情報処理2 第9回

Mathematica 体験 (2)

桂田(かつらだ) 祐史(まさし)

2012年6月13日

この授業用のWWWページは http://www.math.meiji.ac.jp/~mk/syori2-2012/

1 連絡事項

- 前回、時間配分計算を間違って、レポート課題8[1] をするための十分な時間が取れませんでした。締切りは来週19日 (火曜) に延びたので、じっくりやって下さい。
- レポート課題6B[2] の解説を、この資料の付録[3] につけておきますので、参考にして下さい (質問があれば、巡回時にいつでもどうぞ)。

2 前回のやり残し: 数学SNSで TEX 画像を使う

授業で使っている SNS システムは、阿原先生による改造で、TEX を利用した数式が使えるようになっています。せっかくだから体験してみましょう。前回の資料「数学SNSで TEX 画像を使う」[4] に飛びます。

3 Mathematica 体験

前回は、さらっと紹介して、(残っている課題がなくて余裕がある人は) 各自ということでしたが、今日は「電卓的な使用 5.11 結果の簡単化」[5] から順番に説明します。

[1] http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2012-08/node8.html
[2] http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2012-06/node8.html
[3] http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2012-09/node6.html
[4] http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2012-08/node10.html
[5] http://www.math.meiji.ac.jp/~mk/syori2-2012/mathematica/node27.html

4　レポート課題 9

- Oh-o! Meiji を使ってレポートを提出せよ。締め切りは 7 月 3 日 (火曜)18:00 とする (締切りはかなり先ですし、さらに延ばすかもしれません。本日の説明だけでは最後まで到達しないかもしれないので。あわてずにやって下さい。)。
- Mathematica に与えたコマンドと計算結果、その説明を TeX で書き、PDF ファイル (名前は `kadai9.pdf` とする) を提出する。
- 計算の仕方を工夫すること (関数を作ったり、`Table[]` を使ったり…)。
- (繰り返し) 結果が複雑な場合は、簡単化を試みること。
- (繰り返し) 検算が可能な問題については、検算もすること。— 時間に余裕が生じた場合は、ここを頑張ること。コンピューターを使う場合、筆算ではできないような検算も可能になる。

(1) Mathematica に、$\cos\dfrac{2\pi}{n}$ $(n=1,2,\dots,20)$ を計算させなさい。(結果を見て納得が行きますか?)

(2) $\displaystyle\sum_{k=1}^{3}\frac{1}{2^k}, \sum_{k=1}^{5}\frac{1}{2^k}, \sum_{k=1}^{10}\frac{1}{2^k}, \sum_{k=1}^{50}\frac{1}{2^k}$ を計算せよ (なるべくユーザー定義関数を使うこと)。また、それらの値を正確に小数に直せ (十進法では有限小数というのはすぐ分かりますね？)。

(3) 与えられた $\alpha>0$ に対して、$\sqrt{\alpha}$ の近似値を求めるために Newton 法

$$x_1 \text{ は適当に与える},$$
$$x_n = x_{n-1} - \frac{x_{n-1}^2-\alpha}{2x_{n-1}} = \frac{1}{2}\left(x_{n-1}+\frac{\alpha}{x_{n-1}}\right) \quad (n=2,3,\cdots)$$

が利用できる[6]。実際にこれを用いて $\sqrt{3}$, $\sqrt{21}$ の近似値を求めよ。やはり計算の仕方を工夫すること。また得られた結果の精度についても検討せよ。

(4) 次のどちらか一方を解け。

(a) 図 1 を再現せよ。(色々な描き方があります。楕円面と平面は別々に描いてから合成出来ることを知っておくと、自由度が上がるかも。)

(b) 円錐を描け。

(注意　3 次元グラフィックスは、EPS 形式で出力すると、ファイル・サイズが非常に大きくなり、TeX 文書に取り込めなかったり、Oh-o! Meiji にアップロード出来なくなったりするので、一度 JPEG 形式で出力してから、jpeg2ps で EPS 形式に変換することを勧めます。)

[6] Newton 法の一般式は $x_{n+1}=x_n-f'(x_n)^{-1}f(x_n)$ で、$f(x)=x^2-\alpha$ について適用すると上の式が得られる。

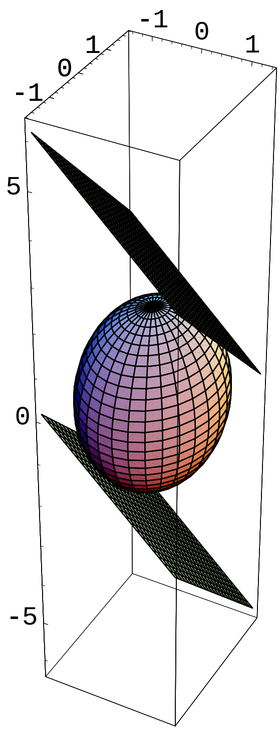

図 1: $\dfrac{x^2}{2}+\dfrac{y^2}{3}+\dfrac{z^2}{4}=1$ と接平面 $x+y+z=\pm 3$

図 1 の描き方のヒント: 球面を描く例は解説文書の中にある (そこではパラメーター曲面としてだったけれど、レベル・セット (等値面) としても描画可能)。それを少し修正すれば $x^2/2+y^2/3+z^2/4=1$ を描くのは簡単である。一方で平面を描くのも簡単 (グラフとして描いたり、やはりレベル・セット (等値面) としても描画可能)。同時に描ければ良いけれど、それは簡単ではないかもしれない。そういう困難を解決する手段が、別々に描いておいたものをまとめて表示する `Show[]`[7] です。

補足 (2012/6/27)

- グラフィックス・オブジェクトは、変数に代入しておくのが良い。

```
g1=Plot3D[...]
...
g=Show[g1,g2]
```

- `Export[ ]` コマンドで保存できる。2012 年度の情報処理教室の環境では、特に指定しないとドキュメント に保存される。フォーマットはファイル名末尾の拡張子で自動的に選ばれる。

```
Export["kadai9graph.eps", g]
```

ドキュメントの下の syori2 フォルダーに保存するのならば

[7] http://www.math.meiji.ac.jp/~mk/syori2-2012/mathematica/node61.html

```
Export["syori2/kadai9graph.eps", g]
```

のように指定すれば良い。

- グラフィックスを PostScript フォーマットで出力した場合、サイズが大きくなることがある (その結果 Oh-o! Meiji のレポート提出システムにはねられることがある)。その場合は (姑息な手段かもしれないが) JPEG で出力して、それから PostScript に変換するのが良い。

```
Export["syori2/kadai9graph.jpg", g]
```

コマンドプロンプトにて JPEG から PostScript に変換

```
jpeg2ps kadai9graph.jpg > kadai9graph.eps
```

(jpeg2ps コマンドは Windows 標準のコマンドではないが、2012 年度情報処理教室の環境には用意されている。)

- (おまけ) LaTeX に取り込むには、

```
\usepackage[dviout]{graphicx}% もしかすると dviout は dvipdfm が良いかも。
  ...
\begin{document}
  ...
\includegraphics[width=10cm]{kadai9graph.eps}
```

5 レポート課題 10(案)

次のいずれかを選択して下さい。

(1) 授業などで現れた問題や例を、Mathematica を使って計算してみる。教科書、授業のノート、プリント、自分が読んだ本 (授業と全然関係無くても良い) などから、自分でやるのは大変そうな計算や、グラフ描画など、適当な問題を探しておいて下さい。

(2) Mathemaitca が計算できない、あるいは間違えた結果を答えるような問題を見つけたら、その理由を分析して、どの辺に限界があるか確めてみる。

(3) 3 次元空間のラプラシアン $\triangle = \dfrac{\partial^2}{\partial x^2} + \dfrac{\partial^2}{\partial y^2} + \dfrac{\partial^2}{\partial z^2}$ の極座標表示を Mathematica を使って計算せよ (微分法の合成関数の微分法の少し面倒目の計算問題)。

A 課題 6B 解説

課題 6B[8]

「円周率計算の歴史」[9] で書いたように、円周率 π の効率的な計算法には、(i) $\arctan = \tan^{-1}$ のような逆三角関数の Taylor 展開を利用する方法, (ii) AGM 公式を利用する方法, (iii) ラマヌジャンの公式 (またはその系列) を利用する方法, などがある。

ここでは、主に単純な (i) の方法を解説する。

A.1 arctan の Taylor 展開を用いて計算する

既に言ってあるように、

http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-06/node7.html

で紹介した `piarctan.BAS` が叩き台になる。これは、与えられた x, N に対して、級数

$$\tan^{-1} x = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2n-1} x^{2n-1} \tag{1}$$

の第 N 部分和 S_N を計算するプログラムである。

piarctan.BAS (再掲)

```
REM piarctan.BAS --- マーダヴァ・グレゴリー・ライプニッツ級数でπを計算
REM arctan x の級数を第 n 項まで計算
INPUT X
INPUT N
F=-X*X
T=X
S=0
FOR J=1 TO N
  A=T/(2*J-1)
  S=S+A
  T=F*T
NEXT J
PRINT "arctan(x)≒";S
PRINT "その４倍";4*S
REM 組込み定数 PI との差を計算してみる
PRINT USING "πとの差=-%.###^^^^^^";4*S-PI
END
```

$x = 1$ の場合の

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots$$

[8] http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-06/node8.html

[9] http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-06/node9.html

は有名であるが、収束は非常に遅い。実際、実験で見た通り、

$$\pi - 4S_N \fallingdotseq \frac{1}{N}$$

が成り立つ (1 億 $(=10^8)$ 項足して、誤差 10^{-8} — 10 桁も合わない)。どんなにコンピューターが速くても、100 桁の精度の値を計算するのは無理である。

しかし、これは x の値として、収束ギリギリの $x=1$ を代入するからである (実際、(1) の収束半径は 1 である, 冪級数は収束円の内部では必ず収束するが円周上の点で収束するかどうかは case by case で収束しても「遅い」場合が多い)。$|x|<1$ なる x に対しては、(1) の収束はぐっと速くなる。

一番簡単なのは、高校生も知っている $\tan\dfrac{\pi}{6}=\dfrac{1}{\sqrt{3}}$ に基づく、$\pi = 6\tan^{-1}\dfrac{1}{\sqrt{3}}$ を用いることである。

`INPUT X` を `X=1/SQR(3)` に変え、4 をかけるところを 6 をかけるように変えたものが次のプログラムである (`OPTION ARITHMETIC DECIMAL_HIGH` にも変えた)。

kadai6b1.BAS

```
REM kadai6b1.BAS --- マーダヴァ・グレゴリー・ライプニッツ級数でπを計算
REM arctan(1/√3) の級数を第N項まで計算
OPTION ARITHMETIC decimal_high
INPUT N
LET X=1/SQR(3)
LET F=-X*X
LET T=X
LET S=0
FOR J=1 TO N
   LET A=T/(2*J-1)
   LET S=S+A
   LET T=F*T
NEXT J
PRINT "6*arctan(x)≒";6*S
REM 組込み定数 PI との差を計算してみる
PRINT USING "πとの差=-%.###^^^^^^":6*S-PI
END
```

最後に `PI` と比較しているのは、カンニングであるが (苦笑)、その結果は次のようになり、$N=210$ くらいで 100 桁精度を実現しているようである (差が大体 3.9×10^{-103} であるから)。

kadai6b1.TXT ($N=210$ の場合)

```
? 210
6*arctan(x)≒ 3.14159265358979323846264338327950288419716939937510582097494459230781 6
πとの差=-3.939E-0103
```

N が増加するにつれて、精度がどのように上がっていくか見るために、`INPUT N` をやめて、全体を

```
FOR N=10 TO 210 STEP 10
...
NEXT N
```

と FOR NEXT ループにして、次の表 2 の結果を得る。

N	誤差
10	-2.143×10^{-6}
20	-1.839×10^{-11}
30	-2.085×10^{-16}
40	-2.654×10^{-21}
50	-3.601×10^{-26}
60	-5.086×10^{-31}
70	-7.387×10^{-36}
80	-1.095×10^{-40}
90	-1.649×10^{-45}
100	-2.514×10^{-50}
110	-3.872×10^{-55}
120	-6.011×10^{-60}
130	-9.399×10^{-65}
140	-1.478×10^{-69}
150	-2.337×10^{-74}
160	-3.710×10^{-79}
170	-5.915×10^{-84}
180	-9.461×10^{-89}
190	-1.518×10^{-93}
200	-2.442×10^{-98}
210	-3.939×10^{-103}

図 2: 項数 N と誤差

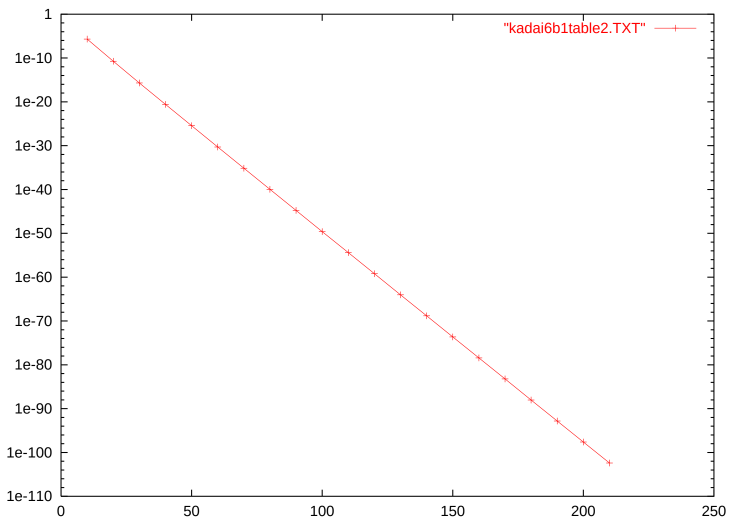

図 3: 項数 N と誤差 (縦軸対数目盛)

誤差が大体等比数列的に (N に関して指数関数的に？) 0 に収束している様子が分かる 本当はこういうのはグラフにするのが良い。ここでは gnuplot (1 年生のときに習ったはず) を使ってみたが、もちろん十進 BASIC で描くのも簡単である。

余談 A.1 (少し数学) この級数は交代級数なので、和と部分和の差は、次の項よりも小さくなることが分かる[10]。従って、誤差が等比数列的に小さくなることが分かる。もう少し考えると、上の級数は $N = 210$ 程度で、誤差が 10^{-100} 位になることも分かる (なぜ？)。 ■

[10] $a_1 > a > 2 > \cdots \geq 0$, $a_n \downarrow 0$ なる $\{a_n\}$ を用いて、$s = \sum_{n=1}^{\infty}(-1)^{n-1}a_n$ と表される級数を交代級数と言う。これは必ず収束し、$|s - s_n| \leq a_{n+1}$ という評価が簡単に得られる。

それでは、要求されていた結果を求めるプログラムを作ろう。課題2B でやったように、FOR NEXT で N を大きくして行って、部分和を計算するプログラムにする。

kadai6b2.BAS

```
REM kadai6b2.BAS --- マーダヴァ・グレゴリー・ライプニッツ級数でπを計算
REM arctan(1/√3) の級数を 10,20,...,250 項まで計算
REM 結果は小数点以下 110 位まで表示
OPTION ARITHMETIC decimal_high
LET FMT$="N=### -%."&REPEAT$("#",110)
LET N=250
LET X=1/SQR(3)
LET F=-X*X
LET T=X
LET S=0
FOR J=1 TO N
   LET A=T/(2*J-1)
   LET S=S+A
   LET T=F*T
   IF MOD(J , 10) =0 THEN
      PRINT USING fmt$:J,6*S
   END IF
NEXT J
REM 組込み定数 PI との差を計算してみる
PRINT USING "πとの差=-%.###^^^^^^":PI-6*S
END
```

結果は次のようになる (TEX で \scriptsize コマンドを使って縮小)。

```
N= 10  3.14159051093808009964275422994425504368823543729459863385301608264097243139275244243408049537664837144194195965
N= 20  3.14159265357140338177371056457791845749708370902558800062450336039110974863967354187227900363909324495379551167
N= 30  3.14159265358979302993126907675698512128890364163387594078160677223250316907504141910094384327780556430672531665
N= 40  3.14159265358979323845998904545815723164682333580898559851810755021711576515774234507828600074005410050567590231
N= 50  3.14159265358979323846264334727215223712766242383933328994947074253583407491260142245980404128750279799354137185
N= 60  3.14159265358979323846264338327899429478611788675967126248193958028428440424653148715465478961463146251880103922
N= 70  3.14159265358979323846264338327950287681011408881114209344980232821142119403271477392517560945005206755050370174
N= 80  3.14159265358979323846264338327950288419705988682105507083891588023987499387955321296887381276928532730590072263
N= 90  3.14159265358979323846264338327950288419716939772598750958858556364736336671762754275139106652717651497248200692
N=100  3.14159265358979323846264338327950288419716939937508067873446993355312916323675359893283010326258393751533470850
N=110  3.14159265358979323846264338327950288419716939937510582058777735023191320405347993715459391151728912750033891584
N=120  3.14159265358979323846264338327950288419716939937510582097493858083854337957828522717360102517299756190923350247
N=130  3.14159265358979323846264338327950288419716939937510582097494459221382757184428319165349115190289161510116917782
N=140  3.14159265358979323846264338327950288419716939937510582097494459230781492806566013451682112808110943615477604646
N=150  3.14159265358979323846264338327950288419716939937510582097494459230781640626284131806763014633707611386690905481
N=160  3.14159265358979323846264338327950288419716939937510582097494459230781640628620862758871391779973144580812870063
N=170  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862212031675396342638517754282
N=180  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803473073620684479093638320
N=190  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534059912106400090011
N=200  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211704355929502943
N=210  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798175415515
N=220  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808014
N=230  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651
N=240  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651
N=250  3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534211706798214808651
πとの差= 2.722E-0122
```

少々字が小さくて見づらいが、$N = 230, 240, 250$ の場合に、小数点以下 110 位までの小数表示が一致していることが分かる。これで小数点以下 100 位までの値は得られていると考えて良いだろう。

円周率 π の小数点以下 110 位まで

```
3.14159 26535 89793 23846 26433 83279 50288 41971 69399 37510
  58209 74944 59230 78164 06286 20899 86280 34825 34211 70679
  82148 08651
```

(要求されていたのは、100 位までだったが、101 位が '8' で、四捨五入を考えると微妙なので、110 位まで表示しておく。)

100 桁の数をそのまま表示されても読みづらいので、10 あるいは 5 桁ごとにブランクをいれるなどすると親切であろう。なお、筆者自身はそれを「手で」行ったが、過去に十進 BASIC にやらせた学生もいた (感心しました)。

(参考) pi100.BAS 5 桁ずつ区切って表示

```
OPTION ARITHMETIC decimal_high
LET FMT$="%."&REPEAT$("#",110)
LET PI100$=USING$(fmt$,PI)
PRINT "3."
FOR i=0 TO 20
   PRINT mid$(PI100$,i*5+3,5)&" ";
   IF MOD(i,10)=9 THEN
      PRINT
   END if
NEXT i
END
```

pi100.TXT

```
3.
14159 26535 89793 23846 26433 83279 50288 41971 69399 37510
58209 74944 59230 78164 06286 20899 86280 34825 34211 70679
82148
```

A.2 AGM, 代表選手 Gauss-Legendre 公式による計算

「AGM(算術幾何平均) を用いる方法」[11] で紹介した公式で計算してみよう。この公式はどうしてこれで円周率が計算できるかを理解するのが難しいが、アルゴリズムそのものは比較的単純で、短いプログラムで計算が実行可能である。

ここでは、鎌田伊織・吉本清夏「π — 計算法の変遷 —」, 2005 年度卒業研究[12] から拝借したプログラムを紹介する。

[11] http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-06/node18.html

[12] http://www.math.meiji.ac.jp/~mk/labo/report/open/2005-kamata-yoshimoto.pdf

GaussLegendre.BAS

```
REM GaussLegendre.BAS --- Gauss-Legendre公式によるπの計算
REM 鎌田伊織・吉本清夏,「π-- 計算法の変遷 ---」, 2005年度卒業研究
OPTION ARITHMETIC DECIMAL_high
OPTION BASE 0
DIM A(100),B(100),T(100)
LET A(0)=1
LET B(0)=1/SQR(2)
LET T(0)=1/4
LET MAXN=10
FOR n=1 TO maxn
   LET a(n)=(a(n-1)+b(n-1))/2
   LET b(n)=SQR(a(n-1)*b(n-1))
   LET t(n)=t(n-1)-2^(n-1)*(a(n)-a(n-1))^2
   PRINT USING "### -%.####^^^^^^":n,PI-(A(n)+B(n))^2/(4*t(n))
NEXT n
LET n=maxn
LET k=(a(n)+b(n))^2/(4*t(n))
PRINT k
END
```

GaussLegendre.TXT

```
  1  1.0134E-0003
  2  7.3763E-0009
  3  1.8313E-0019
  4  5.4721E-0041
  5  2.4061E-0084
  6  2.3086E-0171
  7  1.0586E-0345
  8  1.1110E-0694
  9 -1.5022E-1000
 10 -1.5022E-1000
 3.141592653589793238462643383279502884197169399375105820974944592307816406286208998628
```

わずか 10 回未満の反復で 1000 桁の精度を達成していることに注目。合っている桁数が反復1回ごとにほぼ2倍になるという、いわゆる2次の収束をしている (これは例えば、非線形方程式の解法として紹介した Newton 法の性質と同じ) ので、あっと言う間に 1000 桁を達成！、と言えるでしょうか。

($\tan^{-1} 1$ の級数展開と比べると、もの凄い違いですね。)

A.3 参考: 色々な arctan 公式を比較する

`http://www.math.meiji.ac.jp/~mk/syori2-2012/jouhousyori2-2012-06/node17.html` で紹介した公式を比較してみるのも面白いです。

こういうのは実際に試してみて初めて分かることが多く、挑戦できる腕力のある人は是非やってみることをお勧めします。

いくつかアドバイスをします。

(1) たくさんの arctan を計算するので、arctan の計算を副プログラム (関数またはサブルーチン) にすることをお勧めします。次のサンプル・プログラムは、`piarctan.bas` を外部副関数 (`EXTERNAL FUNCTION`) を使って計算するように書き換えたものです。

```
REM piarctan2.bas --- マーダヴァ・グレゴリー・ライプニッツ級数でπを計算
REM arctan の級数の計算は外部副関数に任せることにした。
DECLARE EXTERNAL FUNCTION arctan
INPUT x
INPUT n
LET  s=arctan(x,n)
PRINT "arctan(x)≒";s
print "その4倍";4*s
PRINT USING "πとの差=-%.###^^^^^^":4*s-PI
END

EXTERNAL FUNCTION arctan(x,n)
REM arctan x の級数を第 n 項まで計算
LET  f=-x*x
LET  t=x
LET  s=0
for j=1 to n
   LET  a=t/(2*j-1)
   LET  s=s+a
   LET  t=f*t
NEXT j
LET  arctan=s
END FUNCTION
```

(2) 級数の部分和 $s_n(x)$ は、n をどこまで大きく取れば十分な精度を持つのか？一般にはそれほど簡単ではありませんが、今考えている arctan の級数の場合は、各項の絶対値が単調に減少する交代級数になるので、次の Leibniz の定理によって簡単に見通しが立ちます (s_n の誤差は、次に足す項の絶対値 a_{n+1} 以下である)。

交代級数に関する Leibniz の定理

数列 $\{a_n\}_{n\geq 1}$ は単調減少 $(a_1 \geq a_2 \geq \cdots)$ で、$\displaystyle\lim_{n\to\infty} a_n = 0$ を満たすとき、級数

$$s = \sum_{n=1}^{\infty} (-1)^{n-1} a_n = a_1 - a_2 + a_3 - a_4 + \cdots$$

は収束する。第 n 部分和を s_n とすると、

$$s_{2n} \leq s \leq s_{2n-1} \quad (n \in \mathbf{N}),$$

$$|s - s_n| \leq a_{n+1} \quad (n \in \mathbf{N})$$

が成り立つ。

次に掲げる結果は、100 桁程度の精度を得るために、級数の和を何項加える必要があるか概算し、実際にそれを実行するとどれくらいので精度になるかを計算したものです。

大体どれくらいか…

```
シャープ
第 205 項まで加えて要求精度を達成しました。
 9.81E-101

マチン 2項公式
第 71 項まで加えて要求精度を達成しました。
第 22 項まで加えて要求精度を達成しました。
 1.20E-101

Gauss 3項公式
第 40 項まで加えて要求精度を達成しました。
第 29 項まで加えて要求精度を達成しました。
第 22 項まで加えて要求精度を達成しました。
 1.23E-102

Gauss 4項公式
第 32 項まで加えて要求精度を達成しました。
第 29 項まで加えて要求精度を達成しました。
第 22 項まで加えて要求精度を達成しました。
第 21 項まで加えて要求精度を達成しました。
 1.09E-103

Gauss 9項公式
第 14 項まで加えて要求精度を達成しました。
第 13 項まで加えて要求精度を達成しました。
第 13 項まで加えて要求精度を達成しました。
第 12 項まで加えて要求精度を達成しました。
第 12 項まで加えて要求精度を達成しました。
第 11 項まで加えて要求精度を達成しました。
第 11 項まで加えて要求精度を達成しました。
第 10 項まで加えて要求精度を達成しました。
第 10 項まで加えて要求精度を達成しました。
 7.45E-106

高野喜久雄の4項公式
第 30 項まで加えて要求精度を達成しました。
第 29 項まで加えて要求精度を達成しました。
第 22 項まで加えて要求精度を達成しました。
第 11 項まで加えて要求精度を達成しました。
 7.29E-105

Stormerの4項公式
第 29 項まで加えて要求精度を達成しました。
第 22 項まで加えて要求精度を達成しました。
第 18 項まで加えて要求精度を達成しました。
第 13 項まで加えて要求精度を達成しました。
 7.73E-104
```

この結果をどう分析すべきか…書きたくてむずむずするけれど、こういうのは自分で考えて納得すべきものだと信じるので書きません (ネタばらしは控えるように)。

(一言だけ： Gauss は色々大理論を作った偉い人だけど、こういう計算も大好きだったのが良く分かります。)