

# 比較対照 METAPOST & Asymptote ~ 第1回

とりあえず今回は、点 ( pair 値 ) をいくつか設定して、その点を通る曲線を引いてみましょう。

## 1 点 ( pair 値 ) の設定

pair 値の設定方法は、基本的に METAPOST と Asymptote で変わりがないのですが、受け取る側の pair 値変数の宣言の仕方が違います。

まず、METAPOST ではデフォルトで z(サフィクス) を pair 値 ( のマクロ ) と設定してくれていますが、Asymptote ではそのような設定はなされていません。また、新たに pair 値変数を宣言するときの書式も METAPOST と Asymptote では少々異なります。

METAPOST では、たとえば zz を pair 値変数として宣言して、値を (2,3) に設定するときは

```
pair zz; zz:=(2,3);
```

のように、2文に分けて設定する必要がありましたが、Asymptote では

```
pair zz=(2,3);
```

のように、1文で一気に入指できます<sup>1</sup>。なお、METAPOST では := が代入、= が等置でしたが、Asymptote は = が代入であることにも注意してください。Asymptote には等置はありません<sup>2</sup>。

次に pair 値の配列について。METAPOST では zz[1], zz[2], ... は zz1, zz2, ... と省略できましたが、Asymptote ではそのような省略は許されません。zz1 と zz2 は配列の要素ではなく、まったく別の変数と解釈されます。配列の宣言の仕方も、METAPOST よりは C 言語における宣言に似ています。

METAPOST で zz[1], zz[2], ... を pair 値の配列として宣言するには

```
pair zz[];
```

としますが、Asymptote では次のようにします。

```
pair[] z = new pair[4];
```

Asymptote では配列の要素数の最大値を宣言時に指定することに注意してください。

宣言した変数に具体的な pair 値を設定するときを使う表記や関数は METAPOST と Asymptote でほぼ同じです<sup>3</sup>。ただ、METAPOST と Asymptote で意味が違う関数や Asymptote のみにある関数もありますので、その辺の違いを紹介しておきます。

まず、表記が違うものは次の表の通りです。

意味	METAPOST での表記	Asymptote での表記
z と同じ向きの単位ベクトル	unitvector(z)	unit(z)
z の方向角 ( 度数法 )	angle(z)	degrees(z)
z と w の内積	z dotprod w	dot(z,w)

とくに、angle の意味が変わっていることにご注意下さい。Asymptote にも angle 関数はあるのですが、こちらは角を弧度法 ( $-\pi < \theta \leq \pi$ ) で返します。ちなみに、angle や degree の逆関数に相当する dir は METAPOST と Asymptote で同じ意味 ( 引数が度数法の方向角、戻り値が指定方向の単位ベクトル ) ですが、Asymptote には引数を弧度法にした expi という関数もあります<sup>4</sup>。

次に、Asymptote だけの関数。

conj(z) — z の「複素共役」を返します。たとえば z が (2,3) なら conj(z) は (2,-3) です。

realmult(z,w) — z と w の成分毎の積です。たとえば realmult((1,2),(3,4)) は (1\*3,2\*4) で (3,8) です。

minbound(z,w) — x 成分, y 成分とも、z と w の小さい方<sup>5</sup>を採用した値を返します。

maxbound(z,w) — x 成分, y 成分とも、z と w の大きい方を採用した値を返します。

length, xpart, ypart<sup>6</sup> や上述の dir などは METAPOST と Asymptote で同じ意味です。また、具体的な数値を掛け算するとき、2z のように 2 と z の間の乗算記号 \* を省略できるところも共通です。

<sup>1</sup>Asymptote で宣言時に値を指定せず単に pair zz; とすると、初期値は (0,0) に設定されます

<sup>2</sup>つまり METAPOST 風に連立方程式で pair 値などを設定することはできません。どうしても連立方程式を使いたいときは、「連立方程式を解く」命令を明示的に使うことになります。

<sup>3</sup>METAPOST には引数を囲う ( ) を省略できる関数が多数ありますが、Asymptote では基本的に引数を囲う ( ) を省略できません。METAPOST に慣れ親しんだ人はとくにご注意ください。

<sup>4</sup>記号の名前は、もちろんオイラーの公式  $\exp(i\theta) = e^{i\theta} = \cos \theta + i \sin \theta$  からきています。

<sup>5</sup>正確には「大きくない方」です。同様に maxbound の方も正確には「小さくない方」です。

<sup>6</sup>Asymptote では z の x 成分, y 成分を z.x, z.y で読み出すことができます。ただし、書き込み ( 成分毎の値の変更 ) はできません。値を変更したい場合は z そのものを変更して下さい。

あと、 $z=(2,3)$ ; のように単位をつけずに表記した場合、単位長が bp になることや、cm、mm などが実は単位ではなく、それぞれ 1cm、1mm を bp 単位で表した変数であることなども METAPOST と Asymptote で共通です。なお、Asymptote には「単位長を変更」する `size`、`unitsize` という命令があるのですが、これは正確には単位長を変更するのではなく、あとで図全体を拡大縮小するコマンドですので注意が必要です。METAPOST や Asymptote の cm や mm は実寸法ではなくただの変数であるため、実寸法のため 1cm と指定していても、実は  $1\text{cm}/1\text{bp} = 28.3465$  ですので、数値 28.3465 と指定したのと同じになります。したがって、`unitsize(10bp)` とすれば、すべてが 10 倍されて 1cm のはずが  $283.465 = 10\text{cm}$  になってしまいます。

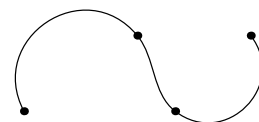
## 2 曲線を引いてみる

では、実際に曲線を引いてみましょう。線を描くだけなら METAPOST と Asymptote で表記はほぼ同じです。ただし、Asymptote では `draw` 関数の引数を囲う ( ) が省略できないことにはご注意ください。

点を打つ命令は METAPOST では `drawdot` ですが、Asymptote では `dot` です。

ついでに、`for` 文を用いて繰り返しを表現してみました。`for` 文がモロに C 言語って感じで、METAPOST とは全然違うことに注意してください。

```
\begin{ASypic}<1cm>(0,0)(4,-1.5)
\sendASY{pair[] z = new pair[4];
z[0]=(0,0); z[1]=(1.5,1); z[2]=(2,0); z[3]=(3,1);
path p=z[0]..z[1]..z[2]..z[3];
draw(p);
for(int i=0; i<4; ++i){dot(z[i],linewidth(3pt));}
}
\end{ASypic}
```



次に線種や色、線の太さを変えてみます。METAPOST では線種を破線に、色を赤に変えるとき、

```
draw p dashed evenly withcolor red;
```

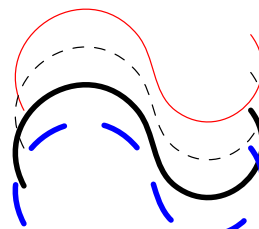
のように `draw` の後置文<sup>7</sup>で指定し、また、線の太さは、

```
pickup pencircle scaled 1pt;
```

のように「ペン先の形状 ( `pen` 値変数 `currentpen` )」の変更として実現していました。

それに対し、Asymptote では線種や色、線の太さすべてを `pen` 値変数<sup>8</sup>がもっており、その `pen` 値変数を `draw` の引数に与える<sup>8</sup>ことで、線種や色、線の太さを変更します。したがって、以下の例で `red` や `blue` は `color` 値変数<sup>9</sup>ではなく `pen` 値変数です。複数の属性を一度に変えるときは + で結ぶことや、`pen` 値 + `real` 値のときその `real` 値は `linewidth(real 値)` と解釈されることにも注意して下さい。( `pen` 値の詳しい解説は長くなりますので、いずれまた。)

```
\begin{ASypic}<1cm>(0,0)(4,-3)
\sendASY{pair[] z = new pair[4];
z[0]=(0,0); z[1]=(1.5,1); z[2]=(2,0); z[3]=(3,1);
path p=z[0]..z[1]..z[2]..z[3];
draw(p,red);
draw(shift(0,-0.5)*p,dashed);
draw(shift(0,-1)*p,linewidth(2pt));
draw(shift(0,-1.5)*p,blue+dashed+2pt);
}
\end{ASypic}
```



上の例ではついでに `transform` 値による `path` 値の変形 ( この場合平行移動 ) をしてみました。METAPOST の `shifted` が Asymptote では `shift` であることや、変形したいもの ( 上の例の場合 `path` 値変数 `p` ) の左から \* を介して作用させることにご注意下さい。( `transform` 値の詳しい解説も長くなりますので、いずれまた。)

<sup>7</sup>今勝手に名付けました。正式名称ありましたっけ？

<sup>8</sup>Asymptote の関数には省略可能な引数が多数あり、省略した場合はデフォルト値が選択されます。先程の例で `pen` 値が指定されていなかったのはそういうことです。

<sup>9</sup>Asymptote にはそもそも `color` 値変数はありません。Asymptote の 3 次元ベクトルは `triple` 値変数といえます。