ハイパフォーマンス
コンピューティング 58－10
（１９９５．１０．１８）

拡張した二重指数分割表現による数値表現法に関する研究

富松 剛

東京大学大学院理学系研究化情報科学専攻

浜田の提案したURR(Universal Representation of Real numbers)は1から離れるに従って，急速に精度が悪化するという欠点がある．この欠点を改善するためにURRを一般化し，拡張した二重指数分割表現を提案する．URRは二重指数$\pm 2^{\pm 2^m}$で実数を大まかに近似するが，本表現では二重指数を$\pm p^{\pm q^m}$と一般化する．本表現により，pとqを大きくするとで，より大きな実数に対しては，より少ないビット数で近似できること，pとqを適切に選ぶことによって，URRに比べより広範囲にわたって，IEEE表現よりも精度の良い表現方法があることがわかった．さらに本表現はURRの欠点でもある急激な精度の悪化も抑えることがわかった．

A Study on Representation of Numbers Based on Extended Double Exponential Cut

Tsuyoshi Tomimatsu

Department of Information Science, Faculty of Science,
the University of Tokyo

Precision of URR (Universal Representation of Real numbers) which was proposed by Hamada goes to the worse when the number is away from ± 1. By generalizing from URR, we propose the floating point representation of the extended double exponential cut of URR. Our representation is based on the double exponential cut of $\pm p^{\pm q^m}$. By selecting suitable values for p and q, we can represent numbers more precise than the IEEE representation. And our floating point representation cut can escape from the demerit in URR representation of getting worse in the precision when the number is away from ± 1.

1 はじめに

数値計算におけるあふれの発生は昔から問題となっている．そのあふれを発生させない様に松井・伊里は指数部を可変とし，事実上あふれの生じない数値表現法を提案した [1]．しかし，その表現法は長さの違うフォーマット間において相互変換が困難となる欠点がある．

その後，あふれの生じない表現法として URR (Universal Representation of Real numbers) が浜田により提案された [2][3][4]．浜田の提案する URR の特徴は，(1) 指数部が可変であり，事実上あふれが発生しない．(2) 長さの違うフォーマット間での変換が容易である．(3) 大小関係が固定小数点と同じである．(4)1 付近の精度が非常に良い．しかし 1 から離れるにしたがって急速に精度が悪化すると言う欠点がある [5]．

そこで，URR を一般化することによって急激に精度が悪化する欠点を抑えた表現法の一つが見つかるかもしれないと考えた．

本報告では一般化した URR に付いて述べ，一般化した立場から見ると精度の悪化を緩やかにし，かつ URR よりも広範囲において，IEEE 表現よりも精度の良い表現方法があることを示す．また URR の簡単な説明と，その一般化の方法に付いて述べ，URR およびその他の表現との精度の比較も行う．

2 URR の一般化

2.1 URR の定義

浜田の提案した URR とは，ある区間を URR を表現するビット列に対応した一定の規則に基づいて分割を繰り返していき，表現すべき実数値が分割区間の上限に属すれば URR を表現するビット列に 1 を連結し，下限に属すれば同様にビット列に 0 を連結していく方法である．

以下具体的に述べる．任意のビット列 Sを実数値の半開区間に対応させる．

$$S : \{x | a \leq x < b\}$$

ビット列 Sに対応する区間の中の任意の実数値を $U(S)$ で表すとき次の関係となる．

$$a \leq U(S) < b$$

Sの右にビット 0 を連結したものを $S0$，ビット 1 を連結したものを $S1$ と表すことにする．S の形と a, b の値によって決まる値 c によって区間を二分し次の対応を行う．

$$a \leq U(S0) < c \quad , \quad c \leq U(s1) < b$$

以後，便宜上 k個の 0 の連結を 0^k，k個の 1 の連結を 1^kと表すことにする．

分割による区間の縮小は偏りなく行われるので，$k \to \infty$ における $U(S0^k)$ が区間の下限に収束することは明らかで，$a \leq U(S) < b$ のとき次の通りである．

$$\lim_{k \to \infty} U(S0^k) = a$$

2.1.1 符号部

符号部は二進整数における大小関係を考慮して以下の様に定義する．

$$\begin{aligned} -\infty \leq U(1) &< 0 \\ 0 \leq U(0) &< +\infty \end{aligned}$$

2.1.2 大まかな分割

続いて正負の区間を ± 1，± 2，$\pm 1/2$ で分割を行い，以下の範囲を得る．

$$\begin{aligned} -\infty \leq U(100) &< -2 \\ -2 \leq U(101) &< -1 \\ -1 \leq U(110) &< -1/2 \\ -1/2 \leq U(111) &< 0 \\ 0 \leq U(000) &< 1/2 \\ 1/2 \leq U(001) &< 1 \\ 1 \leq U(010) &< 2 \\ 2 \leq U(011) &< +\infty \end{aligned}$$

このとき，上限と下限の比が 2 の場合は等比分割に移る．そうでないものは次の二重指数分割に移る．

2.1.3 二重指数分割

以後，便宜上 2^{2^m} を $P^+(m)$，2^{-2^m} を $P^-(m)$ と表す記法を導入する．

$$\begin{aligned} -\infty \le U(10^{m+2}) &< -P^+(m) \\ -P^-(m) \le U(11^{m+2}) &< 0 \\ 0 \le U(00^{m+2}) &< +P^-(m) \\ +P^+(m) \le U(01^{m+2}) &< +\infty \end{aligned}$$

以上の各範囲をそれぞれ，以下の各点で分割する．

$$\begin{aligned} &-P^+(m+1) \\ &-P^-(m+1) \\ &+P^-(m+1) \\ &+P^+(m+1) \end{aligned}$$

最終的に以下の範囲を得ることが出来る．

$$\begin{aligned} -P^+(m+1) \le U(10^{m+2}1) &< -P^+(m) \\ -P^-(m) \le U(11^{m+2}0) &< -P^-(m+1) \\ +P^-(m+1) \le U(00^{m+2}1) &< +P^-(m) \\ +P^+(m) \le U(01^{m+2}0) &< +P^+(m+1) \end{aligned}$$

2.1.4 等比分割

以下の分割を上限と下限の比が 2 になるまで分割を行う．回数的には m 回分割を行えばよい．

$$\begin{aligned} a \le U(S0) &< a\sqrt{\tfrac{b}{a}} \\ a\sqrt{ba} \le U(S1) &< b \end{aligned}$$

2.1.5 等差分割

以下の分割を任意精度が得られるまで行う．

$$\begin{aligned} a \le U(S0) &< \tfrac{a+b}{2} \\ \tfrac{a+b}{2} \le U(S1) &< b \end{aligned}$$

以上の分割はビット列の生成規則を示すもので，必ずしも全ての分割を行う必要はなく，所望のビット数，もしくは精度が得られればそこで打ち切ってもよい．

2.2 拡張した二重指数分割表現

先に述べた URR の定義では，URR は 1 から離れるにしたがって，急速に精度が悪化していくという欠点がある．これは二重指数分割の回数 m が 1 大きくなると二重指数分割部，等比分割部，各々が 1 ビットずつ伸びるために精度は 2 ビットずつ悪化することによる．そこで URR を一般化した場合の m と二重指数分割部と等比分割部の振る舞いに付いて調べてみた．

拡張した二重指数分割表現 (以下拡張表現とよぶ) を定義する．拡張表現は URR を一般化したもので，URR では分割のベースが $\pm 2^{\pm 2^m}$ であったものを $\pm p^{\pm q^m}$ としたものである．

まず拡張表現におけるいくつかの関数に付いて定義する．拡張表現によるビット列 S に対応する区間の中の任意の実数値を $U_{p,q}(S)$ で表す．

また URR と同様に p^{q^m} を $P^+_{p,q}(m)$，p^{-q^m} を $P^-_{p,q}(m)$ と表記する．拡張表現における $p=2$，$q=2$ のときが URR となる．

2.2.1 符号部

符号部は二進整数を考えて以下の様に定義する．

$$\begin{aligned} -\infty \le U_{p,q}(1) &< 0 \\ 0 \le U_{p,q}(0) &< +\infty \end{aligned}$$

2.2.2 大まかな分割

続いて正負の各区間を ± 1，$\pm p$，$\pm 1/p$ で分割し，以下の範囲を得る．

$$\begin{aligned} -\infty \le U_{p,q}(100) &< -p \\ -p \le U_{p,q}(101) &< -1 \\ -1 \le U_{p,q}(110) &< -1/p \\ -1/p \le U_{p,q}(111) &< 0 \\ 0 \le U_{p,q}(000) &< 1/p \\ 1/p \le U_{p,q}(001) &< 1 \\ 1 \le U_{p,q}(010) &< p \\ p \le U_{p,q}(011) &< +\infty \end{aligned}$$

上限と下限の比が p のものは等差分割へ移る．そうでないものは第一の二重指数分割を行う．

2.2.3 第一の二重指数分割

$$
\begin{aligned}
-\infty \le U_{p,q}(10^{m+2}) &< -P^{+}_{p,q}(m) \\
-P^{-}_{p,q}(m) \le U_{p,q}(11^{m+2}) &< 0 \\
0 \le U_{p,q}(00^{m+2}) &< +P^{-}_{p,q}(m) \\
+P^{+}_{p,q}(m) \le U_{p,q}(01^{m+2}) &< +\infty
\end{aligned}
$$

以上の範囲をそれぞれ，以下の点で分割する．

$$
\begin{aligned}
&-P^{+}_{p,q}(m+1) \\
&-P^{-}_{p,q}(m+1) \\
&+P^{-}_{p,q}(m+1) \\
&+P^{+}_{p,q}(m+1)
\end{aligned}
$$

最終的に以下の範囲を得ることが出来る．

$$
\begin{aligned}
-P^{+}_{p,q}(m+1) \le U_{p,q}(10^{m+2}1) &< -P^{+}_{p,q}(m) \\
-P^{-}_{p,q}(m) \le U_{p,q}(11^{m+2}0) &< -P^{-}_{p,q}(m+1) \\
+P^{-}_{p,q}(m+1) \le U_{p,q}(00^{m+2}1) &< +P^{-}_{P,q}(m) \\
+P^{+}_{p,q}(m) \le U_{p,q}(01^{m+2}0) &< +P^{+}_{p,q}(m+1)
\end{aligned}
$$

2.2.4 第二の二重指数分割

$$
\begin{aligned}
+P^{+}_{p,q}(a) \le U_{p,q}(S) &< +P^{+}_{p,q}(b) \\
+P^{-}_{p,q}(a) \le U_{p,q}(S) &< +P^{-}_{p,q}(b) \\
-P^{-}_{p,q}(a) \le U_{p,q}(S) &< -P^{-}_{p,q}(b) \\
-P^{+}_{p,q}(a) \le U_{p,q}(S) &< -P^{+}_{p,q}(b)
\end{aligned}
$$

以上の各範囲をそれぞれ以下の各点で分割する．

$$
\begin{aligned}
&+P^{+}_{p,q}((a+b)/2) \\
&+P^{-}_{p,q}((a+b)/2) \\
&-P^{-}_{p,q}((a+b)/2) \\
&-P^{+}_{p,q}((a+b)/2)
\end{aligned}
$$

以上の分割は $\log_2 \log_2 q$ 回行う．

2.2.5 等比分割

以下の分割を上限と下限の比が p になるまで行う．

$$
\begin{aligned}
a \le U_{p,q}(S0) &< a\sqrt{\tfrac{b}{a}} \\
a\sqrt{\tfrac{b}{a}} \le U_{p,q}(S1) &< b
\end{aligned}
$$

2.2.6 等差分割

以下の分割を任意精度が得られるまで行う．

$$
\begin{aligned}
a \le U_{p,q}(S0) &< \tfrac{a+b}{2} \\
\tfrac{a+b}{2} \le U_{p,q}(S1) &< b
\end{aligned}
$$

拡張表現の分割の様子を図にしたものを図 1 に示す．

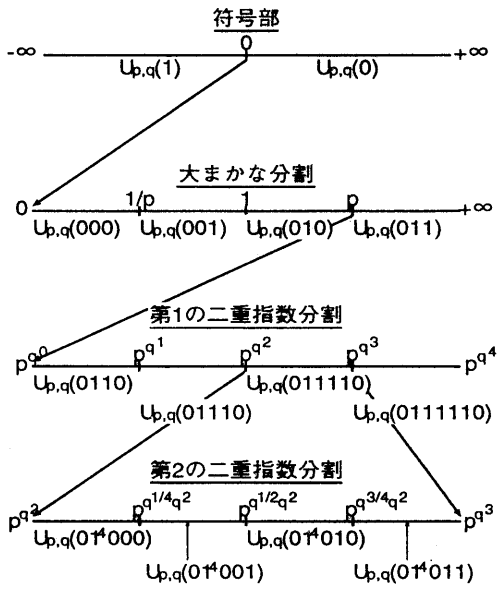

図 1: 拡張表現の分割の様子

3 拡張表現の精度

ここでは拡張表現の持つ精度に関して考察を行う．各分割に要するビット数を計算し，それから精度を求めることが出来る．まず大まかな分割から考える．

3.1 各分割段階で必要なビット数

大まかな分割では，その定義から明らかな様に符号部のビットも含めて，3 ビットが費される．

第一の二重指数分割は，定義から分割に要するビット数は $m+1$ ビットである．

続いて第二の二重指数分割部について考える．簡略化のため $a > 0$ の領域のみ考える．他の領域も同様の解析を行えばよい．

さて第一の二重指数分割で

$$P^{+}_{p,q}(m) \leq U_{p,q}(S) < P^{+}_{p,q}(m+1)$$

の範囲を得た場合，n 回目の第二の二重指数分割後の下限から $j(j = 0, \cdots, 2^n - 1)$ 番目の範囲は以下のようになる．

$$P^{+}_{p,q}\left(m + \frac{j}{2^n}\right) \leq U_{p,q}(S) < P^{+}_{p,q}\left(m + \frac{j+1}{2^n}\right)$$

また，上限と下限の比は以下のようになる．

$$\left\{P^{+}_{p,q}\left(m + \frac{j}{2^n}\right)\right\}^{q^{\frac{1}{2^n}} - 1} = p^{q^m q^{\frac{j}{2^n}}(q^{\frac{1}{2^n}} - 1)}$$

さて，上限と下限の比は等比分割での偏りのない分割を考慮すると，$p^{2^{n'}}$(n'は任意) の形をしていた方がよい．よって以下の条件を満たす必要がある．

$$q^m q^{\frac{j}{2^n}}(q^{\frac{1}{2^n}} - 1) = 2^{n'}$$

すなわち q^m，$q^{\frac{j}{2^n}}$，$q^{\frac{1}{2^n}} - 1$ は全て 2 のベキ乗でなくてはならない．この三条件から $q^{\frac{1}{2^n}}$ は 2 であることがわかる．また $n = \log_2 \log_2 q$ であることも導かれる．結局 n 回分割後の上限と下限は以下のようになる．

$$p^{q^m 2^j} \leq U(S) < p^{q^m 2^{j+1}}$$

続いて，上記の範囲が得られた場合の等比分割に要するビット数について考える．等比分割は上限と下限の比が p になるまで分割を行うので，l回分割を行ったあとの上限と下限は

$$p^{q^m 2^j (1 + \frac{k}{2^l})} \leq U(S) < p^{q^m 2^j (1 + \frac{k+1}{2^l})}$$

比は p でなくてはならないので以下のようになる．

$$p^{q^m 2^j 2^{-l}} = p$$

分割回数 lは以下のようになる．

$$l = j + m \log_2 q$$

等差分割は上限と下限の比が p から始まる．最初の $\log_2 p - 1$ 回の分割は上限と下限の比が 2 より大きく，URR で言う，等差分割部の精度には相当しない．そのため $\log_2 p - 1$ 回の分割だけ特別に，精度に寄与しない分割とする．

3.2 拡張表現の精度

以上より L ビット長の拡張表現で実数値 x を表現する際に精度に寄与する仮数部に割けるビット数は以下の通りになる．

$$\begin{cases} L - 3 - \log_2 p - j \\ \quad - \log_2 \log_2 q - m(\log_2 q + 1) & (x \geq p) \\ L - 3 - \log_2 p & (x < p) \end{cases}$$

L： 拡張表現の長さ

m： 第一の二重指数分割部の長さ

$$m = \left\lfloor \log_q \log_p x \right\rfloor$$

j： 第二の二重指数分割部の区間

$$j = \left\lfloor \log_2 q \left(\log_q \log_p x - m \right) \right\rfloor$$

すなわち $D = L - 3 - \log_2 p - j - \log_2 \log_2 q - m(\log_2 q + 1)$ とすると，拡張表現は $x \geq p$ の範囲においては Dビットの仮数部を持ち，2^{-D-1} の相対誤差で表現が可能といえる．

4 他の表現方法との比較

以上述べた様に，拡張表現は p と qの値を変化させることによって，多様な精度を持った表現が可能であることがわかる．そのため良い表現を選ぶ評価基準を設けねばならない．そこで以下の三点に注目することにする．

1. 拡張表現が IEEE 表現と交差する所での実数が最も大きい表現方法．
2. URR が IEEE 表現と交差する所で一番精度が良い表現方法．
3. IEEE 表現があふれを起こす所で一番精度が良い表現方法．

以上の評価基準は図 2に示した通りである．

これら三点に注目すると，一番悪化が抑えられ，かつ広範囲にわたって IEEE 表現よりも精度の良い表現方法を選ぶことが出来る．そこで，図 3の様に，ある実数値を表現する際の精度を x 軸に p，y軸に qそして，z 軸に精度をプロットして調べた．

その結果，先に挙げた三点において一番よい精度を保っていたものは $p = 4$，$q = 16$ であった．

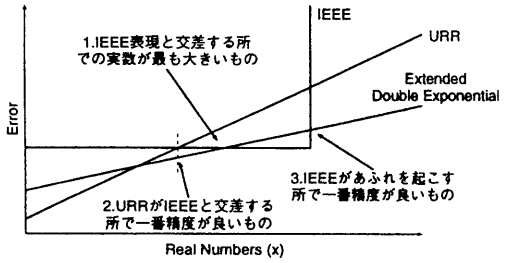

図 2: 評価基準

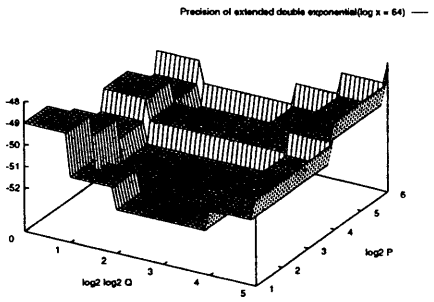

図 3: 2^{64}を表現する場合の精度

次にその $p = 4$, $q = 16$ の拡張表現の精度について，他の表現と比較したものが図4である．この図からわかる様に $p = 4$, $q = 16$ の拡張表現は URR と比較して 1 付近での精度は二箇所において，1 ビット精度が悪いが，URR に比べ IEEE 表現よりも精度のよい範囲が広いことがわかる．また精度は基本的に 1 ビットずつ悪化するので急速な精度の悪化が抑えられていることがわかる．

5 まとめ

本論文では URR を一般化した拡張した二重指数分割表現を提案した．この拡張した二重指数分割表現の立場から検証した結果，急速な精度の悪化を抑えた表現方法 ($p = 4, q = 16$) があることがわかった．また，この表現は URR よりも広範囲において IEEE 表現よりも精度がよいことがわかった．

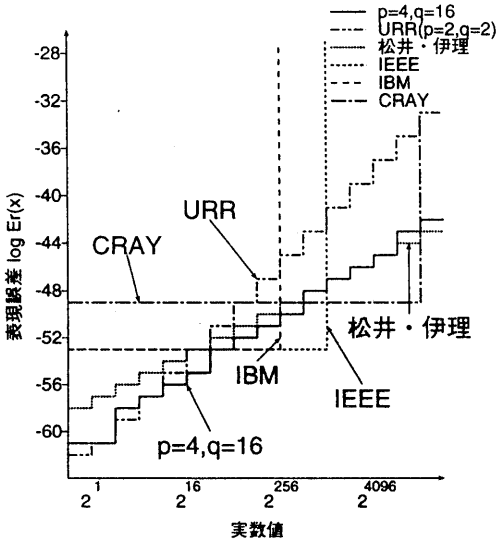

図 4: 精度の比較

今後の予定として，URR および $p = 4$, $q = 16$ のときの拡張表現 (必要ならばそれ以外の拡張表現) で，それぞれ大規模演算を行い有効性を検証する．

参考文献

[1] 松井, 伊理: あふれのない浮動小数点表示方式, 情報処理学会論文誌, Vol.21, No.4, pp.306-313(1980).

[2] 浜田: 新しい数値表現法 URR, 第 25 回プログラミングシンポジウム報告集, pp.84-91(1984).

[3] 浜田: 二重指数分割に基づくデータ長独立実数値表現法 II, 情報処理学会論文誌, Vol.24, No.2, pp.149-156(1983).

[4] 浜田: ディジタル・システムの数値表現法, Bit, Vol.20, No.3, pp.299-315.

[5] 中森, 土井: 3 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, Vol.J71-A, No.7, pp.1468-1469(1988).