JAPLA 研究会資料 2020/1/18

Ｊによるゴスパー曲線のグラフィックスーその２
ー蜂の巣タイリングからどうやってゴスパー図形を描くかー

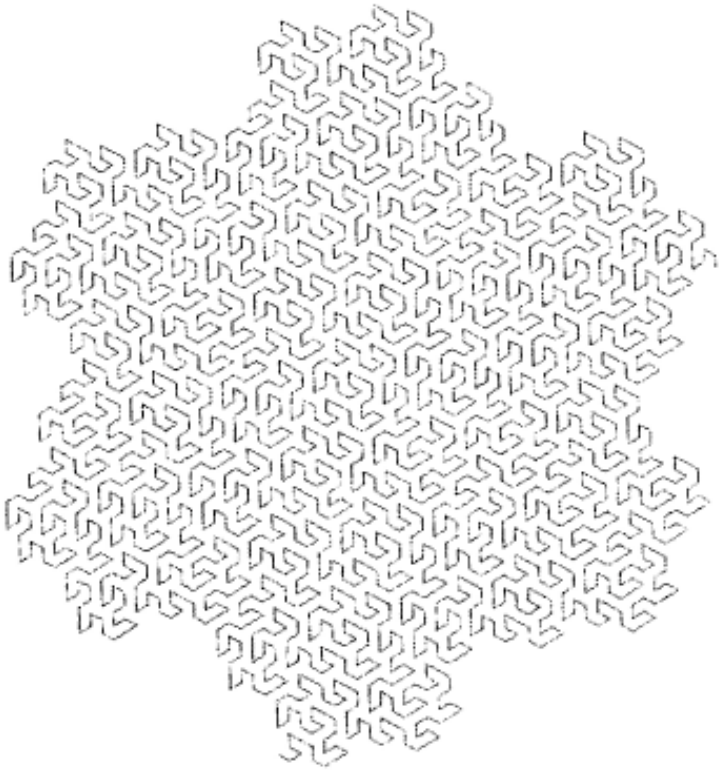
西川　利男

前回の報告[1]、その１では蜂の巣模様のタイリングどまりであったが、今回は前とは違う方法で蜂の巣タイリングを行い、ゴスパー図形[2],[3]を描くことに挑戦した。

[1] 西川利男「Ｊによるゴスパー曲線のグラフィックスーその１－Ｊグラフィックス・ツールの整備と蜂の巣模様のタイリング JAPLA 研究会資料、2019/12/7.

[2] マーチン・ガードナー、一松信訳「マーチン・ガードナーの数学ゲームⅠ」p.92-95 別冊日経サイエンス１７６(2010).

[3] 福田宏、中村義作「エッシャーの絵から結晶構造へ」p.81-92 海鳴社(2010).

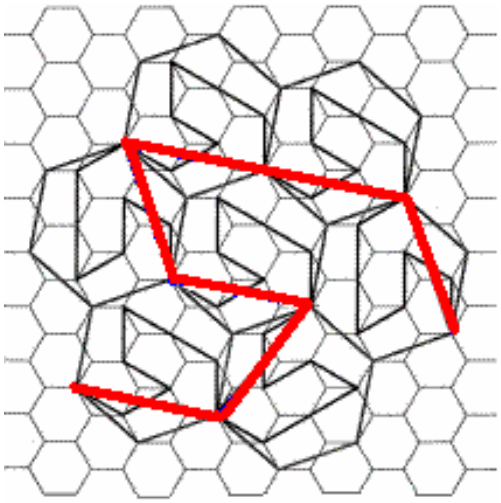


１．ベースゴスパー曲線とローカルゴスパー曲線

ゴスパー曲線とは何かの説明について、前回の報告[1]から採録する。

まずゴスパー曲線を描くには、まず蜂の巣模様(Honeycomb)のタイリング(下図)を作る。、つぎに、その正六角形の適当な頂点を結んで、基本のゴスパー曲線(赤)をつくる。これを元に縮小、連結を繰り返して、次第に細かくして(黒)、ゴスパー曲線とする。

このように、まず蜂の巣模様タイリングを作ることがスタートとなる。



赤の部分をベース・ゴスパー、黒い部分をローカル・ゴスパーと呼ぶことにする。

なお、ゴスパー線のような線分の集合のグラフィックスでは、あるいはゴスパー図形の一つの数学関数としては扱えないので、アート（職人的なわざ？）としてやらざるを得ず、なかなかやっかいである。

１．J グラフィックス・ツール

これまで、オリンピック・エンブレム、反転グラフィックスなどで、すこしずつ整備してきた J のグラフィックス・ツールの定義動詞はつぎのものである。

・図形の回転　　(degree) **rotate** (picture)　原点(0, 0)でおこなう

・図形の拡大、縮小　(value) **ensize** (picture)　原点(0, 0)でおこなう

(enlarge, reduce をまとめて ensize とした)

・図形の移動　　(x, y) **shift**　(picture)

・色つきで図形を描画

(R, G, B) **colorpolylines** picture　複数の点をつないで、指定した色の線を引く

(R, G, B) **colorpolygon** picture　点をつないで多角形の内部を色で塗りつぶす

すべて、右引数の picture データは、点の値(x, y)のボックス集合で示す.。

なお、これらの J プログラム定義は最後に示してある。

２．蜂の巣図形、ゴスパー図形のグラフィックスの考え方

基本の操作はつぎのようである。

・各頂点の座標値(x, y)を求めること

・それぞれの規則に従って、結んで図形を描く

ポイントは、頂点座標の各点を示すインデクス付けをどうするかになる。

３．蜂の巣タイリングとそのインデックス付け

蜂の巣図形の元となる正六角形の点座標 TH をつぎのように決める。

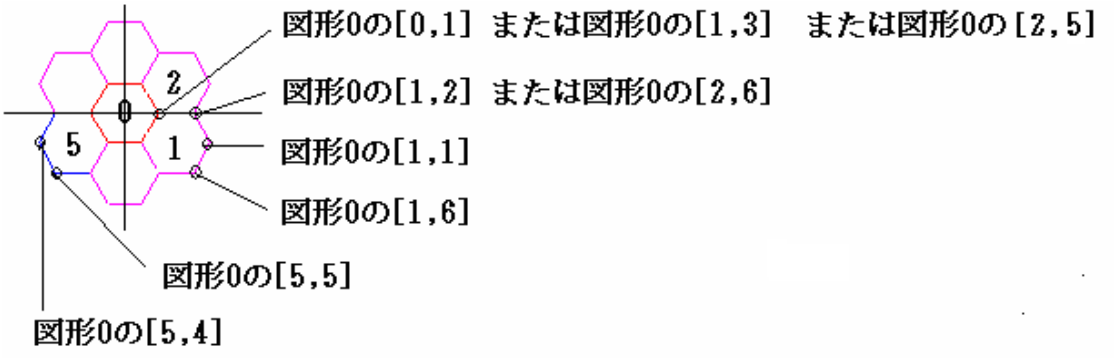
TH =: (2, 0);(1, RT3);(_1, RT3);(_2, 0);(_1, rt3);(1, rt3);(2, 0);(4, 0)

ここで　RT3 =: %: 3　　NB. √3 ,　rt3 =: - RT3　　NB. -√3

以下、具体例で説明する。

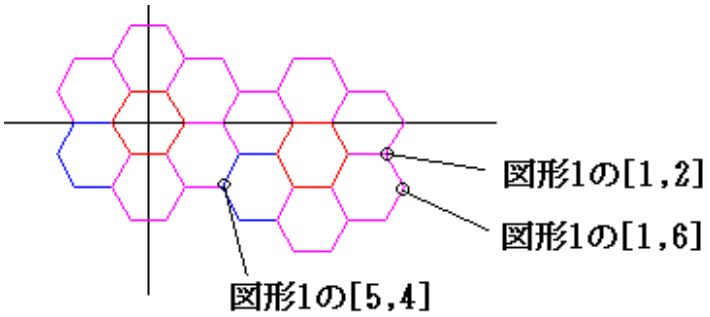
３．１　蜂の巣タイリング図形０のインデックス付け

画面の中央に、中心座標(x, y) = (0, 0)で、正六角形の集合を作って、これを、蜂の巣図形０と呼び、各頂点座標のインデクスを次のように決める。インデクスは[と]とでしめす。座標値(x, y)とは違うことに注意。



３．２　蜂の巣タイリング図形１のそのインデックス付け

上の図形０に接触させて、その右下に図形１を付加する。ここで接触させるには正六角形の辺の長さから、図形１の中心座標(x, y) = (9, rt(=-√3))とすればよい。



このようにして、図形０の周りに、図形1、図形2、..., 図形６を貼り付けて、正六角形７個からなるローカル蜂の巣図形が出来上がる。

４．蜂の巣タイリングのグラフィックス

４.1　インデックス表示の座標値の値を得て、図形を描く

前章のように定義したインデックスは一種の論理的座標と見ることができる。
これから画面上の図形の物理的座標値(x, y)を得るには、インデックスを引数として次のサブルーティン動詞 jj により、計算して求める。

```
jj =: 3 : 0
'X Y' =. y.
gg1 =. (sz * ((X + 3),  (Y + RT3)) )  shift ; sz ensize HX
gg2 =. (sz * ((X + 0),  (Y + 2*RT3))) shift ; sz ensize HX
gg3 =. (sz * ((X + _3), (Y + RT3) ))  shift ; sz ensize HX
gg4 =. (sz * ((X + _3), (Y + rt3) ))  shift ; sz ensize HX
gg5 =. (sz * ((X + 0),  (Y + 2*rt3))) shift ; sz ensize HX
gg6 =. (sz * ((X + 3),  (Y +  rt3)) ) shift ; sz ensize HX
gg0 =. (sz * ((X + 0),  (Y + 0)) )    shift ; sz ensize HX
gg =: (<gg1), (<gg2), (<gg3), (<gg4), (<gg5), (<gg6), (<gg0)  NB. global
)
```

これを用いて、以下の動詞により７つの６角形＝蜂の巣をまとめて描く。

```
NB. draw local_hexagons ================================
draw_7hexagon =: 3 : 0
i =. 0
while. i < 7
  do.
    select. i
      case. 3 do. (0 0 255)   colorpolylines , > 3{gg
      case. 6 do. (255 0 0)   colorpolylines , > 6{gg
      fcase.  do. (255 0 255) colorpolylines , > i{gg
    end.
    i =. i + 1
  end.
glshow ''
)
```

例えば、ボタン JXX1 に対応する蜂の巣図形１を描くには、つぎのようにする。
蜂の巣図形１の中心座標値(x, y) = (9, -√3)であるから、蜂の巣図形１を成す７個の正六角形の頂点座標値はサブルーチン jj を用いて次のように求められる。

```
  GS1 =: jj 9, rt3
  GS1
+-------------------------------------------------------------------+--------- --
|7 0 6.5 0.866025 5.5 0.866025 5 0 5.5 _0.866025 6.5 _0.866025 7 0|5.5 0.866 --
+-------------------------------------------------------------------+--------- --
  $GS1
7
```

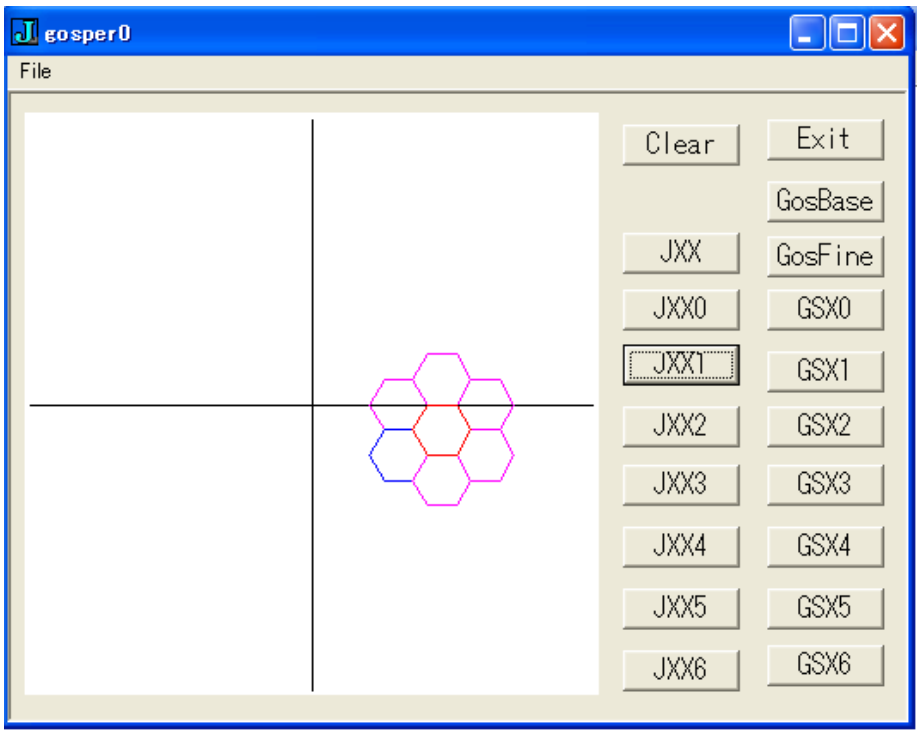
つまり、蜂の巣の最初の７つの正六角形の座標値(x, y)は

```
   > 0{GS1
7 0 6.5 0.866025 5.5 0.866025 5 0 5.5 _0.866025 6.5 _0.866025 7 0
```

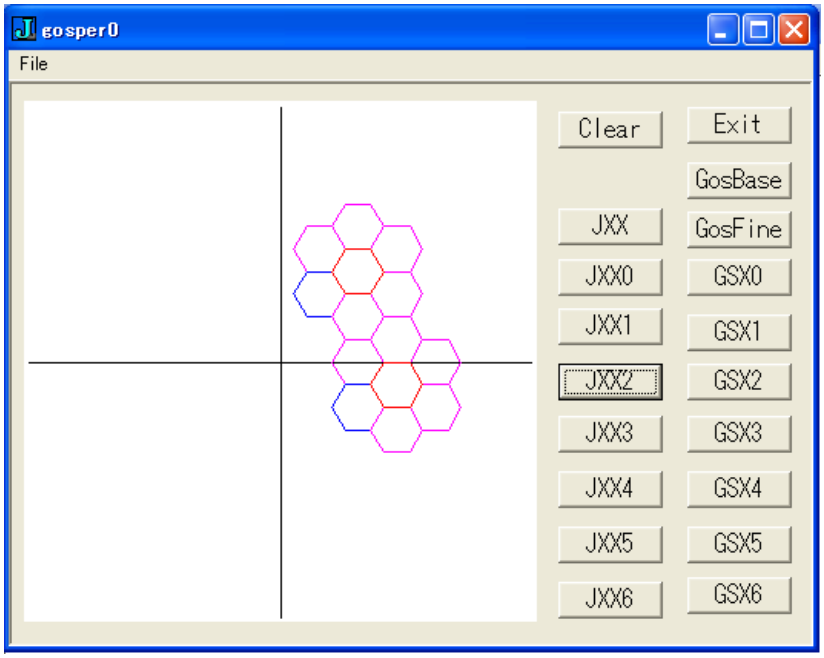
より、 (7, 0), (6.5, 0.866025), ... (7, 0)となる。従って、これを描画するには draw_7hexagon を用いて行う。これを描画実行ボタン JXX1 とすればよい。

```
invergraph_JXX1_button=: 3 : 0
GS1 =: jj 9, rt3
draw_7hexagon GS1
)
```

実行すると次のようになる。



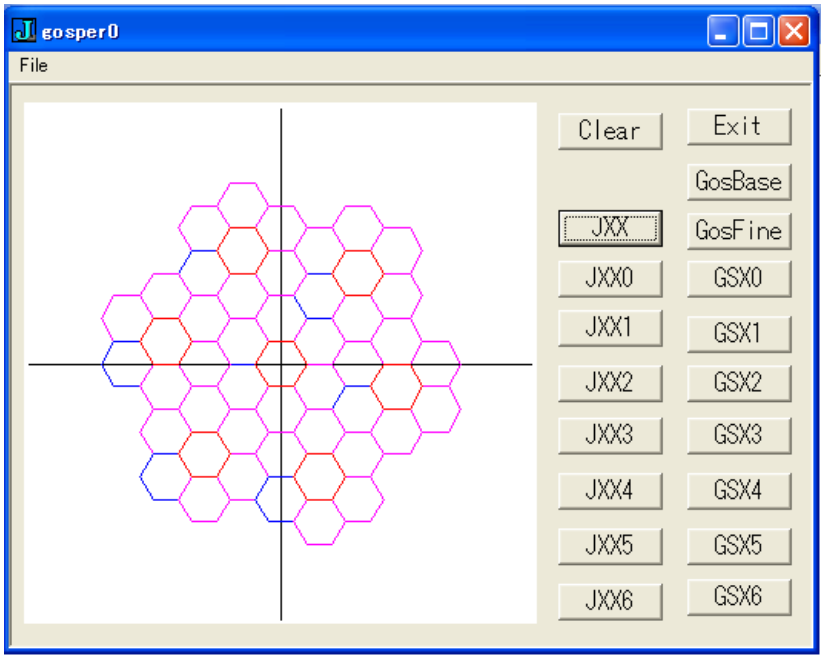
続いて、ボタン JXX2 を押すと、次のようになる。



４．２　蜂の巣タイリングのグラフィックスの実際

Ｊのプログラム run により実行して、ボタン JXX0, JXX1, JXX2, JXX3, JXX4, JXX5, JXX6 を順次押すと、以下のような蜂の巣タイリングが現れる。なお、図形を次々タイリングしたときに、区別できるように、正六角形の辺の色を変えて表示した。この図は前回ものとは異なり、点対称ではなくやや斜めになっている。

この図を一度で表示するには、ボタン JXX を押す。グローバル値 gg がセットされる。



５．ゴスパー図形のグラフィックス

５．１　ローカル・ゴスパー図形の作成

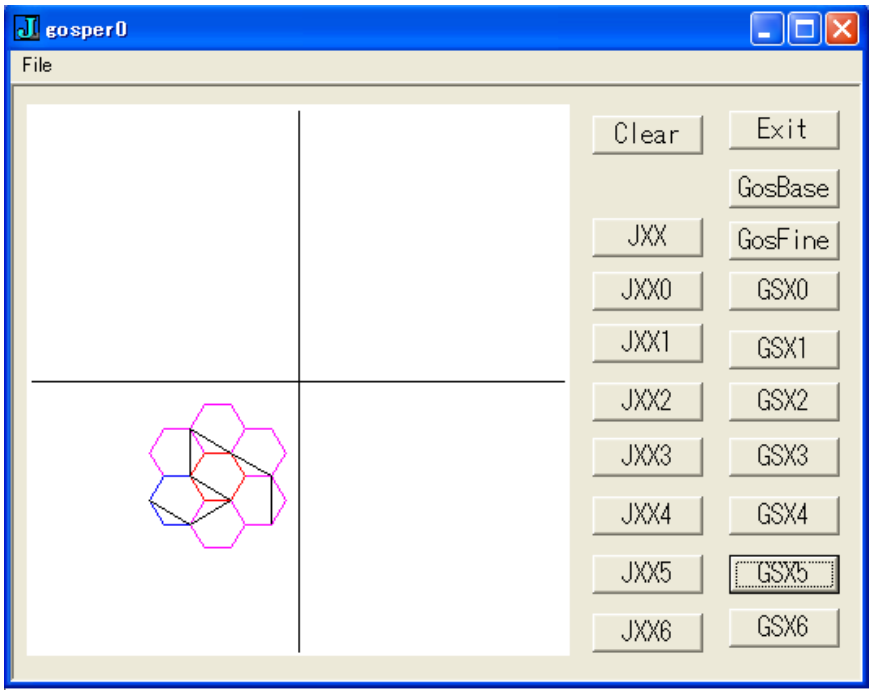
スタートするには左下の図形５から始める。そのためにはボタン JXX5 を押して図形

5

のローカル蜂の巣図形５を表示する。つぎにその隣のボタン GSX5 を押すと、ローカルゴスパー図形５が現れる。

```
invergraph_JXX5_button=: 3 : 0
GS5 =:  jj _6, (_4 * RT3)
```

```
draw_7hexagon GS5
)
gosper0_GSX5=: 3 : 0
gs50 =. > 3{ <"(1) (7,2)$ > (4 -1){ GS5
gs51 =. > 3{ <"(1) (7,2)$ > (5 -1){ GS5
gs52 =. > 3{ <"(1) (7,2)$ > (6 -1){ GS5
gs53 =. > 3{ <"(1) (7,2)$ > (7 -1){ GS5
gs54 =. > 3{ <"(1) (7,2)$ > (2 -1){ GS5
gs55 =. > 3{ <"(1) (7,2)$ > (1 -1){ GS5
gs56 =. > 5{ <"(1) (7,2)$ > (1 -1){ GS5
gs57 =. > 5{ <"(1) (7,2)$ > (6 -1){ GS5
(0 0 0) colorpolylines gs50, gs51, gs52, gs53, gs54, gs55, gs56, gs57
glshow ''
)
```



これをもう少し使いやすくするため、図形番号pとそのインデクス[q, r]から次のように座標値(x, y)を返す両側引数をもつサブルーティン動詞fig_xyを作った。

```
NB. coversion subroutine ====================================
NB.  from Fig.No.p and its indices [q, r]
NB.  to (x, y) values
NB.  e.g.
NB.   gb0 =. 5 fig_xy (4,3) => _5.5 _4.33013
fig_xy =: 3 : 0
:
p =. x.      NB. Fig.No.
'q r' =. y. NB. index
P =. 'GS',(": p)
xy =. r {"(2) (7 2)$ > (<:q) { (". P)
)
```

これを用いると、先のプログラムは、以下のように見やすくなる。

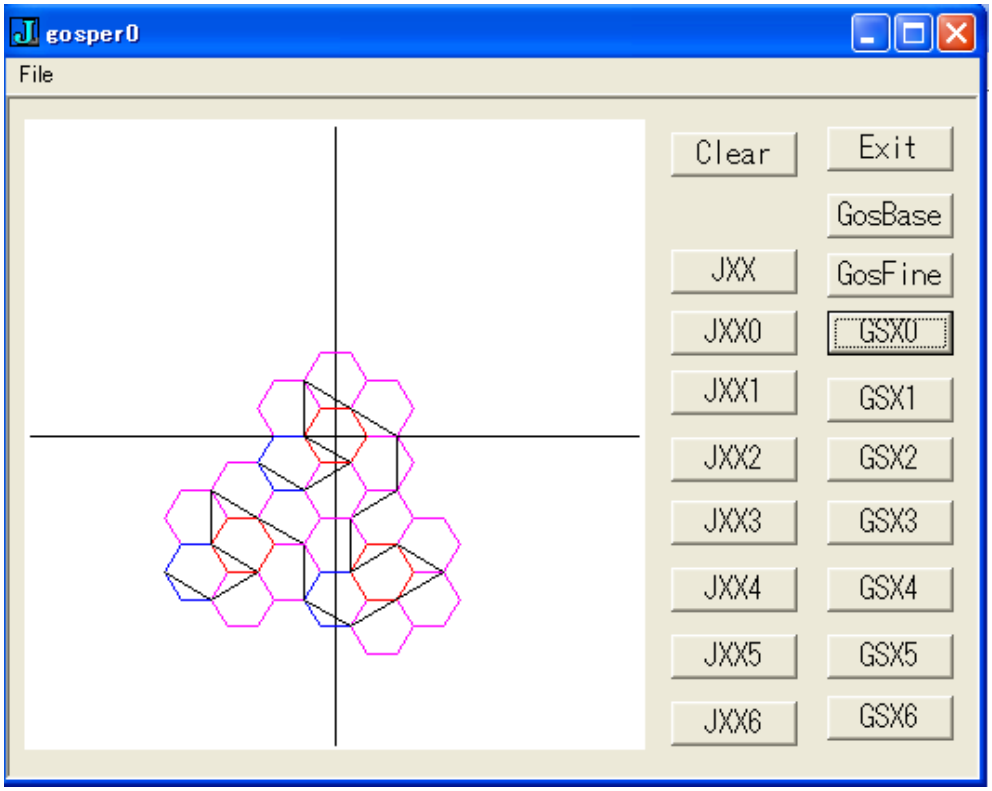
```
gosper0_GSX5_button=: 3 : 0
gs50 =. 5 fig_xy (4, 3)
gs51 =. 5 fig_xy (5, 3)
gs52 =. 5 fig_xy (6, 3)
gs53 =. 5 fig_xy (7, 3)
gs54 =. 5 fig_xy (2, 3)
gs55 =. 5 fig_xy (1, 3)
gs56 =. 5 fig_xy (1, 5)
gs57 =. 5 fig_xy (6, 5)
(0 0 0) colorpolylines gs50, gs51, gs52, gs53, gs54, gs55, gs56, gs57
glshow ''
```

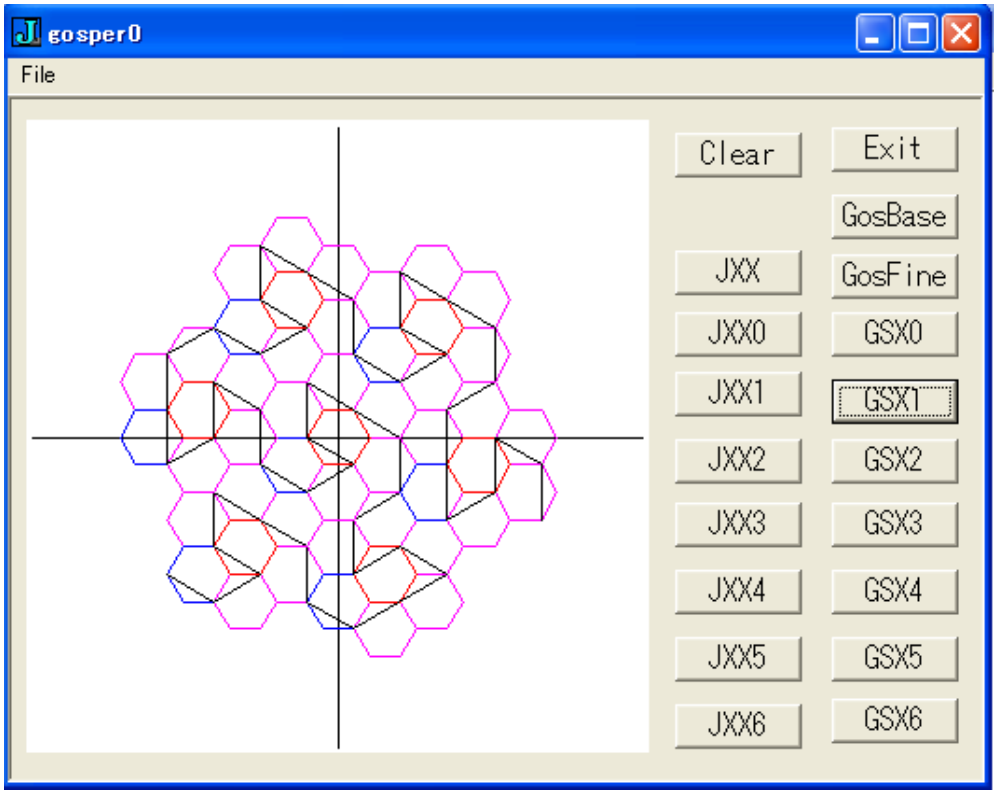
実行するには、以下のようにする。

前提となる蜂の巣図形の頂点座標のx,y値の集合、グローバル値ggを得るためにあらかじめボタンJXXで蜂の巣図形を表示させた後、一度Clearボタンを押して画面をクリアする。その後、ボタンGSX5を押せば、前と同様に、ローカルゴスパー図形５が現れる。

なお、ローカルゴスパー図形のボタンGSX6, GSX0, GSX4, GSX3, GSX2, GSX1 に対する図形の頂点座標の取り出しは、一律ではなくそれぞれ固有であり、Jプログラムは固有のものである。プログラムは後述。

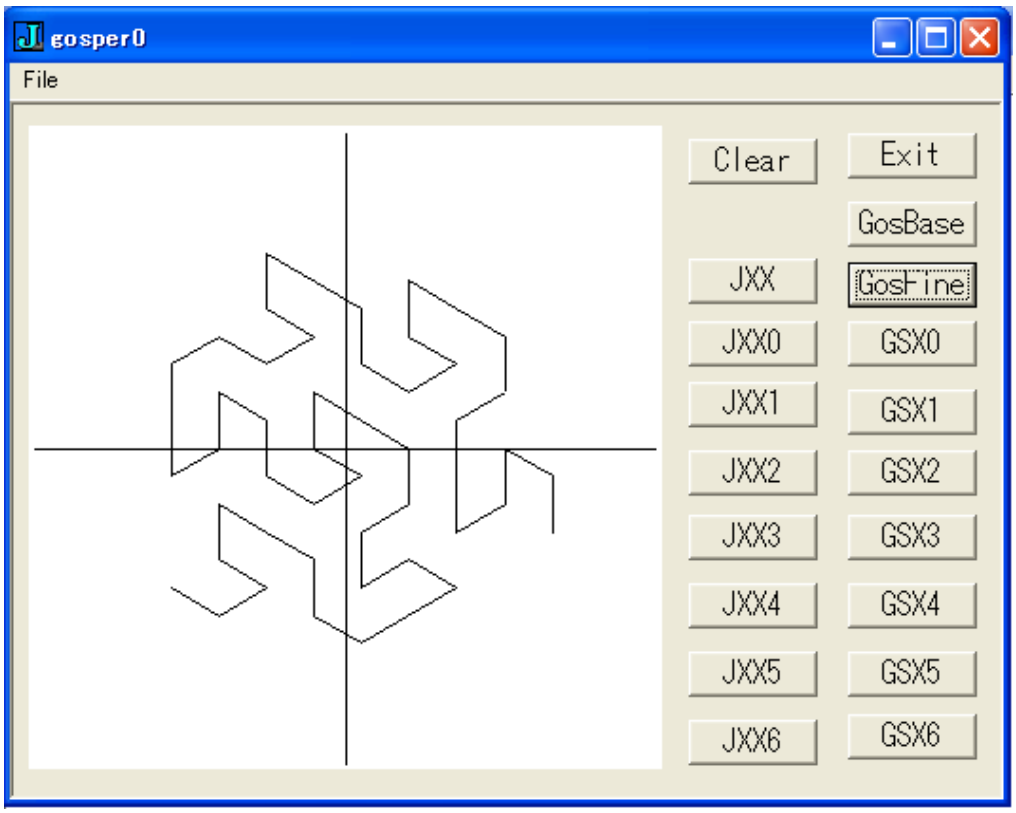
ローカルゴスパー図形全体を次々と連続して見るには、ボタン GSX5 に続いて、GSX6, GSX0, GSX4, GSX3, GSX2, GSX1 と押して行く。





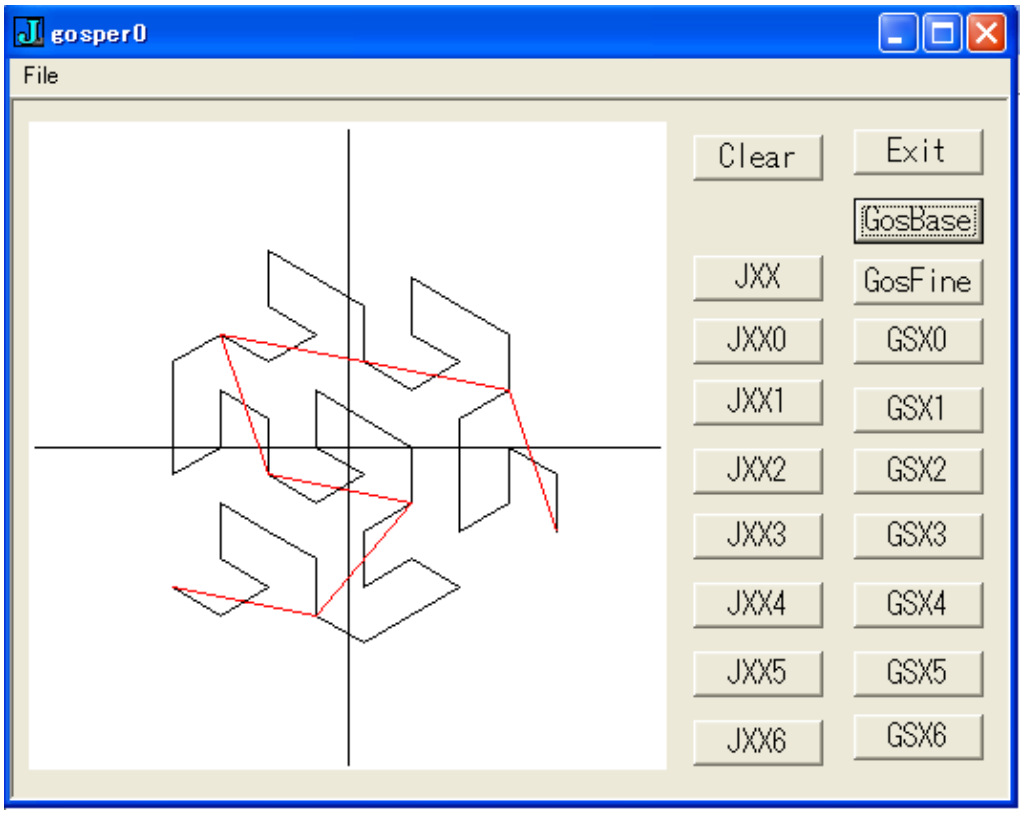
５．２　ローカル・ゴスパー図形のグラフィックス

ローカル・ゴスパー図形だけを描画するには、ボタン JXX で蜂の巣図形を表示し、グローバル値 gg を得た後、Clear ボタンで画面をクリアした後、あらためてボタン GosFine を押す。



６．グローバル・ゴスパー図形のグラフィックス

さらに、グローバル・ゴスパー図形を描画するには、ボタンGosBaseを押す。



７．おわりに－ゴスパー図形のグラフィックス

やっとゴスパー図形にたどり着いたという次第である。

文献にあげた成書[2],[3]では、わずか二、三ページで説明しているこの図形だが、プログラミングして、グラフィックスとして描くとなると、こんなにややこしいとは思っても見なかった。

数学的なコンピュータ処理というより職人技のアートであり、私にとっては、まさに怪物図形との格闘だった。

Jプログラム・リスト

```
NB. Gosper0.ijs by T. Nishikawa revised 2020/1/23
NB. Honeycomb Tile and Gosper Figure Graphics
NB.   imported from Gosper3.ijs, and revised compactly

wr =: 1!:2&2

load 'trig'
load 'gl2'

GOSPER0=: 0 : 0
pc gosper0;
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 220 7 34 11;cc cancel button;cn "Exit";
xywh 4 5 167 151;cc hexgos isigraph;
xywh 178 51 34 11;cc JXX0 button;
xywh 178 65 34 11;cc JXX1 button;
xywh 178 81 34 11;cc JXX2 button;
xywh 178 96 34 11;cc JXX3 button;
xywh 178 112 34 11;cc JXX4 button;
xywh 178 128 34 11;cc JXX5 button;
xywh 178 144 34 11;cc JXX6 button;
xywh 220 51 34 11;cc GSX0 button;
xywh 220 67 34 11;cc GSX1 button;
xywh 220 81 34 11;cc GSX2 button;
xywh 220 96 34 11;cc GSX3 button;
xywh 220 112 34 11;cc GSX4 button;
xywh 220 128 34 11;cc GSX5 button;
xywh 220 143 34 11;cc GSX6 button;
xywh 178 36 34 11;cc JXX button;
xywh 178 8 34 11;cc Clear button;
```

```
xywh 220 37 34 11;cc GosFine button;
xywh 220 23 34 11;cc GosBase button;
pas 6 6;pcenter;
rem form end;
)

run =: gosper0_run
gosper0_run=: 3 : 0
wd GOSPER0
NB. initialize form here
x0 =: 500
y0 =: 500
sz =: 0.5  NB. size
NB. sz =: 0.7
Box =: 0
draw_xy ''
glshow ''
wd 'pshow;'
)

gosper0_close=: 3 : 0
wd'pclose'
)

gosper0_cancel_button=: 3 : 0
gosper0_close''
)

gosper0_Clear_button=: 3 : 0
glclear ''
x0 =: 500
y0 =: 500
draw_xy ''
sz =: 0.5 NB. size
glshow ''
)
```

```
NB. Graphics Utilities by TN ============================================
adj0 =:  3 : 0    NB. centered
1 adj0 y.
:
NB. 500 + 100 * x. * y.
(x0, y0) + "(1) > cut2 100 * x. * y.  NB. y-axis down
NB. (500, 400) + "(1) > cut2 100 * x. * y.  NB. y-axis down
)

adj =:  3 : 0     NB. start left down
1 adj y.
:
NB. (500&+)@(100&*)
200 + 40 * x. * y.
)

NB. ensize = enlarge or reduce
NB. eg. 0.4 ensize TH (data points)
ensize =: 3 : 0
:
siz =. x.
y. * (L:0) siz
)

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


NB. eg. 30 rotate figure(x0, y0, x1, y1, .. ) around (0, 0)
rot =: 3 : 0
```

```
:
'x y' =. y.
t =. x.
((x*cosd t) - (y*sind t)), ((x*sind t) + (y*cosd t))
)

rotate =: 3 : 0
:
ANG =. x.
RXY =. cut2 , > y.
RX =. {."(1) > ANG rot L:0 RXY
RY =. {:"(1) > ANG rot L:0 RXY
, |: RX ,: RY
)

NB. eg. (1, 2) shift x, y
shift =: 3 : 0
:
'Xs Ys' =. x.
, (Xs, Ys) +"(1) > cut2 y.
)

colorpolygon =: 3 : 0
:
glrgb x.
glbrush ''
glpen 1 0
NB. glpolygon , sz adj0 y.
glpolygon , sz adj0 y.
)

colorpolylines =: 3 : 0
:
glrgb x.
glbrush ''
glpen 1 0
```

```
gllines , sz adj0 y.
)

draw_xy =: 3 : 0
glrgb 0 0 0
glpen 1 0
gllines 10 500 990 500
gllines 500 10 500 990
)

circle =: 3 : 0
'x y r' =. y.
TH =. 15 * i. 25        NB. circle = 15 deg * 24 times
XX =. x + r * cosd TH
YY =. y + r * sind TH
, XX ,. YY
)

NB.         imported         from         gosper3.ijs         =         2019/12/11
=======================================
NB. Honeycomb Pattern Values / Global Defined =======
RT3 =: %: 3
rt3 =: - RT3

HX =: (2, 0);(1, RT3);(_1, RT3);(_2, 0);(_1, rt3);(1, rt3);(2, 0)

NB. for test HX
NB. gosper0_HX0_button=: 3 : 0
NB. (255, 0, 0) colorpolylines ; (0.5) ensize HX
NB. glshow ''
NB. )

NB.       subroutinized       programs       using       by       local       center
=========================================
NB. make x, y values for hex_local_hexagons subroutine
NB.  e. g.  jj (_6, (_4 * RT3)) for JX5
```

```
jj =: 3 : 0
'X Y' =. y.
gg1 =. (sz * ((X + 3),  (Y + RT3)) )  shift ; sz ensize HX
gg2 =. (sz * ((X + 0),  (Y + 2*RT3))) shift ; sz ensize HX
gg3 =. (sz * ((X + _3), (Y + RT3) ))  shift ; sz ensize HX
gg4 =. (sz * ((X + _3), (Y + rt3) ))  shift ; sz ensize HX
gg5 =. (sz * ((X + 0),  (Y + 2*rt3))) shift ; sz ensize HX
gg6 =. (sz * ((X + 3),  (Y +  rt3)) ) shift ; sz ensize HX
gg0 =. (sz * ((X + 0),  (Y + 0)) )    shift ; sz ensize HX
gg =: (<gg1), (<gg2), (<gg3), (<gg4), (<gg5), (<gg6), (<gg0)  NB. global
)

NB. draw local_hexagons ================================
draw_7hexagon =: 3 : 0
i =. 0
while. i < 7
  do.
    select. i
      case. 3 do. (0 0 255)   colorpolylines , > 3{gg
      case. 6 do. (255 0 0)   colorpolylines , > 6{gg
      fcase.  do. (255 0 255) colorpolylines , > i{gg
    end.
    i =. i + 1
  end.
glshow ''
)

NB. draw honey_comb figure =============================
gosper0_JXX0_button=: 3 : 0
GS0 =: jj 0, 0
draw_7hexagon GS0
)

gosper0_JXX1_button=: 3 : 0
GS1 =: jj 9, rt3
draw_7hexagon GS1
```

```
)

gosper0_JXX2_button=: 3 : 0
GS2 =: jj 6, (4 * RT3)
draw_7hexagon GS2
)

gosper0_JXX3_button=: 3 : 0
GS3 =: jj _3, (5 * RT3)
draw_7hexagon GS3
)

gosper0_JXX4_button=: 3 : 0
GS4 =: jj _9, RT3
draw_7hexagon GS4
)

gosper0_JXX5_button=: 3 : 0
GS5 =:  jj _6, (_4 * RT3)
draw_7hexagon GS5
)

gosper0_JXX6_button=: 3 : 0
GS6 =:  jj 3, (_5 * RT3)
draw_7hexagon GS6
)

NB. draw gosper figures ======================================
gosper0_GSX0_button=: 3 : 0
gs00 =. 0 fig_xy (6, 5)
gs01 =. 0 fig_xy (6, 1)
gs02 =. 0 fig_xy (2, 3)
gs03 =. 0 fig_xy (7, 3)
gs04 =. 0 fig_xy (7, 5)
gs05 =. 0 fig_xy (5, 3)
gs06 =. 0 fig_xy (4, 3)
```

```
(0 0 0) colorpolylines gs00, gs01, gs02, gs03, gs04, gs05, gs06
glshow ''
)

gosper0_GSX1_button=: 3 : 0
gs10 =. 1 fig_xy (6, 5)
gs11 =. 1 fig_xy (6, 1)
gs12 =. 1 fig_xy (7, 1)
gs13 =. 1 fig_xy (5, 1)
gs14 =. 1 fig_xy (5, 3)
gs15 =. 1 fig_xy (2, 3)
gs16 =. 1 fig_xy (2, 1)
(0 0 0) colorpolylines gs10, gs11, gs12, gs13, gs14, gs15, gs16
glshow ''
)

gosper0_GSX2_button=: 3 : 0
gs20 =. 2 fig_xy (4, 3)
gs21 =. 2 fig_xy (5, 3)
gs22 =. 2 fig_xy (6, 3)
gs23 =. 2 fig_xy (7, 3)
gs24 =. 2 fig_xy (2, 3)
gs25 =. 2 fig_xy (1, 3)
gs26 =. 2 fig_xy (1, 5)
gs27 =. 2 fig_xy (6, 5)
(0 0 0) colorpolylines gs20, gs21, gs22, gs23, gs24, gs25, gs26, gs27
glshow ''
)

gosper0_GSX3_button=: 3 : 0
gs30 =. 3 fig_xy (4, 3)
gs31 =. 3 fig_xy (5, 3)
gs32 =. 3 fig_xy (6, 3)
gs33 =. 3 fig_xy (7, 3)
gs34 =. 3 fig_xy (2, 3)
gs35 =. 3 fig_xy (1, 5)
```

```
gs36 =. 3 fig_xy (6, 5)
(0 0 0) colorpolylines gs30, gs31, gs32, gs33, gs34, gs35, gs36
glshow ''
)

gosper0_GSX4_button=: 3 : 0
gs40 =. 4 fig_xy (6, 5)
gs41 =. 4 fig_xy (6, 1)
gs42 =. 4 fig_xy (7, 1)
gs43 =. 4 fig_xy (5, 1)
gs44 =. 4 fig_xy (5, 3)
gs45 =. 4 fig_xy (2, 3)
gs46 =. 4 fig_xy (2, 1)
(0 0 0) colorpolylines gs40, gs41, gs42, gs43, gs44, gs45, gs46
glshow ''
)

gosper0_GSX5_button=: 3 : 0
gs50 =. 5 fig_xy (4, 3)
gs51 =. 5 fig_xy (5, 3)
gs52 =. 5 fig_xy (6, 3)
gs53 =. 5 fig_xy (7, 3)
gs54 =. 5 fig_xy (2, 3)
gs55 =. 5 fig_xy (1, 3)
gs56 =. 5 fig_xy (1, 5)
gs57 =. 5 fig_xy (6, 5)
NB. gs50 =. > 3{ <"(1) (7,2)$ > (4 -1){ GS5
NB. gs51 =. > 3{ <"(1) (7,2)$ > (5 -1){ GS5
NB. gs52 =. > 3{ <"(1) (7,2)$ > (6 -1){ GS5
NB. gs53 =. > 3{ <"(1) (7,2)$ > (7 -1){ GS5
NB. gs54 =. > 3{ <"(1) (7,2)$ > (2 -1){ GS5
NB. gs55 =. > 3{ <"(1) (7,2)$ > (1 -1){ GS5
NB. gs56 =. > 5{ <"(1) (7,2)$ > (1 -1){ GS5
NB. gs57 =. > 5{ <"(1) (7,2)$ > (6 -1){ GS5
(0 0 0) colorpolylines gs50, gs51, gs52, gs53, gs54, gs55, gs56, gs57
glshow ''
```

```
)
gosper0_GSX6_button=: 3 : 0
gs60 =. 6 fig_xy (4, 3)
gs61 =. 6 fig_xy (5, 3)
gs62 =. 6 fig_xy (6, 1)
gs63 =. 6 fig_xy (1, 3)
gs64 =. 6 fig_xy (7, 3)
gs65 =. 6 fig_xy (2, 3)
gs66 =. 6 fig_xy (2, 1)
(0 0 0) colorpolylines gs60, gs61, gs62, gs63, gs64, gs65, gs66
glshow ''
)

gosper0_JXX_button=: 3 : 0
gosper0_JXX0_button ''
gosper0_JXX1_button ''
gosper0_JXX2_button ''
gosper0_JXX3_button ''
gosper0_JXX4_button ''
gosper0_JXX5_button ''
gosper0_JXX6_button ''
)



gosper0_GosFine_button=: 3 : 0
gosper0_GSX5_button ''
gosper0_GSX6_button ''
gosper0_GSX0_button ''
gosper0_GSX4_button ''
gosper0_GSX3_button ''
gosper0_GSX2_button ''
gosper0_GSX1_button ''
)

gosper0_GosBase_button=: 3 : 0
```

```
gb0 =. 5 fig_xy (4,3)
gb1 =. 6 fig_xy (4,3)
gb2 =. 1 fig_xy (4,3)
gb3 =. 0 fig_xy (4,3)
gb4 =. 3 fig_xy (4,3)
gb5 =. 2 fig_xy (4,3)
gb6 =. 2 fig_xy (6,5)
gb7 =. 1 fig_xy (6,5)
(255 0 0) colorpolylines gb0, gb1, gb2, gb3, gb4, gb5, gb6, gb7
glshow ''
)

NB. conversion subroutine ====================================
NB.  from Fig.No.p and its indices [q, r]
NB.  to (x, y) values
NB.  e.g.
NB.   gb0 =. 5 fig_xy (4,3) => _5.5 _4.33013
fig_xy =: 3 : 0
:
p =. x.      NB. Fig.No.
'q r' =. y. NB. index
P =. 'GS',(": p)
xy =. r {"(2) (7 2)$ > (<:q) { (". P)
)
```