

X_Y-pic

1 はじめに

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ の CD 環境で簡単な可換図式が描けますが、矢印の形状が限られている上、斜めの矢印が描けないなど種々の制約があります。X_Y-pic はこのような制約が無く、各種の図式を描くことができます。

- 任意の 2 点間に矢印が引け、矢先、軸、矢羽を組み合わせて種々の形状が可能で、軸の部分も自由に曲げることができる。
- 可換図式、フローチャート、有向グラフやツリーグラフ、カテゴリー理論のセル 2 つの図式、多角形や楕円などの図形や自由曲線、格子構造の図式、結び目やリンク、などを描くための機能が備わっている。
- 矢印や曲線などは独自のフォントによって (specials を用いない) 標準の $\mathcal{T}\mathcal{E}\mathcal{X}$ の機能内で動作する。一方、出力デバイスに依存するが、tpic specials や PostScript を使った軽く綺麗な出力にも対応し、タイプセットは (綺麗さを除いて) 両者で同一である。これを利用し、拡大してもギザギザが出ない PDF に変換できる。
- 座標の原点の移動のみならず、 X -軸、 Y -軸の単位ベクトルを任意に指定できるので、斜交座標の採用や図形のアフィン変換が容易。
- $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, plain $\mathcal{T}\mathcal{E}\mathcal{X}$, $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ など、種々の $\mathcal{T}\mathcal{E}\mathcal{X}$ の下で同様に動く。

X_Y-pic はいくつかのファイル群から成り立っており、必要なもののみ読み込むことができますが、当面は

```
\usepackage{all}{xy}
```

を $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ の冒頭の `\usepackage{}` の最後に追加しておくのがよいでしょう。以下で扱う $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ や $\mathcal{p}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ の場合、X_Y-pic は、本文の中で

```
\begin{xy}
...
\end{xy}
```

として使うのが基本です。

なお、ここをクリックして得たパッケージを `.\texmf` の存在するディレクトリで展開すれば X_Y-pic がインストールされます。

2 矢印

1. デフォルトの座標単位は mm で，(0,0) から (20,0) に矢印を引くには

```
\begin{xy}
  \ar (20,0)
\end{xy}
```



とします。(20,0) の代わりに実寸を用いて `<2cm,0cm>` のように書くこともできます (TEX で有効な長さを用いる)。

実際には「続けて`\begin{xy} \ar (20,0)\end{xy}`と書く」と、以下のようになります。

「続けて \longrightarrow と書く」

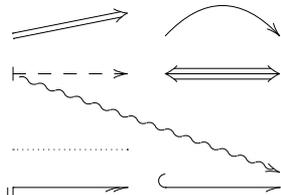
より短く「続けて`\xybox{\ar(20,0)}`と書く」としても同じ結果です。

2. 一般に

`{(a,b) \ar @{arrow} (c,d)}` または `{\ar @{arrow} (a,b);(c,d)}`

によって，(a,b) から (c,d) へ *arrow* で形状を指定した矢印が引かれます (形状指定がないときは，上の標準の矢印)：

```
\begin{xy}
  {(0,20) \ar @{=>} (15,23)},
  {(0,15) \ar @{|-->} (15,15)},
  {(0,15) \ar @{~>} (35,5)},
  {(0,5) \ar @{.} (15,5)},
  {(0,0) \ar @_{| | ->>} (10,0)},
  {(20,20) \ar @/^4mm/ (35,20)},
  {(20,15) \ar @3<->} (35,15)},
  {(20,0) \ar @^{({}->)} (35,0)}
\end{xy}
```



矢印の進む向きに対し，`^` は左直角方向，`_` は右直角方向を表します (左から右へ向かう矢印に対しては，前者は「上」，後者は「下」方向)。

一般に `/ /` はずらしを表し，`@/^4mm/` は「左直角方向に 4mm のずらし」となります。矢印の始点と終点は固定されたままで，上の例では 4mm 上に曲がった曲線の軸を持つ標準の形状の矢印が引かれます。

矢先と矢羽には `< > () | ' ' + / x o {*}` などが使えます。

矢軸には `- . ~ = :` が使えます。

`@` の直後に `^ _ 2 3` (左 / 右半，二 / 三重) のいずれかを付加できます。形状の *arrow* における上の例の `^{({}->)}` は「矢羽 (の左半分)」を表します。

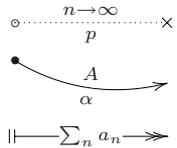
3. 矢印のフォントは Computer Modern や Euler に変更できます:

```
\begin{xy} \ar (0,10);(10,10) \longrightarrow
\SelectTips{cm}{} \ar (0,5);(10,5) \longrightarrow
\SelectTips{eu}{} \ar (10,0) \end{xy}
```

後の { } の中で, ポイント (10/11/12) が指定できます.

4. 矢印には, 以下のように ^ _ | (挿入) によってラベルを付けられます:

```
\begin{xy}
{(0,20) \ar @{{o.x}}(20,20)^{\n\to\infty}_p},
{(0,15) \ar @/_/@{[*]->} (20,10)^A_\alpha},
{(0,0) \ar @{|->>} (20,0)|{\sum_n a_n}}
\end{xy}
```

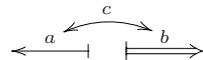


5. 位置やラベルに={id} によって名前をつけると, 後から参照できます:

```
\begin{xy}
(5,8.66)="A", (0,0)="B", (10,0)="C",
\ar@{-}"A";"B" \ar@{-}"B";"C" \ar@{-}"C";"A"
\end{xy}
```



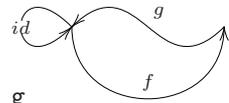
```
\begin{xy}
\ar @{<-|} (0,0);(10,0)^a="A"
\ar @{|=>} (15,0);(25,0)^b="B"
\ar @/^/ @{<->} "A";"B"~c
\end{xy}
```



6. 矢 (羽) の出発方向と, 終点からみた矢先の到着方向とは, この順に @ (方向, 方向)

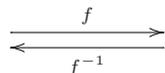
で指定できます. 方向には r (右), rd (右下), d (下), ld (左下), l (左), lu (左上), u (上), ru (右上) があります.

```
\begin{xy}
(0,0)="A", (20,0)="B",
\ar @{lu,ld} "A";"A" | {id}
\ar @{d,d} "A";"B" ~ f \ar @{ld,ru} "B";"A" ~ g
\end{xy}
```



7. 矢印は, 指定した長さ (左方向が正) だけの平行のずらしができます: @ <長さ >

```
\begin{xy}
(0,0)="A", (20,0)="B",
\ar @<1mm> "A";"B" ~ {f}
\ar @<1mm> "B";"A" ~ {f^{-1}}
\end{xy}
```



8. ラベルの直前にラベルの場所指定を置くことができます .

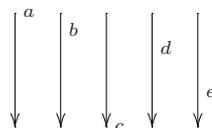
`< > -` : 矢印の末尾/先頭/中央 (`-` は `<>(0.5)` と同等)

(比) : 比で指定

/長さ/ : ずらし

!{位置} : 位置で指定

```
\begin{xy} (0,15)="A", (0,0)="B",
\ar "A";"B"<a
\ar@<6mm> "A";"B"<<<b
\ar@<12mm> "A";"B">c
\ar@<18mm> "A";"B"^(.3)d
\ar@<24mm> "A";"B"~/3mm/e \end{xy}
```

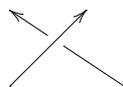


9. “A” と “B” を結ぶ直線との交点の位置は

!{"A";"B"}

によって得られます .

```
\begin{xy}
(0,0)="A", (10,10)="B", (15,0)="C", (0,10)="D",
\ar "A";"B"
\ar "C";"D"|!{"A";"B"}\hole
\end{xy}
```



“C” から “D” への矢印に対し，“A” と “B” を結ぶ直線との交点に空白の穴 `\hole` を空けたものです .

一般に “A” と “B” を結ぶ直線と “C” と “D” を結ぶ直線の交点は

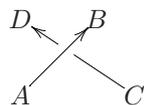
!{"A";"B":"C";"D",x}

によって求められます .

3 要素

1. 図式に要素を配置し，それを矢印で結ぶことができます . 要素は，通常の記号や文字，さらに一般に $\text{T}_{\text{E}}\text{X}$ で表現されたものとしてすることができます .

```
\begin{xy}
(0,0) *{A}="A", (10,10)*{B}="B",
(15,0)*{C}="C", (0,10) *{D}="D",
\ar "A";"B"
\ar "C";"D"|!{"A";"B"}\hole
\end{xy}
```



要素を配置するには，先頭に `*` をつけて `*{要素}` のように書きます .

次のようなものが要素として、先頭に * を付加することで配置されます。

{テキスト} :テキストは数式モードで組まれます。

TeX における通常の \hbox のように組まれた箱でも OK です。

\txt<幅>{テキスト} :テキストは通常の記事として組まれます。

<幅> は省略可能で、テキスト中は強制改行記号 \\ が使用できます。

\xybox{Xy-pic による図式}

これらの要素は、ラベルとしても使用できます。

```
\begin{xy}
\ar (0,0) *{\sum_{n=1}^N 2^{-n}};
(30,0)
*\txt{増加して\\1 に収束}
~{\N\to\infty}
\end{xy}
```

$$\sum_{n=1}^N 2^{-n} \xrightarrow{N \rightarrow \infty} 1 \text{ 増加して } 1 \text{ に収束}$$

2. 要素は大きさや位置が調整でき、枠を付けることなどもできます。

add-op 長さ :大きさの調整。add-op には、+ - = += -= がある。

長さを指定しない場合は、デフォルト値が用いられる。

!ベクトル :位置の調整 (境界までのベクトル R RD D ... など可)

[形] :様々な形 ([o] は丸に) を設定可能で、以下の枠もそのひとつ。

[Fframe:opt] :: opt は省略可。F に (空白無しで続けた) frame は

. - = : 矩形の枠、opt は<長さ>で角の丸みの半径

-- o- : 破線の枠、丸みを帯びた枠

, -, : 影付きの枠 (opt は<長さ>で影のサイズ)

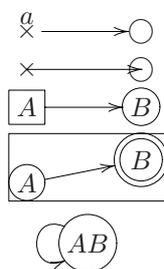
o .o -o oo : 円形の枠、opt は<長さ>で半径を指定

() ^) _ (: (の形の括弧 (左/右/上/下)

\{ \} ^{ } _{ } : { の形の括弧 (左/右/上/下)

枠は ***\frm opt{frame}** または、****\frm opt{frame}** によっても描ける。前者は直前の要素を囲み、後者はその前の要素も囲むものとなる。

```
\begin{xy}
{\ar(0,20) *{\times}**!D{a};(15,20) *{\bigcirc}
\ar(0,15) *=0{\times};(15,15) *=0{\bigcirc}
\ar(0,10) **[F]{A}; (15,10) **[Fo]{B}
\ar *++={A} *\frm{o};
(15,3) *+++={B} *\frm{oo} **\frm{-}},
(8,-8) **[Fo]{AB} ="A",
\ar @(\lu,ld) "A";"A"
\end{xy}
```



3. 行列形式の図式 .

要素を，直接座標を指定せずに行列の成分として配置します .

座標を考慮する必要がなく，可換図式のような図式を描くのには便利です .

列は `&` で，行は `\\` で区切り，以下のように書きます .

```
\xymatrix{
  成分 & 成分 & \cdots & \\
  成分 & 成分 & \cdots & \\
  \cdots & & & }

```

各成分には，配置された要素と，そこを始点とする矢印の情報が有り

- ・ 要素がテキストの場合，`*` は不要 . 先頭が `\` でないなら `{ }` も不要 .
- ・ 矢印の行き先は，成分の番号で指定できる :

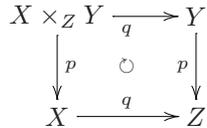
`[hop]` : `hop` は，`r l u d` (右/左/上/下) をいくつか並べたもの

たとえば `[rrd]` は，その場所から 2 つ右で一つ下の成分を意味する .

"行，列" : 例えば `"1,1"` は左上の (1,1) 成分

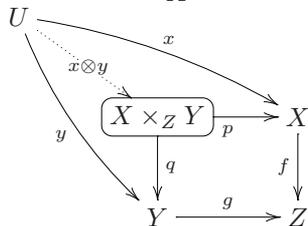
```
\xymatrix{ \ar@{}[rd] | {\circlearrowright}
  X \times_Z Y \ar[r]_q \ar[d]^p & Y \ar[d]_q \\
  X \ar[r]^q & Z }

```



```
\xymatrix{
  U \ar@/_/[ddr]_y \ar@{.}/[dr]
    | {x \otimes y} \ar@/^/[drr]^x \\
  & *+[F-: <3pt>]{X \times_Z Y} \\
  & \ar[d]^q \ar[r]_p & X \ar[d]_f \\
  & & Y \ar[r]_g & Z }

```



4 いろいろな図式

1. 曲がった軸をもつ矢印

円弧と線分を組み合わせると，迂回する矢印を描くことができる . 経路は

・ **目標** : 目標までの線分

・ **方向 目標** : 出発方向から **目標** に向かって $\frac{1}{4}$ 回転する .

最初の方向指定がなければ，前回の方向からの回転となる (次項も) .

・ **方向 回転の向き 方向 目標** : 最初の方向へ出発して直線的に移動し，後の方向で進むと **目標** に到達できる地点で必要な回転をする .

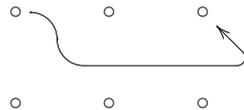
方向には `r r d d l d l l u u r u` がある .

回転の向き は，`^` (反時計回り) または `_` (時計回り)

最後は，同じ向きの矢印を引いて，最終 **目標** に到達させる .

円弧の半径は 10pt であるが / **長さ** により以降変更可能 .

```
\xymatrix{
  {\circ} \ar
    'r[d] ' [rr] '^lu/3pt[rr] [rr]
    &{\circ}&{\circ} \\
  {\circ}&{\circ}&{\circ}
}
```

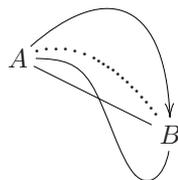


より一般に、始点と終点とを与えたとき、途中の制御点を指定した Bézier 曲線を使って自由曲線が描けます。

****\crv{ 制御点 & 制御点 & ... }**

- ・ 制御点がないときは、二点を結ぶ線分になります。
- ・ 矢印のときと同様 **~*{.}** などを入れて線種指定できます。

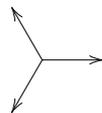
```
\begin{xy}
  (0,10)*[o]+{A};(20,0)*[o]+{B}="B"
  **\crv{}
  **\crv{~*{.}}(10,15)
  **\crv{(10,20)&(20,20)} ?>*\dir{>}, "B"
  **\crv{(5,10)&(10,10)&(15,-10)&(20,-5)}
\end{xy}
```



2. 座標軸の回転

ベクトル: X 軸の単位ベクトルを定める (回転 + スケール変換を行う)。

```
\begin{xy} {\var (8,0)},
  (-.5,\halfrootthree): {\var(0,0);(8,0)},
  (0,0),(-.5,\halfrootthree): \var(0,0);(8,0)
\end{xy}
```



次に、`\xymatrix` を使った $\frac{\pi}{4}$ の整数倍の回転の例を挙げます。

`\xymatrix` の直後の指定で

@方向 : 方向に向けた回転をする (@r:回転なし)

@add-op 長さ : 行と列の空きの設定 (@!:行と列の空きを等しくする)

@Cadd-op 長さ : 列の空き (@C!:列の空きを等しくする)

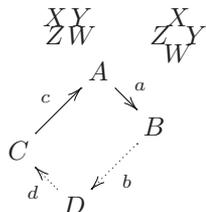
@Radd-op 長さ : 行の空き (@R!:行の空きを等しくする)

@Madd-op 長さ : 要素のデフォルトの空き

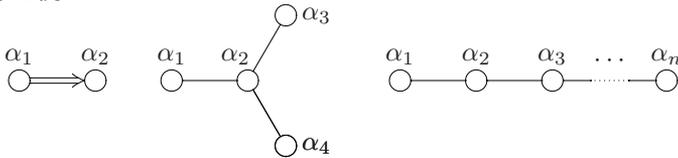
```
\xymatrix@OptM=Opt{X&Y\\Z&W}\qqquad
```

```
\xymatrix@dr@=OptM=Opt{X&Y\\Z&W}
```

```
\xymatrix@dr@C=4mm{
  A \ar[r]^a & B \ar@{.}[d]^b \\
  C \ar[u]^c & D \ar@{.}[l]^d }
```



3. 仕様と例



```
\begin{xy}
  \ar@{=>} *++!D{\alpha_1} *\cir<4pt>{};
  (10,0) *++!D{\alpha_2} *\cir<4pt>{}
  \ar@{-} (20,0) *++!D{\alpha_1} *\cir<4pt>{};
  (30,0) *++!D!R(0.4){\alpha_2} *\cir<4pt>{}="A",
  \ar@{-} "A";(35,8.61) *++!L{\alpha_3} *\cir<4pt>{}
  \ar@{-} "A";(35,-8.61) *++!L{\alpha_4} *\cir<4pt>{};
  \ar@{-} "A";(35,-8.61) *++!L{\alpha_4} *\cir<4pt>{};
  \ar@{-} (50,0) *++!D{\alpha_1} *\cir<4pt>{};
  (60,0) *++!D{\alpha_2} *\cir<4pt>{}="B"
  \ar@{-} "B";(70,0) *++!D{\alpha_3} *\cir<4pt>{}="C"
  \ar@{-} "C";(75,0) \ar@{.} (75,0);(80,0)~*!U{\cdots}
  \ar@{-} (80,0);(85,0) *++!D{\alpha_n} *\cir<4pt>{}
\end{xy}
```

では、以下の円弧を描くコマンドを使っています。

`\cir<半径><方向 回転の向き 方向>`

最初の方向で円周上を回転の向き（`^` は反時計回り、`_` は時計回り）にスタートし、後の方向まで円弧を描きます。`{ }` の中を空として、これらを指定しない場合は円を描きます。

```
\begin{xy}
  *\cir<5mm>{l_d}, (10,0) *\cir<5mm>{r^d}, (22,0) **{M} *\cir{dr_ur}
\end{xy}
```

\Xy-pic は \TeX のメモリーを大量に使うため、複雑な図式を扱うと `TeX capacity exceeded` のエラーが発生することがあります。それが解消できない場合、より軽い

```
\usepackage[all,ps,dvips]{xy}
```

として DVI ファイルを作成して下さい。その後 PS ファイルから EPS ファイル、あるいは PDF ファイルに変換 (`Ghostscript+dvipsk+ps2pdf` など) することができます。

これまで X_y-pic の基本的な機能をいくつか紹介してきましたが，その他にも，楕円や多角形，各種ダイアグラムや結び目などに関する多くの描画機能を持っています．より詳しい仕様やこれらの機能については，dviout に添付の Help，あるいはそこに挙げてある文献などを参照してください．