

π 10 万桁計算

県立松戸高等学校 広川久晴

1 はじめに

今春県立松戸高校に転勤し、数学準備室にある青木洋子先生の蔵書で最初に目に入ったのは、金田康正『 π のはなし』(東京図書)であった。25 年程前になるが一宮商業高校で π の値を小数点以下 3 万桁計算したことがあった。当時パソコンはなく、一部屋使うミニコンと呼ぶもので行った。データ領域として使える記憶容量は 6k バイト、32767 までの整数しか覚えられないメモリが 6000 個と考えられる。このメモリを節約し、プログラムを 6 分割し、それぞれ十数時間も動かしてやっと求めた。

『 π のはなし』は以前読んでおり、いつか計算を再開したいと考えているうちに時は過ぎてしまった。世はパソコンの時代、同じパソコンでも 80 年代に全盛を極めた DOS で動く PC98 シリーズは今はなく、現代は Windows の時代。教科書にある BASIC が使えない、という時代である。我が家にとってある NEC の PC9821 を捨てる前に UBASIC を使って計算しようと考えた次第である。

2 UBASIC

UBASIC は、整数論が専門の木田祐司氏が巨大整数の計算のために作った言語である。10 進法で約 2500 桁までの整数が扱える。10 年ほど前、情報教育センター(現総合教育センター)でも UBASIC の講座を開いたことがあったが、DOS 上で動くため DOS マシンの衰退とともに使われなくなってしまった。

今回はこれを使うのも最後と思い、使ってみた。1つのメモリに 1000 桁入るので 10 万桁の計算も長さ 100 の配列ですむ。

3 公式

『 π のはなし』には 10 億桁の記録を樹立したときに使ったものは、ガウス・ルジャンドルの公式であるとのこと。これは、

$$a = 1, b = \frac{1}{\sqrt{2}}, t = \frac{1}{4}, x = 1$$

から始め、漸化式

$$a' = \frac{a+b}{2}, b' = \sqrt{ab}, t' = t - x(a-a')^2, x' = 2x$$

で計算し、 a' と b' が収束したとき、 $\pi = \frac{a'^2}{t'}$ となるという。この公式を自分で証明できないのが残念だが、今回はこれを使った。

エクセルで試してみるとあっという間に収束する。次表のようにたったの 4 回で 8 桁も求まる。

回数	A	B	T	X
1	1	0.707106781	0.25	1
2	0.853553391	0.840896415	0.228553391	2
3	0.847224903	0.847201267	0.228473291	4
4	0.847213085	0.847213085	0.228473291	8

$$\text{PI} = 3.141592654$$

有名なのはマチンの公式

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

である。ただし、

$$\tan^{-1} \frac{1}{x} = \frac{1}{x} - \frac{1}{3x^3} + \frac{1}{5x^5} - \frac{1}{7x^7} + \frac{1}{9x^9} - \dots$$

1873 年にシャンクスが 707 桁まで計算したが、1946 年に 528 桁目が間違っていたことがわかった歴史的な公式である。私が前回計算したときは、この変形であるガウスの公式

$$\frac{\pi}{4} = 12 \tan^{-1} \frac{1}{18} + 8 \tan^{-1} \frac{1}{57} - 5 \tan^{-1} \frac{1}{239}$$

を使った。

以上 2 つの公式は $\frac{\pi}{4} = \tan^{-1} 1$ と $\tan$ の加法定理 $\tan(\alpha + \beta) = \frac{\tan \alpha + \tan \beta}{1 - \tan \alpha \tan \beta}$ より確かめられる。マチンの公式は $\tan \alpha = \frac{1}{5}$ とすると、倍角の公式で

$$\tan 2\alpha = \frac{2 \tan \alpha}{1 - \tan^2 \alpha} = \frac{5}{12}$$

さらに

$$\tan 4\alpha = \frac{2 \tan 2\alpha}{1 - \tan^2 2\alpha} = \frac{120}{119}$$

つぎに

$$\begin{aligned} \tan\left(\frac{\pi}{4} - 4\alpha\right) &= \frac{\tan \frac{\pi}{4} - \tan 4\alpha}{1 + \tan \frac{\pi}{4} \tan 4\alpha} \\ &= \frac{1 - \frac{120}{119}}{1 + \frac{120}{119}} = -\frac{1}{239} \end{aligned}$$

これより

$$\frac{\pi}{4} - 4\alpha = -\tan^{-1} \frac{1}{239}$$

よって

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

ガウスの公式も

$$\tan \alpha = \frac{1}{18}, \tan \beta = \frac{1}{57}, \tan \gamma = \frac{1}{239}$$

となるので、マチンの公式に戻る。

$$\tan(\alpha + \beta) = \frac{3}{41}, \tan(2\alpha + 2\beta) = \frac{123}{836}, \tan(3\alpha + 2\beta) = \frac{122}{597}$$

$$\tan(3\alpha + 2\beta - \gamma) = \frac{1}{5}$$

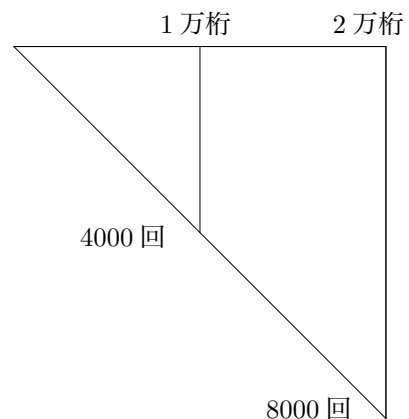
4 収束

(1) マチンの公式

$\tan^{-1} \frac{1}{5}$ の項は 5 で割って足したり引いたりする。10 進法の場合 5 で割るのはとても楽、2 倍して 1 桁ずらすだけである。この項の計算時間をゼロと考えると計算時間のほとんどは 239 で割る時間である。

$\log_{10} 239 = 2.3 \cdots \cdots$ なので1回割ると2.3桁進む。1万桁求めるには4000回の割り算が必要である。この4000回、初めは頭から割るが、だんだん先頭に0が並ぶので早くなってくる。しかし1万桁と2万桁を比べると、2万桁は8000回の割り算が必要なのだがその初回が1万桁の2倍かかるので右図のように計算時間は4倍になる。一般に桁数が N 倍になれば計算時間は N の2乗になる。

$\tan^{-1}$ を使うときはすべて同じである。



(2) ガウス・ルジャンドルの公式

収束は早かった。漸化式 1 回進むと桁数が 2 倍になるのである。この計算の前に初期値 $\frac{1}{\sqrt{2}}$ を求めなければならない。この求め方までは本にはなかった。まず思いつくのは右記の開平法だがプログラムが面倒なのでやめた。そこでニュートン法で求めることにする。ニュートン法なら割り算だけで求まるのである。

$$\begin{array}{r} 7 \\ 7 \\ \hline 1 \quad 4 \quad 0 \\ 0 \\ \hline 1 \quad 4 \quad 0 \\ \hline 1 \quad 4 \quad 1 \\ \hline 1 \quad 4 \quad 1 \end{array}$$

	7	0	7	1	
0.	50	00	00	00	00
	49				
	1	00			
		0			
	1	00	00		
		98	49		
		1	51	00	
		1	41	41	
			10	59	00

$\frac{1}{\sqrt{2}}$ は $y = x^2 - \frac{1}{2}$ と x 軸の交点なので、適当な初期値から始めて、接線と x 軸の交点を新たな x とする。これを繰り返せば x は $\frac{1}{\sqrt{2}}$ に収束する。具体的な漸化式は $x' = \frac{x}{2} + \frac{1}{4x}$ である。収束の判定は $x_k - x_{k+1}$ が 10 万桁以下となったら止める。

$x_0 = 0.75$ から実行させ、頭からいくつめが異なっているか表示させたところ

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 4, 7, 13, 26, 51, 102

であった。1 は 1000 桁を表すので合っている桁数が 2 倍ずつ進んでいることがわかる。ニュートン法では交点に近づくほど曲線と直線の差が小さくなるのである。

ニュートン法は現在、数 B の「数値計算とコンピュータ」の章に入っているので授業ではほとんど扱われていないと思われるがもっと表に出てほしい題材である。

5 掛け算

2 つの配列だけで掛け算をしたいので、下の桁から答えを出していった。右のように 5×9 , 3×8 , 2×7 の和 83 を求め、その 1 の位 3 が答えの小数第 4 位として確定するので 0.532 の右を 3 に置き換える。つぎに、今の 10 の位 8, 5×8 , 3×7 の和 69 を求め、その 1 の位 9 が確定するので 0.5323 の 2 を 9 に置き換える。最後は、今の 10 の位 6, 5×7 の和 41 を求め、0.5393 の 53 を 41 に置き換えた。下の桁が合わないが先頭の 0.419 までは合っている。なお、実際の計算はひとつの数が 1000 桁である。

	0.	5	3	2	
×	0.	7	8	9	
				4	5
				2	4
				1	4
			4	0	
			2	1	
		3	5		
	0.	4	1	9	3

6 割り算

最も苦勞したのが割り算であった。1000 桁が 100 個並んだ数同士の商を求める。苦勞したのは 1 個ずつの商を立てることである。例えば $654 \div 876$ で最初の位の商を立てるとき、私たちは $65 \div 8$ ではなく、876 は 900 に近いので $65 \div 9$ と考えて 7 を立てる。しかし計算機に、9 に近いときは、というのは無理である。 $65 \div 8$ しか見られないので機械は 8 を立てる。余りが負になったら商を 1 減らし、余りに割る数を足し戻す。

これは 1 回でよいのだろうか。10 進だと $91 \div 19$ が最大誤差である。 $91 \div 10$ は 9、実際の商は 4 なので 5 回の足し戻しが必要なきがある。ただ今回は 2000 桁 $\div$ 1000 桁で 1000 桁を立てるので足し戻しは 1 回ですむと考えたがだめであった。初めの 1000 桁が 1 で始まり、次の 1000 桁が 9 で始まれば同じ状況になる。実際の 10 万桁の計算では 25000 回程商を立てて 2 回足し戻したのが 12 例あった。これだけ少ないのはほとんどが 8 から始まる数で割っているからであろう。テストランの最中、 $\pi = \frac{a^2}{t}$ を求める計算では 3 回の足し戻しもあった。これは t が 2 から始まる数だからであろう。

7 実行

テストで1万桁の計算は6分で終わった。気を良くして10万桁に臨んだところ、なんと8時間8分もかかってしまった。分にすれば488分。桁数が10倍になっても計算時間は10数倍で済むはずが、誤算であった。やはり平方根を求めるときに1発の開平法ですむところを、10数回の割り算で収束させていった罰である。それでも前回、3万桁を100時間以上かけて求めたことに比べれば天国である。

なお、収束の桁数は1000桁単位で1, 1, 1, 1, 1, 1, 2, 4, 7, 14, 28, 56, 102と進んだ。ニュートン法と同じく1回で求まる桁数は2倍になっていることが分かる。

前回の結果はフロッピーディスクがない時代だったので紙テープしかなかったが、今回の結果はきちんとフロッピーディスクとUSBメモリにとってある。

今回は、NEC98シリーズのパソコン・DOS・ソフトはUBASICで求めた。どれもみな過去の遺物である。本来ならこの原稿は10年前に出すべきものである。すでにUBASICのプリント命令を受け付ける印刷機はなく、テキストファイルに落とした結果をエクセルを使って編集し印刷した。そのために古いパソコンと今使っているパソコンの間を行ったり来たりしたのは滑稽であった。

付録としてプログラムの初めの部分、 π 小数点以下の初めの4000桁と、96001桁から100000桁を載せておく。なお興味ある方には10万桁のデータをメールで送ることも可能なので連絡下さい。

8 今後

『 π のはなし』によれば10年以上前にパソコンで100万桁計算した人がいるという。それに挑戦するつもりは毛頭ないが、きちんと開平のプログラムを書けば記憶容量の関係から25万桁までは求められそうである。ただ今回の10万桁までの結果は『 π のはなし』にあるものと同じだったので正しいが、それ以上の桁は書いていなかった。そのため、2つの公式で計算しその結果を比較し、数値が正しいことを自分で検証しなければならない。そこまでの時間とエネルギーがあるかどうか心配である。

また今回はUBASICを使ったが、次回発表するなら誰でもがいつでも使えるビジュアルベーシックを使うべきであろう。計算速度も速いしメモリもふんだんにある。しかし、心配は配列である。1メモリに10進で4桁記憶するとすれば20万桁を記憶するのに長さ5万の配列が必要となる。こんなに長い配列が使えるかどうか、心配はそこである。そこに制限がなければ100万桁も夢ではないように思われる。

付録

UBASIC のプログラム, 初めの部分

```
5 '-program pi main 060909
10 word 210:dim A(103),B(103),C(103),D(103),T(103)
20 K=10^1000
30 A(1)=K:T(1)=K¥4:X=1:print time
40 open "B:ROD25MA" for input as #1
50 for I%=1 to 102:input #1,B(I%):next:close
100 '-kurikaesi
110 for I%=1 to 102:C(I%)=A(I%):next
120 gosub *Atbw2
200 '-kakeru
210 R=0:M=0
220 for I%=102 to 1 step -1
230 for J%=1 to I%:S=(B(I%+1-J%)*C(J%))¥K:N=res
240 R=R+S:M=M+N:next J%
250 S=M¥K:D(I%+1)=res:M=R+S:R=0
260 next I%
270 R=M¥K:D(1)=res
280 if R>0 then print "e-kakeru":stop
```

付

ガウス・ルジャンドルの公式の解説, 現代の π の計算方法はインターネットで以下のサイトに詳しく載っている。大浦拓哉「円周率の公式と計算法 <http://www.kurims.kyoto-u.ac.jp/~ooura/pi04.pdf>

