

Dessin géométrique en L^AT_EX avec PStricks

Z, auctore

2 juillet 2007

Résumé

On suppose que le lecteur dispose d'une distribution L^AT_EX et en connaît l'utilisation habituelle. L'inclusion dans les documents L^AT_EX d'images obtenues par captures d'écran de logiciels de dessin à la souris n'est absolument pas satisfaisante ; l'extension PStricks de T. Van Zandt permet de réaliser des figures simples en géométrie plane, en les décrivant par la saisie d'un code directement dans le fichier `.tex`. Ce document présente différentes commandes de cette extension, sans prétention car ce ne sont pas, loin s'en faut, les seules possibilités de celle-ci ! Deux documents viendront en complément, l'un abordant les courbes et les noeuds en `pstricks`, l'autre consacré à l'extraordinaire extension `pst-eucl`.

Généralités

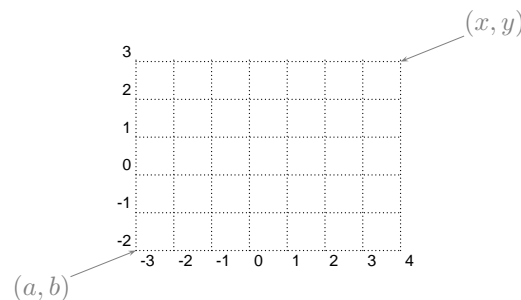
L'extension est appelée dans le préambule du document par `\usepackage{pst-all}`. La compilation doit se faire en L^AT_EX (et pas en pdfL^AT_EX), le `.dvi` étant converti en `.ps` (le `.ps` est éventuellement converti en `.pdf`) pour la visualisation des figures : il faut penser à régler la construction rapide notamment avec Kile, TeXmaker ou TeXnicCenter. La description des figures se fait généralement au moyen des coordonnées cartésiennes notées conventionnellement (x, y) – le séparateur est ici la **virgule**, les décimaux étant saisis à la façon anglo-saxonne. La place pour une figure sur le document est réservée avec l'environnement `pspicture`.

1. Grille [grid]. Les coordonnées (a, b) indiquent le coin inférieur gauche et (x, y) le coin supérieur droit :

```
\psgrid(a,b)(x,y)
```

Exemple 1

```
\begin{pspicture}(-3,-2)(4,3)
\psgrid % équiv. à \psgrid(-3,-2)(4,3)
\end{pspicture}
```

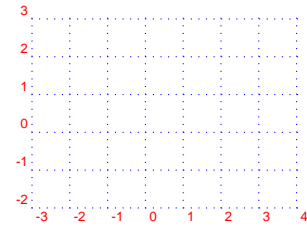


Grille : effet des options

Le quadrillage, les nombres (étiquettes) sur les axes, la couleur, etc. sont paramétrables :

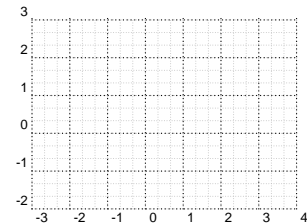
Exemple 2

```
\begin{pspicture}(-3,-2)(4,3)
\psgrid[griddots=5,gridcolor=blue,%
gridlabelcolor=red,gridlabels=5pt]
\end{pspicture}
```



Exemple 3

```
\begin{pspicture}(-3,-2)(4,3)
\psgrid[gridlabels=5pt,subgriddiv=3,%
subgriddots=5,subgridcolor=lightgray]
\end{pspicture}
```



Dans la suite, je laisse apparaître une grille sur les figures pour faciliter la lecture du repérage ; par contre je n'indique plus les

```
\begin{pspicture}(a,b)(x,y) \psgrid \end{pspicture}
```

pour ne pas surcharger le code. Des options globales sont fixées par

```
\psset{showpoints=true, % affichage des points
dotstyle=*, % style de point
dotsize=3pt, % taille de point
linewidth=0.8pt, % épaisseur des traits
subgriddiv=1, % grille divisée aux unités
griddots=10, % nombre de points sur le côté du carré
gridlabels=6pt, % taille des étiquettes
gridwidth=0.5pt, % épaisseur du trait de quadrillage
xunit=0.5, % facteur d'unité en abscisse
yunit=0.5, % facteur d'unité en ordonnée
runit=0.5, % facteur d'unité en radial
gridcolor=lightgray} % couleur de la grille
```

2. Axes. Avec les mêmes conventions que pour la grille :

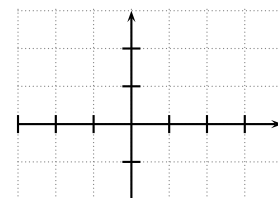
```
\psaxes{flèche}(p,q)(a,b)(x,y)
```

On code les « flèches » de façon intuitive, comme par exemple

```
-> ou *-* ou ->> ou <-> ...
```

Exemple 4

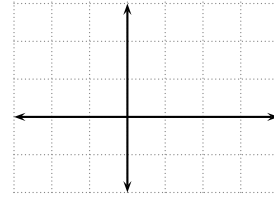
```
\psaxes[labels=none]{->}(0,0)(-3,-2)(4,3)
```



Axes : effet des options

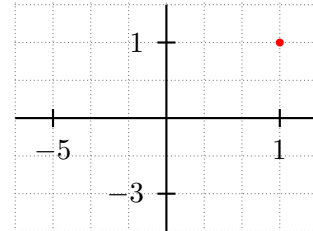
Exemple 5

```
\psaxes[ticks=none,labels=none]{<->}%
(0,0)(-3,-2)(4,3)
```



Exemple 6

```
\psdot(3,2) % voir section 3 pour cette commande
\psaxes[Ox=-2,Oy=-1,Dx=3,Dy=2](0,0)(-4,-3)(4,3)
```



3. Point [dot].

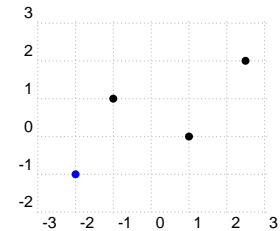
```
\psdot(x,y)
```

```
\psdots(x,y)...(z,t)
```

```
\SpecialCoor \psdots(r;a)
```

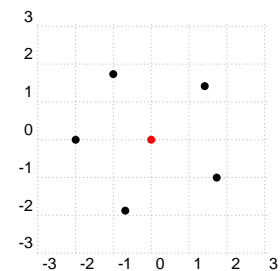
Exemple 7

```
\psdot[linecolor=blue](-2,-1) % un seul point
\psdots(-1,1)(1,0)(2.5,2) % plusieurs points
```



Exemple 8

```
\SpecialCoor % autorise les coordonnées polaires
\psdots(2;45)(2;120)(2;180)(2;250)(2;-30)
\psdot[linecolor=red](0,0)
```



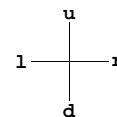
4. Étiquette. Pour attacher un texte à un point (x, y) : donner un nom, placer une légende sur une figure...

```
\uput{s}[d]{r}(x,y){nom}
```

- s : paramètre d'espace (labelsep) entre (x, y) et le nom ;
- d : paramètre **obligatoire** de direction autour de (x, y) pour le nom ;
- r : paramètre de rotation du nom.

Le paramètre de direction peut être un angle (en degrés, sens trigo) ou bien une combinaison de

u pour **up** l pour **left** d pour **down** r pour **right**.

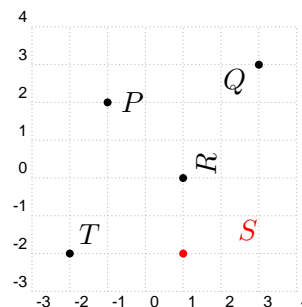


Exemple 9

```
\psdot(-1,2)
\uput[0](-1,2){$P$}

\psdots(3,3)(1,0)(-2,-2)
\psdot[linecolor=red](1,-2)

\uput[-145](3,3){$Q$}% direction du nom
\uput[30]{90}(1,0){$R$} % rotation du nom
\uput{1.5}[20](1,-2){\red $$$} % espacement 1 cm
\uput[ur](-2,-2){$T$} % direction uldr du nom
```



5. Ligne brisée [line]. On énumère les coordonnées des points à relier dans l'ordre voulu.

$\psline{\text{flèche}}(a,b)\dots(x,y)$

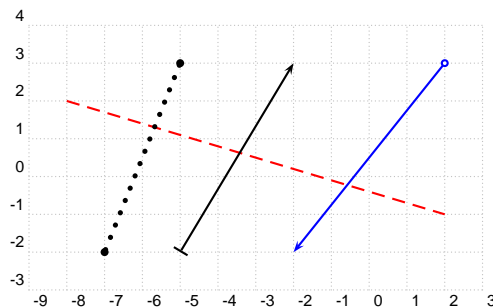
Exemple 10

```
\psline[linecolor=red,%
linestyle=dashed,%
showpoints=false](-8,2)(2,-1)

\psline[linestyle=dotted,%
linewidth=2pt](-6,-2)(-2,3)

\psline{|->}(-3,-2)(2,3)

\psline[linecolor=blue]{<-o}(1,-2)(6,3)
```



Exemple 11

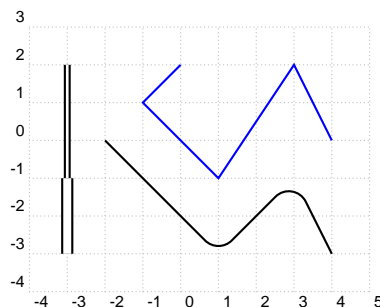
```
\psset{showpoints=false}

\psline[doubleline=true](-3,-2)(-3,-1)

\psline[doubleline=true,%
doublesep=3pt](-3,-1)(-3,-3)

\psline[linecolor=blue]%
(0,2)(-1,1)(1,-1)(3,2)(4,0)

\psline[linearc=.5](-2,0)(1,-3)(3,-1)(4,-3)
```

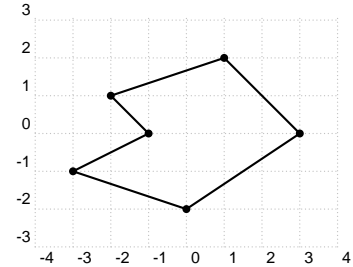


6. Polygone [polygon]. Même syntaxe que la précédente ; la commande ferme automatiquement la ligne brisée.

$$\boxed{\backslash\text{pspolygon}(a,b)\dots(x,y)}$$

Exemple 12

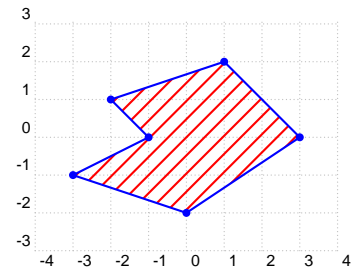
```
\pspolygon(-3,-1)(-1,0)(-2,1)(1,2)(3,0)(0,-2)
```



Polygone : effet des options

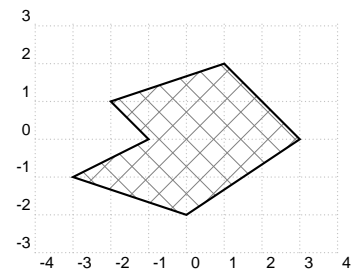
Exemple 13

```
\pspolygon[linecolor=blue,%
  fillstyle=vlines,%
  hatchangle=-45,%
  hatchcolor=red]%
(-3,-1)(-1,0)(-2,1)(1,2)(3,0)(0,-2)
```



Exemple 14

```
\pspolygon[showpoints=false,%
  fillstyle=crosshatch,%
  hatchcolor=gray,%
  hatchwidth=0.3pt,%
  hatchsep=7pt]%
(-3,-1)(-1,0)(-2,1)(1,2)(3,0)(0,-2)
```

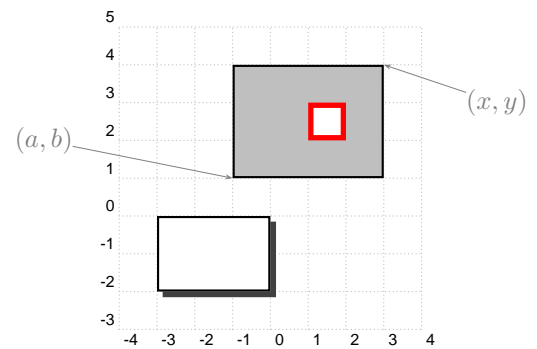


Cas particulier : rectangle [frame]

$$\boxed{\backslash\text{psframe}(a,b)(x,y)}$$

Exemple 15

```
\psframe[shadow=true](-3,-2)(0,0)
\psframe[fillstyle=solid,%
  fillcolor=lightgray](-1,1)(3,4)
\psframe[fillstyle=solid,%
  linecolor=red,%
  linewidth=2pt](1,2)(2,3)
```



7. Cercle [circle].

$$\boxed{\backslash\text{pscircle}(x,y)\{\text{rayon}\}}$$

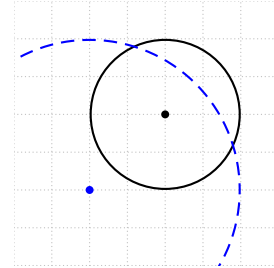
Exemple 16

```
\begin{pspicture*}(-3,-3)(4,4)
\psgrid

\psdot(1,1)
\pscircle(1,1){2}

\psdot[linecolor=blue](-1,-1)
\pscircle[linestyle=dashed,linecolor=blue](-1,-1){4}

\end{pspicture*} % la forme étoilée rogne la figure
```



Cas particulier : arc de cercle

$$\boxed{\backslash\text{psarc}\{\text{flèche}\}(x,y)\{\text{rayon}\}\{\text{angle1}\}\{\text{angle2}\}}$$

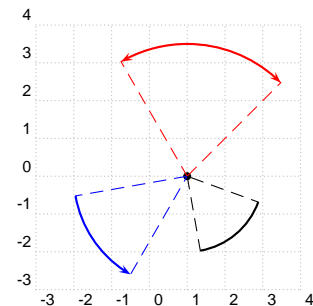
Exemple 17

```
\psdot(1,0)

\psarc[linecolor=red]{<->}(1,0){3.5}\{45}\{120}

\psarc[linecolor=blue]{->}(1,0){3}\{190}\{240}

\psarc(1,0){2}\{-80}\{-20}
```



8. Placement, rotation. On peut placer n'importe quel objet sur la feuille, en lui faisant éventuellement subir une rotation, par rapport à (x, y) .

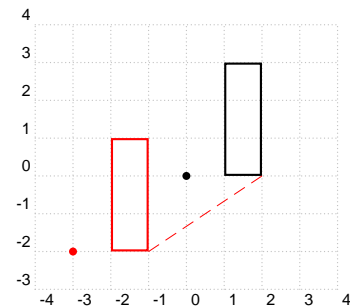
$$\boxed{\backslash\text{rput}\{\text{angle}\}(x,y)\{\text{objet}\}}$$

Exemple 18

```
\psdot(0,0) \psframe(1,0)(2,3)

\psdot[linecolor=red](-3,-2)

\rput{0}(-3,-2)\psframe[linecolor=red](1,0)(2,3)
\psline[linecolor=red,linestyle=dashed,%
linewidth=.02,showpoints=false](2,0)(-1,-2)
```



Le code précédent donne une **translation** du rectangle initial.

Exemple 19

```

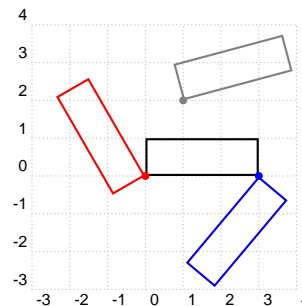
\psframe(0,0)(3,1)

\psdot[linecolor=red](0,0)
\rput{120}(0,0){\psframe[linecolor=red](0,0)(3,1)}

\psdot[linecolor=blue](3,0)
\rput{-130}(3,0){\psframe[linecolor=blue](0,0)(3,1)}

\psdot[linecolor=gray](1,2)
\rput{15}(1,2){\psframe[linecolor=gray](0,0)(3,1)}

```



Les codes précédents donnent une **rotation** du rectangle initial.

Exemple 20

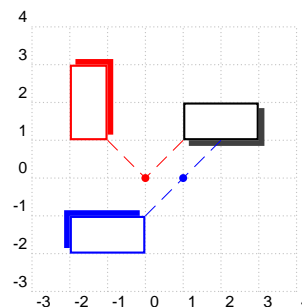
```

\psframe[shadow=true](1,2)(3,1)

\psdot[linecolor=red](0,0)
\rput{90}(0,0){\psframe[linecolor=red,%
    shadow=true,shadowcolor=red](1,2)(3,1)}
\psline[linecolor=red,linestyle=dashed,%
    showpoints=false,linewidth=.02](1,1)(0,0)(-1,1)

\psdot[linecolor=blue](1,0)
\rput{-180}(1,0){\psframe[linecolor=blue,%
    shadow=true,shadowcolor=red](1,2)(3,1)}
\psline[linecolor=blue,linestyle=dashed,%
    showpoints=false,linewidth=.02](2,1)(0,-1)

```



Les deux derniers codes donnent un quart de tour et le demi-tour. Le résultat peut surprendre lorsque le centre ou le coin inférieur gauche du rectangle n'est pas $(0,0)$: il y a d'abord une translation, puis le `\psframe` est défini à partir du (x,y) du `\rput`.

9. Homothétie.

$$\boxed{\backslash\text{scalebox}\{c\}\{\text{objet}\}}$$

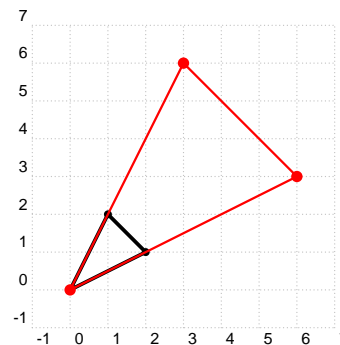
$$\boxed{\backslash\text{scalebox}\{cx\ cy\}\{\text{objet}\}}$$
Exemple 21

```

\pspolygon[linewidth=.1](0,0)(2,1)(1,2)

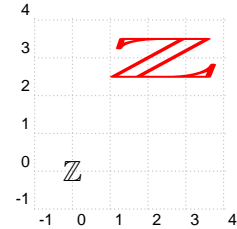
\scalebox{3}{\pspolygon[linecolor=red,%
    linewidth=.02,dotsize=.1](0,0)(2,1)(1,2)}

```



Exemple 22

```
\rput(0,0){$\mathbb{Z}$}
\scalebox{6 2}{\rput(.4,1.5){\red $\mathbb{Z}$}}
```

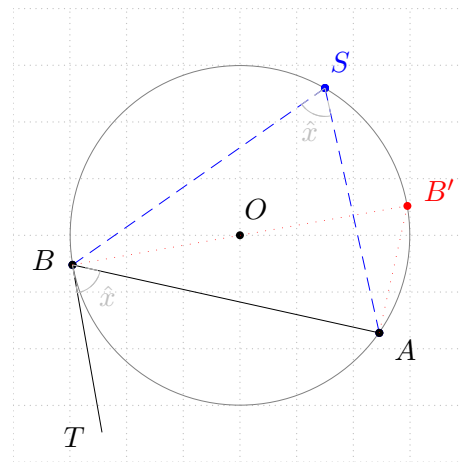


10. Outils d'ébauche. Les logiciels **D^r Geo** et **PST+** permettent de se décharger de la partie fastidieuse du travail en générant l'essentiel du code pstricks, que l'on peut ajuster ensuite au besoin. **D^r Geo** ne fonctionne que sous Linux à ce que je sache. C'est un logiciel de dessin à la souris disposant d'une fonction « Exporter la figure en \LaTeX ». **PST+** fonctionne sous XP et sous Linux. Ce logiciel permet d'obtenir la génération du code pstricks et la prévisualisation de la figure qui est décrite au moyen d'un langage relativement intuitif.

Le code de la figure ci-dessous a été généré pour l'essentiel avec PST+. Sans cela, l'obtention de certaines coordonnées précises au dix-millième aurait été difficile.

Exemple 23

```
\psset{xunit=1.5,yunit=1.5,runit=1.5,linewidth=0.01}
\begin{pspicture*}(-4,-4.5)(4,4)
\psdots(0,0)
\uput{0.3}[60](0,0){$O$}
\pscircle[linecolor=gray](0,0){3}
\psdots(2.4575,-1.7207)
\uput{0.3}[-35](2.4575,-1.7207){$A$}
\psdots(-2.9544,-0.5209)
\uput{0.3}[170](-2.9544,-0.5209){$B$}
\uput{0.6}[-40](-2.9544,-0.5209){\lightgray $\hat{x}$}
\psdots(1.5,2.5981)
\uput{0.3}[60](1.5,2.5981){\blue $S$}
\uput{0.6}[250](1.5,2.5981){\lightgray $\hat{x}$}
\psline[showpoints=false](-2.4335,-3.4754)%
(-2.9544,-0.5209)
\psline[linecolor=blue,linestyle=dashed]%
(2.4575,-1.7207)(1.5,2.5981)
\psline[linecolor=blue,linestyle=dashed]%
(-2.9544,-0.5209)(1.5,2.5981)
\uput{0.3}[190](-2.4335,-3.4754){$T$}
\psline(-2.9544,-0.5209)(2.4575,-1.7207)
\psarc[linecolor=lightgray](1.5,2.5981){0.5}{-145}{-77.5}
\psarc[linecolor=lightgray](-2.9544,-0.5209){0.5000}{-80}{-12.5}
\psdots[linecolor=red](2.9544,0.5209)
\uput{0.3}[30](2.9544,0.5209){\red $B'$}
\psline[showpoints=false,linecolor=red,linestyle=dotted]%
(-2.9544,-0.5209)(2.9544,0.5209)
\psline[showpoints=false,linecolor=red,linestyle=dotted]%
(2.4575,-1.7207)(2.9544,0.5209)
\end{pspicture*}
```



L'un des inconvénients est de devoir donner un grand nombre de coordonnées, ce qui complique les modifications ultérieures de la figure dès que le code dépasse quelques lignes. C'est cette limitation que les **noeuds** ou l'extension **pst-eucl** permettent de dépasser.