

群論と幾何学模様の対称性—その1 —Jグラフィックスからのアプローチ—

西川 利男

1. はじめに—群論とは何か

群論は難しい。微分積分や線形代数とは全く違う考え方をする。現代数学としては極めて重要な分野であるにもかかわらず、従来の数学教育では無用の学とされてきた。群論の始まりがアーベルやガロワの5次方程式の解明からだったのは、われわれにとって不幸なことであった。

群論が難解の理由として、その導入の方法に問題があると思う。まずは具体的に形の見える幾何学図形の対称性の解析分類など、数学の計算とは違う構造の理解、処理の道具として身に付け、その後と同じ方法が方程式の構造など、さらに抽象的ないろいろな方面で有用であるというストーリーを進めたらよいと思う。

われわれJを知るものにとってありがたいことは、Jの動詞と名詞という考え方である。すなわち、群(Group)の構成要素である元(Element)は、操作(Operation)であり動詞(Verb)である。一方、その操作されるもの(Operand)は、つまり図形や数値などは、名詞(Noun)であり、その集まりは集合(Set)である。決して群ではない！

これまで私の興味からルービック・キューブのパズルに関連して群論について何回か報告した [1]。

[1] 西川利男「J言語からの群論の理解—その2」JAPLA 研究会資料 2011/11/26

西川利男「J言語からの群論の理解—その3—直接置換、巡回置換、互換、隣接互換」
JAPLA シンポジウム資料 2011/12/10

一方、志村正人氏は「葛飾北斎の江戸小紋」をとりあげ、そのJグラフィックスを美術的側面から精力的に研究していらっしゃる [2]。

[2] 志村正人「北斎の新形小紋帳から J6 版」JAPLA 2012/12/10

志村正人「北斎の新形小紋帳から (その2) J6 版」JAPLA 2012/7/12

志村正人「北斎の新形小紋帳にフォームをつける (J6 版)」JAPLA 2012/12/6

私は、違う見方として、分子・原子構造から結晶学に向けての群論にアプローチしたいと思っている。そして、まずはごく基本的なところから、Jグラフィックスの力をかりて、プログラムしつつ、進めたいと思う。

群論の本は多数あるが、私にとってアカデミックな数学者の本は「猫に小判」である。今回参考にしたのは以下の本で、読みやすく良い本だと思う。

- [3] 伏見康治、安野光雅、中村義作「美の幾何学」早川書房(2010).
- [4] マルコム・ラインズ、青木薫訳「物理と数学の不思議な関係」早川書房(2012).
- [5] M.A.Armstrong,"Group and Symmetry",Springer(1988).
佐藤信哉訳「対称性からの群論」
- [6] 難波誠「群と幾何学」現代数学社(1997).
- [7] 笹田健一「対称性と数学」技術評論社(2016).

2. 連続帯模様の群＝壁紙群（格子群、結晶群）とは

群論でよくないのは、その用語の使い方であると思う。群の操作を言っているのか、使われる対象分野を言っているのか。あまりに哲学的？で抽象的である。単に対称群など何と言っているのかわからない、まさにペダンティックである。私に言わせれば、まずは使う人の目線から、名前を付けたらよい。数学手順などその枝葉にすぎない。

連続帯模様の群は、壁紙群(Wallpaper Group)が妥当で良いと思う。
物理化学の本では格子群や結晶群という。原子分子の群は、ふつう点群という。

数学的には、次の操作に関連して命名されている。

点群.....一つの点の周りの対称性.....回転、鏡映の操作.....数学的には直交変換
壁紙群...繰り返し移動の操作.....上の操作に連続移動が加わる.....合同変換

3. 壁紙群（格子群、結晶群）の対称操作

志村氏が扱っている江戸小紋のような美的なきれいな模様は最終目標であるが、群論に照らし合わせて、自分で一步一步プログラムするのは最初は無理だ。極易しいところから始めるしかない。伏見、安野、中村の「美の幾何学」p.74-101には、伏見先生の手による「対称性の基本」として、非常にていねいな、かつ正確な解説がある。

ここでは、格子群でよく使われるワイルの記号の説明が一つずつされている。

p2.....2 回回転
pm.....鏡映
pg.....滑り鏡映
p4.....4 回回転
p6.....6 回回転
cm.....中心鏡映

.....

これらの群操作を大規模に行うことで、さまざまな江戸小紋が生みだされる。また、ヨーロッパ中世の紋章もそうだし、イスラムのアラビア模様(Arabesque)もある[8]。

[8] ダウド・サットン、武井摩利訳「イスラム芸術の幾何学」創元社(2011).

4. Jによる図形座標値の3つの方法

複数の点の集まりからなる平面上の図形は、複数の座標値 (X, Y) の集合で表現されるが、Jでは3つの方式があり目的に応じて使い分けることになる。

(1) 1点ごとにボックスで囲む

(x0, y0);(x1, y1);(x2,y2);

例えば (0.5, 1); (1, 2); (1.5, 3); (2, 2)

```
+-----+-----+-----+-----+
=|0.5 1|1 2|1.5 3|2 2|
+-----+-----+-----+-----+
```

(2) n×2 配列 図形の回転、移動などの計算の際に使う

x0 y0

x1 y1

x2 y2

x3 y3

(3) 単純リスト Jのグラフィック命令として

gllines x0, y0,x1,y1,x2,y2,x3,y3

gpolygon x0, y0,x1,y1,x2,y2,x3,y3

Jの利用で、相互の方式はコンパクトなプログラムにより変換できる。

5. Jによる対称操作のプログラム

伏見先生の本[3]から、**p 2**(2回回転)操作をとりあげる。

基本の図形としては不等辺3角形の内部に「J」の文字が入った図形で行う。

3角形PQRのJの定義はつぎのようになる。

P =: (_2, _1)

Q =: (2, _1)

R =: (0.5, 2)

FIG0 =: P; Q; R; P

「J」の文字は半円と長方形の結合として作成した。

th =: 1r12p1 * i. 13

JCHAR0 =: , _0.5 * |: > (cos th);(sin th)

JCHAR1 =: (_0.5, 0), (_1, 0)

JCHAR2 =: , _1 * |: > (cos th);(sin th)

JCHAR3 =: (0.5, 0), (0.5, 2), (1, 2), (1, 0)

JCHAR =: JCHAR0, JCHAR1, JCHAR2, JCHAR3

操作 **p2** は次のような図形の操作で行う。

(1) たとえば、一つの辺 **PQ** の中点 **S** を求めてこれが座標の原点になるよう図形の各点を移動する。

(2) 次にここで、図形の各点を必要な角度だけ回転（ここでは180度）する。

(3) 出来上がった図形の各点を元の位置に移動して戻す。

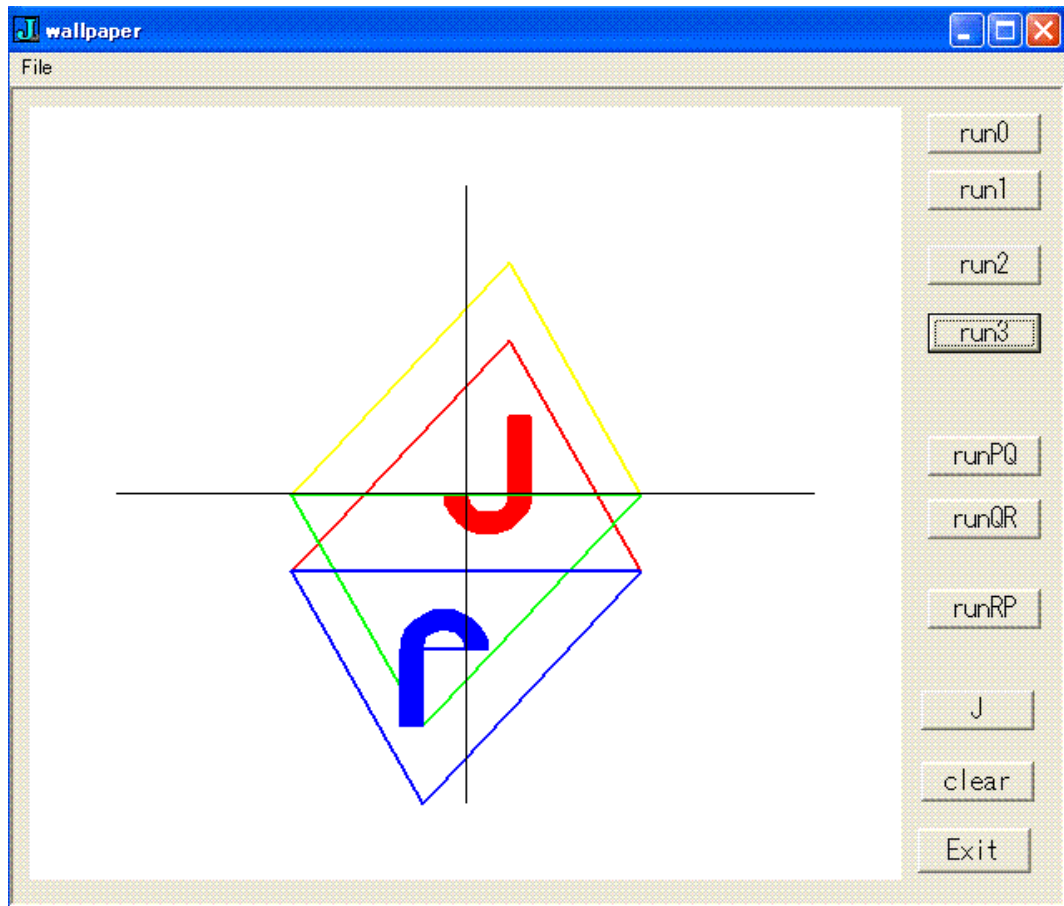
これにより、操作 **p2** が行われる。

以上に対応する **J** のプログラムによる計算のポイントは次のようになる。

```
NB. Start Figure =====
FIG0 =: P; Q; R; P
JCH00 =. 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR NB. adjust size J
JCH0 =: <"(1) (32, 2)$JCH00
NB. Shift =====
S =. (Q + P) % 2
S0 =: (0, 0) - S
FIG1 =: S0 +L:0 FIG0
JCH1 =: S0 +L:0 JCH0
NB. Rotate =====
FIG2XY =. cut2 FIG1
FIG2X =. {."(1) > 180 rot L:0 FIG2XY
FIG2Y =. {:"(1) > 180 rot L:0 FIG2XY
FIG2 =: , |: FIG2X, :FIG2Y
JCH2XY =. cut2 JCH1
JCH2X =. {."(1) > 180 rot L:0 JCH2XY
JCH2Y =. {:"(1) > 180 rot L:0 JCH2XY
JCH2 =: , |: JCH2X, :JCH2Y
NB. Shift Return =====
FIG3 =. (-S0) + L:0 FIG2
JCH3 =. (-S0) + L:0 JCH2
NB. Draw Figure =====
color_pen4 0
gllines adj , > FIG0
glpolygon adj , > JCH0
```

なお、**J** プログラムの詳細なコーディングの実際は最後に示した。

以上の途中経過を示すJグラフィックスは次のようになる。赤い3角形が黄色、緑色を通して、最終的に青い3角形に $p2$ 操作されるのがわかるであろう。



このようにして、群論とくに壁紙群（格子群、結晶群）についての、Jグラフィックスによる基本の操作技術はできたと思っている。つまり $p2$ 操作のプログラムのグラフィックス・プログラムは一応出来上がった。

その他の pm , pg , cm ,...また2回以外の $p4$, $p6$...もあるが、同様にしてプログラムできるはずである。

伏見先生の解説書[3]にそって整備したいと思っている。

Jのプログラムリスティング

NB. Wall Paper Group

NB. wallpaper.ijs by TN 2017/1/16

```
require 'trig'
require 'gl2'

WALLPAPER=: 0 : 0
pc wallpaper;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 272 198 34 12;cc cancel button;cn "Exit";
xywh 5 5 262 207;cc wallpg isigraph;
xywh 273 161 34 11;cc J button;
xywh 273 180 34 11;cc clear button;
xywh 275 7 34 11;cc run0 button;
xywh 275 22 34 11;cc run1 button;
xywh 275 42 34 11;cc run2 button;
xywh 275 60 34 11;cc run3 button;
xywh 275 93 34 11;cc runPQ button;
xywh 275 110 34 11;cc runQR button;
xywh 275 134 34 11;cc runRP button;
pas 6 6;pcenter;
rem form end;
)

run =: wallpaper_run
wallpaper_run=: 3 : 0
wd WALLPAPER
NB. initialize form here
wd 'pshow;'
)
```

```
wallpaper_close=: 3 : 0
wd' pclose'
)
```

```
wallpaper_cancel_button=: 3 : 0
wallpaper_close''
)
```

```
adj =: (500&+)@(100&*)
```

```
NB. cut2 i.10
NB. +---+---+---+---+---+
NB. |0 1|2 3|4 5|6 7|8 9|
NB. +---+---+---+---+---+
cut2 =: 3 : 0
y =. ((-:#y.),2)$y.
<"(1) y
)
```

```
NB. J-Character =====
th =: 1r12p1 * i. 13
```

```
JCHAR0 =: , _0.5 * |: > (cos th);(sin th)
JCHAR1 =: (_0.5, 0), (_1, 0)
JCHAR2 =: , _1 * |: > (cos th);(sin th)
JCHAR3 =: (0.5, 0), (0.5, 2), (1, 2), (1, 0)
JCHAR =: JCHAR0, JCHAR1, JCHAR2, JCHAR3
```

```
wallpaper_J_button=: 3 : 0
NB. glclear ''
glrgb 0 0 255
glbrush ''
glpen 4 0
r =: 0.5
th =: 1r12p1 * i. 13
```



```

JCHAR0 =. , _0.5 * |: > (cos th);(sin th)
JCHAR1 =. (_0.5, 0), (_1, 0)
JCHAR2 =. , _1 * |: > (cos th);(sin th)
JCHAR3 =. (0.5, 0), (0.5, 2), (1, 2), (1, 0)
JCHAR =: JCHAR0, JCHAR1, JCHAR2, JCHAR3
glpolygon adj JCHAR
glshow ''

glrgb 255 0 0
glbrush ''
glpen 4 0
JCHAXY =. cut2 JCHAR
JXR =. {."(1) > 30 rot L:0 JCHAXY
JYR =. {:"(1) > 30 rot L:0 JCHAXY
JXYR =: , |: JXR, :JYR
glpolygon adj JXYR

glshow ''
)

rot =: 3 : 0
:
'x y' =. y.
t =. x.
((x*cosd t) - (y*sind t)), ((x*sind t) + (y*cosd t))
)

NB. Initial Figure =====
P =: (_2, _1)
Q =: (2, _1)
R =: (0.5, 2)
XY =: , > P; Q; R; P
NB. J character =====
J_W =: 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR

```

```
wallpaper_clear_button=: 3 : 0
glclear ''
)
```

```
wallpaper_testp2_button=: 3 : 0
glrgb 0 0 0
glpen 1 0
gllines 100 500 900 500
gllines 500 100 500 900
```

```
glrgb 255 0 0
glbrush ''
glpen 4 0
gllines adj P, Q, R, P
```

```
NB. draw J character
glpen 4 0
glpolygon adj J_W
```

```
NB. shift =====
S =. (Q + P) % 2
S0 =: (0, 0) - S
P1 =. P + S0
Q1 =. Q + S0
R1 =. R + S0
```

```
glrgb 255 255 0
glbrush ''
glpen 4 0
gllines adj P1, Q1, R1, P1
```

```
NB. draw J character
glrgb 0 0 255
glbrush ''
glpen 4 0
J_W1 =. , S0 + "(1) (32, 2)$J_W
```

NB. glpolygon adj J_W1

NB. rotate =====

P2 =. Q1

Q2 =. P1

R2X =. {."(1) > 180 rot L:0 R1

R2Y =. {:"(1) > 180 rot L:0 R1

R2 =. , |: R2X, :R2Y

glrgb 0 255 0

glbrush ''

glpen 4 0

gllines adj , > P2; Q2; R2; P2

NB. draw J character

glrgb 0 0 255

glbrush ''

glpen 4 0

J_W2XY =. cut2 J_W1

J_W2X =. {."(1) > 180 rot L:0 J_W2XY

J_W2Y =. {:"(1) > 180 rot L:0 J_W2XY

J_W2 =. , |: J_W2X, :J_W2Y

NB. glpolygon adj J_W2

NB. return shift =====

P3 =. P2 - S0

Q3 =. Q2 - S0

R3 =. R2 - S0

glrgb 0 0 255

glbrush ''

glpen 4 0

gllines adj , > P3; Q3; R3; P3

J_W3XY =: cut2 J_W2

J_W3 =. (- (0, 1)) + L:0 J_W3XY

glpolygon adj , > J_W3

glshow ''

)

NB. revised version =====

NB. using FIG & JCH in array form

```
draw_xy =: 3 : 0
```

```
glrgb 0 0 0
```

```
glpen 1 0
```

```
gllines 100 500 900 500
```

```
gllines 500 100 500 900
```

```
)
```

```
color_pen4 =: 3 : 0
```

```
if. y. = 0 do. glrgb 255 0 0 end.
```

```
if. y. = 1 do. glrgb 255 255 0 end.
```

```
if. y. = 2 do. glrgb 0 255 0 end.
```

```
if. y. = 3 do. glrgb 0 0 255 end.
```

```
glbrush ''
```

```
glpen 4 0
```

```
)
```

```
wallpaper_run0_button=: 3 : 0
```

```
NB. start figure
```

```
FIG0 =: P; Q; R; P
```

```
JCH00 =. 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR NB. adjust size J
```

```
JCH0 =: <"(1) (32, 2)$JCH00
```

```
color_pen4 0
```

```
gllines adj , > FIG0
```

```
glpolygon adj , > JCH0
```

```
draw_xy ''
```

```
glshow ''
```

```
)
```

```
wallpaper_run1_button=: 3 : 0
```

NB. shift

S =. (Q + P) % 2

S0 =: (0, 0) - S

FIG1 =: S0 +L:0 FIG0

JCH1 =: S0 +L:0 JCH0

color_pen4 1

gllines adj , > FIG1

glpolygon adj , > JCH1

draw_xy ''

glshow ''

)

wallpaper_run2_button=: 3 : 0

NB. rotate 60 degree

FIG2XY =. cut2 FIG1

FIG2X =. {."(1) > 180 rot L:0 FIG2XY

FIG2Y =. {:"(1) > 180 rot L:0 FIG2XY

FIG2 =: , |: FIG2X, :FIG2Y

JCH2XY =. cut2 JCH1

JCH2X =. {."(1) > 180 rot L:0 JCH2XY

JCH2Y =. {:"(1) > 180 rot L:0 JCH2XY

JCH2 =: , |: JCH2X, :JCH2Y

color_pen4 2

gllines adj , > FIG2

glpolygon adj , > JCH2

draw_xy ''

glshow ''

)

```
wallpaper_run3_button=: 3 : 0
```

```
NB. shift return
```

```
FIG3 =. (-S0) + L:0 FIG2
```

```
JCH3 =. (-S0) + L:0 JCH2
```

```
color_pen4 3
```

```
gllines adj , > FIG3
```

```
glpolygon adj , > JCH3
```

```
draw_xy ''
```

```
glshow ''
```

```
)
```

```
wallpaper_runPQ_button=: 3 : 0
```

```
FIG0 =: P; Q; R; P
```

```
JCH00 =. 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR NB. adjust size J
```

```
JCH0 =: <"(1) (32, 2)$JCH00
```

```
color_pen4 0
```

```
gllines adj , > FIG0
```

```
glpolygon adj , > JCH0
```

```
S =. (P + Q) % 2
```

```
S0 =: (0, 0) - S
```

```
FIG1 =: S0 +L:0 FIG0
```

```
JCH1 =: S0 +L:0 JCH0
```

```
FIG2XY =. cut2 FIG1
```

```
FIG2X =. {."(1) > 180 rot L:0 FIG2XY
```

```
FIG2Y =. {:"(1) > 180 rot L:0 FIG2XY
```

```
FIG2 =: , |: FIG2X, :FIG2Y
```

```
JCH2XY =. cut2 JCH1
```

```

JCH2X =. {."(1) > 180 rot L:0 JCH2XY
JCH2Y =. {:"(1) > 180 rot L:0 JCH2XY
JCH2 =: , |: JCH2X, :JCH2Y

```

```

FIG3 =. (-S0) + L:0 FIG2
JCH3 =. (-S0) + L:0 JCH2

```

```

color_pen4 3
gllines adj , > FIG3
glpolygon adj , > JCH3

```

```

draw_xy ''
glshow ''
)

```

```

wallpaper_runQR_button=: 3 : 0
FIG0 =: P; Q; R; P
JCH00 =. 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR NB. adjust size J
JCH0 =: <"(1) (32, 2)$JCH00

```

```

color_pen4 0

```

```

gllines adj , > FIG0
glpolygon adj , > JCH0

```

```

S =. (Q + R) % 2
S0 =: (0, 0) - S

```

```

FIG1 =: S0 +L:0 FIG0
JCH1 =: S0 +L:0 JCH0

```

```

FIG2XY =. cut2 FIG1
FIG2X =. {."(1) > 180 rot L:0 FIG2XY
FIG2Y =. {:"(1) > 180 rot L:0 FIG2XY
FIG2 =: , |: FIG2X, :FIG2Y

```

```

JCH2XY =. cut2 JCH1
  JCH2X =. {."(1) > 180 rot L:0 JCH2XY
  JCH2Y =. {:"(1) > 180 rot L:0 JCH2XY
JCH2 =: , |: JCH2X, :JCH2Y

FIG3 =. (-S0) + L:0 FIG2
JCH3 =. (-S0) + L:0 JCH2

color_pen4 2
gllines adj , > FIG3
glpolygon adj , > JCH3

draw_xy ''
glshow ''
)

wallpaper_runRP_button=: 3 : 0
FIG0 =: P; Q; R; P
JCH00 =. 0.5 * , (0.5, 0) + "(1) (32, 2)$JCHAR NB. adjust size J
JCH0 =: <"(1) (32, 2)$JCH00

color_pen4 0

gllines adj , > FIG0
glpolygon adj , > JCH0

S =. (R + P) % 2
S0 =: (0, 0) - S

FIG1 =: S0 +L:0 FIG0
JCH1 =: S0 +L:0 JCH0

FIG2XY =. cut2 FIG1
  FIG2X =. {."(1) > 180 rot L:0 FIG2XY
  FIG2Y =. {:"(1) > 180 rot L:0 FIG2XY
FIG2 =: , |: FIG2X, :FIG2Y

```



```
JCH2XY =. cut2 JCH1
JCH2X =. {."(1) > 180 rot L:0 JCH2XY
JCH2Y =. {:"(1) > 180 rot L:0 JCH2XY
JCH2 =: , |: JCH2X, :JCH2Y
```

```
FIG3 =. (-S0) + L:0 FIG2
JCH3 =. (-S0) + L:0 JCH2
```

```
color_pen4 1
gllines adj , > FIG3
glpolygon adj , > JCH3
```

```
draw_xy ''
glshow ''
)
```