数 値 解 析 30 － 5
（ 1 9 8 9. 1 0. 7 ）

平方根を近似する高次収束法

小沢 一文
仙台電波高専情報工学科

本論文では、平方根を近似する高次収束法のアルゴリズム群を提案し、それらの多倍長浮動小数点演算における計算効率について考察している。ここで提案されたアルゴリズム群は、収束の次数を任意に高くとれ初期値に無関係に収束するという特徴がある。このアルゴリズム群の反復関数はどれも有理関数になっていて、収束の次数を２とすればニュートン法になり、３とすればＢａｉｌｅｙ法となる。この有理関数を多倍長演算で効率的に計算する方法を提案し、時間計算量の解析を行っている。その結果、乗算に対する除算の時間が比較的小さいとき、５次収束法を二次因子に分解する方法が最も高速であり、逆に、除算にかなり時間を要するときは除算を含まない「逆平方根法」が最も高速であることが判明した。この結果は乗算および除算の方法に関係なく言えるものである。

A Family of Higher-Order Converging Algorithms for Approximating to Square Roots

Kazufumi Ozawa
Department of Information Technology, Sendai National College of Technology,
Kami-Ayashi, Aoba-Ku, Sendai 989-31, JAPAN.
e-mail:ozawa@sendai-ct.ac.jp

We derive a family of higher-order converging algorithms for approximating to square roots. This family includes the Newton and the Bailey iterations, and is globally convergent. The iteration functions of these algorithms are rational functions. In order to reduce the time-complexities in multiple-length arithmetic, we propose the various methods for the computations of the rational functions. Time-complexity analysis shows that the 5-th order algorithm by "square factor decomposition" is the fastest if the time ratio of division to multiplication is relatively small, and that the "reverse square root method" is the fastest if this ratio is considerably large. This result is independent of the methods of the multiplications.

1．まえがき

本論文では平方根に収束する高次収束法の解法群を導出し、その多倍長浮動小数点演算における計算法を工夫することによって、最も高速なアルゴリズムをその解法群の中から見出す。平方根の高精度計算のアルゴリズムに関しては、これまでにDutka J [1] が連分数展開を用いた２次収束法を提案し、それを用いて$\sqrt{2}$を100 万桁計算した。また、金田 [2] はそれを用いていっきに1600万桁計算したという実績がある。これに対して、小沢、海野 [3] はDutka J のアルゴリズムを高次のものへと拡張してきた。しかし、これらのアルゴリズムはどれも整数の平方根のみに適用可能であった。これに対して、本論文で提案するものは、任意の実数に適用可能であり、収束の次数も任意に高くとれるという特徴がある。

2．一次収束法の導出

はじめに、$a>0$の平方根に収束する一次収束列を導出する。まず次の差分方程式を考える：

（２．１）
$$U_m = 2uU_{m-1} - (u^2 - av^2)U_{m-2},$$
$$V_m = 2uV_{m-1} - (u^2 - av^2)V_{m-2}, \quad m=2,3,....$$

方程式(2.1) の一般解は、

（２．２）
$$U_m = c_{11}\lambda_1{}^m + c_{12}\lambda_2{}^m,$$
$$V_m = c_{21}\lambda_1{}^m + c_{22}\lambda_2{}^m, \quad m=0,1,....$$
$$\lambda_1 = u+\sqrt{a}v,\ \lambda_2 = u-\sqrt{a}$$

である。ここで、

（２．３）
$$uv>0$$

を仮定すれば、特性根の間に、

（２．４）
$$|\lambda_1| > |\lambda_2|$$

という関係が成立する。このとき、式（２．２）に現れる係数が、例えば

（２．５）
$$c_{11}=c_{12}=1/2,\quad c_{21}=-c_{22}=1/(2\sqrt{a})$$

を満たせば、数列$\{U_m/V_m\}$はu，vに無関係に$\sqrt{a}$に収束することが直ちに示される。条件(2.5) と矛盾しないU，Vの初期値は、

（２．６）
$$U_0=1,\ U_1=u,\quad V_0=0,\ V_1=v$$

である。以下、条件(2.4) が成り立ち、初期値を上式のように定めた場合のみを考察する。W_m の持つ誤差e_m は、式(2.2),(2.5) より

（２．７）
$$e_m = W_m - \sqrt{a} = 2\mu^m\sqrt{a}/(1-\mu^m), \quad m=1,2,...,$$

として評価される。ここで、$\mu=\lambda_2/\lambda_1$と置いた。この式より、

（２．８）
$$\lim_{m\to\infty} e_m/e_{m-1} = \mu$$

となり、W_m は$\sqrt{a}$に漸近的に一次収束することが示される。ここで、式(2.1) において非平方な整数aに対して、u，vを$u^2-av^2=\pm1$と選べば、式(2.1) はそれぞれaの連分数近似の中間近似分数における分母と分子を与えていることを付記しておく [4]。

3．高次収束法の導出

本章では、一次収束列(2.1) を加速し$\sqrt{a}$に高次収束するような数列を生成するアルゴリズムを導出する。式(2.2) に式(2.5) を代入すると、U_m，V_m は

（３．１）
$$U_m = \sum_i \binom{m}{2i} a^i u^{m-2i} v^{2i} \equiv A_m(u,v),$$

（3．2）　　$V_m = \sum_i \binom{m}{2i+1} a^i u^{m-2i-1} v^{2i+1} \equiv B_m (u, v)$

となる。ここで二項係数は、

$$\binom{i}{j} = 0, \quad i < j, \quad \text{または} \quad j < 0$$

と約束し、Σは二項係数が0にならない範囲でとるものとする。これより、$W_m = U_m / V_m$，m=1,2... を$W_1 = u/v$によって表すと、

（3．3）　　$W_m = G_m (W_1) = A_m (W_1) / B_m (W_1)$

となる。ここで、

（3．4）　　$A_m (x) = A_m (x, 1)$，　$B_m (x) = B_m (x, 1)$

である。有理関数Gは任意のr＞1について、

（3．5）　　$G_r (\sqrt{a}) = \sqrt{a}$,

（3．6）　　$G_r^{(j)} (\sqrt{a}) = 0$,　　$j = 1, 2, \ldots, r-1$,

（3．7）　　$G_r^{(r)} (\sqrt{a}) \neq 0$

を満たしていること、すなわち、有理関数G_rによって反復法

（3．8）　　$x_{k+1} = G_r (x_k)$，　$x_0 > 0$，$k = 0, 1, \ldots$,

を定義すると、数列$\{x_k\}$は$\sqrt{a}$に漸近的にr次収束することが示される　。また、このr次収束列と式(2.1)で定義される一次収束列との関係は、

（3．9）　　$x_k = W_{r^k} = U_{r^k} / V_{r^k}$，　　$k = 0, 1, 2, \ldots$,

である。以上の結果より、反復法(3.8)は$x_0 > 0$を満たす任意の初期値から出発しても$\sqrt{a}$に収束し、漸近的にr次収束することが示される。

ここで、r次収束法の例をいくつか示す：

(1) $r = 2$,　　$x_{k+1} = (x_k^2 + a) / (2 x_k)$　　；ニュートン法,

(2) $r = 3$,　　$x_{k+1} = (x_k^3 + 3 a x_k) / (3 x_k^2 + a)$　　；Ｂａｉｌｅｙ法,

(3) $r = 4$,　　$x_{k+1} = (x_k^4 + 6 a x_k^2 + a^2) / (4 x_k^3 + 4 a x_k)$.

4．高次収束法とその計算効率

本章では、r次収束法(3.8)の計算法を工夫しその時間計算量を求め、何次の解法が最も高速かを考察する。ここでは、式(3.8)のr次収束法の他に$1/\sqrt{a}$に収束する2次収束法

（4．1）　　$x_{k+1} = x_k - x_k (1 - a x_k^2) / 2$

を用いて$\sqrt{a}$の近似を求める解法についても考察する（この解法をここでは「逆平方根法」と名づける）。以下の考察では、初期値x_0として$\sqrt{a}$のｔ桁近似を用い、r次収束法をｐ回反復し$\sqrt{a}$のｓ（$s = r^p t$）桁近似を求めるのに要する時間計算量(time complexity) $T_r (t ; s)$を求める。演算は多倍長浮動小数点演算とし、演算の方式は常に一定の桁で演算する「固定長演算」と必要に応じて桁数の増減可能な「可変長演算」の２つとする。

4.1 固定長演算

ここで演算の桁数をＮとし、Ｎ桁の浮動小数点数の乗算一回当たりの時間計算量を$M (N)$と表し、r次収束法の反復一回当たりの実質的な乗算の回数をc_rで表すと、$T_r (t ; s)$は

（4．2）　　$T_r (t ; s) = p c_r M (N) = \log (s/t) (c_r / \log r) M (N)$

となる。この式より、上式のｒに依存する部分すなわち$c_r / \log r$が全時間を決定することになる。以下、多項式A_r，B_rの計算法をいくつか提案し、提案された方法の実質的な乗算の回数c_rを求め、

それらの時間計算量を比較検討する。

(1) Horner の方法

例えば、$r=2$ mのとき、$X=x^2$ と置いて、

（4．3）　$A_r(x)=(\cdots((X+a_1)X+a_2)\cdots+a_{m-1})X+a_m$,

（4．4）　$B_r(x)=x\{((\cdots(rX+b_1)X+b_2)\cdots+b_{m-2})X+b_{m-1}\}$

となる。ここで、rを単長数と仮定すれば多倍長数の乗算は$2m=r-2$回となる。同様な考察を行うことによって$r=2m+1$のときも$r-2$回の乗算が必要になることがわかる。その他、$X=x^2$ で二乗演算1回、$G=A/B$で除算1回を要する —— 二乗演算は多倍長数を表す桁の間の対称性を考慮すると乗算の半分の時間で計算できるのでここでは特別扱いする。したがって、c_r は

（4．5）　$c_r=r-2+0.5+d$

となる。ここで、dは除算の乗算に対する時間比とする。

(2)二次因子分解法

二次式 $X^2+\alpha X+\beta$を、$X^2+\alpha X+\beta=(X+\alpha/2)^2-\alpha^2/4+\beta$と変形し、$(X+\alpha/2)^2$を二乗演算を用いて計算すると、定数$\alpha/2$と$\alpha^2/4$ の計算に要する時間を無視できるような状況においては、この二次式はHornerの方法で計算するときの半分の時間で計算できることになる。この定数の計算のための時間が無視できるような状況とは、多くのXについて同じ二次式を何回も計算するときがこれにあたる。

ここでは多項式A_r，B_rを二次因子に分解し、各二次因子をいま述べた方法で計算し、計算量を減らす方法を提案し —— これをここでは「二次因子分解法」と呼ぶことにする —— 、その時間計算量の解析を行う。例えば、$r=4m+2$のとき多項式A_r，B_rを二次因子に分解すると、

（4．6）　$A_r(x)=X\prod_{i=1}^{m}(X^2+\alpha_i X+\beta_i)+a_{2m-1}$,

（4．7）　$B_r(x)=rx\prod_{i=1}^{m}(X^2+\gamma_i X+\delta_i)$,

であるから、反復一回当たり、$X=x^2$ の計算を含めて二乗演算$2m+1$回、乗算$2m-1$回、除算1回を要することがわかる。一般のrについて解析すると、

（4．8）　$r\equiv 0,2 \pmod 4$：$c_r=3r/4-1+d$,
　　$r\equiv 1 \pmod 4$：$c_r=3r/4-5/5+d$,
　　$r\equiv 3 \pmod 4$：$c_r=3r/4-3/4+d$

となる。

(3)Ｔｏｄｄの４次式 [5]

任意の4次式$X^4+aX^3+bX^2+cX+d$は、

$$X^4+aX^3+bX^2+cX+d=\{(X-\alpha)^2+\beta\}\{(X-\alpha)^2+X+\gamma\}+\delta$$

のように変形できる[5]。$r=8m+1$のとき、$A_r(x)$はx×（Xの4m次式），$B_r(x)$はXの4m次式となる。ただし$X=x^2$ である。A_r，B_rに含まれる4m次式を4次式の積に分解し、各4次式を二乗演算を用いて計算すると、$r=8m+1$のとき、

（4．9）　$c_r=5r/8-9/8+d$

となる。

(4)Ｋｎｕｔｈの方法 [5]

因数分解などの前処理を許すとき、多項式の計算の効率的な計算方法としてＫｎｕｔｈの方法というのがよく知られている。Ｋｎｕｔｈの方法を$r=7$の場合について説明する。$r=7$とすると、

（4．9）　$A_7(x)=x(X^3+21a\ X^2+35a^2X+7a^3)=x\ \{(\ Y+35a^2)(X+21a)-728a^3\ \}$，

（4．10）　$B_7(x)=7X^3+35a\ X^2+21a^2X+a^3=(7Y+21a^2)(X+5a)-104a^3$，

$X=x^2$，$Y=X^2$

となり、乗算3回、二乗演算2回、除算1回で反復一回が終了する。この方法では、先ほどのHornerの方法と比べ、二乗演算は1回多くなるが乗算が2回減るため、全体としては計算量が減少する。一般の場合について解析した結果は、

$$\begin{aligned} &r\equiv 0,\ 2\ (\mathrm{mod}\ \ 4) &&: c_r=r/2+1+d,\\ &r\equiv 1\ \ \ (\mathrm{mod}\ \ 4) &&: c_r=r/2+1.5+d,\\ &r\equiv 3\ \ \ (\mathrm{mod}\ \ 4) &&: c_r=r/2+0.5+d \end{aligned} \qquad (4.11)$$

である。

(5)Ｐａｔｅｒｓｏｎ－Ｓｔｏｃｋｍｅｙｅｒの方法[5]

この方法は、多項式の計算において定数倍の演算の時間計算量が無視できるとき、m次の多項式の計算がO（$\sqrt{m}$）の時間計算量でできるアルゴリズムである。例えば、r＝21の場合$Y=X^4$と置けば、

（4．12）　$A_{21}(x)=x\ \{(X^2+a_1X+a_2)\ Y^2+(a_3\ X^3+a_4X^2+a_5X+a_6)\ Y+(a_7\ X^3+a_8X^2+a_9X+a_{10})\}$，

（4．13）　$B_{21}(x)=(21\ X^2+b_1X+b_2)\ Y^2+(b_3\ X^3+b_4X^2+b_5X+b_6)\ Y+(b_7\ X^3+b_8\ X^2+b_9\ X+b_{10})$

である。ここでもしａが単長数であると仮定すると、上に示した各係数ａ，ｂも単長数となり、（　）の中で係数×Ｘのベキ（Ｘの各ベキは$Y=X^4$を計算する過程ですでに求まっている）という演算に要する時間は無視できる。上の2つの多項式をＹに関する多項式とみなしてHornerの方法で計算すると、乗算7回、二乗演算2回、除算1回でワンステップ終了する。これは前に示したHornerの方法に比べて大変な計算量の節約になる。一般のｒについて解析すると次のようになる：

$$\begin{aligned} &r\equiv 0\ (\mathrm{mod}\ \ 2): c_r=i+j-\delta_p-\delta_q+k+d,\\ &r\equiv 1\ (\mathrm{mod}\ \ 2): c_r=2\ (i-\delta_p)+k+d. \end{aligned}$$

$k=\lceil\sqrt{m+1}\rceil$，$i=\lfloor m/k\rfloor$，$j=\lfloor(m-1)/k\rfloor$，$p=m-ik$，$q=m-1-jk$

$\delta_p=1\ (p=0),\ 0\ (p>0)$.

4.2 可変長演算

本節では、ｒ次収束法(3.8)を可変長演算のもとで計算したときの時間計算量を求める。まず、ｒ次収束法(3.8)を次のように書き換える：

（4．14）　$x_{k+1}=x_k+\Delta_r(x_k)$，

$$\Delta_r(x_k)=\sum_i\left\{\binom{r}{i}-\binom{r}{2i+1}\right\}a^i\ x_k^{\,r-2i}\ /B_r(x_k)$$

いま、x_kが$\sqrt{a}$のｂ進ｑ桁の近似値であるとすると、x_kの持つ誤差e_kと増分Δ_rとの関係は、

（4．15）　$|e_K/\sqrt{a}|\fallingdotseq|\Delta_r(x_k)/x_k|=O(b^{-q})$

であるから、増分Δ_rの先頭桁はx_kのそれよりおおよそｑ桁だけ下位に位置していることがわかる。したがって、ｑ桁近似x_kに増分Δ_rを加えた値x_{k+1}をｒｑ桁近似にするためには、増分Δ_rは上位（ｒ－1）ｑ桁だけ計算すれば充分であることがわかる。しかしここでは余裕をみて$r^{1-\delta}$ｑ桁（$0\leqq\delta<1$）程度で計算することにする。このような計算を行ったとき、ｔ桁の初期値からｓ桁の近似を得るための時間計算量T_r（ｔ；ｓ）は、

（4．16）　$T_r(t;s)=c_r\sum_{i=1}^{p}M(r^{i-\delta}\ t)$

となる。ここで、M（N）はNの単長増加関数で、$0<\alpha<1$なる定数と任意のN＞1に対して常に、

（4．17）　M（αN）＜αM（N）

が成り立っているものとする。このとき、T_r（t；s）は

（4．18）　T_r（t；s）＜2c_rM（s）

を満たすことが容易に示される。この式は、可変長演算を用いて計算すると固定長演算の高々反復2回分の計算量で全ての計算が完了することを意味している。また、c_rが

（4．19）　c_r＜K（r－1），　r＞1

を満たしているとき、

（4．20）　T_r（t；s）＜KrM（s）

となり、rが小さいほど可変長演算では計算時間も少なくなることが示される。

〔数値例〕実際に多倍長演算を用いて平方根を計算してみる。ここでは演算桁数N＝32768 とし、乗算のアルゴリズムは時間計算量がO（N^2）となるものを用いた。この演算ルーチンにおける各演算の時間比は、二乗演算：乗算：除算＝0.49：1　：2.5 であった。ここではa＝2 とa＝0.3333‥‥（多倍長数）の2つの平方根を相対誤差がおおよそ2.6 ×10^{-8}となる初期値を与え計算した（図1　）。

参　考　文　献

1)Dutka, J:The Square Root of 2 to 1,000,000 Decimals. Math. Comp.Vol.25, pp.927-930(1971).

2)金田康正：円周率 800万桁及び1/sqrt(2),sqrt(2),1600万桁の検証・統計結果，数理解析講究録，Vol.498,pp.66-108(1983).

3)小沢一文，海野啓明：連分数を用いた平方根の高速発生法とその計算効率，情報処理学会アルゴリズム研究会資料，SIG-AL-3, pp.17-24(1989).

4)和田秀男：数の世界（科学ライブラリー），pp.169-175，岩波書店，東京(1981).

5)野崎昭弘：計算機数学（共立数学講座１１），pp.54-95,共立出版，東京(1984).

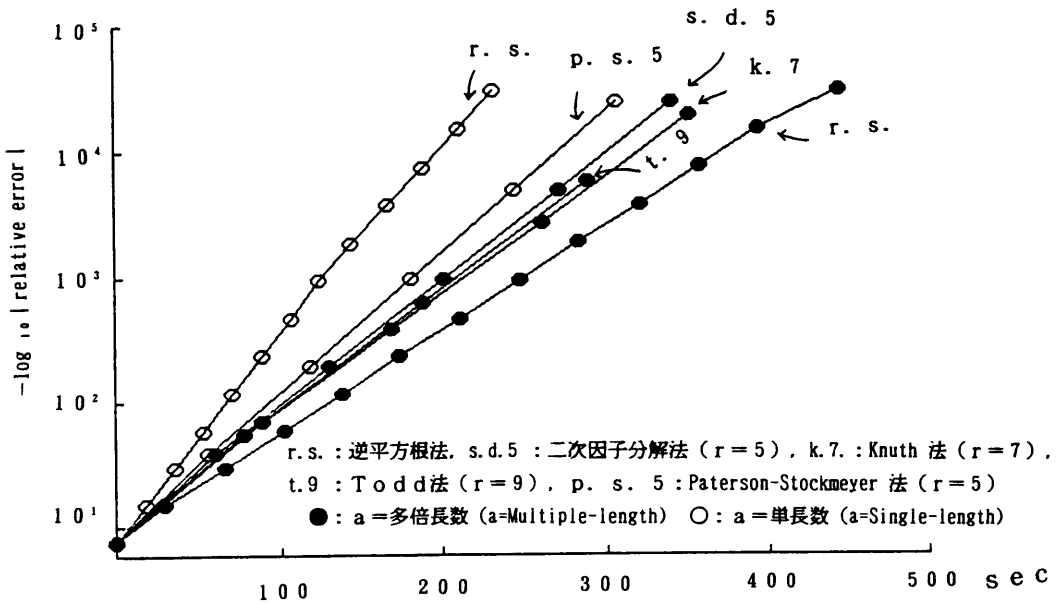

図1．r次収束法の相対誤差と計算時間の関係（固定長演算）