

Cahiers **GUT** *enberg*

☞ UN REGARD SUR LES PIXELS. OBTENTION
DE FONTES DE QUALITÉ POUR IMPRIMANTES
À LASER À 300 DPI GRÂCE À METAFONT

☞ Pierre A. MacKAY

Cahiers GUTenberg, n° 12 (1991), p. 21-35.

http://cahiers.gutenberg.eu.org/fitem?id=CG_1991__12_21_0

© Association GUTenberg, 1991, 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.

Un regard sur les pixels. Obtention de fontes de qualité pour imprimantes à laser à 300 dpi grâce à METAFONT*

Pierre A. MACKAY

University of Washington
Mail Stop DR-10
Seattle, WA 98195, USA
email: mackay@cs.washington.edu

Résumé

Si l'on imprime le même document sur deux imprimantes à laser au même standard industriel de 300 dpi, on peut obtenir des résultats différents selon que ces imprimantes sont du type BPN ou NPB. La plupart des systèmes de création ou de tracé de caractères semblent préférer les BPN et on peut, avec quelque raison, penser que les NPB ne seront jamais vraiment satisfaisants. Quoiqu'il en soit, il est bon que le créateur d'une police de caractère qui envisage une large utilisation sur des imprimantes à 300 dpi soit au courant des difficultés causées par l'utilisation d'un dessin unique pour des technologies différentes.

Abstract

Two otherwise identical documents printed at the industry quasi-standard medium resolution of 300 dots/inch on laser printers can appear very different depending on whether a "write-black" or a "write-white" engine was used to print them. Most font-design and font expression systems appear to favor "write-black" technology, and there is some reason to suspect that "write-white" will never be entirely satisfactory. In any case, it is a good idea for the designer who expects to see a great deal of 300 dots/inch output to be aware of the difficulties involved in trying to support both technologies with the same design.

*Cet article est paru sous le titre « *Looking at the Pixels. Quality Control for 300 dpi Laser Printer Fonts, Especially METAFONTS* » dans *Raster Imaging and Digital Typography II* (R. MORRIS & J. ANDRÉ eds.), Cambridge University Press, 1991, 205-217. Il est reproduit ici avec l'autorisation de l'auteur et celle de David TRANAH, éditeur à Cambridge University Press et nous tenons à les en remercier. Traduction française de Jacques ANDRÉ, Alain COUSQUER et Éric PICHERAL. En tant que traducteurs nous avons été amenés à faire un certain nombre de choix que nous signalerons en général soit par la mention du terme anglais entre parenthèses et en italique, soit par une note en bas de page suivie de la mention {Ndt} : *Note des traducteurs*. Cette même abréviation signale aussi des ajouts que nous faisons par rapport à l'article original. Mise en page avec l'aide de Philippe LOUARN.

1. Contours et matrices de points

Les imprimantes à laser de moyenne définition produisent les caractères sous forme de matrices de points (*bitmaps*) disposés linéairement dans un tableau. Quel que soit le processus de dessin utilisé en amont, on termine toujours par une matrice de pixels formellement discrets qui sont soit colorés (en noir) soit laissés incolores (blanc). La plupart des systèmes actuels de définition de caractères choisissent les pixels à colorier en déterminant s'ils se trouvent entièrement (ou principalement) à l'intérieur ou à l'extérieur d'une courbe fermée placée sur cette grille. Nous considérons dans cet article deux systèmes de contour, METAFONT et PostScript¹. Tous deux sont basés sur les courbes de Bézier et, pour les caractères des alphabets latins, ils procèdent, en général, par remplissage de courbes définies par leurs contours. METAFONT emploie surtout la notion de « plume » (*pen*). Si cette notion est particulièrement adaptée pour certains alphabets et styles de caractères, il est beaucoup plus difficile de l'utiliser pour générer les fontes de caractères latins relevant plus du travail des graveurs que de celui des calligraphes. La quasi-totalité des traits (*stroke*) des caractères des polices latines les plus connues générées par METAFONT sont des contours remplis tandis que les caractères créés par PostScript sont presque exclusivement décrits en termes de contours. La capitale A est composée, en METAFONT, d'au moins quatre contours remplis superposés les uns aux autres : un seul pour les deux diagonales, un pour la barre et un pour chacun des deux empattements (*serifs*) en bas des diagonales. Le même caractère est défini, en PostScript, par une courbe fermée entourant toute la partie noire du caractère et une seconde courbe fermée entourant la partie intérieure blanche c'est-à-dire le contre-poinçon (*counter*).

Malgré ces différences mineures, les deux systèmes rencontrent en fin de compte le même problème : le tracé exact de chaque caractère doit finalement être transformé en une matrice de pixels avant d'être transféré sur le papier. La différence porte sur le moment et l'endroit où se fait la transformation. METAFONT la réalise avant d'envoyer l'image du caractère à l'imprimante tandis que PostScript la fait (généralement, mais pas toujours) dans un processeur résidant dans l'imprimante. L'utilisateur de contours

¹On se réfère ici surtout à METAFONT car, d'une part, c'est le système le mieux connu de l'auteur et, d'autre part, METAFONT permet d'étudier directement les matrices de points et ce de façon simple. On ne juge pas utile d'étudier ici d'autres systèmes utilisant des contours remplis, comme par exemple le système *Purdy-Macintosh* (basé sur des spirales) choisi par *Varityper*, car de toute façon le problème reste le même quelle que soit l'approche utilisée pour remplir les courbes.

PostScript doit accepter que cette transformation ne lui soit pas visible ; par contre, l'utilisateur de fontes METAFONT peut voir exactement les matrices de points que le programme va déposer sur le tambour électrostatique de l'imprimante pour être transférés sur le papier. METAFONT écrit ses trames de sortie (*raster out*) dans un format connu sous le nom de GF (*Generic Font*)² ce qui nous permet au moins d'étudier directement les arrangements de pixels. Cependant, le fait que nous puissions regarder ces pixels ne veut pas dire que le contrôle du positionnement de ces pixels soit facile, même en METAFONT. Dans le manuel de base METAFONT [Knuth86], au chapitre 24 (*Discreteness and Discretion*), Knuth analyse les problèmes en détail et compare certaines techniques spécifiques d'arrondi garantissant la symétrie et la cohérence de certains éléments de lettres qui, sinon, pourraient paraître irrégulières. À basse définition, il faut introduire des distorsions sensibles dans les contours ; il faut alors procéder à de légers déplacements de certains éléments de caractère ce qui pourra surprendre le dessinateur inexpérimenté. PostScript est confronté au même problème et le résoud probablement de la même façon, mais on n'a pas la possibilité d'étudier la matrice de points résultante. Ce problème est partiellement étudié dans [Knuth90] ; on en trouvera une discussion plus approfondie dans [Hobby90].

La définition de 300 dpi, qui est pratiquement une norme pour les imprimantes à laser du commerce, ne produit pas une matrice assez fine pour figurer les détails. Un caractère en corps 10 est réalisé, pour une imprimante à 300 dpi, avec une définition³ de seulement 41 pixels par cadratin (*em*). Un grossissement (*magnification*) d'un facteur de 1,2 (comme cela se fait de façon normale en T_EX et METAFONT) permet d'obtenir 50 pixels par cadratin, ce qui explique l'augmentation de la qualité obtenue en imprimant avec un grossissement de 1,2 et en ramenant photographiquement le texte à la taille voulue. Mais ces deux définitions sont loin des 360 pixels par cadratin que voit un dessinateur sur les épreuves fournies par METAFONT pour 2602 dpi⁴.

Nous avons toutes les raisons d'espérer que la forme d'un caractère produit par METAFONT en mode épreuve, après avoir été transformée par le programme `gftodvi`, montre fidèlement les proportions du caractère choisi

²Voir à ce sujet A. COUSQUER & É. PICHERAL, « Polices, T_EX et Cie », *Cahiers GUTenberg*, 9, juillet 1991, 3-31. {Ndt}.

³*Resolution* signifie en effet « définition » en français (on parle par exemple de la définition d'un appareil d'optique) {Ndt}.

⁴La valeur 2602 peut paraître étrange : c'est en fait le résultat du produit $36 \times 72,27$ (où 72,27 est le nombre de points typographiques dans un pouce) ; c'est une définition courante dans les milieux professionnels.

et qu'une véritable photocomposeuse – nous appelons ainsi tout appareil avec une définition d'au moins 1200 dpi – produira un caractère dont la forme aura la même apparence que celle de l'épreuve. Par contre, à 300 dpi, on a de fortes chances d'avoir des surprises, parfois désagréables. Non seulement la définition n'est pas suffisante pour distinguer ce qui caractérise une bonne fonte pour photocomposeuse, mais de plus le mécanisme de dépôt du toner sur le tambour en vue du transfert sur le papier pose d'autres problèmes.

2. Imprimantes BPN ou NPB

On croit naïvement que le tambour d'une imprimante laser servant aux sorties d'ordinateur fonctionne comme un appareil BPN⁵, c'est-à-dire que la surface du tambour est mise « en blanc » au début du traitement de chaque page, puis que, dans cette région blanche, des zones sont chargées électrostatiquement de façon à attirer les particules noires du toner qui forment ainsi l'image dans la page. En fait, les tambours utilisés dans les photocopieurs fonctionnent le plus souvent de façon inverse ! Dans un copieur, l'effet de la lumière réfléchie depuis les régions blanches de l'original est d'agir sur la surface du tambour qui est donc, au départ, complètement noire⁶. L'effet de la lumière est de détourner les caractères en supprimant des plages noires pour ne laisser que les seules parties qui concernent les caractères. Les fabricants d'imprimantes laser – elles sont toutes construites à partir de copieurs – ont tendance à utiliser cette sorte de tambours, même pour les imprimantes d'ordinateurs. On appelle NPB une telle imprimante⁷.

On représente habituellement les images discrètes à moyenne définition des imprimantes en utilisant un carré ou un rectangle par pixel. Si les

⁵Il nous a été impossible de trouver une traduction « officielle » des expressions *write-black* et *write-white*. Aucune des expressions « écrit en noir/écrit en blanc », « peint le blanc/peint le noir », « négative/positive », « noircissant/blanchissant », « coloriant/détourant », « blanc au noir/noir au blanc » ne convient vraiment. Nous avons finalement préféré employer les abréviations BPN (blanc puis noir) pour *write-black* et NPB (noir puis blanc) pour *write-white*. Elles ont l'avantage, outre le fait de correspondre aux habitudes françaises d'user (voire d'abuser !) d'abréviations, de pouvoir s'utiliser comme substantif, adjectif ou participe. {Ndt}.

⁶Mais pas encore recouverte de toner {Ndt}.

⁷Pierre Mac Kay nous a apporté la précision suivante : « Toutes les LaserWriters sont basées sur les copieurs CanonCX ou CanonSX et sont du type BPN. Les exemples les plus classiques d'imprimante du type NPB sont les imprimantes LN01 et LN03 de Digital Equipment, les machines Ricoh et certaines imprimantes de chez Xerox. Tous les écrans qui utilisent des caractères noirs sur fond blanc sont du type PNB. Le *previewer* Xdvi, dans le monde d'Unix X11, donne des résultats pratiquement illisibles avec des fontes BPN et acceptables avec des fontes NPB. On peut, sous X11, comparer les deux styles dans deux fenêtres voisines, et le résultat est plutôt convaincant ! » {Ndt}.

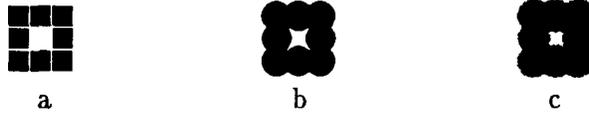


Figure 1. Trois représentations de pixels BPN

appareils pouvaient déposer des particules de toner de cette forme, il n'y aurait pas besoin de considérer la différence entre les machines NPB et BPN. Hélas, la technologie ne le permet pas : les plages noircies ou blanchies par la poudre ressemblent plus à un cercle qu'à un carré, cercle dont la surface est généralement plus grande que celle voulue, sauf si le faisceau du laser est particulièrement fin et convergent.

Si nous avons des pixels carrés, un rectangle de 3×3 avec un trou au milieu devrait alors ressembler à la figure 1a. Cela ne sera jamais le cas mais, si le dépôt du toner dans un appareil BPN était parfait, le rectangle pourrait ressembler à la figure 1b. Elle est déjà plus noire que la figure avec des carrés, mais moins noire que le résultat réel attendu sur papier. Le toner se répand inévitablement en dehors du cercle idéal et la tache BPN, si on la suppose à peu près circulaire, ressemble plus à ● ou à ● qu'à ●. La véritable figure BPN sera même plus noire (figure 1c), du fait des pertes de charge ([Rubinstein88] p. 79–80) et du fait d'interactions physiques de surface des particules de toner aussi bien quand elles sont encore détachées que quand elles sont agrégées par la chaleur de la fusion. Le centre d'un carré (comme celui de la figure 1c) peut alors très bien disparaître.



Figure 2. Pixels d'un appareil NPB

Avec un appareil NPB, l'épaisseur des lignes est atténuée tandis qu'un trou central est agrandi (figure 2). Si le pixel carré idéal avait un côté de 1, le « cercle » NPB devrait avoir au moins un diamètre de $\sqrt{2}$ de façon à ne pas laisser de taches noires régulièrement disposées sur le tambour. S'il était plus petit, du toner pourrait adhérer sur le tambour à l'intersection des quatre cercles (figure 3). De telles petites mouchetures ont bien sûr tendance à disparaître de l'image finale, mais ce n'est pas un effet sur lequel on peut compter ([Rubinstein88] p. 80-1; [Knuth86] p. 33; [Billawala87] p. 30).

Sur le tambour d'un véritable moteur d'impression, le bord des ronds blancs ne sera sûrement pas aussi net mais passera graduellement du gris



Figure 3. Tambour NPB avec des cercles de détournage trop petits

au noir le long des zones noires⁸. Le long des bords, la perte d'intensité du toner noir peut donc être plus grande que ne le montrent ces exemples. De plus, le nettoyage d'un tambour NPB peut être fait en déplaçant un point par balayage ; les petites dents au dessus et au dessous de la ligne risquent alors d'être perdues. Tout se combine et les images sont atténuées en mode NPB.

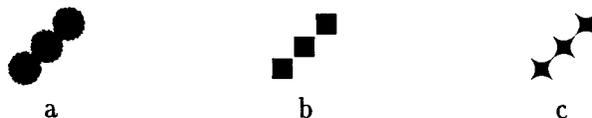


Figure 4. Une diagonale en mode BPN, une idéale et une en mode NPB

L'effet est déjà sensible dans un motif orthogonal de pixels, mais il est encore plus visible dans une ligne diagonale. C'est d'ailleurs dans ces diagonales que les appareils NPB font le plus de dégâts pour des caractères prévus pour être imprimés avec des appareils BPN. L'image BPN de la figure 4a a une surface plus de trois fois supérieure à l'image NPB de la figure 4c. La surface de l'image NPB est la moitié de celle de l'image du centre et, qui plus est, cette image a même perdu la continuité qui existe de façon ténue en 4b. Cette forme ressemble plus à la juxtaposition de trois pixels qu'à une ligne continue. Des particules de toner isolées ont tendance, comme nous l'avons dit plus haut, à disparaître du papier et les pixels NPB d'une diagonale sont alors vraiment séparés.

Tout ceci explique ce qui arrive aux diagonales quand on utilise, avec un appareil NPB, une fonte prévue pour un appareil BPN.

On peut afficher la fonte générique GF (*Generic Font*) produite par METAFONT avec un agrandissement tel que l'on puisse étudier en détail les matrices de points. Le programme de mise au point `gftype` (c'est l'un des divers utilitaires de METAFONT) montre les images tramées d'un fichier GF à basse définition. On n'utilise normalement ce programme que pour valider l'organisation et le format d'un fichier GF ; ce programme ne peut pas être utilisé pour les hautes définitions car l'image des pixels sortirait du cadre

⁸On aurait pu utiliser des demi-teintes (*halftone*) pour le bord des pixels NPB dans la fonte que nous employons pour ces illustrations, mais l'effort n'en vaut pas la peine.

normal du papier. Mais, même à basse définition, les images produites par `gftype` ne sont utiles que de façon marginale. Les sorties brutes faites par `gftype` ont les limitations suivantes :

1. Les pixels sont représentés par des astérisques ce qui rend l'image dépendante des proportions hors-tout de la cellule des caractères d'une imprimante ou d'un écran donné ce qui produit, inévitablement, une distorsion dans la représentation du bitmap du caractère.
2. Il n'y a que peu, voire pas, d'indications des effets produits par un changement de la technique d'impression.

```

*****      *****  Le paquet de macros gfigure.tex de TEX
***          ***      convertit les fichiers en mode image gftype
***          *         en images ramenées à 300 dpi en utilisant
***          *         la fonte pixeldot9. Les macros gfigure.tex
***          *         prennent les images gftype exactement comme
***          **        elles sont produites par le programme et les
***          **        reformatent automatiquement en se basant sur
***          *         la forme particulière du fichier de sortie gftype.
***          ***       On peut faire en sorte que l'impression soit
***          * ***     réalisée avec des pixels carrés BPN ou NPB. Une
***          * ***     fonte à 300 dpi est utilisée par gftype avec la
****         ***      commande (UNIX) :
***          ***
***          ****      gftype -i cmr10.300gf > cmr10.300i
***          ***
***          ***      dont le fichier de sortie est utilisé par les
***          ***      commandes TEX suivantes
***          ***
***          ****      \image cmr10.300i(white)
***          ***
*****      *****  pour produire une image photocomposée de la
gftype image--cmr7  fonte entière.

```

La meilleure façon de montrer l'emploi des macros `gfigure` est de les appliquer à un vrai caractère. La majuscule 'K' de la fonte `cmr7` est l'une des plus affectées par un changement de technique d'impression. La figure 5 en montre quatre aspects différents.

⁹Les sources de `gfigure.tex` et de `pixeldot.mf` doivent être publiés dans un prochain numéro de *TUGboat*.

METAFONT output 1990.12.09:1202 Page 1 Character 75 "The letter K"

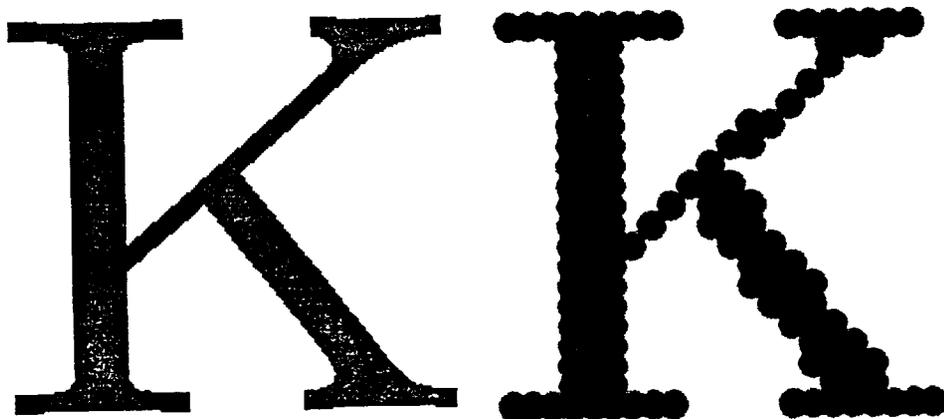
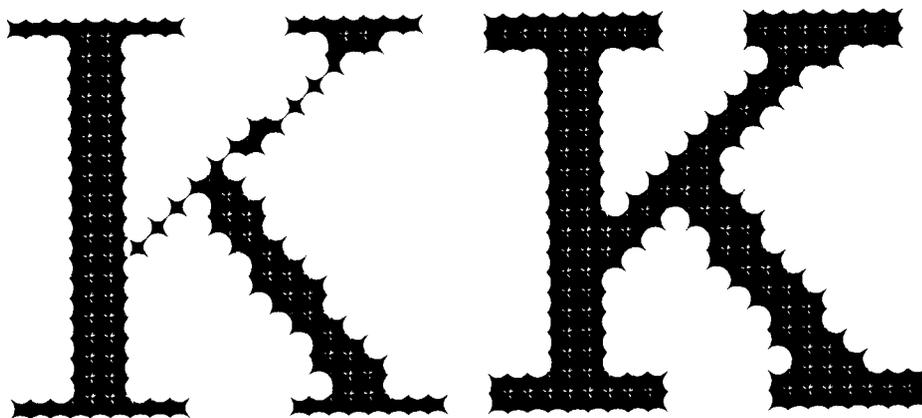


Image BPN vue par un engin BPN



Images BPN et NPB vues par un engin NPB

Figure 5. Capitales K de Cmr7 : épreuve et images à 300 dpi

L'image en haut à gauche est une épreuve produite par METAFONT après suppression de la grille et des points de repère. Celle en haut à droite est l'image `gftype` du même caractère reformatée après compilation pour une machine BPN à 300 dpi. Elle est raisonnablement proche de l'idéal avec une seule anomalie de direction due au fait que la diagonale supérieure traverse la matrice avec un angle juste inférieur à 45° ce qui produit un pli juste au dessus de la jonction avec la diagonale inférieure. Il se trouve que justement METAFONT a produit, à ce point, un trou d'encre (*ink trap*) de chaque côté de la diagonale descendante de façon que l'encre du toner n'aille pas couler dans ces angles et les boucher alors qu'on veut qu'ils soient aigus¹¹.

L'image en bas à gauche semble acceptable en ce qui concerne les horizontales et les verticales (encore que les empattements soient plutôt anémiques) mais la diagonale supérieure est, de toute évidence, inacceptable. L'image grossie montre que les pixels isolés sont minces – en pratique ils disparaîtront le plus souvent pour ne laisser qu'un pâté au dessus de la jonction avec la diagonale inférieure. Ici, les deux trous à encre ne sont pas seulement inutiles, ils tendent à exagérer la minceur du caractère.

Il faut améliorer ce point ; il est alors intéressant de regarder aussi bien ce qui ne fonctionne pas que la solution générale.

METAFONT adapte ses images numérisées à un matériel d'impression donné en affectant des valeurs à un ensemble de paramètres définis dans la macro `mode.def`. Les instructions qui nous intéressent ici sont les suivantes¹² :

```
blacker:=0.0;  
fillin:=0.0;
```

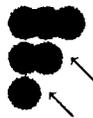
Si l'on veut noircir une image excessivement claire, la première chose à faire est d'accroître la valeur *blacker*, qui augmente la largeur des incréments des pixels entiers ou fractionnaires pour chaque élément du caractère, mais cette action est trop globale. Elle accroît chaque caractéristique d'une quantité liée à la taille d'un pixel et si on veut obtenir l'effet désiré pour n'importe quel trait d'un pixel d'épaisseur dans n'importe quelle direction, il peut être nécessaire de l'augmenter jusqu'à une valeur telle qu'un pixel entier soit

¹¹Dans une imprimante à laser, ce phénomène est mieux décrit, électrostatiquement, comme une perte de charge ([Rubinstein88], p. 79–80).

¹²Une troisième instruction, fixant le pourcentage de dépassement pour la correction optique (*overshoot*) des rondes comme 'o', ne nous intéresse pas ici puisqu'il faut quelque chose comme 80 pixels/cadratin de définition avant de commencer à faire le moindre dépassement.

ajouté à chaque trait du caractère. Cela peut se faire, en théorie du moins, en donnant à *blacker* la valeur 1, en fonction de l'endroit où sont situés les bords des traits par rapport à la grille. Ainsi, sur la lettre 'K' de la figure, donner la valeur 0.5 à *blacker* ne produit aucun résultat visible tandis que, même si la valeur 0.6 donne des résultats satisfaisants pour ce caractère précis, cette valeur, augmentant beaucoup la noirceur de toute la fonte, ne serait pas acceptable. Le résultat est souvent une grande distorsion du caractère et, pour des tailles plus petites, peut conduire à des taches sans forme. Pour un caractère de corps 5, même la valeur de 0.6 est inacceptable car elle noircit tout *sauf* la diagonale !

Le paramètre *fillin* semble plus utile puisqu'il s'applique seulement aux diagonales. *fillin* n'est pas un verbe mais désigne la quantité par laquelle une diagonale paraît plus noire que ne le garantissent ses dimensions, en grande partie à cause du fait qu'un segment linéaire va noircir partiellement plusieurs pixels sur une diagonale alors qu'un seul pixel serait noirci sur une horizontale ou une verticale¹³. Ça donne en général des effets non désirés qu'il faut contrecarrer en amincissant la diagonale avant de la tramer (*rasterize*). La valeur de *fillin* peut également être négative ; elle indique alors en réalité à METAFONT qu'il faut des diagonales plus noires. *fillin* est aussi utilisé dans le cas d'angles aigus entre des pixels BPN adjacents, angles qui peuvent devenir des pièges à encre ou des endroits de perte de charge. Ceci survient aussi sur les bords horizontaux ou verticaux mais cet effet est beaucoup plus prononcé le long des diagonales où les pixels BPN empiètent d'au moins 25 % sur les zones blanches voisines¹⁴.



Le 'K' de *Computer Modern* s'avère être particulièrement sensible à toute variation de la valeur de *fillin*. En effet une valeur aussi petite que -0.05 suffit pour ajouter les pixels nécessaires à la diagonale supérieure pour tous les dessins du corps 5 au corps 17. Malheureusement, si on ne corrige que les diagonales, les empattements paraissent excessivement frêles et d'autres corrections sont alors nécessaires.

Il ne faudrait pas croire que ces problèmes révèlent un défaut spécifique à la méthode de tracé de fontes employée par METAFONT-to-GF. Ils sont en

¹³ On trouvera dans [Hobby90] de bons exemples.

¹⁴ Voir [Knuth86], pages 93–94, pour le calcul de *fillin*.

fait inhérents aux matrices à peine suffisantes pour une définition de 300 dpi et, notamment, à la technique particulière d'encreage NPB.

It must not be imagined that these problems reveal a defect unique in the METAFONT-to-GF method of font expression. Rather, they are inherent in the barely adequate matrix of 300 dots/inch and, more particularly in the peculiar characteristics of write-white inking technology. Consider the following two examples, both set in Adobe Times-Bold, one on a Laser-Writer II NTX (Canon write-black print-engine) and the other on a Texas Instruments Omni-Figure 6. Texte composé en TimesBold : à gauche avec une technologie BPN, à droite avec une NPB

Considérons les deux exemples de la figure 6. Tous deux ont été composés avec le Times-Bold d'Adobe, l'un sur une Laser-Writer II NTX (imprimante Canon de mode BPN) et l'autre sur une Texas Instruments OmniLaser 2115 (imprimante Ricoh de mode NPB). Celui qui a écrit le programme de tramage pour l'Omnilaser a résolu le problème de l'atténuation des lignes en NPB en choisissant de cracher du toner dessus. Les fontes grasses d'Omnilaser sont effectivement très grasses. On obtient le même résultat que si on avait mis, pour un caractère dessiné en METAFONT, la valeur *blacker* à au moins 2.

Quand on s'est rendu compte du problème posé par les matériels de type NPB, on a refusé, pour *Computer Modern*, la méthode brutale et simpliste consistant à épaissir tous les attributs et traits d'un nombre arbitraire de pixels, en partie à cause de l'effet désastreux produit sur les caractères en corps 8 ou plus petit. Actuellement, dans le cas de l'imprimante Ricoh, on donne la petite valeur 0.2 au paramètre *blacker* de METAFONT et on traite les lignes d'un seul pixel d'épaisseur en imposant, pendant le calcul, une contrainte spéciale qui les interdit. L'introduction de la contrainte NPB et l'application de `blacker:=0.2;` et `fillin:=-0.2;` produit la grille de la figure 5 en bas à droite. Ces valeurs semblent être un bon compromis.

Mais une étude plus approfondie de cet ensemble de corrections NPB fait apparaître quelques problèmes. La figure 7 donne les résultats des mêmes corrections pour les corps 9 et 10. Les carrés noirs indiquent les caractères tramés pour un appareil BPN et les carrés blancs les pixels ajoutés par un jeu classique de corrections pour un appareil NPB. Les barres verticales indiquent les approches gauche et droite.

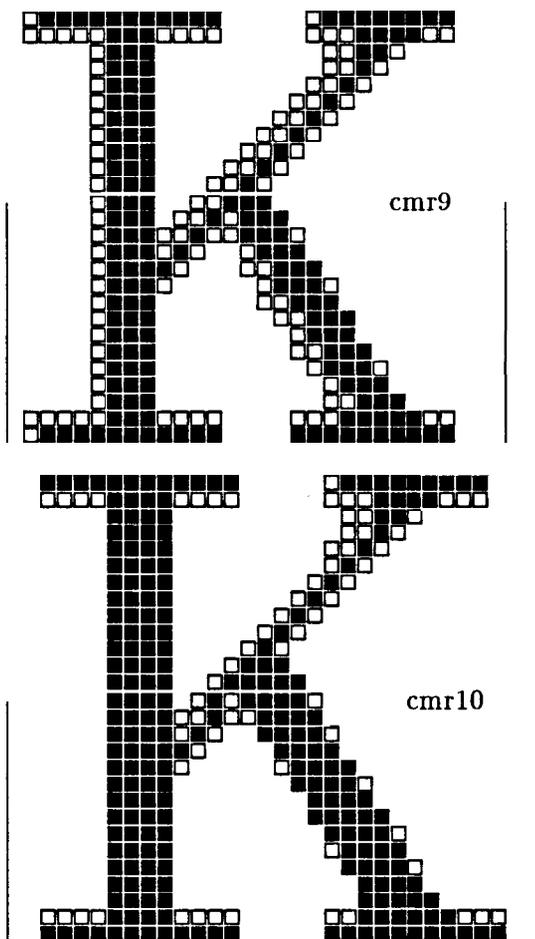


Figure 7. Write-Black on Write-White.

Cet ensemble de corrections pour les imprimantes Ricoh est bien accepté depuis des années, mais peut-être est-il temps de le revoir. En faisant divers essais, je me suis rendu compte récemment que, en pratique, la contrainte NPB convient et qu'il suffit de donner une petite valeur à *blacker* pour traiter les minceurs restant dans les caractères NPB. Par contraste, une valeur négative pour *fillin* semble ou bien n'avoir presque aucun effet ou, au contraire, en faire trop.

Un article récemment paru dans le journal du *TUG* (groupe des utilisateurs de $\text{T}_{\text{E}}\text{X}$), [Yanai et al.90], permet de faire d'autres observations sur les difficultés rencontrées, dans cette définition de 300 dpi, avec les mêmes fontes *Computer Modern* que celles dont nous parlons ici. Il faut reconnaître

Le fait de n'utiliser que la contrainte d'élimination des traits à un seul pixel double l'épaisseur de la diagonale supérieure de cette lettre en corps 9 (et la triple en corps 7). Augmenter la valeur *blacker* noircit le fût et la diagonale inférieure en corps 9 (mais n'a pratiquement aucun effet en corps 7). Par contre, en donnant une valeur négative à *fillin*, on quadruple la diagonale supérieure en corps 9. Une couche inutile de pixels ayant été ajoutée au moins, le caractère est sensiblement plus épais en corps 9 qu'en 10 (en corps 7, l'utilisation de ce paramètre ne produit qu'un léger noircissement de la diagonale inférieure).

que ces fontes sont encore à l'état expérimental, mais elles semblent bien confirmer ce que nous soupçonnons, à savoir que les contours de PostScript ont tendance à produire des images tramées avec une épaisseur d'au moins 2 pixels pour les traits. Si on compare les déliés dans une matrice de points à 300 dpi, où l'on sait avoir des traits à un seul pixel, avec ce que donne PostScript à 300 dpi, il devient évident que ce dernier produit un épaissement. Dans tous ces exemples à 300 dpi (en excluant les matrices produites par METAFONT) il n'y a qu'un seul trait d'un pixel d'épaisseur et en fait c'est une erreur de rendu de PostScript pour la capitale 'R' en corps 12 dans l'émulation de *Computer Modern*. Berry et Yanai notent ([Yanai et al.90] page 527) que « les caractères imprimés depuis les fontes en matrices de points sont plus fins que depuis celles produites par contours »¹⁵. Un tel épaissement concerne d'avantage la famille de styles *Bodoni/Modern* que les autres, mais c'est une contrainte pour le développement de n'importe quelle police à 300 dpi.

3. L'effet de rétrécissement des fontes

Quel que soit le soin apporté au choix des valeurs utilisées pour corriger les différences entre ces techniques d'impression, dans tous les cas les résultats sont plus éloignés des proportions des épreuves de caractères fournies par METAFONT avec des matrices NPB qu'avec celles BPN. Il y a un autre problème qui semble insoluble : faire un dessin sans sacrifier l'indépendance vis-à-vis du matériel. C'est l'effet de rétrécissement (*shrinking effect*) qui se produit pour de nombreuses fontes sur des imprimantes NPB à moins qu'elles ne soient noircies énormément.

Les trois dessins de la figure 8 montrent que la minuscule (ou bas-de-casse) 'o' en BPN est une bonne approximation du caractère idéal (image en mode épreuve) et, qu'au premier abord, le caractère NPB semble acceptable. Du point de vue de la forme, il l'est effectivement ; mais ses dimensions sont sensiblement plus petites qu'en mode BPN. Chaque fois que l'on utilise les deux technologies en même temps, on est sûr que quelqu'un va se plaindre que les sorties produites par T_EX sur une imprimante NPB donnent l'impression d'avoir été imprimées plus noires mais avec un point de moins

¹⁵La différence que note [Yanai et al.90], page 535, suggère qu'il y a une façon plutôt bizarre de créer des traits à un seul pixel en PostScript, mais cette méthode résulte d'une tentative de suppression de chemins superflus et ne sera probablement pas une caractéristique permanente de PostScript.

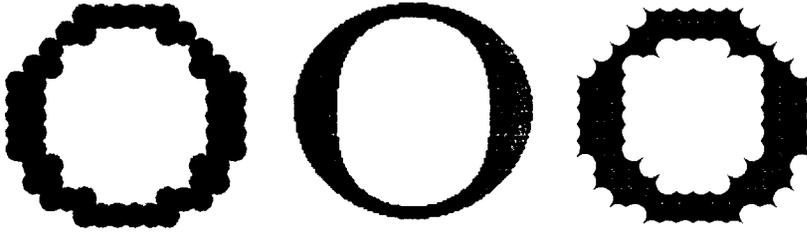


Figure 8. Minuscule 'o' en Cmr7. De gauche à droite : en BPN, idéal et en NPB

que celles sorties sur une imprimante BPN, et c'est vrai ! Dans la version NPB, le caractère a pratiquement perdu un demi-pixel dans ses dimensions extérieures. Cette perte n'est pas aussi insignifiante que ça, même si on la compare avec l'image à pixels carrés et elle devient tout à fait visible quand on la compare à l'image BPN car cette dernière est nécessairement grossie d'un demi-pixel, ce qui fait que la différence est finalement d'un pixel entier.

Avec des petites tailles, de l'ordre de 7 points, un seul pixel peut différencier deux forces de corps ne différant que d'un point. Mais puisqu'il est plutôt rare d'avoir du texte continu en corps 7, il vaut mieux regarder des forces de corps plus normales. Un 'o' en corps 9 en BPN fait 16 pixels de haut et il paraît avoir au moins 16.5 pixels (carrés). En corps 10, un 'o' fait 18 pixels de haut mais en BPN, il semble en avoir 18.5, tandis qu'un 'o' NPB ne semble en avoir que 17.5. Mettez deux pages produites par deux imprimantes, BPN et NPB, côte à côte, et vous vous apercevrez que le rétrécissement des caractères NPB est très visible.

4. Adaptation du dessin à la technologie d'impression

Quelques-unes des difficultés à trouver une solution générale compatible avec ces technologies différentes pourraient être évitées en introduisant des instructions conditionnelles dans les procédures de dessin de METAFONT. Sans préjuger si le jeu en vaut la chandelle, nous pouvons considérer un arrangement qui améliorerait le 'K' en corps 10. Knuth est très explicite quant à la nécessité d'être cohérent lors du positionnement dans la matrice de points des caractères symétriques, comme le 'o', dont les bords extérieurs des courbes gauche et droite sont arrondis à partir des approches gauche et droite, et ce avant tout autre calcul de positionnement. Ce calcul initial du blanc de chaque côté d'un caractère se fait aussi pour des caractères qui

ne montrent aucune symétrie, comme le 'K', ce qui, au moins en corps 10, produit une distorsion bizarre. Dans la figure 7, en bas, juste à droite du fût, la diagonale se décroche d'un pixel, ce qu'on pourrait corriger en étirant le caractère très légèrement. Si on fait une épreuve sur une feuille transparente et qu'on la place sur le caractère, il est évident qu'une version étirée représenterait la forme idéale du caractère. Comme l'indiquent les barres montrant les approches, il y a assez de place ; seule l'extrémité de l'empatement inférieur risquerait de trop avancer vers le caractère suivant. En faisant l'arrondi nécessaire à un endroit différent de la cuisine décrivant ce caractère, il est possible d'augmenter cet effet à basse définition sans détériorer l'apparence à haute définition.

J'essaye actuellement de définir une fonte basée sur des principes complètement différents : le point fixe pour l'arrondi est à peu près au milieu de la chasse et toutes les dimensions sont calculées à partir de là et non depuis les limites d'approche vers l'intérieur. Mais je n'ai pas encore réalisé assez de caractères pour dire si ça marche !

Références bibliographiques

- [Billawala87] Billawala (Neenie), Write-white printing engines and tuning fonts with METAFONT, *TUGboat*, vol. 8 (1), April 1987, pp. 29-32.
- [Hobby90] Hobby (John D.), *Generating Automatically Tuned Bitmaps from Outlines*, Technical Report 148, Murray Hill, New Jersey, AT&T Bell Laboratories, February 1990.
- [Knuth86] Knuth (Donald E.), *The METAFONTbook*, Reading, Mass., Addison Wesley, 1986.
- [Knuth90] Knuth (Donald E.), A note on digitized angles, *Electronic Publishing (EP-odd)*, vol. 3 (2), May 1990, pp. 99-104.
- [Rubinstein88] Rubinstein (Richard), *Digital Typography*, Reading, Mass., Addison Wesley, 1988.
- [Yanai et al.90] Yanai (Shimon) and Berry (Daniel M.), Environment for translating METAFONT to PostScript, *TUGboat*, vol. 11 (4), November 1990, pp. 525-41.