グラフィクスとＣＡＤ　26－2
（1987. 5. 18）

自己相似法による非周期パターンの作成

渡辺泰成[**]　相馬　嵩[*]　出澤正徳[*]
理化学研究所　情報科学研究室[*]　結晶物理研究室[**]

20方相対称性を示すAl-Mn薄膜合金が発見されて以来，そのモデルの基礎となるペンローズ・パターンに対する関心が高まっている．ここでは，自己相似法による2次元非周期パターン（準周期パターン）の生成法について考える．自己相似法の基礎となる基本パターンの分割法を一般的に求める方法は知られていない。世代間の縮小率が大きくなると，分割の仕方も一般的に多くなるが，それを分類することによって，分割法アルゴリズムへの手がかりを探る．

GENERATION OF NON-PERIODIC PATTERNS BY SELF-SIMILAR OPERATION

Yasunari WATANABE,[**] Takashi SOMA[*] and Masanori IDESAWA[*]
Information Science Laboratory,[*] Crystal Physics Laboratory[**]
The Institute of Physical and Chemical Research
2-1 Hirosawa, Wako-shi, Saitama 351-01 Japan

Since the discovery of Al-Mn alloy showing icosahedral phase symmetry, Penrose pattern has been drawing much attention because of its property useful for modeling such metal. We discuss the method of self-similar transformation to generate two-dimensional non-periodic or quasi-periodic patterns like Penrose pattern. No algorithm is known for finding the rules of dividing the basic pattern for the self-similar transformation generating patterns of n-fold symmetry. We try to classify the dividing rules with the aim of obtaining the above algorithm.

1. はじめに

20方相(Icosahedral phase)対称性を示すAl-Mn薄膜合金が発見[1]されて以来，この種の物質に対して数多くの研究がなされて来た。この20方相は，その原子の配列が本来の結晶相やガラス相とは異なる全く新しい物質相と考えられることから，この発見は非常な注目を集めた。この物質の構造に対して，現在までに種々のモデルが提案され実験との比較検討が重ねられて来たが，その解釈について，まだ結着をみるに到っていない。

この20方相は2次元におけるペンローズ(Penrose)パターンの3次元への拡張と考えられるもので，この物質発見以前から研究がなされていた。発見以後は，この新しい物質相に対し準結晶(Quaisi crystal)相と云う名前が広く用いられるようになり，ペンローズ・パターンはその有力なモデルの一つとして大いに関心を集めると同時に，ペンローズ・パターン自身の研究も大きな進展を遂げることになった。[2-4]

そもそもペンローズ・パターンとは，限られた種類のタイルを用いるタイル貼りにおいて非周期パターンしか作らないようなタイルの組によるパターンの一つであり，その性質は代数的に説明することが出来る。従ってその3次元への拡張など一般化は，数学的に容易に行うことが出来る。ここでは，ペンローズ・パターンを含めて一般の2次元非周期パターンの生成法について考える。第2章では，非周期パターン，或いは準周期パターン(Quasiperiodic pattern)の生成法のおさらいをして，第3章で自己相似法について例を上げて説明する。特に自己相似法で生成されるパターンの分類を試みる。

2. 2次元非周期パターン

図1は二種類の菱形タイルから成るペンローズ・パターンを示す。タイルの組合せ方は，図2に示すようにそれぞれの菱形の各辺に二種類の矢印を付けた場合，同じ種類の矢印が同じ向きに重なるようになっていることがわかる。また図1のペンローズ・パターンは，頂点と格子窓(mesh)とを逆にした相対(dual)グラフと一対一に対応すること，このグラフは等間隔平行格子5組を順に $\pi/5$ ずつ回転してできる5重格子(pentagrid)と位相的に等価であることなどが示される。図3(a)に示すように，互に平行な辺を順につないで得られる帯状領域(strip) a-b-c-d-e-f-g …が，図3(b)に示す5重格子の格子線にそっての線分 a-b-c-d-e-f-g に対応している。

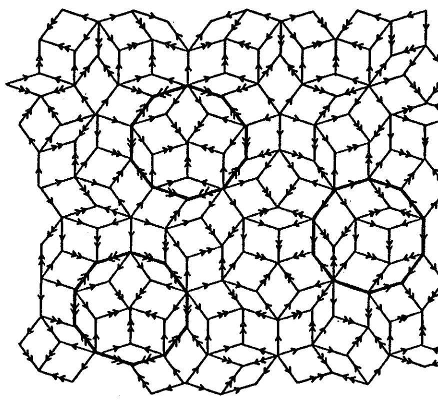


図1. ペンローズ・パターン

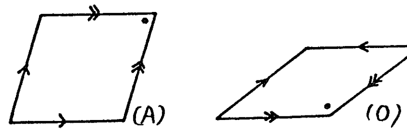


図2　二種類の菱形タイル

de Bruijn[5]はペンローズ・パターンを複素平面上で考え，各頂点を5つの基本ベクトル，$(1, \zeta, \zeta^2, \zeta^3, \zeta^4)$，(但し $\zeta = e^{2\pi i/5}$) を用いて，$k_0 + k_1\zeta + k_2\zeta^2 + k_3\zeta^3 + k_4\zeta^4$ と表わしたとき k_j は条件：

$$\exists z \in C,\ {}^{\forall}j\,(k_j - 1 < \mathrm{Re}(z\zeta^{-j}) + \gamma_j < k_j) \quad (1)$$

から決定されること，更にこの條件は

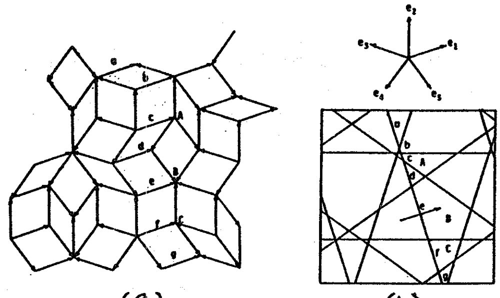
$$V = \{(\Sigma\lambda_j, \Sigma\lambda_j\zeta^{2j}) \mid 0<\lambda_0<1, \cdots 0<\lambda_4<1\} \quad (2)$$

として

$$(\Sigma k_j, \Sigma(k_j - \gamma_j)\zeta^{2j}) \in V \quad (3)$$

と等価であることを示した。但し，ここで γ_j は平行格子を組合せて5重格子を作る際の移動量である。即ち平行格子を

$$\{z \in C \mid \mathrm{Re}(z\zeta^{-j}) + \gamma_j \in Z\} \quad (4)$$

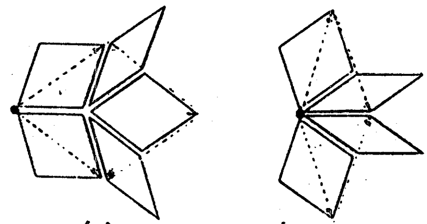


(a) (b)

図3 帯状領域と5重格子

とする。Zは整数の集合を表わすものとする。$\Sigma k_j\zeta^j$ を5次元立方格子空間から2次元のパターン空間への射影と考えると(3)左辺はパターン空間と直交する補空間(3次元)への射影と考えることが出来て條件(3)は，(2)で示す5次元単位立方の補空間への射影に左辺が含まれることを表わしている。このアルゴリズムによるペンローズパターン生成プログラムは石原[6]を，また，n回対称のパターンを生成する方法については石原[7]を参照されたい。Kramer[8]は，12次元から3次元空間への射影としてペンローズパターンの3次元への拡張を行った。

MacKay[9]は，図4(a),(b)に示すように二つの菱形タイルの分割を再帰的に繰返す自己相似変換によりペンローズパターンが得られることを示した。次章ではこの自己相似変換についてくわしく考えてみたい。



(a) (b)

図4 菱形タイルの自己相似分割

3．自己相似変換法

これは上に述べた射影法に比べて直感的な方法である。射影法により得られるパターンのうち，自己相似の條件を満たす特別なものとみなすことができる。その分割法は図4に示す通りであるが，・印を付けた頂は極（pole）と呼ばれ，極に対向する頂点を結ぶ軸に対して対称なパターンとなっている。従って，菱形を指定する場合，半分の山形とそれが軸の上側か下側かを指定するのが都合がよい。極頂点の角度が鋭角か鈍角かで菱形を区別し，鋭角菱形（A: acute rqhombus）と鈍角菱形（O: obtuse rohmbus）と呼ぶ。この自己相似分割により，菱形Aは図4(a)に示すように2個のAと1個のOに，菱形Oは図4(b)に示すように1個のAと1個のOに分割されることがわかる。このことから分割による相似パターンの縮小率 α は，黄金分割比でτに等しいことが示される。また，この分割のように，次世代の菱形が菱形の境界にまたがるような場合には，各次世代菱形に対する極の位置を指定する必要がある。この場合は，極でない頂点に属する頂点を次世代菱形の極として選ぶ必要がある。この分割を繰返し行った極限で菱形AとOの数の比がτに収斂することが示されるが，τが無理数であることからこのペンローズパターンの非周期性が証明される。

3.1 自己相似法の変形

i) $\alpha=\tau^2$

図4による相似変換以外には縮小率τの系列は知られていない。そこでこの系列の第2世代を第1世代とみなし、菱形タイルの並べ換えを行ない別の系列を作ることを考える。図5(a),(b)は基本系列における第2世代を示す。(c)と(d)は、(b)

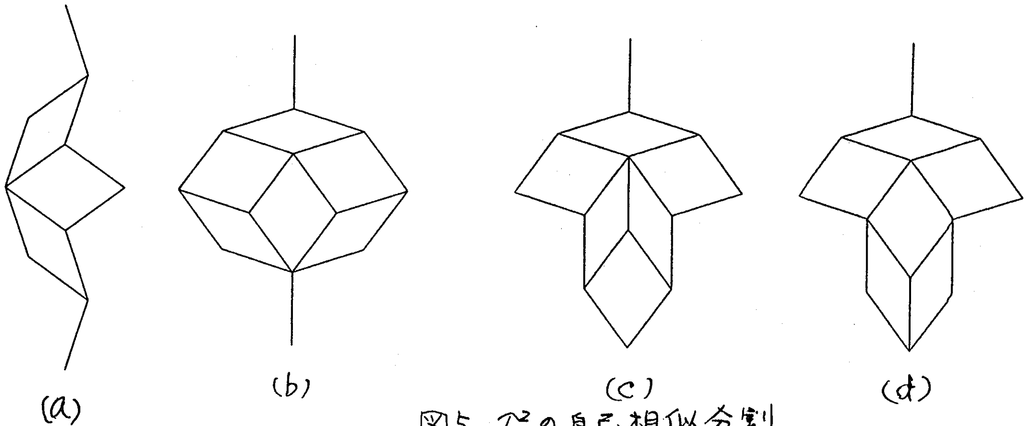


図5 τ^2の自己相似分割

のタイルを並べ替えたもので[10]、それぞれ(1)と組合せてこれを第1世代とする自己相似変換系列を得ることができる。いずれもペンローズパターンとはならないが非周期の條件は満たしている。

ii) $\tau<\alpha<\tau^2$

最小縮小率系列の縮小率の二乗より小さい縮小率をもつ系列で8回対称パターン[11,12]の例をあげて説明する。図6(a),(b)は最小縮小率系列の第1世代を、(c),(d)が変

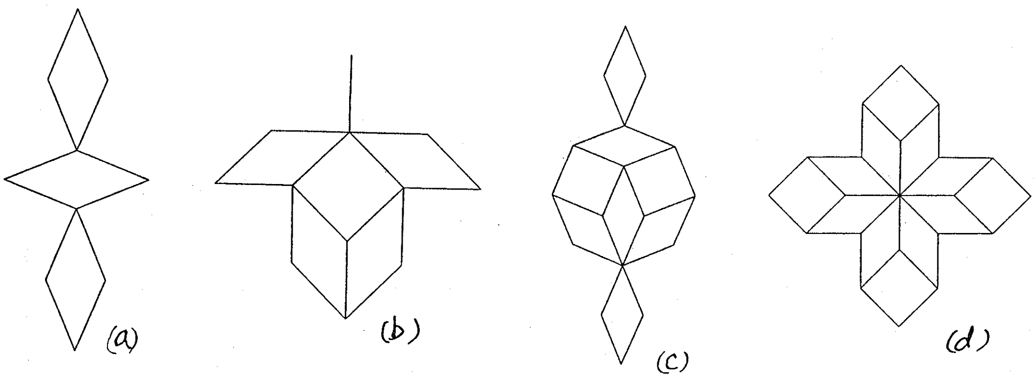
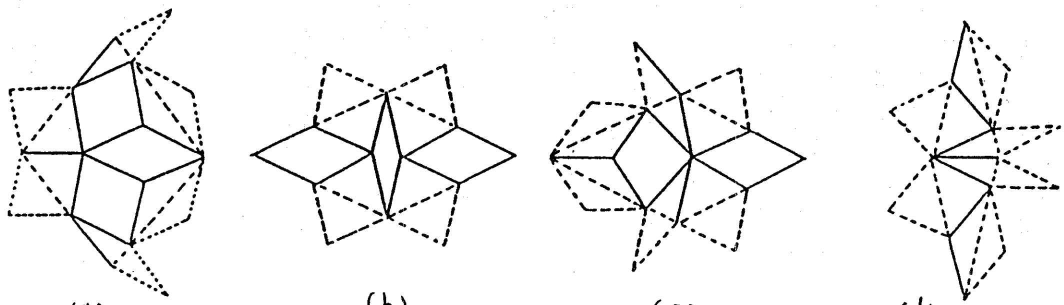


図6 8回対称自己相似分割

形系列の第1世代の分割を示す。縮小率は前者が $1+\sqrt{2}$、に対し後者が $2+\sqrt{2}$ となって上の條件を満たしている。(c)の中央部に8角形があり、これを回転して入れ替えたものは、それぞれ別の系列を作ることがわかる。

iii) 同一タイルに複数個分割法を割当てる。

同一のタイルに対し複数個の分割法を割当て、そのタイルが置かれる相対位置によりそれらの分割法を選ぶ方法が考えられ、これを広義の自己相似変換と呼ぶことにする。Sasisekharan[13]は7回対称の非周期パターンを得るのに図7に示す分割を報告している。(b)と(c)は同じタイルに対する異なる分割を示す。此の場合大きな縮小率の変換を考えれば通常の自己相似変換にできると予想される。



(a)　(b)　(c)　(d)

図7　7回対称自己相似分割

4. むすび

ペンローズ・パターンに代表される非周期タイル貼りパターンの生成法について種々の方法を考察した。自己相似縮小変換については、最小縮小率系列をもとに、縮小率の値による分類を試みた。最小縮小率系列の分割法を求める一般的アルゴリズムは知られていない。代数的な取扱いによりそれが可能と予想される。今は各事例を集める段階だと思われる。

参考文献

[1] Shechtman, D. etal: Phys. Rev. Letters, Vol53, No.20, 12 Nov. 1984, pp 1951-1953.

[2] Physics Today / Feb. 1985, pp 17-19.

[3] Nelson, D.R. and Halperin, B.I., Science Vol 229, No.4710 July 1985, pp 233-238.

[4] Steinhardt, P.J., American Scientist Vol 74, Nov-Dec. 1986, pp 586-596

[5] de Bruijn, N.G., Ned. Akad. Weten. Proc. Ser. A, 43, 1981 pp 39-52, 53-66.

[6] 石原慶一，金属 1985年11月 pp.50-56

[7] 石原慶一，理研シンポジウム準結晶の構造とその周辺，1987年1月 pp 4-7.

[8] Kramer, P. and Neri, R., Acta Cryst. A40, 1984 pp 580-587.

[9] MacKay, A.L., Physica 114A, 1982 pp 609-613.

[10] Ogawa, T., Proc. First Intl. Symp. for Science on Form, 1986, pp. 479-489,

[11] Watanabe, Y., etal. Proc. First Intl. Symp. for Science on Form. 1986 pp. 471-477,

[12] Watanabe, Y., etal Acta, Cryst. A43, 1987, pp 133-134.

[13] Sasisekharan, V., Pramana - J. Phys., Vol 26, No.3, 1986, pp. L283-L293.

Appendix：自己相似法によりペンローズパターンを描くためのプログラム.

```
#include <stdio.h>
#include <math.h>
#include "plotmax.h"

float pi,pi15,pi25,pi35,pi45,a,a1,b,b1;
float min_length;
float xorg,yorg;

main(argc,argv)
int argc;
char *argv[];
{

    long atoi();
    double atof(),atan(),cos(),pow();
    float length,angle,gen,angledeg,xstart=10,ystart=20;

    pi = atan(1.)*4.;
    pi15 = pi/5.;
    pi25 = pi15*2.;
    pi35 = pi15*3.;
    pi45 = pi15*4.;
    a = 2.*cos(pi15);
    a1 = 1./a;       /* a1 = 0.618034... */
    b = 2.*cos(pi25);
    b1 = 1./b;

    length = atof(argv[1]);
    length = 2*length*cos(pi15);
    angledeg = atof(argv[2]);
    angle = angledeg*pi/180.;
    gen = atof(argv[3]);
    min_length = length*pow(a1, gen) + 0.001;

    plots();

    symbol(5.,0.,3.,"parameters: len=",0.,0);
    number(999.,0.,3.,length,0.,1);
    symbol(999.,0.,3.,", ang=",0.,0);
    number(999.,0.,3.,angledeg,0.,1);
    symbol(999.,0.,3.,", gen=",0.,0);
    number(999.,0.,3.,gen,0.,-1);

    plot(xstart, ystart, -3);

    if(strlen(argv[4])) {
        xorg = X0;
        yorg = Y0;
    }

    penrose(length, angle, 1,  1);
    penrose(length, angle, 1, -1);

    plot(0.,0.,999);
}

penrose(length, angle, type, sign)
float length,angle;

int type,sign;
{

    if(type == 1) {
        if(length < min_length) {
            draw_line(length*a1, angle+sign*pi15);
            draw_line(length*a1, angle-sign*pi15);
            move_line(length, angle+pi);
        } else {
            move_line(length*a1, angle+sign*pi15);
            penrose(length*a1, angle-sign*pi45, 1, sign);
            penrose(length*a1*(1.-a1), angle-sign*pi15, 0, -sign);
            move_line(length*a1, angle-sign*pi15);
            penrose(length*a1, angle+sign*pi, 1, -sign);
            move_line(length, angle+pi);
        }
    } else {
        if(length < min_length*(1. - a1)) {
            draw_line(length*b1, angle+sign*pi25);
            draw_line(length*b1, angle-sign*pi25);
            move_line(length, angle+pi);
        } else {
            move_line(length*b1, angle+sign*pi25);
            penrose(length*b1, angle-sign*pi35, 1, sign);
            move_line(length*b1, angle-sign*pi25);
            penrose(length*b1*(1.-a1), angle+sign*pi35, 0, sign);
            move_line(length, angle+pi);
        }
    }
}

draw_line(length, angle)
float length,angle;
{

    double sin(),cos();

    plot(length*cos(angle), length*sin(angle), -2);
}

move_line(length, angle)
float length,angle;
{

    double sin(),cos();

    plot(length*cos(angle), length*sin(angle), -3);
}
```