

TRACÉS DE COURBES EN \LaTeX AVEC TikZ

Pour tracer les courbes avec TikZ, il faut mettre la directive `\usepackage{tikz}` dans le préambule et placer les commandes de TikZ dans l'environnement `\begin{tikzpicture} ... \end{tikzpicture}`. Pour utiliser des couleurs, on utilise `\usepackage{xcolor}`. Il y a peu de couleurs prédéfinies. On peut les déclarer en \LaTeX par la commande `\definecolor{nomcouleur}{rgb}{rouge,vert,bleu}` ou rouge vert bleu sont des codes compris entre 0 et 1. On peut télécharger ici un petit fichier [couleurs.sty](http://fr.wikipedia.org/wiki/Liste_de_couleurs) avec plus de 400 couleurs. Ces couleurs et les codes correspondants ont été pris sur la page de Wikipedia : http://fr.wikipedia.org/wiki/Liste_de_couleurs.

1 TRACÉ DES AXES

► Pour tracer l'axe des x , la commande `\draw` suffit, par exemple : `\draw[>](-6,0)--(6,0) node [below] {x};` Pour un déplacement relatif, on utilise `--++` au lieu de `--`. L'unité par défaut est le centimètre et pour les angles le degré (entre -360° et 720°). La commande `node` permet de placer un « *noeud* » à un endroit, noeud qui peut contenir du texte \LaTeX . Les options éventuelles de toutes ces commandes s'insèrent entre `[..]`.



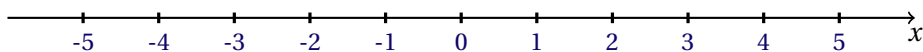
► L'épaisseur du tracé se gère par les options `thin` `very thin` `ultra thin` `thick` `very thick` `ultra thick`

On peut aussi personnaliser par exemple : `line width=5pt`

► La couleur du tracé est obtenue par la directive `color=nomcouleur` dans les options de `\draw`.

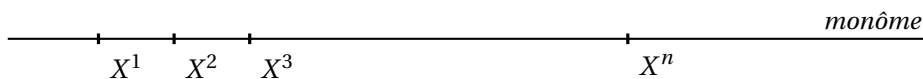
► Le style du tracé (pointillés, tirets, ...) `dotted` `loosely dotted` `densely dotted` `dashed` `loosely dashed` `densely dashed` `double` `double distance = 5pt`

► Pour obtenir des *graduations*, on utilise la commande de boucle `\foreach`, par exemple :



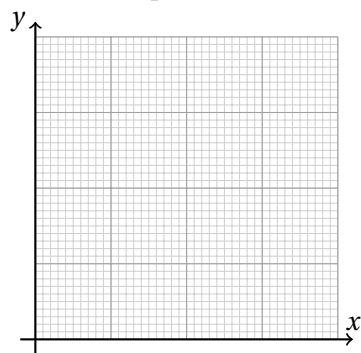
```
\draw [thick, >] (-6,0)--(6,0) node [below] {$x$};
\foreach \x in { -5, ...,5 }
  \draw [very thick] (\x,2pt)--(\x,-2pt) node [below, bleunuit] {\small\x};
```

► Il est tout à fait possible de mettre des valeurs qui ne sont pas « *en accord* » avec la graduation. On utilise `\foreach \i/\j in { 1,...,3, 8/n}` où l'ensemble des valeurs est de la forme *valeur*_i/*valeur*_j. Si *valeur*_j=*valeur*_i, on peut l'omettre :



```
\draw [thick, >] (-0.2,0)--(12,0) node [above left] {\small\textit{monôme}};
\foreach \i/\j in { 1,...,3, 8/n}
  \draw [very thick](\i,2pt)--(\i,-2pt) node [below right] {$X^{\j}$};
```

► Pour tracer un *quadrillage*, on dispose de la commande **grid** de TikZ. l'usage est `\draw (x1,y1) grid (x2,y2);`. On peut préciser usuellement un style de tracé dans les options de **\draw** et pour **grid**, on a **step=xx** qui par défaut est 1. On dispose même de **xstep ystep**. On peut réaliser du « *papier millimétré* » :



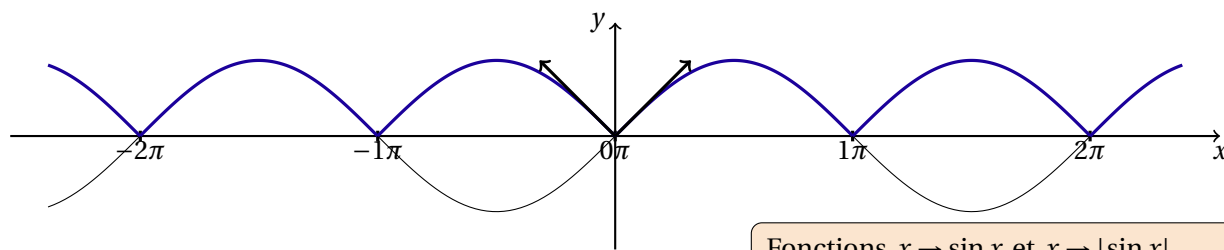
```
\draw [very thin, style=gris!20, step=0.1] (0,0) grid (4,4);
\draw [thin, gris!60] (0,0) grid (4,4);
\draw [thick, ->] (-.2,0)--(4.2,0) node [above] {$x$};
\draw [thick, ->] (0,-.2)--(0,4.2) node [left] {$y$};
```

2 TRACÉ DES COURBES

► Pour tracer une courbe, on utilise **plot**. La variable par défaut est **\x**. Pour la changer, il faut passer **variable=\t** dans les options de **draw**. Par défaut, l'intervalle où la variable prend ses valeurs est $[-5,5]$. Pour adapter cet intervalle, on utilise **domain=a:b**. Par exemple `\draw [domain=0.1:10] plot (\x, {ln(\x)})`; le `{ }` est important pour éviter des erreurs de compilations dues aux parenthèses inattendues.

► Le nombre de points utilisés pour le tracé est 25. C'est bien peu pour certaines courbes qui vont se retrouver affine par morceaux. On change cet état par **samples=nbrdepoints**.

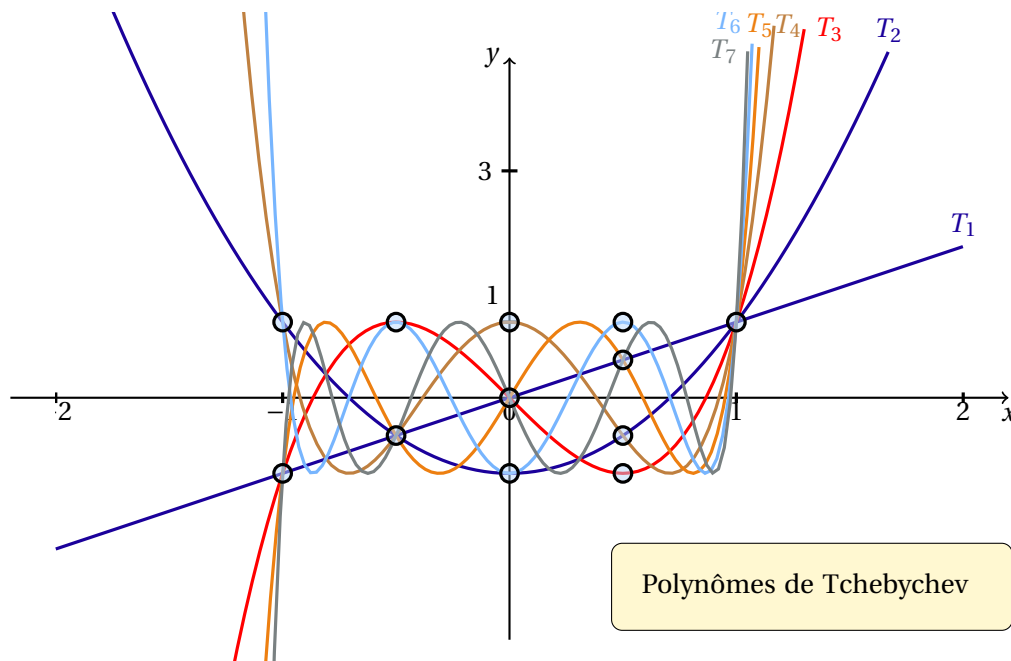
► Les fonctions prédéfinies dans TikZ sont mod, max, min, abs, exp, ln, sqrt, round, floor, ceil, sin, cos, tan, cot, sec, cosec, asin, acos, atan ainsi que la constante pi. Par défaut TikZ utilise les degrés, donc pour les commandes sin cos tan etc..., il faut alors utiliser **sin(rad(\x))** pour ramener en radians. On peut aussi utiliser **sin(\x r)** ou **sin(pi*\x/180)**. Pour l'arcsinus (ainsi que arccosinus et arctangente) « *normal* », utiliser **rad(asin(\x))**



Fonctions $x \rightarrow \sin x$ et $x \rightarrow |\sin x|$

```
\draw [thick, ->] (-8,0)--(8,0) node [below] {$x$}; \draw [thick, ->] (0,-1.5)--(0,1.5) node [left] {$y$};
\foreach \x in { -2,...,2 }
  \draw [very thick] (3.14*\x,-2pt)--(3.14*\x,2pt) node [below] {$\small \x \pi$};
\draw [domain=-7.5:7.5, samples=200] plot (\x, {sin(\x r)});
\tikzstyle {courbe} = [bleu, very thick, domain=\x:\y, samples=100];
\foreach \x/\y in {0/3.14, -6.28/-3.14, 6.28/7.5} \draw [courbe] plot (\x, {sin(\x r)});
\foreach \x/\y in {3.14/6.28, -3.14/0, -7.5/-6.28} \draw [courbe] plot (\x, {-sin(\x r)});
\tikzstyle {tangente} = [very thick, ->];
\draw [tangente] (0,0)--++(1,1); \draw [tangente] (0,0)--++(-1,1);
\draw (5,-1.5) node [text width=6cm, draw, fill=orange!20, rounded corners, inner sep=0.2cm] {Fonctions  $x \rightarrow \sin x$  et  $x \rightarrow |\sin x|$ };
```

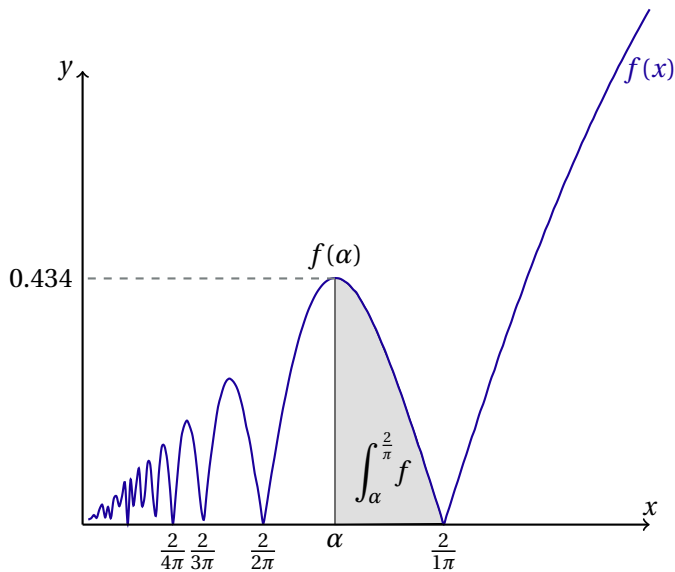
- La commande `\tikzstyle { nomstyle } [options]` permet de définir un style. c'est pratique si le même style doit être appliqué à plusieurs commandes.
- La commande d'environnement `\begin{scope}[scale=xx]` permet de faire un tracé dans une échelle qui convient mieux pour l'étude puis de mettre à une autre échelle pour l'affichage. On dispose aussi des options `xscale` `yscale` et de `rotate=xx`.
- La taille du dessin s'adapte aux commandes : si une courbe a un tracé qui dépasse beaucoup les axes, il peut être utile d'utiliser la commande `\clip` qui accepte les sous-commandes `circle` `ellipse` `rectangle`. Elle « coupe » le dessin. Elle agit uniquement à partir de la ligne où elle apparaît. Lorsque la courbe dépasse de beaucoup, TikZ affiche un message d'erreur : dimension too large. Il faut alors réduire le domaine.



```

\begin{scope}[xscale=3]
  \draw [thick, ->] (-2.2,0)--(2.2,0) node [below] {$x$}; \draw [thick, ->] (0,-3.2)--(0,4.5) node [left] {$y$};
  \clip (-2,-3.5) rectangle (2.5,5.1) ;
  \foreach \x in { -2,...,2 } \draw [very thick] (\x,-2pt)--(\x,2pt) node [below] {$\small\x$};
  \draw [very thick] (-1pt,1)--(1pt,1) node [above left=1mm] {$\small1$};
  \draw [very thick] (-1pt,3)--(1pt,3) node [left=2mm] {$\small3$};
  \draw [bleu, very thick, domain=-2:2, samples=100] plot (\x,{ \x }) node [above] {$T_1$};
  \draw [bleu, very thick, domain=-2:1.67, samples=100] plot (\x,{ 2*\x*\x-1 }) node [above] {$T_2$};
  \draw [red, very thick, domain=-2:1.3, samples=100] plot (\x,{ 4*\x*\x*\x-3*\x }) node [right] {$T_3$};
  \draw [brown, very thick, domain=-2:1.167, samples=100] plot (\x,{ 8*\x*\x*\x*\x-8*\x*\x+1 }) node [right=-1.5mm] {$T_4$};
  \draw [orange, very thick, domain=-1.5:1.1, samples=100] plot (\x,{ 16*\x*\x*\x*\x*\x-20*\x*\x*\x+5*\x }) node [above] {$T_5$};
  \foreach \x/\y in {0/0,0/-1,0/1,1/1,-1/1,-1/-1,-.5/1,-.5/-.5,.5/1,.5/.5,.5/-1,.5/-.5}
    \draw[very thick, black, fill =bleuciel!50, fill opacity=0.5] (\x,\y) ellipse (1.1 pt and 3.3pt) ;
\end{scope}
\draw (4,-2.5) node[ text width=4.5cm,draw,fill=jaunepaille!25,
  rounded corners, inner sep=0.4cm] {Polynômes de Tchebychev};

```

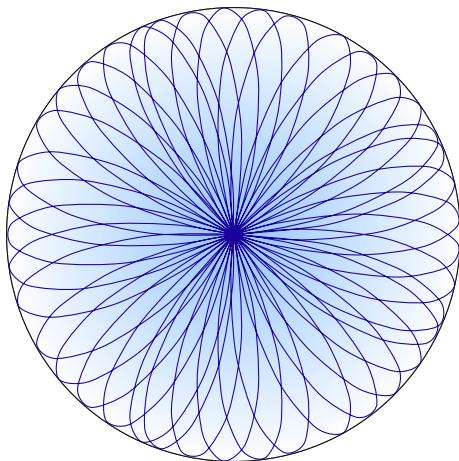


```
\begin{scope}[scale=7.5] % changement d'échelle
\draw[noir,fill=gris!20] % aire grisée
(.445,0) node [below] { $\alpha$ }
--++(0,.434) node [above] { $f(\alpha)$ }
-- plot [domain=.445:2/pi]
(\x, { \x*abs(sin(2/(rad(\x)))) } )
-- cycle;
\draw [bleu, thick, domain=0.01:1, samples=200] plot
(\x,{ \x*abs(sin(2/(rad(\x)))) }) node[below=5mm]{$f(x)$};
\foreach \i in {1,...,4} % repères sur l'axe x
\draw (2/\i*pi,0) node [below] { $\frac{2}{\i\pi}$ };
\draw[thick,->](0,0)--(1,0)node[above]{$x$};%axe x
\draw[thick,->](0,0)--(0,0.8)node[left]{$y$};%axe y
\draw (0.53,.1) % l'intégrale ...
node { $\displaystyle\int_{\alpha}^{\frac{2}{\pi}}f$ };
\draw[thick, dashed, grisplomb](.445,.434)--(0,.434)
node [noir, left] { $0.434$ };
\end{scope}
```

► Comme on le voit, on peut « enchaîner » les *tracés* et les *nodes* dans une même commande. Cela est plus efficace. la commande de tracé `--++` signifie un déplacement relatif (au lieu d'absolu). Les valeurs .445 et .434 ont été obtenus par calcul avec un logiciel extérieur.

► L'option `fill =couleur` dans un tracé fermé (ici par `--cycle`) permet de remplir la figure obtenue par la couleur désirée.

► Les coordonnées polaires dans TikZ se donnent par $(\theta : \rho)$ où θ est en degrés et ρ en centimètres par défaut. On peut ainsi placer des points ou tracer des courbes en polaires.



```
\draw (0,0) circle (3);
\shadedraw [ inner color=bleuciel, outer color=blanc,
opacity=0.8] (0,0) circle (3);
% Courbe polaire d'équation  $\rho = 3 \sin(4.1\theta)$ 
\draw [domain=0:1800, bleu, samples=300, smooth]
plot (\x : { 3*sin(4.1*\x) } );
```

► L'option `smooth` fait ce qu'elle dit et complémente l'option sur le nombre de points `samples=xx`. Comme l'angle est par défaut en degrés, on notera le domaine où $1800 = 10 \times 180$ correspond à 10π .

La commande `\shadedraw` réalise un remplissage en dégradé (ici dégradé radial). Elle est là pour le fun. Les options sont parlantes d'elles-mêmes.

► Pour tracer des courbes en paramétriques, on utilise la syntaxe `\draw[options] plot[options] ({f(\x)} , {g(\x)});`

3 TRACÉ DE COURBES AVEC GNUPLOT

3.1 INSTALLATION DE GNUPLOT POUR USAGE AVEC TIKZ

- Pour l'installer, les Linuxiens consulteront le gestionnaire de package de leur distribution tandis que les Windowsiens pourront utiliser ce lien <http://sourceforge.net/projects/gnuplot/files/>. Personnellement j'utilise la 4.42. J'ai eu quelques problèmes avec les versions supérieures sous Windows. Je n'ai pas trop cherché à comprendre et j'ai rétrogradé vers la 4.42.
- Ensuite il faut s'assurer que le chemin de l'exécutable Gnuplot est bien dans le path. Pour ce, vous ouvrez une « *invite de commandes* » sous Windows ou une « *console* » sous Linux, et vous tapez gnuplot. Si le soft ne se lance pas, consultez la documentation de votre OS pour rajouter le chemin de l'exécutable dans le path.
- Il faut ensuite indiquer au compilateur T_EX d'utiliser un programme extérieur. Ceci est en général désactivé pour cause de dangerosité. Pour ce, on rajoute la directive de compilation `--enable-write18` ou `--shell-escape`, selon l'os/distribution. Pour Windows/MikTeX, j'ai rajouté la première dans les options de ligne de commande de **latex.exe** et **pdflatex.exe**.
- Lors de l'utilisation de Gnuplot, sont créés deux fichiers : `nomtex.xxx.gnuplot` qui contient les commandes et `nomtex.xxx.table` qui contient les coordonnées des points utilisés. Le même xxx (donc le même nom) est utilisé pour chaque dessin (faisant appel à Gnuplot) dans tout le fichier TeX. Il est préférable, pour chaque dessin, de donner à **plot** une option du style **id=yyy** qui remplacera alors le xxx. Personnellement, j'ai eu des problèmes à la compilation sous Windows/MikTeX (fichier `.table` non trouvé) lorsque le fichier `nomtex.tex` contenait des « *espaces* » dans son nom. Si quelqu'un a la solution, je suis intéressé.

3.2 UTILISATION AVEC TIKZ

En préparation...