

## Figures mathématiques avec TikZ (fin)

Ceux qui veulent peuvent travailler sur le document libre au lieu de faire cette feuille de TP. Regarder le § 10.1 pourra néanmoins être utile pour la figure obligatoire du document libre.

### 10.1 Notion de style

On peut passer des options globales à tous les nodes d'une figure. Cela se fait de la façon suivante :

```
\begin{tikzpicture}[every node/.style={text=red,font=\ttfamily\tiny\bfseries}]
\node at (0,0) {c};
\node[above] at (0,0) {h};
\node[below] at (0,0) {b};
\node[left] at (0,0) {g};
\node[right] at (0,0) {r};
\end{tikzpicture}
```

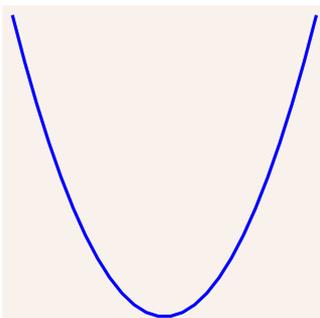
Pour d'autres options de style, voir le manuel de TikZ page 152-154. Pour que le style affecte globalement tous les `tikzpicture` du document, il suffit de mettre, dans le préambule,

```
\tikzset{every node/.style={text=red,font=\ttfamily\tiny\bfseries}}
```

Une autre possibilité utile est de définir un style avec un nom que l'on peut ensuite utiliser comme n'importe quel style. Par exemple, si on fait un document avec beaucoup de tracé de fonctions, on peut vouloir définir un style (dans le préambule) :

```
\tikzstyle{plotted function}=[color=blue,very thick]
```

pour ensuite utiliser



```
\shorthandoff{:}
\begin{tikzpicture}
\draw[plotted function,domain=-2:2] plot ({\x},{\x*\x});
\end{tikzpicture}
\shorthandon{:}
```

**Compléments.** — Ce genre de choses permet par exemple de redéfinir les épaisseurs de traits de tout le document. Voici par exemple des épaisseurs adaptées au package `txfonts` :

```
\tikzstyle{ultra thin}= [line width=0.2pt]
\tikzstyle{very thin}= [line width=0.4pt]
\tikzstyle{thin}=      [line width=0.58pt]
\tikzstyle{semithick}= [line width=0.75pt]
\tikzstyle{thick}=     [line width=0.9pt]
\tikzstyle{very thick}= [line width=1.2pt]
\tikzstyle{ultra thick}= [line width=1.6pt]
```

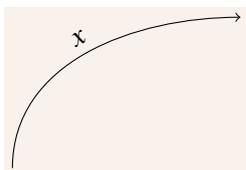
Pour changer le style de flèches<sup>1</sup> de tout le document, on peut faire :

```
\pgfarrowsdeclaimalias{<}{>}{latex}{latex}
```

Avant de changer de style de flèche, faire attention à garder la cohérence du document. Utiliser des flèches de type  $\rightarrow$  alors que toutes les autres flèches du document seront comme  $\longrightarrow$  (dans les maths, par exemple) n'est pas une bonne idée, l'un des avantages de TikZ étant justement d'être plutôt bien configuré par défaut, contrairement à d'autres solutions graphiques. Par exemple, avec `lmodern`, il ne faut surtout pas changer le style de flèches par défaut.

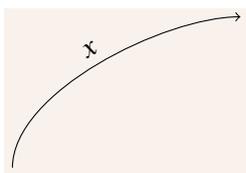
## 10.2 Faire des flèches incurvées

Il y a deux possibilités pour faire des flèches incurvées avec TikZ. La première est l'utilisation de `to` avec spécification de l'inclinaison de départ et d'arrivée :



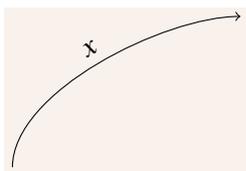
```
\begin{tikzpicture}
\draw[->] (0,0) to[out=90,in=180] (3,2) node[above,sloped,pos=0.5] {$x$};
\end{tikzpicture}
```

On peut aussi utiliser `..` et `controls` pour relier deux points :



```
\begin{tikzpicture}
\draw[->] (0,0) .. controls (0,1) and (2,2)
.. (3,2) node[above,sloped,pos=0.5] {$x$};
\end{tikzpicture}
```

Les points donnés par `controls` contrôlent la direction et la force de tangence de la courbe. On peut les donner avec des syntaxes plus agréables :



```
\begin{tikzpicture}
\draw[->] (0,0) .. controls +(up:1cm) and +(left:1cm)
.. (3,2) node[above,sloped,pos=0.5] {$x$};
\end{tikzpicture}
```

(Cette syntaxe signifie seulement que le point (0, 1) est 1cm au-dessus de (0, 0) tandis que le point (2, 2) est 1cm à gauche de (3, 2).)

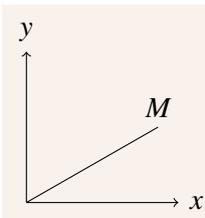
**Exercice 1.** — Reproduire le tableau suivant.

COS	
0	1
$\frac{\pi}{6}$	$\frac{\sqrt{3}}{2}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$
$\frac{\pi}{3}$	$\frac{1}{2}$
$\frac{\pi}{2}$	0

**Solution de l'exercice 1.** — Voici le code :

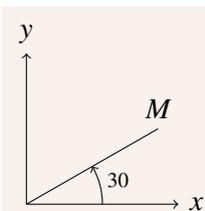
```
\begin{tabular}{|c|c|}
\multicolumn{2}{c}{\begin{tikzpicture}
\draw[->] (0,0) to[out=90,in=90] (1,0) node[above,sloped,pos=0.5] {\cos$};
\end{tikzpicture}} \\
\hline
$0$ & $1$ \\
\hline
$\frac{\pi}{6}$ & $\frac{\sqrt{3}}{2}$ \\
\hline
$\frac{\pi}{4}$ & $\frac{\sqrt{2}}{2}$ \\
\hline
$\frac{\pi}{3}$ & $\frac{1}{2}$ \\
\hline
$\frac{\pi}{2}$ & $0$ \\
\hline
\end{tabular}
```

Si on veut marquer des angles, il peut être utile d'avoir recours aux coordonnées polaires. Le point situé à une distance de 2cm de (0, 0) et faisant un angle de 30° avec l'axe des abscisses  $Ox$  peut être affiché de la façon suivante :



```
\begin{tikzpicture}
\draw[->] (0,0) -- (2,0) node[right] {$x$};
\draw[->] (0,0) -- (0,2) node[above] {$y$};
\draw (0,0) -- (30:2cm) node[above] {$M$};
\end{tikzpicture}
```

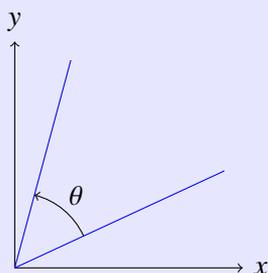
Si on veut marquer l'angle, on peut utiliser la forme arc :



```
\begin{tikzpicture}
\draw[->] (0,0) -- (2,0) node[right] {$x$};
\draw[->] (0,0) -- (0,2) node[above] {$y$};
\draw (0,0) -- (30:2cm) node[above] {$M$};
\draw[->] (0:1cm) arc (0:30:1cm);
\node[font=\scriptsize] at (15:1.25cm) {30};
\end{tikzpicture}
```

Dans  $(0:1cm)$  arc  $(0:30:1cm)$ , le premier point est le point de départ, ici  $(0:1cm)$  ce qui est la même chose que  $(0, 1)$ . Ensuite,  $(0:30:1cm)$  signifie que l'arc va être construit comme s'il provenait d'un cercle de 1cm de diamètre entre les angles 0 et 30 (en degrés) de ce cercle.

**Exercice 2.** — Reproduire la figure suivante :



**Solution de l'exercice 2.** — Voici le code :

```
\begin{tikzpicture}
\draw[->] (0,0) -- (3,0) node[right] {$x$};
\draw[->] (0,0) -- (0,3) node[above] {$y$};
\begin{scope}
\clip (0,0) rectangle (2.75,2.75);
\draw[color=blue] (0,0) -- (75:4cm);
\draw[color=blue] (0,0) -- (25:4cm);
\end{scope}
\draw[->] (25:1cm) arc (25:75:1cm);
\node at (50:1.25cm) {$\theta$};
\end{tikzpicture}
```

### 10.3 Faire des dégradés

On peut, lorsqu'on trace un objet spécifier un couleur non uniforme. Par exemple,

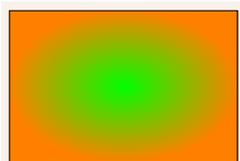


```
\begin{tikzpicture}
\draw[left color=red,right color=violet] (0,0) rectangle (3,2);
\end{tikzpicture}
```

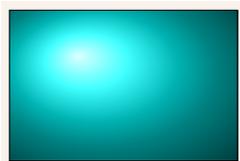
Il y a plein d'autres options que `left color` et `right color`. Voici quelques exemples.



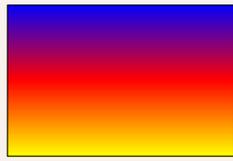
```
\begin{tikzpicture}
\draw[top color=blue,bottom color=yellow] (0,0) rectangle (3,2);
\end{tikzpicture}
```



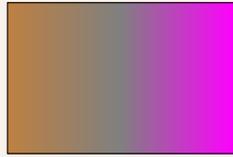
```
\begin{tikzpicture}
\draw[inner color=green,outer color=orange] (0,0) rectangle (3,2);
\end{tikzpicture}
```



```
\begin{tikzpicture}
\draw[ball color=cyan] (0,0) rectangle (3,2);
\end{tikzpicture}
```



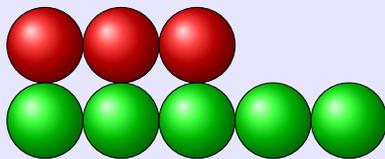
```
\begin{tikzpicture}
\draw[top color=blue,bottom color=yellow,middle color=red]
(0,0) rectangle (3,2);
\end{tikzpicture}
```



```
\begin{tikzpicture}
\draw[left color=brown,right color=magenta,middle color=gray]
(0,0) rectangle (3,2);
\end{tikzpicture}
```

Noter que `middle color` doit être donnée en dernier pour avoir un effet (car les `top color` et `bottom color` changent automatiquement le `middle color`).

**Exercice 3.** — Reproduire le dessin suivant.



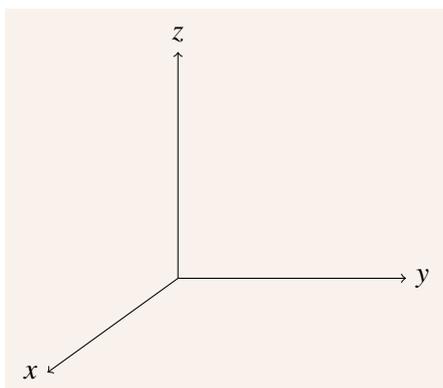
**Solution de l'exercice 3.** — Voici le code de la figure précédente :

```
\begin{tikzpicture}
\draw[ball color=red] (0,0) circle (0.5cm);
\draw[ball color=red] (1,0) circle (0.5cm);
\draw[ball color=red] (2,0) circle (0.5cm);
\draw[ball color=green] (0,-1) circle (0.5cm);
\draw[ball color=green] (1,-1) circle (0.5cm);
\draw[ball color=green] (2,-1) circle (0.5cm);
\draw[ball color=green] (3,-1) circle (0.5cm);
\draw[ball color=green] (4,-1) circle (0.5cm);
\end{tikzpicture}
```

## 10.4 Initiation à la 3D

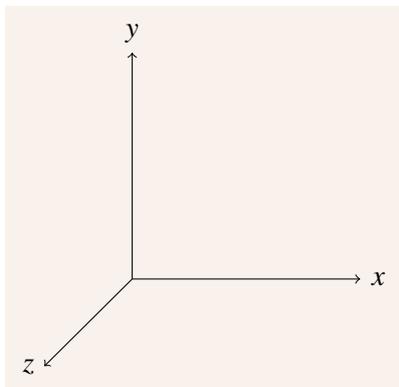
Jusqu'à maintenant, on a toujours travaillé en 2 dimensions, mais TikZ peut faire des graphiques en 3 dimensions, bien que les possibilités soient assez basiques (pas de surfaces 3D, pas de commandes pour les solides, etc.), mais bien suffisantes pour des dessins simples.

Voyons tout d'abord comment faire des axes d'un repère en 3 dimensions.



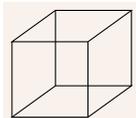
```
\begin{tikzpicture}[x={(-0.572cm,-0.416cm)},y={(1cm,0cm)},z={(0cm,1cm)}]
\draw[->] (0,0,0)--(3,0,0) node[left] {$x$};
\draw[->] (0,0,0)--(0,3,0) node[right] {$y$};
\draw[->] (0,0,0)--(0,0,3) node[above] {$z$};
\end{tikzpicture}
```

(Les valeurs de 0.572 correspond environ au cosinus de  $55^\circ$  et  $0.416 = \sqrt{0.5 - (0.572)^2}$ .) Si on ne spécifie rien, les axes sont orientés en vue plongeante :



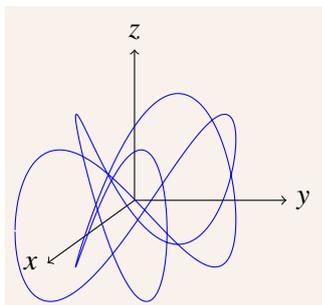
```
\begin{tikzpicture}
\draw[->] (0,0,0)--(3,0,0) node[right] {$x$};
\draw[->] (0,0,0)--(0,3,0) node[above] {$y$};
\draw[->] (0,0,0)--(0,0,3) node[left] {$z$};
\end{tikzpicture}
```

Voici comment tracer par exemple un cube, où l'on ne spécifie pas les valeurs de  $x$ ,  $y$  et  $z$ , ce qui fait que celles par défaut sont utilisées :



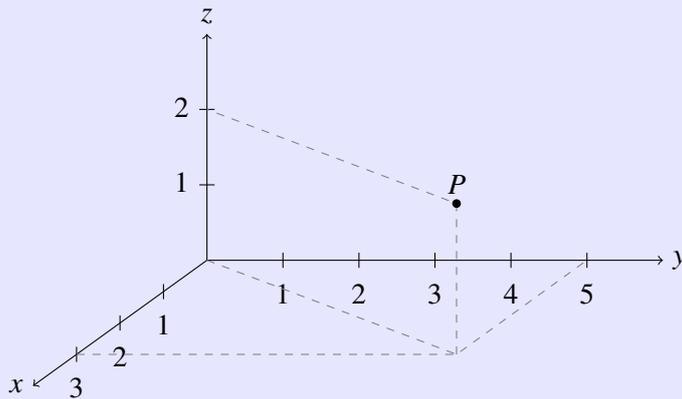
```
\begin{tikzpicture}[x={(-0.572cm,-0.416cm)},y={(1cm,0cm)},z={(0cm,1cm)}]
\draw (0,0,0)--(1,0,0)--(1,1,0)--(0,1,0)--cycle;
\draw (0,0,1)--(1,0,1)--(1,1,1)--(0,1,1)--cycle;
\draw (0,0,0)--(0,0,1);
\draw (1,0,0)--(1,0,1);
\draw (0,1,0)--(0,1,1);
\draw (1,1,0)--(1,1,1);
\end{tikzpicture}
```

Tracer une courbe paramétrée en 3D ne pose aucun problème :



```
\shorthandoff{:}
\begin{tikzpicture}[x={(-0.572cm,-0.416cm)},y={(1cm,0cm)},z={(0cm,1cm)}]
\draw[->] (0,0,0)--(2,0,0) node[left] {$x$};
\draw[->] (0,0,0)--(0,2,0) node[right] {$y$};
\draw[->] (0,0,0)--(0,0,2) node[above] {$z$};
\draw[domain=-pi:pi,samples=500,color=blue] plot ({cos(2*x r)},{cos(3*x r)},{sin(5*x r)});
\end{tikzpicture}
\shorthandon{:}
```

**Exercice 4.** — Reproduire le dessin suivant :



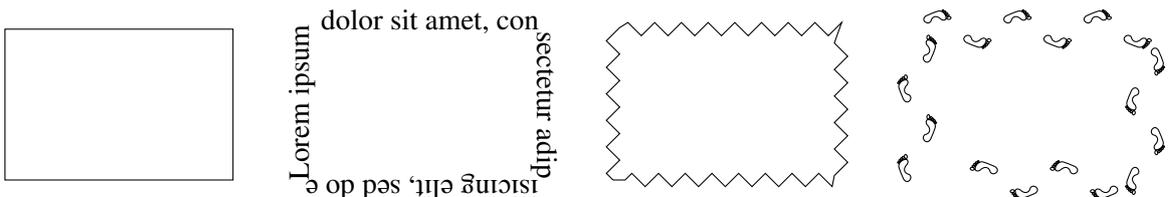
**Solution de l'exercice 4.** — Voici le code du dessin précédent :

```
\begin{tikzpicture}[x={(-0.572cm,-0.416cm)},y={(1cm,0cm)},z={(0cm,1cm)}]
\newcommand{\ticsize}{0.1cm/28.35}% ne pas oublier de diviser par 28.35
\draw[->] (0,0,0)--(4,0,0) node[left] {$x$};
\draw[->] (0,0,0)--(0,6,0) node[right] {$y$};
\draw[->] (0,0,0)--(0,0,3) node[above] {$z$};
\foreach \x in {1,2,3} {
  \draw (\x,0,\ticsize) -- (\x,0,-\ticsize) node[below] {$\x\strut$};
}
\foreach \y in {1,2,...,5} {
  \draw (0,\y,\ticsize) -- (0,\y,-\ticsize) node[below] {$\y\strut$};
}
\foreach \z in {1,2} {
  \draw (0,\ticsize,\z) -- (0,-\ticsize,\z) node[left] {$\z\strut$};
}
\draw[dashed,gray] (3,0,0)--(3,5,0)--(0,5,0);
\draw[dashed,gray] (3,5,0)--(3,5,2)--(0,0,2);
\draw[dashed,gray] (0,0,0)--(3,5,0);
\draw[fill=black] (3,5,2) circle (0.05cm) node[above] {$P$};
\end{tikzpicture}
```

Noter qu'il faut diviser certaines longueurs en centimètres par un facteur 28.35 (pour la même raison déjà mentionnée à la toute fin de la séance n° 5).

## 10.5 Décorations

Les décorations sont un moyen très puissant pour modifier l'apparence d'un trait reliant deux points. Par exemple, pour obtenir les dessins suivants, la seule chose qui change est une simple option :



Pour utiliser les décorations, il faut charger la librairie correspondante :

```
\usetikzlibrary{decorations}
```

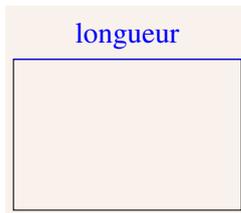
Il peut aussi, selon les options voulues, falloir l'une des sous-librairies suivantes :

```
\usetikzlibrary{decorations.pathmorphing}
\usetikzlibrary{decorations.pathreplacing}
\usetikzlibrary{decorations.shapes}
\usetikzlibrary{decorations.text}
\usetikzlibrary{decorations.markings}
\usetikzlibrary{decorations.fractals}
\usetikzlibrary{decorations.footprints}
```

L'une des décorations les plus utile est celle permettant de faire des accolades pour indiquer une longueur (par exemple). Voyons par exemple comment reproduire le dessin suivant :



Le tracé du rectangle ne pose aucun problème, c'est juste `\draw (0,0) rectangle (3,2);`. Pour l'accolade du haut, on trace d'abord une simple ligne bleue sur le dessus du rectangle :



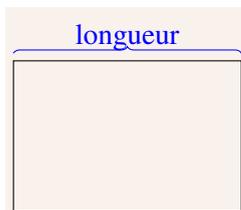
```
\begin{tikzpicture}
\draw (0,0) rectangle (3,2);
\draw[color=blue] (0,2) -- (3,2) node[above,pos=0.5] {longueur};
\end{tikzpicture}
```

puis on la décore ; pour cela, on rajoute l'option `decorate` et on choisit la décoration `brace` :



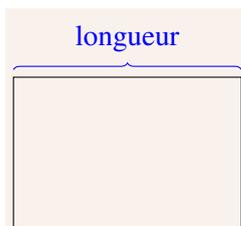
```
\begin{tikzpicture}
\draw (0,0) rectangle (3,2);
\draw[color=blue,decorate,decoration={brace}]
(0,2) -- (3,2) node[above,pos=0.5] {longueur};
\end{tikzpicture}
```

Finalement, on va rehausser l'accolade avec l'option `raise` :



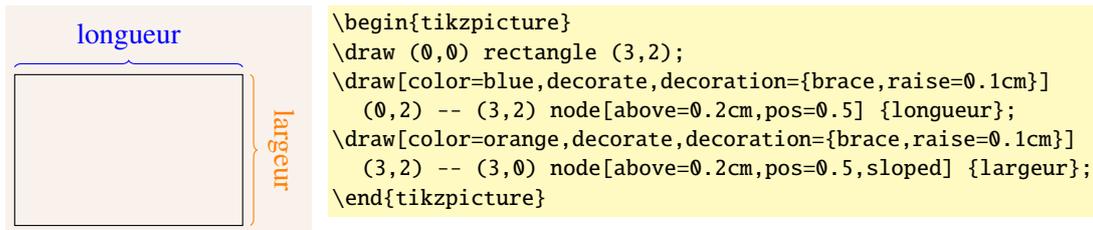
```
\begin{tikzpicture}
\draw (0,0) rectangle (3,2);
\draw[color=blue,decorate,decoration={brace,raise=0.1cm}]
(0,2) -- (3,2) node[above,pos=0.5] {longueur};
\end{tikzpicture}
```

Le texte est maintenant trop bas, donc on le rehausse un peu :

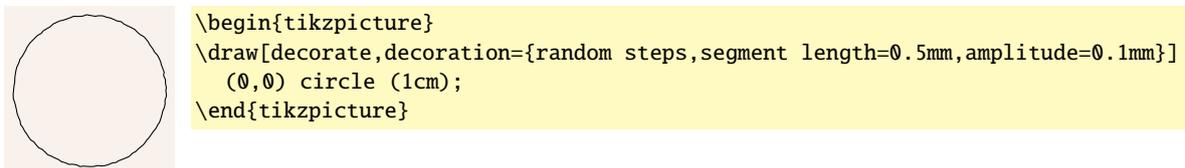
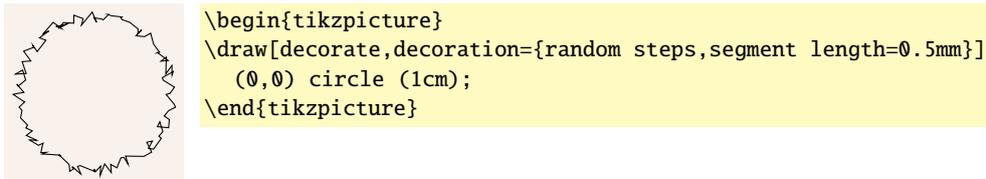


```
\begin{tikzpicture}
\draw (0,0) rectangle (3,2);
\draw[color=blue,decorate,decoration={brace,raise=0.1cm}]
(0,2) -- (3,2) node[above=0.2cm,pos=0.5] {longueur};
\end{tikzpicture}
```

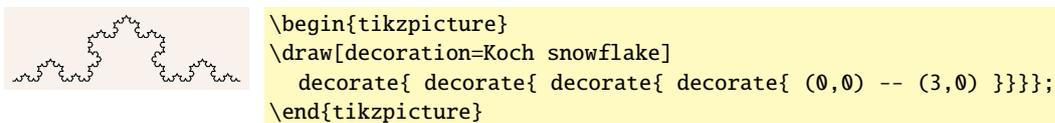
Finalement, on s'occupe de la deuxième accolade de la même façon, en utilisant l'option `sloped` pour pencher le texte du node :



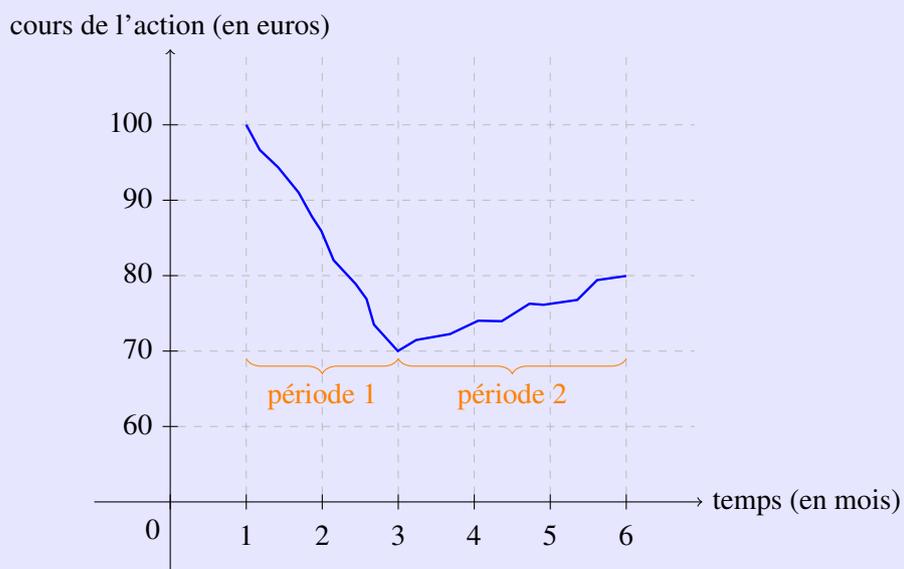
Une autre option utile est celle permettant de rendre un chemin légèrement aléatoire (par exemple pour simuler un phénomène aléatoire ou donner un caractère « dessiné à la main » à une figure) :



Terminons par un dernier type de décoration, les décorations fractales (voir page 270 du manuel de TikZ) :



**Exercice 5.** — Reproduire la figure suivante. Pour faire la grille, on pourra utiliser la commande `grid` qui s'utilise comme `rectangle`



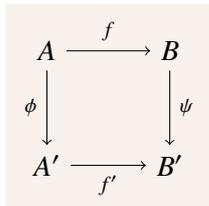
**Solution de l'exercice 5.** — Voici le code de la figure précédente :



de placer les flèches entre les différents éléments, on va leur donner un nom. Cela se fait en mettant juste avant, dans la même case, une étiquette du genre |(haut-gauche)| ou |(A)| si on veut un nom plus explicite. Par exemple,

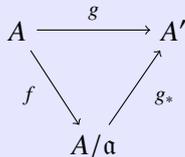
$A$	$B$	<pre>\begin{tikzpicture} \matrix[matrix of math nodes,row sep=1cm,column sep=1cm]{  (A)       A &amp;  (B)       B \\  (Aprime)  A' &amp;  (Bprime)  B' \\ }; \end{tikzpicture}</pre>
$A'$	$B'$	

On peut maintenant tracer une flèche reliant  $A$  à  $B$  simplement en faisant `\draw[->] (A) -- (B)`; Au lieu de mettre `pos=0.5` et `font=` à tous les nodes, il vaut mieux utiliser un style `every node` ainsi que vu au § 10.1, à l'intérieur d'un scope afin de limiter l'action de ce style :

	<pre>\begin{tikzpicture} \matrix[matrix of math nodes,row sep=1cm,column sep=1cm]{  (A)       A &amp;  (B)       B \\  (Aprime)  A' &amp;  (Bprime)  B' \\ }; \begin{scope}[every node/.style={pos=0.5,font=\scriptsize}] \draw[-&gt;] (A)      -- (B)      node[auto=left]  {\$f\$} ; \draw[-&gt;] (A)      -- (Aprime) node[auto=right] {\$\phi\$}; \draw[-&gt;] (Aprime) -- (Bprime) node[auto=right] {\$f'\$} ; \draw[-&gt;] (B)      -- (Bprime) node[auto=left]  {\$\psi\$}; \end{scope} \end{tikzpicture}</pre>
---	--

Les options `auto=left` et `auto=right` servent à placer automatiquement le label dans le sens de la flèche, soit à la gauche de la direction dans laquelle elle pointe, soit à la droite de cette direction. Cela évite d'avoir à hésiter entre `above` et `below` ou entre `left` et `right`.

**Exercice 6.** — Reproduire le diagramme suivant :

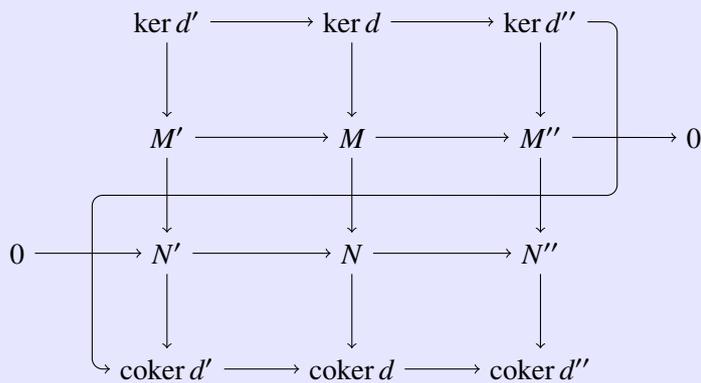


**Solution de l'exercice 6.** — Voici le code de la figure précédente.

```
\begin{tikzpicture}
\matrix[matrix of math nodes,row sep=1cm,column sep=0.33cm]{
|(A)| A\phantom{A'} & & |(Aprime)| A' \\
& & & & |(A/a)| A/\mathfrak{a} & & \\
};
\begin{scope}[every node/.style={font=\scriptsize}]
\draw[->] (A)      -- (Aprime) node[pos=0.5,auto=left]  {$g$} ;
\draw[->] (A)      -- (A/a)   node[pos=0.3,auto=right] {$f\strut$} ;
\draw[->] (A/a)   -- (Aprime) node[pos=0.7,auto=right] {$g_*\strut$};
\end{scope}
\end{tikzpicture}
```

Noter l'utilisation de `\phantom{A'}` dans la case en haut à gauche de la matrice pour éviter que la flèche du haut ne soit légèrement penchée. Noter aussi l'utilisation de `\strut` pour  $f$  et  $g_*$  afin d'assurer un alignement parfait entre eux et l'utilisation de `pos=0.3` et `pos=0.7` au lieu de `pos=0.5` pour tenir compte de la pente.

**Exercice 7 (difficile).** — Reproduire le diagramme du « lemme du serpent » :



Pour faire la flèche en forme de serpent, on pourra utiliser l'option `rounded corners`, des calculs sur les coordonnées ainsi que l'option `++` qui permet de faire dépendre le point du précédent; par exemple, pour tracer une ligne de 1cm vers la droite d'un point (3,4), on peut faire `\draw (3,4) -- ++(1,0)`; au lieu de `\draw (3,4) -- (4,4)`. On utilisera aussi dans le préambule la commande `\DeclareMathOperator{coker}{coker}` pour faire les conoyaux.

**Solution de l'exercice 7.** — Voici le code de la figure précédente :

```
\begin{tikzpicture}
\matrix[matrix of math nodes,row sep=1cm,column sep=1cm]{
& |(kerd')| \ker d' & |(kerd)| \ker d\phantom{d'} & |(kerd'')| \ker d'' \\\
& |(M')| M' & |(M)| M\phantom{M'} & |(M'')| M'' & |(right0)| 0\phantom{M'} \\\
|(left0)| 0\phantom{N'} & |(N')| N' & |(N)| N\phantom{N'} & |(N'')| N'' \\\
& |(cokerd')| \coker d' & |(cokerd)| \coker d\phantom{d'} & |(cokerd'')| \coker d'' \\\
};
\begin{scope}[every node/.style={font=\scriptsize}]
% flèches verticales
\draw[->] (kerd') -- (M');
\draw[->] (kerd) -- (M);
\draw[->] (kerd'') -- (M'');
\draw[->] (M') -- (N');
\draw[->] (M) -- (N);
\draw[->] (M'') -- (N'');
\draw[->] (N') -- (cokerd');
\draw[->] (N) -- (cokerd);
\draw[->] (N'') -- (cokerd'');
% flèches horizontales
\draw[->] (kerd') -- (kerd);
\draw[->] (kerd) -- (kerd'');
\draw[->] (M') -- (M);
\draw[->] (M) -- (M'');
\draw[->] (M'') -- (right0);
\draw[->] (left0) -- (N');
\draw[->] (N') -- (N);
\draw[->] (N) -- (N'');
\draw[->] (cokerd') -- (cokerd);
\draw[->] (cokerd) -- (cokerd'');
% flèche en serpent
\draw[->,rounded corners]
(kerd'')
-- ($ (kerd'') + 0.5*(right0) - 0.5*(M'') $)
```

```

-- ++($M'' - (kerd'') + 0.5*(N'') - 0.5*(M'')$)
-- ++($0.5*(M'') - 0.5*(right0) + (M') - (M'') + 0.5*(left0) - 0.5*(N')$)
-- ++($cokerd' - (N') + 0.5*(N') - 0.5*(M')$)--
(cokerd');
\end{scope}
\end{tikzpicture}

```