

級数計算遊び
紙と鉛筆と電卓で遊ぶ級数集約法
ver1.2

序言

最近、紙と鉛筆と電卓で計算をする機会がめっきり減った。計算機やソフトが良くなったので、ぐにそれに頼ってしまう。公式辞典を見ると様々な公式が「これでもか！」と云わんばかりに並んでいる。級数による π の公式などはその数と種類に頭がクラクラする程で困ったものだ。そこで少し紙と鉛筆と電卓で遊んでみようという趣旨だ。ただそのままいろいろ公式の例を計算するのはつまらないから、ある種の級数の計算を比較的最近に使われている級数集約法と呼ばれているらしい手法で計算する。これによるとそれらの級数計算が本質的に上三角行列の積に帰着され、整数の計算に転換できる。もともとは計算機での計算効率をよくする目的で紹介された手法であるが、手計算してみてもなかなか面白いものなので取り上げてみることにする。

入れ子

多項式

$$f(x) = 7x^6 + 6x^5 + 5x^4 + 4x^3 + 3x^2 + 2x + 1$$

に対して $f(a)$ を計算するのにホーナー法を用いることにする。これは多項式

$$a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$$

の値を計算する際に

$$(\dots((a_0x + a_1)x + a_2)x + \dots + a_{n-1})x + a_n$$

の形に変形して行うもので、利点としてかけ算の回数をそのまま計算するよりも減らすことができるというものである。

このような形は入れ子(ネスト)と呼ばれる。実際の計算では先の $f(x)$ に対して $f(21)$ を求めてみよう。そのままの計算では

$$\begin{aligned} f(21) &= 7 \times 21^6 + 6 \times 21^5 + 5 \times 21^4 + 4 \times 21^3 + 3 \times 21^2 + 2 \times 21 + 1 \\ &= 7 \times 85766121 + 6 \times 4084101 + 5 \times 194481 + 4 \times 9261 + 3 \times 441 + 2 \times 21 + 1 \\ &= 600362847 + 24504606 + 972405 + 37044 + 1323 + 42 + 1 \\ &= 625878268. \end{aligned}$$

ホーナー法では

$$\begin{aligned} f(21) &= ((((((7 \times 21 + 6) \times 21 + 5) \times 21 + 4) \times 21 + 3) \times 21 + 2) \times 21 + 1) \\ &= (((((153 \times 21 + 5) \times 21 + 4) \times 21 + 3) \times 21 + 2) \times 21 + 1) \\ &= (((3218 \times 21 + 4) \times 21 + 3) \times 21 + 2) \times 21 + 1 \\ &= ((67582 \times 21 + 3) \times 21 + 2) \times 21 + 1 \\ &= (1419225 \times 21 + 2) \times 21 + 1 \\ &= 29803727 \times 21 + 1 \\ &= 625878268. \end{aligned}$$

実際に電卓片手に計算してみると、後者のほうがリズムよく計算できることが実感できると思う。

これを見てべき級数の近似値を求めることを考えるのは成り行きというもので

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11} + \frac{x^{13}}{13} - \frac{x^{15}}{15} + \dots$$

は

$$\arctan x = x((((((((((\dots)x^2 - \frac{1}{15})x^2 + \frac{1}{13})x^2 - \frac{1}{11})x^2 + \frac{1}{9})x^2 - \frac{1}{7})x^2 + \frac{1}{5})x^2 - \frac{1}{3})x^2 + 1)$$

とすればさぞかし効率のよいものになるような気がする。ところが実際にかけ算の回数は減るのだけれども計算の手続きはまだまだ難しい。計算機にのせる手順を考えても、効率を度外視すれば、はるかに元の式のほうが簡単である。例えば Mathematica で $x = 1$ において π の近似値を求めるのに、級数を 1000 項ほど計算させるには

```
a=0;
Do[a=a+(-1)^k/(2k+1),{k,0,999}]
N[4*a,10]
```

でよい。ちなみにこの結果は 3.14059... だからまあ収束の遅さだけがよくわかるようなものだが、一方のネストはちょっとした慣れが必要となる。

級数集約法

さて、前置きとホーナー法はこれくらいにしておいて、ある種の級数の効率的な計算方法を考えてみるのだが、そこはそれ、すでによく研究されているからそれを使う。ここでは広島市立大学の右田氏等の論文、「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、情報処理学会論文誌, vol40.No12,1999」で示された方法を簡単に使ってみる。難しい部分もあるが、ここでは遊びの延長で使う程度のレベルに留めるので心配は無用。

さて、それに従うと、これから対象とする級数は

$$\frac{1}{C_0}(A_0 + \frac{B_0}{C_1}(A_1 + \frac{B_1}{C_2}(A_2 + \dots$$

の入れ子の形にできるものとする。ここで A_k, B_k, C_k は高々数十桁の整数とする。

L 項までの部分和を S_L と書くことにすれば

$$S_L = \frac{1}{C_0}(A_0 + \frac{B_0}{C_1}(A_1 + \frac{B_1}{C_2}(A_2 + \dots + \frac{B_{L-3}}{C_{L-2}}(A_{L-2} + \frac{B_{L-2}}{C_{L-1}}A_{L-1})\dots))) \quad (1)$$

$$= \frac{1}{C_0}A_0 + \frac{1}{C_0}\frac{B_0}{C_1}A_1 + \frac{1}{C_0}\frac{B_0}{C_1}\frac{B_1}{C_2}A_2 + \dots + \frac{1}{C_0}\frac{B_0}{C_1}\dots\frac{B_{L-2}}{C_{L-1}}A_{L-1} \quad (2)$$

$$= \sum_{k=0}^{L-1} \frac{\prod_{l=0}^{k-1} B_l}{\prod_{l=0}^k C_l} \quad (3)$$

の形をしているものということになる。

そこで、(1) 式の構造に着目して

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

と置けば,

$$\begin{aligned}
S_L &= \frac{1}{C_0} \left(A_0 + B_0 \frac{1}{C_1} \left(A_1 + B_1 \frac{1}{C_2} \left(A_2 + \cdots + B_{L-3} \frac{1}{C_{L-2}} \left(A_{L-2} + B_{L-2} \frac{1}{C_{L-1}} \left(A_{L-1} + B_{L-1} \cdot 0 \right) \cdots \right) \right) \right) \right) \\
&= \frac{1}{C_0} \left(A_0 + B_0 \frac{1}{C_1} \left(A_1 + B_1 \frac{1}{C_2} \left(A_2 + \cdots + B_{L-3} \frac{1}{C_{L-2}} \left(A_{L-2} + B_{L-2} \frac{1}{C_{L-1}} \left(A_{L-1} + B_{L-1} \cdot R_L \right) \cdots \right) \right) \right) \right) \\
&= \frac{1}{C_0} \left(A_0 + B_0 \frac{1}{C_1} \left(A_1 + B_1 \frac{1}{C_2} \left(A_2 + \cdots + B_{L-3} \frac{1}{C_{L-2}} \left(A_{L-2} + B_{L-2} R_{L-1} \right) \cdots \right) \right) \right) \\
&= \frac{1}{C_0} \left(A_0 + B_0 \frac{1}{C_1} \left(A_1 + B_1 \frac{1}{C_2} \left(A_2 + \cdots + B_{L-3} R_{L-2} \right) \cdots \right) \right) \\
&\vdots \\
&= \frac{1}{C_0} \left(A_0 + B_0 \frac{1}{C_1} \left(A_1 + B_1 R_2 \right) \right) \\
&= \frac{1}{C_0} \left(A_0 + B_0 R_1 \right) \\
&= R_0.
\end{aligned}$$

さらに R_k が有理数であることから整数 α_k, β_k を用いて $R_k = \frac{\alpha_k}{\beta_k}$ と置き直してやると

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

となるからこれを行列表記して

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} B_k & A_k \\ 0 & C_k \end{pmatrix} \begin{pmatrix} \alpha_{k+1} \\ \alpha_{k+1} \end{pmatrix}$$

を得る. これを用いて S_L を書き直してやると

$$\begin{aligned}
\begin{pmatrix} \alpha_0 \\ \beta_0 \end{pmatrix} &= \begin{pmatrix} A_0 & B_0 \\ 0 & C_0 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \\
&= \begin{pmatrix} A_0 & B_0 \\ 0 & C_0 \end{pmatrix} \begin{pmatrix} A_1 & B_1 \\ 0 & C_1 \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \\
&= \begin{pmatrix} A_0 & B_0 \\ 0 & C_0 \end{pmatrix} \begin{pmatrix} A_1 & B_1 \\ 0 & C_1 \end{pmatrix} \begin{pmatrix} A_2 & B_2 \\ 0 & C_2 \end{pmatrix} \begin{pmatrix} \alpha_3 \\ \beta_3 \end{pmatrix} \\
&= \vdots \\
&= \begin{pmatrix} A_0 & B_0 \\ 0 & C_0 \end{pmatrix} \begin{pmatrix} A_1 & B_1 \\ 0 & C_1 \end{pmatrix} \begin{pmatrix} A_2 & B_2 \\ 0 & C_2 \end{pmatrix} \cdots \begin{pmatrix} \alpha_L \\ \beta_L \end{pmatrix} \\
&= \begin{pmatrix} A_0 & B_0 \\ 0 & C_0 \end{pmatrix} \begin{pmatrix} A_1 & B_1 \\ 0 & C_1 \end{pmatrix} \begin{pmatrix} A_2 & B_2 \\ 0 & C_2 \end{pmatrix} \cdots \begin{pmatrix} 0 \\ 1 \end{pmatrix}
\end{aligned}$$

と変形することができるから, あとは最後に $S_L = \frac{\alpha_0}{\beta_0}$ を計算してやればよい. ここであとあと便利

なので $\left\| \begin{pmatrix} a \\ b \end{pmatrix} \right\| := \frac{a}{b}$ と定義しておく. さてこのように行列で表した形は非常に分かりやすく, かつ上三角行列の積がまた上三角行列になることを考えれば計算もし易い.

論文ではこの手法で隣接二項の計算を順次行う方法で 2^n 個の項を 2 分しながらまとめていくことで計算の効率化が図られるとしている。なるほどその通りで優れた手法である。なんでもこの手法で π の値が以前に 10 億桁計算されたい。

計算例

ここでは計算量などこれ以上深入りせずに、この手法でいくつかの級数の計算を主に手計算で行う。今見た通りこの手法によると複雑な級数の計算に対して

- 上三角行列の計算に帰着できる。
- 多倍長整数の乗法が多く現れるが、除法は最後の一回で済む。

という利点がある。

さてそれでは実際にいくつかの級数の S_8 あたりまでを計算してみる。

例

$$\frac{\pi}{6} = \arcsin \frac{1}{2} = \sum_{n=0}^{\infty} \frac{(2n)!}{2^{2n}(n!)^2} \frac{1}{2n+1} \frac{1}{2^{2n+1}}$$

を利用して π の近似値を計算してみる。

$$A_k = 1, B_k = (2k+1)^2, C_k = \begin{cases} x & (k=0) \\ 2k(2k+1) \cdot 4 & (0 < k) \end{cases}$$

のようにとればよいから。

$$\begin{aligned} \frac{\pi}{6} &\approx \left\| \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 3^2 & 1 \\ 0 & 2 \cdot 3 \cdot 4 \end{pmatrix} \begin{pmatrix} 5^2 & 1 \\ 0 & 4 \cdot 5 \cdot 4 \end{pmatrix} \begin{pmatrix} 7^2 & 1 \\ 0 & 6 \cdot 7 \cdot 4 \end{pmatrix} \cdots \begin{pmatrix} 15^2 & 1 \\ 0 & 14 \cdot 15 \cdot 4 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 9 & 25 \\ 0 & 48 \end{pmatrix} \begin{pmatrix} 1225 & 193 \\ 0 & 13440 \end{pmatrix} \begin{pmatrix} 9801 & 521 \\ 0 & 126720 \end{pmatrix} \begin{pmatrix} 38025 & 1009 \\ 0 & 524160 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 11025 & 337737 \\ 0 & 645120 \end{pmatrix} \begin{pmatrix} 372683025 & 282976569 \\ 0 & 66421555200 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 4108830350625 & 22436136605255625 \\ 0 & 42849873690624000 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \frac{22436136605255625}{42849873690624000} \\ &= 0.523598663726 \cdots \end{aligned}$$

したがって、 $\pi \approx 3.14159198235838 \cdots$ という結果を得る。出てきた数を見れば高々 8 項の計算でできたにしてはよい近似となっている。それだけ元の級数の収束が速いということになる。

ここでの計算は、紙と鉛筆とできれば 12 桁の電卓（できれば桁数は多いにこしたことはない）と、なによりも暇な時間があれば良い。上記の例で 12 桁でそのまま計算出来ないのは乗法が 4 回で除法が 1 回だけである。ちなみによく知られたことなので蛇足にはなるけれども 372683025×11025 の計算を 5 桁区切りで行う場合について以下に示しておく。

$$\begin{array}{r} 3726 \ 83025 \\ \times \qquad \qquad 11025 \\ \hline 9153 \ 50625 \\ 410 \ 79150 \\ \hline 410 \ 88303 \ 50625 \end{array}$$

この程度の計算ならば慣れれば数分とかからない。区切りの桁数は電卓の桁数に応じて変えてやればよい。最後の除法だけはまともに電卓だけでやるのは少し難儀で3桁くらい計算したところだいたい嫌気がさしてくる。上の例だと除数の下3桁の0はあとで調整するとして今度は商は1桁ずつ考えていくので除数を10桁区切りにしておいても大丈夫だから

$$\begin{array}{r}
 4284 \ 9873690624 \) \ 22436136605255625 \\
 \underline{214249368453120} \\
 101119975994362 \\
 \underline{85699747381248} \\
 154202286131145 \\
 \vdots
 \end{array}$$

というような感じになるだろうか(計算間違いしていたらゴメン!). 勿論この他にも数桁区切りで計算する方法もある。したがって条件が許せば適当な計算ソフトを使用したほうがよい。Windows付属の電卓でも十分だ。それでもこの例では計算すべてを手作業でこなしても慣れれば約30分程でできる。ちなみに電卓でやらずに算盤でやるほうが速いという人もいるだろう。なにせあれはある意味桁数なんか気にせずできるスーパーツールだから。

さて、なにはともあれ、一応検算を Mathematica にやらせてみる。フリーの MuPAD でもできるのだろうけど私は知らない。興味ある人は <http://www5a.biglobe.ne.jp/ogose/mupad/index.html> などを見ましよう。

```

m={{1,1},{0,2}};
Do[m=m.{{(2k+1)^2,1},{0,8k(2k+1)}},{k,1,7}]
m=N[m.{0,1}/.{a_,b_}->a/b,20]

```

で 0.5235986637263970762 となった。全然再帰的に計算していないけれど8項の検算ならそれでよいのだ。

例

$$\frac{1}{e} = \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!}$$

を利用して自然対数の底 e の近似値を計算してみる。

$$A_k = 1, B_k = -1, C_k = \begin{cases} 1 & (k = 0) \\ k & (0 < k) \end{cases}$$

のようにとればよいから今度はちょっと簡単だ.

$$\begin{aligned}
 \frac{1}{e} &\approx \left\| \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 0 & 3 \end{pmatrix} \cdots \begin{pmatrix} -1 & 1 \\ 0 & 7 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 0 & 20 \end{pmatrix} \begin{pmatrix} 1 & 6 \\ 0 & 42 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 1 & 2 \\ 0 & 6 \end{pmatrix} \begin{pmatrix} 1 & 174 \\ 0 & 840 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 1 & 1854 \\ 0 & 5040 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \frac{1854}{5040} \\
 &= 0.36785714 \dots
 \end{aligned}$$

したがって $e \approx 2.7184460$. となるが, 簡単だから 16 項くらい計算しても手間は知れている. でもこのくらいなら, 少ない項数をもとの級数のまま計算したほうが得かもしれない.

例によって Mathematica で検算すると

```

m={{-1,1},{0,1}};
Do[m=m.{{-1,1},{0,k}},{k,1,7}]
m=N[m.{0,1}/.{a_,b_}->a/b,20]

```

で, 0.36785714285714285714285714 となる. どうせだからもう少し項を多くして近似計算させると

```

m={{-1,1},{0,1}};
Do[m=m.{{-1,1},{0,k}},{k,1,15}]
m=N[m.{0,1}/.{a_,b_}->b/a,20]

```

で, $e \approx 2.71828182845937871587 \dots$ となった.

例

$$G = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)^2}$$

を利用してカタランの定数 G の近似値を計算してみる. 何の数? って感じだが, まあ本で拾った公式だしどうでもよいからやってみる. こういう遊びにいろいろな価値を考えては駄目だ.

$$A_k = 1, B_k = -(2k+1)^2, C_k = (2k+1)^2$$

のようにとればよい.

$$\begin{aligned}
 G &\approx \left\| \begin{pmatrix} -1^2 & 1 \\ 0 & 1^2 \end{pmatrix} \begin{pmatrix} -3^2 & 1 \\ 0 & 3^2 \end{pmatrix} \begin{pmatrix} -5^2 & 1 \\ 0 & 5^2 \end{pmatrix} \begin{pmatrix} -7^2 & 1 \\ 0 & 7^2 \end{pmatrix} \cdots \begin{pmatrix} -15^2 & 1 \\ 0 & 15^2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 9 & 8 \\ 0 & 9 \end{pmatrix} \begin{pmatrix} 1225 & 24 \\ 0 & 1225 \end{pmatrix} \begin{pmatrix} 9801 & 40 \\ 0 & 9801 \end{pmatrix} \begin{pmatrix} 38025 & 56 \\ 0 & 38025 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 11025 & 10016 \\ 0 & 11025 \end{pmatrix} \begin{pmatrix} 372683025 & 2069856 \\ 0 & 372683025 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 4108830350625 & 3755613340800 \\ 0 & 4108830350625 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \frac{3755613340800}{4108830350625} \\
 &= 0.914034657 \cdots
 \end{aligned}$$

ただしこの値はあまり良くない. 級数の収束が遅いためだ.

例

$$\frac{\pi}{2} = \sum_{n=1}^{\infty} \frac{2^n}{2^n C_n \cdot n}$$

を利用して π の近似値を計算してみる. ちなみにこの C は組合せ記号だ. C_k と混同せぬように!

$$A_k = 2^k, B_k = k(k+1), C_k = 2k(2k-1)$$

のようにとればよい.

$$\begin{aligned}
 \frac{\pi}{2} &\approx \left\| \begin{pmatrix} 1 \cdot 3 & 2^1 \\ 0 & 2 \cdot 1 \end{pmatrix} \begin{pmatrix} 2 \cdot 3 & 2^2 \\ 0 & 4 \cdot 3 \end{pmatrix} \begin{pmatrix} 3 \cdot 4 & 2^3 \\ 0 & 6 \cdot 5 \end{pmatrix} \begin{pmatrix} 4 \cdot 5 & 2^4 \\ 0 & 8 \cdot 7 \end{pmatrix} \cdots \begin{pmatrix} 7 \cdot 8 & 2^8 \\ 0 & 16 \cdot 15 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 12 & 32 \\ 0 & 24 \end{pmatrix} \begin{pmatrix} 240 & 640 \\ 0 & 1680 \end{pmatrix} \begin{pmatrix} 1260 & 6144 \\ 0 & 11880 \end{pmatrix} \begin{pmatrix} 4032 & 45056 \\ 0 & 43680 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 2880 & 61440 \\ 0 & 40320 \end{pmatrix} \begin{pmatrix} 5080320 & 325140480 \\ 0 & 518918400 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \left\| \begin{pmatrix} 14631321600 & 32818751078400 \\ 0 & 20922789888000 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\
 &= \frac{32818751078400}{20922789888000} \\
 &= 1.568564769 \cdots
 \end{aligned}$$

で $\pi \approx 3.13713 \cdots$ となった. 見ての通り計算の大変さの割には近似が良くない.

例

$$\frac{\pi}{4} = \arctan \frac{1}{2} + \arctan \frac{1}{3}$$

を利用して π の近似値を計算してみる.

$$\arctan \frac{1}{x} = \sum_{n=0}^{\infty} (-1)^n \frac{1}{(2n+1)x^{2n+1}}$$

だから

$$A_k = 1, B_k = -(2k + 1), C_k = \begin{cases} x & (k = 0) \\ (2k + 1)x^2 & (0 < k) \end{cases}$$

のようにとればよい.

$$\begin{aligned} \arctan \frac{1}{2} &\approx \left\| \begin{pmatrix} -1 & 1 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} -3 & 1 \\ 0 & 3 \cdot 4 \end{pmatrix} \begin{pmatrix} -5 & 1 \\ 0 & 5 \cdot 4 \end{pmatrix} \begin{pmatrix} -7 & 1 \\ 0 & 7 \cdot 4 \end{pmatrix} \cdots \begin{pmatrix} -15 & 1 \\ 0 & 15 \cdot 4 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 3 & 11 \\ 0 & 24 \end{pmatrix} \begin{pmatrix} 35 & 23 \\ 0 & 560 \end{pmatrix} \begin{pmatrix} 99 & 35 \\ 0 & 1584 \end{pmatrix} \begin{pmatrix} 195 & 47 \\ 0 & 3120 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 105 & 6229 \\ 0 & 13440 \end{pmatrix} \begin{pmatrix} 19305 & 113853 \\ 0 & 4942080 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 2027025 & 30796170885 \\ 0 & 66421555200 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \frac{30796170885}{66421555200} \\ &= 0.46364724210 \cdots \end{aligned}$$

$$\begin{aligned} \arctan \frac{1}{3} &\approx \left\| \begin{pmatrix} -1 & 1 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} -3 & 1 \\ 0 & 3 \cdot 9 \end{pmatrix} \begin{pmatrix} -5 & 1 \\ 0 & 5 \cdot 9 \end{pmatrix} \begin{pmatrix} -7 & 1 \\ 0 & 7 \cdot 9 \end{pmatrix} \cdots \begin{pmatrix} -15 & 1 \\ 0 & 15 \cdot 9 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 3 & 26 \\ 0 & 81 \end{pmatrix} \begin{pmatrix} 35 & 58 \\ 0 & 2835 \end{pmatrix} \begin{pmatrix} 99 & 90 \\ 0 & 8019 \end{pmatrix} \begin{pmatrix} 195 & 122 \\ 0 & 15795 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 105 & 73884 \\ 0 & 229635 \end{pmatrix} \begin{pmatrix} 19305 & 1433628 \\ 0 & 126660105 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} 2027025 & 9358305728760 \\ 0 & 29085593211675 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\| \\ &= \frac{9358305728760}{29085593211675} \\ &= 0.32175055398 \cdots \end{aligned}$$

で $\pi \approx 3.141591184 \cdots$ となった. 見ての通り計算の大変さの分だけまあまあよい近似だ. この式は有名な Machin の公式に似ているがどこかで関係があるのかも知れない (そりゃあるに決まっている! とにかくオイラーは偉い). π に関する公式の解説は例えばブルーバックスの一書でも参照してみたら良い. きっと浪漫あふれる解説が載っている. ということでそのあたりは適当な書物に譲るとして, 例によってここではそれ以上突っ込まない.

さて, こうしていくつかの計算をしてみると単純な作業であることはわかるが今ひとつ楽になった気はしない. それは当然大きな数を扱う為で, 手作業ではその煩わしさは仕方がないところだ. 元々は計算機での効率を考えた手法だから乗法の回数を減らすことや隣接二項の計算による桁数の類似が本質的な部分だ. それだけに大きな数の計算で手間取っているのは本来の趣旨からはずれたところで苦労していることになる. それでも与えられた級数の項を頭から計算して小数をいじくるやり方はない方法をひとつ知ったことは大きな意義がある.

これを題材にして例えば, 「多倍長整数の四則演算ができるプログラムを書いてみる」とや「上記の計算法に従って級数の計算をするプログラムを書いてみる」となどを考えてみるのが発展

的な課題となる。そしてそのようなことができれば π の値をたくさん計算するプログラムも書けることだろう。ただし勿論のこととしてそのあたりの話は簡単ではない。効率を最大限に考えるならば、数の積をとっても様々な手法が存在している。ここでそれらを紹介することは適当でないし私の力量を遙かに超える。身の丈に合ったレベルで楽しむということが表題通り「遊び」の本質なのだ。

おまけ (交代級数の加速)

例によっておまけの話だ。級数の効率的な計算を考えるならば本来は級数の収束改善について触れなければならない。これもよく研究されていて、クンマー、オイラー、エイトケン、etc 有名なものがいくつか存在している。詳しくは適当な書物を参照せよ。簡単のために交代級数のオイラー変換だけ触れておく。ややこしいことは抜きにして交代級数

$$S = \sum_{n=0}^{\infty} (-1)^n a_n$$

に対して

$$S_n = \sum_{k=0}^{n-1} \sum_{j=0}^k \frac{(-1)^j}{2^{k+1}} {}_k C_j a_j$$

または

$$S_n = \sum_{k=0}^{n-1} \sum_{j=k+1}^n \frac{(-1)^k}{2^n} {}_n C_j a_k$$

となることを指すのだが、(ちなみにこの2つの式が等しいことを示すのは少し厄介だ。) とにかくこれを用いて例えば

$$\frac{\pi^2}{12} = \sum_{n=0}^{\infty} \frac{(-1)^n}{(n+1)^2}$$

による π の計算をしてみると

t=0;

Do[Do[t=t+(-1)^j*Binomial[k,j]/((2^(k+1))(j+1)^2),{j,0,k}],{k,0,14}]

N[Sqrt[12t],20]

の結果は 3.1415808191 である。わずかに 15 項分でこれなら立派というべきか。比較のために元の式のまま計算してみると、普通に計算させてもよいが折角だから $A_k = 1$, $B_k = -(k+1)^2$, $C_k = (k+1)^2$ として

m={{1,0},{0,1}};

Do[m=m.{{-(k+1)^2,1},{0,(k+1)^2}},{k,0,149}]

N[m.{0,1}/.{a_,b_}->Sqrt[12a/b],20]

としてみると、なんとか 3.1415504949 ということになった。150 項とってやっとこれだけだから厳密な評価は置いておくとしても収束改善の効果がわかる筈だ。

ただし、あたりまえのことだが元々収束の速い級数に適用しても効果は薄い。どころかプロセスが多い分遅くなってしまうこともある。このあたりは注意が必要だ。

さらにおまけのついでにオイラー変換を一般化した方法を含めて加速法を比較しておいた。多項式の取り方で変わってくるけれども一番上が通常のオイラー変換で、その下は違う多項式をとったものだ。たぶん下にいくほうが効率よくとってある筈だ。

```

s=0;
n=15;
Do[Do[s=s+((-1)^k)*Binomial[n,j]/((2^n)*(k+1)^2),{j,k+1,n}],{k,0,n-1}]
N[Sqrt[12s],20]

```

```

u=0;
n=15;
Do[Do[u=u+((-1)^k)*Binomial[n,j]*2^j/((3^n)*(k+1)^2),{j,k+1,n}],{k,0,n-1}]
N[Sqrt[12u],20]

```

```

v=0;
n=15;
Do[Do[v=v+((-1)^k)*Binomial[n+j,2j]*4^j*2^n/((n+j)((3+Sqrt[8])^n)*(k+1)^2),
{j,k+1,n}],{k,0,n-1}]
N[Sqrt[12v],20]

```

これらの結果は以下の通りだからやはり一番下が良いようだ。ちなみに一番下はチェビシェフ多項式を利用しているものである。このくらいの近似値を得ようとするとな元の級数では 2000 項以上は計算しないと駄目だ。

3.1415808191072169047

3.1415926370758042156

3.141592653589960802

だから級数の計算も計算機に乗せる前の前処理が大切でこのあたりはよく研究されているようだ。それでもここに揚げたものは個人的な楽しみで計算したものであるから、もしかしたらどこかで重大なミスを含んでいるかも知れない。実際に使う人は必ずきちんとした理論を身につけて誤りの無きように願いたい。間違ってもこの記事のせいにはするな！

さて、このあたりで紙面が尽きた感じだ。連分数展開のことや交代級数も含めた加速の話の続きとかまだまだ級数に関する話題は尽きないけれども、これ以上は趣味の性格が濃すぎるのでやめておく。

あとがきという名のいいわけ

余り役に立たないものを長々と書いた。余り役に立たないけれども面白いと思えるものがあればいいと思っている。最初に書いたけれども最近手動かすことがめっきり減ってしまった。計算機の発達した今にわざわざ手計算でもないだろうという向きもあるがまあ趣味でやる人がいても良い。文中の例で A_k , B_k , C_k の取り方について詳しく書かなかったけれどもまあ適当にとればいい。 $A_k = 1$ が多いのは特に理由は無い。そうしてあとのを決めているだけだ。Mathematica の使い方は相変わらず不慣れでもう少し熟達したいところだ。しかし高いソフトなのでなかなか使う機会が少ないからいつまでたっても覚えれないのだ。そういうことで少しミスもあるかも知れないのでお断りしておく。級数の収束の加速についてはもう少し詳しくやっても良かったけれども適当な書物もあるようだからおまけにしておいた。まあ単純とも言える変換で百倍も千倍も効率が良くなるのだから脱帽だ。思い切っているんな変換して Mathematica で計算実験してみてもよかった。まあそれは誰かに任せよう。

級数集約法を使って本当に計算機で級数を効率的に計算させようと思うとたぶん Binary Splitting のアルゴリズムや多倍長四則計算アルゴリズム、さらには積についての効率的な方法、詳しくは知ら

ないがFFTなどのアルゴリズムの理解が必要なのであろう。そういうものができる能力の持ち主がいれば自分のパソコンでぜひ円周率を何万桁あるいは何億桁?と計算してみしてほしいものだ。著者にはそのような能力は無いからぜひ誰かに教えてほしいものだ。

参考文献

- 右田・天野・浅田・藤野, 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, 情報処理学会論文誌, vol40.No12,1999
この文書の基本となった論文.
- 解析学1 総合図書 現代応用数学ハンドブック1
例に使った公式はこの本によった.
- <http://numbers.computation.free.fr/Constants/Miscellaneous/seriesacceleration.html>
おまけの交代級数の加速についてはここを参照した.
- H.Cohen,F.Rodriguez Villegas,D.Zagier, Convergence Acceleration of Alternating Series, Experimental Math,Vol.9(2000):No.1,p3-p12.
交代級数の加速についてはこの論文にも記述がある.
- K.Dilcher, Nested Squares and Evaluations of Integer Products, Experimental Math,Vol.9(2000):No.3,p369-p372.
直接触れなかったから参考文献というのも変だがネストに関係したなかなか面白い一文.
- Donald E. Knuth, 中川圭介 訳, 「準数値算法, 算術演算」, サイエンス社,1986年
意識せずともどこかで参考にしている …

2002-12-19 , Typeset by L^AT_EX 2_ε, 京都府立鳥羽高校定時制 稲葉芳成