de Hausdorff de G dans la dimension d.*

On montre les propriétés suivantes :

i) Si G est réunion de deux compacts disjoints G_1 et G_2, alors $H^d(G) = H^d(G_1) + H^d(G_2)$.

ii) Si F est homothétique de G dans le rapport k, alors $H^d(F) = k^d \, H^d(G)$.

iii) Il existe une quantité $\text{Dim}\,(G) \in [0, 1]$ vérifiant :

$$\begin{aligned} \text{Dim}\,(G) &= \inf \{ d \in [0, 1] / H^d(G) = 0 \} \\ &= \sup \{ d \in [0, 1] / H^d(G) = + \infty \} . \end{aligned}$$

Cette quantité $\text{Dim}\,(G)$ est appelée *Dimension de Hausdorff* de G. On trouvera plus de précisions sur cette dimension dans [31] et [55].

iv) Si la mesure de Lebesgue de G est non nulle, alors $\text{Dim}\,(G) = 1$.

Dans le cas qui nous intéresse, le résultat est une conséquence des propriétés 3 et 4. En effet, $G_p(E)$ peut alors s'écrire comme réunion de $p + 1$ compacts disjoints $A_0, \ldots, A_p$, qui lui sont homothétiques dans le rapport $1/a$. Par conséquent, $G_p(E)$ est ce que Kahane et Salem appellent un *Parfait homogène de type* $(p + 1, 1/a)$, donc, d'après leur étude (voir [31]), sa dimension

de Hausdorff est égale à $\mathrm{Log}(p+1)/\mathrm{Log}(a)$ On peut s'en persuader en écrivant que

$$\begin{aligned} H^d(G_p(E)) &= H^d(A_0) + \quad + H^d(A_p) \\ &= (p+1)\, H^d(A_0) \\ &= (p+1)\, a^{-d}\, H^d(G_p(E)) \end{aligned}$$

Par conséquent, si $(p+1)\, a^{-d} \neq 1$, alors $H^d(E)$ est nul ou infini Il suffit alors de montrer que pour $d = \mathrm{Log}(p+1)/\mathrm{Log}(a)$, la mesure de Hausdorff de $G_p(E)$ dans la dimension d est finie et non nulle La propriété iii) donne alors immédiatement le résultat

II A 3 *Problème de minimalité de la décomposition*

Introduction Il est évident que, si (e_n) est une base discrète géométrique ou asymptotiquement geométrique (par exemple une base usuelle de numération), alors si l'on désire approximer l'élément $x = \sum_{n=0}^{\infty} d_n\, e_n$ de $G_p(E)$ par la quantité $x^* = \sum_{n=0}^{N} d_n\, e_n$, le nombre N de chiffres nécessaires pour avoir $| x - x^* | \leqslant \varepsilon$ est proportionnel à $\mathrm{Log}(1/\varepsilon)$

Nous cherchons ici à déterminer si cette approximation est optimale, c'est-à-dire s'il existe une base discrète (e_n) telle que

$$\forall A > 0\,, \quad \exists N \in \mathbb{N}\,, \quad n \geqslant N \Rightarrow p \sum_{k=n+1}^{\infty} e_k \leqslant e^{-nA} \qquad \text{(D)}$$

Ceci serait d'autant plus intéressant que, nous le verrons ultérieurement, le temps de calcul des algorithmes de la partie III est proportionnel au nombre N de chiffres nécessaires à la décomposition approchée de x La réponse à la question matérialisée par (D) est non En fait, nous donnons ici un résultat plus général concernant la représentation des nombres

THÉORÈME 4 *Soit (f_n) une famille de « codages approchés d'éléments d'un intervalle I sur N chiffres de $\{0, 1, \ , p\}$ », c'est-à-dire une application de $\{0, 1, \ , p\}^N$ dans I, alors il existe deux constantes strictement positives K et c vérifiant*

$$\exists x \in I\,, \quad \forall D \in \{0, \ , p\}^N\,, \quad |f_N(d) - x| \geqslant ce^{-KN}$$

Démonstration Soit R_N l'ensemble $\{f_N(D)/D \in \{0, \ , p\}^N\}$ Le cardinal de R_N est inférieur ou égal à $(p+1)^N$, par conséquent, I contient au plus

$(P + 1)^N$ éléments de R_N. Notons ces éléments :

$$x_1 \leqslant x_2 \leqslant \cdots \leqslant x_{(p+1)^N}.$$

Notons aussi x_0 la borne inférieure de I et $x_{(p+1)^N+1}$ sa borne supérieure. Nous montrerons d'abord les résultats intermédiaires suivants :

LEMME 3 : *Il existe* $i \in \{ 0, 1, ..., (p + 1)^N \}$ *tel que*

$$x_{i+1} - x_i \geqslant d/((p + 1)^N + 1)$$

où l'on note $d = x_{(p+1)^N+1} - x_0$.

En effet, sinon on aurait :

$$d = \sum_{j=0}^{(p+1)^N} (x_{j+1} - x_j) < ((p + 1)^N + 1).d/((p + 1)^N + 1).$$

D'où le résultat.

LEMME 4 : *Il existe* x *appartenant à* I *tel que pour tout élément* D *de* $\{ 0, 1, ..., p \}^N$, *on ait :*

$$| x - f_N(D) | \geqslant d/[2((p + 1)^N + 1)] \geqslant d(p + 2)^{-N}/2.$$

La démonstration de ce résultat est immédiate : il suffit de prendre, pour le i) du lemme 3, $x = (x_{i+1} - x_i)/2$.

Le théorème découle de ce dernier résultat, en posant $c = d/2$ et $K = \text{Log}(p + 2)$.

On déduit de ce résultat l'optimalité de la décomposition des réels dans les bases géométriques ou asymptotiquement géométriques. Ceci est très important pour les algorithmes que nous étudierons ultérieurement : en effet, le temps de calcul nécessaire pour obtenir une précision donnée ε en abscisse sera proportionnel au nombre de d_i nécessaires à l'expression de l'argument, et donc à $\text{Log}(1/\varepsilon)$.

II.B. Bases discrètes multiplicatives

Nous nous intéressons maintenant à l'écriture de nombres réels non plus comme somme de termes prédéfinis, mais comme produit de ces termes. Un résultat, le théorème 5, montre qu'une étude exhaustive de ce cas est inutile et que la plupart des résultats concernant les bases additives peuvent être étendus aux bases multiplicatives. Nous verrons ultérieurement que ceci est vrai également pour les algorithmes.

II.B.1. *Définitions*

Dans tout ce qui suit, on notera M l'ensemble des suites réelles (e_n) dont les termes sont strictement supérieurs à 1 et décroissent vers 1, et telles que $\prod_{n=0}^{\infty} e_n$ est fini.

DÉFINITION 3 : *On appellera* Généré multiplicatif d'ordre p *de* $E = (e_n) \in M$, *l'ensemble* :

$$GM_p(E) = \left\{ \prod_{n=0}^{\infty} e_n^{d_n}/d_n \in \{ 0, ..., p \} \right\}.$$

DÉFINITION 4 : *On dira que* $E \in M$ *est une* Base discrète multiplicative d'ordre p *si* $GM_p(E)$ *est un intervalle.*

La fonction exponentielle étant un isomorphisme de groupes de $(\mathbb{R}, +)$ dans $(\mathbb{R}^{+*}, *)$, on déduit le résultat suivant, fondamental pour l'étude des ensembles $GM_p(E)$.

PROPRIÉTÉ 8 : *Soit* $E = (e_n)$ *un élément de* M, *la suite* $\mathrm{Log}(E) = (\mathrm{Log}(e_n))$ *est un élément de* S, *et la fonction exponentielle est une bijection de* $G_p(\mathrm{Log}(E))$ *dans* $GM_p(E)$.

On en déduit immédiatement, grâce à l'étude des suites additives, que $GM_p(E)$ a la puissance du continu, n'a pas de points isolés ; et que si pour tout $n \in \mathbb{N}$, $e_n > \left(\prod_{k=n+1}^{\infty} e_k \right)^p$, alors $GM_p(E)$ ne contient aucun intervalle, et tous ses éléments admettent un système de coordonnées unique sur E. Une autre conséquence de la propriété 8, plus importante pour notre étude est le théorème suivant :

THÉORÈME 5 : $E = (e_n) \in M$ *est une base discrète multiplicative d'ordre* p *si et seulement si* $\mathrm{Log}(E) = (\mathrm{Log}(e_n))$ *est une base discrète additive d'ordre* p, *c'est-à-dire si et seulement si pour tout entier naturel* n, *on a* :

$$e_n \leqslant \left(\prod_{k=n+1}^{\infty} e_k \right)^p.$$

Dans la partie qui suit, consacrée aux algorithmes, nous allons étudier le moyen de calculer les coordonnées d'un réel dans une base discrète additive ou multiplicative, ainsi que les applications de ce travail au calcul d'une classe de fonctions incluant les fonctions élémentaires.

III. ALGORITHMES

III.A. Algorithmes de calcul des d_i

Dans ce premier paragraphe, nous cherchons à calculer rapidement les premiers termes des coordonnées d'un réel donné dans une base discrète. Nous étudierons surtout le cas des bases additives, celui des bases multiplicatives pouvant rapidement en être déduit.

THÉORÈME 6 (*algorithme unidirectionnel*) : *Si* $E = (e_n)$ *est une base discrète d'ordre p, alors pour tout élément t de* $\left[0, p \sum_{n=0}^{\infty} e_n\right]$, *les suites* (t_n) *et* (d_n) *définies comme suit vérifient* $t = \lim_{n\to\infty} t_n = \sum_{n=0}^{\infty} d_n e_n$.

$$\begin{cases} t_0 = 0 \\ d_n = \text{Max}\,\{ j \leqslant p, t_n + j e_n \leqslant t \} \\ t_{n+1} = t_n + d_n e_n . \end{cases}$$

Démonstration : En fait le théorème 6 a déjà été énoncé et démontré, il constitue la condition suffisante du théorème 1. Nous l'avons replacé ici pour plus de clarté.

THÉORÈME 7 (*algorithme bidirectionnel*) : *Si E est une base discrète d'ordre p, alors pour tout élément t de* $\left[-p \sum_{n=0}^{\infty} e_n, +p \sum_{n=0}^{\infty} e_n\right]$, *les suites* (t_n) *et* (d_n) *définies comme suit vérifient* $t = \sum_{n=0}^{\infty} d_n e_n = \lim_{n\to\infty} t_n$.

$$\begin{cases} t_0 = 0 \\ si \quad t_n \leqslant t \quad alors \\ \quad d_n = \text{Max}\,\{ 1 \leqslant j \leqslant p, t_n + (j-1) e_n \leqslant t \} \\ sinon \quad d_n = \min\{ -p \leqslant j \leqslant -1, t_n + (j+1) e_n \geqslant t \} \\ t_{n+1} = t_n + d_n e_n . \end{cases}$$

Démonstration : Comme dans la condition suffisante du théorème 1, on montrera aisément par récurrence la relation :

$$| t_n - t | \leqslant p \sum_{k=n}^{\infty} e_k .$$

Ce qui montrera le résultat désiré.

On peut de la même manière exhiber des algorithmes multiplicatifs bidirectionnel et unidirectionnel.

III.B. Applications

Nous allons maintenant étudier quelques applications des algorithmes précédents au calcul de certaines fonctions, en particulier des fonctions élémentaires. On trouvera plus de détails dans [39] et [41].

ALGORITHME : *Calcul de l'exponentielle.*

Nous avons vu précédemment que, si $1 < a \leqslant p + 1$ alors $(\mathrm{Log}(1 + a^{-n}))$ est une base discrète d'ordre p. L'idée de base de cet algorithme est d'écrire un réel x sous la forme :

$$x = \sum_{n=0}^{\infty} d_n \mathrm{Log}(1 + a^{-n}) \qquad d_n \in \{0, ..., p\}.$$

en utilisant l'algorithme unidirectionnel et d'avoir comme résultat :

$$e^x = \prod_{n=0}^{\infty} (1 + a^{-n})^{d_n}.$$

Sur une machine fonctionnant en base B, les multiplications par une puissance de B peuvent s'exécuter d'une manière très simple : elles se réduisent soit à un *décalage* sur un système à virgule fixe, soit à une *addition à l'exposant* en virgule flottante.

Ici, on aura donc tout avantage à prendre $a = B$, et donc à utiliser la base discrète d'ordre $B - 1$ $(\mathrm{Log}(1 + B^{-n}))$. En effet, une multiplication par $(1 + B^{-n})$ se ramènera à une addition et une multiplication par B^{-n}. Il est évident que la série et le produit infini seront tronqués à un rang N, on peut montrer qu'alors l'erreur relative sur le résultat est majorée par un terme équivalent à B^{-N}.

Voici l'algorithme obtenu :

Exponentielle.

(* cet algorithme calcule l'exponentielle de t *)

$\left(* \text{ converge pour } t \in \left[0, (B-1) \sum_{i=0}^{\infty} \log(1 + B^{-i})\right] *\right)$

(* précision relative : environ B^{-N} *)

(* résultat : la valeur finale de la variable exp *)

```
Début
  x ← 0; exp ← 1;
  pour k = 0 jusqu'à n faire
    début
        d ← 0; u ← 0;
      tant que (u < t) et (d < B − 1) faire
          début
              u ← x + Log(1 + B^−k);
            si  u ⩽ t alors
                début
                  x ← u;
                  exp ← exp + exp.B^−k;
                fin;
            d ← d + 1;
          fin;
    fin;
Fin.
```

L'algorithme présenté ici utilisait le concept de base discrète *additive*, nous allons maintenant en étudier un autre, destiné au calcul du logarithme, et qui utilise la notion de base discrète *multiplicative*.

ALGORITHME 2 : *Calcul du logarithme.*

Sur une machine travaillant en base B, l'idée fondamentale de cet algorithme, qu'on pourrait qualifier de « duale » de l'idée précédente (cette notion de dualité possède d'ailleurs un sens plus profond, présenté dans [39]), est de décomposer un réel x sur la base discrète multiplicative d'ordre $B - 1(1 + B^{-n})$:

$$x = \prod_{n=0}^{\infty} (1 + B^{-n})^{d_n} \qquad d_n \in \{ 0, ..., B - 1 \}.$$

Et d'obtenir alors :

$$\mathrm{Log}(x) = \sum_{n=0}^{\infty} d_n \mathrm{Log}(1 + B^{-n}).$$

Dans l'algorithme qui suit, qui utilise l'algorithme unidirectionnel multiplicatif, la série et le produit infini sont tronqués au rang N, ce qui assure sur le résultat une erreur absolue majorée par B^{-N}.

Voici l'algorithme obtenu :

Logarithme.
(* calcule le logarithme de t *)

$$\left(* \text{ où } t \in \left[1, \left(\prod_{i=0}^{\infty}(1 + B^{-i})\right)^{(B-1)}\right] *\right)$$

(* résultat : la dernière valeur de la variable L *)

```
Début
    x ← 1 ; L ← 0 ;
  pour k = 0 jusqu'à N faire
      début
          d ← 0 ; u ← 1 ;
        tant que (u < t) et (d < B − 1) faire
            début
                u ← u + u.B^−k ;
              si u ⩽ t alors
                  début
                    x ← u ;
                    L ← L + Log(1 + B^−k) ;
                  fin ;
                d ← d + 1 ;
            fin ;
      fin ;
Fin.
```

ALGORITHME 3 : *Calcul de la racine carrée.*

Supposons encore que nous utilisons une machine travaillant en base B. Le principe de base de cet algorithme est de décomposer un réel x sur la base discrète multiplicative d'ordre $(p - 1)$:

$$(1 + B^{-n})^2$$

afin de l'exprimer sous la forme :

$$x = \prod_{n=0}^{\infty} (1 + B^{-n})^{2d_n}$$

et d'obtenir finalement :

$$\sqrt{x} = \prod_{n=0}^{\infty} (1 + B^{-n})^{d_n}.$$

L'algorithme qui suit utilise l'algorithme bidirectionnel multiplicatif. Les produits infinis sont tronqués au rang N, ce qui assure sur le résultat final une erreur relative de l'ordre de B^{-N}.

Racine carrée.

(* cet algorithme calcule la racine carrée de t *)

(* converge pour $t \in \left[1, \prod_{i=0}^{\infty} (1 + B^{-i})^2\right]$ *)

(* précision relative : environ B^{-N} *)

(* résultat : la valeur finale de la variable sq *)

```
Début
  x ← 1 ; sq ← 1 ;
  pour k = 0 jusqu'à N faire
    début
      d ← 0 ; u ← 0 ;
      tant que (u ⩽ t) et (d < B − 1) faire
        début
          u ← x + B^(−2k) x + B^(−k) x + B^(−k) x ;
          si u ⩽ t alors
            début
              x ← u ;
              sq ← sq + sq.B^(−k) ;
            fin ;
          d ← d + 1
        fin
    fin
Fin.
```

Cet algorithme peut être aisément transformé en algorithme de calcul de la racine k-ième d'un nombre en utilisant la base discrète multiplicative d'ordre $(B - 1)$:

$$((1 + B^{-N})^k).$$

ALGORITHME 4 : *Calcul de l'exponentielle complexe et des principales fonctions élémentaires.*

Cet algorithme, présenté en détail dans [41] englobe les algorithmes de calcul du logarithme et de l'exponentielle réelle présentés auparavant et l'algorithme CORDIC de J. Volder pour le calcul des fonctions trigonométriques. Nous le présenterons en base 2 pour plus de simplicité. Son principe de base consiste à

décomposer la partie réelle d'un nombre complexe z sur la base discrète additive d'ordre 1 ($\mathrm{Log}(1 + 2^{-n})$) et sa partie imaginaire sur la base discrète additive d'ordre 1 ($\mathrm{Arctg}(2^{-n})$). On montre alors dans [41] que le schéma itératif :

$$\begin{cases} a_{n+1} = (1 + 2^{-n})^{d_n^x}(a_n - b_n d_n^y 2^{-n}) \\ b_{n+1} = (1 + 2^{-n})^{d_n^x}(b_n + a_n d_n^y 2^{-n}) \\ x_{n+1} = x_n - d_n^x \mathrm{Log}(1 + 2^{-n}) \\ y_{n+1} = y_n - d_n^y \mathrm{Arctg}\, 2^{-n}. \end{cases}$$

Nous permet de calculer les différentes fonctions suivantes :

Exponentielle complexe :

La suite $(a_n + ib_n)$ converge vers l'exponentielle de $(x_0 + iy_0)$, si $x_0 \in \left[0, \sum_{n=0}^{\infty} \mathrm{Log}(1 + 2^{-n})\right]$, $y_0 \in \left[0, \sum_{n=0}^{\infty} \mathrm{Arctg}(2^{-n})\right]$, et si d_i^x et d_i^y sont choisis comme suit :

Si $x_k \geqslant \mathrm{Log}(1 + 2^{-k})$ alors $d_k^x = 1$ sinon $d_k^x = 0$.

Si $y_k \geqslant 0$ alors $d_k^y = 1$ sinon $d_k^y = -1$.

Avec les points de départ :

$$b_0 = 0, \quad a_0 = \left(\left(\prod_{k=0}^{\infty} (1 + 2^{-2k})\right)^{-1/2}\right) \sim 0.6072529... = K.$$

En posant $d_k^y = 0$ pour tout k, on retrouve l'algorithme de calcul de l'exponentielle réelle présenté auparavant ($B = 2$), et en posant $d_k^x = 0$ pour tout k, on retrouve l'algorithme CORDIC de Volder pour le calcul des fonctions sinus et cosinus.

Logarithme réel :

C'est l'algorithme 2 présenté auparavant, que l'on retrouve en posant $d_k^y = 0$ pour tout k, et $d_k^x = 1$ si $a_k + 2^{-k} a_k \leqslant a$, 0 sinon, avec les points de départ $b_0 = 0$, $x_0 = 0$, $a_0 = 1$ (on calcule le logarithme de a).

Arctangente réelle :

On calcule l'arctangente d'un réel quelconque b, en posant : si $b_k > b$ alors $d_k^y = 1$ sinon $d_k^y = -1$; $d_k^x = 0$ pour tout k, avec les points de départ $a_0 = K$, $y_0 = b_0 = 0$. Le résultat désiré est la valeur finale de la variable y. Au bout de N itérations, l'erreur absolue sur le résultat est majorée par 2^{-N}. L'algorithme ainsi obtenu est l'algorithme CORDIC de calcul de l'arctangente.

IV. CONCLUSIONS

L'étude menée ici constitue un approfondissement théorique des notions présentées dans des articles précédents ([39] et [41]). Elle permet d'unifier et de justifier des algorithmes, dont certains, connus depuis longtemps ([57], [58]) ne semblaient pas avoir de parenté profonde.

REFERENCES

[1] M. ABRAMOWITZ and I. A. STEGUN, *Handbook of Mathematical Functions with formulas, graphs, and mathematical tables*, Nat. Bur. Standards, Appl. Math. Series, 55, Washington D.C., 1964.

[2] H. M. AHMED, J. M. DELOSME, M. MORF, *Highly concurrent computing structures for matrix arithmetic and signal processing*, Computer, Jan. 1982.

[3] F. ANCEAU, *Architecture and design of Von Neumann microprocessors*, Nato advanced summer institute, July 1980.

[4] M. ANDREWS and T. MRAZ, *Unified elementary function generator*, Microprocessors and Microsystems, Vol. 2 nº 5, Oct. 1978, pp. 270-274.

[5] P. W. BAKER, *More efficient radix-2 algorithms for some elementary functions*. IEEE Trans. on computers, vol. c-24 nº 11, Nov. 1975, pp. 1049-1054.

[6] P. W. BAKER, *Suggestion for a fast binary Sine/Cosine generator*, IEEE Trans. on Computers, Nov. 1976, pp. 1134-1136.

[7] R. P. BRENT, *Multiple-precision zero-finding methods and the complexity of elementary function evaluation*, Analytic Computational Complexity (Ed. by J. F. Traub), Academic Press, New York, 1975, pp. 151-176.

[8] R. P. BRENT, *Fast multiple-precision evaluation of elementary functions*, J. ACM 23, 1976, pp. 242-251.

[9] R. P. BRENT, *Unrestricted algorithms for elementary and special functions*, Information Processing 80, S. H. Lavington ed., North-Holland Publishing Comp., pp. 613-619.

[10] T. H. CHAN and O. H. IBARRA, *On the space and time complexity of functions computable by sample programs*, Siam J. Comput., Vol. 12, nº 4, Nov. 1983.

[11] T. C. CHEN, *Automatic computation of exponentials, logarithms, ratios and square roots*. IBM J. Res. and Development, Vol. 16, July 1972, pp. 380-388.

[12] C. W. CLENSHAW and F. W. J. OLVER, *Bzyond floating point*, J. of the ACM, Vol. 31, nº 2, April 1984, pp. 319-328.

[13] W. CODY and W. WAITE, *Software manual for the elementary functions*, Prentice-Hall, inc., Englewood cliffs, New-Jersey, 1980.

[14] W. CODY, *Implementation and testing of function software*, *Ibid.*

[15] W. CODY, *Basic concepts for computational software*, *Ibid.*

[16] W. CODY, *Performance testing of function subroutines*, AFIPS Conf. Proc., Vol. 34, 1969 SJCC, AFIPS Press, Montvale, N.J., 1969, pp. 759-763.

[17] J. T. COONEN, *An implementation guide to a proposed standard for floating-point arithmetic*, IEEE Computer, Jan. 1980.

[18] J M DELOSME, *VLSI implementation of rotations in pseudo-euclidian spaces,* proc 1983 IEEE Int Conf on ASSP, Boston, April 1983, pp 927-930

[19] J M DELOSME, *The matrix exponential approach to elementary operations,* Depart of Electrical Engineering, Yale Univ , New Haven

[20] B DE LUGISH, *A class of algorithms for automatic evaluation of certain elementary functions in a binary computer,* Ph D dissertation, Dep Computer sci , Univ of Illinois, Urbana, June 1970

[21] B DERRIDA, A GERVOIS, Y POMEAU, *Iteration of endomorphisms on the real axis and representation of numbers* Commissariat a l'energie Atomique, Service de physique theorique, CEN Saclay

[22] A M DESPAIN, *Fourier transform computers using CORDIC iterations,* IEEE Trans on Computers, Vol c-23 n° 10, Oct 1974

[23] A M DESPAIN, *Pipeline and parallel-pipeline FFT Processors for VLSI implementations,* IEEE Trans on Computers, Vol c-33 n° 5, May 1984

[24] M D ERCEGOVAC, *Radix-16 evaluation of certain elementary functions,* IEEE Trans on Computers, Vol c-22 n° 16, June 1973

[25] M D ERCEGOVAC, *A general method for evaluation of functions in a digital computer,* Computer sci dep , School of Engineering & Applied science, Univ of California, Los Angeles, California 90024

[26] C T FIKE, *Computational evaluation of math functions,* Prentice-Hall, Englewood cliffs, New-Jersey, 1968

[27] W M GENTLEMAN, *More on algorithms that reveal properties of floating-point arithmetics units,* Comm of the ACM, Vol 17, n° 5, May 1974

[28] G W GERRITY, *Computer representation of real numbers,* IEEE Trans Computers, Vol c-31 n° 8, Aug 1982

[29] G H HAVILAND and A A TUSZYNSKY, *A CORDIC arithmetic processor chip,* IEEE Trans on Computers, Vol c-29 n° 2, Feb 1980

[30] J F HART, E W CHENEY, C L LAWSON, H J MAEHLY, C K MESZTENYI, J R RICE, H C TACHER, Jr and C WITZGALL, *Computer Approximations,* Wiley N Y , 1968

[31] J P KAHONE and R SALEM, *Ensembles parfaits et series trigonometriques,* Actualités scientifiques et industrielles 1301, Hermann Paris, 1963

[32] A H KARP, *Exponential and logarithm by sequential squaring,* IEEE Trans on Computers, Vol c-33, n° 5, May 1984, pp 462-464

[33] D E KNUTH, *The art of computer programming,* Vol 2, Addison Wesley, Reading, Mass , 1969

[34] J KROPA, *Calculator algorithms,* Math Mag , Vol 51 n° 2, March 1978, pp 106-109

[35] J D MARASA and D W MATULA, *A simulated study of correlated error propagation in various finite-precision arithmetic,* IEEE Trans on Computers, Vol c-22, n° 6, June 1973

[36] C MASSE, L'itération de Newton *convergence et chaos,* these de troisieme cycle, Universite Grenoble I, Oct 1984

[37] D W MATULA, *Basic digit sets for radix representation,* J of the ACM, Vol 29 n° 4, Oct 1982, pp 1131-1143

[38] J E MEGGITT, *Pseudo Division and Pseudo Multiplication Processes,* IBM of Res and Dev , Vol 6, April 1962, pp 210-227

[39] J M MULLER, *Discrete basis and computation of elementary functions,* IEEE Trans on Computers, Sept 1985, pp 857-862

[40] J. M. MULLER, *Conditionnement de fonctions et représentation flottante des nombres réels*, RR Math. App. nº 453, Grenoble, 1984.

[41] J. M. MULLER, *A hardware algorithm for computing the complex exponential fonction*, RR Math. App. nº 467, Grenoble, 1984.

[42] A. NASEEM and P. D. FISHER, *A modified CORDIC Algorithm*, Preprint Dept. of Electrical Engineering and Systems Science, Michigan State Univ., East Lansing, Michigan 48824.

[43] F. W. J. OLVER, *A new approach to error arithmetic*, SIAM J. Numer. Analysis, Vol. 15 nº 2, April 1978.

[44] G. PAUL and W. WAYNE WILSON, *Should the elementary function library be incorporated into computer instruction sets*, ACM Trans. on Math. Software, Vol. 2 nº 2, June 1976, pp. 132-142.

[45] W. PARRY, *On the β-expansion of real numbers*, Acta math. acad. sci. Hung., 11, 1960, pp. 401-416.

[46] M. PICHAT, *Contribution à l'étude des erreurs d'arrondi en arithmétique à virgule flottante*, thèse d'état, Grenoble, France, 1976.

[47] A. RENYI, *Representations for real numbers and their ergodic functions*, Acta. Math. Acad. Sci. Hungary, 1957, pp. 477-493.

[48] A. RENYI, *On the distribution of the digits in Cantor's series*, Mat. Lapok 7, 1956, pp. 77-100.

[49] F. ROBERT, *Itération machine d'une fonction affine*, RR Math. App. nº 440, IMAG, Grenoble, France.

[50] B. P. SARKAR and E. V. KRISHNAMURTHY, *Economic pseudodivision processes for obtaining square root*, logarithm and arctan, IEEE Trans. on Computers, Dec. 1971, pp. 1589-1593.

[51] C. W. SCHELIN, *Calculator function approximation*, Amer. Math. Monthly 90, 5, May 1983.

[52] H. SCHMID and A. BOGOCKI, *Use decimal CORDIC for generation of many transcendental functions*, Electrical design mag., Feb. 1973, pp. 64-73.

[53] O. SPANIOL, *Computer arithmetic and design*, J. Wiley & Sons, 1981.

[54] W. H. SPECKER, *A Class of algorithms for* ln (x), exp(x), sin (x), cos (x), arctan (x) *and* arcot (x), IEEE Trans. on electronic computers, Vol. ec-14, 1965, pp. 85-86.

[55] C. TRICOT, *Mesures et dimensions*, Thèse d'état, Université Paris-sud, centre d'Orsay, Paris, Dec. 1983.

[56] J. M. TRIO, *Microprocesseurs 8086-8088 Architecture et programmation*, Coprocesseur de calcul 8087, Éditions Eyrolles, Paris, 1984.

[57] J. VOLDER, *The CORDIC Computing technique*, IRE Trans. on Computers, Vol. ec-8, Sept. 1959, pp. 330-334.

[58] J. WALTHER, *A Unified algorithm for elementary functions*, Joint Computer Conference Proceedings, Vol. 38, pp. 379-385.

[59] E. H. WOLD, *Pipeline and parallel-pipeline FFT processors for VLSI implementations*, IEEE Trans. on Computers, Vol. c-33 nº 5, May 1984.