

Cahiers **GUT** *enberg*

☞ METAPOST PRATIQUE

☞ Fabrice POPINEAU

Cahiers GUTenberg, n° 41 (2001), p. 167-175.

http://cahiers.gutenberg.eu.org/fitem?id=CG_2001__41_167_0

© Association GUTenberg, 2001, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

METAPOST pratique

Fabrice POPINEAU

SUPELEC – Campus de Metz
2, rue E. Belin
F-57070 Metz
Fabrice.Popineau@supelec.fr

Résumé. Ces quelques pages ont pour objectif de montrer comment utiliser pratiquement METAPOST. Ce programme est très différent des logiciels classiques de dessin, mais s'intègre extrêmement bien dans la chaîne de composition basée sur T_EX.

Abstract. *In this article, I will explain how to practically use METAPOST. This program is very different from usual drawing programs, but it fits very well in a T_EX based typesetting system.*

1. Où trouve-t-on METAPOST ?

METAPOST est aujourd'hui fourni avec toutes les distributions « complètes » de T_EX :

- T_ET_EX pour Unix,
- le CD-ROM T_EXLIVE pour Unix et Windows,
- M_IK_TE_X pour Windows,
- Oz_TE_X, C_Mac_TE_X (donc MacGUT), et avec T_ET_EX sous OS/X (en revanche pour diverses raisons, il faut T_ET_EX sous OS/X pour utiliser le format C_ON_TE_XT dont il sera question plus loin).

Je me baserai pour tout ce qui suit sur la distribution du T_EXLIVE 6 qui fonctionne sous Unix et Windows. Les exemples de graphiques sont empruntés au manuel de MetaFun écrit par Hans Hagen (voir plus bas).

Le programme principal (`mpost` ou `mpost.exe`) est accompagné de quelques programmes auxiliaires :

- `makempx` : le pilote qui va extraire le texte du fichier `.mp` et le convertir en commandes de bas niveau dans un fichier `.mpx`. Cette opération ne sera exécutée que si le fichier `.mpx` est plus ancien que le fichier `.mp`. Si c'est le cas, les deux programmes suivants seront utilisés pour effectuer la conversion.

- mpto : ce programme extrait les parties `btex . . . etex` et `verbatimtex . . . etex` du fichier `.mp` pour en faire un fichier \TeX
- dvitomp : convertir un fichier `.dvi` en un fichier `.mpx`.

La distribution de base comporte également un ensemble de fichiers de macros permettant entre autres de tracer des graphiques ou des diagrammes à base de boîtes et de cercles. Si votre système respecte l'architecture TDS, les fichiers de macros en question sont situés dans `texmf/metapost`. De même, vous trouverez la documentation et beaucoup d'exemples dans `texmf/doc/metapost` et `texmf/doc/guides/metapost-examples`.

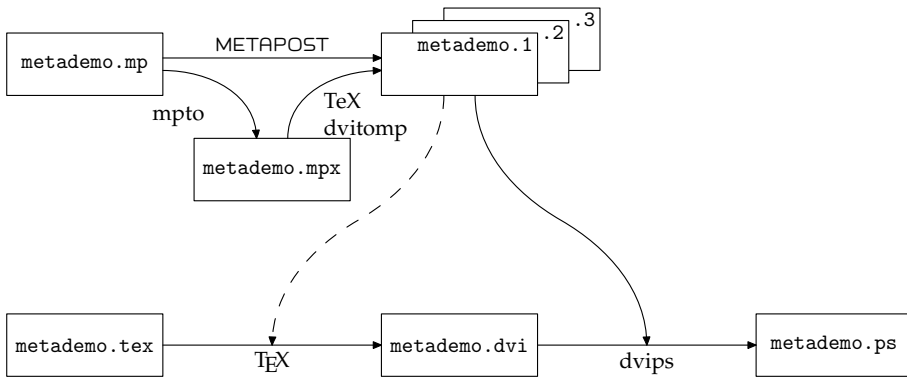


FIG. 1 – Compilation d'un fichier METAPOST

L'ensemble des fichiers source, de la documentation et des contributions à METAPOST est archivé sur le CTAN, par exemple à l'url suivante : <ftp://ftp.dante.de/pub/tex/graphics/metapost/>.

2. Utilisation de base avec \LaTeX

On partira de deux fichiers source :

metademo.tex pour le document \LaTeX ,

metademo.mp pour les graphiques composés par METAPOST.

Le fichier `metademo.mp` listé ci-dessous présente la structure générique d'un fichier METAPOST.

```

Structure 1 prologues:=2 ;
générique de
fichier      color yellow ; yellow = green + red ;
METAPOST.

```

```

5  def star (expr size, n, pos, color) =
    for a=0 step 360/n until 360 :
      draw (origin -- (size/2,0))
        rotatedaround (origin,a)
        shifted pos withcolor color ;
10  endfor ;
    enddef ;

    beginfig(1) ;
      pickup pencircle scaled 2mm ; star(2cm,5,origin, red) ;
15  endfig ;

    beginfig(2) ;
      pickup pencircle scaled 2mm ; star(2cm,7,origin, yellow) ;
    endfig ;
20  beginfig(3) ;
      pickup pencircle scaled 2mm ; star(2cm,11,origin, green) ;
    endfig ;

25  beginfig(4) ;
      pickup pencircle scaled 2mm ; star(2cm,13,origin, blue) ;
    endfig ;

    end

```

Le traitement de ce fichier par la commande ad-hoc va créer des fichiers POSTSCRIPT en nombre égal au nombre de figures décrites dans le fichier source :

```

c:\>mpost metademo.mp
This is MetaPost, Version 0.641 (Web2c 7.3.3.1)
(metademo.mp [1] [2] [3] [4] )
4 output files written: metademo.1 .. metademo.4
Transcript written on metademo.log.

```

Il faudra ensuite inclure ces figures dans le document d'origine. Si on utilise une version récente de \LaTeX , l'extension `graphicx` traitera les fichiers générés par METAPOST comme des fichiers POSTSCRIPT.

Si ce n'était pas le cas, on peut envisager de renommer les fichiers créés pour qu'ils aient l'extension `.eps` et puissent ainsi être traités par \LaTeX . Une autre solution consiste à ajouter une règle `\DeclareGraphicsRule` pour les fichiers METAPOST comme on peut le lire dans le fichier `metademo.tex` ci-dessous. Cette règle spécifie que les fichiers inclus par la commande `\includegraphics` dont l'extension n'est pas reconnue seront traités comme des fichiers POSTSCRIPT.

```

Inclusion de 1 \documentclass[a4paper, 11pt]{article}
graphiques \usepackage[T1]{fontenc}
METAPOST \usepackage[latin1]{inputenc}
ans un fichier \usepackage[frenchb]{babel}
LATEX. 5 \usepackage{graphicx}

% Inutile avec une version récente de LaTeX
% \DeclareGraphicsRule{*}{eps}{*}{}

10 \begin{document}

    Quelques étoiles extraites du manuel MetaFun~:\[4mm]

    \includegraphics[width=1cm]{metademo.1}
15 \hfill
    \includegraphics[width=1cm]{metademo.2}
    \hfill
    \includegraphics[width=1cm]{metademo.3}
    \hfill
20 \includegraphics[width=1cm]{metademo.4}

    \end{document}

```

Ensuite, il ne reste plus qu'à compiler le fichier source, puis à le passer au pilote `dvips`. Nous obtenons finalement le fichier POSTSCRIPT et le résultat ci-dessous.



Dans son comportement par défaut, METAPOST ne crée pas de code POSTSCRIPT auto-suffisant : entre autres, il se contente d'insérer des commentaires indiquant quelles polices sont utilisées et à quelle taille. Si vous utilisez au final le pilote `dvips` – et à l'exclusion de tout autre – il n'y a aucun problème. Si vous voulez utiliser le code POSTSCRIPT généré par METAPOST avec d'autres applications comme Adobe Illustrator par exemple, il faut modifier la variable `prologues` en tête de votre fichier `.mp` :

```
prologues := 2 ;
```

Cette variable vaut 0 par défaut. Une valeur strictement positive exige de METAPOST qu'il génère du code POSTSCRIPT auto-suffisant, autant que faire se peut. La valeur 1 est utilisée avec `troff`. En lui donnant la valeur 2, METAPOST ira chercher dans le fichier `psfonts.map` habituellement lu par `dvips`, quelles sont les polices réellement utilisées. Ces polices ne sont pas incluses dans le fichier généré par METAPOST, donc il

faudra quand même les fournir avec votre fichier sauf si elles sont déjà résidentes dans l'application cible. Évidemment, l'inconvénient de cette approche réside dans le fait qu'on est pratiquement certain d'avoir des problèmes de codage dès que l'on sortira du champ des caractères ASCII. Les vecteurs de codage pris en compte par `dvips` seront ignorés par une autre application.

3. METAPOST et PDF

METAPOST entretient des liens particuliers avec PDF. A priori, METAPOST produit des instructions POSTSCRIPT qui seraient incompatibles avec PDF. Mais le jeu d'instructions utilisé par METAPOST n'est qu'un sous-ensemble assez restreint du langage POSTSCRIPT et qui par chance est compatible avec les opérateurs PDF.

Le système `CONTEXT`¹ fournit un module nommé `supp-pdf` qui permet de convertir directement à partir de `TeX` les fichiers produits par METAPOST en instructions PDF. L'intérêt de ce module est d'être indépendant de `CONTEXT` et donc de pouvoir être utilisé dans d'autres environnements, comme `LaTeX`.

C'est ainsi que l'extension `graphicx`, lorsqu'elle est utilisée depuis un moteur dérivé de `PDFTeX`, reconnaît les fichiers `.mps` comme étant des fichiers contenant des graphiques METAPOST et charge `supp-pdf` pour convertir ces graphiques.

Concernant notre exemple, les fichiers graphiques n'ont évidemment toujours pas l'extension `.mps` et il faut donc cette fois-ci obligatoirement ajouter une règle par défaut pour inclure les graphiques METAPOST :

```
\usepackage{ifpdf}
...
\ifpdf
  \DeclaregraphicsRule{*}{mps}{*}{*}
\else
% Les versions récentes de LaTeX ne nécessitent pas cette règle
% \DeclaregraphicsRule{*}{eps}{*}{*}
\fi
```

On voit donc ici un intérêt majeur à utiliser METAPOST pour créer des graphiques vectoriels : *ils sont compatibles aussi bien avec une sortie POSTSCRIPT qu'avec une sortie PDF*. Un même fichier source servira pour les deux sorties, et on ne conservera pas deux versions des fichiers contenant les graphiques.

¹ `CONTEXT`, dont l'auteur principal est Hans Hagen, est un système complet de composition de textes qui se veut plus moderne et plus ambitieux que `LaTeX` tout au moins par certains de ses aspects. L'adresse <http://www.ntg.nl/context/> fournira plus d'informations sur ce système.

Si l'on veut convertir des graphiques créés avec METAPOST en PDF, CONTEXT fournit un utilitaire bien agréable nommé `mptopdf`. Il s'agit d'un script Perl² qui va automatiquement lancer les commandes nécessaires (`mpost` activé par `texexec`, puis `pdfetex`) à la compilation du fichier METAPOST et à sa traduction en PDF.

CONTEXT fournit également un nouveau format pour METAPOST nommé `metafun` qui charge des macros spécifiques en plus des macros d'origines METAPOST. Les effets autorisés par MetaFun sont assez saisissants comme le montre le graphique de la figure 2, qui affiche un corps de texte par-dessus un dégradé circulaire, dégradé allant du jaune au rouge et centré sur le centre du rectangle englobant le texte.

Ombrage circulaire

FIG. 2 – Ombrages avec MetaFun.

4. Intégration des graphiques dans le document source

Grâce à la primitive `\write18` du système WEB2C, il a été possible de rendre l'inclusion des graphiques METAPOST transparente : la compilation des graphiques est lancée par le moteur T_EX lors de la compilation du document principal.

Attention : pour des raisons de sécurité, la commande `\write18` est inopérante par défaut. Pour la rendre opérante, il faut soit lancer le moteur T_EX avec l'option `-shell-escape`, soit modifier son fichier `texmf.cnf` pour y trouver :

```
% Enable system commands via \write18{...}?
shell_escape = t
```

CONTEXT fut le premier environnement à exploiter intensivement la possibilité d'intégrer des figures METAPOST dans le corps du document source. Le code de la figure 2 est reproduit dans le listing qui suit.

```
Figures 1 \definecolor[a][yellow]
METAPOST  \definecolor[b][s=0.8]
encapsulées \definecolor[c][darkred]
dans un 5 \startuniqueMPgraphic{CircularShade}
document
```

² En fait, le script Perl est accompagné d'un format réduit `mptopdf.efmt` qu'il faut générer auparavant et qui sera utilisé par `pdfetex`.


```

path p ;
p := unitsquare xscaled \overlaywidth
      yscaled \overlayheight ;
circular_shade(p,0,\MPcolor{a},\MPcolor{c}) ;
10 \stopuniqueMPgraphic

\defineoverlay[circular shade]
      [\uniqueMPgraphic{CircularShade}]
\framed[background=circular shade,frame=off]
15 {\bf Ombrage circulaire}

```

L'intérêt de `CONTEXT` réside dans des possibilités extensives d'échanges entre `METAPOST` et `TEX` : `TEX` peut passer des paramètres à `METAPOST` car les macros sont expansées avant la génération du fichier `METAPOST`. `CONTEXT` fournit même un mécanisme très puissant de `\special` de position qui permet de créer des graphiques qui vont parfaitement encapsuler du texte composé par `TEX`. Ceci s'obtient en effectuant plusieurs passes, la première générant un fichier contenant des informations de position sur le texte visé, puis en appelant `METAPOST` pour générer l'image ad-hoc, et enfin en recomposant l'ensemble.

Cette caractéristique assez plaisante de `CONTEXT` qui consiste à fusionner dans le même document figures et texte, a été portée assez tardivement sous `LATEX`, mais est dorénavant disponible grâce à l'extension `emp` (*Embedded MetaPost*). On peut utiliser cette extension pour produire un document analogue au précédent comme le montre l'exemple du listing ci-dessous.

```

1 \documentclass[a4paper,11pt]{article}
  \usepackage{palatino}
  \usepackage[T1]{fontenc}
  \usepackage[latin1]{inputenc}
5 \usepackage{mflogo,graphicx,emp,ifpdf}

\ifpdf
  \DeclareGraphicsRule{*}{mps}{*}{}
\fi
10 \begin{document}

% Commandes incluses dans l'entête du
% fichier metapost
15 \empaddtoTeX{%
  \usepackage{palatino}
  \usepackage[T1]{fontenc}
  \usepackage[latin1]{inputenc}
}

```

Utilisation
package em

```

20  % Début des figures metapost
    \begin{empfile}
      % Définitions génériques
      \begin{empcmds}
25    color yellow ; yellow := red + green ;
      \input mp-spec ;
      \end{empcmds}
      % première figure metapost
      \begin{empdef}[fig1](5cm,5cm)
30    draw fullsquare          withcolor .625red ;
      draw fullsquare rotated 45 withcolor .625red ;
      picture cp ; cp := currentpicture ;
      def copy = addto currentpicture also cp enddef ;
      copy scaled .9 withcolor .625white ;
35    copy scaled .7 withcolor .625yellow ;
      copy scaled .6 withcolor .625white ;
      copy scaled .4 withcolor .625red ;
      copy scaled .3 withcolor .625white ;
      fill fullcircle scaled .2 withcolor .625yellow ;
40    currentpicture := currentpicture scaled 50 ;
      \end{empdef}
    \end{empfile}
    % Compilation au vol du fichier metapost
    \immediate\write18{mpost -tex=latex \jobname}
45
    Un exemple de figure \MP{} incluse dans le corps
    d'un document \LaTeX{} :
    \begin{figure}[ht]
      \begin{center}
50    \empuse{fig1}
      \caption{Un logo avec \MP{}}
      \end{center}
    \end{figure}
55 \end{document}

```

Malheureusement, contrairement à `CONTEX`T, l'utilisation du package `emp` n'autorise pas l'échange de paramètres entre `TEX` et `METAPOST`.

5. Conclusion

J'espère que ces quelques pages auront éliminé les difficultés pratiques liées à l'utilisation de METAPOST et que vous pourrez vous adonner aux joies de la géométrie à l'aide de cet outil ! Bonne chance.