

Inclusions graphiques et homogénéisation des polices

Denis ROEGEL

roegel@loria.fr

mai 1997

1. Introduction

Les figures d'un document `TEX` sont :

- produites directement depuis `TEX` :
`picture`, extensions `epic`, `eepic`, `PSTricks`, `XY-pic`, `DraTeX`, etc.;
- produites par un programme extérieur puis incluses sous forme d'un fichier traité par le pilote :
`tgif`, `xfig`, `gnuplot`, `metapost`, etc.

1.1. Problème d'homogénéisation

- Police différente pour le texte et la figure, typiquement *Computer Modern* pour le texte, et polices *PostScript* pour la figure ;
- Même polices, mais tailles différentes ;

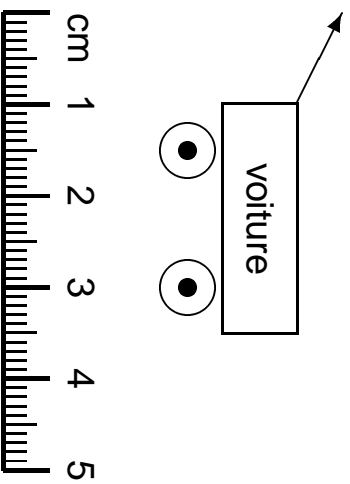
1.2. Solutions

- Le problème ne se pose pas quand la figure est définie dans l'environnement `TEX`, c'est-à-dire lorsque les commandes de `TEX` sont accessibles. C'est le cas en règle générale lorsque la figure est produite directement depuis `TEX`, même si la figure fait intervenir des éléments PostScript, comme avec `PSTricks`.
- Certains programmes extérieurs sont interfacés avec `TEX` et comportent un système permettant d'inclure des passages en `TEX` : `xfig`, `metapost`.
- Dans le cas où l'on dispose simplement d'une figure au format PostScript, on peut utiliser l'extension (*package*) `psfrag` pour effectuer des substitutions introduisant dans la figure des portions de `TEX`.

2. Création de figures depuis TEX

Exemples : environnements `picture` et `PSTricks`

2.1. Environnement `picture`

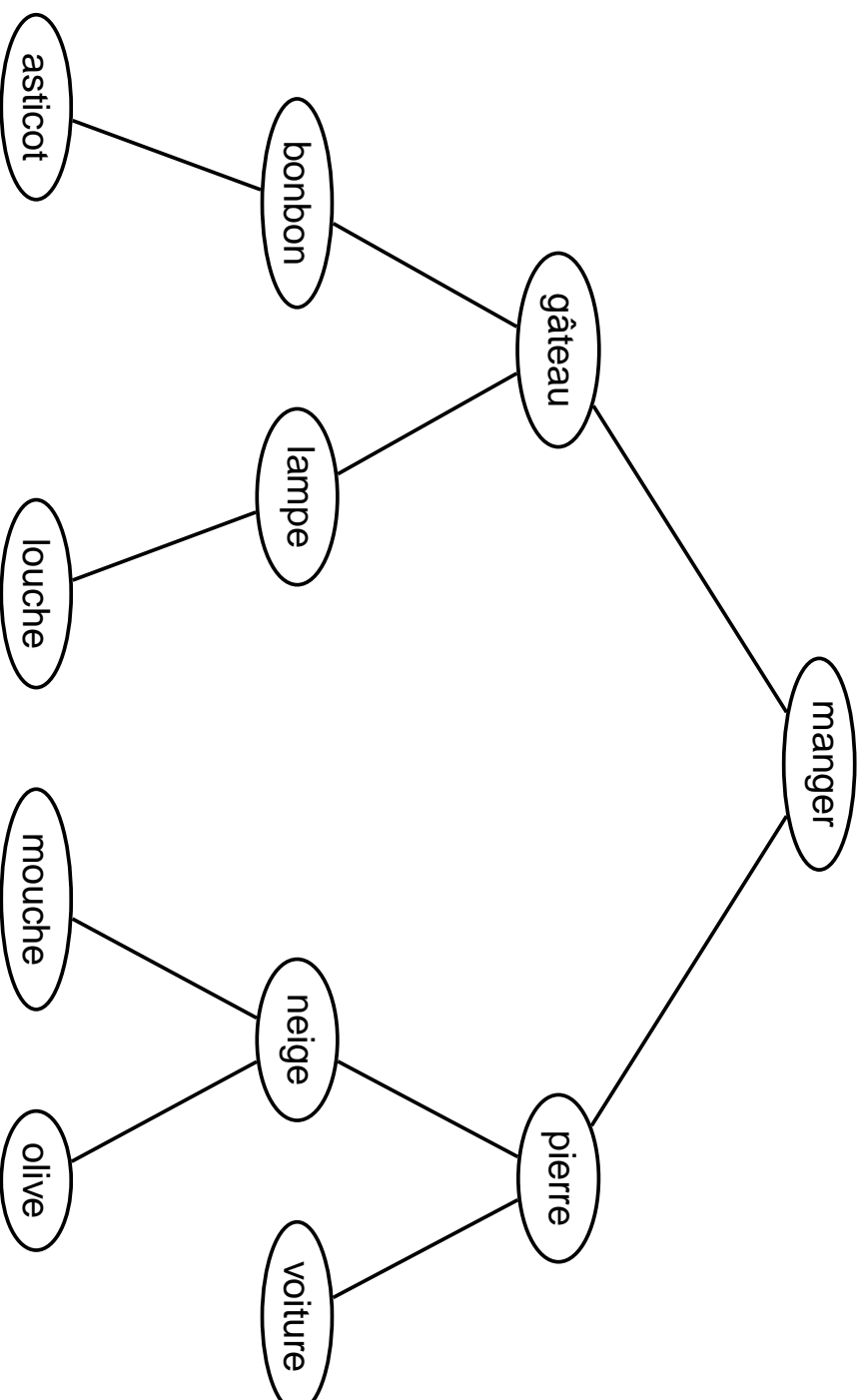


```

\newcounter{cms}
\setlength{\unitlength}{.7mm}
\begin{picture}(50,39)
\put(15,20){\circle{6}}
\put(30,20){\circle{6}}
\put(15,20){\circle*{2}}
\put(30,20){\circle*{2}}
\put(10,24){\framebox(25,8){voiture}}
\put(10,32){\vector(-2,1){10}}
\multiput(10,7)(10,0){5}{%
\addtocounter{cms}{1}
\makebox(0,0)[b]{\arabic{cms}}}
\put(0,7){\makebox(0,0)[bl]{cm}}
\multiput(1,0)(1,0){49}{\line(0,1){2,5}}
\multiput(5,0)(10,0){5}{\line(0,1){3,5}}
\thicklines
\put(0,0){\line(1,0){50}}
\multiput(0,0)(10,0){6}{\line(0,1){5}}
\end{picture}

```

2.2. Extension PSTricks



```

\usepackage{pstree}
...
\def\mytree#1{\pstree{\Toval{#1}}}{
\mytree{manger}{%
  \mytree{g^ateau}{%
    \mytree{bonbon}{
      \mytree{asticot}{}%
      \Tn
    }%
  \mytree{lampe}{
    \Tn
    \mytree{louche}{}%
  }%
}
\mytree{pierre}{%
  \mytree{neige}{
    \mytree{mouche}{}%
    \mytree{olive}{}%
  }%
  \mytree{voiture}{}
}
}

```


2.2.1. Avantages de PSTricks

- Bibliothèque de base très complète ;
- Récupération du contexte TEX.

2.2.2. Inconvénients de PSTricks

- Pas de récupération d'informations de l'exécution PostScript ;
- Tous les calculs doivent donc être faits dans TEX, en même temps que la génération du document ;
- Pas forcément rapide (mais possibilité de compiler les figures à part).

3. Programmes extérieurs interfacés avec T_EX

3.1. X_FI_G

`xfig` est lancé avec l'option `-sp` (pour « `special text` »)

Superposition de la partie PostScript et d'un environnement `picture`. Les deux fichiers, d'extension `.pstex` et `.pstex_t` sont créés à l'exportation.

⇒ Ceci peut être automatisé avec `transfig`.

`transfig -lpstex fig-xfig` génère un `Makefile`

3.1.1. Fichier pstex_t : exemple

```

\begin{picture}(0,0)%
\epsfig{file=fig-xfig.pstex}%
\end{picture}%
\setlength{\unitlength}{0.00083300in}%
%
\begingroup\makeatletter\ifx\SetFigFont\undefined%
\gdef\SetFigFont#1#2#3#4#5{%
\reset@font\fontsize{#1}{#2pt}%
\fontfamily{#3}\fontseries{#4}\fontshape{#5}%
\selectfont}%
\fi\endgroup%
\begin{picture}(2712,645)(1714,-1723)
\put(4426,-1636){\makebox(0,0)[lb]{%
\smash{\SetFigFont{12}{14.4}{\rmdefault}%
{\mddefault}{\updefault}%
texte $\frac{12}{4}$}}
}
\end{picture}

```

3.1.2. Makefile de transfig

```
#
# TransFig makefile
#
all: fig-xfig.tex

# translation into pstex

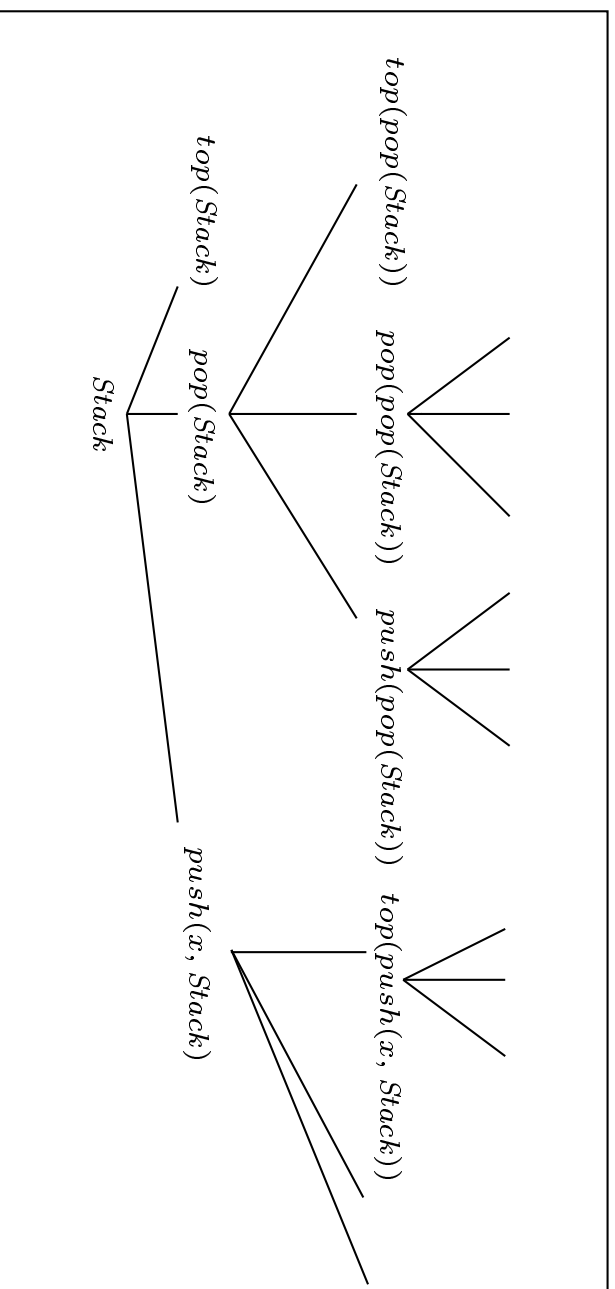
fig-xfig.tex: fig-xfig.ps Makefile
fig2dev -L pstex_t -p fig-xfig.ps fig-xfig.fig > fig-xfig.tex

clean:
    rm -f fig-xfig.tex

fig-xfig.ps: fig-xfig.fig Makefile
fig2dev -L pstex fig-xfig.fig > fig-xfig.ps

clean:
    rm -f fig-xfig.ps
```

3.1.3. Exemple



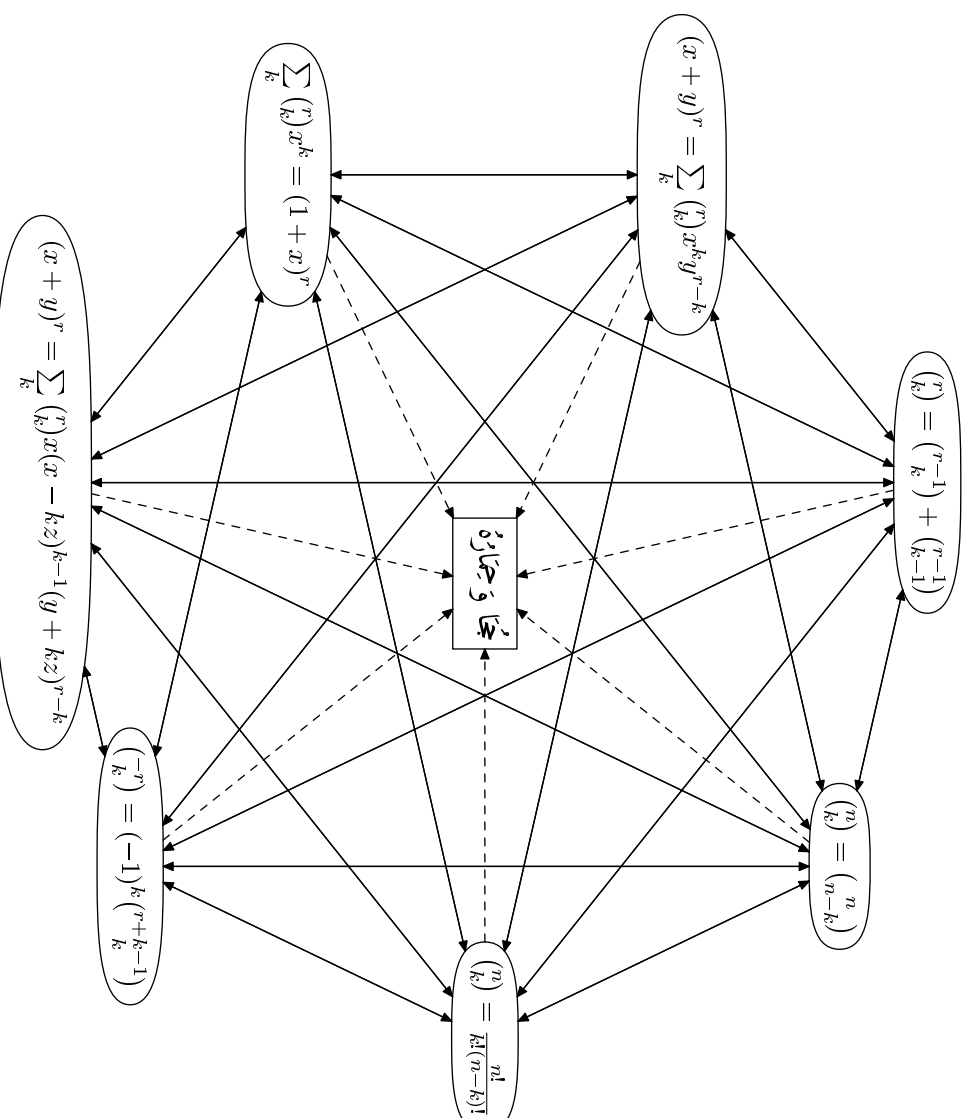
3.1.4. Inconvénients

On ne récupère pas d'informations de T_EX.

Le texte n'est pas forcément bien placé par rapport aux autres éléments graphiques (encadrés de mauvaises dimensions, mauvais centrages, etc.).

3.2. METAPOST

On récupère des informations de TEX, car METAPOST appelle TEX.



3.2.1. Source : texte en arabe

⇒ Utilisation d'ArabTeX

```
verbatimtex
\documentclass{article}
\usepackage{arabtex}
\begin{document}
\vocalize
\setarab
\newbox\arabbox
\setbox\arabbox=\hbox{\RL{^gu.hA wa-.himAruru}}
\setnone
etex
```

3.2.2. Source : figure

```

input boxes
def link(expr i, j)=drawarrow b[i].c--b[j].c
    cutbefore bpath b[i] cutafter bpath b[j]; enddef;
beginfig(1);
numeric u;u=6cm;
circleit.b1(btex  $\sum_{n=1}^n \frac{1}{n}$  etex);
circleit.b2(btex  $\sum_{n=1}^n \frac{1}{n^2}$  etex);
circleit.b3(btex  $\sum_{r=1}^r \frac{1}{r}$  etex);
circleit.b4(btex  $\sum_{r=1}^r \frac{1}{r^2}$  etex);
circleit.b5(btex  $\sum_{r=1}^r \frac{1}{r^k}$  etex);
circleit.b6(btex
     $\sum_{r=1}^r \sum_{k=1}^r x^{k-1}(y+kz)^{r-k}$  etex);
circleit.b7(btex  $\sum_{k=1}^r (-1)^k \binom{r+k-1}{k}$  etex);
for i:=1 upto 7:b[i].dx=b[i].dy;b[i].c=dir((360/7)*(i-1))*u;endfor;
boxit.b0(btex \box\arabbox etex);bo.dx=b0.dy;b0.c=origin;
drawboxed(b0,b1,b2,b3,b4,b5,b6,b7);
for i:=1 upto 7: for j:=1 upto 7:if i<>j:link(i,j);fi;endfor;
drawarrow b[i].c--b0.c cutbefore bpath b[i] cutafter bpath b0 dashed evenly;
endfor;
endfig;
end

```


3.2.3. Avantages

- Collaboration entre éléments graphiques et texte ;
- Langage très puissant.

3.2.4. Inconvénients

- Figure dans un fichier séparé ;
- Pas de récupération immédiate du contexte T_EX.

4. Substitutions dans un fichier PostScript : `psfrag`

L'extension `psfrag` permet de spécifier des substitutions de symboles dans une fichier EPS.

Pour utiliser `psfrag`, on procède comme suit :

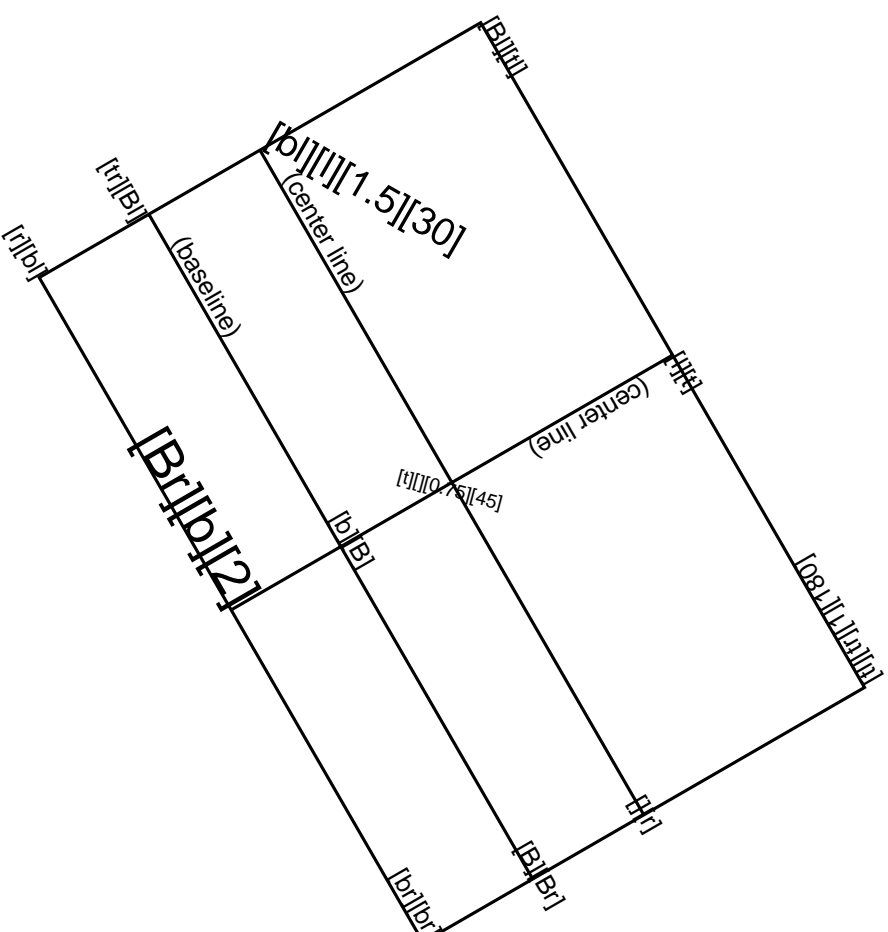
- inclure `\usepackage{psfrag}` dans le préambule ;
- dans le document, utiliser des commandes `\psfrag` pour spécifier le texte EPS à remplacer ;
- utiliser la commande `\includegraphics` normalement.

La syntaxe de `\psfrag` est :

```
\psfrag{<textePS>}[<posn>][<PSSposn>][<scale>][<rot>]{<texte>}
```

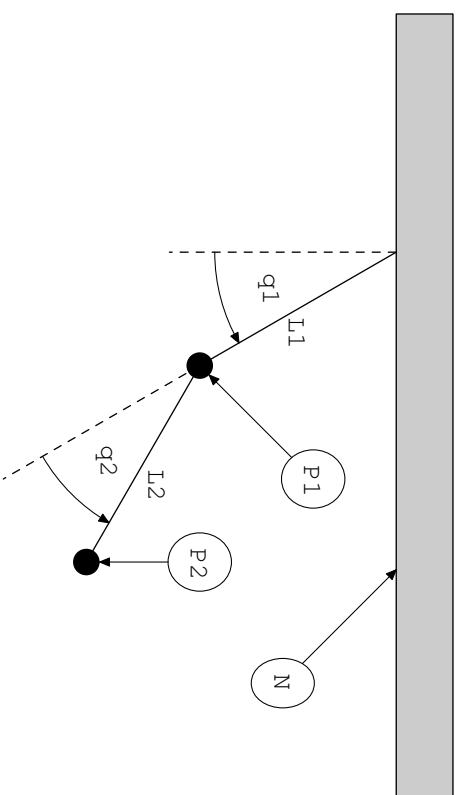
- *textePS* est le texte à remplacer dans le fichier EPS ;
- *posn* est la position de point de placement par rapport au nouveau texte ; par défaut, c'est `BL` ;
- *PSSposn* est la position de point de placement par rapport à l'ancien texte ; par défaut, c'est `BL` ;
- *scale* est le facteur d'augmentation de la taille ; il est préférable d'utiliser `\small`, `\large`, etc., dans les commandes `\psfrag` ;
- *rot* est l'angle de rotation des étiquettes, ce qui peut être utile avec des programmes qui ne permettent de créer des fichiers EPS que avec des étiquettes horizontales.

4.1. Options de placement



4.2. Exemple

`\includegraphics{pend.eps}` inclut le fichier `pend.eps` sans aucun remplacement, ce qui donne :



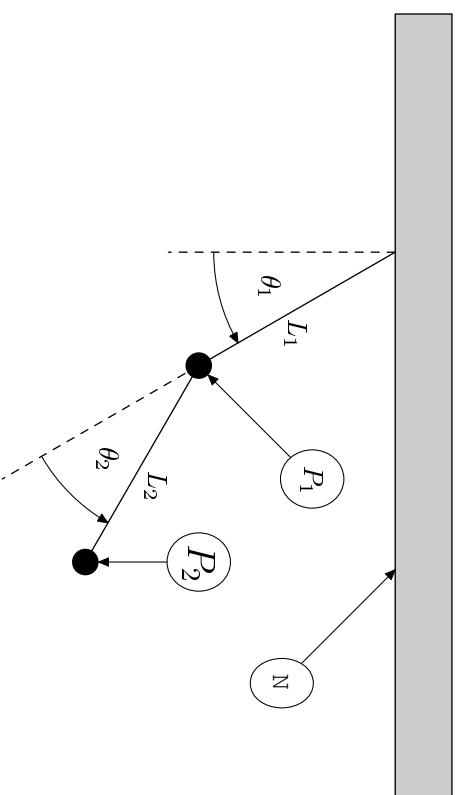
Toutefois, si l'on écrit :

```

\psfrag{q1}{ $\theta_1$ }
\psfrag{q2}{ $\theta_2$ }
\psfrag{L1}{ $L_1$ }
\psfrag{L2}{ $L_2$ }
\psfrag{P1}{ $P_1$ }
\psfrag{P2}{ $P_2$ }
\includegraphics{pend.eps}

```

on obtient :



- toutes les étiquettes n'ont pas besoin d'être remplacées ;
- `\psfrag{pi}{ π }` remplace π , mais pas $\pi/2$, etc.

5. Conclusion

- Lorsque les dessins sont effectués entièrement en `TEX`, tous les paramètres sont disponibles ; éléments graphiques et texte peuvent être harmonisés, mais traitement éventuellement lent (ex. : `XY-Pic`).
- Lorsque les dessins sont effectués par des outils extérieurs, l'homogénéisation des polices peut être atteinte, mais seul `METPOST` permet d'avoir une bonne collaboration entre éléments graphiques et texte ou formules.