

# 北斎の新形小紋帳から (J6 版)

SHIMURA Masato  
JCD02773@nifty.ne.jp

2012 年 12 月 10 日

## 目次

0.1	キャンバス	4
0.2	カラーパレット	5
1	小紋のテクニック (1) ポリゴン	6
1.1	井筒万字 (No.3) の最初のピースの座標	6
1.2	多数の小紋の描画-井筒万字 (No.3) を描く	7
1.3	六つ手万字 (No.17) をポリゴンで描く	11
2	小紋のテクニック (2)-色の変化	14
3	いろいろな小紋のポリゴン	16
3.1	賽形 (No.18)	16
3.2	万字つなぎ (No.19)	18
3.3	万字菱 (No.20)	19
4	小紋のテクニック-図形の線形変換	22
4.1	Homogeneous Coordinates	22
4.2	万字菱 (No.20) を傾ける	23
5	ポリゴンと直線で描く	26
5.1	ひょうれつ麻の葉 (No.9)	26
5.2	麻の葉つくし (No.16)	29
5.3	八つ手麻の葉 (No.01)	31
5.4	松川麻の葉 (No.10)	34
6	小紋のテクニック (4) 直線での色変化	37
7	直線の小紋	38
7.1	釣り四つ目 No15	39

7.2	行きつ戻りつ No13 . . . . .	40
7.3	桜割り No07 . . . . .	41
7.4	かくの七宝 No31 . . . . .	44
7.5	はやわり No67 . . . . .	45
7.6	金網亀甲 No.53 . . . . .	46
7.7	燭光の万字結び No. 22 . . . . .	48
7.8	菱の釣万字 No.26 . . . . .	50
7.9	破れ簞目 No.34 . . . . .	52
7.10	松皮 No61 . . . . .	53
7.11	碁盤筋交い No64 . . . . .	56
7.12	畳 No52 . . . . .	58
7.13	唐様せうじのくづし . . . . .	59
7.14	No79 説明なし . . . . .	60
7.15	早割り碁盤割手 . . . . .	62
7.16	三重格子 No. 58 . . . . .	63
7.17	ねじのふじたね No.41 . . . . .	65
7.18	釣稲妻 No.36 . . . . .	66
7.19	菜籠麻の葉 No.14 . . . . .	68
8	曲線の幾何学 (1)-花卉を描く . . . . .	69
8.1	輪違い麻の葉 No.08 . . . . .	70
9	小紋のテクニック ( 5 ) ベジエ曲線 . . . . .	72
9.1	ベジエ曲線 (マトリクスフォーム) . . . . .	73
9.2	Cubic Bezier Curve . . . . .	73
9.3	小紋の作図データの構成 . . . . .	80
9.4	Cubic 以外のポイント数のベジエ曲線 . . . . .	82
10	ベジエ曲線で描く . . . . .	88
10.1	松波 No.44 . . . . .	88
10.2	No43 七宝のねじ . . . . .	90
10.3	No33 道化もっこう . . . . .	91
10.4	行きつ戻りつ No.11 . . . . .	92
10.5	No.35 わらび . . . . .	92
10.6	結び蕨 No40 . . . . .	94
10.7	四つで蕨 No. 38 . . . . .	97
10.8	流れ万字 No.21 . . . . .	99
10.9	早割りの図 (乱れ縦割り)No.23 . . . . .	100
10.10	尾の上重ね No39 . . . . .	103
10.11	向神通網 No55 . . . . .	105
10.12	阿蘭陀組 (0) NO81 . . . . .	107

10.13	N0.54 捻り四つ手組 . . . . .	108
10.14	うけ葵 No37 . . . . .	110
10.15	文の字繋ぎ No.51 . . . . .	111
10.16	菱割り (らの字)No.59 . . . . .	113
10.17	瓦小口 No77 . . . . .	114
10.18	碁盤割 No.63 . . . . .	115
10.19	四つ手 No29 . . . . .	117
10.20	大井紋高麗型 No.48 . . . . .	118
10.21	藤種 No47 . . . . .	120
11	北斎にフォームを付ける . . . . .	122
11.1	新型小紋帳の版木発見 (1986) . . . . .	122
11.2	デザイナー北斎 . . . . .	122
11.3	フェノロサとビゲロー . . . . .	123
11.4	新形小紋帳にフォームを付ける . . . . .	124
11.5	ラジオボタンを置く . . . . .	125
11.6	コンボドロップを作成する . . . . .	127
11.7	ボタン ( 1 ) . . . . .	127
11.8	ボタン ( 2 ) . . . . .	128
11.9	カラー . . . . .	129
11.10	描画する小紋の数 (縦横) を決める . . . . .	131
11.11	DRAW ボタン (ボタン 3) . . . . .	131
付録 A	J の入手とインストール . . . . .	133
付録 B	北斎の小紋の動かし方 . . . . .	133

## 北斎の幾何デザイン

葛飾北斎は60歳過ぎに「新形小紋帳」という柳亭種彦の序文で始まる60ページ余の小刷を残しており(1824)、各ページの2個ずつの円内に様々な(江戸)小紋が描かれている。種彦が「先生の筆頭より出でて古今にも聞かざる紋をなす」としたためた北斎による100余りの幾何学模様の新形である。

画狂人北斎は1760年に両国本所の大江戸博物館の地に生まれ、1849年永逝した。墓所は元浅草の誓教寺。富岳36景は70代から描き始めた。天保の変革(1842)で種彦は亡くなり北斎も暫し江戸を離れた。

北斎の井筒万字と六つ手万字でポリゴン(多角形)を用いた小紋の基本テクニックを整理する。

### 0.1 キャンバス

- キャンバスはC.Reiterのdwinを用いる。最大は1000×1000 Reiterのaddon *dwin2.ijs*のうち必要箇所は *hokusai\_tool.ijs* に収録したので *Addon* を読み込む必要は無い。
- *dwin* は始点は左下で、終点は右上。(J6は左上が始点)
- サイズは指定でき、マイナスの値も用いることができるがデータにあわせて自動でセットはしてくれないので、サイズ指定のための $xy$ の最小値と最大値を求めるツールを作成した

```
find_maxmin 4 5 calc_each_poly (<IM),<IMPARAM
_3 0 21 21 NB. min(x,y) , max(x,y)
```

```
_3 0 21 21 dwin 'IZUTU-MANJI'
```

- これを組み込んで自動でポップアップするようにした。

```
_3 0 21 21 dwin 'IZUTU-MANJI'
```

- *dwin* は縦横比を自動調整しているので縦横比が大きい小紋のピースを連ねた画像ではピースの歪みが生じる。縦横比を補正してピースの形を保つような手法を *popup\_dwin* に組み込んだ。辺縁部はカットするのでサイズ指定は大きめにすると良い

```
popup_dwin 4 5 calc_each_poly (<IM),<IMPARAM
```

```
find_center=: 3 : 0
```

```
NB. adjust canvas to square
```

```
tmp=. _2<\ 2 0 3 1 {; |. find_maxmin0 y NB. left bottom --> right top
```

```
Center=. -:@; |@:-/@|. L:0 tmp NB. center of gravity
```

```
Wide=. >. -:@:+/ ; 1r4&*@:-/ L:0 tmp NB. band
```

```
tmp1=. _2<\ Wide (+ , -@-) L:0 Center NB. range
```

```
ind=. ;*@>./ L:0 tmp1 NB. check _ or +
```

```
ANS=. <''
```

```
for_ctr. i.2 do.
```

```
if. _1 = ctr{ind do. PK=. >./ ; ctr{tmp1
```

```
else. PK=. <./ ; ctr{tmp1 end.
```

```
ANS=. ANS,<PK
```

```
end.
|. ; 2# L:0 }.ANS
)
```

## 0.2 カラーパレット

RGB 各 256 階調で 1600 万色になる。

	R	G	B	R	G	B	
Red	255	0	0	255	0	255	Magenta
Green	0	255	0	255	255	0	Yellow
Blue	0	0	255	0	255	255	Cyan
Black	0	0	0	255	255	255	White

後年の北斎はプルシャンプルーなどの輸入した合成絵具を用いて明るい藍色を水や空に用いている。

光を重ねた色と絵の具を重ねた色には反対の性質がある。手許のカラーブックには JIS 慣用色 269 が紹介されている。美術や印刷の世界の伝統色は趣があるようだ。日本の伝統色のカラーテーブルを紹介した本もいろいろあるので参照されたい。

J のパッケージにも 16 色の色指定の他に *colortab.ijs* (150 色) と *xwin.ijs* (500 色) が含まれており、読み込めば RGB でなくカラーネームで使うことができる。

Color	R	G	B	Color	R	G	B	Color	R	G	B
赤	190	0	50	緑	0	182	110	青	0	106	182
rose	219	53	97	green	0	154	87	blue	0	111	171
韓紅	230	75	107	常盤色	0	123	80	藍色	43	75	101
紅色	190	0	63	緑青色	77	129	105	瑠璃色	0	81	154
珊瑚色	255	127	143	emerald	0	164	116	marineblue	0	82	107
臙脂色	173	49	64	若竹色	0	163	126	navyblue	52	61	85
茜色	158	34	54	萌黄色	0	83	62	群青色	56	77	152
winered	128	39	63	若葉色	169	192	135	紺藍	53	53	115
scarlet	222	56	56	若草色	170	179	0	濃藍	34	53	70
chinesered	253	90	42					orientalblue	48	66	133
山吹色	248	169	0					江戸紫	97	72	118
橙/orange	239	129	15								
blond	222	178	95								
蜜柑色	235	132	0								

## 1 小紋のテクニック (1) ポリゴン

*Adobe Illustrator* などは持ち合わせていないので座標は手で取った。次のような方法もあるが、なかなかピタリとは決まらない。手始めにポリゴンで描く図形を集めた。

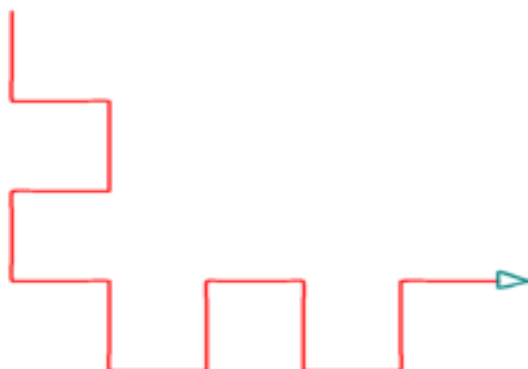
- 方眼紙にフリーハンドで描き、順に右または左回りに座標を取っていく。
- 小紋をスキャンして *eps* で保存し、*Tex* で任意の大きさに拡大し、1 ミリ方眼紙に印刷する。*jpg* でワープロに貼り付けてもよい。

北斎の小紋帳に整理のため最初の八つ手麻の葉を *No.01* として 82 番まで小紋帳の組版の順に付けた。最後の 15 ほどは欄間やオランダ欄間風が多く見られる

### 1.1 井筒万字 (No.3) の最初のピースの座標

### 1.1.1 タートルグラフィックス

井筒万字の最初の部分のデータをタートルグラフィックスで描いてみる。始点は左上とする



```
require 'turtle'

R180=: rt 180 fd 1

R90=:  rt 90 fd 1

L90=:  lt 90 fd 1

show (R180,L90,R90,R90,L90,L90,R90,L90,L90,R90,R90,L90,L90)

save '/temp/logo_izutu'
```

タートルグラフィックスは多角形を簡単に描ける。色を管理するパーツが多少弱いので本格的なカラー描画には推奨しない

### 1.1.2 ポリゴンの座標をとる

1. 一つのピースを方眼紙に描く
2. 始点を決める (任意の点)。これが原点になる。
3. ピースの左と下に  $x, y$  軸を記入する
4. 始点から左又は右回りに順にピースの頂点の座標を取っていく (一筆書き)
5. ポリゴンの終点は始点の一個手前でよい。(自動で連結される)
6.  $J$  に入力するときはラミネート ( , : ) で縦型に。  $dpoly$  では見やすいようにポリゴンのデータは  $(x, y)$  を縦形とする。 $x, y$  をペアにして 2 個ずつカンマで連結し、最後の一個をラミネートで連結すると縦形に

なる。

NB. 井筒組の万字

NB. first piece

IM01=:0 4,0 3,1 3,1 2,0 2,0 1,1 1,:1 0

IM02=:2 0,2 1,3 1,3 0,4 0,4 1,:5 1

IM03=:5 2,4 2,4 3,5 3,5 4,4 4,:4 5

IM04=:3 5,3 4,2 4,2 5,1 5,:1 4

IM=: IM01,IM02,IM03,IM04

NB. center square

IM2=: 2 3,2 2,3 2,: 3 3

IM01,IM02

0 4

0 3

1 3

1 2

0 2

0 1

1 1

1 0

2 0

2 1

3 1

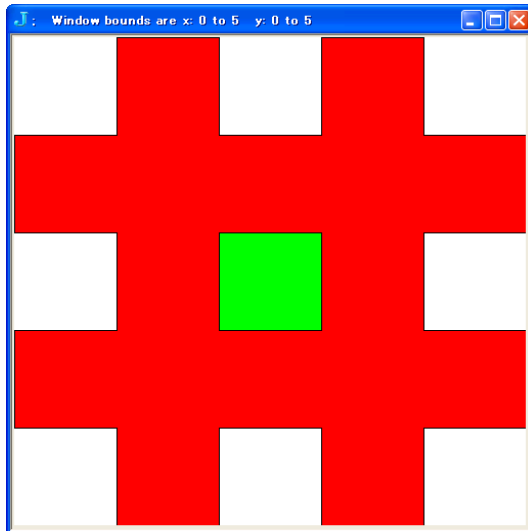
3 0

4 0

4 1

5 1

## 7. 最初のピース。2種類のポリゴンによる寄せ木細工またはジグソーパズル



255 0 0 draw\_dpoly IM

0 255 0 draw\_dpoly IM2

NB. 井筒組の万字

### 1.2 多数の小紋の描画-井筒万字 (No.3) を描く

同時描画 各ピースの座標をボックスで与え *dpoly* を (*L:0*) で用いるとループを用いなくて簡潔な手法で同時に描画できる。

\*1

井筒万字のポリゴン 図形の始点 (左上 (0 4)) から外延のポイントを左回りに (*x,y*) を筆順になぞる。中の小さ

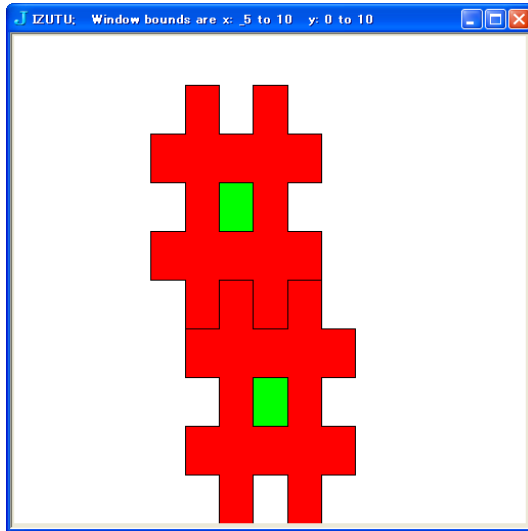
\*1 J はまだマルチ CPU での処理をサポートしていないので並列処理ではないが面倒なループを使用しない簡潔な同時処理ができる。

な正方形は別のポリゴンで描く

ボックスを用いて一度に描画する（同時処理）

組み木 上に一個積む (基準ポイント (0 4) からの移動量 (-1 4) を加える)。

(-1,4) は上の井筒の始点のが左へ 1、上へ 4 上がったことを示す。これは方眼紙に図形を書いて値を得る。



```
_5 0 20 20 dwin 'IZUTU'
```

```
255 0 0 dpoly L:0 IM + "1 L:0 (0 0;_1 4)
```

```
0 255 0 dpoly L:0 IM2 + "1 L:0 (0 0;_1 4)
```

寄せ木（又はタイル）の差分表 ボックスを用いて一度に計算するためのポイント表と差分表を作成する。  
方眼紙に幾つか寄せ木を描いて確認する。マスターピースは左下の (0,4) である。

小紋のフレーム

_3	16	1	17	5	18	9	19
_2	12	2	13	6	14	10	15
_1	8	3	9	7	10	11	11
0	4	4	5	8	6	12	7

差分表

IM\_16

_3	12	1	13	5	14	9	15
_2	8	2	9	6	10	10	11
_1	4	3	5	7	6	11	7
0	0	4	1	8	2	12	3

左下がマスターピースで上と右に貼り付けていく。

原点は (0,4)

差分表はフレーム表から原点 (0,4) を差し引く。

小紋の差分パラメーターの定義 フレーム表や差分表をいちいち作るのは煩瑣なので差分パラメーターから生成する。

最初のピースの始点 ; X 軸方向の xy の差分; Y 軸方向の xy の差分。これは統一様式とした

IMPARAM=: 0 4;4 1;\_1 4

IM2PARAM=: 2 3;4 1;0 4

y 軸	_1 4
	↑
	0 0 ⇒ 4 1
	始点 X 軸

y 軸	_1 4
	↑
	2 3 ⇒ 4 1
	始点 X 軸

差分表の自動生成 差分パラメーターと小紋の数から差分表を生成する

- フレーム表の自動生成



```

4 4 mk_diff_sub0 IMPARAM
+-----+-----+-----+-----+
|_3 12|1 13|5 14|9 15 |
+-----+-----+-----+-----+
|_2 8 |2 9 |6 10|10 11|
+-----+-----+-----+-----+
|_1 4 |3 5 |7 6 |11 7 |
+-----+-----+-----+-----+
|0 0 |4 1 |8 2 |12 3 |
+-----+-----+-----+-----+

mk_diff_sub0=: 4 : 0
'size_raw size_column'=. x
'base dfx dfy' =. y
X0=.|. { base +"1 (|. i.size_column) */ dfx
tmp=. |: >{ L:0 X0 +"1 L:0(|.i.size_raw) */ dfy
tmp - L:0 base
)

```

小紋の各ピースので座標の計算 マスターピースと差分表から小紋の各ピースの座標を一度に計算する

```

4 5 calc_each_poly (<IM),<IMPARAM

calc_each_poly=: 4 : 0
NB. Usage: 4 5 calc_each_poly (<MC),<MCPARAM
NB. x is size-of-matrix
NB. y is (<piece),< parameter
'Piece Parameter'=. y
Piece + ("1) L:0 x mk_diff_sub1 Parameter
)

```

dwin での最小値と最大値 dwin に与える最小値と最大値の目安を取得し dwin に渡す

```

find_maxmin 4 5 calc_each_poly (<IM),<IMPARAM
_3 0 21 21

(0 0 255;4 5) draw_dpoly (<IM),<IMPARAM
(0 255 0 ;4 5) draw_dpoly_over (<IM2),<IM2PARAM

```

自動で小紋を描く 重ね描き用も作成する

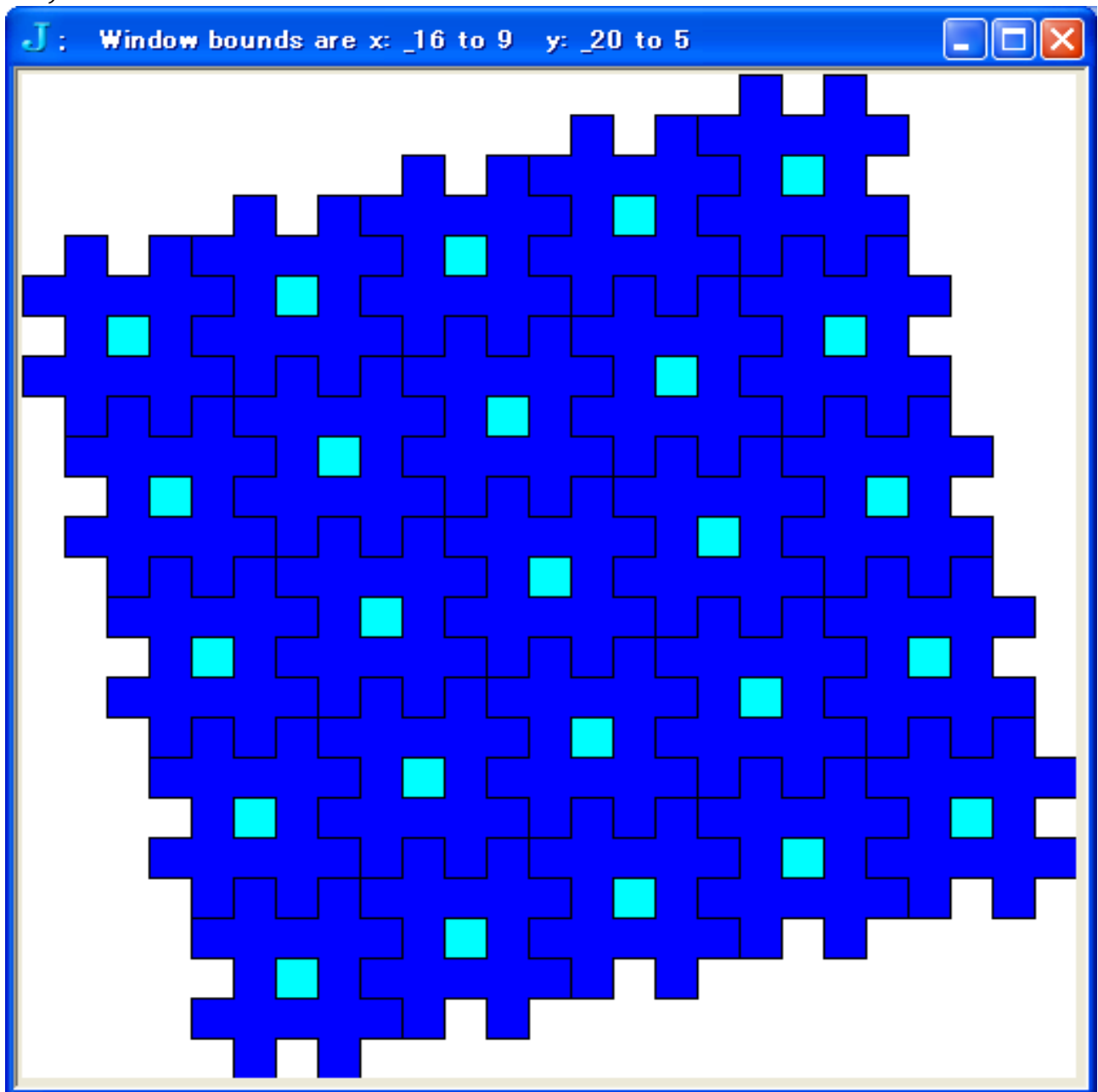
- *draw\_dpoly*

```

draw_dpoly=: 4 : 0
NB. x is color;size_of_matrix /raw & column(ex. 4 5)
NB. y is (<piece_data) , < diff_paramemter
NB. (255 0 255;4 5) draw_dpoly (<MC),<MCPARAM
'Color Size'=. x
tmp=. Size calc_each_poly y
popup_dwin tmp
Color dpoly L:0 tmp
)

```

- *draw\_dpoly\_over*  
draw\_dpoly\_over=: 4 : 0  
NB. for overdrawing on shape/  
NB. x y is same  
'Color Size'=. x  
tmp=. Size calc\_each\_poly y  
Color dpoly L:0 tmp  
)



(246 191 0;244 213 0;92 84 36) hokusai\_im 10 10

一度に描く 色とサイズを指定して一度に描く手順のスクリプトを作成した

```
hokusai_im=: 4 : 0
NB. (246 191 0;244 213 0 ;92 84 36) hokusai_im 6 7
NB. Color 0/1 graduation Color 2 is Center square
'Color0 Color1 Color2'=. x
Size=. y
((Color0;Color1);<Size) draw_dpoly_grad (<IM),<IMPARAM
(Color2;Size) draw_dpoly_over (<IM2),<IM2PARAM
)
```

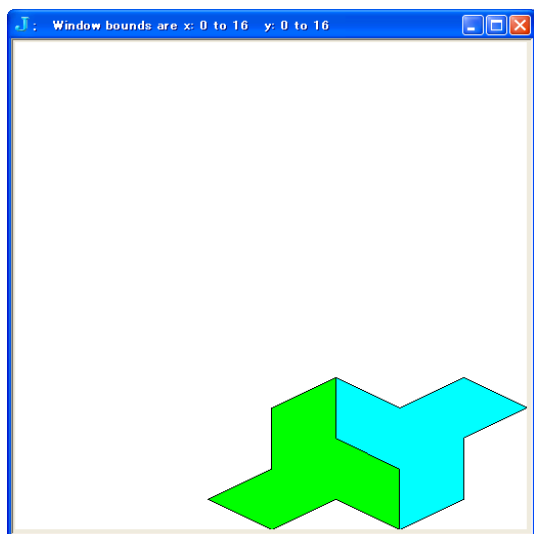
### 1.3 六つ手万字 (No.17) をポリゴンで描く

六つ手万字は  $[A,B]2$  個のポリゴンを組み合わせる。回転と移動でもできるが2個作成したほうが手早い

最初のピースの座標 .

	MM0 ,. MM1
	10 5   10 5
	8 4   10 3
NB. 六つ手万字	8 2   12 2
MM0=:10 5, 8 4, 8 2, 6 1, 8 0, 10 1, 12 0, 12 2, :10 3	6 1   12 0
MM1=: 10 5,10 3,12 2,12 0,14 1,14 3,16 4,14 5,:12	8 0   14 1
4	10 1   14 3
原点 ( 始点 ) は共用している	12 0   16 4
	12 2   14 5
	10 3   12 4

最初のピースを描く .



```
0 0 16 16 dwin ''
0 255 0 dpoly MM0
0 255 255 dpoly MM1
```

ピースはこれを 90 度回転させたものと 2 組 (A,B) で構成されているが B は白抜きとすることもできる。

組み木用テーブル 図に方眼を重ね、フレームのため上と右の座標を求める。

4	20	10	19	16	18
6	15	12	14	18	13
8	10	14	9	20	8
<b>10</b>	<b>5</b>	16	4	22	3

8	10	
<b>10</b>	<b>5</b>	16 4

始点と差分 方眼紙に幾つかの絵を描くか、原画を眺めて原点の始点と各ピースの始点の差を読み取るここでは原点は A,B 共用である。

$$\begin{array}{c} 2 \ 5 \\ \uparrow \\ 10 \ 5 \Rightarrow 6 \ 1 \end{array}$$

これは A,B 一緒である。

差分パラメーターを作成する .

MM0PARAM=: 10 5;6 \_1;\_2 5 NB. for automatic

MM1PARAM=: 10 5;6 \_1;\_2 5 NB. same MM0PARAM

小紋用の差分 小紋の各ピース用の差分を A,B について計算する

4 5 calc\_each\_poly (<MM0),<MM0PARAM

4 5 calc\_each\_poly (<MM1),<MM0PARAM

dpoly の最小値、最大値 dwin に与える最小値と最大値の目安を得る。(自動生成は最初の A で行う)

```
find_maxmin 4 5 calc_each_poly (<MM0),<MM0PARAM
0 _4 36 20
```

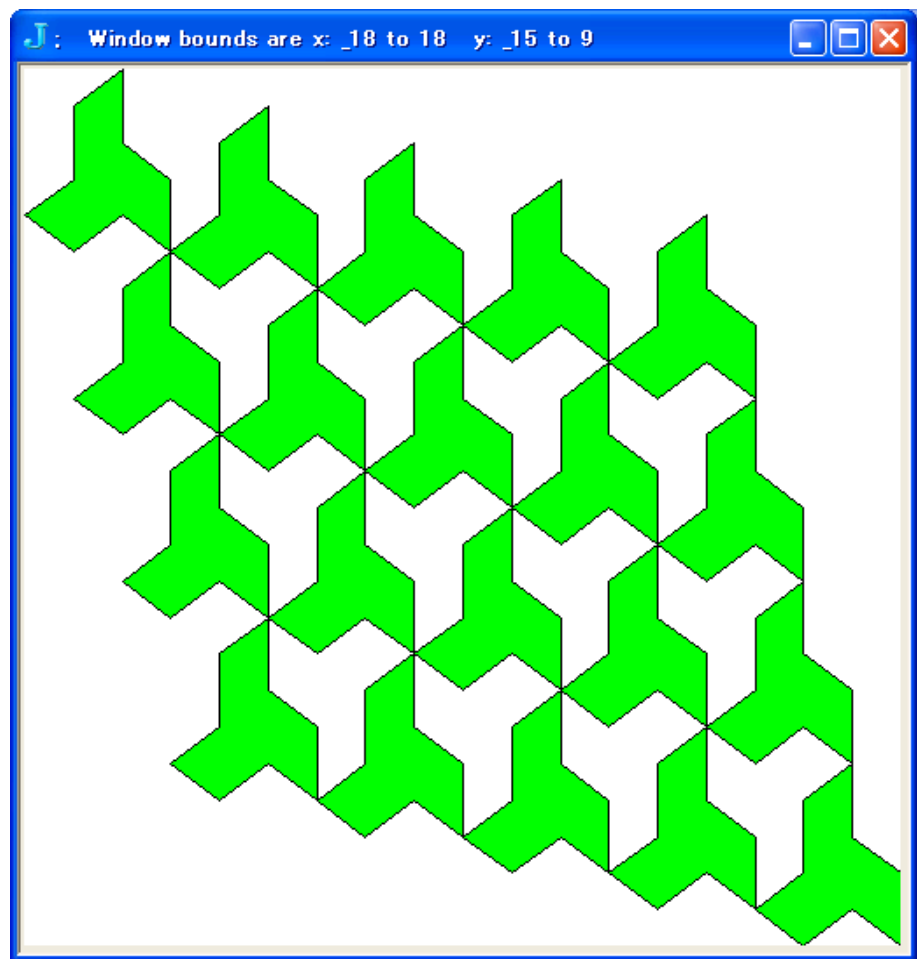
同時描画 .

白抜き

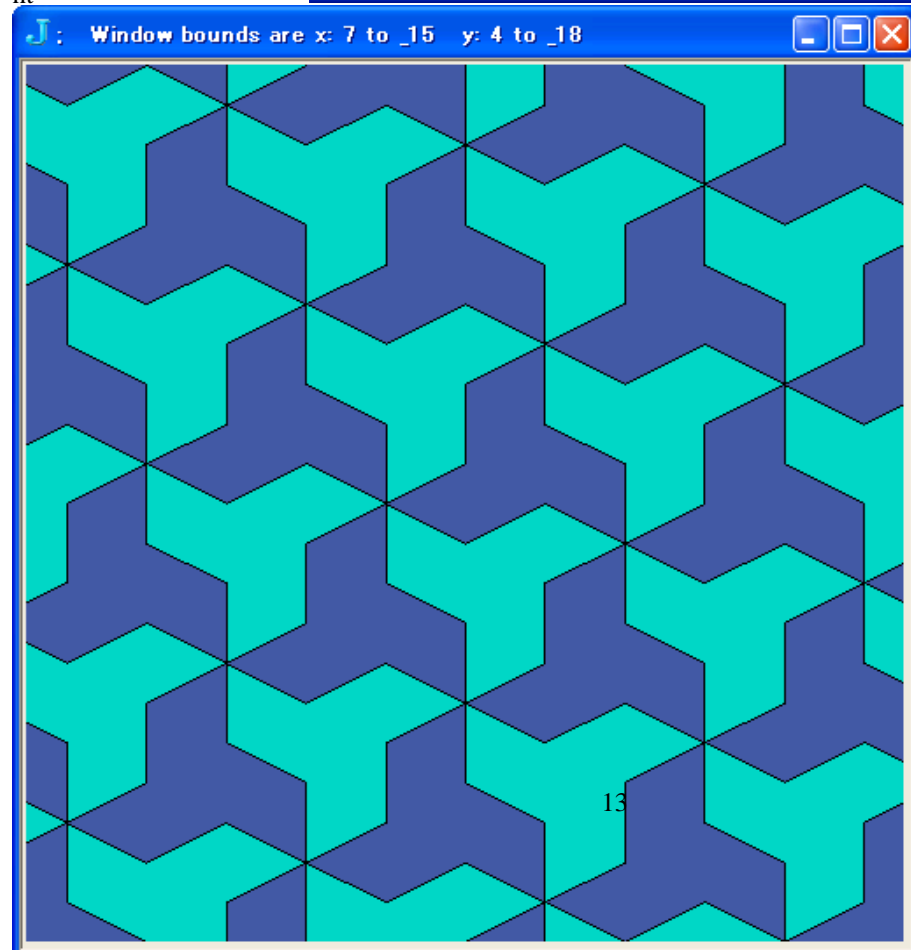
(0 0 255 ;4 5 )draw\_dpoly (<MM0),<MM0PARAM NB. A 単独では白抜き

2 色

(67 71 173; 67 90 160) hokusai\_mm 6 7



ht



## 2 小紋のテクニーク (2)-色の変化

六つ手万字のように A,B2 つの図形がある場合は色による変化を付けやすいが、井筒万字のような主体部分が 1 種類の図形は同じ色だとのっぺりとするので色で変化を付けたい。

0	1	2	3		0	1	2	3	4
4	5	6	7		5	6	7	8	9
8	9	10	11		10	11	12	13	14
12	13	14	15		15	16	17	18	19
16	17	18	19		20	21	22	23	24

ここでは小紋の個数は縦横の大きさを任意に取れるようにしている。よく眺めると赤と黒は右上、左下の斜めの線に沿って並んでいる。これは *J* の *Oblique* 機能によって斜め線に分類できる。

```
</. i. 5 4
+-----+-----+-----+-----+-----+
|0|1 4|2 5 8|3 6 9 12|7 10 13 16|11 14 17|15 18|19|
+-----+-----+-----+-----+-----+

</. i. 5 5
+-----+-----+-----+-----+-----+
|0|1 5|2 6 10|3 7 11 15|4 8 12 16 20|9 13 17 21|14 18 22|19 23|24|
+-----+-----+-----+-----+-----+
```

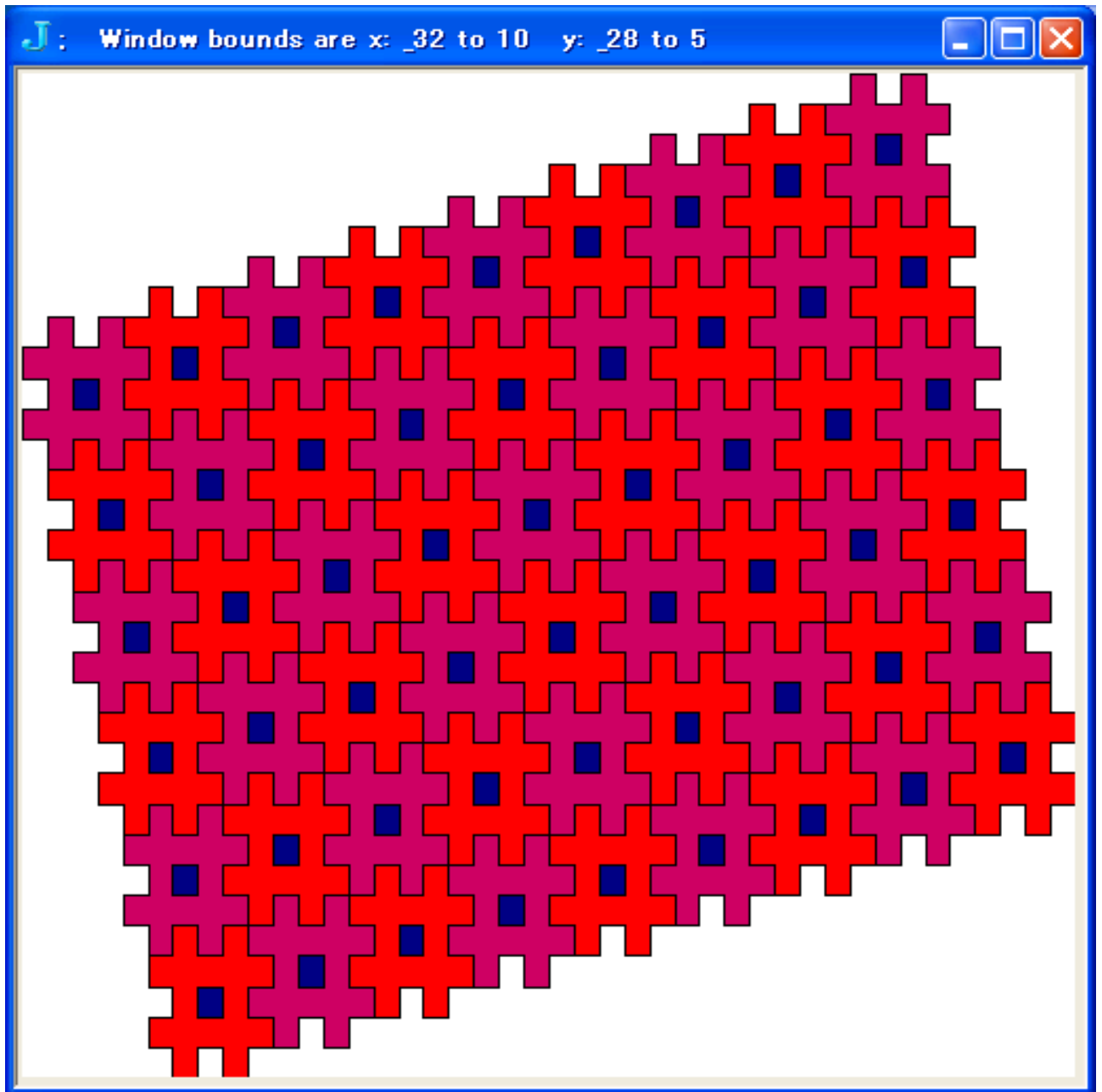
これを交互に紅白に分類し、この数値を指標として差分計算済みのデータを区分して描画関数に渡せばよい。次の簡潔なスクリプトで分離できる

```
index_separate=: 3 : 0
NB. for graduation 2 colors
NB. index_separate 3 5
ind=. i. # Oblic=. </. i. y NB. using oblique
tmp0=. ;(-.2|ind)# Oblic
tmp1=. ;(2|ind)#Oblic
tmp0;tmp1
)

index_separate 5 5
+-----+-----+-----+-----+-----+
|0 2 6 10 4 8 12 16 20 14 18 22 24|1 5 3 7 11 15 9 13 17 21 19 23|
+-----+-----+-----+-----+-----+
```

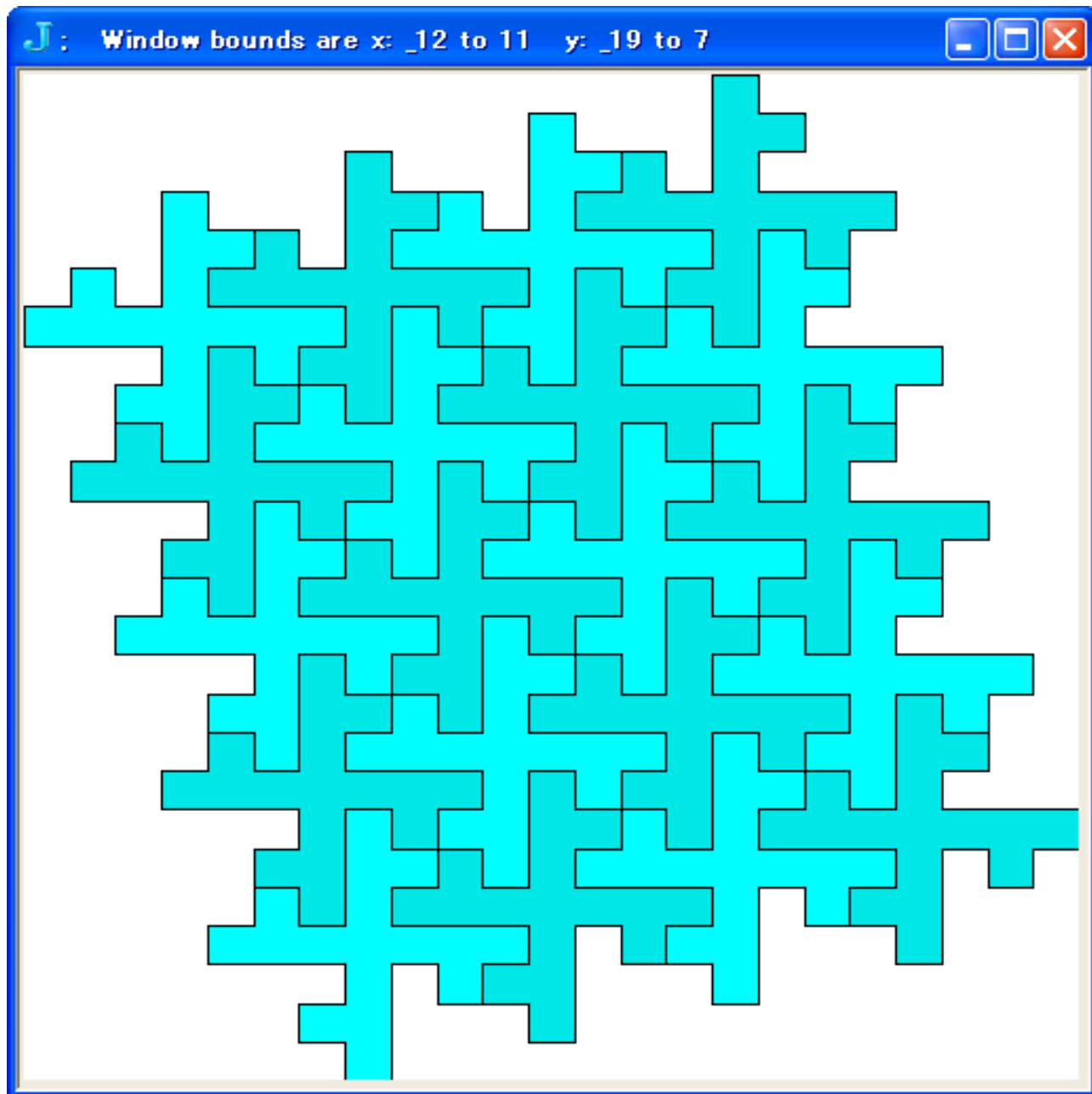
一連の作業を行うため *draw\_dpoly\_grad* を作成した

```
(( 0 0 255 ; 0 0 240);< 4 5) draw_dpoly_grad (<IM),<IMPARAM  
( 0 200 0;4 5) draw_dpoly_over (<IM2),<IM2PARAM
```



### 3 いろいろな小紋のポリゴン

#### 3.1 賽形 (No.18)



最初のピースの入力 .

NB. xx saigata

S0=: 0 3,3 3,3 2,2 2,2 1,3 1,3 0,: 4 0

S1=: 4 3,5 3,5 2,6 2,6 3,7 3,:7 4

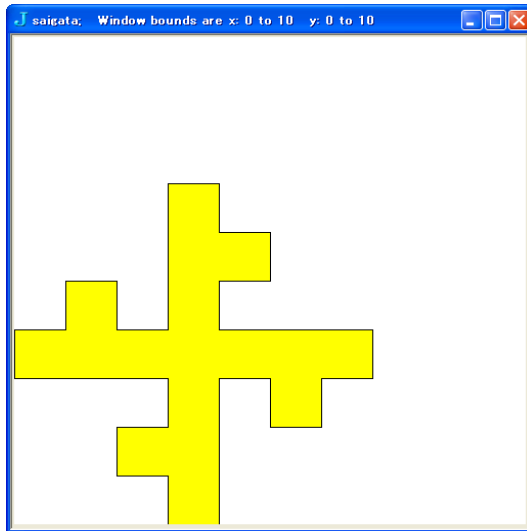


```

S2=: 4 4,4 5,5 5,5 6,4 6,4 7,:3 7
S3=: 3 4,2 4,2 5,1 5,1 4,:0 4
S=: S0,S1,S2,S3

```

最初のピース .



```
255 255 0 draw_dpoly0 S
```

ピースの差分テーブル .

小紋のフレーム

```

4 5 mk_diff_sub0 SPARAM
+---+---+---+---+---+
|12 6|11 10|10 14|9 18 |8 22 |
+---+---+---+---+---+
|8 5 |7 9 |6 13 |5 17 |4 21 |
+---+---+---+---+---+
|4 4 |3 8 |2 12 |1 16 |0 20 |
+---+---+---+---+---+
|0 3 |_1 7 |_2 11|_3 15|_4 19|
+---+---+---+---+---+

```

```

_1 7
  ↑
0 3 ⇒ 4 4

```

```
SPARAM=: 0 3;_1 4;4 1
```

draw\_dpoly\_grad .

```

hokusai_s=: 4 : 0
NB. Usage:(67 71 173; 67 90 160) hokusai_s 6 7
'Color0 Color1'=. x
Size=. y
((Color0;Color1);<Size) draw_dpoly_grad (<S),<SPARAM
)

```

### 3.2 万字つなぎ (No.19)

[A,B]2 個の図形である。

最初のピース .

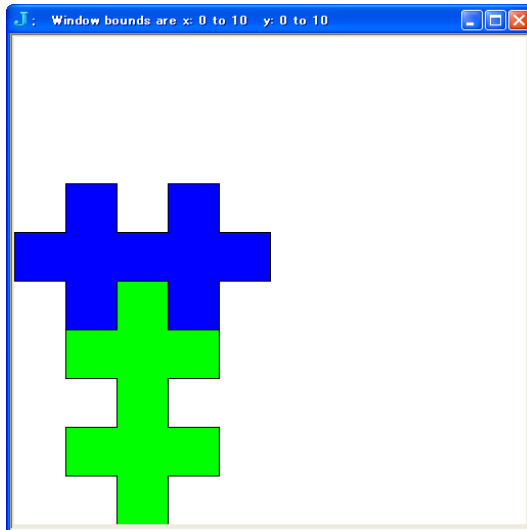
NB. Manji Tunagi

MT0=: 1 4,1 3,2 3,2 2,1 2,1 1,2 1,2 0,3 0,3 1,:4 1

MT0=:MT0,4 2,3 2,3 3,4 3,4 4,3 4,3 5,2 5,: 2 4

MT1=: 0 6,0 5,1 5,1 4,2 4,2 5,3 5,3 4,4 4,4 5,:5 5

MT1=: MT1,5 6,4 6,4 7,3 7,3 6,2 6,2 7,1 7,:1 6



0 0 10 10 dwin ''

0 255 0 dpoly MT0

0 0 255 dpoly MT1

差分パラメータ .

小紋フレーム

1 10  
↑  
1 4 ⇒ 4 1

差分

MTOPARAM=: 1 4;3 \_3; 0 6

0 6  
↑  
1 4 ⇒ 3 \_3

draw\_dpoly .

小紋フレーム

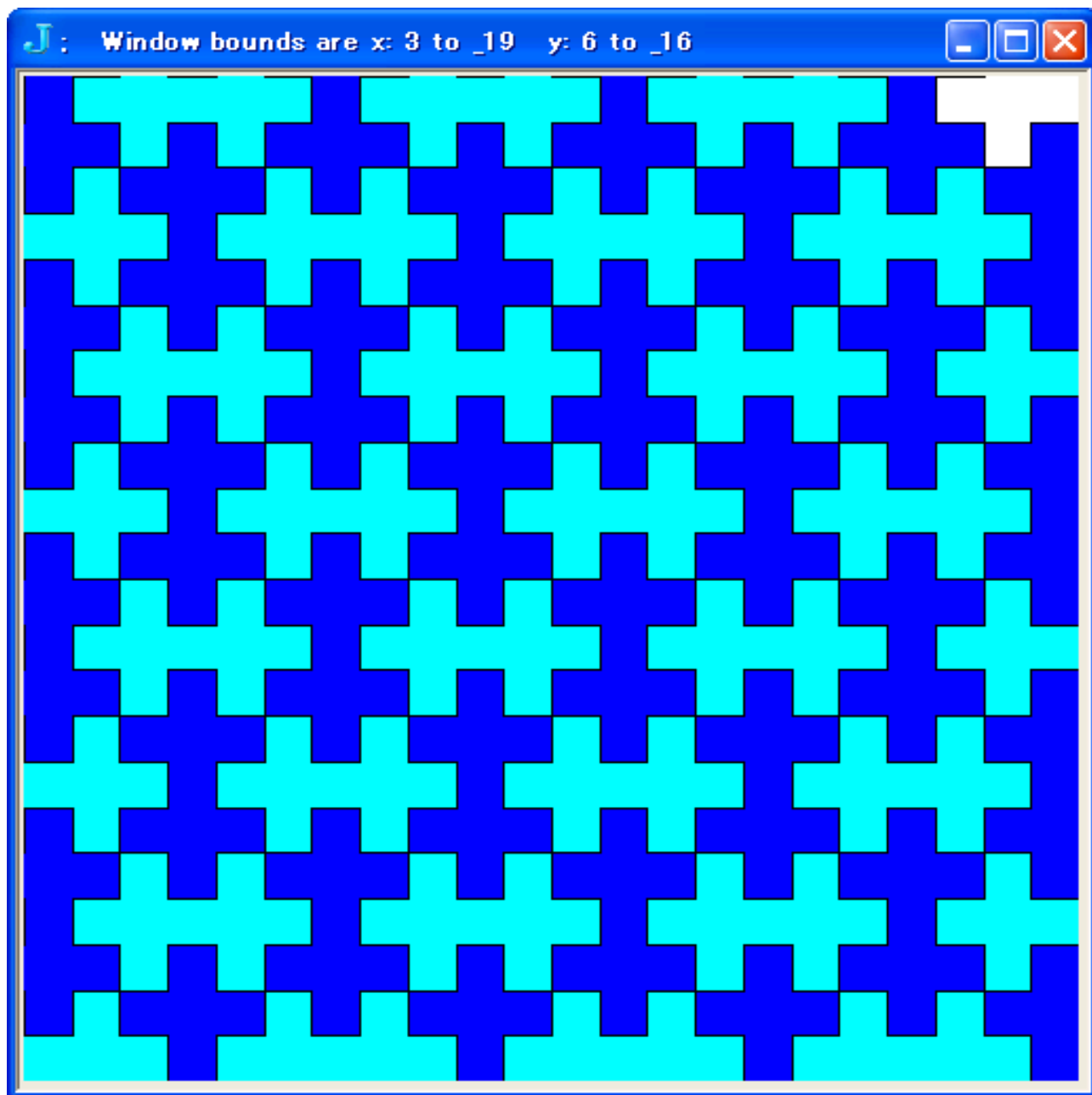
0 12  
↑  
0 6 ⇒ 3 3

差分

MTIPARAM=: 0 6;3 \_3;0 6

0 6  
↑  
0 6 ⇒ 6 0

(144 93 84;230 75 107) hokusai\_mt 6 7



### 3.3 万字菱 (No.20)

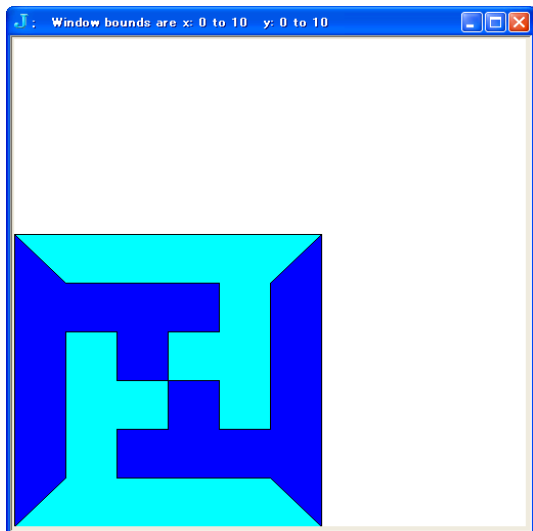
最初のピース .

4 個の万字で菱形を作っている。1 点でつながった 2 個のピースは一個のピースとして扱えるので  $[A,B]$  の 2 個として扱う。

```
MC=: 3 3,4 3,4 2,5 2,5 5,6 6,0 6,1 5,4 5,4 4,3 4,: 3 3
```

```
MC=: MC,3 3,3 2,2 2,2 1,5 1,6 0,0 0,1 1,1 4,2 4,2 3,:3 3
```

MCPARAM=: 3 3;6 0;0 6



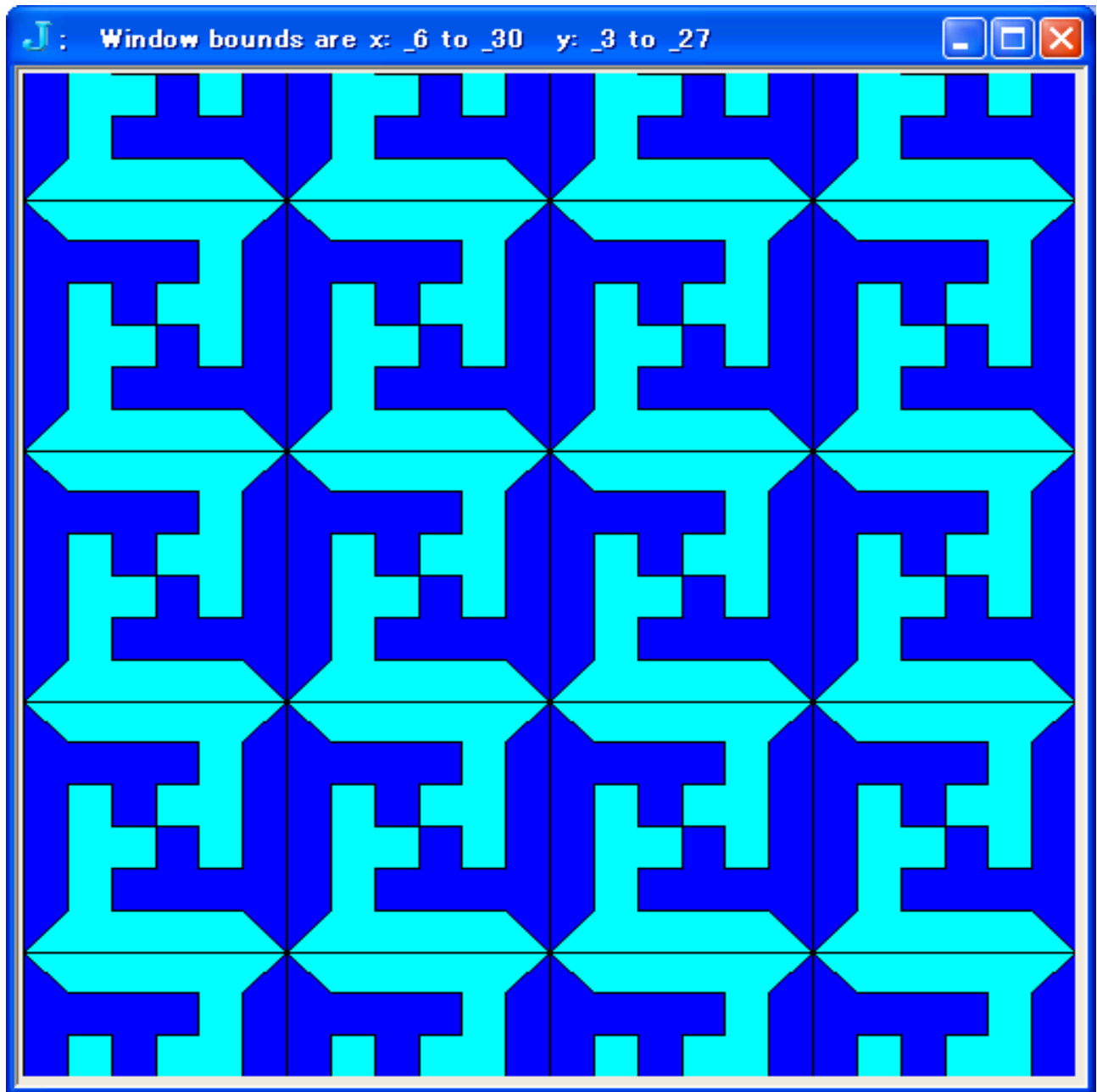
```
0 0 6 6 dwin 'manjihishi'
0 255 255 dpoly MC0
```

差分パラメータ .

MCPARAM=: 3 3;6 0;0 6

差分

```
0 6
↑
3 3 ⇒ 6 0
```



```

hokusai_mc=: 4 : 0
NB. (0 164 116 ;169 192 135) hokusai_mc 6 7
'Color0 Color1 '=. x
'Size'=. y
(Color0;Size) draw_dpoly (<MC0),<MC0PARAM
(Color1;Size) draw_dpoly_over (<MC1),<MC0PARAM
(0 0 0;Size) draw_dline_over (<MC0),<MC0PARAM
(0 0 0;Size) draw_dline_over (<MC1),<MC0PARAM

```

)

## 4 小紋のテクニク-図形の線形変換

### 4.1 Homogeneous Coordinates

図形の伸縮、回転、移動は次のように行える。3×3のマトリクスを用いる場合は *Homogeneous Coordinates* と呼ばれる。本来 3 次元の回転であるが z 軸を固定した場合に相当し 2 次元平面上で作用する。

1. 葉書を串で突き刺して回した場合に相当。

x を 1 にして展開すれば X 軸が、y が 1 では y 軸が、z に 1 を持ってくると z 軸が回転する。今回は 2 次元なので z を 1 にする。

$$\begin{pmatrix} x & . & . \\ . & y & . \\ . & . & z \end{pmatrix}$$

2. 図形データは (x,y) に z 軸分の 1 を加えて計算し、描画のときに削除する
3. 計算は内積演算でおこなう mp=: +/ . \*
4. ポリゴンやラインの x,y のデータには z 軸にあたる列を全て 1 として計算し、グラフィックスに渡す前に Z の列を落とす

Script 同時変換の Script

```
NB. rotm by C.Reiter
elongm=: 3 : '(y,1)*=i.3'

rotm=: (cos, sin,0:),(-@sin,cos,0:),: 0: ,0:,1:

transm=: 3 : '(=i.2), y,1'

mp=: +/ . * NB. inner products
```

回転 rotm 反時計回りに回転する。

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
rotm 1r4p1
0.707107 0.707107 0
_0.707107 0.707107 0
0 0 1
```

伸縮 elongm 横に 2 倍、縦に 3 倍に伸ばす。縮小はマイナスで

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
elongm 2 3
2 0 0
0 3 0
0 0 1
```

移動 transm  $x, y=(3\ 4)$  に移動

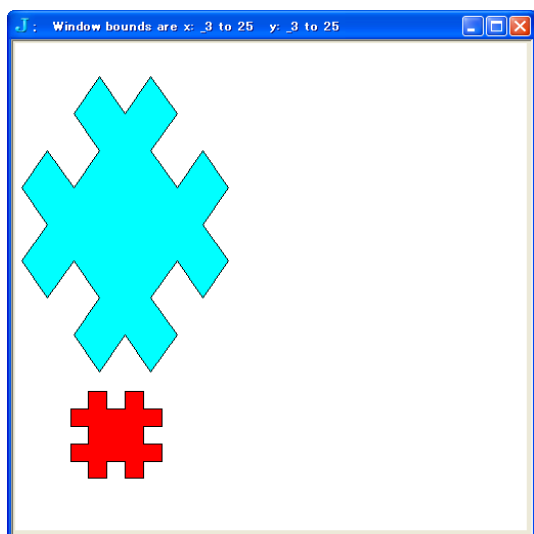
$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix}$$

```
elongm 2 3
1 0 0
0 1 0
3 4 1
```

全部まとめて *homogenous coordinate*

$$(x, y, 1)_{\text{new}} = (x, y, 1) \begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix}$$

```
a=. (rotm 1r4p1) mp (elongm 2 3) mp transm 3 4
1.41421 2.12132 0
_1.41421 2.12132 0
      3      4 1
```

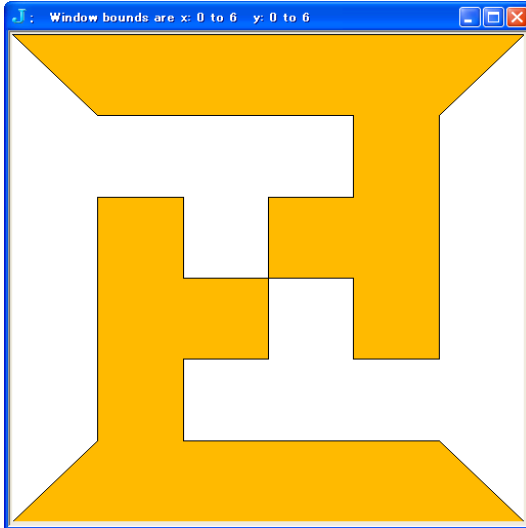


```
_3 _3 25 25 dwin ''
255 0 0 dpoly IM
0 255 255 dpoly a1=. }:"1 (IM,.1)mp a
```

## 4.2 万字菱 (No.20) を傾ける

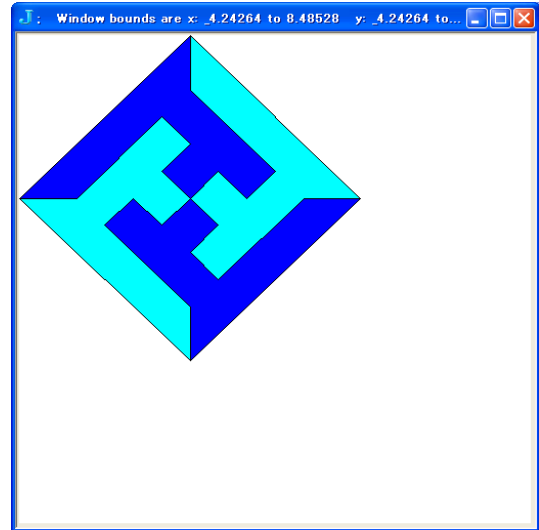
万字菱と回転 左回りに  $\frac{1}{4}\pi$  回転させる

```
255 187 0 draw_dpoly0 MC0
NB. 向日葵色
```



4 5 度回転 左回り (キャンパスの都合で小さく見えるが縮小はしていない)

```
0 0 255 draw_dpoly0 }:"1 (MC0,.1) mp rotm 1r4p1
0 255 255 dpoly }:"1 (MC1,.1) mp rotm 1r4p1
```



小紋のフレームを傾ける 小紋は直交フレームで作ってある。これを指定ラディアン分傾けたフレームを作る

#### 1. 直交フレーム

```
a=. 4 5 mk_diff_sub0 MC0PARAM
+-----+-----+-----+-----+-----+
| 3 21| 9 21| 15 21| 21 21| 27 21|
+-----+-----+-----+-----+-----+
| 3 15| 9 15| 15 15| 21 15| 27 15|
+-----+-----+-----+-----+-----+
| 3 9 | 9 9 | 15 9 | 21 9 | 27 9 |
+-----+-----+-----+-----+-----+
| 3 3 | 9 3 | 15 3 | 21 3 | 27 3 |
+-----+-----+-----+-----+-----+
```

#### 2. 直交フレームの各ベクトルをを回転行列で回転させ、左下の原点との差分を取る

```
(3 4;1r4p1) mk_diff_sub1_diagonal MC0PARAM
+-----+-----+-----+-----+-----+
| 8.48528 8.48528| 4.24264 12.7279 | 0 16.9706      | _4.24264 21.2132|
+-----+-----+-----+-----+-----+
| 4.24264 4.24264| 0 8.48528      | _4.24264 12.7279|_8.48528 16.9706|
+-----+-----+-----+-----+-----+
| 0 0          | _4.24264 4.24264|_8.48528 8.48528|_12.7279 12.7279|
```



+-----+-----+-----+-----+-----+

ピースを回転させる .

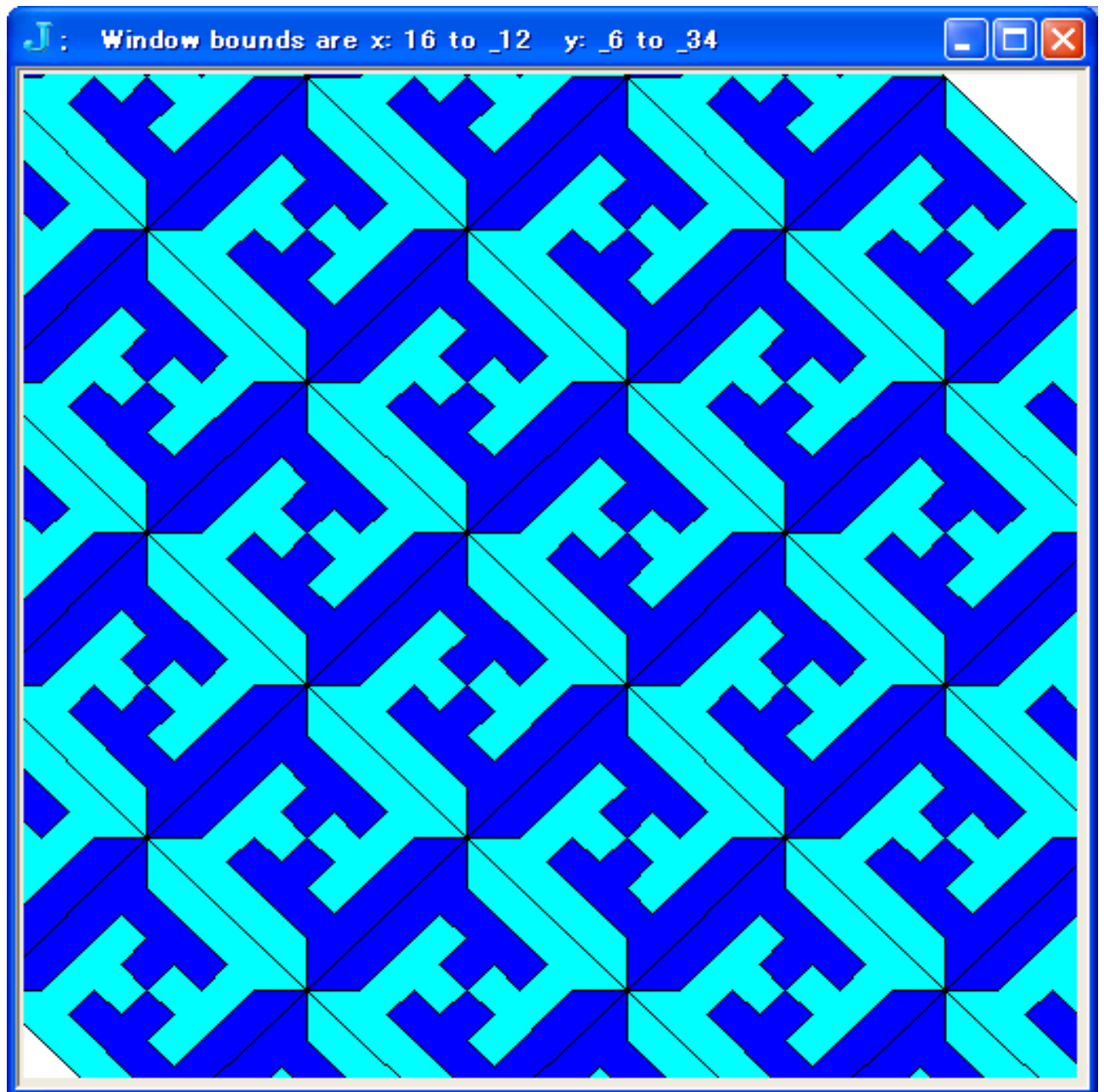
```
}:"1 ( MC0,.1) mp rotm 1r4p1
```

小紋の各座標を計算する 回転させたピースから差分座標を引く

```
(0 0 255;6 7;1r4p1) draw_dpoly_diagonal (<MC0),<MC0PARAM
```

```
(0 255 255;6 7;1r4p1) draw_dpoly_over_diagonal (<MC1),<MC0PARAM
```

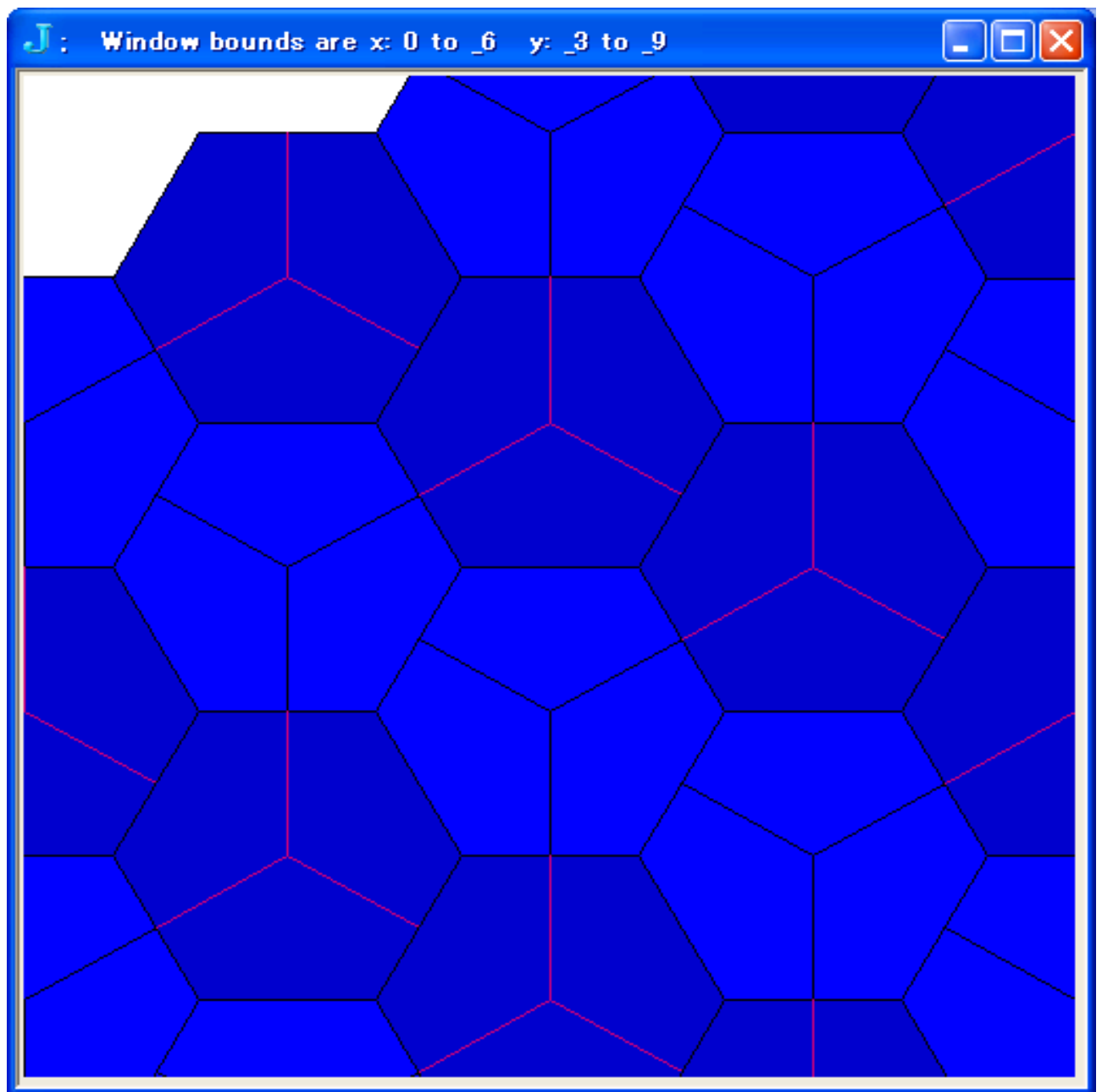
1r4p1 は  $\frac{1}{4}\pi$ ラジアン



## 5 ポリゴンと直線で描く

### 5.1 ひょうれつ麻の葉 (No.9)

```
((0 0 255;0 0 200 );<5 5) draw_dpoly_grad (<HA0),<HA0PARAM  
((0 0 0 ; 200 0 120);<5 5) ha_line_over ''
```



六角形 正多角形を描くには *J* のイディオムが便利である。正多角形の数字を変えれば何角形でも描ける

が、定規とコンパスと言う古典問題ではない。さらに、正多角形の各頂点の座標が得られるので便利である。

1. 6角形なので  $(i.6)\%6$  に  $2\pi$  をかける

```
2p1*(i.6)%6
0 1.0472 2.0944 3.14159 4.18879 5.23599
```

2. 局座標に変換する

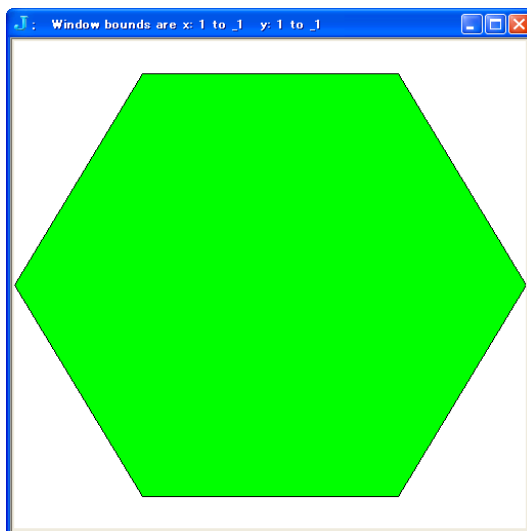
```
r. 2p1*(i.6)%6
1 0.5j0.866025 _0.5j0.866025 _1j1.22465e_16 _0.5j_0.866025 0.5j_0.866025
```

3. 複素数を実部虚部に分離する。

```
clean HA0=: +. r. 2p1*(i.6)%6 NB. hexagon
1      0
0.5 0.866025
_0.5 0.866025
_1      0
_0.5 _0.866025
0.5 _0.866025
```

始点は中央右

```
0 255 0 draw_dpoly0 HA0
```



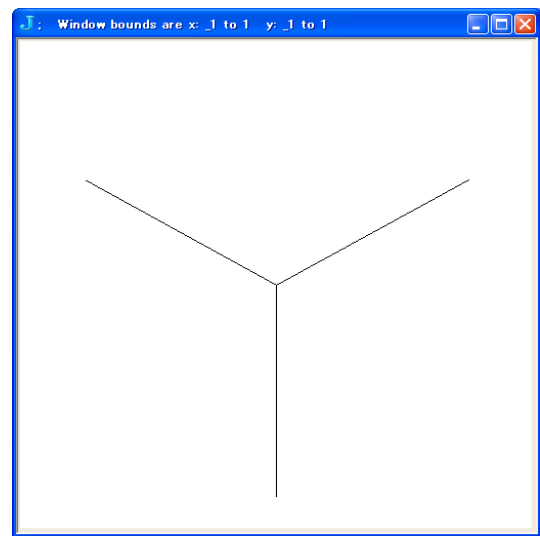
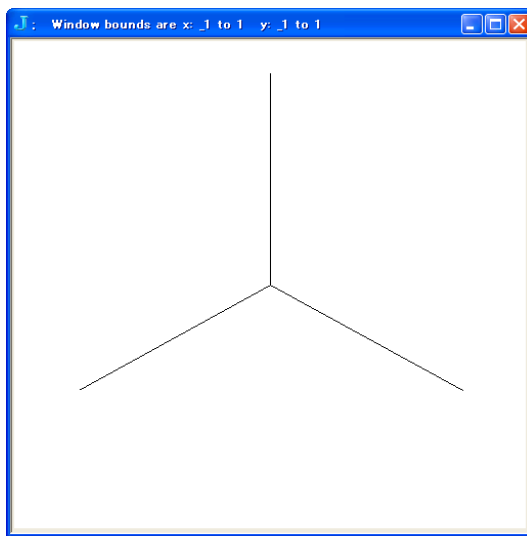
直線 直線は *dline* で描く。座標データの形式はポリゴンと同じであるが、ラインは終点から始点まで自動で

は連結してはくれない。

乱れ麻を直線で描く Y と逆 Y の 2 種類である

HAL0, HAL1 値は方眼の図から計算や読み取りで求めた

HAL0		HAL1	
0	0	0	0
0	0.866025	0	0.866025
0	0	0	0
0.75	0.43317	0.75	0.43317
0	0	0	0
0.75	0.43317	0.75	0.43317



始点は Y の中心である

差分 次の表から始点との差を採る.

$$\begin{array}{c|c} 1 & 1.7325 \\ \uparrow & \\ 1 & 0 \end{array} \Rightarrow \begin{array}{c|c} 2.5 & 0.866 \end{array}$$

HA0PARAM=:1 0 ; 1.5 0.866;0 1.732

$$\begin{array}{c|c} 1 & 1.7325 \\ \uparrow & \\ 0 & 0 \end{array} \Rightarrow \begin{array}{c|c} 1.5 & 0.866 \end{array}$$

HAL1PARAM=:HAL0PARAM=: 0 0 ; 1.5 0.86625;0 1.7325

麻の葉を描く 北斎の Y、逆 Y の組み合わせから規則性は分からなかったのがグラデュエーションのパーツを流用して手順を書き出した。Y、逆 Y はグラデュエーションと一致するが、乱数を入れるとずっと粋になるのではないかな

hokusai\_ha=: 4 : 0

NB. (144 93 84;230 75 107;0 0 0;144 93 84) hokusai\_ha 6 7

'Color0 Color1 Color2 Color3'=. x

Size=. y

tmp0=.Size calc\_each\_poly (<HAL0),<HAL0PARAM

tmp1=.Size calc\_each\_poly (<HAL1),<HAL0PARAM

```

'Ind0 Ind1'=. index_separate Size
'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp1
((Color0;Color1);<Size) draw_dpoly_grad (<HA0),<HA0PARAM
Color2 dline L:0 Gr0
Color3 dline L:0 Gr1
)

```

## 5.2 麻の葉つくし (No.16)

正六角形 前の六角形を流用するが、頂角がある。

```

_1 _1 1 1 dwin ''
0 0 255 dpoly AT

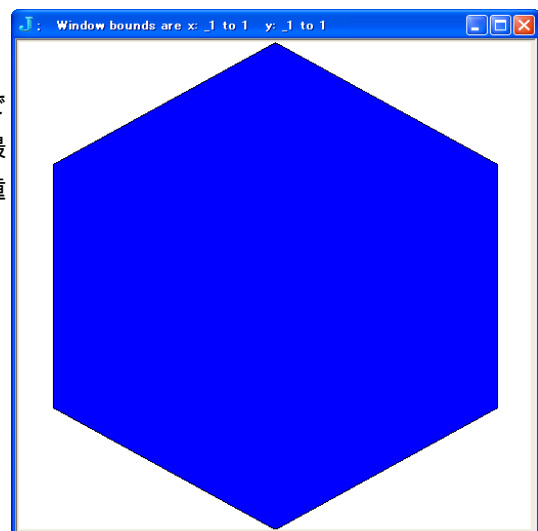
```

AT=: (4|.|.|. "1 HA0),|.1{HA0  
*dline* でも 6 角形を書くために始点と終点を繋いで  
 おこう。ポリゴンは図を重ねると重なった部分は最  
 後の図 (色) が表示される。絵の具のように塗り重  
 ねの痕跡は残らない

```

clean AT
0.866025 0.5
0 1
_0.866025 0.5
_0.866025 _0.5
0 _1
0.866025 _0.5
0.866025 0.5

```



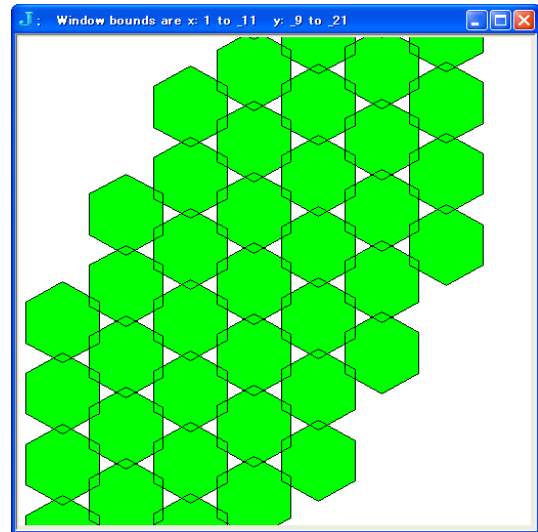
*dline* 塗り重ねて消えた輪郭線を再度描く

差分 図を多く描いて調整した

ATPARAM=: 0.866 0.5; 0 1.75; 1.5 2.65

差分パラメーターの構成

Y	軸		
1.5	2.65		
	↑		
0.866	0.5	⇒	0 1.75
始	点		X 軸

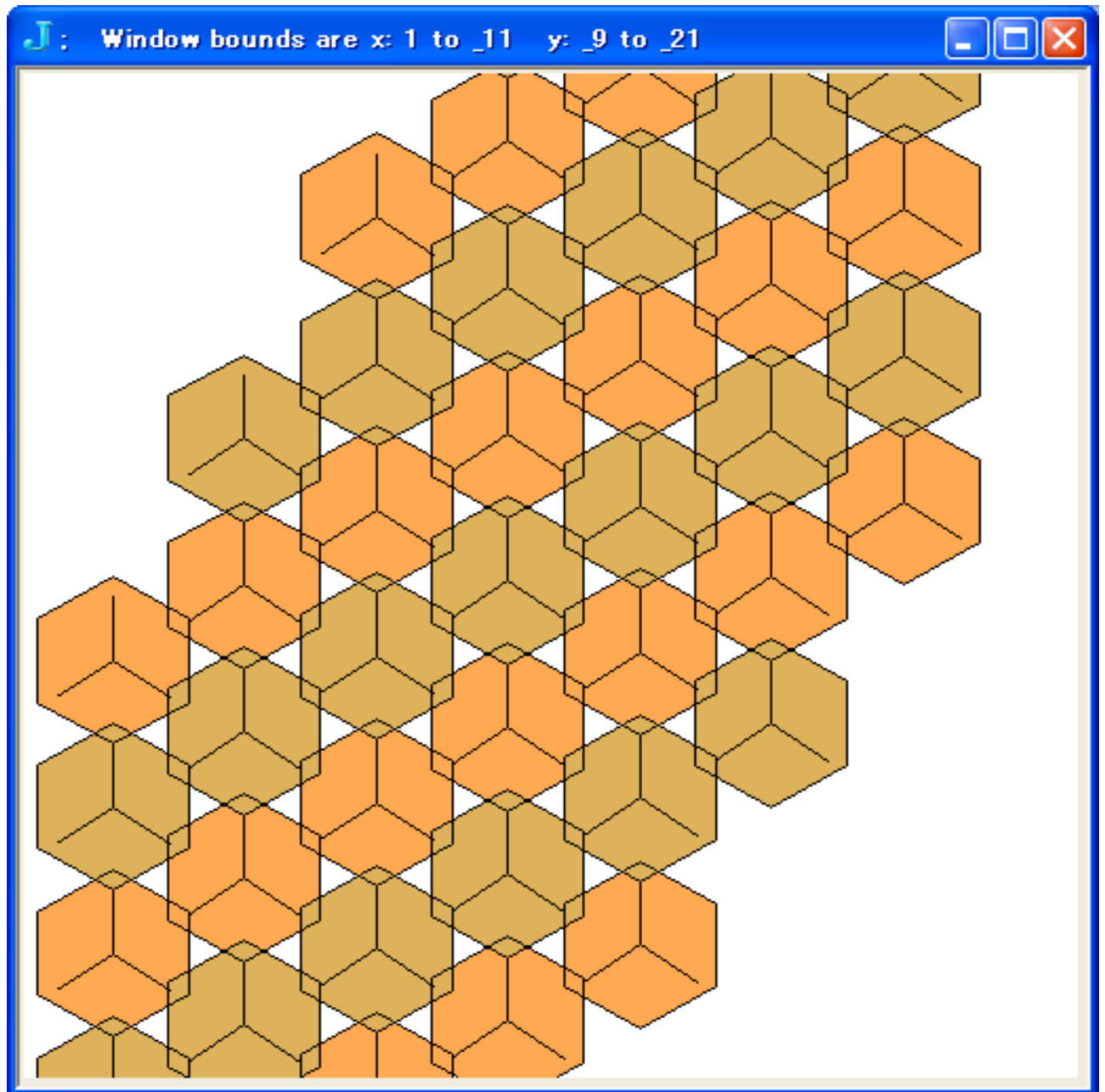


Yの線 draw\_dline\_over で重ね描きする

ATL=: 0 0 , \_0.65 0.45, 0 0, 0 \_0.75, 0 0 ,: 0.65 0.45

(144 93 84;230 75 107;0 0 0) hokusai\_at 6 7

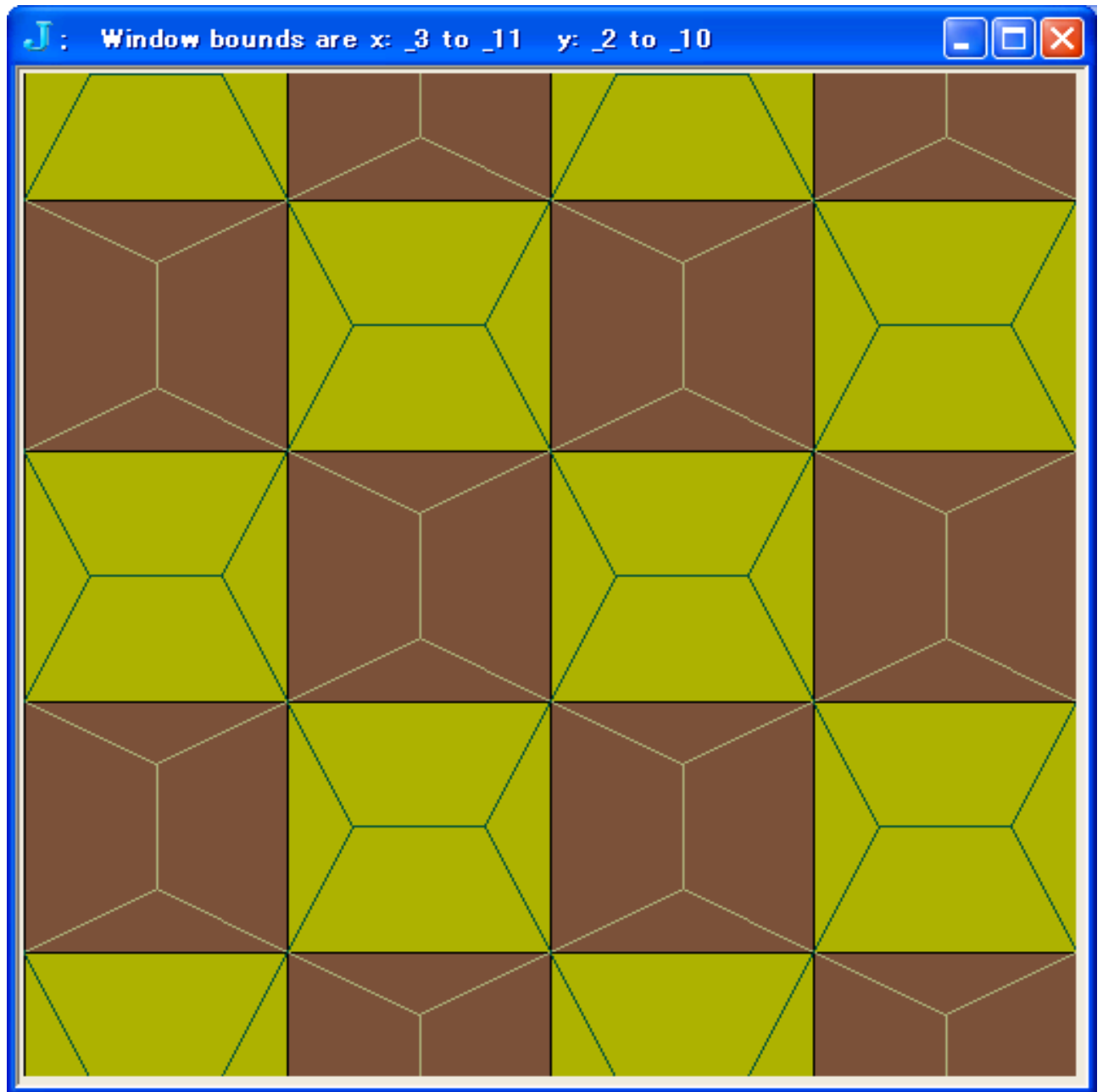
色は山吹色とブロンド



### 5.3 八つ手麻の葉 (No.01)

```
((0 83 62;170 179 0;170 179 0;0 83 62);<7 8) ym_grad_over ''
```

線も色指定できるようにし、パラメーターが多いので手順を単独のスク립トに書いた。色は若竹と萌黄を使っている。バックグラウンドの色が濃い場合は *dline* の色を白 (255 255 255) などの色指定にしないと濃い色に埋没する



正方形 .

NB. -----

NB. 八手麻の葉

NB. 8 hands Hemp leaves

YM0=: 1 1, \_1 1, \_1 \_1 , 1 \_1, : 1 1

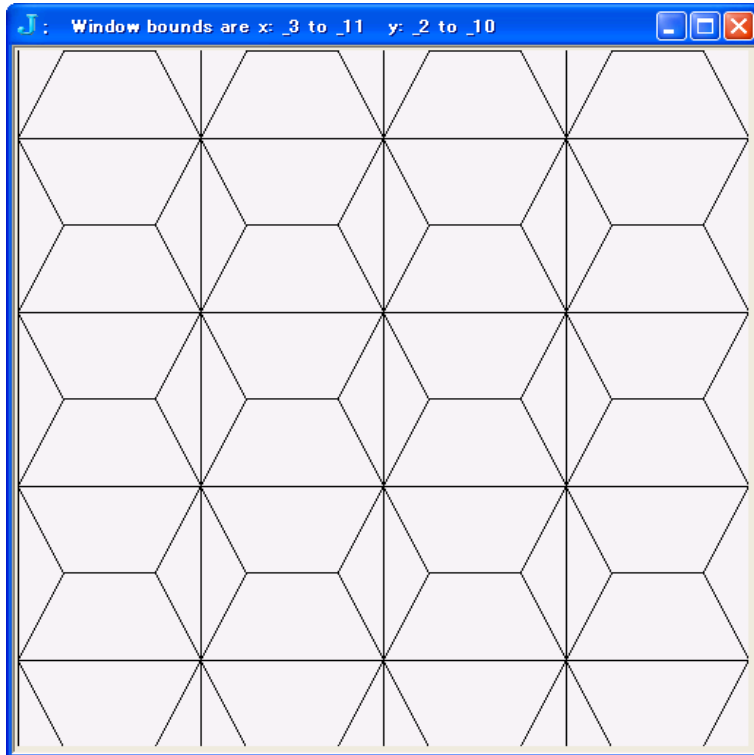
線 クロスの線と付加する線。YML\_2 は YML\_1 を 90 度回転させるための x,y を入れ替える。



```

YML0=: 0 0,_1 1,0 0,_1 _1,0 0,1 _1 , 0 0, 1 1,: 0 0
YML1=: 0 0,0 0.5,1 1,0 0.5,_1 1,0 0.5 ,0 _0.5,1 _1,0 _0.5,: _1 _1
YML2=: |."1 YML1

```



差分 4の図形とも正方形の重心(0,0)の積み重ね分

```
YMPARAM=: 0 0;2 0;0 2
```

手順 手順は引数が多いので script に書き出した

```

hokusai_ym=: 4 : 0
NB. (0 83 62;170 179 0;170 179 0;0 83 62) hokusai_ym 7 8
NB. color 0 1 ->dpoly 2 3 -> dline
NB. color 2 3 is reverse of 0 1 or 0 0 0 /255 255 255
'Size'=. y
'Color0 Color1 Color2 Color3'=. x
tmp0=.Size calc_each_poly (<YML0),<YMPARAM
tmp1=.Size calc_each_poly (<YML1),<YMPARAM
tmp2=.Size calc_each_poly (<YML2),<YMPARAM
'Ind0 Ind1'=. index_separate Size
'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp0
'Gr2 Gr3'=. (<Ind0 { ,tmp1),<Ind1{,tmp2
popup_dwin tmp0
Color0 dpoly L:0 Gr0

```

```

Color1 dpoly L:0 Gr1
Color2 dline L:0 Gr2
Color3 dline L:0 Gr3
)

```

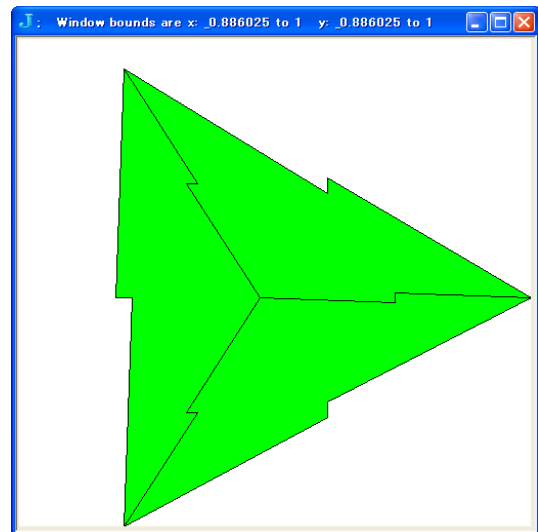
#### 5.4 松川麻の葉 (N0.10)

三角形 3 三角形の 2 つのピース。各辺に稲妻を入れる

```

MA0, .MA1
_0.5 0.886025| 1 1.732
_0.47 0| 0.22 1.299
_0.53 0| 0.28 1.299
_0.5 _0.886025| _0.5 0.886025
0.25 _0.463| 0.22 0.433
0.25 _0.403| 0.28 0.433
1 0| 1 0
0.25 0.463| 0.97 0.866
0.25 0.403| 1.03 0.866
_0.5 0.886025| 1 1.732

```



Y 字稲妻の線分 三角形の重心を起点として一筆書きするため、行きつ戻りつしている。数値は図形からざっと求めたもので精密ではない。

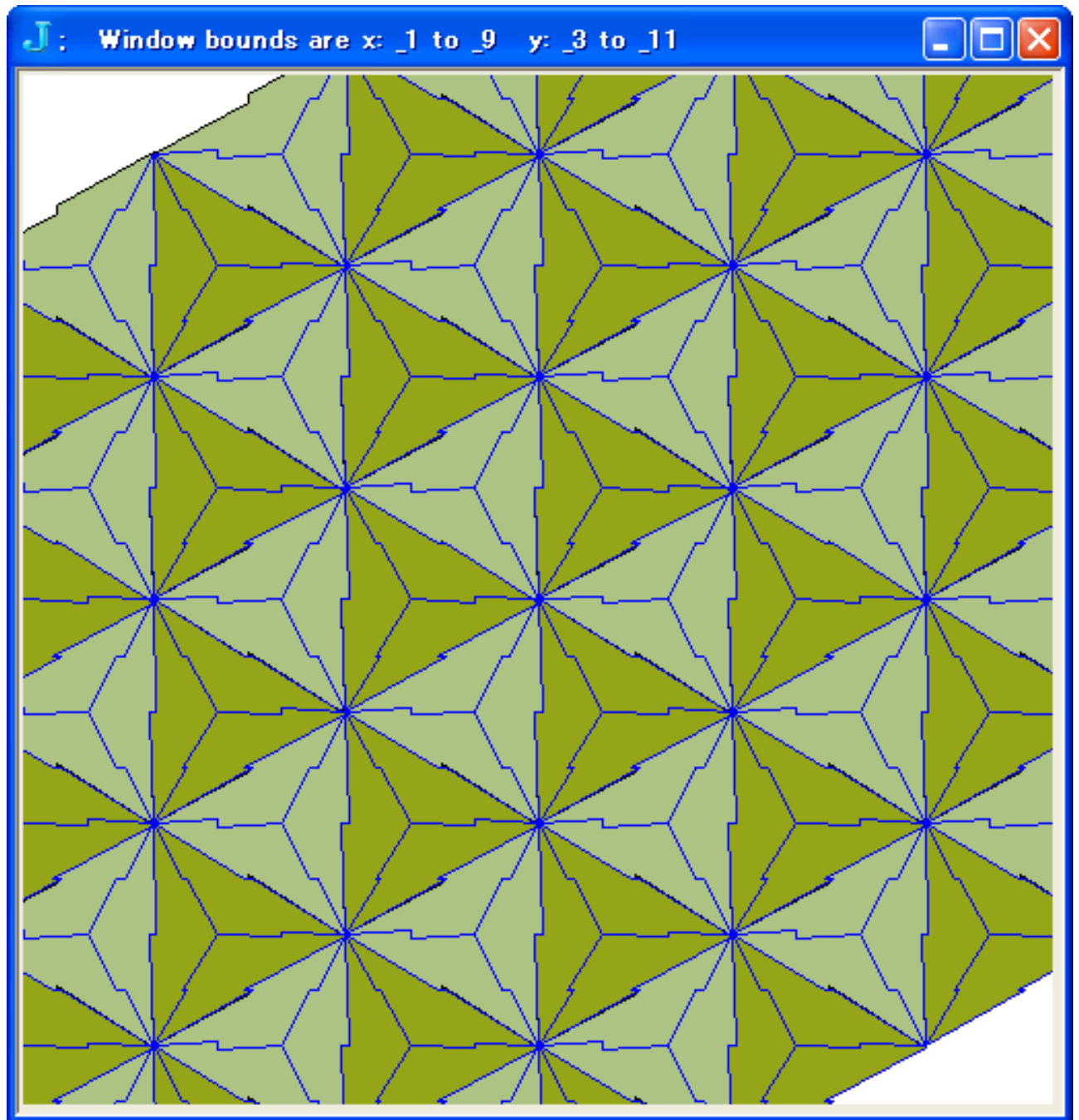
color3 color3 は線の色。バックが濃い場合は白抜き (255 255 255) などが映える

差分パラメータ 2 の三角形と 2 の Y 字の差分。

```

MA0PARAM=: _0.5 0.866025;1.5 0.866025;0 1.73205
MA1PARAM=: 1 1.73205;1.5 0.866025;0 1.73205
MAL0PARAM=: 0 0;1.5 0.866025;0 1.73205
MAL1PARAM=: 1 1.73205;1.5 0.866025;0 1.73205

```



*(169 192 135;151 166 30;0 0 255)hokusai\_ma 6 7*

hokusai\_ma=: 4 : 0

NB. (169 192 135;151 166 30;104 126 82) hokusai\_ma 6 7

NB. y is Size

'Color0 Color1 Color2'=. x

Size=. y

```

tmp0=.Size calc_each_poly (<MA0),<MA0PARAM
tmp1=.Size calc_each_poly (<MA1),<MA1PARAM
tmp2=.Size calc_each_poly (<MAL0),<MAL0PARAM
tmp3=.Size calc_each_poly (<MAL1),<MAL1PARAM
'Ind0 Ind1'=. index_separate Size
'Gr0 Gr1'=. (<Ind0 { ,tmp0),<Ind1{,tmp0
'Gr2 Gr3'=. (<Ind0 { ,tmp1),<Ind1{,tmp2
popup_dwin tmp0
Color0 dpoly L:0 Gr0
Color1 dpoly L:0 Gr1
Color1 dline L:0 Gr1
Color2 dline L:0 tmp2
Color2 dline L:0 tmp3
)

```

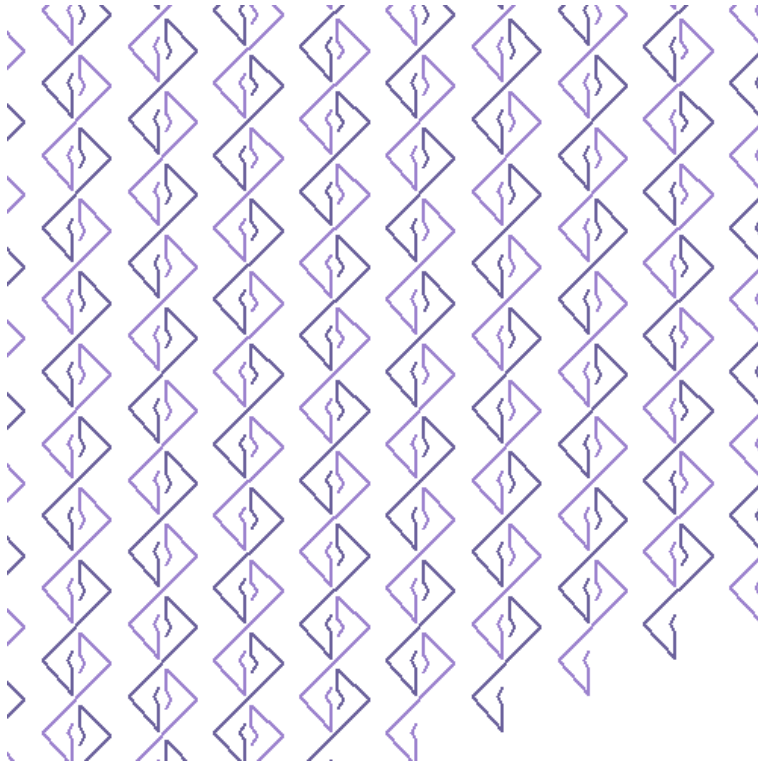
## 6 小紋のテクニック (4) 直線での色変化

小紋を描くとき数本の直線のラインを組み合わせるときはラインごとに色変化が付けられる。これに対してピースが一筆書きできる場合は一気にプログラムしてしまうと、単色でのっぺりしてしまう。そこでポリゴンで行ったのと同様の色変化を付けることができるようにした。プログラムが型紙より便利な点である

直線の色変化 .

```
NB. --2-colors for polygon -----
draw_dpoly_grad_sub=: 4 : 0
NB. x is color;size_of_matrix /raw & column(ex. 4 5)
NB. y is (<piece_data) , < diff_paramemter
NB. ((255 0 0 ;200 0 100), <4 5) draw_dpoly_grad (<IM),<IMPARAM
'Color Size'=. x
tmp=. Size calc_each_poly y
'Ind0 Ind1'=. index_separate Size
(<Ind0 { ,tmp),<Ind1{,tmp NB. 'Gr0 Gr1'
)

draw_dline_grad=: 4 : 0
'Color Size'=. x
'Gr0 Gr1'=. x draw_dpoly_grad_sub y
tmp=. Size calc_each_poly y
popup_dwin tmp
'Color0 Color1'=. Color
Color0 dline2 L:0 Gr0
Color1 dline2 L:0 Gr1
)
```



太い絵筆を作る ペンサイズ 3 *C.Reiter* のオリジナルを改変した

```
dline3=: 3 : 0 "1 2
NB. modified dline// bold line
0 0 0 dline y
:
Y=.x:^:_1 SC 2{"1 y
wd 'psel ',WIN_nam
glrgb x
glpen 3,PS_SOLID NB. pen 3/solid
gllines ,Y
glpaint ''
)
```

## 7 直線の小紋

キャンバスはシリーズ全体を *C.Reiter* の *dwin2* で統一している。左下が原点。キャンバスの大きさは小紋の配列 (例えば  $10 \times 10$ ) を指定したときに用いるデータの最大値を参照して自動的に決めている

## 7.1 釣り四つ目 No15

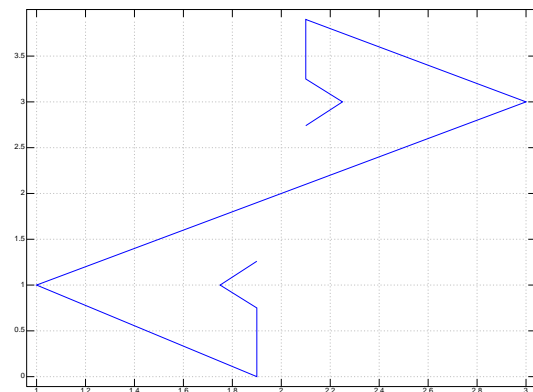
北斎の時代には合成絵の具「ブルشان・ブルー」がもたらされていて北斎も愛用した。  
 小手調べに青で描いてみる

最初のピース 右上方から順に

```

TY
  x      y
2.1 2.74
2.25    3
2.1 3.25
2.1 3.9
  3      3
  1      1
1.9      0
1.9 0.75
1.75    1
1.9 1.26

```



差分 各小紋間の差分 「(始点);(X 方向の右隣の小紋との始点の差分); (y 方向の上の小紋との始点の差分)」  
 TYPARAM=: 2.2 2.6; 2.5 1.05; 0 2.1

各ボックス内での始点を定め、小紋データを差分を基に展開する

Script hokusai\_ty=: 4 : 0

NB. (255 0 0;10 10) hokusai\_ty ''

Color0=.x ] Size=. x

tmp0=. Size calc\_each\_poly (<TY);<TYPARAM  
 (find\_center tmp0) dwin ''

Color0 dline3 L:0 tmp0

イズ 3

)

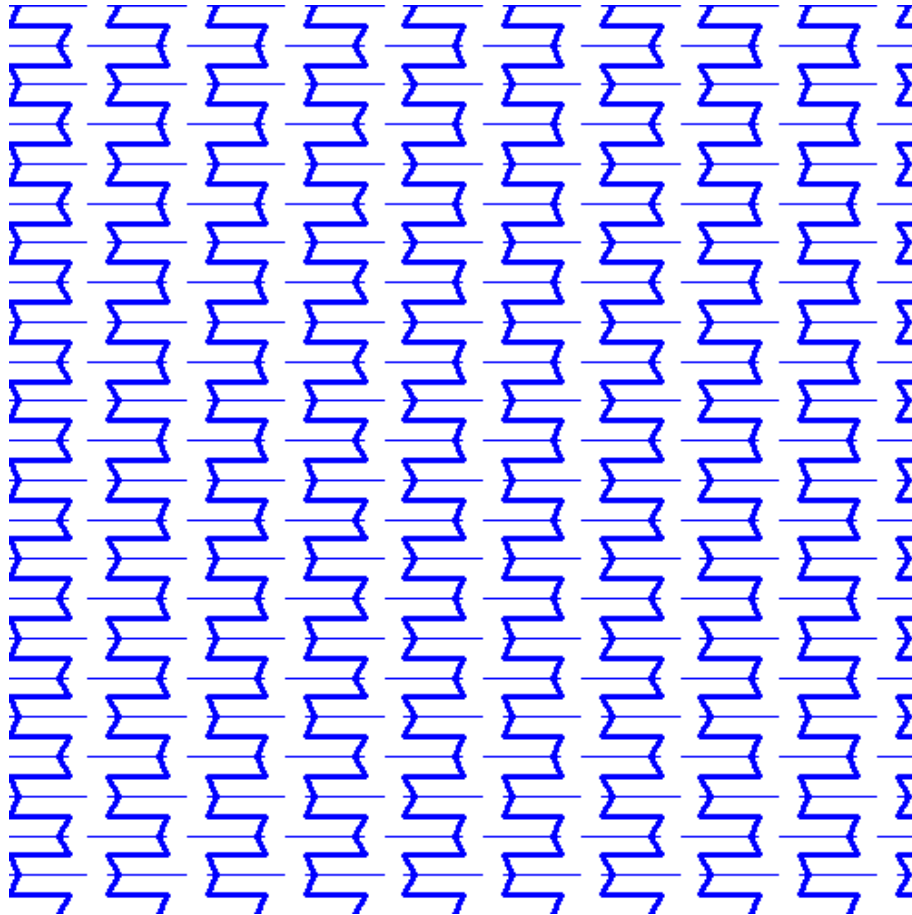
NB. 色と小紋の数を指定

NB.Box の配列で各小紋の位置データを計算

NB. キャンバスの大きさを指定して起動

NB. 各小紋をラインで描く dline3 ペンサ

## 7.2 行きつ戻りつ No13



最初のピース .

Σ 型

IKI

2 0

5 0

4.5 1

5 2

2 2

2.5 3

2 4

2本の横線

IKI2=: 1 1, : 5 1

IKI3=: 2 3, : 6 3

小紋の差分 始点;x 方向の差分;y 方向の差分

IKIPARAM=: 2 0;5 0;0 4

Script .

hokusai\_iki=: 4 : 0

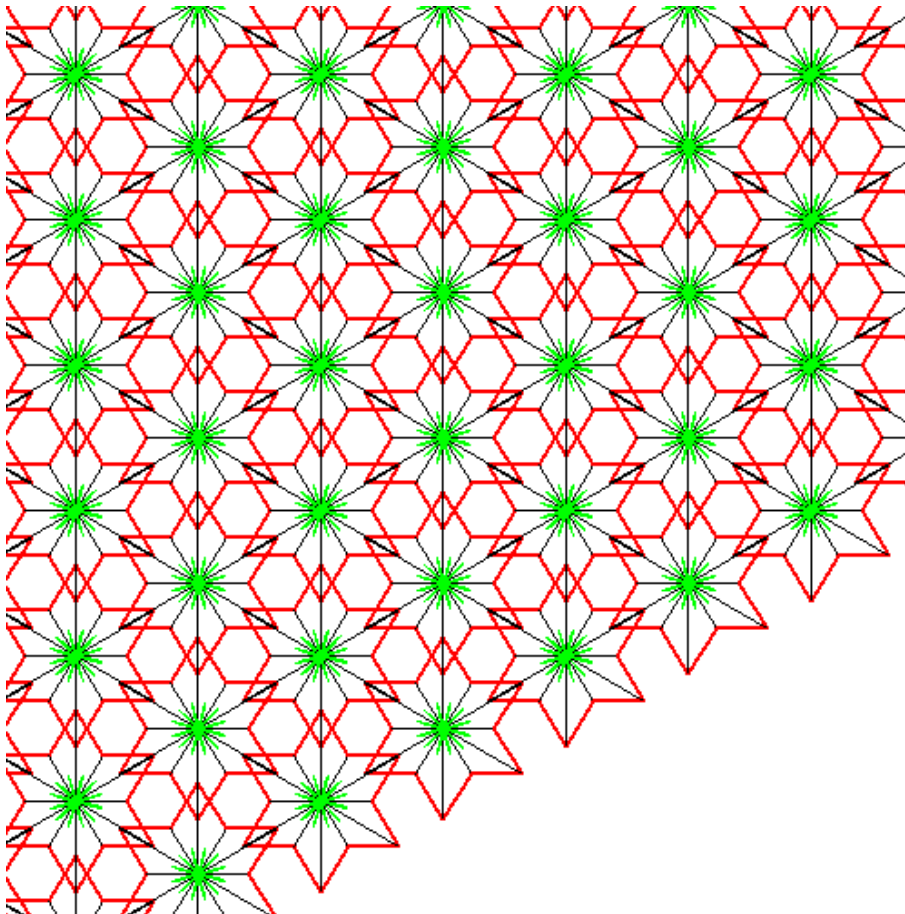


```

NB. 255 0 0 hokusai_iki 10 10
Color0=. x
Size=. y
tmp0=. Size calc_each_poly (<IKI);<IKIPARAM
tmp1=. Size calc_each_poly (<IKI2);<IKIPARAM
tmp2=. Size calc_each_poly (<IKI3);<IKIPARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
Color0 dline2 L:0 tmp1
Color0 dline2 L:0 tmp2
)

```

### 7.3 桜割り No07



桜の外縁 六角形を作る

```
HA0=: +. r. 2p1*(i.6)%6 NB. hexagon
```

•

$$2\pi \times \frac{i.6}{6}$$

2p1\* (i.6)%6

0 1.0472 2.0944 3.14159 4.18879 5.23599

• 極座標に変換

r. 2p1\* (i.6)%6

1 0.5j0.866025 \_0.5j0.866025 \_1j1.22465e\_16 \_0.5j\_0.866025 0.5j\_0.866025

このまま *plot* で描くとガウス座標で 6 角形になるが、*ISIGraph* では複素座標はサポートされていない。

• 実部と虚部に分離しデカルト座標の  $x,y$  とする。

clean +. r. 2p1\* (i.6)%6 NB. clean は 0 のごみ取り

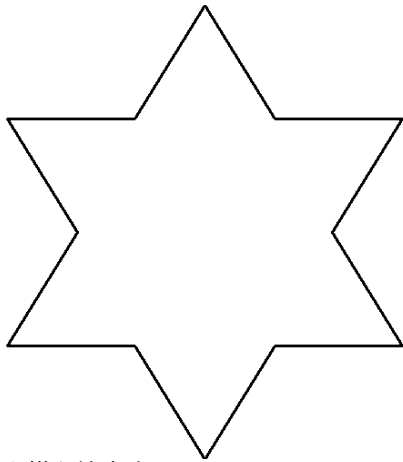
```
x      y
1      0
0.5    0.866025
_0.5    0.866025
_1      0
_0.5   _0.866025
0.5   _0.866025
```

切り込み .

	SW
• 頂角が上に来るよう回転する	0 1
• 6 角形を切り込みを入れ 12 角形にする。図	_0.307 0.5
を描き作業は手で行った	_0.866025 0.5
%: +/ ^&2] 0.5 0.25	_0.559 0
0.559017 NB. (1)	_0.866025 _0.5
	_0.307 _0.5
	0 _1
0.8660225 - 0.559017	0.307 _0.5
0.307006 NB. (2)	0.866025 _0.5
	0.559 0
	0.866025 0.5
	0.307 0.5
	0 1

$$\text{内角の和} = 180^\circ \times n - 360^\circ$$

• 始点は頂上で左回りに描く



```
_1 _1 1 1 dwin ''
0 0 0 dline3 SW
```

花芯 花を描き縮小する

```
SWL1=: }:"1 (( circ_flower 7 7),.1) mp elongm 0.3 0.3
```

- 花卉の多い花を描く *circ\_flower 7 7*(輪違い麻の葉を参照)
- *xy* 方向を夫々 0.3 倍に縮小する回転行列

```
elongm 0.3 0.3
```

```
0.3 0 0
```

```
0 0.3 0
```

```
0 0 1
```

- *SW* の右に *I* を付加し, 内積演算の後 *I* を落とす *mp*:=+/. \*

Script .

NB. 07-桜割り

NB. 07-sakura-wari

```
HA0=: +. r. 2p1*(i.6)%6 NB. hexagon
```

```
SW0=: clean }:"1 |.(4|.|.|.|"1 HA0),|.1{HA0 NB. hexagon
```

```
SW2=: _0.433013 0.5 ,_0.5 0, _0.433013 _0.5,0.433013 _0.5,0.5 0,: 0.433013 0.5
```

NB. kubo

```
SW=:({.SW),~ SW=: ;("2) ,. 2 2 $ L:0 ({ SW0) , L:0 {SW2 NB. interlace
```

```
SWPARAM=: 0 1; 1.5 0.866025;0 1.732013
```

```
SWL0=(0 6;1 7;2 8;3 9;4 10;5 11){ L:0 SW
```

```
NB. plot { |: circ_flower 12 5 // 芯
```

```
SWL1=: }:"1 (( circ_flower 7 7),.1) mp elongm 0.3 0.3
```

```
hokusai_sw=: 4 : 0
```

```
NB. (255 0 0; 0 255 0 ;10 10) hokusai_sw''
```

```
'Color0 Color1'=. x ] Size=. y
```

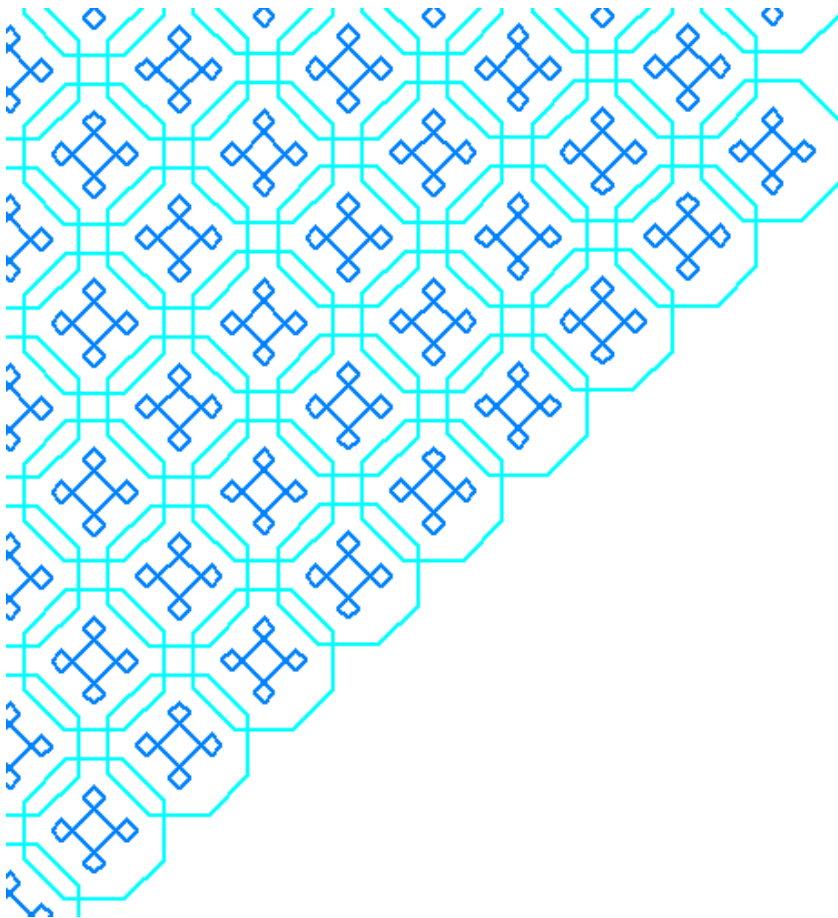
```
tmp0=. Size calc_each_poly (<SW);<SWPARAM
```

```

tmp1=. Size calc_each_poly (<SWL0);<SWPARAM
tmp2=. Size calc_each_poly (<SWL1);<SWPARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
dline L:0 tmp1
Color1 dline L:0 tmp2
)

```

#### 7.4 かくの七宝 No31



八角形 六角形を基に作成し、 $\frac{1}{8}\pi$  回転する

```

HA8=: clean (+. r. 2p1*(i.8)%8) NB. octagon
HA8=: }: "1 ((HA8,1 0),. 1) mp rotm 1r8p1 NB. rotate 1/8 pi

```

井桁 正位できっちり描いてから回転( $\frac{1}{4}\pi$ )、縮小 0.4 倍

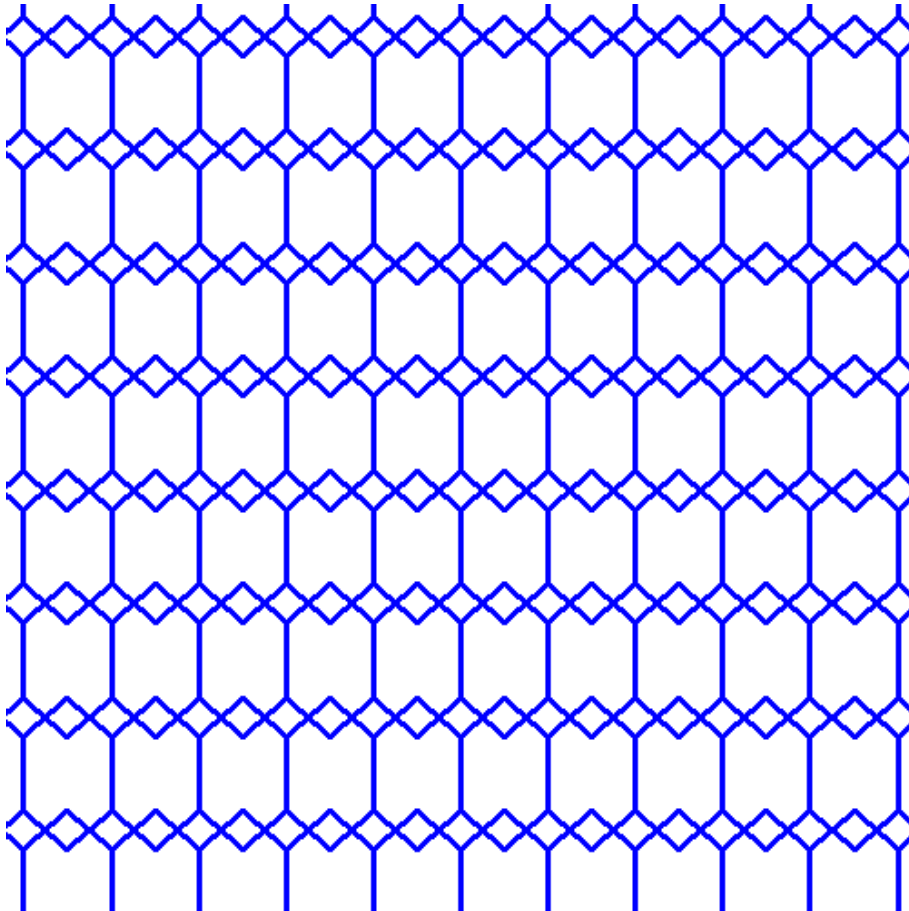
```

KS1=: 1 0.5,1 1,0.5 1,0.5 _1,1 _1,:1 _0.5 NB. i type
KS1=:KS1, _1 _0.5,_1 _1,_0.5 _1,_0.5 1, _1 1,_1 0.5,: 1 0.5

```

```
KS1=:}:"1 ((KS1,.1)mp elongm 0.4 0.4) mp rotm 1r4p1 NB. rotate & smaller
```

## 7.5 はやわり No67



6 角形の回転 六角形を  $\frac{1}{2}\pi$  回転させたものは  $xy$  を入れ替えた形になる。

縮小は  $x$  を  $\frac{2}{3}$  とした

```
HH=: 1 0 ,~ +. r. 2p1*(i.6)%6
```

NB. hexagon

```
HH=: clean }:"1 (HH,.1) mp (rotm 1r2p1) mp elongm 2r3 1
```

NB. rotate top is ^

HH

```
      0      1
_0.57735 0.5
_0.57735 _0.5
      0      _1
0.57735 _0.5
0.57735 0.5
      0      1
```

H6

```
      1      0
0.5 0.866025
_0.5 0.866025
      _1      0
_0.5 _0.866025
0.5 _0.866025
      1      0
```

## 7.6 金網亀甲 No.53

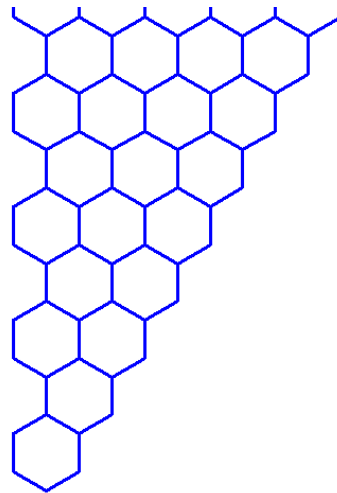
頂点が上に来た正6角形  $\frac{1}{2}\pi$  回転させる。始点は頂上

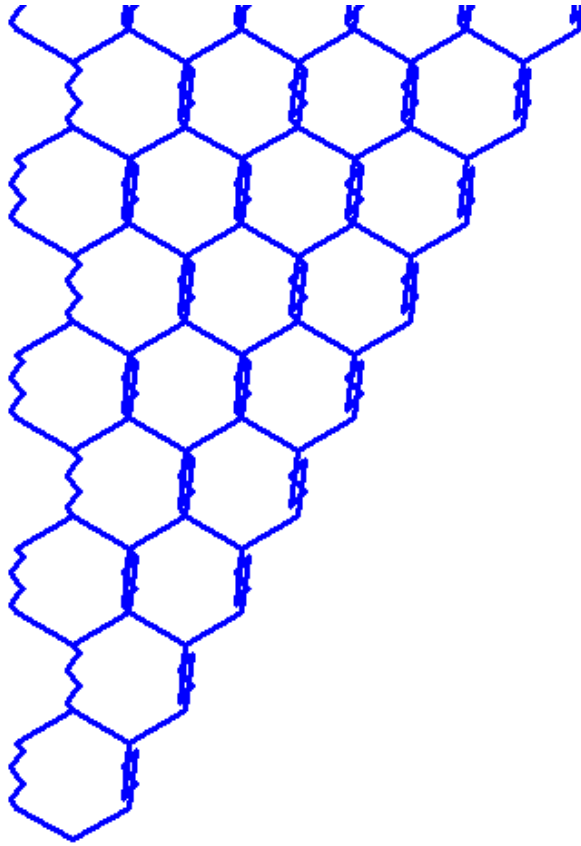
```
HH=: 1 0 ,~ +. r. 2p1*(i.6)%6      NB. hexagon
KK=: clean }:"1 (HH,.1) mp rotm 1r2p1 NB. rotate top is ^
```

亀甲を1色で敷き詰める .

```

      y
    0,4
      ↑
    0,1  →  0.866025, 2.5  x
-----
base (x,y)
KKPARAM=: 0 1;0.866025 1.5;0 3
NB. origin;xdiff;ydiff
```





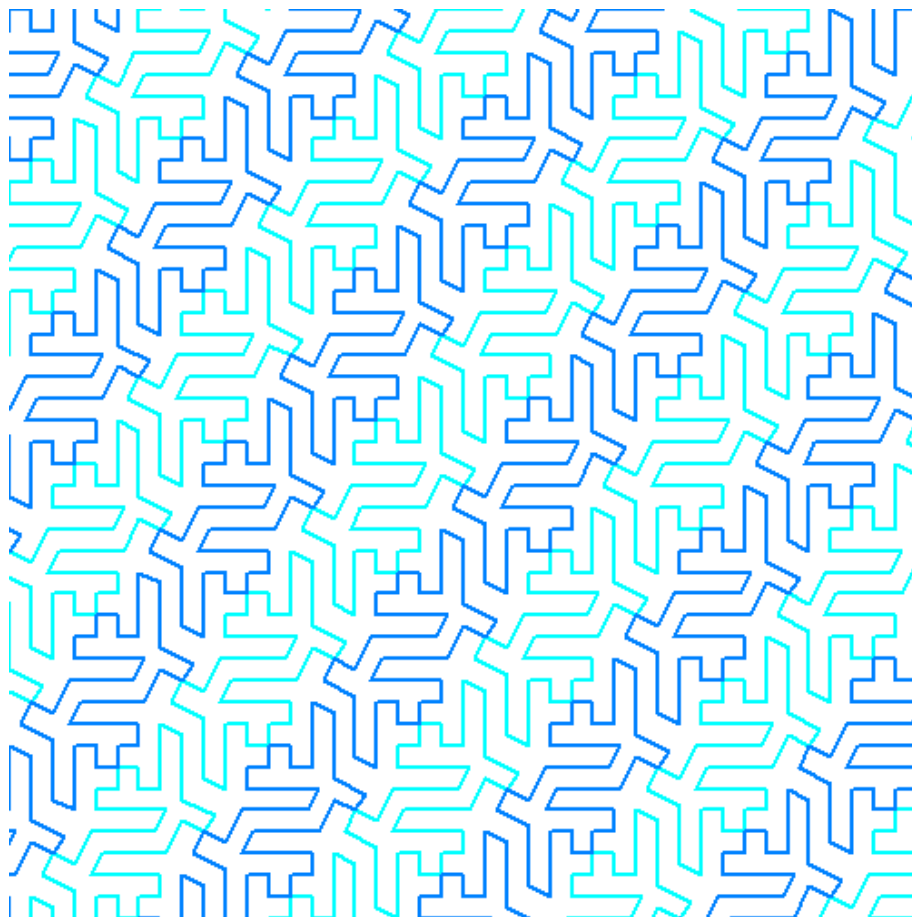
Script 亀甲の敷き詰めの *Script*

```

hokusai_kk0=: 4 : 0
Color0=. x ]   Size=. y
tmp0=. Size calc_each_poly (<KK);<KKPARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
)

```

### 7.7 燭光の万字結び NO. 22



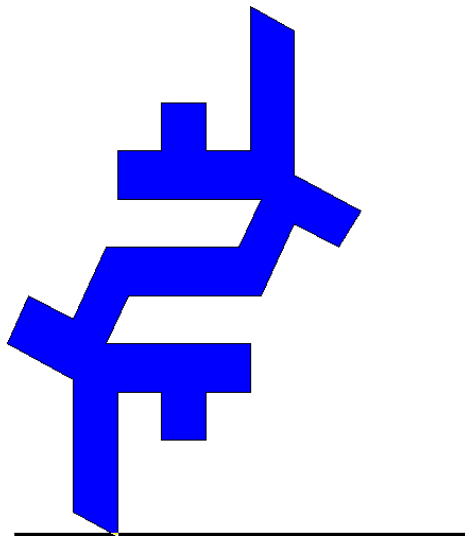
最初のピース 回転していない状態で方眼紙に描く。



		7.25	5	
幾つかのピースを画面に		8	6.5	
描きながら微調整した		9	6	3.75 6
		9.5	6.75	3 4.5
SM0		8	7.5	2 5
4 0		8	10.5	1.5 4
4 3		7	11	3 3.25
5 3		7	8	3 0.5
5 2		6	8	4 0
6 2		6	9	
6 3		5	9	
7 3		5	8	
7 4		4	8	
3.75 4		4	7	
4.25 5		7.25	7	
		6.75	6	

大工の差し金を手にした  
であろう北斎の美のバラ  
ンス感覚も鋭いが、江戸  
の彫師の腕にも感服する

ピースと差分



$$\begin{pmatrix} x \\ (3,7) \\ \uparrow \\ (4,0) \\ \text{origin} \end{pmatrix} \rightarrow (11,1) \quad x$$

SMOPARAM=: 4 0;7 1;\_1 7

Script .

```

NB. 燭光の万字繋ぎ No. 22
SM0=: 4 0,4 3,5 3,5 2,6 2,6 3,7 3,7 4,3.75 4,:4.25 5
NB. start a
SM0=:SM0,7.25 5 ,8 6.5,9 6,9.5 6.75,8 7.5,8 10.5,:7 11
SM0=: SM0,7 8,6 8,6 9,5 9,5 8,4 8,4 7,7.25 7,:6.75 6
NB. start b
SM0=: SM0,3.75 6,3 4.5,2 5,1.5 4,3 3.25,3 0.5,:4 0

SMOPARAM=: 4 0;7 1;_1 7

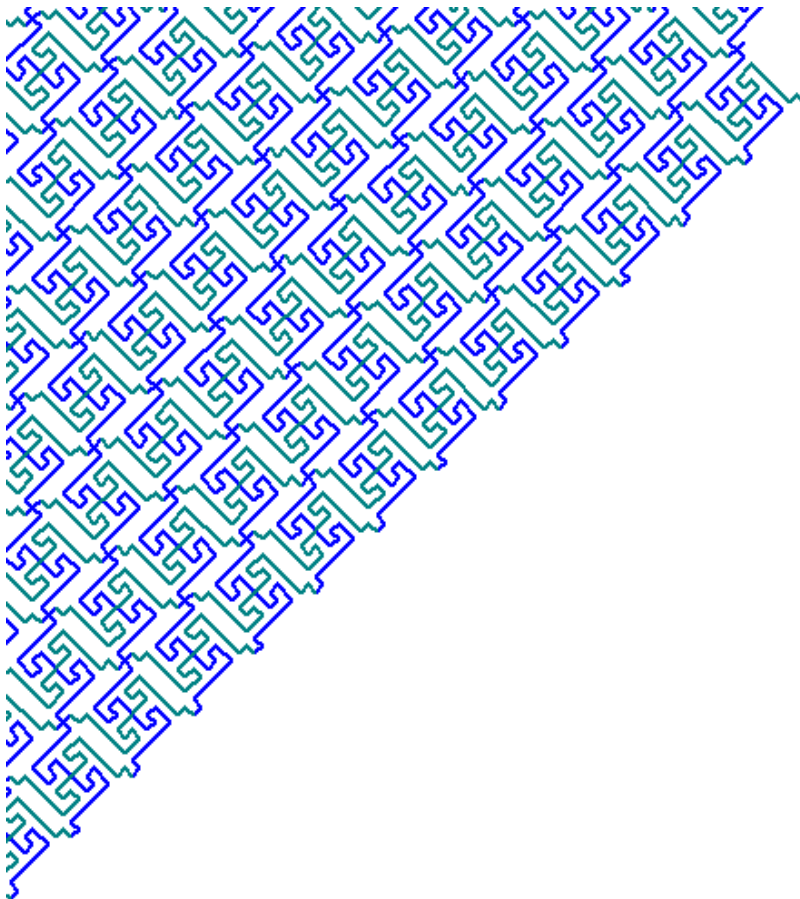
```

```

hokusai_sm=: 4 : 0
NB. 0 0 255 hokusai_sm 10 10
Color0 =. x ] Size=. y
tmp0=. Size calc_each_poly (<SM0);<SM0PARAM
(find_center tmp0) dwin ''
NB. Color0 dpoly L:0 tmp0
Color0 dline3 L:0 tmp0
)

```

## 7.8 菱の釣万字 No.26

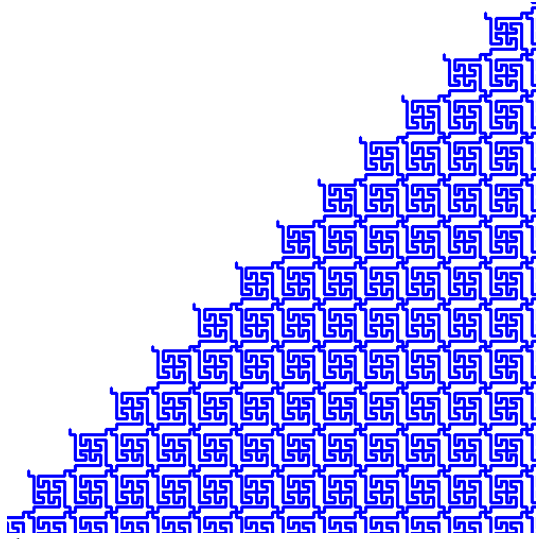


ピース .

```

NB. 菱の釣万字 No.26
HM0=: 1 1,2 1,2 2,7 2,7 4,6 4,6 3,5 3,5 7,4 7,4 6,3 6,3 8,8 8,8 9,:9 9
HM1=: 9 1,9 2,8 2,8 7,6 7,6 6,7 6,7 5,3 5,3 4,4 4,4 3,2 3,2 8,1 8,:1 9
HM0PARAM=: 1 1;8 0;8 8

```



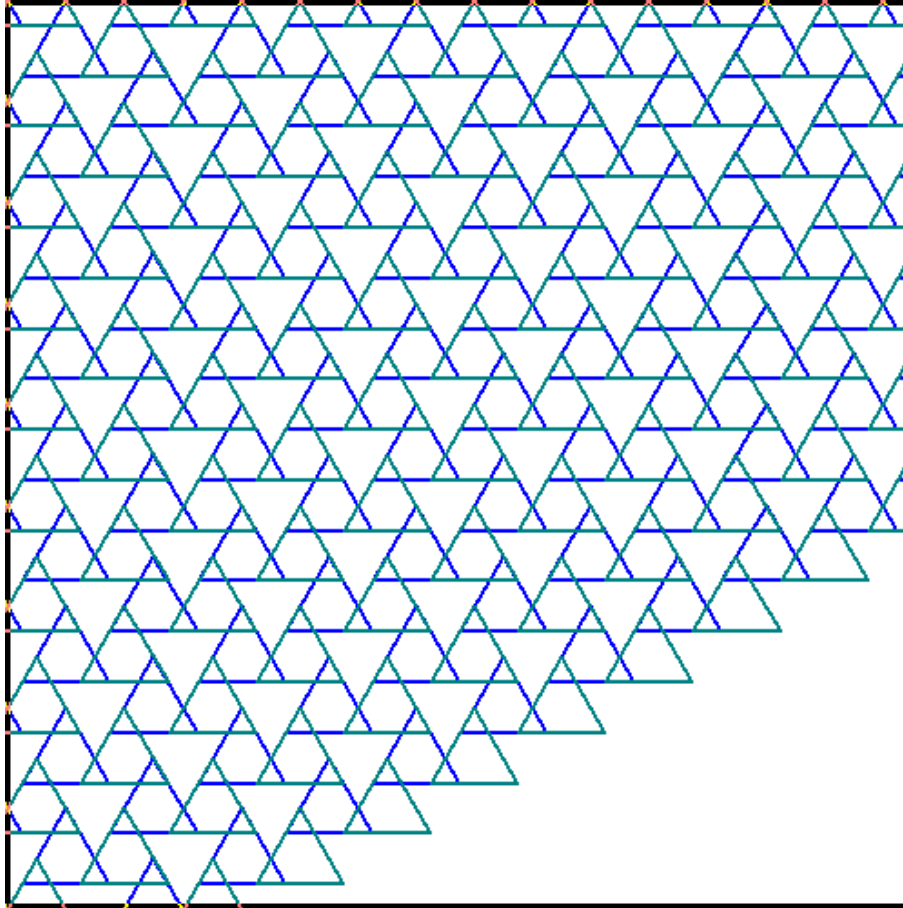
回転  $\frac{1}{4}\pi$  差分パラメーターも自動生成

```
NB. rotate
HM0R=: }:"1 clean (HM0 ,.1) mp rotm 1r4p1
HM1R=: }:"1 clean (HM1 ,.1) mp rotm 1r4p1
HM1PARAM=: 1r4p1 find_rotm_diff HM0PARAM
```

Script .

```
hokusai_hm=: 4 : 0
NB. 0 0 255 hokusai_hm 10 10
'Color0 Color1 '=. x
Size=. y
NB. y is rotm
tmp0=. Size calc_each_poly (<HM0R);< HM1PARAM
tmp1=. Size calc_each_poly (<HM1R);< HM1PARAM
(find_center tmp0) dwin ''
NB. Color0 dpoly L:0 tmp0
Color0 dline2 L:0 tmp0
Color1 dline2 L:0 tmp1
)
```

## 7.9 破れ罫目 No.34



正三角形  $\frac{1}{2}\pi$  回転して頂点を上に持ってくる

```
NB. triangle(0)
```

```
YK0:=clean ({}:"1 ((+.r. 2p1 * (i.3)%3),.1) mp rotm 1r2p1),0 1
```

2 個の三角形 絵を眺めていて次の 2 個の正三角形をピースとすればよいと気づいた。

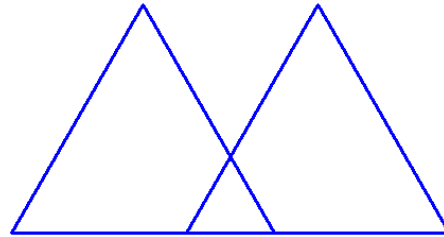
```
NB. triangle(1)
```

```
YK1:= YK0+("1) (0.8666025*4r3), 0
```

```

YK0, ._. , .YK1
      0      1 _ .  1.15547      1
_0.866025 _0.5 _ .  0.289445 _0.5
0.866025 _0.5 _ .  2.0215 _0.5
      0      1 _ .  1.15547      1

```



差分 差分パラメーター (始点;xの差分;yの差分)

微細な重ねあわせが要請される

YKPARAM=: 0 1; 1.73333 1;0 2

Script 正三角形を一個づつ描く。破れは埋め尽くさない飛び (空白) でパラメータで空白を作る。

```

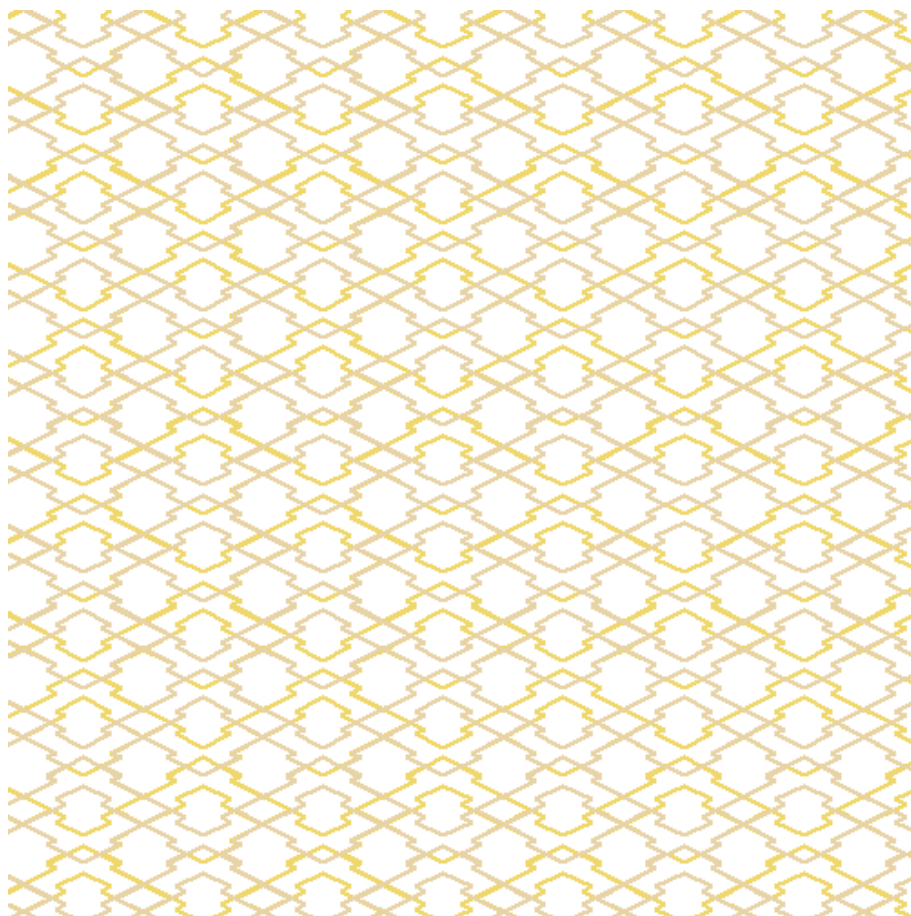
hokusai_yk=: 4 : 0
NB.    0 0 255 hokusai_yk 10 10
Color0=. x ] Size=. y
tmp0=. Size calc_each_poly (<YK0);< YKPARAM
tmp1=. Size calc_each_poly (<YK1);< YKPARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
Color0 dline3 L:0 tmp1
)

```

## 7.10 松皮 No61

ライン系で一番手強く、図形の半分近くが重なる。差分パラメーターをあれこれ試して固定した後、図形を微調整した。

北斎先生も型紙を何通りも切ったり捨てたりしながら、ごみの山の中で苦吟したのではないだろうか。



最初のピース 最初に四角く正立で書く。2 個の図形は色を変えるときに切り離しもできるようにした。

NB. Mk1\_insert is small piece connect/divide both OK

MK0=: 2 2,5 2,5 3,13 3,13 4,15 4,15 6,16 6,16 14,17 14,17 17,14 17,:14 16

NB. figure

MK1\_insert=: 9.5 16,9.5 12.5,8.5 12.5,8.5 8.5,12.5 8.5,12.5 9.5,19 9.5,19 16,:20 16 NB. figure

MK1\_insert=: MK1\_insert,20 20,16 20,16 19,9.5 19,:9.5 16

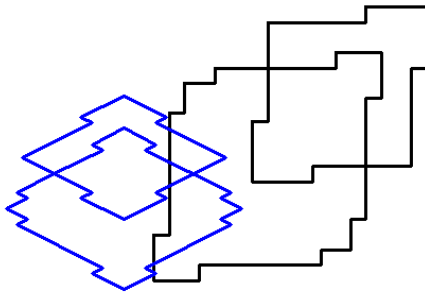
MK0=: MK0,MK1\_insert ,6 16,6 15,4 15,4 13,3 13,3 5,2 5,:2 2 NB. figure A+B / cutoff OK

回転と圧縮  $\frac{1}{4}\pi$  回転させ、縦に  $\frac{1}{2}$  圧縮する

NB. rotate 1/4 pi and compress vertical 1/2

MK01=: clean }:"1 (MK0,.1) mp (rotrm 1r4p1) mp elongm 1.0 0.5

MK11=: clean }:"1 (MK1\_insert,.1) mp (rotrm 1r4p1) mp elongm 1 0.5



```

maxmin MK0
+-----+-----+
|20 20|2 2|
+-----+-----+
maxmin MK01
+-----+-----+
|7.77817 14.1421|_7.77817 1.41421|
+-----+-----+

_10 _10 25 25 dwin ''
0 0 0 dline3 MK0
0 0 255 dline3 MK01

```

差分パラメータ 手で何回も調整した。この後図形を重ねるためにピース (A,B) を削ったり膨らましたりして微調整した。

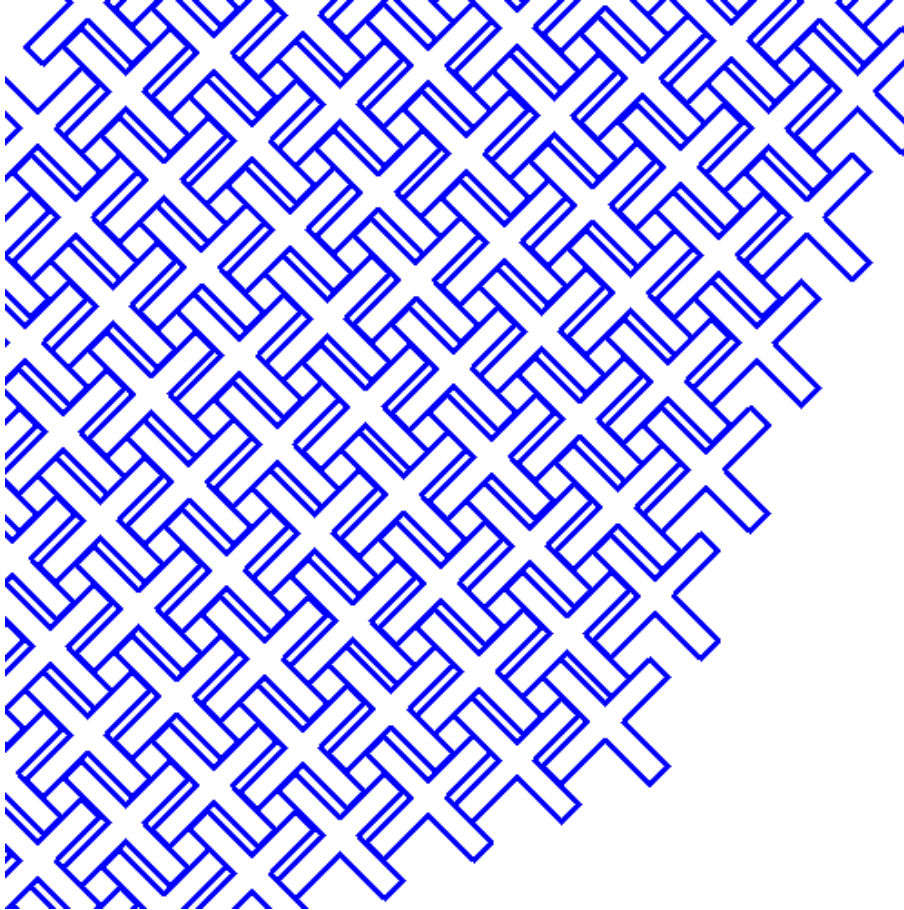
```

MK01PARAM=: 0 0 ;9.25 0;0 6.75 NB. xdiff ydiff is very delicate
Script 色を変える場合は MK01 を切り離し、コメントアウトした 2 行を生かす
MK0=: MK0,6 16,6 15,4 15,4 13,3 13,3 5,2 5,:2 2 NB. figure A+B / cutoff OK

hokusai_mk=: 4 : 0
NB. 0 0 255 hokusai_yk 10 10
Color0=. x ] Size=. y
tmp0=. Size calc_each_poly (<MK01);< MK01PARAM
NB. tmp1=. Size calc_each_poly (<MK11);< MK01PARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
NB. Color0 dline3 L:0 tmp1
)

```

## 7.11 碁盤筋交い No64



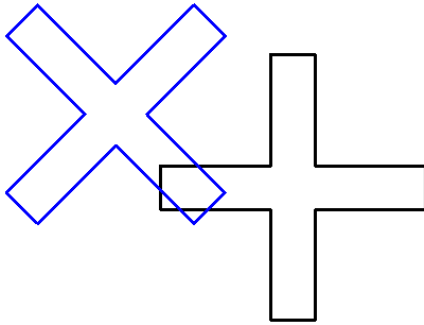
最初のピース 回転前の正立で描く

```
SK0=: 3.5 1,4.5 1,4.5 3.5,7 3.5,7 4.5,4.5 4.5 ,:4.5 7
SK0=:SK0,3.5 7,3.5 4.5, 1 4.5,1 3.5,3.5 3.5,: 3.5 1
SK0R=: clean }:"1 (SK0,.1) mp rotm 1r4p1
```

回転  $\frac{1}{4}\pi$  回転させる。

```
SK0R=: clean }:"1 (SK0,.1) mp rotm 1r4p1
```





小紋の位置パラメータ 位置パラメータは回転前を用いる

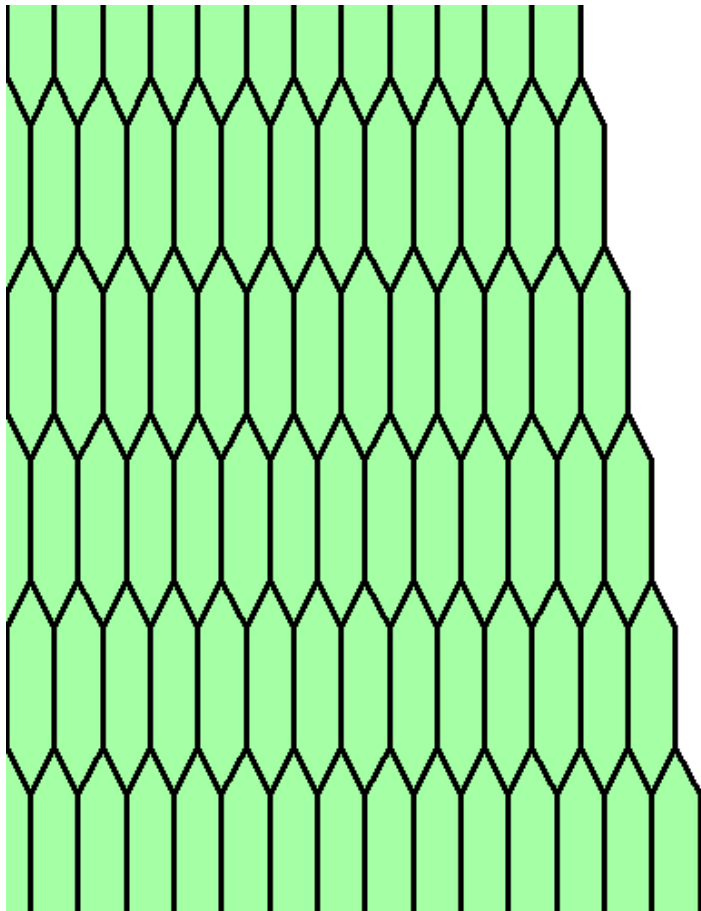
*SKPARAM=: 3.5 1 ;3.5 1.5;2 5 NB. base;x diff;y diff*

$$\begin{pmatrix} y \\ (5.5, 6) \\ \uparrow \\ (3.5, 1) \end{pmatrix} \rightarrow (7, 2.5)$$

Script .

```
hokusai_sk=: 4 : 0
NB. 0 0 255 hokusai_sk 10 10
Color0 =. x ] Size=. y
tmp0=. Size calc_each_poly (<SK0R);< SKPARAM
(find_center tmp0) dwin ''
Color0 dline3 L:0 tmp0
)
```

## 7.12 畳 No52

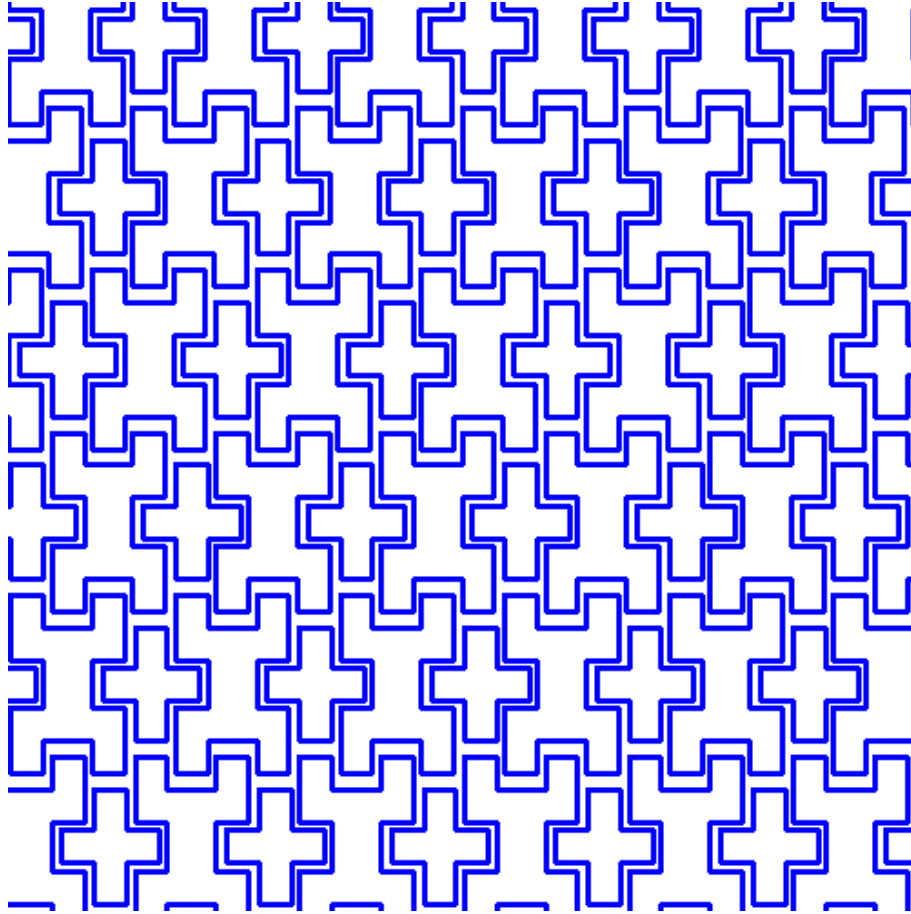


最初のピース .

TT0=: 3 1,4 3,4 8,3 10,2 8,2 3,:3 1

TT0PARAM=: 3 1;2 0;\_1 7

### 7.13 唐様せうじのくづし



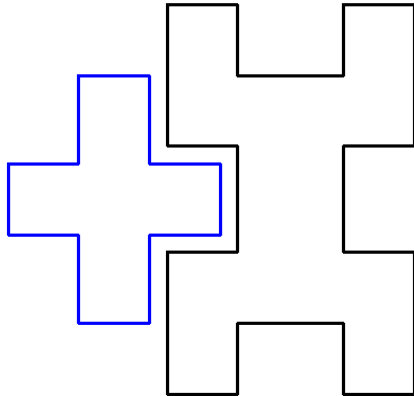
2 個のピース 正立で斜線がないので丁寧に追っていけばよい。障子の棧にあたる部分は空白で表現する

KSY0=: 5 1,7 1,7 3,10 3, 10 1,12 1,12 5 ,10 5,10 8,12 8,:12 12

KSY0=: KSY0,10 12, 10 10,7 10,7 12,5 12,5 8,7 8,7 5,5 ,:5 1

KSY1=: 2.5 3,4.5 3,4.5 5.5,6.5 5.5,6.5 7.5 ,4.5 7.5,4.5 10,2.5 10,:2.5 7.5

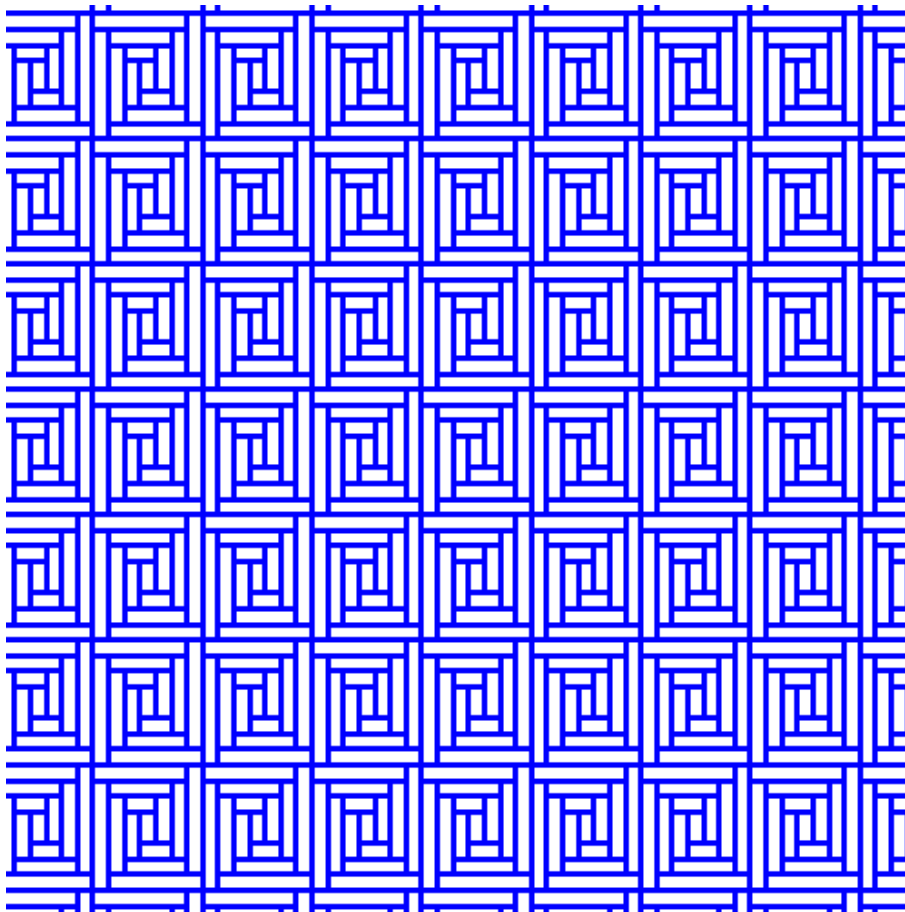
KSY1=: KSY1,0.5 7.5,0.5 5.5,2.5 5.5,:2.5 3



差分  $KSY0$  と  $KSY1$  を線や色を変えて栈の幅の微調整が必要。 $KSY0$  と  $KSY1$  は同じパラメーターで良い

$KSY0PARAM = 5 \ 1; 10.15 \ 0; 2.5 \ 10 \ NB. \ base; xdiff; ydiff$

#### 7.14 No79 説明なし



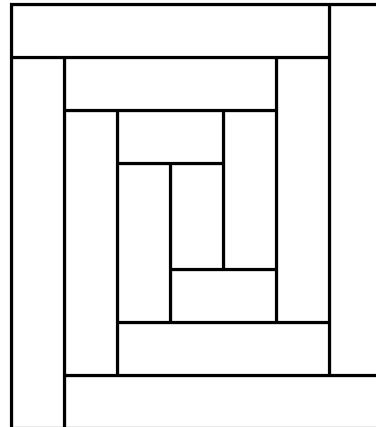
ピース 一筆書きで多少重複しても一気に描こう。タートルグラフィクスは用いない。32 ステップを要する。  
 始点は一筆書き用に右上より一つ左に取った

```
NEX0=: 7 9,1 9,1 8,7 8,7 9,8 9,8 1,1 1,1 8,7 8,7 9,7 2,8 2,2 2,:2 1
NEX0=: NEX0,2 8,2 7,6 7,6 8,6 3,7 3,3 3,:3 2
NEX0=: NEX0,3 7,3 6,5 6,5 7, 5 4,6 4,4 4,4 3,:4 6
```

```
(<./ , >./) NEX0
1 1 8 9
```

少し大きい目に窓を開く

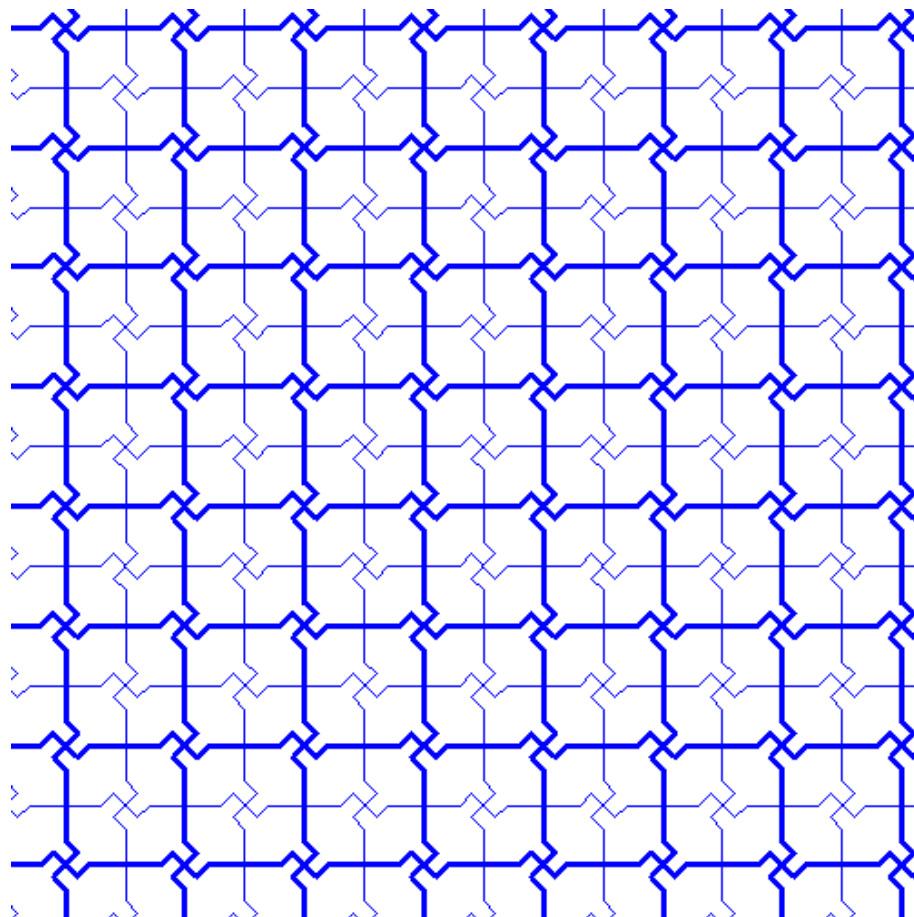
```
0 0 10 10 dwin ''
0 0 0 dline3 NEX0
```



差分 .

```
NEX0PARAM=: 7 9;7 0; 0 8
```

## 7.15 早割り碁盤割手

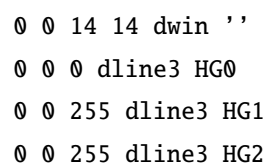


各ピース 外枠は半分でよい

HG0=: 2 2,3 1,4 2,10 2,11 3,12 2,13 3,12 4,12 10,11 11,:12 12

HG1=: 2 7,5 7,6 8,8 6,9 7,:12 7

HG2=: 7 12,7 9,8 8,6 6,7 5,:7 2



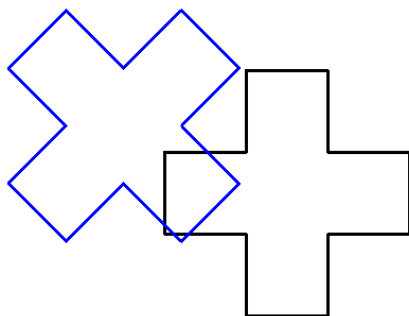
差分  $HG0PARAM=: 2\ 2;10\ 0;0\ 10$

### 7.16 三重格子 No. 58



ピース (0) 正立でポイントを取って  $\frac{1}{4}\pi$  回転させる

```
TK0=: 3 1,5 1,5 3,7 3,7 5,5 5,5 7,3 7,3 5,1 5,1 3,3 3,:3 1
TK0R=: clean }:"1 (TK0,.1) mp rotm 1r4p1 NB. rotate 1/4 pi
```

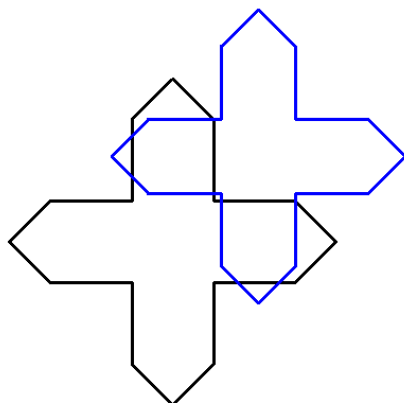


```
maxmin TK0,TK0R,TK1,TK1T
+-----+-----+
|9.7 9.7|_2.82843 0|
+-----+-----+

_3 _3 10 10 dwin ''
0 0 0 dline3 TK0
0 0 255 dline3 TK0R
```

ピース (1) *transm 2.5 2.5* で移動し微調整した

```
TK1=: 3 1,4 0,5 1,5 3,7 3,8 4,7 5,5 5,5 7,4 8,3 7,3 5,1 5,0 4,1 3,3 3,:3 1
TK1T=: clean }:"1 (TK1,.1) mp (elongm 0.9 0.9) mp transm 2.5 2.5 NB. rotate 1/4 pi
```



```
_3 _3 10 10 dwin ''
0 0 0 dline3 TK1
0 0 255 dline3 TK1T
```

ピース (3-1,3-2) 同様に回転と微調整

```
TK2=: 8 10,9 11,9 13,8 14,7 13,7 11,:8 10
TK2R=: clean }:"1 (TK2,.1) mp (rotm 1r4p1) mp transm 0.35 0.5
TK3=: 3 7,5 7,6 8,5 9,3 9,2 8,:3 7
TK3R=: clean }:"1 (TK3,.1) mp (rotm 1r4p1) NB. rotate 1/4 pi
```

差分 差分パラメーターは正立でとって、回転させた

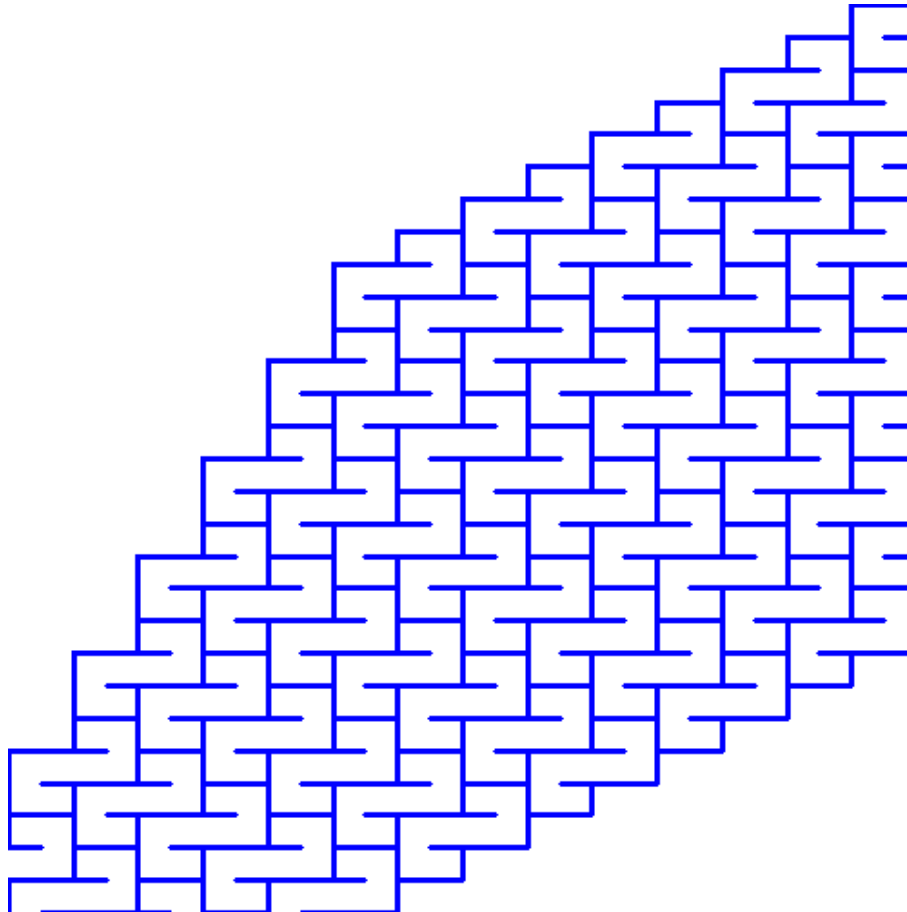


```

TK0PARAM=: 5 5 ;9 0;0 8
NB. base;xdiff;ydiff
TK0RPARAM=: } L:0 (TK0PARAM,(L:0) 1) mp L:0 rotm 1r4p1 NB. rotate 1/4 pi

```

## 7.17 ねじのふじたね No.41

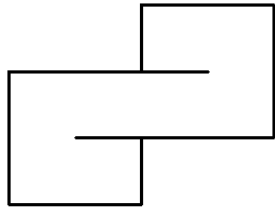


ピース 多少の重複は構わないこととした

```

NF0=: 2 1,4 1,4 2,3 2,6 2, 6 4,4 4,4 3,5 3,2 3,:2 1

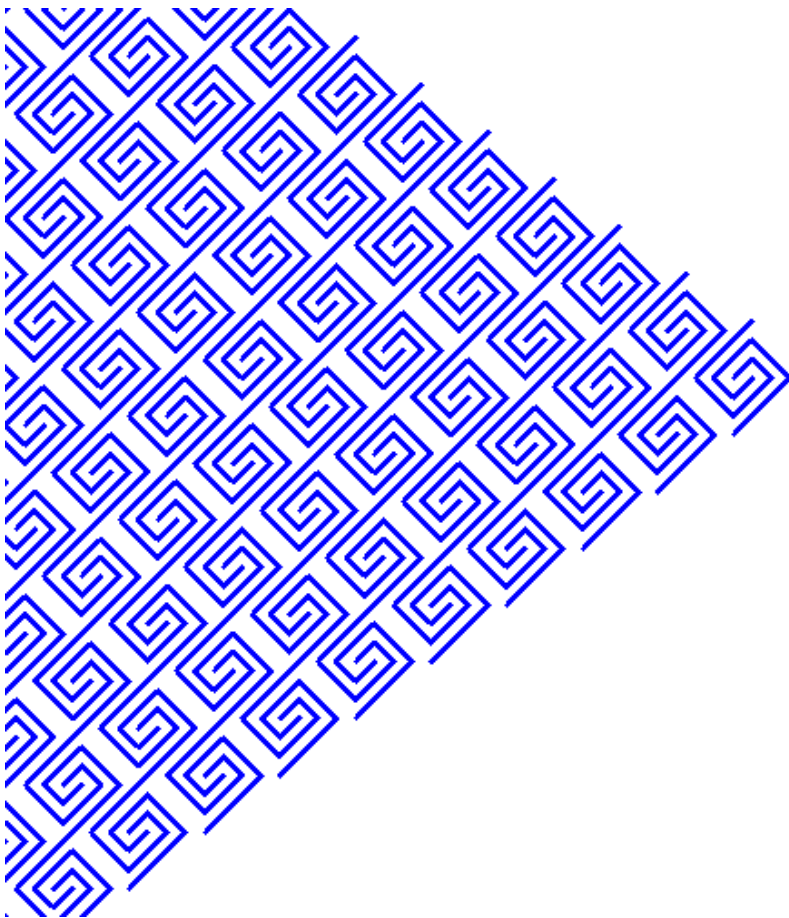
```



```
0 0 8 8 dwin ''
0 0 0 dline3 NF0
```

差分 . NF0PARAM=: 2 1;4 2; 2 3

## 7.18 釣稲妻 No.36



二つのピース  $\frac{1}{4}\pi$  回転させている

```

TI0=: 0 1,6 1,6 5,2 5,2 3,:4 3
TI0R=: clean }:"1 (TI0,.1) mp rotm 1r4p1

```

```

TI1=: 3 4,5 4,5 2,1 2,1 6,:7 6
TI1R=: clean }:"1 (TI1,.1) mp rotm 1r4p1

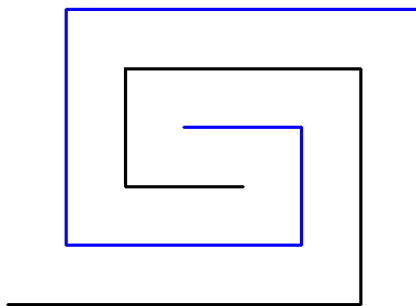
```

差分 差分パラメータも同様に回転させる

```

TI0PARAM=: 0 1;7 _1;_1 6
TI0PARAM=: }: L:0 (TI0PARAM, (L:0) 1) mp L:0 rotm 1r4p1

```

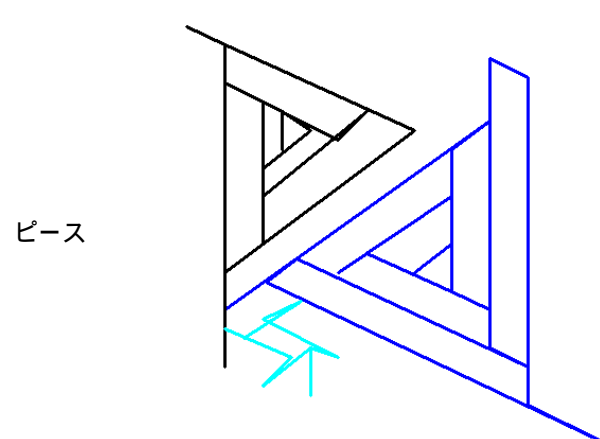
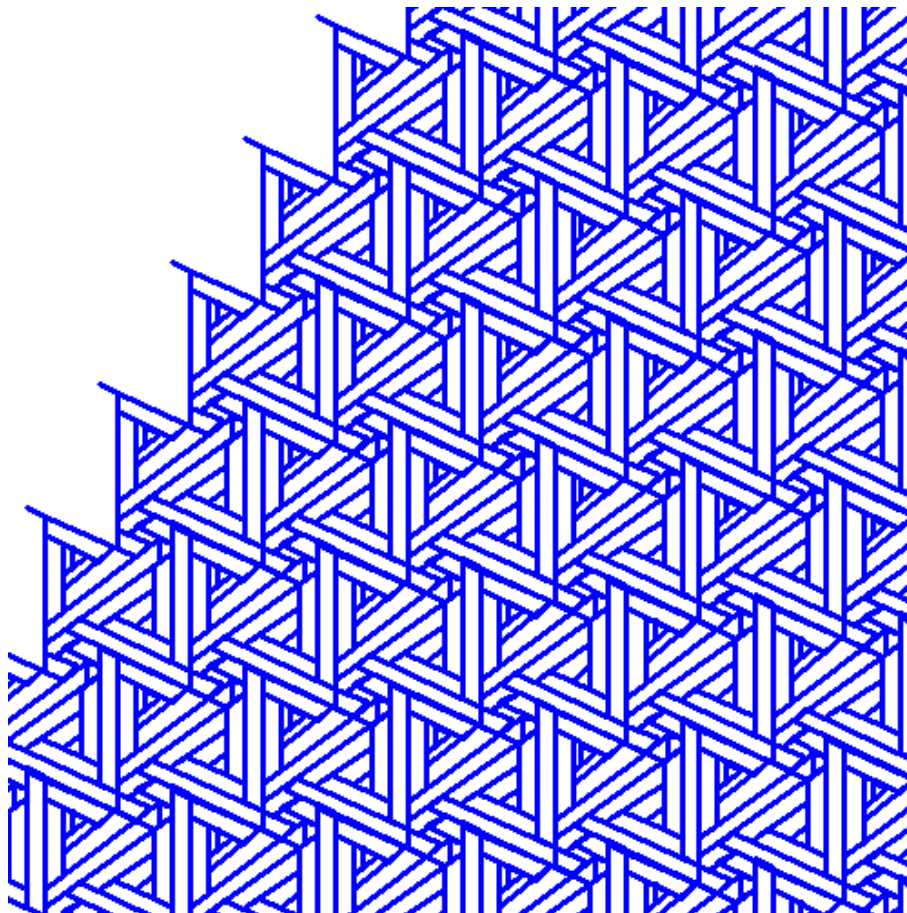


```

_1 _1 8 8 dwin''
0 0 255 dline3 TI1
0 0 0 dline3 TI0

```

## 7.19 菜籠麻の葉 No.14



```

maxmin NA0,NA1
+-----+-----+
|11 10.5|0 _0.5|
+-----+-----+

_2 _2 12 12 dwin ''
0 0 255 dline3 NA0
0 0 0 dline3 NA1
0 255 255 dline3 NA2

```

ピースのデータ 一応全体を書いた上で一本ずつ調整したが、もう少し精査が必要である。北斎先生も和紙を鋏で切り抜いて並べたのかな。

```

NB. (0)
NA0=: 1 3,8 8,8 9.65,9 9.15,9 0.5,11 _0.5,2.1 3.75,:2.9 4.35
NA0=: NA0,9 1.5,8 2,8 8,7 7.25,7 3.5,:8 3
NA0=: NA0,7 3.5,7 6 ,4 4,4.75 4.5 ,7 3.5 ,7 4.75,: 6 4
NB. (1)
NA1=: 0 10.5,1 10,1 1.5,1 4,6 7.75,1 10,:1 9
NA1=: NA1,4 7.5,4.75 8.25 ,2 6 ,2 4.75 ,2 8.5,2 6.75,:3.25 7.75
NA1=: NA1,2.5 8.25,:2.5 7.25
NB. (2) basket
NA2=: 1 2.5,2.75 1.75,2 1,3.25 2,3.25 0.75,3.25 2 ,4 1.75 ,:2 2.75
NA2=:NA2, 3 3.25,: 1.5 2.25

```

差分パラメータ *base;xdiff;ydiff* 最後の 6.75 は図上で調整した

```
NA0PARAM=: 1 3 ;8 _1; 4 6.75
```

## 8 曲線の幾何学 (1)-花卉を描く

戸川隼人「花の CG」を参照して作りかけのスク립トがあったので転用した。

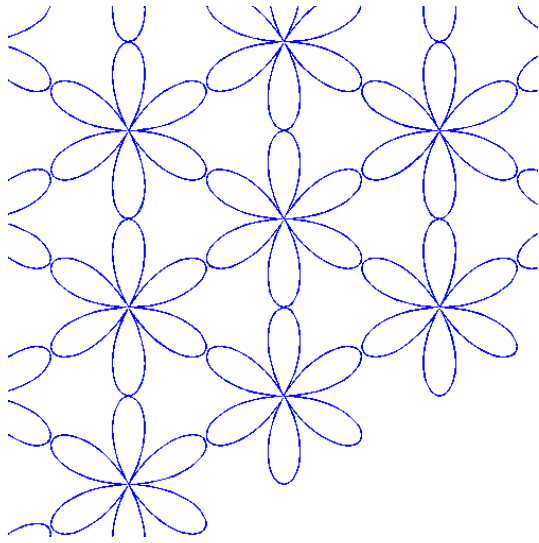
$$r = a\cos\theta + b\sin\theta$$

花の 1 度ごとの座標 (WC) は計算で得る。

```

circ_flower=: 3 : 0
NB. Usage: circ_flower 3 1
NB. petal is 2 * R0 --> 5 is 2.5 // 1 is 0.5
'R0 Beki'=. y NB. 0.07 is Scale parameter
R=. Beki ^~| sin R0 * T=. 0.07 * i. 360
R polar_xy T
)
polar_xy=: 4 : ' |: ;("1),.x * L:0 (cos;sin)L:0 y' NB. y is T

```

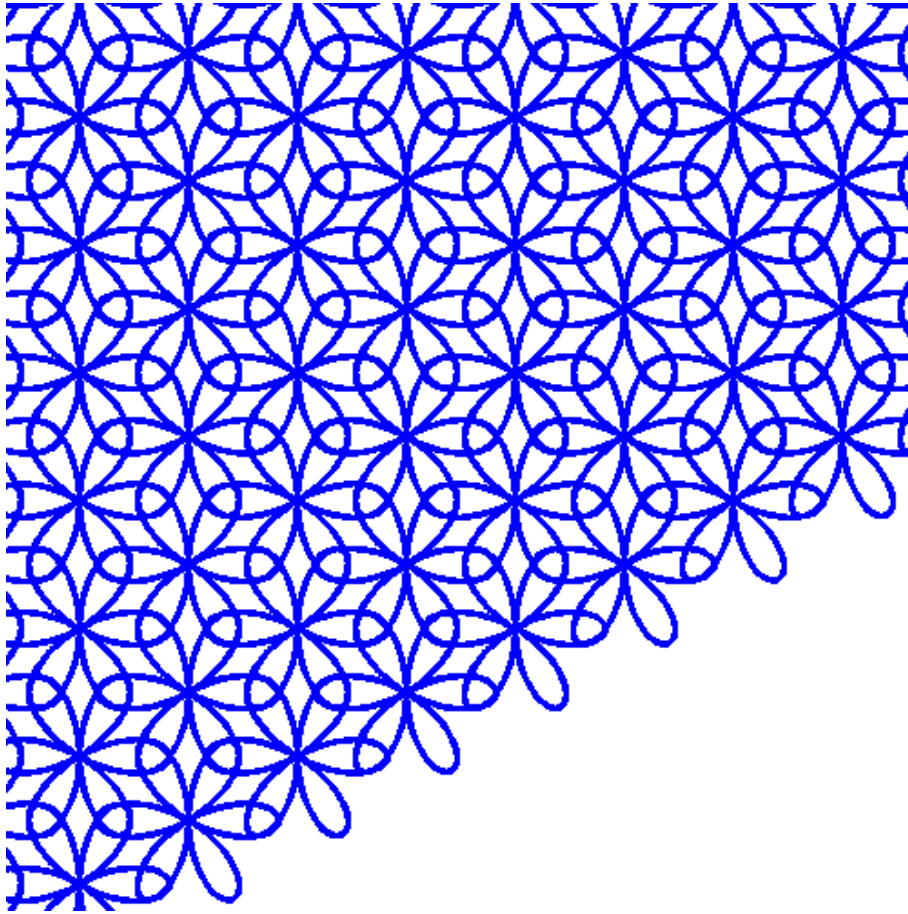


## 8.1 輪違い麻の葉 No.08

麻の葉 6 弁の花で描く

花の回転 北斎の 6 弁の花は六角形でいえば頂点が上には来ない。上の図を  $\frac{1}{6}\pi$  回転させる  
座標の回転はシリーズ (1) で作成した。

```
((circ_flower 3 1),.1) mp rotm 1r6p1
```



マイナス座標を変換  $xy$  の最小値の絶対値を加えて全ての座標をプラスに変換する  
 小紋の差分パラメータ (WCPARAM) 花の座標の  $xy$  の最大値を基に手で微調整した  
 Script(1) 計算部分

```
calc_hokusai_wc=: 3 : 0
NB. u ''
tmp=. circ_flower 3 1                      NB. 6 petal flower
tmp=. }:"1 (tmp,.1) mp rotn 1r6p1          NB. rotate 1/3 pi
Min=. <./ Maxmin=. minmax_dwin tmp
WC=. tmp + - Min                           NB. adjust to all plus
'p1 p2'=. 2}. Maxmin                       NB. find WCPARAM
WCPARAM=. (0 0);((0.6*+:p1),0.8*p2);0,0.8*+:p2
(<WC),<WCPARAM
)
```

グラデュエーション 淡色ではのっぺりするため隣接する 2 色とした。draw\_dpoly\_grad を *line* 用に改良した

```

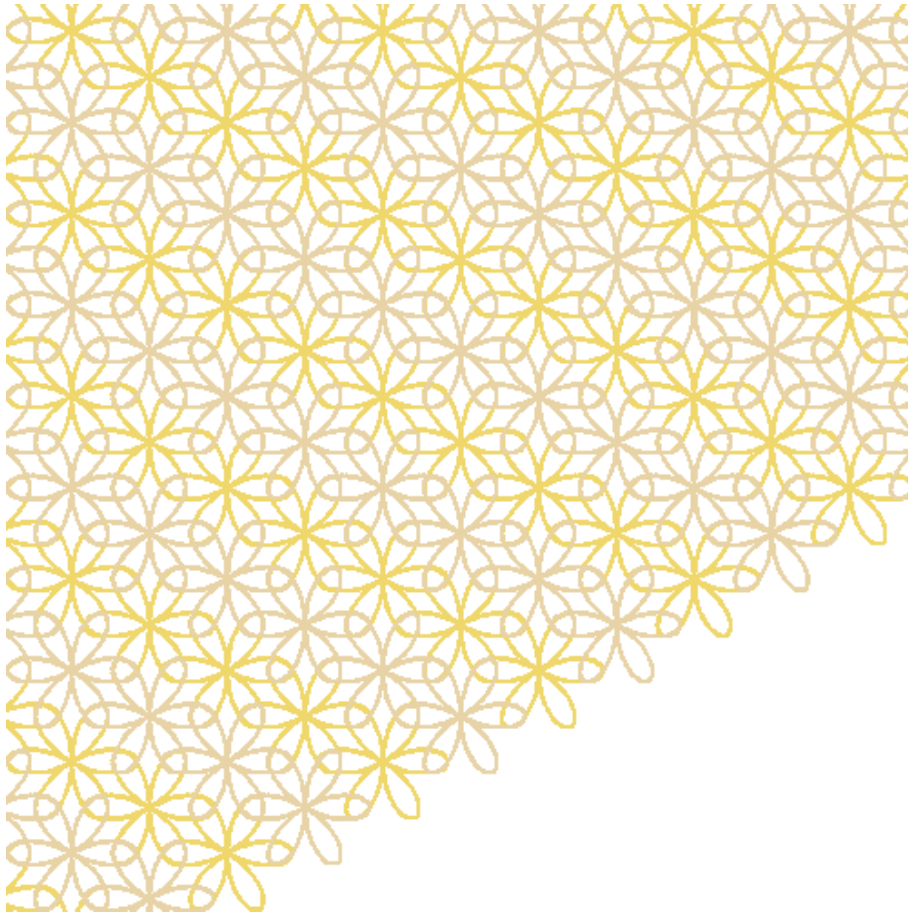
hokusai_wc2=: 4 : 0
NB. depict with 2 colors
NB. (234 213 107;228 211 162;16 17) hokusai_wc2 ''
'Color0 Color1 Size'= . x
((Color0;Color1);<Size) draw_dline_grad calc_hokusai_wc ''
)

```

```

NB. (234 213 107;228 211 162;16 17) hokusai_wc2 ''

```



## 9 小紋のテクニック ( 5 ) ベジエ曲線

コンピュータで曲線を描くには数式による幾何曲線やツールでマウスやペンを用いたドローイングと並んでエンジニアリング由来のスプライン曲線やベジエ曲線もよく用いられている。マトリクスを用いるとベジエ曲線は実にシンプルなスクリプトで表現でき、難解な数式や長いプログラムなしで自由にベジエ曲線を扱うことができる。 *simple is beautiful!*

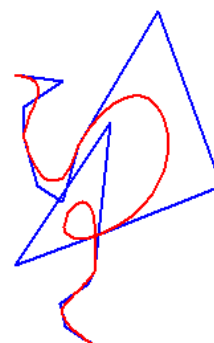
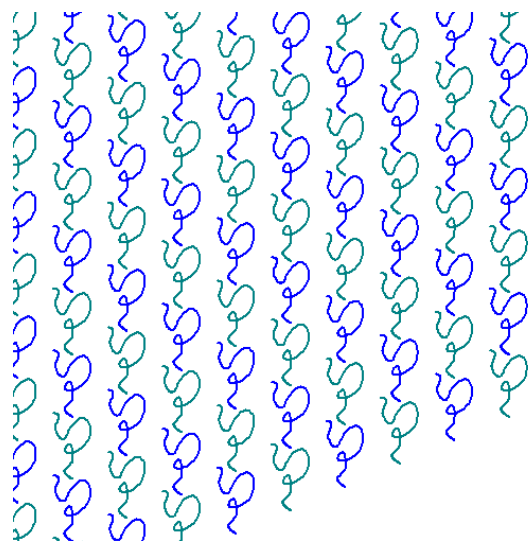
次の左の図は北斎が「新形小紋帳」でデザインしたもので、右のような図を並べ重ねて作成する。数学の幾何曲線では図形は限定されるので、もっと自由に図形を描け、フォントデザインや CAD に用いられてい



るスプライン曲線やベジエ曲線を用いることになる。スプライン曲線は *Truetype* フォントに、ベジエ曲線は *Postscript* フォントに用いられている。操り人形でいえばスプラインは手繰り、ベジエ曲線は糸繰りの様であるがベジエ曲線のほうが自由度が高いと思われる。

ベジエ曲線 (*Cubic Bezier Curve*) の計算は次の 4 行=4 つの関数にまとめることができた。plot を入れても 10 行足らずである。

```
1. mat_bezier4=: 1 0 0 0, _3 3 0 0, 3 _6 3 0, :_1 3 _3 1 NB.Cubic BezierMatrixForm
2. calc_bezier4=: 3 : '(: mat_bezier4 +/ . * y)&p. " 0 steps 0 1 20'
3. form_bezier =: 4 : '({: tmp), L:0 }.{. L:0 tmp=.(- <: x)<\ y'
4. calculus_bezier4 =: [: ;("2)@,. calc_bezier4(L:0)@form_bezier4
```



## 9.1 ベジエ曲線 (マトリクスフォーム)

スプライン曲線は第 2 次大戦後航空機産業で用いられ始めた。ベジエ曲線はシトロエン社のドン・カステイヨ、ルノー社のベジエが開発したが、直ぐには社外には出なかったようだ。Pierre Bezier(1910-1999) は生粋のバリっ子のエンジニアでルノー社に 42 年在籍した。

ベジエ曲線はマトリクスフォームを用いると簡潔に表すことができる。マトリクスフォームを文献 (1) により提示してみよう。

## 9.2 Cubic Bezier Curve

4 点のポイントから 3 次多項式を作成するので *Cubic* と呼ばれる。本稿ではポイント数で 4 として取り扱う。

コントロールポイント 4 点からなる。P<sub>0</sub>, P<sub>3</sub> は固定される。

$$\text{ControlPoints: } \begin{array}{l} P_0 = [x_0, y_0] \\ P_1 = [x_1, y_1] \\ P_2 = [x_2, y_2] \\ P_3 = [x_3, y_3] \end{array}$$

ベルンシュタインの式 .  
\*2

$$J_{n,i} = \binom{3}{i} t^i (1-t)^{3-i}$$

$t$  は媒介変数で  $[0, 1]$  の間の値をとる。パスカルの 3 角形が現れる。

$$\text{BernsteinCubics: } \begin{array}{l} B_0(t) = (1-t)^3 \\ B_1(t) = 3(1-t)^2t \\ B_2(t) = 3(1-t)t^2 \\ B_3(t) = t^3 \end{array}$$

ベジエの公式の導出 .

$$\text{Bezier}(t) = B_0(t)P_0 + B_1(t)P_1 + B_2(t)P_2 + B_3(t)P_3$$

$$\text{Bezier}(t) = (1-t)^3[x_0, y_0] + 3(1-t)^2t[x_1, y_1] + 3(1-t)t^2[x_2, y_2] + t^3[x_3, y_3]$$

$$\text{Bezier}(t) = [x(t), y(t)]$$

$$x(t) = x_3t^3 + 3x_2t^2 - 3x_2t^3 + 3x_1t - 6x_1t^2 + 3x_1t^3 + x_0 - 3x_0t + 3x_0t^2 - x_0t^3$$

$$x(t) = x_0 + (-3x_0 + 3x_1)t + (3x_0 - 6x_1 + 3x_2)t^2 + (-x_0 + 3x_1 - 3x_2 + x_3)t^3$$

ベジエのマトリクスフォーム .

$$\text{BezierMatrixForm: } \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = AX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_0 \\ -3x_0 + 3x_1 \\ 3x_0 - 6x_1 + 3x_2 \\ -x_0 + 3x_1 - 3x_2 + x_3 \end{pmatrix}$$

$$x(t) = c_0 + c_1t + c_2t^2 + c_3t^3$$

$$y(t) = y_3t^3 + 3y_2t^2 - 3y_2t^3 + 3y_1t - 6y_1t^2 + 3y_1t^3 + y_0 - 3y_0t + 3y_0t^2 - y_0t^3$$

$$y(t) = y_0 + (-3y_0 + 3y_1)t + (3y_0 - 6y_1 + 3y_2)t^2 + (-y_0 + 3y_1 - 3y_2 + y_3)t^3$$

---

\*2 Sergei Bernstein ユダヤ系ロシア人の数学者。バリで数学教育を受けロシアで教えた

$$\text{BezierMatrixForm} : \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = AY = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} y_0 \\ -3y_0 + 3y_1 \\ 3y_0 - 6y_1 + 3y_2 \\ -y_0 + 3y_1 - 3y_2 + y_3 \end{pmatrix}$$

$$y(t) = d_0 + d_1 t + d_2 t^2 + d_3 t^3$$

ベジエの多項式と係数 .

$$\text{Bezier}(t) = [x(t), y(t)] = [c_0 + c_1 t + c_2 t^2 + c_3 t^3, d_0 + d_1 t + d_2 t^2 + d_3 t^3]$$

\*3

### 9.2.1 数値例と経過

$J$  で描くからには簡潔なマトリクスフォームで直ちに多項式の係数を得て、 $p.$  で多項式の値 (ベジエ曲線) を求め、グラフィックスに渡すようにする

1. Example 1  $P_0, P_1, P_2, P_3$

---

\*3

- Bernstein の公式 (3 次式) からマトリクスの係数を求める

$$J_{n,i} = \binom{3}{i} t^i (1-t)^{3-i}$$

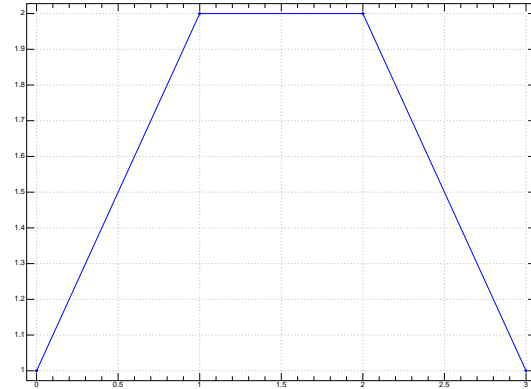
$$\begin{aligned} J_{3,0}(t) &= (1-t)^3 &= 1 - 3t + 3t^2 - t^3 \\ J_{3,1}(t) &= 3t(1-t)^2 &= 3t - 6t^2 + 3t^3 \\ J_{3,2}(t) &= 3t^2(1-t) &= 3t^2 - 3t^3 \\ J_{3,3}(t) &= t^3 \end{aligned}$$

- ベジエマトリクスフォーム (3 次=4 ポイント)。J 流に構成している

$$\text{BezierMatrixForm} : \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = AX = \begin{pmatrix} t^3 & t^2 & t & 1 \\ 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_0 \\ -3x_0 + 3x_1 \\ 3x_0 - 6x_1 + 3x_2 \\ -x_0 + 3x_1 - 3x_2 + x_3 \end{pmatrix}$$

```
'line, marker' plot {@|: L0
pd 'eps c:/temp/Bezier_L0base.eps'
```

```
L0
x y
0 1
1 2
2 2
3 1
```



## 2. Bezier Matrix

```
mat_bezier4=: 1 0 0 0, _3 3 0 0, 3 _6 3 0, _1 3 _3 1 NB.Cubic BezierMatrixForm
```

```
mat_bezier4
1 0 0 0
_3 3 0 0
3 _6 3 0
_1 3 _3 1
```

## 3. 内積演算 $x, y$ を同時に行う

```
mat_bezier4 +/ . * L0
```

```
x(t),y(t)
0 1
3 3
0 _3
0 0
```

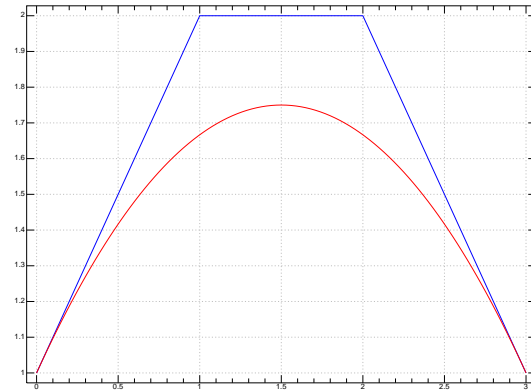
$$\begin{aligned} x(t) &= 3t \\ y(t) &= 1 + 3t - 3t^2 \end{aligned}$$

## 4. $x(t), y(t)$ の多項式を作成し $(x, y)$ での値を計算する。

$J$  の  $p$ . を用いる。3 次多項式になる。 $t$  は区間を  $[0, 1]$  として  $steps$  で細分する。

(0 0 3 &p. ,. 1 3 \_3&p. )steps 0 1 10

0	1
0.03	1.27
0.12	1.48
0.27	1.63
0.48	1.72
0.75	1.75
1.08	1.72
1.47	1.63
1.92	1.48
2.43	1.27
3	1



##### 5. Bezier 曲線の特徴

- $P_0, P_1, P_2, P_3$  のうち両端の  $P_0, P_2$  は固定  
中間の  $P_1, P_2$  はコントロールポイント (=操り人形の紐) として用いる
- $P_0, P_1$  をベクトルと見て中間点を取る。  $P_A$   
 $P_1, P_2$  の中間点を取る。  $P_B$   
 $P_2, P_3$  もベクトルと見て中間点を取る。  $P_C$
- $P_A$  と  $P_B$  を結びその中間点  $P_D$  をとる。  
 $P_B$  と  $P_C$  を結びその中間点  $P_E$  をとる。  
 $P_D$  と  $P_E$  を結びその中間点  $P_F$  をとるとこの  $F$  がアーチの頂上となる。(概ね 7 割り強)

コントロールの  $P_1, P_2$  は実例を沢山描き勘で覚える必要がある

- Example 2

L1

0 1

1 3

2 4

4 2

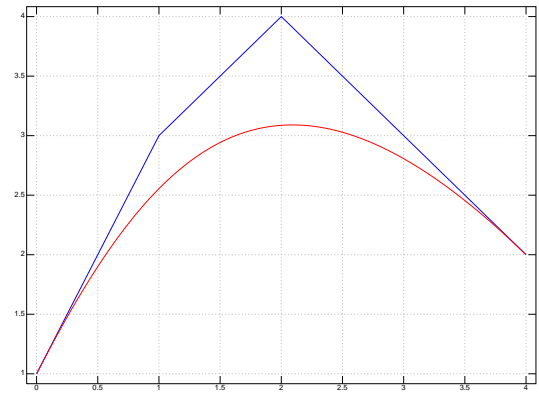
mat\_bezier4 +/ . \* L1

0 1

3 6

0 \_3

1 \_2



$$x(t) = 3t + t^3$$

$$y(t) = 1 + 6t - 3t^2 - 2t^3$$

- *Example 3*

L2

0 1

3 1

2 4

4 2

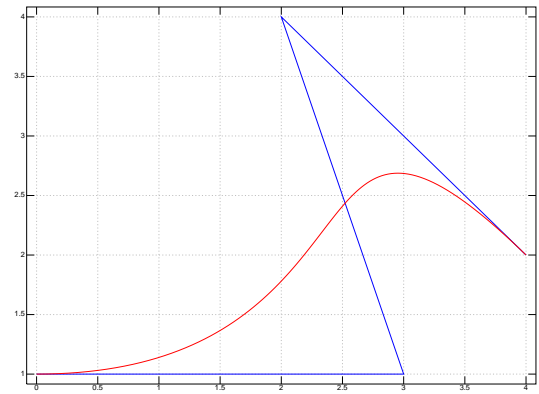
mat\_bezier4 +/ . \* L2

0 1

9 0

\_12 9

7 \_8



$$x(t) = 9t - 12t^2 + 7t^3$$

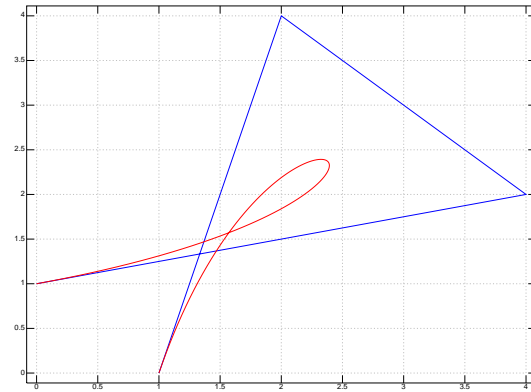
$$y(t) = 1 + 9t^2 - 8t^3$$

- *Example 4*

L3

```
0 1
4 2
2 4
1 0
```

```
mat_bezier4 +/ . * L3
0 1
12 3
_18 3
7 _7
```



$$x(t) = 12t - 18t^2 + 7t^3$$

$$y(t) = 1 + 3t + 3t^3 - 7t^3$$

### 9.2.2 Script

*Cubic Bezier* のコアは次の数行で簡潔に記述できる

```
mat_bezier4=:1 0 0 0,_3 3 0 0,3 _6 3 0,:_1 3 _3 1 NB.Cubic BezierMatrixForm
```

丁寧に計算過程の *Script* を書き下すと次のようになる。

```
calc_bezier4p1=: 3 : 0
tmp0=.: mat_bezier4 +/ . * y
fx=. ({. tmp0)&p.
fy=. ({: tmp0)&p.
fx0 =. fx steps 0 1 20 NB. divide [0,1] for smooth Bezier curves
fy0 =. fy steps 0 1 20
fx0,.fy0
)
```

これは次のように簡潔にまとめる事ができる。  $f(x)$ ,  $f(y)$  は同時計算。

```
calc_bezier4=: 3 : '({: mat_bezier4 +/ . * y)&p. " 0 steps 0 1 20'
```

### 9.2.3 J Grammar

- ラミネート 最後のリストの前に (,:) を置き、横に積み重ねる。2×4 の行列にする  

$$\begin{pmatrix} 3 & -6 & 3 & 0 \\ 1 & 3 & -3 & 1 \end{pmatrix}$$
- +/ . \* 内積演算
- |: *transpose* 縦横を転置する。多項式の係数を横行列にする

```
|: mat_bezier4 +/ . * L0
0 3 0 0      NB.  fx= 3t
1 3 -3 0     NB.  fy=1+3t-3t^2
```

- p. 両項は多項式を当てはめ 右引数 ( $t$ ) ごとの値を計算する。fxy で  $f(x)$ ,  $f(y)$  を同時に計算する
- steps 区間 [0 1] を  $n$  分割する。numeric.ijs に入っている関数

```
(|: mat_bezier4 +/ . * L0) &p."0 steps 0 1 10
```

```
0      1
0.3 1.27
0.6 1.48
0.9 1.63
1.2 1.72
1.5 1.75
1.8 1.72
2.1 1.63
2.4 1.48
2.7 1.27
3      1
```

- ランク " 1 0 は最初にベクトル方向、即ち  $fx$  を、次に  $fy$  を計算するように指定。

## 9.3 小紋の作図データの構成

1. ベジエの計算に与えるデータは方眼紙に描いた図から最初に括弧の  $P_0, P_3, P_6, P_9, \dots$  を採る。  
4 ポイントずつのデータを 1 個ずつボックスでオーバーラップさせて構成する。各ボックスのデータの最初と最後の 2 個は固定される。入力はオーバーラップ分は除く 各象限は 7 個ずつのデータで構成さ

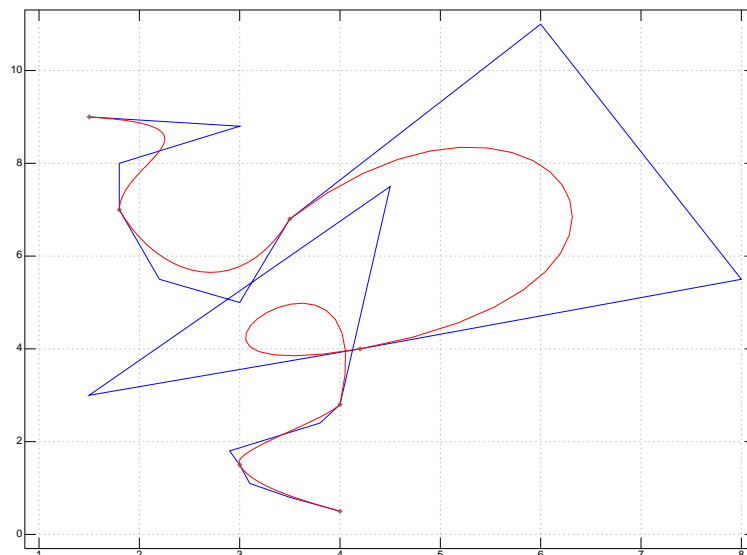


れる

```

RANOJI0
(1.5 9) NB. P0
3 8.8
1.8 8
(1.8 7) NB. P3
2.2 5.5
3 5
(3.5 6.8) NB. P6
6 11
8 5.5
(4.2 4) NB. P9
1.5 3
4.5 7.5
( 4 2.8)
3.8 2.4
2.9 1.8
( 3 1.5)
3.1 1.1
3.5 0.8
( 4 0.5)

```



2.  $J$  に入力する。読みやすいように固定ポイントは括弧で囲む。

$RANOJI0$  はボックスにしたときにプログラムでオーバーラップさせるので、入力ではオーバーラップさせる必要はない。

NB. Hokusai No29 らの字

```

RANOJI0=: (1.5 9),3 8.8 , 1.8 8 ,(1.8 7),2.2 5.5,3 5 ,:(3.5 6.8) NB. 0/1/2
RANOJI0=: RANOJI0, 6 11,8 5.5 ,(4.2 4),1.5 3 ,4.5 7.5 ,:(4 2.8) NB. (2)/3/4/5
RANOJI0=: RANOJI0, 3.8 2.4 ,2.9 1.8 ,(3 1.5), 3.1 1.1 , 3.5 0.8,:(4 0.5) NB. 6/7

```

3.  $J$  の *calculus\_bezier4* に与える 4 ポイントごとのデータを作成する。最下行と最上行はオーバーラップさせる

```

4 form_bezier RANOJI0
+-----+-----+-----+-----+-----+-----+
|1.5 9|1.8 7|3.5 6.8|4.2 4| 4 2.8| 3 1.5|
| 3 8.8|2.2 5.5| 6 11|1.5 3|3.8 2.4|3.1 1.1|
|1.8 8| 3 5| 8 5.5|4.5 7.5|2.9 1.8|3.5 0.8|

```

```
|1.8 7|3.5 6.8|4.2 4| 4 2.8| 3 1.5| 4 0.5|
+-----+-----+-----+-----+-----+-----+
```

#### 4. *form\_bezier* の Script

```
form_bezier =: 4 : '({: tmp), L:0 }.{. L:0 tmp=(- <: x)<\ y'
```

*J* のボックスと *infix* 機能を用いてオーバーラップ無し ( $-3<\backslash$ ) の次の配列を作成し、オーバーラップを付加する

```
_3<\i. 20 2
+---+---+---+---+---+---+---+
|0 1| 6 7|12 13|18 19|24 25|30 31|36 37|
|2 3| 8 9|14 15|20 21|26 27|32 33|38 39|
|4 5|10 11|16 17|22 23|28 29|34 35|    |
+---+---+---+---+---+---+---+
```

#### 5. *Cubic Bezier* の計算。4 点計算の基本形とロングデータ用

```
calculus_bezier4 RAN0JI0
```

```
calc_bezier4=: 3 : '(: mat_bezier4 +/ . * y)&p. " 0 steps 0 1 20' NB. basement
```

```
calculus_bezier4=: 3 : ';"1) calc_bezier4 (L:0) 4 form_bezier y' NB. long data
```

## 9.4 Cubic 以外のポイント数のベジエ曲線

### 9.4.1 3 ポイント (2 次) のベジエマトリクス

あまり複雑でない図形は 3 次のベジエマトリクスで対応できる。

*Bernstein* の公式から 2 次式を求める

$$\begin{aligned} (1-t)^2 &= 1-2t+t^2 \\ 2t(1-t) &= 2t-2t^2 \\ t^2 & \end{aligned}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

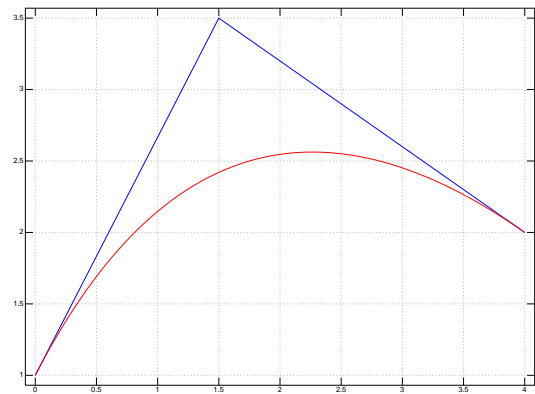
```
mat_bezier3
```

```
1 0 0
```

```
_2 2 0
1 _2 1
```

```
3 plot_bezier0 0 1,1.5 3.5,: 4 2
```

```
pd 'eps c:/temp/besier_t3.eps'
```



#### 9.4.2 4次 (5 ポイント) のベジエ曲線

##### 1. ベルンシュタインの公式

$$\begin{aligned}
 (1-t)^4 &\rightarrow 1 - 4t + 6t^2 - 4t^3 + t^4 \\
 4t(1-t)^3 &\rightarrow 4t - 12t^2 + 12t^3 - t^4 \\
 6t^2(1-t)^2 &\rightarrow 6t^2 - 12t^3 + 6t^4 \\
 4t^3(1-t) &\rightarrow 4t^3 - 4t^4 \\
 t^4 &
 \end{aligned}$$

##### 2. Bernstein Formura を直接に $t$ を先に計算する方法

```
b5_0=:(^&4@-.);(4&* * ^&3@-.);(*&6@^&2 * ^&2@-.);(*&4@^&3 * -.);^&4
      (1-t)^4 4t(1-t)^3 6t^2(1-t)^2 4t^3(1-t) t^4
```

##### 3. 4 次のマトリクスフォーム

```
mat_bezier5
1 0 0 0 0
_4 4 0 0 0
6 _12 6 0 0
_4 12 _12 4 0
1 _4 6 _4 1
```

##### 4. 2 の方法の比較

```
(b5 L0,4 0),. (|: mat_bezier5 +/ . * L0, 4 0)&p. " 0 steps 0 1 10
0 1 0 1 NB. P0
0.4 1.3401 0.4 1.3401
0.8 1.5616 0.8 1.5616
```

```

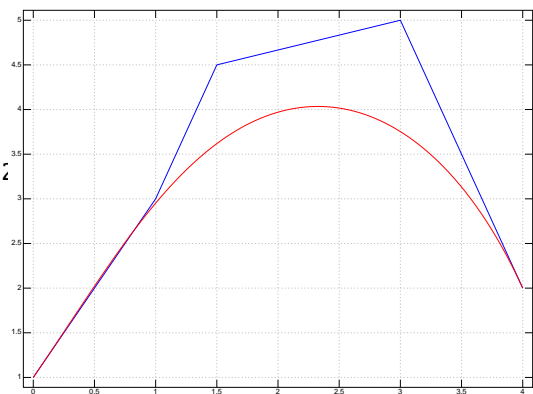
1.2 1.6681    1.2 1.6681
1.6 1.6656    1.6 1.6656
   2 1.5625    2    1.5625
2.4 1.3696    2.4 1.3696
2.8 1.1001    2.8 1.1001
3.2 0.7696    3.2 0.7696
3.6 0.3961    3.6 0.3961
   4    0     4    0    NB. P3

```

```

5 plot_bezier0 0 1, 1 3, 1.5 4.5, 3 5, : 4 2
pd 'eps c:/temp/besier_t5.eps'

```



#### 9.4.3 2ポイント(1次式)のベジエ

2ポイントは直線になる。わざわざ直線をベジエで書くのは、直線区間を一筆書きで描くのに都合が良いからである。

##### 1. ベルンシュタインの式

$$(1-t) + t = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} t \\ 1 \end{pmatrix}$$

##### 2. マトリクス フォーム

```

mat_bezier2
1 0
_1 1

```

#### 9.4.4 5次(6ポイント)のベジエ曲線

*Bernstein* の公式と加藤の三角錐の美しい内部構造が明らかになったところで5次のベジエのマトリクスフォームを作成してみよう。

##### 1. ベルンシュタインの公式

$$\begin{array}{rcl}
(1-t)^5 & \rightarrow & 1 - 5t + 10t^2 - 10t^3 + 5t^4 - t^5 \\
5t(1-t)^4 & \rightarrow & 5t - 20t^2 + 30t^3 - 20t^4 + 5t^5 \\
10t^2(1-t)^3 & \rightarrow & 10t^2 - 30t^3 + 30t^4 - 10t^5 \\
10t^3(1-t)^2 & \rightarrow & 10t^3 - 20t^4 + 10t^5 \\
5t^4(1-t) & \rightarrow & 5t^4 - 5t^5 \\
t^5 & & 
\end{array}$$

## 2. 5 次のマトリクスフォーム

```

mat_bezier6
1   0   0   0   0   0
_5   5   0   0   0   0
10 _20  10   0   0   0
_10  30 _30  10   0   0
 5 _20  30 _20  5   0
_1   5 _10  10 _5  1

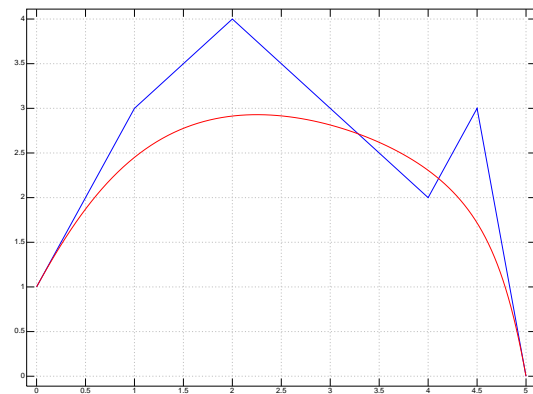
```

## 3. plot

```

6 plot_bezier0 L1,4.5 3,:5 0
pd 'eps c:/temp/besier6t.eps'

```



## 9.4.5 色々な次数の Script

### 1. マトリクスフォーム (1,2,3,4,5 次)

```

NB. ---0.----BezierMatrixForm-----
mat_bezier4=: 1 0 0 0, _3 3 0 0, 3 _6 3 0, : _1 3 _3 1 NB.Cubic BezierMatrixForm
mat_bezier3=: 1 0 0 , _2 2 0, : 1 _2 1
mat_bezier2=: 1 0, : _1 1 NB. bezier2 is linear & divide[0 1] by steps
NB. -----5/6 is Kato's triangle-----
mat_bezier5=: 1 0 0 0 0 , _4 4 0 0 0, 6 _12 6 0 0 , _4 12 _12 4 0, : 1 _4 6 _4 1
mat_bezier6=: 1 0 0 0 0 0, _5 5 0 0 0 0 , 10 _20 10 0 0 0 , : _10 30 _30 10 0 0
mat_bezier6=: mat_bezier6, 5 _20 30 _20 5 0, : _1 5 _10 10 _5 1

```

## 2. 基本のベジエスクリプト

```
NB. ----1 calc single 2/3/4/5/6 points bezier-----
NB. midify steps yourself e.g. stepd (0 1), 100
calc_bezier4=: 3 : '(: mat_bezier4 +/ . * y)&p. " 0 steps 0 1 20'
calc_bezier3=: 3 : '(: mat_bezier3 +/ . * y)&p. " 0 steps 0 1 20'
calc_bezier2=: 3 : '(: mat_bezier2 +/ . * y)&p. " 0 steps 0 1 20'
calc_bezier5=: 3 : '(: mat_bezier5 +/ . * y)&p. " 0 steps 0 1 20'
calc_bezier6=: 3 : '(: mat_bezier6 +/ . * y)&p. " 0 steps 0 1 20'
```

## 3. データフォーム付きの汎用のベジエスクリプト

```
NB. --2---main calc Bezier long data-----
calculus_bezier4=: 3 : ';"1) calc_bezier4 (L:0) 4 form_bezier y'
calculus_bezier3=: 3 : ';"1) calc_bezier3 (L:0) 3 form_bezier y'
calculus_bezier2=: 3 : ';"1) calc_bezier2 (L:0) 2 form_bezier y'
calculus_bezier5=: 3 : ';"1) calc_bezier5 (L:0) 5 form_bezier y'
calculus_bezier6=: 3 : ';"1) calc_bezier6 (L:0) 6 form_bezier y'
```

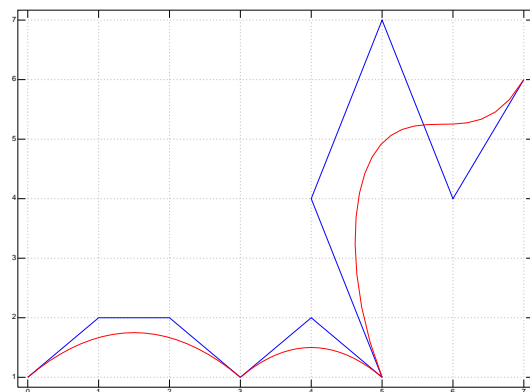
### 9.4.6 異なる次数ベジエ曲線の混合

ベジエ曲線は基本は *Cubic*(4 ポイント) で描くが、図形の途中で次数を変えたり、直線を用いたりすることがある。一筆書きが切れると、別のラインで描画関数も別に指定しなければならないのでなるべく一筆書きにしたい場合に *mixed form* が便利である

• .

```
L4=: L0,4 2,5 1,4 4, 5 7,6 4,:7 6
```

```
4 3 5 plot_bezier_mix L4
```



- ミックスフォームの作成

```

4 3 5 form_bezier_mix L4
+---+---+---+
|0 1|3 1|5 1|
|1 2|4 2|4 4|
|2 2|5 1|5 7|
|3 1|   |6 4|
|   |   |7 6|
+---+---+---+

```

- *mix\_form* 用の関数一覧

1. *form\_bezier\_mix*
2. *calc\_bezier\_mix*
3. *calculus\_bezier\_mix*
4. *plot\_bezier\_mix*

## 10 ベジエ曲線で描く

### 10.1 松波 No.44

松葉をベジエで描く *calc\_besier4 MTNM0* 4 点

曲線はデリケートなのでポイントの座標拾うのに次の方法がある。

- 方眼紙に手で写す
- 北斎の図案をスキャナで読み込んで 1 ミリ方眼の用紙にプリントする。

\*4

4 点を左から  $P_0, P_1, P_2, P_3$  とする。両端の  $P_0, P_3$  は固定点で原図より拾う。 $P_1, P_2$  で図形をコントロールする。松葉ではベジエの練習として  $P_1, P_2$  の  $x$  は随分両サイド寄りにして背も中頃にしたが、図はかなり尖っている。

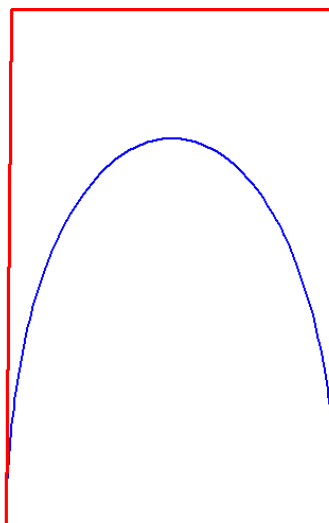
キュービックでは  $P_3$  は次の固定点になり、 $P_4, P_5$  がコントロール、 $P_6$  が固定点になる。

```
0 0 255 draw_dline0 calculus_besier4 MTNM0
255 0 0 dline3 MTNM0
```

```
MTNM0 NB. 松葉本体
x y
3 1 NB. P0 fix
3.1 12 NB. P1
8.9 12 NB. P2
9 1 NB. P3 fix (next P0)
```

$P_0 - P_3$  は右の図の赤い線

*calc\_besier4 MTNM0* で 100 点ずつの  $x, y$  を計算する



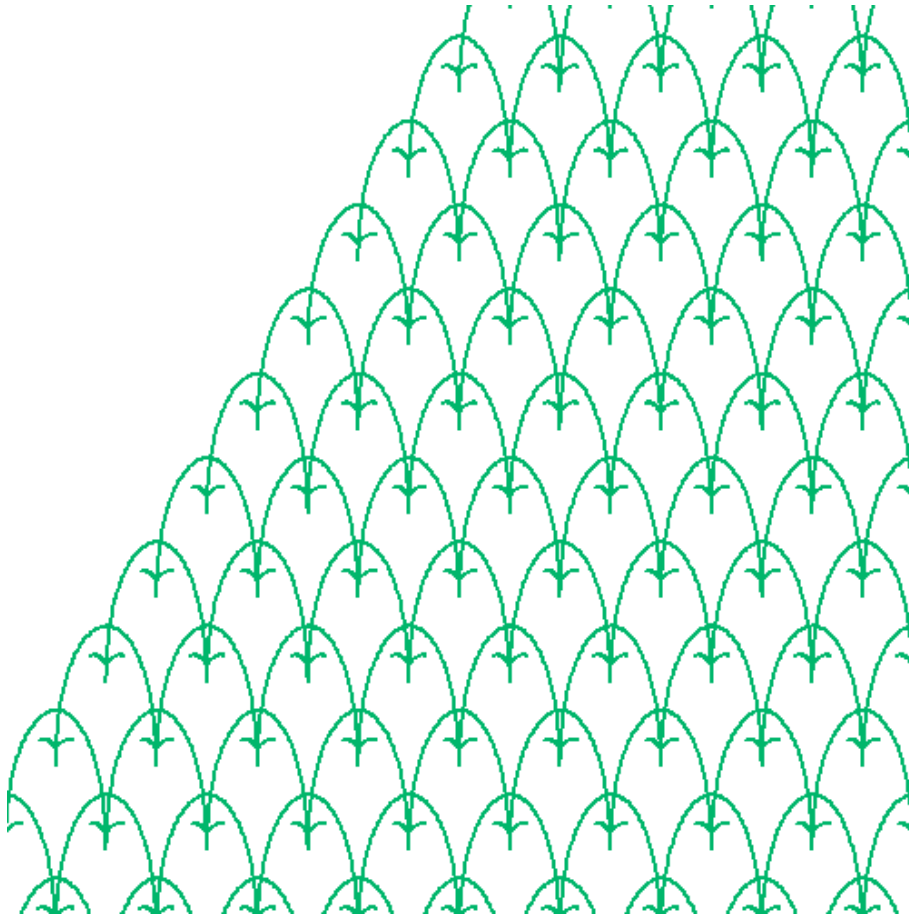
松葉の髭 *calc\_besier3 MTNM1* 松葉の髭は簡略な 3 点で描く

```
MTNM1 NB. 松葉の髭
x y
2 2.5 NB. P0
2.5 2.7 NB. P1
3 2 NB. P2=P0
3.5 2.7 NB. P1
4 2.5 NB. P2
```

\*4 或いはイラストレーターや CAD ツールを使う方法もあるが、筆者は持っていない。



松波 北斎の原画にもっと近づけるには両端に直線の下駄を履かせ、曲線を丸くすることになるが、この図も中々クールだ。



Script .

```
NB. -----
NB. 44-松並
NB. Matsunami
MTNM0=: 3 1,3.1 12,8.9 12,:9 1
MTNM1=: 2 2.5,2.5 2.7,3 2,3.5 2.7,:4 2.5 NB. Hige left
MTNM0PARAM=: 3 1; 6 0;3 5
NB. -----

hokusai_matsunami=: 4 : 0
NB. 0 181 110 hokusai_matsunami 15 15
Color0=. x ] Size=. y
tmp01=. calculus_besier4 MTNM0
tmp11=. calculus_besier3 MTNM1
```

```

NB. -----
tmp0=. Size calc_each_poly (< tmp01); <MTNM0PARAM
tmp1=. Size calc_each_poly (< tmp11); <MTNM0PARAM
(find_center tmp0) dwin ''
Color0 dline2 L:0 tmp0
Color0 dline2 L:0 tmp1
)

```

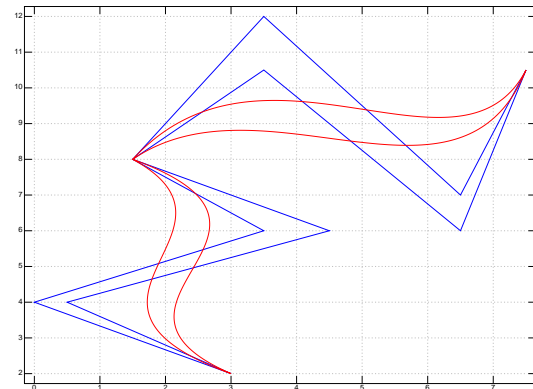
## 10.2 No43 七宝のねじ

ねじり菱形を半分描く ベジエの値を調整しながら半分のみを描く。

```

SPN0
1.5 8
3.5 12
6.5 7.5
7.5 10.5
6.5 6
3.5 10.5
1.5 8
4.5 6
0.5 4
3 2
0 4
3.5 6
1.5 8

```

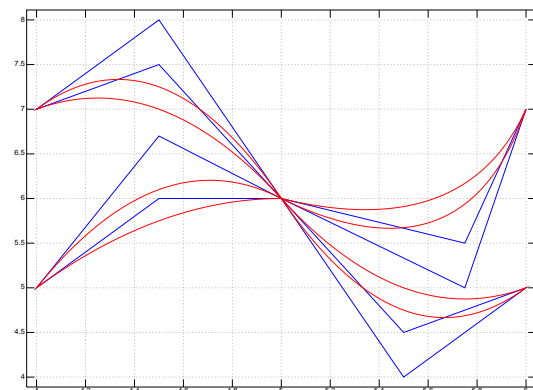


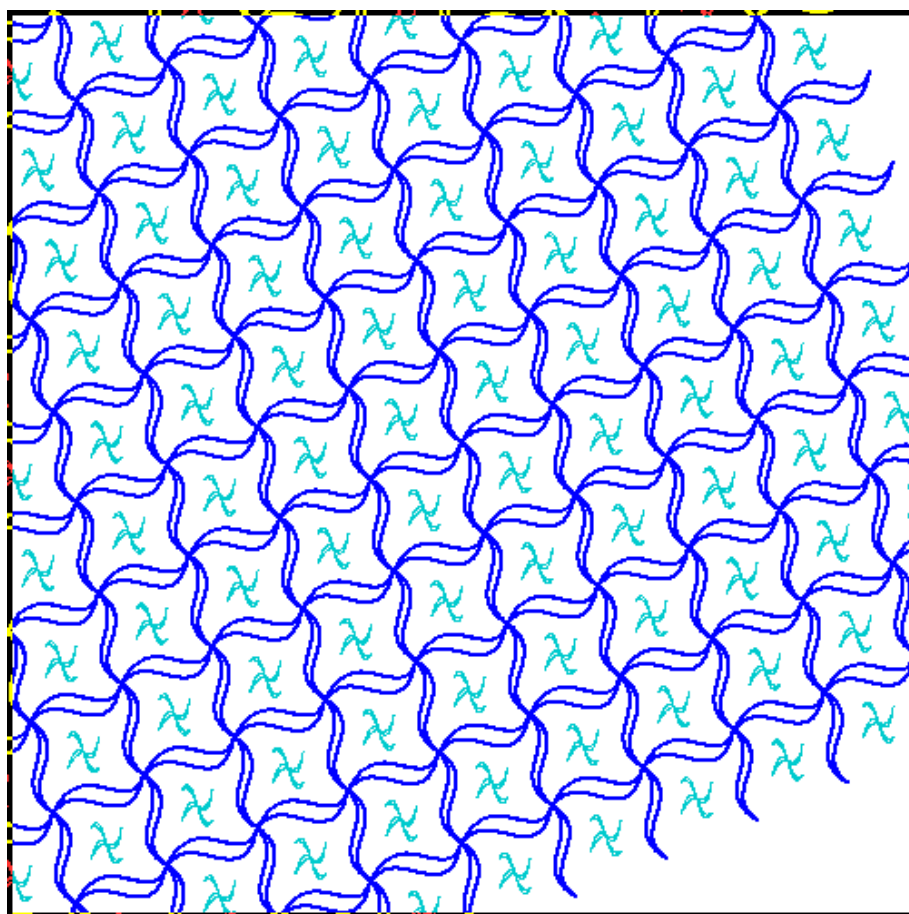
中央の風車 *besier3* で一気に描く。中心から左上、右上、右下、左上の順に往復2本ずつ描く。

```

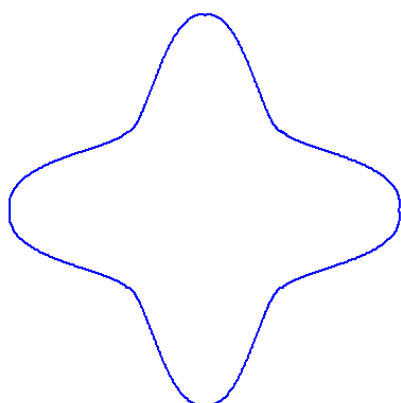
SPN1=: 5 6,4.5 8 ,4 7 ,4.5 7.5 ,:5 6
SPN1=:SPN1,5.75 5 , 6 7 ,5.75 5.5 ,:5 6
SPN1=:SPN1,5.5 4 , 6 5 , 5.5 4.5 ,:5 6
SPN1=:SPN1,4.5 6.7, 4 5 ,4.5 6 ,:5 6

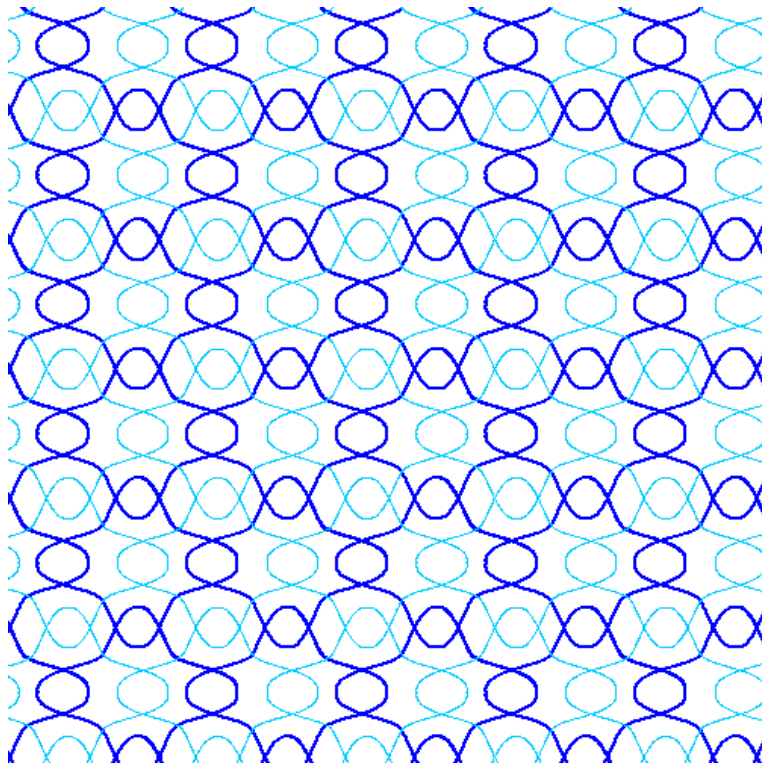
```





### 10.3 No33 道化もっこう





#### 10.4 行きつ戻りつ No.11

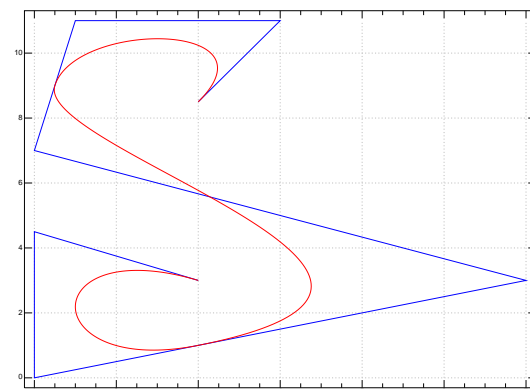
#### 10.5 N0.35 わらび

ピースの作成 ピースは大雑把に絵を描いて固定点を決め、コントロールを入れていった。

全体を 3 区間で描いた。詳細はコントロールを少しずつ動かしながら *plot\_besier* で描いて補正していった

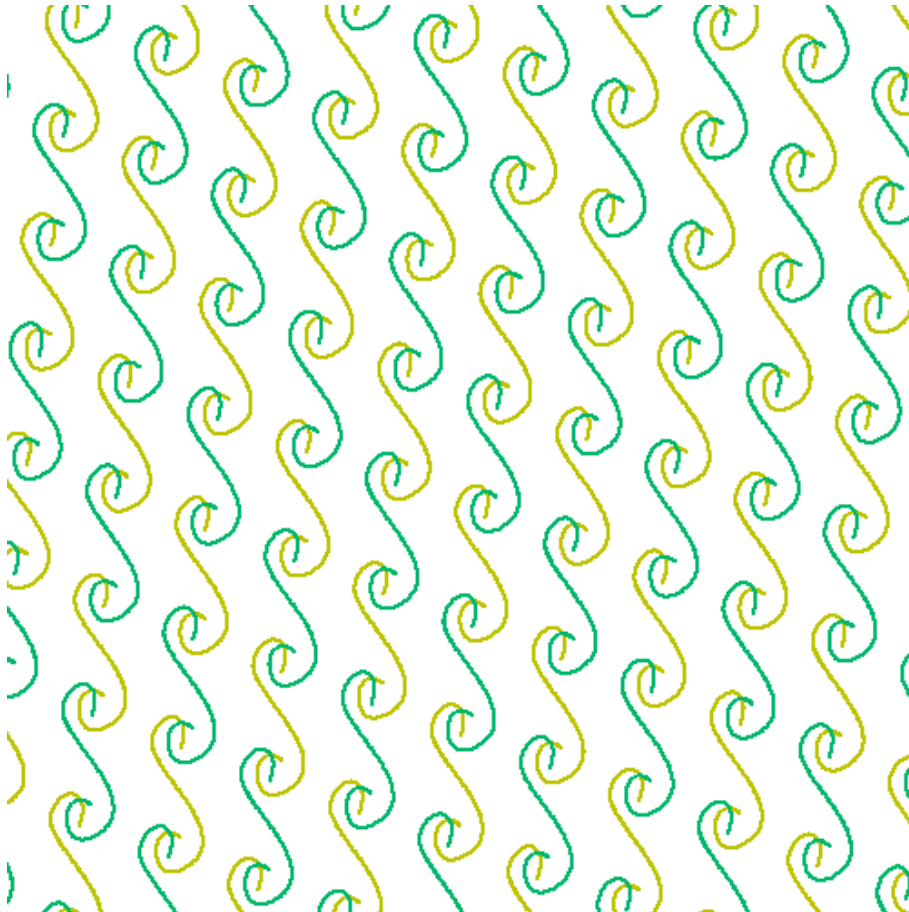
```
WARABI0
3 8.5 NB. fixed point
4 11
1.5 11
1.25 9 NB. fix
1 7
7 3
3 1 NB. fix
1 0
1 4.5
3 3 NB. fix
```

```
4 plot_besier WARABI0
pd 'eps c:/temp/warabi0.eps'
```



差分 差分も画面で微調整した

```
WARAPARAM=: 3 8.5;5.5 _2;0.75 6.75 NB. base;xdiff;ydiff
グラデュエーション 単調な図柄なので2色の近隣色で変化を付けた
(187 192 0;0 182 110;15 15) hokusai_warabi ''
```



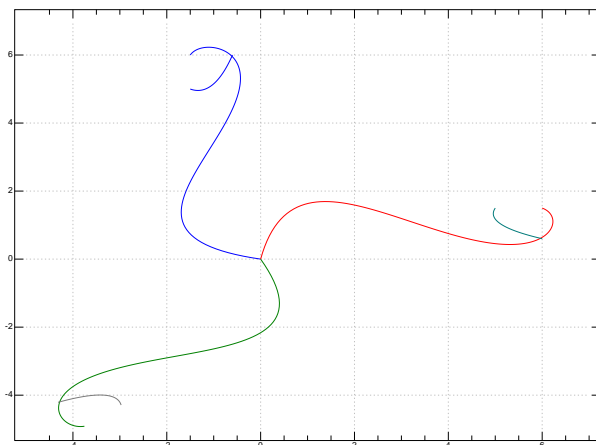
Script .

```
hokusai_warabi=: 4 : 0
NB. 0 0 255 hokusai_warabi 15 15
'Color0 Color1'=. x
Size=. y
tmp01=. calculus_besier4 WARABI0
((Color0;Color1);<Size) draw_dline_grad (<tmp01),<WARAPARAM
)
```

## 10.6 繋ぎ蕨 No40

最初のピースは次のような 6 本の曲線で構成されている。これを一筆書きでは描けないので 1 軸の蕨を 2 本の曲線で書き、後の 2 軸は回転で作成する。ここで大切なことは 3 軸の結節点は  $(0,0)$  にしておくことである。*rotm* による回転は  $z$  軸を回すので、 $(0,0)$  でない場合は振り回され、*transm* で移動させて繋がないといけない。

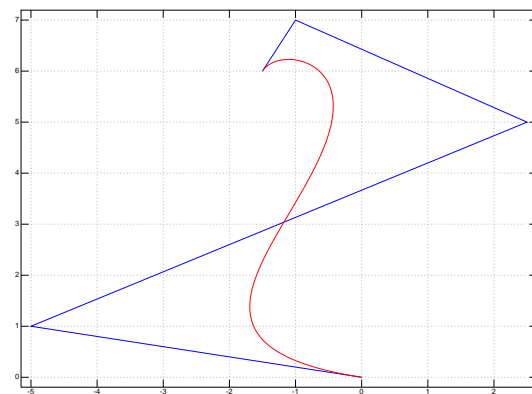
\*5



最初のピース 5 点のベジエを用いて、次のようにポイントを調整しながら蕨の茎を一本描いた

```
5 plot_bezier0 WT0

WT0
_1.5 6
_1 7
2.5 5
_5 1
0 0
```



2,3 本目の蕨 回転を調整しながら決定した

```
WT0=:_1.5 6 ,_1 7 ,2.5 5, _5 1,: 0 0
WT1=:}"1 (WT0,. 1) mp rotm 5r7p1
WT2=:}"1 (WT0,. 1) mp rotm _1r2p1
```

蕨の穂先 3 点のベジエで位置を調整しながら描いた。

\*5 一筆書きで描けない場合は連結線を消すためにペンアップしなければならないが J にはその関数がない。ペンアップとは背景色で線を描くことである。

```
WT3=: _1.5 5, _1 4.75 ,:_0.6 6
WT4=:}: "1 (WT3,. 1) mp rotm 5r7p1
WT5=:}: "1 (WT3,. 1) mp rotm _1r2p1
```

plot の作成 多くの曲線を調整するための plot\_line\_test を作成した

```
a0=. mat_besier5&calc_besier_all WT0
a1=. mat_besier5&calc_besier_all WT1
a2=. mat_besier5&calc_besier_all WT2
a3=. calc_besier3 WT3
a4=. calc_besier3 WT4
a5=. calc_besier3 WT5

plot_line_test (<a0),(<a1),(<a2),(<a3),(<a4),<a5
```

小紋の位置 差分パラメーターを調整して蕨を上手く繋ぐ

1. *TWARABIPARAM*=: 0 0;9 8;0 10
2. (0,0) は最初のピースの原点。
3. (9,8) は右横のピースの差分
4. (0,10) は上方のピースの差分

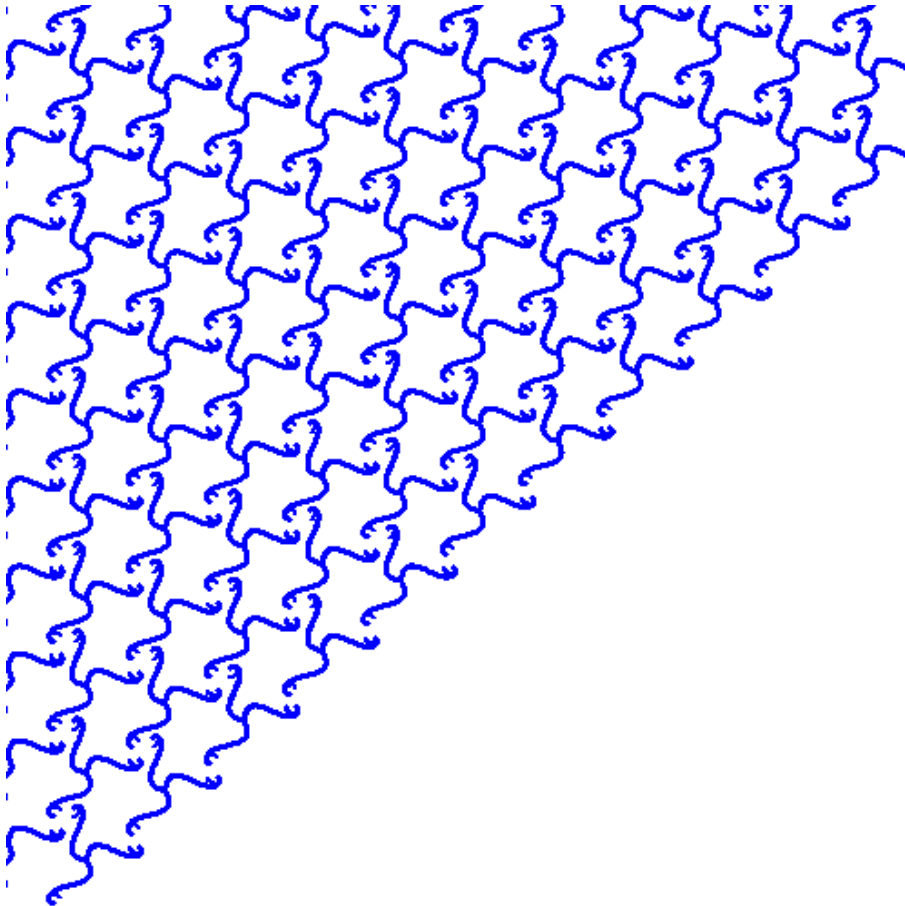
Script 6 に分けて描いた

```
hokusai_tsunagi_warabi=: 4 : 0
NB. Usage: 0 255 255 hokusai_gb0 10 10
Color0=. x ] Size=. y
tmp01=. mat_besier5&calc_besier_all WT0
tmp02=. mat_besier5&calc_besier_all WT1
tmp03=. mat_besier5&calc_besier_all WT2
NB. -----
tmp11=. calc_besier3 WT3
tmp12=. calc_besier3 WT4
tmp13=. calc_besier3 WT5
NB. -----
tmp1=. Size calc_each_poly (<tmp01);<TWARABIPARAM
tmp2=. Size calc_each_poly (<tmp02);<TWARABIPARAM
tmp3=. Size calc_each_poly (<tmp03);<TWARABIPARAM
```

```

NB. -----
tmp4=. Size calc_each_poly (<tmp11);<TWARABIPARAM
tmp5=. Size calc_each_poly (<tmp12);<TWARABIPARAM
tmp6=. Size calc_each_poly (<tmp13);<TWARABIPARAM
NB. -----
(find_center tmp2) dwin ''
Color0 dline3 L:0 tmp1
Color0 dline3 L:0 tmp2
Color0 dline3 L:0 tmp3
NB. -----
Color0 dline2 L:0 tmp4
Color0 dline2 L:0 tmp5
Color0 dline2 L:0 tmp6
)

```

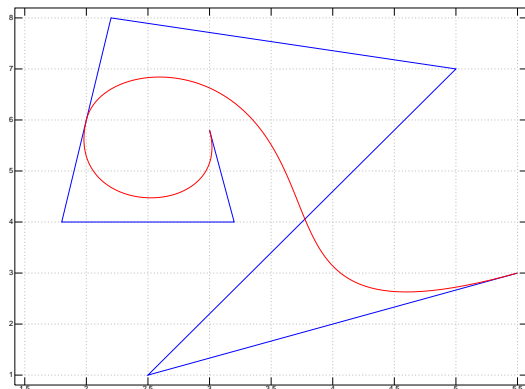




## 10.7 四つで蕨 No. 38

4本一組の蕨が2組ある。蕨は反対向きになっているので各組一本で計2本の蕨を描き、他の6本は回転と移動で作成する

蕨 (0) 4本セット (0) のうちの一本



穂先の先端は3次、胴部分は4次(5ポイント)のベジエを用いた

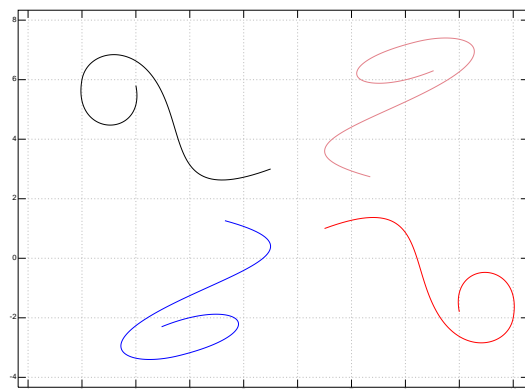
```
WR4_0=: 3 5.8, 3.2 4, 1.8 4 ,:2 6
```

```
WR4_0=: WR4_0,2.2 8,5 7 ,2.5 1 ,: 5.5 3
```

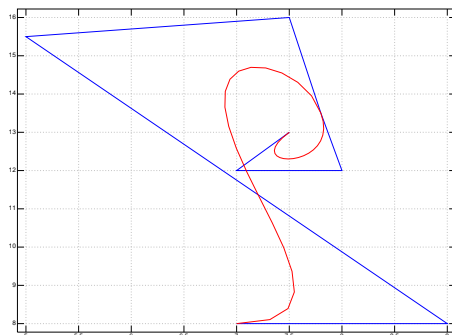
```
4 5 plot_besier WR4_0
```

蕨 4本セット (0) .

```
plot_line_test a1=. calc_yotsude_sub0 ''
```



2本目の蕨 1本目と反対を向いているので最初から描いた



この組の位置は最初の手描きの一本の位置を *transm* で動かして決める。位置パラメーターリストは用いない。

4本組の蕨 (1) .

```

WR4_10=: 7.5 14,7 12      ,8 12      ,:7.8 14
WR4_10=: WR4_10 ,7.5 16,5 15.5,9 8      ,:7 7
WR4_10=: }:"1 (WR4_10,. 1)  mp transm 5 4
WR4_12=: }:"1 (WR4_10,. 1)  mp (rotm 1r2p1) mp transm 23 _2
WR4_11=: }:"1 (WR4_10,. 1)  mp (rotm 1p1)  mp transm 25 21
WR4_13=: }:"1 (WR4_10,. 1)  mp (rotm _1r2p1) mp transm  2 23

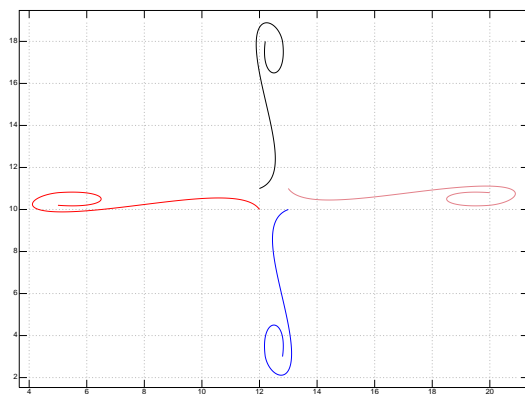
```

```

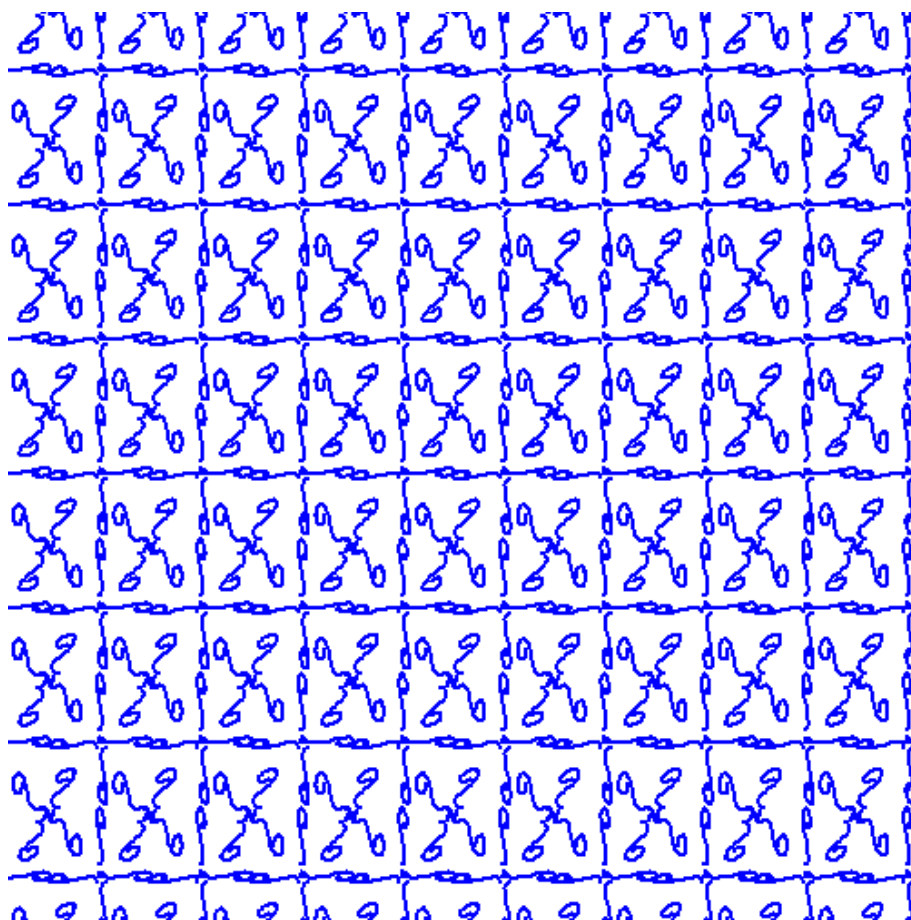
4 5 plot_besier WR4_10

```

plot\_line\_test a2=. calc\_yotsude\_sub1 '' 90度回転させたベジエ曲線はは横に間延びするので 15 パーセント縮小した

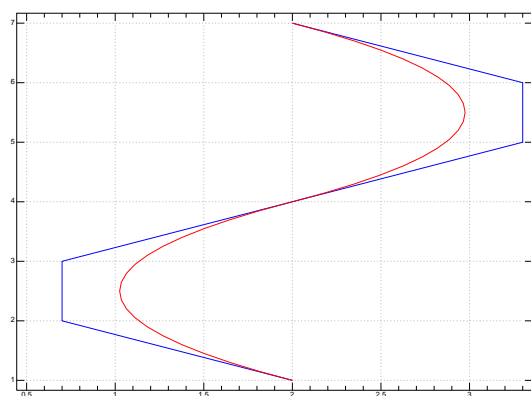


芯 七宝のねじの SPNI を流用し、縮小した



## 10.8 流れ万字 No.21

次のような対称の  $[S]$  を描こうとする場合、5 ポイントまたは 7 ポイントが使いやすい。7 ポイントは作っていないが、対称型は 4 ポイントを 2 個繋げば 7 ポイントになる。次は 4 ポイント (3 次) で描いたものである。



NMANJI1=: 2 7 ,3.3 6, 3.3 5 ,: 2 4

NMANJI1=: NMANJI1, 0.7 3,0.7 2 ,:2 1

NMANJI1

2 7

3.3 6

3.3 5

2 4

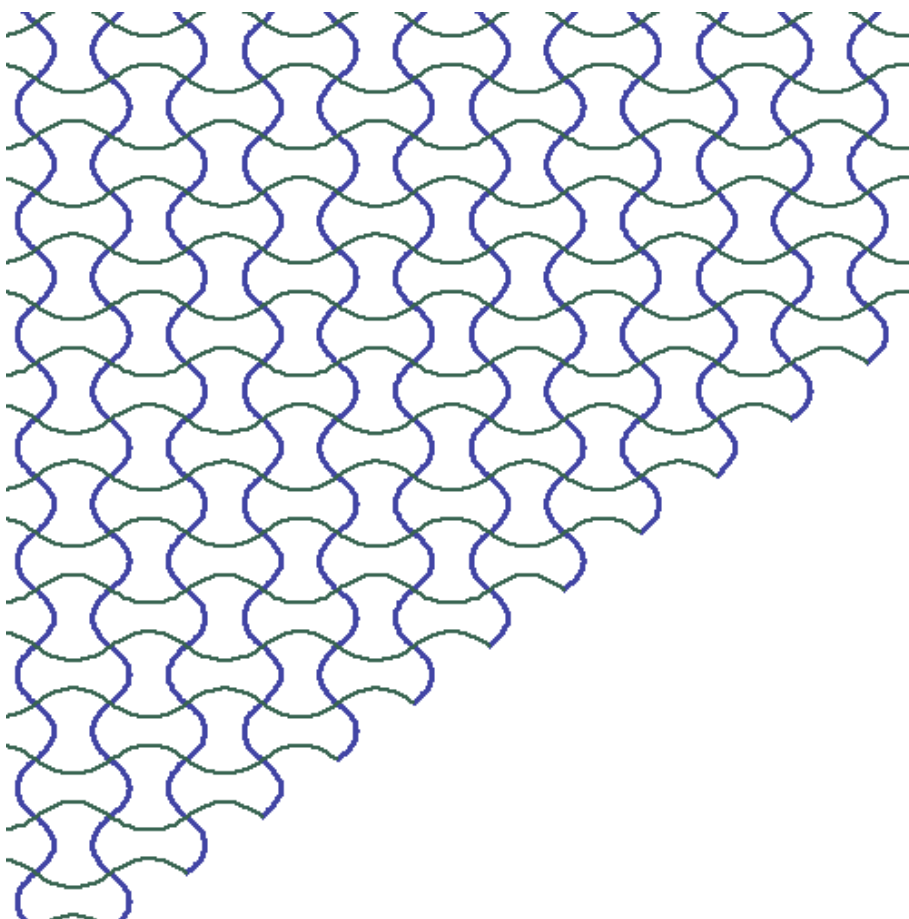
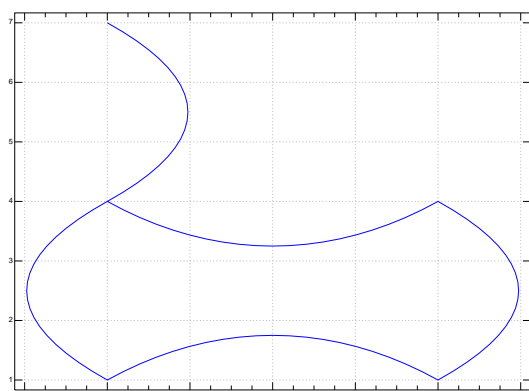
0.7 3

0.7 2

2 1

一筆で書くと次の図形を重ねる。縦線と横線では太さが異なり一筆にはならないので 4 の線分に分割する

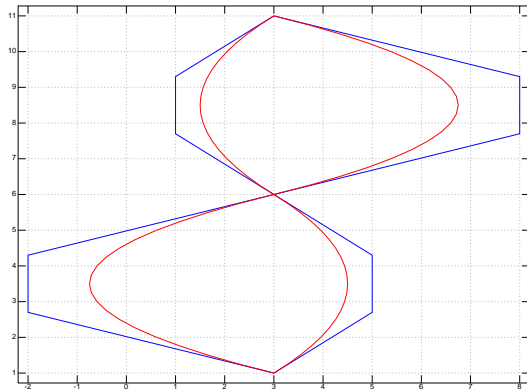
```
plot {@|: calculus_besier4  NMANJI1,({.NMANJI4),(}.|.NMANJI2),}.|.NMANJI3
```



## 10.9 早割りの図 (乱れ縦割り)No.23

最初のピースは2本の縦線で4ポイントのベジエを対称にを使って7ポイント風に描いた。

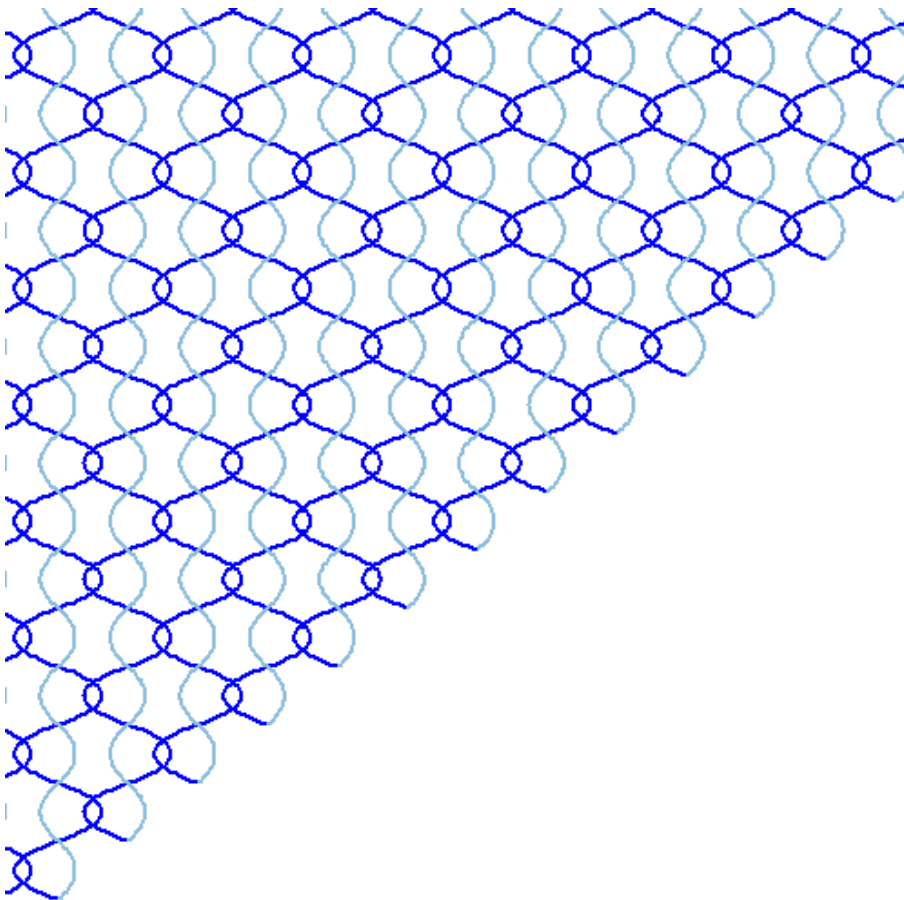
```
4 plot_besier MTATE0,}.|.MTATE1
pd 'eps c:/temp/hokusai_mtate0.eps'
```



```
MTATE0,..|.MTATE1
3 11 _ 3 11
8 9.3 _ 1 9.3
8 7.7 _ 1 7.7
3 6 _ 3 6
_2 4.3 _ 5 4.3
_2 2.7 _ 5 2.7
3 1 _ 3 1
```

x 方向の差分と y 方向の差分。(3,1) はベース。(6,5) は右に並べるピースとの差分 (x,y)。(0,10) は上に積む  
ピースとの差分で 0=真上に 10 ポイント離して積む。

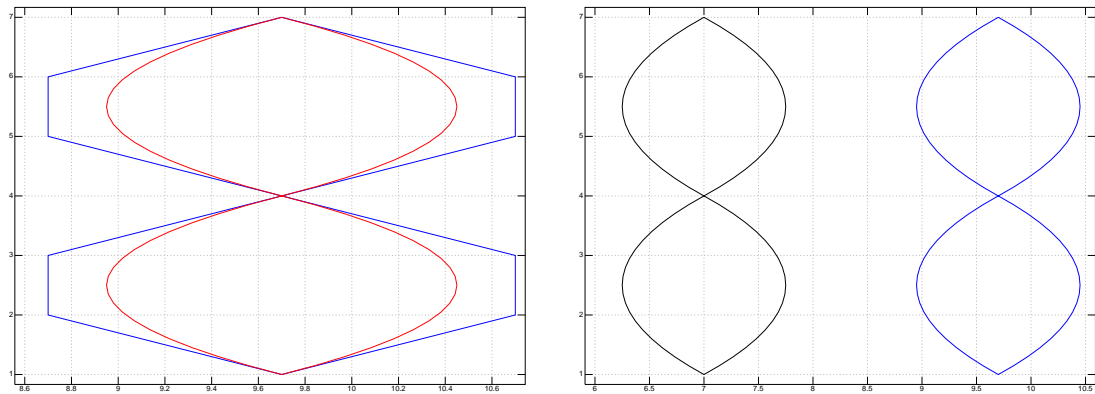
MTATEPARAM=: 3 1;6 5;0 10



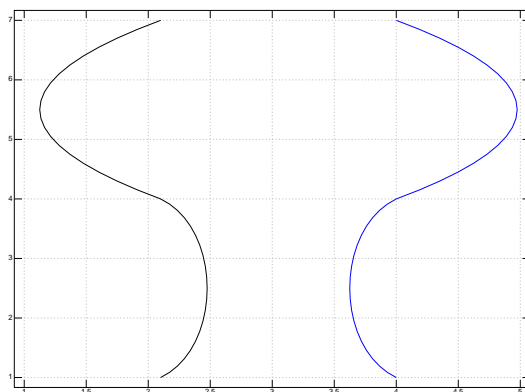
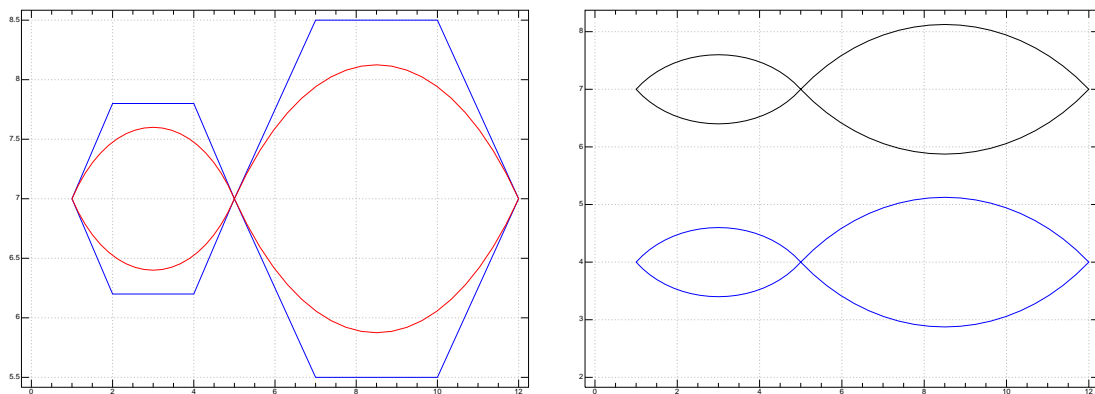
### 10.9.1 組み入れしたて脇 No27

原図を眺めながら最小ピースを切り出す。2 個セットが最小単位になった

ピース 0 右は左のデータを *transm 2.70* で動かした。スケッチを基に図を見ながら調整する。

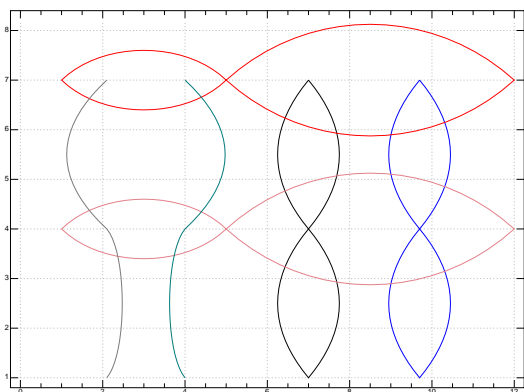


ピース 1 1 個は動かして 2 個描く。



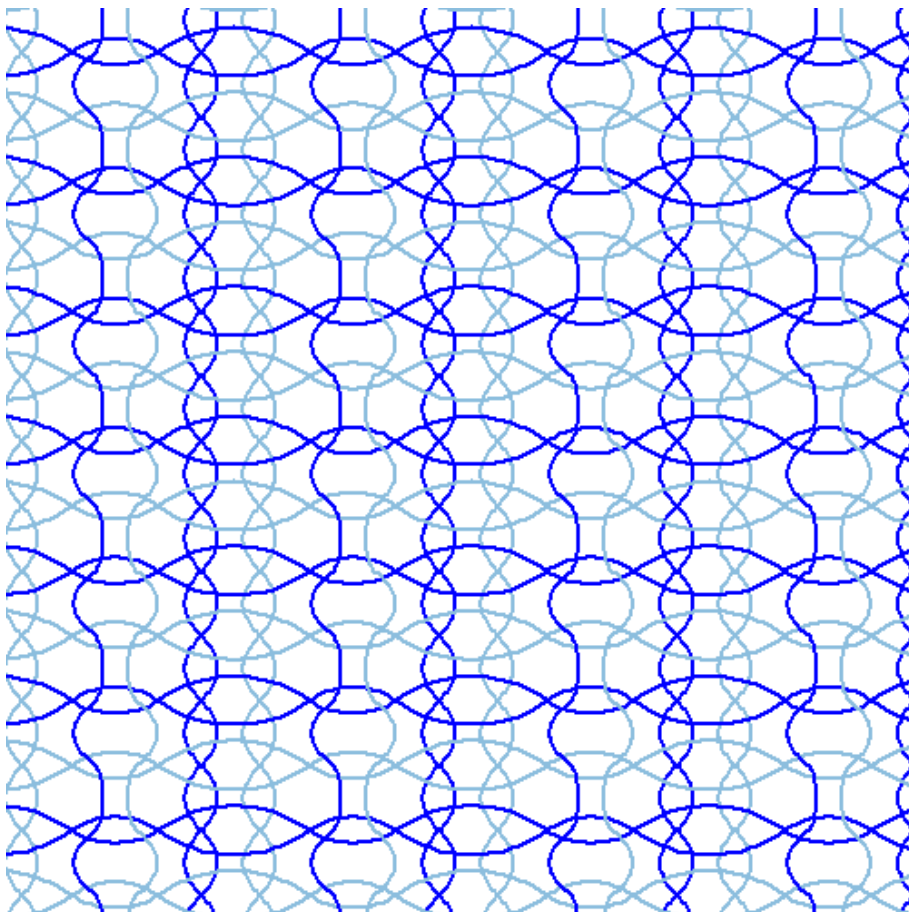
ピース 2 徳利の型。これを縦に積み上げる

ピースの組み合わせ ピースの最小単位はこのようになる



差分パラメータ KUMIIREPARAM=: 3 1;11 0;0 6

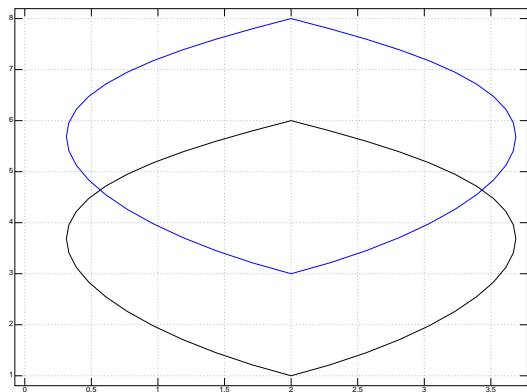
差分パラメータではピース全体の位置決めを行う。2 個セットの中での位置はベジエデータを *transm* で調整する。



## 10.10 尾の上重ね No39

北斎の版図をずっと眺めていて次の図画元素にあたりと考えた。

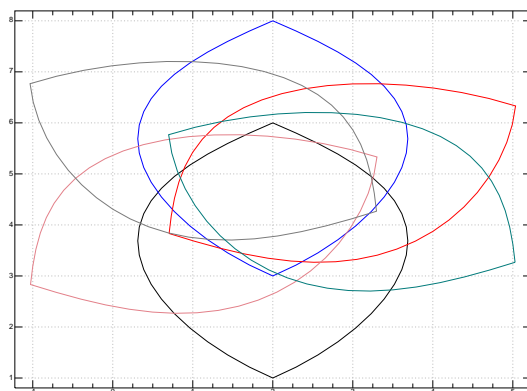
元素 2 個あるから分子が



最初の小紋のピース 小紋のピースは元素 (分子) を左右に 120 度回転させて重ねて花弁を作る。最初の分子を大雑把に作った。北斎の精密な構図やこれを刻んだ江戸の彫師の美観には感服させられる。

小紋のピース内の位置や構成はベジエデータを回転 (*rotm*), 移動 (*transm*), スケール変換 (*elongm*) で *plot\_line\_test* で図を出しながら調整する。このピースが綺麗でないと、良い小紋にならない

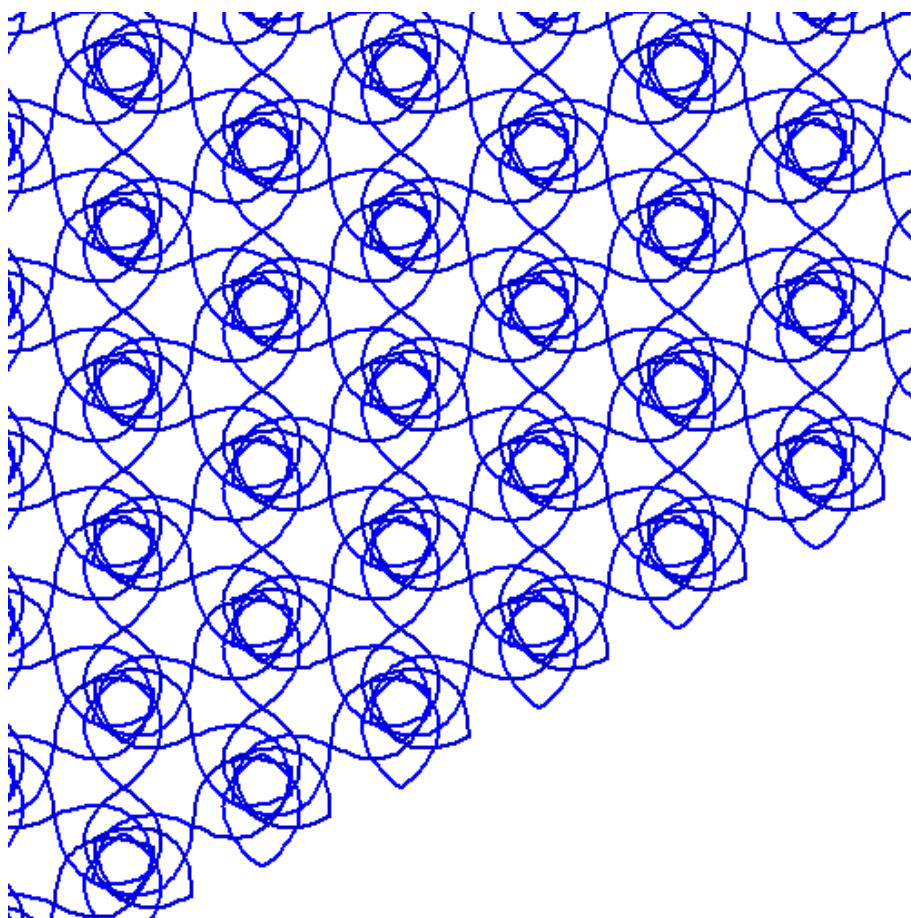
`plot_line_test (calc_okasane_sub0 ''), (calc_okasane_sub1 ''), calc_okasane_sub2 ''`



差分 小紋間の位置はこちらで図を見ながら調整する。花弁を繋ぐには細かい調整が必要である。

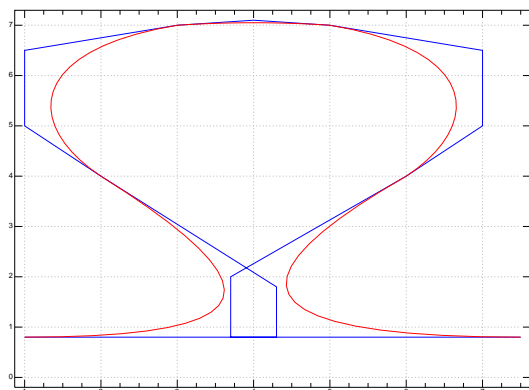
OKASANEPARAM=: 3 1;6.1 3.5;0 7 ピースの始点:X=右隣のピースの位置;y=上方のピースの位置



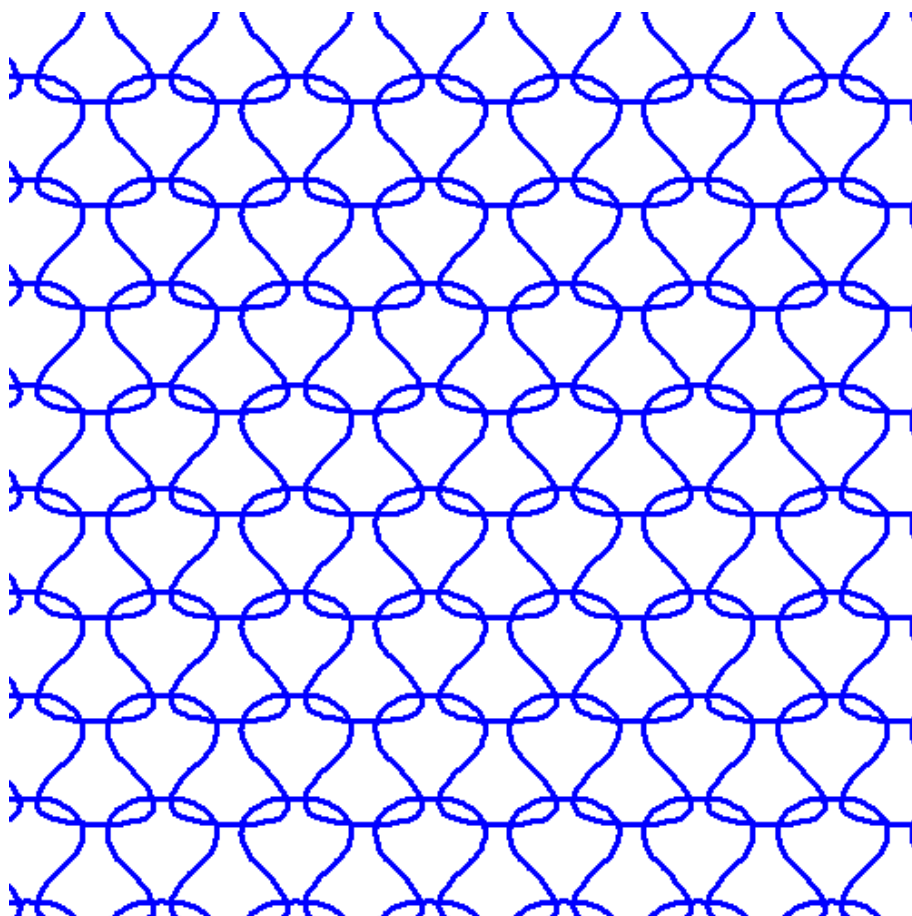


### 10.11 向神通網 No55

最初のピースを44344のミックスタイプのベジエで描いてみた。左右の対称はやさしいが上下の対象は中々複雑である。



JINTUU0  
 (1 0.8)  
 4.3 0.8  
 4.3 1.8  
 (2 4)  
 1 5  
 1 6.5  
 (3 7)  
 4 7.1  
 (5 7)  
 7 6.5  
 7 5  
 (6 4)  
 3.7 2  
 3.7 0.8  
 (7.5 0.8)



## 10.12 阿蘭陀組 (0) NO81

最初のピース。左半分は  $180$  度 ( $\pi = 1p1$ ) 回転させて繋ぐ

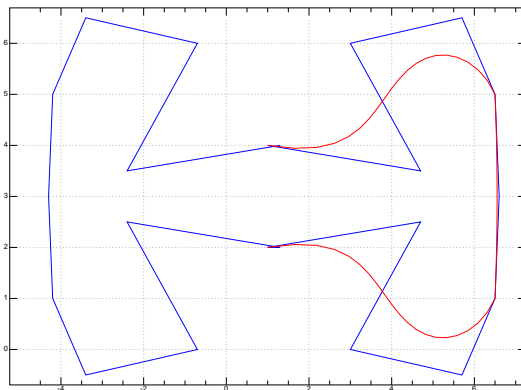
```
5 3 5 5 3 5 plot_besier ORANDAK0,}.ORANDAK1  
pd 'eps c:/temp/hokusai_oranda01.eps'
```

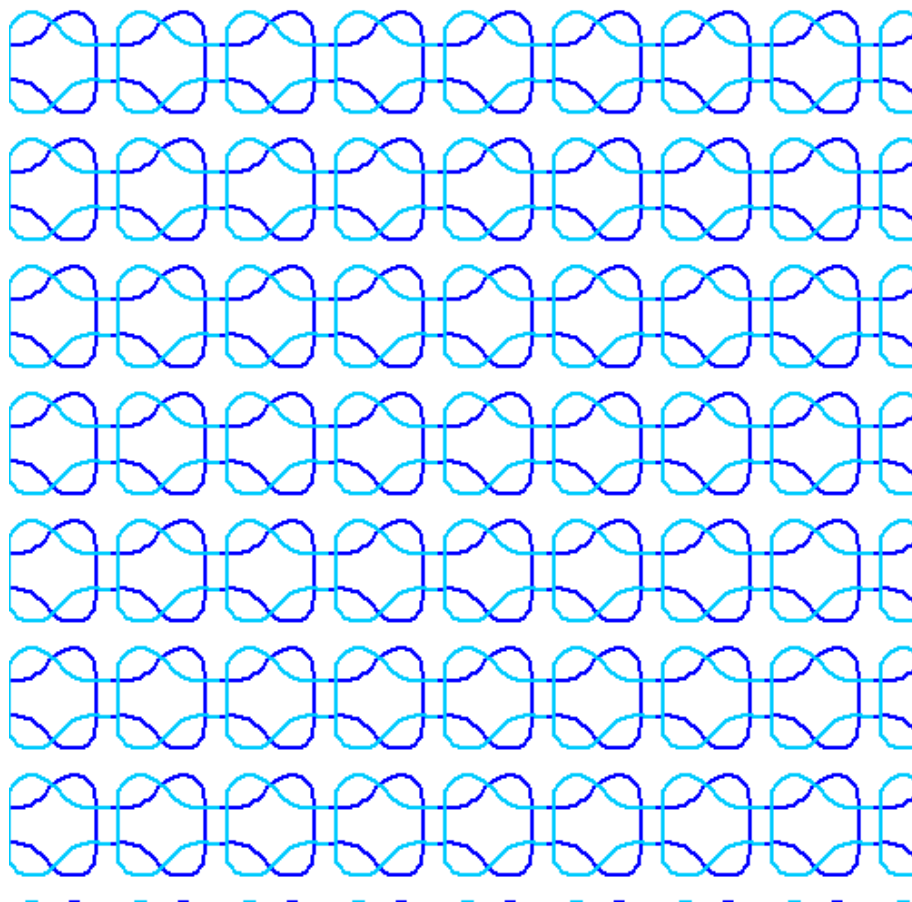
```
ORANDAK0=: (1 4), 4.7 3.5 , 3 6 , 5.7 6.5, (6.5 5), 6.6 3, : (6.5 1)
```

```
ORANDAK0=. ORANDAK0, 5.7 _0.5, 3 0, 4.7 2.5, : (1 2)
```

```
ORANDAK1=: }:"1 (ORANDAK0 ,.1) mp (rotm 1p1) mp transm 2.3 6 NB. rot 180 degree
```

```
ORANDA0PARAM=: 2 1;6 0;0 7
```



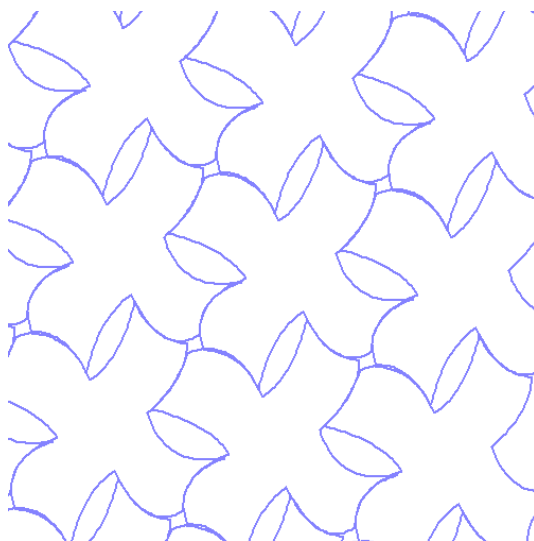
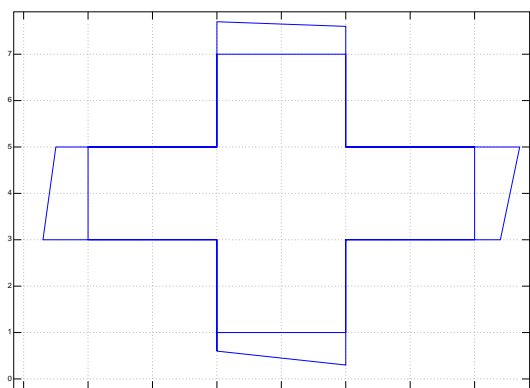


### 10.13 N0.54 捻り四つ手組

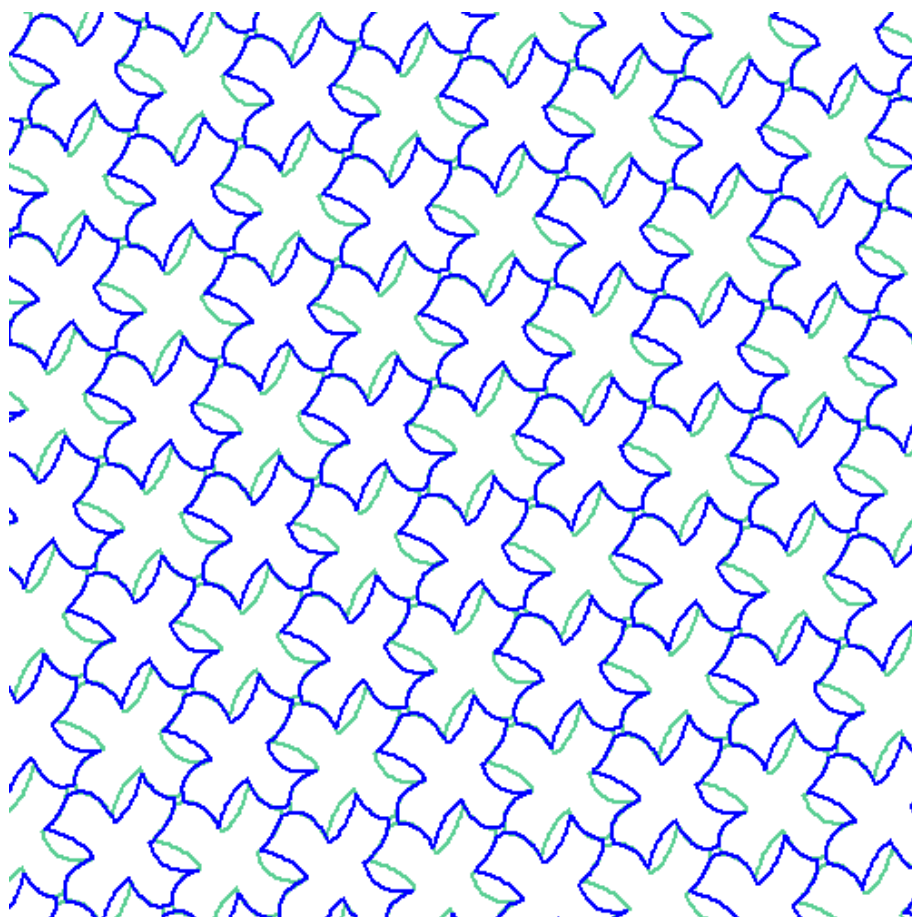
基の図形を回転させてあわせる型は非常にデリケートな位置や図形の調整が求められる。

最初に直立で基本形を描き、 $\frac{1}{3}\pi$  回転させた後、数個のベジエカーブを描いて差分パラメーターで位置決めた後、ベジエのポイントを細かく調整して補正した。

周辺部分は図形をあわせるために加えた部分である

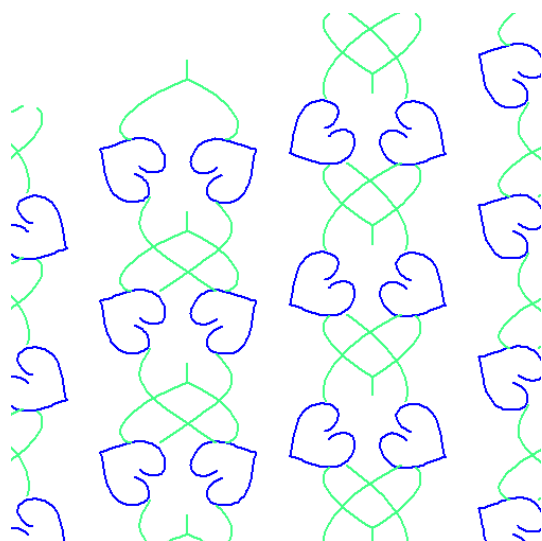
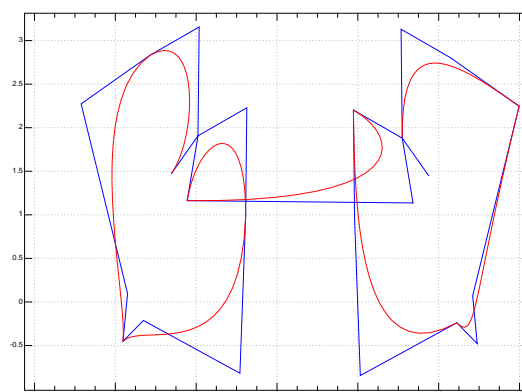
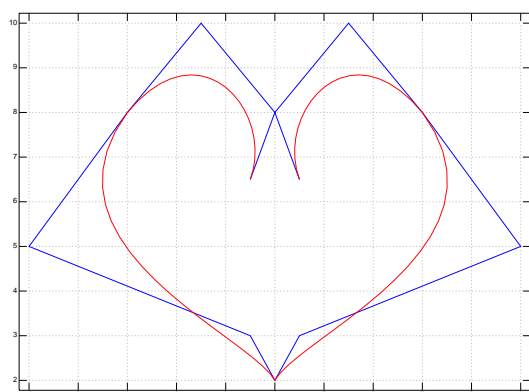


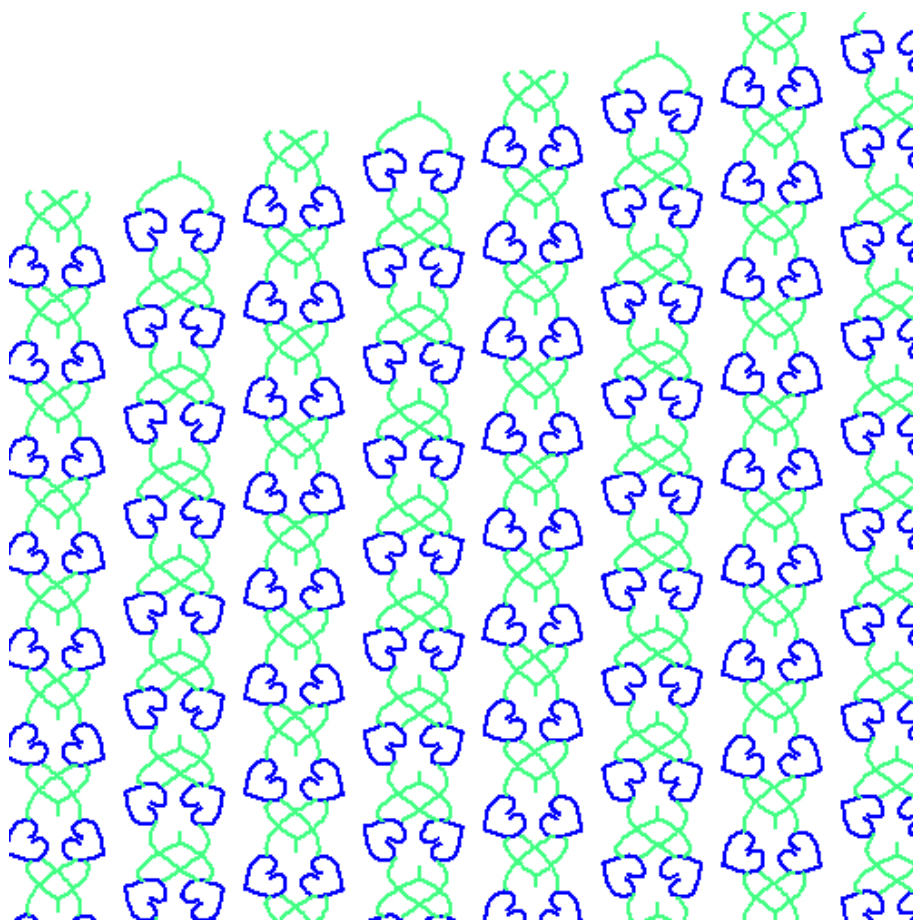
単調にならないよう 2 色で描いた



## 10.14 うけ葵 No37

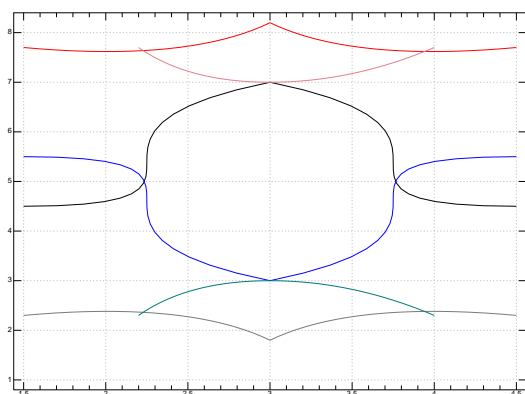
左の原型のように左右対称にポイントを取った方が簡単であり、原型を左右に各々  $\frac{2}{3}\pi$  回転させて、図を見ながら修正した。

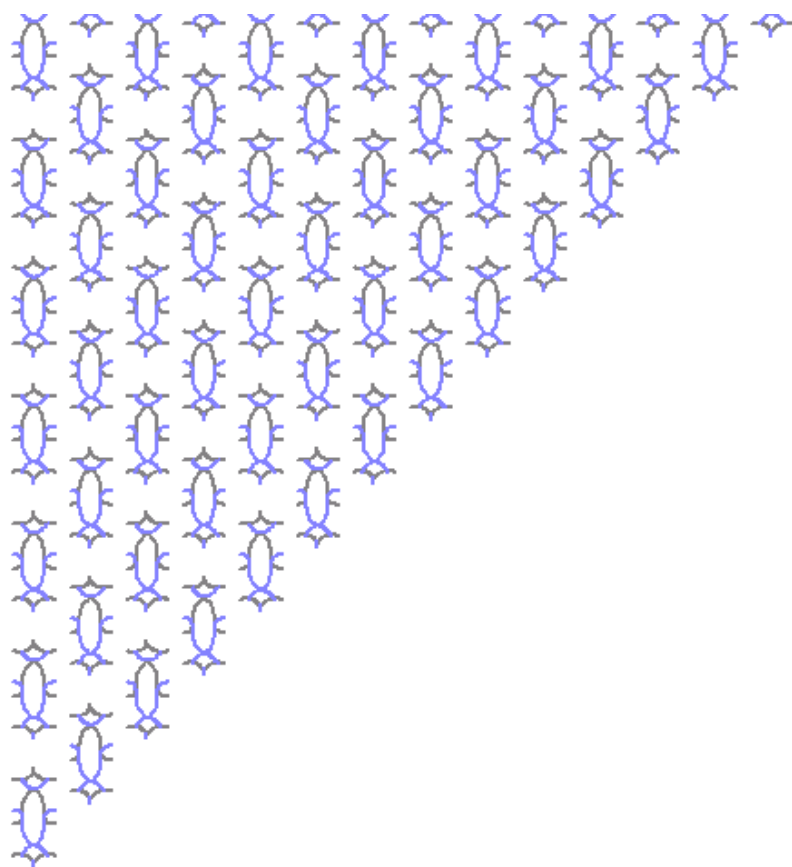
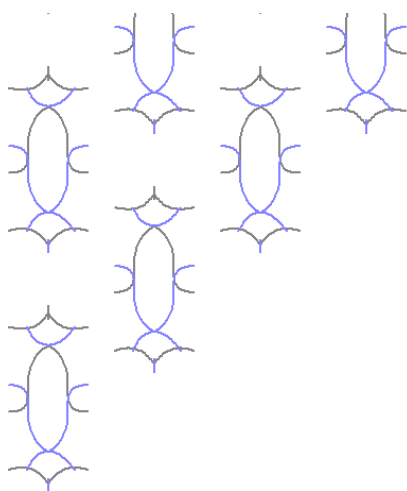




# 10.15 文の字繋ぎ No.51

「文」の字をデザインしたもの。左右対称だが北斎先生の美観に沿うようにベジエのポイントを取るのは試行と調整をを重ねなければならないようだ。



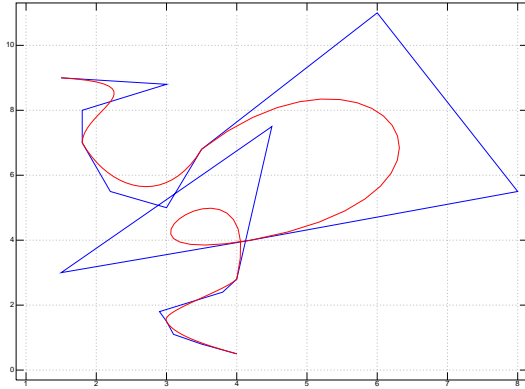


```
plot_line_test (calc_bun_sub0 ''),calc_bun_sub1 ''
pd 'eps c:/temp/hokusai_bun0.eps'
```



## 10.16 菱割り (らの字)No.59

らの字の一筆書きである。大まかに座標を割り振り、図を見ながらベジエポイントを補正した。



RANOJI0

(1.5 9)

3 8.8

1.8 8

(1.8 7)

2.2 5.5

3 5

(3.5 6.8)

6 11

8 5.5

(4.2 4)

1.5 3

4.5 7.5

( 4 2.8)

3.8 2.4

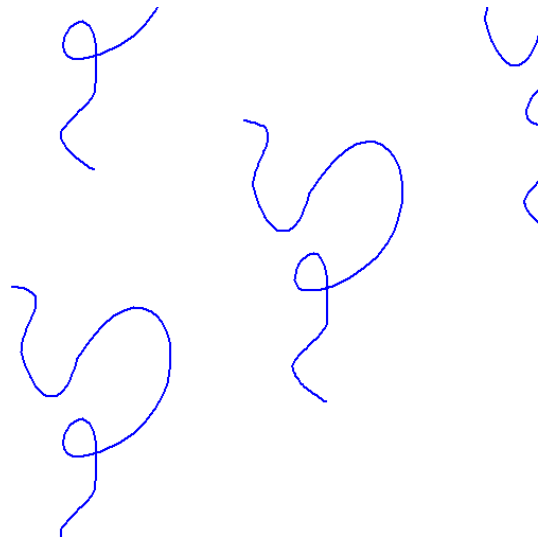
2.9 1.8

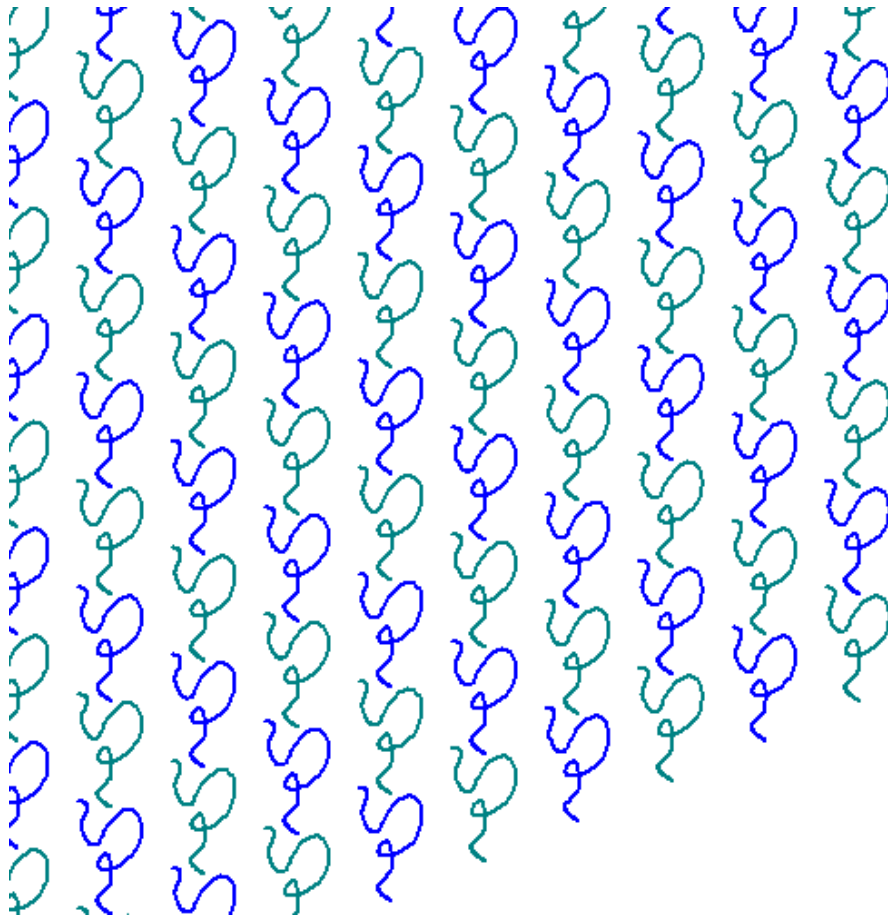
( 3 1.5)

3.1 1.1

3.5 0.8

( 4 0.5)



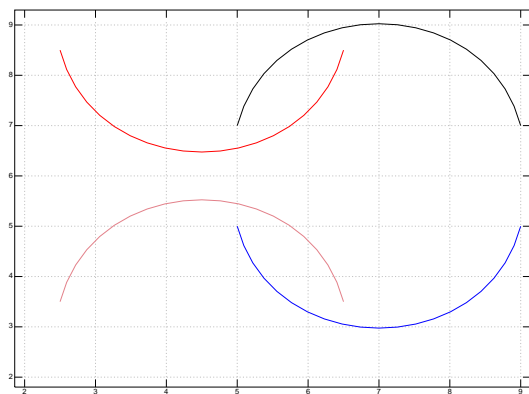


スクリプトは単色と2色の双方を用意した。

```
hokusai_ranoji_grad=: 4 : 0
NB. Usage: (0 255 255 ;0 255 255) hokusai_yotsude_warabi 15 15
'Color0 Color1'= . x
Size = . y
tmp0 =. calculus_besier4 RANOJI0
((Color0;Color1);<Size) draw_dline_grad (<tmp0),<RANOJIPARAM
)
```

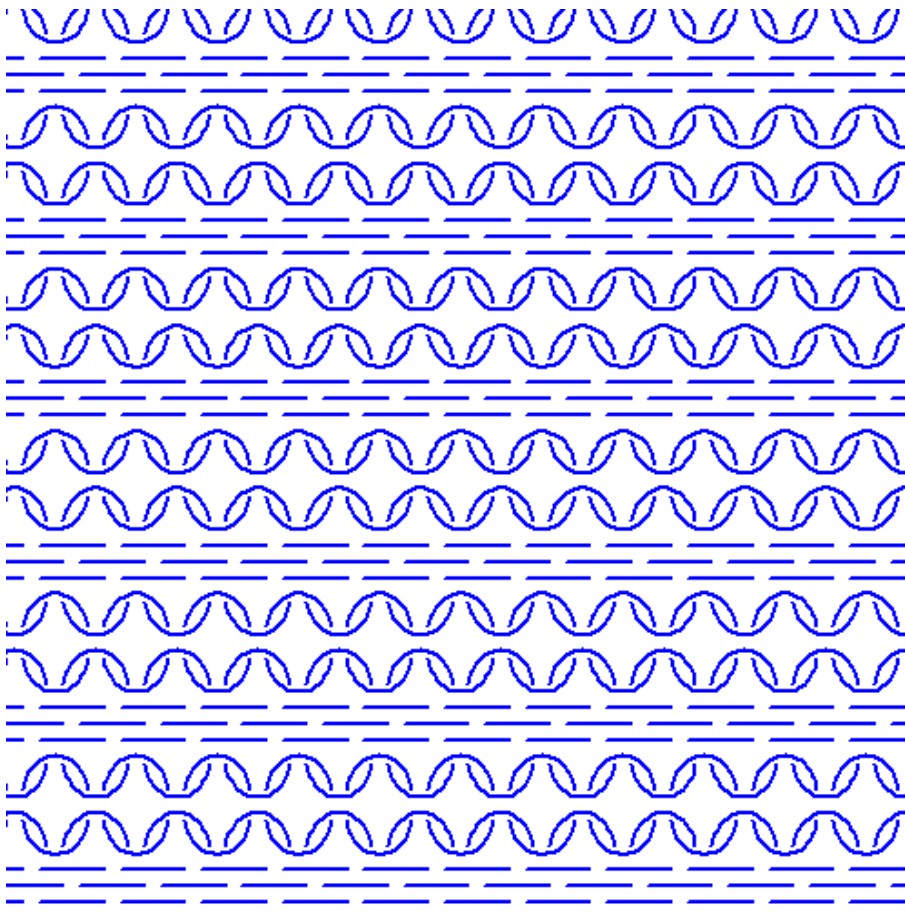
#### 10.17 瓦小口 No77

4枚の瓦の曲線部分のベジエポイントを次のように定めた。上から  $P_0, P_1, P_2, P_3$  である



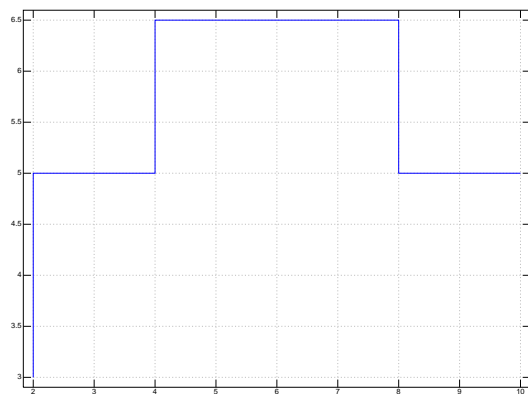
```
KAWARA0;KAWARA1;KAWARA2;KAWARA3
+-----+-----+-----+-----+
|  5   7|  5   5|2.5 8.5|2.5 3.5|
|5.5 9.7|5.5 2.3|  3 5.8|  3 6.2|
|8.5 9.7|8.5 2.3|  6 5.8|  6 6.2|
|  9   7|  9   5|6.5 8.5|6.5 3.5|
+-----+-----+-----+-----+

plot_line_test calc_kawara_sub0 ''
pd 'eps c:/temp/hokusai_kawara0.eps'
```



## 10.18 碁盤割 No.63

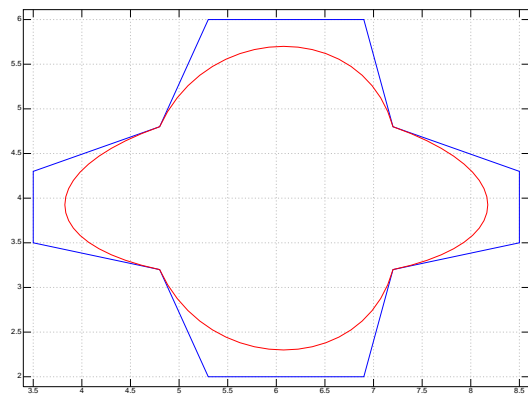
原図を眺めて枠は次の部分で足りると思った。



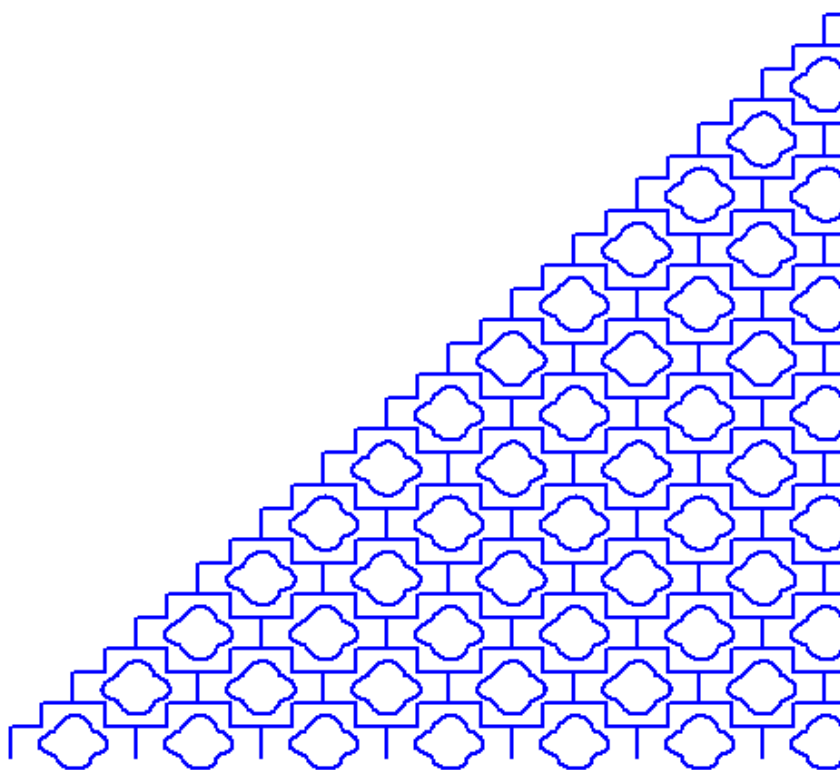
```
H63_line
2 3
2 5
4 5
4 6.5
8 6.5
8 5
10 5
```

花の部分のベジエのポイント

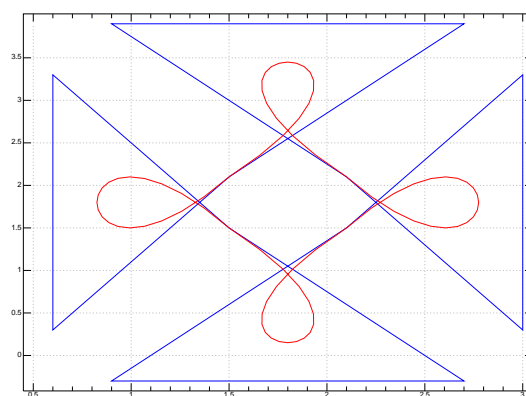
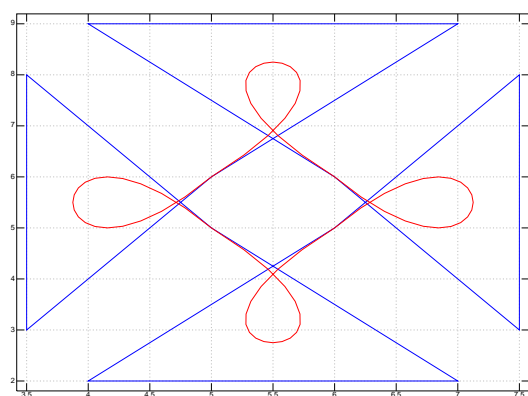
```
4 plot_besier H63_flower
pd 'eps c:/temp_hokusai_h63_0.eps'
```

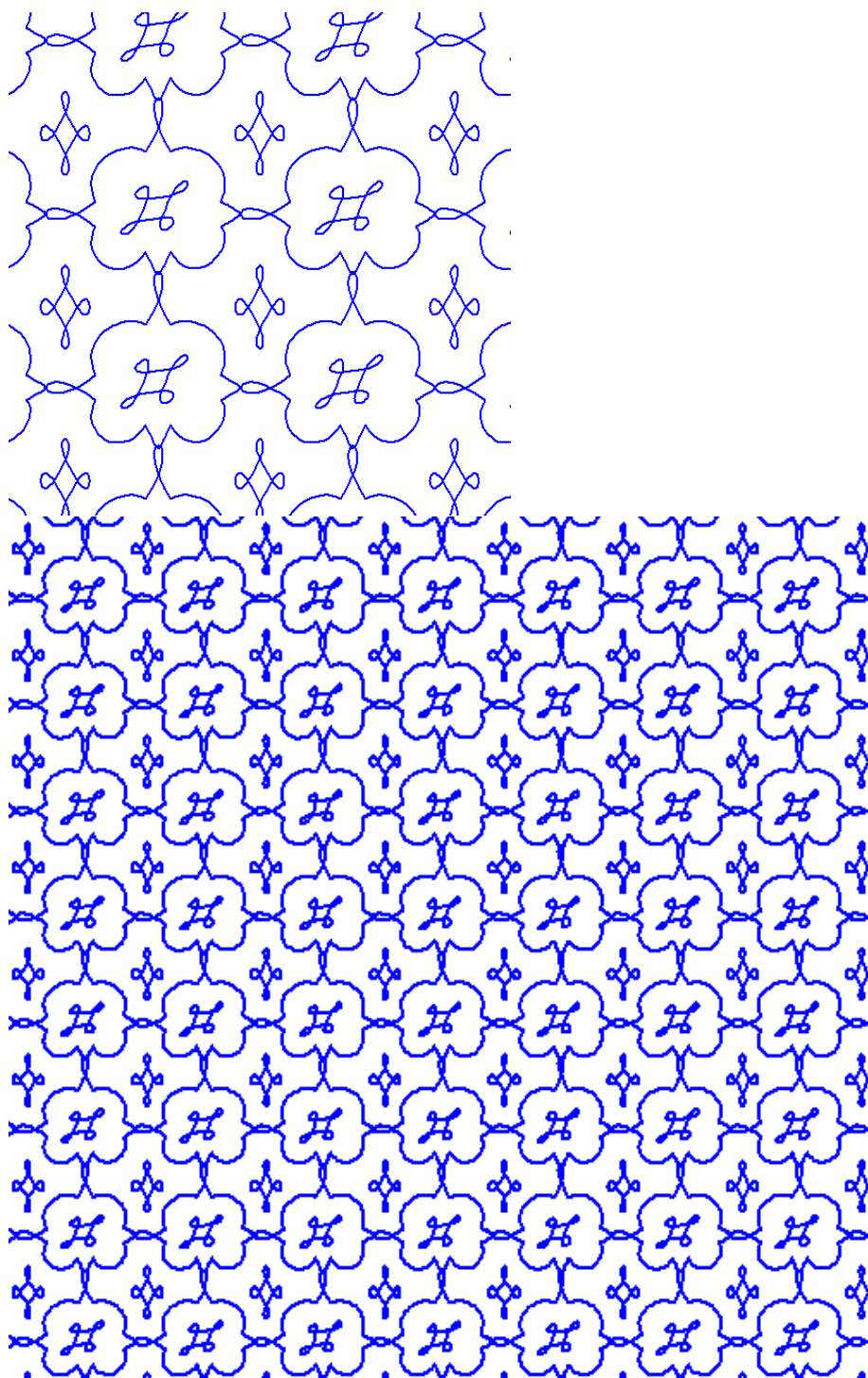


```
H63_flower
(4.8 4.8)
5.3 6
6.9 6
(7.2 4.8)
8.5 4.3
8.5 3.5
(7.2 3.2)
6.9 2
5.3 2
(4.8 3.2)
3.5 3.5
3.5 4.3
(4.8 4.8)
```



# 10.19 四つ手 No29





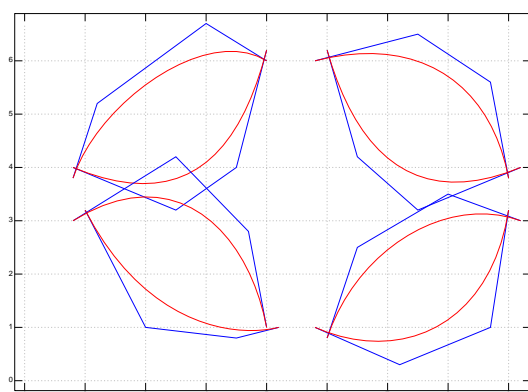
10.20 大井紋高麗型 No.48

compose\_00I ''

+-----+-----+-----+-----+-----+-----+-----+

```
|1.8 3.8|1.8 4| 6 6.2|5.8 6| 6 0.8|5.8 1| 2 3.2|1.8 3|
|2.2 5.2|3.5 3.2|6.5 4.2|7.5 6.5|6.5 2.5|7.2 0.3| 3 1|3.5 4.2|
| 4 6.7|4.5 4|7.5 3.2|8.7 5.6| 8 3.5|8.7 1|4.5 0.8|4.7 2.8|
| 5 6| 5 6.2|9.2 4| 9 3.8|9.2 3| 9 3.2|5.2 1| 5 1|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
calc_besier4 plot_line_once compose_00I ''
pd 'eps c:/teml/hokusai_00i0.eos'
```

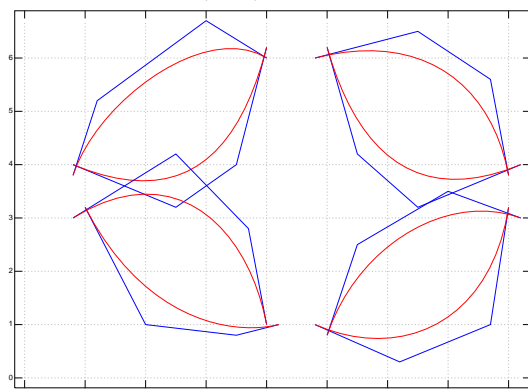


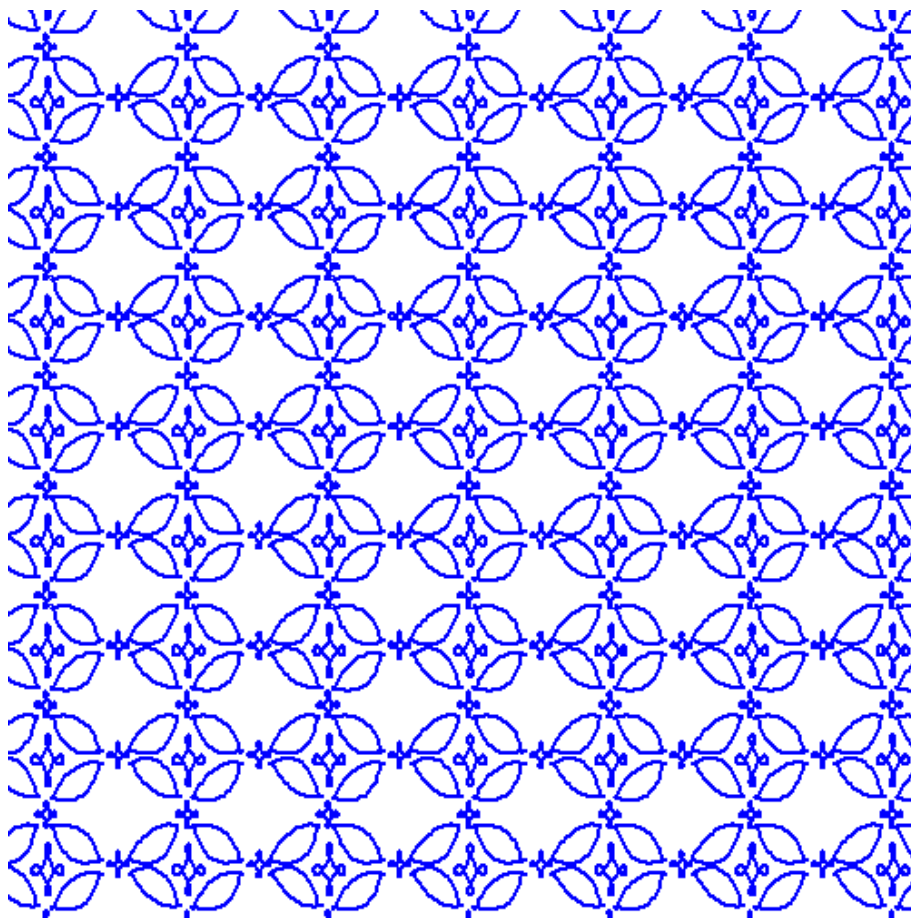
小さな点は花を縮小する。(*elong m*)  
 小紋内の花の位置は *transm* で動かして決定する

```
00I8=: }:"1 (Y29_1 ,.1) mp (elongm 0.6 0.6) mp transm 2.4 0
00I9=: }:"1 (Y29_1 ,.1) mp (elongm 0.4 0.3) mp transm 3.4 5.2
00I10=: }:"1 (Y29_1 ,.1) mp (elongm 0.4 0.3) mp transm _1 2
```

差分パラメータは小紋相互の位置を決める。 *base(x y);right neighbour(x y); upper(x y)*

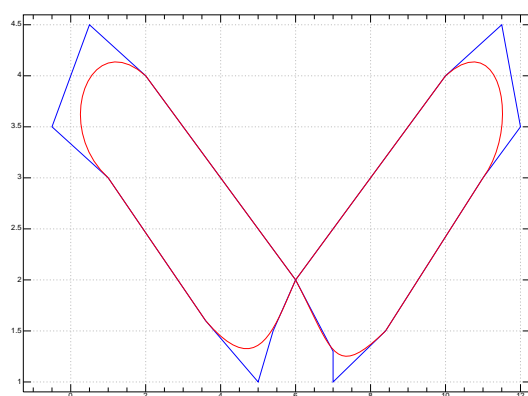
```
OOIPARAM=: 2 4; 9 0; 0 7
```



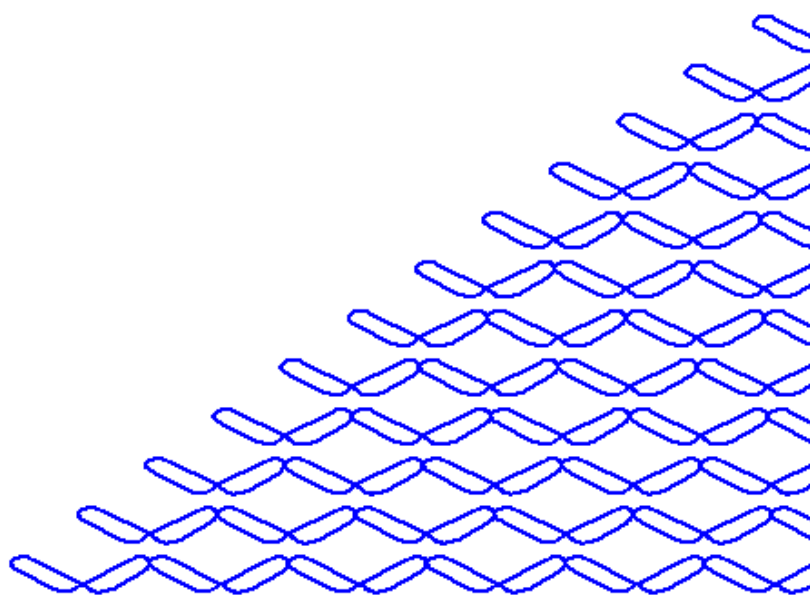
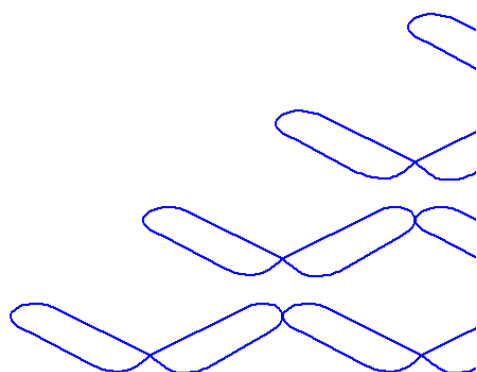


## 10.21 藤種 No47

(FUJIL,FUJIS) plot\_line\_test2 (calc\_besier2 L:0 FUJIL),calc\_besier4 L:0 FUJIS







## 11 北斎にフォームを付ける

### 11.1 新型小紋帳の版木発見（1986）

荻窪の古書店で「ボストンで見つかった北斎展（ボストン美術館の新版木発見）」東京放送1987を見つけた。ボストン美術館には W.S. ピゲローとフェノロサによって明治16年からの7年間に日本で蒐集された4千点の絵画と6万点の木版画があるという。

TBS が1985年に大英博物館の版木で「浮世絵・近江八景全図の謎ー大英博物館秘蔵の版木・100年ぶりに復活」という番組を作成したとき、同博物館のブヌー博士達からボストン博物館にも版木が保存されていると教えられた。

ピゲローコレクションは寄贈の条件として門外不出とされており、調査にあたられた日本浮世絵協会理事長菊池貞夫氏の巻末記事によるとボストン総領事の仲介でボストン博物館と長い交渉の結果同氏とアダチ版画研究所の彫士や刷士が倉庫に荒縄で縛ったままの未整理の版木の調査にあたったとのことである。

短い滞在期間で版木の確認、水洗い、墨摺りを終えて同博物館の所蔵する版木の全容が確認できた。版木は江戸から名古屋、京、大阪の版元に転売されたり、削られて他の絵が彫られたりするが、ピゲローコレクションに入って震災や戦火を免れた貴重なものである。完全にそろったこの山桜の良質な硬い材に彫られた版木は日本に送られ、アダチ版画研究所で色刷りされて、1987年に渋谷の塩とタバコの博物館や地方巡回展覧会が開催された。

北斎の版木には次の3部作の完全な版木があった。

- 絵本 隅田川兩岸一覽 画工 北斎辰政 彫工 安藤円紫 （版元）大阪久宝寺町4丁目 前川善兵衛
  - － 初版は江戸本町 鶴屋喜右衛門
- 東都勝景一覽 画工 北斎辰政 彫工 安藤円紫 （版元）（寛政12年）江戸日本橋一丁目須原屋茂兵衛 下谷池之端仲町 須原屋伊八 天保11年 大阪心斎橋通博労町角 河内屋茂兵衛
  - － 初版は寛政12年 蔦重
- 絵本東都遊

隅田川兩岸一覽の裏には蔦重の版元印のある喜多川歌麿、歌川豊国、勝川春英などの大判の墨刷り用の版木もあった。

また、北斎漫画の一部の版木も見つかっている。ここに「北斎模様画譜」（明治17年版）として「新型小紋帳」（文政7年＝1824）が改題された版があった。題を改め序文は彫り直されているが、他は初版以来のオリジナルであると確認された。

### 11.2 デザイナー北斎

勝川派 北斎は1760年に本所の今の大江戸博物館の地で生まれ、16歳で木版彫刻を学び、19歳で浮世絵師勝川春章の門に入ったが14年間目が出ず師の没後勝川派を離れ、1974年琳派の俵屋宗理を名乗っている。

絵本の挿絵画家 40歳で北斎辰政を名乗り、10年間で挿絵絵本153冊を描いている。(凡そ1000枚)今回の版木はその頃のものである。

デザイナー北斎 50歳から60歳のころ、北斎漫画3000余図や今様櫛キセル形、新型小紋帳、略画早指南、一筆画譜など絵手本を多く描いている。

風景版画の大作 60歳頃から73歳が為一期で富岳36景、諸国瀧廻り、千絵の海、諸国名橋奇覧、琉球八景などの浮世絵版画の代表作を描いた

肉筆画の時代 晩年は天保の改革で水野忠邦に睨まれ江戸を離れることが多く、肉筆画が中心である。1849浅草聖天町で90歳の生涯を終えた。

### 11.3 フェノロサとビゲロー

明治15年(1883年)モース、フェノロ、ビゲローの3人のサアメリカ人が東京から京、奈良へ馬車、馬、駕籠を乗り継いで、美術品蒐集の旅に出た。2週間の旅の間に、噂を聞きつけた多数の古美術商が風呂敷に美術品を包んで押しかけた。

「北斎模様画譜」の版木はビゲローが明治22年(1899)に入手しており、このときボストンへ「そしてさらに、上述のセットよりは小さいものですが、同じく北斎の手になる墨摺りの模様一布地や紙に摺りだす地紋あるいはパターン類のデザイン一冊分の版木セットも手に入れました。これもまたなかなかよいものです」と書き送っている。

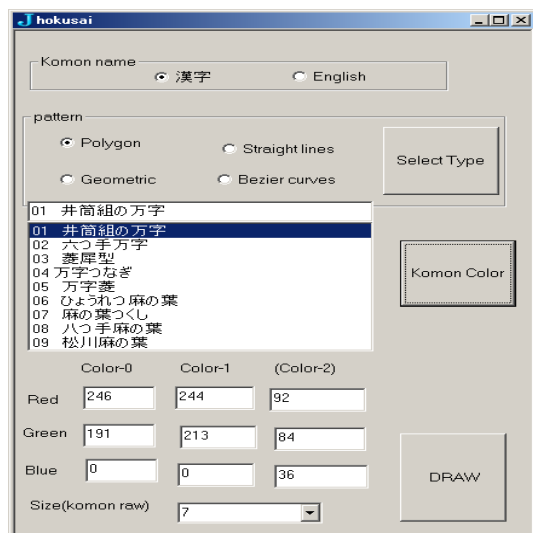
明治政府はモースやフェノロサ達を抱え外国人教師のサラリーは凡そ高級役人の5倍以上払っていた。ビゲローはボストン生まれの裕福な医者の子でハーバード医学校で学び、ドイツ留学やバスツールの下で学んでおり、細菌学の研究者でもあったが、フランス時代に古美術に目覚めている。父は医者になることを薦め、ハーバードで教えたりもするが、父と折が合わず、逃避して日本に渡った。日々得る金利がフェノロサの一月の収入程度はあったようで、26,000点に上る古美術を蒐集し、無償でボストン美術館に寄贈している。ビゲローは三井寺で天台密教の修行を続けていて、三井寺天明院のフェノロサの墓の隣に分骨された墓もある。

\*6

---

\*6 モースの蒐めた陶器4600点は10万ドルと言われた76,000ドルでボストン美術館が買い上げ、フェノロサの日本絵画760点はアメリカの富豪に、ボストン美術館への寄贈を条件に25万ドルも言われる額で譲渡された。  
現在価値は1ドル1円として1万倍すると凡その推測ができる

## 11.4 新形小紋帳にフォームを付ける



### 伊勢型紙

「日本の染色 6 江戸小紋 (華麗な江戸の伝統美)」に伊勢型紙の技法が紹介されている。紀伊藩の保護を受けた伊勢の白子と寺家町 (今の鈴鹿市) で伊勢型紙が彫られ、紀伊藩の保護で関所御免の通行手形や紀伊藩御用の提灯を宿に掲げることができることを許された白子・寺原の型紙株を持った商人の手で全国に売り歩かれた。

\*7

江戸小紋も伊勢型紙で刷られる。ここに小紋の技法も紹介されており、名人だと一寸四方に 1200 の穴を穿ったようだ。彫刀で 1 ミリに 1 個の穴を穿つ技法はそう真似ができるものではないが、幸いにも北斎の小紋デザインには点描がない。

### フォームのデザイン

50 余りの小紋デザインのスクリプトをコマンドラインで打ち込むのは骨が折れるのでフォームにまとめる。フォームの解説は以前に書いた著作と重複するが、J602 版を用いてなるべく丁寧に解説する。

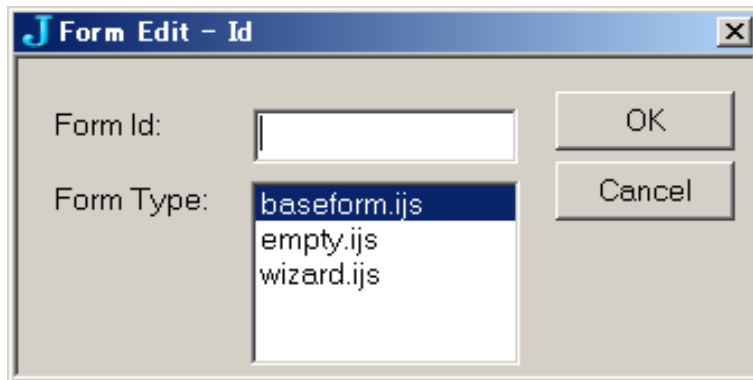
#### 11.4.1 フォームエディタを用いる

フォームエディタを用いて、フォームを作成する。

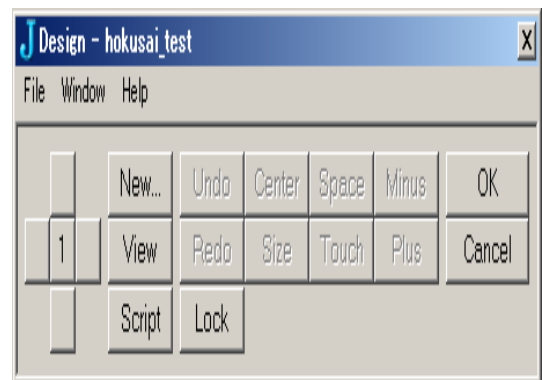
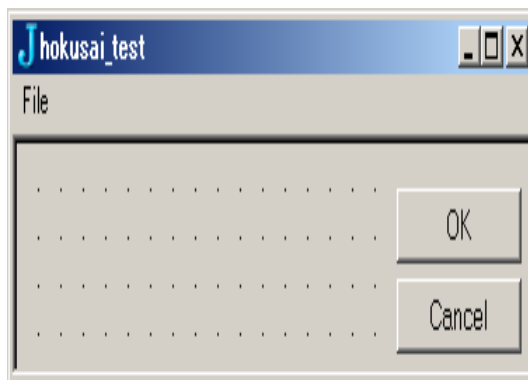
- *ijs* の画面から *Edit* → *Form Editor* をクリックする

---

\*7 根来衆の隠密にも都合がよかったようだ



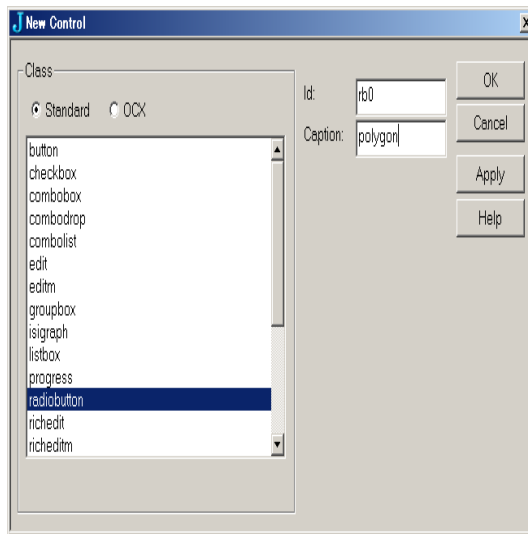
- *Form ID* に *HOKUSAI* と入れた。この名前が作成するフォーム全体のキーワードになる
- 次の2の画面が現れる。この2画面で作業する。



## 11.5 ラジオボタンを置く

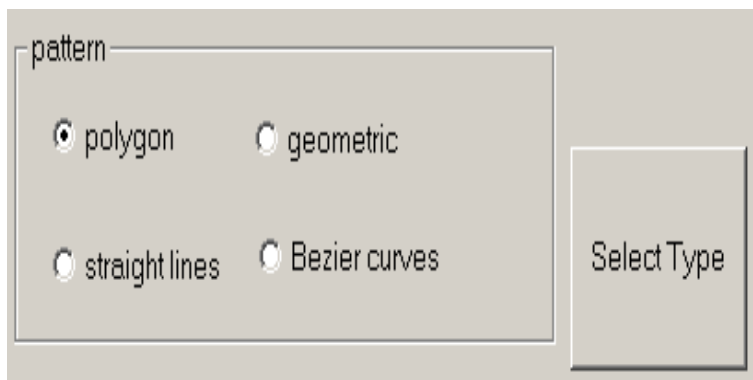
*Combodrop* に 50 余りの小紋のプログラムを登録しても選択が大変なので小分けする。ここでは4に分けた

- *NEW* でグループボックスを選ぶ。最初は全て左上に出てくるのでマウスドラッグで位置と枠の大きさを決める。(いつでも変更できるので大雑把でよい)
- *NEW* でラジオボタンを選ぶ。*id* は変数名になるので *rb0* とした。*caption* はフォーム上の見出しになるので *polygon* とした



- NEW を繰り返しラジオボタンを 4 個作成する (変数名 *rb0,rb1,rb2,rb3*)
- OK ボタンでスクリプトが出力される。以下はラジオボタンの部分。慣れればここを直接変更してもよい。

```
xywh 10 7 147 50;cc type groupbox;cn "pattern";
xywh 20 18 35 13;cc rb0 radiobutton;cn "polygon";
xywh 20 38 51 13;cc rb1 radiobutton group;cn "straight lines";
xywh 75 19 60 12;cc rb2 radiobutton group;cn "geometric";
xywh 76 38 54 11;cc rb3 radiobutton group;cn "Bezier curves";
```



- ラジオボタンはボタンをクリックしたときにグループ内でそのボタンを 1 に、他のボタンを 0 にする (中身は文字列である)
- ラジオボタンの情報を取り出す関数を作る。

```
hokusai_seltype_button=: 3 : 0
SEL=. ". rb0,rb1,rb2,rb3
```

....

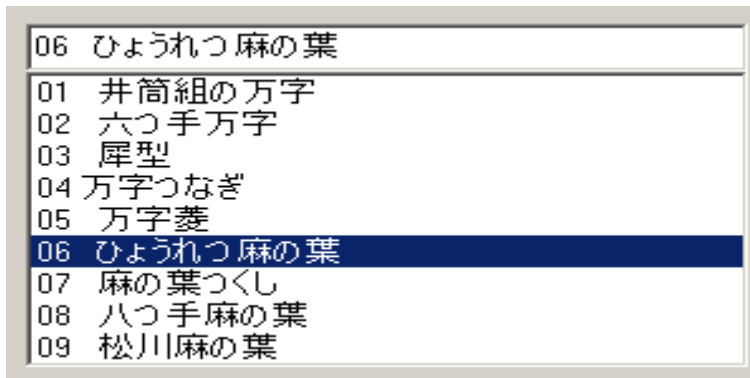
文字を数値化する。(".".) 1000 100 10 1 となる。

	rb0	rb1	rb2	rb3	value
1000	1	0	0	0	1000
0100	0	1	0	0	100
0010	0	0	1	0	10
0001	0	0	0	1	1

もう一つラジオボタンを作ってコンボドロップに表示する言語を「漢字/English」の選択とした

## 11.6 コンボドロップを作成する

- NEW でコンボボックスを選び、ID(変数名) を *komon* とする。マウスで位置と大きさを決める



```
%itemize}
xywh 161 26 48 35;cc seltype button;cn "Select Type";
xywh 9 86 143 78;cc komon combobox rightmove;
```

## 11.7 ボタン ( 1 )

コンボボックスにラジオボタンの情報を送り、小紋パターンを表示する *hokusai\_seltype\_button* を作る

- NEW でボタンを選ぶ。位置と大きさを大雑把に設定する
- *hokusai\_select\_button* のスクリプトを書く。
- コンボボックス内への小紋名の表示関数を作る

```
POLYSET=: '"01 井筒組の万字" "02 六つ手万字" "03 犀型" "04 万字つなぎ" "05
万字菱" "06 ひょうれつ 麻の葉"'
POLYSET=: POLYSET, '"07 麻の葉つくし" "08 八つ手麻の葉" "09 松川麻の葉" '
```

```

set_polygon=: 3 : 0
wd 'set komon ',POLYSET
wd 'setselect komon 0'
)

```

- ボタンにコンボボックスへの表示関数を割り付ける。

```

hokusai_seltype_button=: 3 : 0
SEL=: ". rb0,rb1,rb2,rb3
NB. rb info is 1000 0100 0010 0001->(".)-> 1000 100 10 1
select. SEL
case. 1000 do. set_polygon ''
case. 100 do. set_straight ''
case. 10 do. set_geom ''
fcase. do. set_bezier ''
end.
)

```

コンボドロップ内の小紋名称の英文表記。ラジオボタンで 10/1 で選んだ言語の表示を *Select Type* ボタンに割り付けた。(ここを押さないと実行中には変更されない。)

```

POLYSETE=: "'01 Wellframe Gammadion" "02 6 summit Gammadion" "03 Diamond Rhino"
"04 Connected Gammadion" "05 Diamond Gammadion"'
POLYSETE=: POLYSETE, ' "06 Line-up Hemp leaves" "07 Many Hemp leaves"
"08 8 hand Hemp leaves" "09 Pine Hemp leaves" '

```

```

set_e_polygon=: 3 : 0
wd 'set komon ',POLYSETE
wd 'setselect komon 0'
)

```

## 11.8 ボタン ( 2 )

選ばれた小紋のタイプの情報をカラー窓に伝え、カラー候補と使わないカラー窓を表示する *hokusai\_komontype\_button* を作る。ボタンを押すとこの関数が実行されるボタンを作る

1. 選択した小紋パターンの No. を抜き出す。

```

pick_komon_patarn=: 3 : 0
NB. pol/st/bezier and nr. of band

```



```

SEL=. ". rb0,rb1,rb2,rb3
NB. rb is 1000 100 10 1
KOMON=. ". 2{. komon NB. pick only number from (05 繋ぎ蕨)
SEL;KOMON
)

```

漢字や小紋名を取得して認識させるのは大変なので、先頭に *No.* を付け、この部分を抜き出し、数値化して各小紋描画関数と対応させる

```

    komon NB. SEL=1 komon=01
01 松並

```

## 2. 選択した小紋のカラー情報を加工してセットしている

```

hokusai_komontype_button=: 3 : 0
'SEL0 KOMON0'=: pick_komon_patarn '' NB. type and patarn
CTABLE=. select_colortable SEL0 NB. PSGB-COLOR
CLINE0=. (<: KOMON0){CTABLE NB. select color line
CLINE=. order_set_color CLINE0 NB. remove null set(not infty)
set_color CLINE NB. set color table
)

```

## 11.9 カラー

### 11.9.1 カラーボックス

北斎の小紋帳は墨摺りだが、スクリプトはカラーを付けてみた。最初に規則を設けなかったので後で数えたら、最大で4色は一つ、後は2,3色が多い。

- カラーは *RGB* で入れる。各 0-255 で合計 1600 万色になる。
- カラーブックをプリセットする方法もあるが、リストが長くなりすぎるので止める。各自でカラーブックを手元に色コードを入れてほしい
- カラーの入力窓は3色とし、*RGB* で9個の窓を設ける。*NEW* から *edit* で設定し、*ID* が変数名となる
- 見出しは *NEW* から *static* で記入する

	Color-0	Color-1	(Color-2)
Red	<input type="text" value="144"/>	<input type="text" value="230"/>	<input type="text" value="144"/>
Green	<input type="text" value="93"/>	<input type="text" value="75"/>	<input type="text" value="93"/>
Blue	<input type="text" value="84"/>	<input type="text" value="107"/>	<input type="text" value="84"/>
Size(komon raw)	<input type="text" value="25"/>		

### 11.9.2 カラーテーブル

先頭に個数を取り、最大3個の RGB 毎のボックスとした。各ボックスは ; で区切る

NB. geometric komon

G01=: 2;234 213 107;228 211 162	NB. hokusai_wachigai2
G02=: 2;255 0 0; 0 64 0	NB. hokusai_sw
G03=: 2; 145 76 53; 105 60 52	NB. hokusai_h3
G04=: 2; 0 255 0 ;0 0 255	NB. hokusai_ks
G05=: 2; 234 213 107;228 211 162	NB. hokusai_hy2
G06=: 1; 0 0 255	NB. hokusai_kk1
GCOLOR=: G01,G02,G03,G04,G05,:G06	

### 11.9.3 見本カラーの表示

1. ブランクから RGB を入力するよりも気に入ったカラーやテスト時のカラーを当初に表示する方が良く  
と考え、このため hokusai\_komontypte\_button ボタン (2) を作った。選択すると指定カラーがボッ  
クスに入る。

```
select_colortable=: 3 : 0
select. y    NB. Sel0
  case. 1000 do. CTABLE=. PCOLOR
  case. 100  do. CTABLE=. SCOLOR
  case. 100  do. CTABLE=. GCOLOR
  fcase.    do. CTABLE=. BCOLOR
end.
)
```

2. 入力しないボックスには\_(infinity) を入れる。(infinity は数で横棒ではない。)
3. 選択したカラーの対処法

- (a) 入力を保存し、ボタンでスクリプトを書き換える
  - (b) 変更はスクリプト (カラーテーブル) を手で書き換える
- 不意にボタンを触って書き換えられるのも困るので、カラーテーブルは手で書き換えることとした

#### 1. 各小紋プログラムへの入力

- (a) 各小紋スクリプトのカラーの入力は 'Color0 Color1 Color2'=. x のようになっている
- (b) 各小紋スクリプトは単独でも用いるので、描画関数に空入力のような余計な手法は避けて、*Form* 側で全て捌く
- (c) 2色なら 'Color0 Color1'=. x となっており、ボックスの個数が異なると描画関数が入力を受け付けない
- (d) カラーリストのボックスの数が3で無い場合はブランクに *null* が入ったボックスができる。*null* の部分はカラーテーブルの最初につけた個数で判断し、切り落とす
- (e) *Color0* が1個の場合はボックスを用いなくて3個の数 (*RGB*) として入力しなければならない仕様となっているので、1個の場合はボックスを開く

### 11.10 描画する小紋の数 (縦横) を決める

コンボドロップを用いてサイズを選ぶ。画面の広さと、グラフィックスメモリ、実行時間をみて決める

#### 1. *set* 選択候補の個数はプリセットした

```
set_size=: 3 : 0
wd 'set size "1" "2" "5" "7" "10" "12" "15" "20" "25" '
wd 'setselect size 1'
)
```

#### 2. 描く小紋の数は。横は縦の 1.5 倍 (整数になるよう切り下げた数) とした

```
pick_size=: 3 : '<. 1 1.5 * 2 # ". size ' NB.column make 1.5 times of raw
```

### 11.11 DRAW ボタン (ボタン 3 )

*OK(=DRAW)* ボタンで描く

#### 1. *ok* ボタンに割付

(1) *x (COLOR)* (2) 小紋を描く関数を呼び出す (3) *y* サイズをセットする

```
hokusai_ok_button=: 3 : 0
SIZE=. pick_size '' NB. using y
'SEL0 KOMON'=. pick_komon_patarn '' NB. select nr for komon_script
COLOR=: order_color_table ''
```

```

COLOR hokusai_draw SEL0;KOMON;SIZE
)

```

2. *draw* 関数を決めて、呼び出す

```

hokusai_draw=: 4 : 0
'SEL0 KOMON SIZE'=: y NB. type / fn-name
NB. 'COLOR0 COLOR1 COLOR2 SIZE' =. x
COLOR=: clean_infty x
NB. polygon
if. 1000 = SEL0 do. NB. poly
  select. KOMON NB. drop
    case. 1 do. COLOR hokusai_im SIZE NB. color3 fix
    case. 2 do. COLOR hokusai_mm SIZE
    case. 3 do. COLOR hokusai_s SIZE
  .....

```

## 付録 A J の入手とインストール

- J は <http://www.jsoftware.com> から無償で入手できる。
- WIN32/64 MAC PPC/INTEL LINUX32/64 で J602 とフォームは動く  
J701 は HTML 版で iPad Android 版もあるがグラフィックスの環境は大きく変わっている
- J はレジストリを切っているので、USB、CDROM からでも動く。インストーラーで一度インストールしてから任意の箇所にコピーすればよい。
- コピーした場合は *Jpath* が変更されない (場合が多い)。

## 付録 B 北斎の小紋の動かし方

- プログラムは <http://japla.sakura.ne.jp> から DL できる
- 解凍して任意のフォルダにコピーする。
- *hokusai\_form.ijs* の *hpath* を編集してフォルダの位置を書き込む (変更はこの 1 箇所のみ)
- 変更は *hokusai\_form.ijs* を読み込んで、EDIT の Read only を解除すれば可能
- *hokusai\_run* ” は自動起動にしてある。ijx の画面から打ち込んでもよい。
- RGB カラーはボックスに数字を入れると指定色で描けるが、Script の方は変更されない。変更する場合は *hokusai\_form.ijs* のカラーテーブルの該当箇所を探して書き換える。

## References

永田生慈監修解説「北斎の絵手本 (3)」岩崎美術社 1986  
「ボストンで見つけた北斎展 (ボストン美術館の新版木発見)」東京放送 1987  
ファー・インク編「カラーハンドブック」MDN Corporation 2010  
<http://kinetigran.com/mck/Calculus/BezierCurves/BezierMatrix.html>  
J602 J701 はトロントから DL 出来ます  
<http://www.jsoftware.com>  
スクリプトは次から DL できます  
<http://japla.sakura.ne.jp> の symposium 2011,2012