

Cahiers **GUT**enberg

☞ LES FICHES CUISINE D'ONC' POSTSCRIPT —
FICHE 1 : UNE TABLE TRAÇANTE

☞ Bruno BORGHI

Cahiers GUTenberg, n° 1 (1989), p. 64-66.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989__1_64_0>

© Association GUTenberg, 1989, 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.

Les fiches cuisine d'Onc' PostScript

Fiche N° 1 : une table traçante

B. BORGHI

Metasoft - Rennes

PostScript, c'est formidable, ça permet d'écrire du texte en spirale avec des caractères plus rigolos les uns que les autres, et tout un tas de trucs épatants. C'est vrai.

Le problème, c'est que les applications dans leur très grosse majorité s'intéressent à des problèmes graphiques beaucoup plus terre à terre : tracer des traits présentables, avec du texte à l'horizontale.

Pour cette première fiche cuisine, Onc' PostScript va donc vous parler de l'application à laquelle a été confronté tout programmeur PostScript : connecter une imprimante PostScript à un logiciel à la manière d'une table traçante. Tous les programmes de conversion de sortie type Tektronix, HPGL ou Benson font partie de cette famille.

Le problème est moins simple qu'il n'y paraît, et les solutions que nous allons voir ensemble peuvent s'adapter à des cas plus ambitieux.

1. Une table traçante élémentaire

Nous nous intéressons donc à une application qui génère des fichiers d'impression faisant appel à 4 types de fonctions : initialisation, déplacement plume levée, tracé de segment plume baissée, fin de tracé. Nous supposons que l'application utilise des coordonnées en millimètres.

Pour aujourd'hui, nous allons faire une hypothèse supplémentaire restrictive : le

nombre de segments consécutifs est inférieur à 1400. Cette hypothèse nous permet d'ignorer les limites d'implémentation de PostScript¹.

Plutôt que de générer des coordonnées dans le repère par défaut de PostScript, il est souvent plus efficace de faire un changement de repère pour mettre l'imprimante dans le système de coordonnées de l'application. On évite ainsi beaucoup de multiplications, de chiffres après la virgule et d'erreurs d'arrondi. La programmation la plus immédiate donne des fichiers générés ayant à peu près cette tête :

```
%% Pen Plotter - Version 1
% Init
2.8346 dup scale % unit = mm
0.1 setlinewidth
20 20 translate % adjust origin
0 0 moveto
% Curve
100 120 moveto
54 25 lineto
66 27 lineto
42 23 lineto
stroke
30 125 moveto
30 225 lineto
stroke
showpage
```

Cette version présente un inconvénient : elle est plutôt bavarde. L'utilisation des identificateurs natifs de PostScript augmente artificiellement le volume du fichier et par conséquent les temps de

¹ On peut supprimer cette hypothèse, mais c'est un peu long. Nous réglerons ce problème dans la prochaine fiche cuisine d'Onc' PostScript.

transfert sur l'imprimante. Quelques procédures bien senties permettent de diminuer ce volume.

```

%% Pen Plotter - Version 2
% HEADER
  /bop { % - bop -
    % Beginning Of Page
      2.8346 dup scale
      0.1 setlinewidth
      20 20 translate
      0 0 moveto } def
  /u { % x y u -
    % pen Up
      stroke moveto } def
  /d { % x y d -
    % pen Down
      lineto } def
  /eop { % - eop -
    % End Of Page
      stroke showpage } def
% BODY
bop
100 120 u
54 25 d
66 27 d
42 23 d
30 125 u
30 225 d
eop

```

On gagne ainsi 5 caractères par ligne soit une diminution de volume d'environ 30 %. Il est inutile de réduire la longueur des identificateurs des procédures appelées peu fréquemment : le gain est infime et on y perd en clarté.

Une légère retouche à cette version permet de diminuer le temps d'exécution (à partir de maintenant, seules les procédures modifiées sont réécrites dans les exemples).

```

%% Pen Plotter - Version 3
/u {stroke moveto} bind def
/d /lineto load def

```

`bind` prend en argument une procédure dans laquelle il remplace tous les

identificateurs d'opérateurs par les opérateurs eux-mêmes. Ainsi, à l'exécution, la recherche de ces identificateurs dans la pile de dictionnaires n'est plus nécessaire. `bind` agit donc très exactement comme un éditeur de liens sur les procédures.

Plutôt que de créer une procédure réduite seulement à un opérateur, comme c'était le cas dans la version 2 de `d`, il est aussi simple et plus efficace d'associer directement le nom avec l'opérateur. La version 3 utilise `load` pour charger sur la pile l'opérateur `lineto`, qui est ensuite associé au nom `d`. Au lieu de faire un appel de procédure, une recherche dans la pile des dictionnaires et un appel d'opérateur à chaque exécution de `d`, on ne fait plus qu'un appel d'opérateur.

2. Changements d'échelles

Intéressons nous maintenant à une table traçante un peu plus évoluée : on peut changer le point origine et choisir des facteurs d'échelle en `x` et en `y`. L'origine et les facteurs d'échelle sont spécifiés dans le repère par défaut de la table traçante.

Une procédure supplémentaire est nécessaire.

```

%% Pen Plotter - Version 4
% HEADER
  /setspace { % x y sx sy setspace -
    % x y origin in plotter default space
    % sx sy scaling factors
      4 2 roll translate scale
      0 0 moveto
    } def
% BODY
bop 20 20 1.5 1.5 setspace

```

Conceptuellement, la translation est effectuée avant le changement d'échelle. Cela explique l'ordre des arguments de `setspace` et la nécessité du `roll`. L'argument d'efficacité d'exécution n'est pas do-

minant, car la procédure est appelée peu fréquemment.

Si vous essayez ce programme, vous verrez rapidement que *l'épaisseur du trait des courbes dépend des facteurs d'échelle*, alors qu'elle est constante dans toute table traçante qui se respecte. En particulier, si les facteurs d'échelle sont distincts en x et en y , l'épaisseur des traits varie selon l'orientation².

Nous devons considérer deux espaces distincts : l'espace par défaut de la table traçante dans laquelle est défini l'épaisseur des traits, et l'espace de définition des parcours qui a pu subir des déformations. Il faut modifier la procédure `u` de façon à se trouver dans le repère par défaut de la table traçante à chaque appel de `stroke`. Voici pour terminer la version complète du préambule de notre table traçante.

```

%% Pen Plotter - Version 5
/bop {      % - bop -
% Beginning Of Page
  2.8346 dup scale
  % default unit is mm
  0.1 setlinewidth
  20 20 translate
  % adjust origin
  0 0 moveto } def
/setspace { % x y sx sy setspace -
% x y origin in plotter default space
% sx sy scaling factors
  4 2 roll translate scale
  0 0 moveto } def
gsave
  bop
  /M matrix currentmatrix def
  % plotter default space matrix
grestore
/u {      % x y u -
% pen Up
  gsave
    M setmatrix stroke
  grestore
  newpath moveto
} bind def
/d /lineto load def

```

```

% pen Down
/eop {    % - eop -
% End Of Page
  0 0 u showpage } def

```

La matrice `M` est la matrice du repère par défaut de la table traçante. Sa définition est encadrée par `gsave` et `grestore` afin que l'exécution de `bop` dans le préambule ne modifie pas l'état de l'imprimante. Vous remarquerez l'emploi de `u` dans `eop` pour que le dernier trait soit conforme.

3. Conclusion

On acquiert sans effort les réflexes qui sauvent par l'étude approfondie des problèmes élémentaires.

Grâce à Onc' PostScript et à sa fiche cuisine d'aujourd'hui, vous avez pu voir ou revoir quelques principes de programmation PostScript de portée générale.

- La définition de procédures correspondant au modèle conceptuel de l'application plutôt qu'à celui de PostScript est essentielle pour la clarté, la simplicité et l'efficacité des fichiers générés (principe de *l'imprimante virtuelle*).
- Les identificateurs les plus employés dans le corps du fichier peuvent être réduits à un caractère. On diminue ainsi le volume du fichier et le temps de transmission.
- Des mécanismes existent pour accélérer l'exécution : `bind` et `load`. Utilisons les à bon escient.
- Attention aux systèmes de coordonnées : le repère de définition du parcours n'est pas forcément le repère de tracé.

²On rencontre également ce problème quand on décrit des ellipses avec `arc`.