

# レポート課題に取り組む& 解説

かつらだ まさし  
桂田 祐史

2013年7月10日

この授業用の WWW ページは <http://www.math.meiji.ac.jp/~mk/syori2-2013/>

## 1 連絡事項

- 「レポート課題 11」<sup>1</sup> の締切ですが、7月19日(金曜) 18:00 まで延長します(考えてみたら、これは無理して説明しなくても良いわけですから、授業終了後で構わないわけでした。面白いものがあつたら、後から WWW に何か載せるかもしれませんが。)
- 「レポート課題 10」<sup>2</sup> の (4-a) で楕円体の方程式を

$$\frac{(x+1)^2}{1} + \frac{y^2}{4} + \frac{(z-1)^2}{9} = 1$$

に訂正しました(接平面の方は  $x+y+z = \pm\sqrt{14}$  のままです)。これで接するようになるはずです。

- 『Mathematica 入門』<sup>3</sup> ですが、「微分方程式を解く DSolve[], NDSolve[]」<sup>4</sup> を少し書き足しました。「レポート課題 11」<sup>5</sup> で常微分方程式関係の問題に取り組みやすくなったと思います(ちょっと遅かったかな?)。
- 課題 10 (4-a) で楕円体とその接平面を描く場合の参考に
  - グラフィックスを描く場合に、個別に描いたものを合成することができます。課題 10 (4-a) は使わないでも解けますが、少し説明します。  
「グラフィックス・オブジェクト, Show[] で再表示・合成」<sup>6</sup>

<sup>1</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-11/node3.html>

<sup>2</sup><http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2013-10/node3.html>

<sup>3</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/mathematica/>

<sup>4</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/mathematica/node37.html>

<sup>5</sup><http://www.math.meiji.ac.jp/~mk/syori2/jouhousyori2-2013-11/node3.html>

<sup>6</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/mathematica/node63.html>

- Mathematica では、座標軸の縮尺は適当に決めてくれますが、そうやってお任せにしていると、ときどき結果がこちらの望んでいるようにはならないことがあります。2次元グラフィックスでは、AspectRatio の値をこちらが指示します。例えば Plot[] では、通常 1/GoldenRatio ですが、実際の数値を使うには、AspectRatio->Automatic とします。次の例は  $f(x) = \sqrt{1-x^2}$  の  $0 \leq x \leq 1$  の範囲のグラフ (4分円のはず) を描いてみたものです。AspectRatio->Automatic と指定しないと、円のように見えません。

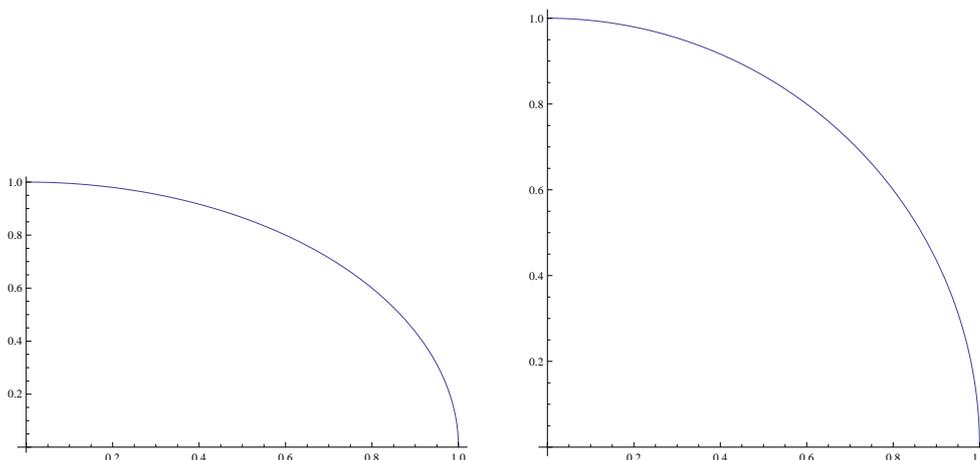


図 1: デフォルトは 1/黄金比      図 2: Automatic で縦横の縮尺が揃う

3次元グラフィックスでは、同様の指示は BoxRatios で行ないます。各軸の縮尺を揃えるには BoxRatios->Automatic とします。

- 他に PlotRange などのオプションも指定した方が良くもしいない (PlotRange->All 等)。

## A レポート課題の解説

### A.1 レポート課題 6B 解説

レポート課題 6B<sup>7</sup> について簡単に解説します。

既に説明したように、

- (1) 10 個の頂点の座標を計算で求めてしまう、
  - (2) タートルグラフィックスを使う、
  - (3) 正五角形の頂点を一つおきに結んで星形を描き PAINT で許してもらう
- などの方法があります。

<sup>7</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-07/node3.html>

### A.1.1 頂点の座標を計算で求める方法

**まずは正五角形から** まず、正五角形を描くことを考えましょう。与えられた自然数  $n$  に対して、円周を  $n$  等分する点の座標を表す式は、あちこちで習っていますね。一周  $2\pi$  を  $n$  等分した角は  $\frac{2\pi}{n}$  ですから、

$$\theta_k := \frac{2\pi k}{n} \quad (k = 0, 1, \dots, n-1)$$

として、

$$x_k := r \cos \theta_k, \quad y_k := r \sin \theta_k \quad (k = 0, 1, \dots, n-1)$$

とおくと、 $(x_k, y_k)$  ( $k = 0, 1, \dots, n-1$ ) は、原点を中心とする半径  $r$  の円周の  $n$  等分点を与えます。

簡単のため  $r = 1$  として、また角度が分かりやすいように (?) 単位を度で表すことにして、点の番号を 0 からでなく 1 から振ることにして、また頂点をずらす角度  $\phi = 90^\circ$  を導入して (真上に頂点が来ると星がきれい)

$$\begin{aligned} \theta_j &:= \frac{360^\circ}{n}(j-1) \quad (j = 1, 2, \dots, n), \\ x_j &:= \cos(\theta_j + \phi), \quad y_j := \sin(\theta_j + \phi) \quad (j = 1, 2, \dots, n) \end{aligned}$$

とおきます (ここらへんは好みの問題で、どうでも良いです)。

次のプログラムで正五角形の輪郭が描けます。

```
REM 正五角形を描く
OPTION ANGLE DEGREES
LET n=5
LET DT=360/n
LET p=90
REM 頂点の座標を求める
DIM x(n),y(n)
FOR j=1 TO n
  t=(j-1)*DT+p
  PRINT t
  LET x(j)=COS(t)
  LET y(j)=SIN(t)
NEXT j
REM 正五角形を描く
SET WINDOW -1,1,-1,1
FOR j=1 TO n
  PLOT LINES : x(j),y(j);
NEXT j
PLOT LINES : x(1),y(1)
END
```

正五角形の内部を塗るのは簡単で、最後 (END 行の前) に次の 2 行を加えるだけです。

```
SET AREA COLOR "red"  
MAT PLOT AREA : x,y
```

pentagon.BAS

```
REM pentagon.BAS --- 正五角形を描く  
OPTION ANGLE DEGREES  
LET n=5  
LET DT=360/n  
LET p=90  
REM 頂点の座標を求める  
DIM x(n),y(n)  
FOR j=1 TO n  
  LET t=(j-1)*DT+p  
  PRINT t  
  LET x(j)=COS(t)  
  LET y(j)=SIN(t)  
NEXT j  
REM 頂点を順に結んで正五角形を描く  
SET WINDOW -1,1,-1,1  
FOR j=1 TO n  
  PLOT LINES : x(j),y(j);  
NEXT j  
PLOT LINES : x(1),y(1)  
REM 塗る  
SET AREA COLOR "red"  
MAT PLOT AREA: x,y  
END
```

**素朴に星形** 正五角形の頂点を一つおきにたどると星になります。素朴にやると

```
PLOT LINES : x(1),y(1);x(3),y(3),x(5),y(5);x(2),y(2);x(4),y(4);x(1),y(1)
```

となるでしょうか (これを FOR NEXT を使って書くことも出来ますが、自主的な課題としましょう)。

```
kadai6b0.BAS
```

```
REM kadai6b0.BAS --- 線画で星
```

```
OPTION ANGLE DEGREES
```

```
LET n=5
```

```
LET DT=360/n
```

```
LET p=90
```

```
REM 正五角形の頂点の座標を求める
```

```
DIM x(n),y(n)
```

```
FOR j=1 TO n
```

```
    LET t=(j-1)*DT+p
```

```
    PRINT t
```

```
    LET x(j)=COS(t)
```

```
    LET y(j)=SIN(t)
```

```
NEXT j
```

```
REM 星を描く (線画)
```

```
SET WINDOW -1,1,-1,1
```

```
PLOT LINES : x(1),y(1);x(3),y(3);x(5),y(5);x(2),y(2);x(4),y(4);x(1),y(1)
```

```
END
```

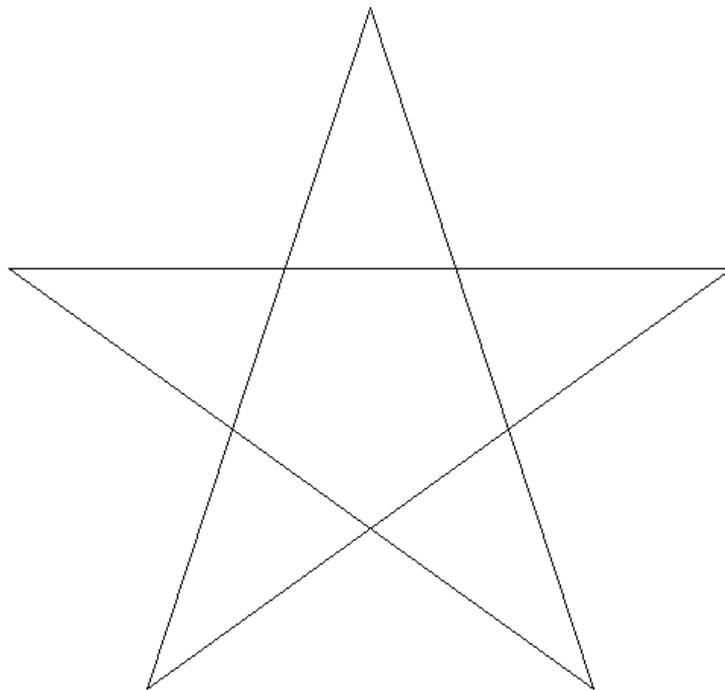


図 3: 線画で星

次のプログラムは、なるべく避けようと言った PAINT を使って正五角形を描くプログラムです。

```
kadai6b1.BAS
```

```
REM kadai6b1.BAS --- 星形を描く
OPTION ANGLE DEGREES
LET n=5
LET DT=360/n
LET p=90
REM 正五角形の頂点の座標を求める
DIM x(n),y(n)
FOR j=1 TO n
  LET t=(j-1)*DT+p
  PRINT t
  LET x(j)=COS(t)
  LET y(j)=SIN(t)
NEXT j
SET WINDOW -1,1,-1,1
DRAW grid
SET AREA COLOR "red"
PLOT LINES : x(1),y(1);x(3),y(3);x(5),y(5);x(2),y(2);x(4),y(4);x(1),y(1)

REM 以下どろくさく塗る (PAINT6 連発)
PAINT 0,0
FOR t=1 TO 5
  paint 0.9*x(t),0.9*y(t)
NEXT t
END
```

なお、PLOT LINES を強引に PLOT AREA に置き換えると、図5のように真ん中が白く抜けた失敗図が出来ます。

なお、PLOT LINES の前に SET LINE COLOR "red" とすると、星形の中がきれいに塗れます (輪郭線は消えますが)。

**頑張って星形の10個の頂点を計算する** 紙の上に星形を(ある程度正確に)描いて、頂点の座標を求めましょう。頂点は「外側の円周上にあるもの」と「内側の円周上にあるもの」の二種類あり、それぞれ正五角形の頂点をなすことは容易に分かります。

半径1の円に内接する星形とすると、外側の円周上にある頂点  $(x_j, y_j)$  ( $j = 1, 2, 3, 4, 5$ ) は、上で提示した式で計算出来ることになります。

内側の円周上にある頂点  $(\bar{x}_j, \bar{y}_j)$  ( $j = 1, 2, 3, 4, 5$ ) は(適当に番号を振ります)、次の性質を持つことは明らかです。

(1) ある正数  $r$  ( $0 < r < 1$ ) を半径とする原点中心の円周上にある。

(2)  $(\bar{x}_j, \bar{y}_j)$  の角度は、 $(x_j, y_j)$  の角度と  $(x_{j+1}, y_{j+1})$  の角度のちょうど真ん中(ただし  $(x_6, y_6) = (x_1, y_1)$  とする)。

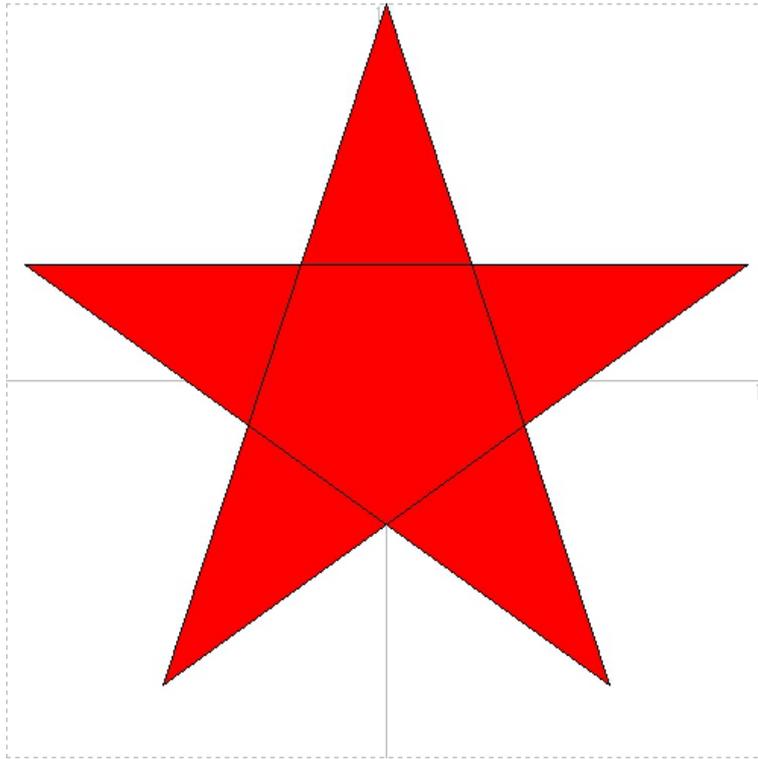


図 4: PAINT 六連発で描きました。

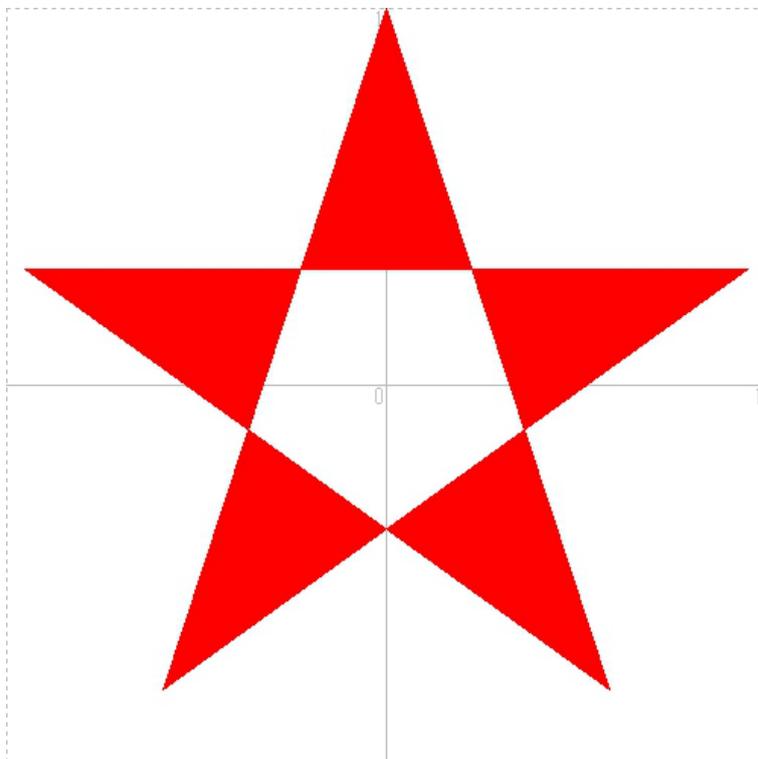


図 5: PLOT AREA:  $x(1),y(1);x(3),y(3);...$  とすると

$r$  さえ求まれば、後は正五角形を描くのとほとんど同様の計算です。とりあえず  $r$  はユーザーに入力してもらうことにすると、次のようなプログラムが書けます。

```
kadai6b4.BAS
REM kadai6b4.BAS --- 星を描く (後一步バージョン)
OPTION ANGLE DEGREES
SET WINDOW -1,1,-1,1
DIM X(10),Y(10)
INPUT PROMPT "内側の円の半径 r": r
LET DT=360/10
FOR j=1 TO 10
  LET T=(j-1)*DT+90
  REM j が奇数か偶数かで場合分け
  IF MOD(j,2)=1 THEN
    LET x(j)=COS(t)
    LET y(j)=SIN(t)
  ELSE
    LET x(j)=r*COS(t)
    LET y(j)=r*SIN(t)
  END IF
NEXT j
SET AREA COLOR "red"
MAT PLOT AREA : x,y
END
```

$r$  ( $< 1$ ) の値を色々変えて試してみましょう。 $r = 0.4$  くらいが良さそうです。

正確な星形を与える  $r$  は、図形の相似比を考えたりして求められます。詳細は自分で解きたい人のお楽しみで取っておきますが(あしからず)、結果は

$$r = \frac{\sin 18^\circ}{\cos 36^\circ} = \frac{\cos 72^\circ}{\cos 36^\circ} = \frac{(\sqrt{5}-1)/4}{(\sqrt{5}+1)/4} = \frac{\sqrt{5}-1}{\sqrt{5}+1} = \frac{3-\sqrt{5}}{2} = 0.38196601\dots$$

です。

kadai6b4.BAS の

```
INPUT PROMPT "内側の円の半径 r": r
```

を

```
LET r=(SQR(5)-1)/(SQR(5)+1)
```

に変えると、図 6 のようにきれいな星が描けます。



図 6: 完成版

### A.1.2 タートルグラフィックスで描く方法

星形の輪郭を辿ってみましょう。いつも進む距離は同じで、曲がり方が、右に  $144^\circ$ 、左に  $72^\circ$ 、... と交互になっている、と分かります。すると、

タートルグラフィックスで星を描く

```
FOR i=1 TO 5
  CALL walk(L)
  CALL right(144)
  CALL walk(L)
  CALL left(72)
NEXT i
```

とすれば星形の輪郭が描けるはずですが。

TURTLESTAR1.BAS

```
REM TURTLESTAR1.BAS --- タートルグラフィックスで星の輪郭を描く
OPTION ANGLE DEGREES
REM right(),left(),walk(),jump()
REM 初期化
SUB init
  LET direction=0
  LET xp=0
```

```

    LET yp=0
END SUB
REM 右に曲がる
SUB right(t)
    LET direction=direction-t
END SUB
REM 左に曲がる
SUB left(t)
    LET direction=direction+t
END SUB
REM s 歩ジャンプする
SUB jump(s)
    LET xp=xp+s*COS(direction)
    LET yp=yp+s*SIN(direction)
    PLOT LINES: xp,yp
END SUB
REM s 歩進む
SUB walk(s)
    PLOT LINES: xp,yp;
    CALL jump(s)
END SUB
REM ----- start -----
LET L=100
SET WINDOW -2*L,2*L,-2*L,2*L
CALL init
FOR i=1 TO 5
    CALL walk(L)
    CALL right(144)
    CALL walk(L)
    CALL left(72)
NEXT i
END

```

TURTLESTAR3.BAS 上のプログラムに PAINT を一つ加えると塗り潰しできますが、MAT PLOT AREA を使って塗るにはどうしたら良いか、少し工夫をしてみましょう。

亀に記憶力を持たせて、星形の道を辿ってから、通過地点の座標を答えさせるという手もあります。しかし高機能の亀はらしくないので、今どこにいるかを尋ねる imadoko(x,y) というサブルーチンを作って、それを使って描くようにしたのが次のプログラムです。

```

REM TURTLESTAR3.BAS --- タートルグラフィックスで星を描く
OPTION ANGLE DEGREES
REM right(),left(),walk(),jump()
REM 初期化

```

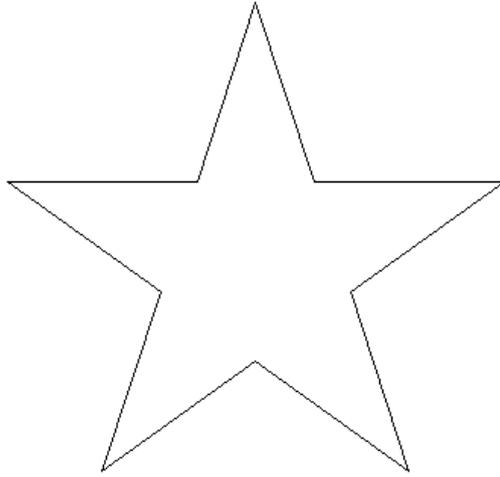


図 7: タートルグラフィックスで星を描く (輪郭線バージョン)

```
SUB init
  LET direction=0
  LET xp=0
  LET yp=0
END SUB
REM 右に曲がる
SUB right(t)
  LET direction=direction-t
END SUB
REM 左に曲がる
SUB left(t)
  LET direction=direction+t
END SUB
REM s 歩ジャンプする
SUB jump(s)
  LET xp=xp+s*COS(direction)
  LET yp=yp+s*SIN(direction)
  PLOT LINES: xp,yp
END SUB
REM s 歩進む
SUB walk(s)
  PLOT LINES: xp,yp;
```

```

CALL jump(s)
END SUB
REM 現在位置を答える
SUB imadoko(x,y)
  LET x=xp
  LET y=yp
END SUB
REM ----- start -----
LET L=100
SET WINDOW -2*L,2*L,-2*L,2*L
CALL init
DIM x(10),y(10)
LET n=1
FOR i=1 TO 5
  CALL walk(L)
  CALL imadoko(x(n),y(n))
  LET n=n+1
  CALL right(144)
  CALL walk(L)
  CALL imadoko(x(n),y(n))
  LET n=n+1
  CALL left(72)
NEXT i
SET AREA COLOR "red"
MAT PLOT AREA : x,y
END

```

## A.2 レポート課題8B

### 課題8B<sup>8</sup>

「円周率計算の歴史」<sup>9</sup> で書いたように、円周率  $\pi$  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.2.1 $\arctan$ の Taylor 展開を用いて計算する

既に言っているように、

<http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-08/node7.html>

<sup>8</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-08/node8.html>

<sup>9</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-08/node9.html>



図 8: タートルグラフィックスで星を描く

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

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

にやや強引に<sup>10</sup>  $x = 1$  を代入した

$$(2) \quad \frac{\pi}{4} = \tan^{-1} 1 = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{2n-1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

の右辺の級数の第  $N$  部分和  $S_N$  を計算するプログラムである。

---

<sup>10</sup>収束円上の点での収束は一般には保証されない。今の場合交代級数になるので収束する(ただし条件収束である)。冪級数の和に関する Abel の定理により、その和が  $\tan^{-1} 1$  になることが分かる。

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 1 の級数を第 n 項まで計算
REM 次の行は最初 INPUT X としてあったのを修正しました
X=1
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
```

この (2) は有名であるが、収束は非常に遅い。実際、実験で容易に確認できるように

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

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の計算が速くても、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 (繰り返しになるが、(1) の収束半径は 1 であり、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、高校生でも分かる  $\tan \frac{\pi}{6} = \frac{1}{\sqrt{3}}$  に基づく、 $\pi = 6 \tan^{-1} \frac{1}{\sqrt{3}}$  を用いることである。

$X=1$  を  $X=1/\text{SQR}(3)$  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)。

kadai8b1.BAS

```
REM kadai8b1.BAS --- マーダヴァ・グレゴリー・ライプニッツ級数で $\pi$ を計算
REM arctan(1/ $\sqrt{3}$ ) 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)  $\doteq$ ";6*S
REM 組込み定数 PI との差を計算してみる
PRINT USING "  $\pi$  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  $3.9 \times 10^{-103}$  であるから)。

kadai8b1.TXT ( $N = 210$  の場合)

```
? 210
6*arctan(x)  $\doteq$  3.141592653589793238462643383279502884197169399375105820974944592307816
 $\pi$  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 9 の結果を得る。

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事が分かる<sup>11</sup>。従って、誤差が等比数列的に小さくなる事が分かる。もう少し考える

<sup>11</sup> $a_1 > a > 2 > \dots \geq 0, a_n \downarrow 0$  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。

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

図 9: 項数  $N$  と誤差

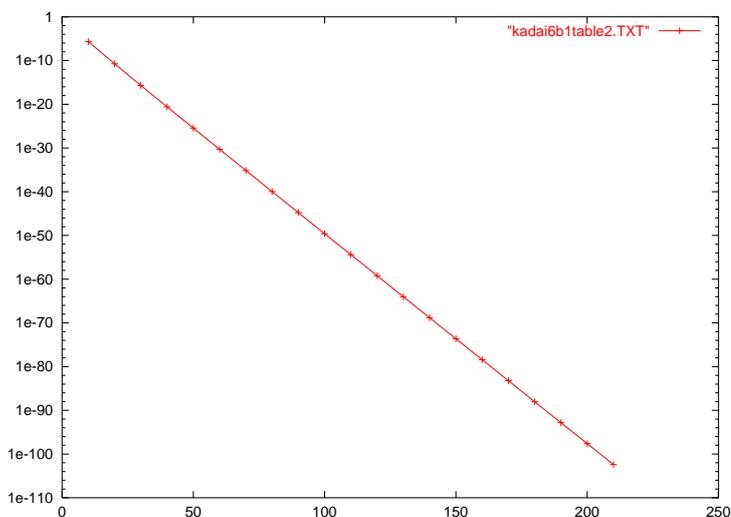


図 10: 項数  $N$  と誤差 (縦軸対数目盛)

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?)。 ■

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。

kadai8b2.BAS

```
REM kadai6b2.BAS --- マーダヴァ・グレゴリー・ライプニッツ級数で $\pi$ を計算
REM arctan(1/ $\sqrt{3}$ ) 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 "  $\pi$  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  3.1415926535714033817737105645779184574970837090255880062450336039110974863967354187227900363909324495379551167
N= 30  3.14159265358979302993126907675698512128890364163387594078160677223250316907504141910094384327780556430672531665
N= 40  3.1415926535897932384599890454581572316468233358089855985181075502171157651577423450782860007400541005067590231
N= 50  3.14159265358979323846264334727215223712766242383933328994947074253583407491260142245980404128750279799354137185
N= 60  3.14159265358979323846264338327899429478611788675967126248193958028428440424653148715465478961463146251880103922
N= 70  3.1415926535897932384626433832795028841971693993751058205877735023191320405347993715459391151728912750033891584
N= 80  3.14159265358979323846264338327950288419705988682105507083891588023987499387955321296887381276928532730590072263
N= 90  3.14159265358979323846264338327950288419716939772598750958858556364736336671762754275139106652717651497248200692
N=100  3.14159265358979323846264338327950288419716939937508067873446993355312916323675359893283010326258393751533470850
N=110  3.14159265358979323846264338327950288419716939937510582097494459230781640628620862758871391779973144580812870063
N=120  3.14159265358979323846264338327950288419716939937510582097493858083854337957828522717360102517299756190923350247
N=130  3.14159265358979323846264338327950288419716939937510582097494459221382757184428319165349115190289161510116917782
N=140  3.14159265358979323846264338327950288419716939937510582097494459230781492806566013451682112808110943615477604646
N=150  3.14159265358979323846264338327950288419716939937510582097494459230781640626284131806763014633707611386690905481
N=160  3.1415926535897932384626433832795028841971693993751058209749445923078164062862086210364283482534211704355929502943
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
```

$\pi$ 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  $\pi$  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 桁ごとに空白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.2.2 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」<sup>12</sup>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。

<sup>12</sup><http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-08/node18.html>

ここでは、鎌田伊織・吉本清夏「 $\pi$  — 計算法の変遷 —」, 2005 年度卒業研究<sup>13</sup> から拝借したプログラムを紹介する。

GaussLegendre.BAS

```
REM GaussLegendre.BAS --- Gauss-Legendre 公式による  $\pi$  の計算
REM 鎌田伊織・吉本清夏, 「 $\pi$  -- 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
3.1415926535897932384626433832795028841971693993751058209749445923078164062862089986
```

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、あっという間に 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。)

<sup>13</sup><http://www.math.meiji.ac.jp/~mk/labo/report/open/2005-kamata-yoshimoto.pdf>

### A.2.3 参考: 色々な arctan 公式を比較する

<http://www.math.meiji.ac.jp/~mk/syori2-2013/jouhousyori2-2013-06/node17.html>で紹介した公式を比較してみるのも面白いです (少なくとも私には)。

こういうのは実際に試してみても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
X=1
INPUT N
LET s=arctan(x,n)
INPUT 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 ( $a_1 \geq a_2 \geq \dots$ ) で、 $\lim_{n \rightarrow \infty} a_n = 0$  を満たすとき、級数

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

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も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。)