

Turtle Graphics

MShimura
JCD02773@nifty.com

2007年8月24日

目次

1	Turtle Graphics ことはじめ	1
2	Turtle 事始め	1
2.1	基本コマンド	2
3	program /noun define	6
3.1	Logo 風	6
3.2	J 風	6
4	いろいろな形	8
4.1	Spiral	8
4.2	color	11
4.3	複雑な形	12
5	高度なテクニック	16

1 Turtle Graphics ことはじめ

タートルグラフィックスと言えば *Logo* が有名であった。*Fraser Jackson* の貢献により、ほぼ同様に J でも使えるようになった。

ガイダンスが J の LAB に入っている。^{*1}

詳細な LAB なので、マニュアルの代用にもなる。Lab を参照しながら turtle を動かしてみよう。この LAB は印刷すれば 20 ページを越え、マニュアルとしても十分利用できる。

Logo のテキストが手元に有ればほとんどそのまま使えるようだ。

最初に *turtle* を読み込む。LAB から動かす場合は *Studio* → *Lab* でダイアログボックスの Turtle Geometry, J Turtle User guide を選ぶ。

^{*1} Graphics ではなく General Interests に入っている。

1.0.1 原著の reference

H. Abelson and A. diSessa, Turtle Geometry: The computer as a medium for exploring mathematics. MIT Press, 1980.

2 Turtle 事始め

(Lab を用いない場合は、) 最初に、

```
'require 'turtle'
```



図 1 part-00

```
show fd 1 rt 72
```

1 進み 72° の右ターンの準備をする

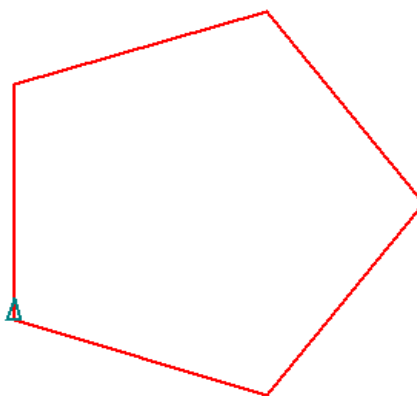


図 2 part-1

```
show repeat 5 fd 1 rt 72
```

1 進み 72° のターンの 5 回繰り返し正 5 角形を描く

2.1 基本コマンド

2.1.1 move, turn

fd forward

bk backwards

rt turns right

lt turns left

これらは次の様に 3 組の数値をセットする。

Turtle の内部定義を少し除いてみよう。

極座標への変換は *show, showf showp* で行っているようだ。

2.1.2

```
ang=[: r. [: - ] * 0.017453292519943295"]_
```

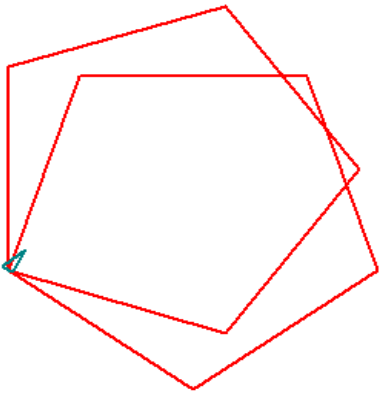


図3 part-2

show repeat 2,(repeat 5 fd 1 rt 72) rt 18
 正5角形を 18° で2回回転させる。

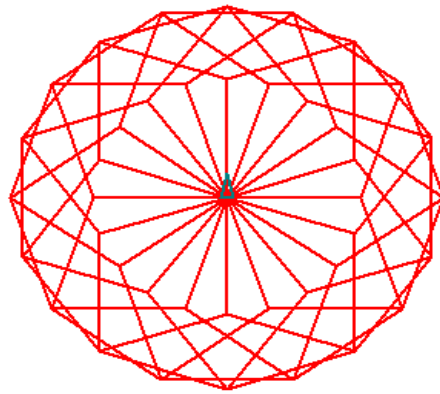


図4 part-3

show repeat 20,(repeat 5 fd 1 rt 72) rt 18
 正5角形を 18° で20回回転させる。

show repeat 8 fd 8 lt 45 fd 24 rt 90
 8進み左へ 45° ターン、24進み右へ 90° ターンを8回
 繰り返す

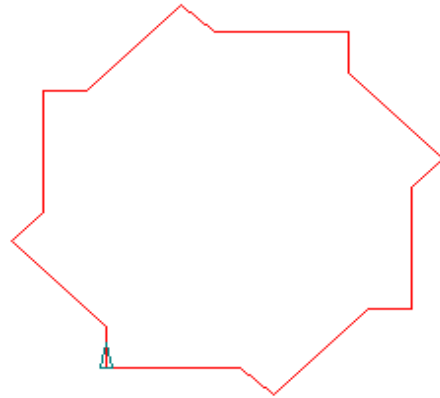


図5

fd,bk

```
fd=:3 : 0
' s' =. {. , y
(s,1 2),1 }. ,y
:
x ,fd y
)
```

```
fd 5 rt 60
5 1 2 0 0.5j_0.866025 1
```

```
fd 3
3 1 2
```

```
bk=:3 : 0
' s' =. {. , y
((-s),1,2),1 }. ,y
:
x ,bk y
)
```

```
bk 5
_5 1 2
```

rt,lt

```
rt=:3 : 0
' deg' =. {. , y
(0,(ang deg),1),1 }. ,y
:
x ,rt y
)
```

```
rt 60
0 0.5j_0.866025 1
```

```
lt=:3 : 0
' deg' =. {. , y
(0,(ang -deg),1),1 }. ,y
:
x ,lt y
)
```

```
lt 30
0 0.866025j0.5 1
```

2.1.3 show

show		標準
showf	fix	中心から始め、上下左右各 100 ステップの固定画面
showp	plot	plot 画面に描く

2.1.4 save/clip

clip	clip” で、クリップボード経由でツールで save する。	clip” NB. pd は不要
save	save 'temp/turtle_29.emf'	emf は可能 save eps はなさそうだ。

3 program /noun define

手順をプログラムする

3.1 Logo 風

LOGO 風には `to` を用いる。

LOGO は次の書式である。

```
to mysquare :side
repeat 4 [fd :side rt 90]
end
```

J の LOGO 風は

to 'mysquare **side**;'repeat 4 fd **side** rt 90'

と'';' で構成する。

```
show mysquare 2 fd 4
```

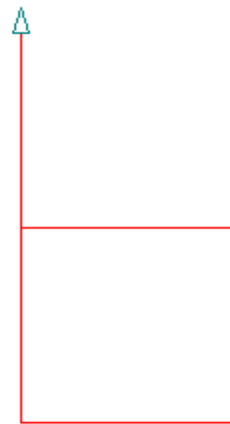


図 6 mysquare

3.2 J 風

noun define が便利である。

```

a=: noun define
rt 30 fd 40
rt 60 fd 40
rt 60 fd 40
NB. rt 120 fd 80
)
show a

```



図 7 showf

```

a=: noun define
rt 30 fd 40
rt 60 fd 40
rt 60 fd 40
rt 120 fd 80
)
show a

```



図 8 showf

repeat
指定回数の copy

```
repeat 3, fd 6 rt 120
```

```

6 1 2 0 _0.5j_0.866025 1
6 1 2 0 _0.5j_0.866025 1
6 1 2 0 _0.5j_0.866025 1

```

4 いろいろな形

4.1 Spiral

ループを簡単に入れ込み、その上に repeat で更に繰り返している。[は始まりの方を用いる。] はエラーが出る。

```
show repeat 30 ;'fd (a =: a+1) rt 120' [ a =: 0  
5 = penta  
show repeat 60 ;'fd (a =: a+1) rt 45' [ a =: 0  
4 = rect  
show repeat 40 ;'fd (x=:x+1) rt 90' ] x=.0
```

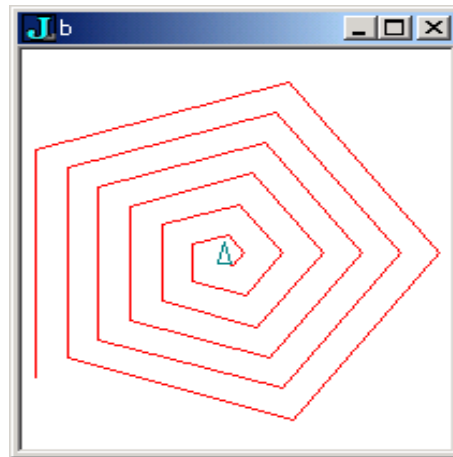


図 9 spiral

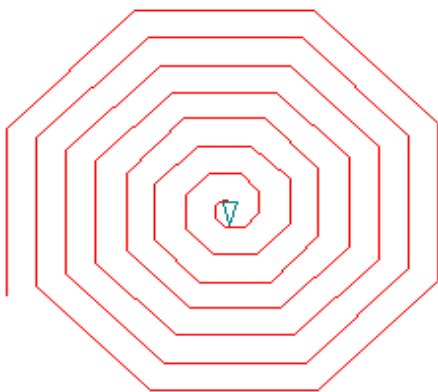


図 10 spiral

4.1.1 for

for による繰り返し

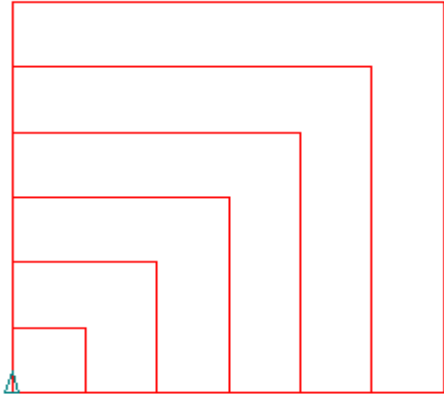


図 11 sq for

sh show sq for 1 2 3 4 5 6 7

```

12 6 $ sq for 1 2 3
1 1 2 0 0j_1 1
1 1 2 0 0j_1 1
1 1 2 0 0j_1 1
1 1 2 0 0j_1 1
2 1 2 0 0j_1 1
2 1 2 0 0j_1 1
2 1 2 0 0j_1 1
2 1 2 0 0j_1 1
3 1 2 0 0j_1 1
3 1 2 0 0j_1 1
3 1 2 0 0j_1 1
3 1 2 0 0j_1 1

```

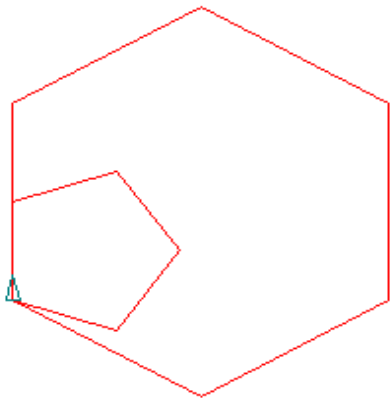


図 12 poly

show poly 1 72 , poly 2 60

poly 2 72, poly 2 60 では同じサイズ

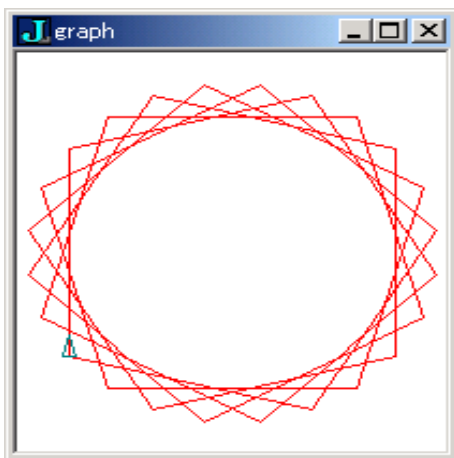


図 13 poly

show poly 1 75

$75^\circ \times 4 = 300^\circ$, 余りり 60°

余りとの最大公約数まで回転する。

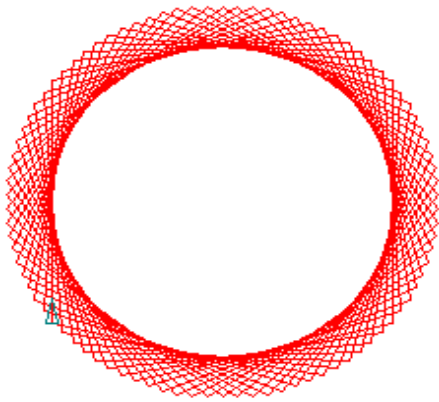


図 14 poly

```
show poly 1 76
show poly 1 138
show fill poly 1 138
NB. fill は塗りつぶし
```

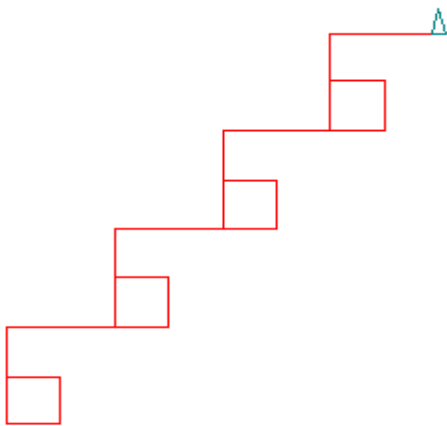


図 15 penup down

```
show repeat 4 penup sq 1 fd 2 rt 90 fd 2 lt 90
```

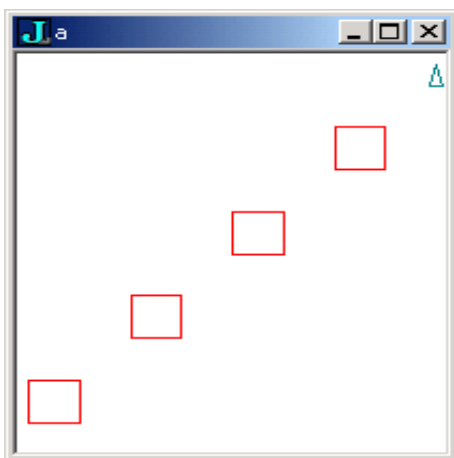


図 16 penup

```
show repeat 4 penup sq 1 pu fd 2 rt 90 fd 2 lt 90
```

4.1.2 いろいろな形

4.2 color

色はJのCOLOR16の指定が利用できる。デフォルトはRED

```
show pencolor GREEN fd 2
```

詳細には rgb で指定する。

```
show pencolor (123 34 12) fd 2
```

fill は塗りつぶし

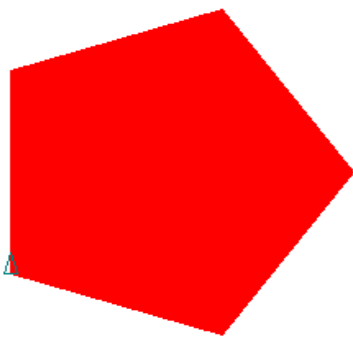


図 17 fill

```
show fill poly 1 72
```

fill 色の塗りつぶし デフォルトは赤

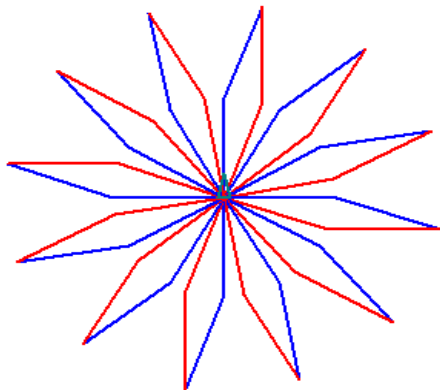


図 18 flower

```
show repeat 12 pencolor BLUE part 2 pencolor RED part
```

```
2 rt 30
```

4.2.1 point goto

point で地点を記憶させられる。

```
a=. rt 45 fd 3 point 1 rt 30 fd 2 point 2 rt 60 fd 4 point 3
```

```
show a
```

```
show a , goto 1
```

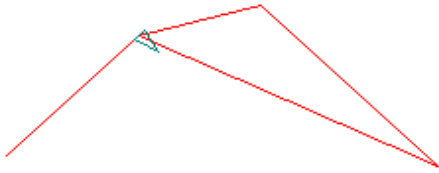


図 19 goto

```
show repeat 5 ;'point (x=. x-1) fd 2 rt 72' [ x=. 6 NB. 5 角形
```

```
show goto 2,goto 4 goto 3 goto 1 goto 5 goto 4
```

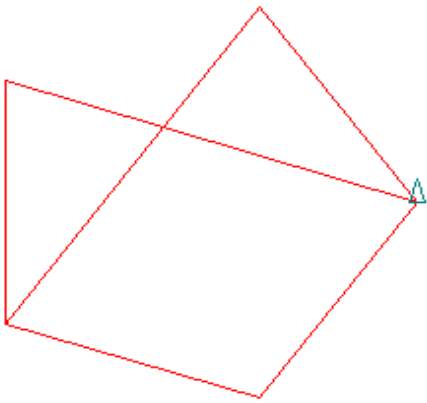
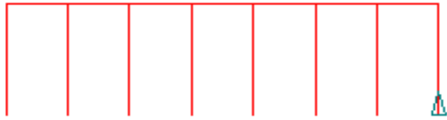


図 20 point

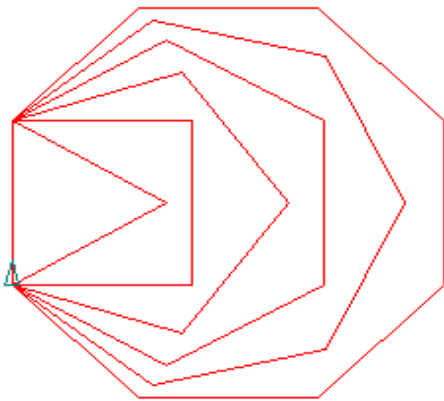
4.3 複雑な形



show repeat 7 , shelter

```
3 6 $ shelter
10 1 2 0 0j_1 1
  5 1 2 0 0j_1 1
10 1 2 0  _1 1
```

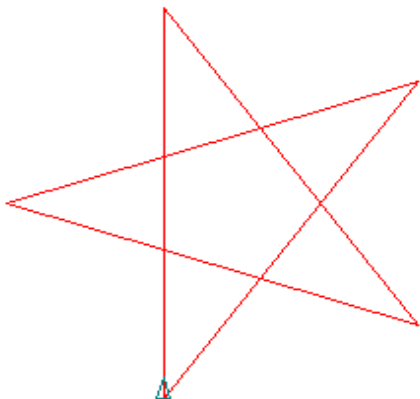
☒ 21 shelter



```
tri  =: repeat 3 fd 1 rt 120  NB. A triangle
squ  =: repeat 4 fd 1 rt 90
pent =: repeat 5 fd 1 rt 72   NB. A pentagon
hex  =: repeat 6 fd 1 rt 60   NB. hexagon
sep  =: repeat 7 fd 1 rt 360%7 NB. septagon
oct  =: repeat 8 fd 1 rt 45   NB. octagon
```

show tri;squ;pent;hex;sep;oct

☒ 22 multi,3-8



show poly 1 144

☒ 23 star



```
show ({1 2 3 4 5;3 2 1 } start rt 18 poly 0.5 144
```

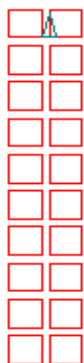
☒ 24 twinkle stars

5 高度なテクニック



```
show ({1 2 3 4 5;1 2 3) start (rt 18 poly 0.5 144)
```

図 25



```
show ({1 2 ;1 2 3 4 5 6 7 8 9 10)  
start sq 0.8
```

図 26 cars

```
show (23 { . , a j./a =. i.10) start sq 0.8
```

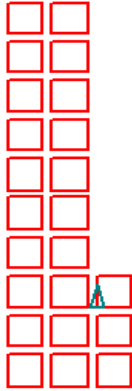


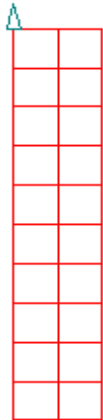
図 27 box

```

j./~ (i.10)
0 0j1 0j2 0j3 0j4 0j5 0j6 0j7 0j8 0j9
1 1j1 1j2 1j3 1j4 1j5 1j6 1j7 1j8 1j9
2 2j1 2j2 2j3 2j4 2j5 2j6 2j7 2j8 2j9
3 3j1 3j2 3j3 3j4 3j5 3j6 3j7 3j8 3j9
4 4j1 4j2 4j3 4j4 4j5 4j6 4j7 4j8 4j9
5 5j1 5j2 5j3 5j4 5j5 5j6 5j7 5j8 5j9
6 6j1 6j2 6j3 6j4 6j5 6j6 6j7 6j8 6j9
7 7j1 7j2 7j3 7j4 7j5 7j6 7j7 7j8 7j9
8 8j1 8j2 8j3 8j4 8j5 8j6 8j7 8j8 8j9
9 9j1 9j2 9j3 9j4 9j5 9j6 9j7 9j8 9j9

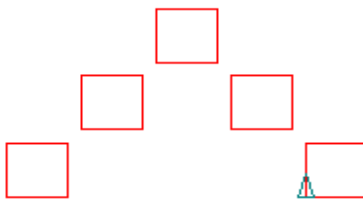
```

ベクトル化して 23 個取り出す。



```
show blocksquares 10 2 1
```

図 28 block



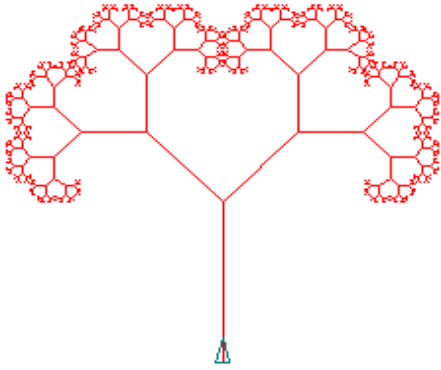
```

show 1j3 2j4 3j5 4j4 5j3 start sq 0.8
      3j5
    2j4      4j4
  1j3              5j3

```

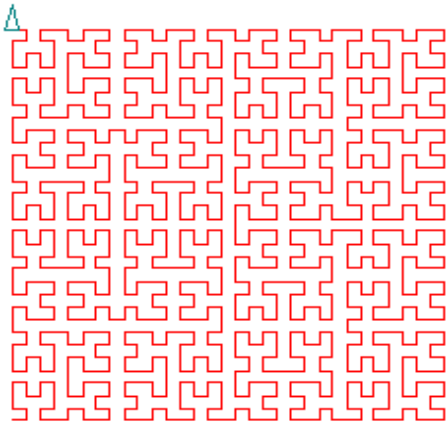
横方向は整数部で、縦方向は虚数部で表す。

図 29 position



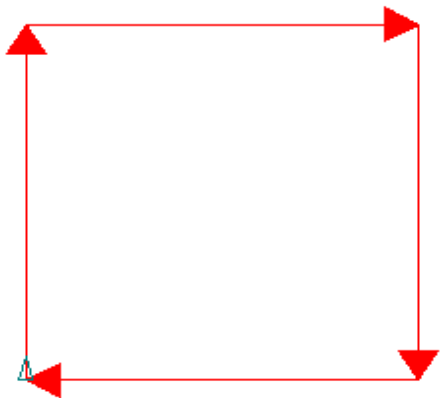
show branch 2 10

図 30 branch



show rhilbert 5 5

図 31 hilbert



(example)

show repeat 4 fd 10 solida 1 rt 90

show repeat 4 fd 10 arrow 1 rt 90

show fd 10 rt 90 arcr 10 45 arrow 1 rt 90 fd 10

図 32 arrow



17show arcl "1 [1 1 2 3 5 8 13 21 ,.90

フィボナッチ数

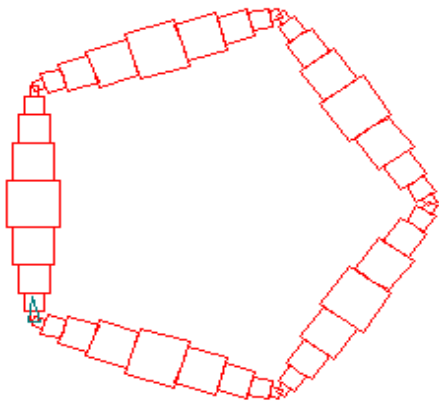


図 34 recursive

```
b =: recursive sq "0] 1 2 3 4 5 4 4 3 2 1
show repeat 5 ,(b) rt 72
72°の5角形で模様(b)を繰り返し描く
```

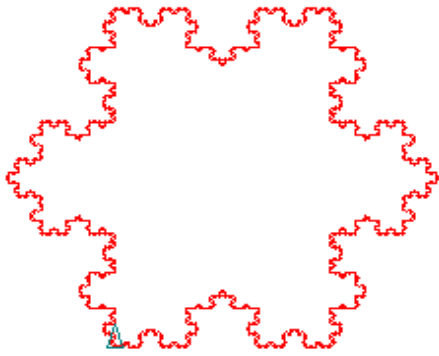


図 35 koch

```
show snowflake 4 5
```

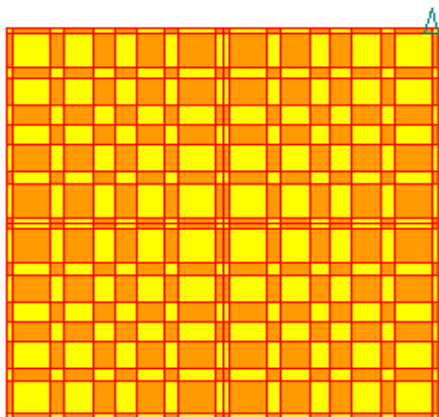
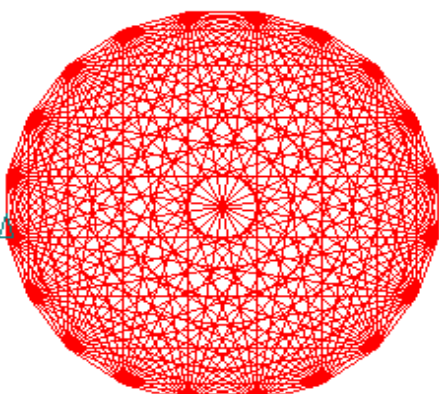


図 36 aya

```
show ( js,(0j30+js),(30j0+js),30j30+js) start <"1
jn0,jn1,jn1,jn0
詳細は LAB の最後を参照
```



```
Fraster Jackson の傑作
show repeat 20 ;' point (n=.n+1) fd 1 rt 18' [ n=. _1
18
to 'goreturn a b'; goto b goto a'
to 'goreturnall a'; goreturn for a,i.20'
show goreturnall for i.20
```