Cahiers GUTenberg

UN PILOTE GRAPHIQUE ENTRE LE LOGICIEL STATISTIQUE S ET PICTEX

Olivier NICOLE

Cahiers GUTenberg, nº 3 (1989), p. 21-31.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1989___3_21_0>

Un pilote graphique entre le logiciel statistique S et PiCTeX

Olivier NICOLE

INRA-CRJ, Laboratoire de Biométrie, 78 JOUY en JOSAS

En raison de la disparition prochaine du système d'exploitation Multics, l'INRA a entamé il y a deux ans une migration vers le système UNIX. C'est grâce à la grande portabilité des logiciels et surtout au fait qu'ils sont du domaine public, donc implantables sur toutes les machines, que TeX a été adopté, d'abord par le laboratoire de Jouy, puis peu à peu par l'ensemble des laboratoires du département de Biométrie.

L'arrivée d'un nouveau système d'exploitation a entraîné le choix de nouveaux logiciels. Le département de Biométrie a participé durant l'été 1987 à l'évaluation du « système d'accueil » S. Finalement S a été retenu, il sera implanté sur certaines des machines équipant l'INRA.

Nous avions donc à notre disposition un système d'édition de textes mathématiques très performant, une base logicielle statistique répondant à nos besoins, mais il manquait le lien entre les deux, un moyen simple d'exporter un graphique réalisé sous S pour l'inclure dans un texte TeX.

1. TeX en Biométrie

TeX est apparu à l'INRA au début de 1986. C'est François CHAHUNEAU, alors chercheur au laboratoire, qui a rapporté TeX d'un voyage aux États-Unis et au Canada.

Une première version de PCTeX a été implantée sur un micro-ordinateur compatible IBM-AT. Rapidement, TeX, et surtout LaTeX se sont révélé être les outils d'édition de textes offrant les possibilités de composition mathématique dont nous avions toujours eu besoin.

D'autres versions de PCTeX ont alors été acquises et installées sur notre réseau de micro-ordinateurs. Le secrétariat n'a plus utilisé alors que LaTeX pour l'ensemble de la frappe des textes, mathématiques ou non. Un micro-ordinateur était également disponible en libre-service pour l'ensemble des membres du laboratoire, afin de « faire du TeX ».

Quand la première machine UNIX est apparue au laboratoire, une carte équipée d'un micro-processeur 32032 de chez NSC et de 4 méga-octets de mémoire, fonctionnant dans un PC, c'est tout naturellement que la version C de TeX y a été implantée. Par la suite, TeX a été mis à disposition sur toutes les nouvelles machines UNIX installées au laboratoire.

L'expérience ayant été probante, les autres laboratoires du département se sont mis à LaTeX. Actuellement, la quasi-totalité des laboratoires de Biométrie utilisent TeX, au moins pour les textes mathématiques.

L'expérience de TeX à l'INRA est pour le moment limitée presque exclusivement au département de Biométrie, seules quelques actions individuelles ont réussi à faire pénétrer TeX dans d'autres laboratoires.

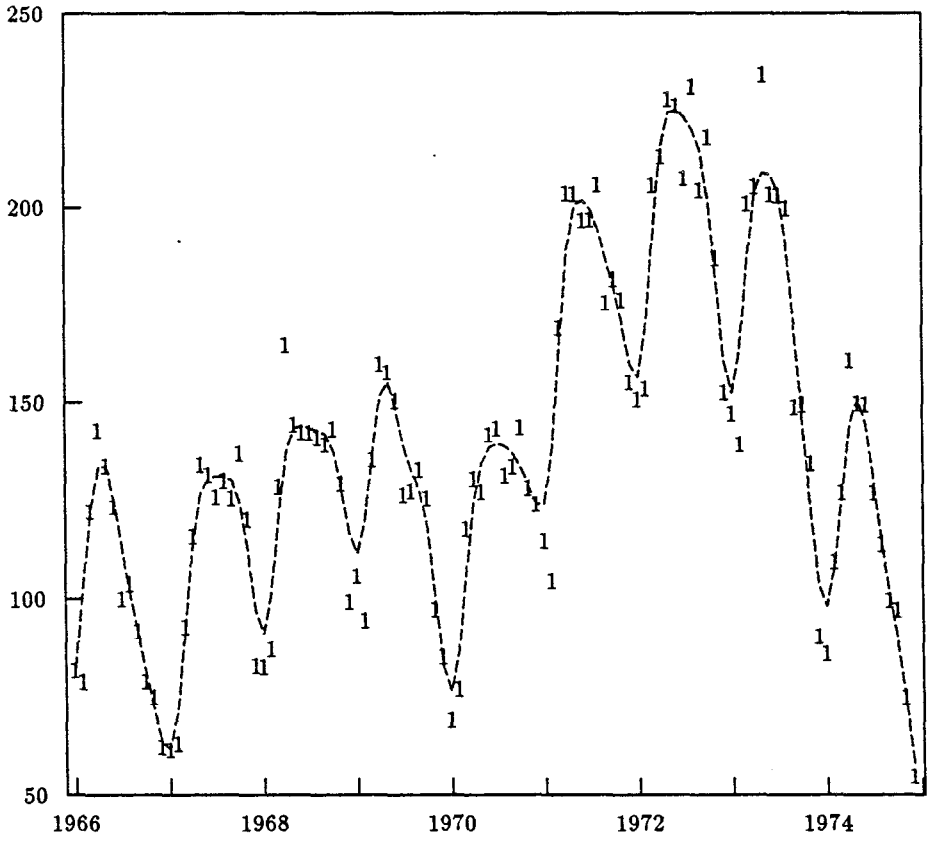


Figure 1 : Série chronologique représentant le nombre de déménagements aux États-Unis. Données brutes et courbe lissée.

2. Le « système d'accueil » S

L'expérience de Consistent System [2] « système d'accueil » développé au MIT sur Multics nous a conduit à rechercher un produit similaire sous UNIX. Notre choix s'est porté sur S, développé par R. A. BECKER et J. M. CHAMBERS d'AT&T Bell Labs [1].

S offre d'abord un système performant de gestion et d'organisation des données de l'utilisateur. Ces données sont par exemple des vecteurs, des matrices, des séries chronologiques, etc.

S possède aussi une large bibliothèque de fonctions statistiques, couvrant des domaines allant des statistiques élémentaires (moyenne, variance...) à l'analyse numérique, en passant par les séries chronologiques par exemple.

Si certains types d'analyses manquent dans S, ils peuvent facilement y être ajoutés, car S possède un système sophistiqué d'interfaçage des programmes écrits par l'utilisateur. Ensuite les programmes ajoutés dans S s'utilisent avec une syntaxe qui reste cohérente avec celle des commandes natives de S. C'est en cela que S est un

≪ système d'accueil ≫. Ainsi S a été enrichi par le département de Biométrie qui y a ajouté des fonctions d'analyse de données, de régression non-linéaire...

S possède aussi un grand choix de fonctions graphiques, permettant de réaliser simplement des graphiques relativement complets. Si en dessin en trois dimensions S a des lacunes, il est en revanche performant dans la représentation de courbes, d'histogrammes... La majeure partie des graphiques produits au laboratoire le sont au travers de S, mais nous ne possédions pas de moyen simple d'utiliser ces graphiques dans un article, un rapport composé avec TEX. La seule solution était de créer un fichier PostScript à partir du dessin produit par S, et d'inclure ce fichier au moment de la traduction du texte du fichier DVI vers le fichier PostScript.

Mais cette solution n'était pas satisfaisante. D'abord il faut noter la lourdeur de sa mise en œuvre, car inclure un fichier PostScript n'est pas immédiat, le résultat dépend du pilote utilisé, et donc la technique n'est pas portable entre DOS et UNIX. Ensuite cette solution souffre de l'absence de possibilité d'utilisation de la puissance de TEX en matière de composition de textes mathématiques, il n'était donc pas possible de mettre par exemple comme légende d'une courbe les mots ≪ Paramètre θ_2 ≫.

La solution idéale consistait donc à écrire un pilote graphique qui permette de produire à partir des fonctions graphiques S un fichier directement interprétable par TEX, qui réalisera lui-même le dessin.

3. Le choix de PICTEX

J'ai décidé de retenir la bibliothèque de macro-instructions PICTEX car c'est celle qui offrait le plus de souplesse. Les possibilités graphiques offertes par LATEX sont trop pauvres et ses extensions EPIC et EEPIC ne sont pas disponibles sur nos machines. Par ailleurs, divers exposés au cours du congrès GUTenberg du mois de mai 1989 m'ont convaincu de la justesse de ce choix [5].

PICTEX [3] est constitué d'un ensemble de macro-instructions TEX permettant de réaliser simplement des graphiques scientifiques tels que courbes, histogrammes...(PICTEX n'est pas adapté au dessin d'art.) TEX ne possède pas à l'origine de moyen de réaliser des graphiques. PICTEX supplée à cela en traçant les courbes à l'aide d'une succession de petits points suffisamment rapprochés pour sembler former une ligne continue.

Ainsi les graphiques réalisés avec PICTEX font partie intégrante du texte et bénéficient de tous les avantages de TEX : indépendance de la machine, du périphérique d'impression... On pourra d'autre part utiliser toute la puissance de TEX pour réaliser des légendes comportant des formules mathématiques.

La version de PICTEX disponible au laboratoire est la version 1.1 du 21 septembre 1987.

Dans le programme que j'ai écrit, je n'utilise qu'un sous-ensemble limité de PICTEX, à savoir le tracé de lignes, la modification du type de trait et de son épaisseur, et le tracé de texte.

Je n'ai volontairement pas utilisé les possibilités offertes par PICTEX pour tracer des filets (*rules*) horizontaux et verticaux car il s'est avéré qu'ils n'étaient pas parfaitement positionnés par rapport aux autres éléments du dessin. Je pense que ce problème est dû au pilote de conversion entre le fichier `.dvi` et le fichier PostScript, mais il est ainsi éliminé.

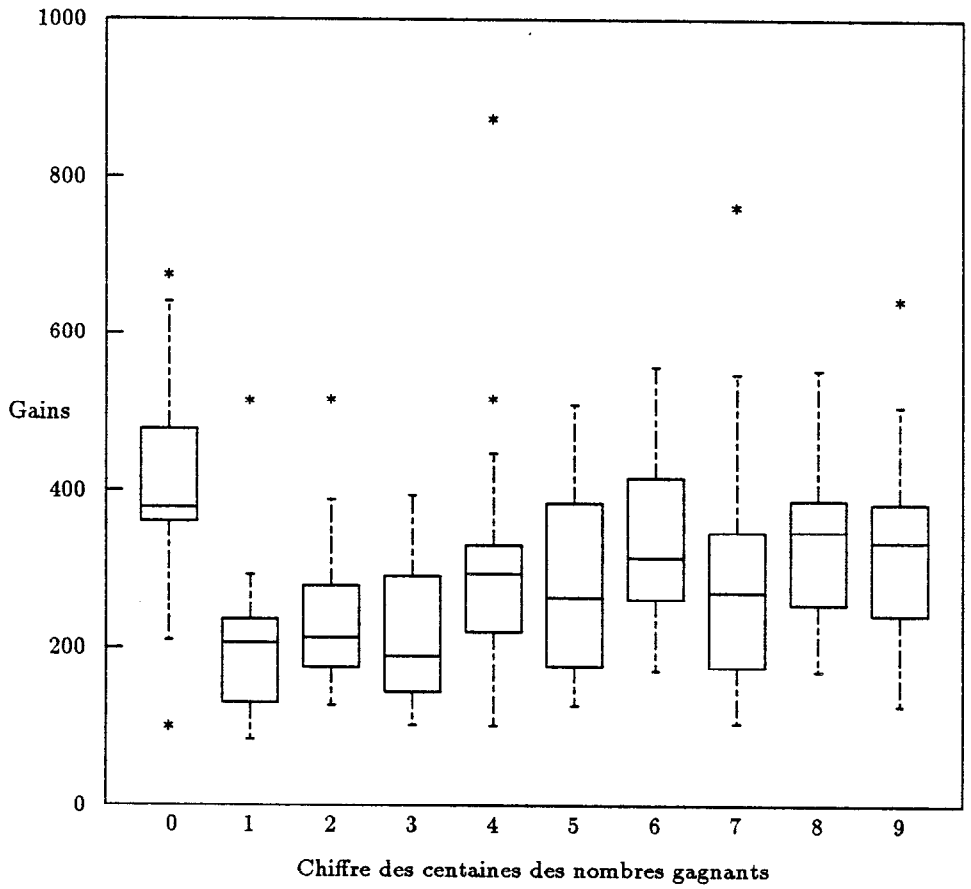


Figure 2 : Boîtes à moustaches représentant les gains en fonction du chiffre des centaines du nombre gagnant. Il faut noter les gains élevés de la première boîte.

4. Le pilote graphique

Il est écrit en C, et divisé en procédures élémentaires pour initialiser le dessin, tracer un trait, écrire du texte, finir le dessin... Ces différentes procédures sont appelées par les fonctions graphiques de S au cours de la réalisation du dessin.

Le graphique sera stocké par le pilote dans un fichier nommé `Slatex.out`, sous forme d'une suite de commandes PICTEX. Ce fichier pourra ensuite être inclus dans un texte TEX.

4.1. L'initialisation

Cette procédure comporte l'initialisation de variables utilisées par les fonctions graphiques de S, comme par exemple la taille maximale du dessin ou la possibilité d'écrire autrement qu'horizontalement.

```
F77_SUB(zparmz)(par,n)
float par[] ; long *n ;
{
  int i ;
  for(i=1 ; i<=39 ; i++) am(i)=0. ;
  am(20)=700. ; /* taille des caracteres */
  am(21)=700. ;

  am(22)=0. ; /* taille du dessin en x */
  am(23)=36000. ;

  am(24)=0. ; /* taille du dessin en Y */
```

```
am(25)=36000. ;

am(26)=.5 ; /* position du caractere par */
am(27)=.5 ;  /* rapport au point courant */
/* ici le caractere est centre */

am(28)=5./36000. ; /* taille du dessin */
am(29)=am(28) ; /* en pouces */

am(31)=0. ; /* pas de rotation */
/* des carateres */
am(1)=1. ; /* caracteres de taille */
/*  variable */
```

L'unité retenue est le centième de point, le graphique fera donc au maximum 360 × 360 points, soit environ 12,5 centimètres. Les coordonnées fournies par les fonctions graphiques de S sont des valeurs entières ; pour avoir une précision suffisante, il a fallu calculer en centièmes de point et définir une zone de 36000 × 36000.

Ensuite le fichier est initialisé :

```
  /* ouverture du fichier */
  fp=fopen("Slatex.out","w") ;
  /* initialisation de PiCTeX */
  fprintf(fp,
"\\bgroup\\beginpicture\\catcode'!=11\n") ;
  fprintf(fp,"\\setplotarea x from 0 to ") ;
  fprintf(fp,"360, y from 0 to 360\n") ;
  fprintf(fp,
"\\setcoordinatesystem units <1pt,1pt>\n") ;
```

Le dessin est défini comme un groupe TEX et initialisé par la commande `\beginpicture`, puis le code du ! est redéfini, car c'est le caractère utilisé par PICTEX pour protéger ses commandes internes. La zone de dessin est définie pour faire 360 points au carré et l'unité utilisée par PICTEX sera le point. Il faut noter que les caractères \ doivent être doublés en C pour apparaître correctement dans le fichier résultat.

Le fichier contenant les commandes PICTEX sera écrasé à chaque passage dans la procédure d'initialisation, donc à chaque initialisation d'un graphique.

Enfin deux variables globales à l'emsemble des procédures sont initialisées, elles indiquent respectivement que l'on est en train de tracer un trait et que le graphique n'est pas vide.

```
  in_put=0 ;
  in_plot=0 ;
}
```

4.2. Le positionnement du point courant

La procédure `zseekz` permet de donner au point courant les coordonnées `ix` et `iy` passées en paramètre.

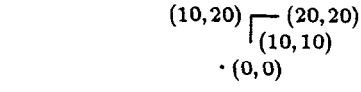
```
F77_SUB(zseekz)(ix,iy)
long int *ix,*iy ;
{
  if (in_put == 1)
     {
       fprintf(fp,"} [lb] at %f %f\n",
              cur_x, cur_y) ;
       in_put=0 ;
     }
```

Dans un premier temps, si on était en train de tracer un trait, on le termine. Cet ensemble d'instructions est présent au début de nombreuses procédures de tracé.

Cela est dû au fait que si plusieurs traits sont tracés à la suite, ils sont enchaînés dans la même commande de tracé \put. Par exemple, si l'on trace un trait avec l'ensemble de coordonnées suivant : $\{(10,10)(10,20)(20,20)\}$, on génère une seule commande :

```
\put{\!start(10,10)
\!ljoin(10,20)
\!ljoin(20,20)
} [lb] at 0 0
```

C'est plus compact qu'avec une commande \put pour $\{(10,10)(10,20)\}$ et une autre pour $\{(10,20)(20,20)\}$. Et voici le résultat produit :



Mais cette procédure permet aussi de modifier le type de trait et son épaisseur. Le nouveau type de trait est spécifié dans `am(8)` :

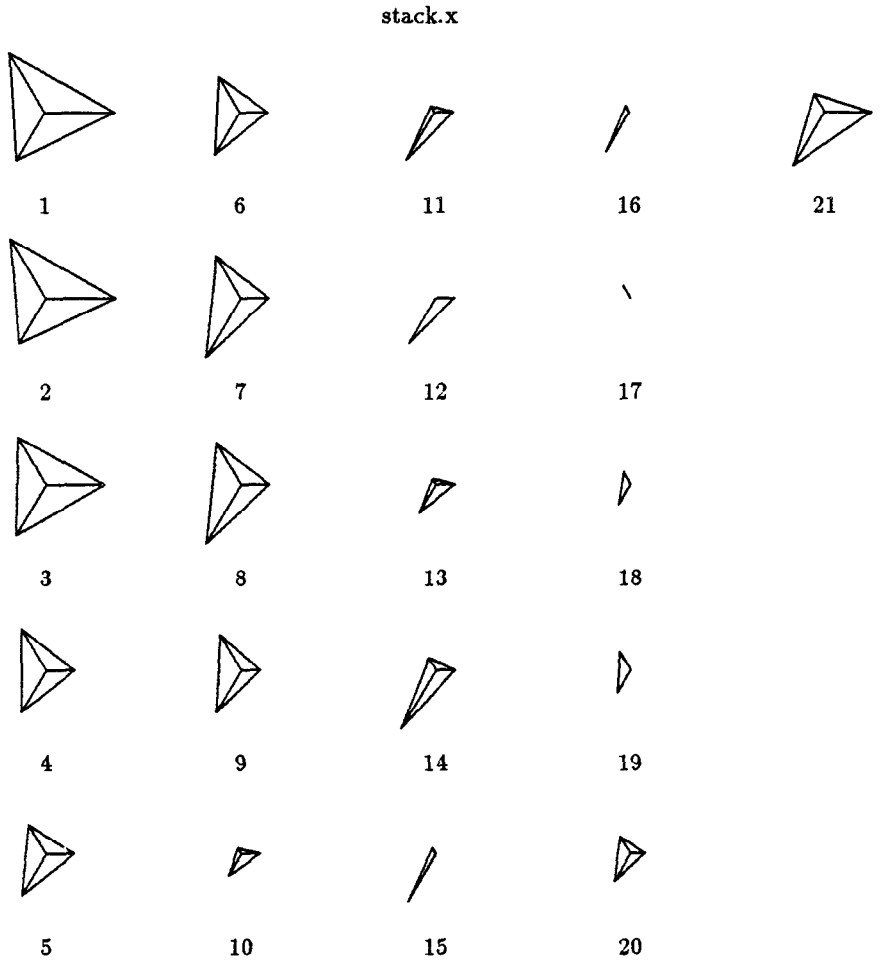


Figure 3 : Graphique en étoile représentant trois variables sur chacune des vingt observations.

```
if(line_type != am(8))
   {
     /* modification du type de trait */
     line_type=am(8) ;
     if (line_type <= 1.)
       fprintf(fp,"\\setsolid\n") ;
     else if (line_type <= 2.) fprintf(fp,
         "\\setdashpattern <4pt,2pt>\n") ;
     else if (line_type <= 3.)
       fprintf(fp,"\\setdots <2pt>\n") ;
...
   }
```

Selon la valeur du type de trait on ajoute dans le fichier une commande PICTEX pour passer en trait plein, en tirets longs, en pointillés...

De même, `am(9)` indique la nouvelle épaisseur du trait :

```
if(line_width != am(9))
   {
     /* modif de l'epaisseur du trait */
     line_width=am(9) ;
     if (line_width <= 1.) fprintf(fp,
         "\\setplotsymbol({\\fivrm .})\n") ;
     else if (line_width <= 2.) fprintf(fp,
         "\\setplotsymbol({\\sixrm .})\n") ;
...
     else fprintf(fp,
         "\\setplotsymbol({\\twfvrm .})\n") ;
   }
```

Le trait est toujours dessiné à l'aide de petits points à des tailles différentes. Les tailles utilisées correspondent à celles définies par LATEX, `\fivrm`, `\sixrm`... `\twfvrm`.

Enfin la procédure stocke la valeur du

point courant dans les variables cur_x et cur_y. Il s'agit de la valeur des variables ix et iy, passées en paramètre, divisée par 100, pour les raisons expliquées au paragraphe précédent.

```
  cur_x=((float) *ix) /100. ;
  cur_y=((float) *iy) /100. ;
}
```

4.3. Le tracé d'un trait

Cette procédure permet de tracer un trait allant du point courant jusqu'au point spécifié par les paramètres ix et iy.

```
F77_SUB(zlinez)(ix,iy)
long int *ix, *iy ;
{
  x=((float) *ix)/100. ;
  y=((float) *iy)/100. ;
```

On commence par calculer les valeurs x et y du point d'arrivée du trait en divisant ix et iy par 100.

Ensuite on initialise une commande de tracé \put si besoin est.

```
  if(in_put == 0)
    {
      /* initialisation du trait */
      fprintf(fp,
              "\\put{\\!start(%f,%f)\n",
              cur_x,cur_y) ;
      cur_x=0. ;
      cur_y=0. ;
      in_put=1 ;
    }
```

L'initialisation de la commande \put consiste surtout à indiquer le point de départ du trait, qui est le point courant stocké dans cur_x et cur_y. La commande de tracé d'un trait est placée aux coordonnées $(0,0)$, c'est pour cela que cur_x et cur_y prennent ensuite la valeur 0.

Ensuite on trace le trait proprement dit.

```
  fprintf(fp,"\\!ljoin(%f,%f)\n",x,y) ;
  in_plot=1 ;
}
```

On ajoute pour cela une commande \!ljoin dans le fichier PICTEX. La variable in_plot sert à indiquer que l'on a effectivement commencé à dessiner quelque chose.

4.4. L'écriture de texte

La procédure ztextz permet d'écrire du texte sur le graphique. Au point de vue du pilote de S, ce n'est pas une procédure élémentaire. La procédure élémentaire correspondante ne permet d'écrire qu'un seul caractère à la fois, ce qui n'a pas d'intérêt par rapport à TEX. Si elle existe, la procédure ztextz est la seule utilisée pour les caractères. De ce fait, le code de la procédure ztextz est moins immédiat que celui des autres procédures de ce pilote.

Les paramètres xx et yy indiquent la position de la chaîne de caractères à écrire. Celle-ci est contenue dans buf et a une longueur n. Le paramètre pos indique si la chaîne doit être centrée, appuyée à gauche ou appuyée à droite par rapport au point (xx, yy).

```
F77_SUB(ztextz) (xx, yy, buf, n, pos, m)
float *xx, *yy, *pos ;
char buf[] ;
int *n, m ;
{
  /* Test de chaine vide */
  if (*n<=0)
    {
      F77_SUB(zejecz)() ;
      F77_SUB(zerrpz)("ztextz",
        "Number of characters not positive",
        6L,34L) ;
    }
```

Si la chaîne de caractères est vide, c'est alors une erreur et il ne faut rien écrire.

Sinon il faut d'abord fermer la commande de tracé de trait en cours si elle existe.

```
  if(in_put == 1)
    fprintf(fp,"} [lb] at %f %f\n"
      ,cur_x, cur_y) ;
```

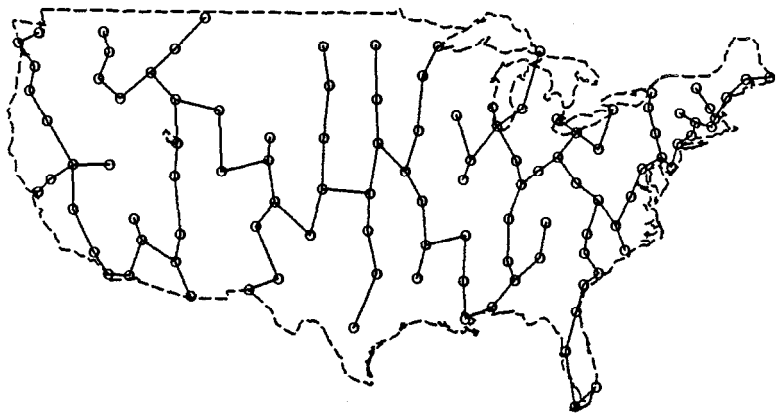


Figure 4 : Arbre de recouvrement minimal reliant les grandes villes des États-Unis. L'arbre tient approximativement compte de la rotondité de la terre.

Puis on modifie le corps des caractères en fonction de la valeur de `am(18)/am(89)`. Cette procédure n'étant pas élémentaire on n'a pas accès directement à la valeur du corps à utiliser, mais on doit la recalculer.

```
if(size!=am(18)/am(89))
  {
    /* modif du corps des caracteres */
    size=am(18)/am(89) ;
    if (size <= .45) fprintf(fp,"\\tiny\n") ;
...
    else if (size <= 1.05)
          fprintf(fp,"\\small\n") ;
    else if (size <= 1.25)
          fprintf(fp,"\\normalsize\n") ;
...
    else fprintf(fp,"\\Huge\n") ;
  }
```

On utilise les commandes LaTeX pour modifier le corps des caractères : `\tiny`... `\small`... `\Huge`. Le corps utilisé par défaut, correspondant à une taille comprise entre 0,95 et 1,05 est le corps `\small`.

Enfin on ajoute dans le fichier PiCTeX la commande `\put` de tracé de texte proprement dite.

```
in_put=0 ;
/* debut du trace de texte */
fprintf(fp,"\\put{") ;
/* ecriture des caracteres du texte */
for(i=*n ;i-- ;buf++)
  fprintf(fp,"%c",*buf) ;
/* calcul des coordonnees */
x=(*xx*am(37)+am(36))/100. ;
y=(*yy*am(39)+am(38))/100. ;
/* fin du trace et positionnement *.
if (*pos <= 0.33)
  fprintf(fp,"} [l ] at %f %f\n",x,y) ;
else if (*pos <= .66)
  fprintf(fp,"} [  ] at %f %f\n",x,y) ;
else fprintf(fp,"} [r ] at %f %f\n",x,y) ;
in_plot=1 ;
}
```

Le texte est copié caractère par caractère dans le fichier PICTEX à l'aide de la boucle `for`. Puis les coordonnées exactes sont calculées, avec toujours la division par 100. Enfin, selon la valeur du paramètre `pos`, la chaîne est positionnée : appuyée à gauche `[l ]`, centrée `[ ]` ou appuyée à droite `[r ]`.

On termine en indiquant par la variable `in_plot` que quelque chose a bien été dessiné.

4.5. Le changement de graphique

Cette procédure est appelée à chaque changement de graphique, pour terminer le dessin en cours et commencer le suivant.

```
F77_SUB(zejecz)()
{
  if(in_put == 1)
    {
      fprintf(fp,
        "} [lb] at %f %f\n",
        cur_x, cur_y) ;
      in_put=0 ;
    }
```

Encore une fois s'il existe une commande de tracé de trait en cours, il faut la fermer.

Puis on va terminer le dessin courant et initialiser le suivant.

```
  /* teste si on a dessine */
  if(in_plot == 1)
    {
      /* fin du dessin en cours */
      /* et changement de page */
      fprintf(fp,
"\\endpicture\\egroup\\clearpage\n") ;
      /* dessin suivant */
      fprintf(fp,
"\\bgroup\\beginpicture\\catcode`!=11\n") ;
      fprintf(fp,"\\setplotarea x from 0 to") ;
      fprintf(fp," 360, y from 0 to 360\n") ;
      fprintf(fp,
"\\setcoordinatesystem units <1pt,1pt>\n") ;
    }
}
```

Pour des raisons internes aux fonctions graphiques de S, la procédure `zejecz` est appelée immédiatement à la suite de la procédure `zparmz` d'initialisation. Pour éviter que l'initialisation ait lieu deux fois, on n'effectue celle-ci que si quelque chose a été effectivement dessiné. On teste donc la variable `in_plot`.

Terminer le graphique en cours consiste à fermer le dessin par la commande `\endpicture` puis à fermer le groupe TEX. Ensuite on change de page.

Pour initialiser le nouveau graphique, on utilise exactement les mêmes commandes que celles de la procédure `zparmz`.

4.6. La sortie du pilote

La procédure `zwrapz` est appelée lorsque S sélectionne un autre pilote graphique, ou lorsque l'utilisateur quitte le système S. Elle sert essentiellement à fermer le fichier PICTEX.

```
F77_SUB(zwrapz)()
{
  if(in_put == 1)
    {
      fprintf(fp,
        "} [lb] at %f %f\n",
        cur_x, cur_y) ;
      in_put=0 ;
    }
```

On termine la commande de tracé de trait éventuellement en cours.

Puis on va terminer le graphique.

```
  fprintf(fp,"\\endpicture\\egroup\n") ;
  fclose(fp) ;
}
```

Les commandes PICTEX de fin de graphique sont les mêmes que celles de la procédure précédente. Ensuite on ferme simplement le fichier PICTEX avec l'instruction `fclose`.

5. Mise en œuvre du pilote

5.1. Du point de vue S

Le pilote s'utilise exactement comme les autres pilotes graphiques, il s'appelle `latex`. Quand le pilote est lancé, tous

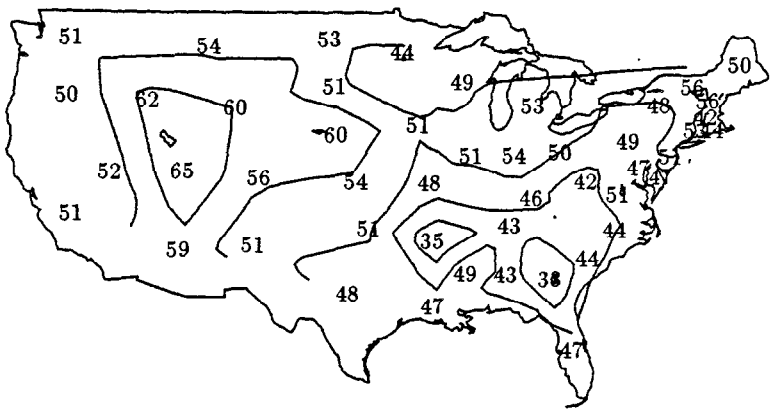


Figure 5 : Pourcentage du vote républicain aux élections présidentielles de 1976 aux États-Unis.

les ordres graphiques viennent ajouter des commandes PICTEX dans le fichier de nom `Slatex.out`.

Il faut prendre la précaution de renommer le fichier créé afin qu'il ne soit pas écrasé par le prochain appel du pilote.

La seule contrainte que je n'ai pas réussi à éliminer, car elle me semble due au fonctionnement même de S, est que si l'on veut introduire un \ dans une chaîne de caractères, il faut le quadrupler; par exemple pour mettre comme légende de l'axe des x « Paramètre θ_2 », il faudra taper `xlab="Param\\\\'etre $\\\\theta_2$"`. Mais cela peut aussi être modifié simplement dans le fichier PICTEX.

Dans une version suivante du pilote, il sera possible de préciser deux arguments optionnels lors de l'activation du pilote. Ces arguments donneront la hauteur et la largeur, exprimées en points typographiques, du graphique.

5.2. Du point de vue TEX

Le pilote a été écrit pour être utilisé avec LATEX. Il faut ajouter dans le fichier contenant le texte à composer les commandes de chargement de la bibliothèque de macro-instructions PICTEX. Cela se fait à l'aide de trois commandes `\input`, qui doivent être placées après l'instruction `\documentstyle`. Le fichier aura donc par exemple l'allure suivante :

```
\documentstyle{...}
\input prepictex
\input pictex
\input postpictex
```

```
\begin{document}
...
```

Les fichiers `prepictex` et `postpictex` sont nécessaires pour utiliser PICTEX avec LATEX.

Ensuite le fichier PICTEX est chargé dans le texte par une autre commande `\input` à l'endroit où l'on souhaite que le graphique apparaisse. Par exemple :

```
\'a l'endroit o\'u l'on souhaite
que le graphique apparaisse.
Par exemple :

\begin{figure*}
\input dessin29
\caption{Pourcentage du vote...}
\end{figure*}
```

Il me semble conseillé d'utiliser la commande `\caption` (qui existe dans l'environnement `figure` de LATEX) pour mettre des titres sur les graphiques. On pourra ainsi faire référence automatiquement à ces graphiques par le mécanisme `\label-\ref` de LATEX.

6. Conclusion

Le pilote graphique entre S et PICTEX représente un des maillons manquants à la « boîte à outils » logicielle idéale telle qu'elle a été imaginée par François CHAHUNEAU [4].

La production d'un rapport à la suite d'un travail statistique est grandement améliorée par ce système simple d'inclusion des données sous forme graphique, sans avoir de gymnastique compliquée à faire.

Les deux points faibles de ce système restent d'une part sa lenteur, il faut une bonne dizaine de minutes pour composer un dessin raisonnablement compliqué sur un Sun3/50. D'autre part si les graphiques sont trop denses, ils font « exploser » la mémoire de TEX. Cela est encore plus critique du fait que les traits horizontaux et verticaux ne sont pas tracés au moyen de filets.

Si l'on se limite à des graphiques assez simples, je pense que ce système est quand même très utile.

Tous les exemples graphiques sont tirés du manuel d'introduction à S [1].

Références bibliographiques

[1] R. A. BECKER, *S: An Interactive Environment For Data Analysis And Graphics* Wadsworth ed., 1984.

[2] J. P. LEY *et al.*, *Manuel Consistent System*, ouvrage commun du laboratoire de Biométrie et de la mission Informatique, publ. INRA, 1985.

[3] M. J. WICHURA, *The PICTEX Manual*, TEXniques, Publications for the TEX Community, #6, distr. TUG, 1987

[4] F. CHAHUNEAU et O. NICOLE, *Une opération pilote « réseau » à l'INRA*, Technique et Science Informatiques, vol. 7, n° 2, 1988.

[5] Cahiers GUTenberg, Actes du congrès GUTenberg 1989, n° 2, 1989.