

9.3 マンデルブロ集合をもう少し

インターネットには多くのマンデルブロ集合の描画がある。大抵は色分けされていて、見た目も綺麗である。前節のプログラムに色をつけるわけにはいかないものの、色違いが分かるようにすることはできる。

programming list [mandel2.cpp]

```
1: #include <iostream>
2: #include <complex>
3:
4: int recur(std::complex<double> c) {
5:     std::complex<double> z(0, 0);
6:
7:     for(int i = 1; i < 21; i++) {
8:         z = z * z + c;
9:         if(abs(z) > 2) return i; // divergence.
10:    }
11:    return 0; // convergence.
12: }
13:
14: int main() {
15:     for(int y = 0; y < 40; y++) {
16:         double im = (20 - y) / 16.;
17:
18:         for(int x = 0; x < 40; x++) {
19:             double re = (x - 32) / 16.;
20:
21:             std::complex<double> z(re, im);
22:             int color = recur(z);
23:             if(color > 0) {
24:                 std::cout << " " << color % 8 + 1;
25:             } else {
26:                 std::cout << " ";
27:             }
28:         }
29:         std::cout << std::endl;
30:     }
31:     std::cout << std::endl;
32:
33:     return 0;
34: }
```

マンデルブロ集合がなぜ色分けできるのかということ、発散までにかかった回数で色を分けているからだ。そこで `recur()` 関数は、単に発散したときに 1 を返すのではなく、そのときの `i` の値を返すようにした。そうすれば `i` の値によって色を変えることができる。ただし当初の `recur()` 関数は、20 回のカウンタのために `i` を 0 から数え始めたが、それでは 0 が返る可能性があるので具

合が悪い。そこでカウントは 1 からに変更した。

`recur()` が返す値は 22:行目で使われる。実際には色の代わりに数値で画面表示するため、24:行目のように剰余で計算した。実行すると、それとなく色を感じられるのではないだろうか。

でも、このプログラム、というよりマンデルブロ集合の真骨頂を見るにはちょっとの修正がいる。現状、

Terminal の行範囲	変換式	im 座標の範囲
$y : 0 - 40$	$(20 - y)/20$	$-1.25 \leq \text{im} \leq 1.25$
Terminal のカラム範囲	変換式	re 座標の範囲
$x : 0 - 40$	$(x - 32)/16$	$-2 \leq \text{re} \leq 0.5$

のようにして範囲変換をしているので、たとえば $0.75 \leq \text{im} \leq 1$ 、 $-0.125 \leq \text{re} \leq 0.125$ の範囲（これも正方形の範囲）に出力を変更したいときは、どのような変換式が必要だろうか。それは Terminal の $0 - 40$ を変換すればよいので、

Terminal の行範囲	変換式	im 座標の範囲
$y : 0 - 40$	$(160 - y)/160$	$0.75 \leq \text{im} \leq 1$
Terminal のカラム範囲	変換式	re 座標の範囲
$x : 0 - 40$	$(x - 20)/160$	$-0.125 \leq \text{re} \leq 0.125$

とすればよい。これを Terminal の 40 行 40 列の範囲として見るわけだから、この場合は狭い範囲が拡大されて表示されていることになる。

TRY! 上記の範囲に変更した `[mandel12.cpp]` を実行してみよ。

上記の範囲は、マンデルブロ集合の右端から $1/4$ 付近の上部である。最初の `[mandel12.cpp]` の出力画面では、右上方に小さな“穴”が開いている付近、その拡大図になる。小さな穴から想像できない複雑さがあるようだ。実は、それがマンデルブロ集合の不思議なところで、収束・発散の境目付近を拡大しても明確な境界が見えるわけではなく、別の複雑な模様が浮かび上がる。さらに範囲を狭めて、つまり拡大率を大きくしても一向に明確な境界が見えることはない。

こうなると、いろいろな範囲を設定したくなるものだが、範囲を変えるたびにプログラムを書き換えるのでは効率が悪い。`re` 座標と `im` 座標を指定すると、たとえばその点を描画範囲の左上隅として縦横 0.025 の範囲（つまり `[mandel12.cpp]` で描画した範囲の $1/10$ の正方形）を表示するようにしてみよう。

`[mandel12.cpp]` に手を加えるのが目的なら、描画範囲を変えるのとは別の方法について述べておこう。それは、漸化式 $z_{n+1} = z_n^2 + c$ をの扱いを変えることである。マンデルブロ集合は

$z_{n+1} = z_n^2 + c$ において、 c をいろいろな定数にして $z_0 = (0, 0)$ から収束・発散を調査するものであった。これを、 c は一つの定数に固定して $z_0 = (re, im)$ を変化させて収束・発散を調査すると、ジュリア¹集合と呼ばれるものになる。他にも漸化式を少し変えるなどして、マンデルブロ集合の亜種を作ることができるし、実際たくさんのものが研究されている。

TRY! ものは試しなので、ジュリア集合を表示させてみよ。その際は、描画範囲も修正する必要があるかもしれない。

¹ガストン・モーリス・ジュリア (1893–1978) : フランスの数学者