

## 9.4 もっと狭い範囲で

では早速、マンデルブロ集合の狭い範囲を見ることにしよう。プログラムの仕様は、`re` 座標と `im` 座標を指定すると、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 `[mandel2.cpp]` で描画した範囲の 1/10 の正方形）を表示するものである。

---

programming list [\[mandel3.cpp\]](#)

---

```
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:     std::complex<double> w;
16:     std::cout << "input '(a, b)' as top-left: ";
17:     std::cin >> w;
18:
19:     for(int y = 0; y < 40; y++) {
20:         // double im = 1.25 - 0.0625 * y;
21:         double im = w.imag() - 0.00625 * y;
22:
23:         for(int x = 0; x < 40; x++) {
24:             // double re = -2 + 0.0625 * x;
25:             double re = w.real() + 0.00625 * x;
26:
27:             std::complex<double> w(re, im);
28:             int color = recur(w);
29:             if(color > 0) {
30:                 std::cout << " " << color % 8 + 1;
31:             } else {
32:                 std::cout << " ";
33:             }
34:         }
35:         std::cout << std::endl;
36:     }
37:     std::cout << std::endl;
38:
39:     return 0;
40: }
```

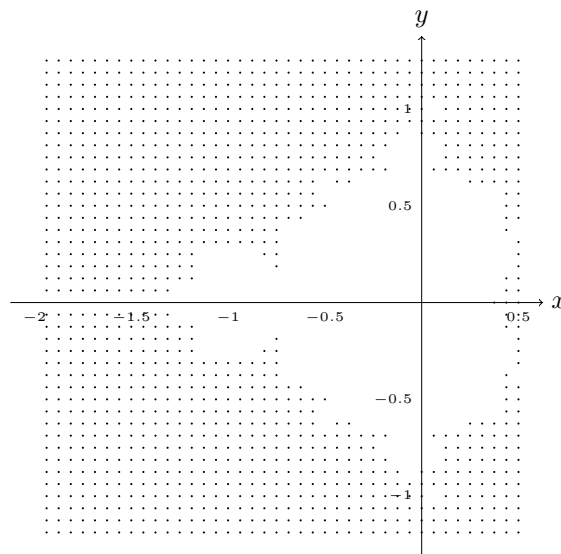
---

変更部分は、あらかじめ決まっていた範囲を、入力した値に差し替えられるようにすることである。よって `recur()` 関数に変更はない。左上隅にあたる座標の入力を促すコードを追加するだけだ。それが 15:-17:行目である。

今回、入力した値は座標変換の必要はなく、単に 40 回に分けて収束・発散を調べればよい。だから、入力された  $(x, y)$  をそのまま変化させればよい。それが 21:, 25:行目である。20:, 24:行目でコメントアウトになっているコードは、本来のマンドルブロ集合を表示する。コメントアウト行を差し替えると本来のマンドルブロ集合が表示されるが、入力要求には  $(-2, 1.25)$  を入力すること。

今回はその  $1/10$  の大きさの範囲なので、 $x, y$  の増分も  $1/10$  になっていることに注意されたい。この増分を変えれば、さらに異なる描画範囲を指定できる。

**TRY!** このプログラムで、いろいろな範囲を描画してみよ。参考のために、本来のマンドルブロ集合の簡素な座標図を示しておこう。もちろん、収束・発散の境目あたりの座標を指定しないと模様にならない。図の輪郭付近の範囲を指定しよう。



**TRY!** 描画範囲ををもっと狭くして試してみよ。ただし、そうすると境目あたりの座標をピンポイントで指定するのは難しくなる。それならば、座標を入力したらそれを少しずつ移動させ、連続表示すればよい。たくさん表示するほど境目を鑑賞できる確率が高まる。ちょっと大変かもしれないが挑戦してみよう。

さあ、この地の見所はたくさんあるようだが、Terminal に出力するのでは視界の悪い景色を眺めているようなものだ。旅の装備は軽いから多くは望めないけど、もっと詳しく展望したいなら、グラフィックが描けるような環境でプログラムしよう。マンドルブロ集合の出力プログラムは十数

行であっても、丁寧なグラフィック表示をさせるならそれ以上のコードを書かなくちゃならない。  
でも、絶景を眺めるなら苦勞を惜しんじゃいけないな。え？ インターネットで見られるって？  
たしかにね。