5. . . の旅

5.1　黄金比

さあ、これから 5. . . の旅が始まる。と言っても 5 から旅が始まるのではなく、旅の途中に 5 と出会うのが今までとは違うところだが。

まず、フィボナッチ数列の 2 項の比

$$\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \frac{34}{21}, \cdots$$

に再び登場してもらおう。

ずっと前に、`[fratio.cpp]` で計算したから分かっているだろうが、比の値は $1.618033\cdots$ となる。ところでフィボナッチ数列の 2 項の比は、(後の項)/(前の項) で求めたものだ。しかし、比なんだから別に (前の項)/(後の項) で求めたっていいだろうに。

EX. フィボナッチ数列の比を (前の項)/(後の項) で計算してみよ。電卓でも計算できるが、**1.3 節**で書いた `[fratio.cpp]` の 8:行目を変更すればすぐだ。

さあ、比の値はどうなった？　ちょっと面白い結果になったのではないかい？　比の値は $0.618033\cdots$ である。きっちり 1 だけ違っているだろう。誤差はないのかって？　そう、まったく誤差はないのだ。この面白い性質を持つ比の値は**黄金比**と呼ばれている。一般には、$1.618033\cdots$ の方を黄金比と呼ぶことが多い。

黄金比とは一体どういう数なのだろうか。計算は簡単にできるので確かめてみよう。

まず、$1.618033\cdots$ に収束する真の値を x としよう。後から求めた比 $0.618033\cdots$ は、x の逆比 $1/x$ のことである。それが x より 1 小さい値に等しいのだから

$$\frac{1}{x} = x - 1 \tag{5.1}$$

が成り立つ。(5.1) は簡単な 2 次方程式となるが、フィボナッチ数の比は正の値なので、$x > 0$ の解を求めると

$$x = \frac{1+\sqrt{5}}{2}$$

であることがわかる。黄金比は**無理数** $\sqrt{5}$ を含む数なのだ。ここに 5 が登場してきた。解は正の数だけに限っているものの、方程式からひとつの解しか得られないということは、(5.1) の性質を満たす数は黄金比だけということである。

コンピュータにシミュレートさせてみよう。ここでは (5.1) を移項して

$$x = \frac{1}{x} + 1 \tag{5.2}$$

で x を探ることにする。そのほうが x の値を直接計算できるからだ。ところで、探るという言葉が気になるね。なぜそのような言葉遣いをしたかはプログラムを見てもらいたい。

programming list [grsimula.cpp]

```
 1: #include <iostream>
 2:
 3: int main() {
 4:     int i;
 5:     float x;
 6:     std::cout << "input an initial value : "; std::cin >> x;
 7:     for(i = 1; i < 20; i++) {
 8:         x = 1 / x + 1;
 9:         std::cout << x << std::endl;
10:     }
11:
12:     return 0;
13: }
```

プログラムは実に簡単だ。4:行目でカウンタ変数の `i` を、5:行目で黄金比になるであろう `x` を用意した。6:行目はいつもと変わらぬ入力要請だ。

7:行目の `for` 文は 20 回ほどの繰り返しだが、このぐらいで十分収束してくれる。

8:行目で $1/x + 1$ の計算をし、次に再利用できるよう x へ代入している。

その経過を表示しているのが 9:行目だ。`for` 文によって 19 回先までの x の値が表示される。

TRY! 入力要求に対して、色々な浮動小数点数を入力してみよ。初期値が何であれ、黄金比に収束する様子が分かるだろう。

あれ？　コンピュータが 2 次方程式を解くんじゃないんだ、と感じただろう。そう、コンピュータは計算をする機械であって、問題を解く機械ではない。問題を解く手順は人間が与えるのだ。では、この手順は何をしているのだろう。どう見ても (5.2) と同等な—つまり両辺に x を掛けた—方程式

$$x^2 = 1 + x$$

を解いているのではないね。でも、解である黄金比が求められている。

それなら $x^2 = 1 + x$ を移項した式、$x = x^2 - 1$ を使っても同じことだろう。そこでプログラムの 8:行目は `x = x * x - 1;` に変えてみる。さあ、準備は整った。プログラムを走らせよう。

TRY! 入力要求に対して、色々な浮動小数点数を入力してみよ。初期値が何であれ黄金比に収束する、とは言えないはずだ。

どうして？　同じ関係の方程式を使ったはずなのに。その秘密はグラフを描けば見えてくる。そして、プログラムがしていることを追ってみれば分かるのだけれど、今はそこに足を踏み入れないでおこう。