Cahiers GUTenberg n°16 — Février 1994

# Présentation de PSTricks

Denis GIROU

*CNRS/IDRIS, BP 167, 91403 Orsay Cedex, France,* **<Denis.Girou@idris.fr>**

**Résumé**. Les macros-commandes de l'extension **PSTricks**, développées par Timothy Van ZANDT, offrent un ensemble impressionnant de nouvelles possibilités aux utilisateurs de (LA)TEX, en leur donnant directement accès à une très large part des fonctionnalités de PostScript, y compris le traitement complet de la couleur. De plus, un développement associé à **PSTricks**, **Seminar**, permet de réaliser facilement des transparents de grande qualité. Cet article veut présenter, à travers un grand nombre d'exemples, des plus simples aux plus complexes, un panorama complet des différents apports de **PSTricks**, en se plaçant du strict point de vue d'un utilisateur.

**Abstract**. *The macro-commands in the* **PSTricks** *package, developed by Timothy Van* ZANDT, *offer impressive new capabilities to (LA)TEX users, by giving them direct access to much of the power of PostScript, including complete support of color. In addition,* **Seminar**, *a development associated with* **PSTricks**, *allows users to easily make transparencies of great quality. This article will present, through a large number of examples, from the simplest to the more complex, the diverse features of* **PSTricks**, *from the point of vue of an end user.*

## Sommaire

*Présentation de*

**PSTricks**

Denis Girou

Institut du Développement
et des Ressources en
Informatique Scientifique

Centre National de la
Recherche Scientifique

BP 167 — 91403 Orsay Cedex

Messagerie : `Denis.Girou@idris.fr`

*Quand quelqu'un nous rêve ensemble — nous nous rencontrons.*
Marina TSVÉTAÏEVA
Lettre à Rainer Maria RILKE du 2 août 1926 [1]

# 1. Introduction

LES avantages de (LA)TEX dans la publication assistée par ordinateur sont connus, en tout premier lieu par les lecteurs des **Cahiers GUTenberg**. Mais ses manques le sont aussi, hors même les interminables débats du *tel-tel* [2]. Car, surtout aujourd'hui qu'une multitude de logiciels de la micro-informatique l'autorise, on a souvent envie d'introduire dans ses documents des graphiques, des couleurs, des effets particuliers de formatage et de mise en page, toutes choses que ne permet pas directement (LA)TEX, parce que Donald KNUTH n'en avait nul besoin au moment où il a conçu et réalisé TEX, et que de toute façon les possibilités de l'informatique de cette époque ne l'auraient pas permis...

MAIS aujourd'hui ces manques sont de moins en moins bien acceptés, au vu de ce qu'offrent les autres logiciels. Depuis déjà assez longtemps, deux voies sont suivies pour remédier à ces graves déficits [3] : METAFONT, selon la direction donnée directement par Donald KNUTH (voir notamment [**MetaPost**], [**mfpic**] et [Hœnig 92] — mais en tout état de cause cela n'autorise pas l'utilisation de la couleur), et PostScript, qui permet en principe *presque tous* les traitements et offre également l'emploi illimité de la couleur, ce qui est devenu essentiel à présent avec l'évolution des matériels d'affichage et d'impression.

OUTRE la simple insertion de fichiers PostScript à la taille voulue, tâche pour laquelle il y a maintenant plusieurs solutions tout à fait satisfaisantes, il y a déjà eu un grand nombre de développements effectués ici et là pour permettre de disposer, depuis (LA)TEX, de certaines des possibilités offertes par PostScript. Ces développements sont de nature et d'importance très diverses. Parmi les plus intéressants, citons : [**TEXdraw**], [**LameTEX**], [**colordvi.sty**], [**ColorRgb.TEX**]. Néanmoins, celui qui nous semble de loin le plus riche est **PSTricks**, de Timothy Van ZANDT [4].

1. Rainer Maria RILKE, Boris PASTERNAK, Marina TSVÉTAÏEVA, *Correspondance à trois*, Éditions Gallimard, 1983.

2. Tel écran-tel écrit ... ou *wysiwyg*.

3. Sans parler des nombreuses extensions apportées directement aux environnements standard de LATEX, en particulier [**epic.sty**], [**eepic.sty**], [**PICTEX**], et des récents développements **DraTEX** et **AlDraTEX** [Gurari 94].

4. Pour la disponibilité des fichiers correspondants, voir les pages 2 à 4.

C'EST un vaste ensemble de macros-commandes utilisables directement depuis (LA)TEX. Elles permettent d'avoir du texte avec les couleurs choisies, de définir des objets graphiques simples (lignes, polygones, cercles, flèches, etc.) ou complexes (grilles, arbres, etc.) — que l'on peut bien sûr combiner — et de manipuler des parties du texte et du document (rotations, changements d'échelles, transformations). L'éventail des possibilités est donc considérable et n'est guère limité que par l'imagination... De plus, comme avec (LA)TEX, on peut mettre en œuvre toutes les capacités d'un langage de programmation puissant afin de construire des applications complexes, de faire générer le code par des pré-processeurs, etc.

LE manuel de référence et d'utilisation de **PSTricks** est très complet (plus d'une centaine de pages...), avec un grand nombre d'exemples simples, mais nous souhaitons présenter ici, du strict point de vue d'un utilisateur final et hors de toute considération technique particulière, les caractéristiques essentielles de ce logiciel, et ceci à l'aide d'une série d'exemples, des plus simples aux plus sophistiqués. Les illustrations de base sont adaptées à partir de celles du manuel lui-même, et les autres sont soit repris de nos propres utilisations et développements antérieurs, soit conçus pour cet article, afin d'illustrer auprès d'un public diversifié les vastes possibilités offertes par **PSTricks**.

IL est toutefois clair que le but de cet article n'est pas d'*apprendre* à utiliser **PSTricks** (le manuel est fait pour cela), mais de montrer assez exhaustivement ses possibilités, et de convaincre de son grand intérêt...[5]

# 2. Commandes de base

## 2.1. Couleurs dans le texte

La mise en couleur du texte se fait très simplement, suivant le même principe qu'en LATEX le changement de police : **{\blue couleur}**[6,7]. Plus d'une centaine de couleurs sont prédéfinies, et l'on peut bien sûr en ajouter.

5. La version de **PSTricks** utilisée pour réaliser ces exemples a été la version 0.93a, augmentée d'un certain nombre de nouvelles extensions en cours de développement, qui seront incluses dans la version 0.94.

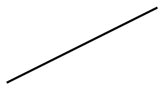
6. Dans quelques cas (au début d'un **\item**, dans certaines entités d'un tableau, etc.), il est nécessaire de faire précéder les commandes de changement de couleur de la macro **\leavevmode**, pour leur garantir une *portée* correcte.

7. Il est parfaitement possible de spécifier des couleurs en mode **verbatim**, en utilisant des commandes de l'extension **FancyVerb**, mise prochainement à disposition par Timothy Van Zandt. Toutefois, le codage est un plus complexe. Se reporter à la documentation.

## 2.2. Objets de base

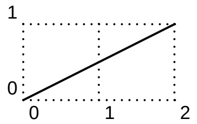
Par défaut, l'on est dans un système de coordonnées en X et Y dont l'unité est de 1 cm, et dont l'origine est toujours définie en bas à gauche.

Ainsi tracer une ligne depuis l'origine s'obtient par :



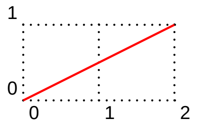
```
\psline(2,1)
```

puisque le système de coordonnées est défini ainsi [8] :

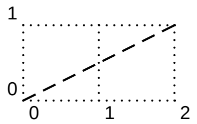



```
\psline(2,1)
```

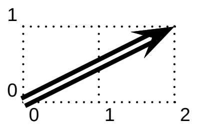
De plus un grand nombre d'attributs permettent de modifier les caractéristiques des objets représentés [9] :



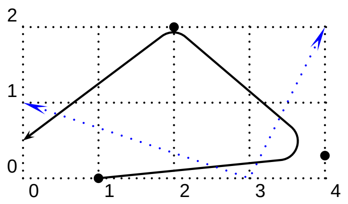

```
\psline[linecolor=red](2,1)
```




```
\psline[linestyle=dashed](2,1)
```



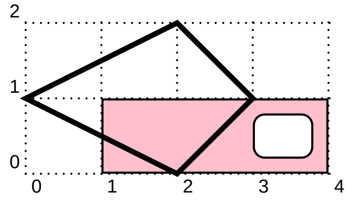

```
\psline[linewidth=0.6mm,doubleline=true,
        doublesep=0.5mm]{->}(2,1)
```



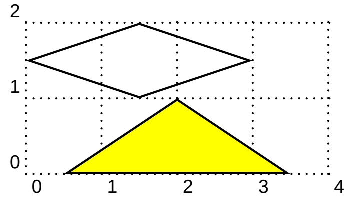

```
\psline[linearc=0.25,showpoints=true]{->}%
        (1,0)(4,0.3)(2,2)(0,0.5)
\psline[linestyle=dotted,linecolor=blue,
        arrowlength=3]{<->}(0,1)(3,0)(4,2)
```

8. Il est également possible de travailler en coordonnées polaires – voir un exemple page 51.

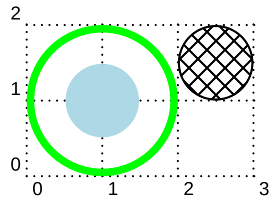
9. Il faut utiliser un environnement **pspicture** sitôt que l'on veut positionner plusieurs objets les uns par rapport aux autres dans le même espace.



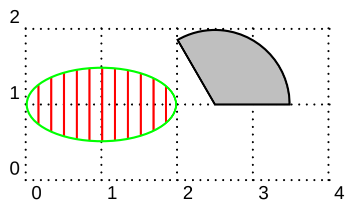
```
\psframe[fillstyle=solid,fillcolor=pink]
        (1,0)(4,1)
\psframe[fillstyle=solid,fillcolor=white,
         framearc=0.5](3,0.2)(3.8,0.8)
\pspolygon[linewidth=0.7mm,dimen=inner]
          (0,1)(2,2)(3,1)(2,0)
```



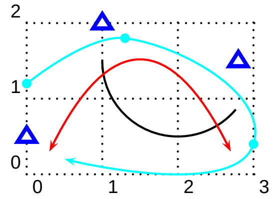
```
\psdiamond(1.5,1.5)(1.5,0.5)
\pstriangle[fillstyle=solid,
            fillcolor=yellow](2,0)(3,1)
```



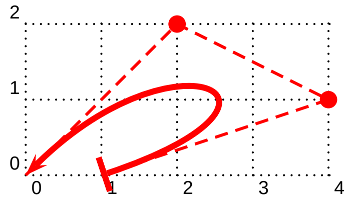
```
\pscircle[linewidth=1mm,linecolor=green]
         (1,1){1}
\pscircle[linestyle=none,fillstyle=solid,
          fillcolor=lightblue](1,1){0.5}
\pscircle[fillstyle=crosshatch](2.5,1.5)
         {0.5}
```



```
\psellipse[linecolor=green,
           fillstyle=vlines,hatchangle=0,
           hatchcolor=red](1,1)(1,0.5)
\pswedge[fillstyle=solid,
         fillcolor=lightgray]
        (2.5,1){1}{0}{120}
```



```
\psdots[linecolor=blue,dotstyle=triangle,
        dotscale=2](0,0.5)(1,2)(2.8,1.5)
\pscurve[linecolor=cyan,showpoints=true]
        {->}%
        (0,1.2)(1.3,1.8)(3,0.4)(0.5,0.2)
\psarc(2,1.5){1}{180}{320}
\parabola[linecolor=red]{<->}%
         (0.3,0.3)(1.5,1.5)
```



```
\psbezier[linewidth=0.8mm,linecolor=red,
          showpoints=true]{|->}%
         (1,0)(4,1)(2,2)(0,0)
```

## 2.3. Commandes de base sur du texte

Outre les opérations de transformations décrites au paragraphe suivant, et qui s'appliquent aussi bien à des objets graphiques qu'à du texte, certaines commandes sont spécifiquement dévolues à des opérations portant sur du texte :

Texte à encadrer

```
\psframebox{Texte à encadrer}
```

**Texte à encadrer**

```
\psframebox[fillstyle=solid,
            fillcolor=black]
           {\bf\white Texte à encadrer}
```

```
\psframebox{Les enfants}
\psframebox
  {\psframebox[linecolor=red]{ont mangé}
   \psframebox[linecolor=red]
     {\psframebox[linecolor=blue]{les pommes}
      et
      \psframebox[linecolor=blue]{les pêches}%
     }%
  }
```

Les enfants ont mangé les pommes et les pêches

Texte à encadrer

```
\psdblframebox[linecolor=green]
              {\blue Texte à encadrer}
```

Texte à encadrer

```
\psshadowbox[fillstyle=solid,
             fillcolor=lemonchiffon]
  {\begin{tabular}{c}
     \red Texte à\\\red encadrer
   \end{tabular}}
```

Texte à encadrer

```
\pscirclebox[doubleline=true]
  {\begin{tabular}{c}
     Texte à\\encadrer
   \end{tabular}}
```

Texte à encadrer

```
\psovalbox[linecolor=red]
          {\blue Texte à encadrer}
```

**Texte**

```
\psdiabox{\bf Texte}
```

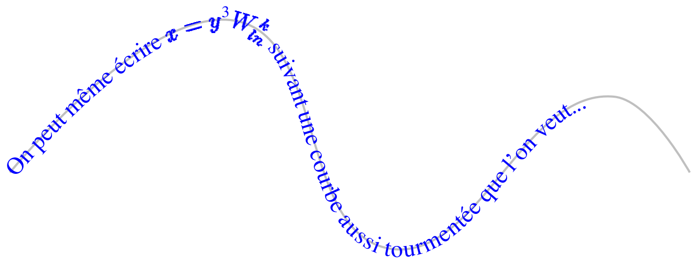

```
\pstribox[shadow=true,fillstyle=gradient,
          gradbegin=pink,gradend=red]
         {\white Texte}
```

**Texte à ombrer**

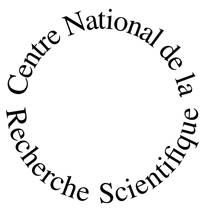
```
\psshadow{\Large\bf Texte à ombrer}
```

Il est également possible de faire écrire du texte suivant une courbe prédéfinie :

```
\psset{linecolor=lightgray}
\pstextpath
  {\pscurve(-5,-2)(-2,0)(0,-3)(3,-1)(4,-2)}
  {\blue On peut même écrire $x=y^3 W_{ln}^k$
   suivant une courbe aussi tourmentée que l'on veut...}
```

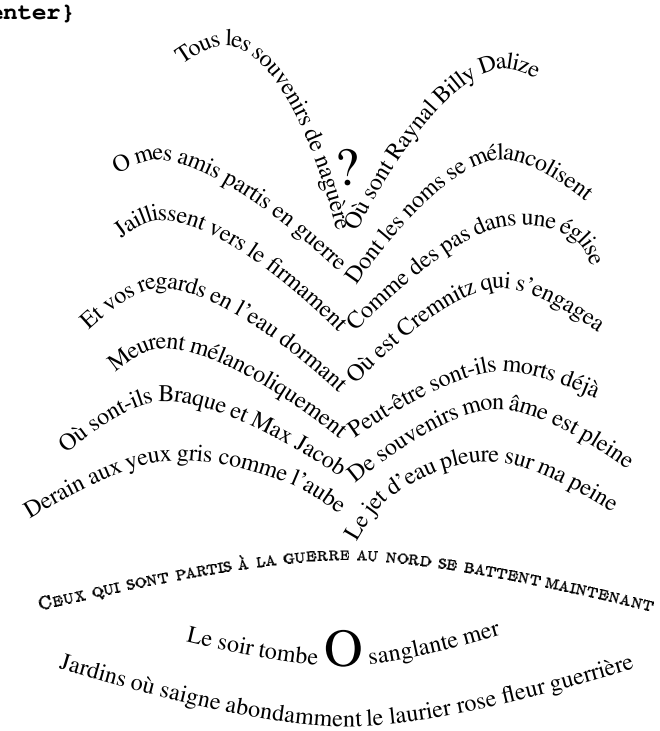



```
\psset{linestyle=none}
\pstextpath[c]{\psarcn(0,0){1.15}{180}{0}}{Centre National de la}
\pstextpath[c]{\psarc(0,0){1.15}{180}{0}}{Recherche Scientifique}
```




```
\begin{center}
  \begin{pspicture}(9,10)
    \psset{linestyle=none}
    {\small
     \pstextpath[r]{\pscurve(1.5,8)(3,9)(4.45,6.5)}
       {Tous les souvenirs de naguère}
     \pstextpath[r]{\pscurve(1,7)(2,7.5)(4.45,6)}
       {O mes amis partis en guerre}
     .............................................
     \pstextpath[r]\pscurve(0.2,2.7)(2,3.5)(3.8,3.5)
                           (4.45,2.8)}
       {Derain aux yeux gris comme l'aube}
     \rput(4.5,7.3){\bf\Huge ?}
```

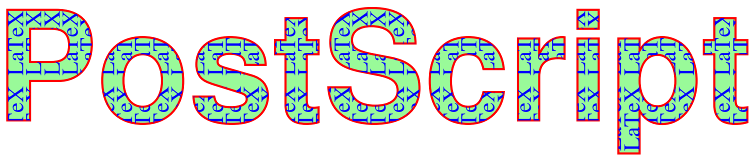
```
      \pstextpath[l]{\pscurve(4.55,6.5)(6.5,8.7)(7.8,8)}
        {Où sont Raynal Billy Dalize}
      ..........................................
      \pstextpath[c]{\pscurve(2.1,1.2)(4.5,0.8)(6.8,1.3)}
        {Le soir tombe {\bf\Huge O} sanglante mer}
      \pstextpath[c]{\pscurve(0,1)(4.5,0)(9,1)}
        {Jardins où saigne abondamment le laurier rose fleur
         guerrière}
      }
   \end{pspicture}
 \end{center}
```




Extrait d'un calligramme de Guillaume Apollinaire
*La colombe poignardée et le jet d'eau*

```
\font\bigpala=pplb at 2cm
\pscharpath[linestyle=none,fillstyle=gradient,
             gradbegin=lightblue,gradend=blue,gradmidpoint=0.5]
            {\bigpala LaTeX}
```

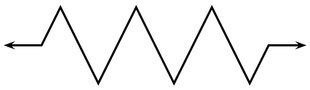
```
\font\bighelv=phvb at 2cm
\font\tinyrm=ptmr at 3mm
\newcounter{compteur}
\setcounter{compteur}{500}
\pscharclip[linecolor=red,fillstyle=solid,fillcolor=palegreen]
            {\rput(0,0){\bighelv PostScript}}
\baselineskip=1mm
\rput[t]{90}(-5,0)
  {\vbox
     {\tinyrm\blue
      \loop
        \addtocounter{compteur}{-1}
        \ifnum\value{compteur}>0
          LaTeX
      \repeat}}
\endpscharclip
```



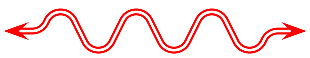

## 3. Objets complexes prédéfinis

### 3.1. Zigzags et ressorts

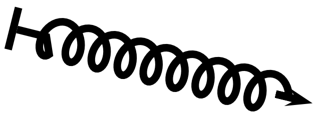
Des commandes permettent de créer directement des lignes en zigzag et des ressorts :



```
\pszigzag{<->}(4,0)
```



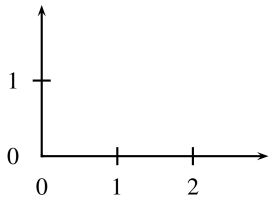
```
\pszigzag[coilarm=0.5,linearc=0.2,
          doubleline=true,linecolor=red]
         {<->}(4,0)
```



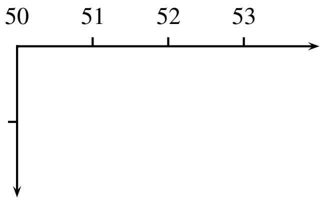
```
\pscoil[coilarm=0.5,linewidth=1mm,
        coilwidth=0.5]{|->}(4,-1)
```

## 3.2. Axes

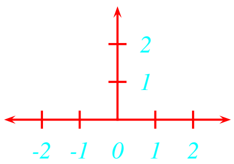
Une commande génère directement des axes, et divers paramètres permettent de les personnaliser :




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



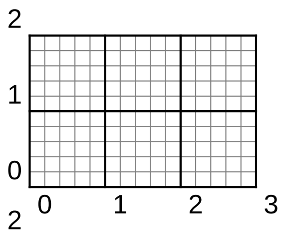

```
\psaxes[tickstyle=bottom,Ox=50,labels=x]
       {->}(4,-2)
```



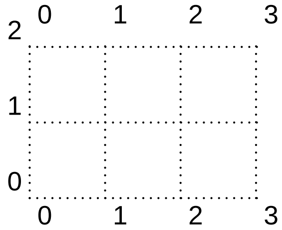

```
\renewcommand{\pshlabel}[1]{\em\cyan #1}%
\renewcommand{\psvlabel}[1]{\em\cyan #1}%
\psaxes[unit=0.5cm,linecolor=red]{<->}%
       (0,0)(3,0)(-3,3)
```

## 3.3. Grilles

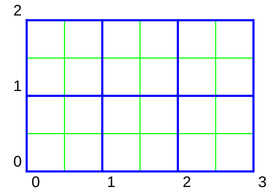
Une commande de haut niveau permet de définir des grilles (on en a en réalité déjà fait usage pour expliciter certains des exemples précédents...), qu'on peut bien sûr personnaliser de multiples façons grâce à un grand nombre de paramètres :



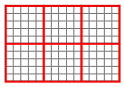

```
\psgrid(3,2)
```



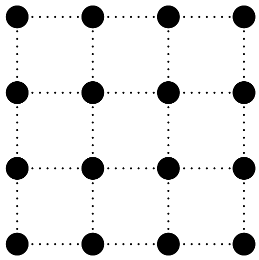

```
\psgrid[subgriddiv=0,griddots=10](3,2)
```




```
\psgrid[gridcolor=blue,subgriddiv=2,
        subgridcolor=green,gridlabels=2mm]
       (3,2)
```

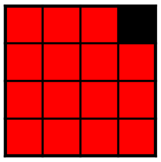


```
\psgrid[unit=0.5cm,gridcolor=red,
        gridlabels=0](3,2)
```

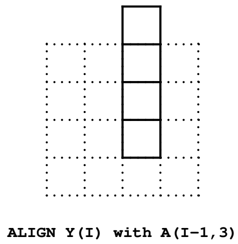


```
\psgrid[griddots=10,gridlabels=0,
        subgriddiv=1](3,3)
\psgrid[griddots=1,gridlabels=0,
        gridwidth=3mm,subgriddiv=1](3,3)
```

Les grilles peuvent être employées pour des besoins très divers (on en trouvera d'autres exemples dans les pages suivantes) :



```
\psset{unit=0.5cm}
\psframe[fillstyle=solid,fillcolor=red]
        (4,4)
\psframe[fillstyle=solid,fillcolor=black]
        (3,3)(4,4)
\psgrid[gridlabels=0,subgriddiv=0](4,4)
```




```
\psset{unit=0.5cm}
\newpsstyle{StyleGrille}
  {gridlabels=0,subgriddiv=0}
\psgrid[style=StyleGrille,griddots=5](4,4)
\psgrid[style=StyleGrille](2,1)(3,5)
\rput(2,-1){\scriptsize\tt ALIGN Y(I)
                           with A(I-1,3)}
```

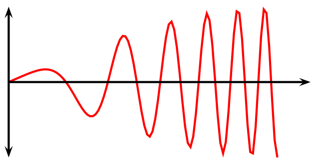
## 3.4. Tracés de courbes

Nous avons décrit un certain nombre de commandes pour tracer des courbes (**`\psdots`**, **`\psline`**, **`\pspolygon`**, **`\pscurve`**, **`\psbezier`**[10]). Mais on dispose aussi de possibilités pour représenter des courbes correspondant à des

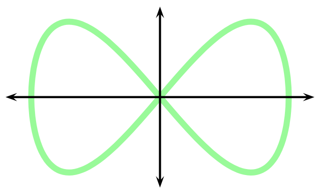
10. Il en existe quelques autres.

fonctions mathématiques, même si ce n'est là qu'une fonctionnalité accessoire et limitée (puisque les calculs sont faits par PostScript, qui n'est pas très bien adapté à cela), ainsi que de quelques ordres pour tracer des courbes à partir d'un ensemble (ou d'un fichier) de coordonnées.

```
\psplot[xunit=0.01mm,yunit=3mm,
        linecolor=orange,plotpoints=300]
       {0}{3060}{x cos}
```

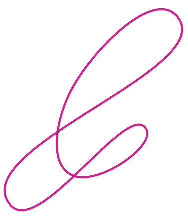


```
\psset{xunit=0.4mm}
\psplot[plotpoints=100,linecolor=red]{0}
       {89}{x sin x 2 div 2 exp cos mul}
\psline{<->}(0,-1)(0,1)
\psline{->}(100,0)
```



```
\psset{xunit=1.7cm}
\parametricplot[linecolor=palegreen,
                linewidth=0.8mm,
                plotstyle=ccurve]
               {0}{360}{t sin t 2 mul sin}
\psline{<->}(0,-1.2)(0,1.2)
\psline{<->}(-1.2,0)(1.2,0)
```

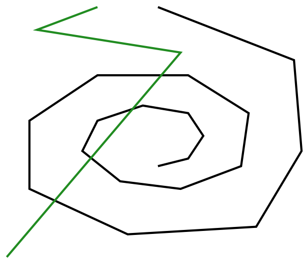
Si l'on a un ensemble de points à joindre, on peut, plutôt que d'utiliser une commande comme **`\pspolygon`**, soit les donner en paramètres de la commande **`\listplot`** qui offre des possibilités supplémentaires, soit les reprendre d'une variable où on les a préalablement stockés, soit les lire dans un fichier (notamment dans le cas où ils ont été générés par un logiciel comme Mathematica ou [gnuplot]).



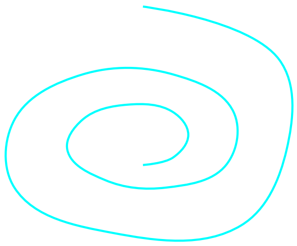
```
\listplot[linecolor=violetred,
          plotstyle=ccurve]
         {1 2 2.4 3 0.2 0.5 2 1.5 1 0.9}
```

Soit le fichier **`MaSpirale.data`** contenant :

```
[(2.1,1.4)(2.5,1.5)(2.7,1.8)(2.5,2.1)(1.9,2.2)(1.3,2)(1.1,1.6)
 (1.6,1.2)(2.4,1.1)(3.2,1.4)(3.3,2.1)(2.5,2.6)(1.3,2.6)(0.4,2)
 (0.4,1.1)(1.7,0.5)(3.4,0.6)(4,1.6)(3.9,2.8)(2.1,3.5)]
```



```
\fileplot{MaSpirale.data}
\savedata{\DonneesA}[(0.1 0.2)(2.4 2.9)%
                     (0.5 3.2)(1.3 3.5)]
\dataplot[linecolor=forestgreen]
        {\DonneesA}
```



```
\readdata{\DonneesB}{MaSpirale.data}
\dataplot[linecolor=cyan,plotstyle=curve]
        {\DonneesB}
```

## 3.5. Nœuds (pour arbres, graphes, diagrammes, organigrammes, etc.)

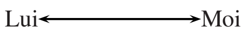
Les nœuds sont une solution générale pour relier entre eux des éléments d'information et positionner des libellés relativement à ceux-ci. Un grand nombre de macros-commandes existent, qui offrent un ensemble très complet d'outils pour réaliser aussi bien des arbres que des graphes, des organigrammes, des diagrammes mathématiques ou syntaxiques, etc. Là aussi, l'éventail des possibilités est considérable, et nous nous contenterons ici d'illustrer les aspects essentiels de cet ensemble de commandes.

! Mais en fait leur usage peut être très diversifié !

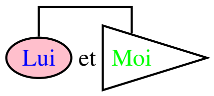
Ces domaines d'application étant d'usage si courant et si essentiel, il existe un grand nombre de développements hétérogènes pour augmenter les capacités de (LA)TEX dans ces domaines, et en faciliter l'utilisation. **PSTricks**, s'il n'a pas sur tous ces aspects des fonctionnalités équivalentes à certaines de ces extensions [11]

11. **AlDraTEX** [Gurari 94], d'apparition récente, veut également offrir des solutions unifiées sur ces aspects.
Pour les arbres, il existe un grand nombre de solutions. Les plus riches et sophistiquées sont [**TreeTEX**], [**tree.sty**] — limitées aux arbres binaires —, [**trees** et **trees.sty**] et [**tree**]. Cette dernière est une solution *Plain* TEX extrêmement puissante, conçue par un linguiste pour réaliser des diagrammes syntaxiques linguistiques très complexes, via un pré-processeur traitant une représentation abstraite des arbres. Les capacités de **PSTricks** sont considérables dans ce domaine, sa principale limitation étant en fait la relative lourdeur de la programmation pour la réalisation d'arbres complexes, en l'absence actuelle de pré-processeur.
Pour les diagrammes mathématiques, là aussi un grand nombre d'extensions ont été réalisées. Les principales sont [**catmac.sty**], [**diagram**], [**diagrams.tex**] et [**XYpic**] — un ensemble plus complet de références se trouve dans la documentation de [**diagrams.tex**]. Notons aussi qu'un groupe de travail du projet LATEX3 est consacré à ce thème.

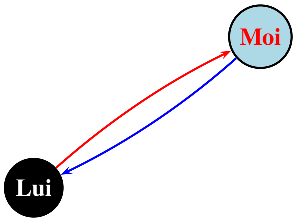
— dans quelques cas parce que ceux-ci possèdent des macros spécifiques de très haut niveau que n'a pas aujourd'hui **PSTricks**, dans d'autres parce que leur puissance provient de ce qu'ils implémentent des pré-processeurs [12] —, a le triple avantage d'offrir un environnement intégré, d'apporter toute la richesse de PostScript (ce qui n'est quasiment jamais offert par ces extensions particulières), et d'avoir la potentialité de construire des fonctions encore plus évoluées, soit par le biais de nouvelles macros-commandes, soit par celui de pré-processeurs spécialisés.



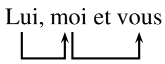

```
\rnode{A}{Lui} \hspace{2cm} \rnode{B}{Moi}
\ncline{<->}{A}{B}
```




```
\ovalnode[fillstyle=solid,fillcolor=pink]
         {A}{\blue Lui}
et \trinode[trimode=R]{B}{\green Moi}
\ncbar[angle=90]{A}{B}
```

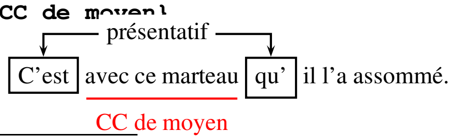



```
\cnodeput[fillstyle=solid,fillcolor=black]
         (0,0){A}{\bf\white Lui}
\cnodeput[fillstyle=solid,
          fillcolor=lightblue]
         (3,2){B}{\bf\red Moi}
\ncarc[linecolor=red]{->}{A}{B}
\ncarc[linecolor=blue]{->}{B}{A}
```



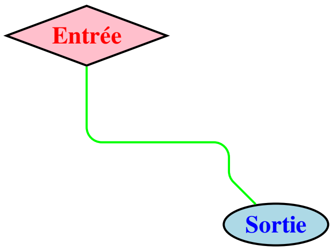

```
\rnode{A}{Lui}, \rnode{B}{moi} et
  \rnode{C}{vous}
\ncbar[offsetB=0.5mm,angleA=-90,
       nodesep=1mm]{->}{A}{B}
\ncbar[offsetA=0.5mm,angleA=-90,
       nodesep=1mm]{->}{B}{C}
```

```
\Rnode{A}{\psframebox{C'est}} avec ce marteau
\Rnode{B}{\psframebox{qu'}} il l'a assommé.
\ncbar[angle=90,arm=3mm]{<->}{A}{B}
\ncput*{présentatif}
\ncline[linecolor=red,offset=-3mm,arm=0,nodesep=1mm]{A}{B}
\nbput{\red CC de moyen}
```

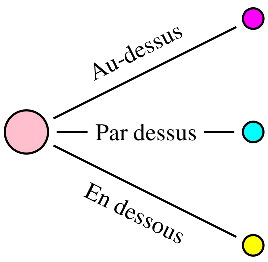



Pour les diagrammes syntaxiques au sens strict (appelés par les anglo-saxons *railroad diagrams*), existe [**rail.sty**].
Pour les organigrammes, on dispose principalement de [**flow**] et de [**Flow.sty**].
Pour les graphes, à notre connaissance n'existe guère de spécifique que [**Graph-TEX**], d'apparition récente, qui est un pré-processeur écrit en **Perl** générant des ordres [**TEXdraw**].

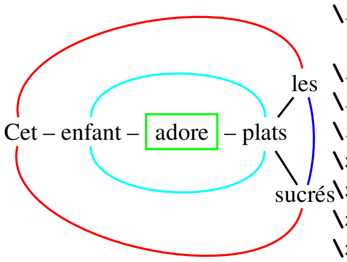
12. C'est le cas notamment de [**flow**] pour les organigrammes, qui permet par son macro-langage spécialisé de traiter des cas complexes, alors que la programmation avec les ordres **PSTricks** de base devient vite dissuasive, comme on peut s'en rendre compte dans l'exemple donné ci-après, et de [**tree**] pour les arbres. Mais il serait parfaitement envisageable d'utiliser ces systèmes de description abstraite de haut niveau — ou d'autres, comme celui utilisé par l'intéressant logiciel de représentation d'arbres [*daVinci*] — pour générer des ordres **PSTricks**.



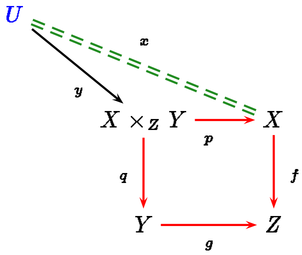

```
\rput(0,2.5)
  {\dianode[fillstyle=solid,
            fillcolor=pink]
           {A}{\bf\red Entrée}}
\rput(2.5,0)
  {\ovalnode[fillstyle=solid,
             fillcolor=lightblue]
            {B}{\bf\blue Sortie}}
\ncangles[linecolor=green,
          angleA=-90,angleB=135,
          armA=1cm,armB=0.5cm,linearc=0.2]
         {A}{B}
```



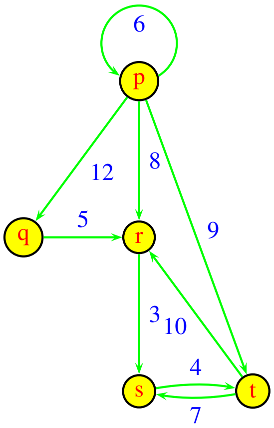

```
\cnode[fillstyle=solid,fillcolor=pink]
      (0,0){3mm}{Racine}
\cnode[fillstyle=solid,fillcolor=magenta]
      (3,1.5){1.5mm}{NoeudA}
\cnode[fillstyle=solid,fillcolor=cyan]
      (3,0){1.5mm}{NoeudB}
\cnode[fillstyle=solid,fillcolor=yellow]
      (3,-1.5){1.5mm}{NoeudC}
\psset{nodesep=1mm,nrot=:U}
\ncline{Racine}{NoeudA}\naput{Au-dessus}
\ncline{Racine}{NoeudB}\ncput*{Par dessus}
\ncline{Racine}{NoeudC}\nbput{En dessous}
```




```
\psset{unit=0.5cm,nodesep=0.8mm}
\rnode{A}{Cet} --
\rnode{B}{enfant} --
\rnode{C}{\psframebox[linecolor=green]
                     {adore}} --
\rnode{D}{plats}
\rput(0.3,1.5){\rnode{E}{les}}
\rput(0.3,-1.5){\rnode{F}{sucrés}}
\ncarc[arcangle=90,linecolor=red]{A}{E}
\ncarc[arcangle=-90,linecolor=red]{A}{F}
\ncarc[arcangle=90,linecolor=cyan]{B}{D}
\ncarc[arcangle=-90,linecolor=cyan]{B}{D}
\ncarc[arcangle=15,linecolor=blue]{E}{F}
\ncline{D}{E}
\ncline{D}{F}
```



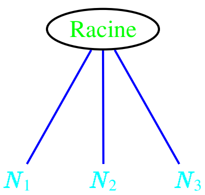

```
$
 \begin{psmatrix}[colsep=1cm,rowsep=1cm]
   \blue U \\
           & X\times_Z Y & X \\
           & Y           & Z
   \psset{arrows=->,nodesep=1mm}
   \everypsbox{\scriptstyle}
   \ncline{1,1}{2,2}_{y}
   \ncline[linecolor=forestgreen,
           linestyle=dashed,
           doubleline=true]
          {-}{1,1}{2,3}^{x}
   \ncline[linecolor=red]{2,2}{3,2}<{q}
   \ncline[linecolor=red]{2,2}{2,3}_{p}
   \ncline[linecolor=red]{2,3}{3,3}>{f}
   \ncline[linecolor=red]{3,2}{3,3}_{g}
 \end{psmatrix}
$
```



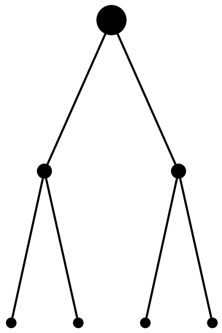

```
\begin{psmatrix}[fillstyle=solid,
                 fillcolor=yellow,
                 mnode=circle,colsep=1]
         & \red p \\
  \red q & \red r \\
         & \red s & \red t
\end{psmatrix}
\psset{linecolor=green,arrows=->,
       labelsep=1mm,shortput=nab}
\blue
\nccircle{1,2}{0.5cm}^{6}
\ncline{1,2}{2,1}^{12}
\ncline{1,2}{2,2}^{8}
\ncline{1,2}{3,3}^{9}
\ncline{2,1}{2,2}^{5}
\ncline{2,2}{3,2}^{3}
\ncline{3,3}{2,2}^{10}
\ncarc[arcangle=10]{3,3}{3,2}^{7}
\ncarc[arcangle=10]{3,2}{3,3}^{4}
```

## 3.6. Arbres

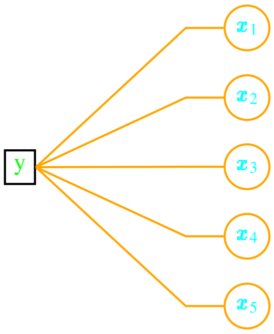
De nombreuses macros-commandes sont spécialement dévolues à la représentation d'arbres, offrant un ensemble très puissant de fonctionnalités.



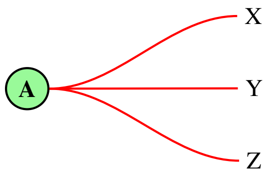

```
\pstree[linecolor=blue,nodesepB=1mm]
       {\Toval{\green Racine}}
       {\TR{\cyan $N_1$}
        \TR{\cyan $N_2$}
        \TR{\cyan $N_3$}}
```



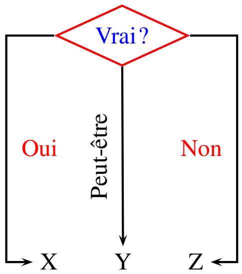
```
\pstree{\Tc*{2mm}}
        {\pstree{\Tc*{1mm}}{\TC* \TC*}
         \pstree{\Tc*{1mm}}{\TC* \TC*}}
```




```
\renewcommand{\psedge}
  {\ncdiag[armA=0,angleB=180,armB=0.5cm]}
\pstree[linecolor=orange,treemode=R,
         treesep=3mm,levelsep=3cm]
        {\Tr{\psframebox{\green y}}}
        {\cyan
         \Tcircle{$x_{1}$}
         \Tcircle{$x_{2}$}
         \Tcircle{$x_{3}$}
         \Tcircle{$x_{4}$}
         \Tcircle{$x_{5}$}}
```

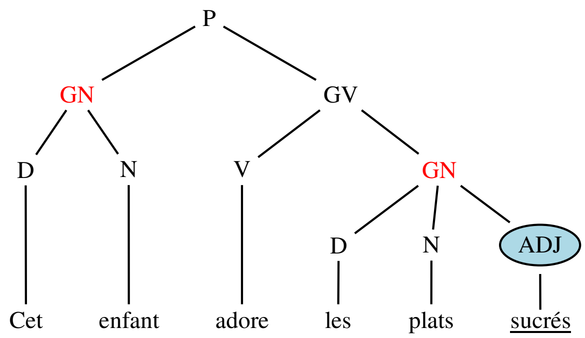



```
\renewcommand{\psedge}
  {\nccurve[angleB=180,nodesepB=1mm]}
\pstree[linecolor=red,treemode=R,
         levelsep=3cm]
        {\Toval[fillstyle=solid,
                 fillcolor=palegreen]
                {\bf A}}
        {\Tr{X}
         \Tr{Y}
         \Tr{Z}}
```




```
\pstree[arrows=->,levelsep=3cm,
         nodesepB=1mm]
        {\Tdia[linecolor=red]
               {\blue Vrai ?}}
        {\TR[edge={\ncbar[angle=180]}]{X}
            \trput{\red Oui}
         \TR{Y} \nbput[nrot=90]{Peut-être}
         \TR[edge=\ncbar]{Z}
            \tlput{\red Non}}
```

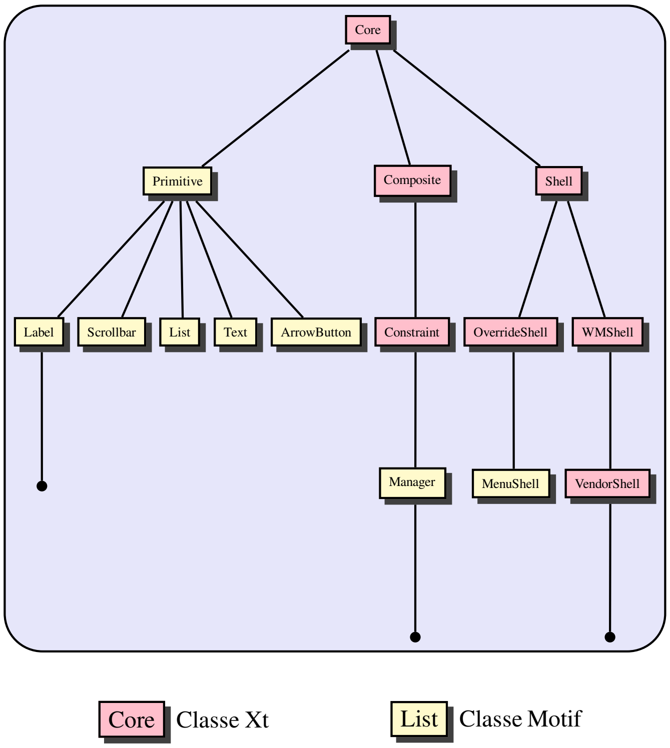
```
\psset{levelsep=1cm,nodesep=1mm}
\pstree{\Tr{P}}
        {\pstree{\Tr{\red GN}}
                 {\pstree{\Tr{D}}
                          {\pstree{\Tn}{\TR{Cet}}}
                  \pstree{\Tr{N}}
                          {\pstree{\Tn}{\TR{enfant}}}}
         \pstree{\Tr{GV}}
                 {\pstree{\Tr{V}}
                          {\pstree{\Tn}{\TR{adore}}}
                  \pstree{\Tr{\red GN}}
                          {\pstree{\Tr{D}}{\TR{les}}
                           \pstree{\Tr{N}}{\TR{plats}}
                           \pstree{\Toval[fillstyle=solid,
                                          fillcolor=lightblue]
                                         {ADJ}}
                                  {\TR{\underline{sucrés}}}}}}
```




```
\psset{treesep=1mm}
\newcommand{\MonNoeud}[2]
  {\Tr{\psshadowbox[fillstyle=solid,fillcolor=#1]{\tiny #2}}}
\newcommand{\NoeudXt}[1]{\MonNoeud{pink}{#1}}
\newcommand{\NoeudMotif}[1]{\MonNoeud{lemonchiffon}{#1}}
\rput(0,0){\Large\blue Ensemble des classes de {\em widgets}
                        Motif}
\rput(0,-4.8)
  {\psframebox[fillstyle=solid,fillcolor=lavender,
                linearc=0.5cm,cornersize=absolute]
     {\pstree{\NoeudXt{Core}}
              {\pstree{\NoeudMotif{Primitive}}
                       {\pstree{\NoeudMotif{Label}}
                                {\TC*}
                        \NoeudMotif{Scrollbar}
                        \NoeudMotif{List}
                        \NoeudMotif{Text}
                        \NoeudMotif{ArrowButton}}
```

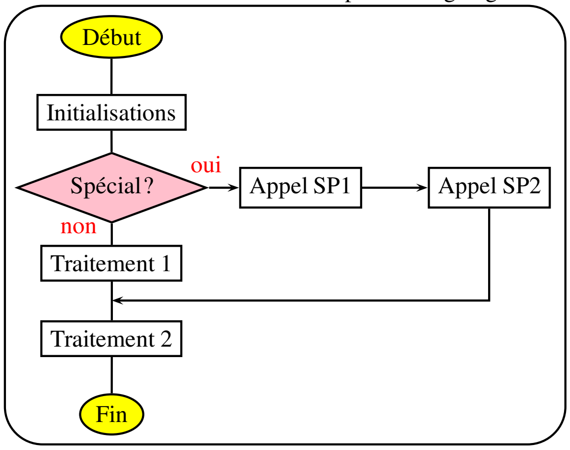
```
\pstree{\NoeudXt{Composite}}
       {\pstree{\NoeudXt{Constraint}}
               {\pstree{\NoeudMotif{Manager}}
                       {\TC*}}}
\pstree{\NoeudXt{Shell}}
       {\pstree{\NoeudXt{OverrideShell}}
               {\NoeudMotif{MenuShell}}
        \pstree{\NoeudXt{WMShell}}
               {\pstree{\NoeudXt{VendorShell}}
                       {\TC*}}}}}}
\rput(-2,-10)
  {\psshadowbox[fillstyle=solid,fillcolor=pink]{Core}
   Classe Xt}
\rput(2,-10)
  {\psshadowbox[fillstyle=solid,fillcolor=lemonchiffon]{List}
   Classe Motif}
```

Ensemble des classes de *widgets* Motif



```
\centerline{Utilisation des arbres et des n\oe uds pour un
            organigramme}
\newcommand{\Tboite}[1]{\Tr{\psframebox{#1}}}
\psset{levelsep=1cm}
\psframebox[linearc=0.5cm,cornersize=absolute]
  {\pstree{\Toval[fillstyle=solid,fillcolor=yellow]{Début}}
     {\pstree{\Tboite{Initialisations}}
        {\pstree{\pstree[arrows=->,levelsep=2.5cm,treemode=R]
                        {\Tdia[fillstyle=solid,
                               fillcolor=pink]{Spécial ?}}
                        {\pstree{\Tboite{Appel SP1}
                                 \taput{\red oui}}
                                {\Tr{\Rnode{A}
                                       {\psframebox
                                          {Appel SP2}}}}}}
                {\pstree{\Tr{\Rnode{B}{\psframebox
                                         {Traitement 1}}}
                                       \tlput{\red non}}
                        {\pstree{\Tboite{Traitement 2}}
                                {\Toval[fillstyle=solid,
                                        fillcolor=yellow]
                                       {Fin}}}}}}
  \ncbar[angleA=-90,armB=0,nodesepB=0.25cm]{->}{A}{B}}
```

Utilisation des arbres et des nœuds pour un organigramme



### 3.7. Superpositions

Même si l'intérêt essentiel en est dans la réalisation de transparents, et que dans ce cas des commandes particulières existent (voir plus loin l'article de Michel GOOSSENS et Sebastian RAHTZ), il est directement possible de générer des *blocs* d'informations automatiquement formatés pour de futures superpositions :

```
\begin{overlaybox}
  \psoverlay{all}
  \psframebox[framearc=0.2,linewidth=0.6mm]
    {\parbox{3.2cm}
       {\footnotesize
        \psoverlay{Resultats}
        Première :
          {\psoverlay{Premiere}
             {\hspace*{\fill}325 points\\
              \hspace*{\fill}\red Sophie {\sc Stiquet}}} \\
        Deuxième :
          {\psoverlay{Deuxieme}
             {\hspace*{\fill}283 points\\
              \hspace*{\fill}\green Sébastienne
              {\sc Toutseuls}}}
       }
    }
\end{overlaybox}
\putoverlaybox{Resultats}  \putoverlaybox{Premiere}
                           \putoverlaybox{Deuxieme}
```

| | | |
|---|---|---|
| Première :<br><br>Deuxième : | 325 points<br>Sophie STIQUET | <br><br>283 points<br>Sébastienne TOUTSEULS |

## 4. Transformation des objets

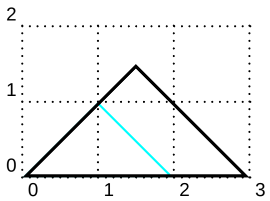
On peut appliquer des transformations diverses aux objets (graphiques ou textuels, simples ou composés) : changements d'échelle, rotations [13], déformations, projections dans l'espace virtuel à 3 dimensions. Et cela multiplie les possibilités...

Texte étiré
Idem

```
\begin{tabular}{c}
  \scaleboxto(3,1){Texte étiré} \\
  \scaleboxto(3,1){Idem}
\end{tabular}
```

13. Les changements d'échelle et les rotations de caractères sont réalisées beaucoup plus fidèlement avec des fontes PostScript résidantes.

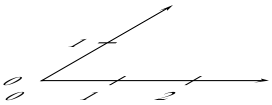

```
\begin{tabular}{c}
  \scalebox{2}{Taille imposée} \\
  \scalebox{0.8}{Taille imposée} \\
  \scalebox{1 4}{Idem} \\
  \scalebox{-1 2}{Miroir} \\
  \scalebox{1 -2}{Miroir}
\end{tabular}
```



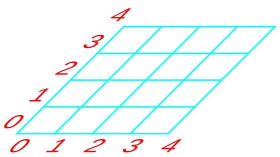

```
\pstriangle[linecolor=cyan](1,0)(2,1)
\scalebox{1.5}{\pstriangle(1,0)(2,1)}
```

```
\pstilt{45}{Mot}
\pstilt{-45}{\blue Mot}
\psTilt{45}{\red Mot}
\psTilt{140}{\green Mot}
```

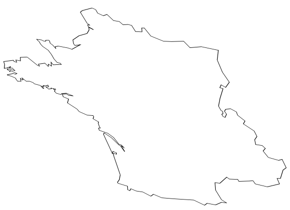

```
\pstilt{45}{\psshadowbox{\Huge\red Texte}}
```



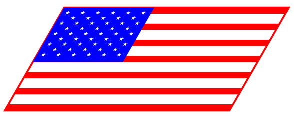
```
\pstilt{30}{\psaxes{->}(3,2)}
```



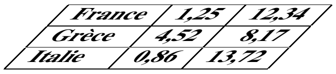

```
\pstilt{45}
  {\psgrid[unit=0.5cm,
           subgriddiv=0,gridcolor=cyan,
           gridlabelcolor=red](4,4)}
```



```
\epsfxsize=3cm
\pstilt{120}{\epsffile{france.eps}}
```

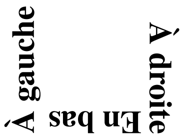


```
\psset{unit=2mm}
\pstilt{60}{\DrapeauAmericain}
```



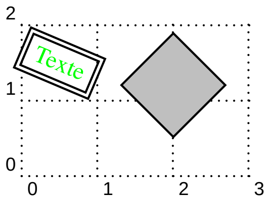

```
\pstilt{45}
  {\bf
   \begin{tabular}{|l|r|r|}
     \hline
     France & 1,25 & 12,34 \\ \hline
     Grèce  & 4,52 &  8,17 \\ \hline
     Italie & 0,86 & 13,72 \\ \hline
   \end{tabular}}
```

En ce qui concerne les rotations, elles peuvent être d'angle fixe, mais on peut également placer arbitrairement les objets, ce qui permet de faire toutes les annotations que l'on peut désirer sur ceux-ci :

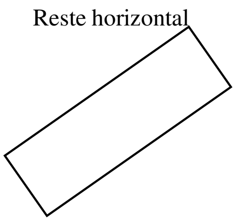



```
{\Large\bf \rotateleft{À gauche}
           \rotatedown{En bas}
           \rotateright{À droite}}
```

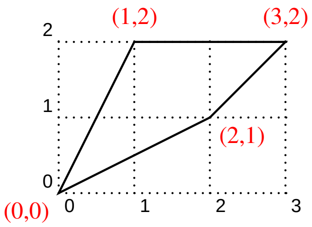

```
\rotateleft{\psshadowbox
              {\red Texte retourné}}
```



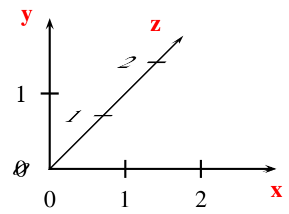

```
\rput{-23}(0.5,1.5)
  {\psdblframebox{\green Texte}}
\rput{45}(2,0.5)
  {\psframe[fillstyle=solid,
            fillcolor=lightgray](1,1)}
```



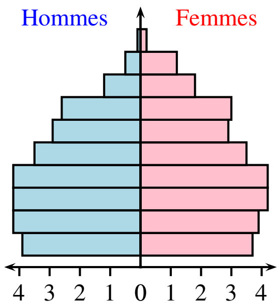

```
\rput{35}
  {\psframe(-1,0)(2,1)
     \rput[br]{*0}(2,1){Reste horizontal}}
```




```
\pspolygon(0,0)(2,1)(3,2)(1,2)
\uput[dl](0,0){\red (0,0)}
\uput[dr](2,1){\red (2,1)}
\uput[u](3,2){\red (3,2)}
\uput[u](1,2){\red (1,2)}
```




```
\psaxes{->}(3,2)
\pstilt{45}
  {\psaxes[ticks=y,labels=y]{->}(3,2.5)}
\rput(3,-0.3){\bf\red x}
\rput(-0.3,2){\bf\red y}
\rput(1.4,1.9){\bf\red z}
```

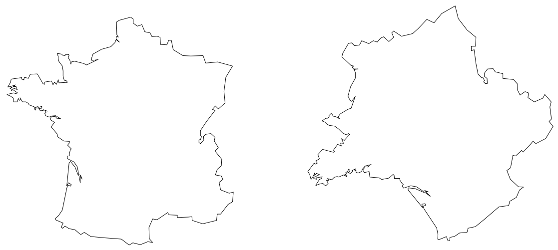



Population française (1990)
Source INSEE — En millions

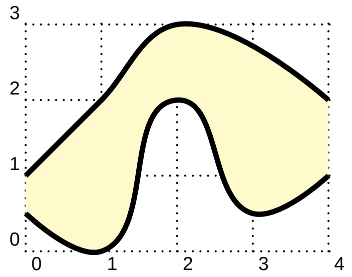
```
\newcommand{\PyramideAgesFemmes}
 {\psframe(0,0)(3.7,1)\psframe(0,1)(3.9,2)
  \psframe(0,2)(4.2,3)\psframe(0,3)(4.2,4)
  \psframe(0,4)(3.5,5)\psframe(0,5)(2.9,6)
  \psframe(0,6)(3,7)   \psframe(0,7)(1.8,8)
  \psframe(0,8)(1.2,9)\psframe(0,9)(.2,10)}
\newcommand{\PyramideAgesHommes}
 {\psframe(0,0)(3.9,1)\psframe(0,1)(4.2,2)
  \psframe(0,2)(4.2,3)\psframe(0,3)(4.2,4)
  \psframe(0,4)(3.5,5)\psframe(0,5)(2.9,6)
  \psframe(0,6)(2.6,7)\psframe(0,7)(1.2,8)
  \psframe(0,8)(0.5,9)\psframe(0,9)(.1,10)}
\psset{xunit=4mm,yunit=3mm,
       dimen=middle,labelsep=1mm,
       fillstyle=solid,fillcolor=pink}
\rput(2.5,0){\PyramideAgesFemmes}
\rput(5,10.5){\small\red Femmes}
\psset{fillcolor=lightblue}
\rput(2.5,0)
  {\scalebox{-1 1}{\PyramideAgesHommes}}
\rput(0,10.5){\small\blue Hommes}
\makeatletter
\renewcommand{\pshlabel}[1]
  {\@tempcnta=#1\relax
   \ifnum\@tempcnta<0
     \@tempcnta=-\@tempcnta
   \fi
   \the\@tempcnta}
\makeatother
\rput(2.5,-0.5)
  {\psaxes[tickstyle=bottom,ticks=x,
           labels=x]{<->}(4.5,11.5)
   \psaxes[tickstyle=bottom,ticks=x,
           labels=x,Dx=1]{->}(-4.5,11.5)}
\rput(2.5,-3){Population française (1990)}
\rput(2.5,-4){\scriptsize Source INSEE
              --- En millions}
```

On peut aussi appliquer une rotation à tout un fichier PostScript :

```
\epsfxsize=3cm
\rput(0,-1.5){\epsffile{france.eps}}
\epsfxsize=3cm
\rput{45}(4,-1.5){\epsffile{france.eps}}
```



Il est également possible, bien que cela soit un peu plus complexe à programmer, de remplir l'espace défini par exemple entre deux courbes :



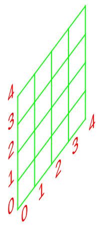

```
\pscustom[linewidth=0.7mm]
  {\pscurve(0,1)(1,2)(2,3)(4,2)
   \gsave
     \pscurve[liftpen=1]
             (4,1)(3,0.5)(2,2)(1,0)(0,0.5)
     \fill[fillstyle=solid,
           fillcolor=lemonchiffon]
   \grestore}
\pscurve[linewidth=0.7mm]
        (4,1)(3,0.5)(2,2)(1,0)(0,0.5)
```

D'autre part, il existe une commande très puissante pour projeter un objet dans l'espace virtuel à trois dimensions. Là aussi, cela ouvre un éventail impressionnant de possibilités...

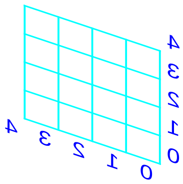

```
\psset{viewpoint=0.5 -1 0.5}
\ThreeDput[normal=0 0 1]{\Huge\blue Texte}
```

```
\psset{viewpoint=-0.5 -1 1}
\ThreeDput[normal=1 0 0]
  {\psshadow{\Huge Texte}}
```

```
\psset{viewpoint=1 -1 1}
\ThreeDput[normal=0 1 0]
  {\psshadowbox{\Huge\red Texte}}
```

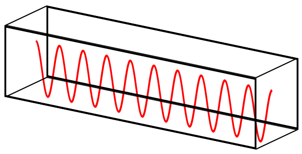


```
\psset{viewpoint=0.5 -1 1}
\ThreeDput[normal=1 0 0]
  {\psgrid[unit=0.5cm,subgriddiv=0,
           gridcolor=green,
           gridlabelcolor=red](4,4)}
```

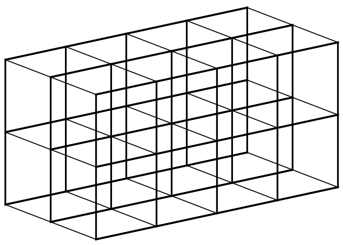


```
\psset{viewpoint=0.5 -1 1}
\ThreeDput[normal=0 1 0]
  {\psgrid[unit=0.5cm,subgriddiv=0,
           gridcolor=cyan,
           gridlabelcolor=blue](4,4)}
```

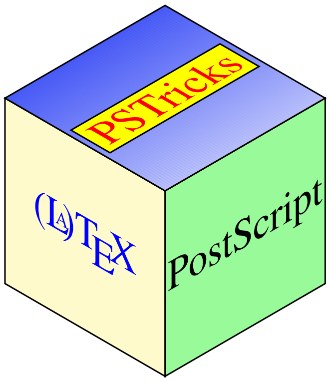
```
\epsfxsize=3cm
\psset{viewpoint=0 -5 1}
\ThreeDput[normal=0 0 1](0,0,3)
  {\epsffile{france.eps}}
```



```
\psset{viewpoint=0.75 0.5 0.3}
\ThreeDput[normal=1 0 0](1,0.5,0.5)
  {\psCoil[coilaspect=0,coilwidth=0.75,
           coilheight=0.5,linecolor=red]
          {0}{3600}}
\ThreeDput[normal=1 0 0]{\psframe(4,1)}
\ThreeDput[normal=0 0 1]{\psframe(1,4)}
\ThreeDput[normal=1 0 0](1,0,0)
                         {\psframe(4,1)}
\ThreeDput[normal=0 0 1](0,0,1)
                         {\psframe(1,4)}
```



```
\psset{viewpoint=4 -3 1.5}
\newpsobject{MaGrille}{psgrid}
  {subgriddiv=0,gridlabels=0}
\ThreeDput[normal=1 0 0]{\MaGrille(4,2)}
\ThreeDput[normal=1 0 0](1,0,0)
                         {\MaGrille(4,2)}
\ThreeDput[normal=1 0 0](2,0,0)
                         {\MaGrille(4,2)}
\ThreeDput[normal=0 0 1]{\MaGrille(2,4)}
\ThreeDput[normal=0 0 1](0,0,1)
                         {\MaGrille(2,4)}
\ThreeDput[normal=0 0 1](0,0,2)
          {\MaGrille(2,4)}
```




```
\psset{dimen=middle,viewpoint=-1 -1 1}
\ThreeDput[normal=0 -1 0]
  {\psframe[fillstyle=solid,
            fillcolor=palegreen](3,3)
   \rput(1.5,1.5){\Huge PostScript}}
\ThreeDput[normal=-1 0 0](0,3,0)
  {\psframe[fillstyle=solid,
            fillcolor=lemonchiffon](3,3)
   \rput(1.5,1.5){\Huge\blue \AllTeX}}
\ThreeDput[normal=0 0 1](0,0,3)
  {\psframe[fillstyle=gradient,
            gradmidpoint=1,gradend=white]
           (3,3)
   \rput(1.5,1.5)
     {\psframebox[fillstyle=solid,
                  fillcolor=yellow]
                 {\Huge\red PSTricks}}}
```

## 5. Répétition d'actions

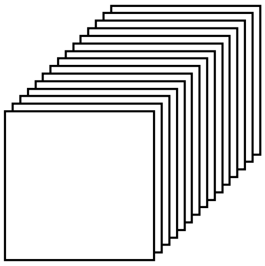
Des macros-commandes très puissantes existent pour répéter des actions de construction d'objets. Elles sont essentielles pour constituer des figures complexes, puisque, par de tels assemblages, on peut ainsi définir des objets composites riches, qui peuvent ensuite être manipulés avec les différents opérateurs vus précédemment. Cela ouvre donc des possibilités proprement infinies ! C'est pourquoi nous donnons ici une large panoplie d'exemples.

Mot Mot Mot Mot Mot

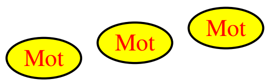
```
\multido{}{5}{Mot }
```

LettreA LettreB LettreC

```
\multido{\i=65+1}{3}
  {Lettre{\protect\red\char\i} }
```

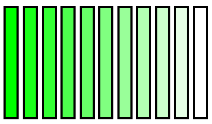


```
\multirput(-0.1,-0.1){15}
  {\psframe[fillstyle=solid,
            fillcolor=white](2,2)}
```

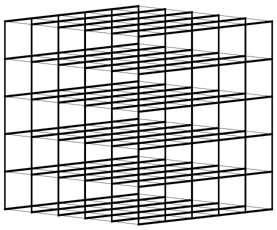



```
\multirput(1.2,0.2){3}
  {\psovalbox[fillstyle=solid,
              fillcolor=yellow]{\red Mot}}
```

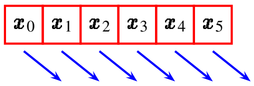
```
\psset{unit=2mm}
\newcommand{\MonZigzag}
  {\psline[linecolor=red]
           (0,0)(.5,1)(1.5,-1)(2,0)}%
\multips(2,0){8}{\MonZigzag}
```



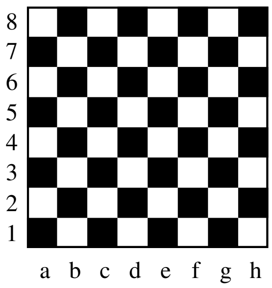
```
\psset{xunit=2.5cm,yunit=15cm}
\multido{\n=0+.1}{11}
  {\newrgbcolor{MaCouleur}{\n\space 1. \n}
   \rput(\n,0.1)
     {\psframe[fillstyle=solid,
               fillcolor=MaCouleur]
              (0.08,0.1)}}
```



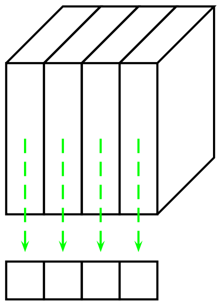
```
\psset{unit=0.5cm,viewpoint=4 -4 0.65}
\newpsobject{MaGrille}{psgrid}
  {subgriddiv=0,gridlabels=0}
\multido{\i=0+1}{6}
  {\ThreeDput[normal=1 0 0]
     (\i,0,0){\MaGrille(5,5)}
   \ThreeDput[normal=0 0 1]
     (0,0,\i){\MaGrille(5,5)}}
```




```
\psset{unit=0.5cm}
\multips(1,0){6}
  {\psframe[dimen=middle,linecolor=red]
            (1,1)
   \psline[linecolor=blue]{->}%
           (0.5,-.2)(1.5,-1)}
\multido{\i=0+1}{6}
  {\uput{-0.2}[0](\i,0.5){$x_{\i}$}}
```




```
\psset{unit=4mm}
\newcounter{lettre}
\psframe(8,8)
\multido{\i=0+2}{4}
  {\multips(0,\i)(2,0){4}
     {\psframe*(1,1)
      \psframe*(1,1)(2,2)}}
\multido{\i=1+1}{8}
  {\rput(-0.5,-0.5)
     {\rput[B](\i,-0.5)
        {\setcounter{lettre}{\i}
         \alph{lettre}}
      \rput(0,\i){\i}}}
```



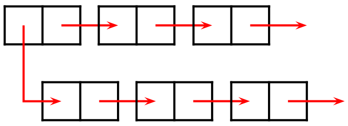
ALIGN X(I,J,K) with C(J)

```
\psset{unit=0.5cm,dimen=middle}
\multips(1,0){4}
  {\psframe(1,4)
   \psline(0,4)(1.5,5.5)(2.5,5.5)(1,4)
   \psline[linecolor=green,
           linestyle=dashed]{->}%
          (0.5,2)(0.5,-1)
   \psframe(0,-1.25)(1,-2.25)}
 \psline(4,0)(5.5,1.5)(5.5,5.5)
 \rput(2.75,-3)
   {\scriptsize\tt ALIGN X(I,J,K)
                   with C(J)}
```

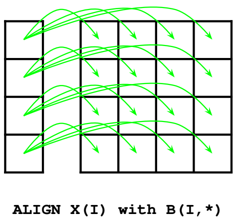
Novembre 1993

| Lun | Mar | Mer | Jeu | Ven | Sam | Dim |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| 29 | 30 | | | | | |

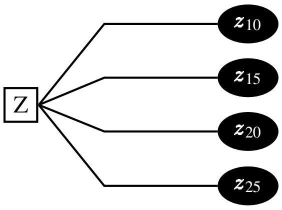
```
\psset{unit=0.5cm}
\newpsstyle{MonStyle}
  {linestyle=none,fillstyle=solid}
\rput(3.5,7){\large Novembre 1993}
\rput(0.5,5.5){\tiny Lun}
\rput(1.5,5.5){\tiny Mar}
\rput(2.5,5.5){\tiny Mer}
\rput(3.5,5.5){\tiny Jeu}
\rput(4.5,5.5){\tiny Ven}
\rput(5.5,5.5){\tiny Sam}
\rput(6.5,5.5){\tiny Dim}
\pspolygon[style=MonStyle,
           fillcolor=lightgray]
          (0,0)(0,5)(5,5)(5,1)(2,1)(2,0)
\psframe[style=MonStyle,fillcolor=yellow]
        (5,1)(7,5)  % samedis et dimanches
\psframe[style=MonStyle,fillcolor=pink]
        (0,4)(1,5)  % 01/11
\psframe[style=MonStyle,fillcolor=pink]
        (3,3)(4,4)  % 11/11
\psgrid[subgriddiv=0,gridlabels=0](7,6)
\newcounter{ligne}
\newcounter{colonne}
\setcounter{ligne}{5}
\setcounter{colonne}{1}
\rput(-0.4,-0.5)
  {\multido{\i=1+1}{30}
     {\rput(\value{colonne},\value{ligne})
           {\i}
      \addtocounter{colonne}{1}
      \ifnum\value{colonne}=8
        \setcounter{colonne}{1}
        \addtocounter{ligne}{-1}
      \fi}}
```



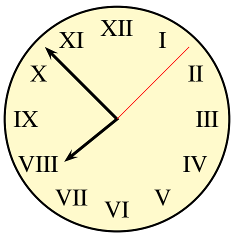
```
\psset{unit=0.5cm}
\multirput(1,-2){2}
  {\multirput(2.5,0){3}
     {\psgrid[subgriddiv=0,gridlabels=0]
              (2,1)
      \psline[linecolor=red]{->}%
              (1.5,0.5)(3,0.5)}}
\psline[linecolor=red]{->}%
        (0.5,0.5)(0.5,-1.5)(1.5,-1.5)
```



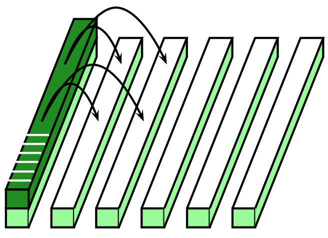

```
\psset{unit=0.5cm}
\newpsobject{MaGrille}{psgrid}
  {gridlabels=0,subgriddiv=0}
\MaGrille(1,4)
\MaGrille(2,0)(6,4)
\newcounter{val}
\multido{\ia=0+1}{4}
  {\multido{\ib=2+1}{4}
     {\FPadd{\ib}{-1}{\val}
      \rput(0.5,\ia)
        {\pscurve[linecolor=green,
                    linewidth=0.15mm]{->}%
           (0,0.5)(\val,1.3)(\ib,0.5)}}}
\rput(3,-1){\scriptsize\tt ALIGN X(I)
                              with B(I,*)}
```



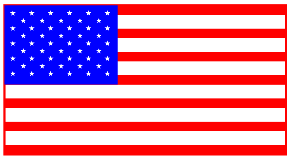

```
\renewcommand{\psedge}
  {\ncdiag[armA=0,angleB=180,armB=1.5cm]}
\pstree[treemode=R,treesep=2mm,
        levelsep=3cm]
       {\Tr{\psframebox{Z}}}
       {\multido{\i=10+5}{4}
          {{\Toval[fillstyle=solid,
                    fillcolor=black]
                  {\bf\white $z_{\i}$}}}}
```




```
\psset{unit=1.5cm}
\pscircle[fillstyle=solid,
          fillcolor=lemonchiffon]{1}
\SpecialCoor
\degrees[1.2]
\newcounter{heure}
\setcounter{heure}{1}
\multido{\n=1.4+-0.1}{12}
  {\rput[N](0.8;\n){\Roman{heure}}
   \addtocounter{heure}{1}}
\psline[linewidth=0.4mm]{<->}%
        (0.6;0.73)(0;0)(0.9;0.45)%Aiguilles
\psline[linewidth=0.1mm,linecolor=red]
        (0.9;0.15)                 %Trotteuse
```

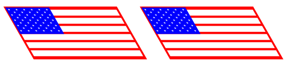



```
\psset{xunit=0.6cm,dimen=middle}
\newcommand{\bloc}
  {\psframe[fillstyle=solid,
            fillcolor=palegreen](.5,.25)
   \pspolygon[linestyle=none,
              fillstyle=solid,
              fillcolor=palegreen]
             (.5,0)(.5,.25)(2,2.5)(2,2.25)
   \psline(0,.25)(1.5,2.5)(2,2.5)(.5,.25)
   \psline(.5,0)(2,2.25)(2,2.5)}
 \newpsobject{MonPolygone}{pspolygon}
   {fillstyle=solid,fillcolor=forestgreen}
 \rput(-7,-1)
   {\multirput(1,0){6}{\bloc}
    \MonPolygone(0,.25)(0,.5)(.5,.5)
                (.5,.25)
    \MonPolygone(.5,.25)(.5,0.5)(2,2.75)
                (2,2.5)
    \MonPolygone(0,.5)(1.5,2.75)(2,2.75)
                (.5,.5)
    \multirput(.05,.62)(.08,.12){6}
      {\psline[linecolor=white](.5,0)}
    \multirput(.8,1.4)(.5,.75){2}
      {\pscurve{->}(0,0)(.625,.5)(1.25,0)
       \pscurve{->}(0,0)(1.125,.75)
                   (2.25,0)}
    \multido{\i=1+1}{6}
      {\rput(-0.75,0)
         {\rput(\i,-0.3){$P_{\i}$}}}}
```



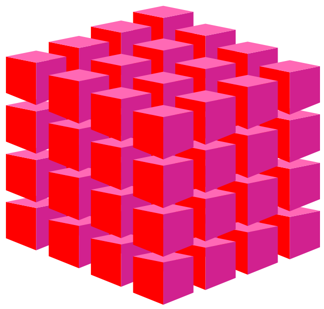
```
\psset{unit=0.25cm}
\newcommand{\etoile}
  {\pspolygon[linestyle=none,
              fillstyle=solid,
              fillcolor=white]
     (.06,0)(.095,.105)(0,.18)(.115,.18)
     (.15,.285)(.185,.18)(.3,.18)
     (.215,.105)(.24,0)(.15,.065)}
\psframe[linecolor=red](15,8)
\multips(0,8)(0,-1.231){7}
  {\psframe[linestyle=none,
            fillstyle=solid,fillcolor=red]
           (15,-0.6154)}
\psframe[linestyle=none,fillstyle=solid,
         fillcolor=blue](0,3.7)(6.1,8)
\multido{\n=4.2+0.8}{5}
  {\multips(0.35,\n)(1,0){6}{\etoile}}
\multido{\n=4.6+0.8}{4}
  {\multips(0.9,\n)(1,0){5}{\etoile}}
```

Si l'on définit l'objet précédent comme la commande **`\DrapeauAmericain`**, on peut alors manipuler ce nouvel objet comme un objet unique [14] :



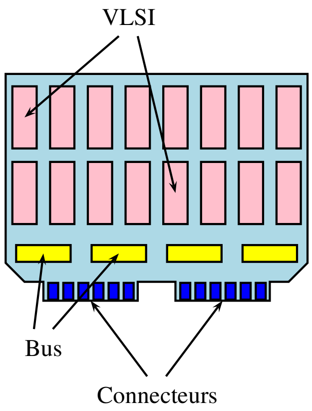
```
\psset{unit=1mm}
\multirput(18,0){2}
  {\pstilt{120}{\DrapeauAmericain}}
```


```
\font\medpala=pplb0 at 1.5cm
\psset{viewpoint=.5 -.75 .3,normal=1 0 0}
\rput(-3.7,-2.5)
  {\multido{\n=0+0.04}{26}
     {\newrgbcolor
        {MaCouleur}{0 \n\space \n}
      \ThreeDput(\n,\n,0)
        {\psframe[linestyle=none,
                  fillstyle=solid,
                  fillcolor=MaCouleur]
                 (7,3)}
      \hspace*{-0.8mm}}}
\rput(-1.95,-0.1)
  {\multido{\n=0+0.04}{26}
     {\newrgbcolor
        {MaCouleur}{1 \n\space \n}
      \ThreeDput(\n,\n,0)
        {\medpala\MaCouleur PSTricks}
      \hspace*{-0.8mm}}}
```



```
\psset{unit=4mm,linestyle=none,
       fillstyle=solid}
\newcommand{\MonCube}
  {\pspolygon[fillcolor=red]
             (0,0)(0,1.2)(1,1)(1,-0.4)
   \pspolygon[fillcolor=violetred]
             (1,-0.4)(1,1)(2,1.2)(2,0)
   \pspolygon[fillcolor=hotpink]
             (0,1.2)(1,1.4)(2,1.2)(1,1)}
\rput(-6.5,-1.4)
  {\multirput(-1.4,-0.5){4}
     {\multirput(0,1.6){4}
        {\multirput(1.4,-0.6){4}
           {\MonCube}}}
```

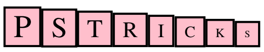
14. C'est de cette façon qu'on peut obtenir des effets analogues à celui employé pour la couverture de cette revue : c'est le même objet **`\Transparent`** qui est à la fois projeté sur le mur « virtuel », et, réduit et retourné, superposé sur le projecteur.



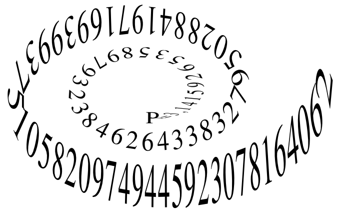

```
\psset{unit=0.5cm,fillstyle=solid}
\pspolygon[fillcolor=lightblue]
    (0,1)(0,6)(8,6)(8,1)(7.5,0.5)(7,0.5)
    (7,0)(4.5,0)(4.5,0.5)(3.5,0.5)(3.5,0)
    (1,0)(1,0.5)(0.5,0.5)
\multido{\i=4+-2}{2}
  {\multips(0.15,\i)(1,0){8}
     {\psframe[fillcolor=pink](0.7,1.7)}}
\multips(0.25,1)(2,0){4}
  {\psframe[fillcolor=yellow](1.5,0.5)}
\multido{\n=1.1+3.5}{2}
  {\multips(\n,0)(0.4,0){6}
     {\psframe[fillcolor=blue](0.32,0.5)}}
\psline{->}(3,7)(0.5,4.85)
\psline{->}(3.5,7)(4.5,2.85)
\rput(3.25,7.5){VLSI}
\psline{<-}(1,1.25)(0.75,-0.75)
\psline{<-}(3,1.25)(1.25,-0.75)
\rput(1,-1.25){Bus}
\psline{<-}(2.25,0)(3.75,-2)
\psline{<-}(5.75,0)(4.25,-2)
\rput(4,-2.5){Connecteurs}
```

En définissant les commandes suivantes pour obtenir le Ième caractère d'une chaîne, on peut ensuite appliquer une transformation différente à chacun :

```
\newtoks\Caracteres
\newtoks\CopieCaracteres
\newcommand{\ExtCarA}[1]{\expandafter\FCA\the#1\ExtCarA\FCA}
\long\def\FCA#1#2\FCA{#1\Caracteres=\expandafter{#2}}
\newcommand{\ExtCarB}[1]{\expandafter\FCB\the#1\ExtCarB\FCB}
\long\def\FCB#1#2\FCB{\Caracteres=\expandafter{#2}%
                      \CopieCaracteres=\Caracteres}
\newcounter{PosCar}
\newcommand{\CaractPos}[2]
  {\CopieCaracteres=\Caracteres%
   \setcounter{PosCar}{1}%
   \loop%
     \ifnum\value{PosCar}<#2%
       \ExtCarB{#1}%
       \addtocounter{PosCar}{1}%
   \repeat%
   \ExtCarA{\CopieCaracteres}}
```



```
\Large
\rput(-1.5,3)
  {\multido{\i=20+20}{8}
     {\Caracteres=\expandafter{PSTRICKS}%
      \pstilt{\i}
        {\CaractPos{\Caracteres}
                   {\multidocount}}}}
\rput(-1.5,0.2)
  {\multido{\n=4+-0.5}{8}
     {\Caracteres=\expandafter{PSTRICKS}%
      \scalebox{1 \n}
        {\CaractPos{\Caracteres}
                   {\multidocount}}}}
\rput(-1.5,-2.5)
  {\multido{\n=1+-0.1}{8}
     {\Caracteres=\expandafter{PSTRICKS}%
      \psframebox[dimen=middle,
                  fillstyle=solid,
                  fillcolor=pink]
        {\scalebox{\n\space \n}
           {\CaractPos{\Caracteres}
                      {\multidocount}}}}}
```



```
\psset{linestyle=none}
\readdata{\spirale}{spirale.data}
\rput(-5.1,-3.5)
  {\pstextpath
    {\dataplot[plotstyle=curve]{\spirale}}
    {\multido{\n=0.7+0.04}{78}
       {\Caracteres=\expandafter{Pi=3,141%
         59265358979323846264338327950288%
         41971693993751058209749445923078%
         78164062}%
        \scalebox{1 \n}
          {\CaractPos{\Caracteres}
                     {\multidocount}}}}}
```

## 6. Mise en valeur des tableaux

On peut bien évidemment utiliser les commandes de mise en couleur du texte à l'intérieur des cellules d'un tableau [15]. Mais il est également facile de superposer les tableaux sur des rectangles de couleurs (*pures* ou *dégradées*). Enfin, il existe des commandes spéciales pour colorier les cellules et les lignes.

Voici deux exemples qui illustrent ces diverses possibilités :

```
\newsavebox{\Tableau}
\savebox{\Tableau}
 {\begin{tabular}{|l|c|c|c|l|}
     \NAC{red}\hline\ENAC \NAC{red}\hline\ENAC
     \LCC
     \pink&\pink&\pink&\pink&\pink \\
     \multicolumn  {1}{|c|}{\cyan\bf Produits} &
       \multicolumn{1}{|c|}{MVS} &
       \multicolumn{1}{|c|}{AIX} &
       \multicolumn{1}{|c|}{VP}  &
       \multicolumn{1}{|c|}{Destination(s)} \\
     \ECC
     \NAC{red}\hline\ENAC \NAC{red}\hline\ENAC
     {\bf ALCHEMY} &X& & & Grappe \\ \hline
     {\bf AMBER}   &X& & & C90 \\ \hline
     {\bf AMPAC}   &X&X& & {\leavevmode\red Abandon} \\ \hline
     {\bf CRYSTAL} &X& & & $\Rightarrow$ CNUSC \\ \hline
     {\bf CSMP}    &X& & & {\leavevmode\red Abandon} \\ \hline
     {\bf DGEOM}   & & &X& C90 \\ \hline
   \end{tabular}}
   \psframebox[linestyle=none,fillstyle=gradient,
                 gradmidpoint=1,gradbegin=white,
                 gradend=lightblue,framesep=0]
               {\usebox{\Tableau}}
```

| Produits | MVS | AIX | VP | Destination(s) |
|---|---|---|---|---|
| **ALCHEMY** | X | | | Grappe |
| **AMBER** | X | | | C90 |
| **AMPAC** | X | X | | Abandon |
| **CRYSTAL** | X | | | ⇒ CNUSC |
| **CSMP** | X | | | Abandon |
| **DGEOM** | | | X | C90 |

15. Dans certains cas, il est nécessaire de faire précéder ces commandes de la macro `\leavevmode` pour leur garantir une *portée* correcte.

```
\begin{tabular}{|l|c|c|c|l|}
  \hline
  \multicolumn  {1}{|c|}{\cyan\bf Produits} &
    \multicolumn{1}{|c|}{MVS} &
    \multicolumn{1}{|c|}{AIX} &
    \multicolumn{1}{|c|}{VP}  &
    \multicolumn{1}{|c|}{Destination(s)} \\
  \hline \hline
  \LCC
                &\lightblue&           &      & \\
  {\bf ALCHEMY} &          &           &      & Grappe\\ \hline
  {\bf AMBER}   &          &           &      & C90 \\ \hline
  \ECC
  \LCC
                &\lightblue&\palegreen&      & \\
  {\bf AMPAC}   &          &           &      &
                          {\leavevmode\red Abandon} \\ \hline
  \ECC
  \LCC
                &\lightblue&           &      & \\
  {\bf CRYSTAL} &          &           &      &
                          $\Rightarrow$ CNUSC \\ \hline
  {\bf CSMP}    &          &           &      &
                          {\leavevmode\red Abandon} \\ \hline
  \ECC
  \LCC
                &          &           &\pink& \\
  {\bf DGEOM}   &          &           &      & C90 \\ \hline
  \ECC
\end{tabular}
```

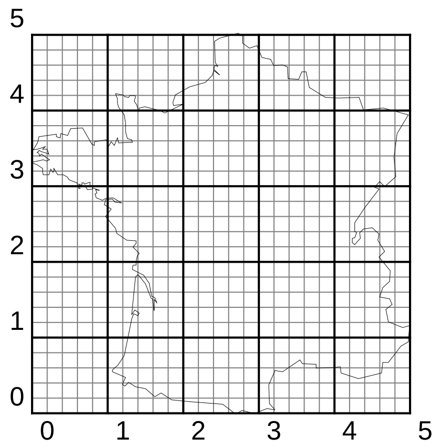
| Produits | MVS | AIX | VP | Destination(s) |
|---|---|---|---|---|
| **ALCHEMY** | | | | Grappe |
| **AMBER** | | | | C90 |
| **AMPAC** | | | | Abandon |
| **CRYSTAL** | | | | ⇒ CNUSC |
| **CSMP** | | | | Abandon |
| **DGEOM** | | | | C90 |

# 7. Quelques exemples complexes

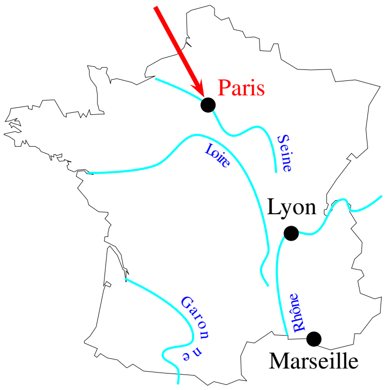
## 7.1. Annotation du contenu d'un fichier PostScript

On peut grandement s'aider d'une grille pour obtenir les coordonnées des informations à ajouter au contenu d'un fichier PostScript.

```
\rput(-2.5,-5){\psgrid(5,5)}
\epsfxsize=5cm
\rput(0,-2.5){\epsffile{france.eps}}
```




```
\epsfxsize=5cm
\rput(0,-2.5){\epsffile{france.eps}}
\rput(-2.5,-5)
  {\pstextpath[c](0,1mm)
     {\pscurve[linecolor=cyan]
        (1.6,1.4)(2.5,0.6)(2.1,0.4)(2.3,0.2)(2.3,0)}
     {\scriptsize\blue Garonne}
   \pstextpath[c](0,-2mm)
     {\pscurve[linecolor=cyan]
        (1.1,2.8)(2,2.9)(2.6,3.3)(3.5,1.8)(3.4,1.5)(3.5,1.3)}
     {\scriptsize\blue Loire}
   \pstextpath[r](0,1mm)
     {\pscurve[linecolor=cyan]
        (2,4.05)(2.7,3.7)(3,3.3)(3.4,3.4)(3.6,2.8)}
     {\scriptsize\blue Seine}
   \pstextpath[l](0,-2.5mm)
     {\pscurve[linecolor=cyan]
        (3.75,0.65)(3.75,2)(4.1,2)(4.5,2.45)(4.6,2.3)(5,2.5)}
     {\scriptsize\blue Rhône}
   \qdisk(2.7,3.7){1mm}
   \uput[ur](2.7,3.7){\small\red Paris}
   \psline[linewidth=0.6mm,linecolor=red]{->}(2,5)(2.65,3.8)
   \qdisk(4.1,0.6){1mm}
   \uput[d](4.1,0.6){\small Marseille}
   \qdisk(3.8,2){1mm}
   \uput[u](3.8,2){\small Lyon}
  }
```



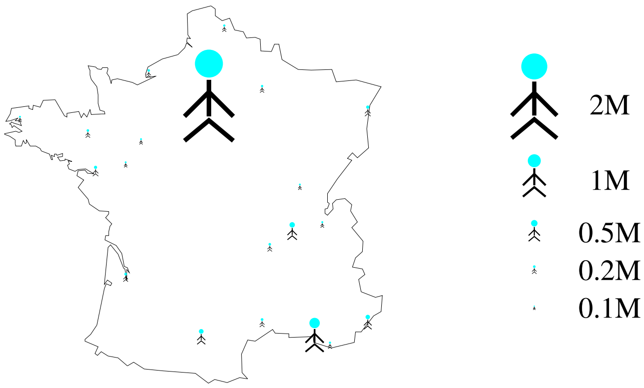

## 7.2. Annotations statistiques

```
\begin{center}
  \begin{pspicture}(9,6)
    \rput(2.5,6){\Large Principales agglomérations françaises
                     (1990)}
    \rput(2.5,5.7){Source INSEE}
    \epsfxsize=5cm
    \rput(2.5,2.5){\epsffile{france.eps}}
    \newcommand{\Homme}
      {\pscircle[linestyle=none,fillstyle=solid,fillcolor=cyan]
                (0,0.25){0.1}
       \psline(0,-0.075)(0,0.15)
       \psline(-0.15,-0.075)(0,0.075)(0.15,-0.075)
       \psline(-0.15,-0.225)(0,-0.075)(0.15,-0.225)}
    \rput(2.7,3.7){\scalebox{2.152}{\Homme}} % Paris
    \rput(4.1,0.6){\scalebox{0.801}{\Homme}} % Marseille
    .......................................
    \rput(1.6,2.9){\scalebox{0.141}{\Homme}} % Angers

    \rput(7,3.7){\scalebox{2}{\Homme}}\rput(8,3.7){2M}
    \rput(7,2.7){\scalebox{1}{\Homme}}\rput(8,2.7){1M}
    \rput(7,2){\scalebox{0.5}{\Homme}}\rput(8,2){0,5M}
    \rput(7,1.5){\scalebox{0.2}{\Homme}}\rput(8,1.5){0,2M}
    \rput(7,1){\scalebox{0.1}{\Homme}}\rput(8,1){0,1M}
  \end{pspicture}
\end{center}
```

Principales agglomérations françaises (1990)

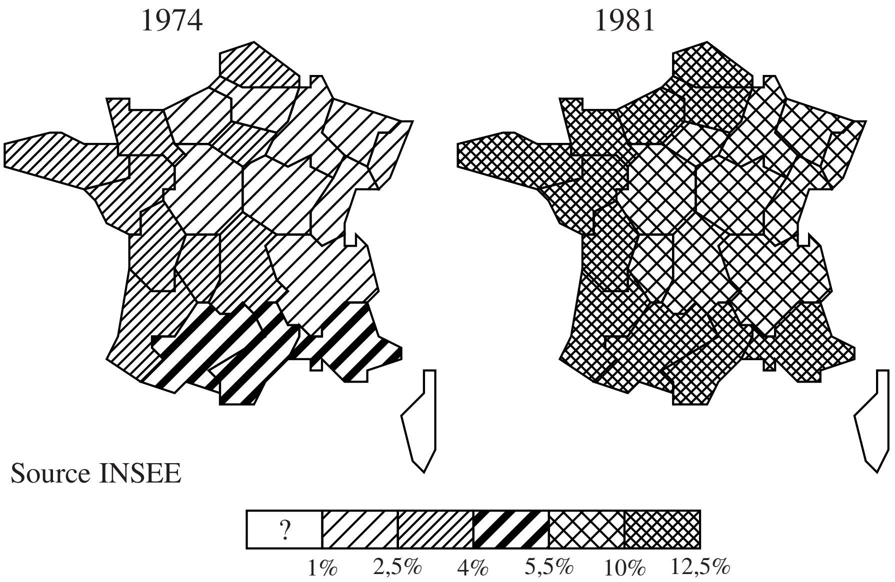
Source INSEE




## 7.3. Coloration de cartes

La coloration de cartes est une tâche qui peut être effectuée sans grandes difficultés avec **PSTricks** :

Taux de chômage en 1974 et 1981



La première chose à faire est d'obtenir les coordonnées des différents ensembles à manipuler (dans notre exemple les régions françaises) [16].

```
\savedata     {\nord}[{344 74 344 44 389 29 449 74 449 89 374 89
                      344 74}]
\savedata {\picardie}[{329 89 344 74 374 89 449 89 434 134 419
                      149 359 134 359 104 329 89}]
................................................................
```

Ensuite, il faut définir les styles de remplissage qui seront utilisés :

```
\newrgbcolor{RougeA}{1. 0.8571 0.8571}
\newrgbcolor{RougeB}{1. 0.7143 0.7143}
......................................................
\newpsstyle{StyleCouleurA}
  {plotstyle=polygon,fillstyle=solid,fillcolor=RougeA}
\newpsstyle{StyleCouleurB}
  {plotstyle=polygon,fillstyle=solid,fillcolor=RougeB}
......................................................
```

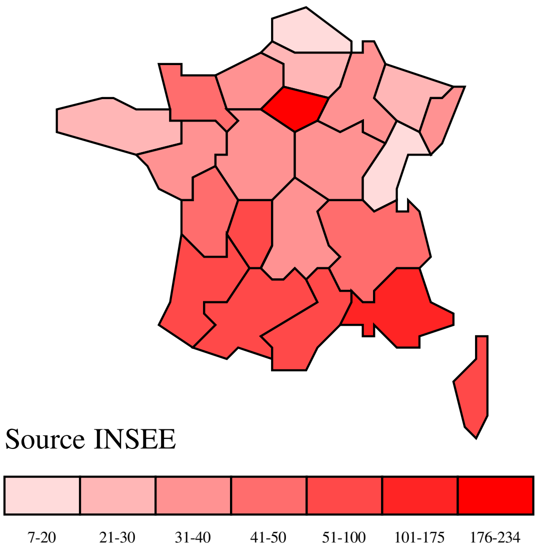
Puis il ne reste plus qu'à remplir les différentes aires suivant les styles adéquats, en fonction des valeurs à représenter [17].

```
\psset{unit=0.01cm}
\rput(0,0){\large Cas de SIDA mi-1992 (par million d'habitants)}
\rput(-340,-50)
  {\scalebox{1 -1}
     {\dataplot[style=StyleCouleurA]{\nord}
      \dataplot[style=StyleCouleurB]{\picardie}
      ........................................
      \dataplot[style=StyleCouleurE]{\corse}}}
\rput(0,-750)
  {\psset{dimen=middle}
   \psframe[style=StyleCouleurA](-350,0)(-250,50)
   ..............................................
   \psframe[style=StyleCouleurG](250,0)(350,50)
   \rput(-300,-30){\scriptsize 7-20}
   ....................................
   \rput(300,-30){\scriptsize 176-234}}
\rput[l](-350,-650){Source INSEE}
```

16. Nous les avons ici spécifiées de manière grossière à l'aide du logiciel [xfig], qui définit l'origine en haut à droite, et non en bas à droite comme **PSTricks**, ce qui explique la présence de l'ordre `\scalebox{1 -1}` dans l'exemple...

17. Il est en fait aisément envisageable de construire un pré-processeur qui générerait automatiquement les cartes suivant les coordonnées des différentes zones, les valeurs qui leur sont associées, et les attributs définis.

Cas de SIDA mi-1992 (par million d'habitants)




## 8. Graphiques de gestion

Il est possible d'utiliser **PSTricks** pour réaliser des graphiques d'un très haut niveau de complexité, puisque l'on dispose de toutes les briques de base pour cela. Toutefois, une telle programmation devient vite lourde et inextricable sur des cas compliqués ... à moins que l'on ne puisse développer pour le but visé un programme qui génère les ordres **PSTricks** correspondants. Plusieurs voies sont utilisables. Les récentes versions du logiciel [gnuplot][18], par exemple, comportent un pilote **PSTricks**[19], capable de convertir en ordres **PSTricks** les tracés obtenus via [gnuplot].

Pour notre part, nous avons réalisé un programme (écrit en **SHELL** et **AWK**) qui permet de générer des graphiques dits de *gestion* (camemberts, histogrammes en 2D ou 3D, lignes et surfaces)[20] à partir de courts fichiers contenant les données et les paramètres décrivant la représentation souhaitée. Si une telle application ne peut évidemment offrir qu'une petite partie des fonctionnalités d'un logiciel aussi riche et complexe que [xmgr], il n'empêche qu'une très grande variété de résultats peuvent être obtenus, et que la qualité de ceux-ci est remarquable, en

18. Logiciel du domaine public pour le tracé de courbes 2D et 3D.

19. Écrit par Raymond TOY, **<toy@soho.crd.ge.com>**.

20. Peu de choses existaient dans ce domaine avec (LA)TEX, jusqu'à l'apparition de **AlDraTEX** [Gurari 94]. Pour les histogrammes, la solution la plus évoluée était [**bar.sty**].

raison de la richesse des macros-commandes de **PSTricks**. De plus, comme le résultat produit par le programme est une suite d'ordres **PSTricks**, il est bien évidemment possible d'intervenir après coup à ce niveau-là et d'opérer des changements particuliers.

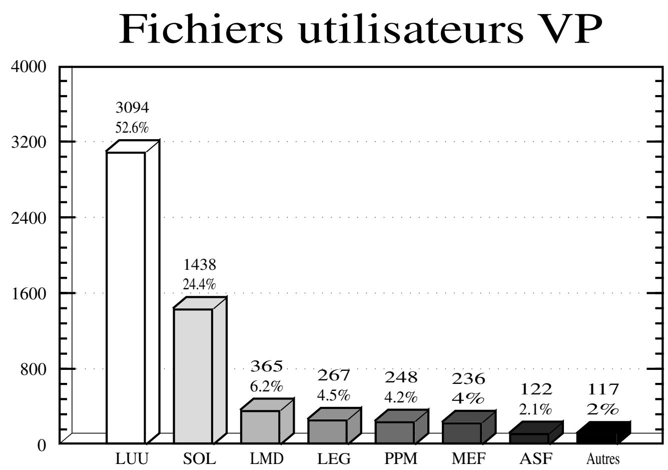
Voici quelques exemples extraits de ceux, nombreux, qui sont donnés dans la documentation de **pstchart.sh**.

À partir d'un fichier de données *brutes* comme celui-ci [21] :

```
3094 | LUU
1438 | SOL
 365 | LMD
 267 | LEG
 248 | PPM
 236 | MEF
 122 | ASF
  57 | DRT
  33 | AMB
  18 | TPR
   9 | RRS
```

la commande suivante permet de générer le graphique ci-dessous :

```
pstchart.sh vbar dimx=8 3d nb-values=8 \
            print-percentages print-values \
            grayscale=white-black data-change-colors \
            title="Fichiers utilisateurs VP" \
            label-others="Autres" center <users.data
```




21. Qui peut en fait être lui-même généré par une autre application — voir un tel exemple dans la documentation de **pstchart.sh**.

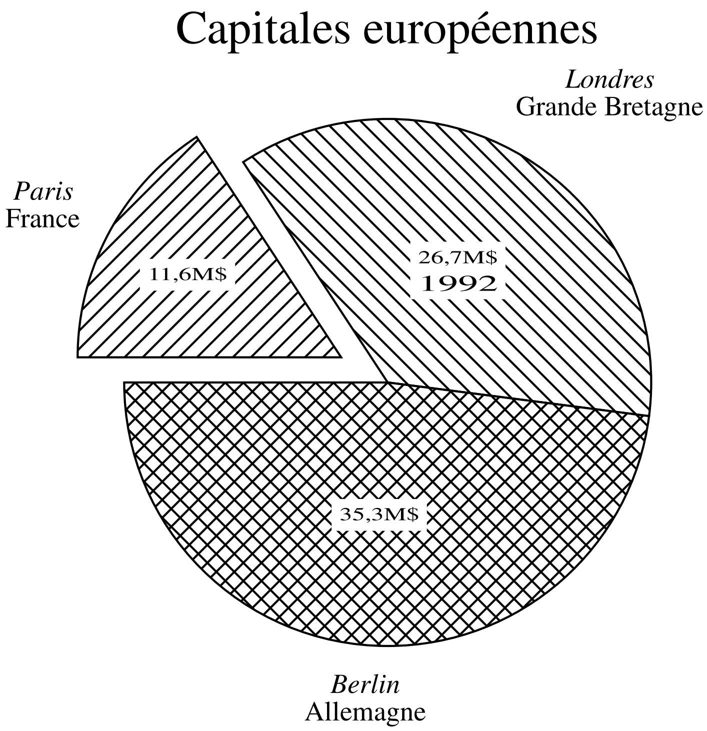
Mais si l'on spécifie certains paramètres, on peut influer sur **toutes** les caractéristiques du graphique :

```
# file1.data : fichier de données pour les exemples de graphiques

#      | Titre
TITLE | Capitales européennes

# Aspects des représentations
#       | Style
ASPECT | hlines
ASPECT | vlines
ASPECT | crosshatch

# Valeur  | Premier libellé                | Échelle || Second libellé ||||Part séparée
   11.6   | \em Paris\\France              | 0.9     || 11,6M\$        ||||true
   26.7   | \em Londres\\Grande Bretagne   | 0.9     || 26,7M\$\\1992
   35.3   | \em Berlin\\Allemagne          | 0.9     || 35,3M\$
```




**pstchart.sh pie dim=7 center <file1.data**

Voici quelques autres exemples illustrant les possibilités majeures de cette application :

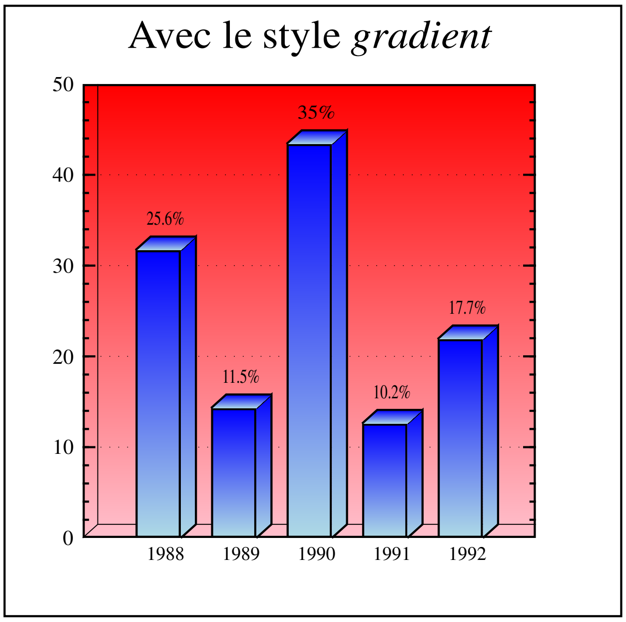
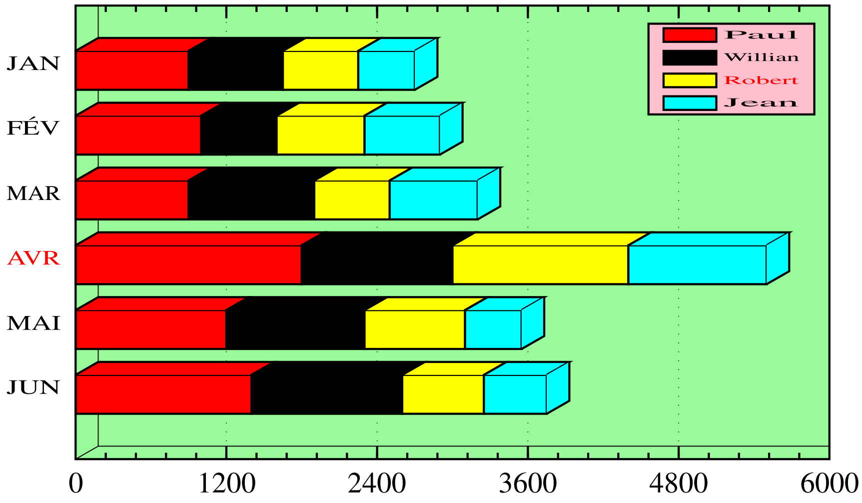
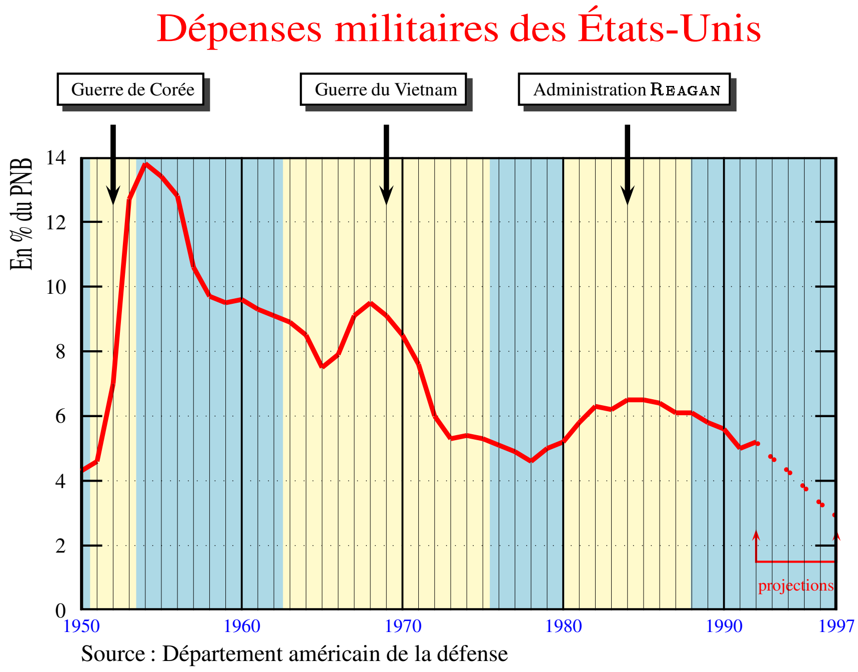



FIG. 1 – *Autre exemple*

```
pstchart.sh vbar dim=6 3d print-percentages \
boxit center figure <file2.data
```




```
pstchart.sh hbar dimx=10 dimy=6 stack 3d center <multsets.data
```




```
pstchart.sh hlines dimx=10 dimy=6 max=14 noshowpoints \
    input-begin=dod.add center <dod.data
```

## 9. Réalisation de transparents

La réalisation de transparents est une tâche souvent indissociable de la production de documents. SLITEX a apporté en son temps un certain nombre de fonctionnalités très utiles, mais il est évident qu'aujourd'hui il est nécessaire de pouvoir disposer de solutions plus satisfaisantes [22].

Or Timothy Van ZANDT a également créé une classe de documents de nom **Seminar**, qui, tout en assurant une conversion facile depuis des fichiers SLITEX, offre de nettes améliorations (dont un traitement très complet de la couleur), et est bien sûr étroitement couplé avec **PSTricks**. Michel GOOSSENS et Sebastian RAHTZ en détaillent plus loin les caractéristiques et les apports. Aussi nous ne donnerons ici que quelques exemples de transparents que nous avons réalisés pour nos propres besoins, dans le seul but de convaincre de la richesse et de la qualité des documents qu'on peut ainsi obtenir...

22. Plusieurs tentatives avaient déjà été faites, soit en améliorant SLITEX lui-même, et notamment le support de la couleur ([Love 90] par exemple), soit en créant de nouveaux outils, comme [**FoilTEX**].

# Le multi-tâches

# sur machines CRAY

**Service d'assistance aux utilisateurs**

**Denis Girou**

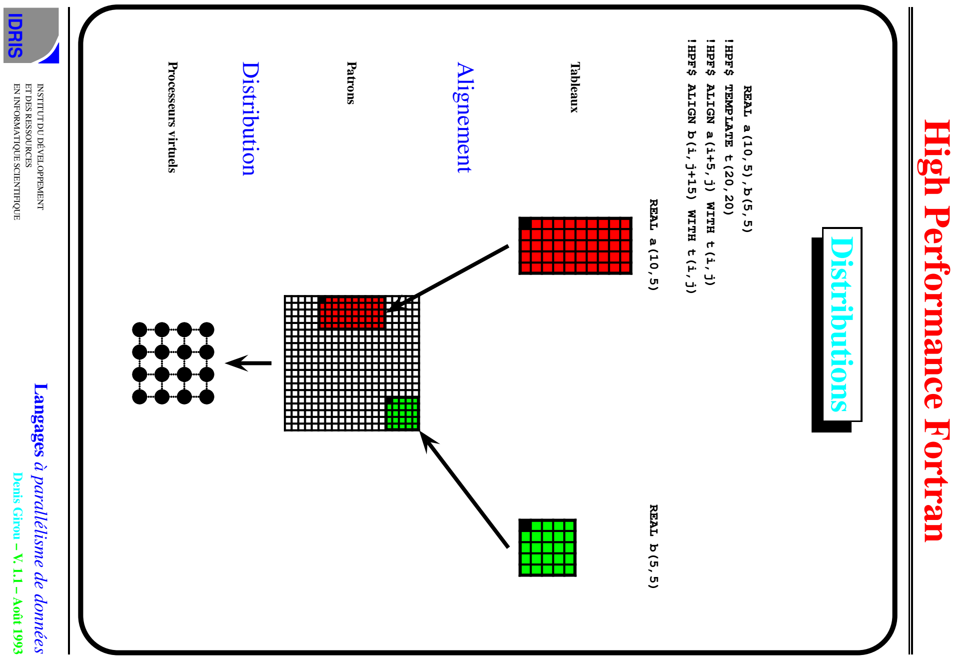
**Messagerie : Denis.Girou@idris.fr**





## Système de compilation `cf77`

- divisé en deux pré-processeurs, le compilateur lui-même et l'éditeur de liens,
- plus de pré-processeurs dans `cf90`,
- pré-processeur `FPP` :
  - restructuration du code pour la vectorisation et éventuellement la parallélisation,
  - insertion de directives **CDIR@** pour la vectorisation,
  - insertion de directives **CMIC@** pour la parallélisation,
  - on peut analyser le travail de l'optimiseur en étudiant la sortie de `FPP` : fichier `nom.m`.
- processeur intermédiaire `FMP` :
  - étude possible de la portée des variables,
  - traduction des directives **CMIC@** et **$**.
- + outils d'aide et d'analyse de performances.




## 10. Conclusion

Au terme de ce long panorama, nous espérons avoir convaincu de l'importance et de l'intérêt de **PSTricks** et de **Seminar**[23]. À travers tous ces exemples, illustrant les aspects très divers de ces extensions à (LA)TEX, nous espérons que les lecteurs concevront plus facilement comment utiliser ces commandes pour leurs propres applications, et y auront trouvé la source de quelques idées nouvelles.

Ces extensions offrent l'accès aisé et direct à une large part des fonctionnalités présentes dans PostScript, ce qui autorise un très grand nombre d'effets inaccessibles avec (LA)TEX. Par leur nombre, leur richesse et leur puissance, les macros-commandes de **PSTricks** permettent donc de répondre à la majorité des besoins non satisfaits par (LA)TEX, en unifiant des possibilités qu'on trouvait parfois ailleurs, mais de façon disparate, et généralement d'une manière plus pauvre. L'accès complet à la couleur est aussi présent, ce qui est dès aujourd'hui, et le sera encore plus dans le proche avenir, un facteur essentiel d'enrichissement des documents. De plus, comme on l'a déjà souligné, le fait de disposer de ces potentialités via un langage de programmation offre des possibilités strictement hors de portée des produits purement interactifs, et pourra par exemple permettre de développer des pré-processeurs pour des besoins spécifiques.

23. Outre **FancyBox**, auquel nous avons déjà fait allusion, un autre développement intéressant de Timothy Van Zandt est **Poster**, qui permet, comme son nom l'indique, de réaliser aisément des *posters*, en agrandissant fortement le contenu des pages logiques, dont les différents morceaux prédécoupés sont imprimés sur plusieurs pages physiques.

# Références

[**bar.sty**] par Joachim BLESER, TH Darmstadt Hochschulrechenzentrum, Allemagne.

[**catmac.sty**] par Michael BARR, **<barr@linc.cis.upenn.edu>**, McGill University, Canada.

[**colordvi.sty**] par Jim HAFNER, **<hafner@almaden.ibm.com>**, IBM Research Division, Almaden Research Center, USA.

[**ColorRgb.TEX**] par Christophe CÉRIN, **<cerin@lri.fr>**, Faculté de Mathématique et d'Informatique, Université de Picardie, et Laboratoire de Recherche en Informatique, Université de Paris Sud, France.

[*daVinci*] par Michael FRÖHLICH et Mattias WERNER, **<davinci@informatik.uni-bremen.de>**, Universität Bremen, Allemagne.

[**diagram**] par Francis BORCEUX, **<borceux@agel.ucl.ac.be>**, Université de Louvain-la-Neuve, Belgique.

[**diagrams.tex**] par Paul TAYLOR, **<pt@doc.ic.ac.uk>**, Imperial College of Science, Technology and Medicine, Department of Computing, Grande Bretagne.

[**eepic.sty**] par Conrad KWOK, **<kwok@iris.ucdavis.edu>**, USA.

[**epic.sty**] par Sunil PODAR, **<podar@sbcs.csnet>**, SUNY at Stony Brook, Department of Computer Science, USA.

[**flow**] par Terry BROWN, **<browntl@lincoln.ac.nz>**, Nouvelle-Zélande.

[**Flow.sty**] par Marion van GEEST-SLORT, Center for the Automation of Weapon and Command Systems, Royal Netherlands Navy, Pays-Bas.

[**FoilTEX**] par Jim HAFNER, **<hafner@almaden.ibm.com>**, IBM Research Division, Almaden Research Center, USA.

[gnuplot] par Thomas WILLIAMS et Colin KELLEY, **<info-gnuplot@dartmouth.edu>**, USA.

[**Graph-TEX**] par John PLIAM, **<pliam@ima.umn.edu>**, Institute for Mathematics and its Applications, University of Minnesota, USA.

[Gurari 94] Eitan M. GURARI, *TEX and LATEX: Drawing and Literate Programming*. McGraw-Hill, New-York, USA, 1994.

[Hœnig 92] Alan HŒNIG, **<ajhjj@cunyvm.edu>**, John Jay College, Department of mathematics, USA, *When TEX and* METAFONT *work together*, conférence EuroTEX 92, publiée par les **Cahiers GUTenberg**, numéro 14, avril 93, pages 1-19.

[**LameTEX**] par Jonathan MONSARRAT, **<jgm@cs.brown.edu>**, USA.

[Love 90] David LOVE, **<d.love@daresbury.ac.uk>**, SERC Daresbury Laboratory, Warrington, Grande-Bretagne, *Experiments in TEXnicolour — A* SLITEX *Sub-style for Colour Printers*, **TUGboat**, Volume 11, N° 4, November 1990, pages 652-656.

[**MetaPost**] par John D. HOBBY, ATT Bell Laboratories, USA.

[**mfpic**] par Tom LEATHRUM, **<moth@dartmouth.edu>**, USA.

[**PICTEX**] par Michael WICHURA, **<wichura@galto.uchicago.edu>**, Chicago University, USA.

[**rail.sty**] par L.W.J. ROOIJAKKERS, **<lwj@cs.kun.nl>**, University of Nijmegen, Pays-Bas.

[**TeXdraw**] par Peter KABAL, **<kabal@aldebaran.ee.mcgill.ca>**, McGill University, Department of Electrical Engineering, USA.

[**tree**] par Greg LEE, **<lee@uhccux.uhcc.hawaii.edu>**, University of Hawaii, Department of Linguistics, Honolulu, USA.

[**trees** et **trees.sty**] par Avery ANDREWS, utilisant **tree-dvips.sty** par Emma PEASE, **<emma@csli.stanford.edu>**, CSLI, Stanford University, USA.

[**TreeTeX**] par Anne BRUEGGEMANN-KLEIN, **<abk@sun1.ruf.uni-freiburg.dbp.de>**, Universität Freiburg, Institut für Informatik, Allemagne, et Derick WOOD, University of Waterloo, Department of Computer Science, Canada.

[**tree.sty**] par Edward M. REINGOLD, **<reingold@cs.uiuc.edu>** et Nachum DERSHOWITZ, **<nachum@cs.uiuc.edu>**, University of Illinois, USA.

[xfig] par Brian V. Smith, **<bvsmith@lbl.gov>**, Lawrence Berkeley Laboratory, USA.

[xmgr] par Paul J. TURNER, **<pturner@amb4.ccalmr.ogi.edu>**, Oregon Graduate Institute of Science and Technology, Beaverton, Oregon, USA.

[**XYpic**] par Kristoffer H. ROSE **<kris@diku.dk>**, Université de Copenhague, Danemark.