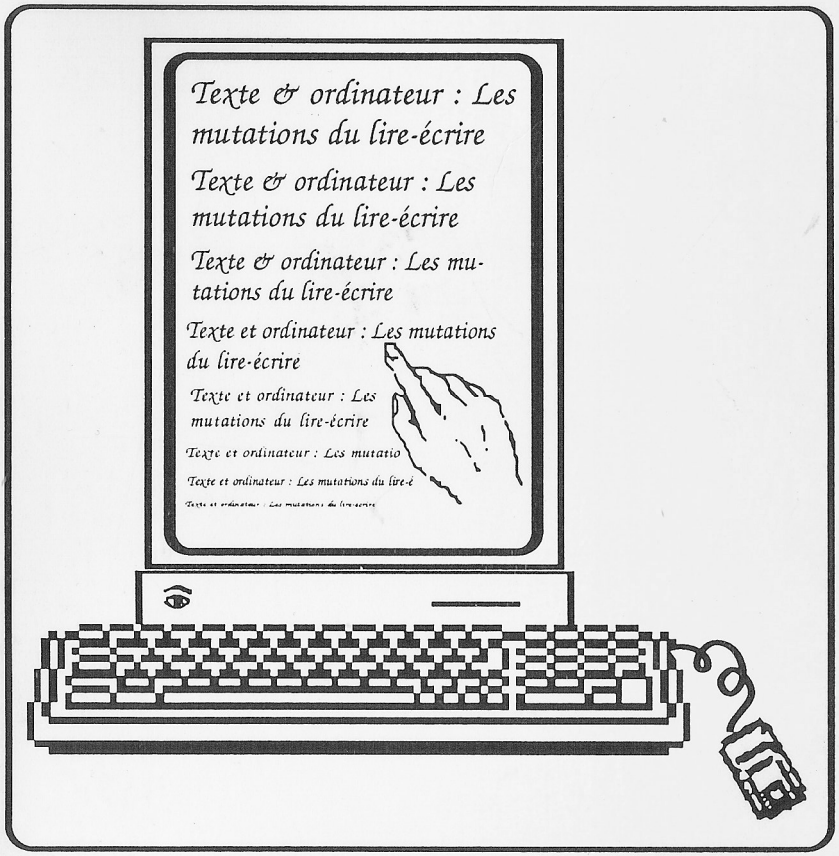
TEXTE ET ORDINATEUR

Les Mutations du Lire-Écrire

Sous la direction de
Jacques ANIS et Jean-Louis LEBRAVE



ÉDITIONS DE L'ESPACE EUROPÉEN

Texte et ordinateur : les mutations du lire-écrire

sous la direction de
Jacques ANIS et Jean-Louis LEBRAVE

Actes du colloque interdisciplinaire tenu à l'Université Paris X Nanterre, 6-7-8 juin 1990.

Volume hors-série de la revue LINX,
publication du Centre de Recherches
Linguistiques de Paris X Nanterre

Éditions de l'Espace Européen
89 rue Sartoris
92250 La Garenne-Colombes

Jacques André
IRISA/INRIA–Rennes

Nouveaux caractères pour une nouvelle typographie

IL Y A QUELQUES DIZAINES d'années, les enfants jouaient avec des soldats de plomb. Aujourd'hui, ils jouent encore à la guerre, mais sur micro-ordinateur.

Il y a quelques dizaines d'années, le plomb servait aussi à la fabrication des caractères qui, aujourd'hui, sont numérisés en ordinateur. Mais, tandis que les auteurs de jeux de stratégie, d'aventures ou autres jeux de rôles ont fait preuve d'inventions inouïes, il faut bien reconnaître que l'invention en typographie numérisée fait plutôt défaut (ce qui, soit dit en passant, n'est pas pour déplaire à une grande partie de la profession[1] !). Non pas qu'il n'y ait pas eu d'invention, technologique, pour mettre au point la production de fontes, leur numérisation et leur flashage sur imprimante ou photocomposeuse à laser. Mais cette technologie ne semble pas s'accompagner de recherches sur de nouvelles possibilités de création de caractères.

Pourtant, les techniques actuelles d'impression offrent de quoi produire une révolution dans la création, et l'emploi, de caractères. Ces quelques pages voudraient attirer l'attention des créateurs de caractères sur cet aspect, méconnu, de l'emploi de l'informatique en typographie.

1. Un processus de fabrication constant

La grande invention de Gutenberg c'est le processus de fabrication des caractères. De façon schématisée, un "graveur de poinçon" taille des caractères en acier, lettre par lettre, corps par corps. Ces poinçons servent alors à créer des moules où seront coulés les caractères utilisés par les imprimeurs. Le poinçon ne sert que rarement, lorsque les moules sont usés. C'est l'œuvre originale[2]. Tout le reste du processus n'est que copie. Copie

---

1 Voir à ce sujet Roger Laufer, « Calligraphie synthétique animée », *Culture technique*, n° 17, mars 1987, 273-275.

2 Ce sont les poinçons qui sont conservés à l'Imprimerie Nationale ou au Vatican. C'étaient les poinçons qui étaient protégés par copyright : voir à ce sujet Charles

pour obtenir un moule, copie pour obtenir un caractère (un multiple), copies enfin lors de l'impression. Copies dont la plus grande qualité est la fidélité par rapport à l'original. La liberté typographique n'existe pas (sauf en calligraphie lorsque chaque lettre est tracée à la main selon l'inspiration du moment). Seule compte la rigueur de la copie.

L'utilisation des pantographes pour tailler mécaniquement les poinçons a aggravé ce processus de copie : même les poinçons sont copiés entre eux, du corps 8 au corps 16 par exemple. La photocomposition n'a fait que renforcer ce processus de copie. Les fondeurs sont devenus photographes, copient les poinçons originaux et vendent des disques ou tambours d'où sont copiés les caractères sur les films. Et l'impression au laser entérine le processus : par copie on obtient l'image numérisée d'un caractère qui est utilisée à chaque occurrence de ce caractère dans une page ou une laize.

2. La machinerie des fontes informatiques

Soyons un peu technique et voyons comment un caractère est effectivement tracé sur une feuille d'imprimante. Utilisons pour cela la *font machinery* de PostScript[1]. Tout d'abord par un processus de saisie (utilisation d'un logiciel spécialisé pour le tracé de caractères, par exemple Typo ou Fontographer, ou d'un logiciel permettant de "scanner" un caractère existant, Ikarus par exemple) les caractères sont mis en mémoire (par exemple sur une disquette). Contrairement à ce que l'on croit souvent, ces caractères sont en fait définis par des courbes mathématiques décrivant leurs contours.

Lorsqu'un logiciel veut imprimer un ensemble de caractères, par exemple « été » en Times romain, corps 12, il doit d'abord spécifier la police[2] voulue. En PostScript on écrira par exemple :

```
/Times-Roman findfont 12 scalefont/
```

Bigelow, « Du piratage des fontes », *Technique et Science Informatiques*, Dunod éd., vol. 6(3), 1987, 251-260.

1 PostScript est un langage de description de page permettant d'utiliser indifféremment un grand nombre d'imprimantes à laser ou photocomposeuses de quatrième génération. On consultera sur ce langage les ouvrages d'Adobe chez Addison Wesley : *PostScript Language*... L'un d'eux a été traduit en français (*PostScript par l'exemple*, chez InterÉditions, Paris, 1988).

2 On utilise ici le mot *fonte* pour désigner l'ensemble des caractères dans tous les corps, dans toutes les orientations, pour une famille donnée, une graisse donnée, etc., par exemple le *Times gras italique* constitue une fonte, le *Times romain maigre* une autre. On utilise *police* pour désigner un sous-ensemble d'une fonte pour un corps donné, dans une orientation donnée. Par exemple le *Times gras italique corps 12 orienté à 45° avec l'horizontale* constitue une police. La notion de fonte correspond à ce qui se vend, celle de police à ce qu'on utilise (c'était le tiroir de la casse, puis un disque de photocomposeuse avec une certaine valeur du zoom, etc.). Voir à ce sujet, Jacques André, « Métrique des fontes en typographie traditionnelle », *Cahiers Gutenberg* n°4, 1989, 9-22.

Pour imprimer le premier « é » désiré, on appelle alors l'instruction *(é) show* qui va lancer toute la "machinerie" suivante :

1 on appelle la définition mathématique des contours du caractère « é ».

2 Un programme, dit de *scan conversion*, traduit alors cette spécification en un ensemble de points noirs ou blancs dépendant du caractère considéré, de sa taille (donc de son corps) et de la définition de l'imprimante utilisée. L'avantage de cette méthode est qu'elle prend moins de place que si on gardait toutes les matrices de points possibles. Elle a surtout l'avantage de pouvoir adapter le tracé d'un caractère en fonction de la grille de l'imprimante et de faire un certain nombre de corrections donnant en général une très bonne approximation du caractère désiré[1].

3 La matrice de points est alors envoyée sur la page à imprimer ;

4 elle est aussi cachée en mémoire.

Avec l'instruction *(t) show*, le même processus se produit, cette fois avec les courbes du « t ».

Le nouvel appel de l'instruction *(é) show* va cette fois se dérouler différemment : puisque la matrice de points correspondant au « é » a été cachée en mémoire, on ne va pas la recalculer par un nouvel appel à l'algorithme de *scan conversion* (qui coûte très cher en temps). On l'utilise directement.

Si maintenant on voulait réimprimer ce même mot « été » en corps 14, on referait tous les calculs pour obtenir des matrices de points plus grandes.

3. Premières conséquences

Si l'on regarde bien le mécanisme que nous venons de décrire, on peut remarquer deux choses :

3.1. Adaptation des tracés au corps

Le corps d'un caractère est connu avant que l'algorithme de conversion ne soit appelé. On peut donc (on devrait même) adapter le tracé d'un caractère au corps, non pas en ayant autant d'algorithmes de tracés que de corps possibles, mais en modélisant ces tracés. Prenons un exemple simple.

Le Didot est un caractère dont les empattements sont réduits à des filets. Pour que ceux-ci soient visibles quel que soit le corps ils doivent avoir une taille inversement proportionnelle au corps (plus le caractère est grand, plus le filet est, proportionnellement, petit). Ainsi,la figure 1 (p. 50) a-t-elle été construite en prenant des filets horizontaux dont l'épaisseur a pour valeur corps/250. Bien sûr, cet exemple est particulièrement simple. Mais on pourrait paramétrer de la même façon les coordonnées de la courbe décrivant par exemple le contre-poinçon d'un « e » Garamond. Et si

---

1 On trouvera dans Jacques André et Roger Hersch (eds.), *Raster Imaging and Digital Typography*, Cambridge University Press, 1989, divers articles sur ces techniques de numérisation de caractères.

on ne sait pas bien le faire aujourd'hui, ceci ne fait que prouver l'intérêt des études en cours sur la modélisation des caractères[1].

3.2. Un début de dynamisme

Puisque les caractères sont définis par des courbes mathématiques, on peut donner à certains points de coordonnées des valeurs non constantes, par exemple le résultat de tirage de nombres au hasard. C'est ce qu'a fait Donald Knuth dans la police Punk[2] : Les coordonnées des points sont calculées par une fonction de génération de nombres aléatoires ce qui donne une illusion d'écriture manuscrite. Mais une fois une lettre "calculée", elle garde les mêmes dimensions tant que l'on ne change pas de police. Ainsi, voir figure 2 (p. 50), tous les « A » sont identiques, tous les « E » sont identiques.

4. Fontes dynamiques

La *font machinery* décrite précédemment ne fonctionne en fait pas obligatoirement toujours comme cela. On peut en effet demander de ne pas cacher la matrice de points en mémoire. Ce qui oblige à recalculer chaque caractère à chaque fois que l'on l'imprime. Ceci dégrade les temps d'impression mais donne des possibilités incroyables de "dynamisme". Voici d'abord quelques exemples[3] avant de décrire une vraie police construite sur ce principe.

4.1 Tracés aléatoires

Avec le même principe de tirage de nombres aléatoires, on peut[4] reprendre la police Punk de Knuth et tracer des « A », par exemple, de façon qu'ils soient tous différents dans un même texte.

Scrabble est une police de caractères qui imite les pièces du jeu de même nom en entourant chaque lettre d'un carré à bords arrondis[5]. Si on

---

1 On aura une idée des études en cours, basées en général sur la théorie du signal, en consultant Robert Morris, « Image processing aspects of type », in *Document Manipulation and Typography* (JC van VLiet ed.), Cambridge Unviersity Press, 1988, 139-155.

2 Donald Knuth, « A Punk Metafont », *TUGboat*, vol. 9(2), 1988, 152–168. En fait, cette police a été dessinée en utilisant son logiciel METAFONT. Voir à ce sujet: Donald Knuth, « Le concept de méta-fonte », *Communication et langage*, 1985, et Douglas Hofstadter, « Metafont, métamathématiques et métaphysique », in *Ma thémagie*, InterEditions, 1988, 249-288.

3 On trouvera quelques précisions techniques dans Jacques André et Bruno Borghi, «Dynamic fonts» in *Raster Imaging and Digital Typography* (André & Hersch eds.), Cambridge University Press, 1989, 198-204 et *idem, PostScript Language Journal*, vol. 2(3), 1989, 6-8.

4 et on l'a effectivement fait : Jacques André et Victor Ostromoukhov, « Punk : de Metafont à PostScript », *Cahiers Gutenberg*, n°4, décembre 1989, 23–28.

5 Cette police fait un peu plus : la chasse est soit horizontale (de gauche à droite), soit verticale (du haut vers le bas) pour simuler une partie.

affecte à ces pièces un angle par rapport à l'horizontale et si on tire au hasard la valeur de cet angle chaque fois que l'on imprime une pièce, on peut obtenir une présentation comme celle de la figure 3 (p. 50). A noter que si l'on repasse ce même programme, on aura une nouvelle disposition de pièces.

4.2 Paramétrisation de la forme de caractères

Le principe de tracer un caractère en fonction de son corps peut être étendu à d'autres facteurs. En calculant un caractère lors de son utilisation, et non une fois pour toutes, on a accès à de nombreux paramètres, par exemple la position du caractère par rapport aux marges de la justification, par rapport à son voisin, etc. Toutes choses qui peuvent être mises à profit pour une typographie fine.

Un exemple typique d'emploi de cette possibilité est lié aux langues sémitiques. En hébreu ou en arabe traditionnels, la justification se faisait dans les noirs, au contraire des langues européennes où elle se fait dans les blancs : pour justifier un texte français, par exemple, on aggrandit la taille des espaces de façon à pousser à l'extrême droite de la marge les derniers caractères de la ligne (figure 4, p. 51, ligne du haut). Dans les langues sémitiques au contraire, on allonge certaines lettres. Dans la ligne du bas de la figure 4 (avec des caractères européens), les arches du « m » sont allongées (ce n'est pas le caractère en entier qui est allongé : la taille des futs ne change pas). Cet allongement a été calculé en fonction de la place restante à cet instant précis.

La figure 5 (p. 51) est inspirée de la calligraphie arabe. Pourtant, elle utilise une fonte PostScript unique : tous les « i » et « j » sont du même corps, l'ordre d'impression étant *(iiijii) show*. Chaque définition de caractère dépend du rectangle noir (extrèmités gauche et droite des caractères) et du caractère précédent (trait plus ou moins bas). Ceci n'est possible que parce que ces dépendances sont évaluées à chaque occurrence de caractère.

5. Une fonte complète et dynamique

Les exemples précédents ont montré la faisabilité de caractères dynamiques. Il était important de tester en vraie grandeur ces principes sur une vraie police. C'est ce qu'a permis le Delorme. Il s'agit d'un caractère modulaire conçu par Christian Delorme et Roger Laufer[1].

---

1 « Les premières ébauches ont été faites avant nos rencontres de Lure 1986. Mais la toute première impulsion m'a été donnée lors des rencontres de l'année précédente, lorsque des propos ont été tenus par Roger Laufer sur les éléments variables des caractères : à partir de certains éléments de base, modulables, on pourrait modifier des signes et des lettres. Les processus informatiques invitent en effet à une certaine combinatoire. C'était l'un de mes objectifs pour les lettres faites pour les ordinateurs... » . Extrait de *Carrefours télématiques* n° 9/10, 1986 (entretien de Christian Delorme avec Françoise Holz-Bonneau et Jean-Paul Martin).

Le Delorme est composé de modules simples et en nombre très limité :

- des segments de droite horizontale,
- des segments de droite verticale,
- des segments de droite oblique, à ±30d° avec l'horizontale,
- des points[1].

Tracer un caractère revient alors à tracer ces segments en suivant l'ordre du ductus. Parmi d'autres principes de ce caractère, citons :

- la suppression de la notion de capitale ou bas de casse : les uns et les autres peuvent être intimement mélangés;
- il n'y a pas d'arrondis, mais une illusion donnée par des raccords étudiés ;
- il n'y a ni plein ni délié (il s'agit donc d'une linéale dans la classification Vox);
- on peut avoir, ou pas, des empattements ;
- la proportion des noirs et des blancs, des vides et de pleins élimine tout risque de bouchage quel que soit le corps.

La figure 6 (p. 51) montre le carton du caractère « a » (ou du « e » en le retournant à la « newman ») et la figure 7 (p. 52) quelques caractères. La figure 8 (p. 52) est beaucoup plus importante : elle montre des propriétés dynamiques de ce caractère. On peut y noter, par exemple, que

- Il y a plusieurs orientations du « O » : ceux de « SATOR » et de « AREPO » sont tronqués en haut à gauche tandis que celui de « OPERA » l'est en bas à droite et enfin celui de « ROTAS » en haut à droite. Pourquoi ? pour pouvoir répartir les blancs dans le mot, c'est-à-dire remplacer le crénage traditionnel.
- Le « A » de « SATOR » est plus court que celui au début de « AREPO ». Ceci permet de répartir les noirs, de donner plus de mouvement dans la lecture, etc.

En fait ceci a été fait avec une fonte normale, utlisant donc le procédé de garder en mémoire les matrices de points des caractères. Il a fallu par contre définir un grand nombre de caractères et laisser à l'utilisateur le soin d'utiliser telle ou telle variante. Mais ceci aurait aussi pu être fait complètement automatiquement par une police dynamique. Une première version a d'ailleurs fonctionné en 1989 à l'EPFL à Lausanne sur ce principe. Une nouvelle version est en cours de programmation et va fonctionner comme le prototype de 1989 (seuls les ductus ont changé) : Lorsqu'un caractère est appelé, on recalcule complètement ses contours. Mais on le fait en tenant compte du contexte : largeur de la justification, caractère avant, caractère après, etc. Ceci permet beaucoup de libertés :

---

1 En fait, les segments étant arrondis à leurs extrémités, on peut considérer le point comme un segment de longeur nulle.

– choix de caractères en fonction des caractères voisins (comme les orientations des « O » de l'exemple précédent, mais aussi choix de variantes pour certaines lettres),

– calcul de l'approche d'un caractère en fonction du précédent, comme pour le crénage dans les fontes normales : ceci utilise les propriétés topologiques de chaque caractère (un « M » est formé à droite d'une barre verticale, un « i » est aussi formé d'une barre verticale, l'espace entre un « M » et un « i » sera donc plus large qu'entre un « C » (creux à droite) et un « A » (pointu à gauche),

– déformation de certains signes, par exemple allongement dynamique de la descendante du « A » en début de mot ou en fin de mot (après retournement du « A »), ou de la barre du « T » à droite ou à gauche selon les voisins,

– ligatures automatiques : un seul signe remplaçant les trois de « ffl », mais aussi ligatures horizontales pour assurer la continuité du tracé. La figure 9 (p. 52) utilise les caractères de 1989, mais montre les ligatures horizontales, faites complètement automatiquement, entre, par exemple, les signes « hristi »,

– calcul du gris : la simplicité des modules (rectangles arrondis) permet de calculer simplement la surface de noir d'une ligne. Il est alors possible de la diminuer ou de l'augmenter (en faisant varier dynamiquement l'œil, ou au contraire la graisse, de certains signes) de façon à obtenir une valeur constante du gris,

– etc.

Cette police Delorme est encore en cours de définition, néanmoins elle montre bien les possibilités de dynamisme.Tout ceci n'est possible que parce que l'on utilise une technologie moderne permettant de calculer à chaque instant la forme d'un caractère, parce que l'on dispose de modèles (même aussi simplistes que les nôtres; on attend beaucoup des modèles basés sur la théorie du signal!) et peut-être surtout parce qu'il y a derrière cela la volonté de sortir du carcan rectangulaire du caractère en plomb, carcan qui n'a plus aujourd'hui aucune raison d'être.

Figures

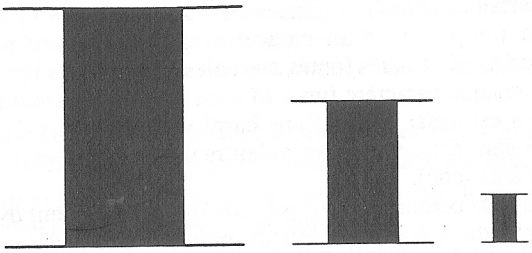


Figure 1 : Puisqu'il connaît le corps d'un caractère au moment de son tracé, un programme de tracé de caractères peut donner aux filets une épaisseur en fonction de ce corps. Ici ces trois « I » à la Didot ont des filets horizontaux d'épaisseur pratiquement constante quel que soit le corps.

Figure 2 : Exemple de caractères aléatoires : Punk de Knuth

Figure 3 : Une utilisation de la police Scrabble

the room
the room

Figure 4 : Justification européenne (on agrandit les blancs) et sémitique (on allonge dynamiquement certaines lettres, ici le m).

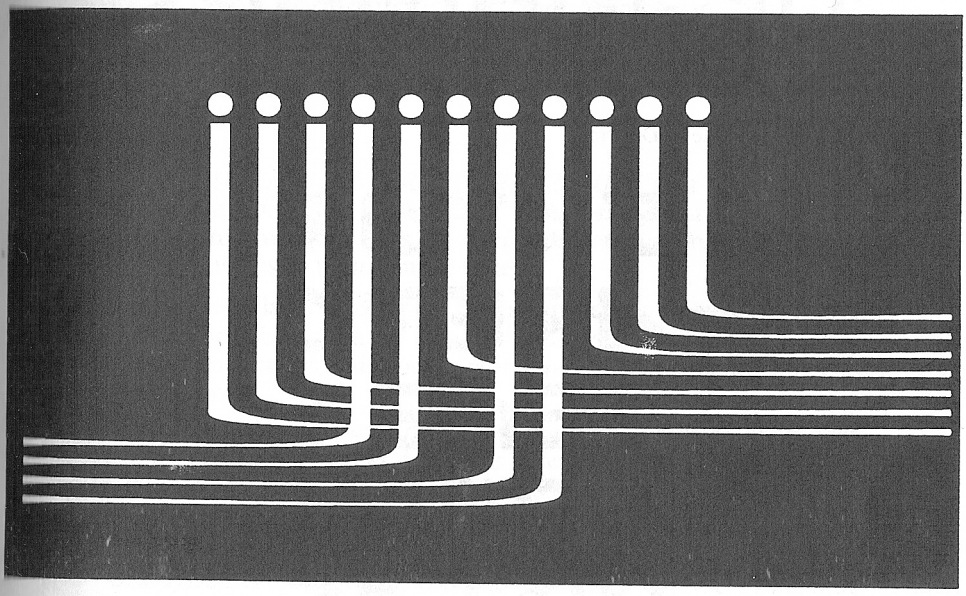


Figure 5 : Calligraphie par ordinateur : pour le programme PostScript, tous ces « i » et « j » sont du même corps. Mais la position relative des traits horizontaux et leur longueur sont calculées en fonction du précédent et de la largeur de la boîte noire.

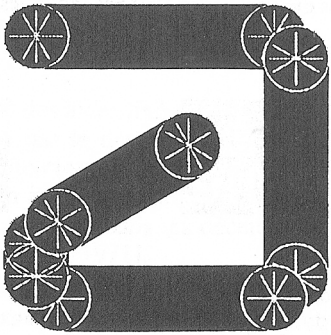


Figure 6 : Carton du « a » Delorme.

Figure 7 : Un texte en Delorme.

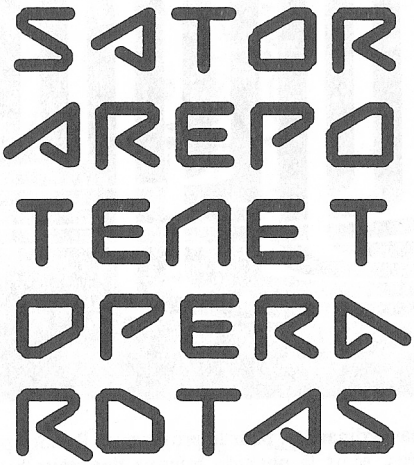


Figure 8 : Variantes dynamiques en Delorme (notez les diverses orientations des *O*).

Figure 9 : Deux modes pour le Delorme (version 1989) : à gauche le mode classique, à droite le mode dynamique (les ligatures, entre « hristi » ont été calculées automatiquement à l'impression).