3.4 いよいよ π の精確な値を？

もう一度 π を求めるマチンの式

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

に登場してもらおう。

われわれは配列を効果的に使うことで、$1/(2n-1)239^{2n-1}$ を小数点以下いくらでも計算できるようになっている。当然のことながら $4/(2n-1)5^{2n-1}$ の計算も可能だ。マチンの式ではこれらの差を計算するわけだが、これも配列の計算で十分対処できる。そして n が大きくなるにしたがって、$4/(2n-1)5^{2n-1} - 1/(2n-1)239^{2n-1}$ の結果をを交互に足したり引いたりしている。しかし、これとて配列を使えば解決する。最後に、ここまでの値を 4 倍するのだが、配列の掛け算に自信がなければ、ここまでの値を 4 回足せば済む。これで π の精確な値が求められるのだ。

何を言いたいのかって？　要するにいまのわれわれには、π の計算に必要でありながら未だ知らない知識はないということだ。これで π の精確な値を望むだけ計算できる。本当だ。ただし配列の値を関数に受け渡す話をしていないので、いまプログラムを組んだら `main()` メソッドにすべてを詰め込まなくてはならないだろう。それは、あまり気持ちのよいものではないが、一刻も早く π の値を 1000 桁ほど計算したいのなら、これまでの知識で何とかなる。

TRY!　π の値を、小数点以下 1000 桁まで計算したくないかい？

π の値を小数点以下 1000 桁まで計算するには、配列は 250 個必要になる。なぜなら、一つの配列で 4 桁を記憶するからだ。もっとも **Java** においては、`int` 型でも 9 桁までは保証される。この豆旅では 4 桁仕様の `int` に乗っているのだが、君たちが 9 桁仕様の `int` に乗り換えるなら配列は 112 個で済む。

ではマチンの式は、第何項まで計算する必要があるのだろう。対数を使って大雑把に計算しておこう。$1/(2n-1)5^{2n-1}$ の方が $1/(2n-1)239^{2n-1}$ より収束が遅いので、$1/(2n-1)5^{2n-1}$ が 10^{-1000} より小さくなれば、その先の計算は不要になる。よって

$$\frac{1}{(2n-1)5^{2n-1}} < 10^{-1000}$$

を解けばよい。ただし n が十分大きい場合、$(2n-1)$ の桁数は 5^{2n-1} の桁数よりはるかに小さい。そこで計算を簡単にするためにも

$$\frac{1}{5^{2n-1}} < 10^{-1000}$$

を解けば十分である。

両辺について底 10 の対数をとると

$$\begin{aligned}
-(2n-1)\log_{10}5 &< -1000 \\
2n-1 &> \frac{1000}{\log_{10}5} \\
n &> \frac{1}{2}\left(\frac{1000}{\log_{10}5}+1\right) \\
n &> \frac{1}{2}\left(\frac{1000}{0.69897\ldots}+1\right) \\
n &> 715.8\ldots
\end{aligned}$$

であるから、716 項あたりで 10^{-1000} 程度の精度が得られる。余裕を持って 720 項まで計算をしておけば十分過ぎるだろう。

ここで 720 項で十分という確信が持てるだろうか？　もしかしたら 800 項先まで計算をすると、繰り上がりの影響で小数点以下 1000 桁の数字が増えたりしないのだろうか。とくに

$$1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}+\cdots \tag{1}$$

という級数を知っている者なら気になるはずだ。

(1) は 1 億項まで加えると 10^{-8} の精度になってくる。しかしこの時点では 10^{-8} どころか 10^{-7} の桁さえ決定しない。なぜなら 1 億項から先を加え続けると、10^{-7} の桁が繰り上がるからだ。それどころか、どの桁も永久に決定しないのだ。その理由はもちろん、(1) が発散する級数だからである。

ではマチンの式は発散しないのか？　その通り、発散しないのである。だったら違いはどこにあるのだろう。これは非常に難しい問題である。もちろんこの豆旅では、その解明を目的としていない。いまは、より精度の高い π の値をコンピュータに計算させるのが目的なのだ。

さて、配列を 250 個ほど用意すると小数点以下 1000 桁まで格納できる。そこでマチンの式の 720 項までを計算すれば、小数点以下 1000 桁の精度を確保できる。これでなんとかなりそうだ。ただ

し前に述べたように、ここまでの知識では野暮ったいプログラムにしかならない。`main()` メソッドにすべてを詰めこまなくてはならないからだ。豆旅のこの地点で、π の小数点以下 1000 桁を期待した君たちには申し訳ないが、野暮な長ったらしいプログラムを載せるわけにはいかないのだ。せっかくの観光地を目前にして、立ち入り禁止にしたみたいで心苦しいが。

正しいアルゴリズムによるプログラミングリストは、書店で簡単に見つけられるはずだ。いま妙なプログラムで苦労するより、少し理解できない部分を含む正しいプログラムを動かすほうがよい。いずれにしても、この豆旅は秘境を巡るものではないのだ。さらに知りたい人には、別のツアーをお勧めする。