

級数の再帰的集約による多倍長数の計算法と π の計算への応用

右田 剛 史[†] 天野 晃[†]
浅田 尚 紀[†] 藤野 清 次^{††}

数学定数である円周率 π の多数桁の計算法として、級数展開を用いる方法と Gauss-Legendre の公式などを用いる反復計算法が知られている。従来、 N 桁の π の値を得る計算量は、級数を用いると $O(N^2)$ 、反復計算法を用いると $O(N(\log N)^2)$ とされ、 N が大きいときには反復計算法の方が有利であると考えられていた。本稿では、まず級数計算が 2×2 行列の積として表現できることを示し、次に隣接する 2 つの行列の積を再帰的に計算することによって級数を集約し、多倍長乗算を利用して級数の和を $O(N(\log N)^3)$ の計算量で求める方法について述べる。本手法を用いることによって、桁数 N が大きくなっても、級数による計算が反復計算法と同等の時間でできることを確認した。また、3.2 万桁から 5.3 億桁の π の計算時間の変化を調べた結果、級数の和を求める Chudnovskys の公式は、反復計算による Gauss-Legendre の公式よりも高速に計算できることが分かった。

Recursive Reduction of Series for Multiple-precision Evaluation and Its Application to Pi Calculation

TSUYOSHI MIGITA,[†] AKIRA AMANO,[†] NAOKI ASADA[†]
and SEIJI FUJINO^{††}

A multiple-precision constant π of N digits is so far calculated by sum of series in $O(N^2)$ computation time as well as by using iterational algorithm in $O(N(\log N)^2)$ time. In this paper, we first prove that the series for π calculation can be represented as a product of 2×2 matrices, and then propose a fast algorithm for calculating sum of series in $O(N(\log N)^3)$ time by recursively reducing the adjacent matrices. Using this algorithm, computation time of sum of series becomes comparable to that of iterational algorithm, and experimental results on calculating 32,000 to 530 million digits of π have shown that the sum of series using Chudnovskys formula is computed faster than the iterational algorithm using Gauss-Legendre algorithm.

1. はじめに

数学定数である円周率 π の多数桁の計算法として、級数展開を用いる方法（以下、級数法と呼ぶ）と、Gauss-Legendre の公式などを用いる反復計算法（以下、反復法と呼ぶ）が知られている。表 1 にそれらの代表的な公式を示す^{1)~4)}。

表 1 の級数のように線形収束する級数の場合、 N 桁の精度で π を求めるには、計算すべき項数を L とすると、1 項ずつ小数に展開して足し合わせる単純な計算法では計算量は $O(LN)$ となり、 L は N に比例するので

計算量は $O(N^2)$ となる。これに対して反復法の場合、 N 桁の精度で π を求める計算量は $O(N(\log N)^2)$ である^{*}。このため、級数法よりも Gauss-Legendre の公式や Borweins の 4 次収束の公式などの反復法を用いる方がはるかに計算量が少ないとされ、最近の π の近似値計算では反復法の使用が主流となっている^{1),4)}。

しかし、文献 1) には、具体的な計算方法は示されていないが、級数法の計算量を $O(N(\log N)^3)$ にすることが可能であると示唆されている。本稿では、式 (1) の形式で表される級数の和の計算が複数個の 2×2 行列の積に帰着できることを示し、文献 5) の recursive doubling と同様の考え方を用いて級数を再帰的に集約することにより、級数の和を $O(N(\log N)^3)$ の計算量で求める方法について述べる。ただし、recursive

[†] 広島市立大学情報科学部知能情報システム工学科
Department of Intelligent Systems, Faculty of Information Sciences, Hiroshima City University

^{††} 広島市立大学情報科学部情報工学科
Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University

^{*} N 桁の多倍長乗算の計算量を $O(N \log N)$ とすると、反復法による N 桁の π の計算量は $O(N(\log N)^2)$ となる^{1),2),4)}。

表1 π の計算公式 (上段: 級数法, 下段: 反復法)Table 1 Formulas for π calculation by series (upper) and iterational algorithms (lower).

Machin	$\pi = 16 \arctan \frac{1}{5} - 4 \arctan \frac{1}{239}$, $\arctan x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{2k+1}$
Ramanujan	$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)! [1103 + 26390n]}{(n!)^4 396^{4n}}$
Chudnovskys	$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k (6k)! 13591409 + 545140134k}{(3k)!(k!)^3 (640320)^{3k+3/2}}$
Gauss-Legendre	$\left\{ \begin{array}{l} \text{初期値: } A_0 = 1, B_0 = \sqrt{1/2}, C_0 = 1 \\ \text{反復: } A_{k+1} = (A_k + B_k)/2, B_{k+1} = \sqrt{A_k B_k}, C_{k+1} = C_k - 2^k (A_k - B_k)^2 \\ 4A_k B_k / C_k \text{ が } \pi \text{ に収束する.} \end{array} \right.$
Borweins	$\left\{ \begin{array}{l} \text{初期値: } A_0 = 6 - 4\sqrt{2}, Y_0 = \sqrt{2} - 1, \\ \text{反復: } Y_{k+1} = \frac{1 - (1 - Y_k^4)^{1/4}}{1 + (1 - Y_k^4)^{1/4}}, A_{k+1} = A_k (1 + Y_{k+1})^4 - 2^{2k+3} Y_{k+1} (1 + Y_{k+1} + Y_{k+1}^2) \\ A_k \text{ が } 1/\pi \text{ に収束する.} \end{array} \right.$

doubling⁵⁾、級数の和を単精度や倍精度で求める場合に、隣接する項の計算を並列に行うことで高速化(計算時間を短縮)するのに対して、本稿で提案する手法は、各項が有理数である級数の和の多倍長計算を行う場合に、隣接項の計算が桁数の大きな偏りを生じないことに着目して高速化(計算量を低減)するという違いがある。

本手法が対象とする級数の一般形を次に示す。

$$S_L = \sum_{k=0}^{L-1} \frac{\prod_{l=0}^{k-1} B_l}{\prod_{l=0}^k C_l} A_k \quad (1)$$

$$= \frac{1}{C_0} \left(A_0 + \frac{B_0}{C_1} \left(A_1 + \frac{B_1}{C_2} \left(A_2 + \dots \right) \right) \right) \quad (2)$$

ここで、 A_k, B_l, C_l は比較的小さな整数(数十桁以下)であり、 $|B_l| < |C_l|$ とする。

表1の π の級数法の公式は無限級数であるが、 π の近似値を有限桁の級数として求める場合、いずれの級数も式(1)を含む形に変形できる。したがって、本手法を用いることにより π の級数法による計算を反復法と同程度の時間で行うことが可能となる。また、本稿で述べる級数の計算法は、 π の計算だけでなく三角関数や指数関数など多くの初等関数に適用可能であり、その応用範囲はきわめて広い。

以下、2章では級数の行列表記とその集約について述べ、3章では級数の集約を π の計算に適用し反復法と比較した結果について述べる。また、本手法を自然対数の底 e の計算に適用した結果についても述べる。

2. 級数の集約

式(2)に対する従来の計算法は次のようなものである。多倍長の小数 $1/C_0$ を初項とし、順次 B_k を掛け C_{k+1} で割ることによって次の数列を作る。

$$\left\{ \frac{1}{C_0}, \frac{1}{C_0} \cdot \frac{B_0}{C_1}, \frac{1}{C_0} \cdot \frac{B_0}{C_1} \cdot \frac{B_1}{C_2}, \dots \right\}$$

この数列の各項に A_k の重みを掛けて総和を求めると、 N 桁の精度の級数の和は $O(N^2)$ の計算量となる。

式(1)の形式の級数の和を効率良く計算するために、隣接する2項を再帰的にまとめ1項に集約することを考える^{*}。簡単のため項数 L を2の累乗個とすると、式(1)の2項ずつの和は次のようになる。

$$S_L = \sum_{k=0}^{L/2-1} \frac{\prod_{l=0}^{2k-1} B_l}{\prod_{l=0}^{2k} C_l} \left(A_{2k} + \frac{B_{2k}}{C_{2k+1}} A_{2k+1} \right) \quad (3)$$

$$= \sum_{k=0}^{L/2-1} \frac{\prod_{l=0}^{k-1} (B_{2l} B_{2l+1})}{\prod_{l=0}^k (C_{2l} C_{2l+1})} (A_{2k} C_{2k+1} + B_{2k} A_{2k+1}) \quad (4)$$

ここで、

^{*} 後⁶⁾は、同様の考え方によって級数計算を高速化する方法を提案しているが、計算の詳細や実験的評価は示されていない。

$$\begin{cases} A'_k &= A_{2k}C_{2k+1} + B_{2k}A_{2k+1} \\ B'_l &= B_{2l}B_{2l+1} \\ C'_l &= C_{2l}C_{2l+1} \end{cases} \quad (5)$$

とおくと、式(4)は式(1)と同じ形になるので、再び同様の変形を施すことができる。

したがって、

$$\begin{cases} \text{初項} \begin{cases} A_k^{(0)} &= A_k \\ B_k^{(0)} &= B_k \\ C_k^{(0)} &= C_k \end{cases} \\ \begin{cases} A_k^{(n+1)} &= A_{2k}^{(n)}C_{2k+1}^{(n)} + B_{2k}^{(n)}A_{2k+1}^{(n)} \\ B_k^{(n+1)} &= B_{2k}^{(n)}B_{2k+1}^{(n)} \\ C_k^{(n+1)} &= C_{2k}^{(n)}C_{2k+1}^{(n)} \end{cases} \end{cases} \quad (6)$$

の漸化式が得られ、 $A_0^{(n)}/C_0^{(n)}$ が 2^n 項までの級数の和を表すことになる。以下では、この漸化式に対応する行列表記を導入し、計算量が $O(N(\log N)^3)$ となることを示す。

2.1 行列表記

式(2)の再帰構造に着目し、次の数列 R_k を考える。

$$R_k = \frac{1}{C_k}(A_k + B_k R_{k+1}) \quad (7)$$

級数の有限項の和を考える場合 R_k は有理数になるので、 $R_k = \alpha_k/\beta_k$ (α_k と β_k は整数、 $\beta_k \neq 0$) とすると、

$$\begin{aligned} \frac{\alpha_k}{\beta_k} &= \frac{1}{C_k} \left(A_k + B_k \frac{\alpha_{k+1}}{\beta_{k+1}} \right) \\ &= \frac{A_k \beta_{k+1} + B_k \alpha_{k+1}}{C_k \beta_{k+1}} \end{aligned}$$

と書くことができる。これを 2×2 行列と 2 次元ベクトルの積として表記する。

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} B_k & A_k \\ 0 & C_k \end{pmatrix} \begin{pmatrix} \alpha_{k+1} \\ \beta_{k+1} \end{pmatrix} \quad (8)$$

ここで、 A_k, B_k, C_k を表す係数行列を

$$M_k \equiv \begin{pmatrix} B_k & A_k \\ 0 & C_k \end{pmatrix}$$

と定義すると、 α_0 と β_0 は次のように書ける。

$$\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} = M_0 M_1 M_2 \cdots M_{L-1} \begin{pmatrix} \alpha_L \\ \beta_L \end{pmatrix} \quad (9)$$

級数の L 項までの和 S_L は、式(7)の数列において $R_L = 0$ (式(9)では $\alpha_L/\beta_L = 0$) としたときの R_0 の値 (式(9)では α_0/β_0 の値) として得られる。

そこで、2次元ベクトルを要素間の除算によってスカラ化する演算子 $// \cdot //$ を次のように定義すると、

$$\left\| \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\| \triangleq \frac{\alpha}{\beta} \quad (10)$$

級数の和 S_L は、

$$S_L = \left\| M_0 M_1 M_2 \cdots M_{L-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \quad (11)$$

となる。 S_L を効率良く求めるために、次に述べる方法で行列の積 $M_0 M_1 M_2 \cdots M_{L-1}$ を計算した後に、ベクトル $(0 \ 1)^T$ との積 $(\alpha_0 \ \beta_0)^T$ を求め、最後に α_0/β_0 の除算を行う。

なお、 2×2 行列の乗算は通常 8 回の乗算を含むが、行列 M_k は上三角行列であり、上三角行列の積はつねに上三角行列となるため、行列 M_k の積は式(6)の漸化式の乗算回数と同じ 4 回である。

$$\begin{pmatrix} b & a \\ 0 & c \end{pmatrix} \begin{pmatrix} B & A \\ 0 & C \end{pmatrix} = \begin{pmatrix} bB & aC + bA \\ 0 & cC \end{pmatrix}$$

2.2 行列の積の計算方法

行列の積は結合法則を満たすので、隣接行列の積を再帰的に集約しながら計算することを考える。たとえば、8 個の行列の積は次の順序で計算する。

$$(((M_0 M_1)(M_2 M_3))((M_4 M_5)(M_6 M_7)))$$

この方法は、 L 個の行列 $M_0 \sim M_{L-1}$ の最大要素の桁数を K とすると、 K 桁の行列積が $L/2$ 回、 $2K$ 桁の行列積が $L/4$ 回、以下同様となり、全体の計算量は、求めるべき桁数を N とすると、後述するように $O(N(\log N)^3)$ となる。

一方、次の順序で行列の積を求めると、 N 桁の精度で計算する場合の計算量は $O(N^2)$ となる。

$$(((((((M_0 M_1) M_2) M_3) M_4) M_5) M_6) M_7))$$

どちらの方法でも乗算回数は同じであるが、隣接行列の積を再帰的に集約する方法は、乗算の桁数に大きな偏りが生じないため、多倍長乗算の計算量を抑制する効果がある。

なお、実際の計算では行列の数が 2 の累乗個とは限らないが、任意の箇所に単位行列を挿入することによって行列の数を 2 の累乗個にすることができる。

π および e を求める級数を行列の積によって表記したものを表 2 に示す。また、 $\pi = 6 \arcsin(1/2)$ の先頭 8 項の行列表記を用いた計算例を図 1 に示す。

2.3 集約による級数の和の計算量

行列表記による級数の和の式(11)において、行列の積には多倍長の乗算と加減算のみが含まれ、1 項

表2 π と e の級数の行列表記
Table 2 Formulas for π and e calculation in matrix form.

Machin	$\pi = 16 \arctan(1/5) - 4 \arctan(1/239)$ $\arctan \frac{1}{p} \approx \left\ \left(\begin{array}{cc} -1 & 1 \\ 0 & p \end{array} \right) \left(\begin{array}{cc} -3 & 1 \\ 0 & 3p^2 \end{array} \right) \cdots \left(\begin{array}{cc} -(2k+1) & 1 \\ 0 & (2k+1)p^2 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\ $
Ramanujan	$\pi \approx \frac{9801/(2\sqrt{2})}{\left\ \left(\begin{array}{cc} 1 \cdot 3 & a \\ 0 & 1 \end{array} \right) \left(\begin{array}{cc} 3 \cdot 5 \cdot 7 & a+b \\ 0 & c \end{array} \right) \cdots \left(\begin{array}{cc} (2k+1)(4k+1)(4k+3) & a+kb \\ 0 & k^3c \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\ }$ ただし, $a = 1103, b = 26390, c = 3073907232$
Chudnovskys	$\pi \approx \frac{426880\sqrt{10005}}{\left\ \left(\begin{array}{cc} -5 & a \\ 0 & 1 \end{array} \right) \left(\begin{array}{cc} -3 \cdot 7 \cdot 11 & a+b \\ 0 & c \end{array} \right) \cdots \left(\begin{array}{cc} -(2k+1)(6k+1)(6k+5) & a+kb \\ 0 & k^3c \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\ }$ ただし, $a = 13591409, b = 545140134, c = 10939058860032000$
自然対数の底	$e \approx 2 + \left\ \left(\begin{array}{cc} 1 & 1 \\ 0 & 2 \end{array} \right) \left(\begin{array}{cc} 1 & 1 \\ 0 & 3 \end{array} \right) \cdots \left(\begin{array}{cc} 1 & 1 \\ 0 & k+2 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\ $
自然対数の底の逆数	$e^{-1} \approx \left\ \left(\begin{array}{cc} -1 & 1 \\ 0 & 2 \end{array} \right) \left(\begin{array}{cc} -1 & 1 \\ 0 & 3 \end{array} \right) \cdots \left(\begin{array}{cc} -1 & 1 \\ 0 & k+2 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\ $

$$\begin{aligned} \pi &= 6 \arcsin \frac{1}{2} = 6 \sum_{k=0}^{\infty} \frac{(2k)!}{2^{2k}(k!)^2} \cdot \frac{1}{2k+1} \cdot \frac{1}{2^{2k+1}} \\ &= 6 \left(\frac{1}{2} + \frac{1}{2 \cdot 3 \cdot 2^3} + \frac{1 \cdot 3}{2 \cdot 4 \cdot 5 \cdot 2^5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 7 \cdot 2^7} + \cdots \right) \\ &= \frac{1}{2} \left(6 + \frac{6}{2 \cdot 3 \cdot 4} \left(1 + \frac{3^2}{4 \cdot 5 \cdot 4} \left(1 + \frac{5^2}{6 \cdot 7 \cdot 4} (\cdots) \right) \right) \right) \\ &\approx \left\| \left(\begin{array}{cc} 6 & 6 \\ 0 & 2 \end{array} \right) \left(\begin{array}{cc} 3^2 & 1 \\ 0 & 2 \cdot 3 \cdot 4 \end{array} \right) \left(\begin{array}{cc} 5^2 & 1 \\ 0 & 4 \cdot 5 \cdot 4 \end{array} \right) \left(\begin{array}{cc} 7^2 & 1 \\ 0 & 6 \cdot 7 \cdot 4 \end{array} \right) \cdots \left(\begin{array}{cc} 15^2 & 1 \\ 0 & 14 \cdot 15 \cdot 4 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\| \\ &= \left\| \left(\begin{array}{cc} 54 & 150 \\ 0 & 48 \end{array} \right) \left(\begin{array}{cc} 1225 & 193 \\ 0 & 13440 \end{array} \right) \left(\begin{array}{cc} 9801 & 521 \\ 0 & 126720 \end{array} \right) \left(\begin{array}{cc} 38025 & 1009 \\ 0 & 524160 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\| \\ &= \left\| \left(\begin{array}{cc} 66150 & 2026422 \\ 0 & 645120 \end{array} \right) \left(\begin{array}{cc} 372683025 & 282976569 \\ 0 & 66421555200 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\| \\ &= \left\| \left(\begin{array}{cc} 24652982103750 & 134616819631533744 \\ 0 & 42849873690624000 \end{array} \right) \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \right\| \\ &= 134616819631533744/42849873690624000 = 3.14159 \cdots \end{aligned}$$

図1 行列表記を用いた $\pi = 6 \arcsin(1/2)$ の計算例
Fig. 1 An example of $\pi = 6 \arcsin(1/2)$ calculation by using the matrix from.

に集約された後に除算によって級数の和 S_L を計算する。 N 桁の乗算の計算量を $M(N)$ とし, $M(N) = O(N \log N)$ とする。一方, N 桁の加減算の計算量は $O(N)$, 除算の計算量は $O(M(N))$ となるが, 加減算や除算の計算量は集約全体の計算量に比べると十分に少ないものとして, 級数の和の計算量を評価する。

A_k, B_k, C_k の最大桁数を K , 計算する項数を L とすると, 行列数を半分にする 1 回目の集約では K 桁の乗算を $2L$ 回行うので $2LM(K)$ の計算量となる。2 回目の集約では, 桁数は倍, 回数は半分になるので, $LM(2K)$ の計算量となる。 $\log_2 L$ 回の集約によってすべての行列は 1 項に集約されるので, この計算量は次のように近似できる。

$$\begin{aligned} &2L[M(K) + (1/2)M(2K) + \\ &\quad (1/4)M(4K) + (1/8)M(8K) + \cdots] \\ &\approx 2L[(\log_2 L)M(K) + K(\log_2 L)^2/2] \end{aligned}$$

N 桁の精度で計算する場合, 適当な定数 l, k を用いて, 計算すべき項数は $L = lN, A_k, B_k, C_k$ の最大桁数は $K = k \log N$ と書けるので, 総計算量は $O(N(\log N)^3)$ となる。

3. 実験と考察

級数の集約によって π と e の値を求める実験を行い, π に関しては, 高速な計算法として知られる反復計算法と比較した。実験には次の計算機を用いた。

- (i) Gateway G6-200 (PentiumPro 200MHz,

表3 級数法による104万桁の π と e の計算時間(秒)
Table 3 Computation time of one million digits of π and e .

公式	本手法	従来法
Machin	507	113443
Ramanujan	218	73971
Chudnovskys	129	42631
自然対数の底 e	54	14320

64 MB, FreeBSD2.2.1R)

(ii) SGI OCTANE (R10000 195 MHz, 384 MB, IRIX6.4)

なお、級数の集約計算では数値の桁数に関して次のような性質がある。図1の例では、行列の各要素の値は15桁以上の数値となっているが、最後に除算(α_0/β_0)を行って得られる級数の和は5桁の精度しかない。このように、計算途中で行列の各要素の値は必要な精度 N 桁よりも大きくなる場合がある。ここでは、計算途中で N 桁を超える数値はすべて N 桁で切り捨てることにする。このような切捨てを行っても、最後の行列の要素はほぼ N 桁の精度で求められ、その結果、級数の和を $N-\epsilon$ 桁まで正しく求めることができる。この ϵ は乗算や除算の精度によって決まる数で、 N が十分に大きいときには $N \gg \epsilon$ という性質がある。

3.1 級数の集約の評価

級数の集約による高速化の効果を調べるために、従来の級数計算法(計算量 $O(N^2)$)と本手法を比較した。 π の計算は、小数点以下 $2^{20}-4=1,048,572$ 桁の精度で行い、計算機(i)を使用した。表3および図2に計算時間を示す。この結果、級数の集約を用いた場合、従来の級数計算法の $1/340 \sim 1/200$ の時間で計算できることが分かった。なお、計算誤差は下位48桁以下であった。

同じ桁数の e の計算では、表3の π の計算で最も高速なChudnovskysの公式による129秒と比べてさらに半分以下の時間で計算できることが分かった。これは、 e の計算において行列の左上の要素がつねに ± 1 となり、乗算の必要がないためである。なお、表2の2種類の式で e を計算し結果を照合したところ計算誤差は下位12桁であった。

3.2 級数法と反復法の比較

3.2.1 104万桁の π の計算

π の計算の反復法として、Gauss-Legendreの公式とBorweinsの4次収束の公式を用いて約104万桁の π を計算する実験を計算機(i)で行った。図2の右にそれらの計算時間を示す。級数の集約を行った級数法

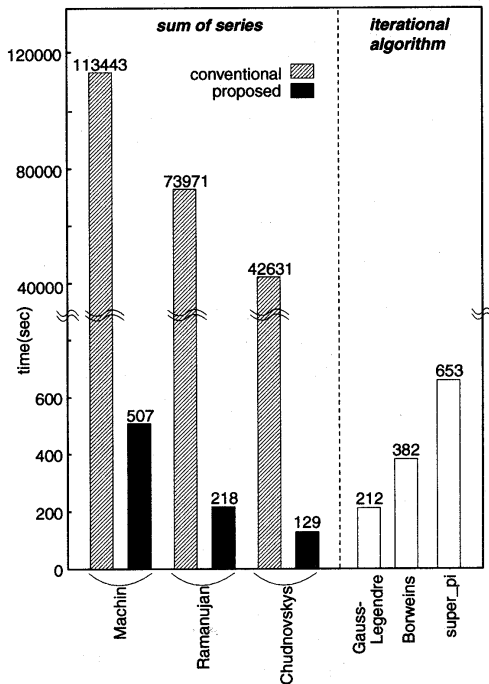


図2 104万桁の π の計算時間の比較
Fig. 2 Computation time of one million digits of π .

の結果と比較すると、Chudnovskysの公式はGauss-Legendreの公式とBorweinsの4次収束の公式のいずれよりも高速に計算しうることが分かる \star 。なお、反復法による計算誤差は下位120桁以下であった。

参考のため、Gauss-Legendreの公式を用いて π を計算するsuper_pi $\star\star$ を計算機(i)で実行し計算時間を計測した。OSや計算桁数の違いがあり単純な比較はできないが、1,048,576桁を計算するのに653秒要した。

3.2.2 5.3億桁の π の計算

級数の集約によるChudnovskysの公式の計算量は $O(N(\log N)^3)$ であり、Gauss-Legendreの公式の計算量 $O(N(\log N)^2)$ に比べて $\log N$ が掛かっているため、計算桁数の増大によって不利になることが予想される。

そこで両公式による3.2万桁から5.3億桁までの π の計算時間を比較した。なお、計算機(i)では主記憶容量が不足するため計算機(ii)を用いた。図3に桁数と

\star 文献(4), (8)では、Gauss-Legendreの公式とBorweinsの4次収束の公式による計算量はほぼ等しいとされているが、本研究では従来法 $^{7), 8)}$ よりも計算量を削減したGauss-Legendreの公式を実装している。

$\star\star$ ftp://www.cc.u-tokyo.ac.jpで公開されている。

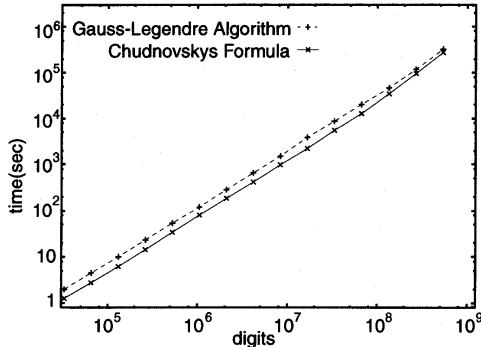


図3 π の桁数と計算時間の変化 (Chudnovskys の公式と Gauss-Legendre の公式の比較)

Fig. 3 Comparison of computation time between Chudnovskys formula and Gauss-Legendre algorithm.

表4 計算機 (ii) による π と e の計算時間
Table 4 Computation time of π and e on machine (ii).

公式	104 万桁 (分)	5.3 億桁 (時間)
Chudnovskys	1.33	75.05
Gauss-Legendre	2.01	88.77
自然対数の底 e	0.56	12.63

計算時間の関係を、表4に104万桁と5.3億桁の計算時間を示す。1億桁以下の計算では Gauss-Legendre の公式は Chudnovskys の公式の平均1.6倍の時間を要し、桁数による時間比率の変化はほとんど見られない。ただし、今回の実験では1億桁以上の計算において Chudnovskys の公式の計算速度が低下し、両公式による計算時間の差が少なくなった。この原因については後述する。

なお、5.3億桁の計算では、両公式で $2^{29} - 4 = 536,870,908$ 桁目まで計算した結果、誤差は下位236桁であった。また、計算機(ii)上で e の値を536,870,788桁計算すると12時間38分を要した。

3.3 考察

$O(N(\log N)^3)$ の級数法が $O(N(\log N)^2)$ の反復法と同程度の時間で計算できる理由を考察するために、級数の集約と Gauss-Legendre の公式の計算量を主要項の係数まで含めて考える。これらの計算量の大部分は多倍長乗算のための FFT であるので^{*}、 N 桁数の FFT の計算量 $F(N)$ を基準として、 N 桁の π の計算量を概算する。

まず、Gauss-Legendre の公式の計算量について考

える。この公式では、各反復における計算桁数はすべて N 桁であり、同一の演算を反復する。今回の実装では、各反復の計算量は $10F(N)$ 程度、反復回数は $(\log_2 N) - 2$ 回、初期値計算と後処理の計算量は合計 $14F(N)$ 程度であるため、全体の計算量は $(10(\log_2 N) - 6)F(N)$ 程度となる。

次に、級数の集約の計算量について考える。集約過程の行列を

$$\begin{pmatrix} b_i & a_i \\ 0 & c_i \end{pmatrix}$$

と表すと、各数値が十分に大きいときには c_i と a_i の桁数はほぼ等しく、 $|c_i| \gg |b_i|$ となる。 c_i は初期行列の右下要素の積であり、その桁数は容易に計算できるので、 c_i の桁数を用いて行列の積の計算量を見積もる。右下要素が m 桁の2つの行列の積の計算量は $10F(m)$ 程度である。すべての項を集約した後の行列を

$$\begin{pmatrix} B & A \\ 0 & C \end{pmatrix}$$

とすると、 C の桁数は $\log_{10} \prod_{k=0}^{L-1} C_k$ であり、Chudnovskys の公式の場合、ほぼ $(0.0637 \log_2 N + 0.796)N$ 桁となる。これを $p(N)N$ とすると、集約に含まれるすべての行列の積の計算量は、

$$\begin{aligned} & 10F(p(N)N/2) + 2 \cdot 10F(p(N)N/4) + \\ & 4 \cdot 10F_m(p(N)N/8) + \dots \\ & \approx 2.5p(N) \log_2(p(N)N)F(N) \end{aligned}$$

となる。後処理の計算量は、今回の実装では $17F(N)$ 程度である。したがって、全体の計算量は、 $\{0.16(\log_2 N)^2 + 2.0(\log_2 N) + 17\}F(N)$ 程度となる。

N 桁数の FFT の計算量 $F(N) = O(N \log N)$ を用いて以上の結果を整理すると以下ようになる。

- Gauss-Legendre の公式の計算量の $N(\log_2 N)^2$ の係数は約10である。
- 級数の集約の計算量の $N(\log_2 N)^3$ の係数は約0.16、 $N(\log_2 N)^2$ の係数は約2.0である。

したがって、前述の実験で用いた N が 2^{30} 程度の値では、Gauss-Legendre の公式より級数の集約の計算量が少なく、Chudnovskys の公式を用いる方が高速に計算できることが分かる。

上で求めた計算量の式を用いて100万桁と6700万桁の計算量を概算した結果を表5に示す。この桁数では計算量の比は1.5~1.6程度であり、この結果はおおむね実験結果と合っていることが分かる。なお、 N が大きくなるに従い $N(\log N)^3$ の影響が大きくな

^{*} ここでは、乗算に FFT 以外のアルゴリズムを用いる場合については考慮しない。

表5 π の計算量の理論値
Table 5 Theoretical complexity of π calculation.

	2^{20} 桁	2^{26} 桁
Chudnovskys	$121F(2^{20})$	$169F(2^{26})$
Gauss-Legendre	$194F(2^{20})$	$244F(2^{26})$

るが, 上記の式では N が 2^{47} 桁*程度までは, 理論上 Chudnovskys の公式の方が Gauss-Legendre の公式より計算量の点で有利である. ただし実際の計算時間は, 使用する計算機のハードウェア資源や実装上の工夫に大きく依存するので, 両者の優劣を比較することは簡単ではない.

一方, 実験において 1 億桁以上の計算で Chudnovskys の公式の計算速度の低下が生じた理由は, 主に乗算桁数の違いによるものである. N 桁の π を計算するには, 今回用いた Gauss-Legendre の公式では一度に $N/8$ 桁の乗算を行えば十分な性能が得られるのに対して, 級数の集約では N 桁の乗算を行う必要がある. しかし計算機 (ii) では, 主記憶容量の制限により 1 億桁を超える乗算を FFT によって高速に行うことはできないため, FFT による乗算よりも低速な Karatsuba の方法¹⁾を使わざるをえなかったことが速度低下の原因である. 主記憶容量に制限がある場合の計算速度の低下を回避する方法については, 現在検討中である.

4. おわりに

数学定数や初等関数の計算法として, 級数を用いる方法と効率の良い反復計算による方法が知られている. 一般に級数の計算は反復計算法に比べ計算時間的に不利とされていた. 本稿では, この多倍長級数の高速な計算法を提案し, 従来法では $O(N^2)$ の計算量であった N 桁の計算を $O(N(\log N)^3)$ の計算量で行えることを示した.

本手法の有効性を示すために π と e の級数計算を取り上げ実験を行った. その結果, 従来は π の値の計算には級数よりも反復計算法を用いる方が格段に有利とされていたが, 級数の集約を行うことによって同程度の計算量となることが分かった. 特に, 3.2 万桁から 5.3 億桁の π の計算に, 級数の和による Chudnovskys の公式を用いると, 反復計算による Gauss-Legendre の公式よりも高速に計算できることが明らかになった.

本手法は, 並列処理による高速化手法である文献 5)

と同様の手法であり, 級数の集約の各段階で隣接する行列の積を独立に計算することが可能であるため, 並列処理による高速化の効果が期待できる. これについて評価する予定である.

本研究の一部は, 広島市立大学大学院情報科学研究科自主プロジェクト演習の補助を受けて行った.

参考文献

- 1) Borwein, J.M., Borwein, P.B. and Bailey, D.H.: Ramanujan, Modular Equations, and Approximations to Pi or How to compute One Billion Digits of Pi, *The American Mathematical Monthly*, Vol.96, pp.201-219 (1989).
- 2) Brent, R.P.: Fast Multiple-Precision Evaluation of Elementary Functions, *J. ACM*, Vol.23, pp.242-251 (1976).
- 3) Salamin, E.: Computation of π Using Arithmetic-Geometric Mean, *Math. Comp.*, Vol.30, pp.565-570 (1976).
- 4) 高橋大介, 金田康正: 分散メモリ計算機による円周率の 515 億桁計算, 情報処理学会論文誌, Vol.39, pp.2074-2083 (1998).
- 5) Kogge, P.M. and Stone, H.S.: A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations, *IEEE Trans. Comput.*, Vol.C-22, No.8, pp.786-793 (1973).
- 6) 後 保範: 逆数型無限級数の n 桁計算の計算量を削減する前処理方式, 京都大学数理解析研究所「数値解析における前処理の研究」予稿集, p.9 (1998).
- 7) 大浦拓哉: 円周率公式の改良と高速多倍長計算の実装, 情報処理学会研究報告, 98-HPC-74, pp.25-34 (1998).
- 8) 高橋大介, 金田康正: 多数桁の円周率を計算するための公式の改良: ガウス-ルジャンドルの公式とボールウェインの 4 次の収束の公式, 情報処理学会論文誌, Vol.38, No.11, pp.2406-2409 (1997).

(平成 11 年 1 月 11 日受付)

(平成 11 年 10 月 7 日採録)

右田 剛史 (学生会員)



1998 年広島市立大学情報学部知能情報システム工学科卒業, 現在同大学院情報科学研究科修士課程に在学中. コンピュータビジョン, 数値計算の高速アルゴリズムに興味を持っている. 電子情報通信学会会員.

* 現時点で報告されている π の最高計算桁数は約 $2^{35.6}$ 桁である⁴⁾.

**天野 晃** (正会員)

1988年京都大学工学部電気工学科卒業。1990年同大学院修士課程修了。1993年同博士課程単位取得退学。同年京都大学工学部助手。1995年より広島市立大学助教授。工学博士。コンピュータビジョン、人工知能の研究に従事。電子情報通信学会会員。

**藤野 清次** (正会員)

1974年京都大学理学部卒業。1993年博士号取得(工学, 東京大学)。1994年3月より広島市立大学情報科学部教授。現在に至る。1997年度応用数理学会論文賞。数値計算, 並列計算に興味を持つ。応用数理学会会員。

**浅田 尚紀** (正会員)

1979年京都大学工学部電気工学科卒業。同大学院博士課程, 福井医科大学助手, 京都大学工学部助手, 岡山大学工学部助教授を経て1995年から広島市立大学情報科学部教授, 現在に至る。1989年シカゴ大学客員研究員。工学博士。コンピュータビジョン, コンピュータグラフィクス, 人工知能, 医用画像理解の研究に従事。1993年情報処理学会論文賞。IEEE CS, ACM, 電子情報通信学会, 人工知能学会, 医用画像工学会各会員。
