

Dessins avec Pstricks

Sumaia SAAD EDDIN

08 Octobre 2013

- 1 Qu'est-ce que PSTricks
- 2 Origine de système de coordonnées
- 3 Dimensions des objets PSTricks
- 4 Réserver de l'espace à un dessin
 - Exemple
- 5 Unité
- 6 Grilles
 - Options
 - Exemple
- 7 Les Lignes
- 8 Options
 - exemple
 - Exemple
- 9 Figues usuelles
 - Rectangles et polygones
 - Cercles, disques et ellipses
 - Secteurs et arcs
- 10 Repère
- 11 (Dé)Placer des objets

Qu'est-ce que **PSTricks** ?

- **PSTricks** est une extension de LaTeX qui permet d'utiliser la majeure partie des possibilités de **PostScript**.

Qu'est-ce que **PSTricks** ?

- **PSTricks** est une extension de LaTeX qui permet d'utiliser la majeure partie des possibilités de **PostScript**.
- **PSTricks** est un outil de préparation de dessins.

Qu'est-ce que **PSTricks** ?

- **PSTricks** est une extension de LaTeX qui permet d'utiliser la majeure partie des possibilités de **PostScript**.
- **PSTricks** est un outil de préparation de dessins.
- **PSTricks** est ensemble de commandes directement utilisables depuis **LATEX**.

Qu'est-ce que **PSTricks** ?

- **PSTricks** est une extension de LaTeX qui permet d'utiliser la majeure partie des possibilités de PostScript.
- **PSTricks** est un outil de préparation de dessins.
- **PSTricks** est ensemble de commandes directement utilisables depuis **LATEX**.
- **PSTricks** peut être chargé via

Code source :

```
\usepackage{pst - all}
```

Origine de système de coordonnées

L'origine du système de coordonnées utilisés par **PSTricks** est le point courant de **Latex**

Code source :

```
Bonjour \psline(2,1)
```

Résultat :

Bonjour



Origine de système de coordonnées

L'origine du système de coordonnées utilisés par **PSTricks** est le point courant de **Latex**

Code source :

```
Bonjour \psline(2,1)
```

Résultat :

Bonjour



Code source :

```
Bonjour \psline(2,1) \psline(1,1)
```

Résultat :

Bonjour



Dimensions des objets PSTricks

Remarque

Les objets graphiques créés par **PSTricks** n'ont pas de dimension propre, c'est-à-dire que **LATEX** ne leur réserve aucun espace.

Dimensions des objets PSTricks

Remarque

Les objets graphiques créés par **PSTricks** n'ont pas de dimension propre, c'est-à-dire que **LATEX** ne leur réserve aucun espace.

Code source :

```
Bonjour \psline(2,1) les amis !
```

Résultat :

Bonjour les amis !

Réserver de l'espace à un dessin

Pour réserver de l'espace à un dessin, on utilisera l'environnement **pspicture** :

Syntaxe

```
\begin{pspicture}(x_0,y_0)(x_1,y_1)  
\end{pspicture}
```

Réserver de l'espace à un dessin

Pour réserver de l'espace à un dessin, on utilisera l'environnement **pspicture** :

Syntaxe

```
\begin{pspicture}(x_0,y_0)(x_1,y_1)  
\end{pspicture}
```

où

- (x_0, y_0) est le point bas gauche.
- (x_1, y_1) est le point haut droit du rectangle délimitant la figure.

Réserver de l'espace à un dessin

Pour réserver de l'espace à un dessin, on utilisera l'environnement **pspicture** :

Syntaxe

```
\begin{pspicture}(x_0,y_0)(x_1,y_1)  
\end{pspicture}
```

où

- (x_0, y_0) est le point bas gauche.
- (x_1, y_1) est le point haut droit du rectangle délimitant la figure.

Remarque :

Si (x_0, y_0) est omis, il est remplacé par $(0, 0)$.

Code source :

Bonjour

```
\begin{pspicture}(0,0)(2.1,1.1)
```

```
\psline(2,1)
```

```
\end{pspicture}
```

les amis !

Résultat :

Bonjour



les amis !

Quelle est la différence entre **Pspicture** et **Pspicture***

Code source :

Bonjour

```
\begin{pspicture}(0,0)(2.1,1.1)
```

```
\psline(3,1)
```

```
\end{pspicture}
```

les amis !

Résultat :

Bonjour  les amis !

Quelle est la différence entre **Pspicture** et **Pspicture***

Code source :

Bonjour

```
\begin{pspicture*}(0,0)(2.1,1.1)
```

```
\psline(3,1)
```

```
\end{pspicture*}
```

les amis !

Résultat :

Bonjour  les amis !

Unité :

L'unité graphique de **Pstricks** est **1 cm** par défaut.

- Si on veut la changer, on écrit (par exemple) avant le `\begin{pspicture}` :

Syntaxe

```
\psset{xunit = 2cm, yunit = 0.5cm}
```

Unité :

L'unité graphique de **Pstricks** est **1 cm** par défaut.

- Si on veut la changer, on écrit (par exemple) avant le `\begin{pspicture}` :

Syntaxe

```
\psset{xunit = 2cm, yunit = 0.5cm}
```

- Si les **deux unités graphiques sont les mêmes**, on peut écrire :

Syntaxe

```
\psset{unit = 2cm}
```

Unité :

L'unité graphique de **Pstricks** est **1 cm** par défaut.

- Si on veut la changer, on écrit (par exemple) avant le `\begin{pspicture}` :

Syntaxe

```
\psset{xunit = 2cm, yunit = 0.5cm}
```

- Si les **deux unités graphiques sont les mêmes**, on peut écrire :

Syntaxe

```
\psset{unit = 2cm}
```

- Si **toutes les lignes ont pour épaisseur 2.5 pt**, on peut écrire :

Syntaxe

```
\psset{linewidth = 2.5pt}
```

Les unités utilisable sont :

- **millimètre** : mm
- **centimètre** : cm
- **pouce** : in
- **point** : pt (0,35 mm)
- **largeur d'un** *m* : em
- **hauteur d'un** *x* : ex

Grilles :

On peut dessiner des grilles.

Syntaxe

```
\psgrid(x1,y1)(x2,y2)
```

Grilles :

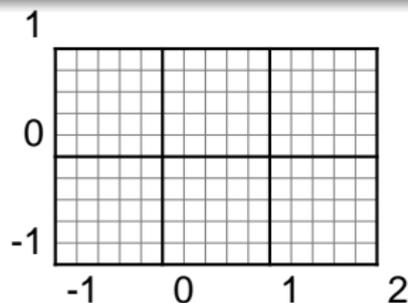
On peut dessiner des grilles.

Syntaxe

$$\backslash psgrid(x_1, y_1)(x_2, y_2)$$

Code source :

```
\begin{pspicture}(-1,-1)(2,1)
\psgrid(-1,-1)(2,1)
\end{pspicture}
```



Remarque :

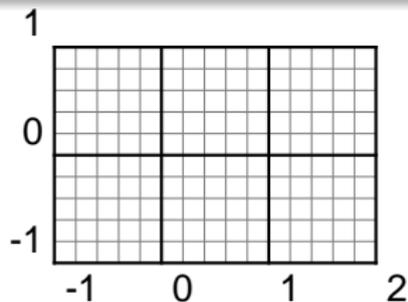
Si la commande `\psgrid` est donnée sans argument, elle occupe toute la figure.

Remarque :

Si la commande `\psgrid` est donnée sans argument, elle occupe toute la figure.

Code source :

```
\begin{pspicture}(-1,-1)(2,1)  
\psgrid  
\end{pspicture}
```



Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`

Longueur du côté du carreau unité.

Il y a plusieurs options :

- `unit=unit` par défaut : *1cm*
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : *0.8pt*
Épaisseur des lignes principales

Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : `0.8pt`
Épaisseur des lignes principales
- `gridcolor=color` par défaut : `black`
Couleur des lignes principales.

Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : `0.8pt`
Épaisseur des lignes principales
- `gridcolor=color` par défaut : `black`
Couleur des lignes principales.
- `griddots=num` par défaut : `0`
Si $num > 0$, les lignes sont en pointillés, avec num points par graduation.

Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : `0.8pt`
Épaisseur des lignes principales
- `gridcolor=color` par défaut : `black`
Couleur des lignes principales.
- `griddots=num` par défaut : `0`
Si $num > 0$, les lignes sont en pointillés, avec num points par graduation.
- `gridlabels=dim` par défaut : `10pt`
Taille des nombres utilisés pour graduer.

Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : `0.8pt`
Épaisseur des lignes principales
- `gridcolor=color` par défaut : `black`
Couleur des lignes principales.
- `griddots=num` par défaut : `0`
Si $num > 0$, les lignes sont en pointillés, avec num points par graduation.
- `gridlabels=dim` par défaut : `10pt`
Taille des nombres utilisés pour graduer.
- `gridlabelcolor=color` par défaut : `black`
Couleur des nombres utilisés pour graduer.

Il y a plusieurs options :

- `unit=unit` par défaut : `1cm`
Longueur du côté du carreau unité.
- `gridwidth=dim` par défaut : `0.8pt`
Épaisseur des lignes principales
- `gridcolor=color` par défaut : `black`
Couleur des lignes principales.
- `griddots=num` par défaut : `0`
Si $num > 0$, les lignes sont en pointillés, avec num points par graduation.
- `gridlabels=dim` par défaut : `10pt`
Taille des nombres utilisés pour graduer.
- `gridlabelcolor=color` par défaut : `black`
Couleur des nombres utilisés pour graduer.
- `subgriddiv=int` par défaut : `5`
Nombre de sous-graduations de la grille.

- `subgridwidth=dim` par défaut : `0.4pt`

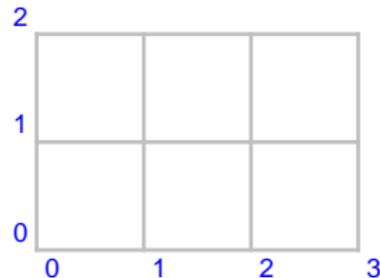
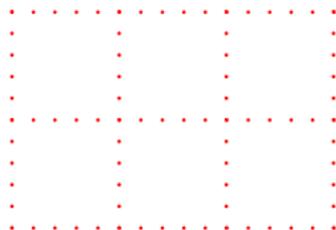
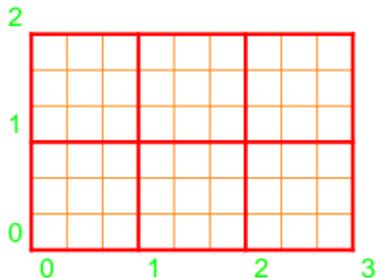
Épaisseur des lignes secondaires.

- `subgridwidth=dim` par défaut : `0.4pt`
Épaisseur des lignes secondaires.
- `subgridcolor=color` par défaut : `gray`
Couleur des lignes secondaires.

- `subgridwidth=dim` par défaut : `0.4pt`
Épaisseur des lignes secondaires.
- `subgridcolor=color` par défaut : `gray`
Couleur des lignes secondaires.
- `subgriddots=num` par défaut : `0`
Comme `griddots`, pour les sous-graduations.

Code source :

```
\psset{gridwidth = 1pt, gridcolor = red, gridlabels = 7pt, gridlabelcolor = green,  
subgriddiv= 3, subgridwidth= 0.1pt, subgridcolor= orange}  
\begin{pspicture}(3,2)  
\psgrid  
\end{pspicture}
```



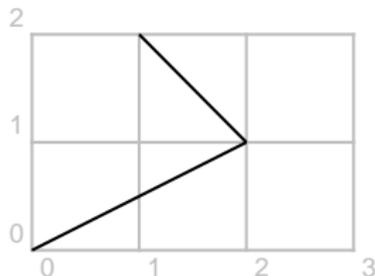
Les droites :

Syntaxe

```

\begin{pspicture}(3,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray, gridlabels =
7cm](0,0)(3,2)
\psline(2,1)
\end{pspicture}

```

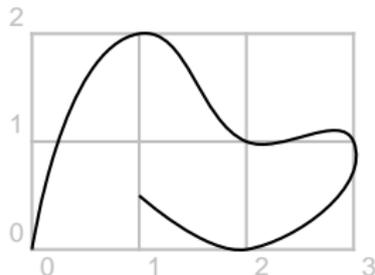


Les courbes :

Pour une courbe passant par des points donnés :

Code source :

```
\begin{pspicture}(3,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray, gridlabels =
7pt](0,0)(3,2)
\pscurve(0,0)(1,2)(2,1)(3,1)(2,0)(1,0.5)
\end{pspicture}
```

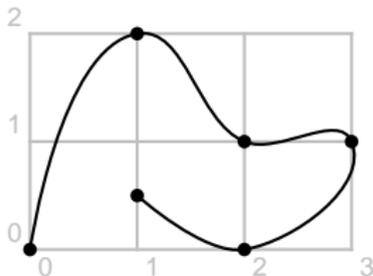


Code source :

```

\begin{pspicture}(3,2)
\psgrid[subgriddiv = 0,gridcolor = lightgray,gridlabelcolor = lightgray,gridlabels =
7pt](0,0)(3,2)
\pscurve[showpoints = true](0,0)(1,2)(2,1)(3,1)(2,0)(1,0.5)
\end{pspicture}

```



Options :

[opt] est une option qui permet de modifier les caractéristiques des objets représentés.

- **Épaisseur** : [linewidth=largeur] modifie l'épaisseur de la ligne.

Options :

[opt] est une option qui permet de modifier les caractéristiques des objets représentés.

- **Épaisseur** : [linewidth=largeur] modifie l'épaisseur de la ligne.
- **Style** : pointillés, ...
[linestyle=dashed] trace la ligne en traitillés et
[linestyle=dotted], en pointillés. On peut combiner les deux.
- **Couleur** : [linecolor=couleur] pour colorier l'objet avec une couleur.
[fillstyle=solid] permet de remplir l'objet par la couleur sollicitée au moyen de la commande
[fillcolor=couleur].

Options :

[opt] est une option qui permet de modifier les caractéristiques des objets représentés.

- **Épaisseur** : [linewidth=largeur] modifie l'épaisseur de la ligne.
- **Style** : pointillés, ...
 [linestyle=dashed] trace la ligne en traitillés et
 [linestyle=dotted], en pointillés. On peut combiner les deux.
- **Couleur** : [linecolor=couleur] pour colorier l'objet avec une couleur.
 [fillstyle=solid] permet de remplir l'objet par la couleur sollicitée au moyen de la commande
 [fillcolor=couleur].
- **Points** : On peut changer le style d'affichage des points (par défaut, un disque noir).

Les 10 styles sont :

(square, square*, *, triangle, triangle*, diamond, diamond*, pentagon, pentagon*).

Options :

[opt] est une option qui permet de modifier les caractéristiques des objets représentés.

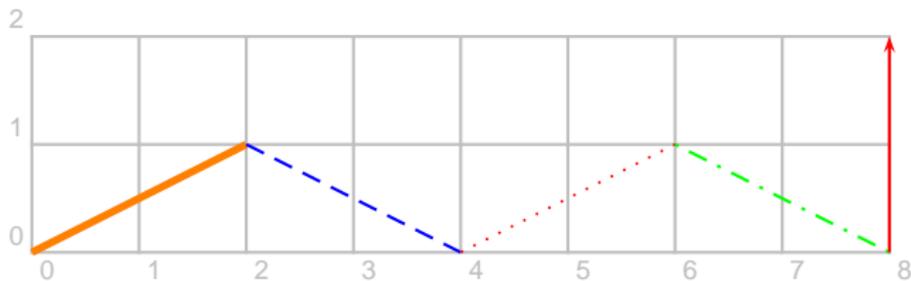
- **Épaisseur** : [linewidth=largeur] modifie l'épaisseur de la ligne.
- **Style** : pointillés, ...
 [linestyle=dashed] trace la ligne en traitillés et
 [linestyle=dotted], en pointillés. On peut combiner les deux.
- **Couleur** : [linecolor=couleur] pour colorier l'objet avec une couleur.
 [fillstyle=solid] permet de remplir l'objet par la couleur sollicitée au moyen de la commande
 [fillcolor=couleur].
- **Points** : On peut changer le style d'affichage des points (par défaut, un disque noir).
 Les 10 styles sont :
 (square, square*, *, triangle, triangle*, diamond, diamond*, pentagon, pentagon*).
- **Longueur des flèches** : [arrowlength=longueur] permet de changer la longueur des flèches.

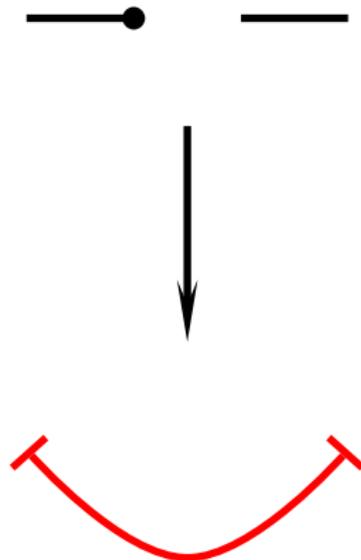
Code source :

```

\begin{pspicture}(5,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray, gridlabels =
7pt](0,0)(8,2)
\psline[linewidth = 2pt, linecolor = orange](0,0)(2,1)
\psline[linestyle = dashed, linecolor = blue](2,1)(4,0)
\psline[linestyle = dotted, linecolor = red](4,0)(6,1)
\psline[linestyle = dashed, dash = 1pt 3pt 5pt 3pt, linecolor = green](6,1)(8,0)
\psline[linecolor = red]{- >}(8,0)(8,2)
\end{pspicture}

```





Options :

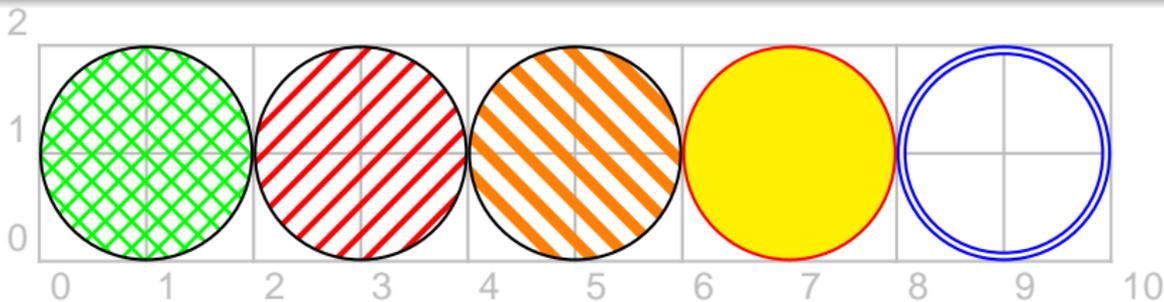
- **Double trait** : `doubleline=true`
- **Hachures** : Pour les figures fermées, on peut définir le type de remplissage :
 - ▶ `fillstyle=valeur` : motif de remplissage, valeur pouvant être :
 - `crosshatch` : hachures croisées à 45° ,
 - `hlines` : hachures simple à 45° ,
 - `vlines` : hachures simple à -45° ,
 - `solid` : plein :
 - ▶ `fillcolor`=couleur de fond ;
 - ▶ `hatchcolor`=couleur des hachures ;
 - ▶ `hatchwidth`=épaisseur du trait ;
 - ▶ `hatchsep`=espacement des traits ;
 - ▶ `hatchangle`=angle des traits.

Code source :

```

\begin{pspicture}(10,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray](0,0)(10,2)
\pscircle[fillstyle = crosshatch, fillcolor = red, hatchcolor = green, hatchwidth = 1pt](1,1){1}
\pscircle[fillstyle = hlines, fillcolor = red, hatchcolor = red, hatchwidth = 1.5pt](3,1){1}
\pscircle[fillstyle = vlines, hatchcolor = orange, hatchwidth = 3pt](5,1){1}
\pscircle[fillstyle = solid, fillcolor = yellow, linecolor = red](7,1){1}
\pscircle[doubleline = true, linecolor = blue](9,1){1}
\end{pspicture}

```



Figures usuelles :

- Rectangles, polygones, etc.
- Paraboles.
- Cercles, disques et ellipses.
- Secteurs et arcs.

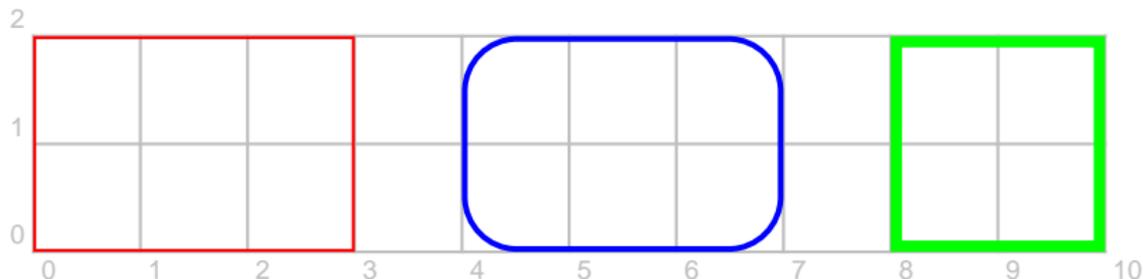
Rectangles :

Syntaxe

`\psframe[opt](x_0, y_0)(x_1, y_1)` permet de tracer un rectangle dont les extrémités de la diagonale ont pour coordonnées (x_0, y_0) et (x_1, y_1) .

Code source :

```
\begin{pspicture}(10,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray, gridlabels =
7pt](0,0)(10,2)
\psframe[linecolor = red](0,0)(3,2)
\psframe[framearc = 0.5, linewidth = 1.5pt, linecolor = blue](4,0)(7,2)
\psframe[linecolor = green, linewidth = 3pt](8,0)(10,2)
\end{pspicture}
```



Polygones :

Syntaxe

`\pspolygon[opt](x_0, y_0)(x_1, y_1) \cdots (x_n, y_n)` permet de tracer un polygone.

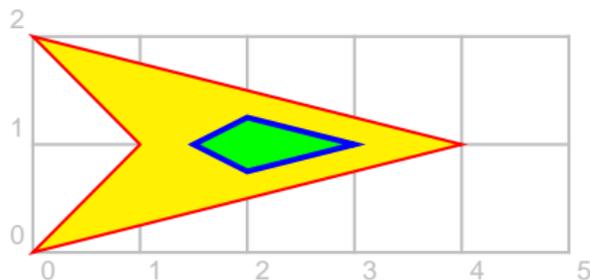
Polygones :

Syntaxe

`\pspolygon[opt](x_0, y_0)(x_1, y_1) \cdots (x_n, y_n)` permet de tracer un polygone.

Code source :

```
\begin{pspicture}(5,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray](0,0)(5,2)
\pspolygon[linecolor = red, fillstyle = solid, fillcolor = yellow](0,0)(4,1)(0,2)(1,1)
\pspolygon[fillstyle = solid, fillcolor = green, linecolor = blue, linewidth = 1.5pt](3,1)(2,1.25)(1.5,1)
\end{pspicture}
```



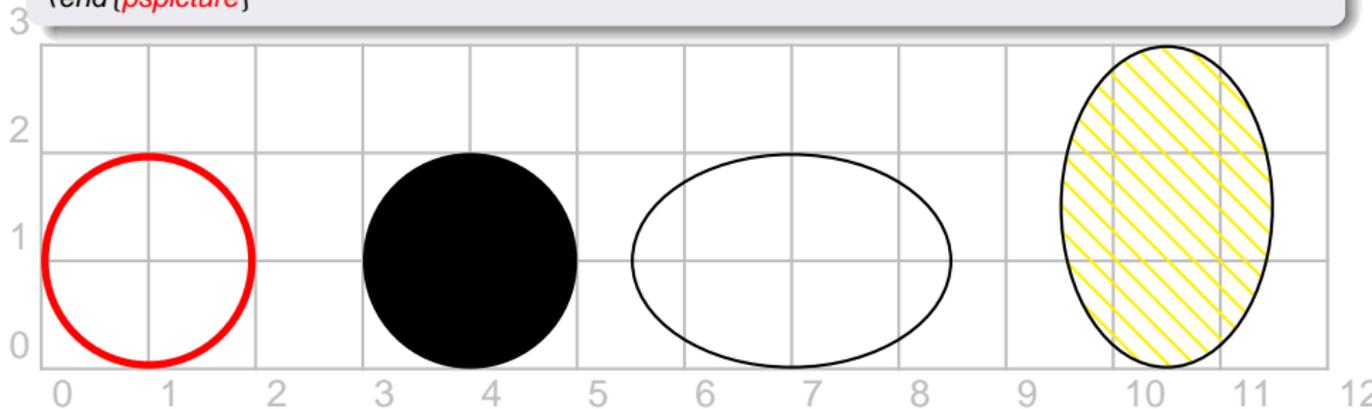
Cercle, disques et ellipses :

Syntaxe

- `\pscircle[opt](x_0, y_0){ r }` permet de tracer le cercle de coordonnées (x_0, y_0) et de rayon r .
- **La version étoilée** dessine le disque (en noir).
- `\psellipse[opt](x_0, y_0)(r_x, r_y)` permet de tracer l'ellipse de centre de coordonnées (x_0, y_0) et de rayon horizontal r_x et de rayon vertical r_y .

Code source :

```
\begin{pspicture}(12,4)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray](0,0)(12,4)
\pscicle(1,1)1
\pscicle*(4,1)1
\psellipse(7,1)(1.5,1)
\psellipse[fillstyle = vlines, hatchcolor = yellow](10.5,1.5)(1,1.5)
\end{pspicture}
```



Secteurs :

Syntaxe

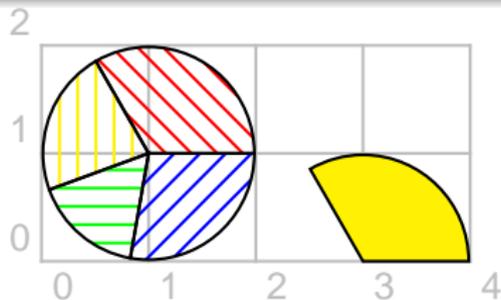
`\pswedge(x0, y0){r}{a}{b}` permet de tracer le secteur de centre de coordonnées (x_0, y_0) de rayon r depuis l'angle a jusqu'à l'angle b .

Code source :

```

\begin{pspicture}(4,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray](0,0)(4,2)
\pswedge[fillstyle = hlines, hatchangle = 135, hatchcolor = red](1,1){1}{0}{120}
\pswedge[fillstyle = hlines, hatchangle = 90, hatchcolor = yellow](1,1){1}{120}{200}
\pswedge[fillstyle = hlines, hatchangle = 0, hatchcolor = green](1,1){1}{200}{260}
\pswedge[fillstyle = hlines, hatchangle = 45, hatchcolor = blue](1,1){1}{260}{360}
\pswedge[fillstyle = solid, fillcolor = yellow](3,0){1}{0}{120}
\end{pspicture}

```



Arcs :

Syntaxe

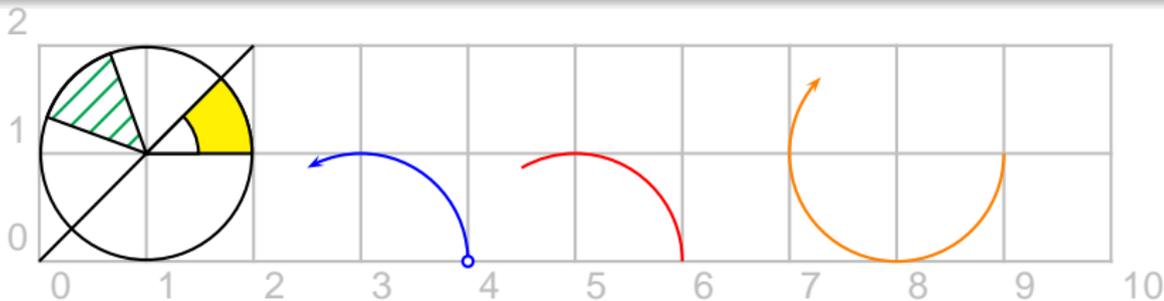
- `\psarc(x0,y0){r}{a}{b}` permet de tracer l'arc de centre de coordonnées (x_0, y_0) de rayon r depuis l'angle a jusqu'à l'angle b .
- Pour dessiner un arc dans le sens indirect, on utilise : `\psarc(x0,y0){r}{a}{b}`.

Code source :

```

\begin{pspicture}(10,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray](0,0)(10,2)
\psarc[linecolor = red](5,0){1}{0}{120}
\psarc[linecolor = blue]{o- >}(3,0){1}{0}{120}
\psarcn[linecolor = orange]{- >}(8,1){1}{0}{135}
\pscircle(1,1){1}
\psline(0,0)(2,2)
\pswedge[fillstyle = solid, fillcolor = yellow](1,1){1}{0}{45}
\pswedge[fillstyle = solid, fillcolor = white](1,1){0.5}{0}{45}
\end{pspicture}

```



Un repère :

Syntaxe

```
\psaxes{(x_min, y_min)(x_max, y_max)}
```

OU

```
\psaxes{fleche}{(x_min, y_min)(x_max, y_max)}
```

Un repère :

Syntaxe

```
\psaxes{(x_min, y_min)(x_max, y_max)}
```

ou

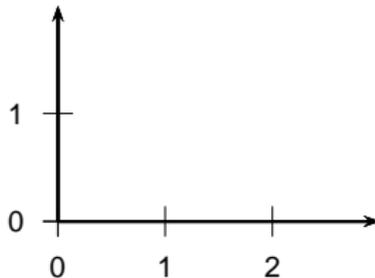
```
\psaxes{fleche}{(x_min, y_min)(x_max, y_max)}
```

Code source :

```
\begin{pspicture}(0,0)(3,2)
```

```
\psaxes{->}(0,0)(3,2)
```

```
\end{pspicture}
```



L'axe « horizontal » seul :

Syntaxe

Pour avoir ce seul axe **xAxis**, on demande de ne pas afficher l'autre axe **yAxis** avec la l'instruction **yAxis=false**.

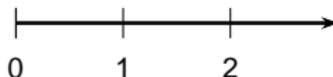
L'axe « horizontal » seul :

Syntaxe

Pour avoir ce seul axe **xAxis**, on demande de ne pas afficher l'autre axe **yAxis** avec la l'instruction **yAxis=false**.

Code source :

```
\begin{pspicture}(0,0)(3,2)
\psaxes[yAxis=false]{- >}(0,0)(3,2)
\end{pspicture}
```

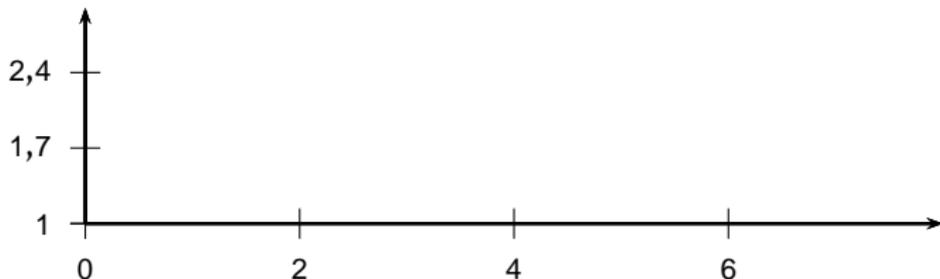


Syntaxe

- Si on préfère la virgule, on utilise l'option **comma**.
- Les options $Dx = \text{valeur}$ et $Dy = \text{valeur}$ permettent de définir le pas entre deux graduations.
- Les options $Ox = \text{valeur}$ et $Oy = \text{valeur}$ permettent de donner la valeur initiale de graduation.

Code source :

```
\begin{pspicture}(0,0)(8,2)
\psaxes[comma,Oy=1,Dy=0.7,Dx=2]{->}(0,0)(8,2)
\end{pspicture}
```



(Dé)Placer des objets :

Syntaxe

```
\rput[point de reference]{angle}(x_0, y_0){objet}
```

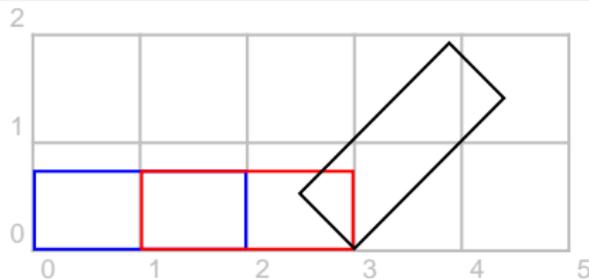
(Dé)Placer des objets :

Syntaxe

$$\backslash rput[\textit{point de reference}]{\textit{angle}}(x_0, y_0)\{\textit{objet}\}$$

Code source :

```
\begin{pspicture}(5,2)
\psgrid[subgriddiv = 0, gridcolor = lightgray, gridlabelcolor = lightgray, gridlabels = 7pt]
\psframe[linecolor = blue](2, .75)
\rrput(1,0){\psframe[linecolor = red](2, .75)}
\rrput{45}(3,0){\psframe(2, .75)}
\end{pspicture}
```



Placer des labels :

Syntaxe

```
\uput[sep. label]{angle}(x,y){texte}
```

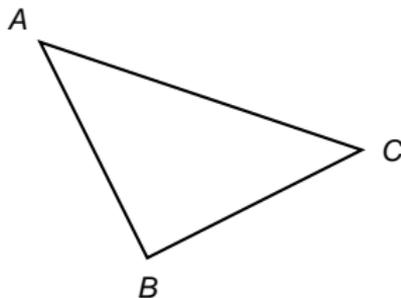
Placer des labels :

Syntaxe

$$\backslash\text{uput}[\text{sep. label}]{\text{angle}}(x,y)\{\text{texte}\}$$

Code source :

```
\begin{pspicture}(5,4)
\pspolygon(1,3)(2,1)(4,2)
\uput[ul](1,3){A} \uput[d](2,1){B} \uput[r](4,2){C}
\end{pspicture}
```



Graphiques et courbes

Syntaxe

```
\package{pstricks – add}
```

Graphiques et courbes

Syntaxe

```
\package{pstricks – add}
```

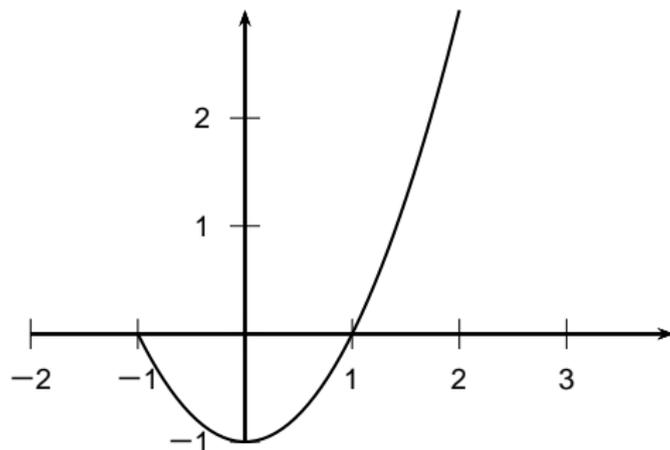
Syntaxe

Pour tracer, sur l'intervalle $[x_{\min}; x_{\max}]$, la courbe représentative de la fonction , la commande de base est : `\psplot[algebraic = true]`

Courbe d'équation $y = x^2 - 1$.

Code source :

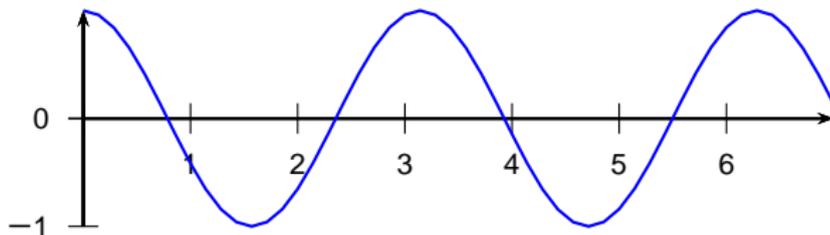
```
\begin{pspicture}(-1,-2)(4,3)
\psset{algebraic=true}
\psaxes{->}(0,0)(-1,-2)(4,3)
\psplot{-1}{2}{x^2-1}
\end{pspicture}
```



Courbe d'équation $y = \cos 2x$

Code source :

```
\begin{pspicture}(-0.5,-1.5)(8.5,1.5)
\psset{algebraic=true}
\psaxes{->}(0,0)(0,-1)(7,1)
\psplot[linecolor=blue]{0}{7}{cos(2*x)}
\end{pspicture}
```

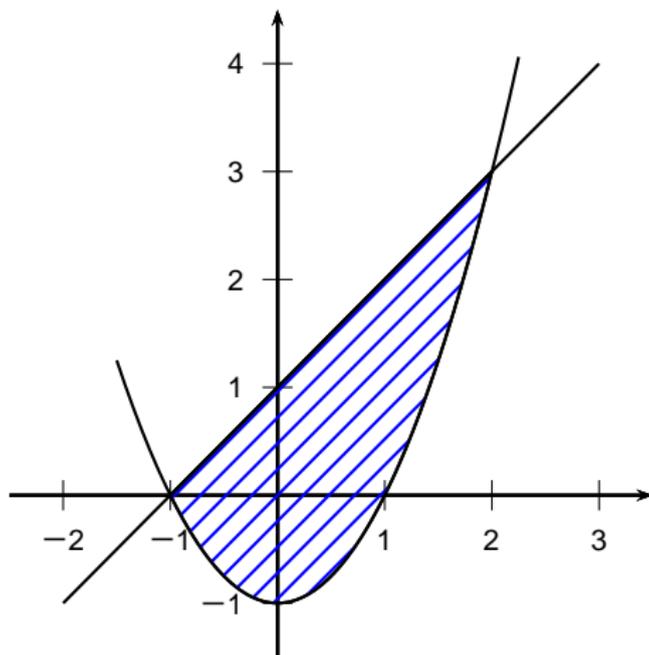


Comment représenter l'intégrale $\int_{-1}^2 ((x^2 - 1) - (x + 1)) dx$

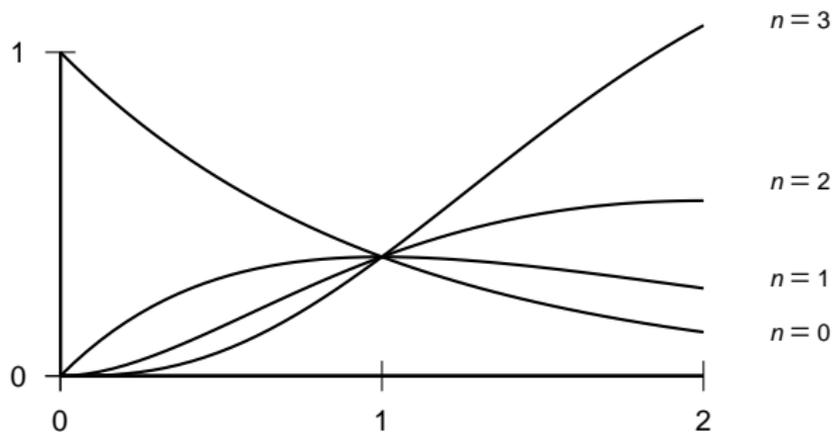
Code source :

```
\begin{pspicture}(-2.5,-1.5)(3.5,4.5)
\psset{algebraic=true}
\psaxes{->}(0,0)(-2.5,-1.5)(3.5,4.5)
\psplot{-1.5}{2.25}{x^2-1}
\psplot{-2}{3}{x+1}
\pscustom[fillstyle=hlines,hatchcolor=blue]{\psplot{-1}{2}{x^2-1}\psplot{-1}{2}{x+1}}
\end{pspicture}
```

Comment représenter l'intégrale $\int_{-1}^2 ((x^2 - 1) - (x + 1)) dx$



Exemple : Comment représenter les courbes des fonctions $x^n e^{-x}$, pour $n \in 0, 1, 2, 3$ sur $[0, 2]$



Courbe d'équation $y = \sin x / x$.

