

Immersion dans TikZ

Nicolas KIELBASIEWICZ

1er février 2024

TikZ ist kein Zeichenprogramm

TikZ ist kein Zeichenprogramm

(=TikZ n'est pas un logiciel de dessin)

L'extension TIKZ, et sa couche bas niveau PGF (= *Portable Graphic Format*) permettent de dessiner toutes sortes de choses.

- ▶ C'est une extension \LaTeX , et pas un logiciel de dessin.
- ▶ On ne dessine pas à la souris (WYSIWYM) : ce n'est pas un logiciel de dessin.
- ▶ On doit donc programmer un dessin ...et le compiler.

TikZ ist kein Zeichenprogramm

(=TikZ n'est pas un logiciel de dessin)

L'extension TIKZ, et sa couche bas niveau PGF (= *Portable Graphic Format*) permettent de dessiner toutes sortes de choses, depuis 2005.

- ▶ C'est une extension L^AT_EX, et pas un logiciel de dessin.
- ▶ On ne dessine pas à la souris (WYSIWYM) : ce n'est pas un logiciel de dessin.
- ▶ On doit donc programmer un dessin ...et le compiler.

Charger TikZ

```
\usepackage{tikz} % chargement du noyau de base de TikZ  
  
\usetikzlibrary{shapes,decorations,math,shadings} % chargement de bibliothèques additionnelles.  
% pas d'espaces entre les noms et les virgules
```

TikZ ist kein Zeichenprogramm

Exemples

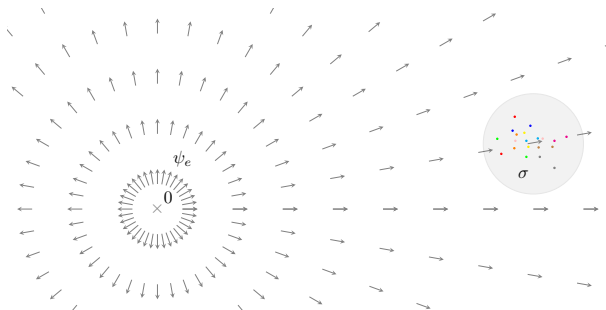



Figure 1 : Source :  KIELBASIEWICZ, « Dynamique gravitationnelle multi-échelle : formation et évolution des systèmes auto-gravitants non isolés »

TikZ ist kein Zeichenprogramm

Exemples

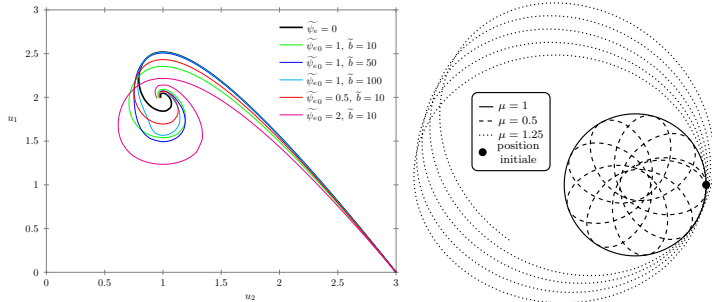



Figure 1 : Source :  KIELBASIEWICZ, « Dynamique gravitationnelle multi-échelle : formation et évolution des systèmes auto-gravitants non isolés »

TikZ ist kein Zeichenprogramm

Exemples

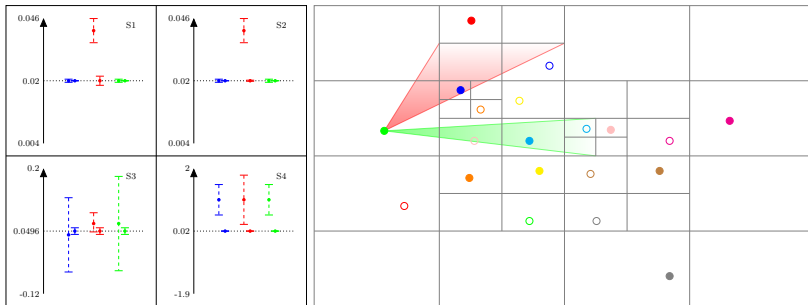



Figure 1 : Source :  KIELBASIEWICZ, « Dynamique gravitationnelle multi-échelle : formation et évolution des systèmes auto-gravitants non isolés »

TikZ ist kein Zeichenprogramm

Exemples

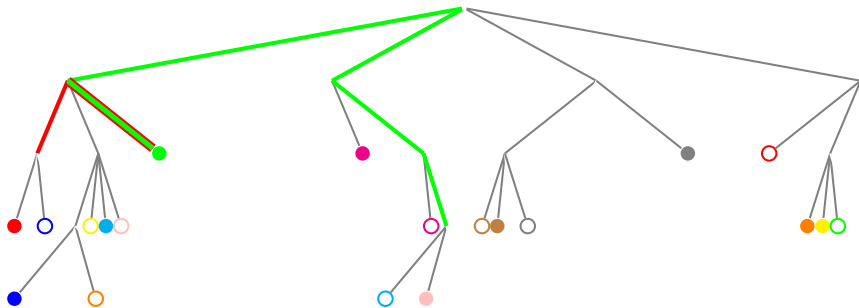



Figure 1 : Source :  KIELBASIEWICZ, « Dynamique gravitationnelle multi-échelle : formation et évolution des systèmes auto-gravitants non isolés »

TikZ ist kein Zeichenprogramm

Exemples

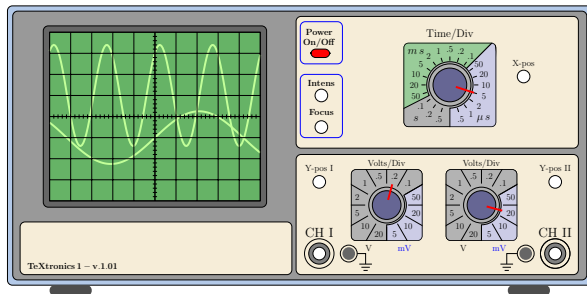


Figure 2 : Source : www.texample.net/tikz/examples/

TikZ ist kein Zeichenprogramm

Exemples

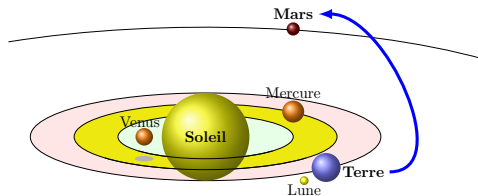
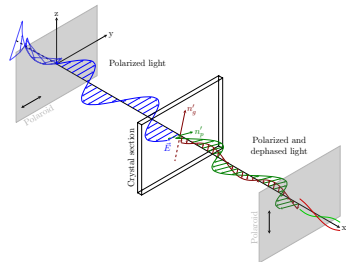
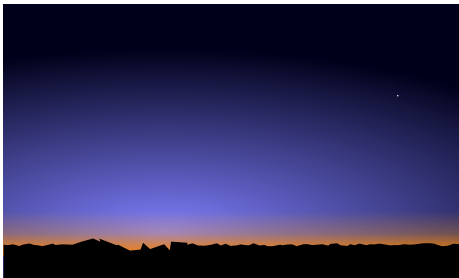


Figure 2 : Source : www.texample.net/tikz/examples/

TikZ ist kein Zeichenprogramm

Exemples



Light behavior in a petrographic microscope with light polarizing device. Only one incident wavelength is shown (monochromatic light). The magnetic field, perpendicular to the electric one, is not drawn.

Figure 2 : Source : www.texample.net/tikz/examples/

Bonnes pratiques

Bonnes pratiques

Générateurs de code TikZ

Gnuplot 2 possibilités : choisir `set terminal table` dans votre script gnuplot et charger le fichier de données dans TikZ. Autre solution : `set terminal tikz`

Bonnes pratiques

Générateurs de code TikZ

Gnuplot 2 possibilités : choisir `set terminal table` dans votre script gnuplot et charger le fichier de données dans TikZ. Autre solution : `set terminal tikz`

GeoGebra Application multiplateforme dédiée à la géométrie de base et utilisée dans l'enseignement de maths au lycée. Cela permet de faire des dessins à la souris et de les exporter en code TikZ.

Bonnes pratiques

Générateurs de code TikZ

Gnuplot 2 possibilités : choisir `set terminal table` dans votre script gnuplot et charger le fichier de données dans TikZ. Autre solution : `set terminal tikz`

GeoGebra Application multiplateforme dédiée à la géométrie de base et utilisée dans l'enseignement de maths au lycée. Cela permet de faire des dessins à la souris et de les exporter en code TikZ.

Matlab Le module `matlab2tikz` permet d'exporter une figure Matlab en TikZ. Il suffit d'invoquer la commande `matlab2tikz('myfile.tikz');` après avoir tracé votre figure en Matlab.

Pour le télécharger : <https://github.com/matlab2tikz/matlab2tikz>

Bonnes pratiques

Externaliser la génération des figures TikZ

- ▶ Une grosse figure TikZ consomme du temps de compilation.
- ▶ Une figure TikZ terminée est quand même recompilée à chaque fois que l'on compile le document.

Bonnes pratiques

Externaliser la génération des figures TikZ

- ▶ Une grosse figure TikZ consomme du temps de compilation.
- ▶ Une figure TikZ terminée est quand même recompilée à chaque fois que l'on compile le document.

Comment faire pour optimiser le temps de compilation ?

Bonnes pratiques

Externaliser la génération des figures TikZ

- ▶ Une grosse figure TikZ consomme du temps de compilation.
- ▶ Une figure TikZ terminée est quand même recompilée à chaque fois que l'on compile le document.

Comment faire pour optimiser le temps de compilation ?

Utiliser la librairie TikZ `EXTERNAL`, qui permet de générer un pdf de chaque figure TikZ ciblée et d'inclure ce pdf ensuite. Il y a un système qui permet de savoir s'il faut régénérer le pdf ou non.

Bonnes pratiques

Externaliser la génération des figures TikZ

- ▶ Une grosse figure TikZ consomme du temps de compilation.
- ▶ Une figure TikZ terminée est quand même recompilée à chaque fois que l'on compile le document.

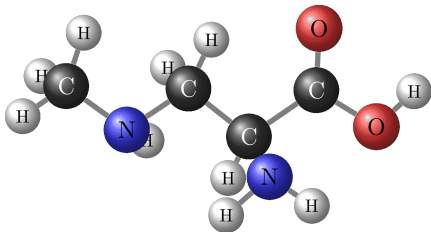
Utilisation de la librairie `EXTERNAL` dans une présentation


```
\usetikzlibrary{external}  
\tikzexternalize[prefix=mesimages/] % le dossier qui contiendra les images  
...  
\begin{document}  
\tikzexternaldisable % désactiver l'externalisation partout  
...  
\tikzexternalenable % réactiver l'externalisation ponctuellement  
\input{monimage.tikz}  
\tikzexternaldisable  
...
```

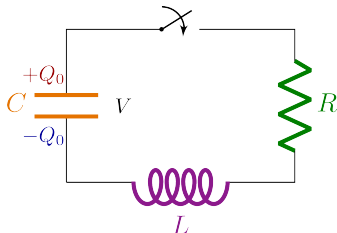
Bonnes pratiques


Pour aller plus loin

- ▶ La doc de TikZ (~1300 pages!!!)
- ▶ www.texample.net/tikz/examples/
- ▶ www.tikz.net/



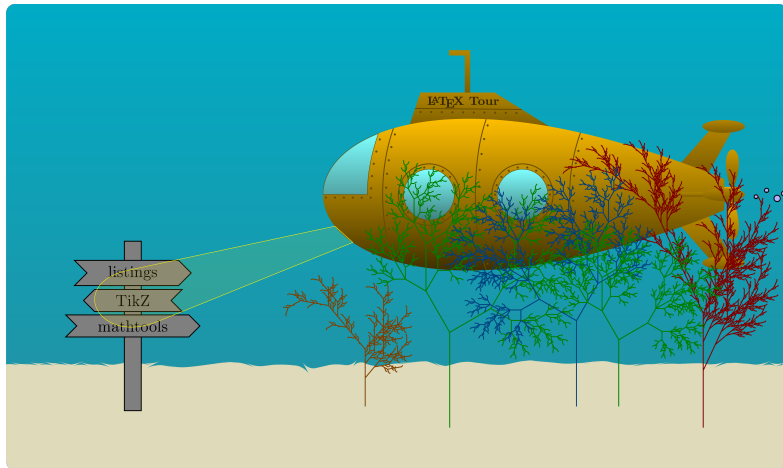
 TELLECHEA, *chemfig* – Draw molecules with easy syntax



 REDAELLI et al., *CircuiTikZ* – Draw electrical networks with TikZ

Bonnes pratiques

Pour aller plus loin



Commandes de base

Commandes de base

Lignes et surfaces



Coordonnées et nœuds

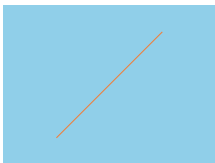
```
\begin{tikzpicture}  
  \fill[tPrim!50!white] (0,0) rectangle (4,3);
```

```
\end{tikzpicture}
```

\fill C'est la commande pour dessiner une surface colorée. On peut choisir la couleur du contour avec l'option **draw**.

Commandes de base

Lignes et surfaces



Coordonnées et nœuds

```
\begin{tikzpicture}
\fill[tPrim!50!white] (0,0) rectangle (4,3);
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\coordinate (p3) at (1,2);
\draw[tAccent] (p1) -- (p2);

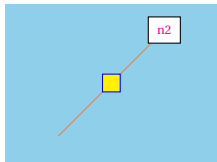
\end{tikzpicture}
```

\draw C'est la commande pour dessiner une surface au contour coloré. On peut choisir la couleur du fond avec l'option **fill**.

\coordinate C'est la commande pour donner un nom à un point. **Notion de variable**

Commandes de base

Définir des nœuds



Coordonnées et nœuds

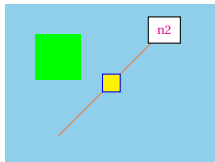
```
\begin{tikzpicture}
\fill[tPrim!50!white] (0,0) rectangle (4,3);
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\coordinate (p3) at (1,2);
\draw[tAccent] (p1) -- (p2);
\node[fill=yellow, draw=blue] (n1) at (2,1.5) {};
\node[fill=white, draw, text=magenta, font=\tiny] (n2) at (p2) {n2};

\end{tikzpicture}
```

\node C'est la commande pour définir un nœud, une boîte contenant du texte (plus généralement du code \LaTeX). Un nœud vide a une taille, car il y a du padding.

Commandes de base

Définir des nœuds



Coordonnées et nœuds

```
\begin{tikzpicture}
\fill[tPrim!50!white] (0,0) rectangle (4,3);
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\coordinate (p3) at (1,2);
\draw[tAccent] (p1) -- (p2);
\node[fill=yellow, draw=blue] (n1) at (2,1.5) {};
\node[fill=white, draw, text=magenta, font=\tiny] (n2) at (p2) {n2};
\node[fill=green, inner sep=0.3cm] (n3) at (p3) {};
\end{tikzpicture}
```

\node C'est la commande pour définir un nœud, une boîte contenant du texte (plus généralement du code \LaTeX). Un nœud vide a une taille, car il y a du padding. Piloté par l'option `inner sep` (ou la paire `inner xsep` et `inner ysep`)

Commandes de base

Définir des styles

Les options principales :

draw Cette option permet de contrôler la couleur des lignes/contours/bordures.

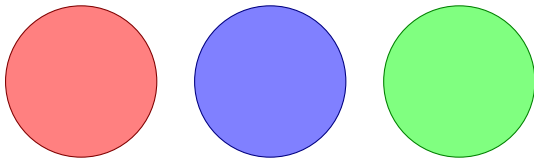
fill Cette option permet de contrôler la couleur des surfaces (= de l'intérieur).

text Cette option permet de contrôler la couleur du texte.

font Cette option permet de contrôler la police de texte.

Commandes de base

Définir des styles



Définir des styles combinées

```
\tikzset{paint/.style={draw=#1!50!black, fill=#1!50!white}}  
\begin{tikzpicture}  
\draw[paint=red] (0,0) circle (1cm);  
\draw[paint=blue] (2.5,0) circle (1cm);  
\draw[paint=green] (5,0) circle (1cm);  
\end{tikzpicture}
```

Cadres et dimensions de l'image

Cadres et dimensions de l'image

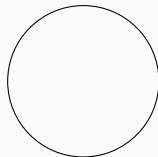
Calcul des dimensions de l'image

Dessiner un cercle de centre $(-1,2)$ et de rayon 1cm

```
\begin{tikzpicture}  
\draw (-1,2) circle (1cm);  
\end{tikzpicture}
```

ou

```
\tikz \draw(-1,2) circle (1cm);
```



Cadres et dimensions de l'image

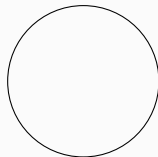
Calcul des dimensions de l'image

Dessiner un cercle de centre (1,2) et de rayon 1cm

```
\begin{tikzpicture}  
  \draw (1,2) circle (1cm);  
\end{tikzpicture}
```

ou

```
\tikz \draw(1,2) circle (1cm);
```



Cadres et dimensions de l'image

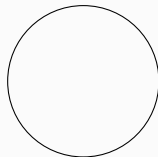
Calcul des dimensions de l'image

Dessiner un cercle de centre (1,2) et de rayon 1cm

```
\begin{tikzpicture}  
\draw (1,2) circle (1cm);  
\end{tikzpicture}
```

ou

```
\tikz \draw(1,2) circle (1cm);
```



Pourquoi le disque n'a pas bougé alors qu'on a changé les coordonnées de son centre?

Cadres et dimensions de l'image

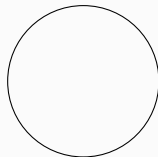
Calcul des dimensions de l'image

Dessiner un cercle de centre (1,2) et de rayon 1cm

```
\begin{tikzpicture}  
\draw (1,2) circle (1cm);  
\end{tikzpicture}
```

ou

```
\tikz \draw(1,2) circle (1cm);
```



Pourquoi le disque n'a pas bougé alors qu'on a changé les coordonnées de son centre?

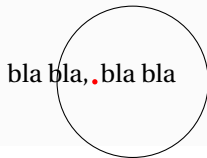
La bounding box d'une figure TikZ est adaptée au contenu dessiné
et l'image est *automatiquement recadrée*.

Cadres et dimensions de l'image

Référentiel absolu

Dessiner un cercle de centre (0.5,0) et de rayon 1cm

```
bla bla ,  
\begin{tikzpicture}[overlay]  
\draw (0.5,0) circle (1cm);  
\end{tikzpicture}  
bla bla
```



overlay Activation globale du référentiel absolu. L'origine des coordonnées de la figure est l'endroit où la figure est censée être placée (visualisé par le point rouge) et la place que la figure occupe est ignorée vis-à-vis de la suite du contenu à traiter.

Cadres et dimensions de l'image

Rogner une figure



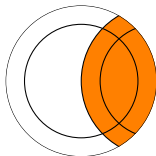
Colorier l'intersection de 2 couronnes définies par 2 cercles

```
\begin{tikzpicture}[scale=0.5]  
  
\clip (-1,0) circle (1.5cm) (-1,0) circle (2cm);  
\fill[fill=orange] (1,0) circle (1.5cm) (1,0) circle (2cm);  
  
\end{tikzpicture}
```

Masque défini à partir des cercles de gauche

Cadres et dimensions de l'image

Rogner une figure



Colorier l'intersection de 2 couronnes définies par 2 cercles

```
\begin{tikzpicture}[scale=0.5]

\clip (-1,0) circle (1.5cm) (-1,0) circle (2cm);
\fill[fill=orange] (1,0) circle (1.5cm) (1,0) circle (2cm);

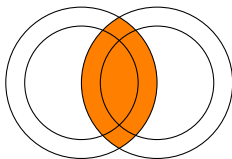
\draw (1,0) circle (1.5cm) (1,0) circle (2cm);
\end{tikzpicture}
```

Le masque s'applique globalement

La bounding box de l'image tient compte de l'intégralité du masque

Cadres et dimensions de l'image

Utilisation des cadres



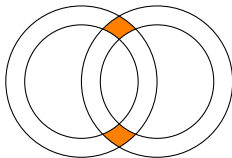
Colorier l'intersection de 2 couronnes définies par 2 cercles

```
\begin{tikzpicture}[scale=0.5]
\begin{scope}
\clip (-1,0) circle (1.5cm) (-1,0) circle (2cm);
\fill[fill=orange] (1,0) circle (1.5cm) (1,0) circle (2cm);
\end{scope}
\draw (-1,0) circle (1.5cm) (-1,0) circle (2cm);
\draw (1,0) circle (1.5cm) (1,0) circle (2cm);
\end{tikzpicture}
```

L'environnement *scope* délimite
une zone de portée (=cadre)

Cadres et dimensions de l'image

Règles de masquages



Colorier l'intersection de 2 couronnes définies par 2 cercles

```
\begin{tikzpicture}[scale=0.5]
\begin{scope}[even odd rule]
\clip (-1,0) circle (2cm) (-1,0) circle (2cm);
\fill[fill=orange] (1,0) circle (2cm) (1,0) circle (2cm);
\end{scope}
\draw (-1,0) circle (2cm) (-1,0) circle (2cm);
\draw (1,0) circle (2cm) (1,0) circle (2cm);
\end{tikzpicture}
```

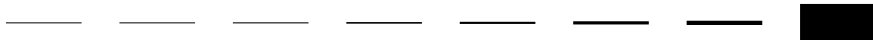
←
Modification de la règle
de superposition

Avec cette option, du point de vue du masquage, les superpositions se compensent 2 à 2.

Décorations

Décorations

Épaisseur de traits

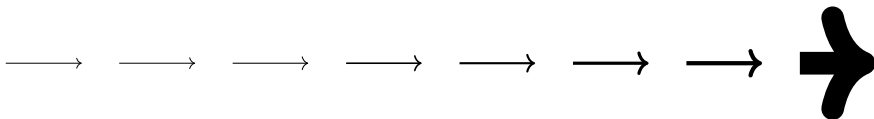


Les épaisseurs de traits

```
\begin{tikzpicture}
\draw[ultra thin] (0,0) -- (1,0);      % line width=0.1pt
\draw[very thin] (1.5,0) -- (2.5,0);  % line width=0.2pt
\draw[thin] (3,0) -- (4,0);            % line width=0.4pt
\draw[semithick] (4.5,0) -- (5.5,0);   % line width=0.6pt
\draw[thick] (6,0) -- (7,0);           % line width=0.8pt
\draw[very thick] (7.5,0) -- (8.5,0);  % line width=1.2pt
\draw[ultra thick] (9,0) -- (10,0);    % line width=1.6pt
\draw[line width=0.5cm] (10.5,0) -- (11.5,0);
\end{tikzpicture}
```


Décorations

Flèches



Flèches et épaisseur de traits

```
\begin{tikzpicture}  
  \draw[ultra thin, ->] (0,0) -- (1,0);  
  \draw[very thin, ->] (1.5,0) -- (2.5,0);  
  \draw[thin, ->] (3,0) -- (4,0);  
  \draw[semithick, ->] (4.5,0) -- (5.5,0);  
  \draw[thick, ->] (6,0) -- (7,0);  
  \draw[very thick, ->] (7.5,0) -- (8.5,0);  
  \draw[ultra thick, ->] (9,0) -- (10,0);  
  \draw[line width=0.3cm, ->] (10.5,0) -- (11.5,0);  
\end{tikzpicture}
```

Décorations

Flèches



Quelques pointes de flèches (bibliothèque *arrows.meta*)

```
\begin{tikzpicture}
\draw[thick, -Stealth] (0,0) -- (1,0);
\draw[thick, <-Latex] (1.5,0) -- (2.5,0);
\draw[thick, -Bar] (3,0) -- (4,0);
\draw[thick, double, -Implies] (4.5,0) -- (5.5,0);
\draw[thick, -Diamond] (6,0) -- (7,0);
\draw[thick, -Triangle] (7.5,0) -- (8.5,0);
\draw[thick, {Triangle[open] Triangle[open]}-{Triangle[open].Triangle[open]}} (9,0) -- (10,0);
\draw[line width=0.3cm, -{Triangle Cap[]}] (10.5,0) -- (11.5,0);
\end{tikzpicture}
```

Décorations

Pointillés et tirets

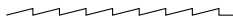
.....

Les pointillés et les tirets

```
\begin{tikzpicture}
\draw[thick, loosely dotted] (0,0) -- (1.5,0);
\draw[thick, dotted] (2,0) -- (3.5,0);
\draw[thick, densely dotted] (4,0) -- (5.5,0);
\draw[thick, loosely dashed] (6,0) -- (7.5,0);
\draw[thick, dashed] (8,0) -- (9.5,0);
\draw[thick, densely dashed] (10,0) -- (11.5,0);
\end{tikzpicture}
```

Décorations

Chemins déformés (bibliothèques *decorations*, *decorations.pathmorphing*)



Les chemins déformés

```
\begin{tikzpicture}  
  \draw[decorate, decoration=saw] (0,0) -- (3,0);  
  \draw[decorate, decoration=zigzag] (4,0) -- (7,0);  
  \draw[decorate, decoration=random steps, segment length=0.2cm] (8,0) -- (11,0);  
\end{tikzpicture}
```

Décorations

Chemins décorés (bibliothèques *decorations*, *decorations.pathreplacing*)



Les chemins décorés

```
\begin{tikzpicture}
\draw[decorate, decoration=brace] (0,0) -- (3,0);
\draw[decorate, decoration=ticks] (4,0) -- (7,0);
\draw[decorate, decoration=ticks, segment length=0.1cm] (8,0) -- (11,0);
\draw (8,0) -- (11,0);
\end{tikzpicture}
```

Décorations

Les marques décoratives (bibliothèques *decorations*, *decorations.shapes*)



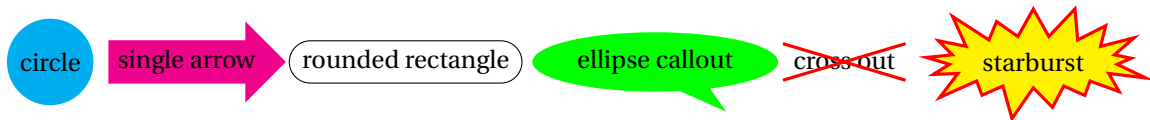
Les marques décoratives

```
\tikzset{paint/.style={ draw=#1!50!black, fill=#1!50}, decorate
  with/.style={decorate,decoration={shape backgrounds,shape=#1,shape size=2mm}}}}
\begin{tikzpicture}
\draw[decorate with=dart, paint=red] (0,0) -- (3,0);
\draw[decorate with=diamond, paint=green] (3,0) -- (6,0);
\draw[decorate with=rectangle, paint=blue] (6,0) -- (9,0);
\draw[decorate with=circle, paint=yellow] (9,0) -- (12,0);
\end{tikzpicture}
```

Diagrammes et arbres

Diagrammes et arbres

Formes des nœuds (bibliothèque *shapes*)

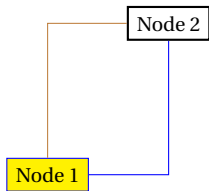


Formes de nœuds

```
\begin{tikzpicture}
\node[circle, fill=cyan] {circle};
\node[single arrow, fill=magenta] at (1.8,0) {single arrow};
\node[rounded rectangle, draw] at (4.7,0) {rounded rectangle};
\node[ellipse callout, callout relative pointer={{325:0.5cm}}, fill=green] at (8,0) {ellipse
  callout};
\node[cross out, draw=red, very thick] at (10.5,0) {cross out};
\node[starburst, draw=red, fill=yellow, very thick] at (13,0) {starburst};
\end{tikzpicture}
```


Diagrammes et arbres

Relier des nœuds



Définition de chemins

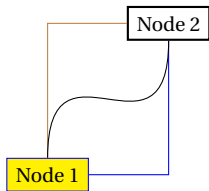
```
\begin{tikzpicture}
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,3);
\node[fill=yellow, draw=blue] (n1) at (p1) {Node 1};
\node[fill=white, draw=black, thick] (n2) at (p2) {Node 2};
\draw[blue] (n1) -| (n2);
\draw[brown] (n1) |- (n2)

\end{tikzpicture}
```

- On peut utiliser la notion d'orthogonalité pour tracer des chemins

Diagrammes et arbres

Relier des nœuds



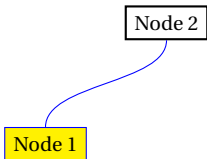
Définition de chemins

```
\begin{tikzpicture}
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,3);
\node[fill=yellow, draw=blue] (n1) at (p1) {Node 1};
\node[fill=white, draw=black, thick] (n2) at (p2) {Node 2};
\draw[blue] (n1) -- (n2);
\draw[brown] (n1) |- (n2);
\draw (n1) .. controls (n1 |- n2) and (n1 -- n2) .. (n2);
\end{tikzpicture}
```

- ▶ On peut utiliser la notion d'orthogonalité pour tracer des chemins
- ▶ On peut définir un chemin courbe à l'aide de courbes de Bézier prenant 2 points de contrôle.

Diagrammes et arbres

Relier des nœuds



Ancres et tangentes aux extrémités

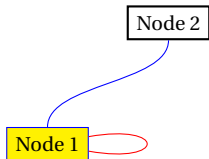
```
\begin{tikzpicture}
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\node[fill=yellow, draw=blue] (n1) at (p1) {Node 1};
\node[fill=white, draw=black, thick] (n2) at (p2) {Node 2};
\draw[blue] (n1) .. controls +(90:1cm) and +(-90:1cm) .. (n2);

\end{tikzpicture}
```

- On peut définir une courbe à partir de ses tangentes aux extrémités.

Diagrammes et arbres

Relier des nœuds



Ancres et tangentes aux extrémités

```
\begin{tikzpicture}
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\node[fill=yellow, draw=blue] (n1) at (p1) {Node 1};
\node[fill=white, draw=black, thick] (n2) at (p2) {Node 2};
\draw[blue] (n1) .. controls +(90:1cm) and +(-90:1cm) .. (n2);
\draw[red] (n1) .. controls +(-10:2cm) and +(10:2cm) .. (n1);

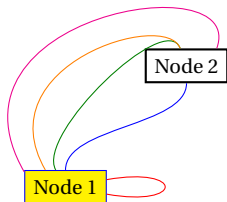
\end{tikzpicture}
```

- On peut définir une courbe qui revient sur le même nœud, ce qui est aussi valable pour les courbes de Bézier avec points de contrôle en coordonnées cartésiennes.

Diagrammes et arbres

Relier des nœuds

Ancres et tangentes aux extrémités



```
\begin{tikzpicture}
\coordinate (p1) at (1,0.5);
\coordinate (p2) at (3,2.5);
\node[fill=yellow, draw=blue] (n1) at (p1) {Node 1};
\node[fill=white, draw=black, thick] (n2) at (p2) {Node 2};
\draw[blue] (n1) .. controls +(90:1cm) and +(-90:1cm) .. (n2);
\draw[red] (n1) .. controls +(-10:2cm) and +(10:2cm) .. (n1);
\draw[green!50!black] (n1) .. controls +(120:1cm) and +(110:1cm) .. (n2);
\draw[orange] (n1.140) .. controls +(120:2cm) and +(110:1cm) .. (n2);
\draw[magenta] (n1.160) .. controls +(120:3cm) and +(70:1.5cm) .. (n2.30);
\end{tikzpicture}
```

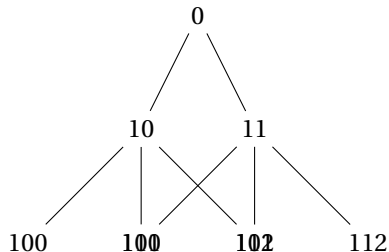
- On peut préciser le point d'ancrage sur le bord d'un nœud. On donne la valeur de l'angle portant le rayon partant du centre du nœud. Le point considéré est l'intersection de ce rayon avec le bord.

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}  
\node {0} child { node {10} child { node {100} } child { node {101} } child { node {102} } }  
          child { node {11} child { node {110} } child { node {111} } child { node {112} } } ;  
\end{tikzpicture}
```



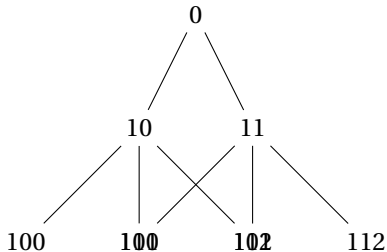
- La construction d'arbres en TikZ se fait par l'utilisation des commandes `\node`, *node* et *child*.

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}  
\node {0} child { node {10} child { node {100} } child { node {101} } child { node {102} } }  
          child { node {11} child { node {110} } child { node {111} } child { node {112} } } ;  
\end{tikzpicture}
```



- La construction d'arbres en TikZ se fait par l'utilisation des commandes `\node`, *node* et *child*.

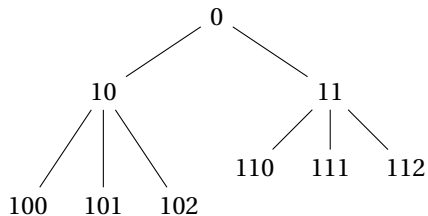
Il n'y a de; qu'à la fin de la définition de l'arbre (fin de la commande `\node`)

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}  
\node {0} [sibling distance=3cm, level distance=1cm] child { node {10} [sibling distance=1cm,  
level distance=1.5cm] child { node {100} } child { node {101} } child { node {102} } }  
child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {  
node {112} } } };  
\end{tikzpicture}
```



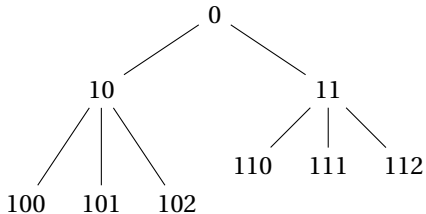
- On peut contrôler l'écartement entre les nœuds d'un même niveau avec l'option `sibling distance`

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}  
\node {0} [sibling distance=3cm, level distance=1cm] child { node {10} [sibling distance=1cm,  
    level distance=1.5cm] child { node {100} } child { node {101} } child { node {102} } }  
    child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {  
        node {112} } } };  
\end{tikzpicture}
```



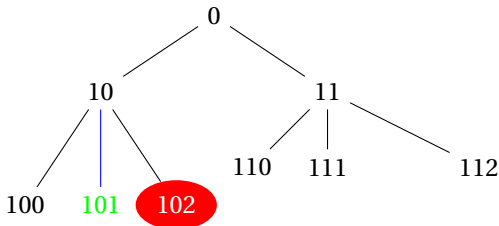
- ▶ On peut contrôler l'écartement entre les nœuds d'un même niveau avec l'option `sibling distance`
- ▶ On peut contrôler la profondeur entre 2 niveaux de l'arbre avec l'option `level distance`

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}
\node {0} [sibling distance=3cm, level distance=1cm] child { node {10} [sibling distance=1cm,
    level distance=1.5cm] child { node {100} } child[blue] { node[green] {101} } child {
    node[ellipse, fill=red, text=white] {102} } }
    child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {
        node[xshift=1cm] {112} } } };
\end{tikzpicture}
```



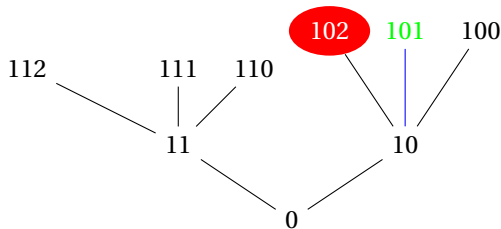
- ▶ Les nœuds d'un arbre se comportent comme des nœuds standards.
- ▶ Styliser la commande *child* permet de styliser les branches de l'arbre

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}
\node {0} [sibling distance=3cm, level distance=1cm, grow=north] child { node {10} [sibling
distance=1cm, level distance=1.5cm] child { node {100} } child[blue] { node[green] {101} }
child { node[ellipse, fill=red, text=white] {102} } }
child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {
node[xshift=-1cm] {112} } };
\end{tikzpicture}
```



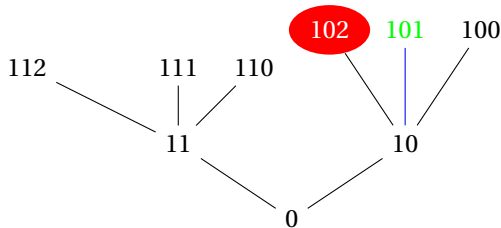
- On peut personnaliser la direction de croissance de l'arbre avec l'option `grow`

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}
\node {0} [sibling distance=3cm, level distance=1cm, grow=north] child { node {10} [sibling
    distance=1cm, level distance=1.5cm] child { node {100} } child[blue] { node[green] {101} }
    child { node[ellipse, fill=red, text=white] {102} } }
child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {
    node[xshift=-1cm] {112} } };
\end{tikzpicture}
```



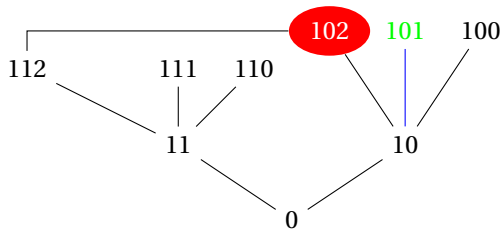
- On peut personnaliser la direction de croissance de l'arbre avec l'option `grow`

Diagrammes et arbres

Créer des arbres

Un arbre à 3 niveaux

```
\begin{tikzpicture}
\node {0} [sibling distance=3cm, level distance=1cm, grow=north] child { node {10} [sibling
    distance=1cm, level distance=1.5cm] child { node {100} } child[blue] { node[green] {101} }
    child { node[ellipse, fill=red, text=white, name=n102] {102} } }
child { node {11} [sibling distance=1cm] child { node {110} } child { node {111} } child {
    node[xshift=-1cm, name=n112] {112} } };
\draw (n102) -- (n112);
\end{tikzpicture}
```

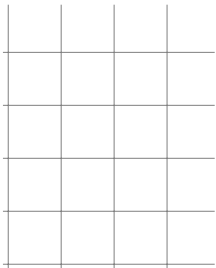


- On peut relier les nœuds a posteriori, il suffit de les nommer.

Tracés de courbes

Tracés de courbes

Directement avec TikZ



Tracés de fonctions analytiques

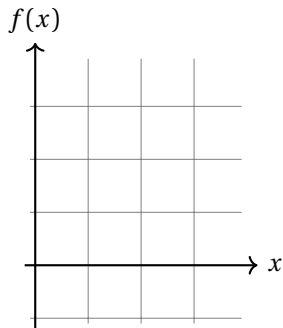
```
\begin{tikzpicture}[domain=0:4, scale=0.7]  
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
```

```
\end{tikzpicture}
```

grid permet de tracer une grille (option *step* pour la finesse)

Tracés de courbes

Directement avec TikZ



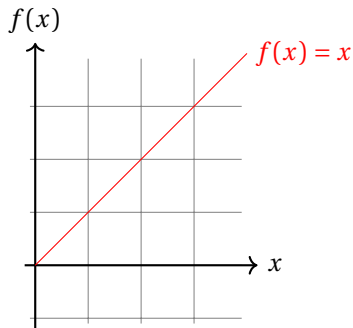
Tracés de fonctions analytiques

```
\begin{tikzpicture}[domain=0:4, scale=0.7]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->, thick] (-0.2,0) -- (4.2,0) node[right] {\(x\)};
\draw[->, thick] (0,-1.2) -- (0,4.2) node[above] {\(f(x)\)};
```

```
\end{tikzpicture}
```


Tracés de courbes

Directement avec TikZ



Tracés de fonctions analytiques

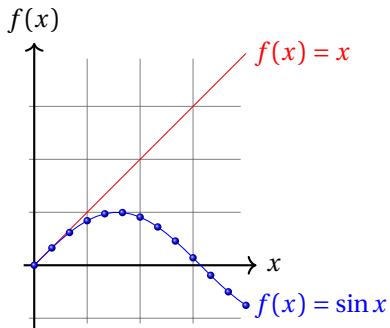
```
\begin{tikzpicture}[domain=0:4, scale=0.7]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->, thick] (-0.2,0) -- (4.2,0) node[right] {\(x\)};
\draw[->, thick] (0,-1.2) -- (0,4.2) node[above] {\(f(x)\)};
\draw[color=red] plot (\x,\x) node[right] {\(f(x)=x\)};

\end{tikzpicture}
```

plot permet de tracer une fonction analytique. L'intervalle de définition est défini avec l'option **domain**

Tracés de courbes

Directement avec TikZ



Tracés de fonctions analytiques

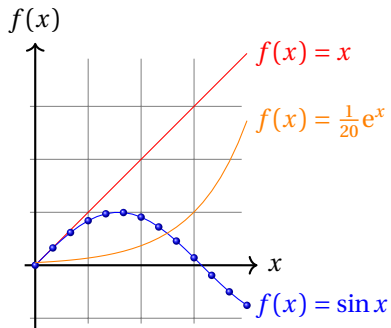
```
\begin{tikzpicture}[domain=0:4, scale=0.7]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->, thick] (-0.2,0) -- (4.2,0) node[right] {\(x\)};
\draw[->, thick] (0,-1.2) -- (0,4.2) node[above] {\(f(x)\)};
\draw[color=red] plot (\x,\x) node[right] {\(f(x) = x\)};
\draw[color=blue, mark=ball, mark repeat=2] plot (\x,{sin(\x
r)}) node[right] {\(f(x) = \sin x\)};

\end{tikzpicture}
```

mark permet d'ajouter une marque (voir la librairie *plotmarks*)

Tracés de courbes

Directement avec TikZ



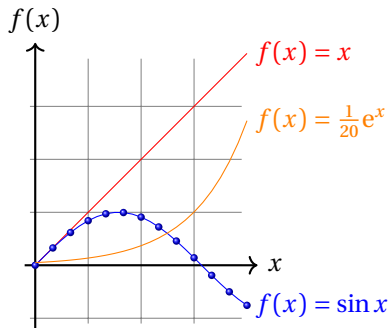
Tracés de fonctions analytiques

```
\begin{tikzpicture}[domain=0:4, scale=0.7]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->, thick] (-0.2,0) -- (4.2,0) node[right] {\(x\)};
\draw[->, thick] (0,-1.2) -- (0,4.2) node[above] {\(f(x)\)};
\draw[color=red] plot (\x,\x) node[right] {\(f(x) = x\)};
\draw[color=blue, mark=ball, mark repeat=2] plot (\x,{sin(\x
r)}) node[right] {\(f(x) = \sin x\)};
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right]
{\(f(x) = \frac{1}{20} \mathrm{e}^x\)};
\end{tikzpicture}
```

plot accepte aussi des fichiers de données.

Tracés de courbes

Directement avec TikZ



Tracés de fonctions analytiques

```
\begin{tikzpicture}[domain=0:4, scale=0.7]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->, thick] (-0.2,0) -- (4.2,0) node[right] {\(x\)};
\draw[->, thick] (0,-1.2) -- (0,4.2) node[above] {\(f(x)\)};
\draw[color=red] plot (\x,\x) node[right] {\(f(x) = x\)};
\draw[color=blue, mark=ball, mark repeat=2] plot (\x,{sin(\x
r)}) node[right] {\(f(x) = \sin x\)};
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right]
{\(f(x) = \frac{1}{20} \mathrm{e}^x\)};
\end{tikzpicture}
```



Voir aussi la librairie *datavisualization*.

Tracés de courbes

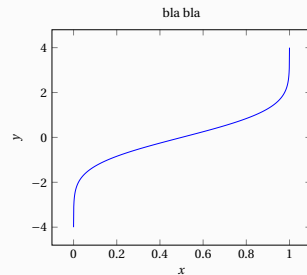
En utilisant PGFPLOTS



Echelle linéaire : Fichier invcum.dat issu de la documentation de PGFPLOTS

```
\begin{tikzpicture}
\begin{axis}[title=bla bla, xlabel={\(\x\)}, ylabel={\(\y\)}]
\addplot [blue] table {samples/invcum.dat};
\end{axis}
\end{tikzpicture}
```

x_0	$f(x)$
# data	
3.16693000e-05	-4.00001451e+00
1.00816962e-03	-3.08781504e+00
1.98466995e-03	-2.88058811e+00
2.96117027e-03	-2.75205040e+00
3.93767059e-03	-2.65736805e+00
4.91417091e-03	-2.58181091e+00



Tracés de courbes

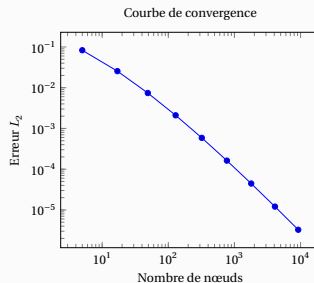
En utilisant PGFPLOTS



Echelle logarithmique : Fichier data_d2.dat issu de la documentation de PGFPLOTS

```
\begin{tikzpicture}
\begin{loglogaxis}[title=Courbe de convergence,
    xlabel={Nombre de nœuds}, ylabel={Erreur  $L_2$ }]
\addplot table {samples/data_d2.dat};
\end{loglogaxis}
\end{tikzpicture}
```

dof	l2_err	level
5	8.312e-02	2
17	2.547e-02	3
49	7.407e-03	4
129	2.102e-03	5
321	5.874e-04	6
769	1.623e-04	7

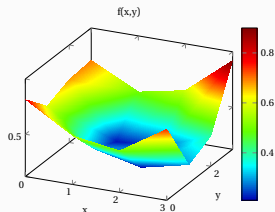


Tracés de courbes

En utilisant PGFPLOTS

Surfaces : Fichier first3d.dat issu de la documentation de PGFPLOTS

```
\begin{tikzpicture}
\begin{axis}[title={f(x,y)}, xlabel={x}, ylabel={y}, colorbar,
            colormap/bluered]
\addplot3[surf, shader=interp] table {samples/first3d.dat};
\end{axis}
\end{tikzpicture}
```



0	0	0.8
1	0	0.56
2	0	0.5
3	0	0.75

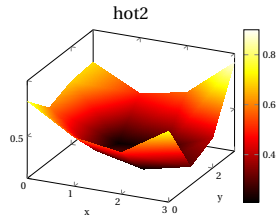
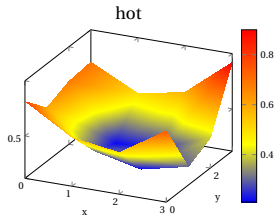
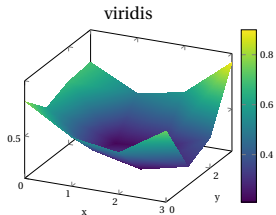
0	1	0.6
1	1	0.3
2	1	0.21
3	1	0.3

0	2	0.68
1	2	0.22
2	2	0.25
3	2	0.4

Tracés de courbes

En utilisant PGFPLOTS

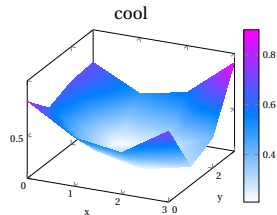
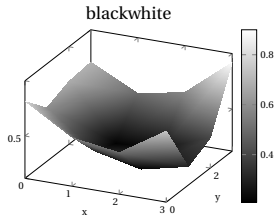
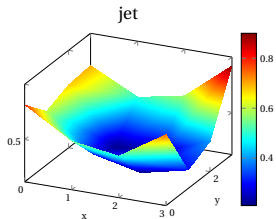
Les colormaps :



Tracés de courbes

En utilisant PGFPLOTS

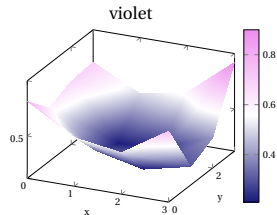
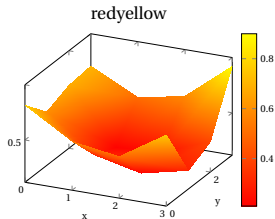
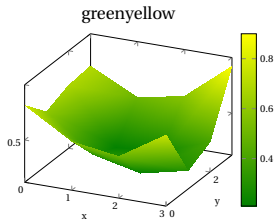
Les colormaps :



Tracés de courbes

En utilisant PGFPLOTS

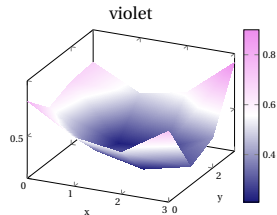
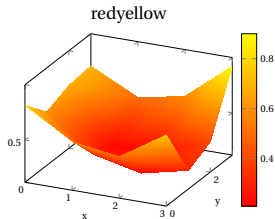
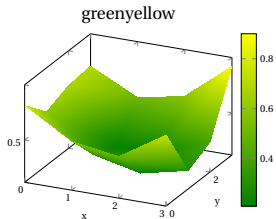
Les colormaps :



Tracés de courbes

En utilisant PGFplots

Les colormaps :



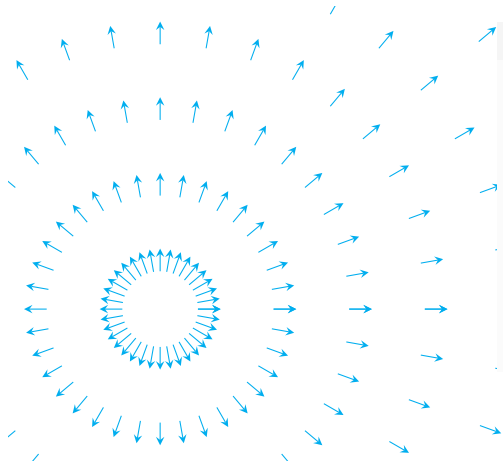
Voir le séminaire InfoMaths de Denis Bitouzé en juin 2023 :

<https://infomath.pages.math.cnrs.fr/talk/2022-2023/pgfplots/>

Autour des coordonnées

Autour des coordonnées

Coordonnées polaires



Champ radial de vecteurs avec \foreach

```
\begin{tikzpicture}
\clip (-2,-2) rectangle +(14,6);
\begin{scope}
\clip (0,0) circle (16);
\foreach \r in {0.5,1.5,2.5,...,15.5} {
  \foreach \theta in {0,10,20,...,360} {
    \draw[cyan, -stealth] (\theta:\r) --
      +(\theta:0.3);
  }
}
\end{scope}
\end{tikzpicture}
```

Autour des coordonnées

Coordonnées polaires

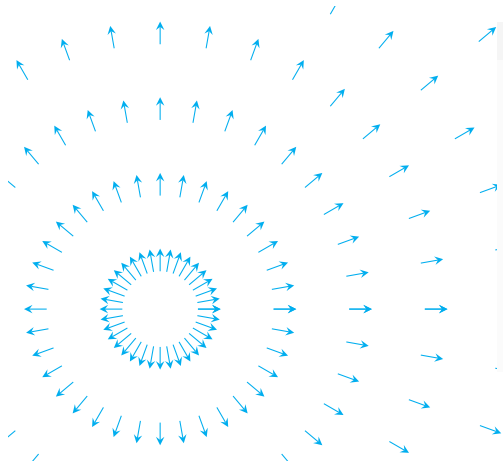
`\foreach` permet de définir un ensemble d'éléments en faisant varier un ou plusieurs paramètres.

Champ radial de vecteurs avec `\foreach`

```
\begin{tikzpicture}
\clip (-2,-2) rectangle +(14,6);
\begin{scope}
\clip (0,0) circle (16);
\foreach \r in {0.5,1.5,2.5,...,15.5} {
  \foreach \theta in {0,10,20,...,360} {
    \draw[cyan, -stealth] (\theta:\r) --
      +(\theta:0.3);
  }
}
\end{scope}
\end{tikzpicture}
```

Autour des coordonnées

Coordonnées polaires



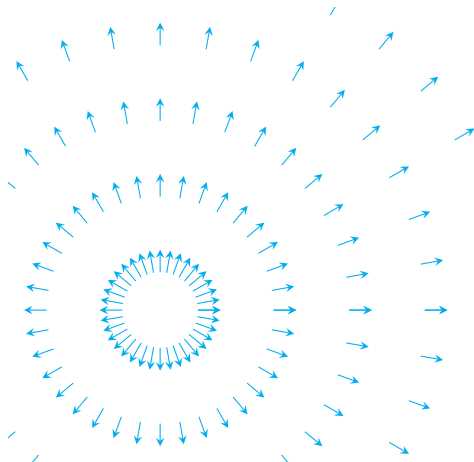
Champ radial de vecteurs avec \foreach

```
\begin{tikzpicture}
\clip (-2,-2) rectangle +(14,6);
\begin{scope}
\clip (0,0) circle (16);
\foreach \r in {0.5,1.5,2.5,...,15.5} {
  \foreach \theta in {0,10,20,...,360} {
    \draw[cyan, -stealth] (\theta:\r) --
      +(\theta:0.3);
  }
}
\end{scope}
\end{tikzpicture}
```

Utilisation des coordonnées polaires

Autour des coordonnées

Coordonnées polaires



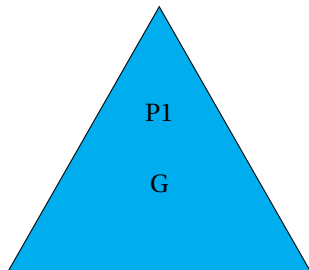
Champ radial de vecteurs avec \foreach

```
\begin{tikzpicture}
\clip (-2,-2) rectangle +(14,6);
\begin{scope}
\clip (0,0) circle (16);
\foreach \r in {0.5,1.5,2.5,...,15.5} {
  \foreach \theta in {0,10,20,...,360} {
    \draw[cyan, -stealth] (\theta:\r) --
      +(\theta:0.3);
  }
}
\end{scope}
\end{tikzpicture}
```

Utilisation des coordonnées relatives

Autour des coordonnées

Coordonnées barycentriques



Coordonnées barycentriques

```
\begin{tikzpicture}
\coordinate (s1) at (0,0);
\coordinate (s2) at (2,3.5);
\coordinate (s3) at (4,0);
\coordinate (g) at (barycentric cs:s1=1,s2=1,s3=1);
\coordinate (p1) at (barycentric cs:s1=1,s2=3,s3=1);

\draw[black, fill=cyan] (s1) -- (s2) -- (s3) -- cycle;
\node at (g) {G};
\node at (p1) {P1};

\end{tikzpicture}
```

Autour des coordonnées

Coordonnées barycentriques

P2

P1

G

Coordonnées barycentriques

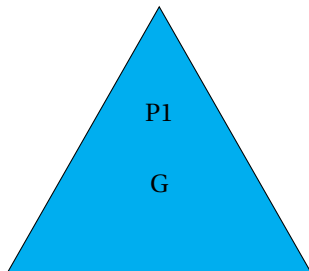
```
\begin{tikzpicture}
\coordinate (s1) at (0,0);
\coordinate (s2) at (2,3.5);
\coordinate (s3) at (4,0);
\coordinate (g) at (barycentric cs:s1=1,s2=1,s3=1);
\coordinate (p1) at (barycentric cs:s1=1,s2=3,s3=1);
\coordinate (p2) at (barycentric cs:s1=-1,s2=6,s3=-1);
\draw[black, fill=cyan] (s1) -- (s2) -- (s3) -- cycle;
\node at (g) {G};
\node at (p1) {P1};
\node at (p2) {P2};
\end{tikzpicture}
```

Autour des coordonnées

Positionnement relatif



P2



Coordonnées relatives

```
\begin{tikzpicture}
\coordinate (s1) at (0,0);
\coordinate [above right=3.5cm and 2cm of s1] (s2);
\coordinate [right=4cm of s1] (s3);
\coordinate (g) at (barycentric cs:s1=1,s2=1,s3=1);
\coordinate (p1) at (barycentric cs:s1=1,s2=3,s3=1);
\coordinate (p2) at (barycentric cs:s1=-1,s2=6,s3=-1);
\draw[black, fill=cyan] (s1) -- (s2) -- (s3) -- cycle;
\node at (g) {G};
\node at (p1) {P1};
\node at (p2) {P2};
\end{tikzpicture}
```

Autour des coordonnées

Positionnement sur une courbe

pos=0.35

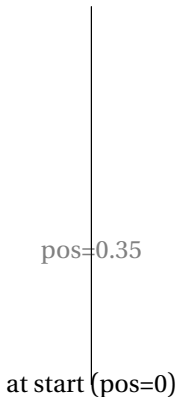
L'option pos et ses alias

```
\begin{tikzpicture}  
  \draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}  
  
  \end{tikzpicture}
```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe



L'option pos et ses alias

```
\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}

\end{tikzpicture}
```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

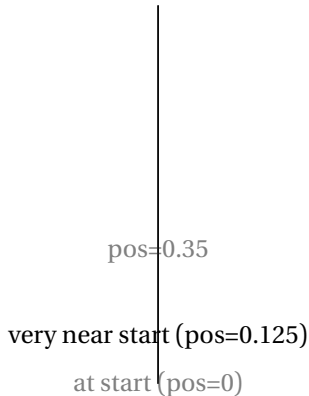
Autour des coordonnées

Positionnement sur une courbe

L'option pos et ses alias

```
\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}

\end{tikzpicture}
```



pos=0.35

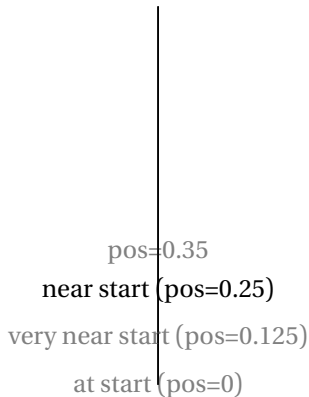
very near start (pos=0.125)

at start (pos=0)

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe



L'option pos et ses alias

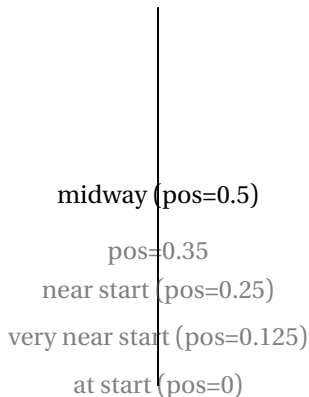
```
\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}
                        node[near start] {near start (pos=0.25)}

\end{tikzpicture}
```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe



L'option pos et ses alias

```
\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}
                        node[near start] {near start (pos=0.25)}
                        node[midway] {midway (pos=0.5)}

\end{tikzpicture}
```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe

near end (pos=0.75)
 midway (pos=0.5)
 pos=0.35
 near start (pos=0.25)
 very near start (pos=0.125)
 at start (pos=0)

L'option pos et ses alias

```

\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}
                        node[near start] {near start (pos=0.25)}
                        node[midway] {midway (pos=0.5)}
                        node[near end] {near end (pos=0.75)}

\end{tikzpicture}
  
```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe

very near end (pos=875)
 near end (pos=0.75)
 midway (pos=0.5)
 pos=0.35
 near start (pos=0.25)
 very near start (pos=0.125)
 at start (pos=0)

L'option pos et ses alias

```

\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}
                        node[near start] {near start (pos=0.25)}
                        node[midway] {midway (pos=0.5)}
                        node[near end] {near end (pos=0.75)}
                        node[very near end] {very near end (pos=875)}

\end{tikzpicture}

```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Positionnement sur une courbe

at end (pos=1)
 very near end (pos=875)
 near end (pos=0.75)
 midway (pos=0.5)
 pos=0.35
 near start (pos=0.25)
 very near start (pos=0.125)
 at start (pos=0)

L'option pos et ses alias

```

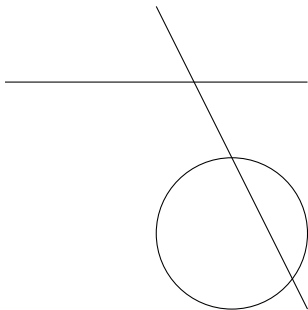
\begin{tikzpicture}
\draw (0,0) -- (0,5) node[pos=0.35] {pos=0.35}
                        node[at start] {at start (pos=0)}
                        node[very near start] {very near start (pos=0.125)}
                        node[near start] {near start (pos=0.25)}
                        node[midway] {midway (pos=0.5)}
                        node[near end] {near end (pos=0.75)}
                        node[very near end] {very near end (pos=875)}
                        node[at end] {at end (pos=1)};
\end{tikzpicture}

```

pos prend une valeur entre 0 (le point de départ de la courbe) et 1 (le point d'arrivée de la courbe)

Autour des coordonnées

Intersections



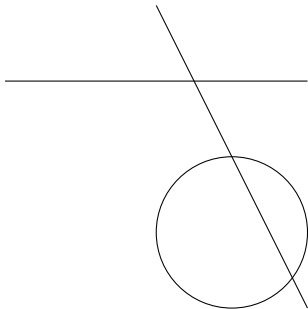
Intersection de droites et d'un cercle (bibliothèque *intersections*)

```
\begin{tikzpicture}  
  \draw[name path=Dr1] (0,3) -- (4,3);  
  \draw[name path=Dr2] (4,0) -- (2,4);  
  \draw[name path=C1] (3,1) circle (1cm);  
  
  \end{tikzpicture}
```

`name path` permet de donner un nom à un chemin

Autour des coordonnées

Intersections



Intersection de droites et d'un cercle (bibliothèque *intersections*)

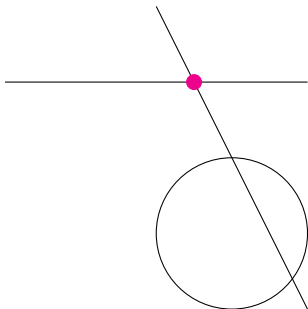
```
\begin{tikzpicture}
\draw[name path=Dr1] (0,3) -- (4,3);
\draw[name path=Dr2] (4,0) -- (2,4);
\draw[name path=C1] (3,1) circle (1cm);
\path[name intersections={of=Dr1 and Dr2, name=i}];

\end{tikzpicture}
```

\path demande les noms des courbes et le préfixe du nom des points résultants.

Autour des coordonnées

Intersections



Intersection de droites et d'un cercle (bibliothèque *intersections*)

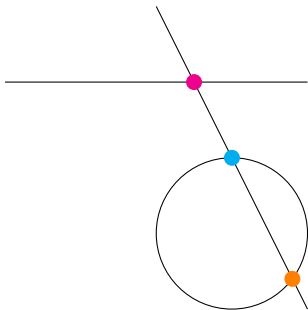
```
\begin{tikzpicture}
\draw[name path=Dr1] (0,3) -- (4,3);
\draw[name path=Dr2] (4,0) -- (2,4);
\draw[name path=C1] (3,1) circle (1cm);
\path[name intersections={of=Dr1 and Dr2, name=i}];
\fill[magenta] (i-1) circle (3pt);

\end{tikzpicture}
```

\path demande les noms des courbes et le préfixe du nom des points résultants.

Autour des coordonnées

Intersections



Intersection de droites et d'un cercle (bibliothèque *intersections*)

```
\begin{tikzpicture}
\draw[name path=Dr1] (0,3) -- (4,3);
\draw[name path=Dr2] (4,0) -- (2,4);
\draw[name path=C1] (3,1) circle (1cm);
\path[name intersections={of=Dr1 and Dr2, name=i}];
\fill[magenta] (i-1) circle (3pt);
\path[name intersections={of=C1 and Dr2, name=j}];
\fill[cyan] (j-1) circle (3pt);
\fill[orange] (j-2) circle (3pt);
\end{tikzpicture}
```

\path demande les noms des courbes et le préfixe du nom des points résultants.

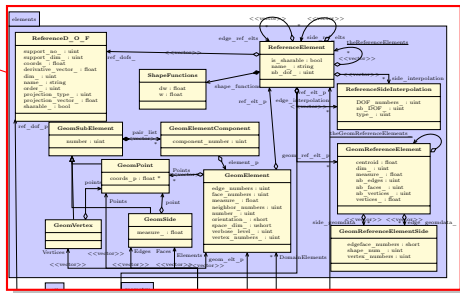
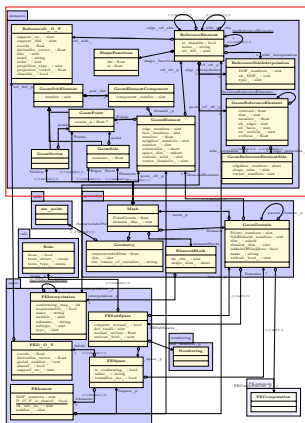
Aller plus loin

Aller plus loin

Agrandissements (bibliothèque *spy*)



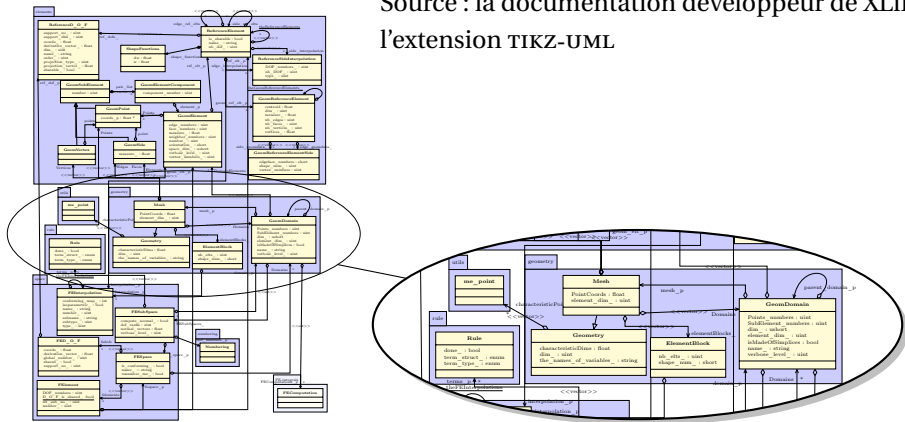
Source : la documentation développeur de XLiFE++ utilisant l'extension TIKZ-UML



Aller plus loin

Agrandissements (librairie *spy*)

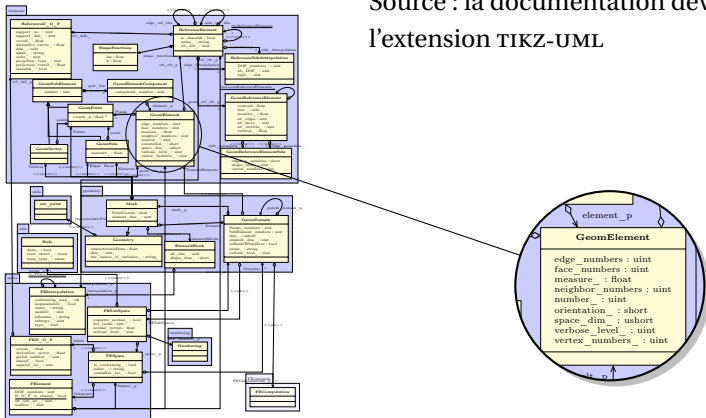
Source : la documentation développeur de XLiFE++ utilisant l'extension TIKZ-UML



Aller plus loin

Agrandissements (librairie *spy*)

Source : la documentation développeur de XLiFE++ utilisant l'extension TIKZ-UML



Aller plus loin

Agrandissements (librairie *spy*)

Utilisation de `\spy`

```
\begin{tikzpicture}[spy using outlines={rectangle , magnification=1.5, connect spies}]
\node at (0,0) {\includegraphics[width=4cm]{images/geometry_elements.pdf}};
\only<1>{\spy [red, thick, width=6cm, height=3.75cm] on (0,1.5) in node[fill=white, drop shadow]
    at (6,-0.5);}
\only<2>{\spy [ellipse, thick, width=7cm, height=2.5cm] on (0,-0.3) in node[fill=white, drop
    shadow] at (6,-1.5);}
\only<3>{\spy [circle, thick, size=2.5cm, magnification=2.5] on (0.1,1) in node[fill=white, drop
    shadow] at (6,-1);}
\end{tikzpicture}
```

- L'option globale *spy using outlines* permet de définir les paramètres par défaut : forme, grossissement, ...

Aller plus loin

Agrandissements (librairie *spy*)

Utilisation de `\spy`

```
\begin{tikzpicture}[spy using outlines={rectangle , magnification=1.5, connect spies}]
\node at (0,0) {\includegraphics[width=4cm]{images/geometry_elements.pdf}};
\only<1>{\spy [red, thick, width=6cm, height=3.75cm] on (0,1.5) in node[fill=white, drop shadow]
    at (6,-0.5);}
\only<2>{\spy [ellipse, thick, width=7cm, height=2.5cm] on (0,-0.3) in node[fill=white, drop
    shadow] at (6,-1.5);}
\only<3>{\spy [circle, thick, size=2.5cm, magnification=2.5] on (0.1,1) in node[fill=white, drop
    shadow] at (6,-1);}
\end{tikzpicture}
```

- ▶ Si on veut agrandir une image externe, il faut appeler la commande `\includegraphics` à l'intérieur d'un `\node`

Aller plus loin

Agrandissements (librairie *spy*)

Utilisation de `\spy`

```
\begin{tikzpicture}[spy using outlines={rectangle , magnification=1.5, connect spies}]
\node at (0,0) {\includegraphics[width=4cm]{images/geometry_elements.pdf}};
\only<1>{\spy [red, thick, width=6cm, height=3.75cm] on (0,1.5) in node[fill=white, drop shadow]
    at (6,-0.5);}
\only<2>{\spy [ellipse, thick, width=7cm, height=2.5cm] on (0,-0.3) in node[fill=white, drop
    shadow] at (6,-1.5);}
\only<3>{\spy [circle, thick, size=2.5cm, magnification=2.5] on (0.1,1) in node[fill=white, drop
    shadow] at (6,-1);}
\end{tikzpicture}
```

- La commande `\spy` n'accepte pas les chevrons, il faut utiliser `\only`, ...

Aller plus loin

Agrandissements (librairie *spy*)

Utilisation de `\spy`

```
\begin{tikzpicture}[spy using outlines={rectangle , magnification=1.5, connect spies}]
\node at (0,0) {\includegraphics[width=4cm]{images/geometry_elements.pdf}};
\only<1>{\spy [red, thick, width=6cm, height=3.75cm] on (0,1.5) in node[fill=white, drop shadow]
    at (6,-0.5);}
\only<2>{\spy [ellipse, thick, width=7cm, height=2.5cm] on (0,-0.3) in node[fill=white, drop
    shadow] at (6,-1.5);}
\only<3>{\spy [circle, thick, size=2.5cm, magnification=2.5] on (0.1,1) in node[fill=white, drop
    shadow] at (6,-1);}
\end{tikzpicture}
```

- Comme pour les listings de code, il faut utiliser l'option *fragile* sur l'env. *frame*

Aller plus loin

Agrandissements (librairie *spy*)

Utilisation de `\spy`

```
\begin{tikzpicture}[spy using outlines={rectangle , magnification=1.5, connect spies}]
\node at (0,0) {\includegraphics[width=4cm]{images/geometry_elements.pdf}};
\only<1>{\spy [red, thick, width=6cm, height=3.75cm] on (0,1.5) in node[fill=white, drop shadow]
    at (6,-0.5);}
\only<2>{\spy [ellipse, thick, width=7cm, height=2.5cm] on (0,-0.3) in node[fill=white, drop
    shadow] at (6,-1.5);}
\only<3>{\spy [circle, thick, size=2.5cm, magnification=2.5] on (0.1,1) in node[fill=white, drop
    shadow] at (6,-1);}
\end{tikzpicture}
```



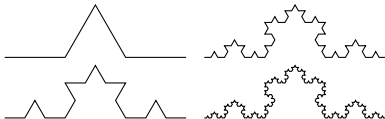
Si vous utilisez *spy* mais que vous utilisez aussi l'extension BABEL (ce que l'on est censé faire dans tout document \LaTeX), il faudra aussi charger la librairie *babel* de TIKZ.

Aller plus loin

Fractales (bibliothèques *decorations*, *decorations.fractals*)

Fractales

```
\begin{tikzpicture}[scale=0.8, decoration=Koch snowflake]  
\draw decorate{ (0,0) -- (3,0) };  
\draw decorate{ decorate{ (0,-1) -- (3,-1) } };  
\draw decorate{ decorate{ decorate{ (3.3,0) -- (6.3,0) } } };  
\draw decorate{ decorate{ decorate{ decorate{ (3.3,-1) -- (6.3,-1) } } } };  
\end{tikzpicture}
```



Aller plus loin

Matrices de nœuds (librairie *matrix*)

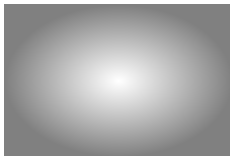
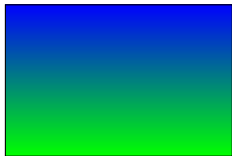
	A			2				
						12		
	B							11
					P			
	-1	6				-2		
								G
5							N	
		2				8		
				A			9	

Matrices de nœuds

```
\begin{tikzpicture}[scale=.2, font=\tiny]
\fill[white] (-10, -10) rectangle (10, 10);
\matrix (s) [matrix of nodes, nodes in empty cells, column sep=-\pgflinewidth, row
sep=-\pgflinewidth, nodes={draw, minimum size=0.4cm, anchor=center, inner sep=0pt}]
{
& A & & & 2 & & & & \\
& & & & & & 12 & & \\
& B & & & & & & & 11 \\
& & & & & P & & & \\
& -1 & 6 & & & & -2 & & \\
& & & & & & & & G \\
5 & & & & & & & N & \\
& & 2 & & & & 8 & & \\
& & & & A & & & 9 & \\
};
\draw[very thick] (s-1-1.north west) rectangle (s-9-9.south east);
\draw[very thick] (s-1-3.north east) -- (s-9-3.south east);
\draw[very thick] (s-1-6.north east) -- (s-9-6.south east);
\draw[very thick] (s-3-1.south west) -- (s-3-9.south east);
\draw[very thick] (s-6-1.south west) -- (s-6-9.south east);
\end{tikzpicture}
```

Aller plus loin

Dégradés de couleurs et textures

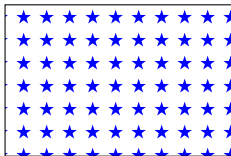
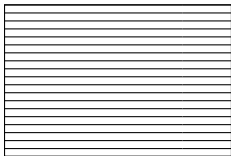


Les dégradés (bibliothèque *shadings*)

```
\begin{tikzpicture}  
  \shade[ball color=blue] (0,1) circle (2ex);  
  \draw[top color=blue, bottom color=green] (1,0) rectangle (4,2);  
  \shade[inner color=white, outer color=gray] (5,0) rectangle (8,2);  
  \shade[left color=green, right color=magenta] (9,0) rectangle (12,2);  
\end{tikzpicture}
```

Aller plus loin

Dégradés de couleurs et textures



Les textures (bibliothèque *patterns*)

```
\begin{tikzpicture}  
  \draw[pattern=horizontal lines] (0,0) rectangle (3,2);  
  \draw[pattern=fivepointed stars, pattern color=blue] (4,0) rectangle (7,2);  
\end{tikzpicture}
```