

J.-P. BENZÉCRI

F. BENZÉCRI

## **État des modifications apportées aux versions du logiciel MacSAIF**

*Les cahiers de l'analyse des données*, tome 18, n° 3 (1993),  
p. 321-344

[http://www.numdam.org/item?id=CAD\\_1993\\_\\_18\\_3\\_321\\_0](http://www.numdam.org/item?id=CAD_1993__18_3_321_0)

© Les cahiers de l'analyse des données, Dunod, 1993, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). 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.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

## ÉTAT DES MODIFICATIONS APPORTÉES AUX VERSIONS DU LOGICIEL MacSAIF

### [ÉTAT MacSAIF]

*J.-P. & F. BENZÉCRI*

Le cahier n°1 du Volume XIV de *CAD* (1989) renferme les premières notices des programmes du logiciel MacSAIF. Divers compléments et des mises à jour ont paru depuis lors. En particulier, la notice [MacSAIF] (in *CAD*, Vol.XV, n°3, 1990) fait l'inventaire des programmes en donnant des références bibliographiques adéquates.

Depuis la parution de cette dernière notice, le format général des fichiers créés et utilisés par le logiciel n'a pas changé; mais de nombreuses modifications ont été introduites, selon les besoins; et de nouveaux programmes ont même été créés, quand il a paru difficile d'intégrer les nouvelles fonctions dans des programmes préexistants. L'objet de la présente note est de rassembler toutes ces nouveautés; dont la plupart ont déjà été mises en œuvre dans des articles parus dans *CAD*.

Bien que cette distinction ne soit pas absolument tranchée, nous dirons qu'un logiciel d'analyse de données a trois sortes de fonctions: mise en œuvre des algorithmes de base d'analyse factorielle et de classification; création de tableaux, destinés à être soumis à l'analyse et à la CAH ou rendant compte des résultats de celles-ci; création de graphiques préliminaires et de graphiques de résultats.

Les algorithmes n'ont pas été notablement améliorés depuis que, en 1982, la recherche en chaîne des voisins réciproques a grandement accéléré la CAH. Sans doute les progrès ultérieurs viendront-ils du calcul parallèle avec plusieurs processeurs, calcul qui permettrait de traiter sur un ordinateur individuel les plus grands ensembles de données.

Mais le traitement de très grands tableaux (e.g. 100k lignes et 1000 colonnes) n'a d'intérêt que si les données sont convenablement préparées et les résultats présentés sous une forme accessible à l'intuition; ce qui nous ramène aux fonctions de création de graphiques et de tableaux, objet principal de la présente mise à jour; de laquelle sont toutefois exclus les programmes de préparation de données linguistiques, objet de plusieurs articles où la

présentation des méthodes va avec le traitement de grands ensembles de données (cf. e.g. [TEXTES GRECS 2], in *CAD*, Vol.XVIII, n°2; 1993).

## 1 Création de graphiques

### 1.1 Le programme 'Grftab' de tracé des courbes des lignes

On rencontre communément des tableaux de données où l'ensemble J des colonnes n'est autre qu'un ensemble T de temps: dates ou instants régulièrement espacés; tandis que chaque ligne *i* enregistre la variation temporelle d'un phénomène, à l'échelle fixée par T. Il peut s'agir des cours trimestriels d'une action ([SICAV], in *CAD*, XVI, n°1, 1991); du volume annuel des exportations d'un produit par un pays (Th. ETTÉ: [CACAO], in *CAD*, XVI, n°1; [CAFÉ], in *CAD*, XVI, n°3); des effectifs totaux, ou des dépenses d'un poste, relevés annuellement pour un secteur de l'économie ([CMPT. FORM.], in *CAD*, XVII, n°1, 1992); de variables hémodynamiques (pression systolique, fréquence cardiaque), enregistrées trois fois par heures sur un sujet libre des ses mouvements (cf. [PRESS. ART. NYCTH.], in *CAD*, Vol. XV, n°4, 1990; et Prat MÉD, MC2 n°9); des cordonnées, mesurées 100 fois par secondes, pour un point de repère, sur un sportif en mouvement rapide...

Dans ces divers cas, il est utile d'illustrer l'analyse multidimensionnelle par des courbes figurant les données d'une ligne; soit à l'état brut, soit peu élaborées. C'est à quoi sert le programme 'Grftab'.

Pour décrire l'usage de 'Grftab', nous suivrons les instructions et demandes présentées à l'écran.

S'affiche d'abord la fonction de 'Grftab'

'ce programme présente les courbes des lignes';

donc, si comme c'est parfois le cas, T était l'ensemble I des lignes du tableau de base, il faudrait transposer celui-ci (par 'zrang') avant d'entrer dans 'Grftab'.

Le tableau des données de base est alors choisi par un dialogue (comme au début de 'qori', de 'cums'...) et mis en mémoire centrale. Et vient la question:

'faut-il restreindre J à un sous-intervalle O ou N';

en effet, d'une part, l'utilisateur peut souhaiter montrer nettement une sous-période, éliminer certains temps pour lesquels les données sont incomplètes...; et, d'autre part, pour la netteté du tracé, 'Grftab', impose, dans sa version actuelle, que le cardinal de T soit au plus égal à 150.

Il faut comprendre que dans la question:

'faut-il faire des graphiques grands(G) ou petits(P)

on tient compte des exigences arithmétiques imposées par la valeur de CardT: 'Petit' veut dire, aussi petit que le permet CardT; 'Grand' signifie "étendu à

l'écran". Il faut comprendre que l'utilisateur peut désirer des graphiques à petite échelle, pour en disposer plusieurs dans une seule page, à des fins de comparaison (cf. [CACAO], [SICAV], [COMPT. FORM.]...)

La question suivante est d'une importance déterminante:

'faut-il représenter les lignes brutes(B) ou les profils/marge(P) ';

en effet, les lignes brutes sont directement comprises par un lecteur non formé en analyse des données; mais c'est d'après les courbes "profil/marge" que l'on saisit quelles sont les véritables oppositions dont l'analyse donnera une vue synthétique: e.g., taux de croissance inégaux, pour des phénomènes tous croissants. Par profil sur marge, on entend ici le quotient:  $k(i,t)/k(t)$ ; plus exactement:  $k(i,t).k/(k(t).k(i)) = f_j/f_i$ .

Mais se pose aussi la question du cadrage des graphiques:

'faut-il tracer tous les graphiques à la même échelle O ou N ';

Dans le tableau des données brutes, les diverses lignes ont, généralement, des ordres de grandeur différents; et, par conséquent, il s'impose d'étaler chaque courbe avec l'échelle d'ordonnée la plus large compatible avec le cadre choisi pour le tracé: on répondra donc 'Non' à la question précédente. Au contraire, pour comparer des profils, dont les composantes sont des nombres non dimensionnés, avec l'unité '1' qui correspond au profil de marge, il est généralement préférable de répondre 'Oui'.

S'il s'agit de représenter les lignes brutes, et que le tableau ne comprend que des nombres positifs, la ligne inférieure du cadre correspond à la valeur zéro; et la ligne supérieure au maximum; maximum général, si on a répondu 'Oui' à la question précédente; maximum pour la ligne, si on a répondu 'Non'. Dans le cas d'un tableau général comprenant des nombres de signe quelconque, le cadre du graphique est traversé par une ligne horizontale qui donne le niveau du zéro. Dans tous les cas, le graphique s'affiche à l'écran avec les valeurs précises correspondant aux niveaux supérieur et inférieur du cadre.

Si l'on a demandé de représenter des profils, le programme vérifie d'abord que le tableau brut ne comporte ni valeur strictement négative, ni ligne de total nul; et, dans le cas où est décelée une telle anomalie, annonce, avec un commentaire, que seront représentées des lignes brutes:

'ERREUR, la ligne ..... a un total nul' ; OU:

'ERREUR, la ligne ..... comporte un nombre négatif'

'on ne tracera donc pas de profil ' .

S'il s'agit, finalement, de représenter des profils, la ligne inférieure du cadre correspond nécessairement à la valeur '0'; et le cadre du graphique est traversé par une ligne horizontale qui, à présent, donne le niveau un. La ligne

supérieure du cadre correspond soit à la valeur maxima, sur  $J=T$ , des composantes du profil particulier représenté, soit à la valeur maxima, sur  $I \times J$ , des composantes de tous les profils afférents au tableau considéré. Comme pour une ligne brute, le graphique porte les valeurs précises correspondant aux niveaux supérieur et inférieur du cadre (ce dernier étant nul).

Il va sans dire que le graphique peut être saisi par copie d'écran (pour être élaboré par un logiciel du type de MacPaint); et que si, après qu'un graphique a été effacé, l'utilisateur demande de poursuivre, le programme, sous l'en-tête

'ci-dessous sont rappelés les num et sigles des lignes'

présente l'ensemble I.

Mais il importe de signaler que les réponses aux quatre questions rappelées ci-après, valent pour tous les graphiques tracés après qu'on a chargé un tableau de données en mémoire centrale:

'faut-il restreindre J à un sous-intervalle O ou N ';

'faut-il faire des graphiques grands(G) ou petits(P) ';

'faut-il représenter les lignes brutes(B) ou les profils/marge(P) ';

'faut-il tracer tous les graphiques à la même échelle O ou N ';

donc, si on désire, pour un même tableau, changer le cadrage ou passer des données brutes aux profils, ou réciproquement, il convient de répondre 'Non' à la question:

'faut-il afficher une autre courbe O ou N ';

et ensuite 'Oui' à la suivante:

'faut-il poursuivre avec un autre tableau O ou N ';

comme s'il s'agissait véritablement de données nouvelles.

## 1.2 Modifications au programme 'planF' de tracé des graphiques d'analyse de correspondance

Le premier article consacré au programme 'qori' d'analyse des correspondances (cf. [NOT. CORR. CAH], in *CAD*, Vol.XIV, n°1, 1990), présente également le programme 'planF'. Il suffira de rappeler que l'exécution de ce programme débute en demandant le nom du tableau de base, disons 'Disq:Tap'; puis les sigles des ensembles à représenter: 'i', 'j', 'is', 'js'; mais aussi 'iq', 'jq', si l'on a effectué des classifications en faisant jouer la procédure FACOR; et, plus généralement, tout ensemble 'ens' pour lequel existe un fichier de facteurs: 'Disq:TapensFac.w' (créé par 'qlrsup', par 'FACOR'...).

Avec chaque ensemble, est offert le choix de la représentation de ses éléments:

'faut-il figurer les élém de ens... par leurs sigles(S) '  
 'par la première lettre de leur sigle(i), par la dernière(d) '  
 'ou par un caractère unique(U) ';

avec, si la réponse est 'U', la demande:

'le caractère choisi pour ens..., est ';

Il va sans dire que l'usage des options 'i' ou 'd' est conditionné par la forme particulière des sigles choisis: e.g. la première lettre donnant l'âge du sujet, la dernière le traitement qu'il a reçu; etc.. L'option 'U', quant à elle, sert à produire une image de densité du nuage: chaque élément étant figuré par le caractère unique '.' ou 'o'; à moins que des sous-ensembles séparés (lus et affichés l'un après l'autre) ne correspondent à des diagnostics différents, rappelés par le caractère choisi (les fichiers de facteurs pour ces sous-ensembles ayant été créés, à partir du fichier général, par le programme 'soustab'); etc..

Éventuellement, l'utilisateur peut demander deux fois le même ensemble avec des modes d'affichage différents; ce qui permet de doser la densité des nuages.

Les vérifications une fois faites, le programme calcule les bornes des facteurs sur chacun des ensembles (ces bornes étant les seules données conservées en mémoire centrale); puis affiche, à la demande, des séries de graphiques définies en spécifiant les ensembles à conserver; chaque graphique pouvant croiser deux axes quelconques. Chaque affichage d'un ensemble 'ens' se fait par une lecture de 'Disq:TapensFac.w', en sorte que l'utilisateur voit les points pleuvoir sur l'écran; d'autant plus rapidement que le temps de transfert à partir de 'Disq' est plus court.

Les seules modifications majeures apportées à 'planF' font l'objet des deux questions suivantes:

'faut-il adopter la même échelle sur les deux axes O ou N ';  
 'faut-il afficher le plan sur deux écrans superposés O ou N ';

Il n'y a pas lieu d'insister sur l'intérêt de la première question.

Quant à la deuxième, nous dirons que, si la réponse est 'Oui', l'étalement offert à l'axe vertical se trouve à peu près doublé. Il revient à l'utilisateur (comme pour les cartes produites par 'carthag') d'ajuster, par 'MacPaint', les deux moitiés du graphique, saisies séparément. Les repères et références portées, désormais, en marge du nuage facilitent ce montage.

Mis à part le cas d'une image de densité du nuage (cf. *supra*), il n'est pas opportun de porter sur un graphique des centaines de points: un ensemble de cardinal élevé doit être présenté par le nuage des centres des classes issues d'une partition. Mais, même pour un nuage d'effectif modéré, l'étalement sur deux

écrans offre plus de clarté; et permet, dans certains cas, de considérer directement un nuage sans consulter simultanément le tableau du contenu des classes d'une partition.

Nous signalerons que, si l'on a répondu affirmativement aux deux questions, mais que, du fait de la faible amplitude de variation de la coordonnée verticale, la moitié inférieure du graphique devait se trouver vide, le programme supprime celle-ci et affiche le plan comme si l'on avait répondu 'N' à la deuxième question.

### 1.3 Tracé d'arbre dans le programme de classification

Dans la version en usage présentement, deux programmes 'CLS' et 'VAC' réalisent les opérations de classification, tracé d'arbres et aides à l'interprétation. Le programme 'VAC' fournit uniquement des aides à l'interprétation, en termes de variables du tableau de base, pour les CAH réalisées par 'CLS'. Ce dernier programme construit des CAH pour les ensembles de lignes et colonnes d'un tableau principal, déjà soumis à l'analyse des correspondances; ainsi qu'éventuellement pour des ensembles supplémentaires, adjoints au tableau principal par 'qorlsup'. De plus, 'CLS' crée des tableaux et fichiers d'interprétation des CAH en terme de facteur: et, notamment, pour chaque ensemble de centres de classes ou de nœuds, un fichier de coordonnées sur les axes factoriels: e.g., 'Disq:tapiqFac.w', pour l'ensemble des centres de classes de lignes du tableau de base 'Disq:tap'. La seule modification à expliquer se rapporte au tracé d'arbre.

De façon précise, après avoir fixé la partition à extraire de la CAH complète (partition définie par les noeuds les plus hauts ou par des noeuds spécifiés), le programme s'enquiert de la présentation des résultats:

'la largeur (en caractères) choisie pour le listage est ' (largeur de 76 à 136)

'la largeur (en caractères) choisie pour tracer l'arbre est '

et, si la largeur demandée (qui ne peut dépasser 255) est supérieure à celle déjà fixée pour le listage, s'affiche le commentaire

'ATTENTION: l'arbre sera tronqué à droite'

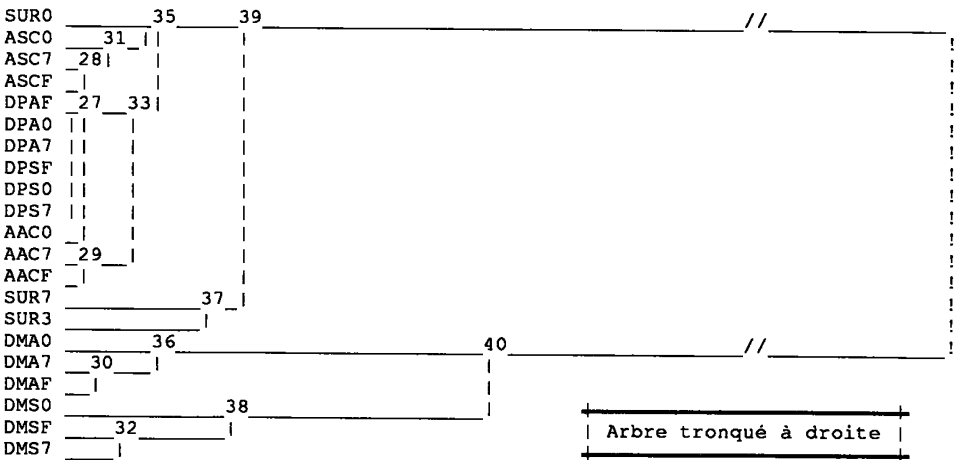
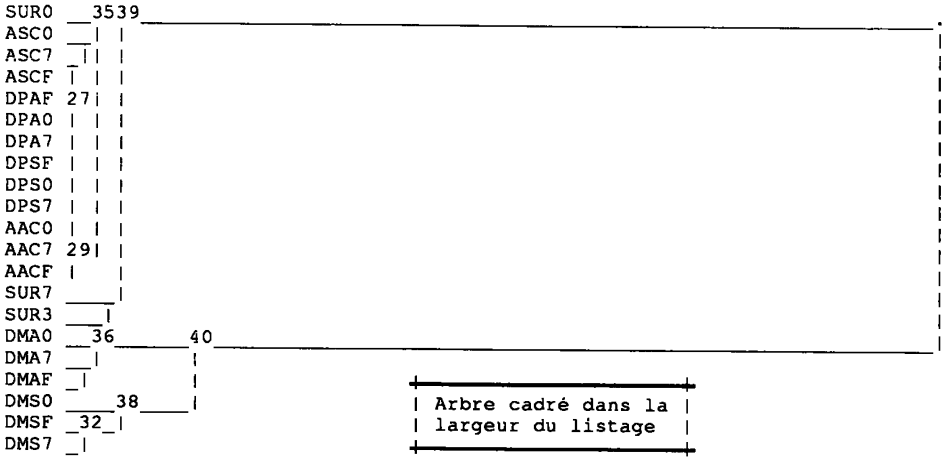
puis vient la question

'faut-il passer une ligne entre 2 classes oui(O) ou non(N) '

suivie d'une demande de confirmation:

'le choix de l'arbre de la partition est-il confirmé O ou N ';

Un dialogue tout semblable définit le format de tracé de l'arbre de la CAH générale (si ce tracé est demandé).



Dans tout cela, relativement à ce qui est dans la notice [NOT. CORR. CAH], seule reste à expliquer la mention: 'l'arbre sera tronqué à droite'.

Prenons l'exemple d'une classification sur 21 éléments, présentée d'abord avec l'arbre cadré sur la largeur du listage. À cette échelle, il apparaît que le niveau du nœud supérieur dépasse de beaucoup tous les autres; en sorte que les subdivisions des classes se lisent mal.

En multipliant par 3 l'échelle horizontale, on distingue mieux les subdivisions; mais, compte tenu de la largeur fixée pour le listage, ne sont dessinées que deux branches (écourtées), sans le pont vertical qui les réunit.



Celui-ci a été ajouté, en utilisant le caractère '!', pour signaler cette addition; et des traits de coupure obliques rappellent que les branches horizontales sont écourtées.

#### 1.4 Croisement de deux de variables dans le programme zrang

NB on affichera successivement un plan et un tableau;  
 pour appeler ou quitter le plan, entrer un caractère;  
 pour marquer les points sur le plan par leur SIGLE entrer '\*\*';  
 pour les marquer par le 1-er caractère du sigle entrer 'i';  
 pour les marquer par un même caractère AUTRE que '\*\*' ou 'i',  
 entrer le caractère choisi

Dans le programme 'zrang', le dialogue qui annonce le croisement de deux variables propose désormais explicitement trois manières d'afficher l'ensemble des individus: soit par leur sigle, soit par l'initiale de leur sigle, soit par un caractère quelconque.

De plus, comme dans 'planF', a été introduite la question:

'faut-il adopter la même échelle sur les deux axes O ou N ';

mais l'affichage sur deux écrans superposés n'est pas prévu dans le dialogue de 'zrang'.

## 2 Création de tableaux

### 2.1 Le programme 'cums', de cumul et combinaison de lignes ou de colonnes

Le programme 'cums' est présenté dans l'article intitulé: "Extension d'un tableau par adjonction de lignes et colonnes créées par cumul", (cf.[CUM. LI. COL.], in *CAD*, Vol. XV n°1, 1990).

Le programme 'cums' procède en chargeant en mémoire centrale un tableau de base - qu'il sera commode de nommer 'Disq:Tap' - puis demande le sigle du tableau à créer - sigle que nous supposons réduit à un seul caractère: '\*'. S'il existe un fichier de commande, 'Disq:Tap $\Sigma^*\Sigma^x$ ', le programme ouvre ce fichier pour y lire les ordres de cumul; sinon, un dialogue s'engage pour obtenir de l'utilisateur les ordres de cumul.

Initialement, le programme 'cums' a été conçu pour adjoindre à un tableau (introduit en mémoire centrale) des lignes ou des colonnes calculées comme sommes de lignes ou colonnes figurant dans le tableau initial, ou déjà créées par cumul. Mais divers choix ont été, par la suite, offerts à l'utilisateur.

Nous présentons ci-après les fonctions du programme; avec la forme précise des commandes.

### 2.1.1 Opérations générales sur les lignes et colonnes

Afin de calculer toute combinaison linéaire à coefficients entiers (et non seulement un simple cumul, où tout coefficient vaut 1), on a utilisé le repère numérique qui marque, sur le listage de commande, le début d'un ordre de cumul, pour coder le coefficient dont on demande d'affecter ce cumul (on ne peut donc introduire un tel coefficient par dialogue à l'écran).

Quant au signe moins, il est introduit, *ipso facto*, si le sigle de l'élément à créer débute par le signe '-'.  
.

On notera que, sans entreprendre de créer des combinaisons linéaires de forme générale, il peut être utile d'affecter une ligne ou colonne d'un coefficient si l'on veut passer au format entier, sans perdre d'information.

#### 2.1.1.1 Coefficient entier positif dans le fichier de commande

On sait qu'un ordre de cumul doit commencer par le nombre 20000, ou par tout autre nombre,  $20000+x$ , compris entre 20000 et 39999; tandis que le listage ' $\Sigma*\Sigma$ ' créé lors du cumul (afin de conserver l'inventaire détaillé des éléments créés) porte le nombre 20000 augmenté du rang de l'élément créé: e.g. 21456 si il s'agit d'une ligne qui prend le rang 1456 (compte tenu des lignes du tableau de base).

Pour affecter la somme créée d'un coefficient entier, il suffit de remplacer le nombre  $20000+x$ , par un nombre qui spécifie un multiplicateur entier positif, *mult*, par l'addition de ce nombre, multiplié par 100000. Par exemple, si au lieu de  $20000+x$ , on écrit  $520000+x$ , la combinaison créée sera multipliée par 5; il faut seulement se souvenir que, le nombre étant lu au format 'entier long', ne peut dépasser 2147483647: ce qui permet d'introduire tout coefficient entier jusqu'à 21474; ce qui paraît suffire dans la pratique.

#### 2.1.1.2 Exemple de fichier de commande avec coefficients

Le texte ci-après (cf. [ÉPUISEMENT], in *CAD*, Vol.XVIII, n°2, §5) commande la création de 7 nouvelles colonnes, obtenues en multipliant par 1000 les colonnes 1 à 7 du tableau de départ.

```
définition de colonnes multipliées par 1000
100020000   axZ1 colonne 1 ;
100020000   axZ2 colonne 2 ;
100020000   axU1 colonne 3 ;
100020000   axU2 colonne 4 ;
100020000   axD1 colonne 5 ;
100020000   axD2 colonne 6 ;
100020000   Scal colonne 7 ;
```

Le fichier créé lors du cumul est, comme de règle, un peu plus explicite: il donne les numéros et les sigles des éléments pris dans chaque cumul.

```

Un1:Ph:phm*QΣ*izudFacΩΣ*Σ
100020008 axZ1 colonne combinaison des colonnes suivantes
  1 Zxe1 ;
100020009 axZ2 colonne combinaison des colonnes suivantes
  2 Zxe2 ;
100020010 axU1 colonne combinaison des colonnes suivantes
  3 Uxe1 ;
100020011 axU2 colonne combinaison des colonnes suivantes
  4 Uxe2 ;
100020012 axD1 colonne combinaison des colonnes suivantes
  5 Dxe1 ;
100020013 axD2 colonne combinaison des colonnes suivantes
  6 Dxe2 ;
100020014 Scal colonne combinaison des colonnes suivantes
  7 Scal ;

```

### 2.1.1.3 Le signe négatif

Pour introduire un coefficient négatif, on agit au niveau du sigle: tout sigle commençant par le caractère '-' commande l'introduction du signe - (indépendamment de l'introduction éventuelle d'un coefficient entier positif, en tête de l'ordre de commande). Le coefficient négatif nous a servi, jusqu'ici, à créer, en deux étapes, une colonne (ou une ligne) différence.

Par exemple, dans [TEXTES LATINS] (cf. Vol. XVI, n°4, 1991), on dénombre les formes se terminant par 'ibus' (désinence du datif-ablatif pluriel des noms et adjectifs de la 3-ème déclinaison; ainsi que des pronoms); en retranchant les formes {quibus, omnibus}, qui sont dénombrées séparément pour elles-mêmes. On crée d'abord la colonne '-bus', opposée du cumul des colonnes {omnibus, quibus}; puis la colonne 'ibus', cumul de {-bus, ibus}; i.e. (ibus - (quibus + omnibus)).

### 2.1.1.4 Calcul de pourcentages

De manière semblable, si le sigle se termine par le caractère '%', on crée une ligne ou colonne dont le total est réduit à 100 par multiplication par une constante appropriée. Plus précisément cette normalisation à 100 est effectuée pour l'ensemble valeurs comprises dans l'étendue du tableau initial: e.g., si d'un tableau  $I \times J$  on passe, par créations successives, à un tableau  $(I+I') \times (J+J')$ , une ligne  $i'$  de pourcentage est une ligne pour laquelle vaut 100 le total  $\sum\{k(i',j)\}$  calculé pour  $j$  dans  $J$  (à l'exclusion de  $J'$ ).

Il importe de noter que, de par la linéarité, l'effet des ordres de cumul et de pourcentages considérés jusqu'ici est commutatif; en ce sens qu'on obtient le même tableau final si l'on crée toutes les lignes de cumul avant toutes les colonnes, ou le contraire; et de même les lignes (ou les colonnes) peuvent être créées dans un ordre arbitraire, pourvu qu'avant chaque élément nouveau soient déjà créés tous ceux qui figurent dans sa combinaison.

Enfin, il a paru commode d'introduire, également par la forme des sigles, des ordres pour des multiplications et des divisions. Mais, avec de tels ordres,

créations de lignes et créations de colonnes peuvent ne pas commuter; le bloc I'  $\times$  J' dépendant de l'ordre des créations.

### 2.1.1.5 Produits et inverses

Pour introduire un coefficient entier, il faut commander la création des cumuls par un fichier; tandis que l'inversion et le produit, comme le signe - et le calcul de pourcentages, étant liés au sigle choisi, peuvent être obtenus dans le dialogue à l'écran. Et c'est pourquoi, quand on ouvre le programme cums, s'affiche le texte suivant, qui explique clairement le format de ces ordres:

```
ce programme crée des lignes et colonnes nouvelles par cumul;
ATTENTION: si le sigle de l'élément à créer commence par le
signe -, les nombres calculés seront affectés de ce signe;
           si le sigle de l'élément à créer commence par le
signe /, les nombres calculés seront inversés;
           si le sigle de l'élément à créer commence par le
signe *, on calcule un produit (non une somme);
```

Il importe de noter que, si le listage de commande demande que l'élément créé soit multiplié par un coefficient entier, cette multiplication est effectuée après une inversion éventuelle commandée par le caractère '/' au début du sigle.

### 2.1.2 Tableau complété et tableaux partiels

Le programme 'cums' sert communément à créer des lignes et colonnes mises en supplément; ou, au contraire, à créer, par cumul, un tableau qui sera pris pour principal. Dans ces conditions, il peut être utile de créer, plutôt qu'un tableau complété, un tableau restreint aux seules lignes ou colonnes qu'on a créées: i.e.  $I \times J$ , nouvelles lignes  $\times$  colonnes initiales; ou  $I \times J_n$ , lignes initiales  $\times$  nouvelles colonnes; ou le transposé de celui-ci,  $J_n \times I$ : ce tableau a en effet l'avantage de pouvoir être adjoint en supplément à l'analyse de  $I \times J$ , par le programme 'qorlsup'; lequel lit en séquence les colonnes supplémentaires, présentées comme des lignes.

C'est pourquoi, quand sont achevés les calculs de cumul, le programme pose à l'utilisateur les questions suivantes:

```
faut-il créer un tableau complet, formé du tableau initial
complété par les lig et col créées par cumul O ou N

faut-il créer un tableau partiel, formé seulement
des colonnes qui ont été créées par cumul, O ou N

faut-il créer un tableau transposé, ayant pour lignes
les colonnes qui ont été créées par cumul, O ou N

faut-il créer un tableau partiel, formé seulement
des lignes qui ont été créées par cumul, O ou N
```

Ces questions ne sont toutefois posées que si des colonnes ou des lignes ont été effectivement créées.

Si le tableau de base est 'Disq:Tap', et que le sigle spécifié pour le tableau à créer est '\*', les tableaux  $(I+I_n) \times (J+J_n)$ ,  $I \times J_n$ ,  $J_n \times I$ ,  $I_n \times J$  ont respectivement pour noms:

Disq:Tap $\Sigma^*$  ; Disq:Tap $\Sigma^*c$  ; Disq:Tap $\Sigma^*b$  ; Disq:Tap $\Sigma^*a$  ;

on notera que les terminaisons ' $\Sigma^*b$ ' et ' $\Sigma^*a$ ' sont conformes aux conventions fixées pour qu'un tableau soit accepté par qorlsup, pour être adjoint à l'analyse de Disq:Tap, respectivement comme ensemble de colonnes ou de lignes supplémentaires.

### 2.1.3 Échantillon d'épreuve et groupes aléatoires

On a prévu le choix d'éliminer systématiquement, de tous les cumuls demandés, une certaine fraction des lignes ou des colonnes.

Dans [MacSAIF] (in CAD, Vol.XV, n°3, 1990; cf. p.364), un alinéa, consacré à [CUM. LI. COL.] indique qu'en début d'exécution, le programme pose la question suivante:

'faut-il éliminer des cumuls une fraction des élmt O ou N ';

et, si la réponse est O (oui), précise la question

'éliminer un demi(2), un tiers(3) ou un quart(4) '.

Si, par exemple, la réponse est 4, le programme 'cums' ne réalise pas exactement les cumuls qui lui sont demandés (par fichier ou en dialogue), mais élimine les éléments dont le rang est divisible par 4. L'intérêt d'une telle élimination pour l'analyse discriminante est expliqué, notamment, dans [TEXTES GRECS] (in CAD, Vol.XVI, n°2, 1991; cf. pp.69-74).

D'autre part, lorsqu'une analyse factorielle montre que se détache le centre d'un groupe d'éléments - par exemple: le groupe des lignes afférentes à des sujets ayant subi un traitement déterminé - il vaut la peine d'apprécier l'excentricité de ce centre en comparant celui-ci au nuage des centres de groupes créés par tirage aléatoire. Le programme 'cumhas' a pour fonction de définir de tels groupes aléatoires par un listage d'ordres de cumul (suivant le format accepté par 'cums'). L'essentiel du dialogue de 'cumhas' tient en deux questions:

'le cardinal de l'ensemble dont on extrait des parties est ';

'le cardinal des parties à créer est '.

D'après les réponses, Card et Part, données à ces questions, 'cumhas' crée, en bref, des listes de Part nombres, tirés aléatoirement entre 1 et Card. Le nombre des listes à tirer est spécifié en réponse à la question complémentaire:

'le nombre de sous-ensembles à définir est '.

Le fichier d'ordre de cumul est créé par 'cumhas', sous le nom 'Disq:Σhasx', comme s'il s'agissait de cumuler des lignes. L'utilisateur remplace, s'il le désire, 'ligne' par 'colonne'; et donne au fichier le nom approprié à l'usage qu'il en veut faire: e.g. 'Disq:TapΣzΣx', si le tableau de base est 'Disq:Tap'; et le sigle du tableau à créer, 'z'. Si, cf. §2.1.2, on crée seulement un tableau de lignes de cumul, celui-ci aura pour nom: 'Disq:TapΣza'; et pourra être adjoint, immédiatement, par 'qorlsup', en supplément à l'analyse de 'Disq:Tap'.

## 2.2 Le programme 'wITxJ' de présentation d'un tableau ternaire

De nombreuses études économiques, notamment les études diachroniques du marché mondial d'un produit (cf. [CACAO], [CAFÉ]), sont fondées sur un tableau ternaire de données; i.e. un tableau mettant en correspondance trois ensembles dont l'un est ordinairement le temps (e.g.  $k(i, j, t)$  = exportations du pays  $i$  vers le pays  $j$ , pendant l'année  $t$ ).

Les analyses de correspondance portent toujours sur des tableaux rectangulaires qui sont, soit l'un des tableaux de marge du tableau donné, soit un tableau (souvent mis en supplément) présentant, sous forme rectangulaire, l'ensemble parallépipédique des données. C'est d'ailleurs sous une telle forme rectangulaire que les données sont généralement saisies.

L'objet du programme 'wITxJ' (dont le nom même évoque une présentation rectangulaire d'un parallépipède) est de faire communiquer entre elles les diverses présentations des données; ce qui permet, de plus, de calculer facilement un tableau de marge binaire, à l'aide du programme 'cums', commandé par un listage ayant une ligne par bloc à cumuler.

Nous publions ci-après l'essentiel du dialogue de commande du programme 'wITxJ'.

S'affiche d'abord le commentaire:

'ce programme transforme ITxJ en TIxJ (ou réciproquement)'

'ou, encore, IxTJ en IxJT (ou réciproquement)';

Le programme demande alors le nom du tableau, 'Disq:tap', sur lequel on doit opérer; et charge ce tableau en mémoire centrale. Puis vient une question (où, comme on le verra, le verbe 'regrouper', doit aussi se comprendre comme 'permuter'):

'faut-il regrouper les lignes(L) ou les colonnes(C) '

Si la réponse est 'L', le programme affiche le nombre total des lignes du tableau, tel qu'il est donné:

'carI x carT = '(carti);

et vient un commentaire qui ne laisse aucun doute sur la structure de produit qu'a l'ensemble des lignes:

'ce programme transforme ITxJ en TIxJ (ou réciproquement)'

'on suppose que CHACUN des blocs consécutifs de lignes,'

'tel qu'il se présente au DÉPART, a pour cardinal carB';

si, par exemple, la présentation initiale est ITxJ, il y a, pour chaque i, un bloc de lignes consécutives indicé par  $t \in T$ ; e.g., dans l'exemple proposé ci-dessus, pour chaque pays exportateur i, un bloc de carT lignes afférente chacune à une année et  $\text{carB} = \text{carT}$ ; le programme demande alors le cardinal des blocs:

'le nombre carB est '...

Le programme vérifie alors que le nombre fourni en réponse divise exactement le cardinal de l'ensemble des lignes du tableau rectangulaire sur lequel on doit opérer; si ce n'est pas le cas, s'affiche un commentaire d'erreur; et après six erreurs l'exécution de 'wITxJ' tourne court... Mais si le nombre carB est acceptable et confirmé, 'wITxJ' crée un tableau 'Disq:tapBI.w'; i.e. un tableau où les lignes sont dans l'ordre demandé; et qui est, de plus, comme l'indique le suffixe '.w', un tableau de nombres réels écrits en format binaire (quel qu'ait été le format du tableau initial 'Disq:tap': entiers ou réels en format de texte, entiers écrits en binaire...).

Si, à la question:

'faut-il regrouper les lignes(L) ou les colonnes(C) ',

l'utilisateur a donné la réponse 'C', l'on procède, pour la permutation des colonnes, suivant une voie tout analogue à celle qu'on suit pour permuter les lignes. Il suffit donc de citer les répliques, sans commentaire.

'ce programme transforme IxJT en IxTJ (ou réciproquement)'

'on suppose que CHACUN des blocs consécutifs de colonnes,'

'tel qu'il se présente au DÉPART, a pour cardinal carB';

'le nombre carB est '...

Si ce nombre divise exactement le nombre total des colonnes, 'wITxJ' crée un tableau 'Disq:tapBJ.w'; i.e. un tableau où les colonnes sont dans l'ordre demandé; et qui est un tableau de nombres réels écrits en format binaire.

### 2.3 Le programme 'MARdiag' pour l'estimation de la diagonale d'un tableau carré d'après le critère de la trace minima

On sait que, dans une matrice de flux (par exemple une matrice de flux migratoires de commune à commune entre deux recensements consécutifs), les

termes diagonaux manquent. On peut certes considérer que ce qui reste en place au point  $i$  (dans l'exemple: la population non migrante) constitue un flux de  $i$  vers  $i$ ; mais cette assimilation engendre une diagonale très chargée qui perturbe l'analyse, avec de fortes valeurs propres et des facteurs qui n'expriment nettement que l'association de tel ou tel point  $i$  avec lui-même.

Dans diverses études, un raisonnement plausible a permis de calculer des termes diagonaux d'un ordre de grandeur acceptable, en fonction de l'ensemble des données disponibles (flux proprement dits et population totale); et l'analyse a été satisfaisante. Cependant K. Ben SALEM, a proposé une méthode générale pour créer la diagonale d'un tableau carré sans traiter d'autres données que les termes extradiagonaux (i.e., dans l'exemple, les flux): le critère étant, en bref, que, dans le tableau carré ainsi complété, la trace soit aussi faible que possible, ce qui prémunit, *a fortiori*, contre la présence des très fortes valeurs propres indésirables (cf. [TRAC. MANQ.], in *CAD*, Vol XVII, n°1, 1992).

Dans l'article [TRAC. MANQ. FLUX], (in *CAD*, Vol. XVII, n°3, 1992), un exemple d'application est présenté en détail, avec un algorithme de calcul. Le programme 'MARdiag' met en œuvre cet algorithme. Il nous suffira d'en citer le dialogue avec un bref commentaire, renvoyant à l'article.

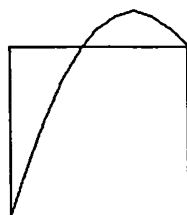
S'affiche d'abord l' en tête:

'ce programme crée une diagonale pour un tableau de flux'

Puis le nom, 'Disq:Tap', du tableau à modifier est acquis par le dialogue usuel; et 'MARdiag' vérifie qu'il s'agit d'un tableau carré, faute de quoi, l'exécution tourne court.

Dans une première étape, la diagonale manquante est calculée en ne retenant du tableau Tap donné que la marge du tableau obtenu en symétrisant ce tableau Tap (complété par une diagonale nulle).

Au fil du calcul, s'affiche la courbe convexe, dont l'intersection avec le côté supérieur du carré  $(0,1) \times (0,1)$  régit le calcul de la diagonale (cf. article, p.267). Et, sous le nom de 'Disq:TapB.w', est enregistré un tableau qui ne diffère du tableau donné qu'en ce que la diagonale en a été créée, uniquement d'après les termes extradiagonaux de 'Tap', sans tenir aucun compte des nombres, pourvus ou non de sens, inscrits initialement sur la diagonale.



Diag:FLhCa

S'affiche alors la question

'faut-il optimiser la trace sur le tableau symétrisé O ou N '

Si la réponse est 'Oui', on construit un nouveau tableau de la forme:

$$P_{II} + t \delta_{II}^I \cdot a_I ;$$



où  $p_{II}$  est le premier tableau complété qu'on vient de construire;  $\delta_{II}^I \cdot a_I$  est la matrice diagonale ayant même marge que le symétrisé de  $p_{II}$ ; et  $t$  est déterminé par le critère que soit minima la trace du symétrisé du nouveau tableau. Celui-ci est enregistré sous le nom de 'Disq:Tapr.w'.

## 2.4 Le programme 'coupli' de découpage des lignes d'un texte

Il est fréquent que partant d'un tableau dont toutes les lignes ont même format (à l'exception, peut-être, d'une en-tête), on doit produire un tableau qui ne conserve, de chaque ligne, que des segments déterminés; en introduisant, au besoin, des séparateurs: blancs, traits verticaux... D'autre part, on peut désirer ne conserver, de l'ensemble des lignes, que celles satisfaisant à des conditions logiques simples: en ce cas, nous parlerons de *filtre*.

Voici comment procède, à cet effet, le programme 'coupli' (dans sa version actuelle).

### Première question:

'le texte à modifier est ';

nous supposons que le texte demandé est 'Disq:Txt', et que le programme en a vérifié l'existence.

### Deuxième question:

'faut-il prendre toutes les lignes(T)

ou une fraction de l'ensemble des lignes(F) ';

si la réponse, confirmée est 'F', l'utilisateur spécifie, par dialogue, la fraction à retenir pour le traitements :

'la fraction à conserver est (1/nu) ; nu = ';

'le rang de la première ligne à prendre est rg ; rg = ';

le dénominateur  $nu$  peut être un entier de 1 à 100; le rang  $rg$  est un entier compris entre 1 et  $nu$ . Les lignes retenues sont alors celles de rang { $rg, rg+nu, rg+2.nu, \dots$ } (éventuellement, parmi les lignes satisfaisant à un filtre, dont le format est expliqué ci-dessous).

### Troisième question:

'y a-t-il un fichier modèle O ou N ';

si la réponse, confirmée, est 'Oui', le format des lignes à créer est défini par un fichier de commande; sinon, on procède par dialogue à l'écran. Nous commencerons par ce dernier cas, qui le plus simple. Après avoir spécifié, par dialogue, pour le nom du tableau à créer (cf. *infra*), un sigle, que nous supposons ici réduit au caractère '\$', on procède au découpage proprement dit.

### 2.4.1 Découpage par dialogue à l'écran

On peut seulement conserver certains segments pour les mettre bout à bout, sans interposer de caractères, ni imposer de filtre logique. D'où la première question:

'le nombre de segments à garder par ligne est ';

et, une fois ce nombre 's' confirmé, 'coupli' demande successivement les bornes de chacun des 's' segments (appelés encore blocs) par les deux questions:

'le rang du 1-ier caractère du bloc β... est '

'le rang du dernier caractère du bloc β... est ';

Dans sa version actuelle, le programme effectue de nombreuses vérifications pour assurer que les segments successifs n'empiètent pas; et qu'ils sont demandés dans l'ordre où ils se rencontrent dans la ligne: e.g., le segment 2..13 avant le segment 17..23 . Mais ces restrictions pourraient être supprimées au besoin; comme on l'a fait dans le cas du découpage avec filtre, au §2.4.3.

Quand le format a été fixé, le programme ouvre en lecture 'Disq:Txt'; et, simultanément, en écriture, un nouveau fichier 'Disq:Txt\*§'; (i.e., le sigle choisi, ici '§', termine le nom du texte créé). Chaque fois qu'est lue une ligne de 'Txt', on en prélève les segments spécifiés dans le dialogue; et ceux-ci sont immédiatement écrits dans 'Txt\*§'. Si la ligne lue dans 'Txt' n'est pas assez longue pour fournir tous les segments demandés, on complète celle-ci à l'aide du caractère '^', répété autant de fois qu'il est nécessaire.

Si seule une fraction des lignes est retenue (cf. 2-ème question), les nombres {rg, nu} sont indiqués par des lettres, dans le nom du fichier texte modifié créé, avant le caractère '\*'. De façon précise, le rang, *rg*, de la première ligne retenue, est noté par deux minuscules; et le taux d'échantillonnage, *nu*, par deux capitales; la principale utilité de cette mention est qu'elle évite qu'un nouveau tableau, n'en efface un déjà créé avec le même modèle, donc le même sigle.

### 2.4.2 Découpage d'après un fichier modèle sans filtre logique

Il faut d'abord trouver le modèle; d'où la question:

'le sigle du fichier modèle est ';

si, par exemple, ce sigle est 'μ', le nom du fichier modèle doit être: 'Disq:Txtμ'. Le programme en vérifie l'existence; et, en cas d'échec, le dialogue se poursuit; avec cette limite qu'après 6 erreurs on passe obligatoirement au mode de découpage par dialogue.

Supposons que 'Disq:Txtμ' existe. Voici quelle en doit être la structure.

### 2.4.2.1 Structure du fichier modèle

Après une ligne de titre, (obligatoirement terminée par un aller à la ligne), les blocs sont chacun spécifiés sur une ligne. Cette ligne commence par deux nombres qui sont, comme dans le dialogue à l'écran, le rang, dans la ligne de 'Txt' du premier et du dernier caractère du bloc. Immédiatement après le deuxième nombre, vient obligatoirement le caractère ';' (en fait, tout caractère, autre qu'un chiffre, conviendrait). La fin de la ligne, ou modèle de bloc, sert à spécifier le format à donner au bloc. Voici comment.

Ce modèle de bloc est une suite (non vide) de caractères, comprenant éventuellement des blancs qui ne jouent pas ici un autre rôle qu'une lettre, un chiffre ou un signe de ponctuation; le caractère '^' jouant seul un rôle distingué. De façon précise, il doit y avoir autant d'occurrences de '^' qu'il y a de caractères dans le bloc,  $\beta$ , lu dans 'Txt'; et ces occurrences sont remplacées, dans le modèle de bloc, par les lettres successives de  $\beta$ . (Si le nombre de '^' ne concorde pas, le modèle est sans effet; et le bloc  $\beta$ , lu dans 'Txt', est conservé sans modification.)

### 2.4.2.2 Exemple de fichier modèle

Il s'agit d'extraire certaines des réponses à un questionnaire.

```
Disq:Txtμ ; modèle pour extraire les réponses à 14 questions
5 8; .
142 155; ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^ ^
```

Le fichier proposé spécifie deux blocs. Le premier comprend les quatre caractères de rang 5 à 8; sans changement de format, puisque le modèle de bloc est réduit à '.'. Le deuxième bloc comprend les 14 caractères qui, dans la ligne lue sur 'Txt', vont du rang 142 au rang 155; ces caractères doivent être recopiés en introduisant des blancs comme séparateurs.

### 2.4.2.3 Création du texte modifié

On procède comme dans le cas du dialogue à l'écran; à ceci près que, d'une part, le fichier créé, ouvert en écriture, reçoit le nom 'Disq:Txt\*μ', où figure explicitement le sigle du modèle; et que, d'autre part, chaque fois qu'est lue une ligne de 'Txt', les segments spécifiés sont mis au format demandé, avant d'être écrits dans 'Txt\*μ'.

### 2.4.3 Découpage d'après un fichier modèle, avec filtre logique

Le dialogue s'engage comme on l'a expliqué au §2.4.2. Mais, si le sigle du fichier de commande commence par le caractère 'f' (par exemple, sigles  $f_1$ ,  $f_2$  dans les exemples ci-après), le programme traite ce fichier autrement: il supprime les contraintes relatives aux segments retenus; et interprète, s'il y a

lieu, certaines instructions comme définissant un filtre. Nous nous bornerons ici à considérer des exemples.

```
Disq:Txtf1 ; premier essai de découpage avec filtre
20038 7
20067 a..x
24 32;
7 9;
40 46;
```

Avec le fichier 'Disq:Txtf1', le filtre logique impose que, dans la ligne retenue, le 38-ème caractère soit un '7'; et le 67-ème, soit compris dans l'intervalle [a..x] du numérotage ASCII ; on combine trois segments, qui n'empiètent pas, mais dont l'ordre séquentiel n'est pas respecté, car celui qui va du 7-ème au 9-ème caractère est placé entre les deux autres; lesquels viennent après lui dans la ligne qui a été lue.

Il apparaît qu'un ordre de filtrage commence par un entier, somme du nombre 20000 et du rang du caractère sur lequel porte la condition spécifiée. Celle-ci fixe un caractère unique, ou un intervalle, noté, comme d'usage par ses extrémités séparées par deux points.

```
Disq:Txtf2 ; deuxième essai de découpage avec filtre
20038 7
20067 a..d
20067 ..
24 32;
17 29;
40 46;
20067 u..x
```

Sur l'exemple du fichier 'Disq:Txtf2', on voit que les conditions de filtrage peuvent être proposées dans un ordre quelconque, et mêlées à la description des segments à concaténer; (l'ordre de présentation n'étant significatif que pour ceux-ci). On notera que le deuxième segment empiète sur le premier. De plus, trois conditions portent sur le 67-ème caractère: celles-ci agissent de façon disjonctive; le 67-ème caractère pouvant être un blanc (que, pour éviter toute confusion, on convient de représenter par deux points consécutifs); ou être compris soit dans l'intervalle [a..d], soit dans l'intervalle [u..x].

## 2.5 Spécification des éléments supplémentaires en analyse des correspondances

On sait que le programme 'qori' d'analyse des correspondances offre à l'utilisateur de commander, par dialogue, la mise en supplément de lignes ou de

colonnes, soit une par une, soit par blocs. On peut, d'autre part, éliminer, une fois pour toutes, un sous-ensemble de lignes ou de colonnes en substituant au tableau à analyser un sous-tableau créé par le programme 'soustab'.

Mais il est apparu que, dans certaines applications, il convenait de mettre en supplément des sous-ensembles dont la structure est trop complexe pour qu'ils puissent commodément être décrits dans un dialogue; mais qu'il n'est pas permis de retrancher par 'soustab'.

C'est le cas, notamment, dans certaines applications cartographiques. Voici pourquoi. L'ensemble I des lignes du tableau analysé doit coïncider avec l'ensemble des unités territoriales du pays étudié; les régions à distinguer sur la carte étant définies par une partition de l'ensemble  $I_p$  des éléments principaux; tandis que l'ensemble  $I_s$  des éléments supplémentaires est laissé en blanc.

Il est, en particulier, naturel de mettre en supplément les lignes ne contenant que des zéros; en d'autres termes, les unités territoriales d'où est absent le phénomène étudié (e.g. telle activité économique). Autrement, ces éléments s'agrègent, à la CAH, avec le premier élément non nul venu; constituant une classe où seul ce dernier élément mérite de n'être pas en blanc. Or dans la pratique, il peut être difficile d'énumérer, par dialogue, les lignes nulles; tandis qu'il est assez facile d'en constituer la liste (notamment d'après les résultats d'une première CAH; où, ainsi qu'on l'a dit, ces éléments sont groupés; et donc mis en liste).

Désormais, le programme 'qori' accepte des fichiers de commande spécifiant un ensemble d'éléments supplémentaires, soit lignes, soit colonnes.

De façon précise, supposons qu'on analyse le tableau 'disq:tab'. Si, à la question:

y a t il ligne supplémentaire(S) ou non(N)

l'utilisateur a répondu 'S', le programme cherche s'il existe un fichier 'disq:tabisupx'; et, si tel est le cas, lit sur ce fichier la liste des numéros des lignes supplémentaires. De même, pour les colonnes supplémentaires, le programme cherchera un fichier 'disq:tabjsupx'.

Reste à préciser la structure d'un fichier de commande 'supx'.

Ce fichier est un texte, commençant par un titre libre, terminé par un aller à la ligne. Viennent alors les numéros des éléments supplémentaires, donnés dans un ordre quelconque; avec, pour désigner des blocs, les mêmes conventions que dans les listages de commande destinés à d'autres programmes: c'est-à-dire que la présence d'un nombre  $10000+x$ , supérieur à 10000, est interprété comme demandant la mise en supplément de  $x$ , ainsi que de tous les éléments dont le rang est compris entre  $x$  et le prochain entier rencontré dans le listage de commande.

Il importe de noter que la procédure de lecture des nombres est telle que tout caractère du listage 'supx' autre qu'un chiffre (e.g.: un aller à la ligne, une lettre, un signe de ponctuation...) est interprété comme un séparateur ayant même signification qu'un espace blanc. L'utilisateur peut donc, à sa convenance, inclure, dans 'supx', des repères et commentaires. Par exemple, le listage suivant:

```
disq:tabjsupx : exemple de supplémentaires
premier bloc : 10007 45
deuxième bloc : 10115 133
isolés       : 151 211 221
```

commandera la mise en supplément des deux blocs {7...45} et {115...133}; et des trois éléments {151, 211, 221}.

### 3 Conclusion: le tableur de l'avenir

Existents depuis plusieurs années des logiciels intégrés, appelés en anglais 'spread-sheet' et en français 'Tableur', dont la fonction est, en bref, de modifier un tableau et d'en tirer des graphiques. Ces logiciels ont une présentation brillante et un menu de fonctions très étendu; mais comme ils ne comportent pas de procédure d'analyse de données proprement dite, toute la vélocité de l'utilisateur, certes bien servi par l'outil informatique, ne suffit pas à offrir de la structure des données des vues globales telles que celles qu'élaborent rapidement l'analyse des correspondances et la CAH.

Dans les présentations les plus élémentaires, par courbes et histogrammes, les tableurs manquent de mettre en relief les différences pertinentes; faute d'introduire la notion de profil (comme le fait 'Grftab", cf. §1.1) qui élimine des courbes particulières la tendance générale; ou de cadrer de multiples façons les histogrammes en y insérant les noms des individus (comme le fait 'zrang'; mais l'écriture régulière de noms suppose l'emploi systématique de sigles dans les étiquettes; plutôt que de numéros ou de noms usuels).

Parfois, devant ces diagrammes circulaires qu'on appelle en anglais 'tarte' (pie-chart) et en français 'fromage' (ou camembert), on se demande si le but n'est pas de faire mousser l'information; alors qu'il convient de la condenser en usant de toutes les ressources du calcul et du tracé!

On souhaite donc disposer dans l'avenir d'un tableur qui, intégrant dans la présentation des données les notions générales de codage et de créations d'éléments que mettent en œuvre, dans MacSAIF, 'zrang', 'zBurt', 'cums'..., offrirait à l'utilisateur les transformations profondes que réalisent l'analyse factorielle et la CAH, pour en présenter graphiquement les résultats.

De tels progrès sont encore plus urgents dans l'accès aux bases de données, où, sans une élaboration linguistique préalable fondée sur l'analyse des

données, les cribles logiques qui viennent à l'esprit sont d'un si médiocre rendement que l'abondance des informations accable l'utilisateur plus qu'elle ne le sert.

**Note: Versions du logiciel et compatibilité avec les divers systèmes d'exploitation**

Les programmes du logiciel MacSAIF existent sous différentes versions, distinguées par un indice numérique: ainsi, du programme 'qori' d'analyse des correspondances, existent des versions qori2, qori5, qori1, qori6, qori7... Quant au format des fichiers qu'elles lisent où qu'elles créent, ces versions sont totalement compatibles entre elles; mais elles diffèrent quant à l'utilisation des ressources des divers microordinateurs MacIntosh.

La version '2', compilée en TML Pascal, est, croyons-nous, compatible avec les divers systèmes d'exploitation; mais elle n'utilise ni les coprocesseurs arithmétiques, ni les éventuelles instructions propres à un processeur 68020, 68030: elle se limite aux ressources d'un 68000; et ne bénéficie des perfectionnements des nouveaux ordinateurs que pour l'accélération du cycle de base.

La version '5', compilée en TML Pascal II, requiert un processeur 68020 ou 68030 et un coprocesseur arithmétique: elle nous satisfait pleinement quant à la rapidité; mais n'est pas totalement compatible avec le système 7; ce qui semble sans remède, le compilateur n'ayant pas été revu.

La version '1' est compilée en TML Pascal II, sans sortir des limites des ressources du 68000: malgré cela, elle dépasse en célérité la version '2'.

Les versions '7' et '6', respectivement écrites pour 68020 (ou 30) et coprocesseur, ou en code 68000 strict, ont été compilées en THINK Pascal. Elles sont certainement compatibles avec les versions les plus récentes des systèmes d'exploitation; remplissent avec élégance les fonctions graphiques, ne diffèrent pas de la version '5' quant à la rapidité des calculs en mémoire centrale; mais, au moins si l'on conserve tels quels les algorithmes de la version '5', les entrées et sorties sont moins rapides qu'avec cette dernière.

Nous venons de faire allusion aux algorithmes... Le passage de TML Pascal à TML Pascal II a présenté quelques difficultés; notamment parce que TML II, affecte à toute chaîne de caractères 256 octets, même si l'on déclare string[5]; en revanche, une fois la stratégie de passage établie, THINK accepte presque tels quels les programmes destinés à TML II.

### Références bibliographiques

K. Ben SALEM, M.-M. THOMASSIN: «Estimation de la diagonale d'un tableau carré d'après le critère de la trace minima; application à l'analyse des flux de population à La Martinique entre 1975 et 1982», [TRAC. MANQ. FLUX], in *CAD*, Vol. XVII, n°3, pp. 261-274; 1992.

J.-P. & F. BENZÉCRI: «Programmes d'analyse des correspondances et de classification ascendante hiérarchique: notice d'utilisation, [NOT. CORR. CAH], in *CAD*, Vol. XIV, pp. 7-34; 1989.

J.-P. & F. BENZÉCRI: «Programmes de création de tableaux: notice d'utilisation, [NOT. CRÉ. TAB.], in *CAD*, Vol. XIV, n°1, pp. 35-54; 1989.

J.-P. & F. BENZÉCRI: «Programme de représentation cartographique des résultats d'une analyse multidimensionnelle: notice d'utilisation, [NOT. PROG. CART.], in *CAD*, Vol. XIV, n°1, pp. 55-72; 1989.

J.-P. & F. BENZÉCRI: «État des recherches sur la cartographie automatique des données statistiques, [STAT. CART.], in *CAD*, Vol. XIV, n°4, pp. 449-466; 1989.

J.-P. & F. BENZÉCRI: «Liste des programmes du logiciel MacSAIF, Système d'Analyse des InFormations, et description de leurs fonctions», [MacSAIF], in *CAD*, Vol. XV, n°3, pp. 359-366; 1990.

J.-P. & F. BENZÉCRI: «Typologie de textes grecs d'après les occurrences de formes de mots, [TEXTES GRECS 2], in *CAD*, Vol. XVIII, n°2, pp. 449-466; 1993. [Cet article se termine par une bibliographie relative aux programmes de préparation de données linguistiques.]

J.-P. BENZÉCRI, C. HERBIN, G. D. MAÏTI, D. de VERNEJOUL, D. SAUR: «Efficacité comparée de deux traitements antihypertenseurs chez des patients suivis par enregistrement de la pression artérielle au cours du nyctémère», MC2 n°9: [PRESS. ART. NYCTH.], in [Prat MÉD.], pp. 345-364 (cf. *CAD*, Vol. XV, n°4, 1990).

J.-P. & F. BENZÉCRI, M.-M. THOMASSIN: «Utilisation par le logiciel MacSAIF de fonds de carte digitalisés créés par divers logiciels», [CARTE DIGIT.], in *CAD*, Vol. XVI, n°3, pp. 333-340; 1991.

Th. ETTÉ: «Le marché mondial du cacao sur onze années, de 1976/1977 à 1986/1987, [CACAO], in *CAD*, Vol. XVI, n°1, pp. 35-50; 1991.

Th. ETTÉ: «Le marché mondial du café sur six années, de 1983/1984 à 1988/1989, [CAFÉ], in *CAD*, Vol. XVI, n°3, pp. 273-292; 1991.



M. OUASSOU: «Action comparée d'un médicament et d'un placebo et épuisement de leurs effets», [ÉPUISEMENT], in *CAD*, Vol. XVIII, n°2, pp.133-142; 1993.

**N.B.** Les lecteurs désireux d'acquérir les programmes du Logiciel **Mac-SAIF**, ('Système d'Analyse des InFormations'), dont le présent article décrit les récentes mises à jour, s'adresseront à la

---

Société **STATMATIC** : 4, rue de Fécamp, 75012 Paris;  
et: 6/8, Avenue Salvador Allende, 93-804, Épinay/Seine;  
Téléphone : (16.1) 48.27.05.49 / (16.1) 47.92.07.86;  
Fax : (16.1) 48.27.32.48.

---