

Institut National Polytechnique de Lorraine

Ecole Nationale Supérieure
de Géologie

Centre de Recherches
Pétrographiques et Géochimiques

INDICATRICES FLOUES APPLIQUEES
A LA RECONNAISSANCE AUTOMATIQUE
DES DIAGRAPHIES

THESE

Présentée pour l'obtention du
titre de DOCTEUR-INGENIEUR

*Soutenue publiquement le 5 octobre 1985
par*

Jean-Marie EPITALON



Devant le Jury :

Président	J.P. HATON
Rapporteur	J.L. MALLET
.....	F. DE BEAUCOURT
.....	C. CHAMBON
.....	J.J. ROYER



D 136 003050 2

1360030502

Institut National Polytechnique de Lorraine

Ecole Nationale Supérieure
de Géologie

Centre de Recherches
Pétrographiques et Géochimiques

[M] 1985 EPITALON, J.-M

INDICATRICES FLOUES APPLIQUEES
A LA RECONNAISSANCE AUTOMATIQUE
DES DIAGRAPHIES

T H E S E

Présentée pour l'obtention du
titre de DOCTEUR-INGENIEUR

Soutenue publiquement le 5 octobre 1985
par

Jean-Marie EPITALON



Devant le Jury :

Président	J.P. HATON
Rapporteur	J.L. MALLET
Examineurs	F. DE BEAUCOURT
	C. CHAMBON
	J.J. ROYER

REMERCIEMENTS

Je remercie en premier lieu mon professeur, M. Mallet qui a dirigé ma thèse. Me proposant un sujet intéressant, me guidant dans les moments difficiles, ne ménageant ni les critiques, ni les encouragements, il m'a permis de travailler dans d'excellentes conditions et m'a communiqué sa passion pour l'informatique.

Je tiens à remercier également la Société Nationale Elf-Aquitaine qui est à l'origine du projet et qui a participé à son développement. Elle m'a par ailleurs fourni les données numériques indispensables à la mise au point du programme informatique créé et mis à ma disposition l'imprimante qui a édité ce texte.

Que Messieurs Haton et Chambon qui me font l'honneur de juger ce mémoire soient assurés de ma reconnaissance.

Je n'oublierai pas non plus M. de Beaucourt qui s'est intéressé d'une façon particulière à mon travail, m'apportant parfois un soutien moral indéniable.

MM. Royer et Jacquemin m'ont prodigué leur aide et leurs conseils, tant dans le domaine de l'informatique que dans le domaine de l'analyse des données, qu'ils en soient vivement remerciés.

Je dois également de grands remerciements à Mme Volle qui a eu la lourde tâche de taper et d'arranger ce texte et à M. Gaize qui en a dessiné toutes les figures.

Enfin, je remercie toutes les personnes qui m'ont permis de travailler dans une ambiance sympathique, en particulier Mme Bkouche, (ex. Mlle Palen), M. Danis, M. Isnard, M. Dreux, Mme Marchal et Mme Sarre.

SOMMAIRE

I)	PRESENTATION DE LA COMMANDE.....	1
	1) Objectif final de la commande DIA.....	1
	2) Bref aperçu des méthodes employées.....	1
	3) Présentation des résultats.....	6
	4) Exemple d'utilisation.....	7
II)	L'ASPECT INFORMATIQUE.....	11
	1) Introduction.....	11
	2) Nature et structure des données manipulées.....	11
	3) Les règles de la conversation.....	18
	4) La convivialité.....	19
	5) Architecture de la commande.....	20
III)	LA RELATION HOMME-MACHINE : L'AIDE A LA DECISION DANS LA CONSTRUCTION D'UN MODELE.....	23
	1) La présentation numérique du noyau courant.....	23
	2) Les éditions graphiques.....	24
	3) La cross-validation.....	28
	4) Possibilités de développement.....	29
IV)	BASES MATHÉMATIQUES DE LA DISCRIMINATION.....	30
	1) Approximation d'une densité.....	30
	2) Indicatrices floues.....	33
	3) Analyse discriminante.....	35
V)	MISE EN OEUVRE DU CALCUL DES PROBABILITES.....	37
	1) Rappels.....	37
	2) Application.....	38
	3) Forme des courbes.....	39
	4) Différence entre la théorie et la mise en oeuvre.....	42
	5) Intérêt de la deuxième formule.....	45
	6) Limite de l'influence des poids.....	45
	7) Ordre de grandeur des entités calculées.....	47
	8) Le calcul de la fenêtre C.....	48
VI)	L'ALGORITHME DES CENTRES DYNAMIQUES.....	57
	1) Description de l'algorithme.....	57
	2) Utilisation qui est faite de cet algorithme.....	58
	3) Le nombre de centres initiaux.....	60
	4) Le nombre d'itérations.....	60
	5) La métrique.....	60
VII)	LES ESSAIS ET LES APPLICATIONS.....	64
	1) Test de la méthode des centres dynamiques.....	64
	2) Les premières données réelles : le puits S1.....	68
	3) Le deuxième ensemble de données réelles : le puits S2.....	70
	4) Application aux diagraphies dans des séries salifères.....	73

VIII)	ANALYSE DIAGRAPHIQUE.....	81
	1) Introduction.....	81
	2) Résumé des opérations à effectuer.....	81
	3) Liste des sous-commandes abordées dans cette section.....	81
	4) Chargement des données.....	82
	5) Chargement d'un modèle.....	82
	6) Définition des composantes.....	83
	7) Définition d'un sous-ensemble de niveau du puits.....	85
	8) L'analyse faciologique.....	86
	9) le lissage des résultats.....	87
	10) La visualisation des résultats.....	88
	11) La sauvegarde des résultats.....	90
	12) Conclusion	93
	13) Exemple concret d'analyse diagraphique.....	94
IX)	CONSTRUCTION D'UN MODELE.....	97
	1) Introduction.....	97
	2) Résumé des opérations à effectuer.....	97
	3) Liste des sous-commandes abordées dans cette section.....	97
	4) Chargement des données.....	98
	5) Définition des composantes.....	98
	6) Définition d'un sous-ensemble de niveaux du puits.....	98
	7) Vérifications préliminaires et définition du nom du nouveau GDT-model.....	98
	8) Création des noyaux.....	99
	9) Attribution d'un nom de faciès aux noyaux.....	101
	10) Correction du modèle.....	110
	11) Sauvegarde du modèle.....	111
	12) Exemple de construction de modèle.....	111
X)	EDITEUR DE MODELES.....	114
	1) La syntaxe des instructions et la notion de noyau courant.....	114
	2) Entrée dans la sous-commande.....	115
	3) Les composantes de travail.....	115
	4) Edition des noyaux : Les instructions LIST, TITLE et DISP.....	115
	5) La cross-validation : Les options CV, CVR, LCV, LCVR, COEF.....	117
	6) Modifications d'un modèle : Les instructions CHANGE, DEL, LITO.....	118
	7) L'instruction SAVE ou S.....	119
XI)	SOUS-COMMANDES DE DIA.....	120
	1) La sous-commande WELLDEF.....	120
	2) La sous-commande SETDEF.....	121
	3) La sous-commande MODDEF.....	122

4)	Les sous- commandes WELLDIR et MODDIR.....	123
5)	La sous-commande TOOLS.....	125
6)	La sous-commande EDTO.....	131
7)	La sous-commande EXTWELL.....	132
8)	La sous-commande CBLD.....	133
9)	La sous-commande CDEL.....	135
10)	La sous-commande SETBLD.....	135
11)	La sous-commande MODBLD.....	137
12)	La sous-commande ANADIA.....	142
13)	La sous-commande RUN.....	143
14)	La sous-commande SMTH.....	144
15)	La sous-commande VISU.....	145
16)	La sous-commande INCLUD.....	148
17)	La sous-commande EDMOD.....	154
18)	La sous-commande WELLSAV.....	158
19)	La sous-commande MODSAV.....	160
20)	Les sous-commandes CROSSV et CROSSVR.....	161

AVANT-PROPOS

Le but de cette thèse est la reconnaissance automatique des diagraphies. La méthode utilisée est une discrimination de données numériques basée sur les indicatrices disjonctives floues. (cf. MALLET J.-L., Propositions for fuzzy characteristic functions in data analysis, In : COMPSTAT 1982 (Physica Verlag, WIEN, 1982).

Ces indicatrices floues, inspirées de la notion des noyaux de Parzen-Rozenblatt (cf. {2} et {3}), sont liées à des points caractéristiques obtenus au moyen de l'algorithme des centres dynamiques.

Du point de vue pratique, nous avons construit un programme appliquant les méthodes décrites ci-dessus. Ce programme appelé "commande DIA" fait partie du système GEOL (cf. {12} {13} et {14}) qui contient déjà plusieurs autres commandes correspondant chacune à une application particulière.

La commande DIA a donc une structure informatique typique des commandes du système GEOL dont elle utilise en particulier la notion de tableau de données GEOL. La relation homme-machine de ce programme a été définie en concertation avec les utilisateurs potentiels de la Société Nationale Elf-Aquitaine.

Par ailleurs, cette société nous a fourni les données de plusieurs sondages diagraphiques enregistrées lors de forages pétroliers. C'est sur ces données que nous avons testé notre programme.

INTRODUCTION

1) Définitions géologiques

- lithofaciès des roches :

D'après la définition de MOORE en 1949, le lithofaciès est la somme des caractéristiques physiques (par exemple : la texture, la porosité, etc.) chimiques et biologiques d'une roche.

Le sous-sol, dans les séries sédimentaires, est constitué de couches de différents lithofaciès. Ces couches ne sont pas nécessairement horizontales, ni d'épaisseur constante.

- un puits de sondage :

C'est un trou foré dans le sous-sol à partir de la surface. Il est le plus souvent vertical. Le sondage peut cependant être oblique ou dévier de la verticale.

2 types de sondages :

. les sondages carottés qui remontent des cylindres de roches de diamètre égal au trou de sondage. En raison de leur coût, ce type de forage n'est que rarement utilisé tout le long du puits.

. les sondages percutants qui brisent la roche et la remontent sous forme de fragments appelés "cuttings", mélangés avec la boue de forage injectée.

- colonne lithologique :

C'est la succession des lithofaciès du sous-sol le long d'un axe vertical. Par extension, c'est la succession des lithofaciès des roches traversées par un puits de sondage.

2) Détermination de la colonne lithologique d'un sondage

Une première information sur la lithologie est apportée par les cuttings du forage. Elle n'est qu'imparfaite du fait des turbulences de la boue de forage lors de sa remontée, du fait des éboulements des parois du puits et des dissolutions de certaines roches (sels, silts).

Depuis l'invention des diagraphies par les frères Schlumberger en 1927, la connaissance des roches du sous-sol se fait essentiellement à partir des diagraphies.

- diagraphie : (terme anglais : "log").

D'après O. Serra, une diagraphie est un enregistrement continu en fonction de la profondeur des variations d'une caractéristique donnée des roches traversées par un sondage.

Il est admis que les diagraphies peuvent donner une bonne idée de la lithologie du sous-sol. Cette évidence a été à l'origine du concept d'électrofaciès défini par O. Serra, comme l'ensemble des mesures diagraphiques qui caractérisent une couche et permettent de la distinguer des autres couches.

Il existe donc des correspondances dans un contexte géologique donné entre le lithofaciès et l'électrofaciès.

Par exemple, dans des séries sédimentaires, une zone présentant une forte résistivité, une faible radioactivité et une grande vitesse du son est en général composée de calcaires compacts ou de grès quartzite.

- La reconnaissance des diagraphies consiste à utiliser cette particularité pour établir la colonne lithologique d'un sondage dans lequel ont été effectuées des diagraphies. Jusqu'à présent, ceci est fait "à la main". Des géologues spécialisés dans ce domaine établissent la colonne lithologique d'un sondage en examinant les courbes diagraphiques obtenues sur ce sondage, disposées les unes

à côté des autres sur un même document (cf. figure ci-dessous)

Depuis peu, grâce au développement de l'informatique, des méthodes de reconnaissance automatique ont vu le jour, notamment chez les Sociétés Schlumberger et Compagnie Française des Pétroles.

Nous nous sommes attachés à créer un logiciel de reconnaissance automatique des diagraphies basé sur l'utilisation des indicatrices floues.

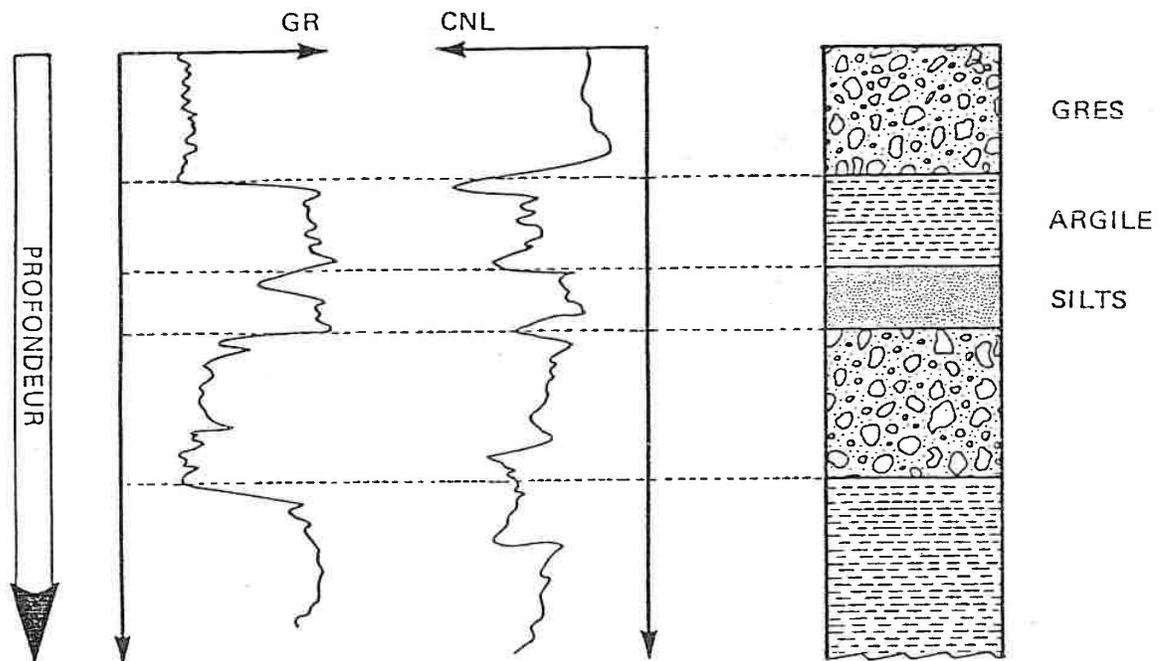


Figure 1. Sondages diagraphiques et leur interprétation lithologique

1) APPROCHE DE LA METHODE EMPLOYEE

Le système créé suit les deux phases de la reconnaissance manuelle des diagraphies :

1re phase : Avec l'aide de l'utilisateur, le système mémorise les lithofaciès présents sur un sondage et leurs caractéristiques diagraphiques, constituant ainsi une base de connaissances.

2e phase : Le système applique ces connaissances au sondage étudié et éventuellement à d'autres sondages comportant les mêmes faciès.

1) L'idée de base : la partition de l'espace des variables

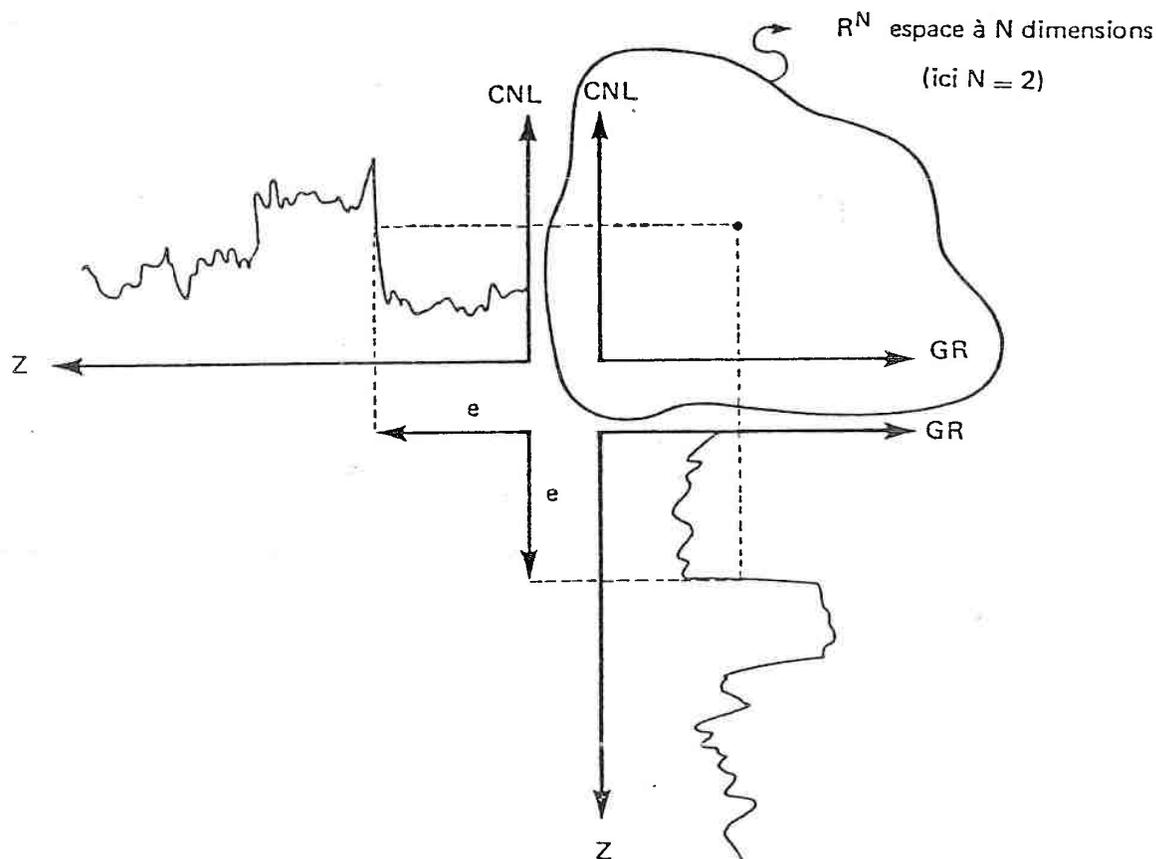


Figure 2. Représentation d'un point de données dans l'espace des variables

A tout échantillon de mesure, pris à un niveau quelconque du sondage étudié, comportant N mesures de natures différentes, on peut associer un point dans un espace à N dimensions. On appelle cet espace, l'espace paramétrique ou l'espace des variables.

Remarque : Cet espace ne peut s'imaginer que si $N < 3$. Dans le cas où $N = 2$, cet espace est un plan, et la représentation des points dans ce plan s'appelle un cross-plot.

Un électrofaciès est alors matérialisé par une zone de cet espace. (Ces zones sont des surfaces, dans le cas où $N = 2$, et des volumes, dans le cas où $N = 3$).

Mémoriser un électrofaciès revient à mémoriser la zone qu'il occupe dans l'espace des variables.

Ex. Dans le cas d'un cross-plot GR CNL

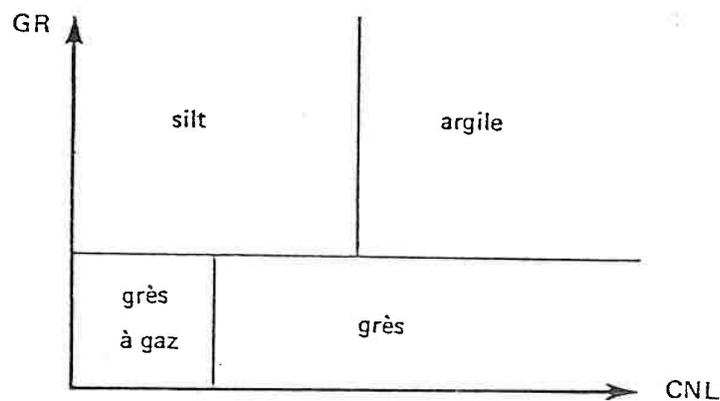


Figure 3. Séparation d'un faciès dans l'espace des variables. Cas d'un cross-plot GR-CNL

2) Notion de partition floue

Comme le suggère la figure 4, dans la méthode employée, chaque zone est mémorisée à l'aide de points qui la décrivent grossièrement :

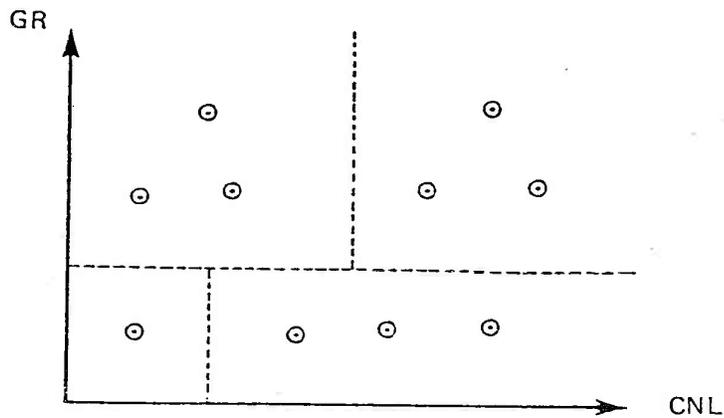


Figure 4

Ces points sont appelés "noyaux" et correspondent à peu près aux pôles utilisés par les diagraphistes dans les cross-plots. Chaque noyau porte donc un nom de faciès. L'ensemble des noyaux forme ce que l'on appellera un "modèle".

Pour un point quelconque de l'espace, on calcule sa probabilité d'appartenance à chaque zone en fonction de l'éloignement relatif de ce point aux noyaux de la zone concernée.

Les résultats bruts de l'analyse sont donc, pour chaque niveau du puits, un ensemble de probabilités, que ce faciès appartienne à un faciès ou à un autre. Dans le cas général, le faciès le plus probable a une probabilité très forte et ce faciès est pour ainsi dire certain.

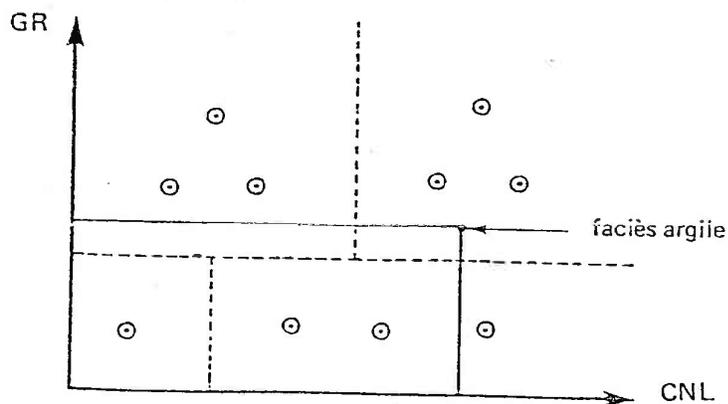


Figure 5. Attribution d'un faciès à un point de l'espace des variables.

3) Présentation des résultats

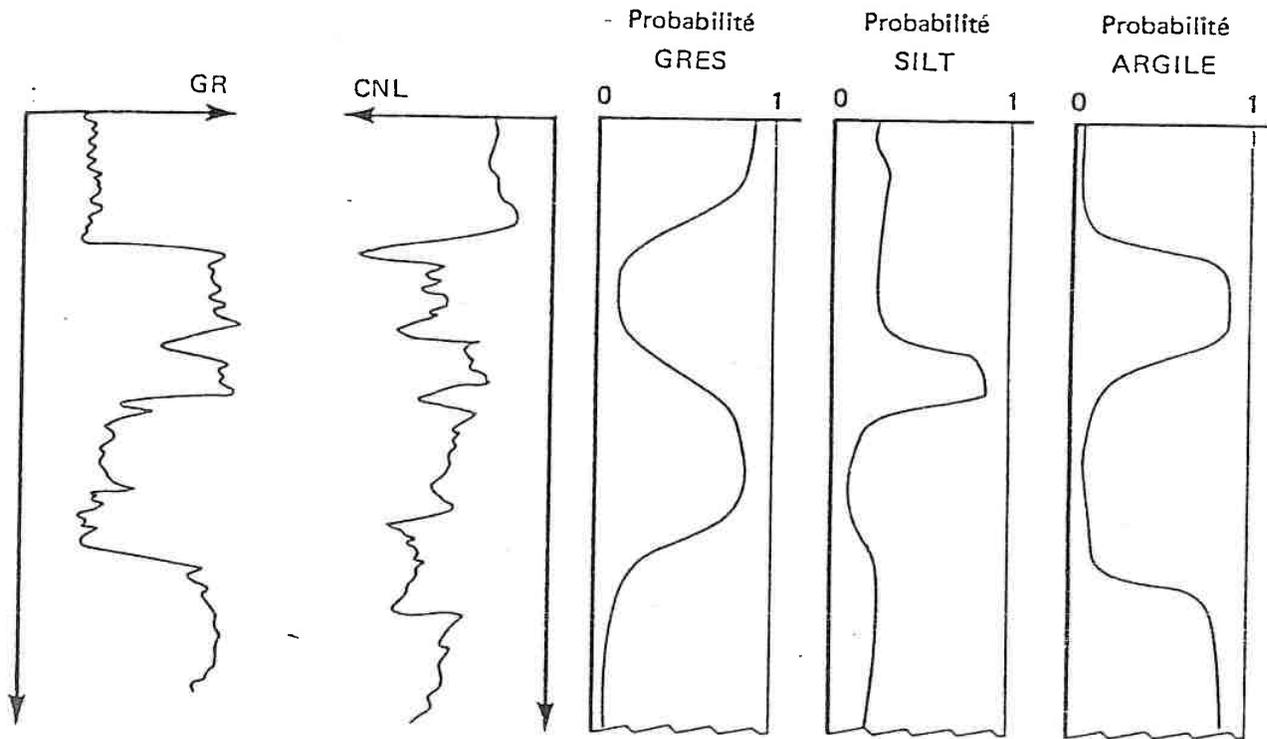


Figure 6. Résultats bruts de l'analyse

On peut envisager de faire une colonne lithologique prenant en compte ces probabilités comme des pourcentages de composition. cf. figure suivante

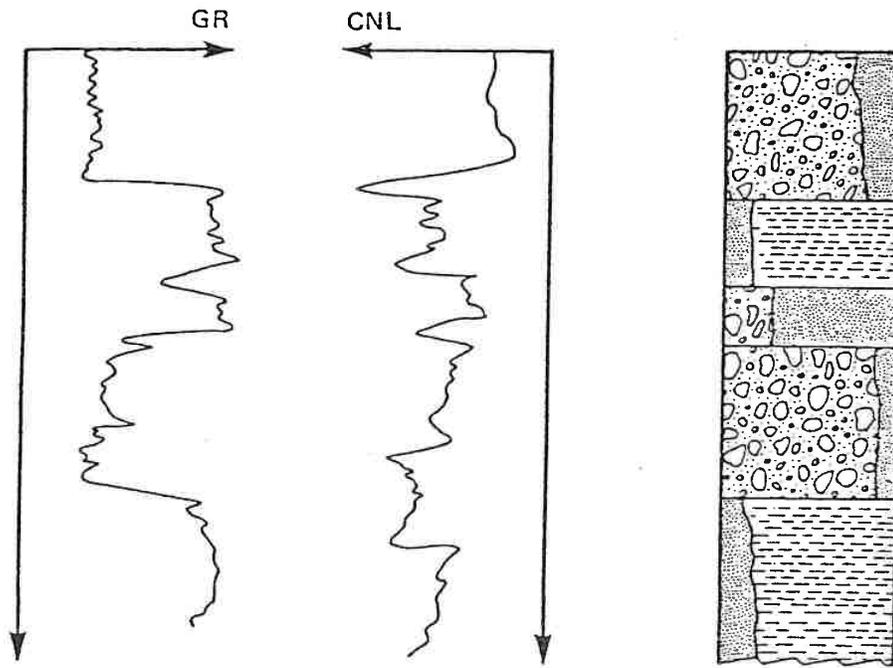
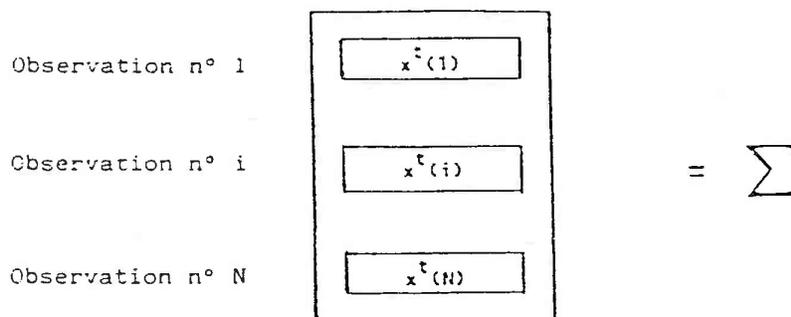


Figure 7. Exemple d'interprétation à deux faciès

II) BASES MATHÉMATIQUES DE LA DISCRIMINATION

1) APPROXIMATION D'UNE DENSITÉ

Soit X une V.A. à valeurs dans \mathbb{R}^n et admettant une densité de probabilité f . Soit d'autre part $\{x = \{x(1), \dots, x(N)\}$ un N -échantillon de X que l'on peut assimiler à un tableau de données.



Nous nous proposons de trouver une fonction f approchant f_x avec pour objectif d'utiliser celle-ci dans un algorithme d'analyse discriminante basé sur la notion d'indicatrice floue. (cf [1])

a) Ensemble D_n

Soit D_n l'ensemble des V.A. à valeurs dans \mathbb{R}^n et ayant une densité $f(x)$ satisfaisant les trois conditions suivantes :

1) $f(x)$ est de la forme suivante ...

$$f(x) = \sum_{s \in S} p(s) \Delta(x|s, C) \quad \forall x \in \mathbb{R}^n$$

ou $S = \{s_1, \dots, s_v\}$ est un ensemble fini de points appartenant à \mathbb{R}^n . On dit que S est le "support" de f , que C est sa "fenêtre" et que les points $s \in S$ sont des points étalons.

2) Chaque fonction $\Delta(x|s, C)$ est une densité de probabilité ayant s pour moyenne, C pour matrice de covariance et telle que :

$$0 \leq \Delta(x|s, C) \leq \Delta(s|s, C) < +\infty \quad \forall x \in \mathbb{R}^n$$

$$\lim_{\|x\| \rightarrow \infty} \Delta(x|s, C) = 0$$

$$\|x\| \rightarrow \infty$$

3) Les coefficients p_i sont positifs ou nuls.

On peut remarquer que les trois conditions réunies entraînent que

$$\sum p_i = 1$$

b) Propriété

Toute variable aléatoire à valeurs dans R^n peut être approchée par une V.A. appartenant à D^n . Du point de vue pratique, trois problèmes se présentent :

- 1) Quel est le support $S = \{s_1, s_2, \dots, s_v\}$ à prendre en compte pour positionner les noyaux ?
- 2) Quelle est la matrice C (appelée "fenêtre") à utiliser ?
- 3) Quels sont les poids p_i ?

Nous proposons dans ce qui suit une solution informatique efficace pour ces deux problèmes.

c) Choix du support S et des coefficients (p_i) associés

a) Méthode des noyaux de Parzen-Rosenblatt

Dans la méthode des noyaux de Parzen-Rosenblatt, on a l'habitude de prendre pour S le N-échantillon Σ de la V.A. X étudiée :

$$S \equiv \Sigma \equiv \{x(1), \dots, x(N)\}$$

Les coefficients p_i associés sont alors pris tous égaux à $1/N$

$$p(s) = 1/N \quad \forall s \in S$$

Du point de vue informatique, si N est grand, cette façon de procéder peut conduire à des calculs longs pour $f(x)$ et pas forcément très précis (à cause des erreurs d'arrondi). A ce sujet, on vérifie facilement que le volume V de calculs nécessaires pour calculer $f(x)$ est du type

$$v = k \cdot \text{Card}(S) \cdot n$$

avec k = coefficient de proportionnalité
et n = dimension de l'espace des valeurs de X

Pour cette raison, nous avons été conduits à proposer une variante de la méthode de Parzen-Rosenblatt visant à déduire de Σ un support S tel que

$$\text{Card}(S) \ll \text{Card}(\Sigma)$$

B) Méthode proposée

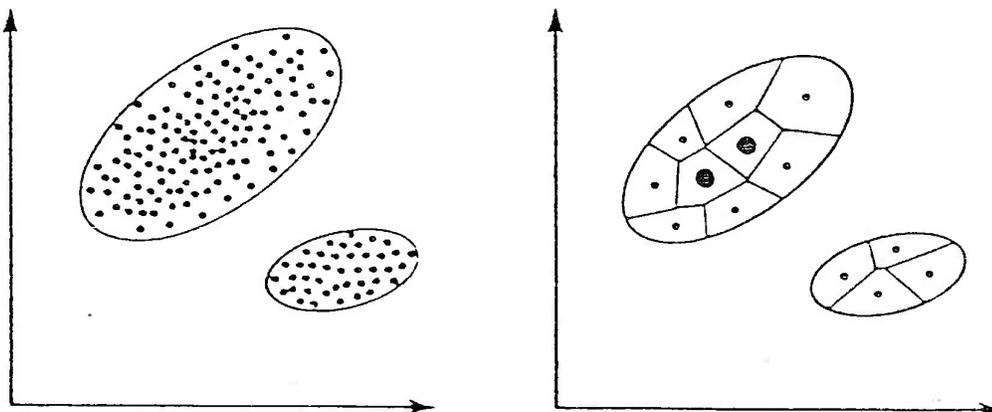
Supposons donc que N soit très élevé (par ex : $N > 300$).

Comme le suggère la figure 8, nous proposons de procéder de la façon suivante :

- 1) On se donne v (très inférieur à N)
- 2) A partir de v centres initiaux pris au hasard dans Σ et à l'aide de la méthode des centres dynamiques, on partitionne l'ensemble des points $\{x(i)\}$ de Σ en v classes $\{\Sigma_j\}$ de centres $\{s_j\}$ et l'on pose :

$$S = \{s_1, s_2, \dots, s_v\}$$

$$p(s_j) = \frac{\text{Card}(\Sigma_j)}{\text{Card}(\Sigma)}$$



a) ensemble E

b) ensemble S : la taille des points s_j est proportionnelle à $p(s_j)$

Figure 8 : Obtention du support S à partir de l'ensemble en utilisant la méthode des centres dynamiques.

Cette façon de procéder donne des résultats tout à fait satisfaisants dans la mesure où l'on a pris la précaution de centrer et réduire les N données E . Cette réduction préalable est nécessaire car elle permet de supprimer les éventuelles distorsions qui apparaissent lorsque les composantes des vecteurs $x(i) \in E$ ont des ordres de grandeur très différents.

d) Choix de la fenêtre C

a) Méthodes analytiques

Le choix de C est un problème délicat auquel de nombreux auteurs se sont attachés. (cf. {2}, {3}, {4}).

Les critères de choix sont généralement analytiques (par ex. : critère IMSE, cf. {4}) et sont acceptables lorsque la variable parente X est scalaire ($n = 1$), mais sont souvent instables lorsque X est vectorielle ($n > 1$). Après avoir effectué de nombreux essais dans ce sens (critère IMSE), nous nous sommes orientés vers une technique purement informatique que nous jugeons tout à fait stable et satisfaisante pour les applications pratiques.

B) Méthode proposée

Pour chaque point $s_j \in s$, on recherche son plus proche voisin $s \in S$, et l'on calcule la distance d_j entre ces deux points.

$$d_j = \min_{\substack{s \in S \\ s \neq s_j}} d(s_j, s)$$

On calcule alors la valeur moyenne σ des distances d_j :

$$\sigma = \frac{1}{v} \sum_{j=1}^v d_j$$

... et l'on pose : $C = \sigma \cdot I$

avec I = matrice unité d'ordre n

Evidemment, comme pour le choix de S , cette façon de procéder n'a d'intérêt que si les valeurs $s_j \in S$ ont été obtenues à partir de données Σ centrées réduites. Si nécessaire, on peut introduire un coefficient de lissage λ en posant :

$$C = C_\lambda = \lambda \cdot \sigma \cdot I$$

En général, cette opération n'est pas nécessaire, mais lorsqu'on l'utilise, elle conduit à une fonction $f(x)$ d'autant plus lisse que λ est élevé, d'où le nom attribué à λ .

γ) Accélération du calcul de d

Pour chaque point $s_j \in S$, la recherche de d_j nécessite le calcul des $(v-1)$ distances entre s_j et les autres points de S et la recherche de la valeur minimale de ces $(v-1)$ distances.

Pour accélérer le calcul de cette distance, on partitionne d'une façon grossière l'ensemble S en m classes $\{S_1, S_2, \dots, S_m\}$ par le moyen de l'algorithme des centres dynamiques et l'on pose :

$$d^*j = \min_{\substack{s \in S_k \\ s \neq s_j}} d(s_j, s)$$

S_k = classe contenant s_j

La plupart du temps, on constate que l'on a

$$d^*j = d_j$$

En conséquence, nous proposons de poser :

$$\sigma = \frac{1}{v} \sum_{j=1}^v d^*j$$

2) INDICATRICES FLOUES

a) Rappel

Pour un ensemble A appartenant à R^n , on appelle "indicatrice floue" de A toute fonction $\phi_A(\cdot)$ telle que :

$$0 \leq \phi_A(x) \leq 1 \quad \forall x \in R^n$$

Généralement, on choisit $\phi_A(\cdot)$ de telle façon que la valeur $\phi_A(x)$ puisse être interprétée comme un indice de proximité de x avec l'ensemble de référence A . (cf. Kaufmann 1971)

b) Indicatrice floue (disjonctive) de $s \in S$

Soit X une V.A. à valeurs dans R^n et admettant une densité du type suivant:

$$f(x) = \int_{s \in S} p(s) \Delta(x|s, C)$$

Nous poserons : $\tilde{R}^n = \{x \in R^n : f(x) \neq 0\}$

Dans une précédente publication (cf. {1}), nous avons introduit la notion d'indicatrice floue disjonctive I_s associée à chaque point s de S en posant :

$$I_s(x) = p(s) : \frac{\Delta(x|s,C)}{f(x)} \quad \forall x \in \tilde{R}$$

c) Indicatrice floue (disjonctive) d'une partition de S

Soit $\{G_k\}$ une partition de S en K classes

$$S = G_1 \cup G_2 \cup \dots \cup G_K$$

avec $G_i \cap G_j = \emptyset \quad \forall i \neq j$

Nous appellerons indicatrice floue (disjonctive) de G_k et nous noterons $I_{G_k}(\cdot)$ la fonction telle que :

$$I_{G_k}(x) = \sum_{s \in G_k} I_s(x) \quad \forall x \in \tilde{R}^n$$

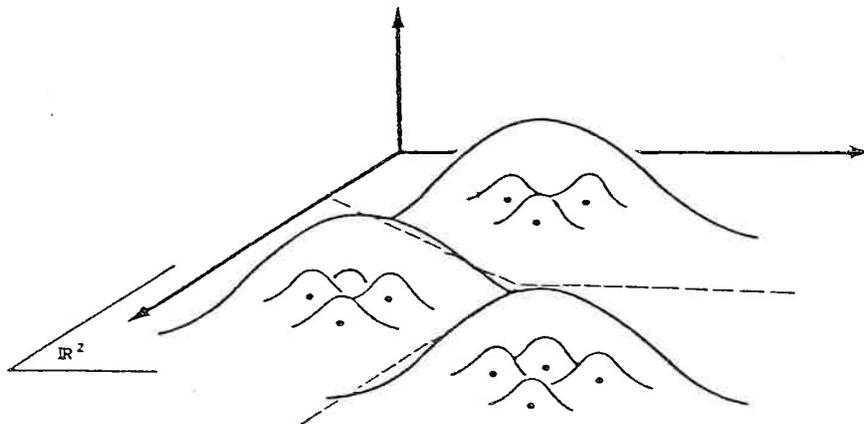


Figure 9 : partition de l'espace.

Comme le suggère la figure 9, les indicatrices floues $I_{G_k}(\cdot)$ induisent une partition de l'espace \tilde{R}^n en K classes G_k telles que :

$$x \in G_k \iff I_{G_k}(x) \geq I_{G_h}(x) \quad \forall h \neq k$$

Ces indicatrices $I_{G_k}(\cdot)$ sont elles-mêmes disjonctives car on a :

$$\sum_{k=1}^K I_{G_k}(x) = 1 \quad \forall x \in \tilde{R}^n$$

On peut interpréter $I_{G_k}(x)$ comme la probabilité de x d'appartenir à la classe G_k .

Remarque : on peut poser

$$\tilde{G}_k = \{x \in \tilde{R}^n : I_{G_k}(x) \geq I_{G_h}(x) \quad \forall h \neq k\} \quad \text{si } k \geq 1$$

$$\tilde{G}_0 = \tilde{R}^n - \tilde{R}^n = \{x \in \tilde{R}^n : f(x) = 0\}$$

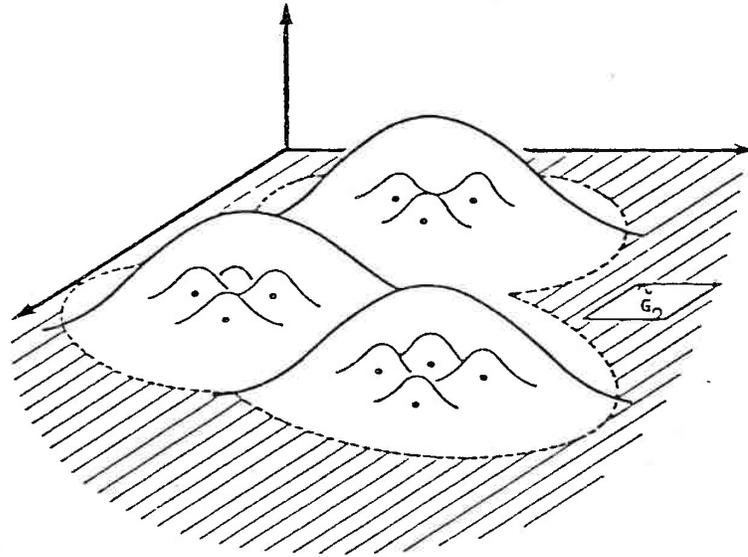


Figure 10

Dans ces conditions, $\{\tilde{G}_0, \tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_k\}$ constitue une partition de l'espace R_n en $(K+1)$ classes, la classe \tilde{G}_0 jouant le rôle d'une classe vide de points. Dans la pratique, on se fixe un seuil ϵ et comme le suggère la figure 10, on pose :

$$\tilde{G}_0 = \{x \in R^n : f(x) \leq \epsilon\}$$

3) ANALYSE DISCRIMINANTE

a) Notion de modèle

α) Définition

Nous appellerons "modèle", toute partition de S en K classes $\{G_k\}$.

β) Construction d'un modèle

La construction d'un modèle est réalisée de façon conversationnelle: après avoir déterminé les points s à partir de Σ , l'ordinateur les propose l'un après l'autre à un expert humain afin que celui-ci les classe dans des groupes homogènes $\{G_1, G_2, \dots, G_k\}$. Ces groupes constituent les classes d'une partition de S en K classes.

Comme le suggère la figure 11, un modèle est constitué par :

- un ensemble S de points s appartenant à R^n
- un ensemble de poids $p(s)$
- un ensemble de K groupes ($G(s) = k$ si $s \in G_k$)

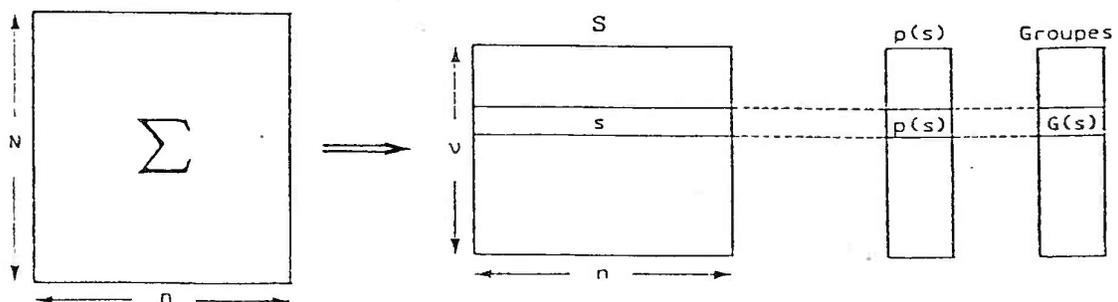


Figure 11

γ) Cross-validation d'un modèle

Etant donné un modèle $\{G_k\}$, pour tout point s appartenant au support S , on peut calculer les probabilités d'appartenance $I_{G_k}(s)$ de s à chaque classe G_k . Si le modèle possède une certaine "cohérence" interne, alors on doit avoir pour chaque classe G_k :

$$I_{G_k}(s) \geq I_{G_h}(s) \quad \forall h \neq k \quad \forall s \in G_k$$

S'il n'en est pas ainsi pour un point $s_j \in G_k$, il y a peut-être lieu de reconsidérer le choix de l'affectation de s_j au groupe G_k lors de la construction du modèle.

δ) Fusion de deux modèles

Etant donnés deux modèles obtenus à partir de deux N -échantillons Σ_1 et Σ_2 , on peut songer à construire un modèle synthétique en posant :

$$S = S_1 \cup S_2$$

$$p(s) = \begin{cases} \alpha \cdot p_1(s) & \dots \text{ si } s \in S_1 \\ (1-\alpha) p_2(s) & \dots \text{ si } s \in S_2 \end{cases}$$

avec $\alpha \in [0,1]$

b) Application à l'analyse discriminante

Soit Σ un tableau de données comportant N individus sur chacun desquels on a mesuré n variables. On suppose que l'on a construit un modèle à partir d'un tableau Σ^* comportant des individus sur lesquels on a mesuré n^* variables. On se propose de classer les N individus de Σ dans les K classes G_k définies par le modèle dont on dispose. Dans la pratique, on peut rajouter la $(K+1)$ ième classe constituée par \tilde{G}_0 .

Pour discriminer les éléments $x(i)$ du tableau Σ , on utilise la règle de décision suivante :

$$x(i) \in \tilde{G}_k \iff I_{G_k}(x(i)) \geq I_{G_h}(x(i)) \quad \forall h \neq k$$

On considère alors que :

$$I_{G_k}(x(i)) = \text{probabilité d'appartenance de } x(i) \text{ à } G_k.$$

Si le modèle contient les variables (X_1, \dots, X_{n^*}) et le tableau de données Σ contient les variables (X_1, \dots, X_n) avec $n < n^*$, alors on extrait un nouveau modèle à partir du premier en ne gardant que les variables (X_1, \dots, X_n) et l'on applique la règle de décision décrite plus haut sur la base des n variables de Σ .

III) MISE EN OEUVRE DU CALCUL DES PROBABILITES

1) Rappels

indicatrice floue d'un point $s \in S$

$$I_S(x) = p(s) \times \frac{\Delta(x|s,C)}{\sum_{s \in S} p(s) \Delta(x|s,C)} \quad \forall x \in \mathbb{R}^n$$

indicatrice floue d'une partie G_k de S

$$I_{G_k}(x) = \sum_{s \in G_k} I_S(x) \quad \forall x \in \mathbb{R}^n$$

$$\Leftrightarrow I_{G_k}(x) = \frac{\sum_{s \in G_k} p(s) \Delta(x|s,C)}{\sum_{s \in S} p(s) \Delta(x|s,C)}$$

discrimination

$$\begin{aligned} & I_{G_k}(x) \geq I_{G_h}(x) \quad \forall h \neq k \\ \sim_{x \in G_k} & \Leftrightarrow \text{et} \\ & \sum_{s \in S} p(s) \Delta(x|s,C) \geq \epsilon \end{aligned}$$

2) Application

Les fonctions $\Delta(x|s,C)$ choisies sont des gaussiennes centrées au point s , de fenêtre C et appliquées à la donnée vectorielle x réduite. (cf. IV 8)

Dans la version finale du programme, nous utilisons une gaussienne isotrope dont la formule est :

$$\Delta(x|s,C) = \exp \left\{ \frac{-d^2(x,s)}{2c} \right\}$$

ou c est une valeur scalaire

et $p(s)$ = poids du noyau attribué lors de sa construction

$d(x,s)$ = distance réduite entre les points x et s

$$d^2(x,s) = \sum_{i=1}^n \frac{(x_i - s_i)^2}{\sigma_i^2}$$

n = nombre de variables

x = vecteur de coordonnées (x_i)

s = vecteur de coordonnées (s_i)

σ_i = écart-type de l'ensemble des points $s \in S$

pour la variable V_i

$d^2(x,s)$ s'écrit aussi sous la forme

$$d^2(x,s) = (x-s)^t \Sigma^2 (x-s)$$

avec $\Sigma = \begin{vmatrix} 1/\sigma_1 & & \dots & 0 & 0 \\ \cdot & 1/\sigma_2 & & & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & & & & \cdot \\ 0 & 0 & \dots & & 1/\sigma_n \end{vmatrix}$

= matrice diagonale
contenant les écarts-types

On appelle D_k la densité du groupe G_k

$$D_k(x) = \frac{\sum_{s \in G_k} p(s) \Delta(x|s, C)}{\sum_{s \in G_k} p(s)}$$

et l'on pose

$$I_{G_k}(x) = \frac{D_k(x)}{\sum_{j=1}^k D_j(x)}$$

$$I_{G_k}(x) \geq I_{G_h}(x) \quad \forall h \neq k$$

$\tilde{x} \in G_k \iff$ et

$$\sum_{j=1}^k D_j(x) \geq \epsilon$$

3) Forme des courbes

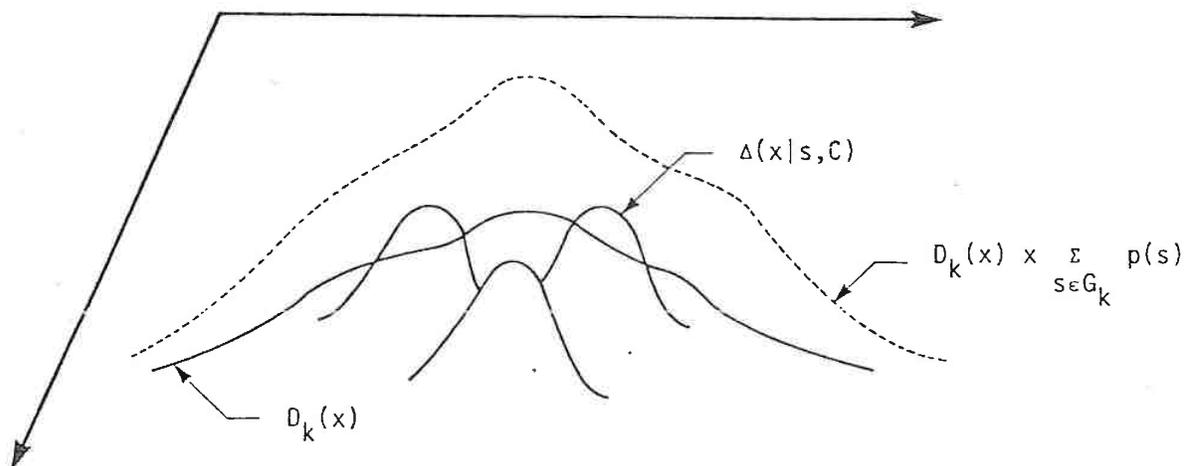


Figure 12. Forme de la courbe $D_k(x)$ dans un espace à deux dimensions

Indicatrice floue I_{G_k} :

Exemple avec 2 classes dans un espace à 1 dimension :

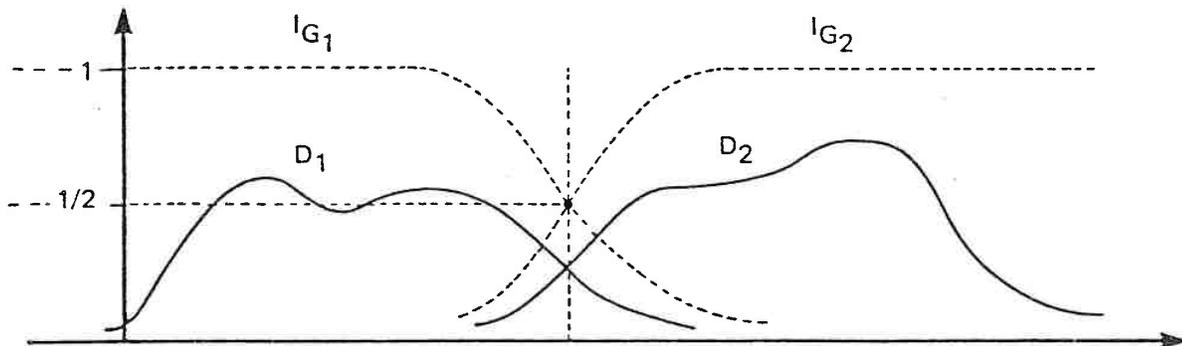


Figure 13. Forme des indicatrices floues dans un espace à une dimension

Le problème des points à l'infini :

Comme le suggère la figure 14, si l'espace est partagé totalement entre les classes, alors, même loin de toute classe, il existe des probabilités non nulles et souvent une probabilité très forte pour une classe.

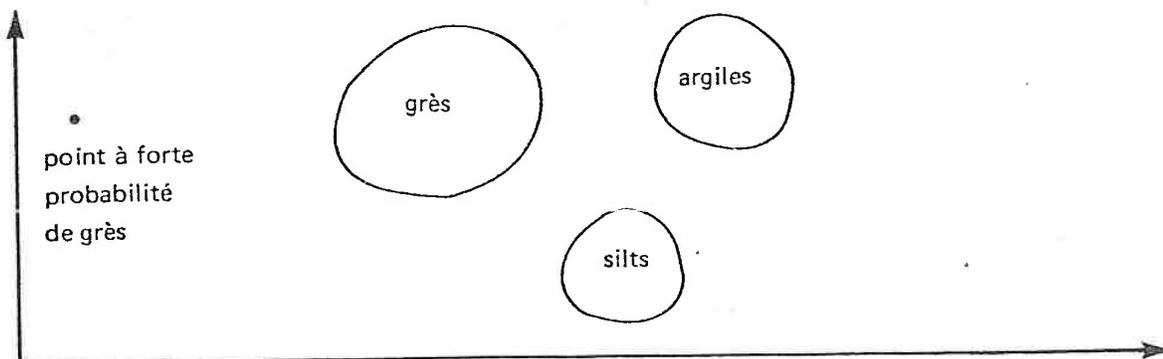


Figure 14

C'est pourquoi on décide de créer une classe de plus, la classe \tilde{G}_0 où toutes les probabilités seront égales :

$$\tilde{G}_0 = \{x \in \mathbb{R}^n : f(x) < \epsilon\}$$

$$\text{avec } f(x) = \sum_{k=1}^K D_k(x)$$

et K = nombre de classes de la partition de S

En fait, nous nous servons d'une particularité de l'ordinateur. Celui-ci ne sait pas stocker une valeur décimale positive plus petite que 10^{-38} (machine à mots de 32 bits). En dessous de cette valeur, il retient la valeur 0. Il suffit pour nous de tester si

$$f(x) = \sum_{k=1}^K D_k(x) \text{ égale } 0 \text{ ou non.}$$

Influence de ceci sur la forme des courbes

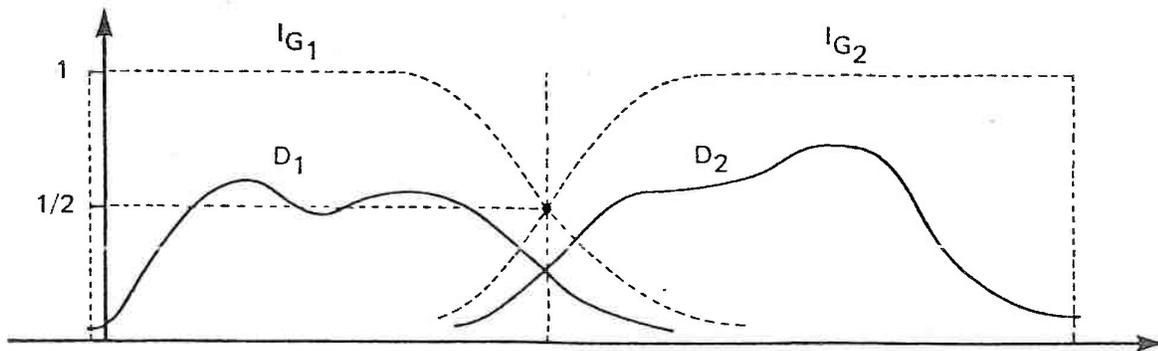


Figure 15. Forme des indicatrices floues après rajout de la classe G_0

4) Différence entre la théorie et la mise en oeuvre

Théorie

$$I_{G_k}(x) = \frac{\sum_{s \in G_k} p(s) \Delta(x|s,C)}{\sum_{s \in S} p(s) \Delta(x|s,C)}$$

$$I_{G_k}(x) = \frac{\sum_{s \in G_k} p(s) \Delta(x|s,C)}{K \sum_{j=1}^K \sum_{s \in G_j} p(s) \Delta(x|s,C)}$$

$$(1) \quad I_{G_k}(x) = \frac{P_k D_k(x)}{\sum_{j=1}^K P_j D_j(x)}$$

avec P_k = poids de la classe G_k

$$P_k = \sum_{s \in G_k} p(s)$$

Dans la mise en oeuvre :

$$(2) \quad I_{G_k}(x) = \frac{D_k(x)}{\sum_{j=1}^K D_j(x)}$$

Dans le cas de la formule (2), le poids des classes n'est pas pris en compte.

Exemple : deux classes dans un espace à 1 dimension.

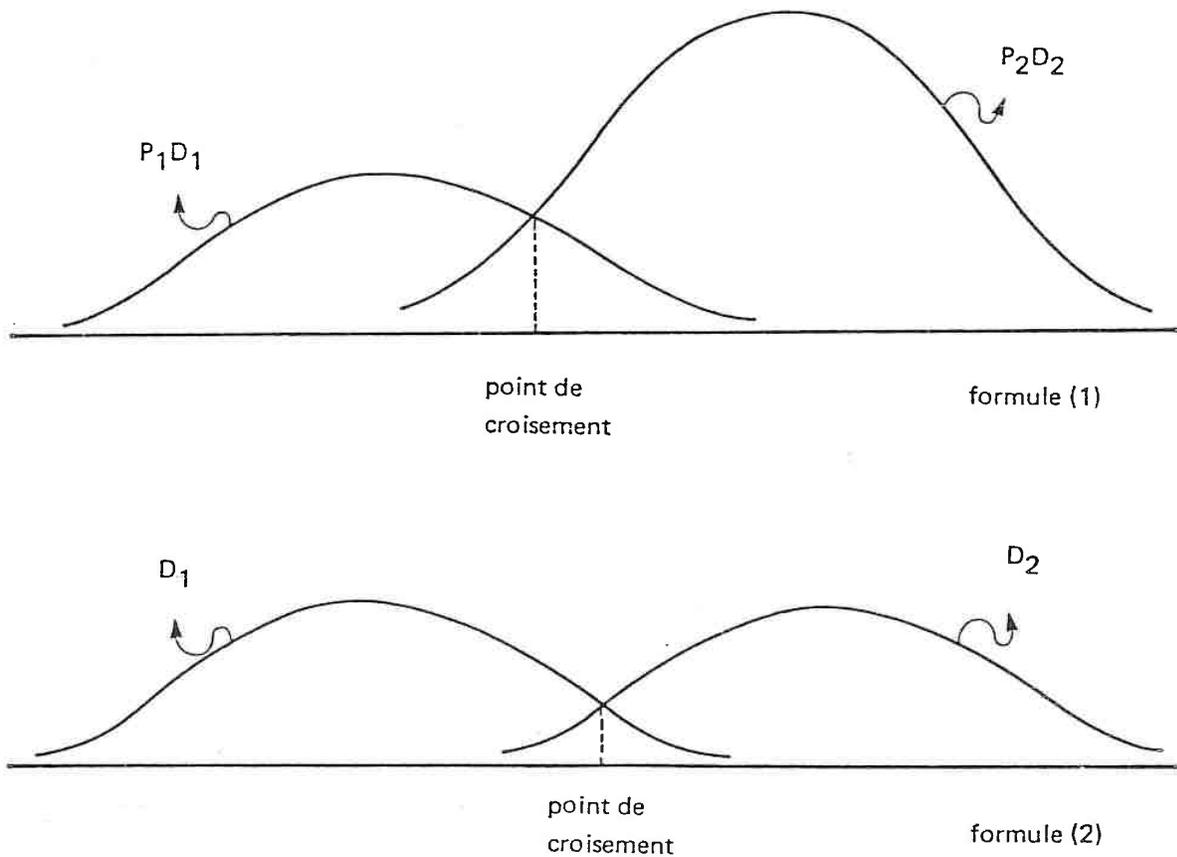


Figure 16. Influence des deux formules sur la forme des courbes

On voit que dans le premier cas, la classe ayant le plus fort poids a une influence prépondérante.

Le point de croisement qui marque la limite $I_{G_1} = I_{G_2}$ est déplacé et le domaine où $I_{G_2} > I_{G_1}$ augmente.

Dans le cas de la formule (1)

$$I_{G_k}(x) = \frac{\sum_{s \in G_k} p(s) \Delta(x|s,C)}{\sum_{j=1}^{\kappa} \sum_{s \in G_j} p(s) \Delta(x|s,C)}$$

Dans le cas de la formule (2)

$$I_{G_k}(x) = \frac{\sum_{s \in G_k} \frac{p(s)}{p_k} \Delta(x|s,C)}{\sum_{j=1}^{\kappa} \sum_{s \in G_j} \frac{p(s)}{p_j} \Delta(x|s,C)}$$

Dans le cas de la formule (2), le poids de noyaux est divisé par le poids de la classe à laquelle ils appartiennent. Les poids des noyaux ne jouent donc un rôle qu'à l'intérieur d'une même classe, ce qui peut être intéressant pour obtenir une courbe D_k en forme de "cloche".

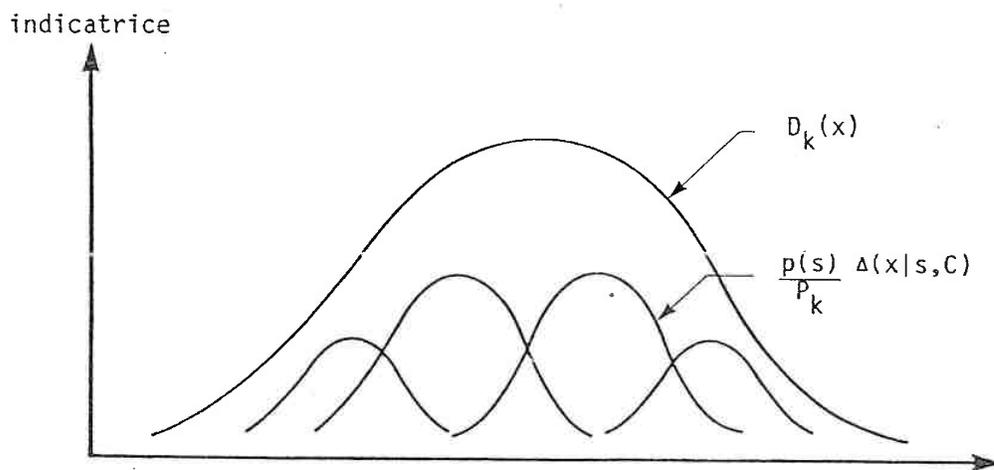


Figure 17. Exemple de courbe $D_k(x)$ en forme de "cloche"

5) Intéret de la deuxième formule

La deuxième formule rétablit l'équilibre entre deux classes ayant des poids différents.

En effet, il se peut que des classes remplissant le même volume dans l'espace des variables aient des poids très différents à la fin de la construction du modèle. Il suffit pour cela que le puits dont on a tiré le modèle comprenne une grande épaisseur de formations d'un même faciès et une très faible épaisseur de formations d'un autre faciès.

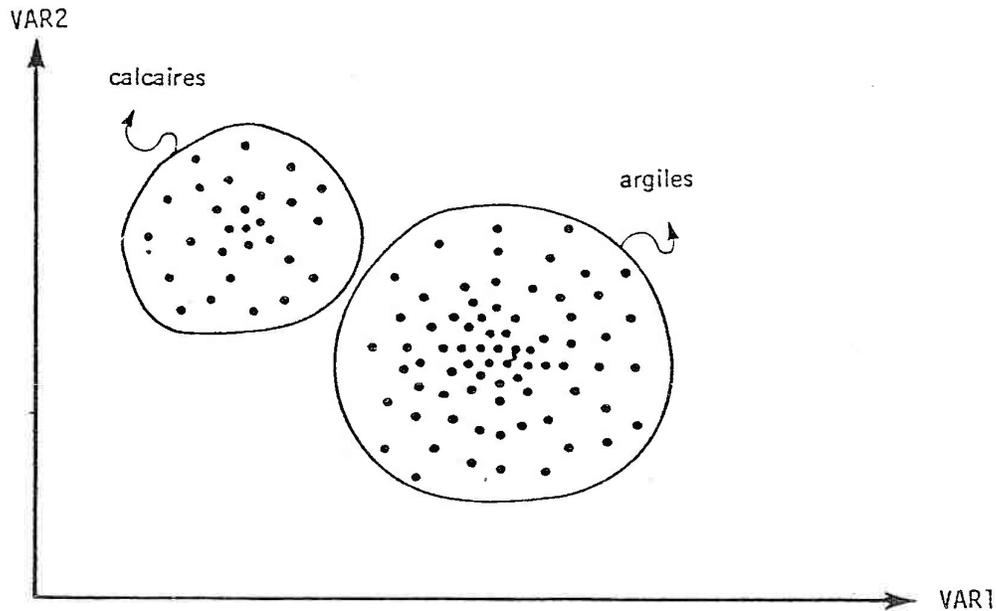


Figure 18. Cross-plot des données d'un puits traversant beaucoup de séries argileuses et peu de séries calcaires.

Dans cet exemple, le nombre de noyaux obtenus dans les argiles sera beaucoup plus grand que le nombre de noyaux obtenus dans les calcaires. Le poids de la classe "argiles" sera beaucoup plus grand que le poids de la classe "calcaire".

6) Limite de l'influence des poids

$$\Delta(x|s,C) = \exp \left\{ - \frac{d^2(x,s)}{2c} \right\}$$

Ces courbes étant exponentielles, elles décroissent très vite.
 Les multiplier par un poids n'accroît souvent pas beaucoup leur aire
 d'influence.

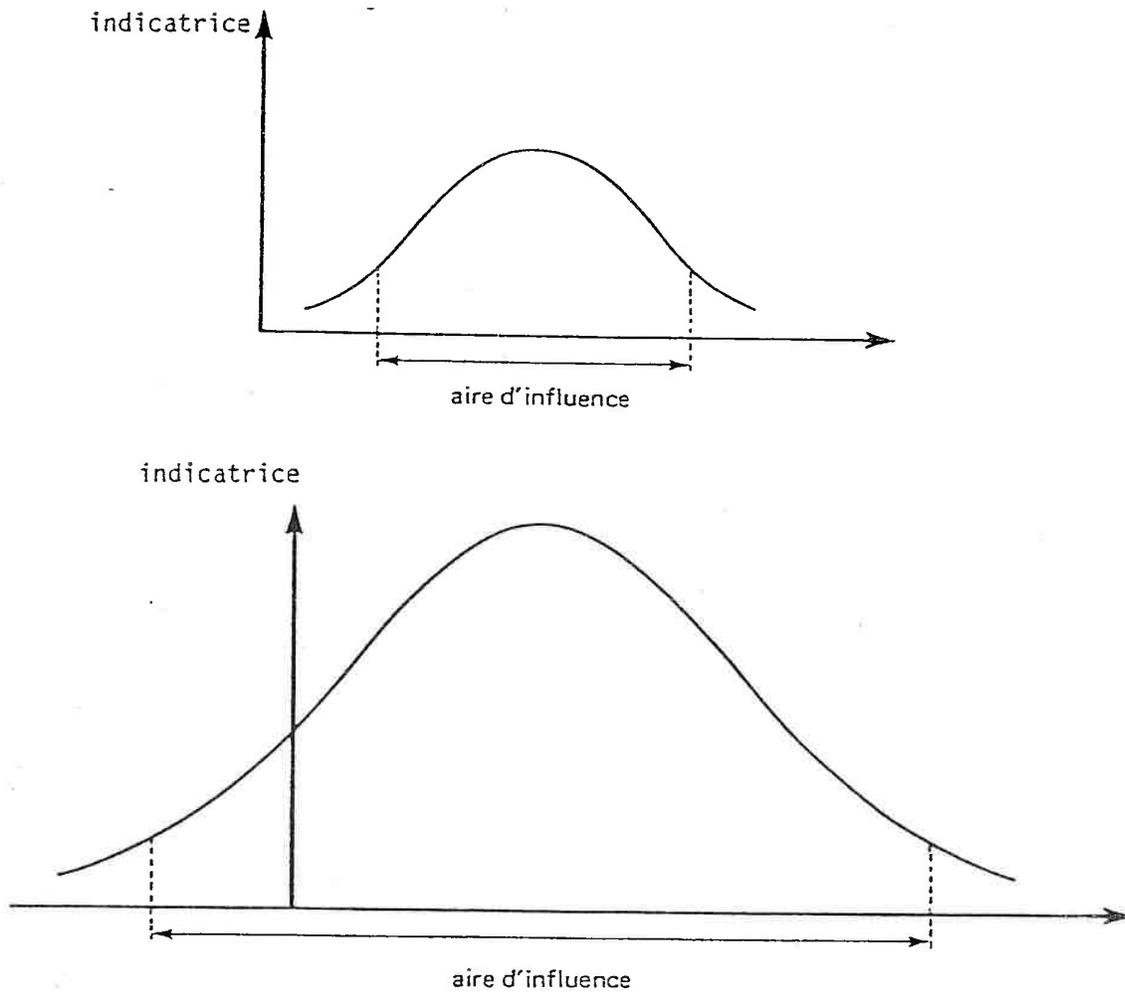


Figure 19. Influence du poids affecté à une indicatrice

Si $1/2c = 2$, alors un poids de valeur 1 000 appliqué à $\lambda(x|s,C)$ produit la même variation qu'une diminution d'environ 3,5 pour $d^2(x,s)$.

$$1\ 000 \times \exp\left\{-\frac{d^2(x,s)}{2c}\right\} = \exp\left\{-\frac{d^2(x,s) + \alpha}{2c}\right\}$$

$$\Leftrightarrow 1\ 000 = \exp\left\{-\frac{\alpha}{2c}\right\}$$

$$\Leftrightarrow \frac{\alpha}{2c} = 7$$

Les données étant réduites, les distances $d^2(x,s)$ sont de l'ordre de 10 à 40 suivant le nombre de variables. Une variation de 3,5 est donc atteinte très facilement.

indiatrice

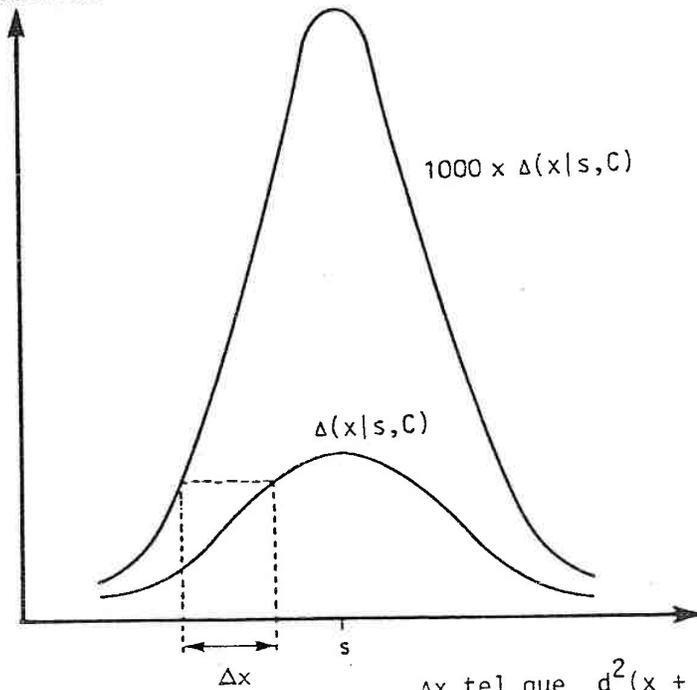


Figure 20

7) Ordre de grandeur des entités calculées

$d^2(x,s)$ est une distance entre données réduites. Elle augmente avec le nombre de variables.

Quand le nombre de variables varie entre 2 et 10,
 $d^2(x,s)$ varie en moyenne entre 10 et 40,
 $1/2c$ varie en moyenne entre 0.5 et 2,
 $\Delta(x|s,C)$ varie en moyenne entre 10^{-10} et 10^{-38} .

Plus c est grand, plus les fonctions $\Delta(x|s,C)$ ont de grandes valeurs et plus ces valeurs sont proches les unes des autres. Les probabilités sont alors proches les unes des autres.

Au contraire, plus c est petit, plus les fonctions $\Delta(x|s,C)$ ont de petites valeurs, plus elles ont des ordres de grandeur différents et plus les probabilités sont tranchées.

Ex. : 0.99 0.008 0.001 0.001

Dans ce cas, les poids affectés aux noyaux ne jouent plus aucun rôle.

Conclusion : Les poids des noyaux jouent un rôle minime, mais l'étude du problème est intéressante. Si on affectait un poids de 1 à tous les noyaux, le poids d'une classe serait égal au nombre de noyaux. Alors en utilisant la formule qui tient compte du poids des classes (formule (2)), on obtiendrait des résultats encore valables.

8) Le calcul de la fenêtre C

Les données étant réduites au préalable, la formule générale de calcul d'une gaussienne anisotrope est :

$$\Delta(x|s,C) = \exp \left\{ -\frac{1}{2}(x-s)^t T C^{-1} T(x-s) \right\}$$

où T est une matrice diagonale contenant les inverses des écarts-types sur chaque variable.

$$T = \begin{vmatrix} 1/\sigma_1 & & \dots & 0 & 0 \\ \cdot & 1/\sigma_2 & & & 0 \\ \cdot & & & \cdot & \\ \cdot & & & \cdot & \\ 0 & & & \cdot & \\ 0 & 0 & \dots & & 1/\sigma_n \end{vmatrix}$$

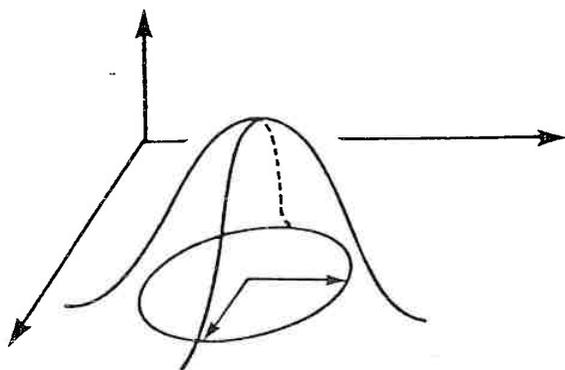
et où C est une matrice carrée d'ordre n (n = dimension de l'espace des variables). C est appelée "fenêtre" de la gaussienne.

Si C est une matrice isotrope,

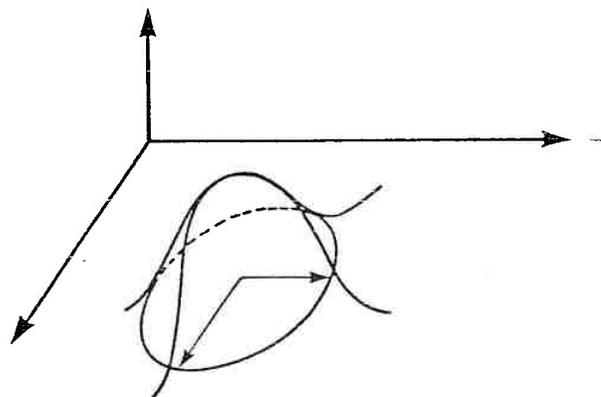
$$C = \begin{vmatrix} c & & \dots & 0 & 0 \\ \cdot & c & & & 0 \\ \cdot & & & \cdot & \\ \cdot & & & \cdot & \\ 0 & & & c & \cdot \\ 0 & 0 & \dots & & c \end{vmatrix}$$

ou c est une valeur scalaire

alors $\Delta(x|s,c) = \exp \left\{ \frac{-d^2(x,s)}{2c} \right\}$



formule isotrope



formule anisotrope

Figure 21. Indicatrices isotropes et anisotropes

a) La méthode du MISE

MISE veut dire "Mean Integrated Square Error". C'est une méthode statistique appliquée à l'estimation de la fonction densité d'une variable aléatoire par la méthode des noyaux de Parzen-Rosenblatt. Elle calcule la fenêtre C des indicatrices centrées sur les noyaux.
cf. {3} et {4}

b) La première version du programme

Dans une première version, nous avons appliqué la méthode du MISE à tous les points du support S , ce qui donnait aux gaussiennes (les fonctions $\Delta(x|s,C)$) une orientation similaire à l'orientation du nuage total des points.

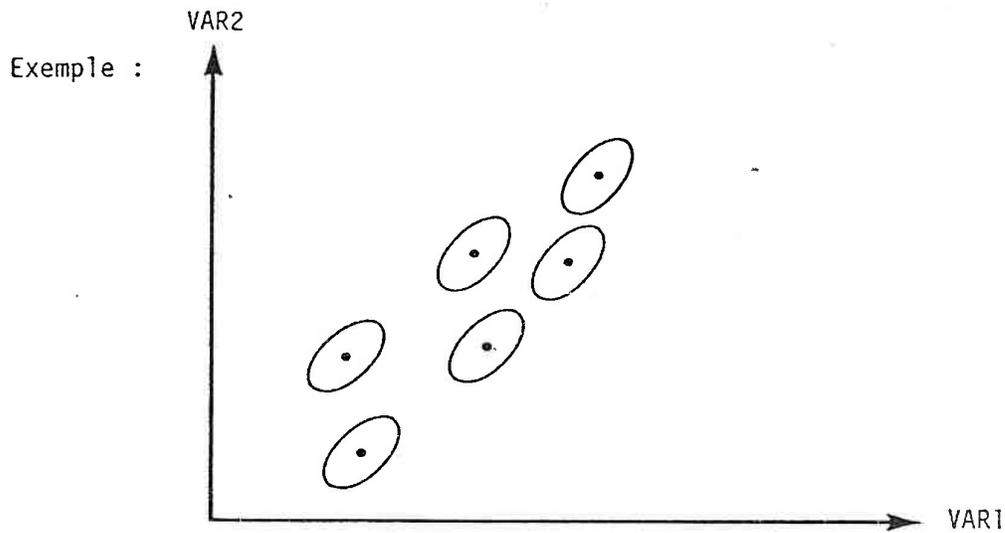


Figure 22. Trace de la base des gaussiennes dans un espace à deux dimensions

c) La deuxième version

Dans une deuxième tentative, nous avons appliqué la méthode du MISE a chaque classe de points en particulier afin d'obtenir une orientation des gaussiennes spécifiques à chaque classe.

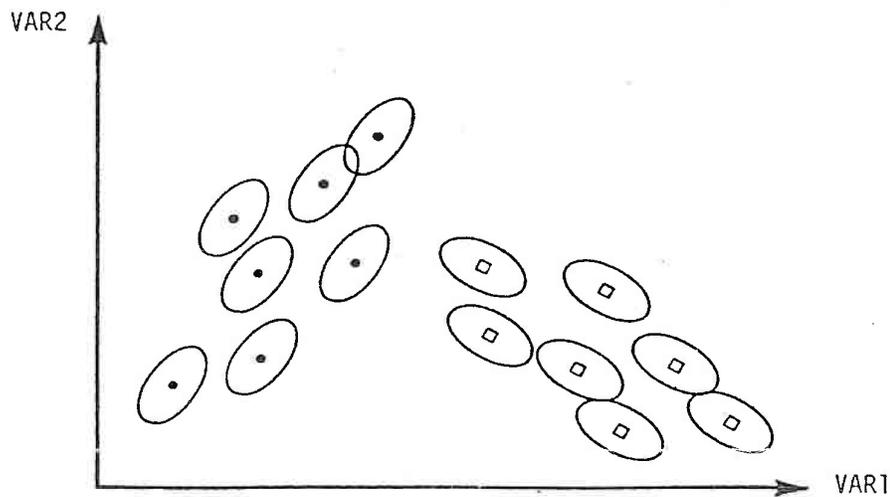


Figure 23. Trace de la base des gaussiennes avec orientations spécifiques

On faisait préalablement une analyse en composantes principales sur chacune des classes, ce qui permettait de connaître les directions principales de chacune. On faisait ensuite une transformation linéaire pour obtenir la position de ces points dans le système de coordonnées que constitue ces directions principales.

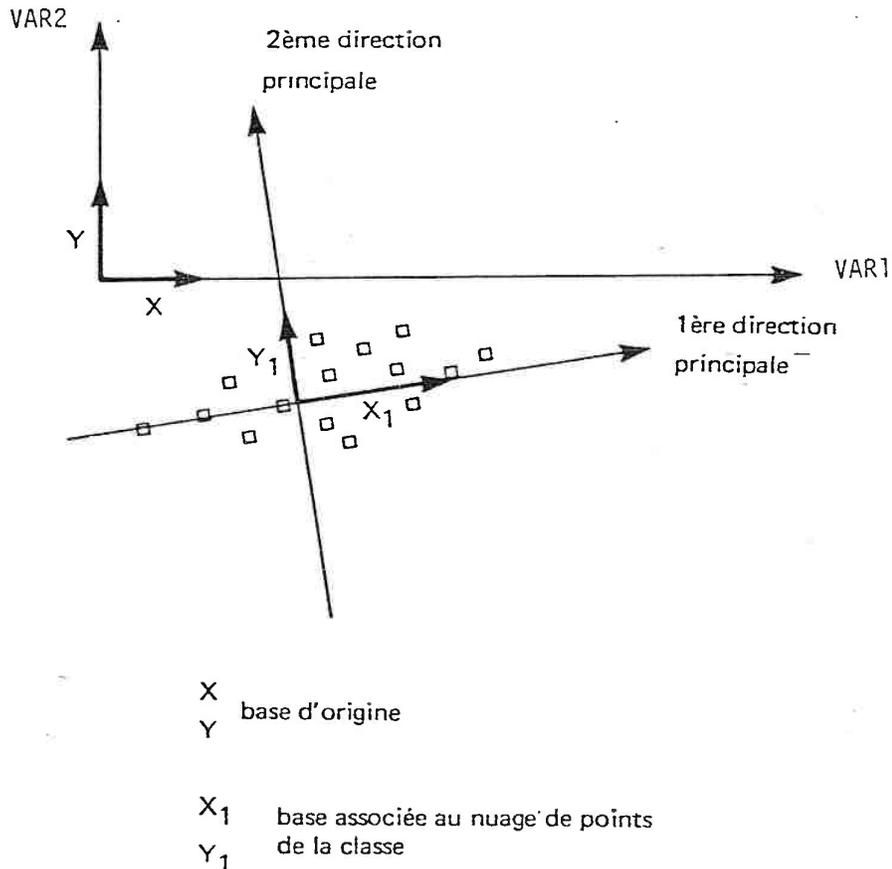


Figure 24. Nouveau système d'axes obtenu par analyse en composantes principales

Le calcul des probabilités passait aussi par cette transformation :

Pour un point quelconque de l'espace, on calculait d'abord ses coordonnées dans le nouveau système d'axe. Ensuite, on calculait les indicatrices floues dans ce nouveau système d'axes.

Pour les classes possédant peu de points, cela posait un problème.

- les classes à un élément n'étaient pas traitées. On appliquait

ensuite à ce point une gaussienne isotrope sans transformation des coordonnées

- les classes ayant moins d'éléments que la dimension de l'espace : dans ce cas, la classe est forcément comprise dans un sous-espace de l'espace.

Ex. : Dans un espace à deux dimensions, une classe formée de deux points est forcément comprise dans une droite. Une direction "principale" de la classe est alors réduite à 0.

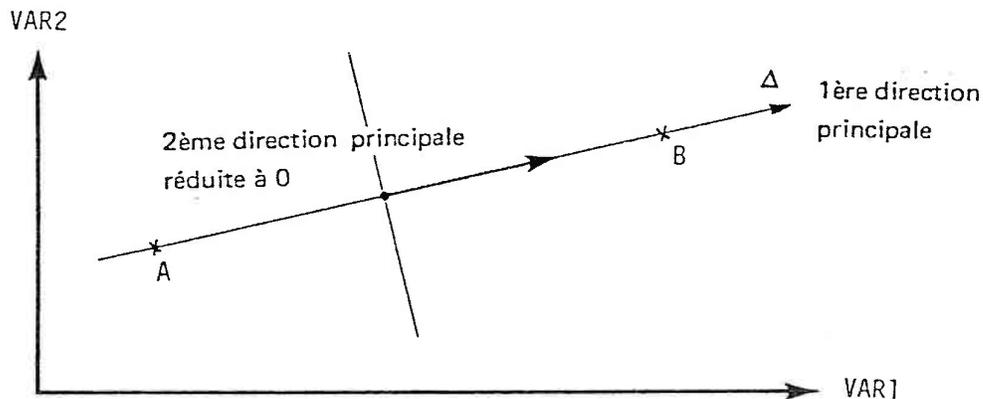


Figure 25. Cas d'une classe de deux points dans un espace à deux dimensions

La transformation linéaire qu'entraînerait une telle disposition serait une projection sur le sous-espace engendré (ici la droite Δ)

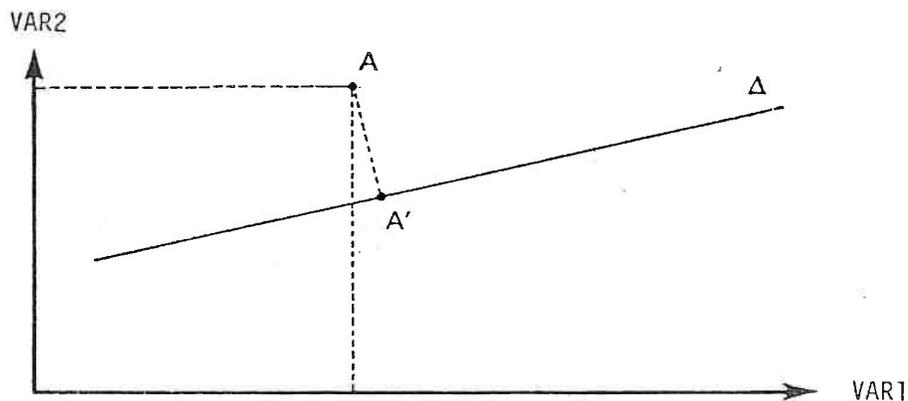


Figure 26. Transformation du point A en A'

Il fallait donc remplacer la valeur 0 dans la matrice de transformation par une valeur afin d'obtenir des directions principales non nulles. (cf. figure 27)

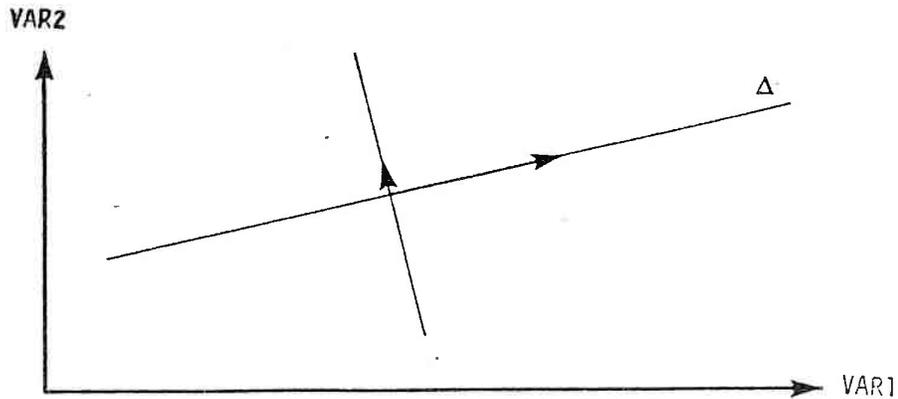


Figure 27

d) Les raisons de l'abandon des deux premières méthodes

La raison majeure est que la méthode du MISE donnait des résultats non adaptés aux besoins demandés. Les valeurs C_i obtenues étaient parfois dix fois plus petites que ce qu'il aurait fallu.

Si ces valeurs sont trop petites, on obtient des probabilités qui sont très contrastées : soit égales à 1, soit à 0.

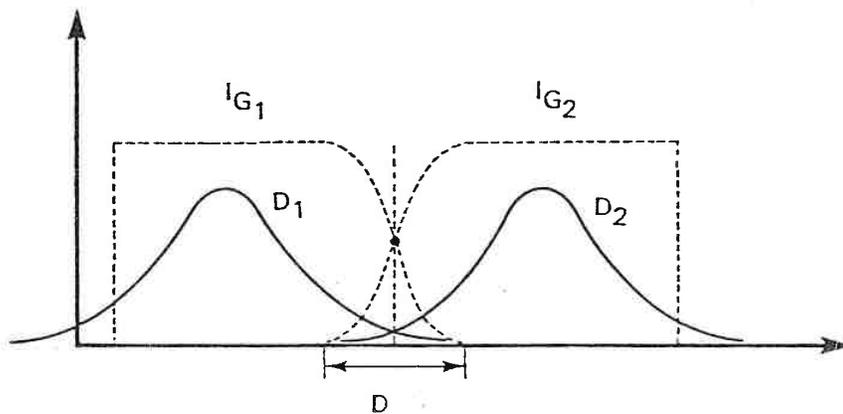


Figure 28. Cas où les gaussiennes sont étroites (C_i petits)

D est le domaine où les fonctions D_k sont de même ordre de grandeur et où, par conséquent, les probabilités ne sont ni très proches de 1 ni très proches de 0.

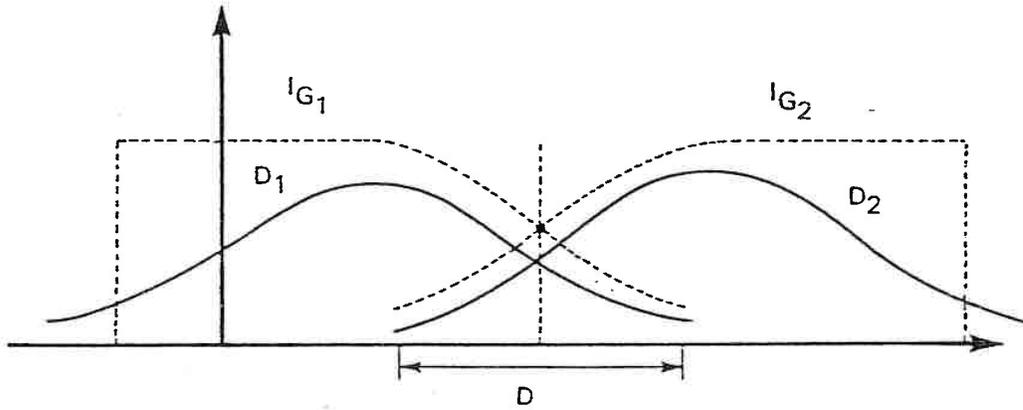


Figure 29. Cas où les gaussiennes sont étalées (C_i grands)

Une autre raison est le temps que prenait le calcul des probabilités à cause de la matrice C anisotrope. La matrice C obtenue était en fait une matrice diagonale.

$$C = \begin{vmatrix} c_1 & & \dots & 0 & 0 \\ & c_2 & & & 0 \\ & & \ddots & & \cdot \\ & & & \cdot & \cdot \\ 0 & & & & \cdot \\ 0 & 0 & \dots & & c_n \end{vmatrix}$$

$$\text{alors } \Delta(x|s,C) = \exp \left\{ -\frac{1}{2} \sum_{i=1}^n \frac{(x_i - s_i)^2}{c_i} \right\}$$

$$\text{avec } x = \begin{vmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{vmatrix} \quad \text{et} \quad s = \begin{vmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{vmatrix}$$

et n = dimension de l'espace des variables
= nombre de variables

La deuxième méthode quant à elle était complexe. Quand le nombre de variables était grand (par exemple 10), il arrivait souvent que le nombre d'éléments d'une classe soit inférieur au nombre de variables.

e) La méthode utilisée actuellement

Nous avons finalement choisi une méthode plus simple. La matrice C est isotrope et il n'y a qu'un seul coefficient c à calculer. Ce coefficient a la grandeur du carré d'une distance. Les points du support ayant été préalablement centrés et réduits, c est pris égal à la moyenne des distances (au carré) entre un point s et son plus proche voisin.

$$d_j = \min_{\substack{s \in S \\ s \neq s_j}} d(s_j, s)$$

$$c = \frac{1}{v} \sum_{j=1}^v d_j \quad \text{avec } v = \text{nombre de points de } S$$

On peut accélérer le calcul de c en ne cherchant le plus proche voisin d'un point que parmi un petit groupe de points globalement plus proches de ce point que les autres.

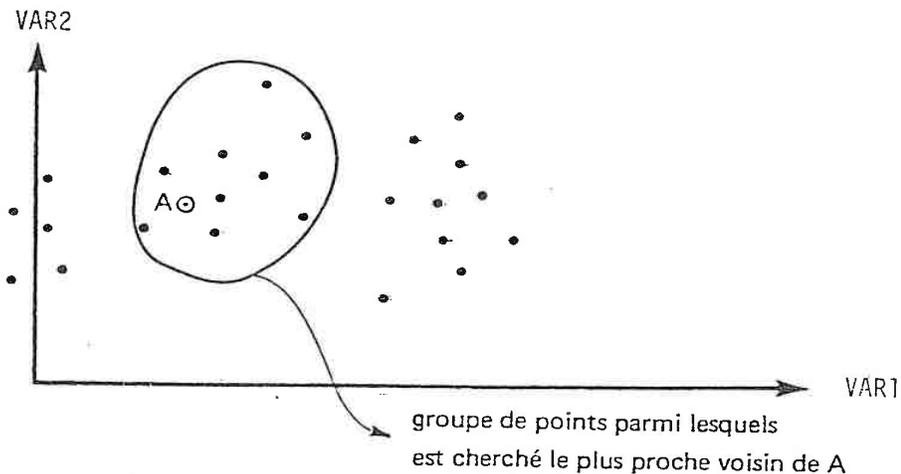


Figure 30

En général, cette méthode donne de bons résultats. Cependant, nous avons laissé à l'utilisateur la possibilité de pondérer cette valeur de c par un coefficient.

Au cours des essais que nous avons faits, nous avons du utiliser

pour ce coefficient des valeurs comprises entre 0.7 et 1.5, ce qui montre que l'ordre de grandeur de c est bon.

f) Réflexions

$$\Delta(x|s, c) = \exp \left\{ - \frac{d^2(x, s)}{2c} \right\} = \exp \left\{ - \sum_{i=1}^n \frac{(x_i - s_i)^2}{2c \sigma_i^2} \right\}$$

Les données étant centrées, réduites, les valeurs $\frac{x_i - s_i}{\sigma_i}$ sont grossièrement comprises entre -4 et +4.

Prenons une valeur moyenne de 2,5 pour $\frac{\|x_i - s_i\|}{\sigma_i}$

$$\text{alors } \sum_{i=1}^n \frac{(x_i - s_i)^2}{\sigma_i^2} = 2,5^2 \times n$$

où n = nombre de variables

$$S = \sum_{i=1}^n \frac{(x_i - s_i)^2}{2c \sigma_i^2} = \frac{6,25 \times n}{2c} = 3,1 \times \frac{n}{c}$$

or, il est intéressant d'avoir une fonction $\Delta(x|s, c)$ telle que $10^{-4} < \Delta(x|s, c) < 1$

ceci impose que $0 < S < 10$ grosso modo

$$0 < S < 10 \implies 3,1 \times \frac{n}{c} < 10 \implies c > \frac{n}{0,31}$$

Ceci donne des valeurs trop grandes : il est difficile en effet d'obtenir des indicatrices $> 10^{-4}$ sans que les probabilités soient assez floues.

IV) L'ALGORITHME DES CENTRES DYNAMIQUES

La construction d'un modèle est réalisée en deux phases :

- dans la première phase, le système calcule les noyaux du modèle,
- dans la deuxième phase, le système entretient une conversation avec l'utilisateur afin qu'il détermine ces noyaux, c'est-à-dire qu'il les classe.

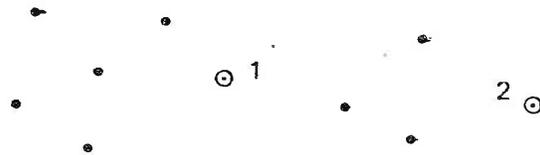
La première phase se fait à l'aide de l'algorithme des centres dynamiques : le système extrait un support S du tableau de données (les données du sondage).

1) Description de l'algorithme

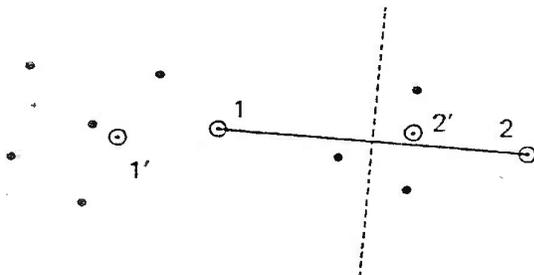
Cet algorithme sert à séparer d'une manière automatique les groupes d'un nuage de points dans un espace de dimension définie. C'est un algorithme itératif. On choisit au départ N points de l'espace. Ce sont les centres initiaux. A chaque étape de l'algorithme, on construit une partition de l'espace de telle façon que chaque partie soit le mieux regroupée autour d'un de ces N centres. Puis on calcule les barycentres de ces N classes qui deviennent à leur tour les points de base (ou centres) pour obtenir une nouvelle partition de l'espace. Ainsi les N centres se déplacent à chaque itération, d'où le nom de "centres dynamiques".

Ex. :

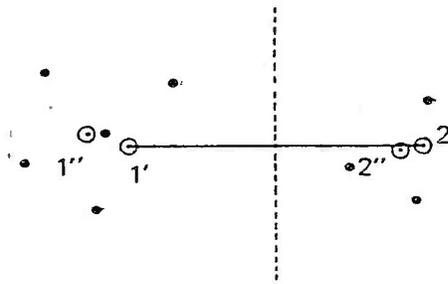
Etat initial : les centres
initiaux sont les points 1 et 2



après 1 itération, les centres 1 et 2
sont remplacés par les centres 1' et 2'



2ème itération



3ème itération : l'état est stable.
Les centres 1'' et 2'' restent inchangés.

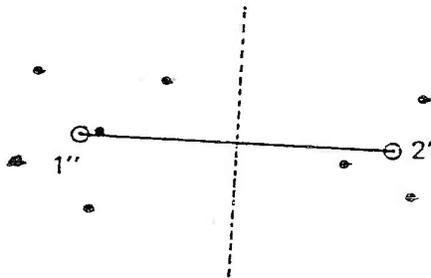


Figure 31

Nous pouvons voir sur cette figure un exemple avec deux classes dans un espace à deux dimensions. L'algorithme converge très vite vers une position stable. L'inconvénient de cette méthode est que le nombre de classes doit être fixé à l'avance.

2) Utilisation qui est faite de cet algorithme

Habituellement, l'utilisateur de cet algorithme donne un nombre de centres initiaux égal au nombre de classes de points qu'il pense obtenir :

Exemple :

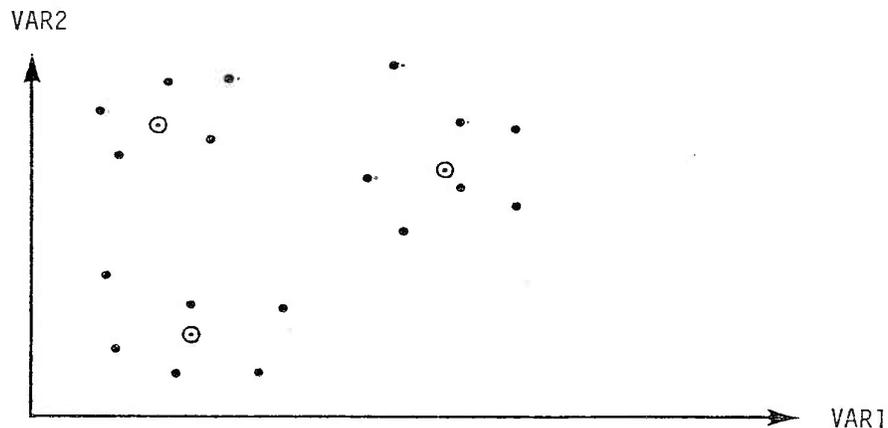


Figure 32. Les trois centres finaux partagent l'ensemble des points en trois classes bien distinctes

Il se peut cependant qu'on obtienne un nombre de centres finaux inférieur au nombre de centres initiaux. Le hasard peut faire en effet qu'à une itération donnée, un centre ne soit pas assez proche d'aucun point du nuage pour former une classe. Il est alors éliminé et les calculs se poursuivent avec un centre de moins.

Exemple :

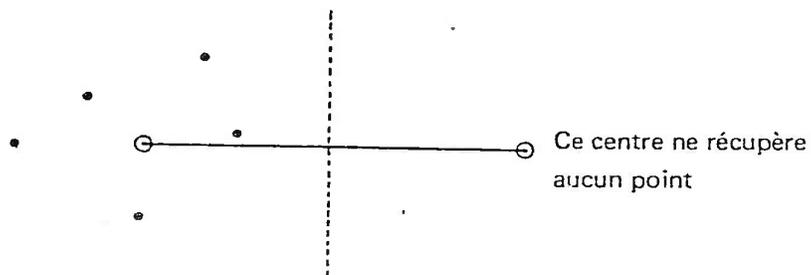


Figure 33. Perte d'un centre dans l'algorithme des centres dynamiques

Dans notre cas, le but n'est pas le même. Nous cherchons simplement à obtenir un ensemble de points représentatifs de l'ensemble des points de données. Les ensembles naturels de points peuvent être partagés en deux ou trois classes.

Exemple :

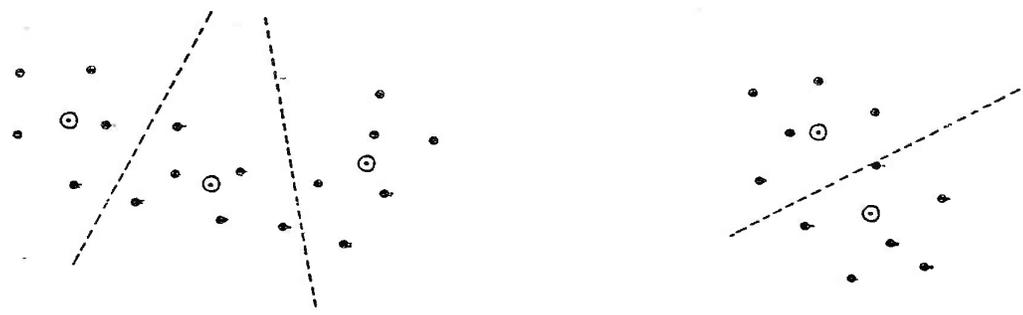


Figure 34

Comme ces ensembles naturels de points correspondent le plus souvent à des faciès diagraphiques, plusieurs noyaux dans un modèle portent le même nom de faciès. Les centres initiaux sont tirés au hasard parmi les points de données. Ceci évite qu'un centre initial soit éliminé dès la première itération.

3) Le nombre de centres initiaux

En conséquence, le nombre de centres qui doit être demandé en entrée est bien supérieur au nombre de faciès. Pour construire un modèle à partir d'un sondage comprenant 10 variables et 3 000 points de données, nous avons du utiliser 200 noyaux caractérisant au total 10 faciès. Il faut faire remarquer que plus le nombre de variables est grand, plus l'espace des variables est vaste et plus il faut de noyaux pour décrire la place qu'occupent les faciès dans cet espace.

4) Le nombre d'itérations

La proportion de centres sur le nombre total de points influe sur la vitesse de convergence. En effet, on conçoit facilement que plus cette densité est grande, moins les centres auront la possibilité de bouger et plus vite ils trouveront une position définitive.

Le nombre d'itérations n'a donc pas besoin d'être grand. Une à trois itérations suffisent.

5) La métrique

L'algorithme des centres dynamiques est basé sur le calcul des distances entre deux points. La métrique est une matrice carrée d'ordre N qui sert à définir le produit scalaire entre deux vecteurs de R_n .

Soit deux vecteurs x' et x'' de \mathbb{R}^N et M une matrice carrée symétrique d'ordre N supposée définie positive. (Ses valeurs propres sont alors toutes positives ou nulles.)

$$x' = \begin{pmatrix} x'_1 \\ x'_2 \\ \cdot \\ \cdot \\ x'_N \end{pmatrix} \quad x'' = \begin{pmatrix} x''_1 \\ x''_2 \\ \cdot \\ \cdot \\ x''_N \end{pmatrix} \quad M = \begin{pmatrix} M_{11} & M_{12} & \cdot & \cdot & M_{1N} \\ M_{21} & M_{22} & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ M_{N1} & \cdot & \cdot & \cdot & M_{NN} \end{pmatrix}$$

M est appelée métrique dans l'espace \mathbb{R}^N si on définit le produit scalaire entre les vecteurs x' et x'' par la relation suivante :

$$\langle x' | x'' \rangle = x'^t M x'' = \sum_{ij} x'_i M_{ij} x''_j$$

La longueur d'un vecteur est alors définie par :

$$\|x\|^2 = \langle x | x \rangle = x^t M x$$

La distance entre deux vecteurs est définie par :

$$\begin{aligned} d^2(x', x'') &= \|x' - x''\|^2 \\ &= (x' - x'')^t M (x' - x'') \end{aligned}$$

Si M est la matrice unité, alors :

$$d^2(x', x'') = (x' - x'')^t (x' - x'') = \sum_{i=1}^N (x'_i - x''_i)^2$$

On retrouve la formule de Pythagore ou formule de la distance Euclidienne. Cette formule accorde la même importance à toutes les directions de l'espace.

Cependant, en analyse des données, il n'en va pas de même car chaque direction de l'espace est associée à une variable aléatoire qui s'exprime avec son unité particulière.

Exemple : Les résistivités exprimées en OHM.M ont des valeurs comprises entre 0.01 et 10 000., alors que les indices d'hydrogène varient entre 0. et 100. Pour ramener toutes ces valeurs dans les mêmes limites, on affecte chaque axe d'un poids égal à l'inverse de la variance de la variable aléatoire associée à cet axe.

alors :

$$d^2(x', x'') = \sum_{i=1}^N \frac{1}{\sigma_{X_i}^2} (x'_i - x''_i)^2$$

avec σ_{X_i} = écart-type de la V.A. X_i associée au $i^{\text{ème}}$ axe.

La métrique est alors égale à

$$M = \begin{vmatrix} 1/\sigma_{X_1}^2 & . & . & . & . & 0 & 0 \\ . & 1/\sigma_{X_2}^2 & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . \\ 0 & 0 & . & . & . & . & 1/\sigma_{X_N}^2 \end{vmatrix}$$

Cette manière de calculer les distances équivaut à réduire les données avant de calculer les distances Euclidiennes. Dans le programme, on appelle par commodité cette métrique la métrique de réduction.

Une autre métrique appelée "métrique de Mahalanobis" est également très utilisée et consiste à poser :

$$M = [E\{(X - m_X) \cdot (X - m_X)^t\}]^{-1}$$

M est la matrice inverse de la matrice des covariances. Cette métrique tient donc compte des coefficients de corrélation entre les diverses variables.

Par exemple, un point à forte résistivité LLS sera assez proche d'un point à forte résistivité LLD quelle que soit sa propre résistivité LLD.

Le programme laisse à l'utilisateur la possibilité de choisir la métrique qu'il désire. Cependant, la métrique unité n'est pas réellement intéressante. La métrique de Mahalanobis, par contre, semble légèrement trop sophistiquée pour des résultats qui diffèrent peu de ceux obtenus avec la métrique de "réduction".

Cf.paragraphe "Essais et applications".

Pour ces trois options : le nombre de centres demandés, le nombre d'itérations et la métrique, ce système laisse le choix à l'utilisateur. Mais si l'utilisateur ne les définit pas, alors le système choisit des valeurs "standards".

Nombre de noyaux : 50
Nombre d'itérations : 2
Métrique : métrique de réduction.

Dans la pratique, ces deux dernières options n'auront pas besoin d'être changées souvent.

V) L'ASPECT INFORMATIQUE

1) Introduction

Ce programme a été écrit en compatibilité avec le logiciel GEOL. Toute la partie informatique en est influencée. En particulier, les structures de données et le type de la conversation générée par le programme.

2) Nature et structure des données manipulées

a) Définition d'un GDT

Au cours d'une session d'analyse diagraphique, il apparaît nécessaire de définir :

- un fichier contenant les données relatives au forage étudié,
- un fichier contenant le modèle approprié.

Ces fichiers sont des fichiers GDT (Geol Data Table).

Un fichier GDT et un fichier de données ayant une structure propre au système GEOL. Ils ont été conçus pour le stockage de données associées à des points de données répartis d'une façon régulière dans un espace de dimension 1, 2 ou 3.

Ex. Données mesurées à intervalles réguliers le long d'un axe (dimension 1) Ce type de fichier est appelé DT1

Données mesurées aux noeuds d'une grille régulière sur un plan (dimension 2). Ce type de fichier est appelé DT2

etc... De même il existe des fichiers de type DT3.

Les données irrégulièrement réparties peuvent aussi être stockées dans un fichier GDT. Ce fichier est alors dit de type DT.

Un fichier GDT est un fichier non formaté contenant pêle-mêle des données alpha-numériques, réelles et/ou entières. Il est chargé entièrement en mémoire centrale lors de l'exécution, exploité en mémoire centrale, puis éventuellement réécrit sur mémoire de masse.

Les données sont réparties sous forme de composantes et d'unités statistiques, selon la structure représentée sur la figure 35.

- Une composante est un ensemble de données de même nature. (ex: mesure de la résistivité LLS sur un sondage)
Il existe ainsi des composantes de type réel, entier, alpha-numérique ou booléen.

- Une unité statistique est un ensemble de données de natures différentes associées généralement à un point géographique. (ex: ensemble des mesures prises à une profondeur donnée d'un puits de forage)

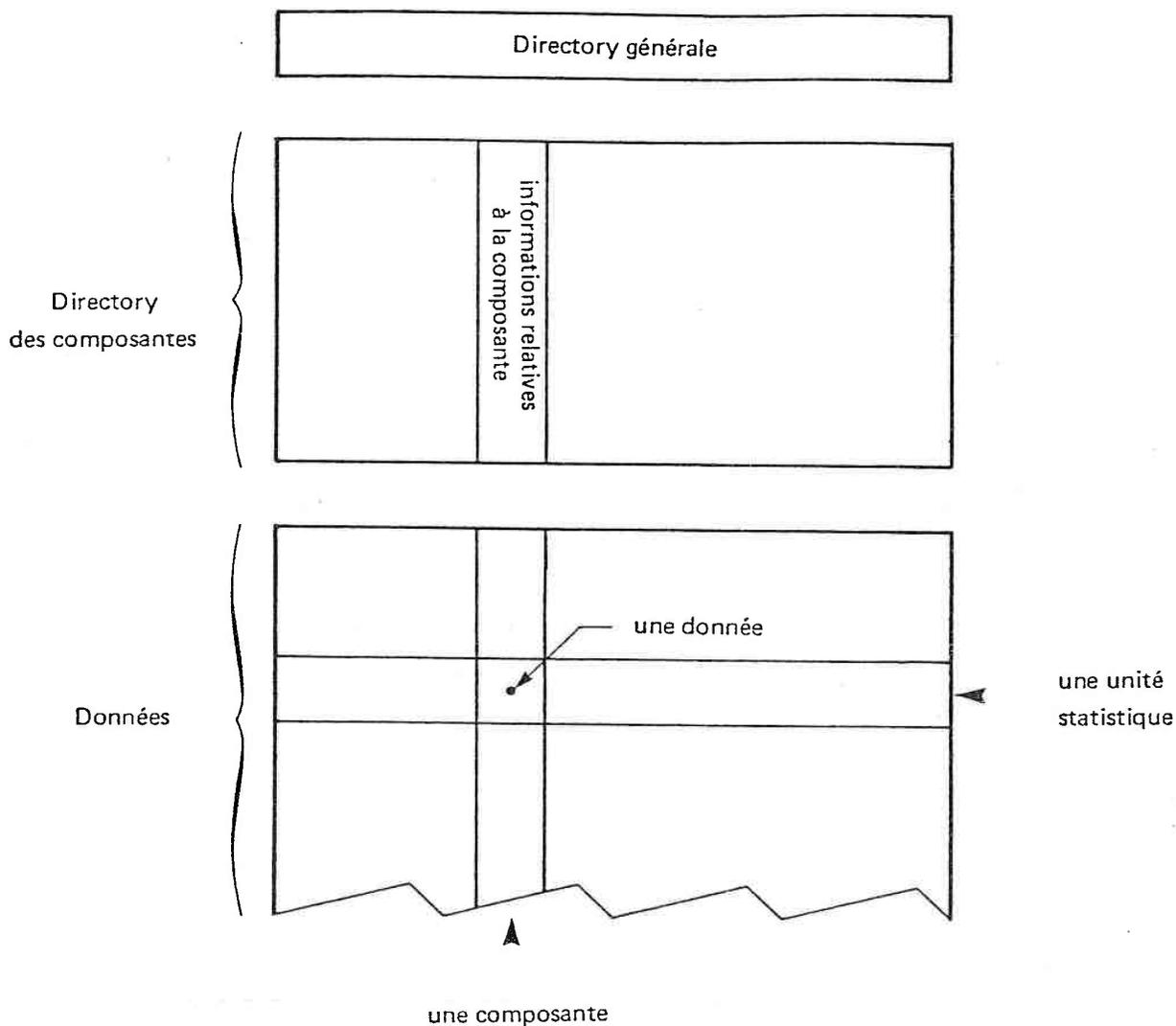


Figure 35. Structure d'un GDT

Certaines composantes, appelées composantes "set" ("set" veut dire "ensemble" en anglais) ne contiennent pas de données. Elles servent à définir des sous-ensembles d'unités statistiques du GDT à raison d'une composante par sous-ensemble.

La valeur d'une composante set au niveau d'une unité statistique contient "-1" si l'unité statistique n'appartient pas au sous-

ensemble, sinon elle contient le numero de l'unité statistique suivante appartenant aussi au sous-ensemble. Nous supposons que dans un GDT, l'ordre dans lequel sont rangées les composantes est sans importance et que chaque composante est repérée à l'aide d'un nom propre (identificateur), c'est à dire une chaîne de 1 à 6 caractères alpha-numériques commençant obligatoirement par une lettre.

Il existe aussi des composantes dites vectorielles qui sont un ensemble de composantes de même type. Cet ensemble de composantes est groupé sous un même nom (par exemple CVEC et chaque élément est alors désigné par CVEC [i], i étant le rang de l'élément)..

Le début du fichier GDT contient sa "directory", c'est à dire un ensemble d'informations relatives aux données qui suivent.

- titre du fichier
- nom (identificateur), titre et type (entier, réel, set, etc...)
- de chaque composante du fichier.

Ces fichiers sont écrits en binaire sur disque et ne sont donc pas accessibles aux éditeurs de texte courants. Ils peuvent être édités grace aux commandes EDWEL et EDMOD (respectivement pour le "CURRENT-GDT-well" et le "Current GDT-model").

Il est possible d'éditer les "directories" de chacun de ces fichiers courants par l'appel de :

- la sous-commande WELDIR pour le "Current GDT-well"
- la sous-commande MODDIR pour le "Current GDT-model"

b) Les GDT-well

Type d'un GDT-well

On appelle GDT-well le type de fichier contenant les données d'un sondage diagraphique. Dans le cas d'enregistrements diagraphiques, les données sont réparties le long d'un axe (en l'occurrence, la profondeur). On parlera de DT1 (Data Table à une dimension). Il faut que les points de mesure soient espacés régulièrement le long du sondage..

Les composantes d'un GDT-well

Les composantes initiales d'un GDT-well sont les enregistrements diagraphiques numérisés ayant subi éventuellement certaines corrections classiques en géophysique.

Le géologue pourra éventuellement y rajouter des composantes contenant des propriétés physiques de la roche (ex. : la porosité) et d'une manière générale toute nouvelle information dépendant de la

profondeur et pouvant intéresser l'utilisateur.

En particulier, il pourra rajouter une ou plusieurs composantes "set" correspondant aux portions sondage qu'il désire individualiser.

Lors de l'exécution de la commande, l'utilisateur peut réduire les calculs à un sous-ensemble du puits en précisant, à l'aide de la sous-commande SETDEF, le nom de la composante "set" correspondant au sous-ensemble choisi.

Enfin, les résultats seront aussi stockés sous forme de composantes à la suite de l'analyse lithologique :

- une composante lithologie (obligatoire)
- d'autres composantes contenant les autres lithologies possibles le long du puits et/ou les probabilités d'existence de ces lithologies.

Le current GDT-well

Il n'y a en permanence qu'un seul GDT-well en mémoire centrale sur lequel sont effectués les travaux. C'est pourquoi il est préférable de rassembler toutes les données relatives à un même sondage dans un même fichier de données.

Le fichier contenu en mémoire centrale est appelé "Current GDT-well". Il est choisi par l'utilisateur au moyen de la sous-commande WELLDEF qui provoque le chargement en mémoire centrale du fichier existant sur le disque magnétique.

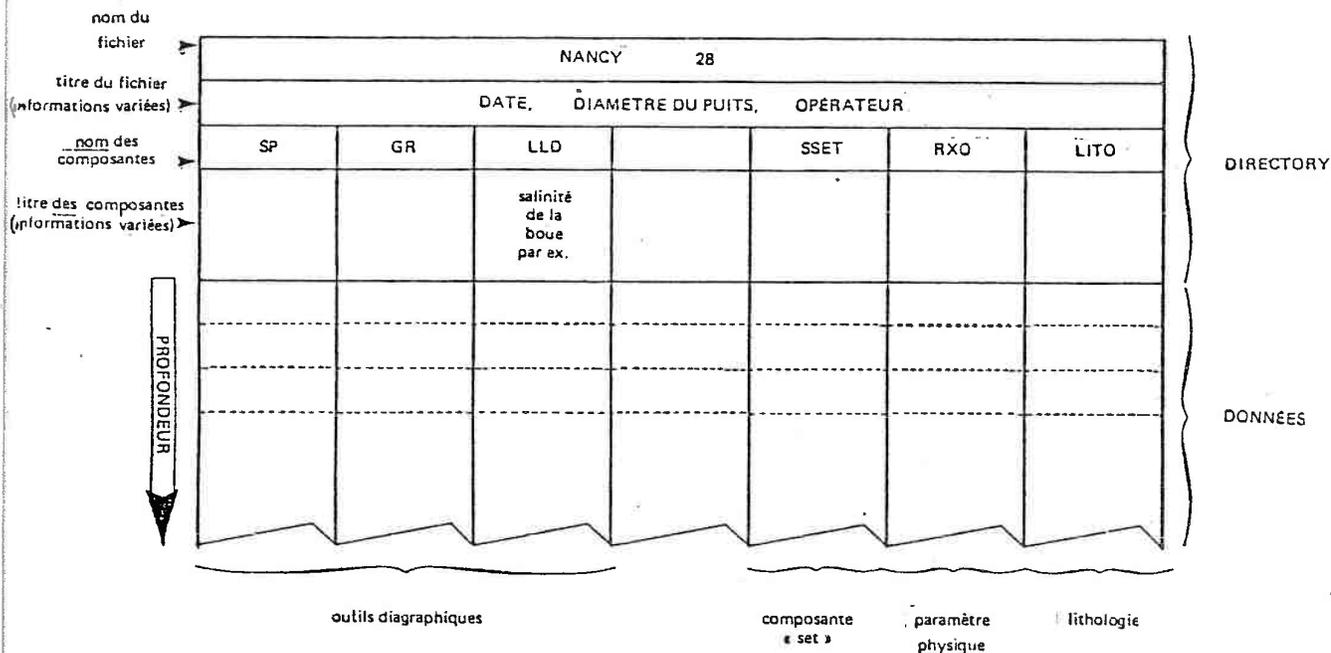


Figure 36. Exemple de GDT-well

c) Les GDT-model

Type d'un GDT-model

On appelle GDT-model le type de fichier contenant les données d'un modèle. Un GDT-model rassemble des données synthétiques. En effet, les échantillons d'un modèle ne correspondent pas à des enregistrements réels effectués sur le terrain. Ils ne correspondent a

aucun forage précis. L'ordre dans lequel ils sont rangés n'importe pas.

Les données n'étant pas associées à des points répartis régulièrement dans l'espace (contrairement aux données d'un puits), elles sont stockées dans un fichier de type DT.

Les composantes d'un GDT-model

Un GDT-model comprend :

- des composantes analogues à celles des GDT-well contenant des mesures diagraphiques, des propriétés physiques (ou toute information numérique susceptible de discriminer les faciès). Il est essentiel que des composantes d'un GDT-well et d'un GDT-model contenant des données de nature exactement semblable (ex.: résistivité LLS en OHM.m) portent le même nom (même identification) dans chaque GDT.

Inversement, il est essentiel que deux composantes portant le même nom dans un GDT-well et un GDT-model contiennent des données de nature exactement semblable.

- une composante contenant le nom de faciès attribué à chaque échantillon du modèle. Cette composante porte dans tous les GDT-model le nom de code "FACIES".
- une composante contenant le poids affecté à chaque échantillon du modèle. Cette composante porte invariablement le nom de code "WEIGHT" (qui veut dire "poids" en anglais).

Le current GDT-model

De même que pour le puits, un seul modèle peut être chargé en mémoire centrale. On l'appelle le "Current GDT-model".

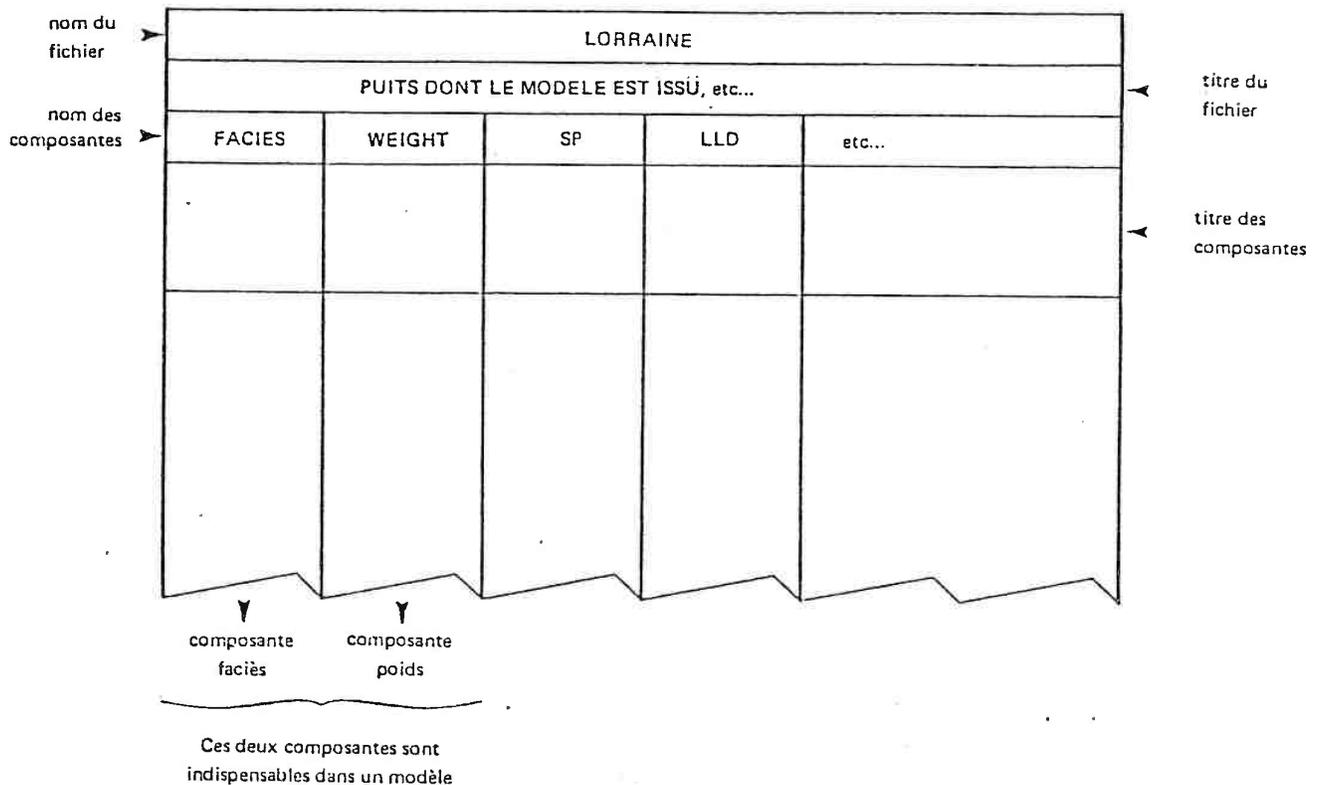


Figure 37. Exemple de GDT-model

d) Les composantes de travail

Le Data-Tools

Le Data-Tools contient les noms de toutes les composantes du GDT-well correspondant à des outils diagraphiques (d'où le nom de Data-Tools) et plus généralement les noms de toutes les composantes susceptibles d'être utiles à la discrimination des faciès.

NB : Il est indispensable que de telles composantes portent des noms répertoriés par l'utilisateur et que par contre aucune composante étrangère ne porte un de ces noms répertoriés. Le répertoire de ces noms est contenu dans un fichier séquentiel ayant pour nom "tooldia". Ce fichier peut être modifié à tout moment par l'utilisateur avec l'éditeur du système d'exploitation courant.

Le Current-Tools

Il est possible que l'utilisateur ne veuille point employer l'ensemble de toutes les composantes du "Data-Tools".

On appelle "Current-Tools" l'ensemble des composantes que l'utilisateur souhaite conserver pour effectuer les opérations qu'il veut sur le "Current GDT-well".

Lors du chargement d'un puits, le "Current-Tools" est égalisé au "Data-Tools". La sous-commande TOOLS permet ensuite à l'utilisateur de modifier à sa guise le "Current-Tools".

Le Working-Tools

Lors de la phase "analyse faciologique" proprement dite, il est nécessaire d'utiliser à la fois un GDT-well et un GDT-model. Le "working-Tool" est alors l'ensemble des composantes présentes à la fois dans le "Current-Tools" et dans le "Current-GDT-model" : ce sont les composantes de travail.

Remarque :

Lors de la construction d'un modèle à partir d'un GDT-well, c'est le "Current-Tools" qui est pris en compte.

3) Les règles de la conversation

Comme toutes les commandes du système GEOL, la commande DIA est entièrement conversationnelle. L'utilisateur tape sur un terminal d'ordinateur le non propre de la commande et à partir de là s'engage un dialogue homme-machine permettant de préciser les modalités du travail à effectuer.

Lorsque le système attend une intervention de l'utilisateur, soit en lui posant une question précise, soit en éditant un témoin d'attente tel que T#, M#, A# ou #, alors l'utilisateur peut :

- abandonner le travail en cours en répondant par "STOP". Le programme revient alors au niveau "SUBCOMMAND = ?", niveau où le programme attend la précision d'une nouvelle tâche de la part de l'utilisateur (voir V 5)).
- répondre par "?" ; le système édite alors une explication sur l'objet de la question et sur la signification des réponses possibles, puis pose à nouveau la question.
- taper directement sur la touche <Carriage Return> ; le système remonte alors à la question posée précédemment et l'utilisateur peut ainsi modifier la réponse qu'il avait faite à cette question.

- faire avancer le déroulement du programme en introduisant une réponse adéquate.

Dans le cas d'une réponse inadéquate, soit parce qu'elle est erronée, soit parce qu'elle est incohérente avec les réponses antérieures, le système édite un message d'erreur et pose la question de nouveau.

Certains paramètres peuvent ne pas être définis par l'utilisateur au cours de l'exécution. Le système leur affecte alors la valeur qui leur a été donnée par l'utilisateur lors de son dernier passage au même endroit du programme (valeur standard) et si l'utilisateur n'a encore jamais défini ces paramètres au cours de la session, le système leur affecte une valeur définie dans le programme (valeur normalisée). C'est le cas notamment pour le nombre de noyaux demandés à l'entrée de l'algorithme des centres dynamiques.

Enfin, toutes les commandes du système GEOL produisent, lors de leur exécution, un fichier contenant le dialogue utilisateur-machine complet de la session. Ce fichier-dialogue peut être alors réutilisé lors d'une autre session. Evidemment, pour un dialogue identique, l'exécution sera identique. L'utilisateur peut changer certaines réponses du dialogue à l'intérieur du fichier-dialogue à l'aide d'un éditeur classique. Il peut autrement supprimer à l'intérieur du fichier-dialogue l'ensemble question-réponse concerné. Lors de l'exécution de la session avec ce dialogue, le système s'arrêtera à cette question, enregistrera la réponse au clavier de l'utilisateur et poursuivra l'exécution à l'aide du fichier-dialogue.

4) La convivialité

Les règles de conversation décrites ci-dessus permettent une très grande convivialité du système.

En effet, les réponses de l'utilisateur sont vérifiées, ses erreurs sont détectées, parfois même corrigées. Il peut obtenir des informations sur l'utilisation de la commande à l'intérieur même de la commande. Il peut changer les modalités d'exécution qu'il avait déjà précisées sans recommencer une nouvelle session. Enfin, il peut réaliser pendant la même session plusieurs fois le même travail, quel qu'il soit, avec de nouveaux paramètres.

- Ex. :
- création de plusieurs modèles avec nombres de noyaux différents
 - analyse faciologique d'un puits avec des modèles différents.

5) Architecture de la commande

Cette commande est destinée à accomplir des tâches indépendantes (construction d'un modèle, analyse faciologique), aussi est-elle dotée d'une structure arborescente, c'est à dire qu'elle présente plusieurs niveaux d'utilisation et à chacun de ces niveaux, l'utilisateur a le choix entre plusieurs tâches possibles.

Le niveau principal est marqué par la question
SUBCOMMAND = ?
à laquelle l'utilisateur doit répondre par un nom de sous-commande (qui est une chaîne de 8 caractères).

L'entrée sur le clavier d'un de ces noms de sous-commandes entraîne

- pour certaines, l'exécution d'une tâche bien précise,
- pour d'autres, l'entrée dans un niveau inférieur d'où l'on peut accéder :
 - * soit à des sous-sous-commandes (cas de la sous-commande ANADIA)
 - * soit à des instructions (cas des sous-commandes EDWEL, EDMOD)

En général, une instruction est une directive de l'utilisateur contenant en elle-même tous les paramètres utiles à l'exécution de la tâche qu'elle lance, alors qu'une sous-commande entraîne un dialogue homme-machine lors de son exécution.

Certaines sous-commandes ont la possibilité d'être introduites sous forme d'instructions comportant plusieurs membres, ce qui permet ainsi d'occulter une partie des questions et de réduire le temps de mise en oeuvre de ces sous-commandes. C'est le cas des sous-commandes LOGSAV, MODSAV et INCLUD.

a) Liste des sous-commandes

* chargement et définition des données

WELLDEF charge le "Current GDT-well"
SETDEF - définit un sous-ensemble de données du
"Current GDT-well"
MODDEF charge le "Current GDT-model"
TOOLS définit le "Current-Tools"

* visualisation et modifications des données d'un puits

WELLDIR édite la directory du "Current GDT-well"
EDWELL édite les données du "Current GDT-well" et permet des
modifications sur celles-ci
EXTWELL crée un fichier plus grand à partir du
"Current GDT-well" et le sauve sur disque
CBLD crée une nouvelle composante dans le "Current GDT-well"
SETBLD crée une composante "set" dans le "Current GDT-well"
CDEL supprime une composante du "Current GDT-well"
(Component DElete)
EDTO édite le "Data-Tools" et le "Current-Tools"

* visualisation et modifications des données d'un modèle

MODDIR édite la directory du "Current GDT-model"
EDMOD édite les données du modèle et permet certaines
modifications sur celles-ci.
CROSSV cross-validation du "Current GDT-model"
MODMNG création d'un modèle à partir de deux autres (cette
sous-commande n'est pas encore réalisée)

* traitement numérique des données

MODBLD (Model Build) Construction d'un nouveau GDT-model
à partir du "Current GDT-well"
ANADIA (sous-commande RUN)
ANALyse DIAgraphique du "Current GDT-well"

à partir du "Current GDT-model"

ANADIA (sous-commande SMTH)
Lissage des résultats de la sous-commande RUN

* visualisation et sauvegarde des résultats

ANADIA (sous-commande VISU)
Edition des résultats des sous-commandes RUN et SMTH

ANADIA (sous-commande INCLUD)
Inclusion des résultats sous forme de composantes
dans le "Current GDT-well"

WELLSAV Sauvegarde du "Current GOT-well" sur disque
magnétique

MODSAV Sauvegarde du "Current GDT-model" sur disque
magnétique.

b) Cas particulier des sous-commandes WELLDEF et MODDEF

L'exécution de certaines sous-commandes nécessite l'emploi d'un
GDT-well ou d'un GDT-model. Si l'utilisateur appelle une de ces sous-
commandes sans auparavant avoir chargé un puits ou un modèle, alors le
système appelle automatiquement la sous-commande appropriée (WELLDEF
ou MODDEF) et édite la question

CURRENT GDT-WELL = ?

ou CURRENT GDT-MODEL = ?

VI) ANALYSE DIAGRAPHIQUE

1) Introduction

Dans cette section, nous allons présenter la démarche à suivre dans le cas de l'analyse d'un puits (sans visualisation des données, ni construction de modèle, etc.)

Nous aurons à employer successivement diverses sous-commandes que nous étudierons brièvement. Le lecteur pourra se reporter à la section X de ce manuel pour connaître le mode d'emploi complet de ces sous-commandes.

2) Résumé des opérations à effectuer

L'analyse diagraphique nécessite la définition préalable d'un ensemble de données à traiter (GDT-well) (VI 4) et (VI 6) et d'un ensemble d'échantillons définissant les faciès possibles (GDT-model) (VI 5). Ensuite peuvent se dérouler les calculs aboutissant à des résultats qui sont sous forme de probabilités (VI 7) : probabilités pour qu'un échantillon de l'ensemble des données soit d'un faciès ou d'un autre). L'utilisateur peut alors vérifier ces résultats à l'écran avant de les sauvegarder.

La sauvegarde se fait en deux temps :

- dans un premier temps, les résultats sont introduits dans le "Current GDT-well" (VI 11)
- dans un deuxième temps, le "Current GDT-well" doit être sauvegardé sur disque magnétique. (VI 12)

3) Liste des sous-commandes abordées dans cette section

sous-commande	:	paragraphe de cette section
-----	:	où elle est abordée
WELLDEF	:	VI 4)
MODDEF	:	VI 5)
EDTO	:	VI 6)
TOOLS	:	VI 6)
SETDEF	:	VI 7)
	:	VI 8)
	:	VI 9)
ANADIA	:	VI 10)
	:	VI 11)
EDWELL	:	VI 12)
WELLSAV	:	VI 12)
	:	

4) Chargement des données

A l'appel de la commande, le système édite la question
SUBCOMMAND = ?

Pour demander au programme de charger des données en mémoire centrale, l'utilisateur doit taper le nom de sous-commande "WELLDEF" (pour WELL DEFINITION).

Le système demande alors à l'utilisateur quel fichier GDT il doit charger en posant la question :
CURRENT GDT-WELL = ?

Une fois que l'utilisateur a répondu un nom de fichier qui existe réellement et qui soit bien un GDT-well, le système revient au niveau SUBCOMMAND = ?

Exemple :

- > SUBCOMMAND = ?
- WELLDEF
- > CURRENT GDT-WELL = ?
- WELL.DTI
- > SUBCOMMAND = ?

5) Chargement d'un modèle

Le chargement d'un modèle s'effectue au moyen de la sous-commande MODDEF (pour MODEL DEFINITION).

Le système édite la question
CURRENT GDT-MODEL = ?

à laquelle l'utilisateur doit répondre un nom de fichier contenant bien un GDT-model. Puis, le système revient au niveau "SUBCOMMAND = ?".

Exemple :

- > SUBCOMMAND = ?
- MODDEF
- > CURRENT GDT-MODEL = ?
- MODEL.DT
- > SUBCOMMAND = ?

Remarque : Ces deux sous-commandes jouent des rôles indépendants, et n'ont pas à être appelées suivant un ordre précis.

6) Définition des composantes

Rappel : Une composante, on l'a vu au paragraphe V 4), est un ensemble de valeurs réparties le long du puits à pas régulier. Le "Data-Tools" est l'ensemble des composantes du "Current GDT-well" qui sont susceptibles de servir à la discrimination des faciès.

Le "Working-Tools" est l'ensemble des composantes de travail, intersection entre le "Current -Tools" et les composantes présentes dans le "Current GDT-model".

a) La sous-commande EDTO

Au chargement du "Current GDT-well", le "Current Tools" est égalisé au "Data-Tools". (cf. V 2 b).

On peut vérifier ceci en appelant la sous-commande EDTO tout de suite après la sous-commande WELLDEF.

Cette sous-commande rappelle en premier lieu le "Current GDT-well" :

```
CURRENT GDT-WELL : WELL.DT1
```

puis elle édite les composantes du "Data-Tools" et du "Current -Tools" (ici égaux)

```
"CURRENT-TOOLS" = "DATA-TOOLS"
```

```
SP GR NPHI LLS LLD etc.
```

et le système revient au niveau "SUBCOMMAND = ?".

b) La sous-commande TOOLS

Pour sélectionner les composantes à partir desquelles devra s'effectuer l'analyse faciologique (le Working-Tools), l'utilisateur devra donc extraire un nouveau "Current-Tools" à partir du "Data-Tools". Ceci se fait au moyen de la sous-commande "TOOLS".

L'utilisateur tape donc le nom de sous-commande "TOOLS" en réponse à la question "SUBCOMMAND = ?".

Le système rappelle alors le nom du fichier courant et les composantes du "Data-Tools" et du "Current-Tools" (sous-commande EDTO)

65

```
CURRENT GDT-WELL : WELL.DTI
"CURRENT-TOOLS" = "DATA-TOOLS" :
SP GR  NPHI  LLS  LLD  etc.
```

Puis le système édite le témoin d'attente T# : l'utilisateur peut alors effectuer les opérations qu'il veut sur le Current-Tools. Dans notre cas, il s'agit de supprimer des composantes. Ceci se fait à l'aide de l'instruction REM.

```
Ex. : T#
      REM  NPHI  GR
```

Le système édite alors le nouveau "Current-Tools"

```
NEW  "CURRENT-TOOLS" :
SP   LLS  LLD  etc.
T#
```

L'édition du témoin d'attente "T#" indique que le système attend une ou plusieurs instructions parmi les suivantes ADD, REM, CHG, EDTO, OK, OKSAVE. ADD sert au rajout d'une composante, CHG (change) sert à la substitution d'une composante. (cf. X).

Quand l'utilisateur a fini de préciser le "Current-Tools", il doit simplement frapper la touche <Carriage Return> pour revenir au niveau "SUBCOMMAND = ?".

Remarque :

Supposons qu'il n'y ait pas encore de "Current-GDT-well" chargé en mémoire centrale. Si l'utilisateur doit utiliser la sous-commande TOOLS pour définir un "Current-Tools" à partir du "Data-Tools" d'un GDT-well, il n'est pas obligé d'appeler la sous-commande WELLDEF auparavant.

Dans ce cas, à l'entrée dans la sous-commande TOOLS, le système demande un nom de fichier GDT-well à charger en posant la question suivante :

```
CURRENT GDT-WELL = ?.
```

Exemple d'utilisation de la sous-commande TOOLS associée à la sous-commande WELLDEF :

```

opérations détaillées           : chargement automatique
                                : du "CURRENT GDT-WELL"

-> SUBCOMMAND = ?               : -> SUBCOMMAND = ?
- WELLDEF                       :
-> CURRENT GDT-WELL = ?         : - TOOLS
- WELL.DT1                      :
-> SUBCOMMAND = ?               : -> CURRENT GDT-WELL = ?
- TOOLS                          :
-> CURRENT GDT-WELL : WELL.DT1   : - WELL.DT1

```

```

"CURRENT TOOLS" = "DATA TOOLS"
SP GR NPHI LLD LLS MNOR
T#
- REM NPHI GR
-> NEW "CURRENT TOOLS" :
SP LLS LLD MNOR
T#
- ADD GR
-> NEW "CURRENT-TOOLS"
SP GR LLD LLS MNOR
T#
- <CR>
-> SUBCOMMAND = ?

```

7) Définition d'un sous-ensemble de niveau du puits

Elle se fait à l'aide de la sous-commande SETDEF. Cette sous-commande ne définit pas véritablement un sous-ensemble quelconque, elle permet seulement de choisir un sous-ensemble prédéfini sous forme d'une composante du "Current GDT-well" (composante de type "set"). La création d'une composante "set" se fait à l'aide de la sous-commande SETBLD (pour SET Build).

A l'appel de la sous-commande SETDEF, le système demande à l'utilisateur quel ensemble il veut choisir avec la question

```
CHOSEN SET = ?
```

Si l'utilisateur désire travailler sur un sous-ensemble de niveaux, il doit répondre par le nom de la composante "set" le définissant.

```
-> SUBCOMMAND = ?
- SETDEF
-> CHOSEN SET = ?
- SSET
-> SUBCOMMAND = ?.
```

8) L'analyse faciologique

L'utilisateur doit taper le nom de sous-commande "ANADIA".

Le système rappelle de nouveau :

```
- le "current GDT-well" et le "Current-Tools"
- le "current GDT-model"
- le "Working-Tools" : les composantes de travail.
```

```
Ex. : -> SUBCOMMAND = ?
- ANADIA
-> CURRENT GDT-WEL : WELL DT1
```

```
"CURRENT TOOLS" :
SP GR LLD LLS MNOR
```

```
"WORKING TOOLS" :
SP GR LLD LLS (si MNOR n'est pas présent
A# sur le modèle)
```

A ce moment, le système attend une sous-commande dont quatre sont particulières à la sous-commande ANADIA :

RUN , VISU , INCLUD et SMTH.

Toutes les sous-commandes accessibles au niveau "SUBCOMMAND = ?" le sont également à ce niveau. (On désignera par la suite ce niveau par le niveau "A#")

Si l'utilisateur est satisfait des composantes de travail, il peut lancer l'exécution des calculs de probabilités avec l'emploi de la sous-commande RUN.

Le système effectue alors les calculs sans rien éditer. Il répond "COMPUTING PROBABILITIES TERMINATED" quand les calculs sont finis et revient au niveau A#.

Exemple synthétique de définitions de données :

```
-> SUBCOMMAND = ?
- TOOLS
-> CURRENT GDT-WELL = ?
- WELL.DT1
-> "CURRENT-TOOLS" = "DATA-TOOLS"
SP GR NPHI LLD LLS MNOR
T#
```

```

- REM NPHI GR
-> NEW "CURRENT-TOOLS" =
    SP LLD LLS MNOR
    T#

- <CR>
-> SUBCOMMAND = ?
- SETDEF
-> CHOSEN SET = ?
- SSET
-> SUBCOMMAND = ?
- MODDEF
-> CURRENT GDT-MODEL = ?
- MODEL.DT
-> SUBCOMMAND = ?

```

9) le lissage des résultats

Il se peut que les résultats obtenus soient irréguliers, c' est à dire que le faciès le plus probable ne présente pas de continuité : par exemple, aucun banc d'un même faciès plus épais que deux ou trois points de mesure du puits.

exemple :

```

ARGILES
CALCAIRES
CALCAIRES
ARGILES
CALCAIRES
ARGILES
ARGILES

```

L' utilisateur peut alors essayer de trouver le faciès moyen sur une séquence au moyen de la sous-commande SMTH de ANADIA. Cette sous-commande ne fait pas en fait un lissage directement sur les faciès les plus probables, ce qui serait difficile car ce sont des données alpha-numériques peu appropriées à ce genre de manipulations. Elle fait un lissage sur les probabilités de chaque niveau.

La probabilité d'un niveau du puits pour un faciès est recalculée en faisant la moyenne arithmétique des probabilités pour ce même faciès. Puis le faciès le plus probable pour chaque niveau du puits est recalculé à partir de ces nouvelles probabilités.

exemple :

	proba. du FACIES 1	proba. du FACIES 2	proba. du FACIES 3	faciès le plus probable
	.7	.2	.1	FACIES 1
	.2	.1	.7	FACIES 3
	.3	.3	.4	FACIES 3
<hr/>				
moyenne des probabilités	.3	.2	.5	FACIES 3

La moyenne calculée est une moyenne glissante : elle est faite uniquement sur quelques niveaux entourant le niveau traité. On appelle "fenêtre" l'ensemble des niveaux à partir desquelles est calculée cette moyenne.

A l'entrée de la sous-commande le système s'assure qu'aucun lissage sur les résultats présents n'a encore été fait. Si un lissage a déjà été effectué, le système avertit l'utilisateur mais en lui laissant la possibilité de refaire un nouveau lissage à partir des résultats lissés :

```
WARNING : RESULTS YET SMOOTHED  
DO YOU REALLY WANT TO SMOOTH (Y/N) ?
```

Si l'utilisateur répond "Y" à cette question, un nouveau lissage est effectué.

Ensuite le système demande la longueur de la fenêtre de lissage, c'est à dire le nombre de niveaux sur lesquels se fera la moyenne glissante. Ce nombre doit être un nombre (positif) impair.

```
NUMBER OF NODES ? (nombre de noeuds)
```

Une fois que l'utilisateur a entré ce nombre de niveaux, le système effectue le lissage sur l'ensemble des résultats puis édite le message suivant :

```
SMOOTHING TERMINATED
```

et revient au niveau "A#".

10) La visualisation des résultats .

Elle se fait au moyen de la sous-commande "VISU" de ANADIA. C'est-à-dire qu'en réponse au témoin d'attente A# édité par la sous-commande ANADIA, l'utilisateur doit taper le nom de sous-commande "VISU".

La sous-commande VISU permet la visualisation des résultats sous une forme invariable : Pour chaque niveau du puits, les probabilités d'appartenance à tous les faciès sont éditées sur la même ligne ainsi que le faciès le plus probable.

Ex.: Si l'on a trois faciès codés sous les noms A (Argile), C (Calcaire) et G (Grès), on aura une édition de la forme :

profondeur	faciès	A	C	G
2 000	A	0.90	0.06	0.04
2 001	A	0.70	0.25	0.05
2 002	C	0.45	0.50	0.05

La sous-commande VISU fonctionne d'une manière semblable à un éditeur de texte.

Au moyen d'instructions, l'utilisateur peut se déplacer d'un niveau du puits à un autre, éditer les résultats correspondant à une suite de niveaux, sauver ces résultats dans un fichier formaté, changer le nom de faciès attribué à un ou à une suite de niveaux, etc...

Exemple d'utilisation simplifiée

```
-> A#
- VISU
-> V#
- TITLE
-> COORX      FACIES      A      C      G
  V#
- L 1 5
  2000      A      0.90      0.06      0.04
      2001      A      0.70      0.25      0.05
      2002      C      0.45      0.50      0.05
      2003      C      0.30      0.65      0.05
      2004      C      0.02      0.80      0.18
  V#
- <CR>
-> A#
```

Selon le principe du système GEOL, l'entrée d'un "carriage return" fait remonter le programme d'un niveau. Ici cela permet de sortir de la sous-commande TOOLS.

11) La sauvegarde des résultats

Au moyen de la sous-commande VISU, l'utilisateur, on l'a vu, pouvait sauvegarder dans un fichier séquentiel ces résultats. Au moyen de la sous-commande INCLUD de ANADIA, il peut inclure ces résultats à l'intérieur du fichier GDT-well, toujours en mémoire centrale. Les résultats sont ainsi une fois pour toutes attachés au fichier de données. Au moyen de la sous-commande EDWELL, l'utilisateur pourra toujours éditer dans des fichiers séquentiels les résultats stockés dans le fichier de données.

Les résultats peuvent être stockés sous forme de faciès ou sous forme de probabilités ou les deux à la fois. C'est à dire, que pour un niveau du puits, l'utilisateur pourra ne conserver que le faciès le plus probable (qui sera alors considéré comme le faciès attribué à ce niveau), ou qu'un nombre de faciès les plus probables. Sinon, pour chaque niveau, il pourra conserver les probabilités que ce niveau appartienne à un faciès ou à un autre.

a) Les options de sauvegarde

A l'appel de la sous-commande INCLUD, le système pose la question
RESULTS = ?
demandant quel type de résultats l'utilisateur désire sauvegarder.

L'utilisateur doit répondre :

"PROB" s'il veut sauvegarder le faciès le plus probable ainsi que toutes les probabilités. (les mêmes résultats que ceux présentés par VISU)

F'num' (ou num est un entier) s'il veut sauvegarder les "num" faciès les plus probables dans l'ordre des probabilités décroissantes.

FP'num' s'il veut sauvegarder les num faciès les plus probables, toujours dans l'ordre des probabilités décroissantes avec en plus leurs probabilités.

Remarque : 'num' est un entier qui ne doit pas être détaché du sigle "F" ou "FP"

Exemple : -> A#
 - INCLUD
 -> RESULTS = ?
 - FP3

b) Mode de sauvegarde

Les résultats sont sauvegardés dans des composantes du "Current GDT-well". Ces composantes doivent obligatoirement avoir un nom qui les définisse et qui permette de les rappeler lors de l'édition d'un GDT-well.

Le système demande à l'utilisateur s'il veut les définir lui-même ou s'il veut que l'ordinateur s'en charge.

DEFINE COMPONENTS' NAMES ? (Y/N)

Si l'utilisateur ne veut pas les définir lui-même, il doit répondre "N". Le système charge alors les résultats définis précédemment, puis revient au niveau "A#"

Si l'utilisateur répond "Y", le système demandera un nom et un titre pour chaque composante à sauvegarder

Exemples :

Si l'utilisateur a répondu FPI à la question "RESULTS = ?"

- > DEFINE COMPONENTS' NAMES ? (Y/N)
- Y
- > NAME FOR "FACIES 1" COMPONENT = ?
- FACI <----- nom de composante = 6 caractères non blancs
commencant par une lettre.
- > TITLE = ?
- FACIES LE PLUS PROBABLE <---- le titre peut contenir jusqu'a
72 caractères quelconques
- > NAME FOR "PROBABILITY 1" COMPONENT = ?
- PROB1
- > TITLE = ?
- PROBABILITE DU FACIES LE PLUS PROBABLE
- > A#

Si l'utilisateur a répondu PROB à la question "RESULTS = ?"

- > DEFINE COMPONENTS' NAMES ? (Y/N)
- Y
- > NAME FOR 'A ' PROBABILITY COMPONENT = ?
- APROB
- > TITLE = ?
- PROBABILITE QUE LE NIVEAU SOIT FAIT D'ARGILE
- > NAME FOR 'C ...' PROBABILITY COMPONENT = ?
- CPROB
- > TITLE = ?
- PROBABILITE QUE LE NIVEAU SOIT FAIT DE CALCAIRE
- > NAME FOR 'G ...' PROBABILITY COMPONENT = ?
- GPROB
- > TITLE = ?
- PROBABILITE QUE LE NIVEAU SOIT FAIT DE GRES
- > NAME FOR MOST LIKELY FACIES = ?
- FACIES
- > TITLE = ?
- FACIES LE PLUS PROBABLE
- > A#

Remarque :

Pour connaître les noms et titres donnés par le système aux composantes sauvegardées automatiquement, se référer au chapitre XI.

12) Conclusion

a) Vérification

Les composantes résultats introduites dans le "Current GDT-well" peuvent être vérifiées à l'aide de la sous-commande EDWELL (éditeur de données d'un puits).

Exemple :

Si l'utilisateur a répondu FPI à la question "RESULTS = ?"

```
-> A#
- EDWELL
-> #
- TITLE

-> COORX  SP  GR  NPHI  LLD  LLS  MNOR  FACI  PROBI
#
- L 1 5
-> 2000.  -  -  -  -  -  -  A  0.90
    2001.  -  -  -  -  -  -  A  0.70
    2002.  -  -  -  -  -  -  C  0.50
    2003.  -  -  -  -  -  -  C  0.65
    2004.  -  -  -  -  -  -  C  0.80
#
- <CR>  <---- touche "Carriage Return"
-> A#
```

b) Sortie de la commande DIA

On sort de la sous-commande ANADIA (marquée par le témoin d'attente A#) en tapant de nouveau la touche "Carriage Return".

```
-> A#
- <CR>
-> SUBCOMMAND = ?
```

De même, à ce niveau, il suffit de taper sur la touche <CR> pour sortir de la commande. Mais il faut auparavant sauver le "Current GDT-well" avec ses résultats dans un fichier sur disque. Ceci se fait à l'aide de la sous-commande WELLSAV.

Si l'utilisateur veut sauver son "Current GDT-well" sous le même nom que celui qu'il possède dans la commande et par le fait écraser l'ancien fichier sur disque, il doit entrer la sous-commande "WELLSAV" sans rien derrière. S'il veut sauver le "Current GDT-well" dans un autre fichier, il doit faire suivre un nom de fichier au nom de sous-commande WELLSAV.

```

Ex. :      (SUBCOMMAND = ?
1er cas   (
           (WELLSAV

           (SUBCOMMAND = ?
2e cas    (
           (WELLSAV NEW-WELL.DT1

```

Dans les deux cas, le système demande si l'utilisateur veut changer le titre de ce nouveau GDT-well :

```
NEW TITLE ? (Y/N)
```

Si l'utilisateur répond "Y" , il doit entrer un nouveau titre de GDT-well (72 caractères maximum en format libre)

```

Ex. :      -> NEW TITLE ? (Y/N)
           - Y
           -> TITLE ?
           - PUIITS CONTENANT RESULTATS OBTENUS AVEC MODEL.DT
           -> SUBCOMMAND = ?

```

Remarque :

Si l'utilisateur tente de sortir de la commande sans avoir sauvé les derniers résultats obtenus dans le "Current GDT-well" ou sans avoir sauvé ce GDT-well sur disque, le système le lui rappelle en lui laissant la possibilité de le faire à ce moment :

```

Ex. :      -> SUBCOMMAND = ?
           - <CR>

           -> INCLUDE CURRENT RESULTS ? (Y/N)
ou
           SAVE CURRENT GDT-WELL ? (Y/N)

```

13) Exemple concret d'analyse diagraphique

```

-> SUBCOMMAND = ?
- WELLDEF
-> CURRENT GDT-WELL = ?
- WELL.DT1
-> SUBCOMMAND = ?
- SETDEF
-> CHOSEN SET = ?
- SSET
-> SUBCOMMAND = ?

```

```

- TOOLS
-> CURRENT GDT-WELL : WELL.DT1
"CURRENT-TOOLS" = "DATA-TOOLS" :
LLS LLD SP GR
T#
- REM SP
-> NEW "CURRENT-TOOLS" :
LLS LLD GR
T#
- <CR>
-> SUBCOMMAND = ?
- MODDEF
-> CURRENT GDT-MODEL = ?
- MODEL.DT
-> SUBCOMMAND = ?
- ANADIA
-> CURRENT GDT-WELL : WELL.DT1
CURRENT GDT-MODEL : MODEL.DT
"CURRENT-TOOLS" :
LLD LLS GR
"WORKING TOOLS" :
LLD GR (en supposant que le modèle ne
A# contient pas la composante LLS)
- RUN
<---- ici se passe un temps de calcul
plus ou moins long

-> COMPUTING PROBABILITIES TERMINATED
A#
- VISU
-> V#
- TITLE
-> COORX FACIES A C G
V#
- L 1 5
-> 2000 A 0.9 0.06 0.04
2001 A 0.7 0.25 0.05
2002 C 0.45 0.50 0.05
2003 C 0.30 0.65 0.05
2004 C 0.02 0.80 0.18
V#
- <CR>
-> A#
- INCLUD FPI
-> DEFINE COMPONENTS' NAMES ? (Y/N)
- Y
-> NAME FOR "FACIES 1" COMPONENT = ?
- FAC1
-> TITLE = ?
- FACIES LE PLUS PROBABLE
- NAME FOR "PROBABILITY 1" COMPONENT = ?

```

- PROBI
-> TITLE = ?
- PROBABILITE DU FACIES LE PLUS PROBABLE
-> A#
- <CR>
-> SUBCOMMAND = ?
- WELLSAV
-> NEW TITLE ? (Y/N)
- N
-> SUBCOMMAND = ?

VII) CONSTRUCTION D'UN MODELE

1) Introduction

Dans cette section, nous allons présenter la démarche à suivre dans le cas de la construction d'un modèle à partir des données d'un puits contenu dans un GDT-well. Pour avoir plus de détails sur toutes les commandes abordées dans cette section, se rapporter à la section X.

2) Résumé des opérations à effectuer

On suppose que l'utilisateur vient de lancer la commande, il faut d'abord définir un tableau de données.

Dans une première phase, le système va, à partir des points de données, calculer des points caractéristiques appelés noyaux ("KERNELS" dans le programme). (cf. I')

Dans une deuxième phase, il va présenter un à un ces noyaux à l'utilisateur pour que ce dernier leur attribue un nom de faciès.

Dans une dernière phase, la création proprement dite du modèle étant terminée, le système laisse à l'utilisateur la possibilité de le vérifier et de le modifier. Cette dernière phase est en fait l'utilisation de la sous-commande d'édition de modèles.

Enfin, l'utilisateur pourra sauvegarder ce nouveau modèle sur disque magnétique en vue d'une utilisation ultérieure.

3) Liste des sous-commandes abordées dans cette section

sous-commande	:	paragraphe où elle est abordée
WELLDEF	:	VII 4)
TOOLS	:	VII 5)
SETDEF	:	VII 6)
MODBLD	:	VII 7)
EDMOD	:	VII 10)
MODSAV	:	VII 11)
	:	

4) Chargement des données

Il se fait au moyen de la sous-commande WELLDEF.
Cf. VI 4)

5) Définition des composantes

Les composantes de travail seront celles contenues dans le "Current-Tools". Lors du chargement du "Current GDT-well" le "Current-Tools" est rendu égal au "Data-Tools". Donc, si l'utilisateur veut travailler avec toutes les composantes du GDT-well il peut immédiatement passer à la suite. Sinon il doit appeler la commande "TOOLS" qui permet de modifier le "Current-Tools". (Cf. VI et X)

6) Définition d'un sous-ensemble de niveaux du puits

Ceci se fait au moyen de la sous-commande SETDEF
(Cf. VI 7)

7) Vérifications préliminaires et définition du nom du nouveau GDT-model

La sous-commande MODBLD effectue la construction proprement dite d'un modèle : (MODEL BUILD)

Le système s'assure à l'entrée de cette sous-commande de deux choses :

- s'il existe déjà en mémoire centrale un modèle récemment créé ou modifié, auquel cas il demande à l'utilisateur s'il veut le sauvegarder sur disque. En effet, il n'est prévu de place en mémoire que pour un seul modèle : le "Current GDT-model". Il demande ensuite à l'utilisateur le nom et éventuellement le titre du nouveau GDT-model à créer.

- s'il existe un GDT-well en mémoire centrale. Si aucun GDT-well n'a été encore chargé, alors le système pose la question
CURRENT GDT-WELL = ?

Remarque :

Si l'utilisateur définit ainsi un "Current GDT-well", il ne peut cependant pas définir de la même façon un sous-ensemble de composantes du "Data-Tools" ni un sous-ensemble de niveaux du "Current GDT-well".

Ex. : -> SUBCOMMAND = ?
 - MODBLD
 -> SAVE CURRENT GDT-MODEL ? (Y/N)
 - Y
 -> NEW GDT-MODEL NAME = ?
 - NEWMOD.DT
 -> TITLE (Y/N) ?
 - Y
 -> TITLE = ?
 - titre du nouveau GDT-model

 -> CURRENT GDT-WELL = ?
 - WELL.DT1

8) Création des noyaux

La création des noyaux se fait au moyen de l'algorithme des centres dynamiques. L'utilisateur doit définir plusieurs paramètres nécessaires à cet algorithme.

Avant de commencer les calculs, le programme pose donc la question

OPTIONS LIST = ? à laquelle l'utilisateur doit répondre une ou plusieurs des options suivantes (séparées par des espaces blancs ou des virgules) :

NKER, NIT, METR, NORM, STD, STDE

Les trois premières réponses sont relatives aux trois paramètres

à définir :

- + le nombre de noyaux à créer (NKER)
- + la métrique utilisée (la métrique est la manière de calculer une distance entre deux points) (METR)
- + le nombre d'itérations que doit effectuer l'algorithme (NIT)

Le nombre de noyaux :

C'est un paramètre important ; il est conseillé à l'utilisateur de le choisir lui-même à chaque création de modèle en fonction du nombre de niveaux du puits et du nombre approximatif de faciès qu'il a l'intention de définir. Remarque : le nombre de noyaux obtenus est parfois légèrement inférieur au nombre demandé, en particulier quand le nombre de données et le nombre de noyaux demandés sont grands.

La métrique :

l'utilisateur a le choix entre trois options :

- la métrique unité
- la métrique de réduction
- la métrique dite de "Mahalanobis"

La métrique de réduction est la plus intéressante. Comme son nom l'indique, elle réduit les données. C'est celle qui est employée par défaut. La métrique unité est déconseillée bien que ce soit celle qui engendre les calculs les moins longs, car elle ne réduit pas les données. Elle est à écarter des que les valeurs prises par les composantes ont des ordres de grandeur très différents. La métrique de Mahalanobis est la plus précise de toutes, mais elle risque d'engendrer de plus longs calculs.

Le nombre d'itérations:

Il est fixé initialement à deux. Il est peu utile de le changer. Toutefois, il convient de l'augmenter si le nombre de noyaux est inférieur à deux pour cent environ du nombre d'échantillons du puits (ou du sous-ensemble choisi)

En conclusion, seul le nombre de noyaux à définir est important et sera le seul paramètre à changer souvent.

Si l'utilisateur ne définit pas un de ces trois paramètres, alors des valeurs standard sont prises par le système. On appelle "valeur standard" la valeur qu'avait un paramètre lors du dernier passage de l'utilisateur dans la même sous-commande et pendant la même session. Si l'utilisateur rentre pour la première fois dans la sous-commande, alors cette valeur standard est prise égale à une valeur fixée par le programme appelée "valeur normalisée". Les trois valeurs normalisées sont :

nombre de noyaux = 50
métrique = métrique "de réduction"
nombre d'itérations = 2

L'option NORM sert à choisir ces trois valeurs normalisées pour la suite des calculs. L'option STD sert à choisir les trois valeurs standards. L'option STDE sert à faire éditer par le système les trois valeurs standards du moment.

Syntaxe

En réponse à la question "OPTION LIST = ?", l'entrée au clavier des options NKER, METR ou NIT entraîne une question du système pour chaque option entrée.

NKER ==> NUMBER OF KERNELS = ?
METR ==> METRIC = ?
NIT ==> NUMBER OF ITERATIONS = ?

Ex : -> OPTION LIST = ?
- NKER, NIT
-> NUMBER OF KERNELS = ?
- 50
-> NUMBER OF ITERATIONS = ?
- 3

La question METRIC = ? réclame quant à elle une réponse sous forme d'option ; à cette question, il faut répondre

UNIT si l'on désire la métrique unité
VAR si l'on désire la métrique de réduction
COV si l'on désire la métrique de Mahalanobis

Ex. : -> OPTION LIST = ?
 - NKER,METR
 -> NUMBER OF KERNELS = ?
 - 50
 -> METRIC = ?
 - COV
 -> début des calculs ...

A la fin des calculs, le système donne le nombre de noyaux trouvés et les édite tous en précisant leur "profondeur" (voir plus loin) et le pourcentage de points de données qu'ils regroupent autour d'eux.

Ex. :

NUMBER OF KERNELS					
47					
NO	DEPTH	%	LLS	GR	NPHI
1	2000.3	.5	4.5	60.0	24.5
2	3303.5	.2	10.3	48.2	28.3
3	2540.1	.4	100.3	14.0	15.5
4
5	etc				

9) Attribution d'un nom de faciès aux noyaux

Ce paragraphe décrit la deuxième phase de la construction d'un modèle où l'utilisateur (un expert en diagraphies de préférence) doit décider quel nom de faciès il doit attribuer à chaque noyau.

Le système présente chaque noyau à l'utilisateur d'une façon numérique. Si l'utilisateur le désire, le système peut présenter certains graphiques afin d'aider l'utilisateur dans sa décision. Enfin, il existe la possibilité de faire une cross-validation du noyau. Cette cross-validation indique le nom de faciès le plus cohérent avec les noyaux déjà déterminés.

a) Présentation numérique d'un noyau.

Un noyau est donc un point de l'espace paramétrique (espace des variables). Il ne correspond cependant pas à un point de mesure du puits.

Exemple dans un espace à deux dimensions :

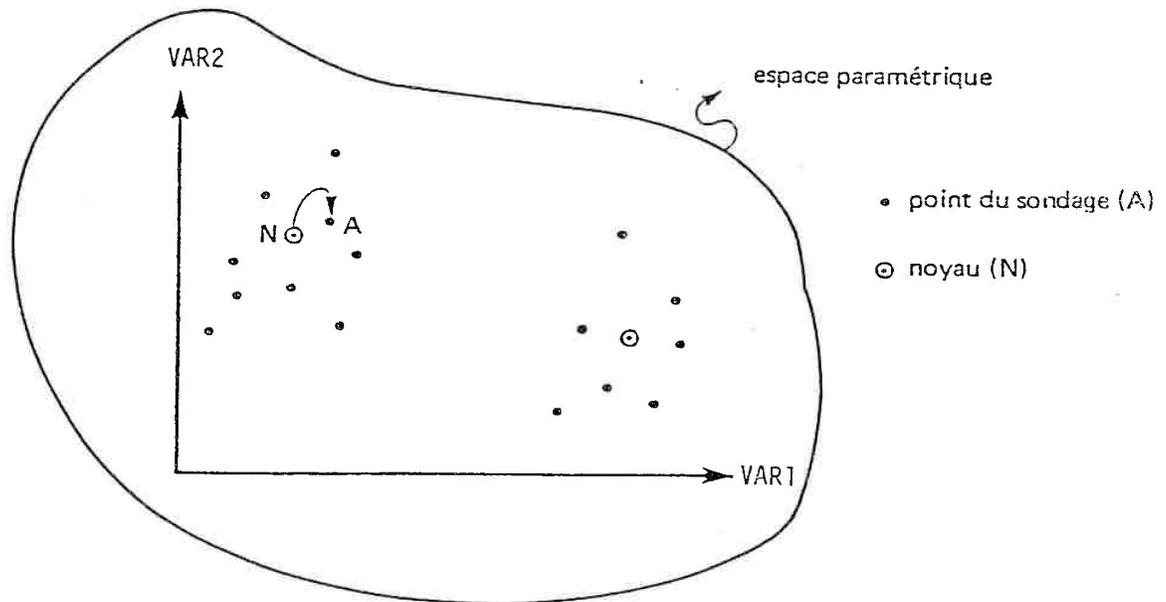


Figure 38

Pour présenter un noyau, le système utilise deux sources d'informations :

- les coordonnées du noyau dans l'espace paramétrique,
- la profondeur du point de mesure du sondage le plus proche du noyau dans cet espace.

A titre indicatif, il donne aussi le pourcentage de points de la classe associée à ce noyau par rapport au nombre total de points de sondage.

Ex. : -> KERNEL NO : 3
 PERCENTAGE OF POINTS AROUND IT : 8.7

DEPTH	SP	GR	NPHI
2000.0	-	-	-

A la suite de cette présentation, le système demande à l'utilisateur de donner un nom de faciès à ce noyau :

FACIES NAME = ?

b) Présentation graphique

Le système peut allier à cette présentation numérique une présentation graphique de ces points. C'est ce à quoi sert l'option DISP (pour DISplay)

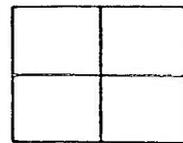
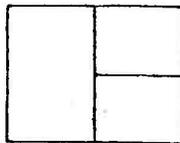
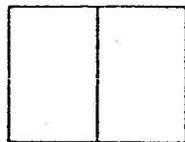
En réponse à la question

FACIES NAME = ?

trois types de dessin sont possibles :

- + dessin de courbes diagraphiques
- + cross-plots des points (noyaux) du modèle
- + projection des points du modèle sur un plan discriminant au mieux les points de même faciès.

Ces types de dessin peuvent être dessinés simultanément sur l'écran jusqu'au nombre de 4. L'écran est alors partagé en deux, en trois ou en quatre.



* dessin de courbes diagraphiques

On utilise l'information apportée par le point du sondage le plus proche du noyau.

Une portion de courbes diagraphiques aux alentours de ce point est dessinée à l'écran. Les courbes diagraphiques sont choisies par l'utilisateur. (cf. figure 39)

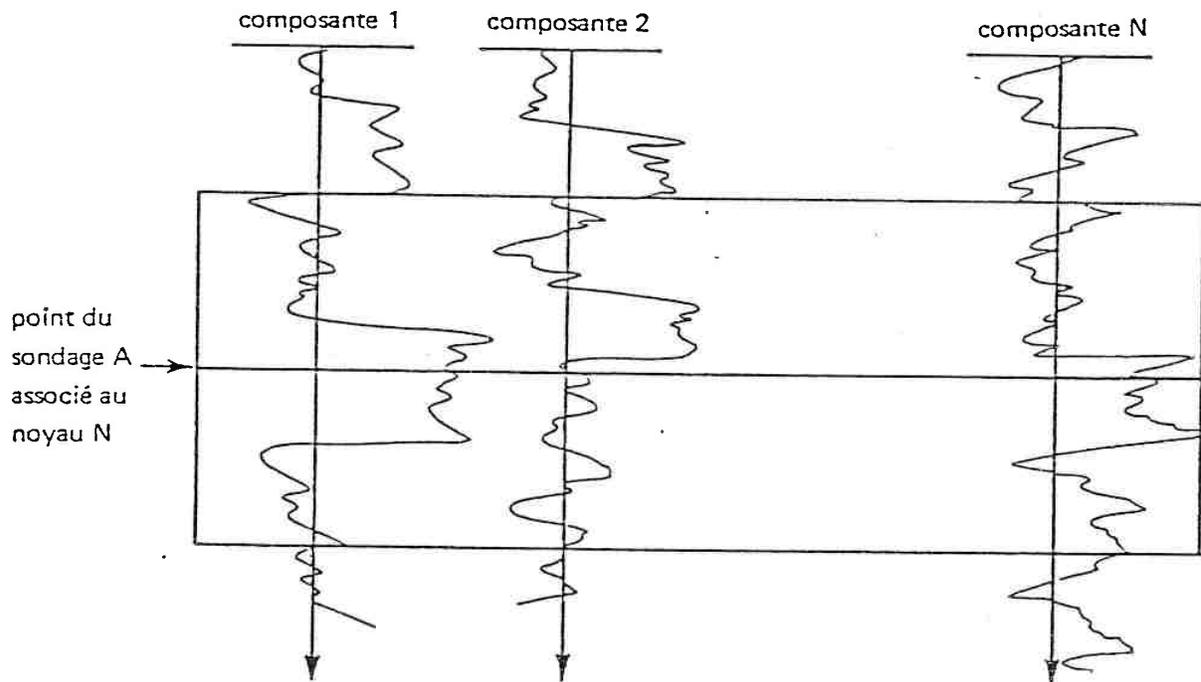


Figure 39. Visualisation des courbes diagraphiques

* cross-plots

Les noyaux étant des points de l'espace paramétrique, ils peuvent donc être représentés en tant que tels. L'espace paramétrique étant en général de dimension supérieure à 2 et même 3, la seule façon de les représenter est la projection sur un plan.

Si le plan est engendré par deux variables, les noyaux sont représentés d'une façon familière. Dans le domaine des diagraphies, on parle de cross-plot.

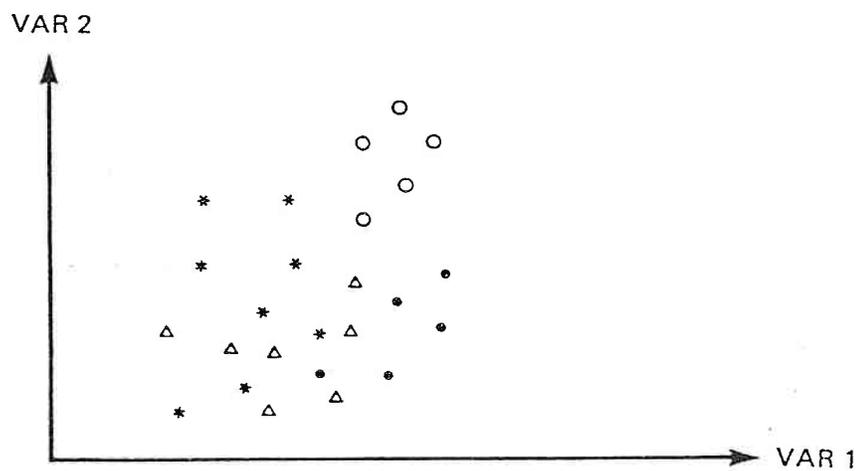


Figure 40. Exemple de cross-plot

Il arrive que certaines variables soient plus discriminantes que d'autres et que l'utilisateur connaisse la répartition des faciès dans l'espace engendré par ces variables.

Exemple :

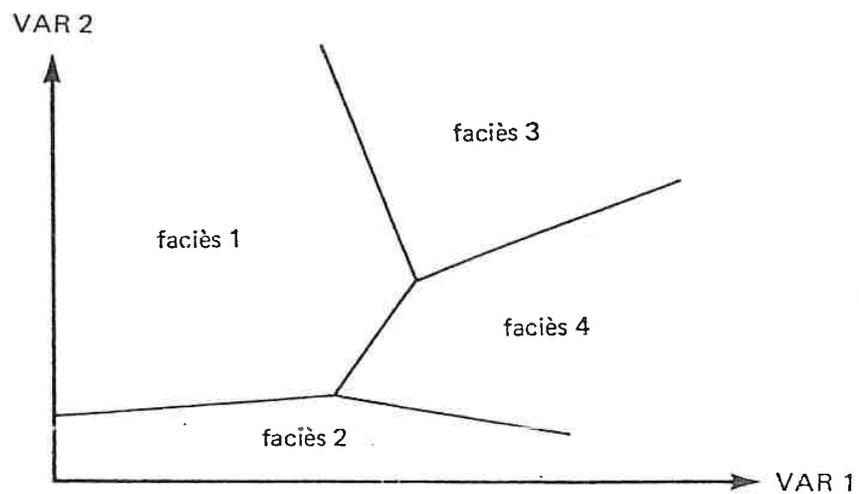


Figure 41. Espace partitionné en zones connues

La représentation graphique des noyaux sur le plan engendré par ces deux variables permet alors à l'utilisateur de classer un noyau d'un seul coup d'oeil sur le dessin. Ceci est plus rapide et à la fois plus sûr (si l'utilisateur connaît bien la répartition des faciès dans le plan considéré) que de se pencher sur les données numériques que sont les valeurs des composantes pour le noyau à déterminer.

A l'écran, le système dessinera les points avec une couleur différente suivant le faciès qui leur a été attribué. Les noyaux qui ne seront pas encore déterminés seront dessinés d'une couleur différente de toutes les autres. Le noyau à déterminer sera lui aussi d'une couleur encore différente de toutes les autres.

Ces projections permettent en outre à l'utilisateur de vérifier à tout instant qu'il ne s'est pas trompé, voyant que les noyaux affectés d'un même faciès sont regroupés dans une zone limitée du dessin.

* projection sur un plan discriminant

On peut choisir un plan de projection tel que les classes constituées des noyaux de même faciès se distinguent le mieux, regroupées sur elles-mêmes et éloignées les unes des autres. Certaines peuvent se superposer cependant.

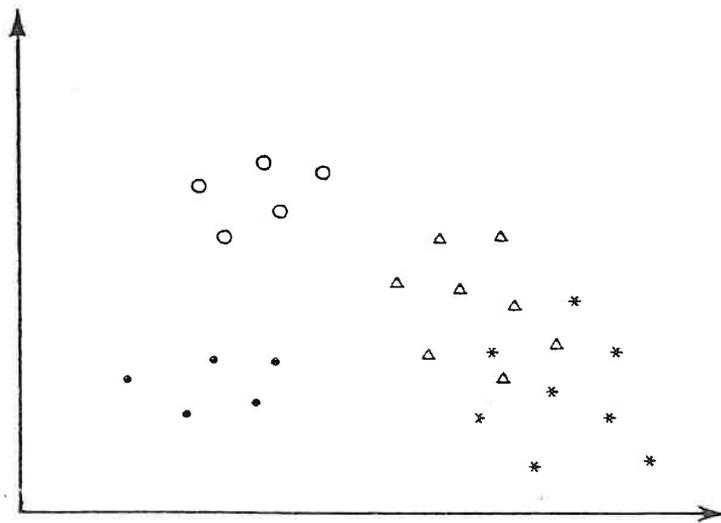


Figure 42. Exemple de projection discriminante

Les points de chaque classe ici distingués par un symbole différent le sont dans le système par une couleur différente. Le noyau courant (celui qui est à déterminer) est lui représenté par une couleur distincte.

Dans un cas semblable à celui de la figure 43, ce type de projection peut aider l'utilisateur à choisir quand il hésite entre deux faciès à attribuer à un noyau et à voir si son modèle est bien cohérent.

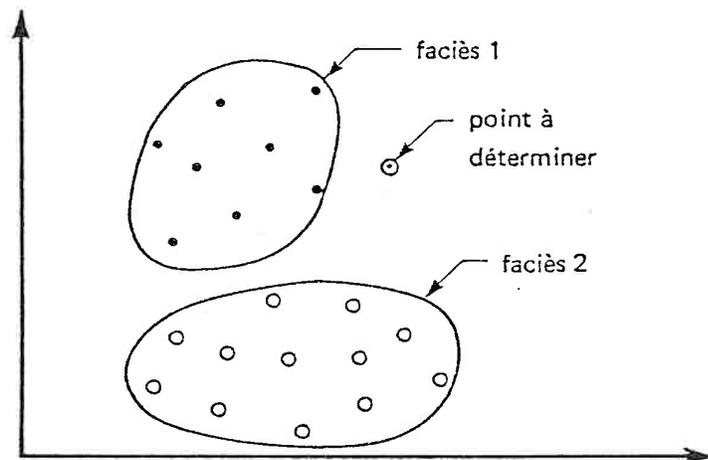


Figure 43

Remarques :

Si trop peu de noyaux ont un nom de faciès ou si ceux qui en ont ont tous le même, alors cette option est interdite car le plan de projection serait mauvais.

A part le dessin des courbes, toutes ces représentations graphiques sont accessibles dans la sous-commande EDMOD.

Comment obtenir ces représentations graphiques :

Par l'option DISP en réponse à la question "FACIES NAME = ?" à n'importe quel moment de la présentation des noyaux.

Le système pose alors la question :

DISPLAY OPTIONS = ?

à laquelle l'utilisateur doit répondre une ou plusieurs des options suivantes CURV, PLOT, PLO2, PLO3, PLO4, DISC, STD, STDE, NIL

* L'option CURV sert à obtenir le tracé des courbes diagraphiques.

Le système demande d'abord le nombre de courbes à tracer
NUMBER OF DISPLAYED CURVES = ?
puis les courbes elles-mêmes :
DISPLAYED CURVES = ?

Les noms des courbes sont en fait des noms de composantes pris dans le CURRENT-TOOLS. L'utilisateur peut les entrer tous sur une même ligne séparés par des espaces blancs ou un par un.

* Les options PLOT, PLO2, PLO3 et PLO4 servent à obtenir à l'écran respectivement un, deux, trois ou quatre cross-plots. Pour chacun des cross-plots, le système demande avec la question

COMPONENTS OF PLOT = ?

le nom des deux composantes du cross-plot. De même, ces deux noms sont pris parmi les noms du CURRENT-TOOLS et peuvent être entrés un par un ou sur la même ligne.

* L'option DISC sert à obtenir la projection des points du modèle sur un plan discriminant.

* L'option STD (pour standard) sert à obtenir les options qui ont été utilisées lors de la dernière visualisation graphique. Cette option s'emploie seule.

* L'option STDE sert à éditer ces options "standard", c'est à dire quelles options ont été utilisées lors de la dernière représentation graphique avec leurs caractéristiques (le nom des composantes). Cette option s'emploie seule également et le système revient ensuite à la question

DISPLAY OPTIONS = ?

* L'option NIL sert à arrêter la représentation graphique.

Dans le cas de l'utilisation des options CURV et d'une option de cross-plots, les questions subsidiaires sont posées par le système toujours dans le même ordre (option CURV d'abord)

Ex. : -> DISPLAY OPTION = ?
- PLO2, CURV
-> NUMBER OF DISPLAYED CURVES = ?
- 3
-> DISPLAYED CURVES (1-3) = ?
- ----
-> COMPONENTS OF PLOT NO 1 = ?
- ----
-> COMPONENTS OF PLOT NO 2 = ?
- ----

Quand la conversation relative aux options graphiques est terminée, le système présente à nouveau le noyau courant, puis pose à nouveau la question "FACIES NAME = ?"

c) Attribution d'un nom de faciès à un noyau

Le système présente les noyaux les uns après les autres en terminant toujours pour chacun d'eux par la question FACIES NAME = ?

L'utilisateur doit entrer un nom de faciès, terme d'au plus six caractères appartenant à une liste prédéfinie.

Ex. :

```
-> KERNEL NO : 1
    PERCENTAGE OF POINTS AROUND IT : 0.5
    DEPTH   LLS   LLD   GR
    3 000.0  8.5   16.5  30.0
    FACIES NAME = ?
- GRES
-> KERNEL NO : 2
    PERCENTAGE OF POINTS AROUND IT : 0.2
    DEPTH   LLS   LLD   GR
    1 530.0  1.0   2.1   60.0
    FACIES NAME = ?
```

Cette liste est en fait contenue dans un fichier que l'on doit créer à l'aide du système propre de l'ordinateur. Elle n'est pas modifiable en cours de session.

L'utilisateur peut taper à la place d'un nom de faciès l'option "NONE" pour indiquer qu'il ne veut pas attribuer de nom de faciès au noyau en question.

Il peut aussi taper l'option "SAME" qui signifie qu'il ne désire pas changer le nom déjà en place. (Cette option est utilisable dans le cas où l'utilisateur est revenu en arrière, sinon elle est équivalente à l'option NONE).

Dans ces trois cas de réponse, le système considère le noyau courant comme classe et présente le noyau suivant.

Si l'utilisateur veut connaître la liste de tous les faciès permis, il doit taper l'option "LITO". Le système édite alors cette liste et présente à nouveau le noyau courant.

```
Ex. :   -> KERNEL NO . 1
        PERCENTAGE OF POINTS AROUND IT : 0.5
        DEPTH   LLS   LLD   GR
        3 000.0  8.5   16.5  30.0
        FACIES NAME ?
        - LITO
        -> LIST OF FACIES NAMES :
            ARGILE
            CALCAI
            GRES
            SILT
            etc ...

        KERNEL NO 1
        PERCENTAGE OF POINTS AROUND IT : 0.5
        DEPTH   LLS   LLD   GR
        3 000.0  8.5   16.5  30.0
        FACIES NAME = ?
```

Enfin l'utilisateur peut entrer l'option "QUIT" qui force le système à abandonner la détermination des noyaux et à quitter la sous-commande MODBLD. Le modèle créé est conservé tel quel et le système entre dans la sous-commande_EDMOD (éditeur de modèles).

d) Options supplémentaires

A la question "FACIES NAME = ?", l'utilisateur peut répondre par certaines instructions de l'éditeur de modèle :

```
LIST
CV et CVR
LCV et LCVR
SMTH
```

Le mode d'emploi de ces sous-commandes sera abordé dans la section suivante qui traite de l'éditeur de modèle.

Les instructions CV et CVR permettent la cross-validation des noyaux d'un modèle. La cross-validation d'un noyau d'un modèle est le calcul des probabilités pour ce noyau d'appartenir à chacune des classes du modèle.

e) Possibilités de développement

- Le système expert.

Il serait souhaitable de pouvoir intégrer à la commande un système expert capable de conseiller l'utilisateur lors de la détermination d'un noyau.

Certains moteurs d'inférences (programmes systèmes experts) peuvent appeler des sous-programmes classiques. Par contre, l'inverse qui consiste dans l'appel d'un système expert à l'intérieur d'un système classique pose certains problèmes. Les systèmes experts sont souvent écrits dans un langage différent du FORTRAN, mais le problème majeur est la transmission des données : un système expert est un système indépendant, lisant les données qu'il lui faut sous une forme spéciale dans un fichier ou au clavier. Ici il faudrait un système qu'on appellerai pour chaque noyau et auquel on transmettrait les données (ce sont les coordonnées du noyau présenté) sous forme d'arguments.

- Détermination automatique du faciès en prenant comme source d'information le sondage dont on extrait le modèle.

Supposons qu'un sondage ait déjà fait l'objet d'une analyse diagraphique (manuelle ou non) et que l'on veuille en extraire un modèle. Lors de la construction du modèle, le noyau est d'habitude présenté à l'utilisateur avec la profondeur du point du sondage qui lui ressemble le plus. Le système pourrait aller lire dans le fichier contenant les données du sondage le nom du faciès correspondant à la profondeur indiquée.

f) Conclusion

Dans le cas normal, l'utilisateur passe en revue tous les noyaux un par un. Quand il détermine le dernier (soit par un nom de faciès, soit par l'option SAME ou l'option NONE), le système considère la détermination des noyaux comme terminée.

10) Correction du modèle

L'utilisateur a la possibilité cependant de modifier les noms de faciès qu'il vient de taper. Ceci se fait à l'aide de la sous-commande EDMOD qui est appelée automatiquement par le système. Le système édite le signal caractéristique de cette sous-commande :

M#

L'utilisateur a alors accès à toutes les instructions de la sous-commande EDMOD. Deux instructions nous intéressent particulièrement : l'instruction CHANGE (ou C) et l'instruction DEL.

* L'instruction CHANGE permet de changer le nom de faciès d'un ou de plusieurs noyaux consécutifs. Elle permet également de supprimer le nom de faciès de certains noyaux.

Sa syntaxe est : "CHANGE (ou C) n1 n2 faciès" où n1 est le premier noyau et n2 le dernier noyau concernés et "faciès" le nom de faciès attribués aux noyaux compris entre n1 et n2 inclus,

Ex. : l'instruction "CHANGE 1 3 ARGILE" provoque l'attribution du faciès ARGILE aux trois premiers noyaux.

Si le faciès n'est pas présent dans la ligne d'instruction, alors le système pose la question

FACIES NAME = ?

Ex. :
-> M#
- C 12
-> FACIES NAME = ?
- NONE
-> M#

Cette série d'instructions conduit à la suppression du nom de faciès du noyau numéro 12 (s'il en avait un)

* L'instruction DEL (pour "delete") permet de supprimer complètement un ou plusieurs noyaux du modèle courant. Sa syntaxe est "DEL n1 n2"

Ex. : "DEL 12 13" provoque la suppression complète des noyaux 12 et 13.

La numérotation des noyaux est alors rétablie :
le numéro 14 devient ici le numero 12
le numéro 15 devient ici le numero 13
etc ...

Après l'exécution de chacune des instructions, le système réédite le signal "M#"

Remarque :

Lorsque l'utilisateur lance la sous-commande RUN de ANADIA (exécution de l'analyse faciologique), les noyaux du modèle utilise qui n'ont pas de faciès n'entrent pas en jeu. Mais entre deux exécutions, l'utilisateur peut leur réattribuer un nom de faciès grâce à la sous-commande EDMOD, ce qu'il n'aurait pas pu faire s'ils avaient été complètement détruits.

11) Sauvegarde du modèle

Pour sortir de l'éditeur de modèle, l'utilisateur n'a qu'à appuyer sur la touche "carriage return". Le système revient alors au niveau "SUBCOMMAND = ?" (si l'utilisateur a tape la touche <CR> par erreur, il peut revenir dans la sous-commande EDMOD en tapant à ce niveau "EDMOD").

Le modèle ainsi créé est utilisable tout de suite pour la sous-commande RUN de ANADIA. Pour que ce modèle puisse servir lors d'une prochaine session, l'utilisateur doit le sauvegarder dans un fichier (fichier GDT). Il doit se servir pour cela de la sous-commande MODSAV en tapant "MODSAV" en réponse à la question "SUBCOMMAND = ?"

12) Exemple de construction de modèle

```
- DIA
-> SUBCOMMAND = ?
- TOOLS
-> CURRENT GDT-WELL = ?
- WELL .DTI
-> "CURRENT-TOOLS" = "DATA TOOLS" :
  LLD  LLS  SP  .GR
  T#
- REM SP
-> NEW "CURRENT-TOOLS" :
  LLD  LLS  GR
  T#
- <CR>
-> SUBCOMMAND = ?
- MODBLD
-> NEW GDT-MODEL = ?
- MODEL.DT
-> TITLE DEFINITION ? (Y/N)
- Y
-> TITLE = ?
- titre du nouveau modèle
-> OPTION LIST = ?
```

- NKER
 -> NUMBER OF KERNELS = ?
 - 30
 -> NUMBER OF KERNELS
 30

NO	%	DEPTH	LLS	LLD	GR
1	3.5	3000.0	8.5	16.5	30.0
2	2.2	1530.0	1.0	2.1	60.0
etc
.
.
30	3.0	-----	---	---	---

ANOTHER ITERATION (Y/N) ?

- N

-> KERNEL NO 1
 PERCENTAGE OF POINTS AROUND IT : 3.5
 DEPTH LLS LLD GR
 3000.0 8.5 16.5 30.0
 FACIES NAME = ?

- DISP
 -> DISPLAY OPTIONS = ?
 - CURV, PLO2, DISC
 -> NUMBER OF DISPLAYED CURVES = ?
 - 3

-> DISPLAYED CURVES = ?
 - LLS LLD GR
 -> COMPONENTS OF PLOT NO 1 = ?
 - LLS GR
 -> COMPONENTS OF PLOT NO 2 = ?
 - LLD GR

KERNEL NO 1
 PERCENTAGE OF POINTS AROUND IT : 3.5
 -> visualisation à l'écran du dessin des courbes
 et des cross-plots

DEPTH	LLS	LLD	GR
3000.0	8.5	16.5	30.0

FACIES NAME = ?

- GRES
 -> KERNEL NO 2
 PERCENTAGE OF POINTS AROUND IT : 2.2
 DEPTH LLS LLD GR
 1530.0 1.0 2.1 60.0
 FACIES NAME = ?

..... présentation de tous les noyaux
 jusqu'au dernier.

-> KERNEL NO 30
PERCENTAGE OF POINTS AROUND IT : 3.0
DEPTH LLS LLD GR
2520.0 18.5 24.5 24.0
FACIES NAME = ?
- GRES
-> M#
- C 1 2
-> FACIES NAME = ?
- SILTS
-> M#
- DEL 13
-> M#
- <CR>
-> SUBCOMMAND = ?
- MODSAV
-> SUBCOMMAND = ?

VIII) EDITEUR DE MODELES

L'éditeur de modèles est un niveau du programme (niveau directement inférieur au niveau "SUBCOMMAND = ?") où le système attend une instruction de l'utilisateur en éditant le terrain d'attente M#. Il peut être appelé depuis les niveaux "SUBCOMMAND = ?" et "A#".

Ex. : la suite d'instructions :

```
-> SUBCOMMAND = ?  
- ANADIA  
-> A#  
- EDMOD  
-> M#
```

a le même effet que la suite d'instructions :

```
-> SUBCOMMAND = ?  
- EDMOD  
-> M#
```

Une instruction est une commande de l'utilisateur contenant en elle-même toutes les informations nécessaires à son exécution. En cas d'erreur ou d'instruction incomplète, le système pose parfois en plus certaines questions.

Les instructions de EDMOD sont : LIST, TITLE, CV, LCV, COEF, CHANGE, LITO, DEL, SAVE, DISP.

Par ces instructions, l'éditeur de modèle permet :

- l'édition des noyaux du modèle (numérique, graphique)
- la cross-validation des noyaux
- la modification (d'une façon restreinte) de ces noyaux.

1) La syntaxe des instructions et la notion de noyau courant.

La plupart des instructions ont une action sur les noyaux du modèle. Pour toutes les instructions de ce type, les noyaux concernés sont inscrits dans la ligne d'instruction de la même manière :

Instruction n1 n2

ou "Instruction" est le nom de l'instruction (par ex. LIST) n1 et n2 sont respectivement les numéros d'ordre dans le modèle du premier et du dernier noyau concerné. Tous les noyaux de numéros compris entre n1 et n2 (n1 et n2 inclus) sont concernés.

Ex. : "LIST 1 3" provoque l'édition des trois premiers noyaux du modèle.

Le noyau courant est le dernier noyau qui a été traité par une instruction : après "LIST 1 3" , c'est donc le noyau n0 3.

2) Entrée dans la sous-commande

A l'entrée dans la sous-commande, le système précise quel est le "Current GDT-model" (celui sur lequel l'utilisateur va travailler).

```
Ex. : -> SUBCOMMAND = ?  
      - EDMOD  
      -> CURRENT GDT-MODEL : MODEL.DT.
```

S'il n'y a pas de "Current GDT-model", le système appelle automatiquement la sous-commande MODDEF.

3) Les composantes de travail.

L'éditeur de modèles ne permet pas à l'utilisateur de choisir ses composantes de travail. Si le modèle est seul en mémoire, c'est à dire s'il n'y a pas de GDT-well défini, toutes les composantes du modèle sont prises en compte.

Par contre, s'il existe un GDT-well en mémoire, alors les composantes prises en compte seront celles du "Working-tools", ensemble de composantes présentes à la fois dans le GDT-model et le "Current-tools".

Contrairement à la sous-commande ANADIA, la sous-commande EDMOD n'édite aucun message concernant les composantes de travail. Cependant, le système prévient l'utilisateur s'il n'existe pas de GDT-well en mémoire.

WARNING : NO DEFINED GDT-WELL

```
Ex. : -> SUBCOMMAND = ?  
      - EDMOD  
      -> CURRENT GDT MODEL : MODEL.DT  
          WARNING : NO DEFINED GDT-WELL  
          M#
```

4) Edition des noyaux : Les instructions LIST, TITLE et DISP

L'édition d'un ou de plusieurs noyaux se fait avec l'instruction LIST qui a la syntaxe suivante :

```
L          : édition du noyau courant  
L n1       : édition du noyau de numéro n1  
L n1 n2    : édition des noyaux dont les numéros sont compris entre
```

n1 et n2.

L'instruction LIST fait apparaître à l'écran le numéro, le nom et le poids du noyau concerné, puis les coordonnées, c'est-à-dire la valeur des composantes de travail pour ce noyau.

La nature de ce qui est écrit à l'écran peut être affichée sous forme d'un titre au moyen de l'instruction TITLE.

```
Ex. :  -> M#
        - TITLE
        -> NO  FACIES  WGHT  LLS  LLD  GR
          M#
        - L 1 2
          1  GRES    3.5   8.5   16.5  30.0
          2  ARGILE  2.2   1.0   2.1   60.0
          M#
```

Le poids d'un noyau est un nombre compris entre 0 et 100 qui exprime son importance dans le modèle. Le système l'a calculé lors de la construction du modèle. Il correspond au "PERCENTAGE OF POINTS AROUND IT", résultat de l'algorithme des centres dynamiques.

Dans le titre, il est nommé WGHT (pour "WEIGHT" qui veut dire "poids" en anglais).

L'éditeur de modèle possède aussi la possibilité d'éditeurs graphiques :

- cross-plots des points du modèle
- projection des points du modèle sur un plan discriminant au mieux les faciès.

De même que pour la construction d'un modèle, l'utilisateur a droit à au plus 4 dessins simultanés.

La définition des options de dessin se fait de la même façon que dans la sous-commande MODBLD, c'est-à-dire par l'intermédiaire de l'option DISP.

```
Ex. :  -> M#
        - DISP
        -> DISPLAY OPTIONS = ?
        - PLO2, DISC
          etc
          .
          .
```

Pour plus de précisions sur ces options graphiques, se reporter au paragraphe VII 9) b)

Remarques :

- L'option graphique CURV n'est pas utilisable dans le cas de l'édition d'un modèle.
- Dans le cas où le modèle a été créé pendant la session, le système rajoute dans le texte d'édition de chaque noyau une autre donnée intéressante : la profondeur dans le sondage d'origine du point le plus représentatif du noyau considéré, (cf. VII 9)a). L'édition prend alors la forme suivante :

```

- TITLE
-> NO  DEPTH  FACIES  WGHT  LLS  LLD  GR
    M#
- L  1
    1  3000.0  GRES    3.5   8.5  16.5  30.0

```

5) La cross-validation : Les options CV, CVR, LCV, LCVR, COEF

a) Principe de la cross-validation

Pour tout noyau du modèle qu'on considère alors comme un point quelconque de l'espace des variables, il est possible de calculer les probabilités, qu'il appartienne à un faciès ou à un autre.

b) Présentation

Deux types de présentation sont possibles :

- l'option CV entraîne l'édition de toutes les probabilités pour le noyau concerné :

```

Ex. :  -> M#
        - CV 1
        -> KERNEL NO 1
            FACIES NAME : GRES
            FACIES  *  PROBABILITY
            GRES    *  0.85
            ARGILE  *  0.1
            SILTS   *  0.05
            M#

```

- l'option CVR (pour "cross-validation réduite") entraîne l'édition uniquement du faciès calculé comme le plus probable.

```

Ex. :  -> M#
        - CVR 1
        -> FACIES : GRES  PROBA = 0.85

```

Les instructions LCV et LCVR cumulent les effets de l'instruction LIST et respectivement des instructions CV et CVR.

```

Ex. : -> M#
      - LCV 1
      -> 1   GRES   3.5   8.5   16.5   30.0
          FACIES *   PROBABILITY
          GRES   *   0.85
          ARGILE *   0.10
          SILT   *   0.05

```

c) Syntaxe

Elle est la même que pour l'instruction LIST. Exemple :
 L'instruction "LCV 3 5" traite les noyaux numérotés 3, 4 et 5.

d) Intérêt de la cross-validation :

La cross-validation permet de vérifier si le modèle est cohérent. En effet, s'il est cohérent, on doit retrouver par cette méthode le faciès que chaque noyau possède réellement dans le modèle. Ex. : La cross-validation d'un noyau ayant pour nom de faciès "GRES" doit nous donner une probabilité plus grande pour le faciès "GRES".

```

FACIES NAME : GRES
FACIES *   PROBABILITE
GRES   *   0.85
ARGILE *   0.10
SILT   *   0.05

```

La cross-validation peut être utile comme aide à la détermination d'un noyau lors de la construction d'un modèle. En supposant qu'il y ait déjà beaucoup de noyaux déterminés et que l'utilisateur hésite sur le nom de faciès à donner à un noyau, il peut s'aider de la cross-validation pour trancher.

N.B. L'instruction COEFF sert à changer un paramètre utile au calcul des probabilités. Elle correspond à la sous-commande COEFF (cf. X).

6) Modifications d'un modèle : Les instructions CHANGE, DEL, LITO

L'éditeur de modèles ne permet pas de modifier les noyaux mêmes, c'est à dire les valeurs des composantes. Mais par contre, il permet effacer des noyaux du modèle (instruction DEL) et de changer leur nom de faciès (instruction CHANGE ou C).

L'instruction LITO permet de prendre connaissance de la liste de tous les noms de faciès permis.

L'utilisation de ces trois instructions a déjà été abordée au sujet de la création d'un modèle (cf. VII 10)

7) L'instruction SAVE ou S

Cette instruction permet d'imprimer dans un fichier les résultats que certaines instructions impriment à l'écran. Ces instructions sont : LIST, CV, LCV, CVR, LCVR, LITO.

L'instruction SAVE se combine avec ces instructions de la façon suivante SAVE (ou S) <instruction>

```
Ex. :  SAVE  LIST  1   10
       ou   S    CV   10  11
       ou   S    L    1
       etc...
```

Un fichier de sortie doit être défini au préalable à l'aide de l'instruction : SAVE DEF.

Le système répond alors SAVING FILE NAME = ?

L'utilisateur doit taper un nom de fichier nouveau. Ce fichier sera un fichier image carte de 80 caractères au plus. (Attention aux lignes trop longues ; elles seront tronquées).

Si, lors d'une instruction SAVE autre que "SAVE DEF", aucun fichier de sauvegarde n'a été défini, alors le système pose automatiquement la question
SAVING FILE NAME = ?
avant d'exécuter l'instruction.

A l'exécution, rien n'apparaît à l'écran, sauf les messages d'erreurs éventuelles, puis le témoin d'attente "M#".

IX) LES ESSAIS ET LES APPLICATIONS

1) Test de la méthode des centres dynamiques

Pour tester la partie du programme correspondant aux centres dynamiques, nous avons simulé des données de sondage diagraphiques.

a) Les données synthétiques

Nous nous sommes inspirés d'un cross-plot GR-CNL : (Gamma-Ray, Porosité Neutron) fait par la S.N.E.A.(P) (Société Nationale Elf-Aquitaine) sur les données d'un de ses puits de forage. Ce cross-plot comportait cinq faciès représentés en courbes d'isodensité de points et avait grossièrement la forme représentée sur la figure 44.

A partir de ces formes, nous avons reproduit un ensemble de données à deux variables en prenant des échelles arbitraires. Nous avons affecté à chacune de ces données un numéro qui simule la profondeur du point de données dans le puits. Nous avons donc ce que nous voulions : des données regroupées par paquets.

b) Les centres dynamiques

Nous avons fait des essais avec peu de centres pour vérifier le bon fonctionnement du programme. Les points sont ici regroupés en 5 groupes. Si on prend 5 centres initiaux, on obtient presque à tous les coups, en peu d'itérations (1 ou 2), les centres des 5 groupes présents. Comme le montre la figure 45, avec la métrique de réduction, on obtient pour centres, les points numérotés 6, 23, 63, 99 et 135.

Avec la métrique de Mahalanobis, on obtient pour centres les points numérotés 6, 30, 57, 92 et 135, ce qui est peu différent.

Nous avons ensuite fait des essais avec beaucoup de centres : 30 centres sur 162 données. Dans ce cas, la position des centres est stable au bout de 3 itérations. Comme le montre la figure 46, chaque groupe de points contient plusieurs centres.

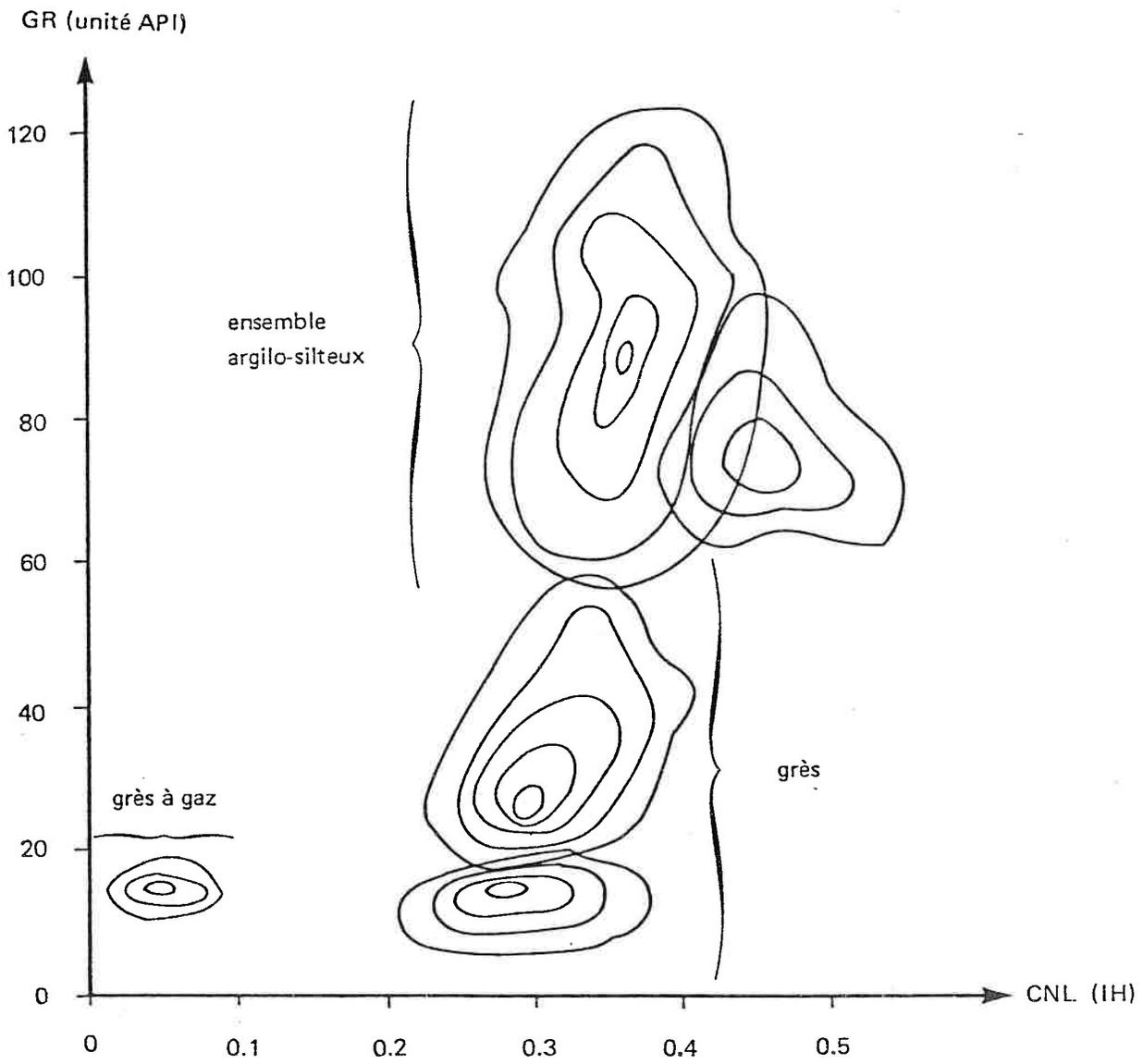


Figure 44. Cross-plot GR-CNL en courbes isodensité

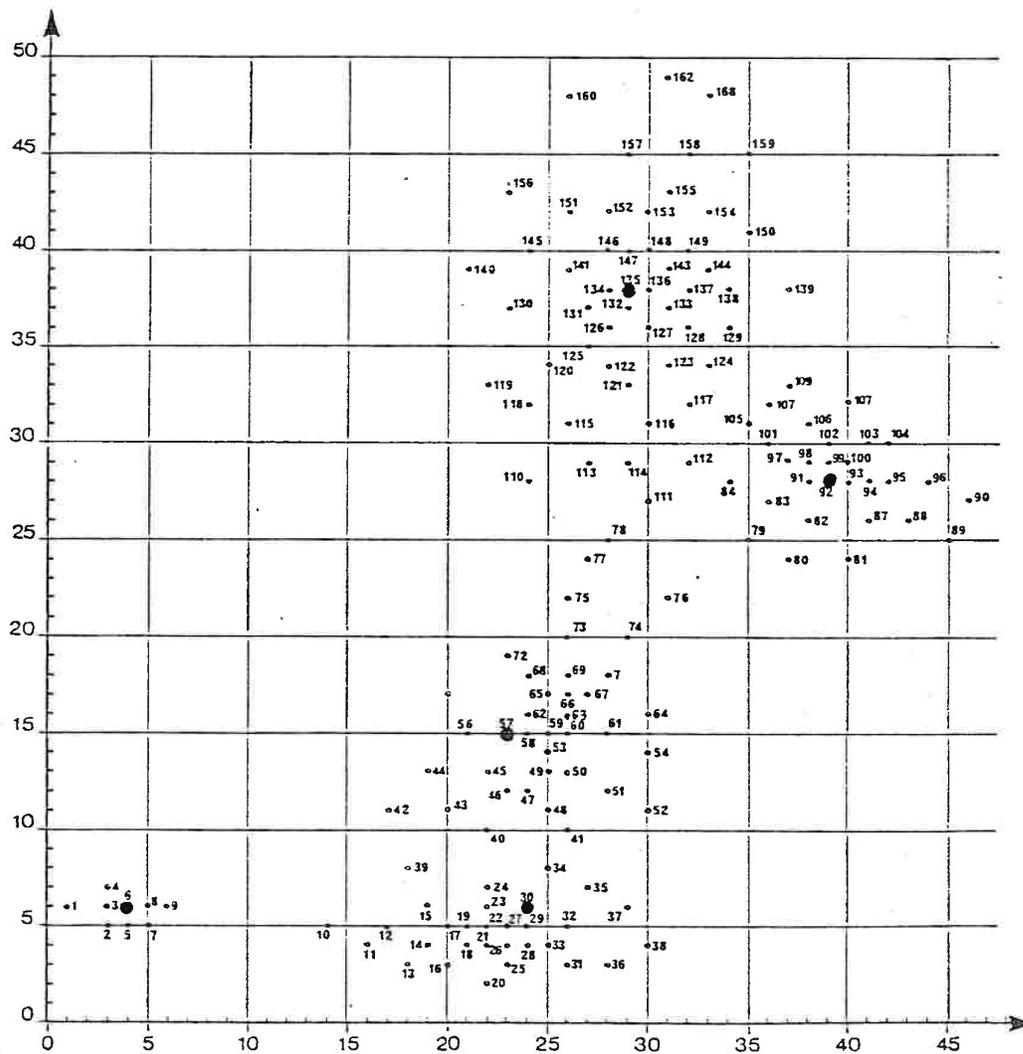


Figure 45. Cinq centres finaux parmi les 162 données synthétiques

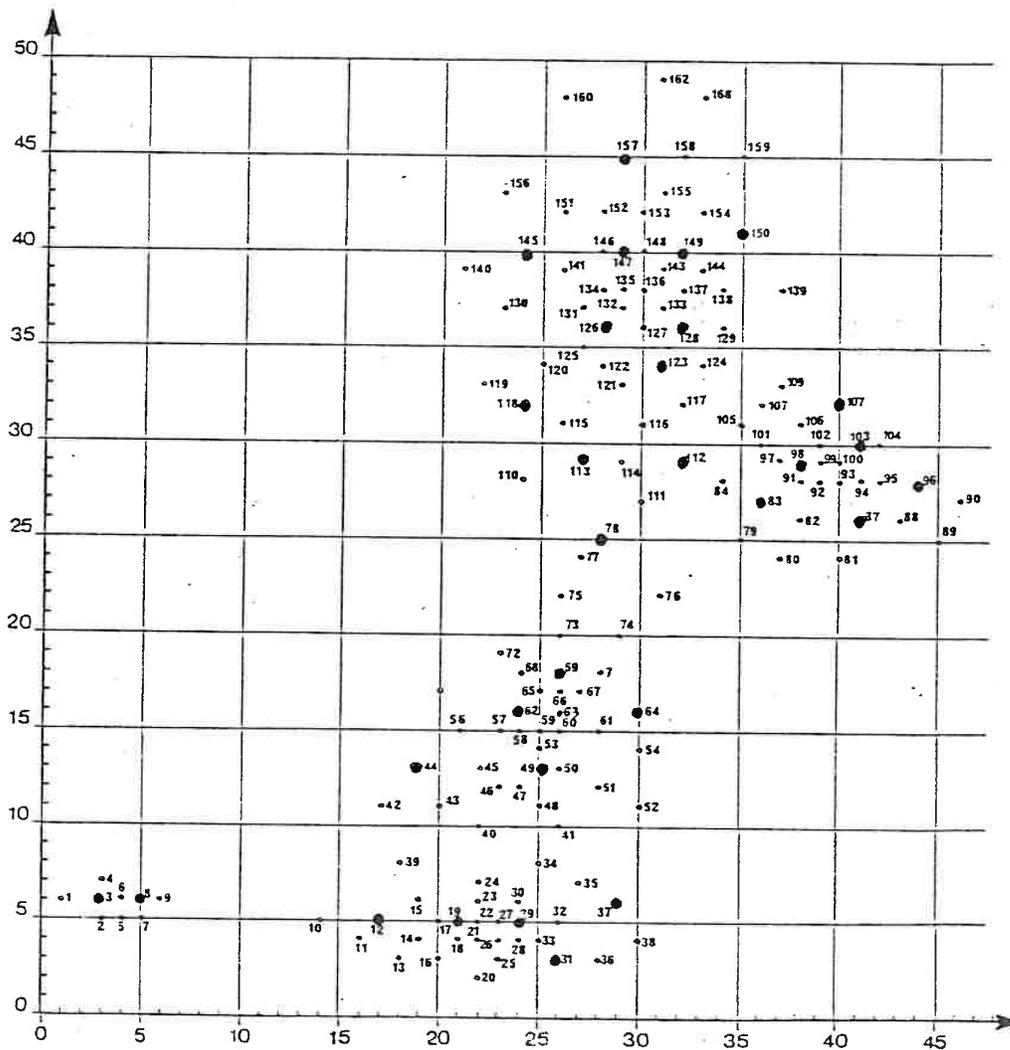


Figure 46. Trente centres finaux

2) Les premières données réelles : le puits S1

Elles sont fournies par la S.N.E.A.(P). Elles sont prises dans une série sédimentaire détritique de delta contenant par endroits des hydrocarbures.

Pour des raisons de confidentialité, les noms des sondages utilisés sont factices.

Les outils utilisés sont :

SP	potentiel spontané
GR	gamma-ray
NPHI	neutron
LLS	résistivités
LLD	
MINV	
MSFL	
MNOR	
CALI	diamètreur
DELTAT	vitesse du son dans la roche (SONIC)

La S.N.E.A.(P) nous avait communiqué une première détermination semi-automatique de la lithologie qu'elle avait faite. Tous les niveaux du puits portaient donc déjà un nom de faciès pris parmi les onze suivants :

GRG	grès grossiers
GRGH	grès grossiers à hydrocarbures
GRM	grès moyens
GRMH	grès moyens à hydrocarbures
TI	silts
TIH	silts à hydrocarbures
LL	niveaux carbonatés
AA	argile franche
AT	argile silteuse
TA	silt argileux
NL	niveaux cavés

Les faciès les plus fréquents sont les faciès détritiques : GRG, GRM, TI, TA, AT, et TA.

Il y a quelques passages de niveaux carbonatés : LL.

Il y a d'épais bancs de faciès à hydrocarbures : GRGH, GRMH, TIH

Il y a très peu de niveaux cavés : NL.

a) La construction d'un modèle

Nous avons tâtonné pour trouver le bon nombre de noyaux. Nous avons d'abord fabriqué un modèle à 50 noyaux. Il s'est avéré insuffisant lors de l'analyse diagraphique. Les onze faciès choisis étaient trop peu discernés par le programme comparativement à la première détermination. Nous avons ensuite fabriqué un modèle à 200 noyaux. Celui-la par contre s'est avéré bon tout de suite.

Lors de la construction d'un modèle, ne sachant pas comment déterminer les noyaux que nous présentait l'ordinateur, nous nous sommes fiés à cette première détermination. C'est-à-dire que nous regardions quel était le faciès choisi lors de la première détermination à la profondeur indiquée à l'écran.

Ex : KERNEL NO : 3

DEPTH	LLS	LLD	GR	NPHI
2300.0	-	-	-	-

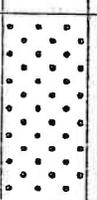
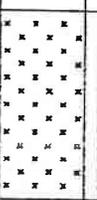
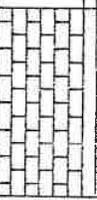
Si à la profondeur 2300 m du puits, la première détermination avait indiqué le faciès TI, nous attribuions le faciès TI au noyau n° 3.

Cette détermination était basée sur des coupures faites sur les différentes composantes du sondage.

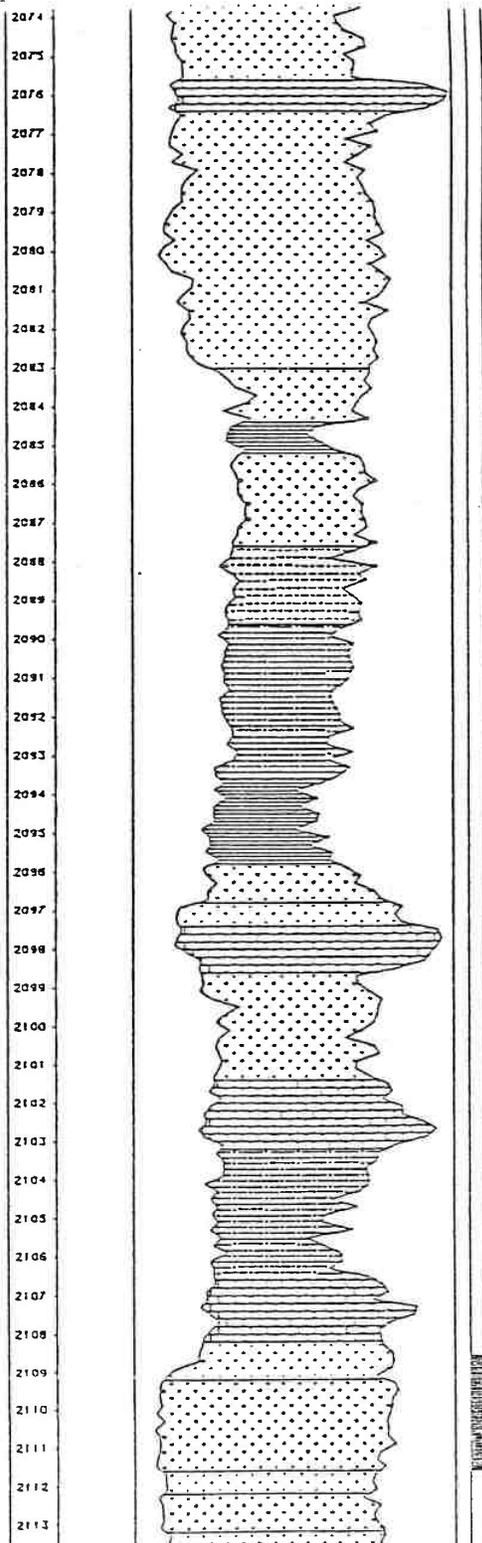
Ex. : Si le Gamma-Ray mesuré était inférieur à 24, alors l'échantillon était classé parmi les grès grossier (GRG).

b) L'analyse diagraphique

Les résultats obtenus sont présentés sur la figure 38. Sur cette figure, on voit deux colonnes lithologiques d'une même partie du puits S1. La première est celle qui a été déterminée semi-automatiquement, puis arrangée par la S.N.E.A.(P). La deuxième a été obtenue grâce à notre programme.

FACIES	CODE	FIGURE
GRES GROSSIER	GG	
GRES MOYEN	GM	
SILT	T	
SILT ARGILEUX	TA	
CALCAIRE	C	
ARGILE	A	
ARGILE SILTEUSE	AT	
CAVE	L	

Prof. (m)



Prof. (m)

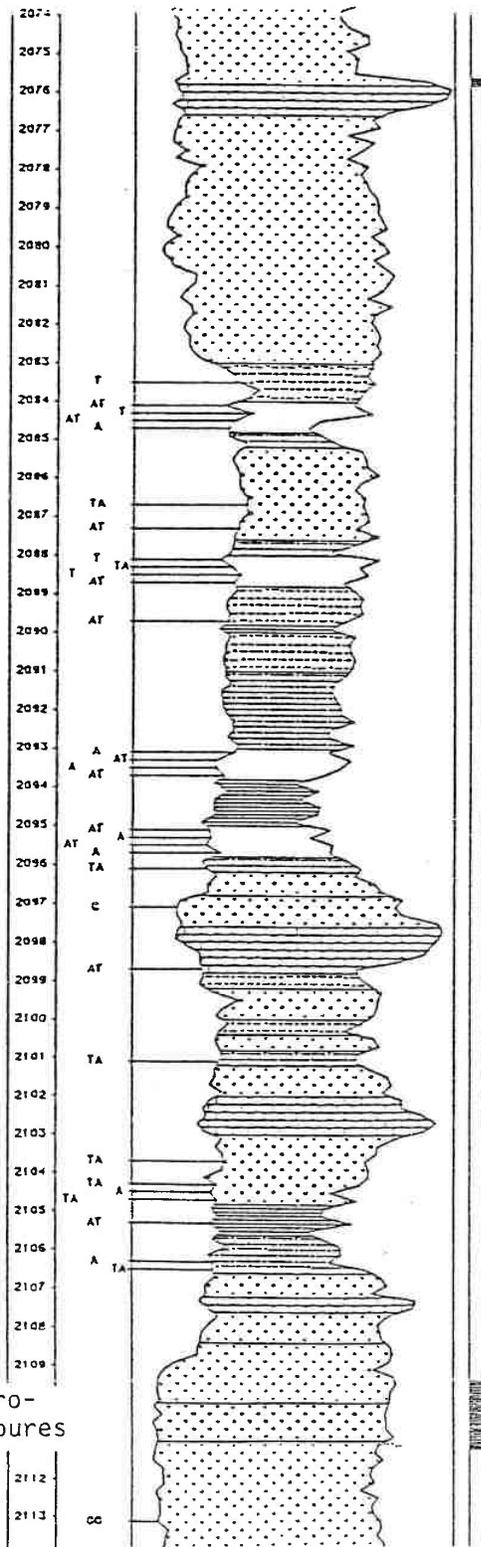


Figure 48 - Détection d'hydrocarbures par le système.

Dans la colonne de gauche, les couches sont épaisses et bien marquées. Dans la colonne de droite, la détermination est beaucoup plus "fouillée". Il existe beaucoup de fines couches. Cependant, les deux déterminations sont très ressemblantes. Les faciès grès (GRG et GRM, représentés en pointillés) sont en particulier bien reconnus. Les ensembles silto-argileux sont confus sur la deuxième colonne. Ceci vient du fait que les faciès argile (AA), argile silteux (AT), silts argileux (TA) et silts (TI) sont très voisins dans l'espace paramétrique.

Sur la figure 48, une autre partie du puits S1 est représentée avec les deux colonnes lithologiques.

La présence d'hydrocarbures est signalée par grisé dans la petite colonne à droite de la colonne lithologique.

On peut voir que la zone à hydrocarbures, présente sur cette partie du puits, est bien reconnue par notre programme, avec cependant une légère différence d'épaisseur. Notre programme a décelé des hydrocarbures au sommet de cette partie du puits. C'est apparemment une erreur : les hydrocarbures se signalent généralement par une forte résistivité et une très faible mesure de l'outil neutron. Ici, c'est la proximité du calcaire qui, en augmentant la mesure de résistivité des grès, a induit le système en erreur. On remarque, d'autre part, sur la figure 39 que les niveaux calcaires ont été bien reconnus.

La valeur des probabilités dépend du coefficient de lissage que l'on applique. Le coefficient de lissage est un coefficient qui élargit ou diminue la "fenêtre" des gaussiennes servant au calcul des probabilités.

Nous avons fait des calculs avec la première méthode de calcul de la fenêtre (méthode du MISE). La fenêtre obtenue était trop petite et c'est ce qui nous a amenés à introduire ce "coefficient de lissage". A ce moment, il fallait prendre un coefficient de lissage de 20. Maintenant, avec la nouvelle méthode de calcul de la fenêtre, le coefficient de lissage à choisir varie entre 0.5 et 2.

Dans le cas de ce puits, avec ce modèle, un coefficient de lissage de 0.5 donnait des résultats trop brutaux : (probabilité 1 pour un faciès et 0 pour tous les autres).

Un coefficient de 0.7 donnait de bons résultats et un coefficient de 1 donnait des résultats légèrement trop flous.

Il faut arriver à trouver le coefficient de lissage tel que les probabilités ne soient pas trop abruptes (pour avoir les informations supplémentaires que procurent les probabilités) et tel qu'en prenant le faciès le plus probable on obtienne des bancs épais de même faciès.

c) Création d'un modèle plus grand

Après avoir fait ce modèle, nous avons fait aussi un modèle à 400 noyaux : ce modèle a été construit comme le précédent à partir des résultats de la première détermination, mais il n'a pas donné de meilleurs résultats que le modèle à 200 noyaux. Un avantage cependant : le faciès NL était présent dans ce modèle et donc présent dans la détermination des faciès.

3) Le deuxième ensemble de données réelles : le puits S2

Nous avons ensuite appliqué le modèle à 200 noyaux à un deuxième puits de la même région. Ce deuxième puits était formé approximativement des mêmes faciès que le précédent, avec cependant un peu moins d'hydrocarbures.

Ce puits contenait deux mesures de résistivité de moins que le premier : MINV (résistivité micro-inverse) et MNOR (résistivité micro-normale). Tous les outils du modèle n'étaient donc pas utilisés. Les noyaux utilisés étaient, si l'on peut dire, la projection des noyaux du modèle sur un espace de dimension inférieure à celle de l'espace initial.

D'après la théorie, si on avait utilisé un modèle fabriqué uniquement avec ces outils, on n'aurait pas obtenu des résultats très différents. Le nombre de noyaux risque ici d'être un peu plus grand ce qui, à priori, est un avantage car alors la définition des classes est plus fine, mais parfois est un inconvénient car ces noyaux supplémentaires peuvent ajouter de la confusion.

En utilisant les données brutes, c'est-à-dire telles qu'elles avaient été mesurées, (SP GR NPHI LLS LLD MSFL CALI DELTAT), nous

avons obtenu des résultats médiocres. Nous obtenions plus de niveaux à hydrocarbures qu'il n'y en avait réellement. Nous nous sommes aperçus que ceci venait de la mesure de la PS (polarisation spontanée).

En effet, les valeurs de la PS dans les grès du premier puits variaient entre 10 et 50 et les valeurs les plus fortes parmi celles-ci correspondaient aux grès à hydrocarbures (GRGH et GRMH). Dans le modèle, les noyaux de grès à hydrocarbures avaient aussi en moyenne des valeurs de PS plus fortes (15-30). Or, dans le deuxième puits, les valeurs de la PS étaient toutes beaucoup plus grandes (aux alentours de 60 pour les grès). Ce qui fait que chaque niveau de grès était reconnu comme un niveau de grès à hydrocarbures.

Il a fallu transformer la mesure de polarisation spontanée (SP spontaneous potential, en anglais). En effet, d'un puits à l'autre, la mesure de la polarisation spontanée peut varier. Par contre, la différence entre la polarisation spontanée mesurée et la polarisation spontanée des argiles pures à cette même profondeur a une signification physique valable pour tous les sondages.

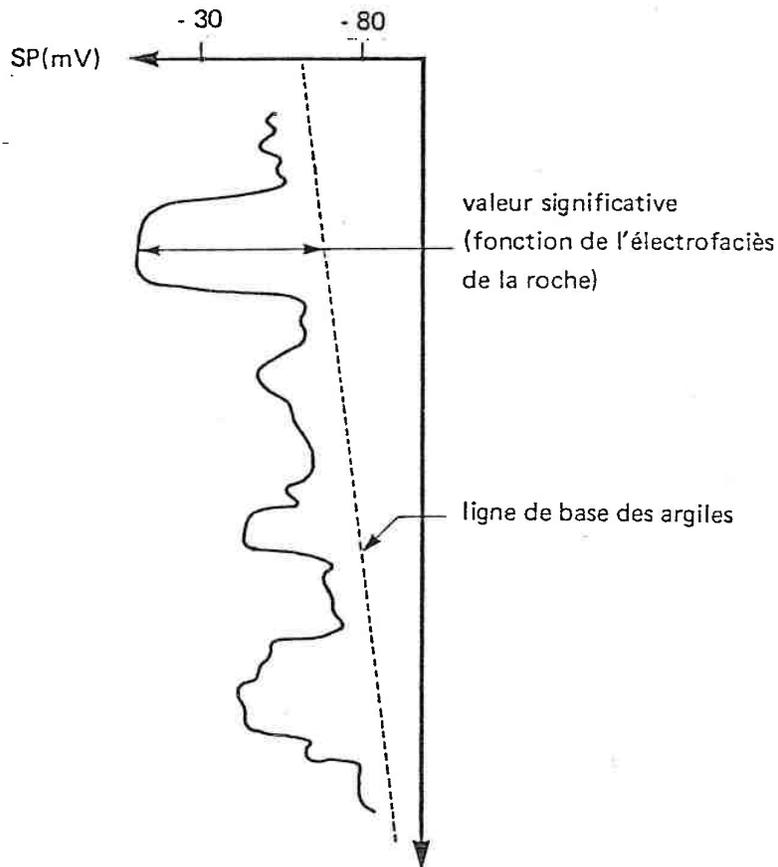


Figure 49. Image classique d'une mesure de PS sur un sondage

Nous avons donc créé une nouvelle composante dans les deux puits appelée SPNEW égale à cette différence de potentiels (en mV). Nous avons créé un nouveau modèle contenant le même nombre de noyaux (200 ==> 193) à partir du premier puits et nous l'avons appliqué au deuxième puits en remplaçant donc l'outil SP par sa nouvelle valeur SPNEW.

Nous avons constaté que le nombre de niveaux à hydrocarbures avait nettement diminué. Il y avait encore quelques confusions entre les grès grossiers GRG et les grès moyens à hydrocarbures GRMH.

Dans l'ensemble, les résultats ne sont pas aussi bons que ceux obtenus sur le premier puits (S1). C'est une conséquence de ce que le modèle utilisé pour l'analyse n'a pas été tiré du sondage analysé.

Les faciès ont été définis à partir du premier puits et les caractéristiques diagraphiques d'un faciès sur le deuxième puits (S2), ne sont parfois pas les mêmes que celles du même faciès sur le premier puits (S1).

Pour remédier à ceci, il faut affiner le modèle :

- soit, prendre plus de noyaux dès le début, mais ce n'est pas vraiment bon,
- soit, remanier le modèle au fur et à mesure des analyses sur des puits différents.

Actuellement, ceci ne peut se faire qu'en changeant le nom des noyaux ou en supprimant des noyaux du modèle courant.

Avec la sous-commande MODMNG, qui devrait permettre de fusionner deux modèles, on pourra éventuellement fabriquer un modèle propre à traiter les puits d'une région géologique entière en fusionnant plusieurs modèles faits à partir de puits de cette région, puis en enlevant les noyaux en trop.

4) Application aux diagraphies dans des séries salifères

Le sondage à analyser était formé de quatre types principaux de faciès :

- argile		(ARG)
- halite (sel gemme) NaCl		(SG)
- sylvinite	KCl	(SYL)
- karnalite		(KARN)

Le faciès sylvinite apparaît d'une manière très franche, alors que la karnalite apparaît souvent mélangée avec le sel gemme. Il a fallu donc désigner une autre classe de données : le mélange sel gemme-karnalite (SK). Nous avons donc à déterminer cinq faciès : ARG SG SYL KARN et SK

a) Discrimination des faciès

Elle se faisait facilement à la main à partir de 3 mesures :

- la résistivité au Laterolog (LL)
- la radioactivité naturelle (GR) (Gamma-ray)
- la réponse à l'outil neutron (NEUT).

La résistivité servait surtout à repérer les niveaux argileux car ceux-ci ont une résistivité bien inférieure aux niveaux salifères. L'outil neutron est presque aussi efficace pour repérer les niveaux argileux car l'argile est le plus perméable de tous les faciès. Enfin la radioactivité est très sélective pour les faciès sylvinite (SYL) et karnalite (KARN).

Remarque : Il existe un "pôle sel gemme", un "pôle Sylvinite" et un "pôle Karnalite". C'est à dire que les points de chacune de ces trois classes sont regroupés autour d'un point précis appelé pôle pour cette classe. Ceci est valable dans l'espace des variables comme dans tout sous-espace de l'espace des variables. La figure suivante montre l'existence de ces pôles dans le sous-espace engendré par les variables GR et NEUT.

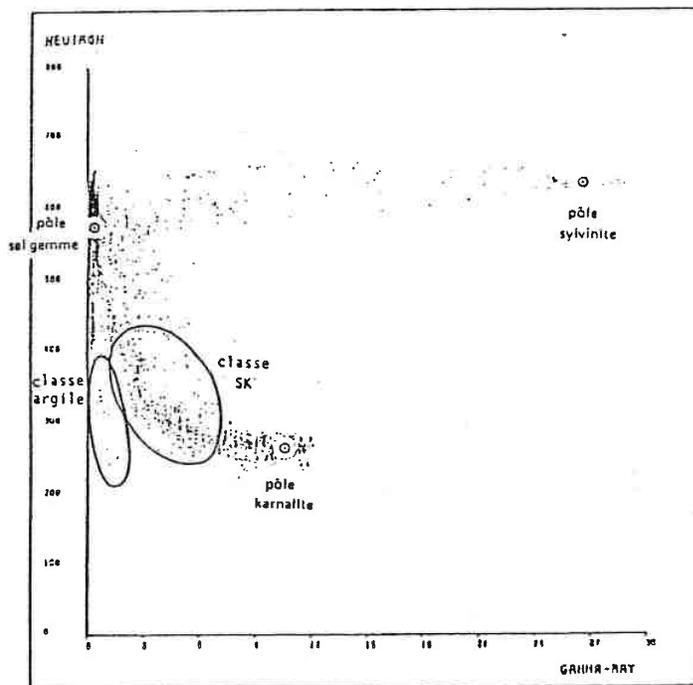


Figure 50. Cross-plot GR-NEUT

b) Détermination automatique

Le modèle à 75 noyaux

Nous avons créé en premier un modèle à 75 noyaux à partir du sondage.

Dans l'ensemble, les résultats obtenus étaient satisfaisants. Les trois outils diagaphiques suffisaient à différencier les cinq faciès. Les limites des couches trouvées ne correspondaient qu'approximativement avec les limites observées sur les carottes extraites du sondage mais ce phénomène est commun à toute détermination automatique de la lithologie à partir de diagaphies.

En effet, on peut voir que sur le cross-plot GR-NEUT, la classe SK (mélange de sel gemme et de karnalite) est floue : les points du sondage appartenant à cette classe sont dispersés. Par conséquent les points du modèle appartenant à cette classe le sont aussi. De même pour la classe argile car les niveaux argileux sont souvent un mélange de sel gemme et d'argile pure.

Cependant, dans le cas de la sylvinite, ce phénomène est accentué au point que les fines couches de sylvinite n'ont pas toutes été reconnues.

Le modèle à 10 noyaux

Le modèle à 75 noyaux contenait beaucoup de noyaux de la classe SG (la moitié environ). La classe SG était donc "trop bien" définie. De même pour les autres classes dans des proportions moindres.

Nous avons donc créé un autre modèle avec 10 noyaux seulement. Les résultats obtenus ont été sensiblement les mêmes avec cependant une meilleure détermination des couches de sylvinite.

Le modèle potassique

Nous avons créé un modèle à partir de l'ensemble des niveaux du sondage que la première analyse avait déterminés comme appartenant à des faciès non halitiques dans l'espoir de pouvoir séparer ces niveaux en différentes classes relatives à la teneur en potassium du niveau.

Exemple :

KARN1 : niveau karnalitique à teneur en potassium faible.
KARN2 : niveau karnalitique à teneur en potassium forte.

Cette tentative a été infructueuse mais elle a montré un détail intéressant :

Appliquée à tout le sondage, et donc en particulier aux niveaux halitiques, l'analyse a attribué à ceux-ci un faciès inconnu. Le problème des points à l'infini évoqué au chapitre IV 3) a donc bien été résolu.

c) Le problème des fines couches de sylvinite

Sur le cross-plot GR-NEUT, il existe une trainée de points entre le pôle "sel gemme" et le pôle "sylvinite". Ces points correspondent aux niveaux de sel gemme proches d'une couche de sylvinite. En effet, comme le montre la figure suivante, la mesure de gamma-ray en un point du sondage est influencée par les niveaux adjacents à ce point. Ainsi, aux alentours d'une couche de sylvinite, la valeur de Gamma-Ray devient supérieure à la valeur normale.

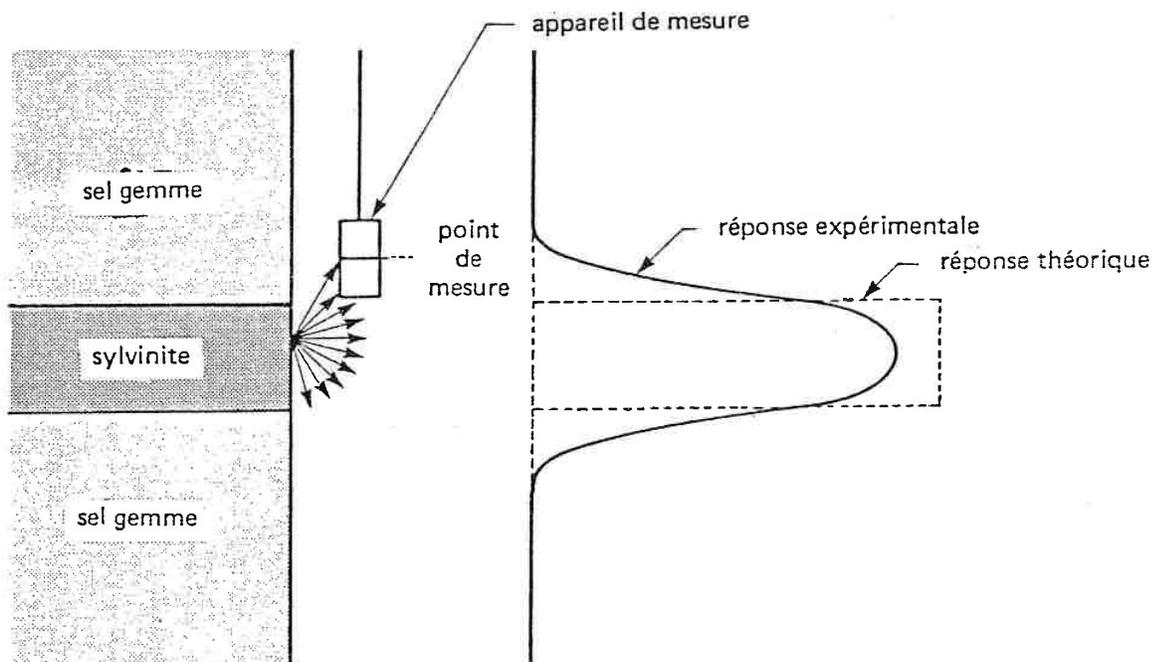


Figure 51. Influence d'une couche de sylvinite sur le Gamma-Ray

Lors de la construction du premier modèle, ces points situés entre les deux pôles ont provoqué l'apparition de noyaux au même endroit.

Dans un premier essai, nous avons attribué à ces noyaux le nom de faciès "sel gemme" (SG). Lors de l'analyse diagraphique, ceci eut pour conséquence la disparition des petites couches de sylvinite. En effet, celles-ci avaient des pics de gamma-ray beaucoup moins hauts que ceux des grosses couches car elles sont influencées par les niveaux supérieurs et inférieurs qui sont en général beaucoup moins radioactifs.

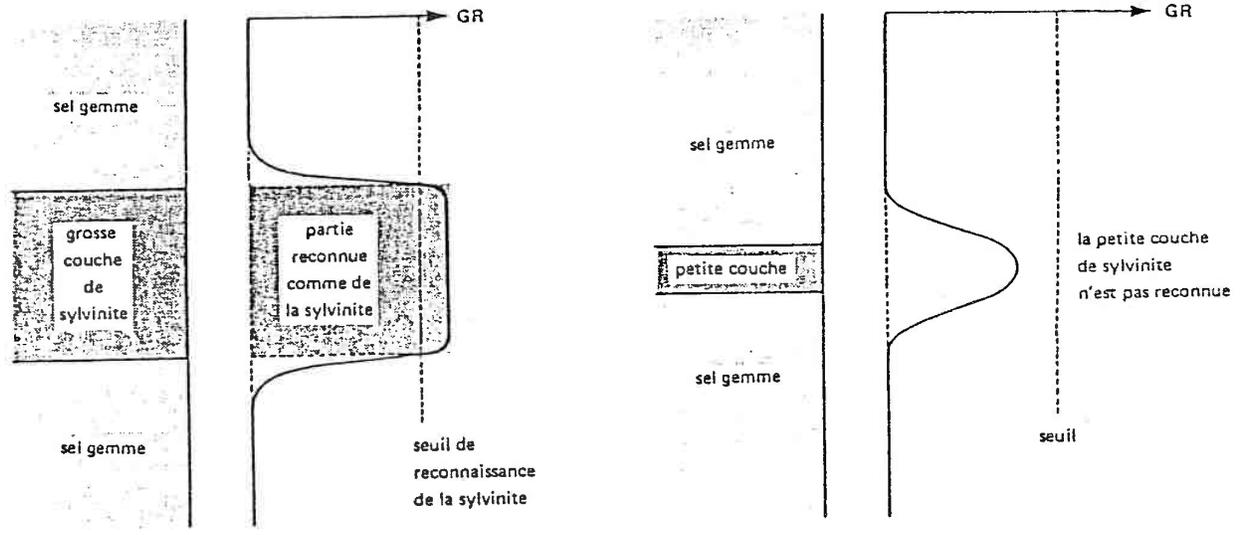


Figure 52

Les petites couches n'ont donc pas été reconnues lors de l'analyse diagraphique. Par contre, les grosses couches ont été bien reconnues, peut être rendues un peu plus petites que dans la réalité.

Dans un deuxième essai, nous avons affecté à ces noyaux le nom de faciès "karnalite" (KARN). Nous avons obtenu, lors d'une nouvelle détermination, un effet inverse. Les petites couches de sylvinite étaient repérées, mais la largeur trouvée était bien supérieure à leur largeur réelle. Les grosses couches, quant à elles, étaient toujours bien repérées quoique trouvées légèrement plus larges, cette fois-ci. On retrouvait donc bien l'ensemble des niveaux "karnalite", mais le problème des fines couches n'était pas réglé.

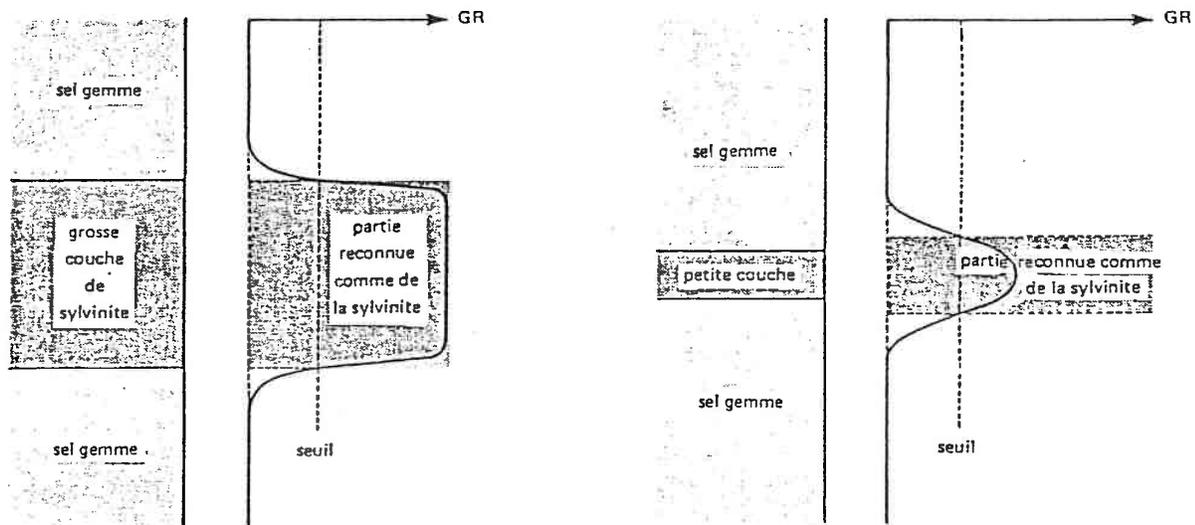


Figure 53

Une autre solution aurait été de supprimer virtuellement ces noyaux en ne leur attribuant pas de noms de faciès puisqu'en effet ces noyaux ne sont représentatifs de rien.

La coupure entre le faciès "sel gemme" et le faciès "sylvinite" aurait été faite par le programme. Cette coupure serait plus naturelle : les niveaux se trouvant en plein milieu d'une couche de karnalite seraient bien déterminés, les niveaux se trouvant en plein milieu d'une couche de sel gemme aussi, les niveaux intermédiaires seraient affectés d'une probabilité de karnalite grossièrement proportionnelle à leur valeur de gamma-ray.

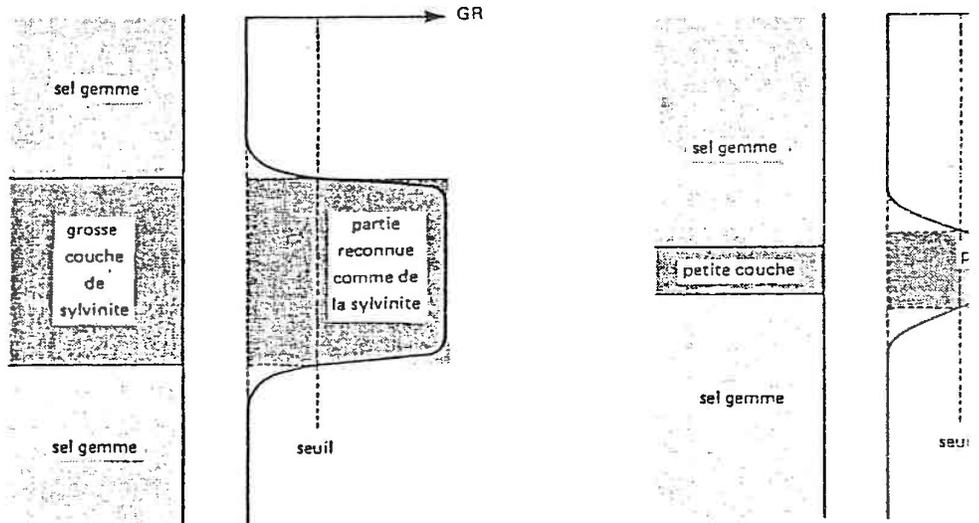


Figure 53

Une autre solution aurait été de supprimer virtuellement les noyaux en ne leur attribuant pas de noms de faciès puisqu'eux-mêmes ne sont représentatifs de rien.

La coupure entre le faciès "sel gemme" et le faciès "karnalite" aurait été faite par le programme. Cette coupure se fait naturellement : les niveaux se trouvant en plein milieu d'une couche de karnalite seraient bien déterminés, les niveaux se trouvant en plein milieu d'une couche de sel gemme aussi, les niveaux intermédiaires seraient affectés d'une probabilité de karnalite grossièrement proportionnelle à leur valeur de gamma-ray.

es qui
nerait
Des

à
il n'est
e" et

re au
urs du
qu'on
point

re la
iès du

on du
nnue à
cette
peut
on du

ns une
nouveau
t une
lleurs

- Les épaisseurs des couches de sylvinite trouvées étaient plus proche de la réalité.
- Les fines couches ont été reconnues.

45) Les temps de calcul

a) Pour le calcul des centres lors de la construction d'un modèle

$$t_1 = k_1 \times \text{NIT} \times \text{NCENT} \times \text{NDATA} \times \text{NVAR}$$

Le temps de calcul croît proportionnellement ou presque avec le nombre d'itérations (NIT), le nombre de centres (NCENT), le nombre de points de données (NDATA) et le nombre de variables (NVAR).

Pour donner un exemple : Avec 2 itérations, la recherche de 200 centres parmi 3 000 points de données comprenant 10 variables utilise 20 mn de temps CPU sur un ordinateur HP 9000.

On a vérifié sur un autre exemple la relation de proportionnalité : avec un nombre d'itérations toujours égal à 2, la recherche de 150 centres parmi 6 000 points de données comprenant 3 variables a pris 11 minutes de temps CPU.

dans le premier cas :

$$\frac{\text{NIT} \times \text{NCENT} \times \text{NDATA} \times \text{NVAR}}{t} = \frac{2 \times 200 \times 3\,000 \times 10}{20} = 6 \times 10^5$$

dans le deuxième cas :

$$\frac{\text{NIT} \times \text{NCENT} \times \text{NDATA} \times \text{NVAR}}{t} = \frac{2 \times 150 \times 6\,000 \times 3}{11} = 5 \times 10^5$$

b) Pour le calcul des probabilités

$$t_2 = k_2 \times \text{NDATA} \times \text{NVAR} \times \text{NCENT}$$

Il est strictement proportionnel au nombre de points de données (NDATA) et globalement proportionnel au nombre de variables (NVAR) et au nombre de noyaux (NCENT) du modèle utilisé.

Un exemple : Le temps de calcul pour l'analyse de 3 000 points de données, avec 10 variables et 200 noyaux prend 20 mn de temps CPU.

217A

X) ANNEXE : MODE D'EMPLOI DES SOUS-COMMANDES

1) La sous-commande WELLDEF

Fonction :

L'objectif de la sous-commande WELLDEF est fondamental puisqu'il s'agit de définir le puits de sondage sur lequel on désire employer la commande DIA, à savoir le "Current GDT-well".

Appel de la sous-commande :

Elle peut être appelée directement par l'utilisateur soit à partir du niveau "SUBCOMMAND = ?", soit à partir du niveau "A#" (voir sous-commande ANADIA).

Elle peut également être appelée de manière automatique par le programme des lors que l'existence du "Current GDT-well" est nécessaire et que ce dernier n'a pas encore été défini. Cet appel automatique peut provenir des sous-commandes TOOLS, EDTO, ANADIA, WELLDIR, MODBLD et EDWELL.

Organisation :

Lorsque l'on entre dans la sous-commande, deux situations sont possibles :

a) aucun fichier n'a encore été choisi :

En début de session, il convient de définir un premier fichier de sondage en répondant à la question :

CURRENT GDT-WELL = ?

L'utilisateur peut alors :

- répondre par STOP, "?" ou <CR>;
- fournir le nom du sondage désire : son existence est alors vérifiée puis, s'il s'agit bien d'un GDT-well, le système le charge en mémoire.

b) il existe déjà en mémoire un "Current GDT-well" :

Au cours d'une session, l'opérateur peut rappeler autant de fois qu'il le veut la sous-commande WELLDEF afin de charger d'autres sondages à étudier. Il reçoit alors un message lui rappelant quel est le "GDT-well" courant :

CURRENT GDT-WELL :

puis il est invité à répondre à la question

NEW CURRENT GDT-WELL = ?

il peut alors fournir les mêmes réponses que dans le cas précédent : STOP, "?", nouveau nom de fichier, ou bien <CR> s'il désire en fin de compte conserver le même sondage (aucune modification n'est alors faite).

Exemple :

```
-> SUBCOMMAND = ?
- WELLDEF
-> CURRENT GDT-WELL = ?
- NANCY-28

-> SUBCOMMAND = ?
- WELLDEF
-> CURRENT GDT-WELL : NANCY-28
NEW CURRENT GDT-WELL = ?
EPINAL-A2

-> SUBCOMMAND = ?
- WELLDEF
-> CURRENT GDT-WELL : EPINAL-A2
NEW CURRENT GDT-WELL = ?
- <CR>
```

2) La sous-commande SETDEF

Fonction :

Cette sous-commande permet de désigner un sous-ensemble de niveaux de mesures du sondage sur lequel seront effectués tous les futurs travaux.

Appel de la sous-commande :

Cette sous-commande ne peut être appelée que directement par l'utilisateur depuis les niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

Si aucun GDT-well n'est chargé en mémoire centrale, alors le système appelle la sous-commande WELLDEF. Ensuite, le programme pose la question :

CHOSEN SET = ?

à laquelle l'utilisateur doit répondre par le nom d'une composante de type SET existant dans le GDT courant ou par les identificateurs "ALL"

ou "OMEGA" désignant l'ensemble du GDT-well.

Ce sous-ensemble de travail n'est modifié que par l'appel de cette sous-commande ou par l'appel de WELLDEF.

Exemple :

```
-> SUBCOMMAND = ?
-   SETDEF
-> CURRENT GDT-WELL = ?
-   OBAGI.DT1
-> CHOSEN SET = ?
-   SSET
-> SUBCOMMAND = ?
```

3) La sous-commande MODDEF

Fonction :

Ayant pour but de définir le "Current GDT-model" que l'utilisateur emploiera pour effectuer l'analyse mathématique, cette sous-commande est en fait très semblable à WELLDEF.

Appel de la sous-commande :

Tout comme la précédente, la sous-commande MODDEF peut être appelée :

- soit "manuellement" depuis les niveaux "SUBCOMMAND = ?" ou "A#";
- soit automatiquement depuis les sous-commandes RUN (sous-commande ANADIA), MODDIR, EDMOD, CROSSV ou CROSSVR, quand l'existence d'un modèle est à priori nécessaire.

Après avoir rempli sa fonction, le programme retourne bien entendu au niveau d'appel.

Organisation :

A l'entrée dans cette sous-commande, il existe également deux cas de figure possibles :

- a) aucun modèle n'a encore été choisi :
- Le système pose alors à l'utilisateur la question suivante :

```
CURRENT GDT-MODEL = ?
```

il est alors possible de répondre par :

- STOP, "?" ou bien <CR>;
- un nom de "GDT-model" dont le système vérifiera l'existence avant de le charger en mémoire.

b) il existe déjà un modèle :
Après avoir rappelé à l'utilisateur le nom de ce modèle,

CURRENT GDT-MODEL :

le système pose la question :

NEW CURRENT GDT-MODEL = ?

Les réponses attendues sont les mêmes que précédemment. Si l'utilisateur désire garder l'ancien GDT-model, il doit taper <CR>. Le système revient alors au niveau d'appel sans avoir fait aucun changements.

Exemple :

```
-> SUBCOMMAND = ?
- MODDEF
-> CURRENT GDT-MODEL = ?
- STOP

-> SUBCOMMAND = ?
- MODDEF
-> CURRENT GDT-MODEL = ?
- LORRAINE

-> SUBCOMMAND = ?
- MODDEF
-> CURRENT GDT-MODEL : LORRAINE
NEW CURRENT GDT-MODEL = ?
- MEUSE
-> SUBCOMMAND = ?
```

4) Les sous- commandes WELLDIR et MODDIR

Fonction :

Ayant des structures de GDT, les "fichiers-sondages" et les "fichiers- modèles" contiennent un certain nombre d'information dans leur directory.

Le rôle des sous-commandes WELLDIR et MODDIR est d'éditer ces informations, respectivement pour le "Current GDT-well" et pour le "Current GDT-model".

Appel de ces sous-commandes :

WELLDIR et MODDIR sont accessibles uniquement par appel manuel depuis les niveaux "SUBCOMMAND = ?" et "A#". Après édition de la directory souhaitée, le système revient au niveau d'appel.

Organisation :

Si le fichier dont on veut éditer la directory n'a pas été auparavant chargé en mémoire centrale, la sous-commande d'édition WELLDIR ou MODDIR appelle automatiquement la sous-commande de définition adéquate WELLDEF ou MODDEF.

Lorsque le fichier est présent en mémoire centrale, l'édition des informations est immédiate.

Le système édite en premier le titre du fichier puis, dans un tableau, pour chaque composante du fichier, il édite le nom, le type (réel, entier, alpha-numérique, booléen ou set), le nombre de "sous-composantes" si c'est une composante vectorielle et son cardinal si c'est une composante set (c'est à dire le nombre d'unités statistiques contenues dans ce set).

Ensuite le système édite :

- le type du fichier (DT, DT1, DT2 ou DT3),
- le nombre de composantes existant dans ce fichier,
- le nombre maximal de composantes que peut contenir ce fichier,
- le nombre d'unités statistiques total du fichier,
- et enfin les valeurs définissant la position des données dans l'espace si ce fichier est de type DT1, DT2 ou DT3.

exemple :

- > SUBCOMMAND = ?
- WELLDIR
- > CURRENT GDT-WELL = ?
- WELL.DT1

-> SIMULATION DE LOG COMPOSITE

```

* COMPONENT'S * TYPE * VECTOR * CARDINAL *
* NAME * * LENGTH * ( SET ) *
*****
* * * * *
* DEPTH * REAL * * *
* NPHI * REAL * * *
* GR * REAL * * *
* FACIES * ALPHA * 2 * *
* SET1 * SET * * 76 *
*****

```

```

TYPE OF THE DATA-TABLE : DT1
NUMBER OF USED COMPONENTS : 6
MAXIMUM NUMBER OF COMPONENTS : 6
MAXIMUM NUMBER OF STATISTICAL UNITS : 162

```

```

XMIN = .100E+01 XMAX = .162E+03 NX = 162 STEPX = .100E+01

```

Le système termine en demandant à l'utilisateur s'il veut connaître les titres attribués à chaque composante du fichier GDT :

```

DO YOU WANT TO EDIT THE TITLES RELATIVE TO
THE ABOVE COMPONENTS ? (ANSWER Y/N)

```

Si l'utilisateur répond "N", le système revient au niveau d'appel; sinon, il édite les titres de chaque composante du fichier.

ne 11 Exemple :

- DEPTH : DEPTH
- NPHI : POROSITE NEUTRON
- GR : GAMMA-RAY
- FACIES : FACIES LE PLUS PROBABLE
- SET1 : SOUS-ENSEMBLE DES DONNEES VERIFIANT "NPHI > 24."

5) La sous-commande TOOLS

Fonction :

Lors de la définition du "Current GDT-well" au moyen de la sous-commande WELLDEF, le programme construit entièrement le "Current-tools" à partir du "Data-tools" (cf. II 2 c) correspondant au sondage choisi (sauf si auparavant l'utilisateur avait sauvegardé un

201

"Current-tools" au moyen de l'instruction OKSAVE; nous le verrons plus loin). Ce n'est qu'au moyen de cette sous-commande que l'utilisateur est en mesure d'apporter des modifications sur l'ensemble des outils courants.

Appel de la sous-commande :

TOOLS est seulement accessible par appel direct depuis les niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

a) Si aucun sondage n'a encore été défini :

Un fichier de données étant indispensable pour travailler, TOOLS appelle automatiquement la sous-commande WELLDEF. Après chargement du fichier en mémoire, le système édite le "Current-tools" et le "Data-tools" qui sont ici égaux :

Ex. :

```
"CURRENT TOOLS" = "DATA-TOOLS" :  
GR      NPHI
```

b) Si le "Current GDT-well" est déjà défini :

Que la commande ait déjà été appelée au cours de la session ou non, le système rappelle le nom du fichier courant :

```
CURRENT GDT-WELL : ....
```

puis édite les deux ensembles d'outils :

Exemples :

- s'ils sont identiques :

```
"CURRENT-TOOLS" = "DATA-TOOLS"  
GR      NPHI
```

- s'ils sont différents :

```
"DATA-TOOLS" :  
GR      NPHI  
"CURRENT-TOOLS" :  
GR
```

Après être ainsi entré dans la sous-commande, le système édite un témoin "I#", indiquant qu'il attend une instruction de la sous-commande TOOLS.

Les instructions :

Il existe, en dehors des "réponses classiques" : STOP, "?" et <CR>, huit instructions ayant chacune une fonction bien déterminée. Ces instructions sont les suivantes :

- REM : (REMove) Soustraction d'outils;
- ADD : (ADDition) Addition d'outils;
- CHG : (CHanGe) Remplacement d'outils;
- DEF : (DEFinition) Définition d'outils;
- DT : (Data-Tools) Egalisation du "Current-tools" au "Data-tools";
- EDTO : (EDition TOols) Edition des deux ensembles;
- OK : Acceptation des outils courants;
- OKSAVE : Acceptation avec sauvegarde.

Toutes ces instructions ont des syntaxes spécifiques mais il existe également une syntaxe globale qui permet l'utilisation simultanée de certaines d'entre elles. Des règles syntaxiques ayant du être choisies, les réponses de l'utilisateur sont automatiquement contrôlées et éventuellement sanctionnées par des messages personnalisés (WARNING ou SYNTAX ERROR).

Toutefois, malgré la diversité des cas envisagés, il est certain que l'utilisateur averti ne sera pas pénalisé et qu'il disposera ainsi d'un maximum de sécurité. Quant à l'opérateur débutant, il appréciera certainement les messages courts et précis qui permettent de maîtriser rapidement toutes les possibilités de la sous-commande.

a) instruction REM

Elle a pour fonction de soustraire un ou plusieurs outils du "Current-tools".

syntaxe :

REM suivi des noms d'outils à soustraire séparés par des espaces blancs. Les noms des outils peuvent être donnés dans n'importe quel ordre et leur nombre est indéfini.

Exemple :

```
-> SUBCOMMAND = ?
- TOOLS
-> "CURRENT-TOOLS" = "DATA-TOOLS" :
    TOOL2 TOOL4 TOOL7 TOOL9
    T#

- REM TOOL9 TOOL2
-> NEW "CURRENT-TOOLS" :
    TOOL4 TOOL7
```

T#

REM peut être employé plusieurs fois sur une même ligne ou être utilisée en même temps que les instructions ADD et CHG.

b) instruction ADD

Elle permet au contraire d'ajouter au "Current-tools" un ou plusieurs outils, dans la mesure où ceux-ci sont disponibles sur le sondage étudié bien entendu.

syntaxe :

ADD suivi des noms d'outils à ajouter séparés par des espaces blancs. Les règles de syntaxe sont strictement analogues à celles de l'instruction REM. ADD peut être utilisée simultanément avec les instructions REM et CHG.

Exemple :

```

-> "CURRENT-TOOLS" = "DATA-TOOLS" :
    TOOL2 TOOL5 TOOL6 TOOL8 TOOL9
    T#

- REM TOOL8 TOOL9 TOOL5
-> NEW "CURRENT-TOOLS" :
    TOOL2 TOOL6
    T#

- ADD TOOL5 TOOL9
-> NEW "CURRENT-TOOLS" :
    TOOL2 TOOL5 TOOL6 TOOL9
    T#

```

c) instruction CHG

Il s'agit d'une combinaison des deux précédentes qui a pour effet de remplacer certains outils du "Current-tools" par d'autres outils disponibles dans le "Data-tools". Cette instruction doit permettre à l'utilisateur averti de définir très rapidement les nouveaux outils courants.

syntaxe :

CHG <noms d'outils à enlever> / <noms d'outils à ajouter>

Dans chaque liste de noms d'outils, les noms d'outils doivent être séparés par un ou plusieurs espaces blancs et leur nombre est indéfini. Il n'est pas nécessaire de séparer une liste d'outils de l'opérateur "/" par un espace blanc.

Une liste peut être vide et les séquences du type suivant sont

autorisées :

```

CHG / TOOL2 TOOL6
CHG TOOL4 /
CHG TOOL7      (equivalent à "REM TOOL7")

```

CHG peut être employé plusieurs fois sur la même ligne et peut être utilisée simultanément avec les instructions REM et ADD.

Exemple :

```

-> "CURRENT-TOOLS" = "DATA-TOOLS" :
    TOOL3 TOOL4 TOOL7 TOOL8 TOOL9
    T#

- CHG TOOL4 TOOL8 /
-> NEW "CURRENT-TOOLS" :
    TOOL3 TOOL7 TOOL9
    T#

- CHG TOOL7 / TOOL4 CHG /TOOL8
-> NEW "CURRENT-TOOLS" :
    TOOL3 TOOL4 TOOL8 TOOL9
    T#

```

d) instruction DEF

Elle permet de définir directement sur une même ligne l'ensemble des outils courants. Elle autorise un gain de temps appréciable dans certaines situations (voir exemple).

syntaxe :

```
DEF <noms d'outils>
```

Exemple :

```

-> "CURRENT-TOOLS" = "DATA-TOOLS" :
    TOOL2 TOOL3 TOOL5 TOOL7 TOOL8 TOOL10 TOOL14 TOOL19
    T#

- DEF TOOL3 TOOL8 TOOL5
-> NEW "CURRENT-TOOLS" :
    TOOL3 TOOL5 TOOL8
    T#

```

e) instruction DT

Il s'agit ici d'un ordre ne nécessitant pas l'entrée de noms d'outils. Il doit être employé seul et a pour conséquence la construction du "Current-tools" à partir du "Data-tools".

Exemple :

```
-> "DATA-TOOLS" :
    TOOL3 TOOL5 TOOL6 TOOL9 TOOL10
    "CURRENT-TOOLS" :
    TOOL5 TOOL9
    T#

- DT
-> NEW "CURRENT-TOOLS" :
    TOOL3 TOOL5 TOOL6 TOOL9 TOOL10
    T#
```

f) instruction EDTO

Il s'agit d'un ordre d'édition qui s'emploie seul et fait apparaître à l'écran les listes du "Data-tools" et du "Current-tools". Cette instruction fait en réalité appel à la sous-commande EDTO à laquelle il convient de se reporter pour plus de précisions.

g) instruction OK

Cet ordre s'emploie seul. Il entraîne la validation des changements faits par l'utilisateur dans le "Current-tools" et fait sortir le système de la sous-commande TOOLS. Le système revient alors au niveau d'appel.

N.B. : l'entrée d'un <Carriage Return> au niveau "T#" produit le même effet que l'ordre OK.

h) instruction OKSAVE

Son utilisation permet de conserver en mémoire la liste des outils courants. Ainsi, lorsque l'utilisateur appellera ultérieurement la sous-commande WELDEF, le programme ne construira pas le "Current-tools" à partir des outils du nouveau sondage, mais en prenant l'intersection entre ces outils ("Data-tools") et ceux du "Current-tools" sauvegardé. Cet ordre entraîne aussi la fin de la sous-commande TOOLS.

Syntaxe globale : utilisation simultanée de plusieurs instructions

Il est possible d'employer autant de fois qu'on le désire, et dans n'importe quel ordre les instructions REM, ADD et CHG, en veillant toutefois au respect des règles de syntaxe qui leur sont propres.

Ainsi les séquences suivantes sont autorisées :

```
- REM TOOL4 TOOL3 CHG TOOL7/TOOL2
- ADD TOOL9 TOOL6 TOOL3 REM TOOL2 ADD TOOL5
```

6) La sous-commande EDTO

Fonction :

Il s'agit là d'une sous-commande d'édition permettant de faire apparaître sur l'unité de sortie les outils contenus respectivement dans le "Data-tools" et le "Current-tools".

Appel de la sous-commande :

EDTO est accessible :

- soit directement par appel de l'utilisateur depuis les niveaux "SUBCOMMAND = ?", "A#" ou "T#";
- soit automatiquement par le programme, lorsque l'utilisateur peut à priori avoir besoin de se remémorer la situation : c'est le cas par exemple à l'entrée de la sous-commande ANADIA.

Comme toutes les autres, la sous-commande EDTO renvoie le système au lieu d'appel lorsque sa fonction est achevée.

Organisation :

a) si les "Current-tools" et "Data-tools" sont identiques : l'utilisateur voit apparaître à l'écran le message :

"CURRENT-TOOLS" = "DATA-TOOLS" :
suivi de la liste des outils contenus dans ces ensembles.

Exemple :

```
-> SUBCOMMAND = ?
-   EDTO
-> "CURRENT-TOOLS" = "DATA-TOOLS" :
    TOOL1 TOOL3 TOOL7

    SUBCOMMAND = ?
```

b) si les deux ensembles sont différents :

ils sont alors édités séparément, avec successivement les outils disponibles sur le sondage ("Data-tools") puis les outils courants ("Current-tools").

Les outils courants présents sur le sondage sont soulignés par un marqueur comme sur l'exemple suivant :

"DATA-TOOLS" :
TOOL2 TOOL5 TOOL6 TOOL9

"CURRENT-TOOLS" :
TOOL5 TOOL9

Remarque : Si l'on a sauvegardé des outils courants (en utilisant l'instruction OKSAVE) puis redéfini un nouveau fichier de sondage, il peut se faire que certains outils courants ne soient plus disponibles; avant de reconstruire un nouveau "Current-tools", le programme appelle la sous-commande EDTO et édite les outils en soulignant ceux qui sont communs aux deux ensembles :

"DATA-TOOLS" :
TOOL2 TOOL3 TOOL6 TOOL8 TOOL9

"CURRENT-TOOLS" :
TOOL2 TOOL4 TOOL6 TOOL7 TOOL8

puis après avoir été reconstruit, le "Current-tools" est affiché :

NEW "CURRENT-TOOLS" :
TOOL2 TOOL6 TOOL8

7) La sous-commande EXTWELL

Fonction :

Cette sous-commande permet d'augmenter ou de réduire dans la mesure du possible les dimensions du "Current GDT-well". Cette modification de taille se fait dans un autre tableau de données qui est recopié dans un fichier disque différent.

L'extension ou la réduction n'affecte en aucune manière le "Current GDT-well" ni son image stockée dans le fichier disque.

Toutes les données compatibles avec les nouvelles dimensions du tableau sont sauvegardées à la place correspondante dans le nouveau GDT. Les composantes de type set sont également sauvegardées et sont réorganisées de telle sorte qu'elles décrivent l'intersection du sous-ensemble de départ avec le nouvel ensemble d'unités statistiques.

Appel de la sous-commande :

Cette sous-commande est accessible par une commande manuelle de l'utilisateur aux niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

A l'entrée de la sous-commande, le système teste s'il existe un GDT-well en mémoire. Si non, il appelle la sous-commande WELLDEF.

Ensuite, le système demande les dimensions du nouveau GDT-well. A chaque question posée concernant une dimension, l'utilisateur peut répondre par le caractère "*", ce qui entraîne la conservation de l'ancienne dimension correspondante. Une réponse "?" à ces questions entraîne, en plus d'une explication, l'édition de la valeur de cette ancienne dimension.

Le système demande d'abord le nombre de composantes du nouveau GDT-well avec la question :

COMPONENTS NUMBER = ?

L'utilisateur doit y répondre par un nombre au moins égal au nombre de composantes effectivement créées (différent du nombre maximum de composantes).

Le système demande ensuite la coordonnée du premier niveau (= de la première unité statistique) du nouveau GDT-well :

XMIN = ?

puis le nombre de niveaux (= d'unités statistiques) du nouveau GDT-well :

NX = ?

Ainsi, la coordonnée de départ peut être décalée et le nombre de niveaux peut être modifié mais le pas initial reste inchangé. Le nouvel ensemble ainsi créé doit avoir une intersection non vide avec l'ensemble d'origine.

Le programme demande enfin le nom du fichier disque dans lequel le nouveau GDT doit être sauvegardé.

8) La sous-commande CBLD

Fonction :

Cette sous-commande permet de créer une nouvelle composante dans le "Current GDT-well".

Appel de la sous-commande :

Il se fait par appel direct de l'utilisateur depuis les niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

En premier lieu, le système teste s'il existe un "Current GDT-well". Si non, il appelle la sous-commande WELLDIR.

Afin de rendre son utilisation plus agréable, la sous-commande peut prendre deux formes.

a) Entrée du nom de la sous-commande seul.

Il s'établit alors un dialogue entre la machine et l'utilisateur qui peut éventuellement obtenir des explications en tapant le caractère "?".

Le système demande d'abord le nom de cette nouvelle composante (6 caractères maximum).

COMPONENT NAME = ?

Le système vérifie que ce nom n'existe pas et qu'il reste de la place dans le GDT pour introduire cette composante.

Le système demande ensuite le type de cette nouvelle composante :

COMPONENT TYPE = ?

La réponse doit être un des cinq types de variables prévus :

I (Entier) R (Réel) A (ASCII) B (Booléen) S (Set)
pour des variables simples
ou

IV RV AV BV SV
pour des composantes vectorielles.

Dans le cas de composantes vectorielles (réponse IV, RV, AV, DV ou SV), le système demande la dimension du vecteur :

VECTOR LENGTH = ?

Le système créera alors N composantes (N étant la réponse à la dernière question) liées entre elles d'une manière privilégiée.

Exemple :

-> SUBCOMMAND = ? - CBLD -> CURRENT GDT-WELL = ? - WELL.DT1 ->
COMPONENT NAME = ? - CNAME -> COMPONENT TYPE = ? - R

b) Entrée du nom de la sous-commande suivi des réponses aux questions précédentes

Exemples : CBLD TOTO IV 4
 CBLD COMP R

En cas d'erreur dans l'instruction, le système reprend le dialogue à l'endroit de l'erreur détectée, la suite de l'instruction étant perdue.

Enfin, quelle que soit la procédure choisie, le programme demande le titre de la composante. (Le titre est un commentaire de 72 caractères maximum qui permet ainsi de désigner clairement le contenu de la composante.)

TITLE = ?

Dans le cas d'une composante vectorielle, on peut affecter un titre global ou un titre spécifique pour chaque sous-composante du vecteur

GLOBAL TITLE ? (Y/N)

Le système pose alors une fois (réponse "Y") ou N fois (réponse "N") la question :

TITLE = ?

9) La sous-commande CDEL

Fonction :

Cette sous-commande permet de détruire une composante quelconque du "Current GDT-well" et de récupérer ainsi sa place.

Appel de la sous-commande :

Il se fait par appel direct de l'utilisateur à partir des niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

S'il n'existe pas de "Current GDT-well", le système appelle la sous-commande WELLDEF. Puis le système demande le nom de la composante à supprimer :

COMPONENT NAME = ?

10) La sous-commande SETBLD

Fonction :

Cette sous-commande permet de créer un nouveau set dans le "Current GDT-well". Le système peut créer une nouvelle composante set à cette occasion ou réutiliser une composante set déjà existante. Le set est créé en testant les valeurs d'une composante réelle ou entière du "Current-GDT-well" : Un niveau du sondage appartient au nouveau set si la valeur à ce niveau de la composante testée est comprise entre deux bornes numériques données par l'utilisateur.

Appel de la sous-commande :

-137

Il peut se faire par appel direct de l'utilisateur à partir de niveaux "SUBCOMMAND = ?" ou "A#".

Organisation :

En premier lieu, le système charge un GDT-well à l'aide de la sous-commande WELLDEF s'il n'en existait pas auparavant.

Puis le système demande le nom de la composante à tester :
COMPONENT TO BE TESTED ?

Il vérifie que cette composante est bien de type réel ou entier puis demande le nom de la composante set à définir :
CREATED SUBSET NAME ?

Si cette composante existe déjà, il s'assure que l'utilisateur est bien décidé à la réutiliser :

WARNING : COMPONENT ALREADY DEFINED
DO YOU WANT TO REUSE THIS COMPONENT (Y/N) ?

Si l'utilisateur répond "N", le système revient au niveau d'appel. Sinon le système continue en demandant les valeurs maximales et minimales définissant le set :

VALUE FOR "FSUP" ?

et

VALUE FOR "FINF" ?

Dans le cas où l'utilisateur n'a pas défini auparavant un set de travail au moyen de la sous-commande SETDEF, le système calcule le set demandé puis en édite le cardinal (c'est à dire le nombre de niveaux du sondage appartenant à ce set) :

CARDINAL OF THE CREATED SET : ...

Dans le cas contraire, le système laisse le choix à l'utilisateur entre plusieurs options :

- soit ignorer le set courant. Le calcul du set demandé se fait donc sur tous les niveaux du sondage comme dans le cas précédent.
- soit créer une intersection entre le set courant et l'ensemble des niveaux du sondage vérifiant la condition. Ceci est en fait la prise en compte selon les règles du set courant : le calcul du set demandé est fait uniquement sur les niveaux appartenant au set courant. Les niveaux n'appartenant pas au set courant n'appartiendront pas non plus au set demandé.
- soit créer une réunion entre le set courant et l'ensemble des niveaux du sondage vérifiant la condition : le calcul du set demandé est donc fait sur tous les niveaux du sondage et on lui rajoute tous les niveaux du set courant qui n'ont pas été sélectionnés par le calcul précédent.

Le système demande à l'utilisateur de choisir une de ces trois options avec la question :
CHOSEN OPTION ?

Les réponses possibles à cette question sont :

- IGNO : le set courant est ignoré.
- UNIO : le set demandé est égal à l'union des deux sous-ensembles.
- INTE : le set demandé est égal à l'intersection des deux sous-ensembles.

Enfin le système édite à l'écran la valeur du cardinal du set créé :
CARDINAL OF THE CREATED SET : ...

11) La sous-commande MODBLD

Fonction :

La sous-commande MODBLD (de l'anglais MODEl BuILD) sert à créer un modèle à partir du GDT-well chargé en mémoire centrale. Si la construction est menée à bien, ce modèle devient le modèle courant (ce qui veut dire : chargé en mémoire centrale et prêt à être utilisé pour des calculs de probabilités).

Appel de la sous-commande :

Cette sous-commande a un rôle important. Elle ne peut être appelée que directement par l'utilisateur depuis les niveaux "SUBCOMMAND = ?" ou "A#".

Organisation :

- La sous-commande est faite de deux grandes parties :
- la construction des noyaux
 - l'affectation d'un nom de faciès à ces noyaux.

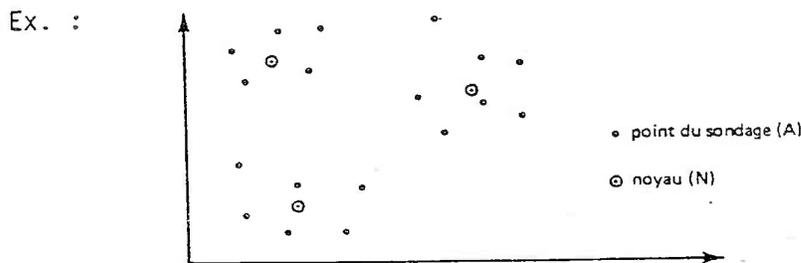
1re partie : Construction des noyaux

Le système s'assure à l'entrée de la sous-commande de deux choses :
- s'il existe déjà en mémoire centrale un modèle récemment créé ou modifié, auquel cas il demande à l'utilisateur s'il veut le sauvegarder sur disque. En effet, il n'est prévu de place en mémoire que pour un seul modèle.

- s'il existe un GDT-well en mémoire centrale. Sinon le logiciel appelle automatiquement la sous-commande WELLDEF et l'utilisateur est obligé de définir un puits de sondage.

Exemple :
- MODBLD
- SAVE CURRENT GDT-MODEL ? (Y/N)
- Y
- CURRENT GDT-WELL = ?
- ...

Les noyaux sont construits à partir des données du sondage à l'aide d'un algorithme de centres dynamiques. Considérons un ensemble de points dans l'espace des variables associé à un puits de sondage : cet algorithme calcule un nombre n de points de cet espace de telle façon que chacun de ces n points soient le barycentre d'un groupe bien distinct de l'ensemble initial des points.



Pour ce faire, l'utilisateur doit préciser le nombre de centres, le nombre d'itérations que doit effectuer la machine dans cet algorithme, et la métrique utilisée dans les calculs de distance.

Le système pose la question
OPTION LIST = ?

L'utilisateur doit répondre une ou plusieurs des options suivantes
NKER, METR, NIT

- NKER sert à définir le nombre de noyaux
- METR sert à définir la métrique
- NIT sert à définir le nombre d'itérations

Si l'utilisateur ne définit pas un de ces trois paramètres, alors ce paramètre garde la valeur qu'il avait au précédent passage dans MODBLD. Si c'est la première fois, ce paramètre prend une valeur fixée par le programme.

L'utilisateur peut utiliser trois autres options possibles
NORM STD STDE

Ces options s'utilisent seules.

NORM sélectionne pour ces trois paramètres trois valeurs fixées

(normalisées) par le programme.
 STD sélectionne les trois valeurs utilisées lors du dernier passage dans MODBLD.
 STDE édite à l'écran les trois valeurs des paramètres utilisés lors du dernier passage dans MODBLD. Puis le système pose de nouveau la question :
 OPTION LIST = ?

Exemple :
 -> OPTION LIST = ?
 - NKEK, METR; NIT
 -> NUMBER OF KERNELS = ?
 - 50
 -> METRIC = ?
 - COV
 -> NUMBER OF ITERATIONS = ?
 - 2

2e partie : Affectation d'un nom de faciès aux centres trouvés.

Pour chaque noyau, le système cherche le point de données du sondage qui lui est le plus ressemblant. Le système alors édite les coordonnées du noyau et la profondeur de ce point de données. Il édite également le pourcentage de points de données qui lui est rattaché.

```
-> KERNEL NO ...
    PERCENTAGE OF POINTS AROUND IT : ...
    DEPTH      VARI      VAR2      VAR3
    2000.5     ....     ....     ....   (coordonnées)
    FACIES NAME = ?
```

Cette présentation peut être accompagnée d'une présentation graphique sur une autre console : (voir plus loin l'option DISP)

Trois types de dessins sont possibles :

- a) Le dessin des courbes de composantes choisies dans le "current-tools" aux alentours du point de données qui ressemble au noyau.
- b) La projection des centres trouvés sur un cross-plot.
- c) La projection des centres trouvés sur un plan discriminant.

L'écran peut contenir au maximum quatre dessins, tous types mélangés et parmi ceux-là plusieurs cross-plots différents.

A la question "FACIES NAME = ?",

l'utilisateur peut répondre :

- un nom de faciès (mot de six caractères au plus faisant partie d'une liste définie à l'extérieur du logiciel)

ou alors répondre :

- NONE : s'il ne veut pas attribuer de nom de faciès au noyau courant,
- SAME : s'il ne veut pas changer le nom de faciès déjà attribué. (Un nom de faciès peut avoir déjà été attribué si l'utilisateur est "revenu en arrière" quand il attribuait un nom de faciès à chacun des noyaux.)
- QUIT : s'il veut que le système cesse de lui présenter les noyaux. Le système quitte alors la sous-commande MODBLD et entre dans la sous-commande EDMOD en conservant le nouveau modèle tel quel.
- Une des instructions de l'éditeur suivantes :
LIST, CV, LCV, CVR, LCVR, DISP

Ex. : LIST 1 *

Le système se repositionne alors au dernier noyau édité par CV, LCV ou LIST.

- DISP : Cette instruction permet de sélectionner ou de changer les dessins édités par MODBLD.

Dans le cas où l'utilisateur a répondu l'option DISP, le système pose la question :
DISPLAY OPTIONS = ?

L'utilisateur doit alors taper une ou plusieurs des options suivantes :
CURV, PLOT, PLO2, PLO3, PLO4, DISC, STD, STDE, NIL

Signification de ces options :

* CURV :
Cette option sert à faire éditer par le système à l'écran une portion des courbes du sondage choisis à l'avance par l'utilisateur en même temps qu'il présente chaque noyau. La portion elle-même des courbes est définie en fonction du noyau. Elle entoure le point du sondage le plus ressemblant au noyau en question. (Ainsi, on espère que ces portions de courbe donneront une bonne idée de la nature du noyau).

* PLOT :
Cette option sert à faire apparaître à l'écran un cross-plot de deux composantes. L'utilisateur doit alors, en réponse à la question :

COMPONENTS OF PLOT = ?

donner le nom des composantes sous forme littérale (noms qui sont contenus dans le current-tools).

* PLO2, PLO3 et PLO4 :

Ces options servent à faire apparaître plusieurs cross-plots à l'écran (respectivement 2, 3 et 4).

Le système pose alors plusieurs fois la question
COMPONENTS OF PLOT n0 i = ?

* DISC : Cette option sert à l'affichage à l'écran des noyaux du modèle en construction projetés sur un plan le plus discriminant par rapport aux classes construites.

* STD : Cette option sert à sélectionner les options de dessin employées lors d'un précédent passage dans MODBLD ou EDMOD. Cette option s'emploie donc seule.

* STDE : Cette option sert à éditer littéralement les options sélectionnées par STD (options "standard"). Cette option s'emploie seule aussi.

Le système redemande ensuite
DISPLAY OPTIONS = ?

Exemple :

```
-> DISPLAY OPTIONS = ?
- STDE
-> CURRENT DISPLAY OPTIONS :
  5 CURVES
  NAMES :  CURV1      CURV2 ...
  SCALES :   20.      .....
  PLO1 :   COMP1     COMP2
  PLO2 :     .....
  DISC
  DISPLAY OPTIONS = ?
```

* NIL : Cette option sert à supprimer toute édition graphique précédente.

Editeur :

Quand tous les noyaux ont été présentés, le système appelle directement l'éditeur de modèles. Le système affiche donc le témoin d'attente "M#" et l'utilisateur peut accéder à toutes les fonctions de l'éditeur de modèles. Il doit taper <CR> pour sortir de l'éditeur et il revient alors au niveau "SUBCOMMAND = ?" ou "A#".

12) La sous-commande ANADIA

Fonction :

La sous-commande ANADIA permet d'accéder à plusieurs sous-commandes dont les noms sont :

- RUN
 - VISU
 - SMTH
- et INCLUD

Appel de la sous-commande :

Cette sous-commande regroupe tous les travaux d'analyse diagraphique, d'ou son nom (Analyse diagraphique - ANA DIA)

- RUN : calcul des probabilités
- VISU : visualisation des résultats
- SMTH : lissage des résultats
- INCLUD : inclusion de ces résultats dans le GDT-well

ANADIA ne peut être appelée que directement par l'opérateur depuis le niveau "SUBCOMMAND = ?".

Organisation :

Le système teste d'abord s'il existe un GDT-well en mémoire. Sinon, il appelle la sous-commande WELLDEF. Ensuite, s'il n'existe pas non plus de GDT-model en mémoire, il le signale et se met en attente d'une sous-commande de ANADIA en éditant le témoin "A#".

S'il en existe un, il cherche parmi tous les outils du modèle ceux qui sont aussi présents dans le current-tools. Ces outils formeront l'ensemble du "working-tools", outils de travail en vue d'une analyse.

Le système édite le "current-tools" et le "working-tools", puis le témoin d'attente "A#".

Exemples :

a) Cas où il n'existe en mémoire ni GDT-well, ni GDT-model

```
-> SUBCOMMAND = ?
- ANADIA
-> CURRENT GDT-WELL = ?
- WELL.DTI
-> CURRENT-TOOLS :
  SP    GR    NPHI    LLS
  WARNING    NO    CURRENT MODEL
  A#
```

b) Cas où il existe en mémoire un GDT-well et un GDT-model

```
-> SUBCOMMAND = ?
- ANADIA
-> CURRENT GDT-WELL : WELL.DTI
  CURRENT GDT-MODEL : MODEL.DT
  CURRENT TOOLS :
  SP    GR    NPHI    LLS
  WORKING TOOLS :
  SP    GR    LLS
  A#
```

A ce niveau (A#), l'utilisateur se trouve devant un grand choix de sous-commandes : toutes celles qui sont accessibles au niveau "SUBCOMMAND = ?" plus quatre autres, les sous-commandes RUN, VISU, SMTH et INCLUD.

Il y a une particularité pour les sous-commandes WELLDEF, MODDEF et TOOLS : comme elles sont susceptibles de modifier le working-tools, le système édite le nouveau working-tools.

13) La sous-commande RUN

Fonction :

La sous-commande RUN sert à lancer les calculs de probabilité. Les résultats (les probabilités plus le faciès le plus probable) sont stockés en mémoire centrale. Ils pourront être ensuite visualisés à l'écran, stockés dans le GDT-well ou écrits dans un fichier séquentiel.

Appel de la sous-commande :

Cette sous-commande n'est accessible qu'au niveau "A#"

Organisation :

Le système teste en premier lieu s'il y a un GDT-model en mémoire. Sinon, il appelle automatiquement la sous-commande MODDEF, édite le nouveau working-tools et revient au niveau "A#".

S'il y a un modèle, le système teste s'il existe des résultats en mémoire qui n'ont pas été sauvés dans le GDT-well.

Si oui, il demande à l'utilisateur s'il veut les sauver dans le GDT-well en posant la question :

INCLUD CURRENT RESULTS (Y/N) ?.

Si l'utilisateur répond "Y" à cette question, les résultats sont sauvés. Le système commence alors le calcul des probabilités.

14) La sous-commande SMTH

Fonction :

Cette sous-commande est utilisée pour lisser les probabilités obtenues à l'aide de la sous-commande RUN. Une moyenne glissante est calculée pour la probabilité de chaque faciès le long du sondage.

Appel de la sous-commande :

Cette sous-commande n'est accessible qu'au niveau "A#".

Organisation :

Le système teste d'abord s'il existe en mémoire des résultats calculés à partir du GDT-well courant et donc susceptibles d'être lissés.

Ensuite, il demande à l'utilisateur sur combien de niveaux il veut que soit calculée la moyenne glissante.

NUMBER OF NODES = ?

Le calcul est alors fait sur l'ensemble des résultats en mémoire. L'utilisateur doit utiliser la sous-commande VISU pour visualiser les nouveaux résultats (probabilités et faciès le plus probable).

15) La sous-commande VISU

Fonction :

Cette sous-commande permet d'éditer les résultats obtenus à partir de la sous-commande RUN. Elle est un mini-éditeur. Elle permet aussi de modifier au gré de l'utilisateur le nom de faciès résultat à une profondeur quelconque sur le sondage.

Appel de la sous-commande :

Cette sous-commande n'est accessible qu'au niveau "A#".

Organisation :

La première condition est qu'il existe des résultats à éditer, sinon le système revient au niveau "A#".

A l'appel de cette sous-commande, le système édite un témoin d'attente "V#". L'utilisateur doit alors frapper une instruction. Le système l'exécute et revient au niveau "V#". Pour sortir de cette sous-commande il suffit de frapper la touche <CR>.

Le dernier niveau de mesures du sondage à être traité par une instruction devient le niveau courant. A l'entrée dans la sous-commande, c'est le sommet du sondage s'il n'y a pas de set défini. C'est le sommet du set s'il y en a un de défini.

Les instructions :

a) instruction L (LIST)

Cette instruction permet d'éditer les résultats obtenus par la sous-commande ANADIA. Ce sont, d'une part, les probabilités pour chaque niveau du sondage d'être de telle ou telle nature lithologique, et d'autre part, la nature lithologique (communément appelée faciès dans ce travail) la plus probable.

syntaxe :

- L édition des résultats du niveau courant
- L n édition des résultats du niveau courant dont le numéro d'ordre dans le GDT-well est n.
- L n1 n2 édition des résultats des niveaux dont les numéros sont compris entre n1 et n2.
- L (x) édition des résultats du niveau dont la profondeur est la plus proche de x

L [x,y] édition des résultats des niveaux dont les profondeurs sont comprises entre x et y.

Ex. L 10
L [2035.1, 2135.5]

b) instruction C (CHANGE)

Cette instruction permet de modifier le nom du faciès le plus probable (stocké comme résultat de l'analyse diagraphique) d'un ou de plusieurs niveaux.

Syntaxe :

- C FAC donne le nom de faciès FAC au niveau courant.
- C n1 FAC donne le nom de faciès FAC au niveau dont le numéro d'ordre dans le GDT-well est n1
- C n1 n2 FAC donne le nom de faciès FAC aux niveaux dont les numéros sont compris entre n1 et n2
- C (x) FAC donne le nom de faciès FAC au niveau dont la profondeur est la plus proche de x
- C [x1,x2] FAC donne le nom de faciès FAC aux niveaux dont les profondeurs sont comprises entre x1 et x2

c) instructions D (DOWN) et U (UP)

Ces instructions permettent de se positionner sur le même niveau au dessous du niveau courant (instruction D) ou au-dessus du niveau courant (instruction U) et d'en éditer les résultats. Dans le cas de la présence d'un set, seuls les niveaux appartenant à ce set sont comptés.

syntaxe :

- U édition des résultats du niveau directement supérieur au niveau courant
- U n édition des résultats du même niveau au-dessus du niveau courant
- D édition des résultats du niveau directement inférieur au niveau courant
- D n édition des résultats du même niveau au-dessous du niveau courant

d) instruction TITLE

Cette instruction permet d'éditer le titre des résultats édités par l'instruction LIST.

e) instructions NUM et NONUM

NUM Fait précéder chaque ligne éditée du numéro du niveau correspondant

NONUM permet de supprimer l'option précédente. Cette dernière option est prise par défaut.

f) instruction S (SAVE)

Cette instruction permet de sauver des résultats dans un fichier séquentiel "image carte" (80 caractères/enregistrement).

syntaxe :

S DEF Définition du fichier de sauvegarde. Le système pose la question

 SAVING FILE NAME = ?

à laquelle l'utilisateur doit répondre par le nom du fichier séquentiel "nouveau" dans lequel seront effectuées les sauvegardes ultérieures.

S L ... (S suivi d'une instruction LIST) sauvegarde des éditions de l'instruction LIST dans le fichier prévu à cet effet. Si aucun fichier n'a été défini, le système pose au préalable la question

 SAVING FILE NAME = ?

S TITL sauvegarde de la ligne de titre dans le fichier. Même chose que pour S L..., si aucun fichier n'a été défini.

Remarque relative à toutes ces instructions :

Le numéro du niveau courant peut être remplacé dans la ligne d'instruction par le symbole "*". La lettre "B" symbolise le dernier niveau du sondage (BOTTOM en anglais). Le symbole "B" n'est pas utilisable dans le cas de la présence d'un set.

Enfin, l'utilisateur a toujours la possibilité :

- de taper "STOP" pour arrêter la sous-commande et revenir au niveau "SUBCOMMAND = ?"
- de taper "?" pour savoir les instructions qu'il peut utiliser.

Exemple d'utilisation de la sous-commande VISU :

- VISU
- > V#

```

- TITL
-> DEPTH  FACIES  GRM  TI  GRG  AA
- V#
- L 1
-> 1999.9  GRM  0.543  0.039  0.418  0.000
- V#
- NUM
-> V#
- TITL
-> SU NUMB  DEPTH  FACIES  GRM  TI  GRG  AA
- V#
- L * 2
-> 1 1999.9  GRM  0.543  0.039  0.418  0.000
   2 2000.1  GRM  0.634  0.052  0.251  0.063
- V#
- C AA (ou C * AA ou C 2 AA)
-> V#
- L
-> 2 2000.1  AA  0.634  0.052  0.251  0.063
- V#
- D 2
-> 4 2000.5  GRM  0.575  0.95  0.309  0.021
- V#
- U 2
-> 2 2000.1  AA  0.634  0.052  0.251  0.063
- V#
- S L 1 10
-> SAVING FILE NAME = ?
- FILSAV
-> V#
- S DEF
-> SAVING FILE NAME = ?
- SAVFIL
-> V#
- S L 100 110
-> V#

```

16) La sous-commande INCLUD

Fonction :

Cette sous-commande sert à stocker des résultats de ANADIA sous forme de composantes dans le GDT-well analysé.

Appel de la sous-commande :

Cette sous-commande peut être appelée par l'utilisateur au niveau "A#". Elle peut aussi être appelée par le système chaque fois que l'utilisateur risque de perdre les résultats courants :

a) lorsque l'utilisateur veut calculer de nouveau des probabilités avec la sous-commande RUN avant d'avoir sauvé les résultats obtenus précédemment.

b) lorsque l'utilisateur est sur le point de sortir de la commande et que les derniers résultats obtenus n'ont pas été stockés ni sauvés.

c) lorsque l'utilisateur est sur le point de charger de nouvelles données.

En effet, le programme ne permet pas de stocker des résultats dans un autre sondage que celui dont ils proviennent. Aussi, dans ces trois cas, on demande à l'utilisateur s'il veut stocker les derniers résultats avec la question

INCLUDE CURRENT RESULTS (Y/N) ?.

Organisation :

Dans le cas d'un appel direct, le système teste s'il existe des résultats, sinon il envoie un message d'erreur approprié et revient au niveau "A#".

Toujours dans le cas d'un appel direct, l'utilisateur peut introduire les "options de stockage" qu'il désire à la suite de nom de sous-commande INCLUD. S'il ne le fait pas, alors le système, comme dans le cas d'un appel automatique, pose la question

RESULTS = ?

L'utilisateur doit y répondre par une option de stockage telle que

PROB, Fn ou FFn.

exemples :

a) commande directe

- > A#
- INCLUD PROB
- > stockage des probabilités.

b) commande indirecte

- > A#
- INCLUD
- > RESULTS = ?
- PROB
- > stockage

- c) appel automatique
- > A#
- RUN
- > INCLUDE CURRENT RESULTS (Y/N) ?
- Y
- > RESULTS = ?
- PROB
- > stockage

Les options de stockage :

a) PROB signifie que toutes les probabilités et le faciès le plus probable seront stockés dans des composantes du GDT-well.

noms des composantes	: ARGILE	: GRES	: CALCAIRE	:
ensemble des probabi-	: 0.5	: 0.01	: .49	:
lités à un niveau du	: 0.33	: 0.21	: 0.46	:
sondage	:	:	:	:

b) Fn signifie que les nièmes faciès les plus probables seront stockés dans les composantes du GDT-well.

Fn = lettre "F" suivie immédiatement d'un nombre entier n

- exemple :
- > A#
 - INCLUD F9
 - > suite de la sous-commande

```

nom des composantes      :   F1   :   F2   :   F3   :
générées automatiquement :         :         :         :
-----
                        : CALCAI : ARGILE : GRES   :
-----
un niveau du sondage    : CALCAI : ARGILE : MARNES :
-----
                        : CALCAI : MARNES : ARGILES :
-----
                        : MARNES : CALCAI : ARGILES :
-----
                        : MARNES : ARGILE : MARNES :
-----
                        :       :       :       :
                        : faciès : 2e faciès : 3e faciès :
                        : le plus : le plus : le plus :
                        : probable : probable : probable :

```

c) FPn signifie que les nièmes faciès les plus probables et leurs probabilités seront stockés dans des composantes dont les noms sont soit générés automatiquement, soit donnés par l'utilisateur.

FPn = sigle "FP" suivi immédiatement d'un nombre entier n.

exemple :

```

-> A#
- INCLUD
-> RESULTS = ?
- FP5

```

✍

```

noms des composantes :      :      :      :      :
générées automatique- : F1  : P1  : F2  : P2  :
ment                       :      :      :      :      :
-----
: CALCAI : 0.935 : ARGILE : 0.01 :
-----
un niveau du sondage : CALCAI : 0.844 : ARGILE : 0.151 :
-----
: CALCAI : 0.630 : MARNES : 0.344 :
-----
: etc   : etc   : etc   : etc   :
-----
:      :      :      :      :
: faciès : proba- : 2e faciès: proba- :
: le plus : bilite : le plus : bilite :
: probable : du faciès: probable : du 2eme :
:      : le plus :      : faciès :
:      : probable :      : le plus :
:      :      :      : probable :

```

Dans les cas de réponse "Fn" et "FPn", le système vérifie si le nombre n n'est pas plus grand que le nombre de faciès. Si oui, il édite un message d'erreur et pose la question
RESULTS = ?

Dans tous les cas de réponse, le système vérifie s'il y a assez de composantes libres dans le GDT-well pour y stocker les résultats demandés. Si non, il édite un message d'erreur et pose à nouveau la question "RESULTS = ?".

Puis le système pose la question suivante :

DEFINE COMPONENTS NAMES (Y/N) ?

qui demande à l'utilisateur s'il veut donner un nom à chacune des nouvelles composantes résultat.

Dialogue dans ces trois cas de réponse :

a) Cas de réponse "PROB"

Si l'utilisateur a répondu "PROB" à la question "RESULTS = ?" et a répondu "Y" à la question "DEFINE COMPONENT'S NAMES (Y/N) ?", alors le système demande pour chaque composante nouvelle un nom et un titre.

Exemple :

- > NAME FOR ARGILE PROBABILITY COMPONENT ?
- ARG
- > TITLE = ?
- PROBABILITE DE L'ARGILE OBTENUE AVEC LE MODELE MODEL.DT
- > NAME FOR CALCAI PROBABILITY COMPONENT ?
- etc ...

Si l'utilisateur donne un nom de composante qui existe déjà dans le GDT-well, alors le système le signale et demande à nouveau un nom de composante.

Si l'utilisateur a répondu "N" à la question
DEFINE COMPONENTS NAMES ? (Y/N)
alors le système choisit automatiquement les noms de composantes. Il donne comme nom à chaque composante-probabilité d'un faciès le nom de faciès complet (6 caractères). Ex. : la composante-probabilité que la roche soit une argile s'appellera ARGILE. La dernière composante, celle qui contient pour chaque niveau du sondage le faciès le plus probable sera appelée FACIES.

Après avoir généré chaque nom de composante, le programme vérifie s'il n'existe pas déjà dans le GDT-well courant une composante de ce nom. Si oui, il quitte à ce niveau le mode automatique et revient en mode manuel. C'est à dire que pour la composante en question et toutes les suivantes, il demande à l'utilisateur un nom et un titre.

b) Cas de réponse Fn ou FPN

* si l'utilisateur répond "Y" à la question
DEFINE COMPONENTS NAMES (Y/N) ?
le système demande pour chaque composante le nom et le titre.

Exemple :

- > NAME FOR FACIES 1 COMPONENT = ?
- réponse de l'utilisateur
- > TITLE = ?
- ...
- > NAME FOR PROBABILITY 1 COMPONENT = ?
- ...
- > TITLE = ?
- ...
- > NAME FOR FACIES 2 COMPONENT = ?
- etc ...

De même que dans le cas "PROB", si l'utilisateur donne un nom de composante qui existe déjà dans le GDT-well, alors le système réitere sa question.

*Si l'utilisateur répond "N" à la question
DEFINE COMPONENT'S NAMES ?

le programme génère automatiquement :

le nom "F1" pour le faciès le plus probable,
le nom "P1" pour la probabilité qui lui est associée,
le nom "F2" pour le deuxième faciès le plus probable,
etc...

De même que dans le cas "PROB", si un nom de composante généré par le programme tel que F1, P1, F2, etc. existe déjà dans le GDT-well, alors le programme quitte le mode automatique et revient au mode manuel pour la composante courante et toutes les suivantes.

Après toutes ces questions, l'ordinateur insère les résultats dans le GDT-well et revient au niveau "A#".

17) La sous-commande EDMOD

Fonction :

C'est l'éditeur de modèles. Elle donne accès à des instructions d'édition, de modifications du modèle courant, de cross-validation.

Appel de la sous-commande :

Elle peut être appelée d'une façon directe par l'utilisateur aux niveaux "SUBCOMMAND = ?" et "A#". Elle est aussi appelée automatiquement par la sous-commande MODBLD.

Organisation :

S'il n'y a pas de modèle en mémoire centrale, le système appelle automatiquement la sous-commande MODDEF. Ensuite il édite le témoin d'attente "M#".

L'utilisateur doit alors entrer une instruction : le système l'exécute et revient ensuite au niveau "M#". Pour sortir de l'éditeur, il faut frapper la touche <CR>.

Au contraire de la sous-commande EDWELL, la sous-commande EDMOD ne permet pas à l'utilisateur de choisir librement les composantes avec lesquelles il va travailler : si un GDT-well est chargé en mémoire centrale, alors les composantes éditées seront les composantes du modèle présentes dans le "CURRENT-TOOLS". Ainsi, l'utilisateur peut régler à peu près les composantes qu'il veut éditer. Si aucun GDT-well n'est chargé, alors toutes les composantes du modèle sont systématiquement prises en compte.

On appelle noyau courant le dernier noyau traité par une instruction. Lorsqu'une instruction ne mentionne pas de noyaux, celui-ci est traité par défaut. A l'entrée dans la sous-commande, le noyau courant est le premier du modèle.

Les instructions :

a) instruction L (LIST)

Cette instruction permet d'éditer le numéro, la "profondeur" s'il y a lieu, le nom de faciès, le poids et les coordonnées d'un ou de plusieurs noyaux. La "profondeur" d'un noyau n'existe que si le modèle courant a été créé pendant la session.

syntaxe :

L édition du noyau courant

L n1 édition du noyau de numéro n1

L n1 n2 édition des noyaux des numéros n1 à n2

b) instruction CV

Cette instruction permet de faire la cross-validation d'un ou de plusieurs noyaux.

CV cross-validation générale. On donne pour le noyau à traiter les probabilités qu'il porte chaque nom de faciès existant dans le modèle.

CVR cross-validation réduite. Quand le noyau à traiter a un nom de faciès et n'est pas le seul à porter son nom de faciès, on donne la probabilité qu'il porte effectivement ce nom de faciès ; dans les deux cas, ces probabilités sont calculées à partir des autres noyaux du modèle.

syntaxe :

CV (ou CVR) cross-validation pour le noyau courant.

CV (ou CVR) n1 cross-validation pour le noyau de numéro n1

CV (ou CVR) n1 n2 cross-validation pour les noyaux dont les numéros sont compris entre n1 et n2

c) instruction LCV

Cette instruction a les effets cumulés de l'option L et l'option CV. La syntaxe est la même que pour ces deux options. Il existe aussi l'instruction LCVR.

d) instruction C (CHANGE)

Cette instruction permet de changer le nom de faciès d'un noyau ou de plusieurs noyaux.

syntaxe :

- C FAC attribue le nom de faciès FAC au noyau courant.
- C n1 FAC attribue le nom de faciès FAC au noyau de numéro n1.
- C n1 n2 FAC attribue le nom de faciès FAC aux noyaux dont les numéros sont compris entre n1 et n2.

e) instructions DOWN (D) et UP (U)

Ces instructions permettent de se positionner sur le nième noyau suivant le noyau courant (D) ou avant le noyau courant (U), et de l'éditer.

syntaxe :

- U édition du noyau précédant le noyau courant
- U n édition du nième noyau précédant le noyau courant
- D édition du noyau suivant le noyau courant
- D n édition du nième noyau suivant le noyau courant

f) instruction DEL

Cette instruction permet de supprimer ou ou plusieurs noyaux.

syntaxe :

- DEL suppression du noyau courant
- DEL n suppression du noyau de numéro n
- DEL n1 n2 suppression des noyaux dont les numéros sont compris entre n1 et n2 inclus

Le noyau courant devient le noyau qui précédait celui ou ceux qui ont été supprimés (sauf dans le cas où le numéro 1 été supprimé, bien sur).

g) instruction TITL

Cette instruction permet d'éditer le titre des lignes produites par l'instruction LIST. Ce titre commence par :
NO DEPTH (éventuellement) FACIES WGHT

WGHT signifie WEIGHT (poids en anglais)
puis le système imprime les identificateurs correspondant aux composantes choisies du modèle.

h) instruction LITO

Cette instruction permet d'éditer la liste de tous les noms de faciès permis.

i) instruction S (SAVE)

Cette instruction permet de sauver des éditions obtenues avec les instructions LIST, CV, LCV, TITL et LITO dans un fichier séquentiel "image carte" (80 caractères/enregistrements).

syntaxe :

S DEF définition du fichier de sauvegarde
 le système pose la question

 SAVING FILE NAME = ?

à laquelle l'utilisateur doit répondre par le nom d'un nouveau fichier séquentiel dans lequel seront effectuées les sauvegardes ultérieures.

- S L ... sauvegarde des éditions des
- S CV ... instruction L, CV, LCV, TITL et
- S LCV ... LITO dans le fichier
- S TITL de sauvegarde
- S LITO

Si aucun fichier de sauvegarde n'a encore été défini, alors le système pose au préalable les questions

 SAVING FILE NAME = ?

Si l'on sort de l'éditeur et que l'on y revient, il n'est plus possible de sauver dans le dernier fichier de sauvegarde et il faut en définir un nouveau.

Exemple d'utilisation :

```

-> SUBCOMMAND = ?
-   EDMOD
-> M#
-   TITL
-> NO   FACIES   WGHT   SP   GR   NPHI
    M#

-   L 1
-> 1   GRM   0.017  21.44  55.23  30.589
    M#

-   LCV
-> 1   GRM   0.017  21.44  55.23  30.589
    FACIES *   PROBABILITY
      GRM  *   0.587
      TI   *   0.343
      GRG  *   0.068
      AA   *   0.002
    M#

-   CV
-> KERNEL NO : 1
    FACIES NAME : GRM
    FACIES *   PROBABILITY
  * GRM  *   0.587
    TI   *   0.343
    GRG  *   0.068
    AA   *   0.002
    M#

-   D
-> 2   GRM   0....  -   -   -
    M#

-   C   AA
-> M#

-   L
-> 2   AA   0...  -   -   -
    M#

```

18) La sous-commande WELLSAV

Fonction :

Cette sous-commande effectue la sauvegarde sur disque du "Current GDT-well".

Appel de la sous-commande :

Elle est appelée directement à partir des niveaux "SUBCOMMAND = ?" et "A#". Elle peut aussi être appelée automatiquement par le logiciel lorsqu'il existe en mémoire centrale un GDT-well contenant des résultats que l'utilisateur est sur le point de perdre, c'est à dire quand il appelle la sous-commande WELLDEF ou tente de sortir de la commande.

Organisation :

Elle n'est pas la même suivant que la sous-commande est appelée automatiquement ou pas.

a) appel direct par l'utilisateur

Le système teste d'abord s'il existe un GDT-well en mémoire. Sinon, il édite un message d'erreur et retourne au niveau "SUBCOMMAND = ?" ou "A#". Puis il édite le nom de ce GDT-well.

CURRENT GDT-WELL :

L'utilisateur peut sauver le "Current GDT-well" dans un autre fichier que le fichier de départ. Il n'a qu'à le préciser dans la ligne d'appel de la sous-commande.

exemple :

WELLSAV "file_name" : Le "Current GDT-well" est sauvé dans un fichier de nom "file_name". S'il existe déjà un fichier de nom "file_name", alors le système édite un message d'erreur et demande un nouveau nom de fichier.

WELLSAV : Le "Current GDT-well" est sauvé sous son nom courant.

Le système édite pour vérifications le nom du fichier de sauvegarde puis demande s'il faut changer le titre du GDT-well :

SAVING FILE NAME :

NEW TITLE ? (Y/N)

Si l'utilisateur répond "N", la sauvegarde s'effectue et le système revient au niveau d'appel. Si l'utilisateur répond "Y", le système demande avec la question

TITLE = ?

un titre qui est un commentaire de 72 caractères au plus. Le titre du fichier courant restant en mémoire n'est cependant pas modifié.

Exemple :

```

-> SUBCOMMAND = ?
- WELLSAV NEWWELL.DT1
-> CURRENT GDT-WELL : WELL.DT1
  SAVING FILE NAME : NEWWELL.DT1
  NEW TITLE (Y/N) ?
- Y
-> TITLE = ?
- nouveau titre du GDT-well

```

b) Dans le cas où cette sous-commande est appelé par le système, un GDT-well est toujours présent en mémoire. Le fichier est alors sauvé sous son nom courant sans modification du titre. Le système n'édite aucun message et passe à la suite des opérations.

19) La sous-commande MODSAV

Fonction :

Cette sous-commande effectue la sauvegarde sur disque du "Current GDT-model".

Appel de la sous-commande :

Elle est appelée directement à partir des niveaux "SUBCOMMAND = ?" et "A#". Elle peut être appelée automatiquement par le logiciel lorsqu'il existe un modèle en mémoire centrale qui a été créé ou modifié pendant la session et que l'utilisateur est sur le point de perdre. Ceci peut se passer dans trois cas :

- en sortie de la commande DIA
- à l'entrée de la sous-commande MODDEF
- à l'entrée de la sous-commande MODBLD

Organisation :

Elle n'est pas la même suivant que la sous-commande est appelée automatiquement ou pas.

a) Appel direct par l'utilisateur

S'il n'y a pas de modèle en mémoire centrale, le système édite un message d'erreur et revient au niveau "SUBCOMMAND = ?" ou "A#". Si oui, il en édite le nom.
CURRENT GDT-MODEL : ...

L'utilisateur peut sauver le "Current GDT-model" dans un autre fichier que le fichier de départ. Il n'a qu'à le préciser dans la ligne d'appel de la sous-commande.

exemple :

MODSAV "file_name" : Le "Current GDT-model" est sauvé dans un fichier de nom "file_name". S'il existe déjà un fichier de nom "file_name", alors le système édite un message d'erreur et demande un nouveau nom de fichier.

MODSAV : Le "Current GDT-model" est sauvé sous son nom courant.

Le système édite pour vérifications le nom du fichier de sauvegarde puis demande s'il faut changer le titre du GDT-well :

SAVING FILE NAME :
NEW TITLE ? (Y/N)

Si l'utilisateur répond "Y", le système demande ce titre (72 caractères maximum) avec la question

TITLE = ?

Le titre du fichier courant restant en mémoire n'est cependant pas modifié. Puis le GDT-model est sauvé sur disque et le système revient au niveau "SUBCOMMAND = ?" ou "A#".

Exemple :

```
-> SUBCOMMAND = ?
- MODSAV NEWMOD.DT
-> CURRENT GDT-MODEL : MODEL.DT
  SAVING FILE NAME : NEWMOD.DT
  NEW TITLE (Y/N) ?
- Y
-> TITLE = ?
- nouveau titre du GDT-model
```

b) Dans le cas où cette sous-commande est appelée par le système, un GDT-model est toujours présent en mémoire. Le fichier est alors sauvé sous son nom courant sans modification du titre. Le système n'édite aucun message et passe à la suite des opérations.

20) Les sous-commandes CROSSV et CROSSVR

Fonction :

Ces sous-commandes permettent de réaliser une cross-validation pour chaque noyau de tout un modèle. L'emploi de CROSSV entraîne une cross-validation générale et l'emploi de CROSSVR entraîne une cross-validation réduite.

Appel des sous-commandes :

Ces sous-commandes sont appelées uniquement par l'utilisateur depuis les niveaux "SUBCOMMAND = ?" et "A#".

Organisation :

Le programme teste s'il y a un modèle en mémoire centrale. S'il n'y en a pas, il appelle la sous-commande MODDEF.

S'il y a un GDT-well de défini, le programme utilisera uniquement les outils du modèle présent dans le "current-tools" du sondage pour effectuer les cross-validations. Sinon, il utilise tous les outils du modèle.

Exemple d'utilisation de la sous-commande CROSSV

```
-> SUBCOMMAND = ?
- CROSSV
-> CURRENT GDT-MODEL : MODEL
WARNING : NO DEFINED GDT-WELL
```

```
KERNEL NO : 1
FACIES NAME : AT
FACIES * PROBABILITY
AT * 0.564
AA * 0.321
TA ... * 0.052
TI ... * ... 0.032
GRM ... * ... 0.031
```

```
KERNEL NO : 2
FACIES NAME : AA
FACIES * PROBABILITY
AT * 0.368
AA * 0.352
TA * 0.253
TI * 0.227
GRM * 0.000
```

```
KERNEL NO : 3
```

```
.
.
.
.
```

etc ... jusqu'au dernier noyau du modèle.

Exemple d'utilisation de la sous-commande CROSSVR

```
-> SUBCOMMAND = ?
- CROSSVR
-> CURRENT GDT-MODEL = ?
- MODELE (par exemple)
-> WARNING NO DEFINED GDT-WELL
```

KERNEL NO : 1
FACIES NAME : AT PROBA : 0.564
KERNEL NO : 2
FACIES NAME : AA PROBA : 0.352
KERNEL NO : 3

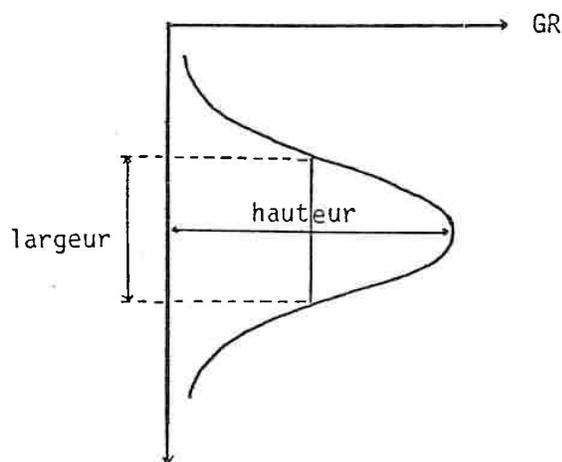
·
·
·

etc ... jusqu'au dernier noyau du modèle.

CONCLUSIONS

Les résultats obtenus sont aussi bons que ceux obtenus à la main, compte tenu que plusieurs diagraphistes peuvent trouver des colonnes lithologiques assez différentes à partir d'un même ensemble de logs et surtout des limites de couches différentes.

Remarques : Les géologues emploient des recettes pour estimer la largeur d'une couche. Par exemple, dans le cas des couches de sylvinite vues dans cet ouvrage, ils prennent pour largeur d'une couche la largeur à mi-hauteur du pic de gamma-ray engendré par cette couche.



Cette astuce remplace la déconvolution que l'on doit effectuer dans le cas d'une analyse automatique des diagraphies.

Outre les problèmes de déconvolution, la reconnaissance automatique des diagraphies nécessite que ces diagraphies aient été normalisées auparavant (cas de la polarisation spontanée, par exemple).

Le temps d'une analyse automatique est assez court (10 à 20 mn de temps CPU pour un puits). C'est ce qui en fait le principal intérêt. En effet, une analyse manuelle occupe un ou plusieurs géologues spécialistes pendant plusieurs heures, rendant le coût de l'analyse manuelle assez élevé.

L'affectation d'un faciès à chaque noyau d'un modèle serait fastidieux si l'utilisateur ne disposait pas d'utilitaires graphiques. Actuellement, les utilitaires graphiques inclus dans la commande ne sont pas perfectionnés. Avec les développements des matériels et logiciels graphiques, il sera possible d'ajouter à la commande des utilitaires graphiques interactifs.

Exemples : sélection d'un noyau à l'aide d'un curseur à l'écran graphique,
sélection d'un faciès à partir d'une liste de faciès affichée à l'écran conversationnel (menu).

La commande créée a des analogies avec les systèmes experts en ce sens que le savoir d'un expert est inclus dynamiquement dans la mémoire de la machine et peut être ensuite réutilisé par un utilisateur non expert.

Il serait intéressant d'ajouter à cette commande des véritables systèmes experts. Dans le cas de la détermination d'un noyau d'abord, un système expert pourrait déjà fournir une première approximation du faciès de ce noyau. On pourrait aussi utiliser un système expert qui arrangerait les résultats fournis par l'analyse automatique.

Exemple : Supposons que l'analyse automatique d'un sondage ait donné pour résultats, dans une zone donnée, la suite de faciès :

ARGILE
ARGILE
ARGILE
ARGILE
SILTS ARGILEUX
SILTS ARGILEUX
SILTS
SILTS
SILTS
SILTS

Le système expert pourrait "comprendre" que le faciès SILTS ARGILEUX n'est qu'un faciès de transition et qu'il y a seulement deux véritables couches : une d'argile et une de silts.

Dans ce même esprit, le système expert tenterait d'éliminer toutes les couches trop fines.

Comparaison de la commande créée avec le programme fait par la Société Schlumberger

Les deux programmes utilisent la même méthode : un modèle décrivant les zones occupées dans l'espace des variables par les faciès et qui calcule pour chaque point de cet espace une probabilité d'appartenance à chacune de ces zones.

Ces deux programmes diffèrent en deux points principaux :

1) Mémorisation des faciès

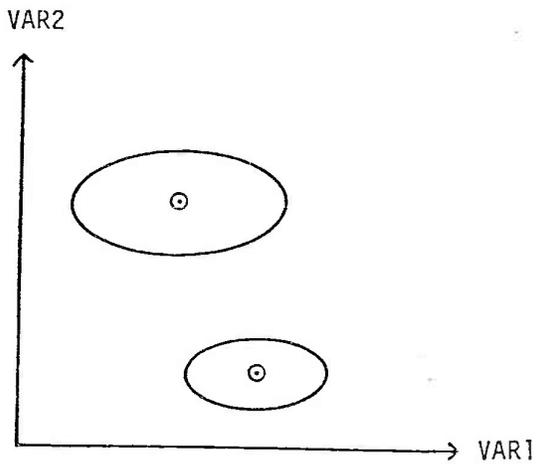
Schlumberger utilise comme modèle unique une base de données contenant les caractéristiques de tous les faciès possibles. Il est facile d'en extraire certains pour créer un nouveau modèle plus adapté au puits analysé. Cependant, le modèle n'est pas créé comme chez nous à partir d'un puits à analyser.

2) Discrimination des données

Chez Schlumberger, chaque classe du modèle n'est caractérisée que par un seul noyau auquel on applique une fonction gaussienne. En revanche, les gaussiennes n'ont pas toutes la même matrice de covariance.

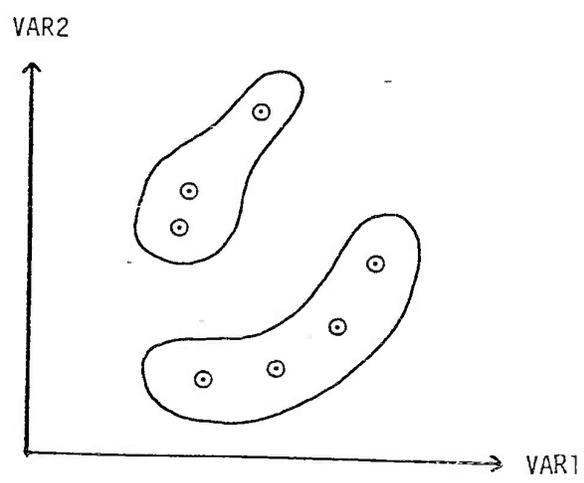
La forme des classes est une ellipse dont les allongements sont parallèles aux axes de coordonnées.

Dans notre programme, le fait qu'il existe plusieurs noyaux par classe permet de donner à celles-ci une forme quelconque.



modèle Schlumberger

Les classes ont des formes elliptiques



notre modèle

Les classes ont des formes quelconques

B I B L I O G R A P H I E

- { 1 } MALLET, J.-L., Propositions for fuzzy characteristic functions in data analysis, In : Caussinus, H., Ettinger, P. and Mathieu J.R. (eds), COMPSTAT 1982 (Physica-Verlag, Wien, 1982)
- { 2 } PARZEN, E., On estimation of a probability density function and mode, In : Annals of math. stat. (1962) XXXIII 1065-1076.
- { 3 } ROSENBLATT, M., Remarks on some non parametric estimates of a density function, In : Annals of math. stat. (1956) XXVII 832-835.
- { 4 } TAPIA, R.A. and THOMPSON, J.R., Non parametric probability density estimation (The Johns Hopkins University Press, Baltimore and London).
- { 5 } WILD, P., Estimation automatique de la densité multivariée et application à l'analyse des données, (Thèse de 3e cycle, I.N.P.L., Nancy, 1984)
- { 6 } MALLET, J.-L., WILD, P., An analog to correspondance analysis with fuzzy characteristic functions, In : "Computer in Earth Sciences for natural resource characterization", Sciences de la Terre, série Informatique, n° 20, 121-135 (CRPG, Nancy, 1984).
- { 7 } ROMEDER, J.-M., Méthodes et programmes d'analyse discriminante (Dunod, Paris, Bruxelles et Montréal, 1973).
- { 8 } DEBRANDES, R., Diagraphies dans les sondages (Publication de l'Institut Français du Pétrole, Technip, Paris, 1982).
- { 9 } SERRA, O., Diagraphies différées. Base de l'interprétation, Tome 1 : Acquisition des données diagraphiques (Bull. Centres Rech. Explor.-Prod. Elf Aquitaine, Mém. 1, Masson, 1979).
- { 10 } PERRODON, A., Géodynamique pétrolière. Genèse et répartition des gisements d'hydrocarbures, 2e éd. (Bull. Centres Rech. Explor.-Prod. Elf Aquitaine, Mém. 2, Masson, 1985).
- { 11 } BOUCHET, P., Définition et implantation d'un langage de manipulation de données, (Thèse de 3e cycle, I.N.P.L., Nancy, 1982).
- { 12 } MALLET, J.-L., Introducing to GEOL, (Document interne, E.N.S.G., Nancy, 1985).

- {13} JACQUEMIN, P., MALLET, J.-L., ROYER, J.-J., Interactive computer and design in the processing of mining and geological data, Int. Conf. CODATA proceedings (1984).
- {14} JACQUEMIN, P., ROYER, J.-J., MALLET, J.-L., GEOL : un logiciel interactif d'analyse, de manipulation de données et de cartographie automatique, In : Le Bulletin de MICADO, n° 71, 1984.
- {15} DELFINER, P.C., PEYRET, O., SERRA, O., Automatic determination of lithology from well logs, (Society of Petroleum Engineers of AIME, Dallas, 1984).
- {16} BONNET, A., GANASCIA, J.-G. and HARY, J., Application de Emycin à la Géologie : analyse de diagraphies (logs), le système Litho, Schlumberger, Clamart, 1982.
-

B I B L I O G R A P H I E

- { 1 } MALLET, J.-L., Propositions for fuzzy characteristic functions in data analysis, In : Caussinus, H., Ettinger, P. and Mathieu J.R. (eds), COMPSTAT 1982 (Physica-Verlag, Wien, 1982)
- { 2 } PARZEN, E., On estimation of a probability density function and mode, In : Annals of math. stat. (1962) XXXIII 1065-1076.
- { 3 } ROSENBLATT, M., Remarks on some non parametric estimates of a density function, In : Annals of math. stat. (1956) XXVII 832-835.
- { 4 } TAPIA, R.A. and THOMPSON, J.R., Non parametric probability density estimation (The Johns Hopkins University Press, Baltimore and London).
- { 5 } WILD, P., Estimation automatique de la densité multivariée et application à l'analyse des données, (Thèse de 3e cycle, I.N.P.L., Nancy, 1984)
- { 6 } MALLET, J.-L., WILD, P., An analog to correspondance analysis with fuzzy characteristic functions, In : "Computer in Earth Sciences for natural resource characterization", Sciences de la Terre, série Informatique, n° 20, 121-135 (CRPG, Nancy, 1984).
- { 7 } ROMEDER, J.-M., Méthodes et programmes d'analyse discriminante (Dunod, Paris, Bruxelles et Montréal, 1973).
- { 8 } DEBRANDES, R., Diagraphies dans les sondages (Publication de l'Institut Français du Pétrole, Technip, Paris, 1982).
- { 9 } SERRA, O., Diagraphies différées. Base de l'interprétation, Tome 1 : Acquisition des données diagraphiques (Bull. Centres Rech. Explor.-Prod. Elf Aquitaine, Mém. 1, Masson, 1979).
- { 10 } PERRODON, A. , Géodynamique pétrolière. Genèse et répartition des gisements d'hydrocarbures, 2e éd. (Bull. Centres Rech. Explor.-Prod. Elf Aquitaine, Mém. 2, Masson, 1985.
- { 11 } BOUCHET, P., Définition et implantation d'un langage de manipulation de données, (Thèse de 3e cycle, I.N.P.L., Nancy, 1982).
- { 12 } MALLET, J.-L., Introducing to GEOL, (Document interne, E.N.S.G., Nancy, 1985).

- {13} JACQUEMIN, P., MALLET, J.-L., ROYER, J.-J., Interactive computer and design in the processing of mining and geological data, Int. Conf. CODATA proceedings (1984).
 - {14} JACQUEMIN, P., ROYER, J.-J., MALLET, J.-L., GEOL : un logiciel interactif d'analyse, de manipulation de données et de cartographie automatique, In : Le Bulletin de MICADO, n° 71, 1984.
 - {15} DELFINER, P.C., PEYRET, O., SERRA, O., Automatic determination of lithology from well logs, (Society of Petroleum Engineers of AIME, Dallas, 1984).
 - {16} BONNET, A., GANASCIA, J.-G. and HARY, J., Application de Emycin à la Géologie : analyse de diagraphies (logs), le système Litho, Schlumberger, Clamart, 1982.
-

LISTE DES FIGURES

Figure 1. Sondages diagraphiques et leur interprétation lithologique.....	1
Figure 2. Représentation d'un point de données dans l'espace des variables.....	3
Figure 3. Séparation d'un faciès dans l'espace des variables. Cas d'un cross-plot GR-CNL.....	4
Figure 4.....	4
Figure 5.....	5
Figure 6. Résultats bruts de l'analyse.....	6
Figure 7. Exemple d'interprétation à deux faciès.....	7
Figure 8. Structure d'un GDT.....	12
Figure 9. Exemple de GDT-well.....	15
Figure 10. Exemple de GDT-model.....	17
Figure 11. Les noyaux dans l'espace paramétrique.....	23
Figure 12. Visualisation des courbes diagraphiques.....	25
Figure 13. Exemple de cross-plot.....	26
Figure 14. Espace partitionné en zones connues.....	26
Figure 15. Exemple de projection discriminante.....	27
Figure 16.....	28
Figure 17. Obtention du support S à partir de l'ensemble en utilisant la méthode des centres dynamiques	32
Figure 18. Partition de l'espace.....	34
Figure 19.....	35
Figure 21. Forme de la courbe $D_k(x)$ dans un espace à deux dimensions.....	40
Figure 22. Forme des indicatrices floues dans un espace à une dimension.....	41
Figure 23.....	41
Figure 24. Forme des indicatrices floues après rajout de la classe G_0	42
Figure 25. Influence des deux formules sur la forme des courbes.....	44
Figure 26. Exemple de courbe $D_k(x)$ en forme de "cloche".	46
Figure 27.....	46

Figure 49. Influence d'une couche de sylvinite sur le
Gamma-Ray..... 76

Figure 50..... 77

Figure 51..... 78

Figure 52..... 79

Figure 53..... 102

Figure 54. Visualisation des courbes
diagraphiques..... 104

Figure 55. Exemple de cross-plot..... 105

Figure 56. Exemple de projection discriminante..... 106

