

# INTRODUCTION À PGF-TikZ

version 0.3 – 2008-11-03

## 1 Introduction

PGF (*Portable Graphics Format*) est une extension à  $\text{\LaTeX}$  qui permet de créer des dessins à l'aide de commandes en-ligne. TikZ [2] est une boîte à outils frontale à PGF qui offre des fonctions pour simplifier la vie des utilisateurs. De nombreux exemples de figures réalisées avec TikZ sont disponibles sur Internet [1]. Elles permettent d'aller plus loin dans l'exploration et l'exploitation des extraordinaires possibilités de PGF/TikZ.

Utiliser TikZ offre l'avantage de limiter la taille du fichier PDF obtenu en évitant d'avoir à générer des images (PNG ou JPEG) pour chaque figure et à les insérer dans le fichier. Par contre, il faut connaître les commandes TikZ pour pouvoir créer son dessin. Cet article a pour but de vous montrer quelques commandes pour vous lancer dans l'aventure PGF/TikZ.

Le fichier le plus simple pour utiliser TikZ est le suivant :

```
\documentclass[11pt]{article}

\usepackage{xcolor} % package pour les couleurs
\usepackage{tikz} % package principal TikZ
\usetikzlibrary{arrows} % librairie optionnelle PGF

\begin{document}
Bla bla bla\dots
\bigskip

\begin{tikzpicture} % figure TikZ
\draw (0,0) — (1,1);
\end{tikzpicture}

\end{document}
```

Comme nous allons le voir, la syntaxe de TikZ est très simple. Chaque ligne décrit une opération à réaliser sur le graphique et les lignes sont terminées par des points-virgules.

## 2 TikZ par l'exemple

### 2.1 Dessiner des lignes sur une grille

La chose la plus simple à dessiner avec TikZ est la ligne entre deux points dont on donne les coordonnées grâce au symbole `—`. Ce symbole indique une liaison par une ligne entre deux éléments. On dessine donc un segment de droite entre les points de coordonnées (0,0) et (1,1). Par défaut, l'unité graphique de TikZ est le centimètre, mais on peut utiliser toutes les unités définies par  $\text{\LaTeX}$ .



```
\begin{tikzpicture}
\draw (0,0) — (1,1);
\end{tikzpicture}
```

Avec TikZ, on peut ajouter très facilement des propriétés à notre ligne, comme la couleur, le type de hachure et l'épaisseur.



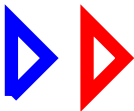
```
\begin{tikzpicture}
\draw[color=blue,dashed,very thick] (0,0) — (1,1);
\end{tikzpicture}
```

On peut définir une liste de points à relier ensemble avec leurs propriétés graphiques (couleur, épaisseur... ) comme on vient de le voir. le mot-clé `cycle` à la fin indique à TikZ qu'on repasse par le point de départ et qu'il faut faire correspondre les coins de la figure à cet endroit.



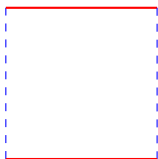
```
\begin{tikzpicture}
\draw[color=red, ultra thick] (0,0) — (0,1) — (1,1) — (1,0) — cycle;
\end{tikzpicture}
```

Si on ne met pas le mot-clé `cycle`, on risque d'avoir une discontinuité dans un coin de la figure. On remarque un problème de liaison pour le triangle bleu en bas à gauche, le triangle rouge est parfaitement construit avec `cycle`.



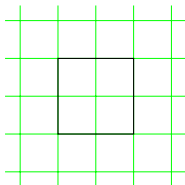
```
\begin{tikzpicture}[line width=5pt]
\draw [color=blue] (0,0) — (0,1) — (0.5,0.5) — (0,0);
\draw [color=red] (1,0) — (1,1) — (1.5,0.5) — cycle;
\end{tikzpicture}
```

Lorsqu'on utilise plusieurs types d'objets différents dans une même figure, TikZ offre des fonctions de définition de styles très utiles.



```
\begin{tikzpicture}[scale=2]
tikzstyle{style1}=[color=red, thick]
tikzstyle{style2}=[color=blue, thin, dashed]
\draw[style=style2] (0,0) — (0,1);
\draw[style=style1] (0,1) — (1,1);
\draw[style=style2] (1,1) — (1,0);
\draw[style=style1] (1,0) — (0,0);
\end{tikzpicture}
```

TikZ propose une fonction très utile lorsqu'on débute, l'ajout d'une grille au dessin pour pouvoir plus facilement se repérer. La même chose que précédemment avec une grille :



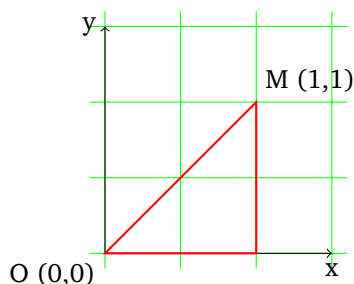
```
\begin{tikzpicture}
\draw[step=0.5cm, color=green, very thin] (-0.7,-0.7) grid (1.7,1.7);
\draw (0,0) — (0,1) — (1,1) — (1,0) — cycle;
\end{tikzpicture}
```

## 2.2 Système de coordonnées

TikZ possède deux systèmes de coordonnées : les coordonnées cartésiennes (abscisse  $x$  et ordonnée  $y$ ) et les coordonnées polaires (module  $r$  et argument (angle)  $\theta$ ). Les deux figures ci-dessous donnent un exemple d'utilisation de chacun des systèmes de coordonnées. Le point  $O$  de coordonnées cartésiennes  $(0,0)$  et polaire  $(0 : 0\text{cm})$  est relié au point  $M$  de coordonnées cartésiennes  $(2,2)$  et de coordonnées polaires  $(45 : 2.82\text{cm})$ .

Les figures suivantes contiennent également des commandes de placement d'information (des nœuds, *node* en anglais) sur lesquels nous reviendrons un peu plus tard.

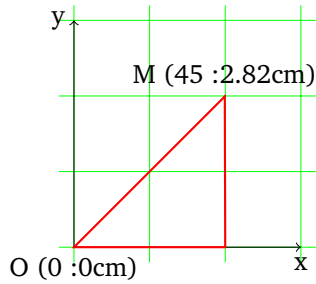
– Coordonnées cartésiennes :



```
\begin{tikzpicture}
\draw [step=1cm, color=green, very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw [->] (0,0) — (3,0);
\draw [->] (0,0) — (0,3);
\path (3,0) node [anchor=north] {x};
\path (0,3) node [anchor=east] {y};
\draw (0,0) node [anchor=north east] {O (0,0)};
\draw (2,2) node [anchor=south west] {M (1,1)};
\draw [color=red, thick] (0,0) — (2,0) — (2,2) — cycle;
\end{tikzpicture}
```

L'unité par défaut est le centimètre pour les distances et le degré pour les angles.

- Coordonnées polaires :



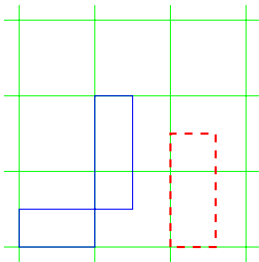
```
\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw [->] (0:0) — (90:3cm);
\draw [->] (0:0) — (0:3cm);
\path (0:3cm) node [anchor=north] {x};
\path (90:3cm) node [anchor=east] {y};
\draw (0:0cm) node [anchor=north] {O (0:0cm)};
\draw (45:2.82cm) node [anchor=south] {M (45:2.82cm)};
\draw [color=red,thick] (0:0cm) — (0:2cm) — (45:2.82cm)
— cycle;
\end{tikzpicture}
```

Pourquoi le point M a-t-il comme coordonnées polaires (45 : 2.82cm) ? Il se trouve sur une droite située à 45 degrés par rapport à l'axe des abscisses (horizontal). Il est sur le triangle rectangle de côtés horizontaux et verticaux de longueur 2cm. On sait que dans un triangle rectangle, l'hypothénuse a pour longueur la somme des carrés des deux autres côtés, soit  $l = \sqrt{2^2 + 2^2} = \sqrt{4 + 4} = \sqrt{8} \approx 2.82cm$ .

## 2.3 rectangles, cercles, ellipses et autres arcs

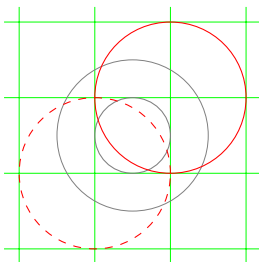
TikZ fournit un ensemble d'objets graphiques à insérer dans les dessins.

- rectangles (*rectangle*) : ils sont définis par deux coins opposés.



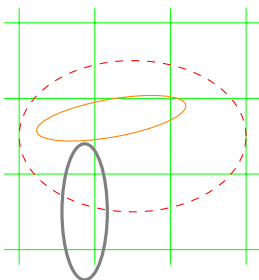
```
\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw[color=blue] (0,0) rectangle (1,1);
\draw[color=red,dashed,thick] (1,0.5) rectangle (2,1.5);
\end{tikzpicture}
```

- cercles (*circle*) : ils sont définis par les coordonnées de leur centre et leur rayon.



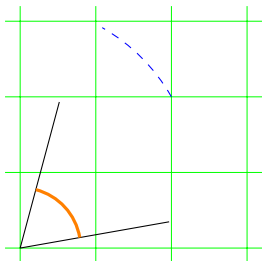
```
\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw [color=gray,very thin] (1.5,1.5) circle (1cm);
\draw [color=gray,thin] (1.5,1.5) circle (0.5cm);
\draw [color=red,dashed] (1,1) circle (1cm);
\draw [color=red] (2,2) circle (1cm);
\end{tikzpicture}
```

- ellipses (*ellipse*) : elles sont définies par les coordonnées de leur centre et leurs longueurs grand axe/-petit axe ainsi que de l'angle de rotation si elles sont tournées.



```
\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw [color=red,dashed] (1.5,1.5) ellipse (1.5cm and 1 cm);
\draw [color=gray,,very thick] (30:1cm) ellipse (0.3cm and 0.9cm);
\draw [rotate=10,color=orange] (1.5,1.5) ellipse (1cm and 0.25cm);
\end{tikzpicture}
```

- arcs (*arc*) : ils sont définis par trois paramètres, les deux premiers sont l'angle de départ, l'angle d'arrivée et le dernier est le rayon de l'arc.

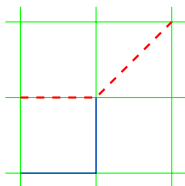


```
\begin{tikzpicture}
\draw [step=1cm,color=green,very thin] (-0.2,-0.2) grid (3.2,3.2);
\draw (0:0cm) — (10:2cm);
\draw (0:0cm) — (75:2cm);
\draw [color=orange,very thick] (10:8mm) arc (10:75:8mm);
\path (32.5:8mm) node [anchor=south west] {$\alpha$};
\draw [color=blue,dashed] (2,2) arc (30:60:2.5cm);
\end{tikzpicture}
```

## 2.4 Coordonnées relatives et absolues

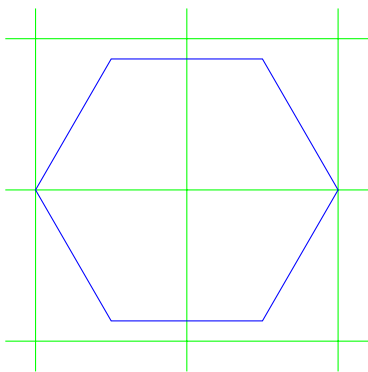
TikZ permet de définir les coordonnées des points à relier soit en relatif : par rapport à la position actuelle, soit en absolu : par rapport à l'origine (0,0) du repère. Dans cette translation de coordonnées, on note par un + un déplacement absolu et par ++ un déplacement relatif.

Dans l'exemple ci-dessous, la ligne bleue est réalisée avec un déplacement absolu de (0,0) vers (1,0) puis (1,1). La ligne rouge est un déplacement relatif à la position précédente, on va donc de (0,1) à (1,1) puis on ajoute (1,1) à la position précédente et on arrive donc en (2,2).



```
\begin{tikzpicture}
\draw [step=1cm,color=green,thin] (-0.2cm,-0.2cm) grid (2.2cm,2.2cm);
\draw [color=blue,thin] (0,0) — +(1,0) — +(1,1);
\draw [color=red,dashed,thick] (0,1) — ++(1,0) — ++(1,1);
\end{tikzpicture}
```

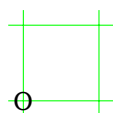
Lorsqu'on utilise des points particuliers dans un dessin, on peut stocker leurs coordonnées et y faire référence dans la suite du dessin (très pratique).



```
\begin{tikzpicture}[scale=2]
\draw [step=1cm,color=green] (-1.2cm,-1.2cm) grid (1.2cm,1.2cm);
\path (0:0cm) coordinate (O);
\path (0:1cm) coordinate (P0);
\path (1*60:1cm) coordinate (P1);
\path (2*60:1cm) coordinate (P2);
\path (3*60:1cm) coordinate (P3);
\path (4*60:1cm) coordinate (P4);
\path (5*60:1cm) coordinate (P5);
\draw [color=blue] (P0) — (P1) — (P2) — (P3) — (P4) — (P5)
— cycle;
\end{tikzpicture}
```

## 2.5 Les nœuds

Les nœuds sont des éléments particuliers d'un dessin qui possèdent leurs propriétés graphiques propres ainsi qu'éventuellement un texte associé. Puisque TikZ fonctionne sous  $\text{\LaTeX}$ , le texte peut évidemment être une formule  $\text{\LaTeX}$ .

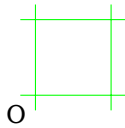


```
\begin{tikzpicture}
\draw [step=1cm,color=green,thin] (-0.2cm,-0.2cm) grid (1.2cm,1.2cm);
\path (0,0) node {O};
\end{tikzpicture}
```

Le problème de positionner le texte "O" sur la grille à la position (0,0) est qu'il masque justement cette position. TikZ fournit une commande d'ancrage des nœuds sous la forme de la propriété "anchor".

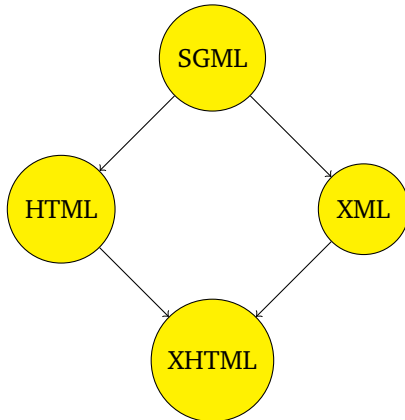
Ces positions "anchor" sont indiquées par les points cardinaux : *north*, *south*, *east*, *west*, *north east*, *north west*, *south east*, *south west*. Attention, l'ancrage est effectué en indiquant où se trouve le point d'ancrage par rapport au texte et non pas le texte par rapport au point d'ancrage. Ainsi pour demander que le texte "O"

associé au point origine du repère de coordonnées (0,0) soit affiché en bas à gauche de ce point, on ajoute "anchor=north east" (et pas south west) dans les propriétés du nœud.



```
\begin{tikzpicture}
\draw [step=1cm,color=green,thin] (-0.2cm,-0.2cm) grid (1.2cm,1.2cm);
\path (0,0) node [anchor=north east] {O};
\end{tikzpicture}
```

Il existe un système très pratique pour gérer les graphes : c'est la définition de nœuds représentant les éléments du graphe.

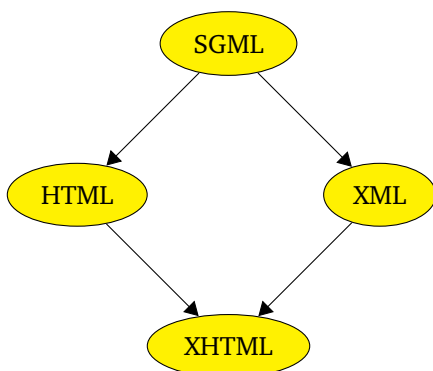


```
\begin{tikzpicture}
\path (0,0) node (n0) [circle,fill=yellow,draw,inner sep=5pt] {SGML};
\path (-2,-2) node (n1) [circle,fill=yellow,draw,inner sep=5pt] {HTML};
\path (2,-2) node (n2) [circle,fill=yellow,draw,inner sep=5pt] {XML};
\path (0,-4) node (n3) [circle,fill=yellow,draw,inner sep=5pt] {XHTML};
%\pgfsetlinewidth{0.02cm};
\draw[>-] (n0) -- (n1);
\draw[>-] (n0) -- (n2);
\draw[>-] (n1) -- (n3);
\draw[>-] (n2) -- (n3);
\end{tikzpicture}
```

Afin d'améliorer l'aspect de certains éléments graphiques, il existe de nombreuses librairies associées à TikZ. Pour les utiliser, il faut ajouter les lignes `\usetikzlibrary{...}` après la commande `\usepackage{tikz}`.

```
\usepackage{tikz}
% Optional PGF libraries
\usetikzlibrary{arrows}
\usetikzlibrary{snakes}
\usetikzlibrary{shapes}
```

Par exemple, on peut remplacer les cercles qui entourent les nœuds dans le graphe ci-dessus par des ellipses (librairie TikZ *shapes*). On peut aussi améliorer l'aspect des flèches (librairie TikZ *arrows*) :

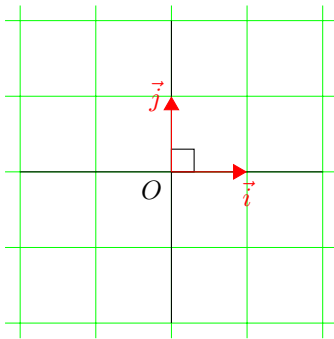


```

\begin{tikzpicture}
% pgflibraryshapes obligatoire !
\path (0,0) node (n0) [ellipse, fill=yellow, draw, inner sep=5pt] {SGML};
\path (-2,-2) node (n1) [ellipse, fill=yellow, draw, inner sep=5pt] {HTML};
\path (2,-2) node (n2) [ellipse, fill=yellow, draw, inner sep=5pt] {XML};
\path (0,-4) node (n3) [ellipse, fill=yellow, draw, inner sep=5pt] {XHTML};
% \pgfsetlinewidth{0.02cm};
\draw[-triangle 60] (n0) — (n1); % pgflibraryarrows !
\draw[-triangle 60] (n0) — (n2); % pgflibraryarrows !
\draw[-triangle 60] (n1) — (n3);
\draw[-triangle 60] (n2) — (n3);
\end{tikzpicture}

```

Un exemple un peu plus conséquent : la création d'un repère orthonormal pour la représentation graphique de fonctions mathématiques.

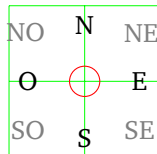


```

\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-2.2,-2.2) grid (2.2,2.2);
\path (0,0) coordinate (origin);
\path (1,0) coordinate (x);
\path (0,1) coordinate (y);
\path (-2,0) coordinate (xm);
\path (2,0) coordinate (xp);
\path (0,-2) coordinate (ym);
\path (0,2) coordinate (yp);
\draw (xm) — (xp);
\draw (ym) — (yp);
\draw (0,0) rectangle (3mm,3mm);
\draw (origin) — (origin) node [anchor=north east] {$O$};
\draw [-triangle 60,color=red] (origin)
— (x) node [anchor=north] {$\vec{e}_i$};
\draw [-triangle 60,color=red] (origin)
— (y) node [anchor=east] {$\vec{e}_j$};
\end{tikzpicture}

```

Une autre façon pour indiquer les positions est de spécifier des valeurs non plus d'ancrage, mais de positionnement. C'est-à-dire qu'on spécifie où se trouve le texte par rapport aux coordonnées à l'aide des mots-clés *above*, *left*, *right*, *below*, *above left*, *above right*, *below left*, *below right* et en précisant une distance : *above=0.4cm*.

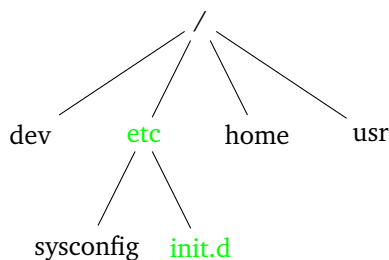


```

\begin{tikzpicture}
\draw[step=1cm,color=green,very thin] (-1,-1) grid (1,1);
\path (0,0) coordinate (origine);
\draw [color=red] (origine) circle (.2cm);
\path (origine) node (s) [below=0.5cm] {S};
\path (origine) node (o) [left=0.5cm] {O};
\path (origine) node (n) [above=0.5cm] {N};
\path (origine) node (e) [right=0.5cm] {E};
\path [color=gray] (origine) node (no) [above left=0.4cm] {NO};
\path [color=gray] (origine) node (ne) [above right=0.4cm] {NE};
\path [color=gray] (origine) node (so) [below left=0.4cm] {SO};
\path [color=gray] (origine) node (se) [below right=0.4cm] {SE};
\end{tikzpicture}

```

On peut aussi créer des arbres très facilement sous TikZ.

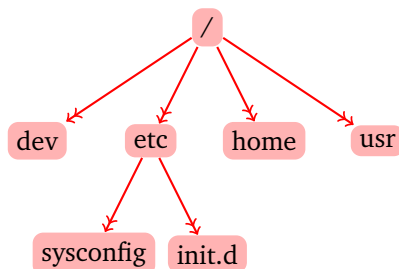


```

\begin{tikzpicture}
\node {}
child {node {dev}}
child {node [color=green] {etc}}
child {node {sysconfig}}
child {node [color=green] {init.d}}
}
child {node {home}}
child {node {usr}};
\end{tikzpicture}

```

Grâce à la définition de styles, TikZ permet de décorer autrement la figure précédente.



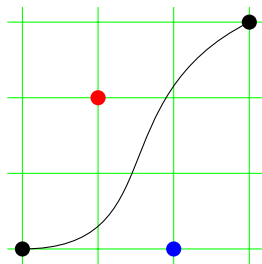
```

\begin{tikzpicture}
\tikzstyle{every node}=[fill=red!30,rounded corners]
\tikzstyle{edge from parent}=[red,->,thick,draw]
\node{/}
  child {node {dev}}
  child {node {etc}}
    child {node {sysconfig}}
    child {node {init.d}}
  child {node {home}}
  child {node {usr}};
\end{tikzpicture}

```

## 2.6 Les courbes

PGF/TikZ permet de définir des courbes de Bézier pour relier deux points. Il faut définir le point de départ et le point d'arrivée ainsi qu'un ou deux point(s) de contrôle qui régissent la trajectoire de la courbe.



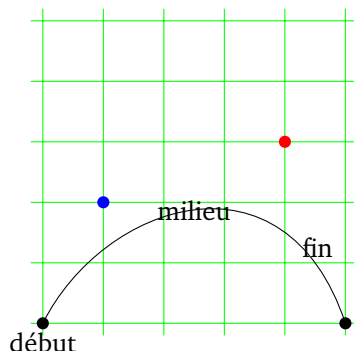
```

\begin{tikzpicture}
\draw[step=1cm,color=green] (-0.2cm,-0.2cm) grid (3.2cm,3.2cm);
\path (0,0) coordinate (src);
\path (3,3) coordinate (dst);
\path (2,0) coordinate (ctrl1);
\path (1,2) coordinate (ctrl2);
\draw (src) .. controls (ctrl1) and (ctrl2) .. (dst);
\fill (src) circle (1mm);
\fill (dst) circle (1mm);
\fill[blue] (ctrl1) circle (1mm);
\fill[red] (ctrl2) circle (1mm);
\end{tikzpicture}

```

## 2.7 Le positionnement de textes sur une courbe

Il est possible de positionner du texte sur une courbe en précisant une position par rapport à la courbe, par exemple : au début, vers la fin, au milieu... C'est assez pratique pour positionner du texte automatiquement sur une courbe. Il existe sept positions pour placer le texte : au début (*at start*), très proche du début (*very near start*), proche du début (*near start*), au milieu (*midway*), proche de la fin (*near end*), très proche de la fin (*very near end*), à la fin (*at end*). On précise où doit se trouver le texte par rapport à la courbe avec les options *below* (en-dessous) et *above* (au-dessus) car, par défaut, le texte est inséré sur la courbe.

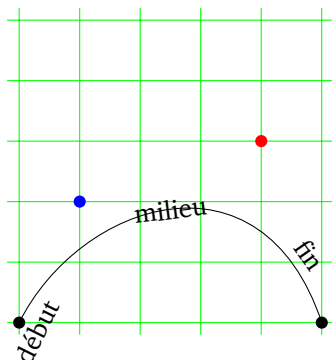


```

\begin{tikzpicture}[scale=0.8]
\draw[step=1cm,color=green] (-0.2cm,-0.2cm) grid (5.2cm,5.2cm);
\path (0,0) coordinate (src);
\path (5,0) coordinate (dst);
\path (1,2) coordinate (ctrl1);
\path (4,3) coordinate (ctrl2);
\draw (src) ..
  controls (ctrl1) and (ctrl2) ..
  node[at start,below] {début}
  node[midway] {milieu}
  node[very near end,above] {fin}
  (dst);
\fill (src) circle (1mm);
\fill (dst) circle (1mm);
\fill[blue] (ctrl1) circle (1mm);
\fill[red] (ctrl2) circle (1mm);
\end{tikzpicture}

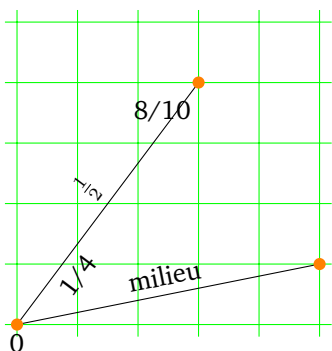
```

Pour améliorer le rendu, il est possible d'ajouter une option pour positionner le texte afin qu'il suive la trajectoire de la courbe avec l'option *sloped*. Le résultat est plus agréable visuellement.



```
\begin{tikzpicture}[scale=0.8]
\draw[step=1cm,color=green](-0.2cm,-0.2cm)grid(5.2cm,5.2cm);
\path(0,0)coordinate(src);
\path(5,0)coordinate(dst);
\path(1,2)coordinate(ctrl1);
\path(4,3)coordinate(ctrl2);
\draw(src)..
controls(ctrl1)and(ctrl2)..
node[at start,below,sloped]{d\'ebut}
node[midway,sloped]{milieu}
node[very near end,above,sloped]{fin}
(dst);
\fill(src)circle(1mm);
\fill(dst)circle(1mm);
\fill[color=blue](ctrl1)circle(1mm);
\fill[color=red](ctrl2)circle(1mm);
\end{tikzpicture}
```

Le positionnement marche aussi pour les lignes. De plus, si on veut être plus précis, on peut utiliser l'option *pos* qui positionne un texte à l'endroit où on le souhaite sur une courbe ou une droite.

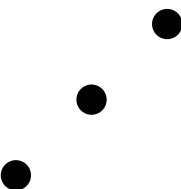


```
\begin{tikzpicture}[scale=0.8]
\draw[step=1cm,color=green](-0.2cm,-0.2cm)grid(5.2cm,5.2cm);
\path(0,0)coordinate(source);
\path(5,1)coordinate(dest1);
\path(3,4)coordinate(dest2);
\draw(source)--node[midway,above,sloped]{milieu}(dest1);
\draw(source)--(dest2);
node[pos=0,below]{0}node[pos=0.25,below,sloped]{1/4}
node[pos=0.5,above,sloped]{\frac{1}{2}}
node[pos=0.8,above]{8/10};
\foreach\i in{source,dest1,dest2}
{\fill[color=orange](\i)circle(1mm);}
\end{tikzpicture}
```

La boucle *foreach* insérée à la fin de l'exemple permet de dessiner un cercle orange de rayon 1mm. Les boucles sont expliquées plus en détail dans la section suivante.

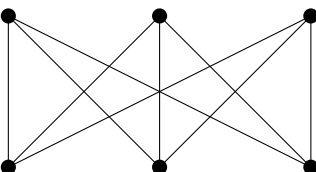
## 2.8 Les boucles

Une fonctionnalité très utile de TikZ/PGF réside dans la possibilité de créer des boucles comme dans un langage de programmation classique.



```
\begin{tikzpicture}
\foreach\i in{1,...,3}
{
\path(\i cm,\i cm)coordinate(P\i);
\fill(P\i)circle(2mm);
}
\end{tikzpicture}
```

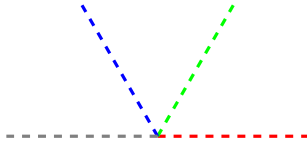
Les boucles sont utiles lorsqu'on doit définir des actions répétitives.



```
\begin{tikzpicture}
\foreach\i in{1,...,3}
{
\path(\i,0)coordinate(X\i);
\fill(X\i)circle(0.5mm);
}
\foreach\j in{1,...,3}
{
\path(\j,1)coordinate(Y\j);
\fill(Y\j)circle(0.5mm);
}
\foreach\i in{1,...,3}
{
\foreach\j in{1,...,3}
{
\draw(X\i)--(Y\j);
}
}
\end{tikzpicture}
```



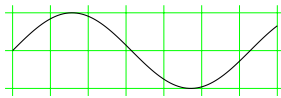
On peut aussi modifier plusieurs valeurs en même temps dans une boucle.



```
\begin{tikzpicture}[scale=2]
\foreach \angle / \color in {0/red,60/green,120/blue,180/gray}
{
\draw [color=\color,very thick,dashed] (0:0cm) — (\angle:1cm);
}
\end{tikzpicture}
```

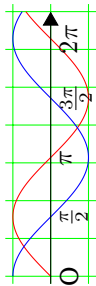
## 2.9 Les fonctions

TikZ offre la possibilité de tracer des fonctions par l'intermédiaire de Gnuplot. On donne la fonction et quelques informations, TikZ génère un fichier gnuplot lors de la compilation PDFLaTeX. Il ne reste plus qu'à lancer la commande "gnuplot \*.gnuplot" dans le répertoire de travail et Gnuplot crée le fichier ".data" qui sera inséré dans le document PDF à la compilation suivante.



```
\begin{tikzpicture}[scale=0.5]
\draw [very thin,color=green] (-0.2cm,-1.2cm) grid (7.2cm,1.2cm);
\draw [raw gnuplot] plot [id=sin0] function{plot [0:7] sin(x)};
\end{tikzpicture}
```

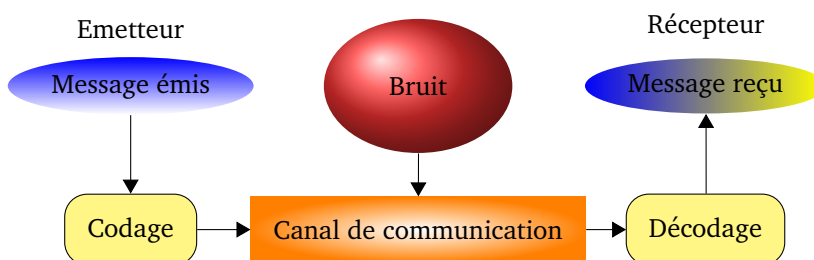
Un exemple un peu plus compliqué pour aller plus loin, on le tourne de 90° (*rotate*) par rapport à l'horizontale et on le met à l'échelle 0.5 (*scale*).



```
\begin{tikzpicture}[scale=0.5,rotate=90]
\draw [step=1cm,color=green] (-0.2cm,-1.2cm) grid (7.2cm,1.2cm);
\draw [color=red,raw gnuplot] plot [id=sin1]
function{plot [0:7] sin(x)};
\draw [color=blue,raw gnuplot] plot [id=cos1]
function{plot [0:7] cos(x)};
\path (3.14/2,0) node [anchor=north,rotate=90] {$\frac{\pi}{2}$};
\path (3.14,0) node [anchor=north,rotate=90] {$\pi$};
\path (3/2*3.14,0) node [anchor=north,rotate=90] {$\frac{3\pi}{2}$};
\path (2*3.14,0) node [anchor=north,rotate=90] {$2\pi$};
\path (0,0) node [anchor=north,rotate=90] {O};
\draw [-triangle 60] (-0.2cm,0cm) — (7cm,0cm);
\end{tikzpicture}
```

## 2.10 Les dégradés

On peut réaliser des dégradés de couleur sous TikZ avec le mot-clé *shade*. On précise les couleurs avec les options décrites ci-dessous. Les couleurs sont volontairement peu judicieuses, mais c'est juste pour l'exemple.



```
\begin{tikzpicture}[scale=0.38]
\path (0,0) node [rectangle,fill=yellow!60,draw,inner sep=0.3cm,rounded corners=0.3cm]
(codage) {Codage};
\path (10,0) node [rectangle,shade,inner color=white,outer color=orange,inner sep=0.3cm]
(canal) {Canal de communication};
\path (20,0) node [rectangle,fill=yellow!60,draw,inner sep=0.3cm,rounded corners=0.3cm]
(decodage) {D\'ecodage};
\draw [-triangle 60] (codage) — (canal);
\draw [-triangle 60] (canal) — (decodage);
\end{tikzpicture}
```

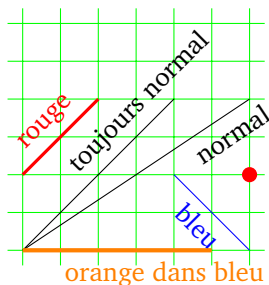
```

\path (0,5) node [ellipse,shade,top color=blue,bottom color=white,inner sep=0.1cm]
(emis) {\Message \'}emis};
\path (emis) node [above=0.5cm] {Emetteur};
\draw [-triangle 60] (emis) — (codage);
\path (20,5) node [ellipse,shade,left color=blue,right color=yellow,inner sep=0.1cm]
(recu) {\Message re\c{c}u};
\path (recu) node [above=0.5cm] {R\'}ecepteur};
\draw [-triangle 60] (decodage) — (recu);
\path (10,5) node [ellipse,ball color=red!80,inner sep=0.5cm] (bruit) {Bruit};
\draw [-triangle 60] (bruit) — (canal);
\end{tikzpicture}

```

## 2.11 Les environnements

Lorsqu'on réalise un schéma, on a souvent besoin de définir des styles qui s'appliquent à plusieurs éléments du schéma, TikZ fournit l'environnement *scope*. Dans le cas de multiples styles définis pour un objet, c'est le style le plus local (défini au plus près de l'objet) qui est appliqué.



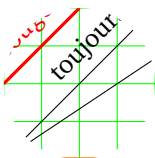
```

\begin{tikzpicture}
\draw [step=0.5cm,color=green] (-0.2cm,-0.2cm) grid (3.2cm,3.2cm);
\draw (0,0) — (3,2) node [very near end,below,sloped] {normal};
\begin{scope}[color=red,very thick]
\draw (0,1) — (1,2) node [midway,above,sloped] {rouge};
\fill (3,1) circle (1mm);
\end{scope}
\draw (0,0) — (2,2) node [very near end,sloped,above]
{toujours normal};
\begin{scope}[color=blue,thin]
\draw (2,1) — (3,0) node [midway,below,sloped] {bleu};
\draw [color=orange,ultra thick] (0,0) — (2.5,0)
node [near end,below,sloped] {orange dans bleu};
\end{scope}
\end{tikzpicture}

```

## 2.12 Le découpage

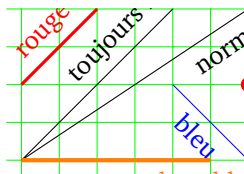
Si on ne veut garder qu'une partie du dessin, on peut ajouter une commande de découpage en précisant le type de la zone de découpage et ses dimensions. On reprend l'exemple précédent en le coupant sous la forme d'un disque et sous la forme d'un rectangle. Il y a juste une ligne de découpage (*clip*) à ajouter juste après le `\begin{tikzpicture}`. Un *clip* dans un *scope* est local au contenu du *scope*.



```

\begin{tikzpicture}
\clip (1,1)
circle (1cm);
[...]
\end{tikzpicture}

```



```

\begin{tikzpicture}
\clip (-0.2,-0.2)
rectangle (3,2);
[...]
\end{tikzpicture}

```

## 3 Conclusion

TikZ est un outil puissant et facile d'utilisation (avec un peu d'habitude) pour dessiner des graphiques de qualité sous  $\text{\LaTeX}$ . Il propose des fonctions simples de mise en page et de dessin qui permettent à tous de créer des figures de qualité avec un encombrement mémoire très faible (plus faible que le chargement d'une image PNG ou JPG externe). L'avantage de la méthode est le gain de place et la souplesse de n'avoir qu'un seul fichier source du document qui contient le texte et les figures. Bonne utilisation de PGF/TikZ. . .

## Références

- [1] MANY CONTRIBUTORS, PGF/TikZ online examples, <http://www.texample.net/tikz/examples>, 2008
- [2] TILL TANTAU, The PGF and TikZ packages, <http://sourceforge.net/projects/pgf>, 2005