# ゴムシートの幾何と PLOT(その 0)

SHIMURA Masato
jcd02773@nifty.ne.jp

2013 年 5 月 23 日

## 目次



## はじめに

J の OpenGL のソースを鑑賞しながら、plot 用に整理した

J の plot と OpenGL util の関数を使うには最初に次のようにセットする。

```
require 'plot numeric trig'
require 'opengl'
coinsert 'jzopenglutil'
```

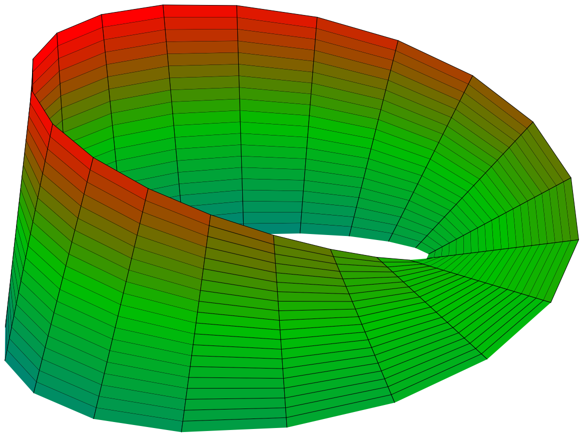
各図形の OpenGL のソースは `system/examples/graphics/opengl/demo` に、openglutil は `system/classes/opengl` に入っている。

# 1 メビウスとクライン

## 1.1 メビウスの帯

ヨハン・リスティングはガウスの学生でアウグスト・メビウスはガウスの助手。半ひねりのメビウスの帯は１８５８にリスティングとメビウスによって示された。

’noaxes’ plot moebius ”

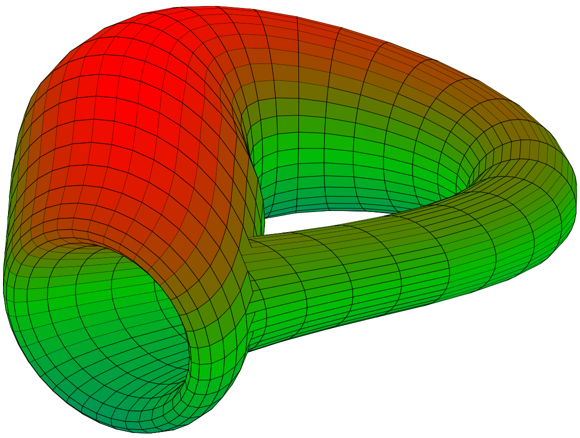


```
moebius=: 3 : 0
u=. steps 0 2p1 20
v=. steps _0.3 0.3 20
x=. (cos u) + ((cos u) * cos u%2) */ v
y=. (sin u) + ((sin u) * cos u%2) */ v
z=. (sin u%2) */ v
DAT=. x; y; z
)
```

## 1.2 クラインの壷

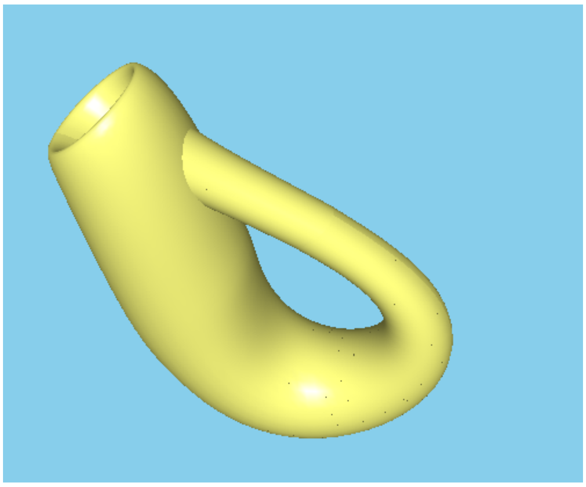
片面しかない曲面の代表格。クラインの曲面 (flache) を壷 (flasche) と言い換えたジョークらしい。古代ペルシャのリュトン

’noaxes’ plot klein2 ”



```
klein2=: 3 : 0
stp=. 19
u1=. steps 0 1p1, stp
u2=. }. steps 1p1 2p1, stp
u=. v=. u1,u2                     NB. linear 0 to 2p1 (wi
r=. 4 * -. 0.5 * cos u            NB. upward parabolic
j=. ((cos u1) */ cos v) , (0 * u2) +/ cos v + 1p1
x=. (6 * (cos u) * 1 + sin u) + r * j
y=. (16 * sin u) + r * ((sin u1), 0 * u2) */ cos v
z=. r * (0 * u) +/ sin v
DAT=. x;y;z
NB. 1.6 * gsfit11 gsmakexyz x;y;z
)
```

```
plot  klein2_gldemo_ ''
```



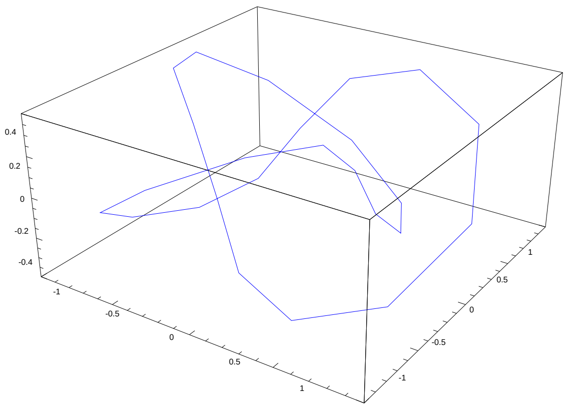
# 2 Knot 系

## 2.1 三つ葉模様

両端が繋がった 3 次元の plot 用の関数が出来れば、OpenGL に落とし込めばよい

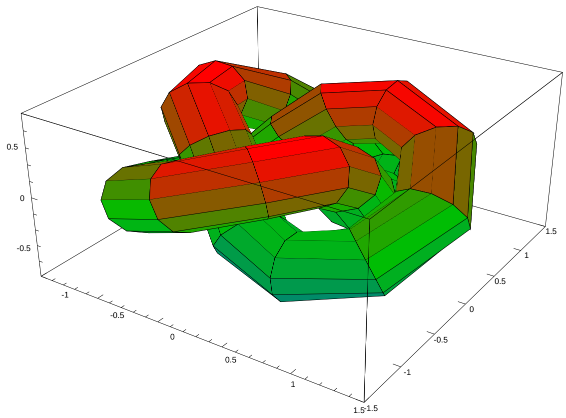
```
V=. "_
r=: 1: + 0.4 V * cos@(1.5&*)
fx=: r * cos
fy=: r * sin
fz=: 0.5 V * sin@(1.5&*)
fn=: (1.7 V * fx,fy,fz) f.
NB. TREFOIL=: fn gsmakefknot (steps 0 16r4p1 25);11;2
```

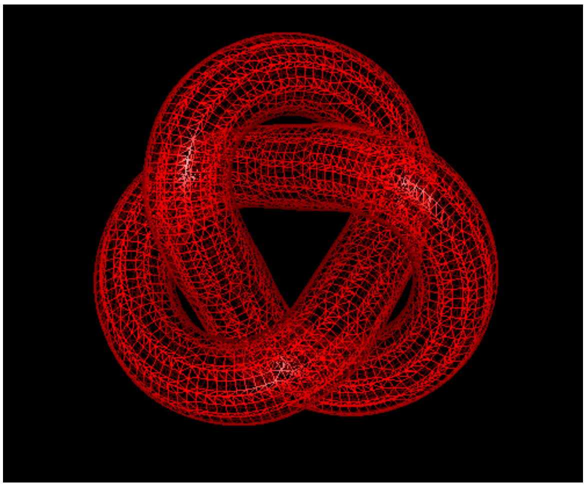
plot (fx;fy;fz) L:0 steps 0 16r4p1 25



OpenGL 用に整形したもの

```
plot <"2 |: TREFOIL
```





## 2.2 リサージュ曲線

リサージュ曲線 (1855) は直交する 2 つの単振動を合成したもので次のように表され、周波数や信号の解析に用いられる。

（*J.A.Lissajous 1822-1880)*

$$x(t) = Acos(\omega_x t - \sigma_x)$$
$$y(t) = Bsin(\omega_y t - \sigma_y)$$

| | | |
|---|---|---|
| $\omega$ | 角振動数 | $0\langle\omega \leq$ |
| | 振幅比 | $\frac{a}{b}$ |
| $\sigma$ | 位相差 | $0 \leq \sigma \leq 360$ |

また、次のようにも書かれる

$$x(t) = asin(\omega_x t + \sigma)$$
$$y(t) = bsin(t)$$

*x,y* のみとして素直にプログラムしてみる。

```
lissajou=: 4 : 0
NB. 1 2 _1r4p1 u steps_gldemo_ _5 5 100
'A B SIGMA'=: x
X0=:   2 o. SIGMA +  A * y
Y0=: 1 o. SIGMA +  B*y
X0;Y0
)
```

```
plot 2 5 _1r4p1 lissajou steps _50 50 10000
```

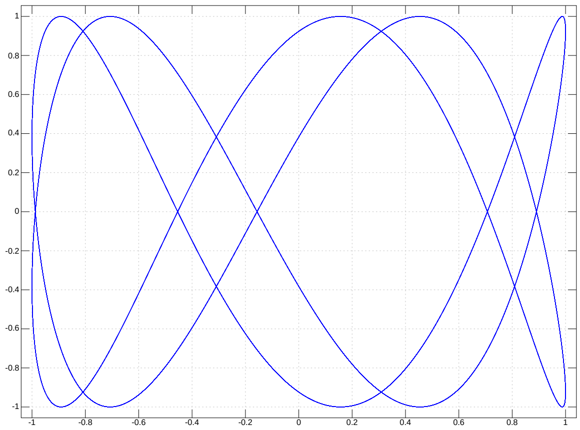


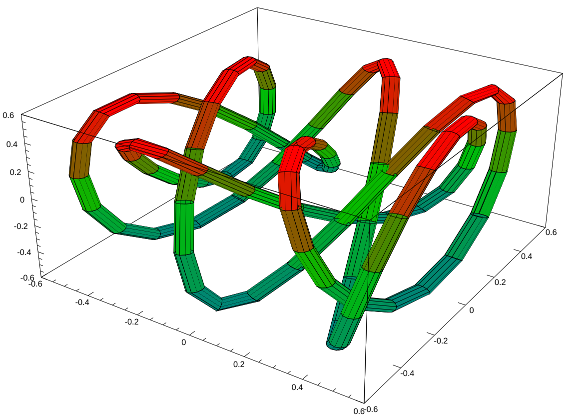
図 1　lissajou(2 5 _1r4p1)

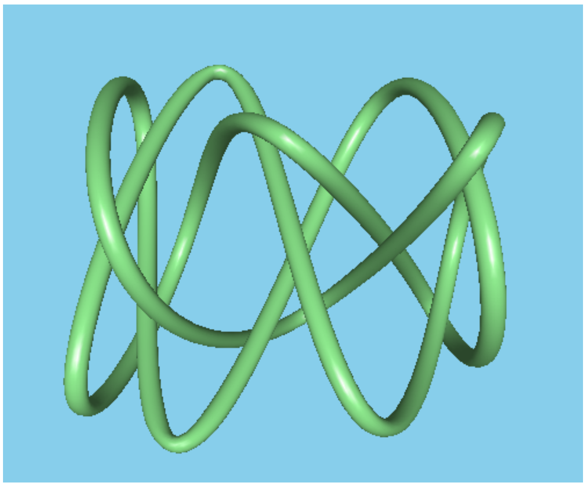
次は *J* の *demo* の計算スクリプトで *3D* である。

*gsmakefknot* は *OpenGL* のデータ生成関数、*gsdrawknot* が描画関数である。

```
fx=: 0: + 2&*
fy=: 1: + 5&*
fz=: 2.1"_ + 7&*
fn=: cos @ (fx,fy,fz) f.
sp=: o. (<:i.92) % 45
KNOT=: fn gsmakefknot sp;11;0.2
```

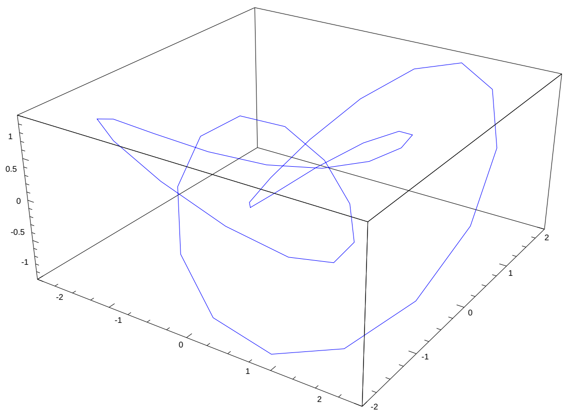
```
plot <"2 |: KNOT
```



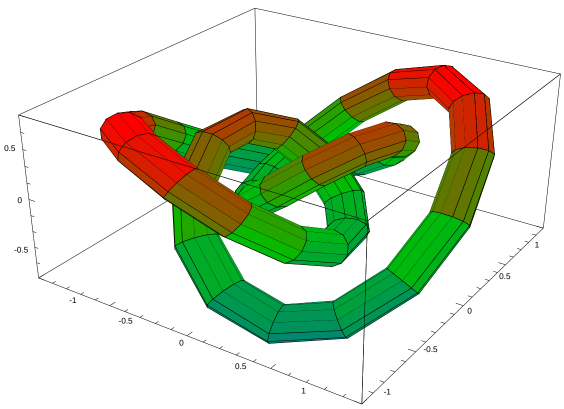


## 2.3 トーラス

```
V=. "_
fx=. cos@- + 1.5 V * cos@(3&*)
fy=. sin + 1.5 V * sin@(3&*)
fz=. sin@(4&*) + sin@(2&*) % 3:
fnt=: (fx,fy,fz) f.
NB. TORUSKNOT=: fnt gsmakefknot (steps 0 2p1 40);11;1
  plot (fxt;fyt;fzt) L:0 steps 0 2p1 40
```




```
 plot <"2  |: TORUSKNOT=: fnt gsmakefknot (steps 0 2p1 40);11;1
```




# 3 アレキサンダー多項式

*1927* *US* のジェームス・ワッデル・アレキサンダーとその学生 *G.B.* ブリッグスによる結び目図と多項式

| | |
|---|---|
| 三つ葉型結び | $x - 1 + x^{-1}$ |
| 8 の字結び | $-x + 3 - x^{-1}$ |
| こま結び | $x^2 - 2x + 3 - 2x^{-1} + x^{-2}$ |
| 縦結び | $x^2 - 2x + 3 - 2x^{-1} + x^{-2}$ |
| 沖中仕結び | $-2x + 5 - 2x^{-1}$ |

# 4 References

イアン · スチュアート 「世界を変えた１７の方程式」*SoftBank Creative 2013*