

TIPE : Quantité d'encre utilisée par une fonte d'écriture

Rapport final

1 Préambule

Ce TIPE vise à étudier la consommation d'encre des différentes fontes couramment utilisées, pour déterminer dans quelle mesure il peut être avantageux d'en utiliser une en particulier.

Sur un ordinateur, les glyphes de ces fontes ont une représentation mathématique sous forme d'équations paramétrées, que l'on exploite pour calculer leurs aires.

On fournit une interprétation graphique des programmes développés *via* l'interface graphique de Caml.

2 Introduction

Le travail réalisé a tout d'abord consisté à convertir et traiter les fichiers de fontes pour les manipuler avec Caml Light.

Une étude mathématique des cubiques de Bézier décrivant les glyphes a permis de mettre en place une méthode de calcul efficace des aires des glyphes, en accord avec le principe de l'algorithme récursif de De Casteljau.

Il a fallu normaliser et pondérer les aires calculées, pour obtenir des valeurs concernant des situations précises.

On a enfin pu donner une représentation graphique des caractéristiques des glyphes que l'on a identifiées.

3 Corps principal

3.1 Modalités d'action

Obtention de données exploitables

On a comparé les fontes suivantes :

- Arial
- Comic Sans MS
- Courier New
- DejaVu Sans
- Garamond
- Times

Les fichiers de ces fontes sont au format TrueType (TTF) ou OpenType (OTF). Le logiciel *fontforge* permet de les convertir au format PostScript Type 1 (PFB).

L'utilitaire *t1disasm* convertit ensuite le PFB en ASM, lisible par Caml Light.

On a sélectionné les caractères suivants :

- Les lettres minuscules, non accentuées (de *a* à *z*)
- Les lettres capitales, non accentuées (de *A* à *Z*)
- Les chiffres (de *0* à *9*)
- Certains signes de ponctuation : *. , ! ? ; ;* ainsi que *' ()*

La description de ces caractères a été isolée manuellement dans les fichiers ASM produits.

Le reste de la programmation a été réalisé avec le langage Caml.

Traitement des données

La description d'un glyphe au format ASM se présente sous la forme :

```
/i {  
  24 245 hsbw  
  42 565 hmoveto  
  13 -6 vlineto  
  ...  
  0 45 5 49 4 41 rrcurveto  
  8 callsubr  
  closepath  
  endchar  
} ND
```

Extrait de la description du glyphe de *i* en Garamond

La description d'un glyphe est donnée dans le langage PostScript : la syntaxe est postfixée, les arguments sont relatifs (au point courant).

- ▷ **hsbw** est ici ignoré.
- ▷ **hmoveto** est une application partielle de **rmoveto**, qui déplace le point courant – horizontalement ici.
- ▷ **vlinto** est une application partielle de **rlinto**, qui déplace le point courant en traçant une ligne droite – verticalement ici.
- ▷ **rrcurveto** est au cœur de ce TIPE car c'est l'instruction de tracé d'une cubique de Bézier.
Ces dernières sont des courbes paramétrées, caractérisées par quatre points de contrôle. Deux de ces points (P_{000} , P_{111}) indiquent les deux extrémités de la courbe, les deux autres (P_{001} , P_{011}) orientent les demi-tangentes aux extrémités.

Si on note les points de contrôle $P_{000} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ $P_{001} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ $P_{011} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ $P_{111} = \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$,
la courbe est paramétrée par :

$$\forall t \in [0, 1] \quad \begin{cases} x(t) = x_0(1-t)^3 + 3x_1(1-t)^2t + 3x_2(1-t)t^2 + x_3t^3 \\ y(t) = y_0(1-t)^3 + 3y_1(1-t)^2t + 3y_2(1-t)t^2 + y_3t^3 \end{cases}$$

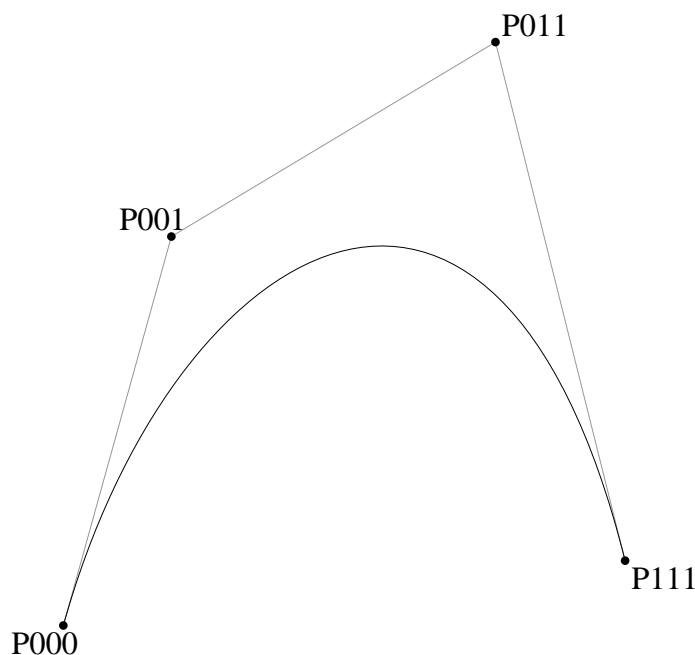


FIGURE 1 – Exemple d'une courbe de Bézier (en noir)

- ▷ **callsubr** appelle une séquence préprogrammée – ici le point du *i*, avec les instructions précédentes. Elle est destinée à être utilisée dans la description de plusieurs glyphes, et est décrite par :

```

dup 8 {
  -28 22 -22 28 vhcureveto
  29 22 22 28 hvcureveto
  29 -22 22 -29 vhcureveto
  -28 -22 -22 -29 hvcureveto
  closepath
  return
} NP

```

Définition de la sous-routine n° 8 en Garamond

- ▷ **closepath** et **endchar** permettent de fermer le glyphe avec des lignes droites.

Un fichier ASM est lu puis enregistré en Caml Light comme liste de chaînes de caractères. Un premier parcours est réalisé pour formater les instructions et leurs arguments, et remplacer les sous-routines par leur description. On peut ensuite effectuer des calculs sur la liste traitée.

Méthode de calcul

Comme on l'a vu, un glyphe est composé de segments et de courbes de Bézier.

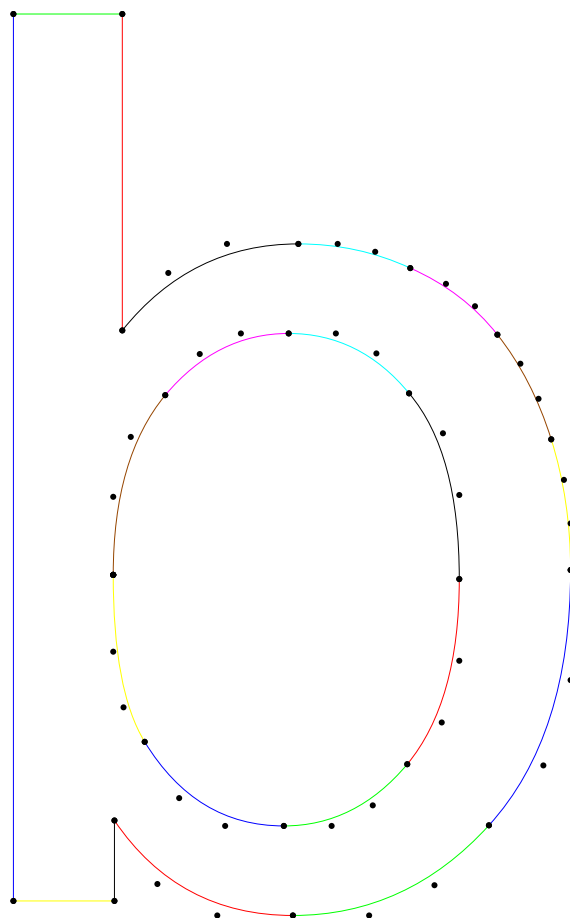


FIGURE 2 – Décomposition du glyphe de *b* en Arial
En noir, les points de contrôle

La formule de Green-Riemann donne l'aire \mathcal{A} d'une portion de plan enclose par un contour paramétré fermé et orienté \mathcal{C} :

$$\mathcal{A} = \frac{1}{2} \int_{\mathcal{C}} (x dy - y dx)$$

Un glyphe est alors parcouru récursivement, les aires de ses sections étant sommées pour obtenir l'aire totale.

Pour pouvoir comparer les fontes, les aires sont calculées par rapport à l'aire d'un carré de côté la hauteur du glyphe de x – c'est une convention. La méthode pour la calculer et de calculer la « bounding box » du glyphe (rectangle contenant la lettre, voir FIGURE 3). On a mis en place cette démarche pour tout les glyphes.

L'exemple du *b* en Arial est parlant. La FIGURE 3 illustre comment chaque section contribue à agrandir la « bounding box » (de la même couleur). La première « bounding box » calculée est la rouge en bas à gauche.

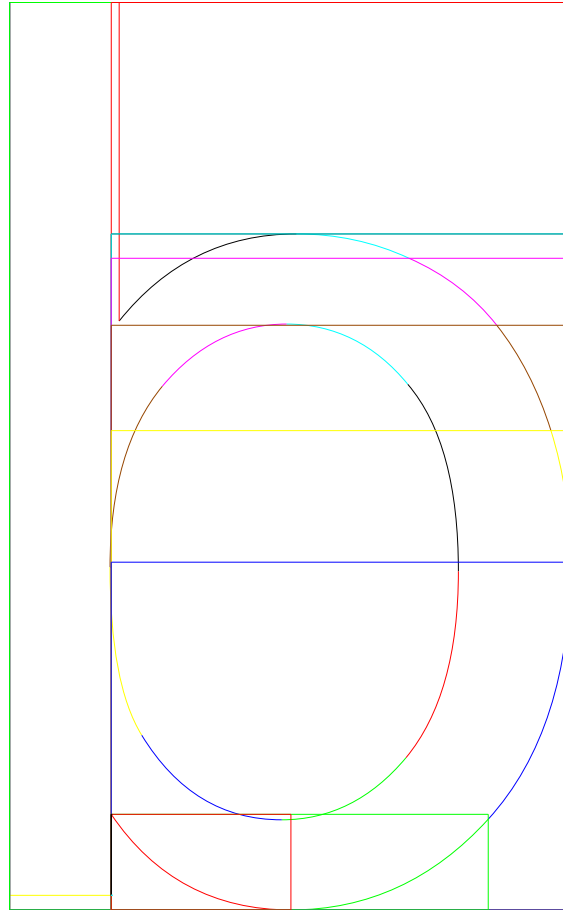


FIGURE 3 – Construction pas à pas d'une « bounding box »

Enfin, on a calculé, pour différents documents, la fréquence d'apparition des caractères étudiés, pour pondérer les aires.

Aspects graphiques

N.B. : On a préféré produire les figures de ce document avec Ocaml plutôt que Caml Light pour pouvoir les exporter en PDF. Cela a demandé la réécriture de certaines fonctions et des conversions manuelles.

Une cubique de Bézier est construite efficacement par l'algorithme de De Casteljau, ce qui en justifie l'utilisation. Ci-dessous, on construit une suite de barycentres, et le barycentre P_{ttt} est sur la courbe.

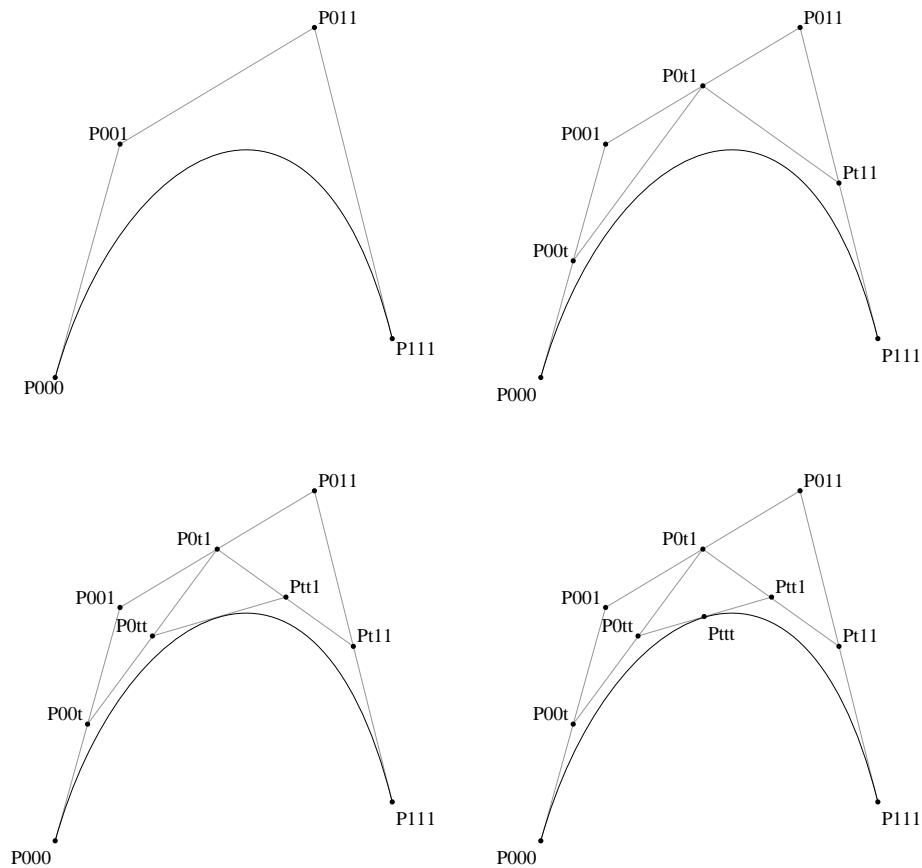


FIGURE 4 – Une étape de l'algorithme de De Casteljau

Les propriétés des cubiques de Bézier permettent d'écrire en oblique, et d'effectuer des rotations.

Times New Roman

FIGURE 5 – Texte en Times New Roman, penché de 0.1 radians, obtenu avec Caml Light

3.2 Restitution des résultats

On donne les fréquences d'apparition calculées pour deux documents.

Lettre	Document	La Bête Humaine ¹	Thèse L. de Broglie ²
e		15.9 %	16.2 %
a		8.68 %	6.26 %
i		7.08 %	6.37 %
t		7.04 %	6.93 %
s		6.98 %	7.32 %
r		6.23 %	5.92 %
o		4.55 %	6.03 %

FIGURE 6 – Fréquences d'apparition

¹ La Bête Humaine, Émile Zola, 1890

² Recherche sur les quanta, Louis de Broglie, 1924

La hauteur du x permet de calculer les aires pondérées relatives suivantes.

Hauteur du x	Document	La Bête Humaine	Thèse L. de Broglie
	Fonte		
1062	Arial	0.377	0.399
1120	DejaVu Sans	0.364	0.385
916	Times	0.363	0.386
409	Garamond	0.356	0.370
1149	Comic Sans MS	0.351	0.374
866	Courier New	0.320	0.330

FIGURE 7 – Aires pondérées relatives

3.3 Analyse – Exploitation – Discussion

La FIGURE 6 met en évidence les importantes fluctuations d'un document à un autre. C'est pourquoi on ne peut pas facilement généraliser les résultats de la FIGURE 7.

D'autre part, aucune fonte ne semble se démarquer des autres, si ce n'est *Courier New*, qui présente cependant l'inconvénient de consommer plus de papier avec sa chasse fixe, et est peu lisible.

4 Conclusion générale

Les glyphes des fontes PostScript Type 1 sont efficacement décrites à l'aide des cubiques de Bézier.

L'étude de ces glyphes, par la manipulation de ces objets mathématiques, permet de conclure que le choix d'une fonte d'écriture n'est pas un paramètre d'influence remarquable sur les coûts d'impression.

Il vaut ainsi mieux se préoccuper de la lisibilité et de l'esthétique, ainsi que du coût d'entretien d'une imprimante, plutôt que du choix de fonte à adopter.

5 Références bibliographiques additionnelles

Aucune référence supplémentaire à mentionner.