

多倍長の円周率計算

吉原 郁夫^{a)}・坂本 亜衣^{b)}・本田 詩織^{c)}・山森 一人^{d)}・棟朝 雅晴^{e)}

Multiple-precision Calculation of π

Ikuo YOSHIHARA, Ai SAKAMOTO, Shiori HONDA,
Kunihito YAMAMORI, Masaharu MUNETOMO

Abstract

Multiple-precision calculation is necessary for precisely solving scientific engineering problems. Extremely long precision is employed to evaluate the mathematical constant, e.g. π , γ (Euler's constant), e (Napier's constant) etc. To develop multiple-precision computing software, we try to calculate π with more than one million decimal digits. The proto-type code is verified by performing calculation with numerical examples and evaluated rapidness of calculation. Hother to π with 16,777,199 decimal digit is obtained.

Keywords: Multiple-precision arithmetic, Fast Fourier transform, Salamin and Brent formula

1. はじめに

多倍長演算とは、倍精度以上の高精度演算である¹⁾。4倍精度以上の表現は、倍精度データ型(10進で約16桁)を繰り返して用いることで表される。これらは、算術オーバーフローの防止、暗号処理、高精度なシミュレーション、計算力学の大規模問題などの正確な演算結果を必要とする場合に用いられる。一方、円周率や数学定数を数百万桁以上求めるには、4倍精度や8倍精度とは別の多倍長演算が必要となる。

本研究では、100万桁以上の桁数を扱う演算を行い、例題として円周率を求めてみる。円周率の計算は昔から多くの人々によって行われており、コンピュータを用いた計算では計算桁数が飛躍的に伸びている。また、桁数の多い円周率計算は、新型計算機が開発されるたびに、「より速くたくさんの桁を計算する」ことにより、新型計算機の性能評価のために使われてきた³⁾。

多倍長演算ソフトウェアの開発と検証については、別報で報告している。本報告では円周率計算について報告する。

a) 情報システム工学科 教授

b) 情報システム工学科 学生

c) 情報システム工学科 学生

d) 情報システム工学科 准教授

e) 北海道大学大学院情報科学研究科 准教授

2. 多倍長演算

2.1 多倍長数の表現

10進の数をk桁ずつ区切り $p=10^k$ 進法とする。たとえば、10進のある数を $k=5$ 桁ずつで区切ると図1のようなイメージとなる。

本研究では、10進の桁数の多い数を4桁または5桁ずつで分割し、円周率を計算する。

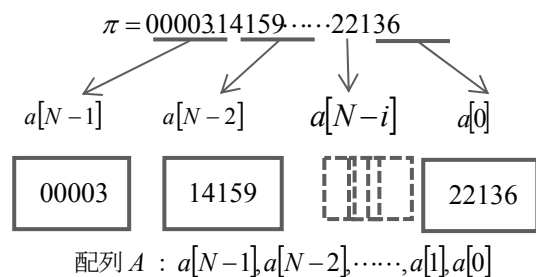


図1. 多倍長数を5桁ずつに分割

2.2 加減算

多倍長での加減算は筆算通り行う。 p 進法 N 桁の整数 $\sum_{i=0}^{N-1} a_i p^i$ と整数 $\sum_{i=0}^{N-1} b_i p^i$ の加減算は式(2.1)ようになる。

$$\sum_{i=0}^{N-1} c_i p^i = \sum_{i=0}^{N-1} (a_i \pm b_i) p^i \quad (2.1)$$

各桁ごとの加減算を行ったあと、以下のような正規化を行う

必要がある。

$c_i > p$ の場合、 c_i の桁あふれ分を c_{i+1} に加える。

$$c_{i+1} = c_{i+1} + c_i / p \quad (2.2)$$

$$c_i = c_i \bmod p \quad (2.3)$$

$c_i < 0$ の場合、 c_i から 1 を借りてきて、 c_i 番目に加える。

$$c_{i+1} = c_{i+1} - 1 \quad (2.4)$$

$$c_i = c_i + p \quad (2.5)$$

桁数が N のとき、計算量は $O(N)$ となる。

2.3 乗算

p 進法での乗算は、式(2.6)のようになる。

$$\begin{aligned} \sum_{i=0}^{2N-2} c_i p^i &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_i b_j p^{i+j} \\ &= \sum_{k=0}^{2N-1} \left(\sum_{i=0}^k a_i b_{k-i} \right) p^k \end{aligned} \quad (2.6)$$

$\sum_{i=0}^k a_i b_{k-i}$ は畳み込み演算と呼ばれる。桁数が N のとき、式(2.6)の計算量は $O(N^2)$ であり、桁数が 2 倍になると、計算量は 4 倍になる。計算量の削減のために、高速フーリエ変換を用いる。

2.4 除算

b/a を計算する場合、まず逆数 $1/a$ を求め、つぎに逆数 $1/a$ と b の積として計算する。

$$\frac{b}{a} = \frac{1}{a} \times b \quad (2.7)$$

$1/a$ は方程式(2.8)の解である。式(2.9)のニュートン・ラフソン法⁴⁾により、式(2.10)を導けば、加減算と乗算だけから逆数 $1/a$ を求めることができる。

$$f(x) = \frac{1}{x} - a \quad (2.8)$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.9)$$

$$x_{n+1} = x_n + x_n(1 - ax_n) \quad (2.10)$$

3. 乗算の高速化

乗算の畳み込み演算はフーリエ変換を行うことで、周波数空間の内積演算に帰着させることができる。

3.1 畳み込み定理

関数 $f(n)$ および $g(n)$ をそれぞれフーリエ変換したものを $F(k)$ および $G(k)$ とする。

$$f(n) \xrightarrow{F} F(k) \quad (3.1)$$

$$g(n) \xrightarrow{F} G(k) \quad (3.2)$$

畳み込みは式(3.3)と定義される。ここで*は畳み込みを意味する記号である。

$$h(n) = (f * g)(n) = \int_{-\infty}^{\infty} f(y)g(n-y)dy \quad (3.3)$$

$$H(k) = F(k) \cdot G(k) \quad (3.4)$$

3.2 高速フーリエ変換

本研究で使うフーリエ変換は、離散フーリエ変換である。

$$F(k) = \sum_{n=0}^{N-1} f(n)W_N^{2kn}, \quad k=0,1,2,\dots,N-1 \quad (3.5)$$

$$W_N = \exp \frac{-j2\pi}{N}, \quad j=\sqrt{-1}$$

離散フーリエ変換には $O(N^2)$ の複素乗算と、 $O(N^2)$ の複素加算が必要である。離散フーリエ変換は、クーリー・チュエキー (James W. Cooley and Jhon W. Tukey)⁵⁾ による高速フーリエ変換^{6,7,8)} で高速に処理できる。

式(3.5)の標本長 N は 2 のべき乗になっているとする。積の総和に関する信号成分 f の指標が偶数か奇数かによって分けることで、標本長が N の離散フーリエ変換を、標本長が $N/2$ の離散フーリエ変換の組み合わせとして書ける。式(3.6)から式(3.7)が得られる。

$$\begin{aligned} F(k) &= \sum_{n=0}^{N/2-1} f(2n)W_N^{2kn} + \sum_{n=0}^{N/2-1} f(2n+1)W_N^{(2n+1)k} \\ &= \sum_{n=0}^{N/2-1} f(2n)W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N/2-1} f(2n+1)W_{N/2}^{kn} \end{aligned} \quad (3.6)$$

$$\begin{aligned} F(N/2+k) &= \sum_{n=0}^{N/2-1} f(2n)W_{N/2}^{kn} + W_N^{N/2+k} \sum_{n=0}^{N/2-1} f(2n+1)W_{N/2}^{kn} \end{aligned} \quad (3.7)$$

これらを用い、偶数指標成分 $T(2k)$ と奇数指標成分 $T(2k+1)$ の重み付き和として表すことができる。

$$F(k) = T(2k) + W_N^k T(2k+1), \quad k=0,1,2,\dots,N-1 \quad (3.8)$$

ただし、

$$T(2k+N) = T(2k) = \sum_{n=0}^{N/2-1} f(2n)W_{N/2}^{kn} \quad (3.9)$$

$$T(2k+1+N) = T(2k+1) = \sum_{n=0}^{N/2-1} f(2n+1)W_{N/2}^{kn} \quad (3.10)$$

式(3.8)の計算手順は図2～図4 (N=2,4,8 の場合) の有向グラフで表される。同図の演算はバタフライ演算と呼ばれる。表記のない矢印に対応する重みは1である。また、奇数指標の中間変数にのみ重み係数 W_N^k がかかる。離散フーリエ変換の標本長を順次半減させていくことで、すべてをバタフライ演算に置き換えることができる(図5)。計算量は $O(N \log_2 N)$ に削減できる。

逆離散フーリエ変換は式(3.11)で定義され、FFTと同様に高速処理が可能である(図6)。

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k)W_N^{-kn} \quad n=0,1,2,\dots,N-1 \quad (3.11)$$

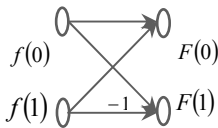


図2. N=2 のバタフライ演算

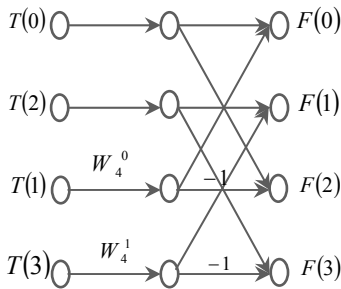


図3. N=4 のバタフライ演算図

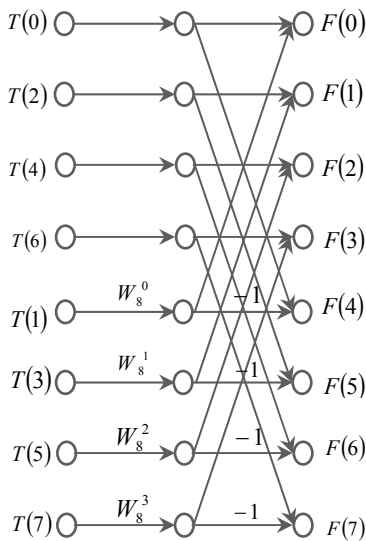


図4. N=8 のバタフライ演算

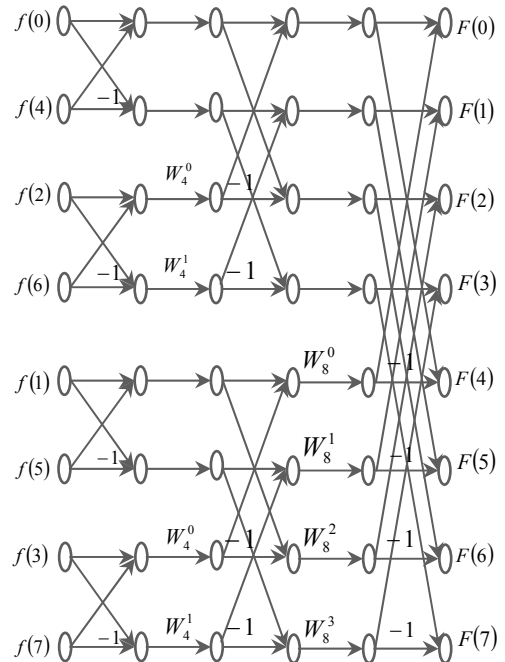


図5. N=8 の高速フーリエ変換

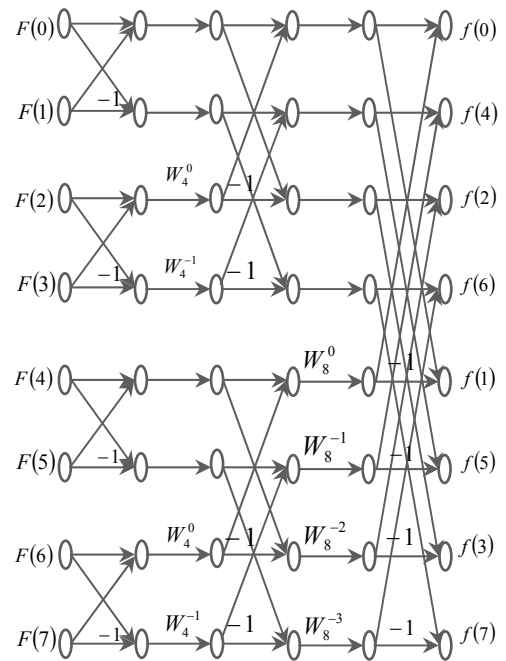


図6. N=8 の高速逆フーリエ変換

4. 円周率計算

計算機を用いた円周率計算^{9,10)}では、アークタンジェント (arctangent)の公式や、ボールウェイン(Borwein)の公式、算術幾何平均(AGM)による方法が使われる。

本研究では、算術幾何平均による方法を使う。

4.1 サラミン・ブレントの公式

初期値を $a_0 = 1$, $b_0 = 1/\sqrt{2}$, $c_0^2 = a_0^2 - b_0^2$ とする。

$$a_{n+1} = \frac{1}{2}(a_n + b_n) \tag{4.1}$$

$$b_{n+1} = \sqrt{a_n b_n} \tag{4.2}$$

$$c_{n+1} = \frac{1}{2}(a_n - b_n) \tag{4.3}$$

$$\pi = \lim_{n \rightarrow \infty} \frac{2a_{n+1}^2}{1 - \sum_{j=0}^n 2^j c_j^2} \tag{4.4}$$

式(4.4)は π に 2 次収束する。

このサラミン・ブレントの公式の演算量を減らした計算法が、改良サラミン・ブレントの公式¹¹⁾である。乗算の回数が半分に削減される。

$$A_n = a_n^2, B_n = b_n^2, C_n = c_n^2$$

で置き換えることで得られる。

4.2 改良サラミン・ブレントの公式

初期値を $A_0 = 1$, $B_0 = 1/2$, $C_0 = A_0 - B_0$ とし、漸化式(4.5)、(4.6)、(4.7)に従って、順次 3 つの数列 A_n , B_n , C_n を作成していく。このとき、式(4.8)も π に 2 次収束する。すなわち、反復を繰り返すごとに求まる桁数が 2 倍に増えていく。

$$A_{n+1} = \frac{1}{2} \left(\frac{1}{2}(A_n + B_n) + \sqrt{A_n B_n} \right) \tag{4.5}$$

$$B_{n+1} = \sqrt{A_n B_n} \tag{4.6}$$

$$C_{n+1} = A_{n+1} - B_{n+1} \tag{4.7}$$

$$\pi = \lim_{n \rightarrow \infty} \frac{A_n + B_n}{1 - \sum_{j=0}^n 2^j C_j} \tag{4.8}$$

4.3 平方根演算

\sqrt{a} を計算するには、まず逆平方根 $1/\sqrt{a}$ を求め、それに a を乗じて求める¹²⁾。逆平方根 $1/\sqrt{a}$ は式(4.10)の解であり、ニュートン・ラフソン法により求める。

$$\sqrt{a} = \frac{1}{\sqrt{a}} \times a \tag{4.9}$$

$$f(x) = \frac{1}{x^2} - a \tag{4.10}$$

$$x_{n+1} = x_n + \frac{x_n}{2} (1 - ax_n^2) \tag{4.11}$$

5. 実験結果および考察

まず、改良サラミン・ブレントの計算をして、反復回数ごとの求まった円周率の桁数を表 1 と図 7 に示す。回帰分析の結果、直線の傾きが 2 であることから、2 次収束といえる。

次に、 $p=10^k$ 進法とし、 $k=4$ と $k=5$ の場合の計算を行う。 $k=4$ の場合、 $p=10,000$ 進法となる。計算結果が正しいことをすでに公表されている結果¹³⁾と比較し、得られた円周率の桁数を表 2 と表 3 に示す。 $k=4$ の場合、 p 進での桁数 4,194,304、すなわち 10 進では 16,777,199 桁まで円周率を求めることができた(表 2)。 $k=5$ の場合、 p 進での桁数 524,288、すなわち 10 進では 2,621,425 桁までしか円周率を求めることができなかった(表 3)。理由として、計算の過程でオーバーフローが起こっていることが考えられる。本研究の現段階では、 $k=5$ 桁より $k=4$ 桁の方が有利といえる。

表1 反復回数ごとの求まった円周率の桁数

計算回数	円周率桁数	12	5,581
1	1	13	11,170
2	3	14	22,348
3	8	15	44,700
4	19	16	89,409
5	41	17	178,823
6	84	18	357,647
7	171	19	715,301
8	345	20	1,430,612
9	694	21	2,861,225
10	1,392	22	5,722,454
11	2,787	23	11,444,915

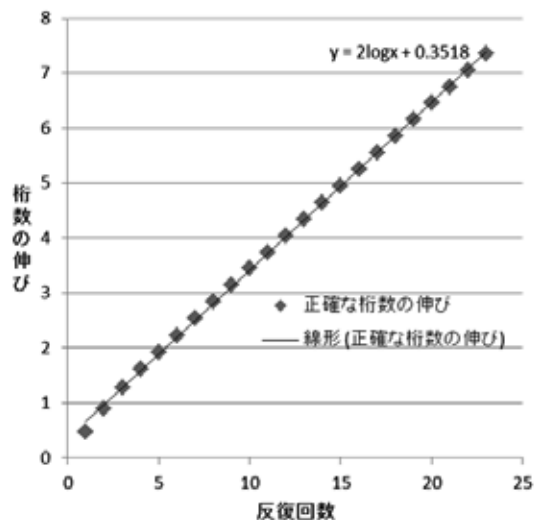


図7. エクセルを使った回帰分析結果

表 2. 4 桁ずつに分割した場合の計算結果

k	桁数	反復計算回数	計算時間 (s)	円周率桁数
4	64	8	0.05	245
	128	9	0.07	501
	...			
	1,024	12	0.98	4,084
	2,048	13	4.8	8,179
	...			
	16,384	16	74.7	65,523
	32,768	17	176.8	131,059
	65,536	18	417.7	262,131
	131,072	19	930.6	524,275
	262,144	20	2215.5	1,048,563
	524,288	21	5077.9	2,097,139
	1,048,576	22	12332.8	4,194,291
	2,097,152	23	33373.6	8,388,595
4,194,304	24	110294.5	16,777,199	

表 3. 5 桁ずつに分割した場合の計算結果

k	桁数	反復計算回数	計算時間 (s)	円周率桁数
5	64	8	0.04	307
	128	9	0.07	636
	...			
	1,024	12	0.95	5,105
	2,048	13	4.6	10,225
	...			
	16,384	16	72.2	81,905
	32,768	17	172.7	163,825
	65,536	18	404.6	327,665
	131,072	19	942.4	655,345
	262,144	20	2214.9	1,310,705
	524,288	21	5076.9	2,621,425
	1,048,576	-	-	-

証実験については、別報で報告している。本報告では、題材として改良サラミン・プレントの公式を用い、円周率を計算した。そして、作成した円周率算出プログラムが2次収束していることを確認し、k=4 の場合に最大、p 進での桁数 4,194,304、すなわち 10 進では 16,777,199 桁まで円周率を求めることができた。

今後の課題としては、k=4 の場合の p 進での桁数を 4,194,304 から、8,388,608、16,777,216、・・・と増やしていき、10 進で 16,777,199 桁以上の円周率を求めたい。また、高速フーリエ変換に代えて高速アダマール変換を用いることで、乗算の効率をもっと上げることができないか研究することが考えられる。

謝辞

本研究の一部は、北海道大学情報基盤センターとの共同研究の一環として行われたもので、関係各位に感謝いたします。

参考文献

- 1) 藤原宏志、磯祐介：“高速多倍長計算環境における数値解析”、日本応用数理学会論文誌、Vol.15、No.3、pp.403-417、2005。
- 2) 永井貴博、吉田仁、黒田久泰、金田康正：“SR11000 モデル J2 における 4 倍精度積和演算の高速化”、情報処理学会論文、Vol.48、No.13、pp.214-222、2007。
- 3) 金田康正： π のはなし、東京図書、1991。
- 4) 水島二郎、柳瀬眞一郎：理工学のための数値計算法、数理工学社、2009。
- 5) J. W. Cooley：“J. W. Tukey: An Algorithm for the Machine Calculation of Complex Fourier Series”、Mathematics of Computation、Vol.19、pp.297-301、1965。
- 6) 佐川雅彦、貴家仁志：高速フーリエ変換とその応用、昭晃堂、1993。
- 7) 金谷健一：これなら分かる応用数学教室、共立出版、2003。
- 8) D. Takahashi：“Parallel implementation of multiple-precision arithmetic and 2,576,980,370,000 decimal digit of π calculation”、Parallel Computing、Vol.36、pp.439-448、2010。
- 9) D. Jean-Paul(著)、畑政義(訳)： π 魅惑の数、朝倉書店、2001。
- 10) 高橋大介、金田康正：“分散メモリ型並列計算機による円周率 515 億桁計算”、情報処理学会論文誌、Vol.39、No.7、pp.2074-2083、1998。
- 11) 大浦拓哉：“円周率公式の改良と高速多倍長計算の実装”、日本応用数理学会論文誌、Vol.9、No.4、pp.165-172、1999。
- 12) 高橋大介、金田康正：“多倍長平方根の高速計算法”、情報処理学会研究報告、95-HPC-58、pp.51-56、1995。
- 13) PI WORLD OF JA0HXV <<http://ja0hvx.calico.jp/pai/start.html>>、(2012/02/13 アクセス)

6. 結論

本研究では、多倍長の円周率計算を行い、円周率を約 1,600 万桁求めることができた。多倍長演算ソフトウェアの開発と検