ハイパフォーマンス
コンピューティング 79－2
（1999．12．3）

Euler 定数の多倍長高速計算法

平山 弘　佐藤 創太郎

神奈川工科大学　機械システム工学科

右田等によって提案された縮約法は、円周率だけでなく多くの計算に適用できる。ここでは、縮約法を 10 進数で N 桁の Euler 定数の計算に適用すると、計算量が $O(N(\log N)^3)$ の計算法になり効率的な計算法になることを示した。実際に、C++言語の高精度ルーチンを使い計算し、非常に効率的な計算法であることを示した。

Fast Calculation Method for High Precision Euler's Constant

Hiroshi Hirayama and Soutarou Satou

Kanagawa Institute of Technology

Recursive reduction of series, proposed by Migita et. al., can be applied to not only high precision computation of π but also many high precision computation. In this paper, it is shown that N digits calculation time for Euler's constant $\gamma = 0.577\cdots$ is $O(N(\log N)^3)$ and its method is very effective. Some numerical computation is down to show that this calculation method is very effective.

1. はじめに

Euler 定数は、

$$\gamma = \lim_{n\to\infty}\left(\sum_{k=1}^{n}\frac{1}{k} - \log n\right) = 0.5772156649015328606065l\cdots$$

と定義されている定数である。多くの重要な数学関数、たとえば、gamma 関数や Bessel 関数などに使われている重要な定数である。この定数は、Euler によって、1735 年に 0.577218 と計算され、1887 年に、天文学者 J.C.Adams によって、263 桁まで計算されている。

コンピュータを使った計算では、1952 年に Wrench によって、328 桁まで計算され、1962 年に D.E.Knuth によって、1271 桁まで計算されている。

同年 D.W.Sweeney[4]によって、 指数積分関数を利用する Bernoulli 数を必要としない計算方法が提案され、3683 桁までの計算が行われている。この計算方法を改良し、1974 年 W.A.Beyer と M.S.Waterman[3]が 7114 桁までの計算を行っている。1977 年に、R.P.Brent[2]は、 W.A.Beyer と M.S.Waterman の方法を一般化して、20,700 桁の計算を行っている。1980 年に、R.P.Brent と E.M.McMillan[1]は Bessel 関数を利用する新しい計算法を複数提案し、その 1 つを計算法を使い 30,100 桁の計算を行っている。これらの計算法は、計算する桁数を N とすると、効率の良い計算法でも $O(N^2)$ の計算時間を必要とし、円周率の計算に比べ非常に遅いものであった。

本論文では、 R.P.Brent が提案している計算方法の一つに、右田等[6]が提案している計算法を適用すると、計算量は円周率と同等のオーダー $O(N(\log N)^3)$ で計算できることを示す。

以下の計算には、コンパイラとして、C++Builder ver. 4、計算機としてパーソナル・コンピュータ(Pentium II 450MHz)を利用した。

2. 計算方法

右田等[6]は、円周率の新しい計算法（縮約法）を提案している。これは、級数Sが、次の形で表現できるとき、すなわち

$$S = \frac{1}{C_0}(A_0 + \frac{B_0}{C_1}(A_1 + \frac{B_1}{C_2}(A_2 + \frac{B_2}{C_3}(A_3 + \frac{B_3}{C_4}(A_4 + \cdots \quad (2.1)$$

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。

$$S = \frac{1}{C_0}(A_0 + \frac{B_0}{C_1 C_2}(A_1 C_2 + A_2 B_1 + \frac{B_1 B_2}{C_3}(A_3 + \frac{B_3}{C_4}(A_4 + \cdots \quad (2.2)$$

と変形できる。ここで、これらの級数を構成している数値が、桁数の小さな数値であるとする。この場合、この計算法を、再帰的に利用すると、高速に計算できる。計算する桁数を N としたとき、N 桁の計算するのに必要な項数と級数に現れる数値の最大値が N に比例する場合、$O(N(\log N)^3)$ で計算できることが知られている。縮約法の計算時間評価では、すべて、乗算は FFT を使って行うとしているが、実際の計算では、ある桁数以上ならば FFT を使った乗算法を使い、それ以下ならば通常の計算法を使った。精度が低い場合、FFT を使った乗算法より、通常の乗算の方が高速であるから、実際の計算は、さらに高速である。

Euler 数については、いろいろな計算方法が知られているが、 Brent 等（ R.P.Brent と E.M.McMillan[1] ）が、最も効率が良いとして提案している方法は、次の公式を利用するものである。この公式を利用すると、それまでの知られている計算法の約半分の計算時間で計算できる。

$$
\begin{aligned}
K_0(x) &= -\left\{\log\left(\frac{1}{2}x\right)+\gamma\right\}I_0(x)+\left\{\frac{\frac{1}{4}x^2}{(1!)^2}+\left(1+\frac{1}{2}\right)\frac{\left(\frac{1}{4}x^2\right)^2}{(2!)^2}+\left(1+\frac{1}{2}+\frac{1}{3}\right)\frac{\left(\frac{1}{4}x^2\right)^3}{(3!)^2}+\cdots\right\} \qquad (2.3)\\
&= -\left\{\log\left(\frac{1}{2}x\right)+\gamma\right\}I_0(x) + S(x)
\end{aligned}
$$

ここで、

$$S(x)=\sum_{k=0}^{\infty}\frac{\left(\frac{x}{2}\right)^{2k}}{(k!)^2}H_k \qquad (2.4)$$

$$H_m=1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{m}=\sum_{k=1}^{m}\frac{1}{k} \qquad (2.5)$$

$I_0(x)$および$K_0(x)$は、それぞれ、第１種変形ベッセル(Bessel)関数、第２種変形ベッセル関数である。

$I_0(x)$および$K_0(x)$は、それぞれ、第１種変形ベッセル(Bessel)関数、第２種変形ベッセル関数である。

$$I_0(x)=1+\frac{x^2}{2^2}+\frac{x^4}{2^2\cdot 4^2}+\frac{x^6}{2^2\cdot 4^2\cdot 6^2}+\cdots=\sum_{k=0}^{\infty}\frac{\left(\frac{x}{2}\right)^{2k}}{(k!)^2} \qquad (2.6)$$

また、xが大きな数値の場合、$K_0(x)$は次のように漸近展開される。

$$
\begin{aligned}
K_0(x) &\approx \sqrt{\frac{\pi}{2x}}e^{-x}\left\{1+\frac{-1}{8x}+\frac{(-1)\cdot(-9)}{2!(8x)^2}+\frac{(-1)\cdot(-9)\cdot(-25)}{3!(8x)^3}+\cdots\right\} \\
&=\sqrt{\frac{\pi}{2x}}e^{-x}\sum_{k=0}^{\infty}(-1)^k\frac{[(2k)!]^2}{(k!)^3(32x)^k}
\end{aligned} \qquad (2.7)
$$

ベッセル関数$I_0(x)$および$K_0(x)$は、縮約法を適用できるが、上の式を見ればわかるように、(2.3)の$S(x)$は、(2.1)の形をしていない。このため、この公式に、縮約法は適用することができない。このため、Brent等がその前に提案している計算方法を適用する。これは、

$$\gamma=S(n)-R(n)-\ln(n) \qquad (2.7)$$

ここで、

$$S(n)=\sum_{k=1}^{\infty}\frac{n^k(-1)^{k-1}}{k!k} \qquad (2.8)$$

$$R(n)=\int_n^{\infty}\frac{\exp(-u)}{u}du\approx\frac{\exp(-n)}{n}\sum_{k=0}^{\infty}k!(-n)^{-k} \qquad (2.9)$$

である。ここで、nは正の整数である。(2.8)の漸近展開式と$R(n)$との差は、次のように評価される。

$$\left| R(n) - \frac{\exp(-n)}{n}\sum_{k=0}^{n-2} k!(-n)^{-k} \right| < 3\exp(-2n) \tag{2.10}$$

nを十分大きな整数にすれば、漸近展開式を評価することによって、任意の精度で計算できる。

また、(2.10)と同じ程度の誤差になるまで、級数の$S(n)$の計算しなければならない。その項数は、Stirlingの公式を利用して、評価すると、項数Mは

$$M = \lfloor \alpha n \rfloor \tag{2.11}$$

ここで、$\alpha + 2 = \alpha \ln(\alpha)$、$\alpha = 4.3191\cdots$である。計算したい桁数を 10 進数で$N$としたとき、

$$n = \frac{1}{2} N \ln(10) \tag{2.12}$$

と選べば、10 進数でN桁の Euler 定数が計算できる。

3. 計算の詳細

3.1 $\log(n)$の計算

$\log(n)$の計算を高速に計算するために、$n = 2^l 3^m 5^n$と表される整数になるように選ぶ。このようにxを選ぶと、$\log\frac{16}{15}$、$\log\frac{25}{24}$、$\log\frac{81}{80}$の計算結果から容易に、$\log n$を計算できる。上の対数は、たとえば、

$$\log\frac{16}{15} = \log\frac{1+\frac{1}{31}}{1-\frac{1}{31}} = 2\left\{\frac{1}{31} + \frac{1}{3\cdot 31^3} + \frac{1}{5\cdot 31^5} + \cdots\right\} \tag{3.1}$$

として、効率的に計算できる。他の二つの対数も同様に変形でき効率的に計算できる。さらに、7、11 の倍数に対しても同様な計算ができるが、あまり計算量を減らすことができないので、この方法を使った。この級数は、縮約法が適用できるのでさらに効率的に計算できる。

3.2 $S(x)$の計算

(2.7)式で与えられる$S(x)$の計算にも、縮約法を適用して計算した。この場合、計算する項数を求めなければならない。(2.11)の式によって、求め使用している。

3.3 e^xの計算

e^xの計算は、定数eを展開式

$$e^{-1} = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \tag{3.2}$$

で計算し、それの二乗を次々計算し、nを 2 進数展開し、これらの数値を掛けることによって計算した。この級数計算にも縮約法が適用できるので、能率的に計算できる。 Brent 等[1]では、この部分も$S(n)$の計算と同様に、級数展開式にnを代入することを想定して計算量を求めている。自然対数の底eではなく、$e^{\frac{1}{2}}$を縮約法で計算すれば高速になる。しかし、eを求めるために乗算の回数が 1 回増加する。全体として計算量が減るようだっ

たら、これを使えばよい。$\frac{1}{2}$だけでなく、$\frac{1}{4}$とかいろいろな数値が考えられるが、どれが最も良い方法であるか、検討を要する。

4. 計算の詳細

計算性能を見るために、縮約法を使った計算と、Brent の方法で計算させ、その時間を測定した。計算には、平山[5]の C++言語のプログラムを改造して利用した。その表 1 を以下に示す。

本論文で提案した、縮約法は、精度が高くなるにつれて、精度にほぼ比例して時間がかかっている。このことから、計算時間が$O(N(\log N)^3)$程度であることを読みとれる。精度が低い場合は、精度が 2 倍になると、3 倍位の時間がかかり、$O(N^2)$の計算時間に近づくことがわかる。

表 1　Euler 定数の計算時間（sec）

計算桁数N	n	本計算法	Brent の計算法
440	1024	0.250	1.352
890	2048	0.711	5.098
1700	4096	1.742	19.098
3500	8192	4.676	79.774
7100	16384	13.650	318.658
14000	32768	32.837	1261.640
28000	65536	74.137	3368.010

Brent 方法は、計算精度が 2 倍になると計算時間は 4 倍となり、$O(N^2)$の計算時間がかかることがわかる。

Brent の方法と比較すると、本方法は精度が低いときでも、高速であることがわかる。これは、円周率を幾何相乗平均を使って計算する計算を行う。

ここでは、計算は 28,000 桁までの結果しか示していないが、本方法では、56,000 桁までの計算では、182.082 秒、116,000 桁の計算で、550.652 秒の計算時間を要した。10 万桁程度の計算になると、計算回数が増加し、それに伴って、高精度数のアロケーション、デアロケーションが頻繁に起こる。そのために計算時間が増加したもの考えられる。このような現象が起きないように、メモリ管理プログラム付きの高精度計算ルーチンを開発中である。

ハードウエア、ソフトウエア共に同じ環境で円周率を計算すると、Euler 定数の計算の約 5 分の 1 の時間で計算できた。

5. 終わりに

Euler定数は、Bessel関数や指数積分関数など応用上重要な関数の計算に現れる定数である。円周率と異なりこれまで高速な計算法が知られていなかったので、高精度計算はあまり行われていない。

今回の提案した計算法では、円周率と同程度のオーダーの計算量でEuler定数を計算できることを示した。

参考文献

[1] Brent R.P., McMilan E.M., “Some New Algorithms for High-Precision Computation of Euler’s Constant”, Math. Comp. Vol.34, 1980,pp.305-312

[2] Brent R.P., “Computation of Regular Continued Fraction for Euler's constant “, Math. Comp. Vol.31, 1977, pp.771-777

[3] Beyer W.A., Waterman M.s., “Error Analysis of Compution of Euler’s Constant”, Math. Comp. Vol.28, 1974, pp.599-604

[4] Sweeney D.W., “On the Compuation of Euler’s Constant”, Math. Comp. Vol.17, 1963, pp.170-178

[5] 平山　弘, " C++言語による高精度計算パッケージの開発",日本応用数理学会, Vol. 5, No.3, pp. 123-134, 1995

[6] 右田,天野,浅田,藤野,"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