

## Jによる円周率 $\pi$ の多桁計算

西川 利男

はじめに

$$\pi = 3.141592653589 \dots\dots$$

昔から、多くの人々が興味を持ち、数学のロマンを感じさせるテーマである。このために一生をささげた人が何人もいる。ベックマンの「 $\pi$ の歴史」は数学だけでなく、技術・文化を含めた歴史にも及び、かつ皮肉をこめたユニークな語り口は、他の本にない楽しい本である。[1]

日本でも、円周率の計算は江戸時代の和算の大きな分野、円理としてとして、多くの人々を魅了した。日本独自の数学である和算は 関孝和を初めとして多くの和算家を生み出した。建部賢弘兄弟などが活躍する面白い小説がある。[2]

現在は、スーパーコンピュータの性能を測る指標として、大変な桁数の円周率の計算がなされている。私の本棚にあった少し古いが、パソコンによる  $\pi$  の計算の本を、夏の暑さボケの解消にと取り出し、Jでやってみた。[3]

### 0. $\pi$ 計算のコンピュータ処理における位置づけ

あらためて、コンピュータ処理、(計算といってもよいが)、というものを考えてみよう。一般に、一緒にまとめて考えられているが、私にとってはつぎのようにかなり性質の異なったものであると思う。

(1) 科学技術をサポートする計算では、有効数字4桁ほどの浮動小数点計算で十分である。あるいは、グラフィックス表示の計算でも同様である。

(2) 整数論の計算、 $\pi$ の多桁計算、組み合わせ論などの計算では、浮動小数点計算などではとんでもない、丸めなしの計算が必要である。J言語の分数表示 x 機能は、そのための最も有効なものである。また、一般には以下のような複数ワードによる多桁整数計算を行う。

(3) さらに、かつては事務計算などとまとめて言われていたが、検索、分類、順序付けなど文字を扱う処理、ファイル処理がある。そして、いまやこの分野の処理のためにコンピュータが在るとさえいえる。

私の考えるに、上の  $\pi$  の多桁計算など項目(2)は、科学技術計算というより、コンピュータのハードウェア、プログラミング言語の基本に関する項目(3)のようなシステム処理に近いのではないかと思う。

---

[1] ペートル・ベックマン、田尾陽一、清水韶光訳「 $\pi$ の歴史」蒼樹書房(1989).

[2] 鳴海風「円周率を計算した男」新人物往来社(2000).

[3] 大野栄一「パソコンで挑む円周率」講談社ブルーバックス(1991).

[4] 木田祐司「多倍長計算 BASIC、ユーザーズマニュアル」日本評論社(1995).

[5] 高木貞治「解析概論」岩波書店(1986).

## 1. $\pi$ 計算の2つのアプローチとプログラミング言語

この課題を考えてみるに、精密な円周率の値を得るのに、小数点以下、多くの数字が並べばそれで良いというだけのものではない。ここには数学の問題以外にプログラミング言語の問題もかかわっている。そこで、J言語の特徴を生かしたいくつかのプログラミングの試みをやってみた。

$\pi$  の多桁計算を問題の本質を変えないで、計算の技法を変える。つまり、小数点以下の計算ではなく、巨大数の多桁整数計算とするのである。

### 1. 1 複数ワードによる多桁計算

多桁整数計算には、通常つぎのような方法がとられることが多い。

1ワードに何桁もの数値、たとえば5桁(通常整数)、あるいは10桁(倍精度整数)を格納し、1つの値にこの複数のワードを使って計算を行う。こうすれば、BASICなどでも、多桁整数計算、そして $\pi$ の多桁計算もできる。[3]

今回、Jでこれも試みてみたが、問題点は桁上げ、借りなどの処理が必要で、特に複数のワードを使ってとなると仲々大変である。

### 1. 2 Jの巨大数計算機能の利用

Jには、x仕様の数という巨大数の加減乗除を行う非常に優れた機能がある。われわれとして、これを利用しないほうはない。

Jの巨大数計算の能力は大変なものである。例えば、階乗の計算について、大野の書[3]p. 115で一般のBASICでは34の階乗でオーバーフローしてしまうと言っている。

ところが、Jではつぎのとおり、巨大数計算のx機能により最後の桁まで計算してくれる。

```
! 100  
9. 33262e157
```

```
! 100x  
93326215443944152681699238856266700490715968264381621468592963895217599993229  
915608941463976156518286253697920827223758251185210916864000000000000000000  
0000
```

この結果は、多桁整数計算として有名なUBASICでの結果と一致している。[4]

[4] 木田祐司「多倍長計算 BASIC、ユーザーズマニュアル」p. 30、日本評論社(1995)。

## 2. Jによる多桁分数による $\pi$ 計算の実際

$\pi$  の計算式としては、大野栄一「パソコンで挑む円周率」から、次の有馬頼懂と会田安明のつぎの分数式により計算した。[1] p. 249

$$\frac{428224593349304}{136308121570117}$$

なお、円周率  $\pi$  はJでは実用上、あえてプログラミングすることなく、次のように手軽に値を得ることができるのは、言うまでもない。

```
pi =: 428224593349304x % 136308121570117x
```

```
43j40 " : pi
```

```
3. 1415926535897932384626433832756974344691
```

さて、有馬頼懂と会田安明の多桁の分数に対して、ここでは、  
分子 (=被除数) (DAL)は多ワード数 1ワード(5桁)×10ワード  
分母 (=除数) (DBL)はx表示の単独の数とする。

```
DA =: 42822, 45933, 49304
```

```
DB =: 13630, 81215, 70117
```

```
DAL =: DA, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

```
DBL =: x: (10000000000x * (0}DB)) + (100000x * (1}DB)) + (2}DB)
```

途中経過も含めた計算の実際は、以下のようなになる。プログラムは最後に示した。

```
DBL muldiv DAL
42822 45933 49304      0      0      0      0      0      0      0
)
(0), 0 -----
42822
   0
)
(1), 0 -----
4282245933
   0
)
(2), 3 -----
428224593349304
408924364710351
)
(3), 14159 -----
1930022863895300000
1929986693311286603
```

```

)
(4), 26535 -----
      3617058401339700000
      3616936005863054595
)
(5), 89793 -----
      12239547664540500000
      12239515160145515781
)
(6), 23846 -----
           3250439498421900000
           3250403466961009982
)
(7), 26433 -----
           3603146089001800000
           3603032577462902661
)
(8), 83275 -----
           11351153889733900000
           11351058823751493175
)
(9), 69743 -----
           9506598240682500000
           9506537322664669931

```

Residue: 60918017830069

Quotient: 0 0 3 14159 26535 89793 23846 26433 83275 69743

\*\*\* end \*\*\*

このような、多桁除算の商(Quotient)として、 $\pi$ の値が得られる。  
 なお、この値は小数点以下29桁まで正しい。

### 3. 級数による展開式—Machinの計算

$\pi$  の無限級数による近似式にはいろいろあるが、現在普通に行われるのは収束の速い  $\arctan$  の展開をもととした、Machin による式である。[5]

$$\frac{\pi}{4} = 4 \arctan \frac{1}{5} - \arctan \frac{1}{239}$$

つまり

$$\pi = 16 \left( \frac{1}{5} - \frac{1}{3 \cdot 5^3} + \frac{1}{5 \cdot 5^5} - \dots \right) - 4 \left( \frac{1}{239} - \frac{1}{3 \cdot 239^3} + \dots \right)$$

である。

[5] 高木貞治「解析概論」p. 186, 岩波書店(1986).

以上の式にしたがって、計算の過程を追ってみよう。

Machin の式は分数の級数近似であるが、ここでは J の x 機能を用いた長大整数の割り算と加減算として、J でプログラミングした。

分子に適当な大きな数、ここでは K=20 桁の数を掛けて整数計算として行う。

$$P = 16 * 1000000000000000000 * \left( \frac{1}{odd * 5^{odd}} - \frac{1}{odd1 * 5^{odd1}} \right)$$

$$Q = 4 * 1000000000000000000 * \left( \frac{1}{odd * 239^{odd}} - \frac{1}{odd1 * 239^{odd1}} \right)$$

where odd = 1, odd1 = 3, ...

また、奇数の組に対して、P の項と Q の項とに分けて、加算と減算を組み合わせる。そして、途中の経過が分かるようにした。

実行は次のようになる。

```
(20, 26) pitakagix 16
i:0, odd 1, odd1 3
P:
  32000000000000000000
   426666666666666667
  315733333333333377024
Q:
  1673640167364016736
   9766636714834
-----
314059702932606091264
```

i:2, odd 5, odd1 7

P:

102400000000000000

2925714285714286

314159177218320433152

Q:

102588926

1283

---

314159177218217845509

i:4, odd 9, odd1 11

P:

91022222222222

2978909090909

314159265261531103232

Q:

0

0

---

314159265261531103232

i:6, odd 13, odd1 15

P:

100824615385

3495253333

314159265358860517376

Q:

0

0

---

314159265358860517376

i:8, odd 17, odd1 19

P:

123361882

4415057

314159265358979464201

Q:

0

0

---

314159265358979464201

i:10, odd 21, odd1 23

P:

159783

5836

314159265358979618148

```

Q:
                                0
                                0
-----
          314159265358979618148
i:12, odd 25, odd1 27
P:
                                215
                                8
          314159265358979618355
Q:
                                0
                                0
-----
          314159265358979618355
i:14, odd 29, odd1 31
P:
                                0
                                0
          314159265358979618355
Q:
                                0
                                0
-----
          314159265358979618355
*** end ***

```

### 終わりにあたって

$\pi$  の多桁計算なるテーマを、夏の暑さ解消になどと始めたものの、大変なテーマであることが分かった。Jにより2つのやり方の仕組みが分かった程度であった。

大野栄一氏の本「パソコンで挑む円周率」には、一昔前のパソコンFM-11を用いてフロッピーファイルを動員して、F-BASICのプログラムにより、計算時間170日23時間をかけて、3万桁の円周率を計算したそうだ。その結果が8ページにわたって同書にのっている。とうていまねできるものではない。

それよりわたくしにとっては、ベックマンの「 $\pi$ の歴史」の面白さ、鳴海風「円周率を計算した男」の小説に、夏をすごせたのが収穫だった。

NB. 巨大数の掛け算 p.112 => Jのx表示で計算可能

NB. 9628321567893105316x \* 6321x

NB. 60860620630652318702436

NB. NB. 有馬、会田の円周率の近似式、大野栄一「パソコンで挑む円周率」 p.110

NB. 2015/6/29

NB. DBL muldiv DAL =>

NB. -----

NB. Residue: 60918017830069

NB. Quotient:0 0 3 14159 26535 89793 23846 26433 83275 69743

DA =: 42822, 45933, 49304

DB =: 13630, 81215, 70117

NB. DBL muldiv DAL

NB. 巨大数の割り算 p.113 => 多桁の割り算

muldiv =: 3 : 0

:

P =. x.

A =. y.

wr 6j0 " : A

N =. #A

NB. wr N

Q =. N # 0

S =. 0{A

Keta =. #S

T =. 0

ST =. ' '

SM =. ' '

i =. 0

while. i <: N

do.

if. 'q' = rd 1 do. '\*\*\* end \*\*\*' return. end.

NB. calculation -----

if. i = 0

do.

NB. wr S, (i){A

S =. 0{A

T =. (<. S % P) \* P

Q =. (<. S % P) (i) }"1 Q

else.

NB. wr S, (i){A

S =. (100000 \* (S)) + (i){A

T =. (<. S % P) \* P

Q =. (<. S % P) (i) }"1 Q

```

end.
ST =. ST, ((":S), (:":T)), '-
wr '(', (":i), '), ', (":i{Q), ' ', 50#'-'
NB.    wr (":i{Q), '-----'

S1 =. _5}. (":S)
S2 =. _5{. (":S)
T1 =. _5}. (":T)
T2 =. _5{. (":T)

NB. display -----
if. i = 0
do.
wr S2
wr T2
else.
KETA =. 5 * (i+1)
NSP =. KETA - (# ": S1, S2)
wr (NSP#' '), S1, S2
wr (NSP#' '), T1, T2
end.
if. i = N - 1 do. goto_fin. end.
S =. S - T
i =. i + 1
end.
label_fin.
wr 'Residue:', ": S - T
wr 'Quatient:', ": Q
'*** end ***'
)

```

```

pitakagix =: 3 : 0
:
K =. x.
'K0 K1' =. K
NB. K0, K1 =. 20, 26

N =. y.
PAI =. 0x
i =. 0
while. i < N
do.
    odd =. 1 + 2*i
    odd1 =. 2 + odd
wr 'i:', (': i),', odd ', (': odd),', odd1 ', (': odd1)
wr 'P:'
    P0 =. (16x * 10x^K0) % (odd * 5x^odd)
    wr PP0 =. ('.:K1), 'j0') ": P0
    PAI =. x: PAI + ('.: PP0)
    P1 =. (16x * 10x^K0) % (odd1 * (5x^odd1))
    wr PP1 =. ('.:K1), 'j0') ": P1
    PAI =. x: PAI - ('.: PP1)
    wr ('.:K1), 'j0') ":PAI

wr 'Q:'
    Q0 =. (4x * 10x^K0) % (odd * (239x^odd))
    wr QQ0 =. ('.:K1), 'j0') ": Q0
    PAI =. x: PAI - ('.: QQ0)
    Q1 =. (4x * 10x^K0) % (odd1 * (239x^odd1))
    wr QQ1 =. ('.:K1), 'j0') ": Q1
    PAI =. x: PAI + ('.: QQ1)

wr '-----'
wr ('.:K1), 'j0') ": PAI
if. 'q' = rd 1 do. '*** end ***' return. end.
    i =. i + 2
end.
'*** end ***'
)

```