数 値 解 析 29 － 3
（１９８９．７．７）

多重指数分割による数値表現における多重度の変動化について

中森 眞理雄，　萩原 洋一

東京農工大学 工学部 数理情報工学科

あらまし

多重指数分割による新しい数値表現方式を提案する．本方式では，指数分割の多重度ｎは指数の大きさが０から離れるほど大きくなり，ＵＲＲなど他の数値表現方式より大きな数を表すことができる．数値の大きさと精度との関係を他の方式と比較する．

A New Floating Point Representation of Numbers Based on Variably Multiple Exponential Cut

Mario Nakamori and Yoichi Hagiwara

Faculty of Technology
Tokyo University of Agriculture and Technology

Abstract

A new real number representation is proposed that is based on multiple exponential cut, where the degree of multiplicity n increases according as the number of bits representing the exponent grows larger. The proposed representation expresses greater number than other representation such as URR.

1．はじめに

計算機で数値を扱う代表的な方式には固定小数点方式と浮動小数点方式とがある．いずれにしても，一つの数値を一定個数の語で表すのが普通である（この他，数値を不定個数の語で表したり，数式を文字・記号列で扱ったりする方式もあるが，本論文で考察しているのとは，応用場面が異なるので，ここでは一応対象外とする）．浮動小数点方式は固定小数点方式より精度が悪いが，プログラマをスケーリングの煩わしさから開放するという利点があるため，今日では，計算機による数値計算には浮動小数点方式を用いるのが常識である．しかし，伝統的な浮動小数点方式は表現できる数値の範囲が狭く，しかも，あふれに対する対策がないことから，数値解析の研究者たちの間で不満が多かった．この不満を解消するために，最近十年間にＩＥＥＥ方式[1,2]，松井・伊理方式[3]，ＵＲＲ[4]など種々の新しい数値表現方式が提案されている．

実用的な大きさの数値に対しては十分な精度を確保し，しかも，表現できる数値の範囲を広げ，あふれに対処するには，一般に次の方法が考えられる．

(a) 指数部，仮数部を可変長にする；
(b) 非数の概念を用いる；
(c) 指数，仮数にけち表現（正規化により１となる冒頭の１ビットを省略する表現）を用いる．

(a)は表現できる数値の範囲を広げるために，(b)はあふれに対処するために，(c)は精度を確保するための方法と考えてよいであろう．

ＩＥＥＥ方式では，(b)と(c)を採用しており，また丸めの方法を何通りか指定できるという特徴がある．ただし，表現できる数値の範囲や精度は伝統的な方式と比べて極端に違うわけではない．

松井・伊理方式は，(a)，(b)，(c) をすべて採用しているが，語長に独立でない．

ＵＲＲは，(a)，(b)，(c)をすべて採用しており，しかも

（α）語長に独立であること，
（β）数値の大小関係が（同じビットパターンのままで）固定小数点数とみなしたときの大小関係と一致すること，
（γ）すべてのビットが０の数値は（ＵＲＲとしても）０を表すこと，
（δ）あらゆるビットパターンが（ＵＲＲ数として）意味をもつこと，

などのすぐれた特徴があるが，数値の絶対値が１から離れるにしたがって精度が急速に悪くなる．

ＵＲＲにおいて数値の絶対値が１から離れたときの精度を改善するために指数部を多段に分けた方式（すなわち，多重指数分割方式）も提案されている[5,6]．なお，松井と伊理は彼らの方式の提案の中で指数部を多段化した版についても言及している[3]．

しかし，いずれにしても，指数部の段の数は固定されている．より根本的には指数部の段数自体を可変にすることであろう．本論文では，指数部の段数を可変にし，指数が１から離れるほど段数を大きくする表現方式を提案する．

本論文は浮動小数点表示方式を論じているが，本質的には整数（指数）を符号化する“自然数の表現の問題”[8]である．自然数の表現については，本論文の方式と似た段数可変の方式が Knuth(7) により提案されている．本論文では，Knuth の方式との違いについても述べる．

以下では，**2**で二重指数分割による数値表現と多重指数分割による数値表現を概観し，**3**で多重度を変化させる新しい数値表現を提案し，諸方式を比較する．**4**では Knuth による方式を述べ，本論文の方式との違いを述べる．

2．多重指数分割による数値表現について

本節では，**3**で提案する浮動小数点方式との比較対象として，既存の多重指数分割による数値表現を簡単に説明する．

2.1 二重指数分割（ＵＲＲ）

本表現方式は語の上位ビットからＳ，Ｌ，Ｅ，Ｆのフィールドに分けられる（**図1**）．各フィールドの意味は次の通りである．

Ｓ：　数値の（したがって仮数の）符号．Ｓ＝０のとき符号は＋，Ｓ＝１のとき－．
Ｌ：　Ｅ部の長さを表す．可変長で$1^{k+1}0$（指数が非負のとき）あるいは$0^{k}1$（指数が負のとき）の形（ｋ≧０）でｋが値である．Ｌ部中の最上位のビットをｔとする．
Ｅ：　指数を表す．ただし，けち表現を用いるので，ｋ＝１のときは実質的には表示されない．
Ｆ：　仮数を２進数で表す．ただし，けち表現を用いる．

指数が負の場合はＥ部を，仮数が負の場合はＦ部を，２の補数で表示する．

S: sign bit

(L, E): exponent part

F: fraction part

図1　URR

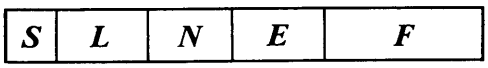
$$v = f \times 2^{e}$$

$$f = \begin{cases} 1 + (0.f_1 f_2 \cdots f_{n-2k})_2 & (s = 0 \text{のとき}) \\ -2 + (0.f_1 f_2 \cdots f_{n-2k})_2 & (s = 1 \text{のとき}) \end{cases}$$

$$e = \begin{cases} 2^{k-2} + (e_{k-3} \cdots e_0)_2 & (s = 0,\ t = 1 \text{のとき}) \\ -2^{k-1} + (e_{k-3} \cdots e_0)_2 & (s = 0,\ t = 0 \text{のとき}) \\ -2^{k-2} + (e_{k-3} \cdots e_0)_2 - 1 & (s = 1,\ t = 1 \text{のとき}) \\ 2^{k-1} + (e_{k-3} \cdots e_0)_2 - 1 & (s = 1,\ t = 0 \text{のとき}) \end{cases}$$

ただし，k＝1のときは

$$e = \begin{cases} 0 & (s \neq t \text{のとき}) \\ -1 & (s = t \text{のとき}) \end{cases}$$



S: sign bit

(L, E): exponent part

F: fraction part

図2　三重指数分割による数値表現

2.2 多重指数分割（多段URR）

URRの自然な拡張として，E部を二段に分割した方式が考えられる．この方式では，語は上位ビットからS，L，N，E，Fのフィールドに分けられる（**図2**）．各フィールドの意味は次の通りである．

S：　数値の（したがって仮数の）符号．＋のときS＝0，－のときS＝1．

L：　N部の長さを表す。可変長で$1^{k+1}0$（指数が非負のとき）あるいは$0^{k}1$（指数が負のとき）の形（k≧0）．kを表す．

N：　E部の長さを2進数で表す．ただし，けち表現を用いる．

E：　指数を2進数で表す．ただし，けち表現を用いる．

F：　仮数を2進数で表す．ただし，けち表現を用いる．

本来のURRではN部がなく，L部が直接E部の長さを表している点がこの方式と違う．また、指数や仮数が負の場合のN部，E部，F部の扱いは本来のURRと同様である．

3．変動型多重指数分割による数値表現の提案

3.1 変動型多重指数分割

本表現方式は語の上位ビットからS，L，E，Fのフィールドに分けられる（**図3**）．各フィールドの意味は次の通りである．

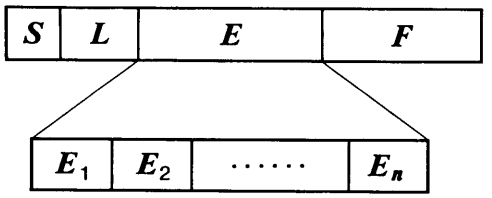
S：　数値の（したがって仮数の）符号．S＝0のとき符号は＋，S＝1のとき－．

L：　段数を表す．可変長で$1^{n+1}0$（指数が非負のとき）あるいは$0^{n}1$（指数が負のとき）の形（n≧0）でnが値である．

E：　L部が10のときは空とし，L部が$1^{n+1}0$あるいは$0^{n}1$（n≧1）のときはn個のフィールドE_1，E_2，…，E_nから成る．E_1はE_2の長さを表し，E_2はE_3の長さを表し，…，E_{n-1}はE_nの長さを表し，E_nは指数を表す．ただし，E_1，E_2，…，E_nはいずれもけち表現を用い，E_1が表すのはE_2のけち表現で省略されるビットを含まない長さであり，E_2が表すのはE_3のけち表現で省略されるビットを含まない長さであり，…，E_{n-1}が表すのはけち表現で省略されるビットも含めたE_nの長さである（E_nの長さだけ，けち表現で省略されるビットの扱いが異なることに注意）．n＝1のときはE_1の長さは1ビット（ただし，けち表現のため冒頭の1ビットは表示されないので，実質的にはE_1は表示されない），n＞1のときはE_1の長さは2ビット（ただし，けち表現のため冒頭の1ビットは表示されないので，実質的には1ビット）．

F：　仮数を2進数で表す．ただし，けち表現を用いる．

指数や仮数が負の場合のE部，F部の扱いはURR



S: sign bit

(L, E): exponent part

F: fraction part

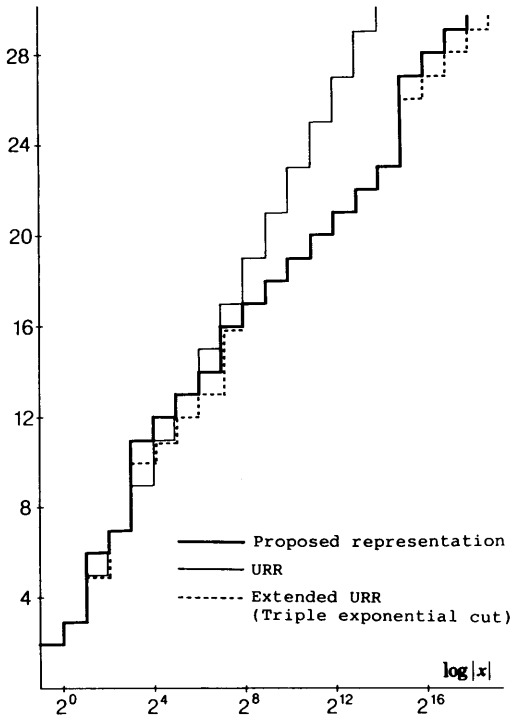
図3　変動型多重指数分割による数値表現

と同様である．ＵＲＲの特徴（α），（β）は本方式でも満たされる．

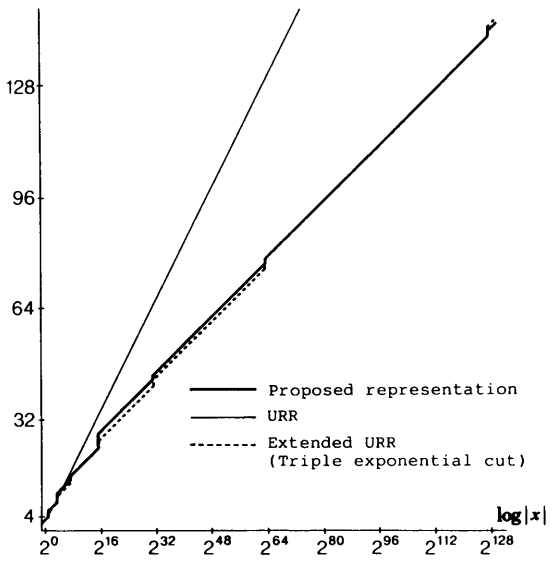
Ｌ部，Ｅ部について具体例を次に示す（＊印は０または１．下線をつけたビットはけち表現のため実際には語中に表示しない）．

Ｌ部	Ｅ部	指数の値
00001 0 0 0110 0 0000000000 0$*^{1022}$		-2^{1022} $\sim -2^{1023}-1$
.....		
00001 0 0 0110 0 1111111111 0$*^{511}$		$-2^{511} \sim -2^{512}-1$
00001 0 0 0111 0 0000000000 0$*^{510}$		$-2^{510} \sim -2^{511}-1$
.....		
00001 0 0 0111 0 1111111111 0$*^{255}$		$-2^{255} \sim -2^{256}-1$
00001 0 1 000 0 0000000 0$*^{254}$		$-2^{254} \sim -2^{255}-1$
.....		
00001 0 1 000 0 1111111 0$*^{127}$		$-2^{127} \sim -2^{128}-1$
00001 0 1 001 0 000000 0$*^{126}$		$-2^{126} \sim -2^{127}-1$
.....		
00001 0 1 001 0 111111 0$*^{63}$		$-2^{63} \sim -2^{64}-1$
00001 0 1 010 0 00000 0$*^{62}$		$-2^{62} \sim -2^{63}-1$
.....		
00001 0 1 010 0 11111 0$*^{31}$		$-2^{31} \sim -2^{32}-1$
00001 0 1 011 0 0000 0$*^{30}$		$-2^{30} \sim -2^{31}-1$
.....		
00001 0 1 011 0 1111 0$*^{15}$		$-2^{15} \sim -2^{16}-1$
0001 0 0 0000 0$*^{14}$		$-2^{14} \sim -2^{15}-1$
.....		
0001 0 0 0111 0$*^{7}$		$-2^{7} \sim -2^{8}-1$
0001 0 1 000 0******		$-2^{6} \sim -2^{7}-1$
0001 0 1 001 0*****		$-2^{5} \sim -2^{6}-1$
.....		
0001 0 1 010 0****		$-2^{4} \sim -2^{5}-1$
0001 0 1 011 0***		$-2^{3} \sim -2^{4}-1$
001 0 0 0**		$-2^{2} \sim -2^{3}-1$
001 0 1 0*		$-2^{1} \sim -2^{2}-1$
01 0		-1
10		0
110 1		1
1110 1 0 1*		$2^{1} \sim 2^{2}-1$
1110 1 1 1**		$2^{2} \sim 2^{3}-1$
11110 1 0 1 0 1***		$2^{3} \sim 2^{4}-1$
11110 1 0 1 0 1 1****		$2^{4} \sim 2^{5}-1$
.....		
11110 1 0 1 1 0 1*****		$2^{5} \sim 2^{6}-1$
11110 1 0 1 1 1 1******		$2^{6} \sim 2^{7}-1$
11110 1 1 1000 1*^{7}		$2^{7} \sim 2^{8}-1$
.....		
11110 1 1 1111 1*^{14}		$2^{14} \sim 2^{15}-1$
111110 1 0 1 0 0 10000 1*^{15}		$2^{15} \sim 2^{16}-1$
.....		
111110 1 0 1 0 0 11111 1*^{30}		$2^{30} \sim 2^{31}-1$
111110 1 0 1 0 1 00000 1*^{31}		$2^{31} \sim 2^{32}-1$
.....		
111110 1 0 1 0 1 111111 1*^{62}		$2^{62} \sim 2^{63}-1$
111110 1 0 1 1 0 1000000 1*^{63}		$2^{63} \sim 2^{64}-1$
.....		
111110 1 0 1 1 0 1111111 1*^{126}		$2^{126} \sim 2^{127}-1$
111110 1 0 1 1 1 10000000 1*^{127}		$2^{127} \sim 2^{128}-1$
.....		
111110 1 0 1 1 1 11111111 1*^{254}		$2^{254} \sim 2^{255}-1$
111110 1 1 1000 1 00000000 1*^{255}		$2^{255} \sim 2^{256}-1$
.....		
111110 1 1 1000 1 11111111 1*^{510}		$2^{510} \sim 2^{511}-1$
111110 1 1 1001 1 0000000000 1*^{511}		$2^{511} \sim 2^{512}-1$
.....		
111110 1 1 1001 1 1111111111 1*^{1022}		$2^{1022} \sim 2^{1023}-1$
.....		

また，００…０（語のすべてのビットが０）である語は，段数が無限大したがって指数が$-\infty$と解釈し，数値０とする．同様に，冒頭以外のビットがすべて１の語は，Ｓに応じて，数値$+\infty$または$-\infty$とする．これらにより，ＵＲＲの特徴(γ)，(δ)がみたされ，結局，(α)，(β)，(γ)，(δ)がすべて本方式でも満たされる．



(a)



(b)

図4　各方式の指数部の長さ

3．2　二重，三重指数分割方式との比較

変動型多重指数分割による方式，ＵＲＲ（２．１），２段化ＵＲＲ（２．２）が指数の表現に要するビット長を**図４**に示す．**図４**において（a）は（b）の左下の部分を拡大したものであり，（b）では細かい階段状の折れ線を描くのが不可能であるので斜めの直線で近似してある．また，語長が３２ビット，６４ビットのとき各方式が表し得る数値の範囲を**表１**に示す（ただし，語の右端からはみ出して表現できないビットは０とみなす）．

表１　表現可能範囲

	指数の最大値，最小値		
	本方式	２段ＵＲＲ	ＵＲＲ
32ﾋﾞｯﾄ	$2^{2^{28}/2}-1, -2^{2^{29}/2}-1$	$2^{2^{28}}, -2^{2^{29}}$	$2^{28}, -2^{29}$
64ﾋﾞｯﾄ	$2^{2^{60}/2}-1, -2^{2^{61}/2}-1$	$2^{2^{60}}, -2^{2^{61}}$	$2^{60}, -2^{61}$

4．Ｋｎｕｔｈの方式との比較

長さの上界が未知の自然数を表現する方式として，Knuth による方式がある[11]．

この方式では，自然数ｉの２進数表現を１α(i)とするとき，ｉをＢ(i)と表現する．ここで，Ｂ(i)は次のように再帰的に定義される．

Ｂ(0) ＝ ０，

Ｂ(i) ＝ １Ｂ(｜α(i)｜)α(i)　（ｉ≧１）

ただし，｜α(i)｜はビット列 α(i) の長さである．

変動型多重指数分割との違いはＥ部の解釈の違いである．すなわち，本論文の変動型多重指数分割においてＥ部を次のように解釈すれば，Knuth の方式と同じになる．

Ｅ：　Ｌ部が１０のときは空であり，値０を表す．Ｌ部が１１０のときは値１を表すが，けち表現のためＥは空である．Ｌ部が１１１０のときは長さが２ビットの１個のフィールドE_1だけから成るが，けち表現のため１ビットだけが表示され，値２または３を表す．Ｌ部が$1^{n+1}0$（ｎ≧２）のときはｎ－１個のフィールドE_1，E_2，…，E_{n-1}から成る．E_1はE_2

表2　変動型多重指数分割による方式と Knuth 方式との比較

変動型多重指数分割による方式	Knuth 方式	表現される値
10	10	0
110 1	110 1	1
1110 10 1*	1110 1*	$2^1 \sim 2^2-1$
1110 11 1**	11110 10 1**	$2^2 \sim 2^3-1$
11110 10 100 1***	11110 11 1***	$2^3 \sim 2^4-1$
11110 10 101 1****	111110 10 100 1****	$2^4 \sim 2^5-1$
11110 10 110 1*****	111110 10 101 1*****	$2^5 \sim 2^6-1$
11110 10 111 1******	111110 10 110 1******	$2^6 \sim 2^7-1$
11110 11 1000 1*^{7}	111110 10 111 1*^{7}	$2^7 \sim 2^8-1$
11110 11 1001 1*^{8}	111110 11 1000 1*^{8}	$2^8 \sim 2^9-1$
.....	.....	.....
11110 11 1111 1*^{14}	111110 11 1110 1*^{14}	$2^{14} \sim 2^{15}-1$
111110 10 100 10000 1*^{15}	111110 11 1111 1*^{15}	$2^{15} \sim 2^{16}-1$
111110 10 100 10001 1*^{16}	1111110 10 100 10000 1*^{16}	$2^{16} \sim 2^{17}-1$
.....	.....	.....
111110 10 100 11111 1*^{30}	1111110 10 100 11110 1*^{30}	$2^{30} \sim 2^{31}-1$
111110 10 101 100000 1*^{31}	1111110 10 100 11111 1*^{31}	$2^{31} \sim 2^{32}-1$
111110 10 101 100001 1*^{32}	1111110 10 101 100000 1*^{32}	$2^{31} \sim 2^{32}-1$
.....	.....	.....
111110 10 101 111111 1*^{62}	1111110 10 101 111110 1*^{62}	$2^{62} \sim 2^{63}-1$
111110 10 110 1000000 1*^{63}	1111110 10 101 111111 1*^{63}	$2^{63} \sim 2^{64}-1$
111110 10 110 1000001 1*^{64}	1111110 10 110 1000000 1*^{64}	$2^{63} \sim 2^{64}-1$
.....	.....	.....
111110 10 110 1111111 1*^{126}	1111110 10 110 1111110 1*^{126}	$2^{126} \sim 2^{127}-1$
111110 10 111 10000000 1*^{127}	1111110 10 110 1111111 1*^{127}	$2^{127} \sim 2^{128}-1$
111110 10 111 10000001 1*^{128}	1111110 10 111 10000000 1*^{128}	$2^{128} \sim 2^{129}-1$
.....	.....	.....
111110 10 111 11111111 1*^{254}	1111110 10 111 11111110 1*^{254}	$2^{254} \sim 2^{255}-1$
111110 11 1000 100000000 1*^{255}	1111110 10 111 11111111 1*^{255}	$2^{255} \sim 2^{256}-1$
111110 11 1000 100000001 1*^{256}	1111110 11 1000 100000000 1*^{256}	$2^{256} \sim 2^{257}-1$
.....	.....	.....
111110 11 1000 111111111 1*^{510}	1111110 11 1000 111111110 1*^{510}	$2^{510} \sim 2^{511}-1$
111110 11 1001 1000000000 1*^{511}	1111110 11 1000 111111111 1*^{511}	$2^{511} \sim 2^{512}-1$
111110 11 1001 1000000001 1*^{512}	1111110 11 1001 1000000000 1*^{512}	$2^{512} \sim 2^{513}-1$
.....	.....	.....

の長さを表し，E_2はE_3の長さを表し，…，E_{m-1}は指数を表す．ただし，E_1，E_2，…，E_{m-1}はいずれもけち表現を用い，E_1が表すのはE_2のけち表現で省略されるビットを含まない長さであり，E_2が表すのはE_3のけち表現で省略されるビットを含まない長さであり，…，等々．$m \geq 3$のときはE_1の長さは2ビット（ただし，けち表現のため冒頭の1ビットは表示されないので，実質的には1ビット）．

Knuth の方式を変動型多重指数分割による数値表現の指数部分と比較したものを**表2**に示す．

5．おわりに

指数部の分割の段数が可変な数値表現を提案し，精度を検討した．また，既存の浮動小数点方式と比較した．本方式は，表現する数値の絶対値が1から大きく離れるときＵＲＲや2段化されたＵＲＲより精度が改善される．しかし，**図4**からわかるように，数値の絶対値が現実的な範囲（語長が１２８ビットまで）ではＵＲＲや2段化されたＵＲＲと同等かそれらより劣るので，浮動小数点数としての実用性に関しては，さらに検討する必要がある．また，本方式を指数部についてだけ注目すると，大きな自然数を表現する方法とみなすこともできるので，暗号への応用も検討する価値があろう．

謝辞

本研究について討論いただいた日本電気株式会社枝廣正人氏，東京農工大学植村俊亮教授，高橋延匡教授，中川正樹助教授，並木美太郎助手に感謝します．本研究の一部は，財団法人大川情報通信基金の援助を受けた．

参考文献

(1) Stevenson, D. et al：“A proposed standard for binary floating-point arithmetic, Draft 8.0 of IEEE”, Computer, **14**, 3, 51-62 (1981).

(2) Kahan, W. and Palmer, J. ：“On a proposed floating-point standard”, ACM SIGNUM Newsletter, Special Issue, 13-21 (1979).

(3) 松井正一，伊理正夫：“あふれのない浮動小数点表示方式”，情報処理学会論文誌，**21**，4，303-313 (1980).

(4) 浜田穂積：“二重指数分割に基づくデータ長独立実数値表現法”，情報処理学会論文誌，**22**，6，521-526 (1981)；“同Ⅱ”，情報処理学会論文誌，**24**，2，149-156 (1983).

(5) 高田正之，西村恕彦：“浜田方式数値表現の変形とその評価”，情報処理学会第26回全国大会講演論文集，1231-1232 (1983).

(6) 中森眞理雄，土井孝：“三重指数分割による数値表現方式について”，電子情報通信学会論文誌 (A)，**J71**-A，7，1468-1469 (1988).

(7) 工藤聖一，伊達 悍：“３重指数型浮動小数点表現の効用”，情報処理学会第36回全国大会講演論文集，19-20 (1988).

(11) Knuth, D. E.：“Supernatural Numbers,” in Klarner, D. A. (ed.), The Mathematical Gardner, 310-325, Prindle Weber and Schmidt, Boston (1980).

(8) 森岳志，中川正樹，高橋延匡，中森眞理雄：“各種浮動小数点表現法の評価方式の実現”，情報処理学会論文誌，**29**，8，807-814 (1988).

(9) 中森眞理雄：“変動式多重指数分割による数値表現方式”，電子情報通信学会論文誌 (A)，**J71**-A（印刷中）(1989).

(10) Elias, P.：“Universal Codeword Sets and Representations of the Integers,” IEEE Trans. Information Theory, **IT-21**, 2, 194-203 (1975).

(12) 横尾英俊：“自然数の表現の立場から見た多重指数分割浮動小数点表示方式”，情報処理学会論文誌，**30**，6，792-794 (1989).