

# 関数のアクアリウム

SHIMURA Masato  
jcd02773@nifty.ne.jp

2014年4月18日

## 目次

1	方程式とパラメーター表示	1
2	極座標、極方程式	7
3	陰関数のグラフ	10

## はじめに

水族館の各水槽の下に魚の解説がしてあるが数式はこの解説のような一面がある。実際に魚が泳ぐところを見たいし、生態や飼育方法も知りたい。このような思いから関数のアクアリウムと題してJの豊富なグラフィック機能を用いて、関数を描いてきた。

## 1 方程式とパラメーター表示

### 1.1 円

円の公式

$$x^2 + y^2 = r^2$$

これを  $y = \sqrt{r^2 - x^2}$  と変形してもそのままグラフに描くことは難しい。  
パラメーター表示 円:  $x^2 + y^2 = r^2$  のパラメーター表示

$$\begin{cases} x = r \cos \theta \\ y = r \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$

これはオイラーの公式を用いて描くことができる

オイラーの公式 .

一番美しい数学の公式と言われるオイラーの公式  $e^{i\theta} = \cos\theta + i\sin\theta$  の幾何学の顔は複素数の極座標である。

J の関数 `r.(Angle/Polar)` で  $e^{i\theta} = \cos\theta + i\sin\theta$  を計算し、`plot` のガウス・アルガン座標を用いて描く。

```
a, > clean L:0 ((cos + j.@sin);(^@j.);r.) (L:0) a=.steps _1p1 1p1 3
```

```
_3.14159      _1.0472      1.0472 3.14159  NB. a
  _1 0.5j_0.866025 0.5j0.866025  _1 NB. cos a + i sin a
  _1 0.5j_0.866025 0.5j0.866025  _1 NB. e^i a
  _1 0.5j_0.866025 0.5j0.866025  _1 NB. r. a
```

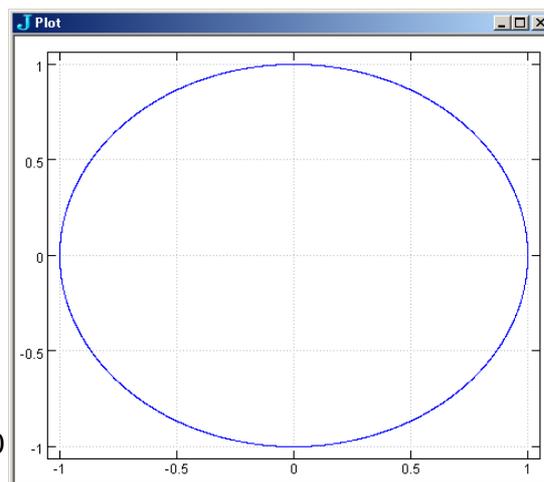
```
require 'plot numeric trig'
```

NB. 同じ図を描く

```
plot r. steps _1p1 1p1 100
```

```
plot ^@j. steps _1p1 1p1 100
```

```
plot(cos + j.@sin)steps _1p1 1p1 100
```



陰関数のグラフ 陰関数をそのままグラフに表す *Implicit Plot* 関数を備えているソフトもあるが、Jには入っていないので工夫する。郡山にはコーシーの中間点の定理を応用して値を比例配分する求め方が紹介されているがもっと簡単に `viewmat` を用いることとする。

- 円の陰関数表示は  $x^2 + y^2 - r^2 = 0$

- 目の細かいメッシュを作成する。

```
aqua4 steps _1 1 5

1    _1    1    _0.6    1 _0.2    1 0.2    1 0.6    1 1
0.6  _1    0.6  _0.6    0.6 _0.2    0.6 0.2    0.6 0.6    0.6 1
0.2  _1    0.2  _0.6    0.2 _0.2    0.2 0.2    0.2 0.6    0.2 1
_0.2 _1    _0.2 _0.6    _0.2 _0.2    _0.2 0.2    _0.2 0.6    _0.2 1
_0.6 _1    _0.6 _0.6    _0.6 _0.2    _0.6 0.2    _0.6 0.6    _0.6 1
_1   _1    _1   _0.6    _1  _0.2    _1  0.2    _1  0.6    _1  1
```

- 陰関数で計算する

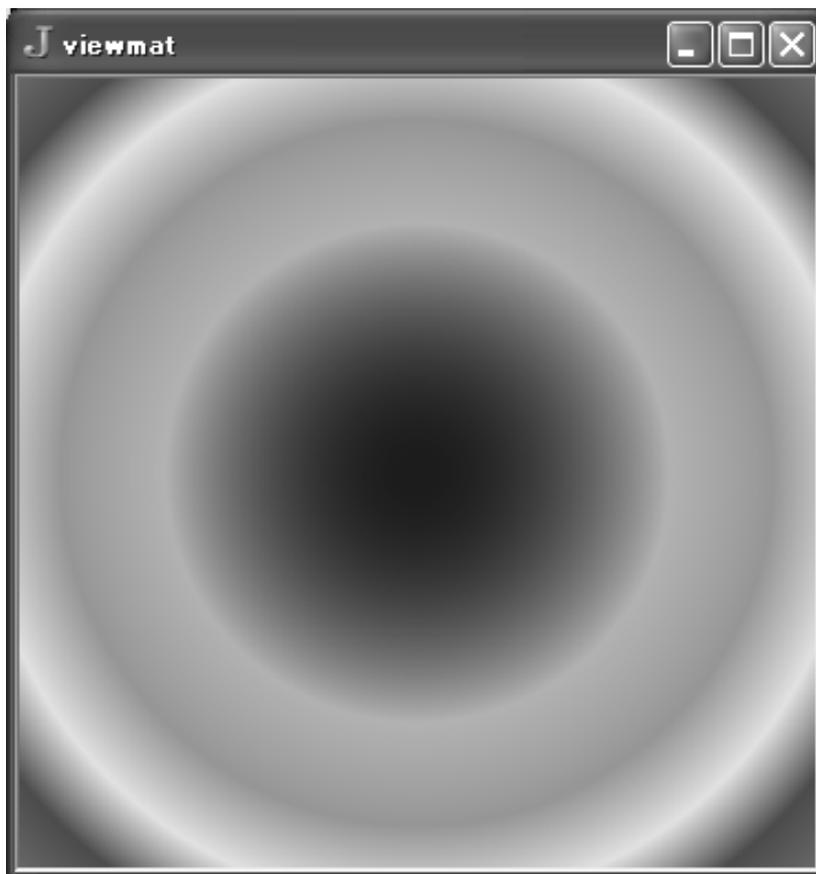
```
implicit_circ0 implicit_calc steps _1 1 5

1    0.36  0.04  0.04  0.36    1
0.36 _0.28 _0.6  _0.6  _0.28  0.36
0.04 _0.6  _0.92 _0.92  _0.6  0.04
0.04 _0.6  _0.92 _0.92  _0.6  0.04
0.36 _0.28 _0.6  _0.6  _0.28  0.36
1    0.36  0.04  0.04  0.36    1
```

- *viewmat* で表示する

```
viewmat implicit_circ0 implicit_calc steps _1p1 1p1 1000
```

陰関数 = 0 のポイントを抜き出すよりも関数全体を表示している。(= 0) を抜き出す計算は容易)



- *Script*

NB. very small mesh

```
aqua4=: 3 : '|. { y ; y ' NB. steps _1p1 1p1 100
```

```
implicit_calc=: 1 : 0
```

```
mesh=: aqua4 y
```

```
clean > u L:0 mesh
```

```
)
```

```
implicit_circ0 =: 3 : '<: (^&2 {. y) + ^&2 {: y'
```

1.  $1 : 0$  は関数を引数にとる副詞型
2. { カタログ (組み合わせの一覧を作る) を利用

3. |. rotate 上下の反転
4. clean : numeric.ijs にあるゴミ取り関数
5. steps a b n : numeric.ijs にある数列生成関数

## 1.2 楕円

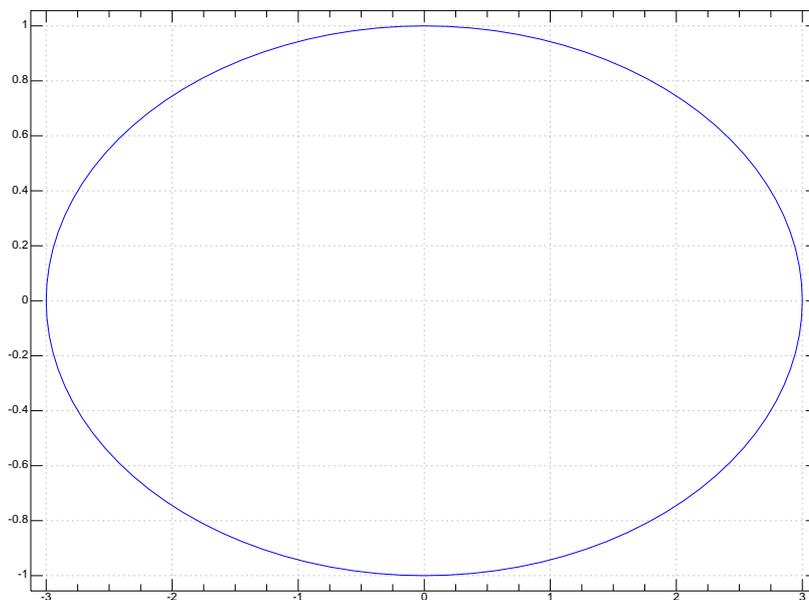
数式

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

パラメーター表示 こちらで描く

$$\begin{cases} x = a \cos \theta \\ y = b \sin \theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$

```
plot 3 1 ellipse steps _1p1 1p1 100
```



```
ellipse=: 4 : 0
```

```
NB. Usage:(plot) 3 1 ellipse steps _1p1 1p1 100
```

```
'a b'=: x
```

```
(a * cos y)+(b * j.@sin y)
```

```
)
```

### 1.3 アステロイド

数式 .

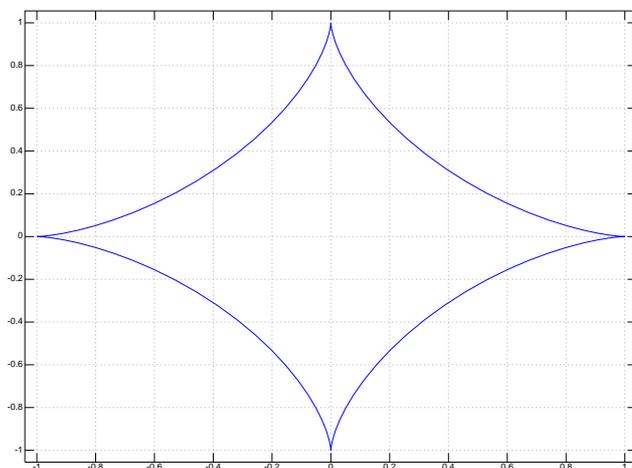
$$x^{\frac{2}{3}} + y^{\frac{2}{3}} = a^{\frac{2}{3}}$$

パラメーター表示

$$\begin{cases} x = a\cos^3\theta \\ y = a\sin^3\theta \end{cases} \quad (a \geq 0)$$

plot パラメーター表示に慣れてきたので直接描いてしまおう

```
plot (^&3@cos;^&3@sin) steps _lp1 lp1 100
pd 'eps c:/asteroid.eps'
```



### 1.4 リサージュ曲線

数式 .

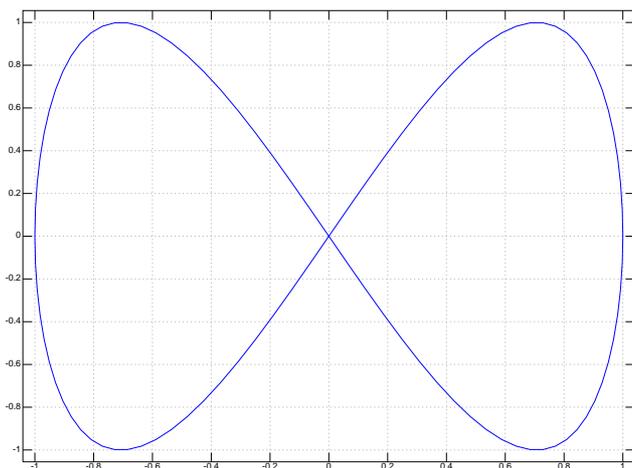
$$y^2 = 4x^2(1 - x^2)$$

パラメータ表示

$$\begin{cases} x = \sin\theta \\ y = \sin 2\theta \end{cases} \quad (0 \leq \theta \leq 2\pi)$$

plot .

```
plot (sin;sin@*&2)steps _1p1 1p1 100
```



## 2 極座標、極方程式

極方程式  $f(r, \theta) = 0$

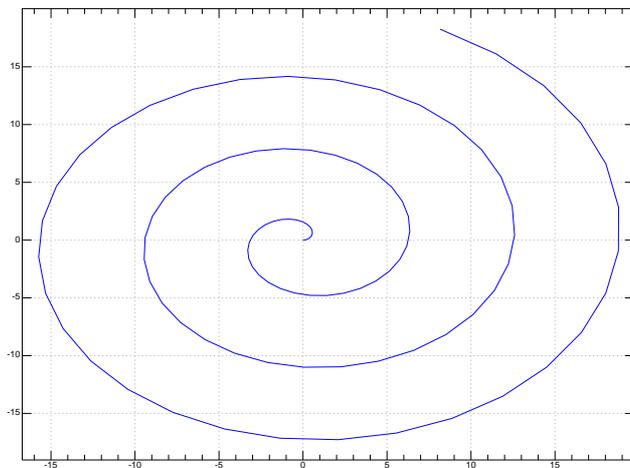
### 2.1 螺旋

アルキメデス螺旋 .

$$r = a\theta \quad (0 \leq \theta \leq 2\pi)$$

```
plot t * r. t=. steps 0 20 100
plot t r. t=. steps 0 20 100 NB. same
plot r.~ t=.steps 0 20 100 NB. same
```

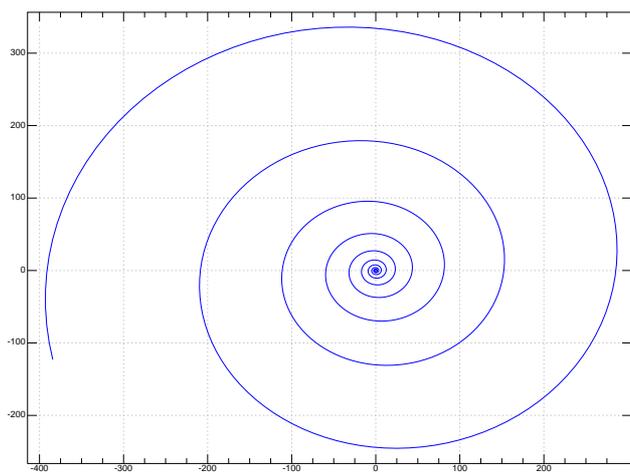
両項の r. は Polar x r. y --> x \* r. y である。



ベルヌイ螺旋 .

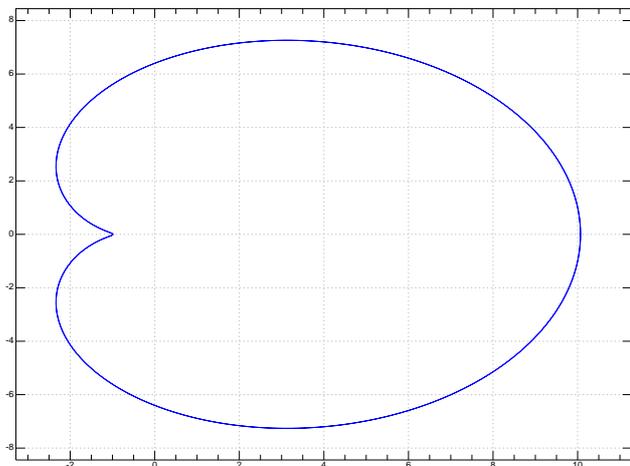
```
plot (^0.1* t) r. t=.steps 0 60 1000
```

係数 0.1 を変化させると巻きの硬さが変化する



## 2.2 カージオイド

```
plot cos 1p1 r. t=.steps _1p1 1p1 100
```

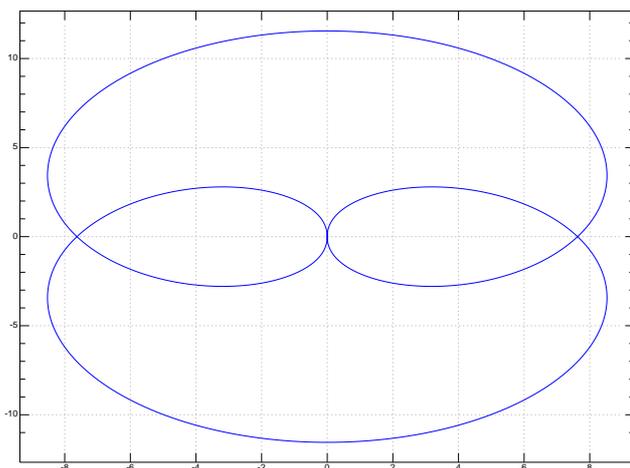


- 最初に円を作る  

```
plot r. steps _1p1 1p1 1000
```
- cos のみでは円のまま。3 倍するとカージオイドになる  
 \*1

## 2.3 Etc

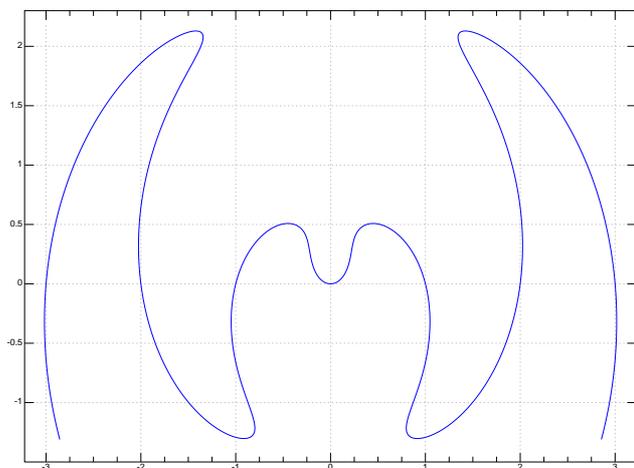
```
plot sin 1p1 r. t=.steps _1p1 1p1 1000 NB. sin 1p1 * r. t
```



```
plot t r.sin 1p1 * t=.steps _1p1 1p1 100
```

---

\*1 郡山の数式とは異なる



### 3 陰関数のグラフ

`viewmat` は細分した区間のマトリクス of 等高線を多色で色付けるので関数全体の構造が理解できる。

$$x^3 - 2xy + y^3 - 0.2 = 0$$

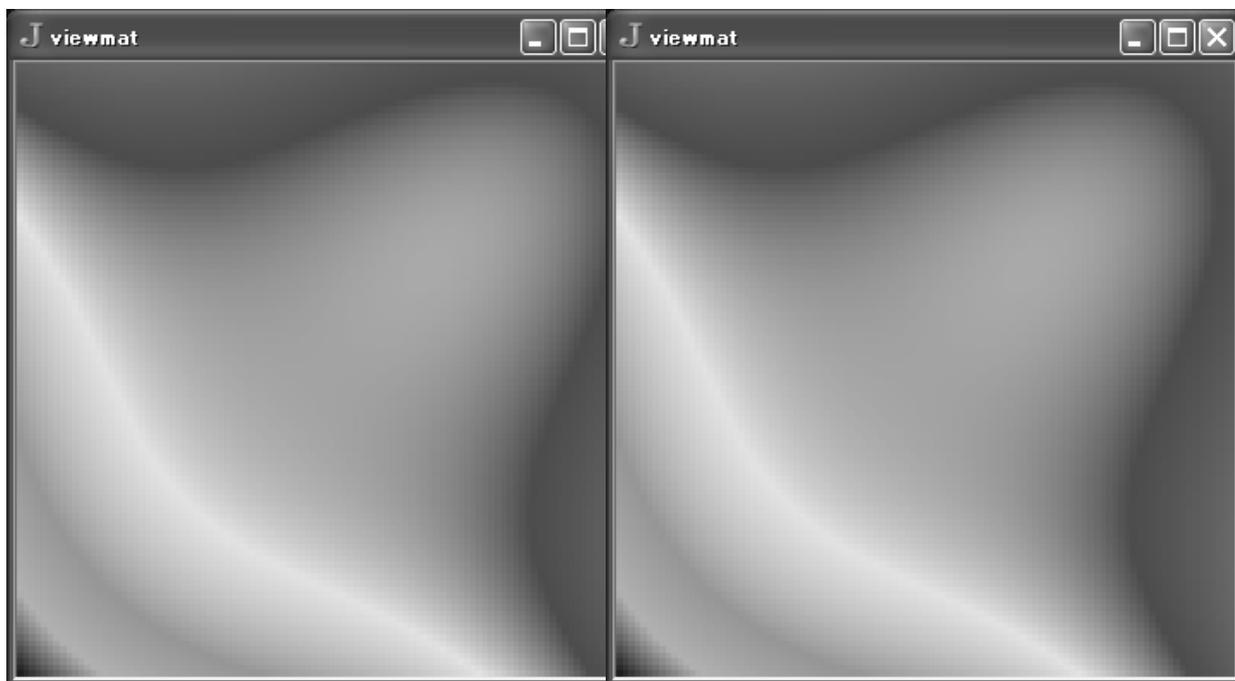
$$x^3 - 2xy + y^3 + 0.2 = 0$$

```
viewmat implicit_fn0 implicit_calc steps _2 2 100
```

```
viewmat implicit_fn1 implicit_calc steps _2 2 100
```

$$x^3 - 2xy + y^3 - 0.2 = 0$$

$$x^3 - 2xy + y^3 + 0.2 = 0$$



## References

郡山彬・原正雄・峯崎俊哉「CGのための線形代数」森北出版 2000  
*Script kohri\_graph0.ijs*