

インドラの真珠 (J6 版) (その 0)

SHIMURA Masato
JCD02773@nifty.ne.jp

2014 年 1 月 17 日

目次

1	同時変換	2
2	複素数で描く	5

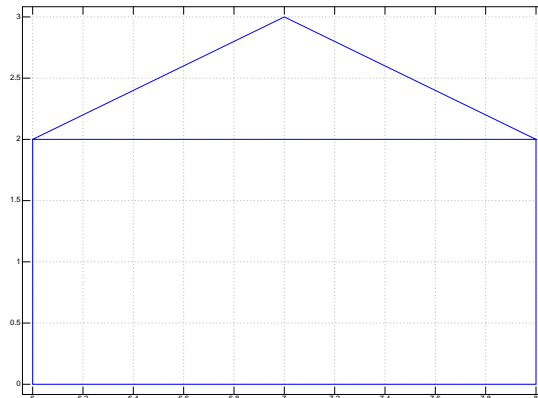
「インドラの真珠-クラインの夢」マンフォード他著 (日本評論社) が 10 年がかりで翻訳された。今回はこれを J で描くための基礎的な準備作業である

オブジェクト (1) - 家を描く

- `require 'plot numeric trig'`
- グラフィック用のツールは北斎の小紋作成のときに作った次の 2 本を互換性を保ちつつ、拡張する
C.Reiter の `dwin2.ijs` をベースにしているが必要な箇所は全て `tool` に移してあるので `addon` は不要。グラフィックスは、J のクラック型で左下が (0,0) である
`hokusai_tool_improve.ijs`
`hokusai_tool_bezier.ijs`
- 家を一筆書きで描き、平面上での拡大 回転 移動の挙動を確認してみよう。

```
H1=: 6 2,8 2,7 3,6 2,6 0,8 0,:8 2
```

```
H1
6 2
8 2
7 3
6 2
6 0
8 0
8 2
```



```
PLOT H1
pd 'eps c:/house0.eps'
```

1 同時変換

C.Reiter のスクリプトをベースにツールに拡張してある。

1.1 拡張/縮小・回転・移動

- 拡大・縮小/Elongation/Scaling

```
3 : '(y,1)* =i.3'
```

```
(x, y, 1)new = (x, y, 1)  $\begin{pmatrix} r & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 
    elongm 1.5 2
    1.5 0 0
    0 2 0
    0 0 1
```

```
PLOT 1.5 2 calc_elongm H1
```

- 回転/rotation

回転角はラジアンで与える。 π で半回転、 2π で1回転である。

```
clean rotm 1r3p1
    0.5 0.866025 0
(x, y, 1)new = (x, y, 1)  $\begin{pmatrix} \cos(t) & \sin(t) & 0 \\ -\sin(t) & \cos(t) & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 
    _0.866025 0.5 0
    0 0 1
```

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

```
transm
```

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

- 移動/transpotation $(x, y, 1)_{new} = (x, y, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & 1 \end{pmatrix}$ `clean transm 2 3`
 $\begin{matrix} 0 & 0 \\ 0 & 1 & 0 \\ 2 & 3 & 1 \end{matrix}$

1. マトリクスの掛け算は内積演算で行う

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

2. 元を揃えるため H1 の右に 1 で揃えた列を加え、計算後落とす。

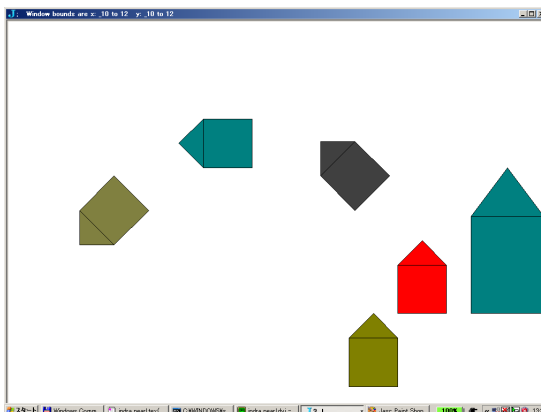
`clean` は計算のゴミ取り

```
calc_elongm=: 4 : 'clean }:"1 (y,.1) mp elongm x'
```

```
calc_rotm=: 4 : 'clean }:"1 (y,.1) mp rotm x'
```

```
calc_transm=: 4 : 'clean }:"1 (y,.1) mp transm x'
```

3. 赤の H1 が変換される様子。



```
_10 _10 12 12 dwin ''
```

```
255 0 0 dpoly H1
```

```
128 128 0 dpoly _2 _3 calc_transm H1
```

```
0 128 128 dpoly 1.5 2 calc_elongm H1
```

```
64 256 64 dpoly 1r2p1 calc_rotm H1
```

```
128 128 64 dpoly 3r4p1 calc_rotm H1
```

```
256 256 64 dpoly 1r2p1 calc_rotm H1
```

4. 同時変換

同時変換は実数ベースではアフィン変換を含んでいるので特にアフィン変換を意識

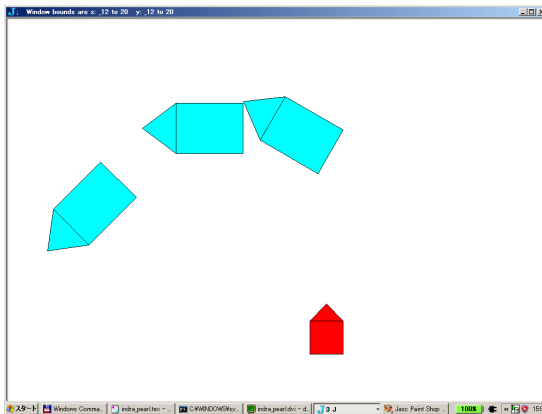
する必要はない。

```
clean (elongm 1.5 2) mp (rotm 3r4p1) mp transm 2 3
_1.06066  1.06066  0
_1.41421  _1.41421  0
          2          3 1
```

```
0 255 255 dpoly }:"1 (H1,.1) mp (elongm 1.5 2) mp (rotm 1r2p1) mp transm 2 3
```

5. 同時変換のためのスクリプトを尽くす。パラメーターは左引数で与える。(r s; π ;a
b)

```
0 255 255 dpoly (1.5 2;1r2p1;2 3) calc_homo H1
```



```
calc_homo=: 4 : 0
```

NB. make Homogeneous matrices

NB. Usage: (1.5 2;1r2p1;2 3) calc_homo H1

NB. x is 3 elements //elong;rot;trans

```
'ep rp tp'=: x
```

```

}:"1 (y,.1) mp (elongm ep) mp (rotm rp) mp transm tp
)

```

2 複素数で描く

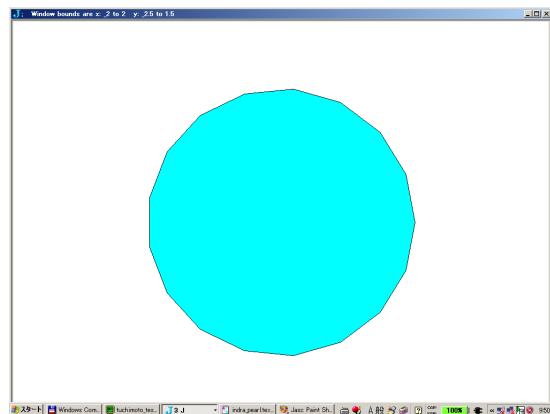
2.1 複素数と J の演算

複素数を用いたガウスの正 17 角形の作図である。

```

_1 _1 1 1 dwin ''
0 255 255 dpoly +. r. 2p1*(i.17)%17

```



1. ガウスの正 17 角形のスクリプト `+. r. (i.17)%17` には J の複素数演算の要素が詰まっている

$$z^{17} - 1 = 0$$

- (a) `2p1` 2π ラジアンで 1 周になる
- (b) `r.` Angle/Polar $e^{i\theta}$
- (c) `+`. Real/Imaginary 実部と虚部に分離する。(x,y) 座標になる。

`plot` はガウス・アルガン座標で複素数も描くが、低レベルの `isigraph` はガウス・アルガン座標をサポートしていないので実部と虚部を分離して x.y 座標に変換する

2. ピタゴレアン

- (a) `3j4` ピタゴラスの 3 角形 3,4,5 の 2 辺 ($x = 4, y = 3$) を複素数で表示する
- (b) 極座標での絶対値を求める。

```
| 4j3
5
```

(c) 極座標上での絶対値と偏角を同時に求める。(*.) length/Angle

```
*. 4j3
5 0.643501
```

(d) 弧度法に変換するには *dfr: degree from radian* 逆は *rfd: radian from degree*。numeric.ijs に入っている。

```
{: dfr *. 4j3
36.8699
```

3. オイラーの公式

(a)

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta)$$

$3 \times e^{\frac{1}{4}\pi}$	<pre>3*^j.1r4p1 2.12132j2.12132 または 3*1x1^j.1r4p1</pre>
-------------------------------	--

$3 \times \cos \frac{1}{4}\pi + i \sin \frac{1}{4}\pi$	$3*(\cos 1r4p1) + j. \sin 1r4p1$ 2.12132j2.12132
$3 \times e^{\frac{1}{4}\pi}$	$3*r. 1r4p1$ 2.12132j2.12132

$r.e^{i\theta}$ を即座に計算する

(b) $r(\cos\theta + i\sin\theta)$ は極座標上で、 $r =$ 絶対値は拡大を、 $\theta =$ 偏角は回転を表す

2.2 幾つかの作図のための複素数の公式

•

$$e^z = e^{x+iy} = e^x e^{iy} = e^x(\cos y + i \sin y)$$

•

$$\begin{cases} e^{i\theta} = \cos\theta + i\sin\theta \\ e^{-i\theta} = \cos\theta - i\sin\theta \end{cases}$$

$$2 \text{ 式を足して} \rightarrow e^{i\theta} + e^{-i\theta} = 2\cos\theta \rightarrow \cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

$$2 \text{ 式を引いて} \rightarrow e^{i\theta} - e^{-i\theta} = 2i\sin\theta \rightarrow \sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

θ を z とすると

$$\cos z = \frac{e^{iz} + e^{-iz}}{2}$$

$$\sin z = \frac{e^{iz} - e^{-iz}}{2i}$$

• 極形式での複素数の積と累乗

$$r_1(\cos\theta_1 + i\sin\theta_1) \cdot r_2(\cos\theta_2 + i\sin\theta_2) = r_1 r_2 (\cos(\theta_1 + \theta_2) + i\sin(\theta_1 + \theta_2))$$

$$z^n = (r(\cos\theta + i\sin\theta))^n = r^n(\cos n\theta + i\sin n\theta)$$

2.3 円と螺旋の作図

- 円 `plot r. steps _1p1 1p1 100`
- アルキメデス螺旋 `plot t* r. t=. steps 0 10p1 1000`
- ベルヌイ螺旋 `plot (^0.1*t) r. t=. steps 0 30 1000`
`r. y --> ^j. y` であり、両項は `x r. y --> x * r. y` となる

2.4 2つの入力形式

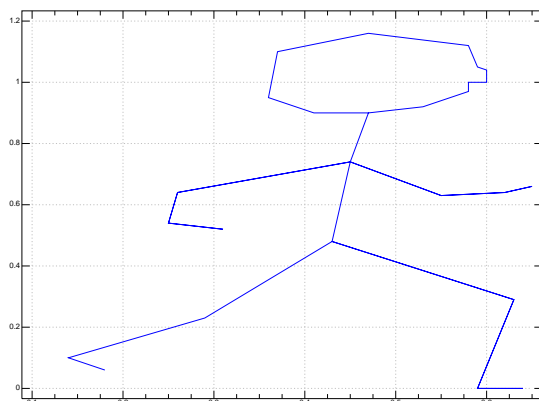
先の家を複素数で入力する。(H1) `plot` はアルガン・ガウス座標にも対応するが、低レベルの `isigraph` は複素入力は受け付けないので (+.) で分離して (x,y) 座標とする。

H1=: 6 2,8 2,7 3,6 2,6 0,8 0,:8 2

HC1=: 6j2 8j2 7j3 6j2 6j0 8j0 8j2 NB. complex number

2.5 オブジェクト(2)-スティックラー博士

1. このシリーズのマスコット・スティックラー博士を作る。大きく座標を取って、 $\frac{1}{10}$ に縮小した。`SI` は (x,y) 座標、`SC1` は複素数のデータである。



2. スティックラー博士を渦巻きに乗せてみよう。渦巻きを粗く取ると各点が拡大と回転の情報を保持しているので、各点を各々スティックラー博士の後ろ足の先端の始点に掛ける。

dline_win は大雑把にキャンバスサイズを自動計算して立ち上げ、ラインを引くツール

```
minmax0=: <./,>./
find_minmax=: 3 : 0
minmax0&minmax0 > minmax0&minmax0 L:0 y
)
```

```
dline_win=: 4 : 0
NB. (bernulle 0 36 180) dline_win SC1
tmp0=. +. L:0 y * L:0 {@> x
tmp1=. find_minmax tmp0
(>. 2 # _2 0 + tmp1) dwin ''
128 128 128 dline L:0 tmp0
)
```

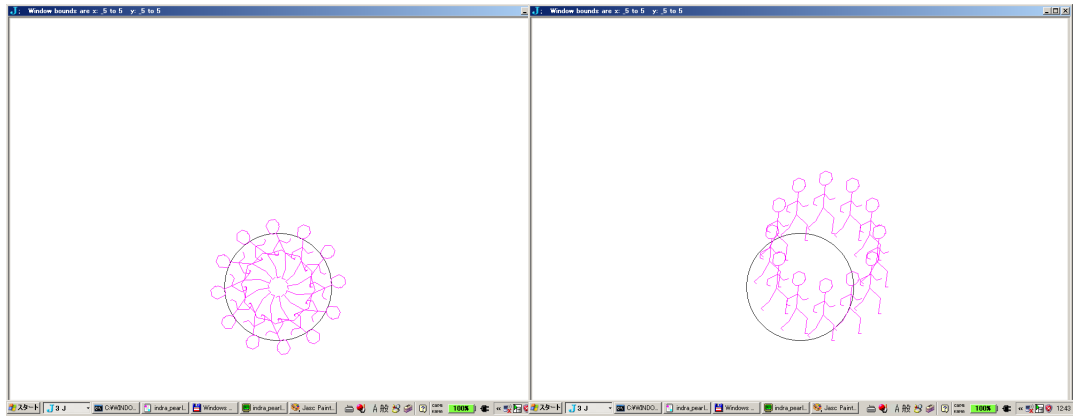
3. 円

- 円を描く。次で円を描く。

```
_5 _5 5 5 dwin ''
dline +. r. steps _1p1 1p1 100 NB. circle
```

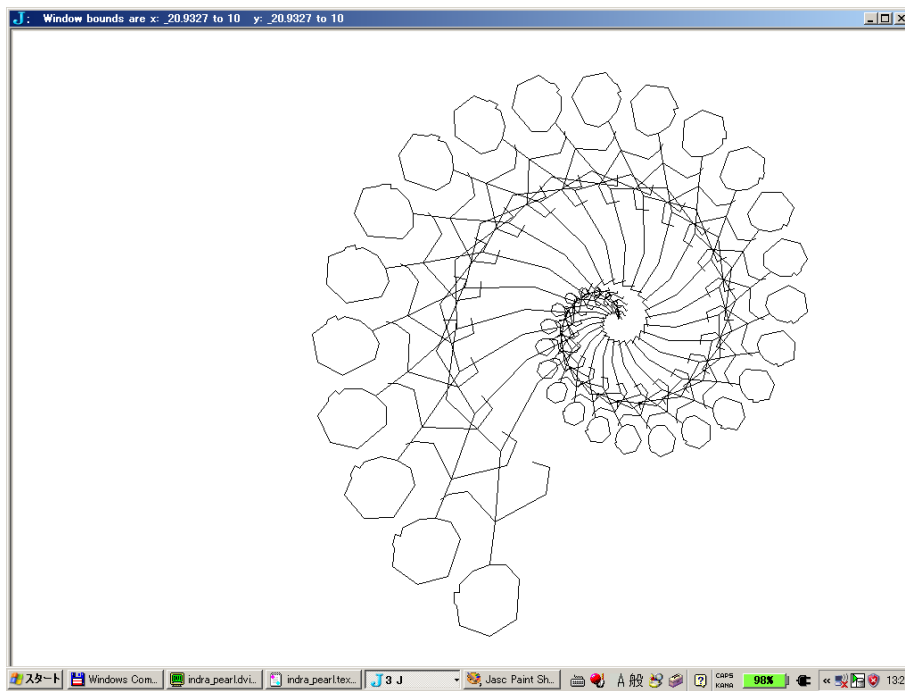
- 円にスティックラー博士を載せる。足す場合は移動のみで回転しない。掛け算では回転する

```
255 0 255 dline L:0 +. L:0 SC1 * L:0 {@> r. steps _1p1 1p1 24
255 0 255 dline L:0 +. L:0 SC1 + L:0 {@> r. steps _1p1 1p1 24
```



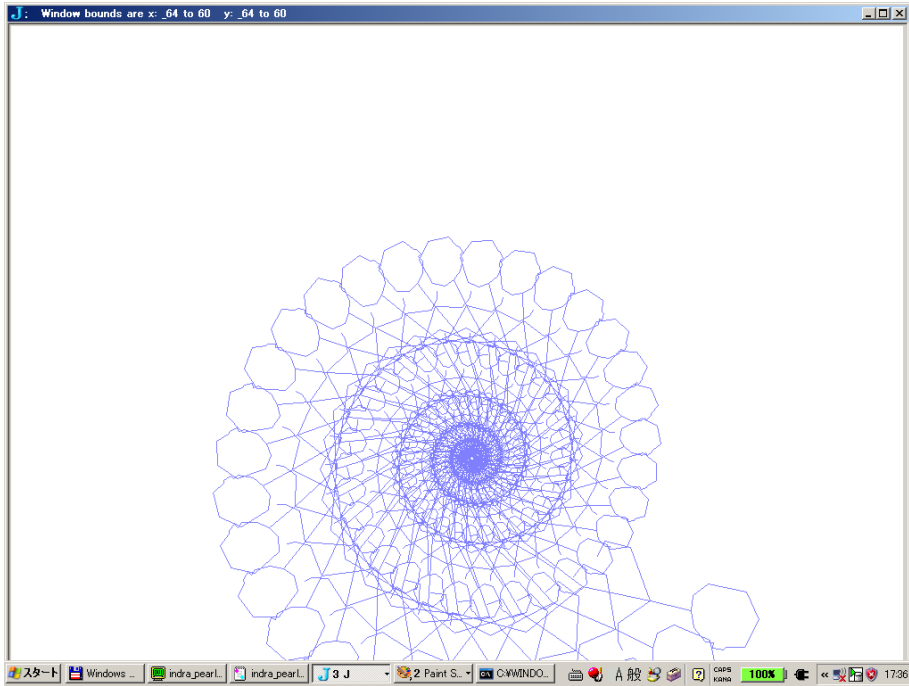
4. アルキメデス螺旋

(archi 0 3p1 36) dline_win SC1



5. ベルヌイ螺旋

(bernulle 0 36 180) dline_win SC1



References