

# *Cahiers* **GUT** *enberg*

☞ L'INCORPORATION DE GRAPHIQUES DANS  
INRST<sub>E</sub>X

☞ Michael J. FERGUSON

*Cahiers GUTenberg*, n° 2 (1989), p. 81-89.

<[http://cahiers.gutenberg.eu.org/fitem?id=CG\\_1989\\_\\_2\\_81\\_0](http://cahiers.gutenberg.eu.org/fitem?id=CG_1989__2_81_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.



## L'incorporation de graphiques dans INRST<sub>E</sub>X

---

Michael J. FERGUSON

*INRS-Télécommunications  
Montréal, Canada*

### Sommaire

Cet article discutera de l'approche utilisée pour l'incorporation de graphiques dans INRST<sub>E</sub>X. La philosophie de base est d'utiliser la capacité de l'imprimante à créer des graphiques et non l'introduction de caractères spéciaux. La capacité de générer des graphiques a été réalisée pour PostScript et QMS graphiques au laser QUIC. Le rôle de l'imprimante sera discuté de même que le besoin d'un logiciel de soutien pour permettre l'incorporation de graphiques produits par d'autres systèmes. Finalement, il y aura une discussion générale au sujet de l'impact de quelques-unes des décisions de design sur le système de graphique du T<sub>E</sub>X.

### 1. Introduction

Ce compte-rendu décrit l'approche utilisée pour incorporer des graphiques dans INRST<sub>E</sub>X. INRST<sub>E</sub>X a été décrit dans plusieurs réunions des Groupes d'utilisateurs T<sub>E</sub>X (TUG) dont notamment à la réunion tenue à Strasbourg en juin 1986 intitulée « T<sub>E</sub>X pour la documentation scientifique ». Le compte-rendu décrit quelques caractéristiques de l'INRS-T<sub>E</sub>X et, ce qui est plus important, il décrit la syntaxe et les caractéristiques de l'option graphique, appelée T<sub>E</sub>XGraph. Plutôt que de discuter de la syntaxe et des capacités de T<sub>E</sub>XGraph, ce rapport discute de quelques-unes des considérations et leçons apprises en créant et en implémentant T<sub>E</sub>XGraph.

### 2. Approches pour ajouter des graphiques à T<sub>E</sub>X

Il y a deux approches principales pour incorporer des graphiques à T<sub>E</sub>X. La première place les caractères à une position précise sur la page tandis que la seconde exploite les possibilités d'imprimer en utilisant la commande T<sub>E</sub>X `\special`. La commande `\special` permet d'envoyer l'information directement à l'imprimante en passant par le programme du « contrôleur d'imprimante ».

L<sub>A</sub>T<sub>E</sub>X [1] était un des premiers programmes de T<sub>E</sub>X à introduire des graphiques. Il utilise la première approche en créant une image et en plaçant des caractères graphiques spéciaux à des coordonnées précises sur la page. Quoiqu'il soit possible de créer des graphiques plus complexes en plaçant précisément de simples points, tel que `< · >`, une partie de la difficulté est réduite en incluant des formes spéciales dans la police de caractères en y ayant accès comme à de simples caractères.

Le travail de spécification peut-être réduit davantage en bâtissant des macro-instructions qui peuvent bâtir des images par une simple évocation de séquences. Wichura a créé un logiciel très puissant, le PICT<sub>E</sub>X [2], qui, quoique basé sur la matrice à point, inclut de puissants macros utilisant ce procédé. L'avantage principal de cette approche est qu'elle ne nécessite rien d'autre de l'imprimante au-delà du procédé normal d'imprimerie T<sub>E</sub>X. Le désavantage principal est le besoin de caractères spéciaux, une restriction à de petits dessins imposée par la structure in-

terne de  $\text{T}_{\text{E}}\text{X}$ , et l'extrême difficulté d'annoter les figures externes.

Le module  $\text{T}_{\text{E}}\text{XGraph}$  de  $\text{INRST}_{\text{E}}\text{X}$  utilise la seconde approche. Il accède et utilise les capacités graphiques de l'imprimante par la commande  $\backslash\text{special}$  et améliore la qualité des spécifications par des macros  $\text{T}_{\text{E}}\text{X}$ . Ceci permet un grand choix de graphiques vectorielles et permet l'insertion de graphiques externes à n'importe quel point de l'image. Quand on inclue un graphique externe,  $\text{T}_{\text{E}}\text{XGraph}$  assume que le « header » nous informe de l'espace requis par le dessin. Quoique cette méthode permet l'insertion relativement simple de graphiques externes, la création d'une librairie de graphiques et l'annotation de graphiques externes, il est cependant dépendant de la capacité du langage de l'imprimante et, malheureusement, du comportement du programme  $\text{T}_{\text{E}}\text{XPrint}$  - le programme qui traduit le fichier  $\text{T}_{\text{E}}\text{X} \backslash\text{tt} *.dvi$  en langage d'imprimerie.

### 3. $\text{T}_{\text{E}}\text{XGraph}$ – Histoire et Caractéristiques

$\text{T}_{\text{E}}\text{XGraph}$  a été introduit comme faisant partie intégrale de  $\text{INRST}_{\text{E}}\text{X}$  en 1985 et a été une des raisons majeures de l'acceptation de  $\text{T}_{\text{E}}\text{X}$  à l'INRS-Télécommunications et à BNR. Le système INRS utilise l'imprimante QMS-LASERGRAPHIX 1200, le langage graphique QUIC [3] et une version de  $\text{T}_{\text{E}}\text{XPrint}$  produite à l'INRS. Cette version de  $\text{T}_{\text{E}}\text{XPrint}$  incorporait la gestion de la police de caractères et interprétait les différents types de commande  $\backslash\text{special}$  correspondant au nom d'un fichier; il mettait aussi le code QUIC en ligne et changeait la direction de l'impression de horizontal à vertical. En fait, il incorporait des instructions pour imprimer dans plusieurs directions de sorte que  $\text{T}_{\text{E}}\text{XGraph}$  pouvait être rappeler plusieurs fois et être toujours en mesure de

faire un travail adéquat. De plus, il changeait de la position d'origine à la position sur la page où la commande  $\backslash\text{special}$  était introduite. Ceci permettait de positionner des graphiques n'importe où sur la page.

Pour rendre utile l'insertion de graphiques, nous avons produit un certain nombre de logiciels qui transposent les sources ordinaires de graphiques en langage graphique QUIC, et inclus une entête indiquant sa grandeur. Comme la plupart de ces sources utilisent une imprimante au laser, il était profitable de développer ces logiciels. Ces sources incluaient un « scanner » de document et des graphiques produits par le logiciel de graphiques faisant partie de notre système FORTRAN. Récemment il nous a été nécessaire d'inclure des graphiques créés sur le MacIntosh avec des logiciels tels que MacDraw [4]. Quoiqu'il était possible d'inclure n'importe quelle forme d'image « raster », il était impossible d'inclure des images MacDraw dans le système QUIC à moins de créer une image « raster ». L'insertion de MacDraw devrait être plus facile dans le système PostScript [5] sauf pour les problèmes de mémoire de l'imprimante provoqués par l'entête de MacDraw. Le complément exact des logiciels requis pour la transposition de graphiques est évidemment dépendant de l'endroit, mais des programmes de soutien sont aussi requis.

$\text{T}_{\text{E}}\text{XGraph}$  a aussi été introduit pour PostScript et il travaille présentement avec un certain nombre de logiciels  $\text{T}_{\text{E}}\text{XPrint}$ . Un exposé traitant de quelques-uns des prérequis imposés à ces logiciels se trouve à la Section 6. Les différences dans les langages QUIC et PostScript a amené des incompatibilités dans les deux versions. Celles-ci sont discutées dans la Section 7. Quoiqu'il serait bien d'inclure l'imprimante Hewlett-Packard LASER JET, ce langage d'imprimante

n'a pas de capacité graphique autre que l'insertion de petites images « raster ». Le support graphique est trop faible pour justifier une insertion complète de *T<sub>E</sub>X*-Graph.

En résumé, *T<sub>E</sub>X*Graph apporte les caractéristiques suivantes:

- La production d'un rapport complet ne nécessitant aucune insertions additionnelles (sauf peut-être des photographies couleur si requises)
- La capacité d'insérer un fichier, par exemple une photo, qui a été transposé en langage d'impression incluant l'information sur la grandeur nécessaire pour *T<sub>E</sub>X*Graph. Ceci requiert un groupe de logiciels qui change la transposition.
- Le mélange de graphiques vectoriels et des polices de caractères *T<sub>E</sub>X* dans le texte. Avec PostScript et quelques prérequis appropriés sur l'imprimante, le texte, utilisant les polices de caractères *T<sub>E</sub>X*, peut maintenant être imprimé à n'importe quel angle.
- La superposition et l'annotation d'un fichier de textes/graphiques.
- L'insertion des images produites à l'extérieur, tel que celles produites sur le MacIntosh.

Quand ceci est combiné avec la capacité de macro du *T<sub>E</sub>X*, ceci permet la production d'une très grande librairie de graphiques. *T<sub>E</sub>X*Graph a déjà été utilisé pour produire les quelques 200 images d'un livre. L'auteur a adapté la syntaxe de *T<sub>E</sub>X*Graph selon ses besoins et a créé une librairie d'images. Un livre de texte est produit assez communément et en assez grande quantité pour rendre valable cette approche de librairie de graphiques.

## 4. Création et Insertion de Graphiques

Une des difficultés majeures qui survient quand on discute de combiner des graphiques et du texte est le manque d'uniformité dans plusieurs aspects du sujet. C'est peut-être pour cette raison que le support pour les graphiques dans *T<sub>E</sub>X* est tellement limité. Les graphiques, et les principes de conception relatifs à leur insertion, ne sont pas facilement transposables en structures linguistiques. Excepté pour quelques cas, tels que quelques livres et manuels techniques, il y a tellement peu de points communs entre les images que la construction d'images par de simples séquences est presque impossible.

La création et l'insertion d'entités graphiques constituent deux aspects séparés mais reliés. Un auteur, spécialement celui qui produit un rapport, est habituellement responsable pour les deux aspects, mais un éditeur ne peut-être responsable que pour les décisions se rapportant à l'insertion de graphiques et à leur disposition. Les différences de perspectives peuvent très bien déterminer la « valeur » de n'importe quel système. *INRST<sub>E</sub>X* et *T<sub>E</sub>X*Graph accomodent les auteurs. Nous présumons que la création et l'insertion de graphiques sont importants.

### 4.1 Création d'Entités Graphiques

La source, la quantité et la régularité des graphiques sont des questions fondamentales pour créer les insertions graphiques de tout document. De plus, il faut choisir la police de caractères d'impression quand il faut inclure du texte avec les graphiques.

Comme tout le monde le sait, les insertions graphiques proviennent de plusieurs sources. Voici une liste d'exemples, probablement très connue:

- texte sur papier
- fax, scanner
- fichier du logiciel
- CAD, MacDraw ...

Il faut nécessairement avoir un ensemble de logiciels pouvant transposer ces sources sous une forme compatible avec l'imprimante pour laquelle ils sont destinés ou au moins sous une forme intermédiaire qui peut-être facilement transposée

Le nombre de graphiques est un facteur important, souvent oublié, dans le processus créatif. L'organisation et la production est fortement influencée d'après le nombre à produire, soit 1, 10, 100 ou 1000 images. Tous les graphiques sont produits un à la fois. Cette réalité semble être l'origine de l'impression que tout graphique est unique. Cependant, au fur et à mesure que le nombre de graphiques augmente, l'accès, l'utilisation et la modification de graphiques déjà existants devient beaucoup plus important. Il est clair que la plupart, peut-être tous les gens trouvent difficile d'évaluer l'utilité d'un graphique d'après une description linguistique. Ceci implique essentiellement un mécanisme de visionnement de graphiques. Donc, la capacité d'éditer et de changer des graphiques déjà existant devient plus importante. Ceci signifie que la représentation doit être accessible à l'auteur et doit être sous forme permettant l'édition. Le besoin d'une structure de base augmente proportionnellement avec le nombre de graphiques. De plus, une présentation structurée permet une plus grande précision au niveau du visionnement et de la signification sémantique du graphique. Des diagrammes en bloc, avec ou sans icônes, sont un exemple important de cette catégorie.

La question de régularité dans un groupe de graphiques est confuse par le niveau d'importance de la représentation.

En général, tous les graphiques existent à un différent niveau de représentation. Les niveaux importants sont ceux qui permettent l'édition et la modification appropriée. Une représentation « raster », bien qu'universelle, n'est pas très utile pour éditer, modifier ou annoter. Une représentation de graphiques vectoriels est probablement mieux mais encore à un niveau très au-dessous des objets qui composent une grande proportion des insertions graphiques. Quoiqu'une représentation basée sur l'objet ait des avantages, nous devons avoir foi dans la transposition des objets à des graphiques pour être satisfait de cette description. Finalement, il y aura toujours un besoin pour figurer le produit final. Ceci est certainement fait plus efficacement sur un écran.

Un point principal qui aide à déterminer l'efficacité des graphiques, spécialement quand ils sont inclus dans un texte, est de pouvoir les visualiser facilement. Pour la majorité des gens, ceci signifie qu'ils soient imprimés sur papier, ou représentés sur un écran. Malheureusement la plupart des logiciels ne permettent pas de voir à l'intérieur de la commande `\special`. Ceci a changé avec l'introduction de « Display PostScript ». L'introduction de l'équivalent de Display PostScript permettrait de voir le document au complet et aurait aussi l'avantage de ne requérir qu'un seul logiciel `TeXPrint`, probablement un logiciel qui interagit mieux que ceux communément employés maintenant.

La conception et des intérêts pragmatiques entre tous les deux en considération dans la sélection de la police de caractères pour graphiques. Dans le passé, la police de caractères **simple sans serif** étaient d'usage parce qu'ils étaient tout ce que l'utilisateur pouvait produire facilement. Maintenant la technologie permet d'utiliser pratiquement toutes les sortes

de caractères. C'est une décision importante de conception à savoir si la police de caractères du graphiques doivent être assortis ou contraster avec ceux du texte du document principal. Le cas le plus courant en ce moment est de faire contraster, encore une fois à cause du processus utilisé pour produire des graphiques. C'est devenu une pratique tellement courante que nous avons tendance à suggérer que c'est une bonne conception. Que cela produise ou non des graphiques plus esthétiques ou plus sémantiquement significatifs, ceci reste à voir. Si le texte d'un graphique peut-être éditer indépendamment, alors il est probablement très simple de le changer de document en document. La police de caractères **simple sans serif** a l'avantage additionnel de pouvoir être réduit ou augmenter et de tourner très facilement sur des appareils à faibles intensité (300 dpi). Nous devons nous assurer en tout temps que les caractères choisis seront agréables et lisibles. Sinon, alors la sélection de la police de caractères la plus appropriée est celle qui contraste agréablement avec presque tous les polices de caractères des documents. A cause de son acceptation très répandue, la police de caractères **simple sans serif** semble approprié si le texte d'un graphique n'est pas facilement accessible. Le texte d'un graphique produit par  $\text{TExGraph}$  peut-être éditer séparément et peut donc être choisi à volonté.

#### 4.2 Insertion des Entités Graphiques

$\text{TEx}$  est orienté en fonction des paragraphes et des boîtes. Il compose des paragraphes probablement mieux que tout autre système sur le marché, mais il le fait en prenant en considération une vue globale du paragraphe.  $\text{TEx}$  doit connaître la forme exacte du paragraphe avant de le composer. Si ceci représentait la

somme totale de l'optimisation de  $\text{TeX}$ , alors il serait facile d'inclure du texte autour d'insertions et d'images. Cependant, avec  $\text{TeX}$ , le problème est plus compliqué par le fait que  $\text{TEx}$  ne sait pas où sur la page un certain paragraphe sera situé. Ceci est déterminé seulement après que  $\text{TEx}$  a décidé où se situera la fin de la page. Ceci signifie que le paragraphe qui doit s'enrouler autour d'un graphique ne peut-être déterminer qu'après que la page est fini ... ce qui est trop tard. Une solution possible à ce problème est de définir des sous-pages de même grandeur et de les remplir dans l'ordre nécessaire pour s'enrouler autour du graphique. Cette technique, peut-être la seule qui fonctionnera avec  $\text{TEx}$ , comporte plusieurs problèmes. En voici une courte liste:

- $\text{TEx}$  ne permet que des pages rectangulaires. Ceci est un problème mineur dans la plupart des cas.
- La définitions des sous-pages doit se faire dans le mode  $\backslash\text{output}$ . Pour le saut de page automatique, le format de page choisi doit être dépendant de l'existence d'un graphique. Ceci doit être présenté dans le document avant que la page ne soit commencée. Ceci n'est pas impossible mais requièrera certainement quelque « iteration ». De plus, le mode  $\backslash\text{output}$  doit être assez versatile pour accomoder plusieurs formes possibles. Celle qui sera appropriée sera probablement seulement déterminée par expérimentation. Une fonction additionnelle dans la programmation du  $\text{TEx}$  serait nécessaire pour permettre cette expérimentation.

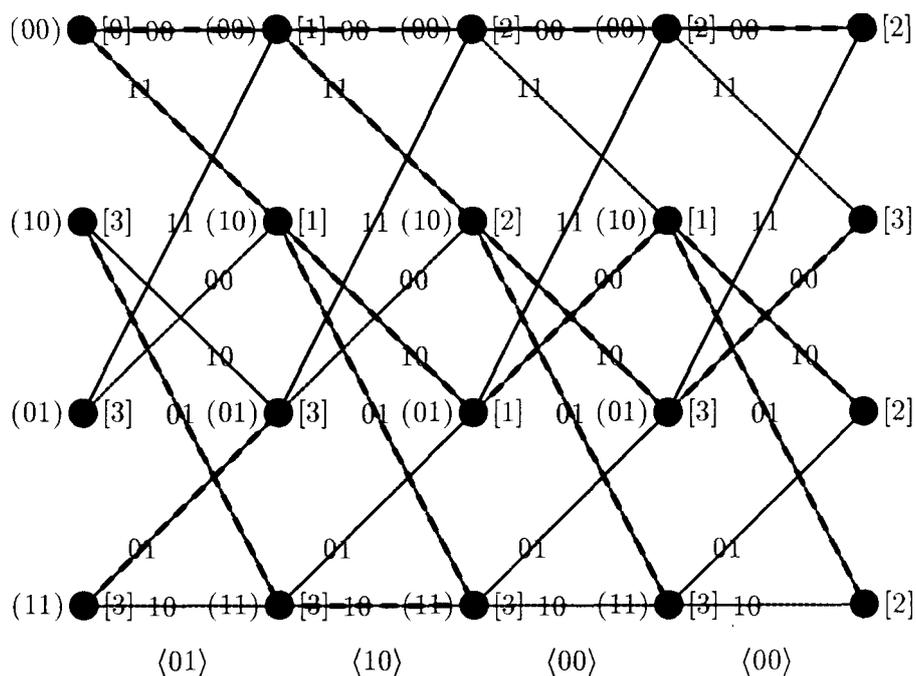


Fig. 1 Exemple de trellis avec un algorithme de Viterbi

$\text{\TeX}$ Graph utilise la capacité de  $\text{\TeX}$  d'insérer des graphiques soit sur une seule colonne d'un texte à plusieurs colonnes, ou sur une page. Pour faire cela,  $\text{\TeX}$ Graph a besoin de connaître la grosseur du graphique. Si le graphique est produit en utilisant les capacités de graphiques vectorielles de  $\text{\TeX}$ Graph, la grosseur du graphique est informatisée. Si le graphique provient d'un fichier extérieure, le format est dans l'entête. Un prérequis souvent nécessaire pour des insertions de graphiques dans les logiciels  $\text{\TeX}$ Printest que la grandeur soit explicitement spécifiée quand on fait l'insertion d'un graphique. Cependant, si le graphique n'est pas exactement de la dimension mentionnée, le logiciel  $\text{\TeX}$ Print ou l'imprimante doit compenser en ajustant la grosseur.

Ceci se fait facilement avec PostScript pourvu qu'aucun des graphiques ne soit en représentation « raster ». Ceci veut dire, par exemple, que le texte dans le graphique doit être d'une police de ca-

ractères « internal stroke ». Il n'est pas possible de mesurer les entrées avec le langage QUIC et avec la plupart des autres représentations. Pour ces raisons, le mécanisme d'insertion de  $\text{\TeX}$ Graph a choisi une forme graphique sans échelle. Le principal avantage pour l'auteur est qu'il n'a pas à déterminer la grosseur du graphique avant l'insertion. Dans la plupart des cas le résultat est très satisfaisant.

## 5. Un exemple spécifique de $\text{\TeX}$ Graph

Une intéressante façon d'utiliser  $\text{\TeX}$ Graph est démontrée à la figure 1. Cette figure semble être très facile et en fait, elle l'est. Elle aurait pu être produite avec MacDraw très rapidement, mais elle a été produite avec peine en utilisant  $\text{\TeX}$ Graph. C'est parce qu'il y a dans cette image plus que ce que l'oeil peut voir.

Le  $\TeX$ Graph et les autres commandes associées de  $\TeX$  produisent non seulement la figure, mais elles incluent l'algorithme de Viterbi lequel calcule le chemin le plus probable de traverser le trellis. Les instructions pour cette figure sont:

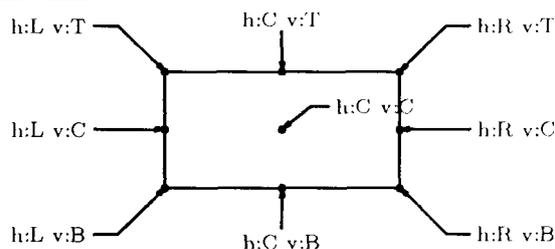
```
\input fourtrps
\figureform{trellis}{Exemple
de trellis avec un algorithme
de Viterbi}{\centergraph{
\xdef\clist{0 3 3 3 }
\trel {\dectrel} 01 10 00 00 }}

```

La raison pour `\clist` est le poids des initiales. `\trel` appelé `\dectrel` amène le calcul des chemins de décisions tandis que `\baretrellis` aurait produit une version sans décision, près à être complété par les étudiant.

## 6. Texte en $\TeX$ Graph

Tel qu'indiqué dans la section 4.1, l'insertion de texte dans les graphiques implique des considérations pragmatiques et de design. Comme les polices de caractères de  $\TeX$  sont très variés,  $\TeX$ Graph permet à  $\TeX$  de les choisir. De sorte que le texte d'un graphique n'est pas inclus dans la commande `\special`.  $\TeX$ Graph connaît l'endroit où sera situé le texte dans le graphique et s'y déplace en utilisant les commandes normales de `\hskip` et `\vskip`. Quoique l'utilisateur peut spécifier une grandeur de boîte spécifique, quand  $\TeX$  essaie de la placer sur la page, elle n'apparaît que comme un point. La police de caractères peut-être spécifiée soit localement ou globalement. Pour rendre le placement de texte précis et facile, une position de référence dans la boîte peut-être spécifiée. Les neuf choix sont indiqués dans la figure suivante:



Ceci veut dire qu'une étiquette peut-être positionnée précisément par rapport au haut, au bas, à droite et à gauche de la page. Ceci rend l'annotation de graphiques très facile.

Il n'y aurait aucun problème si on désirait du texte seulement à l'horizontale. Cependant nous avons besoin de textes à la verticale et possiblement en rotation. Ceci peut-être accompli avec  $\TeX$ Graph en insérant la commande `\special` après `\hskip` et `\vskip` pour accéder à la position du texte qui peut faire une rotation à la boîte contenant le texte et une commande `\special` après la boîte pour défaire le résultat. Ceci est fait avec QUIC en passant du mode portrait au mode paysage et de nouveau au mode portrait tandis que dans PostScript, ceci s'accomplit en modifiant la matrice de transposition avant la boîte et en retournant à l'original après. Pour que ceci fonctionne, le logiciel  $\TeX$ Print doit:

- traiter le texte qui se situe entre les commandes `\special` et s'assurer qu'il apparaît dans cet ordre dans le fichier qui est envoyé à l'imprimante,
- ne pas détruire aucun texte qu'il croit ne pas être sur la page, parce que la commande `\special` peut avoir changée le contenu de la page,
- pour PostScript, définir le `currentpoint` pour qu'il soit accessible dans la commande `\special`,
- ne pas accomplir aucune action, telle un `save` ou un `restore` dans

PostScript avec une commande `\special` ce qui pourrait annuler l'action produite par la commande `\special`.

Ces prérequis découlent de la méthodologie utilisée pour le placement du texte dans `TeXGraph`, mais découlent aussi d'autres considérations. Supposons que la commande `\special` était utilisée pour changer la couleur d'un texte, d'un graphique ou d'une table. Ceci se fait régulièrement pour accentuer certains documents. Il est évident que la destruction de l'ordre de traitement supposé par le fichier `*.dvi` détruirait totalement la manière de procéder de ce changement de couleur. Quoique nous n'utilisons pas de couleur à l'INRS-Télécommunications, nous faisons la rotation de tables. Comme les tables utilisent du texte et des colonnes, une table fera rotation correctement seulement si l'ordre de procéder est donné dans le fichier `*.dvi`.

## 7. Les effets du langage d'imprimante sur l'implantation

Du point de vue de l'utilisateur, le changement de `QUIC` à `PostScript` augmente le pouvoir des graphiques vectoriels de `TeXGraph` et enlève une irritation mineure. La version `PostScript` inclut « splines » sans et avec une pointe de flèche. Ceci permet d'embellir les lignes qui produisent des dessins variés. Cependant, une des pénalités de cette augmentation de puissance est un peu d'incompatibilité. Un exemple de cette incompatibilité est le concept de mettre une région « fill ». `QUIC` utilise un « seeded fill » avec la possibilité de produire un grand nombre de dessins différents. Ce type de remplissage est sensible à la couleur, blanche ou noire, à l'endroit où commence le remplissage et peint toute la

page jusqu'à la frontière de la couleur opposée. Ceci produit des résultats plutôt bizarres quand une ligne noire au milieu d'une page blanche est utilisée comme source. `PostScript` utilise un remplissage polygonal qui ne permet que quelques ombrages de gris. Ce qui est intéressant, c'est que `METAFONT` utilise un remplissage appelé « edge fill ». Ceci est similaire à un remplissage polygonal excepté quand le « poligon » s'entrecroise. Ce qui veut dire qu'il faut utiliser le remplissage avec prudence. `TeXGraph` reconnaît cette inconsistance en définissant un remplissage dans `QUIC` par `\fill` et dans la version `PostScript` par `\pfill`. En général, les incompatibilités entre les versions sont signalées sous forme de différentes commandes dans les différentes versions. Nous espérons qu'il n'y aura aucune différence produites par les programmes `TeXPrint`.

Quand la version `PostScript` de graphiques vectoriels de `TeXGraph` a été produite, nous espérons pouvoir produire tous les graphiques en utilisant `PostScript` seulement. Ceci ne fut pas le cas. Quoique les capacités mathématiques de `PostScript` rendaient plus facile le calcul de certain angle, une portion substantielle de la structure interne des deux systèmes demeurerait plutôt similaire. Le problème se trouvait dans l'implantation de la partie la plus puissante et la plus utile des graphiques vectoriels `TeXGraph`, soit le « segment graphique ». Ce mécanisme de regroupement, qui est à la base de la production d'objets graphiques, regroupe les commandes graphiques et redonne l'ancienne valeur quand on sort de la double commande `\beginsegment ... \endsegment`. Ceci se fait très facilement avec le mécanisme de regroupement de `TeX` mais semble être plus difficile avec le mécanisme de dictionnaire de `PostScript`. Le résultat final fait que la version `PostScript` de `TeXGraph` est très similaire du point de vue interne à la version `QUIC`.

## 8. Conclusions

Bien que la possibilité d'inclure des graphiques dans le système INRST<sub>EX</sub> était très importante pour son acceptation, le fait demeure que la création de graphiques n'est pas amusante. L'avantage principale d'utiliser MacDraw est le sentiment d'accomplissement immédiat qui est donné à l'utilisateur. Ce sentiment est tellement vivifiant qu'il engendre un effort et une concentration extraordinaire. Il importe peu qu'il soit très difficile d'obtenir des résultats précis et agréables ... dans certains cas. Le sentiment général est que c'est un résultat satisfaisant et bien mieux que ce qui pourrait être accompli dans les mêmes limites de temps par d'autres méthodes. Il produit aussi un sentiment de créativité accomplie qu'il est difficile d'argumenter. L'exemple de la section 5 est impossible à accomplir avec MacDraw mais requiert probablement trop de connaissances en programmation pour être un « paradigme » pour la production de graphiques.

Notre système graphique a indiqué quelques modestes prérequis des logiciels

TeXPrint et démontre un besoin de standardisation au niveau de l'imprimante. Peut-être que PostScript pourrait être à la base d'un tel standard. Il serait préférable qu'il soit un peu plus puissant et ouvert.

## Références

1. L. Lamport, **L<sub>A</sub>TeX**, Addison Wesley, New York, 1985.
2. M.J. Wichura, **The PICT<sub>EX</sub> Manual**, *TEXniques Series, Vol. 5*, T<sub>EX</sub> Users Group, Providence, 1988
3. QMS Inc., **QUIC Programming Language, Version 3**, No. 1720460.0100, QMS Inc, Mobile, Alabama, Oct. 1984.
4. Apple Comp. Inc. **MacDraw**, Apple Product No. M1509, Apple Computer Inc., Cupertino, Calif., 1984.
5. Adobe Systems Inc. **PostScript Language Reference Manual**, Addison Wesley, New York, 1985.