

# クォータニオンで回転とグラフィックスを (1)

SHIMURA Masato  
jcd02773@nifty.ne.jp

2019年6月27日

## 目次

1	plot で描く	1
2	Rotation の計算	2
3	円を回転させる	3
4	球のクォータニオンによる回転	6
5	正多面体のクォータニオンによる回転	11

## はじめに

4次元は絵画ではクレー、ピカソ、ダリが最初と Wikipedia にあった。天才の頭の中は覗けないが、幸い作品は観賞できる。エッシャーは双曲幾何の方なので後回しとしよう。

クォータニオンの学習で、3種類の回転プログラムの Script を作成した。

- rotateQ チキチキバンバン方式での回転
- rotateQ\_mat 乗算マトリクスで行う回転方式
- rotateQ\_rotmat 回転マトリクス方式

クォータニオンの習熟運転を行いながら 3D グラフィックスを幾つか描いてみる。

## 1 plot で描く

J の plot は 3D も複素数もサポートしている優れもので最初のグラフィックステストに最適である。

J の plot は左手系でも右手系でもなく xy が平面で z が上向き。3D ツールの Unity は左手系。学際分野は右手系が圧倒的に多い。

J のグラフィックスはベースが gl2,OpenGL は gles である。plot は簡潔に高度な 3D グ

ラフから複素数まで描けるツールである。

ここでは少し本格的な `pd(plot driver)` を用いる方法のテストも行う。

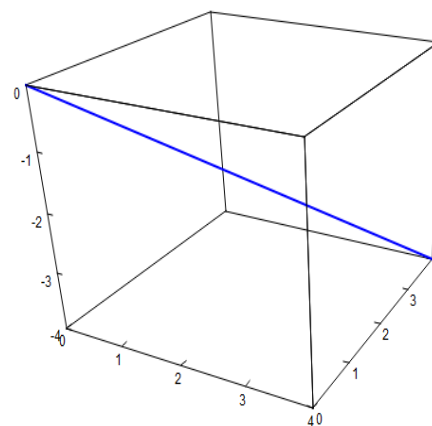
## 1.1 plot の補強

2つの画像を同時に描いたり、**Viewpoint** を変えたりするのに **pd(plotdriver)** を使ってスクリプトを作成する。版面も良いが肉筆浮世絵の一点物も描くことができる。

色々触った後に戻すための `plot` の 3D コマンドの初期値を一通り書き出しておく。`plot_view_3d` を近くにコピーしておいて色々パラメータを変えてみるとよい。

```
plot_view_sub=: 1 : 0
pd 'boxed 0'          NB. --> 1 枠なし
pd 'viewup 0 0 1'
pd 'viewcenter 0 0 0' NB.--> 0 1 0
pd 'viewsize 1 1 0.5' NB.-->1 1 1
pd 'viewpoint 1.6 _2.4 1.5'
)
```

```
plot_Q=: 3 : 0
NB. u 1 4 4 _4
if. 4= # y do. tmp=. }. y
  else. tmp=. y end.
a0=.{0 0 0 ,. tmp
pd 'reset'
". plot_view_3d
pd 'pensize 2'
pd a0
pd 'show'
)
```



## 2 Rotation の計算

### 1. rotation Example

EX.: rotate  $\mathbf{q}$  :  $45^\circ$  about z axis

$$\mathbf{p} = [0, 2i]$$

### 2. 回転クォータニオンの作成

```
qp=. (45;0 0 1) compose_qp 2 0 0
```

### 3. 回転の計算

```
rotateQ qp  
0 1.41421 1.41421 0
```

```
rotateQ_mat qp  
0 1.41421 1.41421 0
```

```
rotateQ_rotmat qp  
0 1.41421 1.41421 0
```

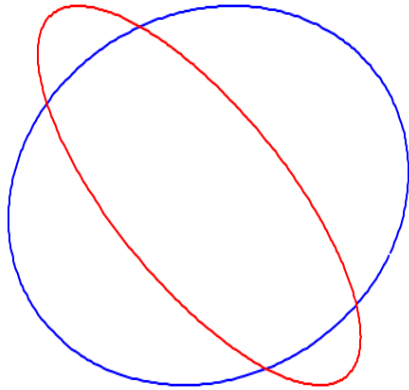
$p' = qp = [0, \sqrt{2}i + \sqrt{2}j]$   
3方式とも安定して計算できている。

## 3 円を回転させる

円をクォータニオンで回転させる

EX1.  $45^\circ = 1r4p1$

1. 回転クォータニオン  
 $\frac{\pi}{3} = 60^\circ$   
axis= 1 1 0      y z 軸で回転
2. p=円のポイント (360 角形 = 360 ポイント)  
point: r. 2p1\*(i.360)%360
3. r.  $e^{i\theta}$  で円の複素座標を求める



4. ラディアンで入力。Jでは  $\pi = 1p1$  は数として扱う。

```
rfd 60
```

```
1.0472
```

```
1r3p1
```

```
1.0472
```

5. 回転クォータニオンの作成

```
mk_q_rad 1r3p1;0 1 1
```

```
0.866025 0 0.353553 0.353553
```

6. 360 ポイント作成

複素数を 2 の実数に分解してから組み上げている。(双複素数=複素数のままだ後に検討)

```
3{. (1r3p1;0 1 1 ) compose_qp_circ { +. circ ''
```

```
+-----+-----+-----+-----+
|0.866025 0|0.866025      0|0.866025      0|
|      0 1|      0 0.999848|      0 0.999391|
|0.353553 0|0.353553 0.0174524|0.353553 0.0348995|
|0.353553 0|0.353553      0|0.353553      0|
+-----+-----+-----+-----+
      q      p      q      p
```

7. 計算

$\omega$  は 0 になる

```
5{. >rotateQ_rotmat L:0 a2
```

```
0      0.5 0.612372 _0.612372
```

```

0 0.489236 0.625368 _0.607916
0 0.478324 0.638174 _0.603275
0 0.467266 0.650785 _0.598449
0 0.456065 0.663198 _0.593442
omega   i       j       k

```

## 8. Script

```

plot_2qp=: 3 : 0
NB. usage: plot_2qp 1r2p1;1 1 1
add_zero=: 3 : ' 0 , L:0 y , (L:0) 0'
compose_qp_circ=: 4 : 0
q0=: mk_q_rad x
p0=: add_zero y
q0 ,. L:0 p0
)

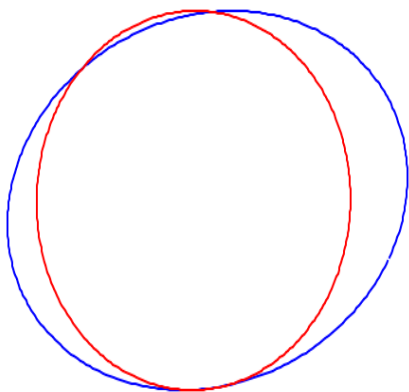
plot_2qp=: 3 : 0
NB. usage: plot_2qp 1r2p1;1 1 1
a1=: { +. circ ''
a2=: y compose_qp_circ a1
a2=: rotateQ_rotmat L:0 a2
pd 'reset'
". plot_view_3d
pd 'pensize 2'
pd {|: (> a1),.0
pd {|: }."1 > a2
pd 'show'
)

```

### EX.2 $90^\circ = 1r2p1$

同じ円を  $90^\circ$  xyz 軸で回転

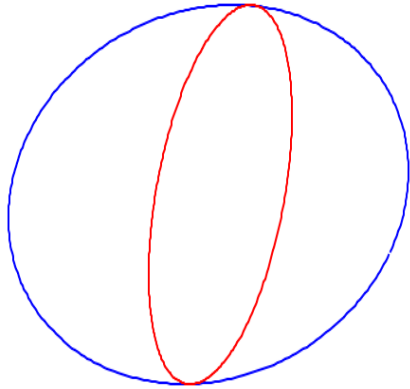
```
plot_2qp 1r2p1; 1 1 1
```



### EX.3 $180^\circ = 1p1$

同じ円を  $180^\circ$  xyz 軸で回転

3D は viewpoint を少し変えても図は変わる。多くを見て感覚をつかもう。



## 4 球のクォータニオンによる回転

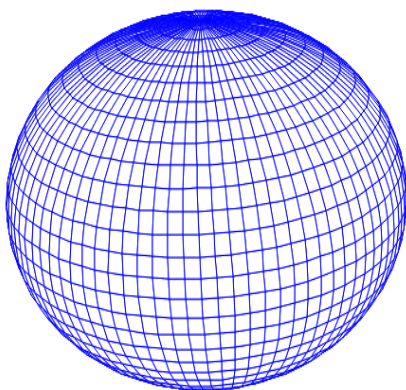
### 4.1 3D をパラメータで描く

3D の球面をオードソックスな方法で描く。

パラメータ (媒介変数) を用いた 3D である。スクリプトは plot と媒介変数作成部に分けた。

$$\begin{cases} x = \cos\theta\cos\phi \\ y = \cos\theta\sin\phi \\ z = \sin\theta \end{cases}$$

```
plot_param_3d@mk_param_phase ''
```



```
plot_param_3d=: 3 : 0
tmp=. y
pd 'reset'
pd 'type wire'
pd ' color blue'
". plot_view_3d
pd tmp
pd 'show'
)
```

```
mk_param_phase=: 3 : 0
NB. usage: mk_param_phase ''
s=. steps _1p1 1p1 90
t=. steps _1r2p1 1r2p1 30
a0=. (cos s) */ cos t
a1=. (sin s) */ cos t
a2=. ($ a1) $ sin t
ax=. a0 ; a1 ; a2
)
```

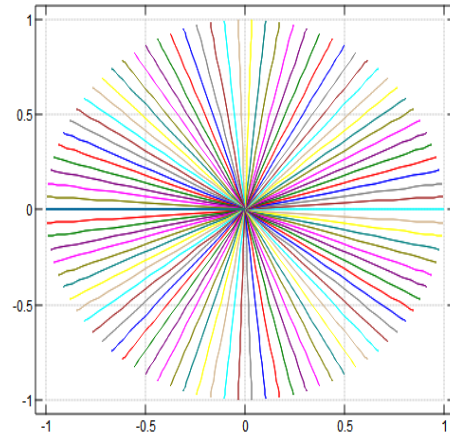
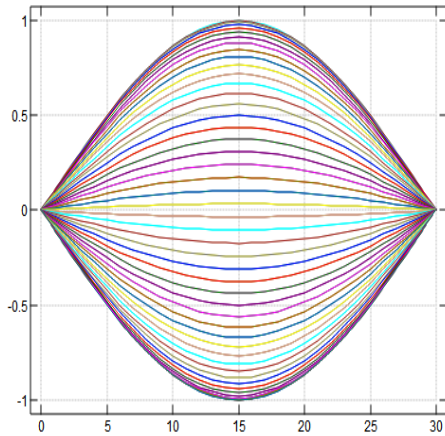
s は -1p1 1p1 でも 0 2p1 でもよいが、t は -1r2p1 1r2p1 でないと半球になる。0- 1p1 は上半身。s も 0- 1p1 では半球 (東経のみ)

z 軸の  $\sin\theta$  を  $\cos\theta$  に変えると、赤道で切った南極が北極の上で繋がる。

x 軸 (a0) を抜き出した。多分経度だろう。

```
plot a0;a1
```

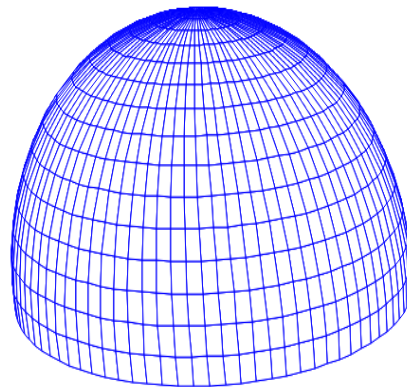
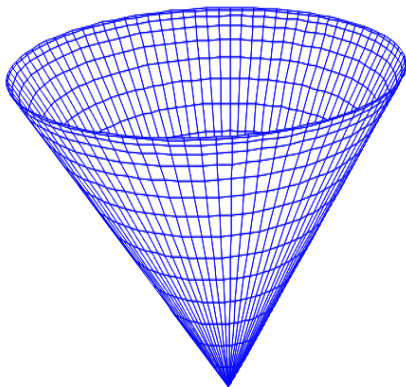
xy の合成。竹ひご編みので横ひごは z で指定。sint を用いる。sins だと形が変わる



```
a3 = cost
```

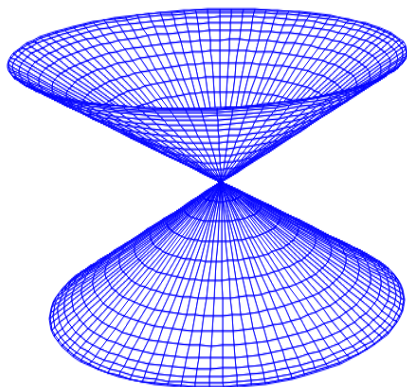
```
t=. steps 0 1p1 30
```

```
a2=: ($ a1) $ sin t
```





```
t=. steps 0 1p1 30
a2=: ($ a1) $ cos t
```



#### 4.2 クォータニオンで回転

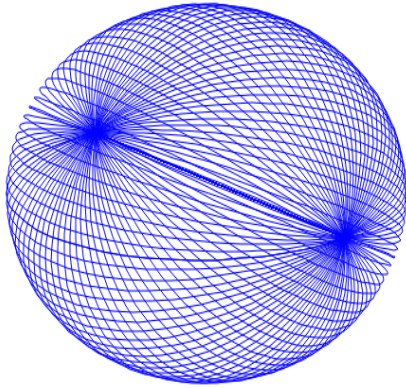
回転クォータニオン 回転角:1r2p1 回転軸:1 1 1 で、パラメータで描いた球を回す。

##### 1. オペレーション

```
a=. calc_qp_phase 1r2p1;1 1 1
'boxed 0' plot { |:}."1 >a
```

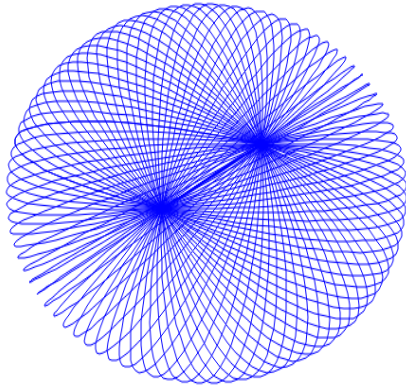
##### 2. 図

3人の虚数海女は元気一杯。



### 3. Script

```
 lineup_p=: 3 : '{ |:> , L:0 y'  
 compose_qp_phase=: 3 : 0  
 NB. u (1r2p1; 1 1 1 )  
 px=. 0 , L:0 lineup_p@mk_param_phase2 ''  
 qx=. mk_q_rad y  
 qx ,. L:0 px  
 )  
  
 calc_qp_phase=: 3 : 0  
 tmp=. compose_qp_phase y  
 rotateQ_rotmat L:0 tmp  
 )
```



```
a=. calc_qp_phase 1r4p1;1 1 1
'boxed 0' plot { |:}."1 >a
```

## 5 正多面体のクォータニオンによる回転

クォータニオンによる正多面体の回転に取り組む。多量なので別稿としたが正6面体はここでも紹介しておこう。

円に内接する一辺2の正方形から正6面体をくみ上げる。

1. 正6面体を作る。実際に正6面体の座標を構成する。

- 中心を (0,0) とするダイヤモンド型の正方形の座標を求める。一辺は2となる。
  - 頂点座標 8 点

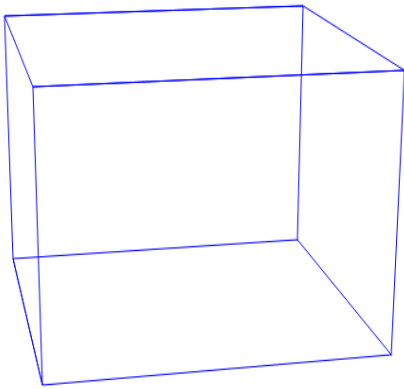
clean	+. r. 2p1* (i.4)%4	NB. hedron 4	1 0 0
			0 1 0
1 0			-1 0 0
0 1			0 -1 0
-1 0			<hr style="width: 100%; border: 0.5px solid black;"/>
0 -1			1 0 1
			0 1 1
			-1 0 1
			0 -1 1

- 12の辺を line で一筆書きで描く。3辺は重複するが分け書きよりスクリプトがシンプルになる。

## 2. Script

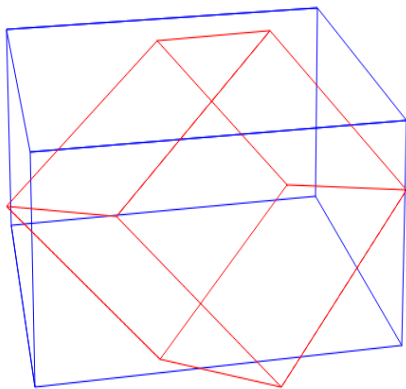
```
mk_6hedron=: 3 : 0
tmp0=: 1 0 0;0 1 0 ;_1 0 0;0 _1 0;1 0 1;0 1 1 ;_1 0 1;0 _1 1
hitohude=. 0 1 2 3 0 4 5 6 7 4 7 3 2 6 5 1
>hitohude { tmp0
)
```

## 3. plot plot\_6hedron ''



## 4. クォータニオンによる回転

$1r2p1 = \frac{1}{2}\pi$  ラディアン指定  
 XYZ 軸で回転  
 plotQ\_6hedron 1r2p1;1 1 1



## 5. Script

```

compose_qp_6hedron=: 3 : 0
NB. usage: compose_qp_hedron 1r4p1;1 1 0
q0=: mk_q_rad y
p0=: { 0, . mk_6hedron ''
q0 ,. L:0 p0
)
  
```

```

plotQ_6hedron=: 3 : 0
NB. Usage: plotQ_6hedron 1r12p1;1 1 1
tmp0=. { |: mk_6hedron ''
tmp1=. rotateQ_rotmat L:0 compose_qp_6hedron
pd 'reset'
". plot_view_sub
pd tmp0
pd { |: } ."1 > tmp1
pd 'show'
)
  
```