

J.-P. BENZÉCRI

F. BENZÉCRI

Sources de programmes d'analyse de données en langage PASCAL : (II) : élaboration des fichiers de facteurs (IIC) : FACOR : aide à l'interprétation en classification ascendante hiérarchique

Les cahiers de l'analyse des données, tome 22, n° 2 (1997), p. 155-166

http://www.numdam.org/item?id=CAD_1997__22_2_155_0

© Les cahiers de l'analyse des données, Dunod, 1997, 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/>

SOURCES DE PROGRAMMES D'ANALYSE DE DONNÉES EN LANGAGE PASCAL: (II) : ÉLABORATION DES FICHIERS DE FACTEURS (IIC) : FACOR: AIDE À L'INTERPRÉTATION EN CLASSIFICATION ASCENDANTE HIÉRARCHIQUE

[SOURCES PASCAL (IIC)]

J.-P. & F. BENZÉCRI

0 Introduction : données traitées pour l'interprétation par FACOR

Ainsi qu'on l'a dit en IIB§0, la structure d'une CAH se comprend mieux si l'on dispose d'aides à l'interprétation; i.e., en bref, si l'on considère, dans l'espace, les centres des classes; l'orientation des dipôles; la contribution de chaque dipôle à l'inertie sur chaque axe... Plus précisément, pour FACOR, l'espace est rapporté au système des axes factoriels.

C'est pourquoi, comme la CAH et le tracé d'arbre, le programme FACOR utilise, pour seule donnée de base, le fichier de facteurs 'Fac.w', afférent à un ensemble E. Mais les résultats de la CAH et la description de la partition retenue sont transmis, à FACOR, sous forme de fichiers.

1 Demande d'aide à l'interprétation: structure du programme 'FAX'

1.1 Le programme principal

Le programme principal consiste simplement en une boucle, appelant trois procédures: 'ouvrir', 'Facx' et 'Facw'; dont les deux dernières constituent des unités séparées: uFacxx.p, et: uFacww.p.

Par 'ouvrir', l'utilisateur choisit le tableau de base et l'ensemble E qui doit avoir déjà été soumis à la CAH. Si le fichier 'Fac.w', ainsi que les fichiers de résultats créés par 'CLH', se trouvent sur le disque qu'on a proposé, la variable d'existence 'rep' est mise à 'O' (oui). En ce cas, s'exécutent les procédures 'Facx' et 'Facw', qui renferment tout ce qui concerne l'aide à l'interprétation proprement dite (cf. *infra*, §§2, 3).

Après avoir lu les fichiers utiles, 'Facx' crée, en mémoire centrale, des tableaux d'après lesquels est créé, comme un fichier de texte, le listage FACOR.

Puis 'Facw', d'après les mêmes tableaux créés par 'Facx', crée deux listages 'Fac.w': l'un 'qFac.w', pour l'ensemble q des centres des classes de la partition retenue pour E (dans la procédure 'arbt' de 'CLS': cf. IIB§3.2.2); l'autre, 'nFac.w', pour l'ensemble des nœuds de la hiérarchie partielle afférente à cette même partition.

```

begin;benzecri;rep='O';
while (rep='O') do begin
  ouvrir;unloadseg(@ouvrir);
  if (rep='O') then begin
    nomf:=concat(nombas,chat);
    Facx(nomf,che,pFc,pGc);unloadseg(@Facx);
    Facw(nomf,pFc,pGc);unloadseg(@Facw);
    write('faut-il faire un autre FACOR oui(O) ou non(N) ');
    readln(rep);if not (rep='N') then rep:='O';
  end;end;
readln(rpv);end.

```

Ces fichiers 'Fac.w' peuvent être soumis à divers programmes que nous avons déjà décrits; notamment: pour le tracé de graphiques, à 'PlanF' ou 'PlanX' (cf. IC); pour l'analyse discriminante, à 'discr' (principalement, comme décrivant l'ensemble des centres: cf. IIA); mais aussi à la CAH elle-même: cf. IIB). Ainsi, dans ([TEXTES GRECS 3D] §5.4, in CAD, Vol.XIX, n°2) après avoir construit, par CAH(50) (i.e. en prenant en compte 50 facteurs), une partition de E, en 47 classes; on recompose cet ensemble par CAH(9); l'avantage étant que CAH(50) reconnaît le mieux les similitudes entre éléments voisins; tandis que CAH(9) offre la vue d'ensemble la plus satisfaisante de l'ensemble des classes créées par CAH(50).

Éventuellement, par rep='O' (plus précisément: rep≠'N'), l'utilisateur peut demander que soient traitées d'autres données.

1.2 Déclarations du programme 'FAX'

```

program FAX;
uses memtypes,quickdraw,osintf,toolintf,uver;
var
  nombas,nomf,chat:str255;
  rep,rvp,che:char;
  pFc,pGc:ptr;
  erl,resu,exar:integer;
{$S segouv}
procedure verficher(sfx:str255);begin
  if (resu=1) then begin
    nomf:=concat(nombas,chat,sfx);
    resu:=verif(@nomf) div 2;
    writeln(' resu =',resu:2,' pour ',nomf);end;
end;
procedure verarber;begin
  resu:=1;verficher('cnabd');verficher('qcqabc');
  verficher('qcmabc');verficher('cqki');verficher('Fac.w');
end;

```

En dehors des noms de fichiers et du sigle de l'ensemble E, le programme FAX n'a pour variables, qui lui soient propres, que des caractères de réponse, et des entiers auxiliaires. Les pointeurs {pFc, pGc} servent à la procédure 'Facx' pour créer des tableaux qui sont transmis à 'Facw'. Le caractère cht, est: 'i', si INR et CTR sont à rapporter aux valeurs propres issues de l'analyse du tableau principal; sinon, che='a', et les INR et CTR sont rapportées à l'inertie du nuage des centres des classes (plus précisément, cf *infra*, §2.1, à l'inertie du nuage des centres relativement à l'origine de l'axe factoriel; origine

qui n'est pas, en général, le centre de gravité afférent à E: mais seulement celui de I_p ou J_p , ensembles des éléments principaux dans l'analyse du tableau de base). La première éventualité n'est proposée à l'utilisateur que si $E=I_p$ ou J_p ; auquel cas les valeurs propres sont effectivement les inerties de l'ensemble E sur les axes.

Par la procédure 'verarber' (qui appelle 'verficher': procédure analogue à 'verichaine', décrite dans IA§1.3.2; et appelée dans 'discri': cf. ID§1) vérifie l'existence d'une chaîne de fichiers utiles à 'FAX'.

1.3 Choix des données et format des contributions: la procédure 'ouvrir'

La procédure 'ouvrir' comprend deux parties: choix des données et choix du mode de calcul des taux: {INR, CTR}.

Le choix des données se fait, à peu près, comme dans la procédure de même nom de 'CLH' (cf. IIB§1.2); avec vérification en chaîne par 'verarber'.

On a déjà dit, au §1.2, comment le calcul des taux est régi par 'che'.

```

procedure ouvrir;
begin erl:=0;rep:='N';
while not ((rep='O') or (erl=6)) do begin
  rep:='O';exar:=1;
  write('le fichier de base est ');readln(nombas);
  if not (setvol(@nombas,10) = noerr) then begin
    rep:='N';erl:=erl+1;writeln('ERREUR le disque manque ') end
  else begin
    write('le sigle de l''ensemble à traiter par FACOR est ');readln(chat);
    verarber;exar:=resu;
    if (exar=0) then begin rep:='N';erl:=erl+1;
      writeln('ERREUR les données manquent pour FACOR') end end;
  if (rep='O') then begin
    writeln('le nom du fichier de base, ',nombas);
    write('et de l''ensemble, ',chat,', sont-ils confirmés oui(O) ou non(N) ');
    readln(rep) end;
  end;
che:='a';
if ((rep='O') and ((chat='i') or (chat='j'))) then begin che:='i';
  writeln('INR et CTR peuvent être rapportées aux valeurs propres');
  writeln('ou à l''inertie du nuage des centres des classes');
  writeln('faut-il rapporter les INR et CTR à l''inertie du');
  writeln('nuage des centres des classes de la partition');
  write('retenue,    oui(O) ou non(N) ');readln(rpv);
  if (rpv='O') then che:='a' end;
if (che='i') then writeln('INR et CTR seront rapportées aux valeurs propres');
end;
{$S segFax} {$U uFacxx5}
procedure Facx (nomen:str255;var chf:char;var pF,pG:ptr);
external;
{$S segFaw} {$U uFacww5}
procedure Facw (nomen:str255;pF,pG:ptr);
external;

```

Les déclarations s'achèvent en donnant les noms des unités où sont les procédures 'Facx' et 'Facw'.

2 Lecture des données; calcul de tableaux dans le système des axes factoriels; création du listage FACOR: la procédure 'Facx'

Après les déclarations, §2.1, nous distinguerons trois parties: lecture des fichiers utiles et calcul de tableaux en mémoire centrale, §2.2; création du listage FACOR, §2.3.

2.1 Déclarations en tête de la procédure 'Facx'

```
UNIT uFacxx5;
INTERFACE uses memtypes, quickdraw, osintf, toolintf, uver5;
procedure Facx(nomen: str255; var chf: char; var pF, pG: ptr);
implementation procedure Facx;
const
  afmax=30; qmax=100; qdmax=200;
type
  cab=array[0..abmax] of single; pcab=^cab;
  ptqda=array[1..qdmax] of pcab; ppqda=^ptqda;
var
  a, c, cq, el, eh, i, j, ij, n, m, ms, q, carq, care, carf, dir, amax, cara, a1, a2, carb,
  CLAS, AINE, BNJM, QLT, PDS, INR, FCT, CO2, CTR, qa, qb, za, zb, zi: integer;
  ffs, trac, poit: single; res, ppp: extended; rpv: char;
  riri: ricla; ptrcl: ptrc;
  nomf: str255;
  fin: file of integer; ft: text;
  cdq: array[1..qdmax] of integer;
  qab: array[1..2] of integer; qde: ptni; nouv: ptr;
  qAm, qBm, cAq, cBq: array[1..qmax] of integer;
  Fqda: ppqda; ptt: ptni;
  lama, lamb: array[1..amax] of single;
  poiq, poim, dq, dm, dd: array[1..qmax] of single;
  function pzq(zz, qq: integer): extended; begin if (zz=0) then
    pzq:=poim[qq-carq] else pzq:=poiq[qq]; end;
```

Comme dans 'arbt' (cf. IIB§3), d'où vient la partition à traiter, le nombre des classes est $\leq q_{\max}=100$; le nombre double, $q_{\max}$, permet de loger des informations afférentes aux nœuds et terminaux de la hiérarchie associée à la partition. La constante: $af_{\max}=30$, est le nombre maximum de facteurs figurant sur le listage 'FACOR'. Ce nombre est élevé, relativement au maximum de 10 pour le listage de 'qori' (cf. IB§3.1 in fine); ou de 8 pour 'qorlsup' (cf. ID§3.2): mais on l'a introduit ici parce que, dans certaines CAH, on a vu des dichotomies interprétables dont l'explication était au-delà des 10 premiers facteurs.

La plupart des variables renvoient à des notions déjà vues ailleurs; notamment dans IIB, pour la CAH. Nous signalons seulement que le tableau usuel, lama, des valeurs propres, est doublé d'un tableau lamb, contenant les inerties calculées sur chaque axe pour l'ensemble E (cf. §1.3: format des taux).

La fonction 'pzq' est analogue à la fonction 'carc' déclarée en tête de 'arbt'. L'une et l'autre fonction, servent à distinguer entre nœud et terminal. Pour 'pzq', le cas $zz=0$, correspond à un nœud.

2.2 Lecture des fichiers afférents à l'ensemble E et à la partition et calcul de tableaux en mémoire centrale

```

begin ptrcl:=ptrc(@riri);
pF:=newptr(2*(titab+2+(3*abmax)));ptt:=ptni(pF);
pG:=newptr(4*qdmax);Fqda:=ppqda(pG);
for a:=1 to qdmax do begin
  nouv:=newptr(4*(abmax+1));Fqda^[a]:=pcab(nouv) end;
nomf:=concat(nomen,'Fac.w');reset(fin,nomf);
for a:=1 to titab-1 do read(fin,c);read(fin,care);close(fin);
nouv:=newptr(2*care);qde:=ptni(nouv);
nomf:=concat(nomen,'cqki');reset(fin,nomf);read(fin,carq);
for q:=1 to carq do begin
  read(fin,c);cdq[q]:=c-20000;read(fin,cq);cq:=cq-10000;
  for el:=1 to cq do begin read(fin,eh);qde^[eh]:=q end end; close(fin);
for q:=1 to carq do begin
  poiq[q]:=0;for a:=1 to aamax do Fqda^[q]^[a]:=0 end;

```

Le tableau ptt^a sert à lire l'en-tête du fichier 'Fac.w' des facteurs pour l'ensemble E. Il sera transmis à la procédure 'Facw', par le pointeur pF, pour créer l'en-tête des fichiers de facteurs 'qFac.w' et 'nFac.w' (cf. *infra*, §3).

Par les pointeurs $Fqda^a[q]$, on accède aux tableaux $Fqda^a[q]^a$, lesquels contiennent les coordonnées (facteurs, poids, distance; et numéros dans la CAH générale de E: $Fqda^a[q]^a[0]$) pour chacun des nœuds et terminaux de la hiérarchie associée à la partition traitée. Ces informations, essentielles, passeront à 'Facw' par le pointeur pG.

En ouvrant (pour le refermer aussitôt) le fichier: 'Fac.w', des facteurs pour E, 'Facx' lit le cardinal care; afin de créer un tableau: qde^a , dans lequel sera le numéro de la classe q ou rentre chaque élément, e, de E.

C'est dans le fichier 'cqki', que la procédure 'Facx' lit le contenu de qde^a ; simultanément, est gardé le numéro $cdq[q]$ afférent à la classe q, dans la hiérarchie générale de E. Ce numéro sera écrit sur le listage FACOR.

Le poids: $poiq[q]$, et les facteurs: $Fqda^a[q]^a$, afférents à la classe q, sont mis à zéro, avant d'être calculés, d'après le fichier 'Fac.w' de E. Les facteurs sont calculés par la formule barycentrique; les poids sont réduits à avoir pour total 1.

```

nomf:=concat(nomen,'Fac.w');reset(fin,nomf);
with riri do begin
  for a:=1 to titab+1 do read(fin,ptt^[a]);
  ptt^[titab]:=carq;amax:=ptt^[titab+1]-2;dir:=7+(2*amax);
  for a:=1 to dir do ptrcl^[a]:=ptt^[61+a];
  for a:=1 to 3*(2+amax) do read(fin,ptt^[a+titab+1]);
  for a:=1 to amax do lama[a]:=Fac[a];
  for el:=1 to care do begin
    for a:=1 to dir do read(fin,ptrcl^[a]);
    q:=qde^[el];poiq[q]:=poiq[q]+Fac[amax+1];
    for a:=1 to amax do Fqda^[q]^[a]:=Fqda^[q]^[a]+(Fac[a]*Fac[amax+1]);
    write(sigler(sis):5);if (el mod 16=0) then writeln end end;
close(fin);writeln;poit:=0;
for q:=1 to carq do begin res:=poiq[q];poit:=poit+res;
  if (res<>0) then res:=1/res;
  for a:=1 to amax do Fqda^[q]^[a]:=Fqda^[q]^[a]*res end;
if not(poit=0) then res:=1/poit else res:=0;
for q:=1 to carq do poiq[q]:=poiq[q]*res;

```

```

nomf:=concat(nomen,'cnabd');reset(fin,nomf);
read(fin,c);read(fin,c);read(fin,c);
cara:=(c+32768) div 256;
for a:=1 to cara do begin lamb[a]:=0;
  for q:=1 to carq do
    lamb[a]:=lamb[a]+(poi[q]*sqr(-Fqda^[q]^a)) end;
if (chf='i') then
  for a:=1 to cara do if (lama[a]<lamb[a]) then chf:='a';
if (chf='a') then for a:=1 to cara do lama[a]:=lamb[a];
trac:=0;for a:=1 to cara do trac:=trac+lama[a];
close(fin);

```

Dans le fichier 'cnabd' (cf IIB§2.4), est lu le nombre: cara des axes utilisés pour la CAH générale. Sur chacun de ces axes, on calcule l'inertie: lamb[a], du nuage des centres des classes relativement à l'origine (cf. §2.1). [Le signe '-', dans la forme: $\text{sqr}(-Fqda^q[a])$, surprendrait un algébriste novice: on l'a mis parce que la procédure sqr, élévation au carré, de TML2, veut un argument 'extended'; format que le 'single': $Fqda^q[a]$, prend instantanément changement de signe]. Si chf='i', les {INR, CTR} sont à rapporter aux valeurs propres: lama[a], (cf. §1.2), on vérifie que celles-ci ne sont pas inférieures aux inerties calculées pour le nuage des centres; auquel cas on conclut qu'un changement de nom de fichier a troublé les données; et l'on se rapporte à l'inerti calculée. Finalement, les dénominateurs qui doivent servir au calcul de {INR, CTR} sont mis dans le tableau lama.

```

nomf:=concat(nomen,'qcmabc');reset(fin,nomf);read(fin,m);
for m:=carq-1 downto 1 do read(fin,qAm[m],qBm[m],cdq[m+carq]);
close(fin);
nomf:=concat(nomen,'qcqabc');reset(fin,nomf);read(fin,q);
for q:=1 to carq do read(fin,cAq[q],cBq[q],a);close(fin);

```

Dans le fichier 'qcmabc' (cf. IIB§3.3.3), on lit la structure hiérachique de la partition considérée. De façon précise, si m est le numéro (de 1 à carq-1), d'un nœud de cette hiérarchie, on a dans: cdq[m+carq], le numéro de ce nœud considéré dans Eé; et dans: qAm[m] et qBm[m], les numéros de ses deux descendants, considérés comme appartenant à la hiérarchie restreinte. On notera que les valeurs: cdq[1..carq], on déjà été lues dans le fichier 'cqki'.

Dans le fichier 'qcqabc' on lit, après la valeur carq+10000, une suite de carq triplets {cAq[q], cBq[q], a}, afférents aux classes la partition retenue: a, est le numéro (non gardé ici) de la classe dans Eé; et, s'il s'agit d'un nœud, cAq[q] et cBq[q] sont, dans Eé, les numéros de son aîné et de son benjamin; sinon: cAq[q]=cBq[q]=0.

Le poids: poi[q], et les facteurs: $Fqda^q[a]$, ont déjà été calculés pour une classe q de la partition de E retenue. Il reste à les calculer pour les nœuds de la hiérarchie restreinte. On procède par voie ascendante: de sorte que, lors du traitement d'un nœud, on a déjà les valeurs afférentes à son aîné et à son benjamin (qu'il s'agisse d'un nœud ou d'une classe).

```

for m:=1 to carq-1 do begin qab[1]:=qAm[m];qab[2]:=qBm[m];
  for a:=1 to amax do Fqda^[m+carq]^a:=0;poim[m]:=0;
  for i:=1 to 2 do if (carq<qab[i]) then begin ms:=qab[i]-carq;
    ffs:=poim[ms];poim[m]:=poim[m]+ffs;
    for a:=1 to amax do
      Fqda^[m+carq]^a:=Fqda^[m+carq]^a+(ffs*Fqda^[ms+carq]^a) end
    else begin q:=qab[i];
      ffs:=poiq[q];poim[m]:=poim[m]+ffs;
      for a:=1 to amax do
        Fqda^[m+carq]^a:=Fqda^[m+carq]^a+(ffs*Fqda^[q]^a) end;
      if (poim[m]>0) then begin ffs:=1/poim[m];
        for a:=1 to amax do Fqda^[m+carq]^a:=Fqda^[m+carq]^a*ffs end end;

```

Les valeurs, d'abord mises à zéro, sont calculées comme sommes de deux termes. Comme pour les classes, on se garde de diviser par zéro.

2.3 Création du listage FACOR

NB : QLT, INR et CO2 sont calculées dans l'espace engendré par les 7 premiers axes utilisés pour construire la CAH
 INR et CTR sont rapportées à l'inertie du nuage des centres des classes de la partition retenue
 Dur:W:PhtBsaFacor

CLAS AINE BNJM QLT PDS INR F 1 CO2 CTR F 2 CO2 CTR F 7 CO2 CTR											
repr sur les axes factoriels des 4 noeuds choisis											
25	24	23	01000	0	0	0	0	0	0	0	0
24	13	21	1000	528	376	144	916	432	31	41	176
23	22	16	1000	472	421	-161	916	483	-34	41	196
22	14	20	1000	353	310	-148	788	307	-49	87	304
repr sur les axes fact des 5 classes de la partition choisie											
13			1000	99	279	254	722	253	136	206	648
21	18	4	1000	429	209	119	909	238	6	3	6
14	1	3	1000	67	34	-74	347	15	-85	452	171
20	19	6	1000	286	315	-166	786	310	-41	48	170
16	2	11	1000	119	164	-198	903	185	10	2	4
CDIP AINE BNJM QLD PDS IND D 1 COD CTD D 2 COD CTD D 7 COD CTD											
repr sur les axes factoriels des 4 dipôles choisis											
25	24	23	10001000	797	305	916	915	65	41	372	3
24	13	21	1000	528	112	136	419	59	130	381	479
23	22	16	1000	472	53	50	131	9	-60	188	112
22	14	20	1000	353	38	92	380	18	-44	87	37

Comme dans IB§3.1, nous soutenons, par un exemple de listage, le commentaire de l'algorithme.

2.3.1 Structure d'ensemble du listage et écriture de l'en-tête

On distinguera, dans le listage FACOR, un en-tête et des bandes de résultats numériques. Chaque bande, peut être divisée, d'une part, horizontalement en 3 tranches; d'autre part, verticalement, en blocs de colonnes. Le listage donné comme exemple n'a qu'une seule bande.

Dans chaque bande, la première et la deuxième tranche concernent respectivement, l'ensemble des nœuds et l'ensemble des classes de la partition

retenue; avec les informations usuelles d'un listage d'analyse factorielle (complétées par la description de la hiérarchie). La troisième tranche, afférente aux dipôles, considère non les vecteurs joignant, dans l'espace ambiant, un point (classe ou nœud) à l'origine; mais les vecteurs joignant les deux descendants d'un même nœud.

Si le nombre: carf, des axes considérés sur le listage est ≤ 10 , il n'y a qu'une seule bande; avec deux premiers blocs de colonnes afférents, respectivement, à la structure hiérarchique et à la qualité globale de la représentation; et, pour les axes de rang 1 à care, une suite de care blocs comprenant chacun trois colonnes. Si carf > 10, il y a plusieurs bandes: la première afférente aux axes de rang 1 à 10; la seconde, aux axes suivant, jusqu'à concurrence de 10; et, si carf > 20, il y a même une troisième bande. On reconnaît, sur le listage, les sigles usités en analyse des correspondances; avec, dans la troisième tranche, la lettre: D, afin de rappeler qu'il s'agit de dipôles.

```
nomf:=concat(nomen,'Facor');rewrite(ft,nomf);
writeln(ft,'NB : QLT, INR et CO2 sont calculés dans l''espace');
write(ft,'engendré par les',cara:3,' premiers axes ');
write(ft,'utilisés pour construire la CAH');writeln(ft);
if (chf='i') then
  writeln(ft,'INR et CTR sont rapportées aux valeurs propres');
if (chf='a') then begin
  writeln(ft,'INR et CTR sont rapportées à l''inertie du nuage des');
  writeln(ft,'centres des classes de la partition retenue') end;
carf:=cara;if (afmax<carf) then carf:=afmax;
while (lama[carf]=0) do carf:=carf-1; writeln(ft,nomf);
a1:=1;if (carf<(a1+10)) then a2:=carf else a2:=a1+9;
while (0<a1) do BEGIN
  carb:=a2+1-a1;
  for a:=1 to 29+(14*carb) do write(ft,' '); writeln(ft);
  write(ft,'|CLAS AINE BNJM| QLT PDS INR|');
  for a:=a1 to a2 do write(ft,' F',a:2,' CO2 CTR|'); writeln(ft);
  for a:=1 to 29+(14*carb) do write(ft,'_'); writeln(ft);
```

Le nombre, carf, des axes considérés sur le listage est, d'abord, pris égal au nombre, cara, des axes ayant servi à la CAH. Mais, d'une part, carf ne peut excéder afmax=30 (cf. supra §3.1); et, d'autre part, ne peut être retenu qu'un bloc d'axes consécutifs sur lesquels l'inertie est non nulle.

La boucle: while (0<a1) do BEGIN, est parcourue autant de fois qu'il y a de bandes; avec pour régir l'écriture des facteurs dans la bande, le rang initial, a1 et le rang final, a2. Pour la 1-ère bande, a1=1 et a2=inf(carf, 10). L'entrée dans une nouvelle itération dépend du calcul de nouvelles valeurs pour {a1, a2}; calcul effectué en fin de boucle par les instructions ci-après:

```
if (a2=carf) then a1:=0;
if (a2<carf) then begin a1:=a2+1;
  if (carf<(a1+10)) then a2:=carf else a2:=a1+9 end;
END;
```

2.3.2 Ecriture de la première tranche d'une bande: les nœuds

```
writeln(ft,'repr sur les axes factoriels des',carq-1:4,' noeuds choisis');
for m:=carq-1 downto 1 do begin res:=0;
  CLAS:=cdq[m+carq];AINE:=cdq[qAm[m]];BNJM:=cdq[qBm[m]];
  PDS:=round(1000*poim[m]);
  for a:=1 to carf do res:=res+sqr(-Fqda^[m+carq]^a));
  dm[m]:=res;
  for a:=carf+1 to cara do dm[m]:=dm[m]+sqr(-Fqda^[m+carq]^a));
  if (dm[m]<2e-9) then dm[m]:=0;
  INR:=round(1000*poim[m]*dm[m]/trac);QLT:=0;
  if not (dm[m]=0) then QLT:=round(1000*res/dm[m]);
  writeln(' ',CLAS:4,AINE:5,BNJM:5,' ');
  write(ft,' ',CLAS:4,AINE:5,BNJM:5,' ',QLT:4,PDS:4,INR:4,' ');
  for a:=a1 to a2 do begin
    res:=Fqda^[m+carq]^a;FCT:=round(1000*res);CO2:=0;
    if not (dm[m]=0) then CO2:=round(1000*sqr(res)/dm[m]);
    CTR:=round(1000*sqr(res)*poim[m]/lama[a]);
    write(ft,FCT:5,CO2:4,CTR:4,' ') end;writeln(ft) end;
```

On notera que, dans {CLAS, AINE, BNJM}, les numéros sont ceux de la CAH générale, lus dans le tableau cdq (cf. *supra*, §2.2). La qualité: QLT, est le rapport du carré de la distance, calculé dans l'espace engendré par les carf axes du listage, au carré de la distance: dm[m], calculé dans l'espace (de dimension: cara) où a été faite la CAH. Ce même: dm[m], est un multiplicateur dans le calcul de INR; qui est, plus exactement, un taux, rapporté à l'inertie totale, trac: calculée avec cara axes; soit pour E (si chf='i', cf. §2.2), soit pour le nuage des centres des classes (chf='a'). Et dm[m] est le dénominateur pour les CO2 afférents à chaque axe.

2.3.3 Écriture de la deuxième tranche d'une bande: les classes

```
writeln(ft,'repr sur les axes fact des',carq:4,' classes de la partition
choisie');
for q:=1 to carq do begin
  c:=cdq[q];write(ft,' ',c:4);n:=c-care;res:=0;writeln(' ',c:4);
  if (0<n) then write(ft,cAq[q]:5,cBq[q]:5) else write(ft,' ');
  PDS:=round(1000*poi[q]);
  for a:=1 to carf do res:=res+sqr(-Fqda^[q]^a));dq[q]:=res;
  for a:=carf+1 to cara do dq[q]:=dq[q]+sqr(-Fqda^[q]^a));
  if (dq[q]<2e-9) then dq[q]:=0;
  INR:=round(1000*poi[q]*dq[q]/trac);QLT:=0;
  if not (dq[q]=0) then QLT:=round(1000*res/dq[q]);
  write(ft,' ',QLT:4,PDS:4,INR:4,' ');
  for a:=a1 to a2 do begin res:=Fqda^[q]^a;FCT:=round(1000*res);CO2:=0;
    if not (dq[q]=0) then CO2:=round(1000*sqr(res)/dq[q]);
    CTR:=round(1000*sqr(res)*poi[q]/lama[a]);
    write(ft,FCT:5,CO2:4,CTR:4,' ') end;writeln(ft) end;
```

Le calcul est tout analogue à celui fait pour les nœuds.

On rappelle que, sur l'ensemble des classes, le poids: PDS, a pour total 1000 millièmes. De plus, le PDS d'un nœud est la somme de ceux de son aîné et de son benjamin. Mais, pour INR ou CTR, le total des termes afférents à l'aîné et au benjamin dépasse, en général, le terme propre au nœud: la différence, se retrouve dans la tranche des dipôles (cf. *infra*).

2.3.4 Écriture de la troisième tranche d'une bande: les dipôles

```

for a:=1 to 29+(14*carb) do write(ft,' '); writeln(ft);
for a:=1 to 29+(14*carb) do write(ft,' '); writeln(ft);
write(ft,'|CDIP AINE BNJM| QLD PDS IND|');
for a:=a1 to a2 do write(ft,' D',a:2,' COD CTD|'); writeln(ft);
for a:=1 to 29+(14*carb) do write(ft,' '); writeln(ft);
writeln(ft,'repr sur les axes factoriels des',carb-1:4,' dipôles choisis');
for m:=carb-1 downto 1 do begin
  CLAS:=cdq[m+carb]; AINE:=cdq[qAm[m]]; BNJM:=cdq[qBm[m]];
  PDS:=round(1000*poim[m]);
  qa:=qAm[m]; if (carb<qa) then za:=0 else za:=1;
  qb:=qBm[m]; if (carb<qB) then zb:=0 else zb:=1;
  ppp:=pzq(za,qa)+pzq(zb,qb); res:=0;
  if not (ppp=0) then ppp:=pzq(za,qa)*pzq(zb,qb)/ppp;
  for a:=1 to carf do res:=res+sqr(Fqda^[qa]^a]-Fqda^[qb]^a]);
  dd[m]:=res;
  for a:=carf+1 to cara do dd[m]:=dd[m]+sqr(Fqda^[qa]^a]-Fqda^[qb]^a]);
  if (dd[m]<2e-9) then dd[m]:=0;
  INR:=round(1000*ppp*dd[m]/trac); QLT:=0;
  if not (dd[m]=0) then QLT:=round(1000*res/dd[m]);
  writeln('||',CLAS:4,AINE:5,BNJM:5,'|');
  write(ft,'||',CLAS:4,AINE:5,BNJM:5,'|',QLT:4,PDS:4,INR:4,'|');
  for a:=a1 to a2 do begin
    res:=Fqda^[qa]^a]-Fqda^[qb]^a]; FCT:=round(1000*res); CO2:=0;
    if not (dd[m]=0) then CO2:=round(1000*sqr(res)/dd[m]);
    CTR:=round(1000*sqr(res)*ppp/lama[a]);
    write(ft,FCT:5,CO2:4,CTR:4,'|') end;writeln(ft) end;
  for a:=1 to 29+(14*carb) do write(ft,' ');writeln(ft);
  if (a2=carf) then al:=0;
  if (a2<carf) then begin al:=a2+1;
    if (carf<(al+10)) then a2:=carf else a2:=al+9 end;
  END;
close(ft);

```

Ainsi qu'on l'a annoncé au §2.3.1, les sigles usités pour la tranche des dipôles diffèrent quelque peu de ceux communs aux nœuds et aux classes. Cette distinction est d'autant plus utile que les dipôles ont mêmes numéros que les nœuds; ceux-là ne se distinguant de ceux-ci que quant aux calculs effectués. On retrouve la distinction entre espace engendré par les carf premiers axes; et espace ambiant. La qualité: QLD, est le rapport du carré de la longueur du bras du dipôle, calculé dans l'espace engendré par les carf axes du listage, au carré: dd[m], calculé dans l'espace (de dimension cara) où a été faite la CAH.

Prenons des exemples dans le listage. L'inertie, INR, afférente au nœud 24, est: 376. Pour l'ainé et le benjamin, {13, 21}, les INR sont: {279, 209}. La différence: ((279+209)-376)=112, est la valeur IND du dipôle 24. De même, dans le bloc de colonnes afférent à l'axe 2, on a des CTR pour {24, 13, 21} qui sont {176, 648, 6}; le CTD, pour 24, est: 479 $\approx$ ((648+6)-176); avec une différence de 1 e-3 imputable aux erreurs d'arrondi. En effet, appelons, en bref: point q, le centre gravité d'une classe q, compté avec sa masse. L'inertie (INR ou CTR) du système des deux points {13, 21} relativement à l'origine peut se décomposer en: inertie (IND ou CTD) de ce système relativement à son centre 24, et inertie (INR ou CTR) du point 24 relativement à l'origine.

2.4 Fin de la procédure 'Facx': résultats transmis à 'Facw'

```
writeln('le listage Facor est créé');
for q:=1 to carq do begin Fqda^[q]^0:=cdq[q];
  Fqda^[q]^amax+1:=poi[q]; Fqda^[q]^amax+2:=dq[q] end;
for m:=1 to carq-1 do begin Fqda^[m+carq]^0:=cdq[m+carq];
  Fqda^[m+carq]^amax+1:=poim[m];
  Fqda^[m+carq]^amax+2:=dm[m] end;
dispose(qde);
end;end.
```

Finalement (ainsi qu'on l'a annoncé au §2.2), par les pointeurs $Fqda^q$, on accède aux tableaux $Fqda^q$, dont chacun contient (pour une classe ou nœud donné: q): son numéro: $Fqda^q[0]$, dans la CAH générale de E ; la suite des facteurs: $Fqda^q[1..carq]$; et le poids et le carré de la distance à l'origine: $Fqda^q[amax+1]$, $Fqda^q[amax+2]$. Ces informations, passent à 'Facw' par le pointeur pG . Tandis que par pF , on accède à l'en-tête destiné aux fichiers 'Fac.w', dûment adaptée par l'instruction: $ptt^{[titab]}:=carq$.

3 Création des fichiers de facteurs pour l'ensemble des classes et l'ensemble des nœuds: la procédure 'Facw'

3.1 Déclarations en tête de la procédure 'Facw'

```
UNIT uFacw5;
INTERFACE uses memtypes, quickdraw, osintf, toolintf, uver5;
procedure Facw(nomen:str255; pF, pG:ptr);
implementation procedure Facw;
const qdmax=200;
type
  cab=array[0..amax] of single; pcab=^cab;
  ptqda=array[1..qdmax] of pcab; ppqda=^ptqda;
var
  fin:file of integer;
  che:char; siq:zngle; nomf:str255;
  a,c,cw,amax,carq,m,q,i0:integer;
  Fqda:ppqda; ptt:ptni; sss:single; ps2,ps3:ptni;
procedure codi; begin siq[0]:=1; siq[1]:=ord(che);
  if (999<i0) then siq[1]:=48+((i0 div 1000) mod 10);
  if (99<i0) then begin siq[0]:=2;
    siq[2]:=48+((i0 div 100) mod 10); end;
  if (9<i0) then begin siq[0]:=siq[0]+1;
    siq[siq[0]]:=48+((i0 div 10) mod 10); end;
  siq[0]:=siq[0]+1; siq[siq[0]]:=48+(i0 mod 10);
end;
```

La plupart des constantes, types et variables déclarés en tête de 'Facw', ont déjà été vus dans 'Facx'.

La procédure 'codi' a pour effet de composer un sigle pour une classe, d'après le numéro de celle-ci, $i0$; et le dernier caractère, che , du nom de l'ensemble à traiter: par exemple si ce nom est 'sa', $che='a'$. Si le numéro est inférieur à 1000, le sigle est formé du caractère: che , suivi du numéro; si le numéro est ≥ 1000 , le caractère che n'est pas pris; et le numéro est écrit avec 4 chiffres. Si le nombre des milliers dépasse 9, on prend dans le code ASCII, un caractère après le 9: ':' pour 10, ';' pour 11... Le maximum est 15, parce qu'avec $card(E) \leq ipmax=8000$ (cf. IA§1.1), on a: $card(E) < 16000$.

3.2 Initialisation des pointeurs

```
begin
ptt:=ptni (pF);Fqda:=ppqda (pG);
carq:=ptt^[titab];amax:=ptt^[titab+1]-2;
ps3:=ptni (@siq);ps2:=ptni (@sss);
che:=nomen[length (nomen)];
```

Ainsi qu'on l'a dit au §2.4, 'Facw', par les pointeurs {pF, pG}, accède aux informations préparées par 'Facx'. Les pointeurs ps3 et ps2 servent, respectivement, à lire un sigle comme une suite de trois entiers; ou un nombre réel comme une suite de deux entiers.

3.3 Écriture des fichiers de facteurs

```
nomf:=concat (nomen, 'qFac.w');rewrite (fin,nomf);
for c:=1 to titab-1 do write (fin,ptt^[c]);
write (fin,ptt^[titab]);write (fin,ptt^[titab+1]);cw:=titab+1;
for c:=cw+1 to cw+(3*(2+amax)) do write (fin,ptt^[c]);
for q:=1 to carq do begin
i0:=round (0+Fqda^[q]^0);codi;
write (fin,ps3^[1],ps3^[2],ps3^[3]);
for a:=1 to amax+2 do begin
sss:=Fqda^[q]^a];write (fin,ps2^[1],ps2^[2]) end;
writeln (sigler (siq)) end;
close (fin);writeln ('le fichier ',nomf,' est créé');
```

```
nomf:=concat (nomen, 'nFac.w');rewrite (fin,nomf);
for c:=1 to titab-1 do write (fin,ptt^[c]);
write (fin,ptt^[titab]-1);write (fin,ptt^[titab+1]);cw:=titab+1;
for c:=cw+1 to cw+(3*(2+amax)) do write (fin,ptt^[c]);
for m:=carq-1 downto 1 do begin
i0:=round (0+Fqda^[m+carq]^0);codi;
write (fin,ps3^[1],ps3^[2],ps3^[3]);
for a:=1 to amax+2 do begin
sss:=Fqda^[m+carq]^a];write (fin,ps2^[1],ps2^[2]) end;
writeln (sigler (siq)) end;
```

```
close (fin);writeln ('le fichier ',nomf,' est créé');
dispose (ptt);
for c:=1 to qdmax do dispose (Fqda^[c]);dispose (Fqda);
end;end.
```

L'en-tête a été reçu de 'Facx', par le pointeur pF; et elle est lue, grâce à ptt, comme une suite d'entiers. On prendra garde qu'initialement: carq=ptt^[titab], ce qui convient pour le fichier 'qFac.w'; mais pour le fichier 'nFac.w', afférent aux nœuds, cette valeur doit être diminuée de 1. Le numéro, i0 (dans Eé), d'une classe ou d'un nœud de la hiérarchie est d'abord lu comme un réel 'single'; mis en 'extended' par addition de zéro; converti en un entier par 'round' et, transformé en un sigle par la procédure 'codi' (cf. §3.1). Les valeurs numériques réelles sont lues par Fqda^.

La procédure 'Facw' s'achève en libérant les zones de mémoire centrale assignées à des pointeurs.