

# 比較対照 METAPOST & Asymptote ~ 第7回

## 1 ラベル

今回はラベルの貼り付け方を説明します。なお、(METAPOST を使ったことがある方は御存知と思いますが) METAPOST や Asymptote で言うラベルは L<sup>A</sup>T<sub>E</sub>X のラベルと異なり、図中に挿入するテキスト (文字列) のことを指します<sup>1</sup>。

METAPOST でのラベルは

```
label<サフィックス>(string 値または picture 値, pair 値)
```

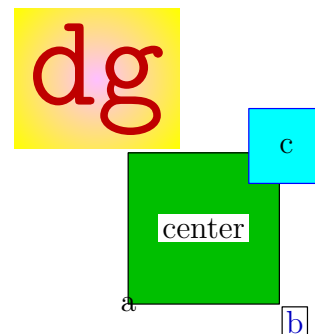
で貼り付けられます。「string 値または picture 値」が貼り付けたいラベルで、pair 値が貼り付け先の座標です。〈サフィックス〉は貼り付け先の座標に対しラベルがどちらの方向に配置されるかを示します。〈サフィックス〉が〈空〉、つまり、〈サフィックス〉がないときは貼り付け先の座標がラベルの中央になるよう貼り付けられます。〈空〉以外で指定できるのは .lft, .rt, .top, .bot, .ulft, .urt, .llft, .lrt のいずれかで、意味は順に、ラベルを貼り付け先の座標の左に、右に、上に、下に、上左に、上右に、下左に、下右に配置する<sup>2</sup>というものです。たとえば、label.lft("1",z1); で点 z1 の左に 1 という文字が貼り付けられます。

Asymptote のラベルも METAPOST と同様に使えるのですが、それだけでなく大幅な機能拡張が行われています。たとえば、配置方向は任意の向きですし、貼り付け先も点 (pair 値) だけでなく〈パス〉 (path 値) も指定でき、さらに、文字列を枠で囲ったり矢印を付けたりといったおまけの機能も充実しています。—ただ、充実しすぎていて1回で説明できそうにありません。とりあえず今回は基本的な使い方だけです。あと、大事なことを1つ — デフォルトでは和文は文字化けします。和文など多言語に対応するには、import unicode; とした上で、(Asymptote から呼び出された) T<sub>E</sub>X 側で babel パッケージを用いる必要があります (詳しくは Asymptote マニュアル p.91 を参照です)。なんかちょっと大事っぽいです。

## 2 基本的な使い方

まず今回は、一番単純な点への配置について。サンプルは次の通りです。

```
\begin{ASYpic}<2cm>(0,0)(1.6,-1.7)
\sendASY{import fontsize;% 50pt フォントの出力のため
filldraw(unitsquare,heavygreen,black);
label("a",(0,0));
label("b",(1,0),SE,heavyblue,Draw);
label("c",(1,1),Align,FillDraw(10pt,10pt,cyan,blue));
label("dg",(0,1),0.5dir(110),font("OT1","cmtt","m","n")+fontsize(50pt)+heavyred,
RadialShade(palemagenta,yellow));
label(Label("center"),(0.5,0.5),Fill(white));
}
\end{ASYpic}
```



label 命令の書式は次の通りです。

```
label(picture 値, Label 値 (必須), pair 値 (必須), align 値, pen 値, filltype 値)
```

以下、引数の意味を順に説明していきます。

○ picture 値引数はラベルを描き込む〈ピクチャー〉を指定します。いつものごとく省略すれば currentpicture になります。

○ ラベル本体は Label 値引数で与えます。Label 値は METAPOST ユーザーには聞き慣れない変数タイプですが、ラベルに関するいくつかの情報をまとめて扱う変数タイプです<sup>3</sup>。ただ、単純な文字列の場合は Label 値の代わりに string 値でかまいません。内部で自動的に string 値を Label 値に変換 (型キャスト) してくれます。

<sup>1</sup>L<sup>A</sup>T<sub>E</sub>X のラベルは相互参照のために貼り付ける「葉」のようなもので、抽象的な意味の「ラベル」です。

<sup>2</sup>ちなみに METAPOST や Asymptote での配置方向の指定は、L<sup>A</sup>T<sub>E</sub>X の \makebox(0,0) などを使った指定法と逆になりますので注意して下さい。たとえば L<sup>A</sup>T<sub>E</sub>X の \makebox(0,0) で [r] (右) を指定すると、配置したいテキストの右端を貼り付け先の座標に合わせるため、テキストは貼り付け先の左側に配置されますが、METAPOST の label で rt (右) を指定しますと、テキストは貼り付け先の右側に配置されます。

<sup>3</sup>Asymptote は C++ のように「構造化されたデータ型」を扱うことができ、またユーザーが新たに設定することもできます。Label 型や align 型もそういった構造化されたデータ型の 1 種です。

- 貼り付け先の座標は pair 値引数で与えます。これはとくに説明は要らないでしょう。
- 貼り付け先の座標に対し、ラベルがどちらに貼り付けられるかは align 値引数で与えます。align 値引数を省略した場合、および、NoAlign を指定した場合は、貼り付け先の座標がラベルの中央になります。また、Align を指定した場合は、貼り付け先の座標がラベルの左下になります<sup>4</sup>。

「任意の方向」に貼り付けたいときには、align 値引数のところに pair 値を与えます（例によって型キャストされ、align 値に変換されます）。この場合、与えた pair 値にラベルマージン<sup>5</sup>をかけた値だけ、貼り付け先の点から離して配置されます。ちなみに、align 値にたとえば dir(90)（つまり上向き単位ベクトル）を指定すれば、貼り付け先の真上の前述の距離だけ離れた位置に、ラベルを囲う矩形の下辺中央を合わせます。また、dir(-45)（つまり右下向き単位ベクトル）を指定すれば、貼り付け先の右下の前述の距離だけ離れた位置に、ラベルを囲う矩形の左上頂点を合わせます<sup>6</sup>。

なお、よく使うのは「上向き」や「右下向き」などの単位ベクトルでしょうから、それらを表す pair 値には名前がついています。名前はいわゆる「16 方位」に対応し、たとえば N は「北」つまり上向きの単位ベクトル（dir(90)）、SSE は「南南西」の単位ベクトル（dir(-112.5)）になります<sup>7</sup>。

- 貼り付けるラベルのフォントのサイズ・種別や色などは pen 値の引数で与えます。省略すれば nullpen が指定され、これは通常さらに currentpen が指定された読み替えられます。

色の指定方法はこれまでと同じです。フォントサイズは fontsize 関数で与えます。書式は次の通りです。

fontsize(real 値 (必須: フォントサイズ), real 値 (行間隔))

行間隔は、省略した場合フォントサイズの 1.2 倍が設定されます。フォントサイズのデフォルト値は 12pt です。なお、標準で存在しないフォントサイズを指定したいときは import fontsize; を行っておくことが必要です<sup>8</sup>。

フォント種別は、 $\LaTeX$  の NFSS で指定するならばたとえば font("OT1", "cmr", "m", "n") のように指定します。例に挙げた引数の意味は OT1 エンコーディング、cmr ファミリー、ミディアム (m) シリーズ（普通の太さ）、ノーマル (n) シェイプ（直立体）です。シリーズ、シェイプは省略可能で省略すればそれぞれ "m", "n" が採用されます。なお、cmr12 のような固定サイズのフォントを使うなら font("cmr12") のようにフォント名だけで OK です<sup>9</sup>。あと、POSTSCRIPT フォントのいくつかについては、指定しやすいよう専用の関数（たとえば Helvetica()）が用意されています。Asymptote のマニュアルの p.40 にあがっていますので参照して下さい。

- さて、最後に filltype です。ラベルを囲う枠線を描いたり枠の内部を塗りつぶしたりできます。一般に filltype 値は「枠」に対する描き方・塗り方を指定する変数タイプで、前回の「タイル塗り」にも出てきました。以下のいずれかを指定します。

- FillDraw(real 値, real 値, pen 値, pen 値) — 2 つの pen 値引数のうち、前者で枠を塗りつぶし、後者で枠線を描きます。一方を省略すればもう一方の色が使われます。両方省略すれば、currentpen になります。また、2 つの real 値引数は順に横方向、縦方向のマージンで、指定された値だけ枠が大きくなります。省略すれば 0 と解釈されます。

- FillDraw — 上と同じように見えますが、こっちは関数ではなく変数です。FillDraw() の省略（つまり関数タイプの引数をすべて省略したもの）に相当します<sup>10</sup>。

- Fill(real 値, real 値, pen 値) および Fill, Draw(real 値, real 値, pen 値) および Draw — 「塗りつぶしのみ」あるいは「枠線のみ」描画するだけで、他は FillDraw と同じです。

- UnFill(real 値, real 値) および UnFill — Fill の「枠内塗りつぶし」を「枠内 clip」に置き換えたものと思えばいいでしょう。色指定の pen 値引数がなくなっただけで、他は Fill と同じです。ただ、label の引数にこれを使うと誤動作 (?) するみたいです（Asymptote ver1.88）。ラベル部分を白抜きするなら Fill(white) の方がいいようです。

- RadialShade(pen 値, pen 値) — 枠内をグラデーションで塗ります。引数は順に中心の色、端の色です。

- NoFill — 塗りつぶしはしません。枠線を描くかは状況次第ですが、label の引数に使ったときは枠線も描きません。label の filltype 値引数を省略すると、この NoFill になります。

<sup>4</sup>したがって、貼り付け先の座標の右上にラベルが貼り付けられます

<sup>5</sup>label に与えた pen 値引数を p とするとき、ラベルマージンは関数 labelmargin(p) で得られる real 値です。これは p の fontsize に（関数でない）定数の labelmargin をかけて算出される値で、定数の labelmargin のデフォルト値は 0.3 です。したがって、12pt のフォントの場合 labelmargin(p) は 3.6pt です。

<sup>6</sup>align 値に 45° の倍数以外の向きを指定した場合、貼り付け先にラベルを囲う矩形のどの位置が対応するのかわかりにくいですが、試してみた結果を直感でいうと、「45° の倍数」のときを基本として補間されている感じです。たとえば align 値に dir(120) を指定すれば、dir(90) のときの矩形下辺中央と dir(135) のときの矩形右下の間の適当な位置が配置されます。

<sup>7</sup>上・下・左・右はそれぞれ up, down, left, right でも表せますが、16 方位の方が簡潔でラベルの位置指定には向いているでしょう。

<sup>8</sup>あと、Asymptote はラベルの文字列の処理に  $\TeX$  を呼び出すのですが、「標準で存在しないフォントサイズを指定」するには  $\LaTeX$  のパッケージファイル fix-cm.sty も必要です。まあ、 $\TeX$  をフルインストールしたなら入っているでしょうが。ちなみに、単に大きい文字が使いたいだけなら「大きいフォント」ではなくラベル自体を拡大する方が楽です。したがって、フォントサイズの設定はあまり意味はないかも知れません。ラベルの拡大は次回説明する予定です。

<sup>9</sup>このとき、フォントサイズの指定は意味がありません。

<sup>10</sup>なお、label の pen 値引数を省略し、FillDraw を指定すると、ラベルの文字と塗りつぶしの色が同じなので、ラベルが見えなくなります。