SPP 講座 デザイン定規による作図

土基 善文

1. 「デザイン定規」を使ってみよう。

1.1. とにかくどんどん描いてみよう. 実際にデザイン定規をお渡ししますから、まずは三つぐらい描いてみて下さい。

- うまく描くにはどこに注意すればいいか考えながらやってみて下さい。
- 一番気に入ったのはどの形ですか。
- 描く前に予想した通りの図になりましたか。

1.2. どんな所に気を付けないとうまく行かないですか。.

- 固定
- すべらない
- 接する
- 無意識に考えていること
- 順序の設定

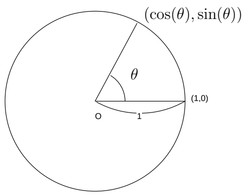
2. グラフをパラメータ表示する

2.1. 三角関数の復習. 単位円上を $(1,0)$ から速度 1 で走る Q ちゃんの t 秒後の座標が

$$(\cos(t), \sin(t))$$

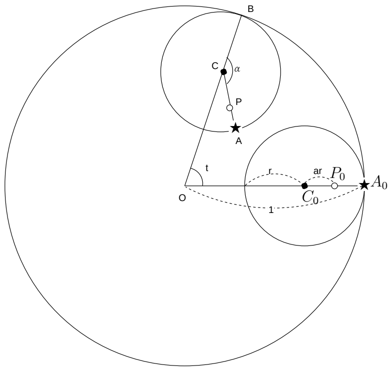
なのでした。

(ここでは **弧度法**をもちいています。小学生のときに習う 60 進法をもちいることは大学の数学ではほとんどありません。そんなことをすると三角関数の微分の公式が必要以上に複雑になるなどの不都合が起こるからです。)



円をパラメータ表示をするのには、三角関数を用いるのがもっとも自然でしょう。(その理由から、三角関数は円関数と呼ばれることもあります。)

2.2. 簡単な図を描く. デザイン定規のグラフ (内トロコイドとも呼ばれます) をデザイン定規なしに考察したり、コンピュータに描かせたりすることを考えてみましょう。まずは簡単な「見取図」を描いてみることが必要です。



2.3. ポイントを図に書き入れる. デザイン定規で描画する際に必要なデータをまずそろえます。実際に描いた手順をよく思い浮かべれば良いでしょう。

(1) 大円の中心。これは原点とします。
(2) 大円の半径。これは基準の長さ 1 とするのが便利でしょう。
(3) 小円の半径。 r とします。
(4) 小円の「鉛筆をいれる穴」の位置。これは小円の中心から ar 離れているとしましょう。もちろん $0 < a < 1$ です。

製図するときに必要なデータ。

(1) 図の t。これは基本になるパラメータです。
(2) 図の α。これは t から求まるはずの数です。

では α をどうやって求めれば良いでしょうか。そのためには、「滑べらずに動かす」ということを表現することになります。結果だけ書くと、

$$\alpha r = (\text{小円の弧 } AB \text{ の長さ}) = (\text{大円の弧 } A_0B \text{ の長さ}) = t.$$

さて、

$$\overrightarrow{OC} = r(\cos(t), \sin(t))$$
$$\overrightarrow{CA} = ar(\cos(t-\alpha), \sin(t-\alpha))$$

ですから A の座標は

$$(1-r)(\cos(t), \sin(t)) + ar(\cos(st), \sin(st)) \qquad (\text{但し } s = 1 - \frac{1}{r})$$

となります。これが内トロコイドのパラメータ表示です。つぎの話題に移る前に、この表示のうち、どの変数が動いて、どの変数が固定されているかしっかり確認してください。

2.4. コンピュータにグラフを書かせてみる. ここでは gnuplot というフリーソフトを用いてみました。配布元は `http://www.gnuplot.info/` です。フリーなので次のような利点があります。

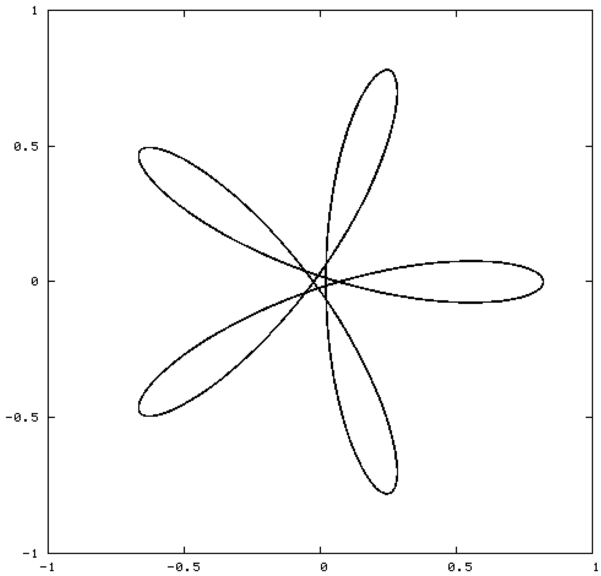
- ネットにつながっている状況なら簡単に手に入れることができる。
- 大勢の人が使っているので必要な使い方の手引も比較的容易に見付けることができる。

2.4.1. プログラム例.

```
############################################################
#
# サンプルプログラム 1
# このプログラムを sample.plt という名前で保存して
#  gnuplot から 開くことで実行できます。
############################################################
# 定数の設定。
r=63.0/105.0;
s=1-1/r;
a=0.7;
l=1000;
# 描画のパラメータの設定。
# 縦横の大きさをそろえる。
# パラメータで描画する。
# 描画のためのサンプル数は 10000 にする。
set size square
set parametric
set samples 10000


# 実際にプロット
#（行末の ’\’ は行が継続していることをあらわす。）
plot [0:l] [-1:1] [-1:1] \
(1-r)*cos(t)+a*r*cos(s*t), (1-r)*sin(t)+a*r*sin(s*t) ;
############################################################
```

2.4.2. 出力結果. サンプルプログラム 1 を実際に gnuplot で動かしてみるとつぎのようなグラフが表示されます。



パラメータを変化させていろいろな図を描かせてみると楽しいし、プログラムやそれに使った考え方が正しいことも確認できます。

もっと凝れば、実際に内トロコイドを描いているようすをアニメーションとして作成すること等もできます。

3. 「葉っぱの数」を数える

3.1. 「葉っぱの数」を定義するには. 「葉っぱ」を数えるには、葉っぱの先端、すなわち原点からの距離が極大であるような点が何点あるか数えるのが良いでしょう。

そのような点に対応する t 全体の集合は a に依存しないので、$a=1$ とおくと簡明ですね。大円の円周 2π を小円の円周 $2\pi r$ で切るのと同じことになります。

$$(\text{葉っぱの端点}) = \{P_k = (\cos(2\pi rk), \sin(2\pi rk)); \qquad (k \in \mathbb{Z})\}$$

3.2. 葉っぱの数の計算法. r が有理数のときをまず考えてみましょう。このときには

$$r = \frac{p}{q}$$

と既約分数のかたちに書いておくと

$$P_k = P_l \Leftrightarrow r(k-l) \in \mathbb{Z} \Leftrightarrow k-l \in q\mathbb{Z}$$

であることから、

$$P_0, P_1, P_2, \ldots P_{q-1}$$

の q 個が相異なり、その他の点 P_k は上の q 個のいずれかに一致することがわかります。つまり、葉っぱの数は q です。

3.3. r **が無理数なら.** 小円が何周まわろうと、最初の場所に鉛筆がもどってくることはありません。葉っぱの数は (あえて言うなら) 無限です。紙に描くと「塗りつぶされる」ようなグラフになります。(が、実は厳密には (線の太さを考えにいれなければ) 完全に塗りつぶされているわけではありません。) 詳しくは各自の考察にお任せいたしましょう。

3.4. 実際のデザイン定規. ちょっと興ざめな話をしましょうか。

実際のデザイン定規では各円周にギザギザがついて、ギアになっています。だから

$$r = \frac{(\text{小円のギアの数})}{(\text{大円のギアの数})}$$

となって、r は必ず有理数です。とくに、「やっているうちに紙が塗りつぶされる」ような組み合わせはないということになります。

問題: 実際のデザイン定規のギザギザの山の数は何個あるでしょうか。

ちゃんと戻ってくることが保証されている有理数の世界と、そんなことはない無理数の世界。どちらが「きれい」でしょう。