ハイパフォーマンス
コンピューティング 78－5
（1999．10．7）

連分数の高速評価方法

平山 弘[1]、 浮川 直章[2]

神奈川工科大学 [1]システムデザイン工学科、[2]機械システム工学科

hirayama@sd.kanagawa-it.ac.jp

本論文では、短い整数で構成された連分数を縮約法によって、十進数N桁の計算を $O(N(\log N)^3)$の時間で計算する高速計算法を提案する。この方法を使えば、桁数の大きな計算の場合、右田、天野、浅田、藤野によって報告された級数縮約法や相加相乗平均法と同等の時間で計算できる。また、この計算方法は、計算精度が小さい場合、従来より高速な計算法を与える。

Fast Multiple-precision Evaluation Method for Continued Fraction

Hiroshi Hirayama[1]　　Naoaki Ukigawa[2]

[1]Dept. of System Design Eng.　　[2]Dept. of Mechanical System Eng.

Kanagawa Institute of Technology

In this paper, we propose a fast algorithm for N digits calculation of continued fraction consist of short integer by recursively reducing adjacent terms of continued fraction in $O(N(\log N)^3)$ time. With this algorithm, calculation time become comparable to that of recursive reduction method for sum of series reported by T. Migita, A. Amano, N. Asada, S. Fujino and AGM method in case of large N. This method also gives faster algorithm than traditional one in case of small N.

1　はじめに

最近、右田等によって、級数を集約することによって、円周率を高速に計算する方法[1]（級数集約法）が提案されている。この方法は、Chudnovsky 兄弟による円周率 10 億桁の計算(1989 年)に使われたと言われる方法である。しかしながら、Chudnovsky 兄弟は、この計算方法の詳細を発表していない。

級数集約法[1]は、級数Sが、次の形で表現できるとき、すなわち

$$S=\frac{1}{C_0}(A_0+\frac{B_0}{C_1}(A_1+\frac{B_1}{C_2}(A_2+\frac{B_2}{C_3}(A_3+\frac{B_3}{C_4}(A_4+\cdots \qquad (1)$$

という形式で表されるとき、次のように変形することができる。

$$S=\frac{1}{C_0}(A_0+\frac{B_0}{C_1C_2}(A_1C_2+A_2B_1+\frac{B_1B_2}{C_3}(A_3+\frac{B_3}{C_4}(A_4+\cdots \qquad (2)$$

この式が正しいことは、簡単な計算で容易にわかる。（2）に変形された後も、（1）の形式を保つことから、このような変形は、式の途中で、何回でも行うことができる。（1）の A_i、B_i、C_iが桁数の少ない数値ならば、隣同士を上のような変形を高速に行えるので、結果として、（1）の式が高速に評価できることになる。これまでの円周率の計算でよく使われてきた算術幾何平均の方法と異なり、1000 桁以下

の比較的精度の低い場合でも、高速に計算できる。縮約の演算は独立に計算できるので並列計算も可能である。さらに、もし乗算にFFTを利用した乗算法を適用すれば、この集約による計算法の計算量は、

Nを計算桁数とすると通常$O(N^2)$に対し、$O(N(\log N)^3)$と非常に高速になる。

本論文では、このような関係式が連分数にも存在し、それを利用することによって、いろいろな連分数を高速で計算することができることを示す。

2 連分数の集約

級数と似たよう計算方法を、連分数に対しても行うことができる。この方法は、Lehmerが提案している、定数を高速に連分数展開する方法の逆計算に相当する。

有限項で打ち切った連分数を次のように記述する。

$$\frac{P_n}{Q_n} = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{\ddots + \cfrac{a_n}{b_n}}}} = b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n} \tag{3}$$

これら連分数は、良く知られているように、次のような漸化式[3]によって計算できる。

$P_{-1}=1$、$Q_{-1}=0$、$P_0=b_0$、$Q_0=1$として

$$\begin{aligned} P_n &= b_n P_{n-1} + a_n P_{n-2} \\ Q_n &= b_n Q_{n-1} + a_n Q_{n-2} \end{aligned} \qquad (n=1,2,\cdots) \tag{4}$$

によって、計算できる。

また、nからmまでの部分連分数を次のように定義する。

$$\frac{P_{n,m}}{Q_{n,m}} = \frac{a_n}{b_n} + \frac{a_{n+1}}{b_{n+1}} + \cdots + \frac{a_m}{b_m} \tag{5}$$

このように定義すれば、$P_n = P_{0,n}$、$Q_n = Q_{0,n}$である。この定義より

$$\frac{P_{l,m}}{Q_{l,m}} = \frac{a_l}{b_l} + \frac{a_{l+1}}{b_{l+2}} + \cdots + \frac{a_{n-1}}{b_{n-1}} + \frac{P_{n,m}}{Q_{n,m}} \tag{6}$$

が成り立つ。この式（6）に関係式（4）を利用して、

$$\begin{aligned} P_{l,m} &= Q_{n,m} P_{l,n-1} + P_{n,m} P_{l,n-2} \\ Q_{l,m} &= Q_{n,m} Q_{l,n-1} + P_{n,m} Q_{l,n-2} \end{aligned} \tag{7}$$

が得られる。この関係式は、mを$m-1$としても成り立つ。

$$P_{l,m-1} = Q_{n,m-1}P_{l,n-1} + P_{n,m-1}P_{l,n-2}$$
$$Q_{l,m-1} = Q_{n,m-1}Q_{l,n-1} + P_{n,m-1}Q_{l,n-2} \qquad (8)$$

（7）、（8）の式を何回も適用することによって、計算を進めることができる。

計算法としては、2項を組み合わせると便利である。最初の1組の分数を

$$\frac{P_0}{Q_0} = \frac{P_{0,0}}{Q_{0,0}} = \frac{b_0}{1} \qquad \frac{P_1}{Q_1} = \frac{P_{0,1}}{Q_{0,1}} = \frac{b_0 b_1 + a_1}{b_1} \qquad (9)$$

と置く。次に

$$\frac{P_{n,n}}{Q_{n,n}} = \frac{a_n}{b_n} \qquad \frac{P_{n,n+1}}{Q_{n,n+1}} = \frac{a_n}{b_n} + \frac{a_{n+1}}{b_{n+1}} \quad (n = 2,4,\cdots) \qquad (10)$$

を計算する。（9）と（10）値を使い、（7）、（8）式を使って、4項をまとめた式、さらに8項をまとめた式と次々計算する。最後に、1つの分数の組に集約される。この分数をわり算すれば、この連分数の値が計算される。（9）、（10）の計算および（7）、（8）の式による項をまとめる計算は、独立に計算できるので、並列にも計算可能である。

上の例では、各項は2項づつまとめたが、計算によっては、さらに多くの項をまとめた計算が効率的である場合も考えられる。

計算項数は、指数部を拡張した倍精度浮動小数点で計算し、収束条件を満たす項数を求める。収束判定には、関係式

$$\frac{P_n}{Q_n} - \frac{P_{n-1}}{Q_{n-1}} = (-1)^{n+1}\frac{a_1 a_2 a_3 \cdots a_n}{Q_n Q_{n+1}} \qquad (11)$$

を利用する。（4）の関係式を利用して、 $Q_i\ (i = 1,\cdots n+1)$ だけを計算し、（11）の分子を別に計算することによって、計算する。計算項数を求める方法として、（4）の漸化式を計算し、漸斤的に計算し計算項数を求める方法があるが、うまく漸斤的に計算できたとしても計算項数の計算が 5%程度の誤差があり、その分だけ、計算を多く行うことになる。この誤差を考えると、上のように直接計算してもほとんど時間差がないことから、直接計算する方法を使っている。

3 計算量

N 桁の数の乗算の計算量を $M(N)$ とする。$M(N)$ は、通常 $O(M(N)) = O(N^2)$ であるが、N が大きな桁数の数のとき、$O(M(N)) = O(N\log N)$ で計算できることが知られている。加減算の計算量は、$O(N)$、除算の計算量は、$O(M(N))$ である。加減算は、乗算に比べ計算量が少ないので無視する。除算は1回だけなので無視し、乗算の回数で計算量を推定する。

a_k、b_k の最大桁数を K、計算する項数を L とすると、（7）（8）の式から、L 回の K 桁の数値の乗算を必要となるので、計算量は、$CLM(K)$ となる（C は定数）。次の段階では、（6）の式を使うと、項数は $L/4$、乗算は各項について4回必要なので、計算量は、$CLM(2K)$ となる。次の段階では、計算

桁数は倍、計算項数は半分になるので、$CLM(4K)/2$ となる。従って、計算量は、$O(M(N))=O(N\log N)$ であることを考慮すると

$$CL\{M(K)+M(2K)+(1/2)M(4K)+(1/4)M(8K)+\cdots\}\approx 2CL(\log L)M(K)$$

が得られる。計算すべき項数Lと計算桁数Nには、ほぼ$L=kN$（kは定数）が成り立ち、さらに、$K=l\log N$（lは定数）が成り立つと仮定すると、計算量は、$O(N(\log N)^3)$となる。この計算量の評価で注意すべきことは、計算桁数Nが、かなり大きいときの評価であることである。計算する桁数が小さい場合、FFTを使った乗算より通常の乗算法が高速になるので、上の評価以上に高速になる。

4 連分数よる計算例

ここでは、連分数を集約法で計算した場合の具体例を示す。時間測定に使用したコンピュータは、メモリー384Mバイト PentiumⅡ450MHzである。コンパイラーとして、C++Builder3を使用した。

4.1 逆正接の計算

連分数展開式は、一般に、べき級数展開式に比較し、収束範囲が広い、収束が速い等の性質があることが知られている。たとえば、$\tan^{-1}x$のべき級数展開式は

$$\tan^{-1}x=x-\frac{x^3}{3}+\frac{x^5}{5}-\frac{x^7}{7}+\cdots \qquad (12)$$

と表現できる。また、連分数展開すると

$$\tan^{-1}x=\frac{x}{1+}\ \frac{x^2}{3+}\ \frac{4x^2}{5+}\ \frac{9x^2}{7+}\ \cdots\ \frac{n^2x^2}{2n+1+}\ \cdots \qquad (13)$$

が得られる。これらの式で、$x=1$として$\pi/4$を1万桁の精度で計算するとすると、（12）の式では、項数を非常に多く取らなければならないため、ほぼ不可能であるが、連分数展開式では、円周率の計算としては効率的ではないが、計算することは可能であり、十分高速である。べき級数として計算した場合、収束が遅いかったり、発散したりする場合、それに相当する連分数展開式が、収束し、実用的に十分な速さで計算できる場合がある。この場合、連分数で計算するのが適切である。

この円周率1万桁の計算を実際行うと、縮約を使った場合、2.96秒、（4）の通常の計算法で122.6秒であった。以下の時間も、この機械での測定値である。FFTの乗算法が効率的でない1000桁の計算でも、この縮約法が効率的で、通常の計算法が、1.18秒に対し、縮約法は、0.1秒であった。

4.2 Machinの公式による円周率の計算

他の公式との比較のために、円周率の計算によく使われるMachinの公式による計算を行う。

$$\pi=16\tan^{-1}\frac{1}{5}-4\tan^{-1}\frac{1}{239} \qquad (14)$$

この公式を使うために、（13）の式を次のように変形する。

$$\tan^{-1}\frac{1}{p}=\frac{1}{p+}\ \frac{1}{3p+}\ \frac{4}{5p+}\ \frac{9}{7p+}\ \cdots\ \frac{n^2}{(2n+1)p+}\ \cdots \qquad (15)$$

（15）の展開式を利用して、円周率を計算すると、5000桁で0.361秒、1万桁で0.951秒、10万桁

で 18.1 秒、20 万桁で 78.3 秒、50 万桁で 287.3 秒、100 万桁で 929.8 秒であった。100 万桁の計算では、公開されている他の円周率計算プログラムと比較して、数倍程度時間がかかる。プログラムに C++言語を使用しているため、非常に桁数の多いテンポラリーな変数のアロケーション、デアロケーションが頻繁に発生するためだと思われる。この計算で使っているプログラムもできるだけ変数のアロケーション、デアロケーションが起きないようにすこし改良したものである。

また、計算の中に同じ数値の乗算が含まれているので、FFT を呼び出す回数を減らし高速化することができると思われる。このような最適化をすれば、他の円周率の高速演算法と同程度速度を得られると思われる。

これまで連分数の高速な計算法が知られていなかったため、連分数展開を利用した高速な円周率計算公式の研究が行われていなかった。このため、連分数を使った効率の良い円周率計算公式が知られていないことも一因と思われる。連分数の研究が進めば、さらに高速な計算公式が開発され、さらに効率的な計算ができると思われる。

4.3 巡回連分数の計算

整数の平方根は、巡回連分数に展開できる。この性質を使うと、1 周期の部分を部分連分数として計算すれば、(7)、(8) により、2 周期の部分連分数が計算できる。さらに 4 周期、8 周期と次々と計算できる。このように、すべての部分連分数が同じになるため、計算量を大幅に減らすことができる。ここでは、具体例として、$\sqrt{2}$ の値を計算する。$\sqrt{2}$ は、次ように連分数展開される。

$$\sqrt{2}=1+\frac{1}{2+}\frac{1}{2+}\frac{1}{2+}\frac{1}{2+}\cdots \qquad (16)$$

この計算の場合、最初の定数 1 を除けば、途中の部分連分数は、すべて同じになるので、計算する必要がなくなり効率的に計算できる。この計算法で計算すると、1 万桁の計算で 0.17 秒であった。

4.4 いろいろな関数値の計算

対数も次の公式を利用して計算できる。

$$\log\left(\frac{1+x}{1-x}\right)=\frac{2x}{1-}\frac{x^2}{3-}\frac{4x^2}{5-}\frac{9x^2}{7-}\cdots\frac{n^2x^2}{-2n+1-}\cdots \qquad (17)$$

この式に $x=1/3$ を代入すると、$\log 2$ の値を計算できる。連分数の縮約法で計算すると 1 万桁を 1.31 秒で計算することができた。

上に示した式以外にも、e^x、$\sin^{-1}x$、$\tan x$ 等[3]の展開式がある。$\tan x$ はべき級数展開式と異なり、次のような簡単な式になる。

$$\tan x=\frac{x}{1-}\frac{x^2}{3-}\frac{x^2}{5-}\frac{x^2}{7-}\cdots\frac{x^2}{-2n+1-}\cdots \qquad (18)$$

このように係数が容易に推定できる形に展開できる。逆に、$\sin x$、$\cos x$ 等の連分数展開式は、係数が簡単な規則で得られる形には展開できない。

e^x は、連分数展開式でも簡単に書けて、

$$e^x=\frac{1}{1-}\frac{x}{1+}\frac{x}{2-}\frac{x}{3+}\frac{x}{2-}\frac{x}{5+}\frac{x}{2-}\frac{x}{7-}\cdots \qquad (19)$$

となる。また、縮約によって

$$e^x = 1 + \cfrac{2x}{2 - x + \cfrac{x^2}{6 + \cfrac{x^2}{10 + \cfrac{x^2}{14 + \cfrac{x^2}{18 + \cdots}}}}} \qquad (20)$$

と書くこともできる。計算の目的によって公式を選ばなければならない。

4.5 積分指数関数の計算

積分指数 $Ei(-x)$ は

$$-Ei(-x) = \int_x^\infty \frac{e^{-t}}{t} dt \qquad (21)$$

と定義される。この式を連分数展開すると

$$Ei(-x) = -\cfrac{e^{-x}}{x + \cfrac{1}{1 + \cfrac{1}{x + \cfrac{2}{1 + \cfrac{2}{x + \cfrac{3}{1 + \cfrac{3}{x + \cfrac{4}{1 + \cdots}}}}}}}} \qquad (22)$$

となり、収束するが、漸斤展開式では

$$Ei(-x) = -\frac{e^{-x}}{x}\left(1 - \frac{1!}{x} + \frac{2!}{x^2} - \frac{3!}{x^3} + \frac{4!}{x^4} - \cdots\right) \qquad (23)$$

となり、発散級数になる。発散級数では、ある程度までの精度では計算できるが、任意の精度までは計算できない。このような場合、連分数展開式は大変便利なものである。

5 まとめ

連分数は、それを構成している数値の桁数が小さな数値ならば、N 桁計算するには、漸近的に $O(N(\log N)^3)$ の計算量で計算する方法を提案した。この方法は、漸近的に高速であるだけでなく、FFT による乗算法が効率的でない 10 進数で数百桁の計算でも効率である。連分数を高速に計算できるので級数では計算しにくい関数も効率的に計算できるようになると期待できる。

6 参考文献

[1] 右田剛史、天野晃、浅田尚紀、藤野清次：級数の集約による多倍長数の計算法とπの計算への応用、情報処理学会研究報告、vol.98, No. 74, pp.31-36(1998)

[2] D. H. Lehmer, Euclid's Algorithm for Large Numbers, Amer. Math. Monthly, Vol. 45, pp.227-233(1938)

[3] L.Lorentzen, H.Waadeland : Continued Fractions with Applications, North-Holland, Amsterdam(1992)