

## 1.2 フィボナッチ数列の 2 項間の比

フィボナッチ数列には面白い性質がある。その一つは、隣り合う 2 項の比を調べることで見えてくる。1, 1 から始まるフィボナッチ数列を 2 項ずつペアにし、『 $\frac{1}{1}$ ,  $\frac{2}{1}$ ,  $\frac{3}{2}$ ,  $\frac{5}{3}$ ,  $\frac{8}{5}$ ,  $\frac{13}{8}$ ,  $\frac{21}{13}$ ,  $\frac{34}{21}$ ,  $\frac{55}{34}$ ,  $\frac{89}{55}$ , ...』のように分数を作る。知りたいのは比の値であるが、その前にこれは『

```
\begin{luacode*}
  a = 0  b = 1
  for i = 1, 10 do
    a, b = b, a + b
    tex.print('\$\\displaystyle\\frac{' .. b .. '}' .. a .. '}$, ')
  end
  tex.print('\$\\dots$')
\end{luacode*}
```

』のようにして出力された。

ここでは  $\frac{b}{a}$  のように出力したかったので、 $\text{T}_{\text{E}}\text{X}$  の命令では `\displaystyle\frac{b}{a}` と書けばよい。しかし、 $a$ ,  $b$  は変数なので直接値を与えるために、`tex.print` 文は少々細切れの見づらいものになってしまった。ご愛嬌である。

で、隣り合う 2 項の比を眺めてもあまり意味はない。比の値に興味があるのだ。フィボナッチ数列の隣り合う 2 項の比を出力するスクリプト『

```
\begin{luacode*}
  a = 0  b = 1
  for i = 1, 20 do
    a, b = b, a + b
    tex.print(b / a .. ', ')
  end
  tex.print('\$\\dots$')
\end{luacode*}
```

』で、比の値を出力してみよう。すると『1.0, 2.0, 1.5, 1.6666666666667, 1.6, 1.625, 1.6153846153846, 1.6190476190476, 1.6176470588235, 1.6181818181818, 1.6179775280899, 1.6180555555556, 1.618025751073, 1.6180371352785, 1.6180327868852, 1.6180344478217, 1.6180338134001, 1.6180340557276, 1.6180339631667, 1.6180339985218, ...』となる。わちゃちゃ！ これはまずい。小数の桁数が多すぎて組版がうまくいってない。この程度のはみ出しが許せるならよいが（そんな人が  $\text{T}_{\text{E}}\text{X}$  を使うだろうか?）、やっぱり  $\text{T}_{\text{E}}\text{X}$  で組版するならこれはいただけない。表示方法を工

夫しくちや。

ところで、フィボナッチ数列というのは、何も 1, 1 から始まると決められているわけではない。たとえば 10, 3 から始めてもよい。すると『10, 3, 13, 16, 29, 45, 74, 119, 193, 312, 505, ...』のようになって、1, 1 から始まる数列とは違うものになる。これは以前のスクリプトと同じなのだが、最初に a の値を表示しなかったので『

```
\begin{luacode*}
  a = 10  b = 3
  tex.print(a .. ', ')
  for i = 1, 10 do
    tex.print(b .. ', ')
    a, b = b, a + b
  end
  tex.print('\dots$')
\end{luacode*}
```

』のように tex.print(a .. ', ') を加えることにした。

しかし、数列は別物に見えても 2 項の比についてはそうでもない。この場合は『

0.3,                    4.33333333333333,   1.2307692307692,   1.8125,                    1.551724137931,  
 1.64444444444444,   1.6081081081081,   1.6218487394958,   1.6165803108808,   1.6185897435897,  
 1.6178217821782,   1.6181150550796,   1.6180030257186,   1.6180458158018,   1.6180294712511,  
 1.6180357142857,   1.6180333296546,   1.618034240502,   1.6180338925892,   1.61803402548,  
 ...

』となる。少しはマシな出力になっただろうか。スクリプトは『

```


$$\begin{array}{l}
\directlua{
  a = 10  b = 3
  for i = 1, 4 do
    for j = 1, 5 do
      tex.print(b / a .. ', ' .. \asluastring{&})
      a, b = b, a + b
    end
    tex.print(\asluastring{\[-5pt]})
  end
  tex.print(\asluastring{\dots \})
}
\end{array}$$


```

』としてみた。

tmt's math page!

さて、いろいろな初期値を試してみれば、いつでも 1.6... 程度の値に収まるように思える。1 にまつわる路を翔んでいるので、目にする数値は当然 1 から始まる。この数値が何かは、もうしばらく翔び回ったときにはっきりするだろう。