1 BASIC で描く自己相似図形

1.1　自己相似図形

1.1.1　　相似縮小変換と自己相似図形

相似縮小変換 f_1, f_2 に対して，$f_1(D) \cup f_2(D) = D$ となる図形 D を自己相似図形といいます。Dは有界であるものと仮定すれば，D を描く手続きを次のように作成することで，D を描くことができます。

D を描く手続き
　　$f_1(D)$を描く
　　$f_2(D)$を描く
手続きの終わり

f_1，f_2 が縮小写像であることから，この再帰手続きを繰り返すと，描くべき図形の直径はディスプレーの解像度に比べて十分に小さくなります。そのとき，問題の図形を 1 点で代替して描くことができます。

1.1.2　　表向き相似変換の例

まず，表向き相似変換に関する例として，f_1 が 原点を中心とする 45° の回転と $\frac{1}{\sqrt{2}}$ 倍の縮小の合成写像，f_2 が 点（1,0）を中心とする 45° の回転と $\frac{1}{\sqrt{2}}$ 倍の縮小の合成写像である場合を考察してみましょう。

原点を中心とする角 α の回転を rotate (α)，原点を中心とする r 倍の拡大を scale (r)，x 軸方向に a, y 軸方向に b の平行移動を　shift (a , b), 変換 T_1, T_2 の合成を $T_1 * T_2$ で表すと，f_1，f_2 は，f_1 = rotate ($\frac{\pi}{4}$) * scale ($\frac{1}{\sqrt{2}}$)，f_2 = shift (1,0) * rotate ($\frac{\pi}{4}$) * scale ($\frac{1}{\sqrt{2}}$) * shift (1,0)

と表すことができます。複素数を用いて表せば $f_1(z) = \left(\frac{1}{2}+\frac{1}{2}i\right)z$，$f_2(z) = \left(\frac{1}{2}+\frac{1}{2}i\right)(z-1)+1$ です。D を描く手続きのなかで $f_1(D)$や $f_2(D)$を描くのに，Full BASIC の図形の変換の機能を用います。

例 1

```
100  PICTURE  D(n)
110    IF  n=18  THEN
120      PLOT POINTS:  0,0
130    ELSE
140      DRAW  D(n+1) WITH ROTATE(PI/4)*SCALE(1/sqr(2))
150      DRAW  D(n+1) WITH SHIFT(-1,0)*ROTATE(PI/4)*SCALE(1/sqr(2))*SHIFT(1,0)
160    END IF
170  END PICTURE
180  SET WINDOW  -1/2 ,  3/2 ,  -1 ,  1
190  SET POINT STYLE  1
210  DRAW  D(0)
220  END
```

100 行から 170 行までは，図形 D を描くための手続き D を定義する部分です。プログラムは 180 行から実行が開始されます。

180 行では，描画領域に x 座標の範囲が $-\frac{1}{2}$ から $\frac{3}{2}$，y 座標の範囲が -1 から 1 までとなる座標系を設定しています。描画領域の形状は正方形なので，x 座標の幅と y 座標の幅とが等しくなるようにしてあります。

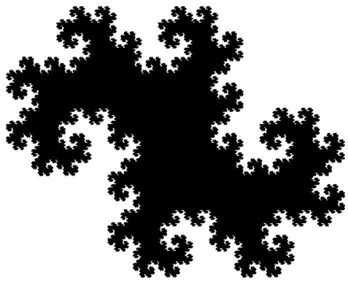
190 行は，120 行の plot points 文で描かれるマークの形状を・に変更しています。

210 行は, 引数 n に 0 を代入して 100 行から始まる絵定義（picture）D を呼び出す文です。100 行から始まる絵定義 D で, 引数 n は再帰の深さを制御するためのものです。140 行と 150 行で再帰的に D を呼び出すときに n を 1 加算しています。 n の値が 110 行の if 文で指定している数値 18 になると再帰呼び出しを終了し，変換された座標系の原点にマークを描きます。

140 行と 150 行が，再帰呼び出しを利用して D を f_1，f_2 で変換して描くことにより $f_1(D)$ と $f_2(D)$を描く部分です。

110 行で指定する定数は，おおむね，f_1，f_2 の縮小率 s と描画領域の縦横方向のピクセル数 p とから決定できます。描くべき図形 D の直径が描画領域の幅より小さいものと仮定すれば, $s^n < \frac{1}{p}$ となるように n を選ぶことが一応の目安になるでしょう。

描画領域のピクセル数を 501×501 として得た実行結果を示します。この図形は，これを並べることによって平面の敷き詰めが可能な図形として知られています。



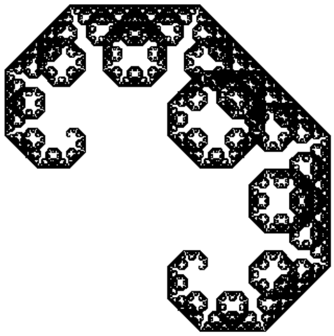
1.1.3　裏向き相似変換の例

x 軸方向に a 倍，y 軸方向に b 倍の拡大を scale(a,b)で表すとき，scale(1,-1)は x 軸に関する対称移動です。これは，複素数平面では共役数をとる演算　$z \to \bar{z}$　に対応します。

複素数を用いて表したとき，$f_1(z)=\left(\frac{1}{2}+\frac{1}{2}i\right)z$，$f_2(z)=\left(\frac{1}{2}+\frac{1}{2}i\right)\left(\bar{z}-1\right)+1$ に関する自己相似図形を求めてみると次のずのようになります。プログラムは，例 1 の 150 行の WITH 以降を

```
SCALE(1,-1)*SHIFT(-1,0)*ROTATE(PI/4)* SCALE(1/sqr(2))*SHIFT(1,0)
```

に修正したものになります。



　　また，同様にして，$f_1(z)=\left(\frac{1}{2}+\frac{1}{2}i\right)z$，$f_2(z)=\overline{\left(\frac{1}{2}+\frac{1}{2}i\right)(z-1)+1}$ に関する自己相似図形を求めてみると次のようになります。プログラムは，150 行の WITH 以降を

```
SHIFT(-1,0)*ROTATE(PI/4)* SCALE(1/sqr(2))*SHIFT(1,0)*SCALE(1,-1)
```

に修正したものです。

1.1.4　　コッホ曲線

　コッホ曲線は，タートルグラフィックスの応用例としてよく知られた図形ですが，それとは別に，自己相似図形の観点から次のように定義することができます。

　コッホ曲線は，底辺 AB の長さが 3 で高さが $\frac{\sqrt{3}}{2}$ の二等辺三角形 CAB で，頂点 A を不動点として，点 B を点 C に移し，点 C を線分 AB 上の点に移すような相似変換を f_1，頂点 B を不動点として，点 A を点 C に移し，点 C を線分 BA 上の点に移すような相似変換を f_2 とするとき，$f_1(D)\cup f_2(D)=D$ となる図形 D です。

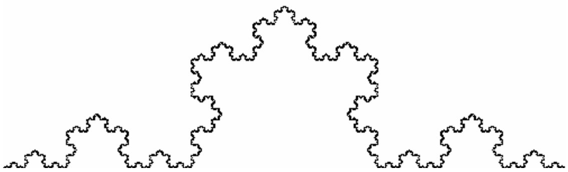
　ここで，各頂点の座標を A (0,0)，B (3,0)，C $(\frac{3}{2},\frac{\sqrt{3}}{2})$ とし，f_1，f_2 を

　f_1＝SCALE (1,−1) * ROTATE (α) * SCALE (r)

　f_2＝SHIFT (−3,0) * SCALE (1,−1) * ROTATE(− α)*SCALE (r) * SHIFT (3,0)

の形に表すものとすると，縮小写像 f_1，f_2 の倍率 r は線分 AC の長さを線分 AB の長さで割って求められ，回転角 α は点 C の偏角として求められます。

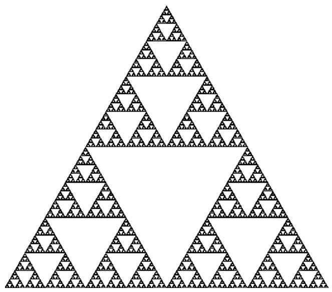
　例 2は，コッホ曲線を描くプログラムです。再帰のネスティングの制御に例 1と異なる方法を採用しています。200 行の絵定義 Koch の引数 s は総合の縮小率を表す変数です。この値が 210 行で定める大きさを下回ったとき再帰呼び出しを止めています。



例 2

```
200 PICTURE Koch(s)
210   IF s<1/500 THEN
220     PLOT POINTS: 0,0
230   ELSE
240     DRAW Koch(s*r) WITH SCALE(1,-1)*ROTATE(alfa)*SCALE(r)
250     DRAW Koch(s*r) WITH SHIFT(-3,0)*SCALE(1,-1)*ROTATE(-alfa)*SCALE(r)*SHIFT(3,0)
260   END IF
270 END PICTURE
280 LET r=SQR((3/2)^2+(SQR(3)/2)^2) / 3
290 LET alfa=ANGLE(3/2,SQR(3)/2)
300 SET WINDOW 0,3,0,3
310 SET POINT STYLE 1
320 DRAW Koch(1)
330 END
```

1.1.5　一般の自己相似図形



例 2で用いた手法を応用すれば，f_1，f_2 で縮小率が異なる場合でも効率よく自己相似図形を求めることができます。そして，さらにこの手法は，任意の相似縮小変換 f_1，f_2，$\cdots f_k$ に対して，それらに関する自己相似図形を求めるのに応用することができます。たとえば，シルピンスキーのガスケットは正3角形の3頂点を中心とする $\frac{1}{2}$ 倍の縮小写像に関する自己相似図形ですが，次に示すようなプログラムで描くことができます。

```
200 PICTURE gasket(n)
210    IF n=10 THEN
220       PLOT POINTS: 0,0
230    ELSE
240       DRAW gasket(n+1) WITH SCALE(1/2)
250       DRAW gasket(n+1) WITH SHIFT(-2,0)*SCALE(1/2)*SHIFT(2,0)
260       DRAW gasket(n+1) WITH SHIFT(-1,-SQR(3))*SCALE(1/2)*SHIFT(1,SQR(3))
270    END IF
280 END PICTURE
290 SET WINDOW 0,2,0,2
300 SET POINT STYLE 1
310 DRAW gasket(0)
320 END
```