tmtmath.sty ファイル

準備と描画環境

tmtmath.sty ファイルは、LaTeX 2ε において次の 2 つの作業を補助するためのマクロ集である。

1. 関数 $y = f(x)$ のグラフを描画する
2. 平面図形を描画する

tmtmath.sty を利用するには、まず tmtmath.sty ファイルを、文書と同じフォルダか所定のフォルダに入れる。文書と同じフォルダに入れれば手軽だが、文書がそれぞれのフォルダに拡散したときに、フォルダごとに tmtmath.sty ファイルが必要になってしまう。できれば所定のフォルダに置いて管理するのがよいだろう。所定のフォルダは、Windows、Macintosh、TEX のバージョンなどで多少の違いがあるようなので、適切な場所に入れてもらいたい[1]。

tmtmath.sty を有効にするには、文書のプリアンブルに

\usepackage{tmtmath}

を宣言しておく。また、媒介変数で表された関数や単位円の描画の際に、pict2e パッケージが必要になる場合がある。そのときのために、文書のプリアンブルにもう 1 行

\usepackage[ドライバ名]{pict2e}

も宣言しておくとよい。ドライバ名は使用環境によって違うようなので、適切なドライバ名を入れてほしい。以上で、tmtmath.sty を使う準備が整った。

関数グラフや図形の描画は picture 環境の中で行うが、ここでは、picture 環境を少しいじった drawpict 環境を設定して、その中に記述することを前提に話を進めることにする。

[drawpict 環境]

```
\begin{drawpict}[#1](#2,#3)(#4,#5)
    ここに各種マクロ、命令を記述する
\end{drawpict}
```

#1 は描画単位を必ず指定する。0.5cm, 2pt などすべての単位が使用できる。

横#2、縦#3 の描画環境が設定される。(#4,#5) は左下隅の座標を指定する。描画環境は用紙幅の中央に置かれる。

[1]詳しくは Internet や TEX の書籍で調べてもらいたい。

drawpict 環境を使って

```
\begin{drawpict}[.35cm](29, 12)(-6, -6)
    あれこれ
\end{drawpict}
```

と書くことは、標準的な TEX において

```
\begin{center}
  \unitlength=.35cm
  \begin{picture}(29, 12)(-6, -6)
    あれこれ
  \end{picture}
\end{center}
```

と書くことと同じである。記述量の面から見れば大差ないので、普通に picture 環境を使ってもかまわない。

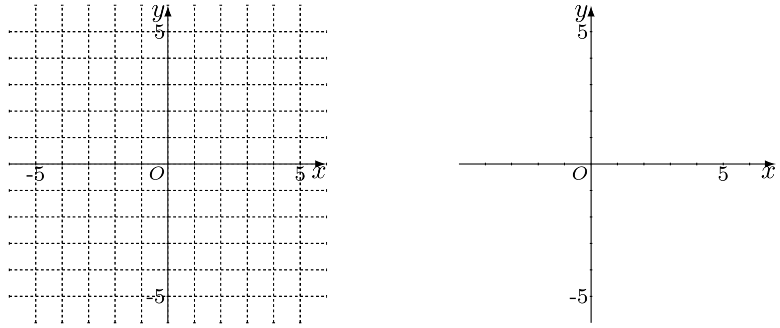
座標平面などの描画

\baseskip#1 **\baseskip{*move*}**

原点を、描画単位の#1 ぶんだけ右へ移動させる。負の値を指定すると左への移動となる。1 つの環境の中に、複数の図を描くときに利用する。また、drawpict 環境では描画は常に中央に置かれるので、左右に寄せるときにも利用する。

▽ 1 つの環境に 2 つの座標軸を描画

```
\begin{drawpict}[.35cm](29, 12)(-6, -6)
  \coordinate[Rg](-6, 6)(-6, 6)
  %
  \baseskip{16} % 原点を 16 移動（この場合は左右の座標系の間が 5 空く）
  %
  \coordinate[R](-5, 7)(-6, 6)
\end{drawpict}
```



原点を 16 単位移動させることは、2 つの図形の原点間の距離が 16 単位になることと同じである。この例では、左図の原点の左に 6 単位、右図の原点の右に 7 単位の図が描かれるので、図の描画幅は $6 + 16 + 7 = 29$ 単位となる。したがって、drawpict 環境の横幅は 29 に設定した。

\coordinate[#1](#2,#3)(#4,#5) **\coordinate**[ax](x_l, x_r)(y_b, y_t)

`#1` は軸指定。

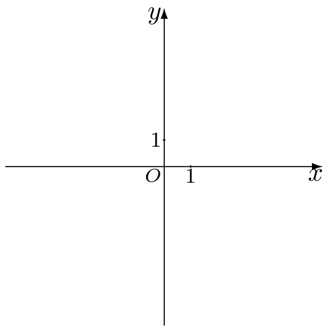
`#1` なし	→	横軸、縦軸の描画（規定値）
A	→	規定値に軸名称を追加（目盛なし）
R | I | d | r	→	実数 | 虚数 | 度 | ラジアン の軸名称と目盛を規定値に付加
g	→	グリッド線を描画（単独もしくは他の軸指定と一緒に使う）

`#2,#3` 間の横軸、`#4,#5` 間の縦軸をもつ座標平面を描画する。**`#2`〜`#5` は整数値で指定すること。** 座標平面の縁にはグリッド線は描画されない。picture サイズと左下隅の座標は`#2`〜`#5` の値を参考にして、`\begin{drawpict}[#1](#3-#2, #5-#4)(#2, #4)` 程度に指定するとよい。

`#2`〜`#5` で指定した値は、実数・虚数目盛ではそのままの値で使用されるが、度・ラジアン目盛では、θ 軸においては 30°・$\frac{\pi}{6}$ 刻み、y 軸においては 0.5 刻み として使用される。

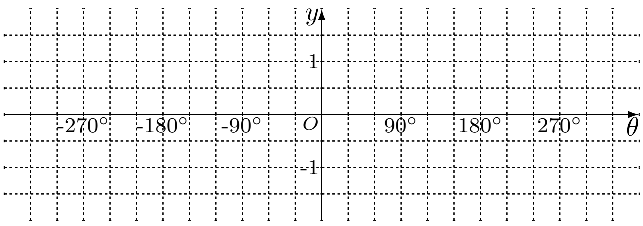
▽ x-y 実数平面 （目盛なし）を $-6 < x < 6$、$-6 < y < 6$ の範囲で描画

```
\begin{drawpict}[.35cm](12, 12)(-6, -6)
  \coordinate[A](-6, 6)(-6, 6)
  \putones % x,y 軸に目盛 1 を表示するマクロ
\end{drawpict}
```



▽ θ-y 座標平面 （目盛、グリッドあり）を $-360^\circ < \theta < 360^\circ$、$-2 < y < 2$ の範囲で描画

```
\begin{drawpict}[.35cm](24, 8)(-12, -4)
  \coordinate[dg](-12, 12)(-4, 4) % 指定範囲注意!
\end{drawpict}
```



θ 軸は 30° 刻みだから $-360^\circ < \theta < 360^\circ$ の範囲は $-12 < \theta < 12$ に、y 軸は 0.5 刻みだから $-2 < y < 2$ の範囲は $-4 < y < 4$ に、それぞれ直してから指定する。

$\boxed{\textbf{\textbackslash locate[\#1](\#2,\#3)}}$ \locate[bwBW/x](x, y)

#1 は必ず指定する。

b | w | B | W → 黒丸（小）｜白丸（小）｜黒丸（大）｜白丸（大）

x → 交差破線を引く（単独もしくは他の表示丸と一緒に使う）

座標 (#2, #3) の位置を黒丸か白丸で示す。

$\boxed{\textbf{\textbackslash linexis\#1\#2(\#3,\#4)}}$ \linexis{a}{op}(y_b, y_t)

#2 はオプション。

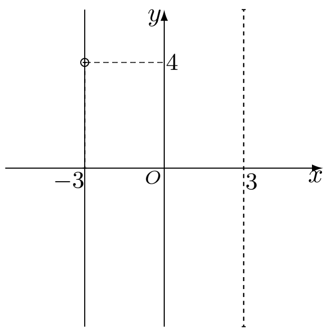
#2 なし → 直線を実線で描画

: → 直線を破線で描画

#1 $= a$ として、直線 $x = a$ のグラフを、y の区間 (#3, #4) で描画する。

▽ $x = -3$ と $x = 3$（破線）を $-6 < x < 6$、$-6 < y < 6$ の範囲の x-y 実数平面に描画

```
\begin{drawpict}[.35cm](12, 12)(-6, -6)
  \coordinate[A](-6, 6)(-6, 6)
  \linexis{-3}(-6, 6)
  \linexis3:(-6, 6)
  \locate[Wx](-3, 4)
  \apeeex(-3.6, -.5){$-3$}(3.3, -.5){$3$}(.3, 4){$4$} % マクロ\apeeex を使用
\end{drawpict}
```



黒丸、白丸は基本的に小さいものを使うことを想定している。丸は`\circle`で描いているので、図の縮尺に応じて丸の大きさが変わる。そのため、図が小さくなると丸も小さくなり判別に苦慮することがある。大きい丸はそのために用意したものであり、例は図が小さめなので大きい丸を使っている。丸の大きさ—黒丸の方が存在感が大きいので、白丸より若干小さい—が気に入らなければ `tmtmath.sty` ファイルを変更してもらいたい。

関数 $y = f(x)$ のグラフの描画

$y = f(x)$ のグラフは`\easyfx[関数式](..)` の "関数式" を定義することで描画される。ただし、`\easyfx` は "関数グラフ自動作成マクロ" ではなく、あらかじめグラフの形状や定義域・値域が分かっているグラフを、適切に描画するためのものである。したがって、定義域・値域を事前に把握、もしくは計算した上で利用しないと正しい描画ができない。基本的に、(横軸 10〜20) × (縦軸 10〜20) 程度の幅でグラフを描き、それが数 cm〜10cm 程度四方の大きさに収まることを想定している。描画は drawpict 環境で行っているが、標準の picure 環境で行っても何ら問題ない。

また、マクロが正常に機能しないときは、変数が他のマクロのものと衝突している可能性がある。適切な変更をしてから利用してもらいたい。

$\boxed{\text{\easyfx[\#1]\#2(\#3,\#4)}}$ \easyfx[**関数式**]$\{op\}(x_l,\ x_r)$

#2 はオプション。

#2 なし	→	関数のグラフを実線で描画
:	→	関数のグラフを破線で描画

逆ポーランド記法で記述した#1 のグラフを区間 (#3, #4) で描画する。#1 に記述できる文字は、下表の演算子および変数・数値だけである。これ以外の文字は受け付けないが、たとえば x2+は英字空白を入れて `x 2 +`のように書くことはできる。また、小数や負の数など、数値が 1 桁でなければ必ず {10} のように中括弧で囲む。見やすさのために 1 桁の数値を中括弧で囲むことは問題ない。

[**二項演算子（二項後オペレーター）**]

演算子	機能	記述例	通常の記述
+	和 A+B を計算する	x2+	$x+2$
-	差 A-B を計算する	x{10}-	$x-10$
*	積 A*B を計算する	xx*	x^2
/	商 A/B を計算する	1x/	$\frac{1}{x}$

[**単項演算子（後置オペレーター）**]

演算子	機能	記述例	通常の記述
r	$\sqrt{\mathtt{A}}$ を計算する	x2+r	$\sqrt{x+2}$
q	$\sqrt[3]{\mathtt{A}}$ を計算する	xq	$\sqrt[3]{x}$
e	$e^{\mathtt{A}}$ を計算する	{-1}x*e	e^{-x}
l	$\log_e \mathtt{A}$ を計算する	xl	$\log_e x$ または $\ln x$
s	$\sin \mathtt{A}$ を計算する	{3.14}s	$\sin \pi$
S	$\sin \mathtt{A}$ を計算する	xSx/	$\frac{\sin x}{x}$
c	$\cos \mathtt{A}$ を計算する	xc	$\cos x$
t	$\tan \mathtt{A}$ を計算する	x2/t	$\tan \frac{x}{2}$
|	|A|を計算する	x2+|	$\lvert x+2 \rvert$

[**変数と数値（オペランド）**]

変数・数値	機能	記述例	通常の記述
x	変数	x, {-1}x*	$x,\ -x$
数値	数値	5, {-2}, {0.333}	$5,\ -2,\ \frac{1}{3}$

■逆ポーランド記法について

逆ポーランド記法は、通常 2 項間に置く演算子を 2 項の直後に書く記法である。逆ポーランド記法は、コンピュータの計算処理において効率的なため、\easyfx に与える関数の記述として採用している。はじめは混乱するだろうが、すぐに慣れると思われる。

通常の式を逆ポーランド記法の式にするには、x に値を代入して計算するときと同じ順に、逐次逆ポーランド記法の式に直していくとよい。直したところが 1 つの項になる。また、逆ポーランド記法の式を通常の式にするには、逆ポーランド記法の式を左から順に見て、演算子を見つけたら逐次直前の 2 項または 1 項を、通常の式に直していくとよい。直したところが 1 つの項になる。式が長いときはそれを繰り返せば、最終的にひとかたまりの逆ポーランド記法による式ができる。

たとえば $x+2$ を逆ポーランド記法にすると

$$x+2 \quad \Rightarrow \quad \texttt{x2+}$$

である。$x-10$ は本来なら $x\,10\,-$ と記述するのだが、\easyfx に 10 が 1 つの項であることを知らせるため、中括弧を用いて

$$x-10 \quad \Rightarrow \quad \texttt{x\{10\}-}$$

とする。関数式には x^2 など、指数が使われることが多いが、\easyfx には指数を計算する演算子はないので

$$x^2 \quad \Rightarrow \quad x\times x \quad \Rightarrow \quad \texttt{xx*}$$

と書くことになる。分数は割り算に見直すことで、$\dfrac{1}{x}$ なら

$$\frac{1}{x} \quad \Rightarrow \quad 1\div x \quad \Rightarrow \quad \texttt{1x/}$$

となる。また、$\dfrac{1}{x-5}$ ならば

$$\frac{1}{x-5} \quad \Rightarrow \quad 1\div(x-5) \quad \Rightarrow \quad 1\div\texttt{(x5-)} \quad \Rightarrow \quad \texttt{1(x5-)/} \quad \Rightarrow \quad \texttt{1x5-/}$$

とするとよい。( ) は 1 つの項であることを明確にするために書いただけなので、本来は一切不要である。実際、逆ポーランド記法では ( ) を使うことはないし、( ) があると\easyfx は処理できない。

\easyfx では、三角関数をはじめとする様々な記号も演算子として扱っている。ただし、これらの記号は 1 項に対する演算を行うものであるから、1 項の直後に記述する。たとえば $\sin x$ は

$$\sin x \quad \Rightarrow \quad \texttt{xs}$$

である。他の記号も同様に記述するので、$\sqrt{x}$、e^x、$\log_e x$、$|x|$ はそれぞれ

xr、xe、xl、x|

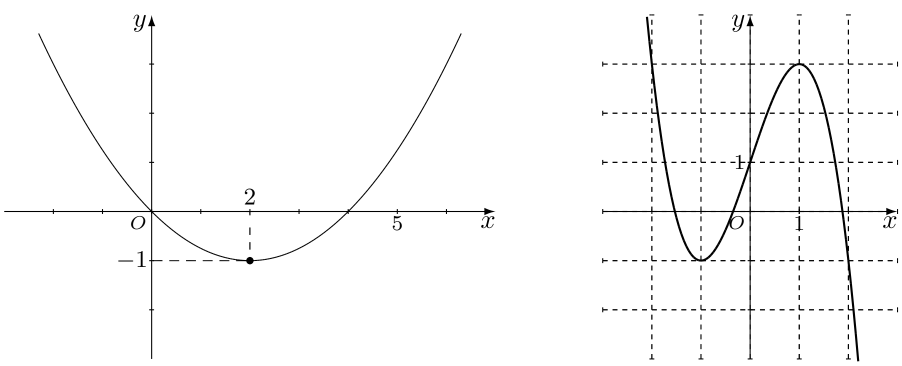
と書く。フォントによっては 1（いち）、l（エル）、|（縦線）は区別しにくいので注意したい。

では、関数式の記述例を示して、\easyfx に与えて処理した結果を示すことにする。[コメント] を参考にしながら、実際に通常の式を逆ポーランド記法の式にしてみれば、思ったほど難しくないことが分かるはずだ。

$\triangledown\ y=\dfrac{1}{4}x^2-x$ と $y=-x^3+3x+1$

$\dfrac{1}{4}x^2-x$	$-x^3+3x+1$
[×÷ の式にする]	[×÷ の式にする]
$0.25\times x\times x-x$	$(-1)\times x\times x\times x+3\times x+1$
[$A\times B\to$ `AB*` より]	[$A\times B\to$ `AB*` より]
`({0.25}x*)` $\times x-x$	`({-1}x*)` $\times x\times x+3\times x+1$
[$A\times B\to$ `AB*` より]	[$A\times B\to$ `AB*` より]
`({.25}x*x*)` $-x$	`({-1}x*x*)` $\times x+3\times x+1$
[$A-B\to$ `AB-` より]	[$A\times B\to$ `AB*` より]
`{.25}x*x*x-`	`({-1}x*x*x*)` $+$ `(3x*)` $+1$
	[$A+B\to$ `AB+` より]
	`({-1}x*x*x*3x*+)` $+1$
	[$A+B\to$ `AB+` より]
	`{-1}x*x*x*3x*+1+`

```
\begin{drawpict}[.65cm](18, 7)(-3, -3)
  \coordinate[R](-3, 7)(-3, 4) % 左のグラフ
  \easyfx[{.25}x*x*x-](-2.3, 6.3)
  \locate[bx](2, -1)
  \apeex(2, .3){$2$}(-.4, -1){$-1$}
  %
  \baseskip{12}
  \coordinate[Rg](-3, 3)(-3, 4) % 右のグラフ
  \putones
  {\thicklines \easyfx[{-1}x*x*x*3x*+1+](-2.1, 2.2)}
\end{drawpict}
```



1/4 を `14/`と記述すると余分な計算をするので、計算式の中では小数値`{.25}`を用いた。

$-x^3$ は左側から順に逆ポーランド記法に直した。右側から順に直せば`{-1}xxx***`となるはずだが、これだと `xxxxxxx******`以上になると処理できない。なぜなら、逆ポーランド記法用にスタックを 6 個しか用意していないからである。スタックを増やすなら `tmtmath.sty` ファイルにおいて、コメントアウトになっている `(if in need)` の 4 箇所を書き換えればよい。

$\triangledown\ y = \dfrac{x+2}{x-2}$ と $y = 2\sqrt{|x-1|}$

$$\frac{x+2}{x-2}$$

[×÷ の式にする]

$$(x+2) \div (x-2)$$

[$A + B \to$ `AB+` より]

$$\texttt{(x2+)} \div (x-2)$$

[$A - B \to$ `AB-` より]

$$\texttt{(x2+)} \div \texttt{(x2-)}$$

[$A \div B \to$ `AB/` より]

`2x+x2-/`

$$2\sqrt{|x-1|}$$

[×÷ の式にする]

$$2 \times \sqrt{|x-1|}$$

[$A - B \to$ `AB-` より]

$$2 \times \sqrt{|\texttt{x1-}|}$$

[$|A| \to$ `A|` より]

$$2 \times \sqrt{\texttt{x1-|}}$$

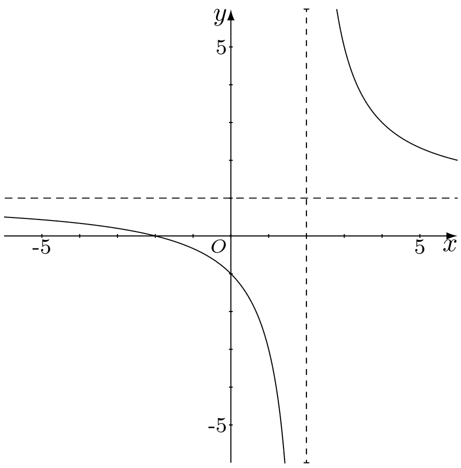
[$\sqrt{A} \to$ `Ar` より]

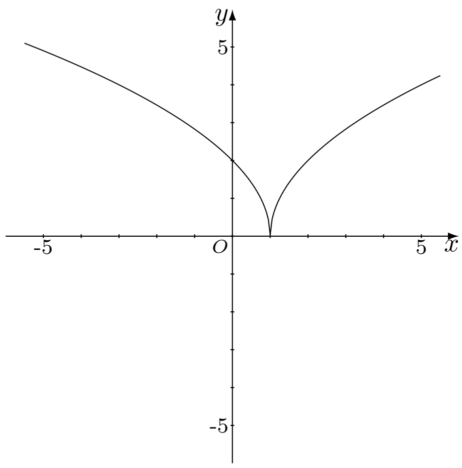
$$2 \times \texttt{(x1-|r)}$$

[$A \times B \to$ `AB*` より]

`2x1-|r*`

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6) % 左のグラフ
  \easyfx[2x+x2-/](-6, 1.43) % (x<2)
  \easyfx[2x+x2-/](2.8, 6) % (x>2)
  \linexis2:(-6, 6) % 漸近線 x=2
  \easyfx[10+]:(-6, 6) % 漸近線 y=1
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[2x1-|r*](-5.5, 5.5)
\end{drawpict}
```





$y = \dfrac{x+2}{x-2} = \dfrac{4}{x-2} + 1$ であるから、漸近線は $x = 2$ と $y = 1$ である。関数 $x =$ (数値) は `\easyfx` で描けないので、`\linexis(数値)` を用いている。関数 $y =$ (数値) は `\easyfx` で描けるが、`\easyfx[数値](..)` と記述したのではだめである。なぜなら、`\easyfx` は演算子を見つけて

計算する仕組みなので、数値だけを単独で使うことはできない。具体的には、`\easyfx[1](..)` は計算式に演算子を含んでいないので、$y=1$ はたとえば $y=1+0$ と見て、`\easyfx[10+](..)` と記述する。

このことから、x だけを単独で使えないことも分かる。関数 $y=x$ は`\easyfx[x](..)` ではなく、たとえば $y=1x$ と見て、`\easyfx[1x*](..)` とする。

$y=\dfrac{x+2}{x-2}$ は $x=2$ のとき分母が 0 になってしまう。`\easyfx` は 0 の除算を避けるようになっていないので、0 を含む区間を指定できない。そのため関数を 2 分割しているのだが、その範囲は面倒でも自分で求めなければならない。いま描いている x-y 座標平面の値域は $-6<y<6$ であるから、

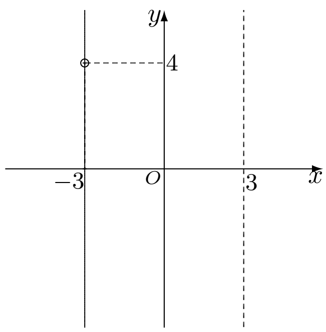
$$\frac{x+2}{x-2}=-6、\quad \frac{x+2}{x-2}=6$$

を実際に解いて、$x\approx 1.43$、$x=2.8$ を得た上で範囲に指定した。

$y=2\sqrt{|x-1|}$ は根号内に絶対値があるので問題ないが、かりに $y=2\sqrt{x-1}$ であれば $x<1$ で根号内が負の値になってしまう。`\easyfx` は根号内の負の値を避けるようになっていないので、根号内が負になる区間を指定できない。そのような場合も、正しい描画範囲を自ら指定しなければならない。

ところで、関数 $x=$ (数値) は`\easyfx` で描けないと言ったが、p.21 の drawfinv 環境を使って

```
\begin{drawpict}[.35cm](12, 12)(-6, -6)
  \coordinate[A](-6, 6)(-6, 6)
  \begin{drawfinv}(-6, 6) % x=-3
    \def\calcdimen{\RPNfx[{-3}0+]}
  \end{drawfinv}
  \begin{drawfinv}:(-6, 6) % x=3
    \def\calcdimen{\RPNfx[30+]}
  \end{drawfinv}
  \locate[Wx](-3, 4)
  \apeeex(-3.6, -.5){$-3$}(3.3, -.5){$3$}(.3, 4){$4$} % マクロ\apeeex を使用
\end{drawpict}
```



のようにすれば描けないことはない。しかし、drawfinv 環境は 1 つの関数ごとに設定しなくてはならず、それなりに手間がかかる。ただし、破線で描いた $x=3$ のグラフをよく見ると、p.4 の出力と比較して端点の描画がわずかに違うことが分かるだろう。`\linexis` は`\dashbox` を用いて描いているからだが、この違いのために手間をかけるかどうかは微妙かもしれない。

$\bigtriangledown\ y=-2^x+1$ と $y=\log_{\frac{1}{2}}(-3x)+1$

$$-2^x+1 \Rightarrow -e^{\log_e 2^x}+1$$

[×÷ の式にする]

$$(-1)\times e^{x\times\log_e 2}+1$$

[$A\times B \to$ `AB*` より]

$$(-1)\times e^{\texttt{(x\{.693\}*)}}+1$$

[$e^A \to$ `Ae` より]

$$(-1)\times \texttt{(x\{.693\}*e)}+1$$

[$A\times B \to$ `AB*` より]

$$\texttt{(\{-1\}x\{.693\}*e*)}+1$$

[$A+B \to$ `AB+` より]

`{-1}x{.693}*e*1+`

$$\log_{\frac{1}{2}}(-3x)+1 \Rightarrow \frac{\log_e(-3x)}{\log_e\frac{1}{2}}+1$$

[×÷ の式にする]

$$\log_e((-3)\times x)\div(\log_e\tfrac{1}{2})+1$$

[$A\times B \to$ `AB*` より]

$$\log_e\texttt{(\{-3\}x*)}\div(-0.693)+1$$

[$\log_e A \to$ `Al` より]

$$\texttt{(\{-3\}x*l)}\div(-0.693)+1$$

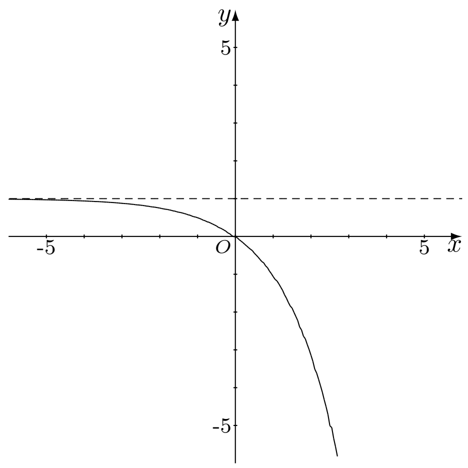
[$A\div B \to$ `AB/` より]

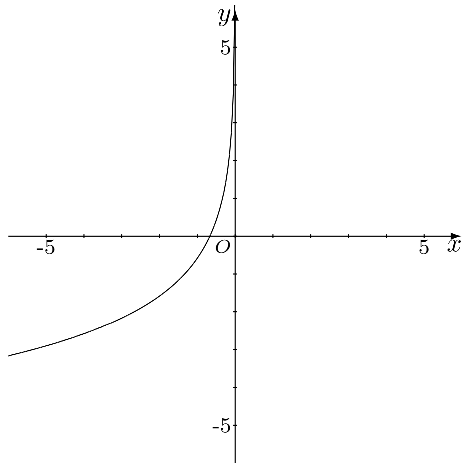
$$\texttt{(\{-3\}x*l\{-.693\}/)}+1$$

[$A+B \to$ `AB+` より]

`{-3}x*l{-.693}/1+`

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6) % 左のグラフ
  \easyfx[{-1}x{.693}*e*1+](-6, 2.7)
  \easyfx[10+]:(-6, 6) % 漸近線 y=1
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[{-3}x*l{-.693}/1+](-6, -.01)
\end{drawpict}
```





$y=-2^x+1$ のグラフであるが、`\easyfx` には e^x を計算する演算子はあっても 2^x を計算する演算子はない。しかし、指数・対数には

$$\boxed{A=e^{\log_e A}}$$

という性質がある。これより

$$2^x=e^{\log_e 2^x}=e^{x\log_e 2}$$

とできるから、

$$-2^x+1=-e^{x\log_e 2}+1\approx -e^{x\times(0.693)}+1$$

を用いて、逆ポーランド記法の計算式に直した。

同様に、$y=\log_{\frac{1}{2}}(-3x)+1$ のグラフであるが、`\easyfx` には $\log_e$ を計算する演算子はあっても $\log_{\frac{1}{2}}$ を計算する演算子はない。しかし、対数には

$$\boxed{\log_A B=\frac{\log_e B}{\log_e A}}$$

という性質がある。これより

$$\log_{\frac{1}{2}}(-3x)+1=\frac{\log_e(-3x)}{\log_e\frac{1}{2}}+1\approx\frac{\log_e(-3x)}{(-0.693)}+1$$

を用いて、逆ポーランド記法の計算式に直した。

指数関数も対数関数も、描画範囲は $-6<x<6$、$-6<y<6$ のすべてではない。そこで指数関数も対数関数も

$$-2^x+1=-6、\quad \log_{\frac{1}{2}}(-3x)+1=6$$

を実際に解いて、$x\approx 2.8$、$x\approx -0.01$ を得た上で範囲に指定した。

ここで、`\easyfx` の計算の仕方について少し説明をしておこう。四則計算のうち、和・差・積を求めるときは TEX の寸法計算を使っているため結構正確なはずである。しかし、除算は繰り返しの引き算で行っているため精度がやや落ちる。指数計算は、近似式

$$e^x\approx\left(\frac{x+512}{x-512}\right)^{256}$$

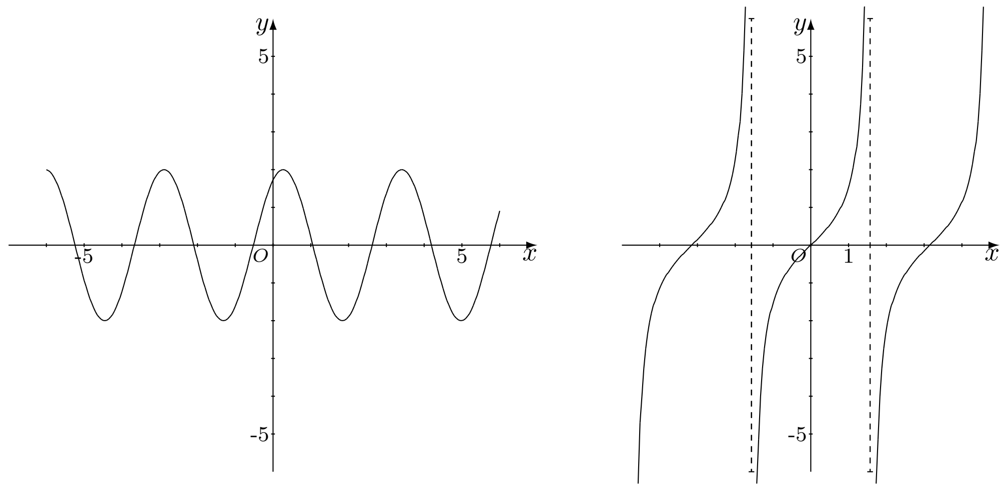
を用いている。x がさほど大きくなければ、まずまずの値が得られる。対数関数は、近似式では精度があまりよくないので、1 から 999 まで 1 刻みにした、有効桁数 4 桁の数表から値を読み取っている。

三角関数についても 0° から 360° まで 1° 刻みにした、有効桁数 4 桁の数表から値を読み取っている。とくに三角関数は、ごく短い周期になる区間で描画精度がかなり落ちるので、注意が必要である。

▽ $y = 2\cos\left(2x - \dfrac{\pi}{6}\right)$ と $y = \tan x$

$2\cos\left(2x - \dfrac{\pi}{6}\right)$	$\tan x$
[×÷ の式にする]	
$2 \times \cos(2 \times x - \frac{\pi}{6})$	xt
[$A \times B \to$ AB* より]	
$2 \times \cos($(2x*)$ - 0.524)$	
[$A - B \to$ AB- より]	
$2 \times \cos$ (2x*{.524}-)	
[$\cos A \to$ Ac より]	
$2 \times$ (2x*{.524}-c)	
[$A \times B \to$ AB* より]	
22x*{.524}-c*	

```
\begin{drawpict}[.5cm](26, 12)(-7, -6)
  \coordinate[R](-7, 7)(-6, 6) % 左のグラフ
  \easyfx[22x*{.524}-c*](6, 6)
  %
  \baseskip{14}
  \coordinate[R](-5, 5)(-6, 6) % 右のグラフ
  \easyfx[xt](-4.57, -1.73)
  \linexis{-1.57}:(-6, 6) % 漸近線
  \easyfx[xt](-1.43, 1.43)
  \linexis{1.57}:(-6, 6) % 漸近線
  \easyfx[xt](1.73, 4.57)
  \putone
\end{drawpict}
```



\easyfx で三角関数のグラフを描画する際、描画区間に与える数値の単位はラジアンである。単位が ° である三角関数のグラフを描画する場合は、次ページの\sindegree を用いる。

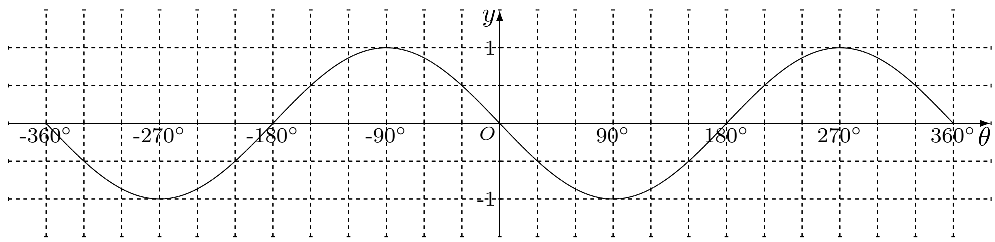
$\boxed{\textbf{\textbackslash sindegree[\#1,\#2,\#3,\#4](\#5,\#6)}}$ **\sindegree[a, b, c°, d]**$(\theta_l,\ \theta_r)$

$\boxed{\textbf{\textbackslash tandegree[\#1,\#2,\#3,\#4](\#5,\#6)}}$ **\tandegree[a, b, c°, d]**$(\theta_l,\ \theta_r)$

#1 $= a$, #2 $= b$, #3 $= c$, #4 $= d$ として、$y = a\sin(b\theta + c) + d$、および、$y = a\tan(b\theta + c) + d$ のグラフを区間 (#5, #6) で描画する。#2 **は 1, 2, 3, 4, 0.5, 0.33, 0.25 のみに対応している。**#3, #5, #6 **は度数の整数値で指定する。**

▽ $y = \sin(-\theta)$ のグラフを $-390° < \theta° < 390°$、$-1.5 < y < 1.5$ の範囲の θ-y 座標平面に描画；座標軸は $\underline{-13 < \theta < 13}$、$\underline{-3 < y < 3}$ を使用

```
\begin{drawpict}[.5cm](26, 6)(-13, -3)
  \coordinate[dg](-13, 13)(-3, 3)
  \sindegree[-1, 1, 0, 0](-360, 360)
\end{drawpict}
```



\sindegree、\tandegree は、1 目盛 $30°$ 刻みの θ 軸と、1 目盛 0.5 刻みの y 軸に描画することが前提である。したがって、$-390° < \theta° < 390°$ の範囲は $-13 < \theta < 13$ で、$-1.5 < y < 1.5$ の範囲は $-3 < y < 3$ である。このことは、たとえば\sindegree[1, 1, 0, 0](-360, 360) と\easyfx[xs](-6.28, 6.28) が同じ関数を表していても、描画される曲線が違うことを意味している。実際、\sindegree のグラフは\easyfx のグラフの 2 倍ほどの大きさで描画される。あくまでも\sindegree、\tandegree は、\coordinate の軸指定に d または r を指定した座標平面と併せて使用するものである。

b の値に -1 は指定できないので、$y = \sin(-\theta) = -\sin\theta$ として描画している。一般に

$$y = a\sin(-b\theta + c) + d \qquad \Rightarrow \qquad y = -a\sin(b\theta - c) + d$$
$$y = a\tan(-b\theta + c) + d \qquad \Rightarrow \qquad y = -a\tan(b\theta - c) + d$$

である。また、$\cos\theta = \sin(\theta + 90°)$ なので、cos のグラフは

$$y = a\cos(b\theta + c) + d \quad \Rightarrow \quad y = a\sin(b\theta + (c + 90°)) + d$$

として描くものとする。

#2 は指定値が決まっているが、tmtmath.sty ファイルに記述されている\def\tobasedeg#1{...}内に、必要な 1 文を追加すれば、指定値を増やすことができる。たとえば、#2 に $\dfrac{5}{3}$ を使いたければ

```
\ifdim\@cyc=1.66\p@\multiply\bdeg5\divide\bdeg3\fi
```

を追加して、#2 に 1.66 を指定すればよい。

$\triangledown\ \dfrac{(x+2)^2}{4}+\dfrac{y^2}{9}=1$ と $x^2-\dfrac{y^2}{2}=-1$

$$\pm\sqrt{9\left\{1-\dfrac{(x+2)^2}{4}\right\}}$$

[×÷ の式にする]

$$\pm\sqrt{9\times\{1-(x+2)\times(x+2)\div 4\}}$$

[$A+B\to$ `AB+` より]

$$\pm\sqrt{9\times\{1-\texttt{(x2+)}\times\texttt{(x2+)}\div 4\}}$$

[$A\times B\to$ `AB*` より]

$$\pm\sqrt{9\times\{1-\texttt{(x2+x2+*)}\div 4\}}$$

[$A\div B\to$ `AB/` より]

$$\pm\sqrt{9\times\{1-\texttt{(x2+x2+*4/)}\}}$$

[$A-B\to$ `AB-` より]

$$\pm\sqrt{9\times\texttt{(1x2+x2+*4/-)}}$$

[$A\times B\to$ `AB*` より]

$$\pm\sqrt{\texttt{91x2+x2+*4/-*}}$$

[$\sqrt{A}\to$ `Ar` より]

$$\pm\texttt{91x2+x2+*4/-*r}$$

$$\pm\sqrt{2x^2-2}$$

[×÷ の式にする]

$$\pm\sqrt{2\times x\times x-2}$$

[$A\times B\to$ `AB*` より]

$$\pm\sqrt{\texttt{(2x*)}\times x-2}$$

[$A\times B\to$ `AB*` より]

$$\pm\sqrt{\texttt{(2x*x*)}-2}$$

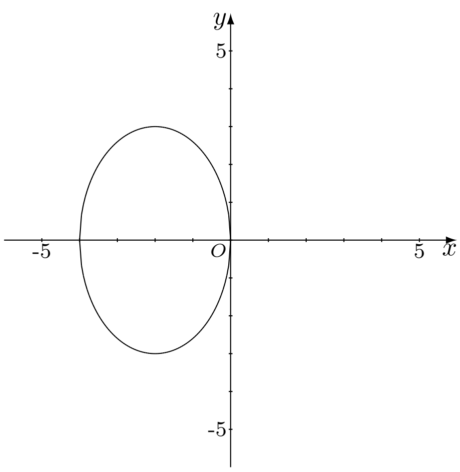
[$A-B\to$ `AB-` より]

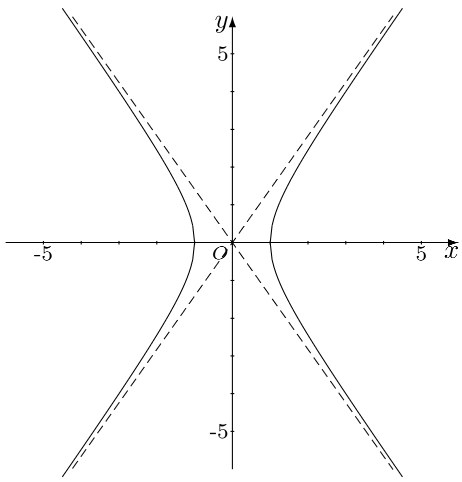
$$\pm\sqrt{\texttt{2x*x*2-}}$$

[$\sqrt{A}\to$ `Ar` より]

$$\pm\texttt{2x*x*2-r}$$

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6) % 左のグラフ
  \easyfx[91x2+x2+*4/-*r](-4, 0) % (y>0)
  \easyfx[{-1}91x2+x2+*4/-*r*](-4, 0) % (y<0)
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[2x*x*2-r](-4.5, -1) \easyfx[2x*x*2-r](1, 4.5) % (y>0)
  \easyfx[{-1}2x*x*2-r*](-4.5, -1) \easyfx[{-1}2x*x*2-r*](1, 4.5) % (y<0)
  \easyfx[{1.414}x*]:(-4.25, 4.25) % 漸近線
  \easyfx[{-1.414}x*]:(-4.25, 4.25) % 漸近線
\end{drawpict}
```





楕円や双曲線のような関数式は、いずれにせよ $y = f(x)$ の形に直さなくてはならない。公式の形として示せば

$$\frac{(x-a)^2}{C} + \frac{(y-b)^2}{D} = 1 \qquad \Rightarrow \qquad y = \pm\sqrt{D\left(1 - \frac{(x-a)^2}{C}\right)} + b$$

$$\frac{(x-a)^2}{C} - \frac{(y-b)^2}{D} = \pm 1 \qquad \Rightarrow \qquad y = \pm\sqrt{D\left(\frac{(x-a)^2}{C} \mp 1\right)} + b$$

である。最終的には $\pm$ の 2 値になるので、`\easyfx` に与える関数式も正負 2 通りが必要となる。

このとき、楕円は単に $y > 0$、$y < 0$ の関数式が 1 つずつで済むのに対し、双曲線は多少面倒である。$f(x,\ y) = 1$ の場合は原点の上下にグラフが描かれるので、$y > 0$、$y < 0$ の関数式は 1 つずつでよい。しかし p.14 で示したように、$f(x,\ y) = -1$ の場合は、原点の左右にグラフが描かれるので、$y > 0$、$y < 0$ 共に 2 分割して描かなくてはならない。つまり、$y > 0$、$y < 0$ の関数式は 2 つずつ必要である。

また、縦長の楕円や双曲線を描画すると、区間の両端でグラフが描かれないことがある。計算精度が粗いために、端点が計算対象にならないことが原因と思われる。必要なら`\qbezier` などでつなぐか、ほんのわずか—たとえば (0.00 何とか) ほど—描画区間を広く取ると、うまく線がつながる場合が多い。

$\triangledown\ y^2 = x^3 - x$

$$\pm\sqrt{x^3 - x}$$

[×÷ の式にする]

$$\pm\sqrt{x \times x \times x - x}$$

[$A \times B \to$ `AB*` より]

$$\pm\sqrt{\texttt{(xx*)} \times x - x}$$

[$A \times B \to$ `AB*` より]

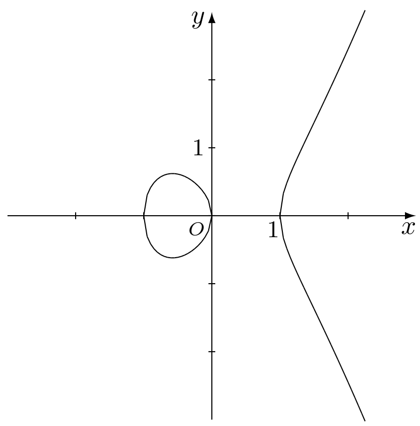
$$\pm\sqrt{\texttt{(xx*x*)} - x}$$

[$A - B \to$ `AB-` より]

$$\pm\sqrt{\texttt{xx*x*x-}}$$

[$\sqrt{\texttt{A}} \to$ `Ar` より]

$$\pm\texttt{xx*x*x-r}$$




```
\begin{drawpict}[.9cm](16, 6)(-3, -3)
  \coordinate[R](-3, 3)(-3, 3)
  \apeex(.9, -.2){$1$}(-.2, 1){$1$}
  \easyfx[xx*x*x-r](-1, 0) \easyfx[xx*x*x-r](1, 2.25) % (y>0)
  \easyfx[{-1}xx*x*x-r*](-1, 0) \easyfx[{-1}xx*x*x-r*](1, 2.25) % (y<0)
\end{drawpict}
```

$y^2 = x^3 - x$ を y について解けば $y = \pm\sqrt{x^3 - x} = \pm\sqrt{(x+1)x(x-1)}$ であるから、根号内が正になる x の範囲は $-1 < x < 0$、$1 < x$ である。したがってグラフ描画は、これら x の範囲で計 4 回必要である。

$\bigtriangledown\ y = xe^{-x}$ と $y = \dfrac{x+2}{x^2+1}$

$$xe^{-x}$$

[×÷ の式にする]

$$x \times e^{(-1)\times x}$$

[$A \times B \to$ `AB*` より]

$$x \times e^{\texttt{(\{-1\}x*)}}$$

[$e^A \to$ `Ae` より]

$$x \times \texttt{(\{-1\}x*e)}$$

[$A \times B \to$ `AB*` より]

`x{-1}x*e*`

$$\frac{x+2}{x^2+1}$$

[×÷ の式にする]

$$(x+2) \div (x \times x + 1)$$

[$A + B \to$ `AB+` より]

$$\texttt{(x2+)} \div (x \times x + 1)$$

[$A \times B \to$ `AB*` より]

$$\texttt{(x2+)} \div (\texttt{(xx*)} + 1)$$

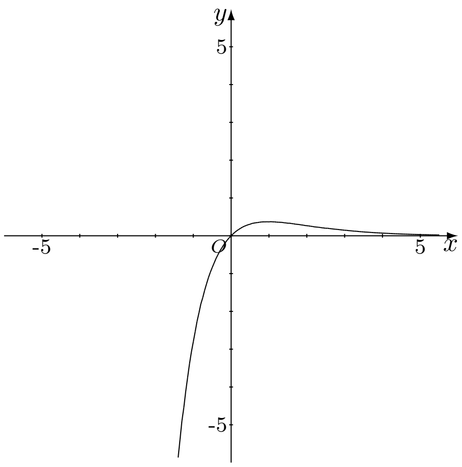
[$A + B \to$ `AB+` より]

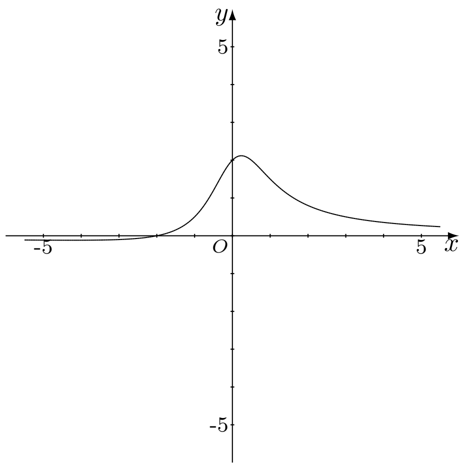
$$\texttt{(x2+)} \div \texttt{(xx*1+)}$$

[$A \div B \to$ `AB/` より]

`x2+xx*1+/`

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6) % 左のグラフ
  \easyfx[x{-1}x*e*](-1.4, 5.5)
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[x2+xx*1+/](-5.5, 5.5)
\end{drawpict}
```



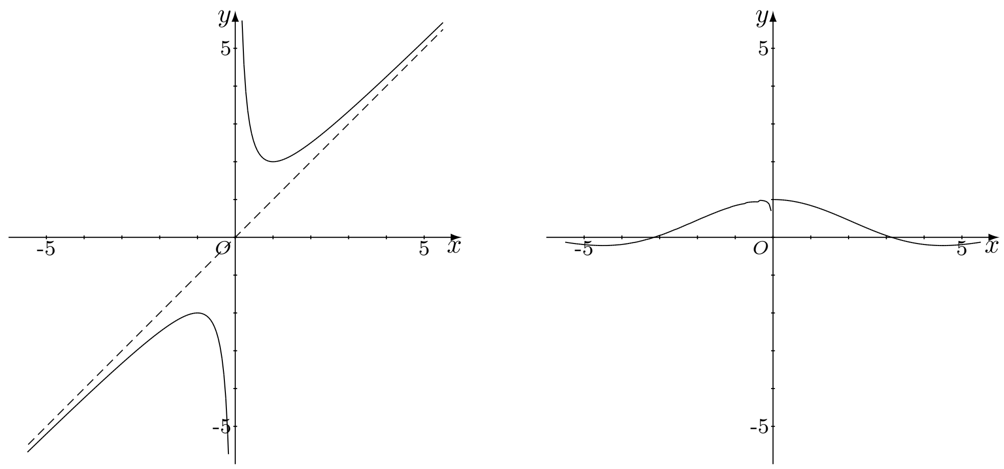


ここからの例に示す関数は、いわゆる合成関数である。単一関数でも合成関数でも、関数式の記述の仕方は大差ないが、合成関数は単一関数を組み合わせて描くことはできない。したがって、合成関数は基本的に`\easyfx`でのみ描画できると思ってよい。単一関数は`\easyfx`の他に、単一関数のマクロ（p.19）でも描画できるが、その必要はないだろう。

$\triangledown\ y = x + \dfrac{1}{x}$ と $y = \dfrac{\sin x}{x}$

$x + \dfrac{1}{x}$	$\dfrac{\sin x}{x}$
[×÷ の式にする]	[×÷ の式にする]
$x + 1 \div x$	$\sin x \div x$
[$A \div B \to$ AB/ より]	[$\sin A \to$ As より]
$x +$ (1x/)	(xs) $\div x$
[$A + B \to$ AB+ より]	[$A \div B \to$ AB/ より]
x1x/+	xsx/

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6) % 左のグラフ
  \easyfx[x1x/+](-5.5, -.18) % (x<0)
  \easyfx[x1x/+](.18, 5.5) % (x>0)
  \easyfx[1x*]:(-5.5, 5.5) % 漸近線
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[xsx/](-5.5, -.05) % (x<0)
  \easyfx[xSx/](.05, 5.5) % (x>0)
\end{drawpict}
```



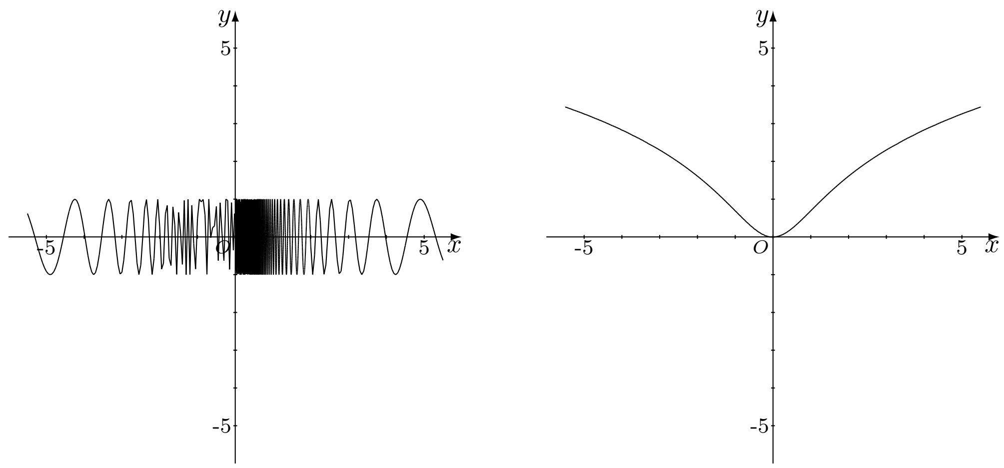
$y = (\sin x)/x$ を\easyfx[xsx/](-5.5, 5.5) として描くと、原点付近でグラフが乱れて描画される。原因は演算子 s が sin の値を数表から読み取るだけのため、$x \approx 0$ では $\sin x = 0$ となるからであろう。これは描画精度を細かくしても解決できない。

しかし演算子 S は、sin の値を数表から読み取る際、単純比例であるが値を補間する。そのため s より明らかにマシな描画をする。\easyfx[xsx/](-5.5, -.05) と\easyfx[xSx/](.05, 5.5) を比較してもらいたい。ただし、計算量が多少増えるので適切に使用しないと、処理時間の増大やオーバーフローの心配が生じる。

$\triangledown\ y = \sin\dfrac{100}{x}$ と $y = \log_e(x^2+1)$

$\sin\dfrac{100}{x}$	$\log_e(x^2+1)$
[×÷ の式にする]	[×÷ の式にする]
$\sin(100 \div x)$	$\log_e(x \times x + 1)$
[$A \div B \to$ `AB/` より]	[$A \times B \to$ `AB*` より]
sin (`{100}x/`)	$\log_e$((`xx*`) + 1)
[$\sin A \to$ `AS` より]	[$A + B \to$ `AB+` より]
`{100}x/S`	$\log_e$ (`xx*1+`)
	[$\log_e A \to$ `Al` より]
	`xx*1+l`

```
\begin{drawpict}[.5cm](26, 12)(-6, -6)
  \coordinate[R](-6, 6)(-6, 6)  % 左のグラフ
  \easyfx[{100}x/S](-5.5, -.02)  % (x<0)
  \begin{drawfunc}(.02, 2) % (x>0)
    \@steps=.001\p@
    \def\calcdimen{\RPNfx[{100}x/S]}
  \end{drawfunc}
  \easyfx[{100}x/S](2, 5.5)
  %
  \baseskip{14}
  \coordinate[R](-6, 6)(-6, 6) % 右のグラフ
  \easyfx[xx*1+l](-5.5, 5.5)
\end{drawpict}
```



$y = \sin(100/x)$ は、演算子に S を用いただけでは粗いグラフにしかならない。原因は、$100/x$ が大きな値になるにつれて、sin の値を数表から読む間隔が広くなるからである。

そこで drawfunc 環境（p.20）を用いて、描画精度を細かくした（$x > 0$ のグラフ）。処理時間の増大やオーバーフローを防ぐためグラフを分割して描画したが、明らかに “それっぽい” であろう。

単一関数（過去の遺物）

逆ポーランド記法がうまく使えないと\easyfx で正しい描画をすることができない。そのようなとき、または、もっとも基本的である関数を描画したいときなどは、\easyfx を定義する以前からある、関数ごとの定義式を用いて描画してもよい。\easyfx があれば、これらの定義は原則不要なので、ここではマクロ名と機能だけを示すことにする。

\abline[#1,#2]#3(#4,#5) \abline[a, b]{op}(x_l, x_r)

#1 $= a$, #2 $= b$ として、$y = ax + b$ のグラフを区間 (#4, #5) で描画する。

\parabola[#1,#2,#3](#4,#5) \parabola[a, b, c](x_l, x_r)

#1 $= a$, #2 $= b$, #3 $= c$ として、$y = ax^2 + bx + c$ のグラフを区間 (#4, #5) で描画する。

\cubecurve[#1,#2,#3,#4](#5,#6) \cubecurve[a, b, c, d](x_l, x_r)

#1 $= a$, #2 $= b$, #3 $= c$, #4 $= d$ として、$y = ax^3 + bx^2 + cx + d$ のグラフを区間 (#5, #6) で描画する。

\fraccurve[#1,#2,#3,#4](#5,#6) \fraccurve[a, b, c, d](x_l, x_r)

#1 $= a$, #2 $= b$, #3 $= c$, #4 $= d$ として、$y = \dfrac{a}{bx+c} + d$ のグラフを区間 (#5, #6) で描画する。漸近線に対する x の値、すなわち $-\dfrac{c}{b}$ が区間 (#5, #6) に含まれてはいけない。

\sqrtcurve[#1,#2,#3,#4](#5,#6) \sqrtcurve[a, b, c, d](x_l, x_r)

#1 $= a$, #2 $= b$, #3 $= c$, #4 $= d$ として、$y = a\sqrt{bx+c} + d$ のグラフを区間 (#5, #6) で描画する。根号内を負にする x の値が区間 (#5, #6) に含まれてはいけない。

\expcurve#1[#2,#3,#4,#5](#6,#7) \expcurve{op}[a, b, c, d](x_l, x_r)

#2 $= a$, #3 $= b$, #4 $= c$, #5 $= d$ として、$y = a \cdot b^{cx} + d$ のグラフを区間 (#6, #7) で描画する。#1 はオプションで、対数関数（$y = a \cdot b^{cx} + d$ の逆関数）を描画するとき-にする。

\hyperoval#1[#2,#3,#4,#5](#6,#7) \hyperoval{op なし/+/−}[a, b, C, D](x_l, x_r)

#2 $= a$, #3 $= b$, #4 $= C$, #5 $= D$ として、2 次曲線 $\dfrac{(x-a)^2}{C} + \dfrac{(y-b)^2}{D} = 1$ のグラフを区間 (#6, #7) で描画する。#1 は (op なし)/+/-のいずれか。$C,\ D > 0$ ならば楕円で、とくに $C = D$ の場合は半径 $\sqrt{C}$ の円になる。また、$CD < 0$ ならば、$\dfrac{(x-a)^2}{C} - \dfrac{(y-b)^2}{D} = 1$ もしくは $\dfrac{(x-a)^2}{C} - \dfrac{(y-b)^2}{D} = -1$ の双曲線になる。オプションがない場合は、区間 (#6, #7) における完全なグラフを、+の場合は $y > 0$ の部分のグラフを、-の場合は $y < 0$ の部分のグラフを描画する。

特殊な描画環境

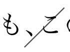
[drawfunc 環境]

```
\begin{drawfunc}#1(#2,#3)
  \@steps=0.01\p@ % 必要に応じて記述する描画精度の値
  \def\calcdimen{\RPNfx[ f(x) の関数式 ]}
\end{drawfunc}
```

#1 はオプションで、: にするとグラフが破線になる。x の区間 (#2, #3) において、$f(x)$ の関数式 に定義した関数を描画する。描画精度の値を記述しないと、規定値の 0.05\p@が使われる。drawpict 環境か picture 環境で使用することが前提だが、文章中でも使用できる。\easyfx は描画精度を変更できないので、描画精度を変更したいときに使用する。実際の使用にあたっては、下線部だけを逆ポーランド記法で記述し、他はそのままでよい。

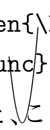
ところで、関数式を定義すると書いたが、drawfunc 環境が内部でしていることは、区間開始の x の値#2 を変数（レジスタ）である\dimen@に代入し、計算された y の値を再び\dimen@に代入するというものである。この計算を\@steps 刻みで#3 まで繰り返し、\qbezier でつなぐことで関数のグラフが描画される。つまり、drawfunc 環境で区間の数値が与えられれば、座標 $(x,\ y)$ が必ず計算され線でつながれることになる。そのため、関数を定義しないまま、この場で

```
\begin{drawfunc}(-5, 10)%
  \calcdimen{\RPNfx[]}
\end{drawfunc}
```

のように使っても、このように $y = x$ の線が描画される。それは、計算式がなくても x の区間 $(-5,\ 10)$ の値が順に x と y に代入されているからである。

しかし、たとえば $f(x)$ の関数式を与えて、この場で

```
\begin{drawfunc}(-5, 6)%
  \calcdimen{\RPNfx[xx*]}
\end{drawfunc}
```



のようにすると、このように放物線が描画される。これは、関数 $y = x^2$ を定義したので、x の区間 $(-5,\ 6)$ の値が順に x^2 に代入された結果が反映されるからである。また、文章中の描画単位は pt であるから、直線と放物線はこの大きさになった。drawpict 環境中で drawfunc 環境を使えば、drawpict 環境で指定された描画単位になる。

[drawfuncp 環境]

```
\begin{drawfuncp}#1(#2,#3)
  \@steps=0.01\p@ % 必要に応じて記述する描画精度の値
  \def\calcdimenx{\RPNfx[ f(t) の関数式 ]}
  \def\calcdimeny{\RPNgx[ g(t) の関数式 ]}
\end{drawfuncp}
```

#1 はオプションで、: にするとグラフが破線になる。t を媒介変数（パラメータ）とする区間 (#2, #3) において、$f(t)$ の関数式 と $g(t)$ の関数式 に定義した関数を描画する。

[drawfinv 環境]

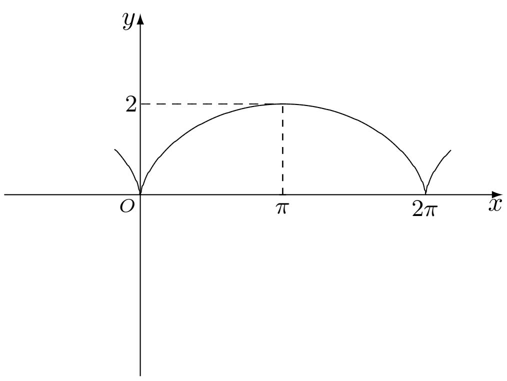
```
\begin{drawfinv}#1(#2,#3)
  \@steps=0.01\p@ % 必要に応じて記述する描画精度の値
  \def\calcdimen{\RPNfx[ f(x) の関数式 ]}
\end{drawfinv}
```

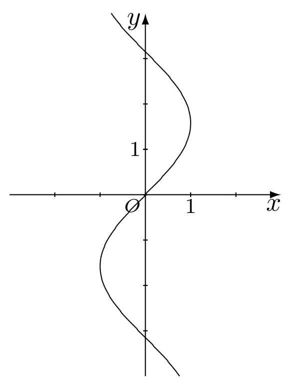
#1 はオプションで、: にするとグラフが破線になる。x の区間 (#2, #3) において、$f(x)$ の関数式に定義した関数の逆関数を描画する。

▽ $x = t - \sin t$, $y = 1 - \cos t$（サイクロイド）　と　$y = \sin x$ の逆関数

$t - \sin t,\ 1 - \cos t$	$\sin x$
[パラメータ t は x と見る] xxs-, 1xc-	xs

```
\begin{drawpict}[.6cm](21, 8)(-3, -4)
  \coordinate[A](-3, 8)(-4, 4) % 左のグラフ
  \begin{drawfuncp}(-1.57, 7.85)
    \def\calcdimenx{\RPNfx[xxs-]}
    \def\calcdimeny{\RPNgx[1xc-]}
  \end{drawfuncp}
  \locate[x](3.14, 2)
  \apeeex(3.14, -.3){$pi$}(6.28, -.3){$2\pi$}(-.2, 2){$2$}
  %
  \baseskip{14}
  \coordinate[R](-3, 3)(-4, 4) % 右のグラフ
  \begin{drawfinv}(-4, 4)
    \def\calcdimen{\RPNfx[xs]}
  \end{drawfinv}
  \putones
\end{drawpict}
```





その他のグラフに関連する描画

$\boxed{\textbf{\textbackslash unitcircle}[\#1]}$ $\quad$ **\unitcircle**$[t\text{-}ax]$

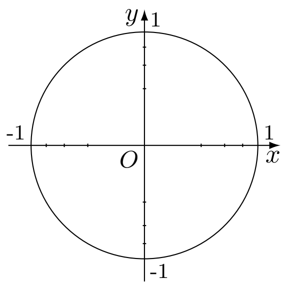
#1 は tangent 線。

#1 なし	$\to$	x 軸、y 軸、原点 O、単位円の描画（規定値）
t	$\to$	規定値に tan 用目盛を記入

三角関数で利用する単位円を描画する。軸には 30°、45°、60° の倍数角に対応する目盛だけが入る。数値は未記入である。

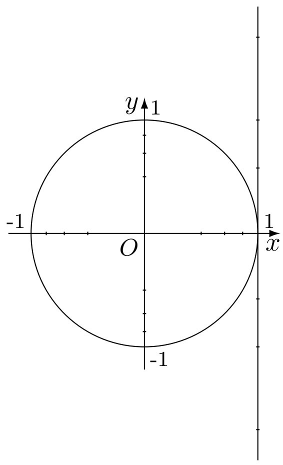
▽単位円を $-1.2 < x < 1.2$、$-1.2 < y < 1.2$ の範囲の x-y 座標平面に描画

```
\begin{drawpict}[1.5cm](2.4, 2.4)(-1.2, -1.2)
  \unitcircle
\end{drawpict}
```



▽単位円と tan 用目盛の描画

```
\begin{drawpict}[1.5cm](2.4, 4.0)(-1.2, -2.0)
  \unitcircle[t]
\end{drawpict}
```



\unitcircle はプリアンブルに\usepackage{pict2e}を宣言して使う。pict2e パッケージがないと円の大きさが 40pt にしかならない。その場合は\easyfx で円を描けばよい。

$\boxed{\text{\rangeline(\#1,\#2)\#3}}$ **\rangeline(l, r){ax}**

#1 から#2 までの数直線。#3 は数直線の名称（不要なら {} にする）。

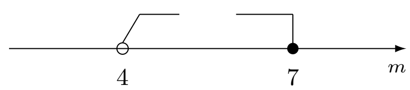
$\boxed{\text{\brange\#1(\#2,\#3)\#4}}$ **\brange{n}(s, e){v}**

$\boxed{\text{\wrange\#1(\#2,\#3)\#4}}$ **\wrange{n}(s, e){v}**

閉区間（黒丸）または開区間（白丸）から伸びる線分を描く。#1 は 0, 1, 2, 3, 4 のいずれかで、数直線からの高さがそれぞれ 0.6, 0.8, 1, 1.2, 1.4 となっている。#2 は始点となる丸の位置を、#3 は端点にあたる位置を表す。#2 < #3 のときは丸から右へ延びる線分を、#2 > #3 のときは丸から左へ延びる線分を描く。#4 は丸位置を示す値で、丸の下に出力される（不要なら {} にする）。

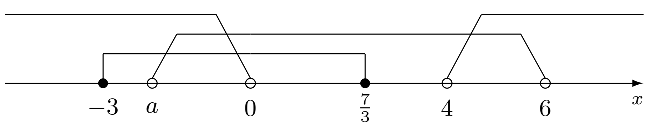
▽ $4 < m \leqq 7$ の範囲を $2 < m < 9$ の範囲の数直線に描画

```
\begin{drawpict}[.75cm](7, 2)(2, -1)
  \rangeline(2, 9){$m$}
  \wrange0(4, 5){$4$}
  \brange0(7, 6){$7$} % 描画見本のため、あえて範囲を (7, 5) にせず!
\end{drawpict}
```



▽ $-3 \leqq x \leqq \dfrac{7}{3}$、$a < x < 6$、$x < 0$, $4 < x$ の範囲を $-5 < x < 8$ の範囲の数直線に描画

```
\begin{drawpict}[.65cm](13, 2)(-5, 0)
  \rangeline(-5, 8){$x$}
  \brange0(-3, 0){$-3$} \brange0(2.333, 0){$\frac{7}{3}$}
  \wrange2(-2, 0){$a$} \wrange2(6, 0){$6$}
  \wrange4(0, -5){$0$} \wrange4(4, 8){$4$}
\end{drawpict}
```



平面図形の描画

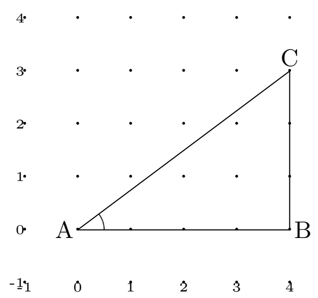
すべてのマクロは、点の位置を座標で示して描画する。\line と\bezier 曲線だけで描くよりはマシという程度かもしれない。はじめに使用できるマクロをすべて提示し、その後で作図例を示す。

〓描画のためのマクロ

$\boxed{\textbf{\textbackslash dotsgrid\#1(\#2,\#3)(\#4,\#5)}}$ $\qquad$ **\dotsgrid{**d**}(**$x_{lb},\ y_{lb}$**)(**$x_{rt},\ y_{rt}$**)**

#1 間隔の格子点を、左下の座標 (#2,#3) 以上、右上の座標 (#4,#5) 未満の矩形範囲に表示する。**整数値しか指定できない**ので、右上の座標 $(x_{rt},\ y_{rt})$ まで格子点を表示させるには (#4,#5) に $(x_{rt}+1,\ y_{rt}+1)$ を指定する。作図が終われば dotsgrid は削除すればよい。

```
\begin{drawpict}[.7cm](4, 4)
  \dotsgrid1(-1, -1)(5, 5)
  \pooly(0, 0)(4, 0)(4, 3)
%点名称の位置、弧の半径などを決める目安に便利だろう
  \vparade[.25]ABC{};{180}{0}{90}0
  \arcdegree(0, 0, .5)(0, 37)
\end{drawpict}
```



$\boxed{\textbf{\textbackslash siide(\#1,\#2)(\#3,\#4)}}$ $\qquad$ **\siide(**$x_1,\ y_1$**)(**$x_2,\ y_2$**)**

$\boxed{\textbf{\textbackslash siiide(\#1,\#2)(\#3,\#4)(\#5,\#6)}}$ $\qquad$ **\siiide(**$x_1,\ y_1$**)(**$x_2,\ y_2$**)(**$x_3,\ y_3$**)**

$\boxed{\textbf{\textbackslash siiiide(\#1,\#2)(\#3,\#4)(\#5,\#6)(\#7,\#8)}}$
\siiiide($x_1,\ y_1$**)(**$x_2,\ y_2$**)(**$x_3,\ y_3$**)(**$x_4,\ y_4$**)**

$\boxed{\textbf{\textbackslash vsiide(\#1,\#2)(\#3,\#4)}}$ $\qquad$ **\vsiide(**$x_s,\ y_s$**)(**$x_e,\ y_e$**)**

[**i の数が頂点の数を示している**]　開いた連鎖線分、すなわち頂点 (#1, #2) から頂点 (#m, #n) までを順に線分で結ぶ。\vsiide のみ、(#1, #2) から (#3, #4) へ向かうベクトルになる。\vsiide が描く矢印は、\vector の引数として-999 から 999 を使うので pict2e パッケージを必要とする。

$\boxed{\textbf{\textbackslash extendline\#1(\#2,\#3)(\#4,\#5)\#6}}$
\extendline{l-*length***}(**$x_l,\ y_l$**)(**$x_r,\ y_r$**){**r-*length***}**

線分 (#2, #3)-(#4, #5) の左・右を、それぞれ長さ#1・#6 だけ延ばした線分を描画する。$x_l = x_r$ のときは、線分 (#2, #3)-(#4, #5) の下・上を、それぞれ長さ#1・#6 だけ延ばした線分を描画する。

$\boxed{\textbf{\textbackslash poooly(\#1,\#2)(\#3,\#4)(\#5,\#6)}}$ **\poooly**$(x_1,\, y_1)(x_2,\, y_2)(x_3,\, y_3)$

$\boxed{\textbf{\textbackslash pooooly(\#1,\#2)(\#3,\#4)(\#5,\#6)(\#7,\#8)}}$

\pooooly$(x_1,\, y_1)(x_2,\, y_2)(x_3,\, y_3)(x_4,\, y_4)$

[o **の数が頂点の数を示している**]　頂点 (#1, #2) から頂点 (#m, #n) までを結んで三角形と四角形を描画する。五角形以上の多角形は`\siide`, `\siiide`, `\siiiide` を組み合わせて描画する。

`\siide` 系マクロと`\poooly` 系マクロは破線や点線が描画できない。図形の描画でどうしても破線や点線を使うなら、図形の描画単位を `pt` でなく `cm` にすれば、1 次関数の式などを使って破線が描きやすい。ただし、この場合は点線は描けない。もし、描画単位が `pt` であれば、`\qbezier[n](..)(..)(..)` の `n` に点の個数を指定することで点線になる。ただし、この場合は破線は描けない。破線と点線を同時に描くなら、描画途中で描画単位を変えながら描くことになるだろう。

$\boxed{\textbf{\textbackslash arcdegree\#1(\#2,\#3,\#4)(\#5,\#6)}}$

\arcdegree{ なし$/+/-/*$**}**$(c,\, d,\, r)(\alpha,\, \beta)$

中心 (#2, #3)、半径 #4 の弧を、角 (#5, #6) の範囲で描画する。#2 $= c$, #3 $= d$, #4 $= r$。#5 $= \alpha°$, #6 $= \beta°$ は整数度を指定し、負の値や 360 より大きい値も使える。ただし、必ず $\alpha < \beta$ とする。#1 が+のときは弧の終点に、-のときは弧の始点に、*のときは弧の終点と始点に矢印を付加する。

$\boxed{\textbf{\textbackslash apex(\#1,\#2)\#3}}$ **\apex**$(x_1,\, y_1)$**X**

$\boxed{\textbf{\textbackslash apeex(\#1,\#2)\#3(\#4,\#5)\#6}}$ **\apeex**$(x_1,\, y_1)$**X**$(x_2,\, y_2)$**Y**

$\boxed{\textbf{\textbackslash apeeex(\#1,\#2)\#3(\#4,\#5)\#6(\#7,\#8)\#9}}$

\apeeex$(x_1,\, y_1)$**X**$(x_2,\, y_2)$**Y**$(x_3,\, y_3)$**Z**

[e **の数が頂点の数を示している**]　各頂点 (#m, #n) の位置に名称#3, #6, #9 を`\small` サイズの文字で表示する。違う文字サイズを指定するとそれが有効になる。5 個以上の名称を表示するときは `\apex`, `\apeex`, `\apeeex` を組み合わせて表示する。また、`\apex(#1,#2)#3` は座標と頂点名の位置が逆に感じるかもしれないが、このほうが`\apex` を`\put` に置き換えやすい。

$\boxed{\textbf{\textbackslash vertex(\#1,\#2,\#3)\#4\#5}}$ **\vertex**$(c,\, d,\, r)$**{**θ**}X**

点 (#1, #2) を中心とし、0° の方向（3 時の方向）にある半径 #3 の動径を角#4（$\theta°$）だけ回転した位置に、#5（X）を表示する（p.29:`\ORBIT` を参照!）。θ は $-360 < \theta < 360$ の範囲の整数値で指定する。`\apex` とうまく使い分けるのがよいだろう。

$\boxed{\textbf{\textbackslash vparade[\#1]\#2\#3\#4\#5;\#6\#7\#8\#9}}$ **\vparade**$[r]\mathbf{V}_1\mathbf{V}_2\mathbf{V}_3\mathbf{V}_4$**;**$\theta_1\theta_2\theta_3\theta_4$

主に頂点の名称を **4 個**まとめて表示するために、\siide 系マクロ、\pooooly 系マクロ、および \extendline で線を引いた 直後に使うこと を想定している。\siide などを実行した直後は一時的に座標の値を保持している。その値を用いて、#2〜#5 までの頂点名称を、それぞれの頂点を中心とし、0° の方向（3 時の方向）にある半径 #1 の動径を#6〜#9 の角度回転した位置に、順に表示する（p.29:\ORBIT を参照!）。θ は $-360 < \theta < 360$ の範囲の整数値で指定する。**表示する $\mathbf{V}_n$ がないときは{}を記述し、角度 θ_n には 0 を当てる。**もし、V_n がないときの 0 と回転角 0° の 0 を区別するなら、正規の 0° を{0}で表すと納まりがよいだろう。実際、

```
\begin{drawpict}[1cm](5, 1)(0, 0)
  \pooooly(0, 0)(1, 0)(1, 1)(0, 1)
  \vparade[.15]ABCD;{-135}{-45}{45}{135}

  \baseskip4
  \siiide(0, 0)(1, 1)(1, 0)
  \vparade[.2]{P$_1$}{P$_2$}{P$_3$}{};{180}{90}{0}0
\end{drawpict}
```

のようにすれば、

のようになる。名称はどれも、頂点から同じ距離だけ離れるので、1 個だけ位置が気に入らないような場合は、その頂点名称は{}に、回転角は 0 にして、別途\vertex を使えばよい。

マクロ名の中に [] や; があるのは煩わしいかもしれないが、{}が多いときは対応が見にくくなるかもしれないので、あえて区切り記号として入れている。邪魔だと思えば、tmtmath.sty ファイルから取り除いてもらいたい。

直後に使うこと について

\vparade を\siide などの実行直後に使うという意味は、\siide などを実行しているグループ内で使うという意味を含んでいる。それは、保持した値が全域的に使えないためである。たとえば上の例で、四角形だけ太線にする場合は

```
{\thicklines\pooooly(0, 0)(1, 0)(1, 1)(0, 1)}
```

または

```
\thicklines\pooooly(0, 0)(1, 0)(1, 1)(0, 1)
\thinlines
```

とするだろう。このとき\pooooly の実行直後とは、

```
{\thicklines\pooooly(0, 0)(1, 0)(1, 1)(0, 1) \vparade...}
```

の位置で使用することである。

```
{\thicklines\pooooly(0, 0)(1, 0)(1, 1)(0, 1)} \vparade...
```

の位置で使用すると、グループの外で実行することになって、\vparade が正しく機能しないので注意が必要である。この仕様が気に入らなければ、値を\global にしてほしい。

\slur#1(#2,#3)(#4,#5)#6

\slur{+/−}$(x_1,\ y_1)(x_2,\ y_2)$**X**

2 点 (#2, #3), (#4, #5) をスラー線で結び、値#6 を\small サイズの文字で表示する。違う文字サイズを指定するとそれが有効になる。#1 はオプションで+か-が指定できる。\slur+は (x_1, y_1) から (x_2, y_2) へ向けて引かれた線分の右側に、\slur-は (x_1, y_1) から (x_2, y_2) へ向けて引かれた線分の左側に、山が出るスラー線になる。値#6 の位置は固定されているが、\makebox(0, 0)[lrtb]{X} の書式を合わせると、多少の調整は可能。オプション#1 を指定しないとスラー線は単に直線となって、値#6 が 2 点 (#2, #3), (#4, #5) の中央、つまり直線上に置かれる。

スラー線のふくらみと値を表示する位置は、マクロ中の\dimendiv(\strip@pt\dimen@, 3) で決めている。3 を小さくすればふくらみが増し、大きくすればふくらみは減る。この位置は、線分の中点から垂直方向へ、中点までの長さの 1/3 のところである。

\putsymbol[#1]#2(#3,#4)(#5,#6)#7

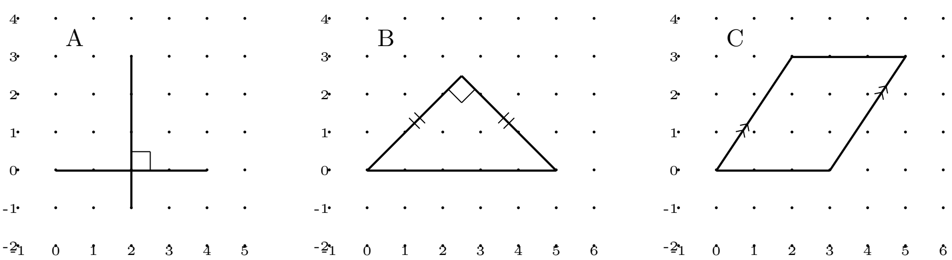
\putsymbol[*symbol*]**{**$length$**}**$(x_1,\ y_1)(x_2,\ y_2)t$

#1 は必ず指定する。

r | e/E | p/P　→　直角記号 | 等辺記号（単線/複線）| 平行記号（単線/複線）

長さ#2 の直角、等辺、平行記号を、線分 (#3, #4)-(#5, #6) の#7 分点の位置に、次ページの\symbol の書式で描画する。\symbol は単独で使用してもよいのだが、傾きなどをあらかじめ計算する必要があるなど、使い勝手がよくない。\putsymbol は\symbol をサブルーティンにしたマクロである。

直角記号は、線分の始点回りに描くときは $t = 0$ を、線分の終点回りに描くときは $t = 1$ を指定するのだが、多少慣れが要るかもしれない。等辺、平行記号は大抵、辺の中点あたりに描くだろうから、$t = 0.5$ を指定すればよい。実際の使用例は、下図を参照してもらいたい。



（A）\putsymbol[r]{.5}(2, 0)(4, 0)0 または \putsymbol[r]{.5}(0, 0)(2, 0)1 または
\putsymbol[r]{.5}(0, 0)(4, 0){.5}

（B）\putsymbol[r]{.5}(2.5, 2.5)(0, 0)0
\putsymbol[E]{.2}(0, 0)(2.5, 2.5){.5} および \putsymbol[E]{.2}(5, 0)(2.5, 2.5){.5}

（C）\putsymbol[P]{.2}(0, 0)(2, 3){.333} および \putsymbol[P]{.2}(3, 0)(5, 3){.666}

\symbol[#1](#2,#3)(#4,#5)#6

$\textbf{\textbackslash symbol}[symbol]\textbf{(}x_0\textbf{,}\ y_0\textbf{)(}dx\textbf{,}\ dy\textbf{)\{}length\textbf{\}}$

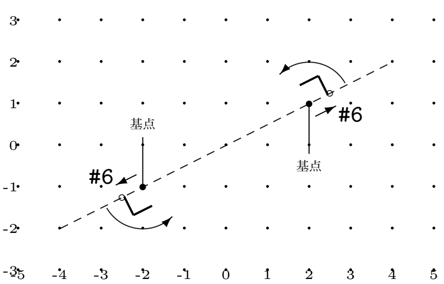
#1 は必ず指定する。

r | e/E | p/P → 直角記号 | 等辺記号（単線/複線） | 平行記号（単線/複線）

$[op = \mathbf{r}]$ **直角記号** 点 (#2, #3) を基点として、傾き $\frac{\#5}{\#4}$ の方向に、長さ #6 進んだ点から、反時計回りに長さ#6 の直角記号を描画する（下図参照）。#2 $= x_0$, #3 $= y_0$, #4 $= dx$, #5 $= dy$, #6 $= length$ は、すべて実数値を指定できるが、**数値は 10 程度以下とする**（傾きが $\dfrac{61}{37}$ のようなときは $\dfrac{6.1}{3.7}$ と考えるか、$\dfrac{1.64}{1}$ と考えるとよい）。

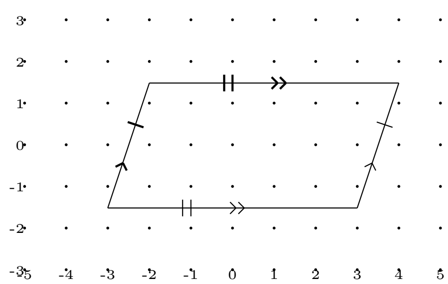
$[op = \mathbf{e/E}]$ **等辺記号** 点 (#2, #3) を基点として、傾き $\frac{\#5}{\#4}$ の方向に垂直な、長さ #6 の 2 倍の線を描画する（下図参照）。#2 $= x_0$, #3 $= y_0$, #4 $= dx$, #5 $= dy$, #6 $= length$ は、すべて実数値を指定できるが、数値は 10 程度以下とする。複線の場合は、傾き $\frac{\#5}{\#4}$ の方向に、長さ #6 進んだ位置に 2 本めが描かれる。

$[op = \mathbf{p/P}]$ **平行記号** 点 (#2, #3) を基点として、傾き $\frac{\#5}{\#4}$ の方向に矢の先が向く、長さ #6 の矢印を描画する（下図参照）。#2 $= x_0$, #3 $= y_0$, #4 $= dx$, #5 $= dy$, #6 $= length$ は、すべて実数値を指定できるが、数値は 10 程度以下とする。複線の場合は、傾き $\frac{\#5}{\#4}$ の方向に、長さ #6 進んだ位置に 2 本めが描かれる。



（左）\symbol[r](-2, -1)(-2,-1){.5}

（右）\symbol[r](2, 1)(2,1){.5}



（左辺上）\symbol[e](-2.33, .5)(1, 3){.2}

（左辺下）\symbol[p](-2.66, -.5)(1, 3){.2}

（上辺左）\symbol[E](0, 1.5)(-1, 0){.2}

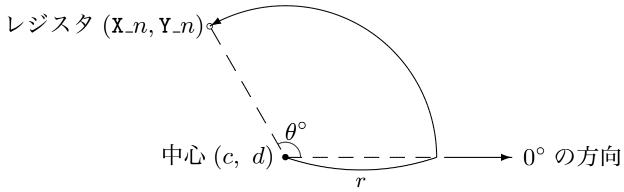
（上辺右）\symbol[P](1, 1.5)(1, 0){.2}

■値保持のためのマクロ

>>座標上の 1 点の値を保持

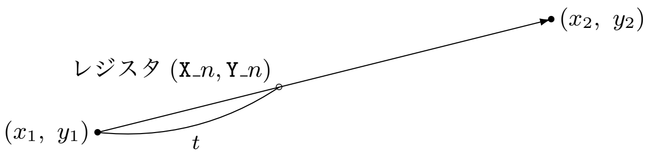
$\boxed{\text{\textbackslash ORBIT\#1(\#2,\#3,\#4)\#5}}$ \ORBIT$n(c,\ d,\ r)\theta$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。点 (#2, #3) を中心とし、0° の方向（3 時の方向）にある半径 #4 の動径を角 θ° だけ回転した点の座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と\Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。θ は $-360 < \theta < 360$ の範囲の**整数値**で指定する。



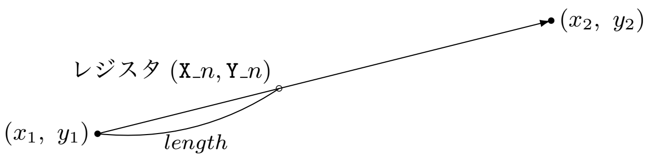
$\boxed{\text{\textbackslash DIVIDE\#1(\#2,\#3)(\#4,\#5)\#6}}$ \DIVIDE$n(x_1,\ y_1)(x_2,\ y_2)t$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。線分 (#2, #3)-(#4, #5) を $t : (1-t)$ に内分または外分する点の座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と\Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。$0 < t < 1$ のとき内分点、$t < 0,\ 1 < t$ のとき外分点となる。



$\boxed{\text{\textbackslash CUTLINE\#1(\#2,\#3)(\#4,\#5)\#6}}$ \CUTLINE$n(x_1,\ y_1)(x_2,\ y_2)\{length\}$

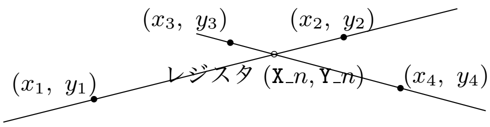
#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。有向線分 (#2, #3)-(#4, #5) 上の始点から、長さ#6 の線分の終点座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と\Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。\DIVIDE が 線分に対する比 で位置を計算するのに対し、\CUTLINE は 線分の長さ で位置を計算することに注意。



\XPOINT#1(#2,#3)(#4,#5)(#6,#7)(#8,#9)

$\backslash\mathbf{XPOINT}n(x_1,\ y_1)(x_2,\ y_2)(x_3,\ y_3)(x_4,\ y_4)$

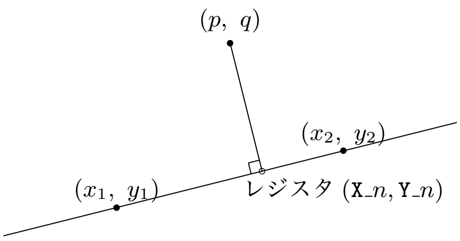
#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。直線 (#2, #3)-(#4, #5) と直線 (#6, #7)-(#8, #9) の交点の座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と \Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。



\PLFOOT#1(#2,#3)(#4,#5)(#6,#7)[2]

$\backslash\mathbf{PLFOOT}n(p,\ q)(x_1,\ y_1)(x_2,\ y_2)$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。点 (#2, #3) から直線 (#4, #5)-(#6, #7) へ下ろした垂線の足の座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と \Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。



\PLHEAD#1(#2,#3)(#4,#5)

$\backslash\mathbf{PLHEAD}n(x_1,\ y_1)(x_2,\ y_2)$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。有向線分 (#2, #3)-(#4, #5) の終点から延びる、有向線分に対し 90° で線分と同じ長さの垂線の頭（終点）の座標を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（\X_n と \Y_n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。

[2] マクロ名は垂線 (perpendicular line) の足 (foot) より。

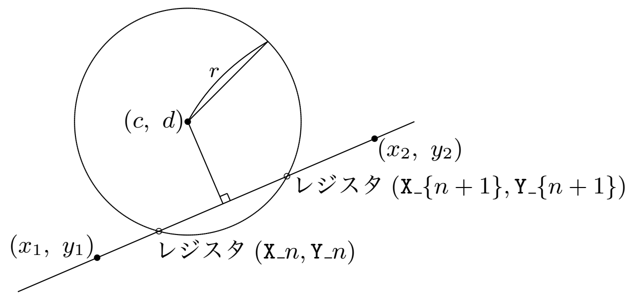
>>>座標上の 2 点の値を保持

\IOCROSSES#1(#2,#3)(#4,#5)(#6,#7,#8)

\IOCROSSES$n(x_1,\ y_1)(x_2,\ y_2)(c,\ d,\ r)$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。直線 (#2, #3)-(#4, #5) と、中心 (#6, #7)、半径#8 の円との 2 つの交点の座標を計算し、その値をレジスタに保持する。値は n 番目と $n+1$ 番目のレジスタ（\X_n と\Y_n）および（\X_{$n+1$} と\Y_{$n+1$}）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。また、n が 4 のときは、$n+1$ 番目のレジスタは 1 番目のレジスタである。

2 つの交点のうち n 番目のレジスタに保持されるのは、垂線の足の x 座標に対して左側にある点である。右側の点は $n+1$ 番目のレジスタに保持される。2 つの交点の x 座標が同じなら、n 番目のレジスタに保持されるのは、垂線の足の y 座標に対して下側にある点になる。

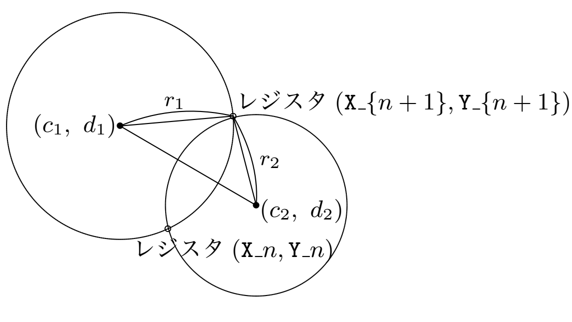


\OOCROSSES#1(#2,#3,#4)(#5,#6,#7)

\OOCROSSES$n(c_1,\ d_1,\ r_1)(c_2,\ d_2,\ r_2)$

#1（レジスタ番号）は 1, 2, 3, 4 のいずれかを必ず指定する。中心 (#2, #3)、半径#4 の円と、中心 (#5, #6)、半径#7 の円との 2 つの交点の座標を計算し、その値をレジスタに保持する。値は n 番目と $n+1$ 番目のレジスタ（\X_n と\Y_n）および（\X_{$n+1$} と\Y_{$n+1$}）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照）。また、n が 4 のときは、$n+1$ 番目のレジスタは 1 番目のレジスタである。

2 つの交点のうち n 番目のレジスタに保持されるのは、円 $(c_1,\ d_1)$ の中心から円 $(c_2,\ d_2)$ の中心へ向かう線分に対して、円 $(c_1,\ d_1)$ の円周に沿って右側の点である。左側の点は $n+1$ 番目のレジスタに保持される。



>>>1 値を保持

$\boxed{\textbf{\textbackslash DISTANCE\#1(\#2,\#3)(\#4,\#5)}}$ $\qquad$ **\DISTANCE**$n(x_1,\ y_1)(x_2,\ y_2)$

`#1`（レジスタ番号）は 1, 2 のいずれかを必ず指定する。2 点 (`#2`, `#3`)、(`#4`, `#5`) 間の距離を計算し、その値をレジスタに保持する。値は n 番目のレジスタ（`\D_`n）に保持され、次に値保持のマクロが実行されるまで保たれる（→p.32「レジスタに関する注意」参照)。他のマクロが 2 つの値を保持するのに対し、このマクロは 1 つの値だけを保持することに注意されたい。

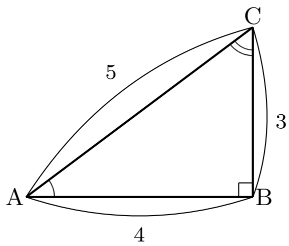
▬レジスタに関する注意

値保持のマクロは`\DISTANCE` を除いて、値を保持するレジスタが共通である。具体的には、レジスタ番号 n で実行したマクロの値は、`\X_`n、`\Y_`n に保持される。したがって、基本的に同じレジスタ番号のマクロが実行されなければ値は保たれるのだが、`\IOCROSSES` と`\OOCROSSES` は n と $n+1$ のレジスタに値を保持するので、`\X_`$\{n+1\}$、`\Y_`$\{n+1\}$ に保持した値は上書きされる。保持した値を使うときは、なるべく保持した直後に利用するのがよいだろう。

また、値を保持するレジスタは`\X_`n, `\Y_`n であると説明したが、実際は`\@Xa`, `\@Ya`, . . . , `\@Xd`, `\@Yd` に保持される。`\X_`n, `\Y_`n は保持した値を取り出すマクロで、`\@Xa` が`\X_1` に、`\@Ya` が`\Y_1` に、. . . `\@Xd` が`\X_4` に、`\@Yd` が`\Y_4` に対応している。必要なら`\@Xe`, `\@Ye` 等を用意して、`\ifcase` 文に追加するとよい。また、大文字が煩わしいと思えば小文字にしてもかまわない。

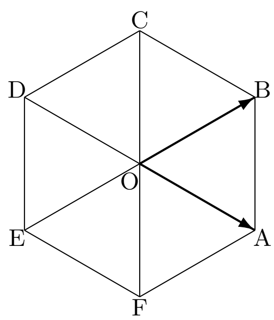
▽直角三角形にまつわる様々な表示

```
\begin{drawpict}[.75cm](4, 4)
  {\thicklines \poooly(0, 0)(4, 0)(4, 3) % 直角三角形
  \vparade[.2]ABC{};{180}{0}{90}0} % \vparade は{\thicklines }内で実行!
  \slur+(0, 0)(4, 0){$4$}
  \slur+(4, 0)(4, 3){$3$}
  \slur+(4, 3)(0, 0){\makebox(0, 0)[t]{$5$}}
  \arcdegree(0, 0, .5)(0, 37) % 角 A
  \putsymbol[r]{.25}(4, 0)(4, 3)0 % 角 B
  \arcdegree(4, 3, .5)(-143, -90) \arcdegree(4, 3, .4)(-143, -90) % 角 C
\end{drawpict}
```



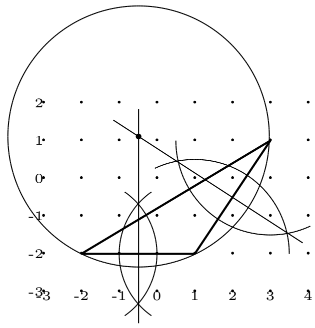
▽正六角形とベクトル

```
\begin{drawpict}[.5pt](200, 200)(-100, -100)
  \siiiide(0, -100)(86.6, -50)(86.6, 50)(0, 100) % 正六角形
  \vparade[8]ABC{};{-45}{45}{90}0 % 頂点
  \siiiide(0, 100)(-86.6, 50)(-86.6, -50)(0, -100)
  \vparade[8]DEF{};{135}{-135}{-90}0
  \siide(0, -100)(0, 100)
  \siide(86.6, -50)(-86.6, 50)
  \siide(86.6, 50)(-86.6, -50)
  \vertex(0, 0, 15){-120}O
  \thicklines
  \vsiide(0, 0)(86.6, -50)
  \vsiide(0, 0)(86.6, 50)
\end{drawpict}
```



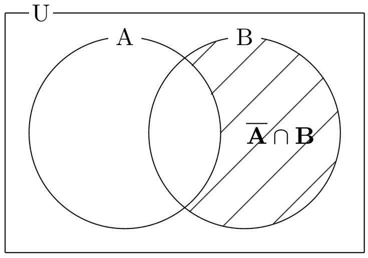
▽外接円の作図

```
\begin{drawpict}[.5cm](6, 5.5)(-3, -1.8)
  \dotsgrid1(-3, -3)(5, 3)
  {\thicklines \poooly(-2, -2)(1, -2)(3, 1)} % 三角形
  \arcdegree(-2, -2, 2)(-55, 55) \arcdegree(1, -2, 2)(125, 235) % 作図線
  \OOCROSSES1(1, -2, 2)(-2, -2, 2) % 垂直二等分線の座標
  \extendline{2.5}(\X_1, \Y_1)(\X_2, \Y_2){.5} % 垂直二等分線
  \arcdegree(1, -2, 2.5)(0, 115) \arcdegree(3, 1, 2.5)(180, 295)
  \OOCROSSES3(1, -2, 2.5)(3, 1, 2.5)
  \extendline{.5}(\X_3, \Y_3)(\X_4, \Y_4)2
  \XPOINT1(\X_1, \Y_1)(\X_2, \Y_2)(\X_3, \Y_3)(\X_4, \Y_4)
  \put(\X_1, \Y_1){\circle*{.15}}
  \DISTANCE1(\X_1, \Y_1)(-2, -2) % 外心と三角形の頂点 (-2, -2) 間の距離計算
  \arcdegree(\X_1, \Y_1, \D_1)(0, 360) % 外接円
\end{drawpict}
```



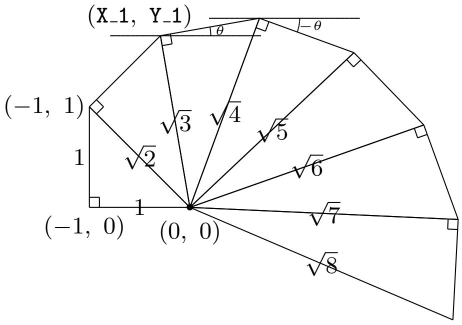
▽集合とベン図

```
\begin{drawpict}[.9pt](150, 100)
  \siiiide(0, 0)(150, 0)(150, 100)(20, 100)
  \siiide(0, 0)(0, 100)(10, 100)
  \arcdegree(50, 50, 40)(100, 440)
  \arcdegree(100, 50, 40)(100, 440)
  \apeeex(15, 100)U(50, 90)A(100, 90)B
  \NEobliqs{0}(78, 88)(86, 109)(90, 123)(77, 133) % マクロ\NEobliqs を利用
  \NEobliqs{-80}(91, 139)(112, 138)(0, 0)(0, 0)
  \apex(115 ,50){\bf$\overline{\textrm A}\cap{\textrm B}$}
\end{drawpict}
```



▽ $\sqrt{2}$, $\sqrt{3}$, $\sqrt{4}$, ... の作図

```
\begin{drawpict}[1.33cm](4, 3.5)(-1, -1.5)
  \pooooly(0, 0)(-1, 0)(-1, 1) \putsymbol[r]{.1}(-1, 0)(0, 0)0
  \apeeex(-.5, 0){$1$}(-1.1, .5){$1$}(-.5, .5){$\sqrt{2}$}
  %
  \ORBIT1(-1, 1, 1){45} % 次の頂点 (\X_1, \Y_1) 計算
  \pooooly(0, 0)(-1, 1)(\X_1, \Y_1) \putsymbol[r]{.1}(-1, 1)(0, 0)0
  \DIVIDE2(0, 0)(\X_1, \Y_1){.5} \apex(\X_2, \Y_2){$\sqrt{3}$}
  %
  \ORBIT2(\X_1, \Y_1, 1){10} % 次の頂点 (\X_2, \Y_2) 計算
  \pooooly(0, 0)(\X_1, \Y_1)(\X_2, \Y_2) \putsymbol[r]{.1}(\X_1, \Y_1)(0, 0)0
  \DIVIDE3(0, 0)(\X_2, \Y_2){.5} \apex(\X_3, \Y_3){$\sqrt{4}$}
  %
  \ORBIT3(\X_2, \Y_2, 1){-20} % 次の頂点 (\X_3, \Y_3) 計算
  \pooooly(0, 0)(\X_2, \Y_2)(\X_3, \Y_3) \putsymbol[r]{.1}(\X_2, \Y_2)(0, 0)0
  \DIVIDE1(0, 0)(\X_3, \Y_3){.5} \apex(\X_1, \Y_1){$\sqrt{5}$}
  %
  \ORBIT1(\X_3, \Y_3, 1){-46}
  \pooooly(0, 0)(\X_3, \Y_3)(\X_1, \Y_1) \putsymbol[r]{.1}(\X_3, \Y_3)(0, 0)0
  \DIVIDE2(0, 0)(\X_1, \Y_1){.5} \apex(\X_2, \Y_2){$\sqrt{6}$}
  %
  \ORBIT2(\X_1, \Y_1, 1){-70}
  \pooooly(0, 0)(\X_1, \Y_1)(\X_2, \Y_2) \putsymbol[r]{.1}(\X_1, \Y_1)(0, 0)0
  \DIVIDE3(0, 0)(\X_2, \Y_2){.5} \apex(\X_3, \Y_3){$\sqrt{7}$}
  %
  \ORBIT3(\X_2, \Y_2, 1){-93}
  \pooooly(0, 0)(\X_2, \Y_2)(\X_3, \Y_3) \putsymbol[r]{.1}(\X_2, \Y_2)(0, 0)0
  \DIVIDE1(0, 0)(\X_3, \Y_3){.5} \apex(\X_1, \Y_1){$\sqrt{8}$}
\end{drawpict}
```



`\ORBIT` に与えた角度は、1 の長さの辺が直角の頂点を通る水平線となす角 θ である（図を参照）。θ は 90° から順に、$\tan^{-1}\frac{1}{1}$, $\tan^{-1}\frac{1}{\sqrt{2}}$, $\tan^{-1}\frac{1}{\sqrt{3}}$, ... を引いた値になっている。$\theta$ の計算—すなわち $\tan^{-1}$ の計算—は実際に Microsoft Excel で計算したのだが、それなりの手間には違いない。

また、\ORBIR1 で計算された値が、次の直角頂点の座標 (X_1, Y_1) の値に使われている。続いて \DIVIDE2 で計算された値は、\apex(\X_2，\Y_2) で辺の長さを表示する位置の値に使われるが、\DIVIDE1 で計算しなかったのは、まだその後で\X_1、\Y_1 の値を使うからである。\DIVIDE1 で計算してしまうと\X_1、\Y_1 の値が変わってしまう。

▽三角形の外接円・内接円のマクロ

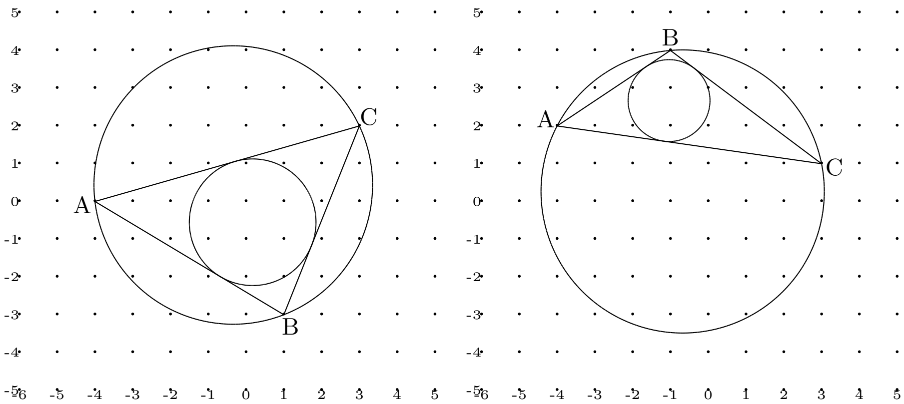
```
\def\TriangleABC(#1,#2)(#3,#4)(#5,#6){%
  \pooly(#1,#2)(#3,#4)(#5,#6)
  %OuterCircle
  \DIVIDE1(#1,#2)(#3,#4){.5}\PLHEAD2(#1,#2)(\X_1,\Y_1)%中点 1"からの垂線の頭 2"
  \DIVIDE3(#3,#4)(#5,#6){.5}\PLHEAD4(#3,#4)(\X_3,\Y_3)%中点 3"からの垂線の頭 4"
  \XPOINT1(\X_1,\Y_1)(\X_2,\Y_2)(\X_3,\Y_3)(\X_4,\Y_4)%直線 1"-2",3"-4"の交点 1"
  \DISTANCE1(\X_1,\Y_1)(#3,#4)%交点 1"と三角形の頂点との距離 1"
  \arcdegree(\X_1,\Y_1,\D_1)(0,360)%交点 1"、距離 1"（半径）の円
  %InnerCircle
  \CUTLINE1(#1,#2)(#3,#4)1%頂点から長さ 1 の位置 1"
  \CUTLINE2(#1,#2)(#5,#6)1%頂点から長さ 1 の位置 2"
  \DIVIDE2(\X_1,\Y_1)(\X_2,\Y_2){.5}%位置 1"、位置 2"の中点 2"
  \CUTLINE3(#3,#4)(#1,#2)1%頂点から長さ 1 の位置 3"
  \CUTLINE4(#3,#4)(#5,#6)1%頂点から長さ 1 の位置 4"
  \DIVIDE4(\X_3,\Y_3)(\X_4,\Y_4){.5}%位置 3"、位置 4"の中点 4"
  \XPOINT1(#1,#2)(\X_2,\Y_2)(#3,#4)(\X_4,\Y_4)%2 つの頂角の 2 等分線の交点 1"
  \PLFOOT2(\X_1,\Y_1)(#1,#2)(#3,#4)%交点 1"からの垂線の足 2"
  \DISTANCE1(\X_1,\Y_1)(\X_2,\Y_2)%交点 1"と垂線の足 2"との距離 1"
  \arcdegree(\X_1,\Y_1,\D_1)(0,360)%交点 1"、距離 1"（半径）の円
}
```

たとえばこのマクロは、任意の三角形の外接円と内接円を同時に描画するものである。外接円は各辺の垂直二等分線の交点を中心とする円、内接円は各角の二等分線の交点を中心をする円である。定木とコンパスによる作図を再現するなら、\OOCROSSES で 2 円の交点を保持しながら描画するのがよい。ただし、\OOCROSSES と\IOCROSSES は回転角を 1° 刻みで計算するため誤差が大きい。p.34 に示した「外接円の作図」は、\OOCROSSES で計算した割にはマシな描画になっているが、内接円を作図するとかなりのずれが生じてしまう。

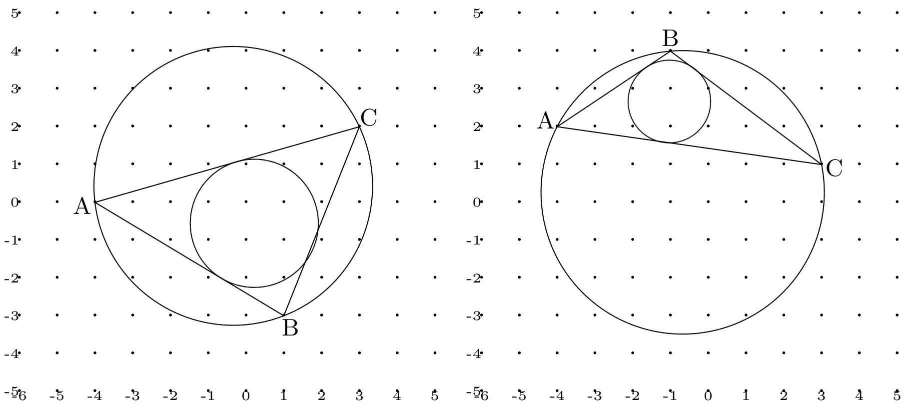
外接円や内接円は、要するに辺や角の中点を求めればよいので、上のマクロでは\CUTLINE や \DIVIDE などで交点を保持ながら描画している。このようにすれば、内接円もそこそこ見られる描画になる。このマクロを実際に利用する場合は、このまま利用したい文書に貼付ければよい。その上で\TriangleABC に引数を与えれば、いつでも外接円と内接円を手軽に描画できることになる。\TriangleABC の後ろで実行される\vparade は、マクロ内の\pooly 以降に\siide 系等のマクロがないため、正しく機能することに注意されたい。

以下に描画結果を示す。最初の 2 例が上のマクロで描画した外接円と内接円である。次の 2 例は内接円の描画の際、\DIVIDE を\OOCROSSES に変更したものである。左の描画は明らかにずれが分かる。右の描画は注意して見る必要があるが、内接円がわずかに辺 AC を越えている。

```
\begin{drawpict}[.5cm](24, 10)(-6, -5)
  \dotsgrid1(-6, -5)(6, 6)
  \TriangleABC(-4, 0)(1, -3)(3, 2) \vparade[.35]ABC{};{-165}{-60}{45}0
  %
  \baseskip{12}
  \dotsgrid1(-6, -5)(6, 6)
  \TriangleABC(-4, 2)(-1, 4)(3, 1) \vparade[.35]ABC{};{150}{90}{-15}0
\end{drawpict}
```



```
\def\TrybungleABC(#1,#2)(#3,#4)(#5,#6){%マクロ定義\TriangleABCを一部変更して、
  :（同上マクロ）                        %同じ三角形で外接円と内接円を描画した。
  %InnerCircle
  :（同上マクロ）
  \OOCROSSES2(\X_1,\Y_1, 2)(\X_2,\Y_2, 2)%\DIVIDE2を変更!
  :（同上マクロ）
  \OOCROSSES4(\X_3,\Y_3, 2)(\X_4,\Y_4, 2)%\DIVIDE4を変更!
  :（同上マクロ）
}
```



覚え書き

〓変数の衝突

各マクロの変数は global に定義して共通に使っているものが多い。マクロを定義する際に見落としがあれば、衝突の可能性がある。変数の衝突のせいで不可解な動作になっていたら、変数名を適切に変更してもらいたい。

〓描画精度

各曲線は微小区間に\qbezier で直線を描き、それをつないでいる。easyfx は描画精度を、実線のときは\@steps=0.05\p@に、破線のときは\@steps=0.02\p@に決めている。精度を変更する場合は、tmtmath.sty ファイルの値を直接変更してもよいが、一時的な描画精度の変更なら p.20 の drawfunc 環境を使えばよいだろう。ただし、除算計算ルーティンの精度は 0.00001 で打ち切っているし、三角比・対数表の精度も 0.0001 であるため、描画精度をそれより小さくしても無駄である。

〓空白の制御

TEX では "改行" は "ひとつの空白" に相当する。tmtmath.sty を記述するに当たっては、見やすさを優先して至る所に改行を入れている。そのため、不必要な空白が出力されることがあるが、通常、図やグラフを描画する際には問題となることはないだろう。もし、図やグラフの描画で予期しないズレが生じたら、不必要な空白が出力されたことを疑ってよい。ほとんどの空白は、行末に%を入れることで出力されなくなる。

〓メモリ?不足

使用環境にもよるが、ソーステキストが処理できないときは、図形描画の量が多いためということもある。また、マクロに与える値が大きいのかもしれない。実際、\XPOINT$n$ は大きな値を指定するとオーバーフローを起こす。tmtmath.sty が適切なレジスタ管理をしていないことが問題なのだが、図形の量やマクロに与える値はほどほどにしておいた方がよい。

〓忘備録

正確なマクロ名や引数の与え方などは忘れやすいので、忘備用に\man マクロを組み入れてある。\man と書いて処理すれば、簡易マニュアルとして次ページの表が表示される。

\baseskip{x}	x だけ右へ移動（負の値なら左）
\coordinate[ax](x_l, x_r)(y_b, y_t)	$ax \to$ (A, R, I, d, r)+(g)
	$ax \to$ d or r では $1x = 30°$, $1y = 0.5$
\putone, \putones	1 目盛表示
\locate[$circ$](x, y)	$circ \to$ (b, w, B, W)+(x)
\linexis{a}{op}(y_b, y_t)	$x = a$（op $\to$: で破線）
\easyfx[RPN]{op}(x_l, x_r)	$y =$ [RPN で記述した関数]（op $\to$: で破線）
\sindegree[a, b, c, d](x_l, x_r)	$y = a\sin(bx + c) + d$;　c, x_l, x_r は整数度
	b $\to$ (1, 2, 3, 4, 0.5, 0.33, 0.25)
\tandegree[a, b, c, d](x_l, x_r)	$y = a\tan(bx + c) + d$;　c, x_l, x_r は整数度
	b $\to$ (1, 2, 3, 4, 0.5, 0.33, 0.25)

\abline[a, b]{op}(x_l, x_r)	$y = ax + b$（op $\to$: で破線）
\parabola[a, b, c](x_l, x_r)	$y = ax^2 + bx + c$
\cubecurve[a, b, c, d](x_l, x_r)	$y = ax^3 + bx^2 + cx + d$
\fraccurve[a, b, c, d](x_l, x_r)	$y = \dfrac{a}{bx + c} + d$
\sqrtcurve[a, b, c, d](x_l, x_r)	$y = a\sqrt{bx + c} + d$
\expcurve{op}[a, b, c, d](x_l, x_r)	$y = a \cdot b^{cx} + d$（指数および対数関数）
	op $\to$ - で $y = a \cdot b^{cx} + d$ $[x_l,\ x_r]$ の逆関数
\hyperoval{op}[a, b, C, D](x_l, x_r)	$\dfrac{(x - a)^2}{C} + \dfrac{(y - b)^2}{D} = 1$
	op $\to$ (なし) or + or -
	$C,\ D > 0 \to$ 楕円;　$C * D < 0 \to$ 双曲線

\unitcircle[t-ax]	t-ax $\to$ t
\rangeline(l, r){ax}	l から r までの数直線;　$ax \to$ 数直線の名称
\brange{n}(s, e){v}	始点 s（黒丸）から e まで伸びる数直線線分
	$n \to$ 0, 1, 2, 3, 4;　$v \to$ 丸位置の値
\wrange{n}(s, e){v}	同上（白丸）

\dotsgrid{d}(x_1, y_1)(x_2, y_2)	d 間隔の格子点を左下座標-右上座標まで描画
\siiide(x_1, y_1)(x_2, y_2)(x_3, y_3)	連鎖線分の描画　(他に \siide、\siiiide)
\vsiide(x_s, y_s)(x_e, y_e)	2 点間の線分ベクトルの描画
\poooly(x_1, y_1)(x_2, y_2)(x_3, y_3)	三角形の描画　(四角形は \pooooly)
\apeeex(x_1, y_1)X(x_2, y_2)Y(x_3, y_3)Z	頂点 X, Y, Z の表示　(他に \apex、\apeex)
\vertex(c, d, r){θ}X	中心 (c, d)、半径 r を θ 回転した位置に X を表示
\slur{op}(x_1, y_1)(x_2, y_2)X	2 点間にスラー線を引き X を表示
	op $\to$ + or -
\arcdegree{op}(c, d, r)(α, β)	中心 (c, d)、半径 r、整数度 $[\alpha°,\ \beta°]$ の弧（$\alpha < \beta$）
	op $\to$ + or - or *
\extendline{l}(x_l, y_l)(x_r, y_r){r}	線分 $(x_l,\ y_l)$-$(x_r,\ y_r)$ の左右に長さ l、r 延ばした線分
\putsymbol[sym]l(x_1, y_1)(x_2, y_2)t	2 点 $(x_1,\ y_1)$、$(x_2,\ y_2)$ の t 分点に線長 l の記号類
	$sym \to$ r（直角） or e/E（等辺） or p/P（平行）
\vparade[r]$V_1V_2V_3V_4$;$\theta_1\theta_2\theta_3\theta_4$	主に頂点名称をまとめて表示

\ORBIT$n$(x, y, r)θ	1 点からの半径 r、角 θ の点 (\X_n, \Y_n) を計算
\DIVIDE$n$(x_1, y_1)(x_2, y_2)t	2 点の $t : (1 - t)$ 分点 (\X_n, \Y_n) を計算
\CUTLINE$n$(x_1, y_1)(x_2, y_2)l	線分上の長さ l の終点 (\X_n, \Y_n) を計算
\XPOINT$n$(x_1, y_1)(x_2, y_2)(x_3, y_3)(x_4, y_4)	2 線分の交点 (\X_n, \Y_n) を計算
\PLFOOT$n$(x_1, y_1)(x_2, y_2)(x_3, y_3)	垂線の足 (\X_n, \Y_n) を計算
\PLHEAD$n$(x_1, y_1)(x_2, y_2)	垂線の頭 (\X_n, \Y_n) を計算
\IOCROSSES$n$(x_1, y_1)(x_2, y_2)(c, d, r)	直線と円の交点を計算
\OOCROSSES$n$(c_1, d_1, r_1)(c_2, d_2, r_2)	円と円の交点を計算
	以上、$n \to$ 1 or 2 or 3 or 4
\DISTANCE$n$(x_1, y_1)(x_2, y_2)	2 点間の距離 \D_n を計算
	$n \to$ 1 or 2

appendix

TEX の picture 環境だけでグラフや図を描くのは大変で、必ずしも思い通りの描画ができるわけではない。しかし、ある程度それが実現すると、「他にこんな機能も...」という欲求が生じるものである。その場合は、他のマクロ集を使うべきであろう。この tmtmath.sty は、あくまでもこれひとつで描画を支援するものなので、手書き以上に手間がかかることはしていない。実際私は、TEX での描画が厳しいときは手書きしている。領域を示す斜線などはその類いになる。とは言うものの、領域を示す斜線を描画するマクロも定義してみた。もちろん、出来が今ひとつで使い勝手も良くないので、不要なオマケ機能と考えてもらえばよいだろう。

$\boxed{\textbf{\textbackslash neobliqs\#1(\#2,\#3)(\#4,\#5)(\#6,\#7)(\#8,\#9)}}$

$$\textbf{\textbackslash neobliqs}\{y_b\}(x_{1l},\ x_{1r})(x_{2l},\ x_{2r})(x_{3l},\ x_{3r})(x_{4l},\ x_{4r})$$

#1 を y 切片にもつ右上がり（傾き 1）の斜線から始め、y 切片を 0.5 ずつ下げながら、4 本の斜線を引く。

$\boxed{\textbf{\textbackslash seobliqs\#1(\#2,\#3)(\#4,\#5)(\#6,\#7)(\#8,\#9)}}$

$$\textbf{\textbackslash seobliqs}\{y_b\}(x_{1l},\ x_{1r})(x_{2l},\ x_{2r})(x_{3l},\ x_{3r})(x_{4l},\ x_{4r})$$

#1 を y 切片にもつ右下がり（傾き -1）の斜線から始め、y 切片を 0.5 ずつ上げながら、4 本の斜線を引く。

$\boxed{\textbf{\textbackslash NEobliqs\#1(\#2,\#3)(\#4,\#5)(\#6,\#7)(\#8,\#9)}}$

$$\textbf{\textbackslash NEobliqs}\{y_b\}(x_{1l},\ x_{1r})(x_{2l},\ x_{2r})(x_{3l},\ x_{3r})(x_{4l},\ x_{4r})$$

#1 を y 切片にもつ右上がり（傾き 1）の斜線から始め、y 切片を 20 ずつ下げながら、4 本の斜線を引く。

$\boxed{\textbf{\textbackslash SEobliqs\#1(\#2,\#3)(\#4,\#5)(\#6,\#7)(\#8,\#9)}}$

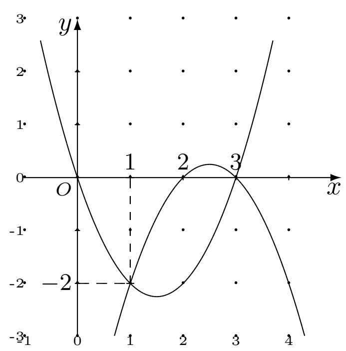
$$\textbf{\textbackslash SEobliqs}\{y_b\}(x_{1l},\ x_{1r})(x_{2l},\ x_{2r})(x_{3l},\ x_{3r})(x_{4l},\ x_{4r})$$

#1 を y 切片にもつ右下がり（傾き -1）の斜線から始め、y 切片を 20 ずつ上げながら、4 本の斜線を引く。

※ 斜線は順に、x の区間 (#2, #3), (#4, #5), (#6, #7), (#8, #9) に引かれる。また、斜線は左から右へ向けて引くことにしている。なぜなら区間を $(x_l,\ x_r)$ としたとき、$x_l < x_r$ である方が自然だと思うからだ。その理由から、右上がりの斜線は順に y 切片が下がるように、右下がりの斜線は順に y 切片が上がるように描画されるのである。

$y = x^2 - 3x$ と $y = -x^2 + 5x - 6$ で囲まれる部分に斜線を引く場合を例にとる。まず、斜線を引く目安のために`\dotsgrid`で格子点を描画しておく（座標があるので必要ないかもしれないが、図形を描画する際は必要だろう）。

```
\begin{drawpict}[.7cm](6, 6)(-1, -3)
  \coordinate[R](-1, 5)(-3, 3)
  \easyfx[xx*3x*-](-.7, 3.7)
  \easyfx[{-1}x*x*5x*+6-](.7, 4.3)
  \locate[x](1, -2)
  \apeeex(1, .3){$1$}(2, .3){$2$}(3, .3){$3$}
  \apex(-.4, -2){$-2$}
  %
  \dotsgrid1(-1, -3)(5, 4) % 格子点
\end{drawpict}
```



右下がりの斜線を引くとすると、最初の斜線の y 切片は -0.5 である。そこで`\seobliqs{-.5}`とし、x の区間はとりあえず `(0, 0)` としておく。斜線が 4 本では足りないので、`\seobliqs{1.5}`も加える（右下がりの場合、次の y 切片は 2 上である—右上がりの場合は 2 下）。

```
\begin{drawpict}[.7cm](6, 6)(-1, -3)}
  \coordinate[R](-1, 5)(-3, 3)
  \easyfx[xx*3x*-](-.7, 3.7)
  \easyfx[{-1}x*x*5x*+6-](.7, 4.3)
  \locate[x](1, -2)
  \apeeex(1, .3){$1$}(2, .3){$2$}(3, .3){$3$}
  \apex(-.4, -2){$-2$}
  %
  \dotsgrid1(-1, -3)(5, 4)
  \seobliqs{-.5}(0, 0)(0, 0)(0, 0)(0, 0)
  \seobliqs{1.5}(0, 0)(0, 0)(0, 0)(0, 0)
\end{drawpict}
```

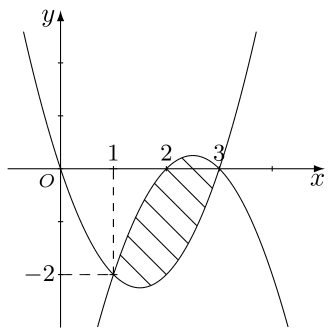
斜線を引く区間の値は、図を見ながらおよその目安で入れてもよいのだが、幸いグラフが 2 次関数なので、交点は楽に計算できる。斜線の左側は $y = -x^2 + 5x - 6$ と斜線の交点になるので、斜線を $y = -x + b$ とおいて $-x^2 + 5x - 6 = -x + b$ を解けばよい。解の公式より $\pm$ の解が求まるが、左側の交点を採用するので $x = 3 - \sqrt{3 - b}$ が交点になる。この b に y 切片である $-0.5, 0, 0.5, 1, \ldots, 3$ まで代入すれば、8 本の斜線の始点（x_l 座標）が分かる。計算は、電卓か表計算ソフトでするとよいだろう。その結果、以下のように左側の値が決まる（斜線は 7 本でよいので 8 本目の値は `0` のままにする）。

```
  \seobliqs{-.5}(1.12, 0)(1.26, 0)(1.41, 0)(1.58, 0)
  \seobliqs{1.5}(1.77, 0)(2, 0)(2.29, 0)(0, 0)
```

次は斜線の終点（x_r 座標）である。こちらは、$y = x^2 - 3x$ と $y = -x + b$ の右側の交点を採用し $x = 1 + \sqrt{1+b}$ が交点になる。同様に、b に $-0.5, 0, 0.5, 1, \ldots, 3$ まで代入する。

```
\seobliqs{-.5}(1.12, 1.7)(1.26, 2)(1.41, 2.22)(1.58, 2.41)
\seobliqs{1.5}(1.77, 2.58)(2, 2.73)(2.29, 2.87)(0, 0)
```

これで`\dotsgrid`を削除すれば出来上がりである。




```
\begin{drawpict}[.7cm](6, 6)(-1, -3)
  \baseskip7
  \coordinate[R](-1, 5)(-3, 3)
  \easyfx[xx*3x*-](-.7, 3.7)
  \easyfx[{-1}x*x*5x*+6-](.7, 4.3)
  \locate[x](1, -2)
  \apeeex(1, .3){$1$}(2, .3){$2$}(3, .3){$3$}
  \apex(-.4, -2){$-2$}
  %
  \seobliqs{-.5}(1.12, 1.7)(1.26, 2)(1.41, 2.22)(1.58, 2.41)
  \seobliqs{1.5}(1.77, 2.58)(2, 2.73)(2.29, 2.87)(0, 0)
\end{drawpict}
```

斜線の始点と終点の x 座標がすぐに分かれば実用になるだろうが、明らかに手書きするほうが速やかである。斜線の間隔は、描画単位が cm なら`\neobliqs`か`\seobliqs`を使って 0.5 に、描画単位が pt なら`\NEobliqs`か`\SEobliqs`を使って 20 になるように決めてある。変更は`tmtmath.sty`ファイルで行ってもらいたい。