



L^AT_EX, un peu, beaucoup

Pascal Dupont

Mots clés : LaTeX, dessins, figures, Tikz



11. TikZ (1)

Pour dessiner, dans un texte en L^AT_EX, de jolis diagrammes, des graphes de fonctions, des figures géométriques, une solution est d'utiliser un logiciel de dessin pour créer la figure, puis de l'importer dans le texte au moyen de la commande `\includegraphics`, que nous avons déjà évoquée dans le neuvième avatar de cette rubrique ⁽¹⁾, en promettant d'y revenir plus longuement.

Cette heure-là n'est pas encore venue.

Parce que, pour l'usage qui est ici en cause, elle a au moins trois inconvénients d'importance.

1. Le lettrage des figures, qui aura été créé par un logiciel de dessin et non par L^AT_EX, jurera de manière affreuse avec le texte.

2. Si la figure doit être redimensionnée, elle subira *globalement* une homothétie, ce qui signifie que des dimensions qui n'auraient pas dû être modifiées le seront tout de même : la taille du lettrage, l'épaisseur des traits, ...

3. La figure est contenue dans un document *externe* au document `.tex` qui contient l'article, ce qui induit des risques d'oubli lorsque nous devons déplacer le document dans un autre dossier ou l'envoyer à un tiers.

Bref, l'usage de `\includegraphics` devrait être réservé, autant que possible, à l'inclusion de documents iconographiques. Pour créer des figures de nature géométrique, il est de loin préférable d'inclure dans le document source toutes les instructions pour la tracer.

Pour ce faire, le L^AT_EX de base propose l'environnement `picture`, qui permet de tracer quelques segments et quelques cercles ; mais ses limitations sont rédhibitoires.

⁽¹⁾ *Losanges* 30, 61–62.

⁽²⁾ J'avoue que ce dernier dilemme est sérieusement passé de mode !

Heureusement, existent deux modules qui étendent très largement les capacités graphiques de L^AT_EX : *PS-Tricks* et *TikZ*.

Ou plutôt, aurais-je dû écrire, *malheureusement, existent deux modules qui étendent très largement les capacités graphiques de L^AT_EX*. En effet, ces deux solutions sont largement incompatibles, ce qui oblige l'utilisateur à se positionner, radicalement, dès sa première utilisation, comme, hélas, dans de nombreux contextes techniques (*Mac* ou *PC* ? *Nikon* ou *Canon* ? *VHS* ou *Betamax* ? ⁽²⁾) Dès qu'on a mis un pied dans l'un de ces mondes, on y est à peu près coincé, les accessoires étant incompatibles avec le monde d'en face ; et pourtant, au départ, le choix s'est fait sur des arguments très ténus : les appareils photographiques conçus à Shinjuku sont-ils nettement meilleurs que ceux nés à Tokyo ? Ces derniers sont-ils nettement moins chers que leurs concurrents ? Non point.)

Il y a une dizaine d'années, je me suis d'abord intéressé à *PS-Tricks*, entraîné par un collègue ; mais fort peu de temps après, je me suis mis à utiliser *Beamer* pour mes cours, et à l'époque *Beamer* et *PS-Tricks* étaient incompatibles (du moins avec le compilateur que j'utilisais). En fait, *TikZ* a été développé par Till TANTAU dans la foulée de *BEAMER*, pour cette raison. Des choses se sont arrangées depuis cette époque, mais le « mal » était fait, j'étais contaminé par le virus de *TikZ*.

En fait, *TikZ* est la partie émergée de *PGF* ; il s'agit d'un langage de description de page, qui étend le PDF (*PGF* est l'acronyme de *Portable Graphics Format*). Rapidement, on pourrait dire que *TikZ* est un ensemble de commandes destiné à manipuler le *PGF* sans trop devoir mettre les mains dans le cambouis.

Figures TikZ

Une figure TikZ sera contenue dans un environnement `tikzpicture`⁽³⁾. Il n'est pas nécessaire d'en définir les limites : c'est, par défaut, le rectangle qui contient tout ce qui doit être dessiné. L'unité par défaut est le centimètre. Pour n'envisager pour l'instant que la solution la plus simple, les points du plan sont repérés par leurs coordonnées cartésiennes.

Chemins

La notion centrale, pour TikZ, est celle de *chemin*, *path* en anglais.

Exemple

```
\path(0,0)--(1,0)--(0,1)--(0,0);
```

Exemple

est un chemin dont le premier nœud est l'origine, qui comprend un segment d'axe *Ox*, jusqu'à l'abscisse 1 (second nœud), qui va de là, toujours via un segment de droite, au point (0,1) (troisième nœud), et qui retourne à l'origine via un segment d'axe *Oy*. On notera que tout chemin voit sa description clôturée par un point-virgule et, par ailleurs, que des blancs (ou des passages à la ligne) peuvent sans façons être ajoutés, pour améliorer la lisibilité.

Nous avons donc ainsi décrit un joli triangle rectangle isocèle. Cependant, nous ne l'avons pas tracé ! En effet, la commande `\path` ne fait que définir le chemin, elle ne le trace pas ; pour le tracer, il faut remplacer la commande `\path` par `\draw` :

Exemple

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

donne



Exemple

Pourquoi cette complication, vous demandez-vous sans doute ? La raison en est que le tracer n'est pas la seule chose que nous pouvons faire d'un chemin ; par exemple, nous pouvons le remplir :

⁽³⁾ Ce n'est pas la seule possibilité. En fait, TikZ regorge de synonymes, qu'on peut voir comme des commandes redondantes. Dans ce texte introductif, nous n'en ferons guère l'inventaire.

Exemple

```
\begin{tikzpicture}
\fill(0,0)--(1,0)--(0,1)--(0,0);
\end{tikzpicture}
```

colorie donc le triangle (noir) :



Exemple

Nous rencontrerons encore plus loin d'autres usages des chemins. Une figure TikZ consiste donc principalement en des commandes de tracé, de remplissage, ..., de chemins — chacun terminé par son point-virgule.

Dans le cas d'un chemin fermé, comme ici, il est possible de remplacer la mention finale du nœud de départ par `cycle` :

Exemple

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

trace aussi le triangle



Exemple

Cette solution, cependant, est préférable, car le raccord du dernier segment au premier se fait de la même façon qu'aux points intermédiaires, ce qui n'est pas le cas dans notre première solution (mais la différence, ici est imperceptible).

Les commandes `\draw` et `\fill` admettent des options, qui viennent, sans surprise, se placer entre crochets à leur suite ; si plusieurs options doivent cohabiter, nous les séparons par des virgules :

Exemple

```
\begin{tikzpicture}
\draw[very thick,dashed]
(0,0)--(1,0)--(0,1)--(0,0);
\end{tikzpicture}
```

produit



Exemple

tandis que

Exemple

```
\begin{tikzpicture}
\fill[semitransparent,magenta]
(0,0)--(1,0)--(0,1)--(0,0);
\end{tikzpicture}
```

remplit notre triangle de magenta partiellement transparent, ce qui signifie qu'il apparaît plus clair, mais aussi que si une autre partie du dessin (ou du texte) se trouvait derrière, il ne serait pas complètement masqué.



Exemple

Il est possible de combiner le tracé et le remplissage en une seule commande :

Exemple

```
\begin{tikzpicture}
\filldraw[thick,fill=magenta]
(0,0)--(1,0)--(0,1)--(0,0);
\end{tikzpicture}
```

crée



Exemple

Dans le chemin qui nous a servi d'exemple jusque ici, les nœuds sont reliés par des segments de droite. D'autres possibilités existent, on s'en doute. TikZ parle d'*opérations* pour désigner ces diverses « jonctions ».

- Omission d'un segment :

Exemple

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

donne



Les deux nœuds intermédiaires ne sont pas reliés. Ceci rend impossible l'utilisation de `cycle` pour boucler le chemin.

Exemple

- Segment parallèle aux axes :

Exemple

Avec

```
\begin{tikzpicture}
\draw(0,0)--(1,0)|-(0,1)--cycle;
```

au lieu de joindre le second nœud au troisième

par un segment, nous les joignons par une ligne brisée dont le premier segment est vertical et le second horizontal, ce qui trace donc un carré :



Si nous avons plutôt utilisé `-|`, la ligne brisée aurait eu son premier segment horizontal et son second vertical, ce qui nous aurait fait repasser par l'origine.

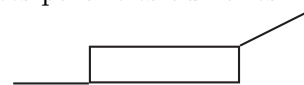
Exemple

- Rectangle :

Exemple

```
\begin{tikzpicture}
\draw
(0,0)--(1,0)rectangle(3,.5)--(4,1);
\end{tikzpicture}
```

trace, entre les deux nœuds concernés, un rectangle à côtés parallèles aux axes :



Exemple

- Cercle et ellipse :

Exemple

```
\begin{tikzpicture}
\draw(0,0)circle(7.5mm);
\draw(-.5,0)circle(2.5mm);
\draw(.25,0)circle(5mm);
\end{tikzpicture}
```

trace trois cercles tangents deux à deux; la mesure entre parenthèses est le *rayon* et non le diamètre.



Exemple

On notera que cette opération *circle* ne modifie pas le nœud courant du chemin, de sorte que dans un cas comme ci-dessous, le tracé se poursuit à partir du centre du cercle. Il en va de même pour l'opération *ellipse*, qui fonctionne de manière analogue; on spécifie les deux demi-axes (parallèles aux axes de coordonnées), en les séparant par `and` :

Exemple

```
\begin{tikzpicture}
\draw(0,0)--(1,.5)
ellipse(1cm and 5mm)-|(2.5,1);
\end{tikzpicture}
```



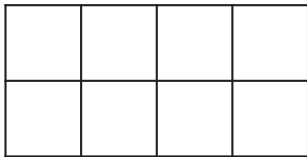
Exemple

- Quadrillage :

Exemple

```
\begin{tikzpicture}
\draw(0,0)grid(4,2);
\end{tikzpicture}
```

trace, entre les deux nœuds concernés, un quadrillage de maille 1 cm :



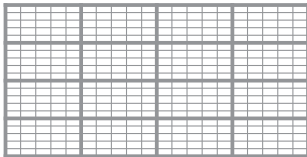
Exemple

Pour modifier la taille des mailles :

Exemple

```
\begin{tikzpicture}
\draw[gray,xstep=2mm,ystep=1mm]
(0,0)grid(4,2);
\draw[gray,very thick,ystep=5mm]
(0,0)grid(4,2);
\end{tikzpicture}
```

donne



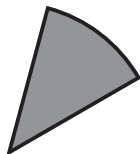
Exemple

- Arc de cercle :

Exemple

```
\begin{tikzpicture}
\filldraw[ultra thick,fill=gray](0,0)
--(1.732,1)arc(30:75:2cm)--cycle;
\end{tikzpicture}
```

trace, et remplit de gris, à partir du dernier nœud, un arc de cercle de rayon 2 cm, partant de l'angle polaire 30° et se terminant à l'angle polaire 75°; le nœud courant du chemin après l'arc est l'extrémité de celui-ci :



Exemple

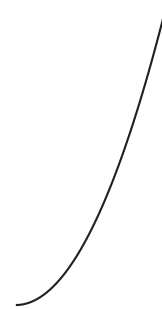
On observera bien, parce que c'est un peu inattendu peut-être, que le nœud courant n'est pas le centre du cercle, mais bien l'origine de l'arc.

- Arc de parabole :

Exemple

```
\begin{tikzpicture}
\draw(0,0)parabola(2,4);
\end{tikzpicture}
```

trace, à partir du dernier nœud, un arc de parabole d'axe vertical; par défaut, le dernier nœud est le sommet :

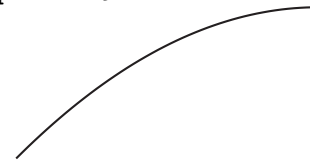


Exemple

On place le sommet au nœud *suivant* en indiquant l'option `bend at end` :

Exemple

```
\begin{tikzpicture}
\draw(0,0)parabola[bend at end](4,2);
\end{tikzpicture}
```

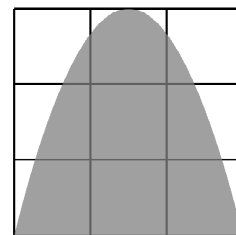


Exemple

Si c'est l'opération *parabola bend* qui est utilisée, le point qui la suit est le sommet, et l'extrémité de l'arc vient ensuite

Exemple

```
\begin{tikzpicture}
\draw[thick](0,0)grid(3,3);
\fill[gray,nearly opaque]
(0,0)parabola bend(1.5,3)(3,0);
\end{tikzpicture}
```



Exemple

(En fait,

```
\draw(a,b)parabola bend(c,d)(e,f);
```

équivaut à

```
\draw(a,b)parabola[bend at end](c,d);
\draw(c,d)parabola(e,f);,
```

de sorte que si c n'est pas la moyenne de a et e , ce sont des arcs de paraboles *différentes* qui seront obtenus : la figure ne sera pas symétrique.)

Dans cet exemple, où l'option `nearly opaque` demande moins de transparence que `semitransparent`, on peut observer l'atténuation du quadrillage derrière le segment de parabole.

– Arc de sinuséide :

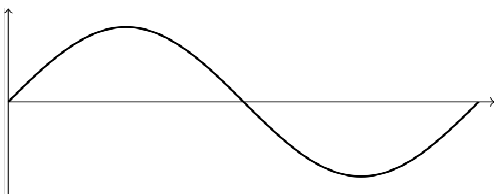
L'opération `sin` trace l'arc de sinuséide correspondant à l'intervalle $[0; \pi/2]$, éventuellement dilaté en abscisse ou en ordonnée ; l'opération `cos` fait de même avec la cosinuséide. Mais évidemment, il s'agit de la même courbe dans les deux cas ; avec `sin`, on part du point d'inflexion et avec `cos` on y aboutit.

Exemple

Avec le code

```
\begin{tikzpicture}
\draw[->] (0,0)--(6.5,0) ;
\draw[->] (0,-1.25)--(0,1.25) ;
\draw[thick] (0,0)sin(1.571,1)cos
(3.142,0)sin(4.712,-1)cos(6.283,0) ;
\end{tikzpicture}
```

nous obtenons



Exemple

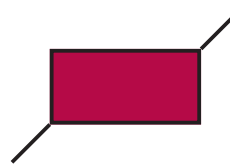
Dans cet exemple, nous découvrons l'option `->` de la commande `\draw`, qui munit le chemin dessiné d'une pointe de flèche à son extrémité.

Positionnement incrémentiel

Dans la construction d'un chemin, il est possible de situer un nœud non de manière absolue, dans le repère, mais par rapport au nœud précédent ; pour ce faire, il suffit de noter `++` avant le couple de ses coordonnées.

Exemple

```
\begin{tikzpicture}
\filldraw[very thick,fill=magenta]
(0,0)--(.5,.5)rectangle++(2,1)--++(.5,.5) ;
\end{tikzpicture}
```



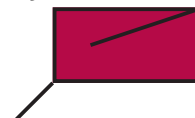
Exemple

De cette façon, nous spécifions le rectangle par ses dimensions, ce qui est sans doute plus parlant.

Il est aussi permis de faire précéder les coordonnées de `+` ; on obtient ici encore un déplacement relatif, mais *sans remise à jour* du nœud de référence pour la suite du chemin. La comparaison de l'exemple ci-dessous avec le précédent devrait rendre cela bien clair.

Exemple

```
\begin{tikzpicture}
\filldraw[very thick,fill=magenta]
(0,0)--(.5,.5)rectangle+(2,1)--+(.5,.5) ;
\end{tikzpicture}
```



Exemple

Coordonnées polaires

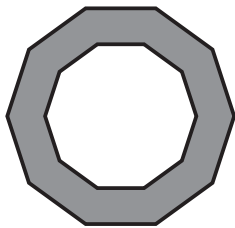
TikZ connaît non seulement les coordonnées cartésiennes, mais également les coordonnées polaires : celles-ci doivent être spécifiées sous la forme $(\text{angle}:\text{rvec})$ où, comme de juste, `angle` est l'angle polaire, en degrés, et `rvec` le rayon vecteur, qui doit être présenté comme une longueur, c'est-à-dire avec son unité. L'esprit est donc le même que lors de l'utilisation de l'opération `arc`, mais il n'y a qu'un seul angle à spécifier.

L'exemple suivant montre comment exploiter cette possibilité pour construire le plus facilement du monde une couronne décagonale ; si nous avions dû la construire en utilisant les coordonnées cartésiennes de ses sommets, nous aurions été obligés de calculer celles-ci, à grand renfort de fonctions trigonométriques.

Exemple

```
\begin{tikzpicture}
\filldraw[thick,fill=gray](0:15mm)--
(36:15mm)--(72:15mm)--(108:15mm)--
(144:15mm)--(180:15mm)--(216:15mm)--
(252:15mm)--(288:15mm)--(324:15mm)--
```

```
(0:15mm)(0:10mm)--(324:10mm)--  
(288:10mm)--(252:10mm)--(216:10mm)--  
(180:10mm)--(144:10mm)--(108:10mm)--  
(72:10mm)--(36:10mm)--(0:10mm);  
\end{tikzpicture}
```



Exemple

Il est possible, bien sûr, d'utiliser les coordonnées polaires dans du positionnement incrémentiel, ainsi que dans le zig-zag à 60° suivant :

Exemple

```
\begin{tikzpicture}  
\draw[ultra thick](0,0)--++(60:8mm)  
--++(-60:8mm)--++(60:8mm)--++(-60:8mm)  
--++(60:8mm)--++(-60:8mm)--++(60:8mm)  
--++(-60:8mm)--++(60:8mm)--++(-60:8mm);  
\end{tikzpicture}
```



Exemple

Pour en savoir plus

Le lecteur l'aura déjà compris, TikZ foisonne de commandes et chacune d'elles admet de multiples options. Dans ce premier article, nous avons seulement voulu indiquer les mécanismes de base. Quelques-unes des caractéristiques les plus intéressantes ne seront présentées que dans une publication à venir.

Indiquons cependant déjà trois références que le lecteur pourra trouver sur la toile et consulter s'il souhaite, tout de suite, en savoir plus.

- Le manuel officiel est disponible à l'adresse : <http://ftp.snt.utwente.nl/pub/software/tex/graphics/pgf/base/doc/pgfmanual.pdf>. Par essence, il contient tout, donc n'est pas le plus abordable, ni le plus commode d'emploi lorsqu'on a besoin d'une référence rapide.
- Un texte plus simple, pas irréprochable sans doute, mais qui a le mérite d'exister (et d'être en français) : *TikZ pour l' impatient*. <http://math.et.info.free.fr/TikZ/>
- Une courte brochure, présentant les notions de base et les illustrant d'exemples : *Minimal TikZ*. <http://cremeronline.com/LaTeX/minimaltikz.pdf>

Pascal DUPONT est chargé de cours à HEC•ULg. ✉ pascal.dupont@ulg.ac.be.



Le carillon du beffroi de Gembloux, inscrit en 2005 sur la liste du patrimoine mondial de l'UNESCO. <http://www.gembloux.be/loisirs/tourisme/gembloux-villages/gembloux-2/le-carillon-du-beffroi>