

正多面体とクォータニオンによる回転

SHIMURA Masato
jcd02773@nifty.ne.jp

2019 年 10 月 17 日

目次

1	正多面体	1
2	6 面体を作る	3
3	正三角形の多面体	5
4	12 面体	10

はじめに

クォータニオンと回転行列を作成し、腕試しその 1 として、パラメータで作成した球の 3 次元座標をクォータニオンで回転させた。

腕試しその 2 として、プラトンに由来する正多面体の 3D 座標を順にラインで描き、回転させてみる。

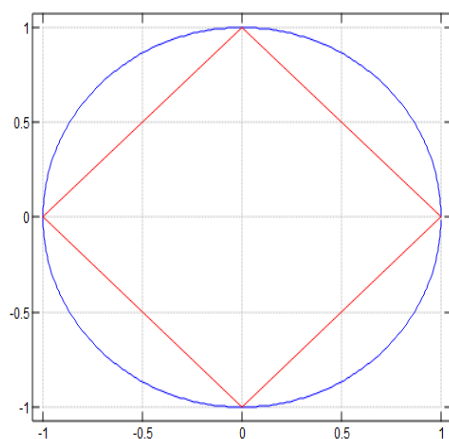
1 正多面体

正多面体の座標は、 J の多角形と円 (360 角形) の頂点座標を求める Script から始める。半径 1 の円に内接し、多角形の中心は 0 となる。

```
hedron=: 3 : 'clean +. r. 2p1 * (i.y)%y '
```

r は $e^{i\theta}$ を計算するものでオイラーの美しい公式 $e^{i\theta} = \cos\theta + i\sin\theta$ から \cos, \sin での多角形の x, y の座標を求めている。

plot_hedron 4



*1

```
plot_hedron=: 3 : 0
a0=. hedron 360
a1=. ({: a1), a1=. hedron y
pd 'reset'
". plot_view_sub
pd a0
pd a1
pd 'show'
)
```

最初の作業

1. ファイルのロード

```
require 'numeric trig plot png'
require '~temp/math/linear_math/vec_quaternion.ijs'
require '~temp/graphics/geometry/quaternion_graphics0.ijs'
```

クオータニオンのファイルのパスは各々の環境で。なかなか一気にには通らないので、一枚ずつロードしてもよい。

2. plot の 3D コマンド

3D は見る角度により図形が大きく変化するので、plot の 3D コマンドを抜き書きして独立させた。

plot の近くにコピーしてお気に入りのパラメーターを探してほしい。回転にはオイラー角のようだ。(NB. はデフォルト値)

```
plot_view_sub=: 1 : 0
pd 'boxed 0' NB. 1
pd 'viewup 0 0 1' NB. 0 0 1 // hanging easy
pd 'viewcenter 0 0 1' NB. 0 0 0
pd 'viewsize 1 1 1' NB. 1 1 0.5
pd 'viewpoint 1.6 _2.4 1.5' NB. 1.6 _2.4 1.5
```

*1 正 12 面体と正 20 面体は黄金比に溢れた座標なので、頂点座標は他から拝借した

)

2 6面体を作る

円に内接する一辺2の正方形から正6面体にくみ上げる。

1. 正6面体を作る。実際に正6面体の座標を構成する。

- 正方形の座標を求める。一辺は2となる。
 - 頂点座標 8点

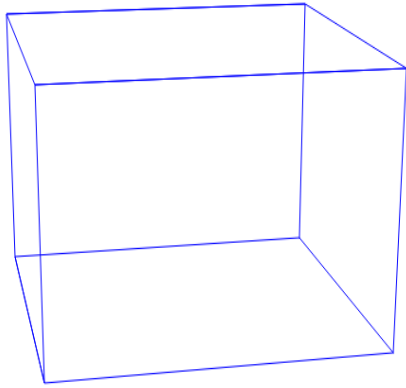
```
clean +. r. 2p1* (i.4)%4 NB. hedron 4
1 0
0 1
_1 0
0 _1
1 0 0
0 1 0
-1 0 0
0 -1 0
-----
1 0 1
0 1 1
-1 0 1
0 -1 1
```

- 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

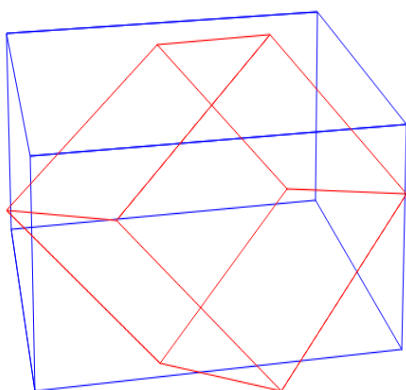
```
mk_6hedron=: 3 : 0
tmp0=: 1 0 0;0 1 0 ;_1 0 0;0 _1 0;1 0 1;0 1 1 ;_1 0 1;0 _1 1
hitohude=. 0 1 2 3 0 4 5 6 7 4 7 3 2 6 5 1
>hitohude { tmp0
)
```

3. plot plot_6hedron ''



4. クォータニオンによる回転

$1r12p1 = \frac{1}{2}\pi$ ラディアン指定
 XYZ 軸で回転
 plotQ_6hedron 1r2p1;1 1 1



正三角形の垂線はピタゴレアンともなる。

#: 0.75

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'
)
  
```

```
0.866025
```

```
sin 1r3p1
```

```
0.866025
```

$$\frac{\sqrt{3}}{2} = 30^\circ$$

上下で 60° となる。

3 正三角形の多面体

3.1 正8面体

内接円から接する菱形を描くと正方形でなく上が少し尖った形になる。ここで斜辺 $\frac{\sqrt{3}}{2}$ で ϕ が内接円の半径 $\frac{1}{2}r = 0.5$ となる ϕ を求める。
手動 AI で 109° を求めると ϕ が得られる。60度ではなく約55度

```
(-: %:3) * sin rfd 35.2644
```

```
0.5
```

```
rfd 35.2644
```

```
0.61548
```

```
180- +: 35.2644
```

```
109.471
```

```
-: 180- +: 35.2644
```

```
54.7356
```

```
mk_8hedron=: 3 : 0
```

```
a0=. (%: 3)* sin 1r6p1
```

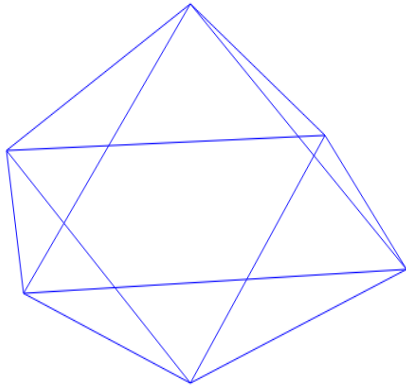
```
tmp0=: 1 0 0;0 1 0 ;_1 0 0;0 _1 0;( 0 0, a0);0 0 ,- a0
```

```
index=. 0 1 2 3 4 1 5 3 0 4 2 5 0
```

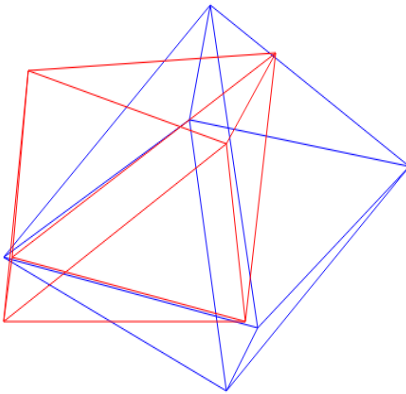
```
>index { tmp0
```

```
)
```

```
plot_8hedron ''
```



`plotQ_8hedron 1r3p1 ; 0 1 1`



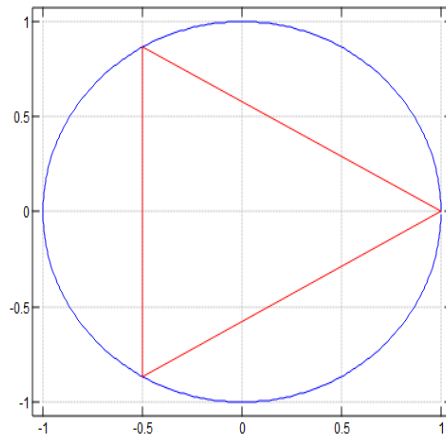
3.2 正4面体

`plot_hedron 3`

```

+. r. 2p1* (i.3) % 3
      1      0
_0.5 0.866025
_0.5 _0.866025
NB.cos=x sin=y

```



頂点座標即ち正三角形の重心は中学数学の花形だが、その上の頂点座標に集う3辺と下敷きの正三角形のなす角度を求めることは稀のようだ。

ここでは頂点座標をJの多角形のイディオムとした。円に内接する、すなわち360角形を描くもの。三角形を描くのに同じスクリプトを用いると正4面体の頂点のxy座標は(0,0)となる。

1. 正三角形のイディオム

図から平面の三角形の一辺は $\sqrt{3}$ 中心は $(0, 0, \sqrt{2})$

$\sqrt{2}$ は白銀比ともいわれる。

正四面体の高さの公式というものがあった。一辺を a とすると

$$h = \frac{\sqrt{6}}{3} a$$

$$\frac{\sqrt{6}}{3} = \sqrt{\frac{2}{3}}$$

(%:2r3)

0.816497

3%~ %:6

0.816497

%: -/ *: (%:3), -: %: 3

1.5

1.5 * L:0 (cos;sin) rfd 70.52876

+-----+

```
|0.5|1.41421|  
+---+-----+
```

底辺の正三角形と重心上で合わさった頂点の角度は手動 AI で求めると 70.52876° となる。

扱いにくい \tan で検証する。末尾の差は円関数の定義誤差か？

```
(%:2)%0.5  
2.82843  
tan rfd 70.52876  
2.82842
```

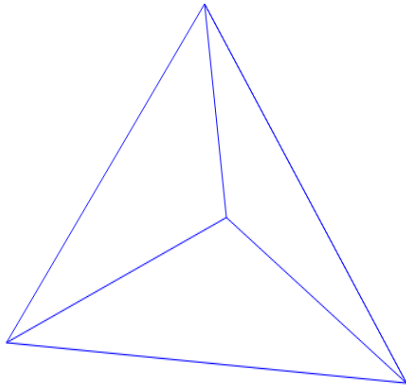
工作だと正三角形を精密に合わさなければならないが、CG ではここに $\sin 60^\circ$ の値を与えても何食わぬ顔をして描いてしまう。

2. Script

```
mk_4hedron=: 3 : 0  
a0=. %: 2 NB. center of gravity is (0 0 ,%:2)  
tmp0=. (_0.5 ,(-a0),0);1 0 0;(_0.5 , a0,0); 0 0 ,a0  
ind=. >0 1 3 0 2 1 3 2{tmp0  
)
```

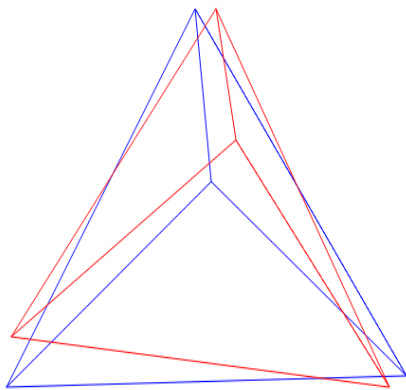
3. plot

```
plot_4hedron "
```

4. クォータニオンによる回転

`plotQ_4hedron 1r4p1;1 1 0`



3.3 正 20 角形

正 4 面体のベースは正三角形、正 8 面体のベースは正 4 角形で、正 20 面体は正 5 角形がベースとなる。

$c = \frac{\sqrt{5}-1}{2}$ としたときの 12 の頂点座標
 黄金比が²ふんだんに入っている。

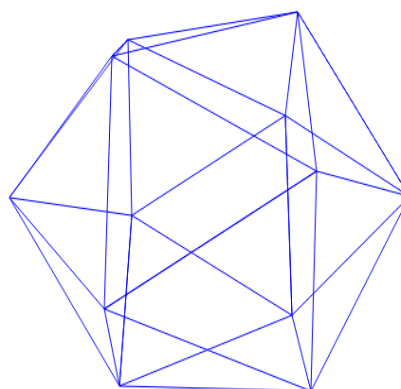
```
-: <: %:5
```

```
0.618034
```

明快な座標は次から得て順序を組み替えた。

https://www.simplex-soft.com/html_transform/3D_data_graph.htm

0	0	c	1
1	-1	0	c
2	-c	-1	0
3	c	-1	0
4	1	0	c
5	0	-c	1
6	-c	1	0
7	-1	0	-c
8	0	-c	-1
9	1	0	-c
10	c	1	0
11	0	c	-1



```
mk_20hedron_foreign2=: 3 : 0
```

```
a0=: -: _1+ %:5
```

```
tmp0=. (0,(a0),1);(-1,0,a0);((-a0),_1,0);(a0,_1,0);(1,0,a0);(0,(-a0),1)
```

```
tmp1=.((-a0),1,0);(-1,0,-a0);(0,(-a0),_1);(1,0,(-a0));((a0),1,0);(0,a0,_1)
```

```
index=. 0 10 4 0 6 1 0 5 1 2 7 1 6 10 11 6 7 11 7 8 11 9 8 9 10 4 9 3 4 5 3 5 2 3 8
```

```
>index { tmp0,tmp1
```

```
)
```

4 12 面体

正に黄金の12面体である。

1. 黄金比は五角形に宿る

$$\frac{1 + \sqrt{5}}{2} = \frac{\sqrt{5} - 1}{2}$$

```
-: >: %:5
```

```
1.61803
```

```
% -: >: %:5
```

0.618034

-: <: %:5

0.618034

2. 72°, 144° から黄金比が現れる

dfr 2r5p1 4r5p1

72 144

3. 計算

+: cos 2r5p1 4r5p1

0.618034 _1.61803

座標データはほりたゆみさんの投稿が出てきたので、図から数値を読み取り、ナンバーを付けて二筆で描いた。黄金分割を用いた数値座標は幾つかのパターンがあるがこのケースは一筆で追えて正確に図が復元できた。

<https://twitter.com/hyrodium/status/369819992658698240>

No	x	y	z	No	x	y	z
0	1	1	1	10	0	$-\frac{1}{\phi}$	$-\phi$
1	0	$\frac{1}{\phi}$	ϕ	11	-1	-1	1
2	0	$-\frac{1}{\phi}$	ϕ	12	$-\phi$	0	$\frac{1}{\phi}$
3	1	-1	1	13	-1	1	1
4	ϕ	0	$\frac{1}{\phi}$	14	$-\frac{1}{\phi}$	ϕ	0
5	ϕ	0	$-\frac{1}{\phi}$	15	$\frac{1}{\phi}$	ϕ	0
6	1	-1	-1	16	1	1	-1
7	$\frac{1}{\phi}$	$-\phi$	0	17	0	$\frac{1}{\phi}$	$-\phi$
8	$-\frac{1}{\phi}$	$-\phi$	0	18	-1	1	-1
9	-1	-1	-1	19	$-\phi$	0	$-\frac{1}{\phi}$

mk_12hedron=: 3 : 0

rphi=: % phi=: -: >: %: 5 NB. golden division

tmp0 =. (1 1 1) ;(0,rphi,phi);(0,(- rphi),phi);(1 _1 1);(phi,0,rphi) NB. penta0

tmp0 =. tmp0,(phi,0,- rphi);(1 _1 _1);(rphi,(- phi),0);((- rphi),(- phi),0)

tmp0 =.tmp0,(_1 _1 _1);(0,(- rphi),- phi);(_1 _1 1);((- phi),0,rphi);(_1 1 1)

NB. 14->

tmp1 =. ((- rphi),phi,0);(rphi,phi,0);(1 1 _1);(0,rphi,- phi)

```

tmp1=. tmp1,(_1 1 _1);((- phi),0,- rphi)
tmp2=: tmp0,tmp1
ind0=. 4 0 1 2 3 4 5 6 10 9 8 7 6 7 3 2 11 8 11 12 13 1
NB. back start is 13
ind1=. 13 14 15 0 15 16 5 16 17 10 17 18 14 18 19 12 19 9
tmp10=:<> ind0{tmp2
tmp11=:<> ind1{tmp2
tmp10,tmp11
)

```

