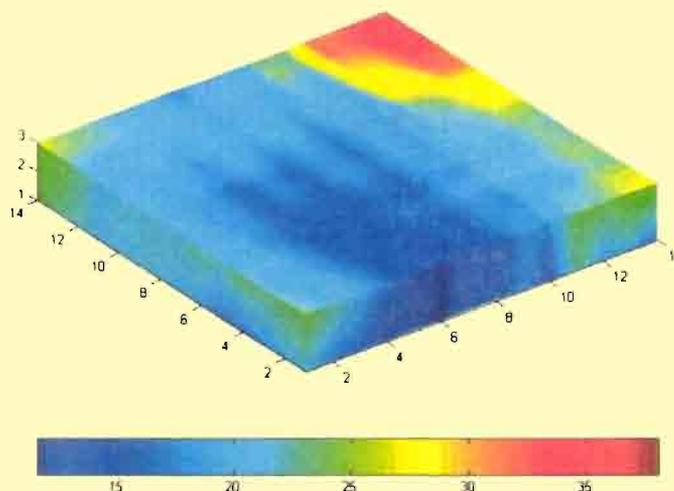


Mémoire de DESS de Géophysique Appliquée
présenté par Nordine BOUZID

**Utilisation de Matlab pour le formatage et le
traitement des données électriques 3D et des
données TDEM**



Année Universitaire 2001-2002

Université Pierre et Marie Curie Paris 6

Département de Géophysique Unité Mixte de Recherche Sisyphe

Institut de Recherche et de Développement (Bondy)

Mémoire de DESS de Géophysique Appliquée
présenté par Nordine BOUZID

**Utilisation de Matlab pour le formatage et le
traitement des données électriques 3D et des
données TDEM**

Année Universitaire 2001-2002

Remerciements

Je tiens ici à exprimer ma profonde gratitude à Monsieur Pierre Andrieux, directeur du DESS de Géophysique Appliquée, pour toute l'aide et les conseils qu'il m'a promulgué lors de ce travail et de m'avoir inciter « à me mettre à la programmation », je remercie monsieur Henri Robain, directeur de l'Unité de Recherche Geovast (IRD, Bondy), pour son soutien et « ses aides en lignes » lors de la mission d'Agen, de la gentillesse et de la sympathie qu'il me témoigne à chacune de nos rencontres, et pour les autres missions auxquelles il m'a permit de participer (Kerbernez, Dannemark). Mes remerciements vont également à Monsieur Yves Albouy, partager son expérience de terrain est toujours très enrichissante, pour sa gentillesse et sa disponibilité de tous les instants. Le travail et les résultats de la prospection effectuée à Agen, est le fruit d'une intense et fructueuse collaboration avec Monsieur Cédric Caron que je remercie ici vivement. Enfin, j'exprime ma reconnaissance à Monsieur Roger Guérin, maître de conférence à l'Université Pierre et Marie Curie, pour sa disponibilité, ses conseils et ses précieuses corrections. J'adresse également mes plus sincères remerciements à tous ceux qui de près ou de loin ont participé à ce mémoire.

SOMMAIRE

Sommaire.....	1
Liste des figures.....	3
Liste des tables.....	3
I Introduction	4
II Brève introduction au langage Matlab.....	5
II.1 Démarrage.....	5
II.1.1 Lancement rapide	5
II.1.2 Aides et commandes utiles	6
II.1.3 Commandes de type calculatrice	7
II.1.4 Exécution des programmes.....	7
II.2 Bref aperçu du langage	8
II.2.1 Premiers éléments.....	8
II.2.2 Manipulation des matrices.....	8
II.2.2.a Création d'une matrice.....	8
II.2.2.b Extraction des valeurs d'une matrice	9
II.2.2.c Construction de matrices de taille croissante	9
II.2.3 Opération sur les matrices	10
II.2.3.a Transposition	10
II.2.3.b Opération scalaire-matrice	10
II.2.4 Opérations entre matrices	11
II.2.4.a Multiplication	11
II.2.4.b Division.....	11
II.2.4.c Division élément par élément	12
II.2.5 Matrices particulières	12
II.2.6 Caractéristiques des matrices	12
II.3 Entrées/Sorties (E/S).....	13
II.3.1 Ecran d'E/S	13
II.3.2 Fichier d'E/S	13
II.3.2.a E/S propres à Matlab.....	13
II.3.2.b E/S standards	13
II.4 Programmer avec Matlab	14
II.4.1 Opérateurs logiques et de relation	14
II.4.2 Instruction et commandes structurées.....	14
II.4.2.a Instruction <i>for</i>	14
II.4.2.b Instruction <i>while</i>	15
II.4.2.c Instruction de sélection <i>if</i>	15
II.4.2.d Instruction de rupture de séquence.....	16
II.4.2.e Autre instruction	16
II.4.3 M-Files ou scripts.....	16
II.4.4 Fonctions	17
II.5 Graphisme	18
II.5.1 Graphique 2D	18
II.5.1.a Courbes : <i>plot</i>	18
II.5.1.b Histogramme : <i>hist</i>	20
II.5.2 Graphiques 3D.....	20
II.5.2.a Ligne dans l'espace : <i>plot3</i>	20
II.5.2.b Grillage en perspective : <i>mesh</i>	21
II.5.2.c Surface avec illumination : <i>surf</i>	22
II.5.2.d Courbe de niveau : <i>contour</i>	22
II.6 « Debugger ».....	23
II.7 Conclusion	24

III	Traitement des prospections multi-électrodes 2D et 3D	25
III.1	Contexte de l'étude	25
III.2	La méthode électrique	26
III.2.1	Principe	26
III.2.2	Les différents dispositifs utilisés	28
III.2.2.a	Le pôle-pôle	28
III.2.2.b	Le pôle-dipôle	28
III.2.2.c	Les quadripôles	29
III.2.3	Matériel utilisé	30
III.3	Le panneau électrique en 2D	31
III.3.1	Principe	32
III.3.2	Mode opératoire	32
III.3.3	Traitement des données 2D	33
III.3.4	Présentation et interprétation des résultats	36
III.4	Le panneau électrique 3D	39
III.4.1	Dispositif et mise en oeuvre	39
III.4.2	Utilisation de Matlab pour le formatage des fichiers 3D	40
III.4.2.a	Format de fichier Electre	41
III.4.2.b	Format de fichier Res3Dinv	42
III.4.2.c	Structure du code	43
III.4.3	Traitement des données électriques 3D	44
III.4.4	Présentation et interprétation des résultats	44
III.4.4.a	Reconnaissance 3D	44
III.4.4.b	Essai de suivi temporel	47
III.4.5	Conclusion	49
IV	Formatage et la représentation des données TDEM	50
IV.1	Sondage électromagnétique en domaine temporel : méthode TDEM	50
IV.1.1	Principe	50
IV.1.2	Sensibilité	52
IV.1.3	Les limites d'utilisation	52
IV.2	Appareil de mesure	53
IV.2.1	L'émetteur NT-20	53
IV.2.2	Le récepteur GDP32II	53
IV.3	Paramètres d'acquisition	54
IV.3.1	Sample Rate (pas d'échantillonnage)	54
IV.3.2	Récepteur et émetteur	55
IV.3.3	Turn-off time (temps de coupure)	55
IV.3.4	Robust Filter	57
IV.3.5	Gain Mode	57
IV.3.6	Fréquence (repetition rate) et cycle (séquence d'acquisition)	57
IV.3.7	Détermination de la position de la 1 ^{ère} fenêtre temporelle	58
IV.4	Format des fichiers de données issus du GDP32II	60
IV.5	Structure du code Matlab	63
IV.6	Exemples d'applications	64
IV.6.1	Représentation des données acquises sur le site de Kerbernez (Bretagne, France)	64
IV.6.2	Représentation des données acquises à Aarhus (Danemark)	64
IV.7	Perspectives et développement	66
V	Conclusion	67
	Bibliographie	67

Liste des figures

Figure 1 : Illustration de la commande plot.....	18
Figure 2 : Customisation du graphique (légende, titre, titres des axes).....	19
Figure 3 : Histogramme d'une distribution normale	20
Figure 4 : Graphique 3D « ligne dans l'espace »	21
Figure 5 : Grillage 3D avec les contours sur le plan de base.....	21
Figure 6 : Surface avec illumination : <i>surfl</i>	22
Figure 7 : Représentation en lignes de niveau	22
Figure 8 : Représentation en lignes de niveau avec coloration inter-lignes de niveau.....	23
Figure 9 : Localisation du verger et implantation des panneaux électriques.....	26
Figure 10 : Principe du sondage électrique à courant continu.....	27
Figure 11 : Le dispositif pôle-pôle	28
Figure 12 : Le dispositif pôle-dipôle	28
Figure 13 : Le dispositif pôle-dipôle avec l'électrode infinie perpendiculaire au dispositif....	29
Figure 14 : Le dispositif wenner α	30
Figure 15 : Les dispositifs dipôle-dipôle et wenner β	30
Figure 16 : Le matériel utilisé pour le panneau électrique	31
Figure 17 : Principe du panneau électrique pour un wenner α (Loke, 1996)	32
Figure 18 : Vue du panneau électrique 2D de 128 m	33
Figure 19 : Exemple de filtrage des effets « C » avec X2ipi.....	35
Figure 20 : Planche synthétique des principaux résultats 2D.....	38
Figure 21 : Dispositif schématique 3D pour le dispositif pôle-pôle.....	39
Figure 22 : Vue d'un panneau électrique 3D.....	40
Figure 23 : Schéma illustrant les processus de formatage des données 3D sous Matlab	43
Figure 24 : Succession verticale des plans horizontaux issus de l'inversion avec Res3Dinv ..	46
Figure 25 : Sections horizontales (0 -0.0.35 m) : le suivi temporel : pôle-pôle	48
Figure 26 : Principe de la méthode TDEM (d'après Descloîtres, 1998)	51
Figure 27 : Vue de la face avant du GDP32II	53
Figure 28 : Détermination de la position de la 1 ^{ère} fenêtre temporelle.....	60
Figure 29 : Descriptif des formats de fichier NanoTem et Zerotem	62
Figure 30 : Structure du code de visualisation et formatage des données TDEM.....	63
Figure 31 : Représentation graphique de $\partial B/\partial t$ sous Matlab exportée au format <i>.bmp</i>	64
Figure 32 : Représentation graphique de $\partial B/\partial t$ (NanoTem) exportée au format <i>.emf</i>	65
Figure 33 : Représentation graphique de $\partial B/\partial t$ (ZeroTem) exportée au format <i>.emf</i>	65

Liste des Tables

Table 1 : Format de fichier Electre.....	41
Table 2 : Format de fichier Res3Dinv [®]	42
Table 3 : Taille de boucle émettrice et Tx Delay associés pour le NanoTem	55
Table 4 : Centre des fenêtres temporelles du NanoTem.....	59

I Introduction

La nature et la disposition des roches qui constituent la Terre sont connues à partir de l'étude des affleurements ou éventuellement de celle de prélèvements de sondages mécaniques plus ou moins profond. Dans bien des cas, l'information n'est pas directement accessible (terrains de recouvrement, espaces aménagés, etc...), la géophysique va alors fournir une grande variété d'outils qui permettront d'obtenir des éléments d'information à partir desquels on pourra compléter le modèle de la structure profonde des terrains. La géophysique dispose de méthodes de prospection non-destructive qui ont des thématiques minières (recherche de site minier), hydrogéologiques (connaissance d'aquifère), environnementales (caractérisation de zones de pollutions), géotechnique (étude de stabilité de terrain et de structure) et archéologique (reconnaissance de structures enfouies). Les résultats obtenus par les différents outils géophysique se présentent sous forme de données numériques qui pourront être formatées, traitées et représentées sous forme d'images (courbes, cartes, volume, etc.). Une grande partie du travail du géophysicien est la manipulation des fichiers de données dans le but de les convertir aux formats particuliers pour les logiciels de traitement et de représentation. L'objectif du stage réalisé au laboratoire de géophysique appliquée de Paris 6 était l'apprentissage, puis l'utilisation de Matlab pour répondre à des problèmes de format de fichier et de représentation de données. La première partie de ce mémoire est en quelque sorte une prise en main rapide de Matlab, présentation succincte qui contient tous les éléments pour débiter avec Matlab. Puis afin d'illustrer l'utilisation de Matlab, pour résoudre des problèmes précis, nous présenterons la prospection électrique 2D et 3D réalisée à Agen, et enfin l'apport de Matlab pour le formatage et la représentation de données TDEM.

II Brève introduction au langage Matlab

Matlab est un environnement facilitant le calcul matriciel complexe (MATrice LABoratory) produit par la société MathWorks (<http://www.mathworks.com>). Il est disponible sur plusieurs plateformes (Windows, Linux, etc.). C'est un langage relativement simple et très efficace, optimisé pour le traitement des matrices. Matlab contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques. On peut enrichir Matlab en ajoutant des « boîtes à outils » (*Toolbox*) qui sont des ensembles de fonctions supplémentaires, dédiées à des applications particulières (traitement de signaux, analyses statistiques, optimisation, etc.). Matlab est un langage interprété qui permet de développer des algorithmes rapidement, de visualiser des données (sous la forme de graphiques 2D ou 3D et d'images, voir de séquences d'images), et de réaliser des interfaces graphiques conviviales (*Graphical User Interface*). Les notes qui suivent ne constituent pas une référence exhaustive sur Matlab, mais soulignent les notions principales de manière succinctes et servent de guide pour « une prise en main rapide ». Le lecteur trouvera en annexe (Annexe I) un récapitulatif des principales fonctions et commandes utilisées dans les différents codes réalisés dans le cadre de ce mémoire de fin d'étude et pourra se référer aux différents ouvrages et liens Internet qui traitent de l'utilisation et de la programmation sous Matlab (Annexe I).

II.1 Démarrage

II.1.1 Lancement rapide

Matlab va interpréter les commandes que vous aurez sauvegardées dans un fichier texte avec l'extension *.m* (*toto.m* par exemple). Il vous faut donc créer un répertoire de travail (*C:\temp\jdupont* par exemple) pour y sauvegarder vos programmes, lancer Matlab et lui demander d'exécuter les commandes sauvegardées dans votre fichier "toto.m". Par défaut le répertoire de travail de Matlab (« *work* ») se trouve dans le répertoire où est installé Matlab par exemple : *C:\program files\matlab6p5\work*. Pour démarrer Matlab, il suffit de cliquer sur l'icône « *Matlab* » si vous êtes sous Windows, ou de taper la commande *Matlab* si vous êtes sous Unix. L'espace de travail de Matlab se présente alors sous la forme d'une fenêtre affichant un prompt *>>* à la suite duquel vous pouvez taper une commande qui sera exécutée après avoir tapé sur la touche *return* (*entrer* ou *ok*). Il faut indiquer à Matlab le répertoire dans lequel vous voulez travailler (celui qui contient vos programmes).

Pour ce faire deux méthodes :

- cliquez dans *File / Set Path / Browse* et sélectionnez votre répertoire de travail, puis *return*. Pour sortir de cette fenêtre, cliquez dans la croix en haut à droite ou dans la barre de menu : *File / Exit Path Brother*.
- après le prompt `>>` de Matlab utilisez les commandes :
pwd : pour connaître le nom du répertoire actif de Matlab.
cd titi : pour aller dans le répertoire *titi*.
cd. : pour remonter d'un cran dans l'arborescence.
dir : pour connaître le nom des fichiers et des répertoires contenus dans le répertoire actif de Matlab. Ces commandes classiques sous unix vous permettent de vous déplacer dans les répertoire et d'accéder au répertoire dans lequel vous voulez rendre Matlab actif, par exemple : *C:\temp\jdupont*.

II.1.2 Aides et commandes utiles

Vous pouvez également utiliser à tout moment l'aide de Matlab qui est très conviviale et bien documentée, pour cela vous pouvez cliquer sur le point ? de la fenêtre ou utiliser la commande *help* qui permet d'obtenir l'aide sur une fonction ou un toolkit (exemple : *help help*). Vous pouvez également vous servir de la boîte de dialogue *help* en choisissant *getting started* pour débiter avec Matlab.

Les commandes suivantes sont utiles pour effectuer une recherche ou connaître le type d'une variable située dans le *workspace* (espace de travail de Matlab) qui liste et garde en mémoire les variables créées par un programme ou l'utilisateur tant qu'elles ne sont pas effacées avec la commande *clear* :

- *lookfor* : recherche d'un mot clé dans toute la documentation (lent).
- *which* : localise fonctions et fichiers.
- *what* : liste des fichiers Matlab dans le répertoire courant (*work* par exemple).
- *exist* : vérifie si une fonction ou une variable existe dans le *workspace*.
- *who, whos* : liste des variables dans le *workspace*.
- *class* : affiche le type de la variable.

II.1.3 Commandes de type calculatrice

Dans le mode interactif, Matlab peut-être utilisé comme une « surpuissante » calculatrice scientifique. L'utilisateur dispose des opérations arithmétiques et d'un ensemble important de fonction de calcul numérique et de visualisation graphique.

Dans Matlab, on trouve des constantes prédéfinies comme : $Pi : 3.14159265358979$; Inf (*Infinite*) : nombre infini ; NaN (*Not a Number*) : n'est pas un nombre, exprime parfois une indétermination ; et une variable pré-définie ans : variable contenant la dernière réponse (*answer*). Une portion de ligne débutant par % est vue comme un commentaire.

Les opérations arithmétiques de base dans Matlab sont : + pour l'addition, - pour la soustraction, * pour la multiplication et / ou \ pour la division. La division à droite et la division à gauche ne donnent pas les mêmes résultats, ce sont donc deux opérations différentes que ce soit pour les scalaires ou pour les vecteurs et matrices. Ainsi, a / b : c'est a qui est divisé par b ; $a \setminus b$: c'est b qui est divisé par a.

```
>> 6 + 3.4           % Effectue l'opération demandée
>> x = -2.34 / 21    % Le résultat de l'opération est affecté à la variable x
>> x                % Permet d'afficher la variable x
>> y = 1 + y.^3      % Opération de mise à une certaine puissance, les variables peuvent
                    % être réaffectées à d'autres valeurs
>> y = cos (x)       % Utilisation de la fonction cosinus
```

Si vous voulez rappeler une commande précédemment tapée, il vous suffit d'utiliser les flèches du curseur ↑ (revient aux commandes précédentes) et ↓ (affiche les commandes suivantes).

II.1.4 Exécution des programmes

Si Matlab est actif dans votre répertoire de travail et que celui-ci contient un fichier texte d'extension *.m*, il suffit de taper le nom du fichier après le prompt >> pour exécuter les commandes sauvegardées dans le fichier. Par exemple tapez :

```
>> toto
```

Attention, vos noms de fichiers ne devront pas contenir des caractères exotiques tels que les accents ou les espaces, sinon Matlab ne pourra pas les exécuter correctement. Veillez par ailleurs à ne pas utiliser des noms de commande trop simple. Si vous sauvegardez votre fichier comme *max.m*, lorsque vous taperez *max* dans la fenêtre de Matlab, il ne saura pas s'il doit interpréter les lignes de commandes contenues dans le fichier *max.m* ou exécuter la commande *max* (évaluation du maximum d'une matrice). Le plus simple est de sauvegarder vos fichiers en prenant comme première lettre un caractère particulier pour éviter toute confusions, *k_toto.m* par exemple ; ou d'utiliser des noms de fichier utilisant le français, comme *moyenne.m* par exemple (en effet, les commandes prédéfinies ont des termes anglais).

II.2 Bref aperçu du langage

II.2.1 Premiers éléments

Avant toutes choses, votre fichier doit débiter par la commande *clear*. Elle permet d'effacer toutes les variables chargées dans la mémoire de Matlab. Le fait de taper la commande `>> var = 3 ;` affecte à la matrice de taille 1x1 la valeur 3. Si on omet le point-virgule à la fin de la commande, la valeur de *var* est affichée après exécution. Une portion de ligne débutant par `%` est vue comme un commentaire. Pour interrompre une ligne d'instruction et la poursuivre à la ligne suivante, taper

Par exemple :

```
>> A = [1 2 3 4 5 ...  
6 7 8] équivaut à A = [1 2 3 4 5 6 7 8]
```

II.2.2 Manipulation des matrices

Les commandes suivantes illustrent les différentes manipulations que l'utilisateur peut effectuer sur les matrices.

II.2.2.a Création d'une matrice

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9] % La virgule (ou l'espace) sépare les colonnes  
% le point-virgule les lignes
```

Ce qui donne :

```
A= 1 2 3  
    4 5 6  
    7 8 9
```

```
>> t = 0 : 0.2 : 2.8 % On incrémente les composantes du vecteur t de 0 à 2.8 par pas de 0.2  
ans = 0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8
```

```
>> signal = sin (t) % Le sinus s'applique à chaque composantes du vecteur t  
ans = 0 0.19 0.38 0.56 0.71 0.84 0.93 0.98 0.99 0.97 0.90 0.80 0.67 0.51 0.33
```

Pour connaître la taille d'une matrice, il suffit de taper la commande *size*.

```
>> size (t) donnera ans = 1 15 % Une ligne sur quinze colonnes
```

II.2.2.b Extraction des valeurs d'une matrice

A (i, j) représente l'élément se trouvant sur la ième ligne, jème colonne

```
>> b = A (2, 3) donne b = 6
```

A (:, j) représente la colonne n°j

```
>> C = A (:, 2) donne
```

```
C = 2
```

```
5
```

```
8
```

A (i:k, :) représente les lignes allant de i à k

```
>> D = A (1:2, :) donne
```

```
D = 1 2 3
```

```
4 5 6
```

A (i:k, j:l) : extraction d'une sous matrice

```
>> E = A (2:3, 2:3) donne
```

```
E = 5 6
```

```
8 9
```

II.2.2.c Construction de matrices de taille croissante

Rappelons que la virgule sépare les colonnes, le point-virgule les lignes.

```
>> F = [A C] donne donc : F = 1 2 3 2
```

```
4 5 6 5
```

```
7 8 9 8
```

```
>> G = [A ; A (2, :)] donne G = 1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
4 5 6
```

Notez la différence entre () et [] : () ensemble d'arguments ; [] énumération d'éléments.

```
>> Z = [] % Matrice vide
```

On peut supprimer une colonne d'une matrice de la façon suivante :

```
>> A (:, 2) = [] donne : A = 1 3
```

```
4 6
```

```
7 9
```

II.2.3 Opération sur les matrices

Soit la matrice notée $m2$ qui vaut :

```
>> m2 = [1:1:3 ; 11:1:13] % Matrice 2 lignes et 3 colonnes
```

```
m2 = 1 2 3
```

```
    11 12 13
```

II.2.3.a Transposition

L'opérateur apostrophe utilisée pour créer un vecteur colonne est en fait l'opérateur transposition: `>> m2'`

```
ans = 1 11
```

```
      2 12
```

```
      3 13
```

II.2.3.b Opération scalaire-matrice

Une telle opération agit sur chaque élément de la matrice:

```
>> m2' * 10 % De même: 4*m, m2-10, m2/4
```

```
ans = 10 110
```

```
      20 120
```

```
      30 130
```

Une exception: `>> m2^2`

```
??? Error using ==> ^
```

```
Matrix must be square.
```

Dans ce cas, Matlab veut calculer le produit matriciel $m2 * m2$. La solution est l'usage du point qui force l'opération sur chaque élément:

```
>> m2 .^ 2
```

```
ans = 1 4 9
```

```
      121 144 169
```

II.2.4 Opérations entre matrices

II.2.4.a Multiplication

```
>> m1 % Définition de m1
```

```
m1 = 1 2 3
```

```
      4 5 6
```

```
      7 8 9
```

```
>> m2 % Rappelons la définition de m2
```

```
m2 = 1 2 3  
     11 12 13
```

```
>> m1 * m2' % Le produit matriciel n'est possible que si les dimensions sont cohérentes
```

```
ans = 14 74  
      32 182  
      50 290
```

```
>> m1 * m2
```

```
??? Error using ==> *
```

Inner matrix dimensions must agree.

La solution consiste en une multiplication élément par élément:

```
>> m2 .* m3 % (m2 et m3 ont les mêmes dimensions)
```

```
ans = 1.0000 4.0000 9.0000  
      11.0000 37.9473 130.0000
```

II.2.4.b Division

```
>> m2/m3 % Division matricielle à droite
```

```
ans = 1.0000 -0.0000  
      9.5406 -1.5960
```

```
>> m2\m3 % Division matricielle à gauche
```

```
ans = -0.5000 -0.8257 -0.4500  
      0 0 0  
      0.5000 0.9419 1.1500
```

II.2.4.c Division élément par élément

```
>> m2./m3 % Chaque élément de m2 est divisé par l'élément équivalent de m3
```

```
ans = 1.0000 1.0000 1.0000  
      11.0000 3.7947 1.3000
```

```
>> m2.\m3 % Chaque élément de m3 est divisé par l'élément équivalent m2
```

```
ans = 1.0000 1.0000 1.0000  
      0.0909 0.2635 0.7692
```

```
>> m3./m2 % Chaque élément de m3 est divisé par l'élément équivalent m2
```

```
ans = 1.0000 1.0000 1.0000  
      0.0909 0.2635 0.7692
```

II.2.5 Matrices particulières

```
>> ones(3) % Créer une matrice [3 ; 3]prenant la valeur 1
ans = 1 1 1
      1 1 1
      1 1 1
>> zeros(2,5) % Créer une matrice [2 ; 5]prenant la valeur 0
ans = 0 0 0 0 0
      0 0 0 0 0
>> rand(1,7) % Nombres aléatoires entre 0 et 1
ans = 0.9355 0.9169 0.4103 0.8936 0.0579 0.3529 0.8132
```

II.2.6 Caractéristiques des matrices

```
>> size(m3) % Dimensions de la matrice m3
ans = 2 3
>> length(m3) % Equivalent à max(size(m3)) : dimension maximum
ans = 3
```

II.3 Entrées/Sorties (E/S)

II.3.1 Ecran d'E/S

```
>> x = input('valeur de départ svp'); % Imprime la chaîne de caractère « valeur de départ
                                     % svp » à l'écran, la valeur tapée au clavier sera
                                     alors affectée à x.
```

II.3.2 Fichier d'E/S

II.3.2.a E/S propres à Matlab

Deux types d'entrée et de sortie sous Matlab :

- format binaire

```
>> save fichierl A B C % Sauvegarde A, B et C dans le fichier fichierl.mat
>> load fichierl % Charge en mémoire A, B et C sauvegardées dans le fichier
                  %fichierl.mat
```

- format texte

Une seule matrice peut-être sauvegardée par fichier:

```
>> save fichier2.dat A -ascii % Sauvegarde les valeurs de A dans le fichier fichier2.dat au
                               % format texte
```

```
>> load fichier2.dat      % Récupère les valeurs contenues dans le fichier fichier2.dat et
                          % les place dans la variable fichier2
```

II.3.2.b E/S standards

Pour lire des valeurs sauvegardées sous un format binaire dans un fichier, une liste d'instructions est nécessaire :

```
>> fidin = fopen ('fichier3.dat', 'r'); % Ouvre le fichier fichier3.dat du répertoire courant
                          % pour le lire et initialise la variable fidin
>> Donnee = fread (fidin, 2000, 'uchar'); % Lis 2000 valeurs binaires sauvegardées comme
                                          % des uchar = unsigned char, c'est à dire des
                                          % valeurs codées sur 8 bits non signés (allant de
                                          %0 à 255) et les place dans le vecteur Donnee
>> fclose (fidin); % Ferme le pointeur affecté au fichier fichier3.dat
```

Pour sauvegarder des valeurs dans un fichier sous format binaire réutilisable par un autre logiciel :

```
>> fidout = fopen ('fichier4.dat', 'w'); % Ouvre le fichier fichier4.dat du répertoire courant
                                          % en mode d'écriture binaire et initialise la variable
                                          %fidout.
>> fwrite (fidout, Donnee, 'uchar'); % Ecrit les valeurs (allant de 0 à 255) de Donnee dans
                                          % le fichier fichier4.dat
>> fclose (fidout); % Libère l'espace mémoire affecté au fichier fichier4.dat
```

II.4 Programmer avec Matlab

II.4.1 Opérateurs logiques et de relation

Les opérateurs logiques et de relation utilisés sous Matlab sont les suivants : < plus petit ; > plus grand ; <= plus petit ou égal ; >= plus grand ou égal ; == égal ; ~= pas égal ; & et ; | ou ; ~ not ; xor(x,y) ou exclusif ; any(x) retourne 1 si un des éléments de x est non nul ; all(x) retourne 1 si tous les éléments de x sont nuls ; isequal(A,B) ; isempty ; ischar etc...

II.4.2 Instruction et commandes structurées

Une instruction peut désigner une instruction simple (expression), conditionnelle, une boucle ou une rupture de séquence.

Une commande structurée permet de réaliser des itérations ou des sélections. Matlab dispose des instructions structurées suivantes : *for*, *while* et *if*. La syntaxe de chacune de ces instructions est illustrée dans les exemples qui suivent.

II.4.2.a Instruction *for*

La syntaxe est la suivante :

```
for variable = expression  
    Instructions  
end
```

Les instructions sont répétées un nombre de fois donné. Les colonnes de « expression » (matrice) sont affectées, l'une après l'autre, à la variable et les instructions sont exécutées. « Expression » est généralement un vecteur ligne de forme *deb : pas : fin*. Les boucles *for* peuvent être imbriquées entre elles et avec les autres instructions de contrôle.

II.4.2.b Instruction *while*

La syntaxe est la suivante :

```
while expression  
    Instructions  
end
```

Les instructions sont répétées aussi longtemps que “expression” est différente de zéro. L’expression peut être simple ou composée d’expressions reliées entre elles par des opérateurs relationnels ou logiques ($=$, $<$, $>$, $<=$, $>=$, $\&$, $|$, \sim , etc.). L’expression peut-être une variable. Les instructions sont exécutées si tous les éléments de cette variable ont leur partie réelle non nulle.

II.4.2.c Instruction de sélection *if*

La syntaxe est la suivante :

```
if expression1  
    (Commandes à exécuter si expression1 est « vrai »)  
elseif expression2  
    (Commandes à exécuter si expression2 est « vrai »)  
else  
    (Commandes à exécuter si aucune expression est « vrai »)  
end
```

L'expression peut être simple ou composée d'expressions reliées entre elles par des opérateurs relationnels ou logiques ($=$, $<$, $>$, \leq , \geq , $\&$, $|$, \sim , etc.). L'expression peut-être une variable. Les instructions sont exécutées si la partie réelle de cette variable à tous ses éléments non nuls.

En cas d'instructions conditionnelles imbriquées, un *else* est toujours associé au *if* le plus proche. Dans le cas de sélections multiples, on utilisera l'instruction *elseif*.

II.4.2.d Instruction de rupture de séquence

Ces instructions sont utiles pour sortir d'une boucle :

- *break*

Cette commande termine l'exécution d'une boucle. Si plusieurs boucles sont imbriquées, *break* permet de sortir de la boucle la plus proche.

- *return*

Cette instruction permet de revenir au fichier M ayant appelé le programme courant ou à la ligne de commandes Matlab.

- *error ('message')*

Affiche le message spécifié, émet un « bip » et interrompt l'exécution du programme.

II.4.2.e Autre instruction

- *eval*

eval (chaîne)

Evalue une chaîne de caractères en exécutant les instructions qui la composent.

eval (chaîne1, chaîne2)

Cette instruction permet de capturer des erreurs. La première chaîne est évaluée, s'il n'y a pas d'erreurs, le retour est réalisé au programme appelant, sinon la deuxième chaîne est évaluée.

Pour récupérer les arguments de sortie de l'expression chaîne, la forme suivante de la commande est utilisée :

$[x, y, z, \dots] = eval (chaîne)$

Cette commande permet d'incrémenter une variable du *workspace*, c'est à dire obtenir les variables nommées *x1*, *x2*, etc.

II.4.3 M-Files ou scripts

Un script (ou *M-file*) est un fichier (*message.m* par exemple) contenant des instructions Matlab. Voici un exemple de script :

```

% Ce script affiche le message que s'il fait beau
% message.m affiche un message
beau_temps = 1;
if beau_temps ~= 0
disp ('Hello, il fait beau')
end
return % (pas nécessaire à la fin d'un M-file)

```

Matlab vous offre un éditeur pour écrire et mettre au point vos M-files:

```
>> edit % Lance l'éditeur de Matlab, voir aussi la barre d'outils
```

Tout autre éditeur de texte convient aussi. Les *M-files* sont exécutés séquentiellement dans le *workspace*, c'est à dire qu'ils peuvent accéder aux variables qui s'y trouvent déjà, les modifier, en créer d'autres etc. On exécute un *M-file* en utilisant le nom du script comme commande: >> *message*

```

Hello, il fait beau

```

II.4.4 Fonctions

On peut écrire des fonctions Matlab que l'on peut ensuite appeler depuis un script.

Voici une fonction *temps* définie dans le fichier *temps.m* :

```

function y=temps (x)
% TEMPS (X) affiche un message suivant le temps qu'il fait
% et retourne le paramètre d'entrée X changé de signe
if length (x)>1 error ('X doit être un scalaire'); end
if x~=0
disp ('Hello, il fait beau')
else
disp ('Espérons que demain sera meilleur!')
end
y = -x;

```

Utilisation de cette fonction dans le prompt de Matlab :

```
>> help temps
```

```

TEMPS(X) affiche un message suivant le temps qu'il fait et retourne le paramètre d'entrée X changé
de signe

```

```
>> temps(1)
```

```

Hello, il fait beau

```

```
-1
```

Remarquez que les variables internes d'une fonction sont locales et n'entrent pas dans le *workspace*. Vous pouvez, dans une fonction, déclarer une ou plusieurs variables globales afin de pouvoir les utiliser depuis l'extérieur de la fonction. Par exemple: `>> global x y`

Matlab offre plusieurs moyens de vérifier les arguments d'entrées et de sorties d'une fonction:

- *nargin* : retourne le nombre d'arguments d'entrée
- *nargout* : retourne le nombre d'arguments de sortie
- *nargchk* : vérifie le nombre d'arguments d'entrée
- *error* : affiche un message d'erreur
- *inputname* : retourne le nom d'un argument d'entrée

II.5 Graphisme

Nous donnons ici les indications minimales. Utilisez *help* et les autres commandes d'aide pour affiner vos connaissances et vos graphiques.

II.5.1 Graphique 2D

```
>> help graph2d % introduction au graphisme 2D et tableau des fonctions disponibles
```

II.5.1.a Courbes : *plot*

```
>> x = linspace(0,2*pi, 30);
```

```
>> y = sin(x);
```

```
% un graphique simple :
```

```
>> plot(x,y) % Pour tracer sous la forme d'une courbe les composantes (x, y) l'axe des x  
           % correspond à l'indice de la composante dans le vecteur, l'axe des y la valeur  
           % de la composante
```

```
>> title('Trigonométrie') % Inscrit la chaîne de caractère « Trigonométrie » en titre au  
                          % dessus du graphe courant
```

Sous Matlab il est possible de sauvegarder la figure *.fig* ou de l'exporter, de nombreuses extensions sont prises en charge (*.tiff*, *.pcx*, *.bmp*, *.emf*, etc.)

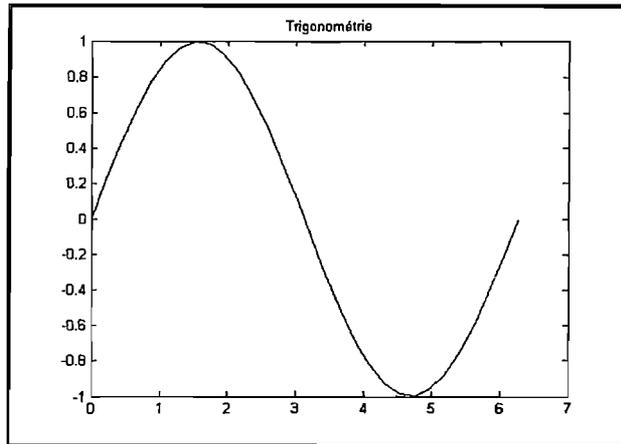


Figure 1 : Illustration de la commande plot

L'utilisateur peut apporter quelques améliorations en utilisant les commandes suivantes:

```
>> grid on
```

```
>> axis ([ 0 2*pi -1 1]) % Définit l'échelle de l'axe x
```

```
>> xlabel ('x')
```

% Quelques améliorations:

```
>> grid on % Cette commande active la grille d'arrière plan
```

```
>> hold on % Ajouter une courbe
```

```
>> z = cos (x);
```

```
>> plot (x, z, 'c+') % On ajoute ici la couleur cyan et le figuré +, sans ligne pour joindre
```

% les points

```
>> xlabel ('X') % Légende de l'axe x
```

```
>> ylabel ('Y') % Légende de l'axe y
```

```
>> legend ('y = sin (x)', 'z = cos (x)'); % Légende du graphique
```

```
>> clf % Efface la figure
```

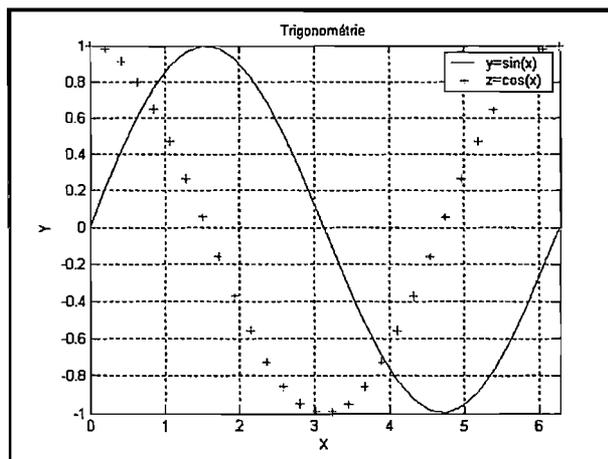


Figure 2 : Customisation du graphique (légende, titre, titres des axes)

Les commandes suivantes sont très utiles :

```
>> subplot (dl, d2, d) % Partage l'écran en une matrice de dimension dl x d2 et affiche le
                        % graphique suivant cette commande dans la région d
>> figure (1)         % Appelle la figure (1)
>> close              % Ferme la fenêtre figure (2)
>> close ('all')     % Ferme toutes les fenêtres de figures
```

Le lecteur trouvera en annexe (Annexe I) d'autres commandes pour la réalisation de graphiques élaborés, comme les commandes *semilogx*, *semilogy* et *loglog* semblables à *plot* mais qui permettent de faire des graphiques *log*, ainsi que la commande *pie* (graphique en camembert), et bien d'autre encore.

II.5.1.b Histogramme : *hist*

```
>> x=-2.9 : 0.2 : 2.9; % Défini l'intervalle et la largeurs des canaux de l'histogramme
>> y=randn (50000, 1); % Génère des nombres aléatoires répartis selon une distribution
                        % normale
>> hist (y, x) % dessine l'histogramme représenté figure 3
```

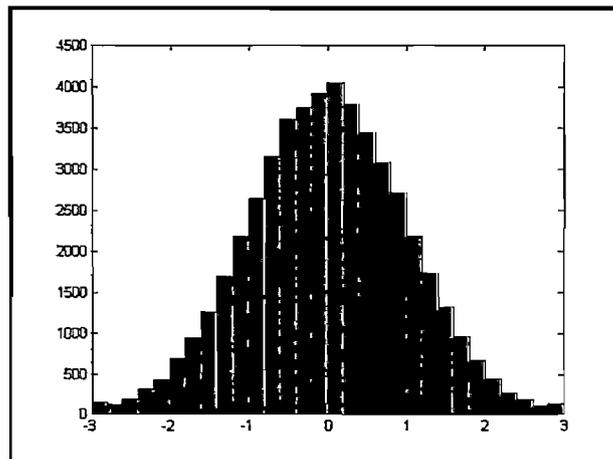


Figure 3 : Histogramme d'une distribution normale

II.5.2 Graphiques 3D

```
>> help graph3d % Introduction au graphisme 3D et tableau des fonctions disponibles
```

II.5.2.a Ligne dans l'espace : *plot3*

```
>> t = linspace (0, 10*pi);
>> plot3(sin (t), cos (t), t)
>> xlabel ('sin (t)'), ylabel('cos (t) '), zlabel ('t') >> grid on % figure 4
```

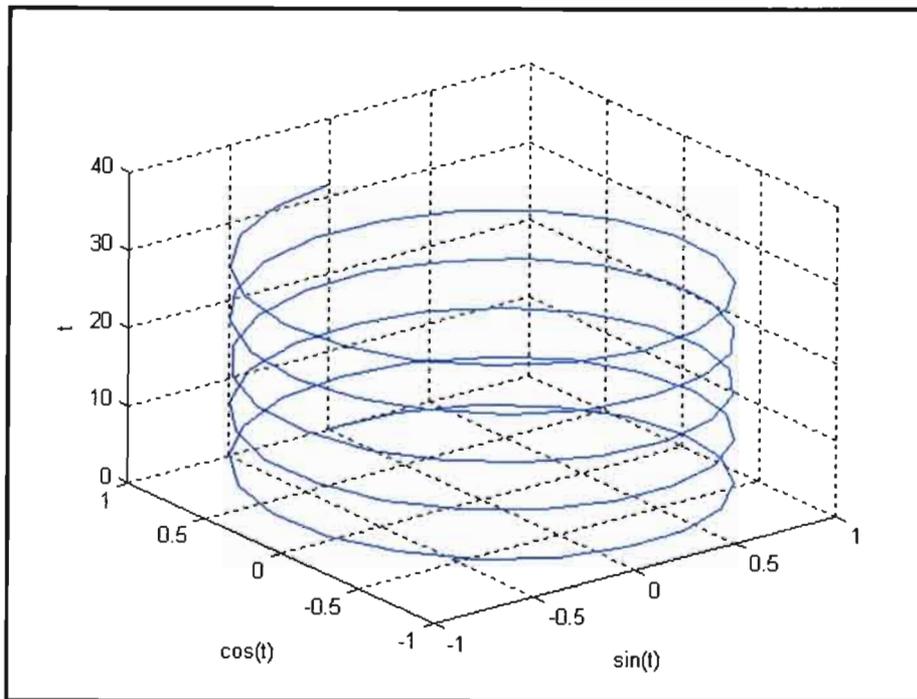


Figure 4 : Graphique 3D « ligne dans l'espace »

II.5.2.b Grillage en perspective : *mesh*

```
>> [X, Y, Z] = peaks (30); % peaks est une fonction à 2D prédéfinie dans Matlab
```

```
>> mesh (X, Y, Z) % Grillage 3D
```

```
>> meshc (X, Y, Z) % Grillage 3D avec les contours sur le plan de base :figure 5
```

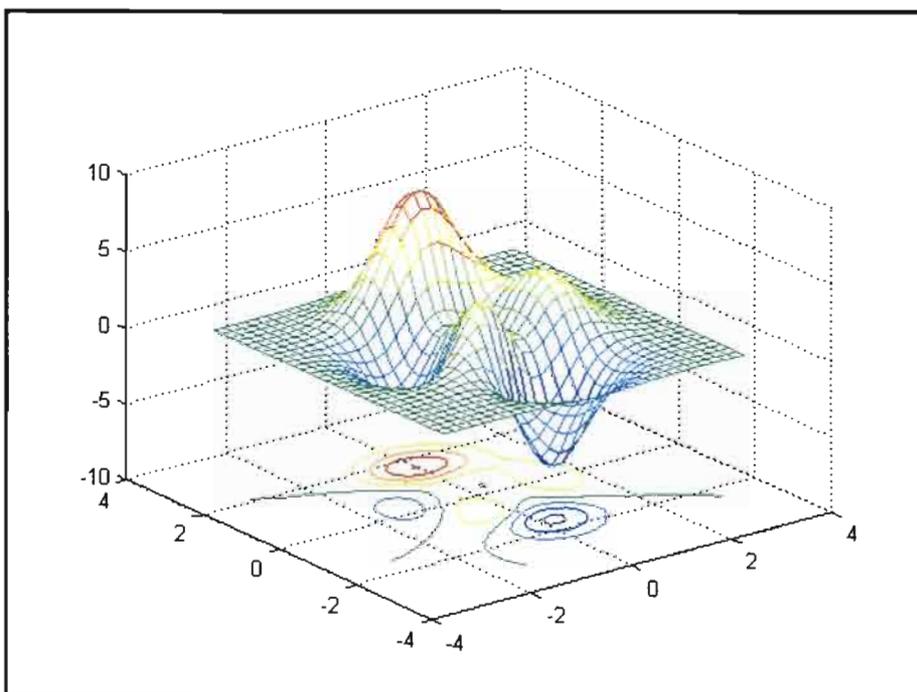


Figure 5 : Grillage 3D avec les contours sur le plan de base

II.5.2.c Surface avec illumination : *surf*

```
>> [X, Y, Z] = peaks (30);  
>> surf(X, Y, Z) % Graphique avec illumination  
>> shading interp % Meilleure interpolation  
>> colormap pink % Choix d'une palette de couleurs prédéfinie  
>> view(-37.5 + 90, 30) % Changement point de vue: view(azimut, élévation)
```

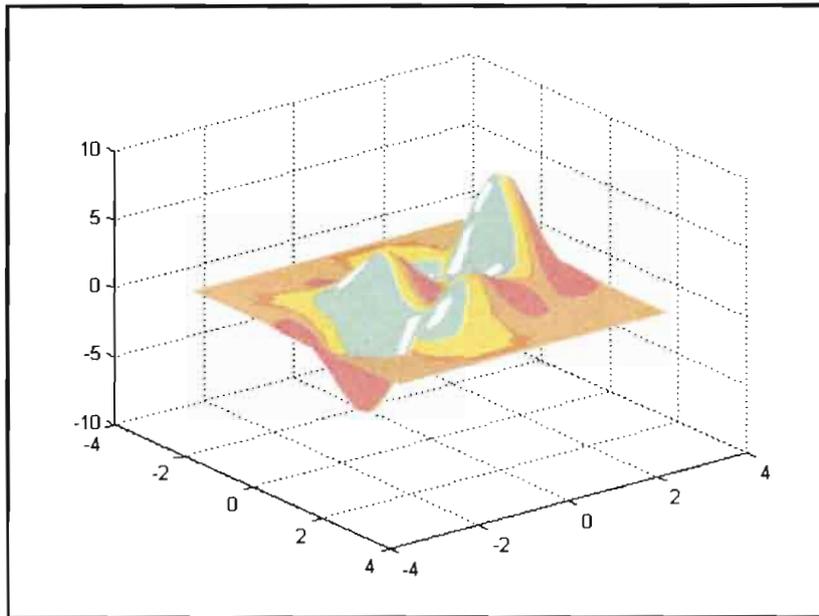


Figure 6 : Surface avec illumination : *surf*

II.5.2.d Courbe de niveau : *contour*

```
>> contour(X, Y, Z, 20) % 20 lignes de niveau
```

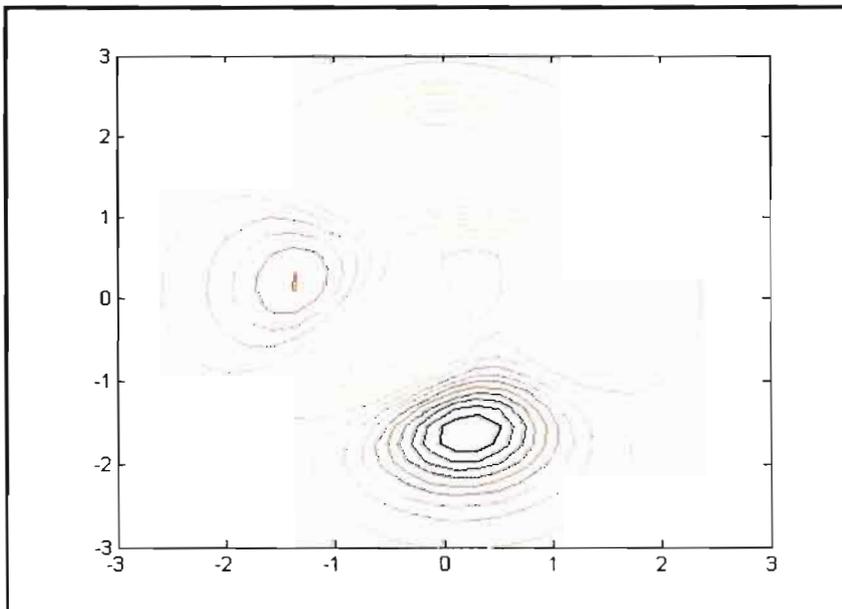


Figure 7 : Représentation en lignes de niveau

```
% Pour transformer ce graphique en échelle de couleurs
```

```
>> colormap ('gray')
```

```
>> pcolor (X, Y, Z)
```

```
>> shading interp
```

```
>> axis ('square') % figure 8
```

```
>> colormap ('default')
```

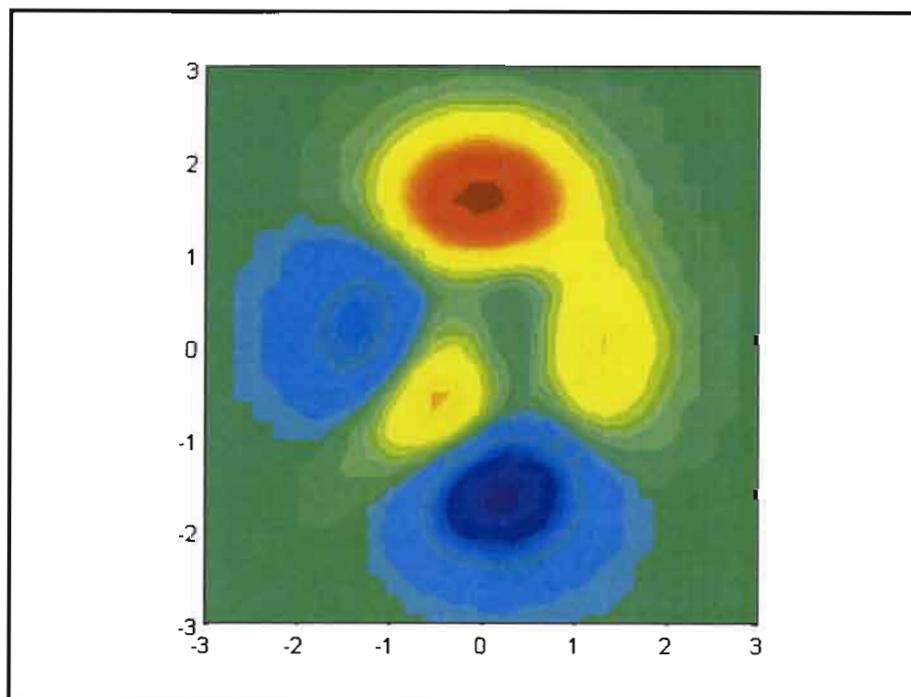


Figure 8 : Représentation en lignes de niveau avec coloration inter-lignes de niveau

II.6 « Debugger »

Pour connaître la liste et la taille des variables utilisées, il suffit de taper la commande *whos* dans l'espace de travail Matlab. Pour interrompre une liste d'instructions, tapez la commande *pause* dans le fichier exécutable; le programme ne se poursuivra que si vous tapez sur une touche quelconque du clavier. Pour avoir la main dans l'environnement Matlab au cours de l'exécution d'un programme, introduire la commande *keyboard* dans le fichier exécutable : le déroulement du programme s'interrompt et vous pouvez observer l'état des variables. Le prompt est alors : *K>>*. Afin de poursuivre l'exécution du programme, taper *return* dans la fenêtre de commande de Matlab. Pour interrompre l'exécution d'un programme, appuyer en même temps sur les touches *Control* et *C* du clavier alphanumérique.

II.7 Conclusion

Dans cette première partie nous avons tenté de donner un bref aperçu des fonctionnalités et du langage Matlab, le lecteur trouvera les compléments d'informations (liste des liens Internet et ouvrages en Annexe II) qui lui sont nécessaires, soit avec les outils d'aide en ligne de Matlab, soit dans les ouvrages suivants:

- **Introduction à Matlab**

J.-T. Lapresté (Ellipses, 1999)

- **Mastering Matlab 6**

D. Hanselman B., Littlefield (Prentice Hall, 2001)

- **Apprendre et maîtriser Matlab**

M. Mokhtari, A. Mesbah (Springer, 1997)

- **Solving problems in scientific computing using Maple and Matlab**

W. Gander, J. Hrebicek (Springer, 1995, second edition)

- **Numerical Methods Using Matlab**

G. Lindfield, J. Penny (Prentice Hall, second edition : 2000)

Matlab est relativement coûteux car il contient un nombre impressionnant de fonctions. Il existe une version étudiant à un prix abordable et un clone (**Octave**), disponible en téléchargement gratuit, dont la compatibilité avec Matlab est assez bonne (site internet : <http://www.che.wisc.edu/octave>).

III Traitement des prospections multi-électrodes 2D et 3D

Du 10 au 16 juillet 2002, une prospection multi-électrodes 2D et 3D a été réalisée, à la demande de monsieur Cédric Caron, dans le cadre de son stage de DESS de géophysique appliquée, au bénéfice de l'Association Climatologique de la Moyenne Garonne (ACMG) avec le soutien du Département de Géophysique Appliquée, UMR 7619 Sisyphe, et de l'UR Geovast, l'Institut de Recherche et de Développement (ex ORSTOM).

L'ACMG, localisée à Agen, est une association loi 1901 créée en 1959 par des arboriculteurs, avec deux principales orientations : elle constitue une station de recherche appliquée à l'agriculture et elle propose aux adhérents des services et conseils fondés sur des études climatologiques, d'irrigation, et de télédétection.

Les objectifs de cette mission étaient dans un premier temps de réaliser une reconnaissance géophysique du site proposé en mettant en œuvre les techniques d'imageries électriques 2D et 3D, puis de réaliser un suivi temporel des variations de la résistivité électrique du milieu en 2D et 3D et enfin de proposer une méthode simple, rapide, et non destructive, à la détermination des besoins en eau des cultures.

III.1 Contexte de l'étude

L'étude a été menée sur un verger de pruniers, situé dans une zone de coteaux calcaires du Lot-et-Garonne près de Laugnac, au nord d'Agen (figure 9). Le climat est de type océanique « dégradé », c'est-à-dire que l'influence océanique reste prépondérante, mais elle est altérée par l'éloignement du littoral qui lui donne une petite nuance continentale. Les précipitations sont relativement fréquentes mais en faible quantité. La climatologie estivale varie extrêmement et la région est sujette à de longues et fréquentes périodes de sécheresse. Les bassins versants de la zone d'étude présentent des ruisseaux à faible débit, régulièrement à sec en été et qui comportent des retenues de stockage d'eau. Des lacs collinaires privés, la Garonne et le Lot servent à l'irrigation. Ce verger de pruniers date de 1990 et sa superficie est de 5 hectares. La pente est faible et le sol argilo-calcaire repose sur des plateaux calcaires. Les rangs sont espacés de 7 m, les pruniers de 6 m le long d'un rang (figure 9).

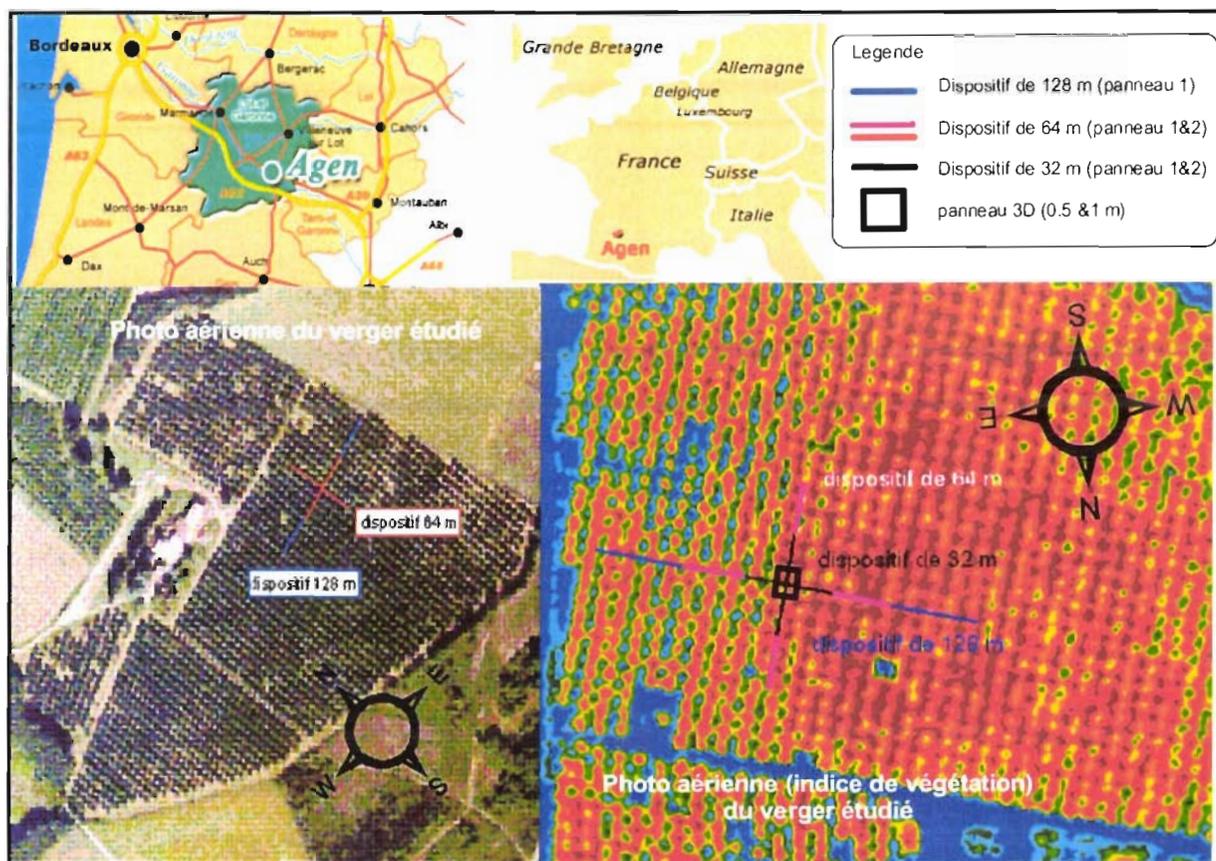


Figure 9 : Localisation du verger et implantation des panneaux électriques

Les profils électriques parallèles aux rangées de pruniers du verger ont pour orientation N15 avec deux longueurs de lignes : 64 m et 32 m, les profils perpendiculaires aux allées sont orientés N105 pour les panneaux de 128 m, 64 m et 32 m, et situés entre les arbres 10 et 11 du rang 25 à 2.25 m du 11^{ème} arbre. Le centre de tous les dispositifs est situé entre les rangs 24 et 25. La prospection 3D est centrée sur l'intersection des lignes parallèles et perpendiculaire aux rangs.

III.2 La méthode électrique

III.2.1 Principe

La méthode électrique permet la mesure de la résistivité apparente du sous-sol, en utilisant la propriété physique du sol qu'est la résistivité électrique (capacité à ne pas conduire le courant, qui dépend de la teneur en eau, de la teneur en argile, de la minéralisation de l'eau et de la granulométrie).

L'injection du courant et la mesure de la différence de potentiel sont réalisés à partir d'électrodes plantées dans le sol.

La mesure de la résistivité apparente est une mesure « intégrante » : elle est représentative d'un certain volume de terrain qui dépend de la position et de la géométrie du dispositif utilisé. Le dispositif de mesure est constitué de deux paires d'électrodes qui constituent un dipôle d'injection (électrodes A et B) et un dipôle de mesure de la différence de potentiel (électrodes M et N) produite par l'injection du courant dans le milieu environnant.

Le principe de cette méthode ainsi que l'allure des lignes de courant qui relient les électrodes, placées à la surface du sol, sont illustrés ci dessous :

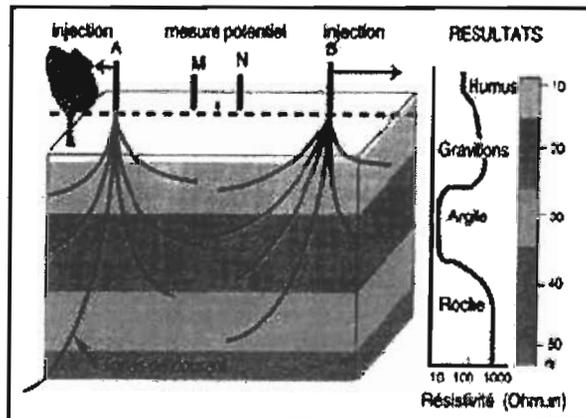


Figure 10 : Principe du sondage électrique à courant continu

Une électrode émettant un courant I au sein d'un demi-espace infini et homogène de résistivité ρ crée au point M un potentiel électrique V de la forme :

$$V_M = \rho \cdot \frac{I}{2\pi r}$$

Entre les électrodes M et N, on mesure la différence de potentiel ΔV due à l'action conjuguée de A et B. La résistivité apparente est donnée par la formule suivante :

$$\rho_a = K \cdot \frac{\Delta V}{I}$$

ρ_a s'exprime en $\Omega.m$, ΔV en mV, I en mA, et où K est le coefficient géométrique, dépendant de la disposition des électrodes :

$$K = \frac{2\pi}{\left(\frac{1}{AM} - \frac{1}{AN}\right) - \left(\frac{1}{BM} - \frac{1}{BN}\right)}$$

En augmentant la distance AB, les lignes de courant atteignent des couches de plus en plus profondes.

La méthode électrique peut-être utilisée pour réaliser un sondage électrique (connaissance de la distribution verticale des résistivités apparentes sous le point de sondage), un trainé électrique (étude des variations latérales des résistivités apparentes le long d'un profil) ou encore un panneau électrique (combinaison des deux précédent).

I.9.2 Les différents dispositifs utilisés

Pour l'établissement d'une carte de résistivité les dispositifs les plus variés peuvent être envisagés. Tout dispositif de mesure comporte quatre électrodes, deux A et B, pour l'envoi de courant « circuit d'émission » et deux M et N, pour la mesure du potentiel ΔV « circuit de mesure ».

I.9.2.a Le pôle-pôle

Les électrodes B et N sont placées à l'infini (figure 11). Ce dispositif est théoriquement simple puisque le potentiel mesuré en M est dû à A seul. Dans la pratique cependant ce dispositif est encombrant à cause de la longueur des câbles nécessaires pour rendre négligeable l'effet des électrodes placées à l'infini.

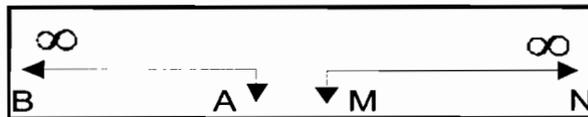


Figure 11 : Le dispositif pôle-pôle

Le coefficient géométrique (B et N étant rejetés à l'infini) se réduit à : $K = 2\pi a$.

I.9.2.b Le pôle-dipôle

On ne rejette à l'infini qu'une des quatre électrodes (figure 12), on obtient un tripôle AMN ou ABM. En vertu du principe de réciprocité, on peut théoriquement utiliser ABM ou AMN. En fait le dispositif avec MN très écarté est peu commode car l'influence des courants vagabonds et des telluriques devient prépondérante. Le tripôle le plus courant est celui où les trois électrodes sont disposées sur une même ligne droite, A étant à l'extérieur de MN. De plus généralement, ou bien ces trois électrodes sont équidistantes, ou bien la distance MN est petite par rapport à AN.

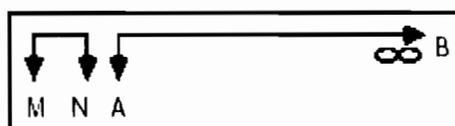


Figure 12 : Le dispositif pôle-dipôle

On peut aussi, au lieu de rejeter B assez loin pour que son influence soit négligeable, le placer de manière qu'il crée le même potentiel en M et N. En milieu homogène, il suffit de le placer sur la médiatrice de MN (figure 13). Ainsi, l'effet de B s'élimine des ΔV mesurés. En pratique, il faut éloigner B suffisamment pour que les dissymétries introduites par la non homogénéité du sous-sol puissent être négligées.

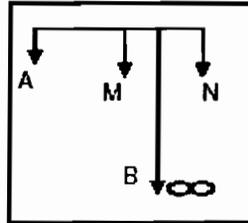


Figure 13 : Le dispositif pôle-dipôle avec l'électrode infinie perpendiculaire au dispositif

Le coefficient géométrique se réduit à (B étant à l'infini) : $K = 2\pi / (1/AM - 1/AN)$ où $AM = na$, n étant un entier, et $AN = (n+1)a$. Nous obtenons finalement : $K = 2\pi an / (n+1)$

Remarque : en ce qui concerne le dispositif pôle-dipôle, lorsque l'électrode B est à l'infini, nous obtenons une pseudo-section dissymétrique. Afin de couvrir tous les points de la même façon et d'avoir une section finale symétrique, il est indispensable de faire la même acquisition avec l'électrode A à l'infini. Il y a donc deux acquisitions successives, une utilisant les électrodes AMN, l'autre les électrodes MNB, et ce sans déplacer les électrodes.

I.9.2.c Les quadripôles

Dans la plupart des quadripôles utilisés, les quatre électrodes sont disposées sur un même alignement. Les deux électrodes de mesure M et N sont le plus souvent, à l'intérieur de l'intervalle AB et en général symétriques par rapport au milieu O de cet intervalle. Parfois il peut être intéressant d'utiliser un dispositif dissymétrique, M et N étant beaucoup plus près de l'un des pôles d'injection, l'influence du second pôle d'injection est alors très faible. Les deux quadripôles que nous avons utilisés lors de cette prospection sont :

- Le wenner α où toutes les électrodes sont équidistantes, $AM = MN = NB = AB/3$ (figure 14).
- Le wenner β (wenner dipôle-dipôle) est un cas particuliers du dipôle-dipôle, où $n = 1$. Toutes les électrodes sont équidistantes et le dipôle d'injection séparé du dipôle de réception, $AB = MN = BM = AB/3$ (figure 15).

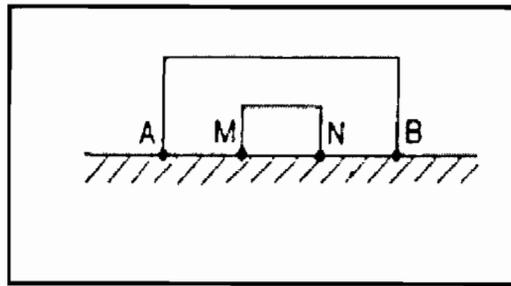


Figure 14 : Le dispositif wenner α

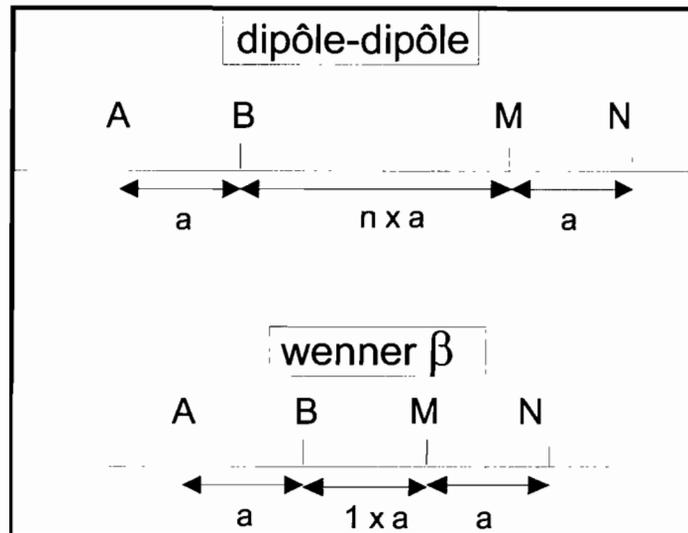


Figure 15 : Les dispositifs dipôle-dipôle et wenner β

Pour le dispositif wenner α le coefficient géométrique vaut $K = 2\pi a$ et pour le dispositif wenner β il est égal à $K = -6\pi a$.

Chacun de ces dispositifs est caractérisé par des sensibilités différentes aux hétérogénéités du sous-sol : le wenner α donne une réponse lisse, le wenner β est sensible aux interfaces verticales et à une profondeur d'investigation moindre, le pôle-dipôle allie à la fois une profondeur d'investigation importante et une bonne sensibilité.

I.9.3 Matériel utilisé

L'appareillage pour la prospection électrique 2D et 3D (figure 16) effectuée sur le site d'Agen se compose :

- D'un résistivimètre Syscal (Iris Instrument) avec un système de commutation interne permettant de piloter 48 (Switch-48) ou 72 (Switch-72) électrodes et une source d'une puissance de 100 W. Le courant de sortie est automatiquement ajusté afin d'optimiser les valeurs de tension d'entrée et d'assurer la meilleure qualité de mesure.

Le système est conçu pour réaliser de manière automatique des séquences de mesures prédéfinies. Le résistivimètre offre une excellente précision des mesures, grâce notamment à :

- une compensation automatique de la polarisation spontanée incluant une correction linéaire,
- une accumulation digitale pour l'amélioration du rapport signal/bruit.
- un calcul de la déviation standard,
- une possibilité de mesurer le bruit ambiant avant de débiter les mesures.
 - Un convertisseur DC/DC (si on utilise une batterie de voiture, pour des injections de tension entre 100 et 800W) ou AC/DC (si on utilise un groupe électrogène, pour des tensions plus fortes) fournit le courant au résistivimètre qui injecte un courant alternatif (train de créneaux) dans le sol.
 - Les capteurs multi-électrodes composés d'un multiplexeur RCM (Remote Control Multiplexeur), qui sert d'interface entre les relais électromécaniques des électrodes et les séquences programmées dans le résistivimètre, et de 4 multinodes (permettant d'utiliser les électrodes numérotées de 65 à 128).
 - De 4 flûtes permettant de connecter chacune 16 électrodes à un multinode

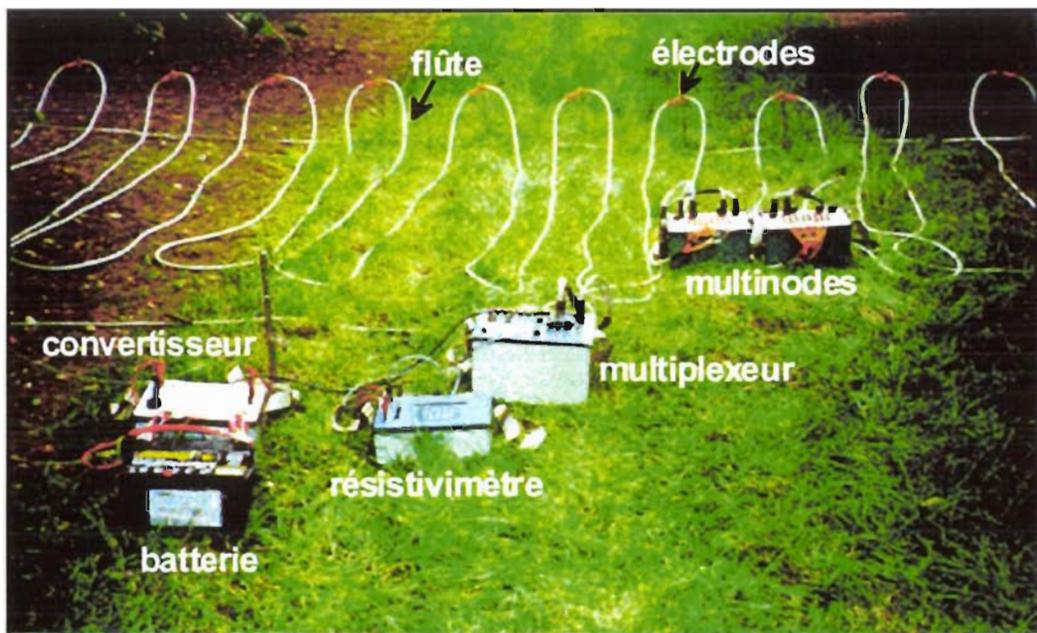


Figure 16 : Le matériel utilisé pour le panneau électrique

III.3 Le panneau électrique en 2D

Le panneau électrique est une technique d'imagerie géophysique qui est un couplage du sondage et du traîné. Ce dispositif, alliant les avantages de ces deux techniques électriques, permet d'obtenir une image en deux dimensions du sous-sol ou « pseudo-section ». L'investigation est à la fois latérale et verticale.

I.10.1 Principe

La figure 17 est une représentation schématique du principe pour le dispositif wenner α , ceci pour trois écartements d'électrodes différents.

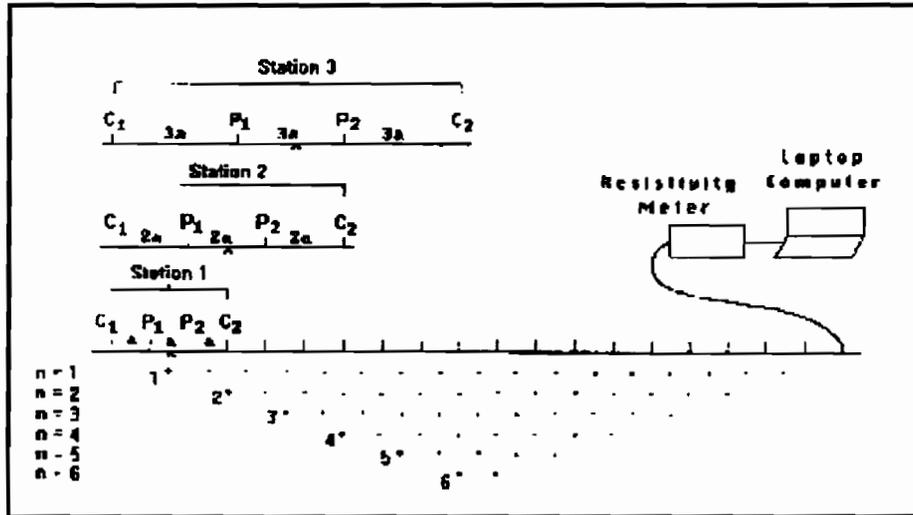


Figure 17 : Principe du panneau électrique pour un wenner α (Loke, 1996)

Les mesures sont effectuées le long d'un profil avec des distances entre électrodes augmentant ou diminuant (selon le choix de départ) régulièrement. Par convention, le point d'investigation pour un écartement inter-électrode donné est représenté sur la médiatrice de C1C2 à la pseudo profondeur n , n qui vaut pour la première nappe (écartement minimum) $1 \times a$, $2 \times a$ pour la deuxième nappe, etc. Soit l'écartement inter-électrode de a , la nappe $n = 1$ est obtenue en translatant le quadripôle le long de la ligne avec C1 la première électrode puis C1 prenant la position de la deuxième électrode (ici P1) et ainsi de suite jusqu'au dernier quadripôle possible (nombre d'électrode -3). On opère de même pour obtenir la nappe $n = 2$ avec un écartement inter-électrode de $2 a$ (station 2, nombre d'électrode -2×3) et ainsi de suite jusqu'à la dernière nappe $n = 6$. Puisque l'éloignement des électrodes mène à des informations de profondeur plus importante, la résistivité apparente mesurée permet une modélisation en 2D représentant la distribution de la résistivité vraie en fonction de la profondeur (résolution du problème inverse). Après traitement des données, l'inversion fournira une image géoélectrique du sous-sol.

I.10.2 Mode opératoire

Les opérateurs disposent au sol 64 électrodes le long d'un profil, de façon à obtenir le meilleur contact possible avec le sol. Nous disposons de 4 flûtes de 16 électrodes, ainsi que de 4 multinodes permettant les connexions entre celles-ci (figure 18).

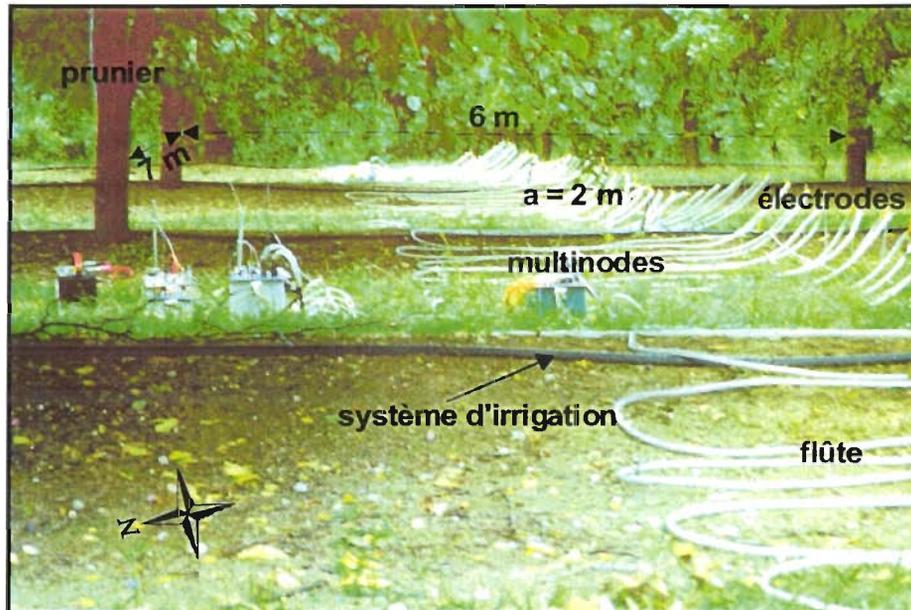


Figure 18 : Vue du panneau électrique 2D de 128 m

Les séquences d'acquisition de chaque dispositif sont programmées sur ordinateur (à l'aide du logiciel X2ipi) et transférées (Electre) dans la mémoire du résistivimètre. Ce dernier pilote l'acquisition et envoie les différentes configurations des électrodes à un multiplexeur qui permet la succession rapide des mesures sans déplacer les électrodes d'injection et de mesure. Pour la prospection 2D sur le site proposé, les configurations pôle-dipôle et wenner β ont été choisies. Avec ces dispositifs nous devons obtenir des informations sur les éventuelles anomalies de résistivité liées aux volumes racinaires des arbres (wenner β), en s'accordant une reconnaissance globale du sous-sol (pôle-dipôle).

III.3.3 Traitement des données 2D

Afin d'obtenir les résistivités interprétées du sous-sol, nous devons inverser les données de résistivités apparentes mesurées (résolution du problème inverse), de façon à ce que l'image obtenue par ce modèle calculé soit la plus proche possible de celle obtenue par les mesures.

Pour cela, nous avons utilisé le logiciel d'inversion Res2Dinv (Loke and Barker, 1996), comportant un algorithme d'inversion basé sur la méthode des moindres carrés. Les paramètres choisis pour l'inversion sont les suivants :

- limitation de la gamme de variation de la résistivité,
- les cellules du modèle ont pour largeur la moitié de l'espacement inter-électrode.

Les résultats sont représentés à l'aide du logiciel Surfer de Golden Software.

Avant la résolution du problème inverse, les données brutes sont « traitées » à l'aide du logiciel X2ipi développé par Bobachev (Université de Moscou et IRD Bondy). Ce logiciel supporte en entrée les fichiers au format binaire issu du Syscal (.seq et .data), et les données ascii issues de Res2Dinv (.dat et .inv) et propose en sortie des fichiers au format Syscal (.seq et .data), Res2Dinv, IPI (.dtg, .dat et .res), SensInv2d. Celui-ci fixe trois paramètres sur lesquels les données peuvent être filtrés, afin de limiter les problèmes dus au bruit :

- Q : facteur de dispersion (nul pour les mesures sans problème, valeur max 1%),
- I : intensité du courant injecté,
- ΔV : différence de potentiel mesurée (si les signaux obtenus sont trop faibles).

La nouvelle version de X2ipi permet également d'effectuer une série de traitement visant à éliminer les effets « P » et « C » créés par les hétérogénéités de surface (à coup de prise) sous les électrodes de potentiel (effets « P » qui apparaissent pour le dispositif pôle-dipôle) et de courant (effets « C »). La figure 19 illustre le résultat du filtrage des données obtenues avec le dispositif wenner β pour un écartement initial inter-électrode de 2 m. Nous avons choisit d'utiliser le filtrage par la méthode de la moyenne glissante, ce choix étant lié au fait que nous nous trouvons dans une situation quasi 1D, c'est-à-dire un sous-sol constitué de couches tabulaires et considérées comme infiniment allongées. La pseudo-section ainsi obtenue apparaît nettement moins bruitée, les isocontours des valeurs de résistivité apparente sont moins « perturbées ». La différence entre ρ_a (résistivité apparente non filtrée) et ρ_a traitée (résistivité apparente obtenue après filtrage par la moyenne glissante) fait clairement apparaître les anomalies liées aux effets « C », ces anomalies sont alignées sur une droite de pente sensiblement égal à 45° et sont dues à la présence dans le proche sous-sol d'hétérogénéités (Robain, 2002).

En éliminant ces artéfacts liés à la mesure de la résistivité apparente, la résolution du problème inverse est meilleure, le modèle solution montre moins d'anomalies liées aux hétérogénéités de surface.

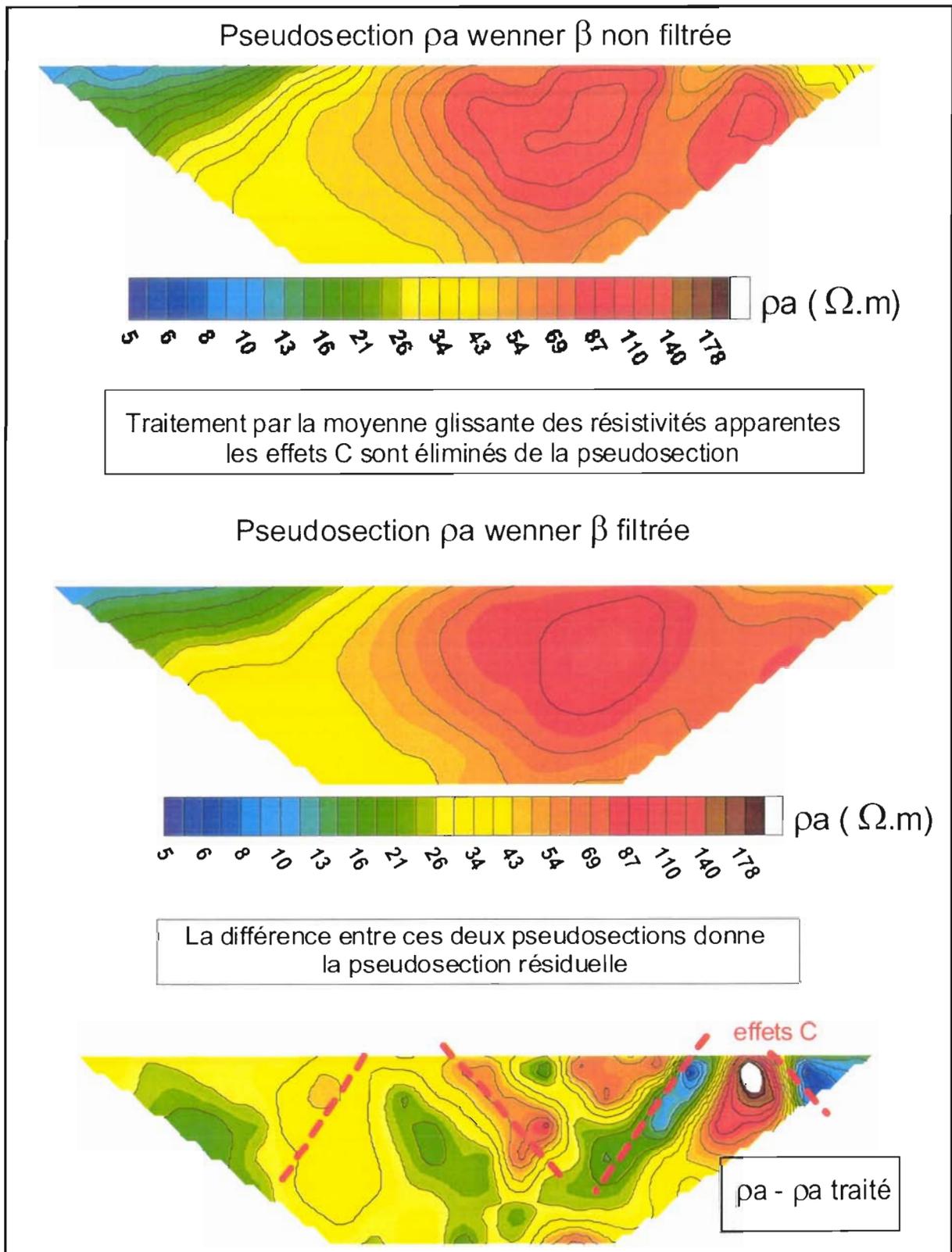


Figure 19 : Exemple de filtrage des effets « C » avec X2ipi

I.10.4 Présentation et interprétation des résultats

Les résultats des profils perpendiculaires présentés dans ce mémoire ont été retraités à l'aide de la nouvelle version de X2ipi, ils apparaissent nettement moins bruités et avec des anomalies mieux résolues que celles proposées par Caron (2002).

La figure 20 est une planche synthétique présentant les principaux résultats après filtrage et inversion des panneaux électriques perpendiculaires aux rangés de pruniers et de même centre. Les profils parallèles aux rangs ne seront pas présentés ici, les résultats (Caron, 2002) de ces derniers ont permis de vérifier l'hypothèse 1D de notre site d'étude.

L'étude des résultats du premier panneau électrique (écartement initial $a = 2$ m, dispositif wenner β) permet d'identifier deux principaux horizons de résistivités interprétées différentes qui reflètent la nature lithologique des terrains sous-jacents et d'appréhender leur répartition latérale. On observe :

i) un terrain superficiel de résistivité $< 30 \Omega.m$ (en bleu/jaune sur la figure) dont l'épaisseur varie entre 1 et 2.5 m, qui constitue la couche d'argile calcaire formée par l'altération météorique du substratum qui s'épaissit fortement à partir de la cote 85 m (présence des arbres de forte production, comme le montre la carte traitée en indice de végétation de la figure 9),

ii) un terrain de résistivité comprise entre 70 et 250 $\Omega.m$ qui constitue le substratum calcaire à la profondeur moyenne de 2.5 m. On observe des variations de résistivité importante (cote 45 m et 62.5 m, $\rho > 185 \Omega.m$, profondeur 6 m et 7 m) au sein de ce substratum qui traduisent des variations de lithologie dans le calcaire.

La seconde pseudo-section (écartement initial de $a = 1$ m, dispositif wenner β) pour laquelle la profondeur d'investigation est moins importante (de l'ordre de 7 m contre 12.5 m pour la coupe précédente), débute à la cote 32 m du précédent panneau et se termine à la cote 96 m. Nous retrouvons ici les deux principaux horizons déterminés précédemment l'épaisseur de la couche conductrice superficielle varie de 1.5 m à environs 3 m.

Les panneaux de 32 m (écartement initial $a = 0.5$ m, wenner β et pôle-dipôle) débutent à la cote 16 m jusqu'à la cote 48 m de la section de 64 m (cote 48 m à 80 m pour le panneau de 128 m). A nouveau nous retrouvons nos deux horizons :

i) une couche superficielle conductrice, de résistivité variant entre 6 et 20 $\Omega.m$, correspondant au sol argilo-calcaire composé à 40 % d'argile. L'épaisseur est nettement inférieure à la gamme de variation des épaisseurs donnée par les précédentes pseudo-sections (environ 80 cm qui est confirmée par les prélèvements effectués à la tarière (Caron, 2002)),

ii) le plateau calcaire dont la résistivité est comprise entre 70 et $> 250 \Omega.m$. Il apparaît ici pour les deux dispositifs des zones très résistantes, sphériques, et situées à l'aplomb des arbres (abscisses 5.5, 12.5, 19.5 et 26.5 m pour le pôle-dipôle et 12.5 m et 19.5 m pour le wenner β), entre 2 m et 3 m de profondeur.

L'étude de la différences entre deux sections (pour un même dispositif) réalisées à 4 jours d'intervalle (les différences étant exprimées en pourcentage et calculées avec la formule suivante : $((\rho_2 - \rho_1) / \sqrt{\rho_2 * \rho_1}) * 100$) fait apparaître :

i) une augmentation de la résistivité en surface (pourcentage positif de 10 à 66 %, jaune/rouge sur la figure), le terrain devient moins humide donc moins conducteur,

ii) une diminution très nette de la résistivité en profondeur (pourcentage négatif de 3 à -25 %, en bleu/vert sur la figure), peut s'expliquer par l'apport d'eau par infiltration ou échange chimique entre le sol et l'arbre,

iii) une différence dans la forme des anomalies liée à la géométrie des dispositifs utilisés, le wenner β montre des interfaces subverticaux, le pôle-dipôle des anomalies « plus arrondies ».

En effet, entre le 10 et le 14 juillet, le verger n'a pas été arrosé et aucune précipitation n'est survenue. Durant ces 4 jours, le sol a eu le temps de réessuyer les 35 mm de pluie tombés entre les 8 et 10 juillet. Les arbres ont consommé une partie de cette eau, une autre partie s'est évaporée. La réserve étant au 2/3 de la RFU (Réserve Facilement Utilisable), les arbres n'étaient pas en stress hydrique, et l'eau a aussi pu s'infiltrer en profondeur. L'hypothèse qui prend en compte la modification de la chimie du sol en profondeur peut expliquer cette diminution de la résistivité. En effet, les échanges entre le sol et les arbres sont importants, surtout si le milieu est saturé. Dans notre cas, la réserve en eau se trouve au 2/3 de la RFU. Les exsudats modifient les propriétés chimiques du sol et le rendent plus conducteur. L'augmentation de la résistivité de la couche superficielle est liée à la diminution de la réserve en eau entre le 10 et le 14 juillet. Elle chute effectivement de 193 à 168 mm (Caron, 2002).

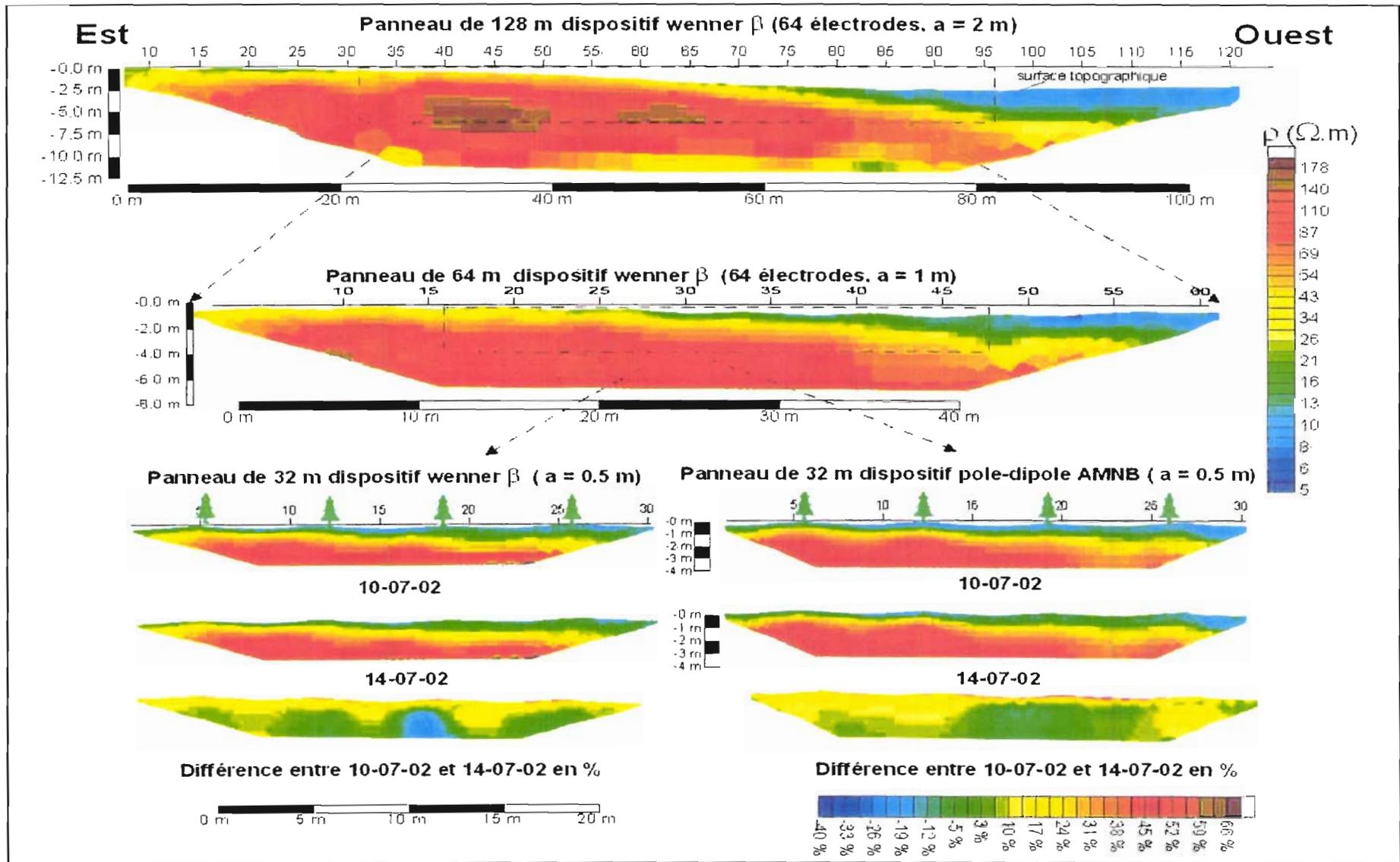


Figure 20 : Planche synthétique des principaux résultats 2D

III.4 Le panneau électrique 3D

Le principe reste identique au panneau électrique 2D, les différences résident dans les dispositifs, les configurations choisies, la mise en œuvre, le traitement, et l'information obtenue qui est un bloc de résistivité au lieu de pseudo-sections de résistivité.

III.4.1 Dispositif et mise en oeuvre

Le dispositif (pôle-pôle) schématique est illustré par la figure 21. Les 64 électrodes sont installées selon une grille carrée, avec un maillage régulier. Pour la prospection 3D, les configurations pôle-pôle et wenner β ont été choisies, avec deux écartements initiaux : $a = 1$ m, $a = 0.5$ m.

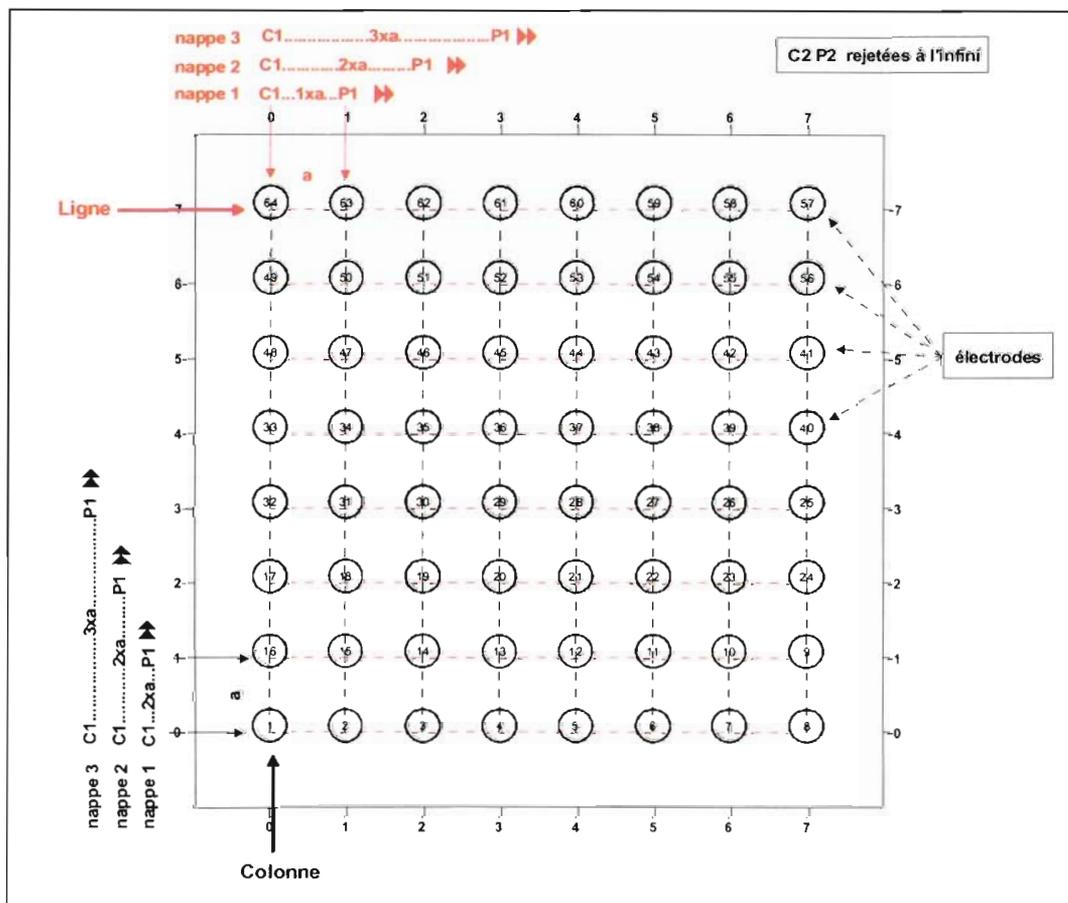


Figure 21 : Dispositif schématique 3D pour le dispositif pôle-pôle

Les séquences (succession des quadripôles) ne comporte que 3 nappes ($n = 3$, pour les deux dispositifs), elles sont saisies (Electre) pour les lignes et les colonnes c'est-à-dire que le volume est construit en jouxtant les panneaux 2D de chacune des 8 lignes et des 8 colonnes du dispositif, puis les mesures seront inversées simultanément en incluant la topographie.

Les séquences ainsi définies sont constituées de la succession de 400 quadripôles soit 400 mesures qui nécessiteront environs 30 à 45 minutes pour leur acquisition selon les configurations choisies.

On peut noter que ce temps d'acquisition est relativement long pour suivre des phénomènes transitoires qui peuvent être très rapide (de l'ordre de la dizaine de minutes). Ces dispositifs ont été choisis afin de nous permettre de réaliser un suivi temporel de la résistivité apparente en incluant dans un coin un arbre (dispositif avec $a = 0.5$ m) et une reconnaissance 3D en incluant un arbre dans chaque coin ($a = 1$ m) du dispositif. La figure 22 est une vue de l'installation d'un panneau 3D, ici la distance inter-électrode est de 0.5 m.

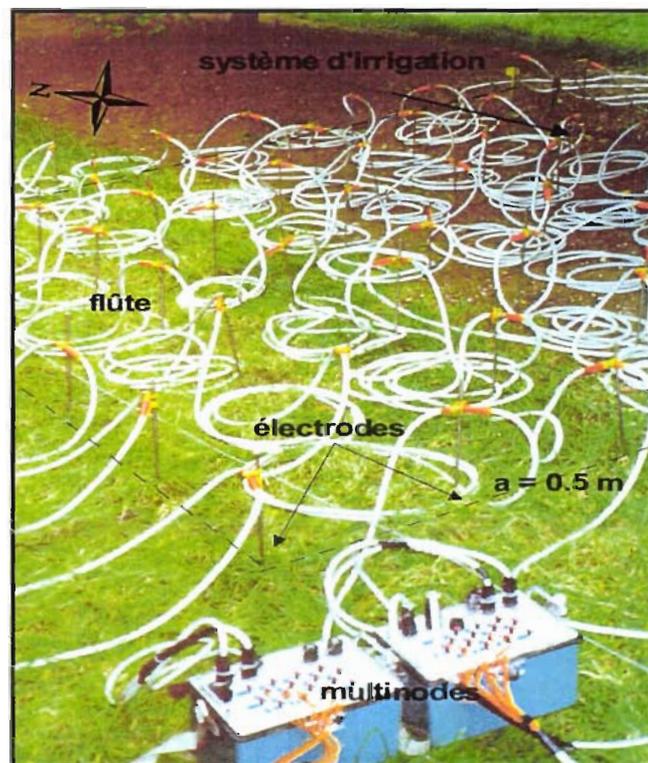


Figure 22 : Vue d'un panneau électrique 3D

III.4.2 Utilisation de Matlab pour le formatage des fichiers 3D

Les mesures acquises à l'aide du Syscal sont stockées en mémoire et sont récupérées par le biais du logiciel Electre. Elles peuvent être exportées au format *ascii* mais ne sont pas directement exploitable par Res3Dinv 2.11 (Geotomo Software, 2001) qui impose en entrée un format particulier. L'utilisation de Matlab pour effectuer ce formatage et l'automatiser s'est avéré être une aide précieuse lors du traitement des données acquises sur ce site (28 panneaux 3D soit 11200 mesures).

III.4.2.a Format de fichier Electre

Le format *ascii (.txt)*, proposé par Electre est le suivant :

Fichier mesure	: G:\GEOPHY-1\ELECTRE\PP3D1\07131845.006									
Titre/Nom de fichier	: PP3D1									
Quadripôles	: 400									
Electrodes	: 128									
Area	: 0.000000 Line : 0.000000									
Stacks de 3 à 10										
Auteur	: Nordine et Lucas-cedric									
Commentaires	: AM 3D 8x8, Nodes 65-128, 1m									

Coordonnées des électrodes										

#	X	Y	Z							
1	-64.0	0.0	0.0							
2	-63.0	0.0	0.0							
3	-62.0	0.0	0.0							
.....										
.....										
64	-1.0	0.0	0.0							
65	0.0	0.0	0.0							
.....										
.....										
126	5.0	7.0	0.0							
127	6.0	7.0	0.0							
128	7.0	7.0	0.0							

Quadripôles										

#	A	B	M	N	K					
1	65	Inf	66	Inf	6.283					
2	66	Inf	67	Inf	6.283					
3	67	Inf	68	Inf	6.283					
.....										
.....										
398	72	Inf	112	Inf	31.416					
399	80	Inf	120	Inf	31.416					
400	88	Inf	128	Inf	31.416					

Résultats										

#	V	I	Rho	q	M	M1	M2	M3	M4	
1	849.013	185.46	28.76	0						
2	1193.449	218.89	34.26	0						
3	1455.437	271.98	33.62	0						
.....										
398	427.597	225.16	59.66	0						
399	476.368	254.35	58.84	0						
400	439.530	244.23	56.54	0						

Table 1 : Format de fichier Electre

I.11.2.b Format de fichier Res3Dinv

Le format *ascii* de fichier pour Res3Dinv (*.dat*), illustré pour le dispositif pôle-pôle (écartement initial de $a = 1$ m) est le suivant :

Fichier PP3D1.dat	Commentaires
AM 3D 8x8, Nodes 65-128, 1m	Titre
8	Grille selon x
8	Grille selon y
1	Ecartement a selon x
1	Ecartement a selon y
2	Type de dispositif
400	Nombre de mesure
0.00 0.00 1.00 0.00 28.76	Coordonnées x et y
1.00 0.00 2.00 0.00 34.26	des électrodes de
2.00 0.00 3.00 0.00 33.62	courant puis de
.....	potentiel, valeur de
.....	$\rho_{apparente}$
.....	
7.00 0.00 7.00 5.00 59.66	
7.00 1.00 7.00 6.00 58.84	
7.00 2.00 7.00 7.00 56.54	Dernière mesure
Topography	Présence topographie
2	Type coordonnées x,y
-0.12 -0.16 -0.19 -0.22 -0.25 -0.27 -0.26 -0.28	Altitude de chaque
-0.33 -0.36 -0.39 -0.44 -0.45 -0.46 -0.47 -0.53	electrode arrangée
-0.47 -0.50 -0.53 -0.56 -0.57 -0.60 -0.63 -0.64	selon la grille du
-0.62 -0.66 -0.68 -0.71 -0.72 -0.76 -0.78 -0.80	dispositif
-0.70 -0.73 -0.76 -0.79 -0.80 -0.82 -0.85 -0.88	
-0.69 -0.71 -0.74 -0.77 -0.78 -0.80 -0.82 -0.84	
-0.67 -0.71 -0.75 -0.78 -0.79 -0.79 -0.82 -0.84	
-0.65 -0.70 -0.75 -0.77 -0.82 -0.82 -0.82 -0.84	
0	5 zéros pour fin de
0	fichier
0	
0	
0	
0	

Table 2 : Format de fichier Res3Dinv®

Dans le système de coordonnées local utilisé (différent sur la Figure 21, dispositif schématique), l'électrode située en haut à gauche a pour coordonnées (0.0, 0.0). Pour les types de dispositif, la nomenclature est la suivante : 2 pour le pôle-pôle, 3 pour le dipôle-dipôle (et wenner β), et 6 pour le pôle-dipôle. Le mot « topography » doit apparaître juste après la dernière valeur de résistivité apparente pour que le processus d'inversion tienne compte de la topographie. La valeur 2 indique que les coordonnées x, y sont les distances le long du sol (en effet, les décamètres, qui permettent de placer les électrodes, sont posés le long du sol) et non les véritables distances horizontales (1 dans ce cas). Les valeurs de la topographie sont données ligne par ligne à partir de l'électrode (0, 0), puis (1, 0) puis (2, 0) jusqu'à (7, 0) ; la deuxième ligne du fichier *.dat* commençant à (0, 1) et se termine à (7, 1) ; pour l'exemple donné ici.

III.4.2.c Structure du code

L'objectif était de réaliser à l'aide de Matlab un script qui à partir du fichier de données au format Electre va créer le fichier *.dat* d'entrée dans Res3Dinv correspondant. La figure 23 illustre de façon schématique la structure de ce script :

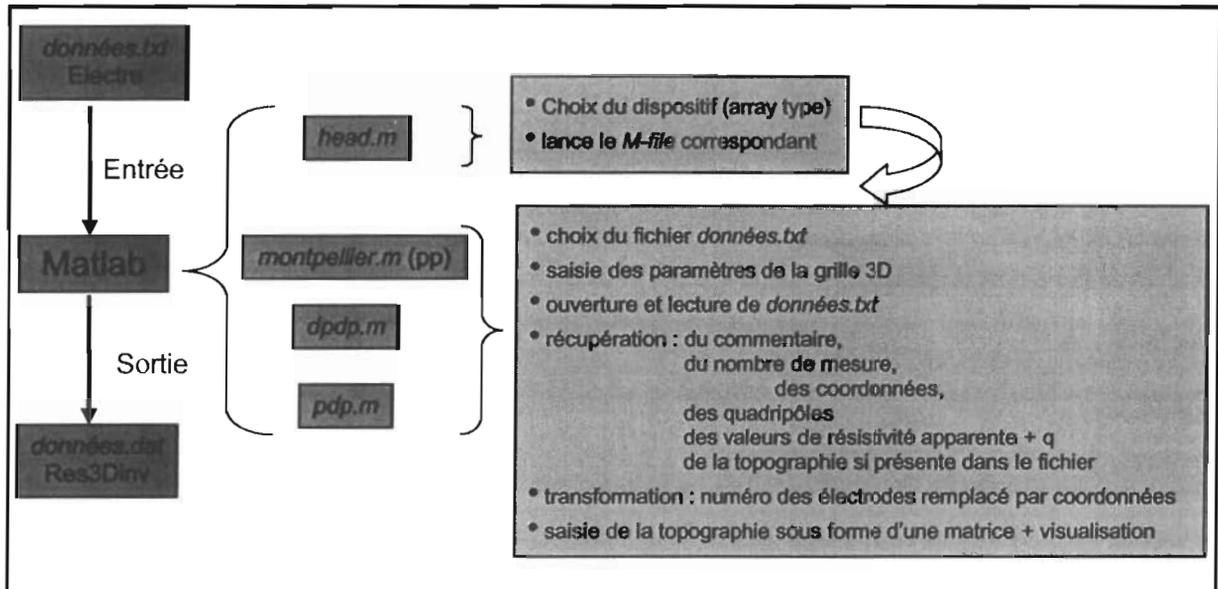


Figure 23 : Schéma illustrant les processus de formatage des données 3D sous Matlab

Pour transformer les données, 4 *M-scripts* sont nécessaires :

- *head.m* : à partir du dispositif choisit ce script (Annexe III) lance l'un 3 scripts suivants qui vont traiter le fichier d'entrée et donner en sortie le fichier *.dat* désiré,
- *montpellier.m* : ce script (Annexe III) formate les données acquises avec le dispositif pôle-pôle, il est adapté à la lecture des fichiers obtenus avec les configurations 3D utilisés à Agen (lecture et transformation des séquences lignes et colonnes), et Montpellier (lecture et transformation des séquences lignes, colonnes, et diagonales) ; il permet également d'intégrer la topographie dans Matlab sous forme d'une matrice et de visualiser cette topographie ; enfin il écrit le fichier de sortie au format Res3Dinv,
- *dpdp.m* : ce script formate les données acquises avec le dispositif dipôle-dipôle, il est adapté à la lecture des fichiers obtenus avec les configurations 3D utilisés à Agen ; il autorise la saisie de la topographie et permet la visualisation de cette dernière, et enfin écrit le fichier de sortie au format Res3Dinv,
- *pdp.m* : ce *M-file* formate les mesures obtenues avec le dispositif pôle dipôle pour les deux configurations AMN et MNB ; il réalise les mêmes opérations décrites précédemment.

L'utilisation de Matlab a permis de traiter les données de 28 panneaux 3D réalisés avec les dispositifs pôle-pôle, wenner β et pôle-dipôle et ceci en un temps record (la transformation des données pour un fichier de mesure ne prend que quelques minutes). L'utilisation de ce code a également permis le formatage des fichiers de mesure obtenus à Montpellier (les résultats ne seront pas présentés dans ce mémoire). Cependant, ce code ne trouve plus réellement d'intérêt dans la mesure où la nouvelle version de Res3Dinv 2.12 accepte en entrée le format Electre, l'avantage indéniable de ce code est la possibilité d'intégrer à n'importe quel moment la topographie de façon très simple et de la visualiser pour corriger d'éventuelles erreurs.

I.11.3 Traitement des données électriques 3D

Les valeurs de résistivité apparente sont inversées à l'aide de Res3Dinv, qui permet d'obtenir une image 3D de la structure du sous-sol, utilisant une routine d'inversion fondée sur la méthode des moindres carrés. Les paramètres choisis pour l'inversion sont les suivants :

- limitation de la gamme de variation de la résistivité
- le modèle initial est un demi-espace homogène
- les cellules du modèle ont pour largeur la moitié de l'espacement inter-électrode

Le processus d'inversion terminé, le résultat est affiché sous forme de couches qui résultent de l'intersection de plans horizontaux ou verticaux avec le volume dont la géométrie dépend du dispositif et du nombre de nappe. La nappe la plus profonde possède moins de points de mesure que la première où la densité de point est maximale. Les résultats sont représentés à l'aide du logiciel Surfer de Golden Software.

I.11.4 Présentation et interprétation des résultats

I.11.4.a Reconnaissance 3D

Nous ne présenterons ici que les résultats (sections horizontales) du panneau électrique pôle-pôle 3D (distance inter-électrode initiale de $a = 1$ m).

La figure 24 illustre la succession verticale des 6 plans horizontaux représentés sous forme d'isovaleurs de résistivité obtenues après inversion suivant un maillage géométrique et précise la localisation du dispositif du suivi temporel. L'étude des 3 premières couches (0.00 – 0.32 m, 0.32 – 0.70 m, 0.70 m – 1.5 m) permettent d'appréhender les variations spatiales des résistivités interprétées au sein de la couche d'argile calcaire et de préciser l'épaisseur de cette dernière sur la portion de terrain prospectée. On observe :

i) à proximité des arbres une résistivité $> 30 \Omega.m$ (en jaune/rouge sur la figure) et a tendance à diminuer progressivement à mesure que la profondeur augmente,

ii) la partie constituant l'inter-rang à plutôt une résistivité $< 30 \Omega.m$, qui augmente avec la profondeur, la couche d'argile calcaire a une épaisseur maximale de 0.72 m sur la portion de terrain considérée.

On constate une homogénéisation des valeurs de résistivité avec la profondeur, les halos de résistivité plus forte autour des arbres disparaissent, nous sommes ici dans un contexte 1D, les variations spatiales de la première couche sont liées aux hétérogénéités de la couche de terre arable.

L'étude des 3 dernières couches (1.5 – 2.43 m, 2.43 -3.50 m, 3.50 -4.72 m) permettent d'appréhender les variations spatiales des résistivités vraies au sein du substratum calcaire. Globalement la résistivité augmente (valeur maximale $134 \Omega.m$), le calcaire devient massif.

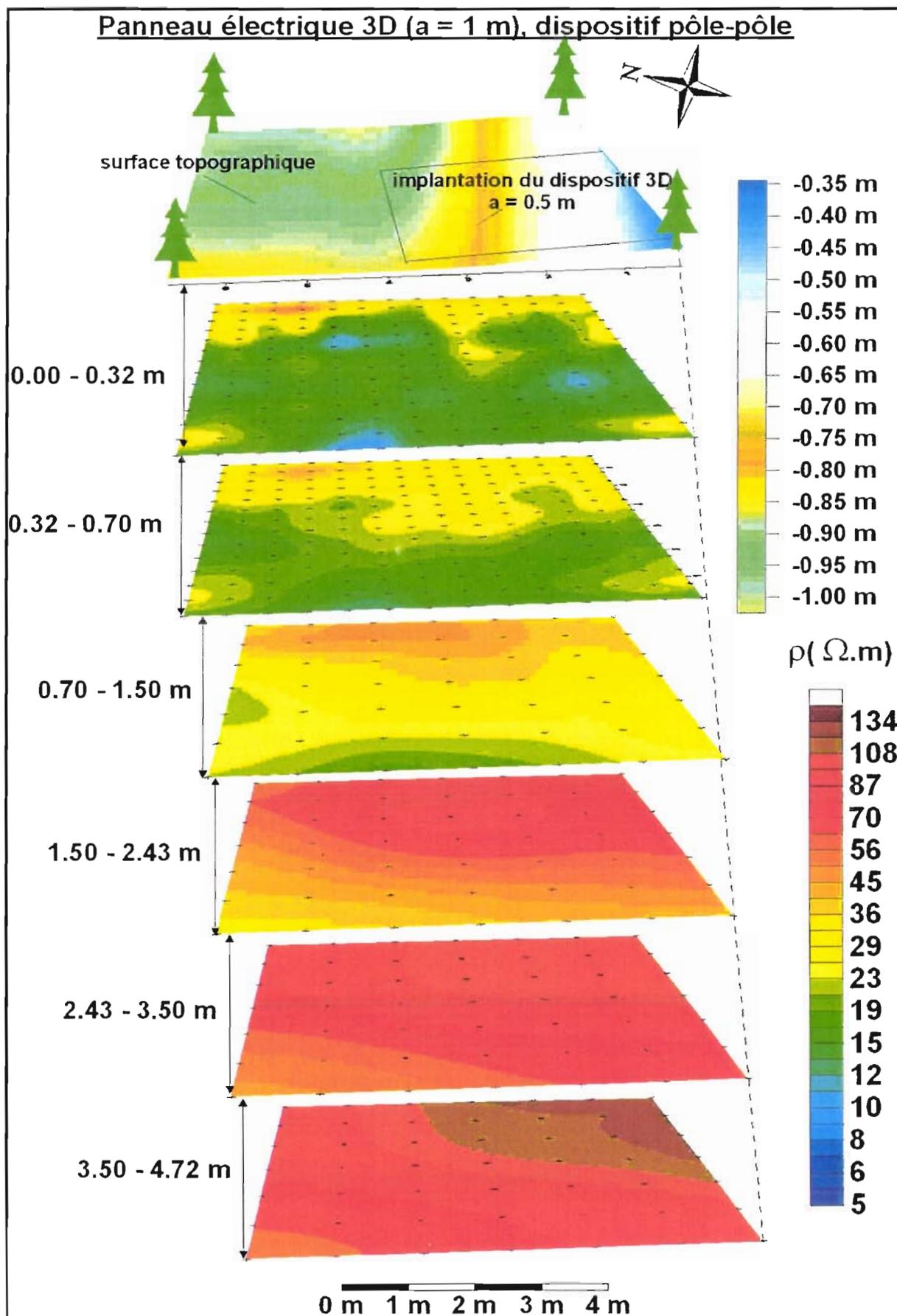


Figure 24 : Succession verticale des plans horizontaux issus de l'inversion avec Res3Dinv

I.11.4.b Essai de suivi temporel

Les résultats de la prospection 2D, pour des écartements inter-électrodes faibles ($a = 0.5$ m), nous ont inspiré un essai de suivi temporel 3D sur une journée en réalisant une acquisition avec deux dispositifs (pôle-pôle et wenner β) constitué de seulement 3 nappes (faible profondeur d'investigation et temps court pour l'acquisition). L'objectif était de tenter de suivre de possible changement de la résistivité autour d'un seul arbre. La figure 25 illustre les neuf plans horizontaux (0.00 m – 0.35 m) qui résultent de l'acquisition successive pendant 9 heures de la séquence pôle-pôle. L'étude de la couche la plus superficielle semblant la plus adéquate pour répondre à notre problématique, les résultats de la reconnaissance 3D montrant une variation spatiale plus importante pour cette couche et des résistivités plus fortes autour des arbres. En réalité, les résultats de ce suivi temporel ne montrent pas de prime abord de variations nettes entre les différentes heures. Les 3 couches, du bas de la figure, représentent respectivement, la différence entre les acquisitions réalisées à 10 h 40 et 8 h 27, 12 h 40 et 8 h 27 et enfin 18 h 27 et 8 h 40. Ces différences ont été obtenues en utilisant la formule précédemment citée (cf. page 37) pour les différences entre les pseudo-sections 2D. Le prunier est situé en bas à gauche. On observe pour ces 3 différences de façon générale :

- i) une nette augmentation de la résistivité (pourcentage positif de 13 à 75 %, jaune/rouge sur la figure), qui pourrait être liée à une diminution de la température de l'air (?),
- ii) une diminution très nette de la résistivité autour de l'arbre (pourcentage négatif de 3 à -50 %, en bleu/vert sur la figure), qui pourrait être due la photosynthèse de l'arbre (augmentation des échanges avec le sol ?).

Les raisons de ces différences ne sont ici pas clairement définies, le suivi temporel ne s'est pas effectué dans de bonne condition climatique (absence de soleil), les données ne peuvent pas être corrélées avec des données de température du sol (variation de la résistivité en fonction de la température), le temps d'acquisition d'une séquence (30 mn pour le dispositif pôle-pôle, avec 3 stacks si $Q < 1$ jusqu'à 10 stacks dans le cas contraire) ne permet pas de suivre des phénomène transitoire très rapide. Néanmoins, l'absence de résultats concrets est une information en soit, elle pointe les lacunes du protocole d'acquisition qui pour suivre de tels phénomènes était trop long (1 stack aurait diminué ce temps, mais dégradé la qualité des mesures), une telle étude nécessiterait un appareil plus rapide, et un appareillage de suivi de la température dans le sol.

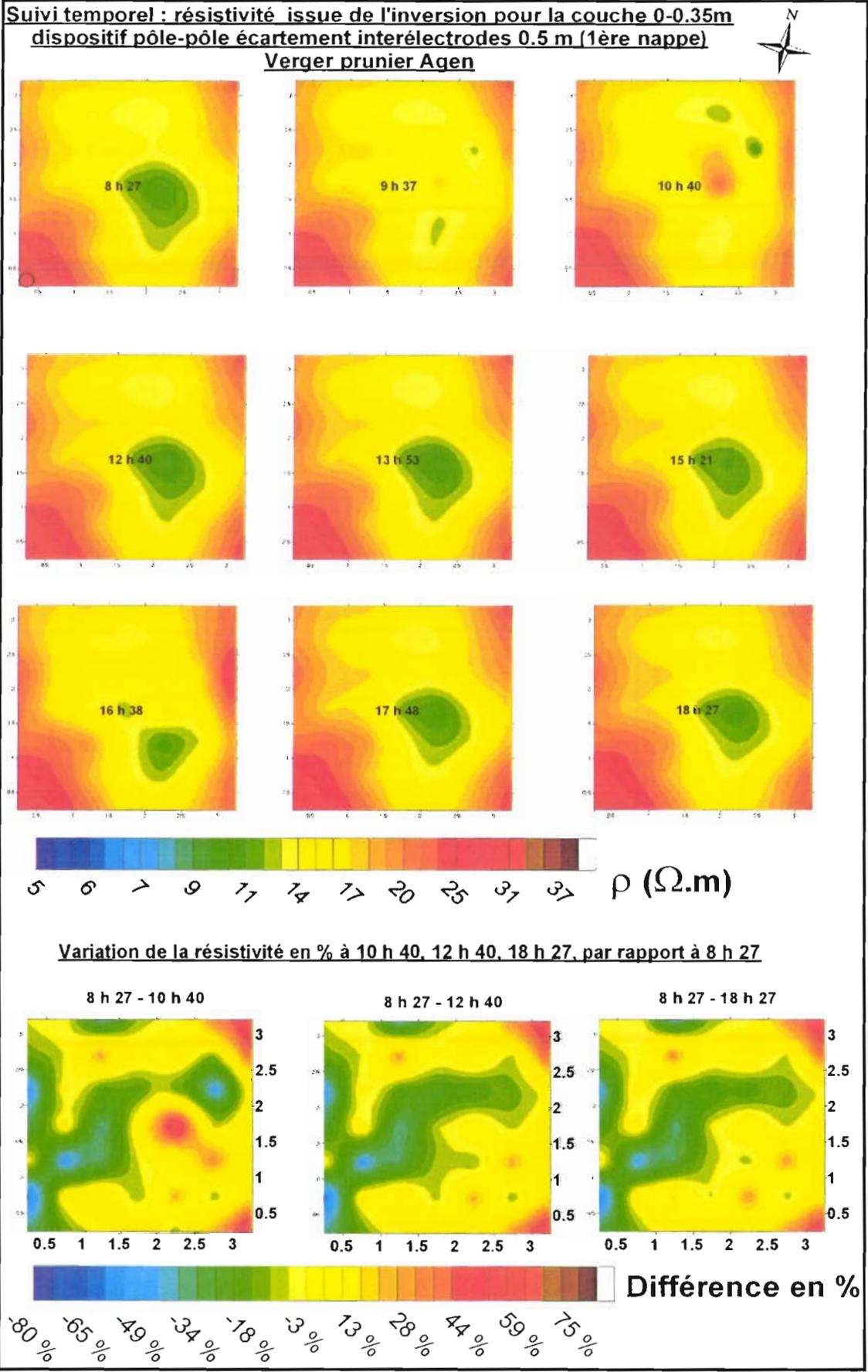


Figure 25 : Sections horizontales (0 -0.0.35 m) : le suivi temporel : pôle-pôle

III.4.5 Conclusion

La prospection multi électrodes sur ce site a permis d'identifier les deux principaux horizons constitutifs du sous-sol à savoir une couche conductrice traduisant la présence des argiles calcaires, et le substratum calcaire. Les dispositifs à faibles écartements font clairement apparaître des anomalies à l'aplomb des arbres dont la résistivité peut varier au cours du temps. Ces anomalies relativement profondes (les racines sont situées dans le premier mètre en dessous de la surface) sont probablement liées à la zone d'influence du système racinaire (zone d'échange entre les racines et le sol qui modifie les propriétés physiques du sol). Il aurait été intéressant de réaliser avec les dispositifs à faibles écartements une ligne électrique plus importante, interceptant au moins 16 arbres afin de vérifier si ces anomalies sont répétitives et sont présentes sous tous les arbres ou non. Pour le moment, ces anomalies de résistivité sont présentes sous 2 arbres sur 4. Comme nous l'avons décrit précédemment le suivi temporel peut donner de bons résultats, il faudrait diminuer le nombre de stacks et le créneau d'injection (500 ms), et lorsque les lacunes liées à l'équipement utilisé seront résolues (appareillage permettant de suivre de façon efficace les flux transitoires).

Cependant, au vue des résultats de la prospection électrique et électromagnétique à l'EM38 (Caron, 2002), il apparaît clairement que les méthodes géophysiques peuvent amener des outils supplémentaires à l'irrigation de précision : d'une part en tant qu'élément de diagnostic à l'implantation des vergers (reconnaissance géologique), d'autre part dans la détermination des zones les plus sèches (suivi temporel).

IV Formatage et la représentation des données TDEM

Ce qui va être présenté dans la suite de notre exposé correspond à la continuité du travail débuté en maîtrise sur la méthode TDEM (Time Domain ElectroMagnetism) et l'appareillage NanoTem/ZeroTem de Zonge. Le précédent travail (Bouzid, 2001) souffrait cruellement du manque de résultats fiables, l'appareil d'acquisition des données ayant un problème au niveau des cartes d'acquisition. Aujourd'hui, ce problème est résolu, mais il manque un outil rapide de formatage et de représentation des mesures, d'où l'idée d'écrire un script sous Matlab permettant d'effectuer ces tâches. Nous n'avons pas la prétention de développer un outil de visualisation type Grapher de Golden Software mais juste un outil rapide utilisable directement sur le terrain (contrôle qualité des mesures). Dans un premier temps nous rappellerons le principe de la méthode, puis les appareils utilisés en précisant les modifications de ces derniers, et enfin le *M-script* en question.

IV.1 Sondage électromagnétique en domaine temporel : méthode TDEM

Les méthodes de sondage électromagnétique visent à déterminer les distributions de la résistivité du sous-sol en fonction de la profondeur. D'une manière générale, elles comprennent les méthodes qui utilisent des champs électromagnétiques naturels comme source (méthode magnétotellurique, MT) qui sont des méthodes fréquentielles (la profondeur d'investigation est proportionnelle à la fréquence du signal) et les méthodes à source contrôlée, dont fait partie la méthode TDEM (Time Domain Electromagnetism) qui est une méthode temporelle. Pour ces dernières, c'est une induction magnétique qui permet d'explorer le sous-sol. La méthode TDEM est à nombre d'induction intermédiaire, c'est-à-dire que la distance émetteur/récepteur est petite et compense les hautes fréquences. « L'effet de peau » et la géométrie du dispositif interviennent sur la profondeur d'investigation qui est fonction du temps (t), de la période (T) et de la distance émetteur/récepteur (L).

IV.1.1 Principe

Le principe (commun à pratiquement tous les appareils) de la méthode TDEM est illustré par la figure 26. Dans la boucle d'émission nous faisons circuler un courant de type créneau avec des période de temps sans injection, caractérisé par le turn-on time (défini comme étant le temps de montée du signal à sa valeur nominale) et le turn-off time (qui est le temps que met le signal pour s'annuler).

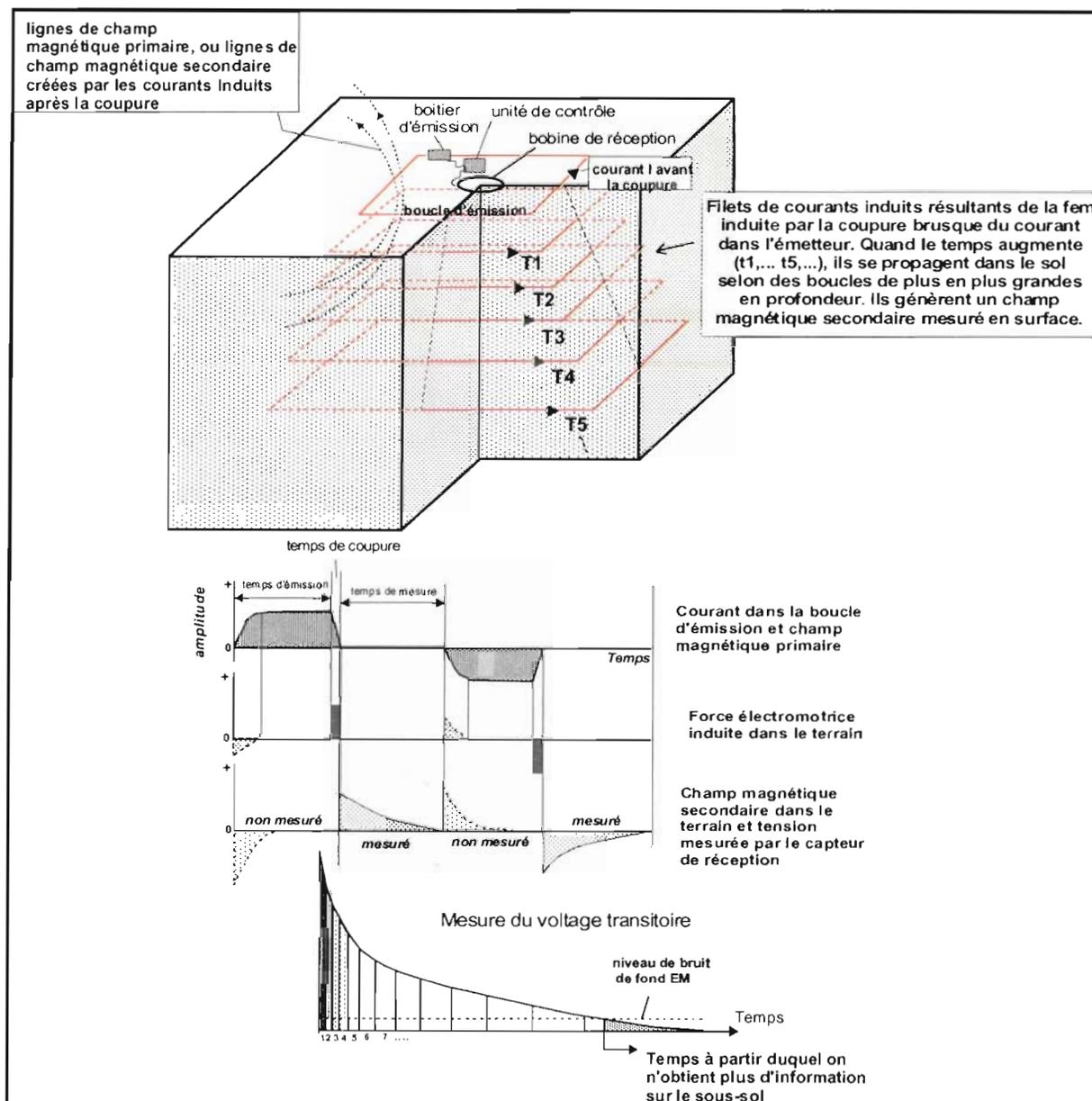


Figure 26 : Principe de la méthode TDEM (d'après Descloîtres, 1998)

Le fait de couper le courant dans la boucle d'émission produit une variation du flux magnétique selon la loi de Faraday. Une force électromotrice (fem) est ainsi créée dans le terrain. Les courants de Foucault (induits par cette fem) produisent un champ magnétique dont l'amplitude décroît avec le temps. Ces courants circulent avec une géométrie similaire à celle du dispositif qui leur a donné naissance. En surface, nous mesurons ce champ transitoire en l'absence du champ primaire. En effet, ce dernier est coupé pendant le temps de la mesure. Ce champ crée un courant dans les spires du récepteur (loi de Bio-Savart). Nous pouvons ainsi mesurer une différence de potentiel aux bornes de la bobine réceptrice. Nous en déduisons la composante verticale de la dérivée du champ secondaire en fonction du temps. Il y a proportionnalité entre cette dérivée et la tension mesurée.

Le champ induit s'exprime en teslas, $\delta B_z/\delta t$ s'exprime en tesla/seconde. Les filets de courant explorant le terrain de plus en plus profondément au fur et à mesure que le temps augmente, l'étude du champ transitoire dans le temps permet d'obtenir des informations sur une tranche de terrain de plus en plus profonde. Les mesures sont effectuées selon des fenêtres de temps. Au début du signal, où la dynamique est la plus grande, les premières fenêtres sont étroites et se succèdent rapidement pour s'élargir lorsque la dynamique diminue.

La méthode TDEM utilise des boucles de différents types qui font office d'émetteur et de récepteur. L'émetteur est généralement une grande boucle carrée disposée sur le sol. Le récepteur est soit une boucle plus petite soit la boucle émettrice elle-même. La profondeur d'investigation est en première approximation égale à la longueur d'un côté de la boucle d'émission (mais elle peut-être bien supérieure, pouvant aller jusqu'à 3 fois la longueur du côté), elle dépend de l'intensité du courant injecté dans la boucle d'émission, de la taille de la boucle émettrice et du temps de mesure (plus ce temps est long, plus nous obtiendrons des informations en profondeur).

I.12.2 Sensibilité

La méthode TDEM est très sensible aux variations terrains conducteurs. La nature contrôlée de la source EM (forme et intensité du courant) et la possibilité de faire varier la taille du dispositif (boucle d'émission) permet une investigation à diverses profondeurs des terrains sous-jacents. Pour un dispositif peu étendu en surface, la profondeur d'investigation est plus importante que dans le cas des sondages électriques, qui nécessitent une grande ligne de câble. Une fois le dispositif installé au sol, l'acquisition des données ne dure que quelques minutes, nous pouvons donc réaliser plusieurs dizaines de sondages par jour. Le dispositif de terrain ne nécessite pas de contact entre le sol et les boucles émettrices et réceptrices, ce qui autorise des prospections sur des surfaces indurées, contrairement aux méthodes électriques (électrodes plantées le long du profil d'investigation).

I.12.3 Les limites d'utilisation

Cette méthode de prospection EM présente les inconvénients suivants :

- la faible résolution des valeurs de résistivités élevées,
- la présence de perturbations EM (lignes électriques, corps métallique à proximité du point de sondage) peuvent rendre l'interprétation impossible,
- la profondeur d'exploration sera toujours limitée par le niveau de bruit ambiant,

- la non résolution de la partie superficielle en raison de l'impossibilité de faire des mesures durant le turn-off time.

IV.2 Appareil de mesure

Les appareils de mesures sont constitués par un émetteur le NT-20 qui crée le signal injecté et un récepteur le GDP-32II (Geophysical Data Processor II new, récepteur multi-canaux, utilisable en CSAMT, multi électrode ZETA, etc., si l'on possède les cartes d'acquisitions correspondantes) qui pilote le système, mesure la tension générée aux bornes de la boucle réceptrice par le champ transitoire et enregistre les données. Ces deux appareils constituent l'équipement TDEM proposé par Zonge Engineering & Research Organisation aux Etats-Unis (Tucson, Arizona, site Internet : <http://www.zonge.com>).

IV.2.1 L'émetteur NT-20

L'émetteur est constitué par un caisson en aluminium blanc avec un panneau avant muni de touches qui permettent de sélectionner la taille de la boucle, et le mode utilisé (comme nous le verrons au paragraphe suivant) ; de vérifier la tension envoyée par la batterie (externe), la température. L'émetteur peut être utilisé sous deux tensions : 12 ou 24 V, fournies par une batterie externe. L'émetteur est piloté par le GDP-32II qui contrôle l'alternance de l'injection et des mesures du champ transitoire.

IV.2.2 Le récepteur GDP32II

Le récepteur (Figure 27) est constitué par un caisson en aluminium blanc dont les spécifications techniques sont les suivantes :

- Taille : 43 cm de long x 22 cm de large x 31 cm de haut
- Poids : 13.7 kg

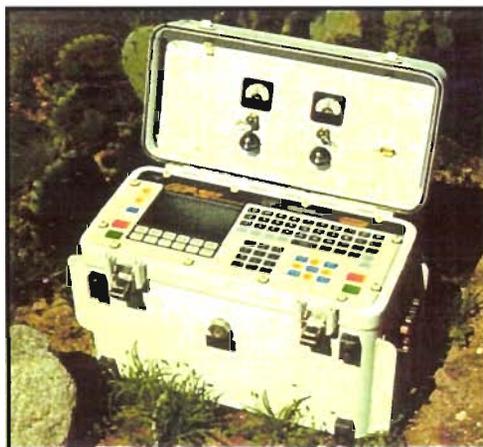


Figure 27 : Vue de la face avant du GDP32II

Le GDP-32II est muni de six canaux de mesure et fonctionne avec une batterie interne amovible de 10 A-h). Il peut être utilisé dans des conditions extrêmes (sous des températures allant de -20°C à 60°C, et sous une forte pluie). Le panneau avant est constitué d'un écran à cristaux liquides qui permet à l'opérateur de suivre les étapes de programmation de l'instrument par la succession de différents menus qui proposent des options que l'utilisateur va sélectionner selon ses besoins. Ce récepteur peut fonctionner en deux modes : le mode NanoTem pour les applications les plus superficielles et le mode ZeroTem prenant le relais pour des profondeurs d'investigation très importante (> 100 m). Le tableau suivant présente les principales caractéristiques des deux modes :

Mode	NanoTem	ZeroTem
Intensité du courant injecté	jusqu'à 3 A	jusqu'à 11 A
Canal utilisé	2	1
Alimentation du NT-20	12 V	24 V
Fréquence utilisée	64 Hz, 32 Hz	32 Hz à 0.0625

IV.3 Paramètres d'acquisition

L'acquisition de données TDEM de bonne qualité implique de configurer correctement les paramètres suivants (dont nous ne donnerons qu'une explication succincte, le lecteur pouvant se référer au manuel d'utilisation du GDP32II, section 12, Transient Electromagnetism Program)

IV.3.1 Sample Rate (pas d'échantillonnage)

Le pas d'échantillonnage détermine l'échantillonnage du signal mesuré dans le récepteur avec un intervalle de temps régulier. Cet intervalle est choisit en fonction de la profondeur d'investigation souhaité, le mode NanoTem possède deux pas d'échantillonnage : 1.2 μ s (833 k échantillons/sec) destiné à des applications très superficielles où la réponse du sous-sol est très dynamique et rapide et implique un pas d'échantillonnage très fin, et 1.6 μ s (625 k échantillons/sec) à utiliser dans les conditions d'investigation standard (meilleur d'après une communication personnelle de Kurt Sorensen, Université d'Aarhus, Danemark). Le mode ZeroTem prend le relais lorsque la dynamique du NanoTem devient trop faible pour échantillonner les temps longs. Le pas d'échantillonnage est fixé à 30.52 μ s pour les fréquences (cadences d'injections) de 32 Hz à 0.0625 Hz.

IV.3.2 Récepteur et émetteur

Nous allons préciser ici les paramètres liés à la boucle d'émission (qui constitue l'émetteur) et au récepteur (boucle, ou capteurs spécifiques : bobine multi-spires ou supraconducteurs SQUID) que l'opérateur doit changer au moment de l'acquisition.

- Array type : c'est le type de dispositif utilisé sur le terrain, boucle coïncidente (coincident loop : COL) où la boucle d'émission et de réception sont confondues, boucle centrée (inloop : INL) où l'émetteur est centré dans la boucle d'émission, dispositif avec offset (fixed loop : FXL, l'émetteur est fixe seule le récepteur est déplacé),
- Receiver moment : équivaut à la surface du capteur (x 10 pour le NanoTem, facteur multiplicatif introduit pour normaliser le signal mesuré, la carte d'acquisition possédant un gain interne de 10), par exemple $5 \times 5 \text{ m} = 25$ pour le ZeroTem et 250 pour le NanoTem,
- Transmitter X, Y : longueurs des côtés selon X et Y, par exemple 50 m pour X et 50 m pour Y dans le cas d'une boucle carré de côté 50 x 50 m,
- Transmitter Type : choisir ZT-20 en mode ZeroTem et NT-20 en mode NanoTem.

IV.3.3 Turn-off time (temps de coupure)

Le temps de coupure correspond au temps, après la coupure du courant dans la boucle d'injection, à partir duquel la mesure du champ transitoire débute. Ce paramètre est celui qui pose le plus de problème, c'est un paramètre intrinsèque à chaque appareil et qui est lié à l'électronique. Pour le Zonge, le turn-off time est la somme du Tx Delay, de l'Antenna Delay et du Filter Delay :

- Tx Delay : correspond au temps que met le courant pour s'annuler dans la boucle d'injection, la table 3 donne les valeurs approximatives des Tx Delay à utiliser pour différentes taille de boucle avec le mode NanoTem :

Taille de la boucle émettrice	Tx Delay
10 x 10 mètres	1.2 μ s
20 x 20 mètres	1.5 μ s
40 x 40 mètres	3.0 μ s
50 x 50 mètres	5.0 μ s
100 x 100 mètres	6.0 μ s

Table 3 : Taille de boucle émettrice et Tx Delay associés pour le NanoTem

Les valeurs seront affinées et précisées en utilisant un oscilloscope. L'opérateur peut également définir la valeur du Tx Delay en réalisant une mesure du champ transitoire avec une des valeurs indicatives de la table puis visualiser les données, si les premières portes sont saturées (valeurs alignées sur un seuil), il faut augmenter progressivement le Tx Delay, la valeur est correcte quand on observe une diminution progressive dans les mesures (sorte de palier faiblement penté), au contraire si les données montrent une nette décroissance, il faut diminuer le Tx Delay.

Pour le mode ZeroTem, l'opérateur peut utiliser la formule suivante pour estimer le Tx Delay :

$$\boxed{\text{Tx Delay} = (2 * l^{1.25}) / (5 + R)}$$

Le Tx Delay est en μs , l est la longueur du côté de la boucle d'émission (m), R est la résistance totale (Ω) de la boucle d'émission. Par exemple, pour une boucle de 100 x 100 m avec une résistance totale de 4 Ω , la formule donne un Tx Delay de 70 μs . Dans le cas d'une boucle d'émission rectangulaire, utilisez pour l la moyenne des côtés de la boucle (exemple pour une boucle 300 x 600 m, $l = (300 + 600) / 2 = 450$ m pour l),

- Antenna Delay : c'est le temps de retard lié au récepteur, il est de 2 μs pour le NanoTem (Zonge donne ce temps quelque soit la taille de la boucle), et de 15 μs pour le ZeroTem (pour une bobine). L'Antenna Delay est un paramètre dont la valeur doit être précisée par une série d'expérimentation sur le terrain (variation de la taille de la boucle de réception et comparaison de la réponse en fixant la valeur à 2 μs pour les deux modes et/ou en fixant la taille de la boucle et en faisant varier la valeur de l'Antenna Delay). L'influence de l'Antenna Delay doit être importante pour la résolution du problème inverse pour le NanoTem et à un degré moindre pour le ZeroTem (temps de mesure plus long),
- Filter Delay : ce paramètre est le temps de retard lié au filtre anti-repliement, il vaut 1 μs pour le NanoTem et 26 μs pour le ZeroTem, ces valeurs ne peuvent pas être modifiées par l'opérateur.

IV.3.4 Robust Filter

Robust est un filtre utilisé lors de l'échantillonnage du signal pour chaque fenêtre de temps, qui possède différentes options qui ne sont disponibles qu'en mode ZeroTem (l'option None est la seule utilisable en NanoTem) :

- None : moyenne linéaire standard, à utiliser en présence de ligne à haute tension
- Median : utilisation de la médiane
- Trimmed mean : calcul de la moyenne après élimination des valeurs aberrantes (+/- 12.5 %, 25 % ou 37 %)

IV.3.5 Gain Mode

L'opérateur peut choisir la valeur du gain : 1, 2, 4, 8, 16 ; les mesures de bruit peuvent être effectuées avec le gain maximum (16). Des tests ont été menés afin d'évaluer l'influence du gain (site test Danemark), les conclusions sont provisoires. On observe que le gain augmente avec le signal mesuré pour les temps longs.

IV.3.6 Fréquence (repetition rate) et cycle (séquence d'acquisition)

La période d'injection est choisie en fonction des profondeurs d'investigation souhaitées. Prenons par exemple un champ transitoire de durée 10 ms, c'est-à-dire d'une période totale de 40 ms, ce qui correspond à une fréquence d'injection de 25 Hz. En réalité, le terme de fréquence est peu approprié pour ce type de signal temporel, aussi les termes de taux de répétition (repetition rate) ou cadence d'injection (fréquences : 64 Hz, 32 Hz, 16 Hz, 8 Hz, etc.) sont employés tout en conservant l'unité, le hertz. Puisque la dynamique du signal est importante, certains constructeurs ont choisi de fractionner l'acquisition en plusieurs cadences d'injection, chacune comportant 20 à 30 fenêtres (3 cadences de 20 fenêtres pour le PROTEM47 de Geonics et jusqu'à 6 cadences de 32 fenêtres pour le GDP32II de Zonge). De cette manière, le signal sera échantillonné convenablement, les séries de mesures issues de différentes cadences d'injections se recouvrant partiellement dans le temps. La durée d'une séquence d'acquisition (nombre de cycle) est choisie par l'opérateur. Cette séquence d'acquisition correspond à une période unitaire (injection, mesure, injection, mesure = 1 cycle) qui sera répétée 128, 256, 512, 1024, 2048 ou 4096 fois. La mesure unitaire réalisée sur un transitoire est répétée des centaines de fois et moyennée, ce qui permet d'améliorer considérablement le rapport signal sur bruit.

A titre d'exemple, 128 cycles d'acquisition avec une cadence de 32 Hz produiront 512 courbes (128 x 2 (2 mesures par cycle) x 2, x 2 si on a choisit 2 stacks) qui seront moyennées pour obtenir une seule courbe de $\partial B/\partial t$ en fonction du temps.

IV.3.7 Détermination de la position de la 1^{ère} fenêtre temporelle

La mesure d'un transitoire débute après la fin du turn-off time, l'appareil calcule la position du centre de la première fenêtre échantillonnée de façon automatique. La figure 28 illustre à partir d'un exemple concret la détermination de la position de la 1^{ère} fenêtre échantillonnée après le début de la coupure du courant dans la boucle d'émission (principe identique pour les deux modes).

Considérons une demi période unitaire avec une injection puis une mesure en mode NanoTem, un pas d'échantillonnage (Sample Rate) de 1.6 μs et un Tx Delay de 3.5 μs . La mesure du transitoire va débiter 6.5 μs après le début de la coupure (somme du Tx Delay et de Delay (Antenna Delay + Filter Delay) soit 3.5 μs + 2 μs + 1 μs = 6.5 μs).

En se référant à la table 4 qui donne la position du centre, le début et la fin de chaque fenêtre pour un turn-off time de zéro, il apparaît que la valeur de la 5^{ème} porte (6.4 μs) est inférieure à 6.5 μs , la 6^{ème} porte (8.0 μs) pourra être échantillonnée correctement (fenêtre entière).

La position du centre de la 1^{ère} porte est calculée de la façon suivante : 5 x 1.6 = 8.0 μs (n° de la porte inférieur au temps de coupure x pas d'échantillonnage). La différence entre la position du centre et le temps de coupure donne le temps, à partir de la fin de la coupure, où débute la mesure du champ transitoire. La première porte sera donc à 1.513 μs , la 2^{ème} à 1.531 μs + 1.606 μs = 3.137 μs c'est à dire en ajoutant cette différence au temps du centre des fenêtres donnée par la table 4.

La 1^{ère} fenêtre temporelle est donc à 8.031 μs après le début de la coupure du courant. Ces temps correspondent aux temps qui sont calculés automatiquement par le GDP32II (calcul selon le principe décrit ci-dessus et illustré dans la figure 28) et inscrit dans le fichier de résultat. Pour le mode ZeroTem le principe est le même, seul le temps de coupure et le pas d'échantillonnage change, pour obtenir le centre des fenêtres reporter vous au manuel du GDP32II (section 12) ou à l'Annexe IV.

12.23 NanoTEM WINDOW CENTERS

12.23.1 Sample Period: 1.6 Microseconds Zero Delay After Current Turnoff

WINDOW #S	POINTS PER WINDOW	WINDOW CENTERS	WINDOW WIDTHS	
			BEGINNING	END
1	1	0.0 μ sec	0.0 μ sec	0.0 μ sec
2	1	1.6	1.6	1.6
3	1	3.2	3.2	3.2
4	1	4.8	4.8	4.8
5	1	6.4	6.4	6.4
6	1	8.0	8.0	8.0
7	2	10.4	9.6	11.2
8	2	13.6	12.8	14.5
9	2	16.8	16.1	17.7
10	3	20.8	19.3	22.5
11	3	25.6	24.1	27.3
12	5	32.0	28.9	35.3
13	6	40.8	36.9	45.0
14	7	51.2	46.6	56.2
15	9	63.9	57.8	70.7
16	11	79.9	72.3	88.3
17	15	100.6	89.9	112.4
18	18	127.7	114.0	142.9
19	23	161.2	144.5	179.9
20	29	202.7	181.5	226.4
21	36	254.6	228.1	284.3
22	47	320.7	285.9	359.7
23	58	404.5	361.4	452.9
24	72	508.3	454.5	568.5
25	92	639.0	570.1	716.3
26	116	804.9	717.9	902.6
27	145	1.013 m sec	904.2	1.135 m sec
28	184	1.276	1.137 m sec	1.431
29	231	1.607	1.433	1.802
30	289	2.022	1.804	2.266
31	369	2.546	2.268	2.859

Table 4 : Centre des fenêtres temporelles du NanoTem

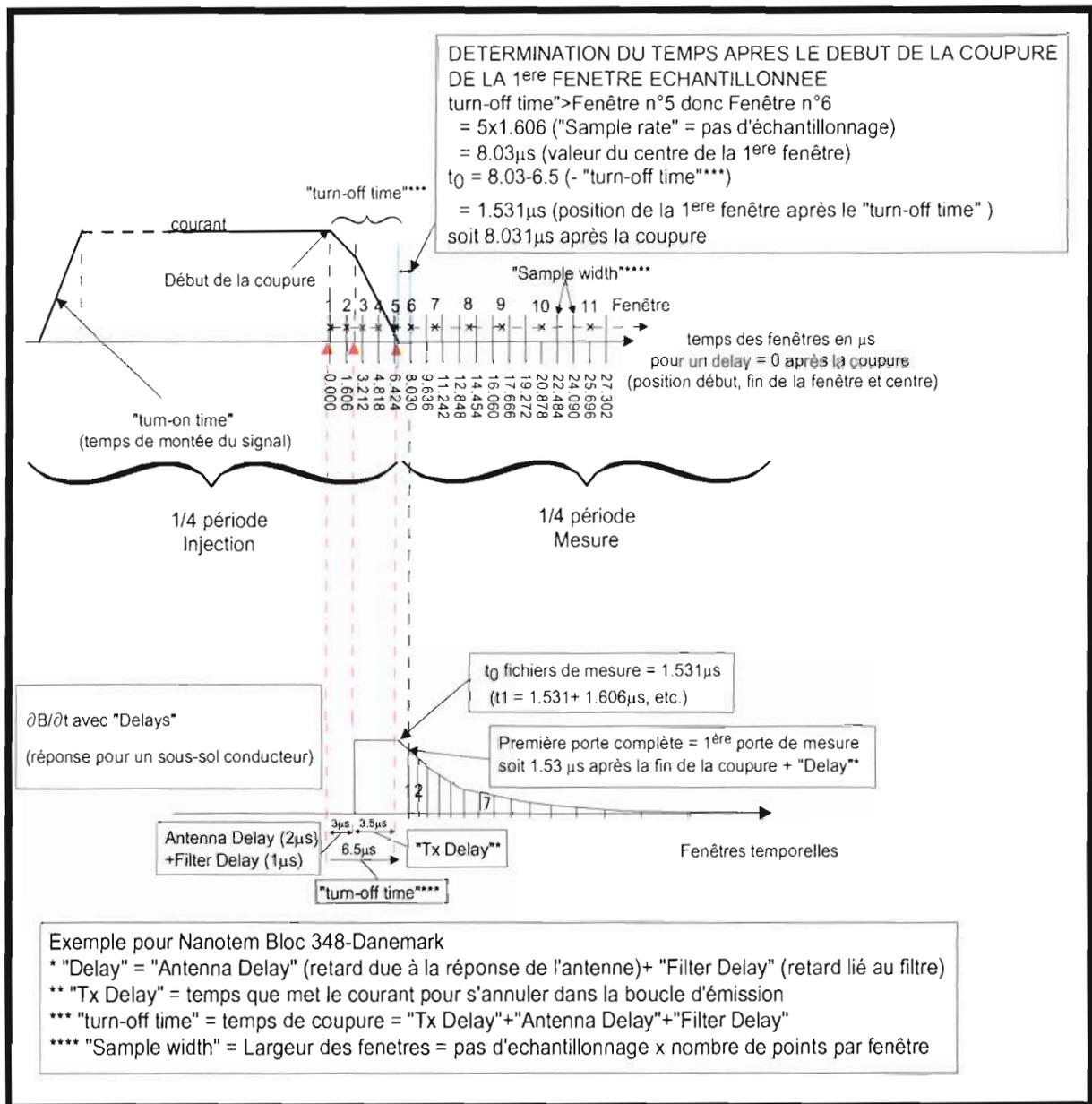


Figure 28 : Détermination de la position de la 1^{ère} fenêtre temporelle

IV.4 Format des fichiers de données issus du GDP32II

Les mesures sont stockées dans la mémoire du GDP32II, puis transférées à un PC via un câble série RS232 et en utilisant la connexion HyperTerminal disponible sous Windows. Le fichier de données comporte deux parties différentes (appelées aussi blocs) :

i) un bloc d'entête qui comprend un numéro, le mode utilisé (NANO pour le mode NanoTem ou TEM pour le mode ZeroTem), la date, l'heure, la tension de la batterie qui alimente le NT-20/ZT-20, le type de dispositif (boucles centrées, coïncidentes, offset, etc.).

Les deux lignes suivantes comportent des informations sur l'opérateur, le site, la localisation du sondage. La ligne en dessous contient le moment du récepteur (RxM, surface en $m^2 \times 10$ pour le NanoTem car la carte d'acquisition NanoTem a un facteur multiplicatif 10, en m^2 pour le mode ZeroTem) ; les longueurs x et y de la boucle réceptrice, le nombre de tour de la boucle d'émission (#T). La ligne suivante contient les valeurs des temps qui définissent le temps de coupure définis au moment de l'acquisition par l'opérateur :

- Tx Delay : c'est le temps que met le signal pour s'annuler dans la boucle d'émission
- Antenna Delay : temps inhérent à la boucle de réception et aux filtres, valeur toujours égal à $2 \mu s$
- Alias in : le filtre anti-repliement est actif (out : inactif)

Robust est un filtre utilisé lors de l'échantillonnage du signal sur chaque fenêtre de temps, qui possède différentes options qui ne sont disponibles qu'en mode ZeroTem (None, Median et Trimmed mean). Ensuite, les informations sur les cartes d'acquisition apparaissent : numéro du canal (1 pour la carte ZeroTem, 2 pour la carte NanoTem), DiffAmp signifie que la carte d'entrée connectée est un amplificateur différentiel, Notch+60 implique que la carte possède un filtre qui rejette les fréquences liées aux ligne à haute tension (60/180/300/540 Hz), le symbole « + » indique un changement dans les niveaux de rejet. La dernière ligne contient les numéros de série du panneau avant du GDP32II et du module de calibration, la température et le taux d'humidité dans le boîtier.

ii) un bloc de mesure avec un numéro de bloc qui s'est incrémenté de 1, la 2^{ème} ligne contient le mode utilisé, la date, l'heure, la tension de batterie qui alimente l'émetteur, le type de dispositif (boucles centrées, coïncidentes, offset, etc.). La 3^{ème} ligne montre les positions de la boucle d'émission (Tx) et de réception (Rx), à préciser si l'on utilise un dispositif avec offset. La 4^{ème} ligne contient la fréquence utilisée (64/32 Hz pour le NanoTem, et 32/16/8/4/2 Hz pour le ZeroTem) ; le nombre de cycle (cadence de répétition), l'intensité du courant injecté (en A), le pas d'échantillonnage (Sampling Delay), la valeur de l'anti-aliasing. Puis nous trouvons sous forme de colonne, les valeurs des centres des fenêtres temporelles (Wn), les mesures du champ transitoire 5 (Mag 2) en V/A (normalisé par le gain si l'opérateur a sélectionné un gain différent de 1), les valeurs de résistivités apparentes ($\Omega.m$) calculées avec la formule des temps longs. Le format *ascii* (.txt) des fichiers de données NanoTem et ZeroTem avec un descriptif des différentes lignes est illustré par la figure 29 :

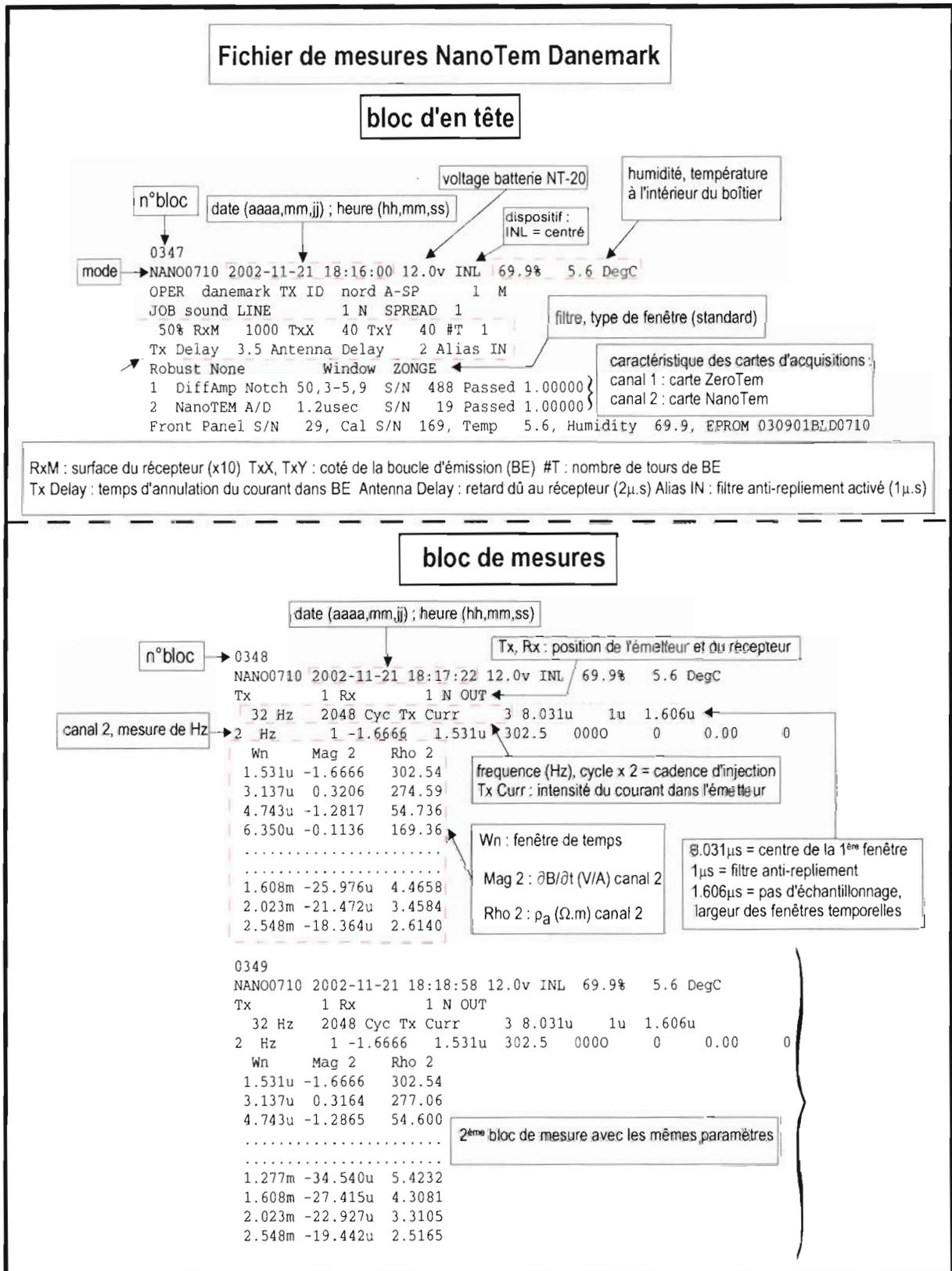


Figure 29 : Descriptif des formats de fichier NanoTem et Zerotem

IV.5 Structure du code Matlab

L'utilisation de Matlab nous a permis de formater et représenter les mesures du champ transitoire obtenu avec l'équipement TDEM proposé par Zonge. La figure 30 illustre de façon schématique la structure et le fonctionnement de ce M-file, les boîtes en pointillé et en gris clair montre les perspectives de développement du code.

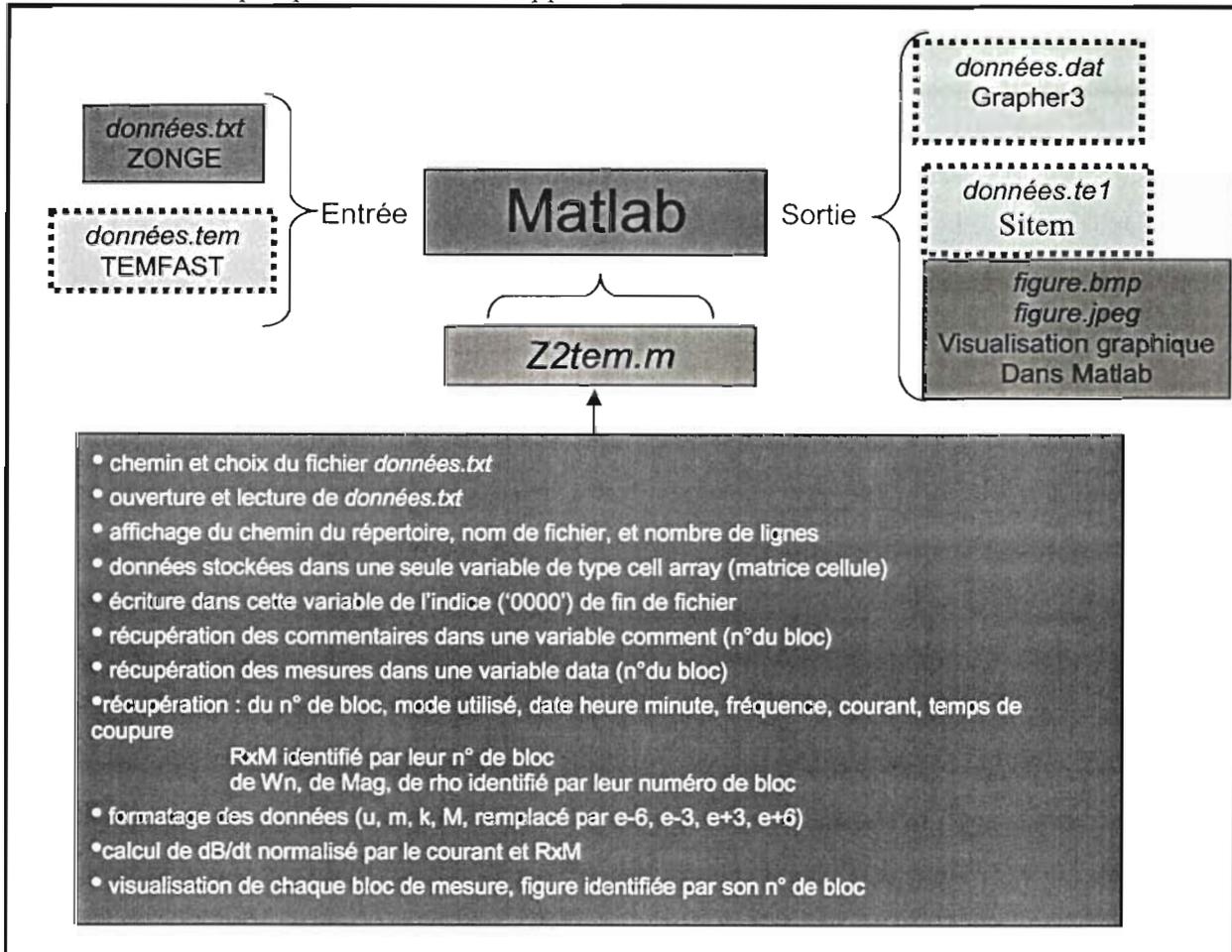


Figure 30 : Structure du code de visualisation et formatage des données TDEM

L'utilisation de ce code permet le formatage des fichiers de mesure TDEM, il constitue un outil de visualisation rapide sur le terrain des données pour un contrôle qualité. Pour représenter 20 courbes le temps nécessaire est de l'ordre de 2 à 3 minutes. Dans la mesure où la méthode TDEM est relativement rapide, il est courant de réaliser une centaine de sondage par jour, la rapidité de traitement de Matlab et de ce code trouve tout son intérêt. Le code complet est présenté en Annexe V.

I.17 Exemples d'applications

Les points suivant illustreront la fonctionnalité de ce code et donnera en exemple quelques courbes de la mesure du champ transitoire représentées dans Matlab et exportées au format *.bmp* et *.emf*.

I.17.1 Représentation des données acquises sur le site de Kerbernez (Bretagne, France)

Lors du camp de terrain des étudiants de DESS de géophysique appliquée de Paris 6, 600 mesures TDEM ont été réalisées sur le site de Kerbernez (Bretagne, France) du 24 au 30 novembre 2002. Nous montrerons ici un exemple de courbes (mesure en mode ZeroTem fréquence 32 Hz, sondage TDEM n°15) obtenues en utilisant notre code. Les résultats ne seront pas interprétés dans notre propos. La figure 31 est exportée à partir de Matlab au format *.bmp*.

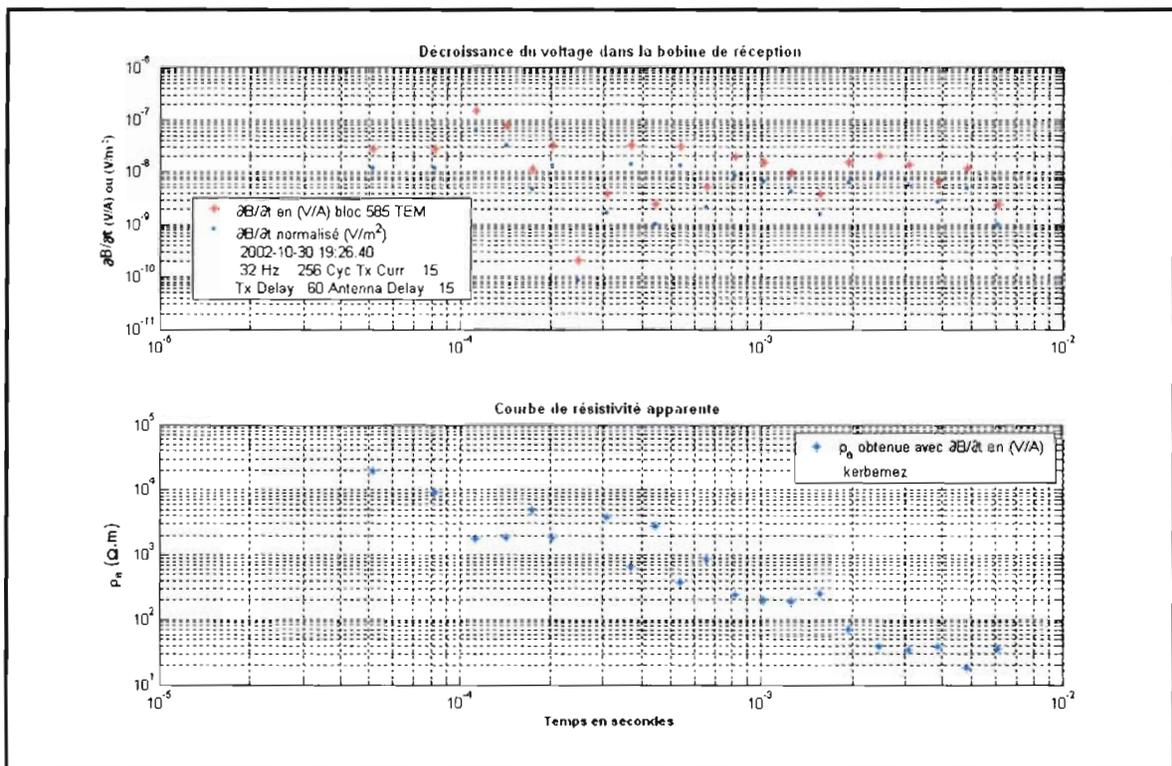


Figure 31 : Représentation graphique de $\partial B/\partial t$ sous Matlab exportée au format *.bmp*

I.17.2 Représentation des données acquises à Aarhus (Danemark)

Une mission, de tests et comparaisons d'équipement TDEM (GEONICS, ZONGE, TEMFAST), a été réalisée conjointement avec l'IRD sur le site test de l'Université d'Aarhus au Danemark du 19 au 23 novembre 2002. Différents tests ont été menés afin de calibrer les réponses obtenues par les différents appareils à une courbe de réponse de référence. Au cours de ces 3 jours de terrain, environ 400 mesures furent acquises.

Comme précédemment, les résultats ne seront pas interprétés ici, l'intérêt est d'illustrer en image la sortie graphique du code. Les figures 32 et 33 sont exportées à partir de Matlab au format *.emf*, respectivement pour le NanoTem et le ZeroTem.

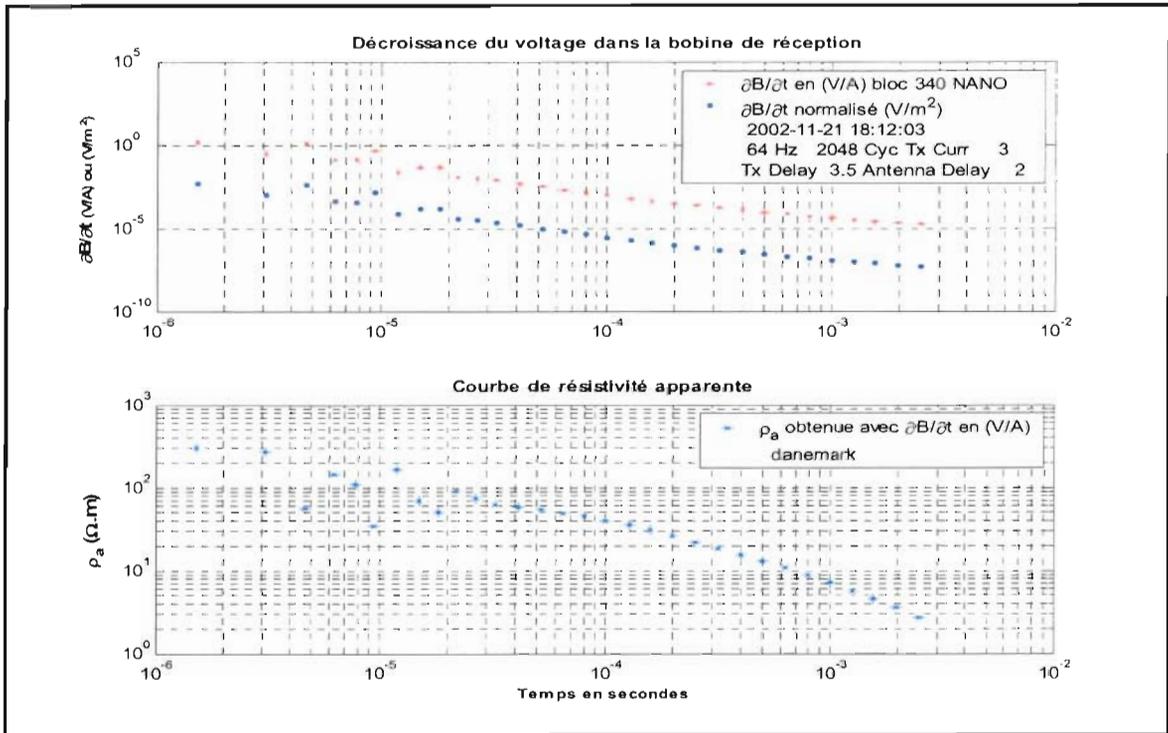


Figure 32 : Représentation graphique de $\partial B/\partial t$ (NanoTem) exportée au format *.emf*

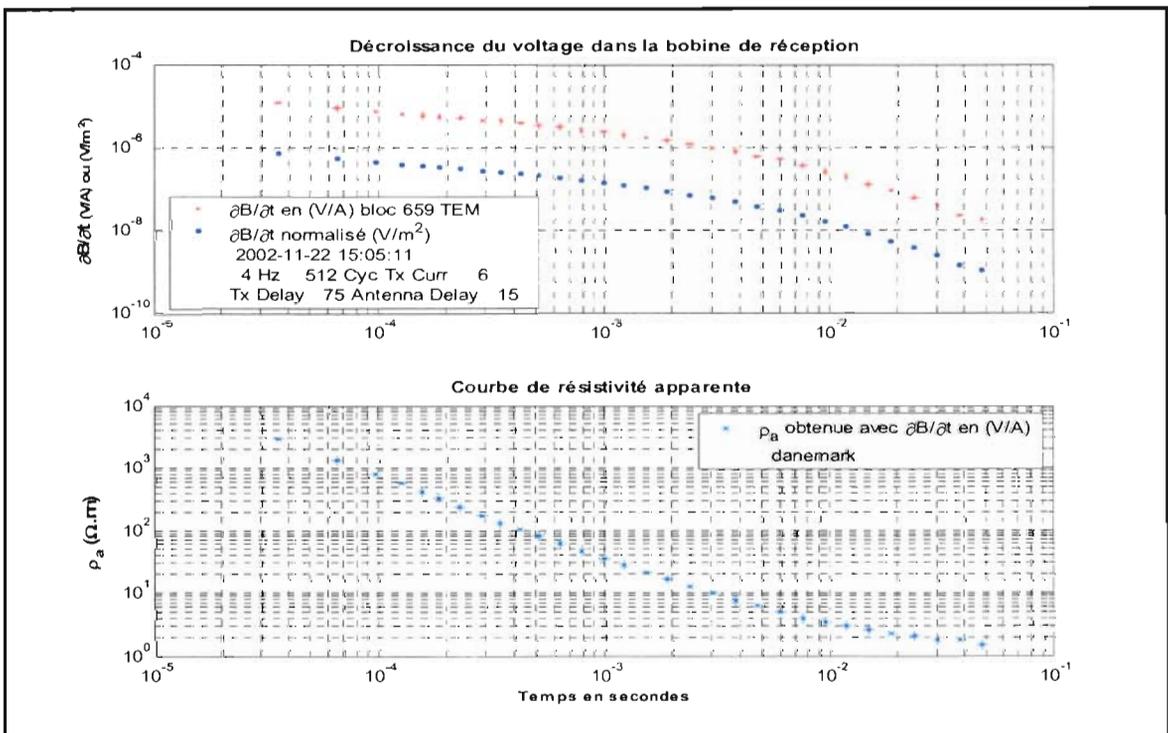


Figure 33 : Représentation graphique de $\partial B/\partial t$ (ZeroTem) exportée au format *.emf*

IV.7 Perspectives et développement

La mission à Kerbernez a initié l'idée de développer sous Matlab un outil de visualisation rapide des données avec le GDP32II, la première opération, après l'acquisition des données, est la représentation correcte de ces dernières. L'utilisation de ce code est aisée même pour un utilisateur non initié à Matlab. Suite à la mission au Danemark, ce code doit être développé dans plusieurs directions (illustré en pointillé et gris clair sur la figure 30) :

i) prévoir les sorties au format *.dat*, pour la représentation éventuelle dans Grapher3 de Golden Software,

ii) préparer les données au format de fichier *.tel* qui est le format d'entrée du logiciel danois Sitem avec choix des mesures à convertir à ce format, et représentation des blocs choisis dans Matlab (le code présenté fait tout le traitement et la représentation des données de façon automatique),

iii) développer le code de façon à formater, représenter et convertir le format de données du TEMFAST 32 au format d'entrée dans Sitem,

iv) à long terme, de transformer ce code afin qu'il accepte directement les données au format binaire, c'est-à-dire éliminer l'étape de transfert des données via HyperTerminal.

V Conclusion

Ce stage nous aura permis de nous familiariser avec le langage de programmation Matlab et de façon plus général à la compréhension et l'écriture d'un algorithme, de comprendre et réaliser des instructions logique (boucle *for*, *if*, *while*, etc.). Les scripts présentés ici ne sont pas issus du travail d'un informaticien mais d'un géophysicien utilisant quelques routines qui facilitent et automatisent le formatage et la représentation des données géophysiques.

Les résultats de la prospection multi-électrode sur le site d'Agen a clairement montré que les méthodes géophysiques peuvent amener des outils supplémentaires à l'irrigation de précision, des projets de développement de ces outils combinés à la télédétection à l'échelle européenne sont discutés en ce moment.

Le script de formatage des données TDEM présenté dans ce mémoire est amené à être développé, notamment en intégrant de nouvelles routines qui convertirons les fichiers, obtenus avec d'autres équipement TDEM, afin d'être utilisés dans les logiciels de traitement danois (Sitem et Semdi).

L'utilisation de Matlab pour le développement de script qui ont servi à l'automatisation pour le formatage des données électriques 3D et le développement d'un outil pour le contrôle qualité des données du Zonge, s'est révélée être une aide précieuse.

BIBLIOGRAPHIE

BOUZID N.. – Mise au point de protocoles d'acquisitions et évaluation des données TDEM. – Mémoire de Maîtrise de Géologie Fondamentale, Université Pierre et Marie Curie, Paris, 2001, 53 p.

CARON C.. - Application de la géophysique a l'arboriculture de précision, Etude des hétérogénéités de l'alimentation hydrique. – Mémoire de DESS de Géophysique Appliquée, Université Pierre et Marie Curie, Paris, 2002, 88 p.

DESCLOITRES M.. - Les sondages électromagnétiques en domaine temporel (TDEM) : Application à la prospection d'aquifères sur les volcans de Fogo (CAP VERT) et DU PITON de la Fournaise (La Réunion). – Thèse de Doctorat de Géophysique, Université Pierre et Marie Curie, Paris, 1998, 238 p.

GEOTOMO SOFTWARE. – Res2Dinv ver.3.4. – Rapid 2D Resistivity & IP inversion using the least-squares method. – Geoelectrical Imaging 2D & 3D, 2001, 94 p.

GEOTOMO SOFTWARE. – Res3Dinv ver.2.11. – Rapid 3D Resistivity & IP inversion using the least-squares method. – Geoelectrical Imaging 2D & 3D, 2002, 51 p.

LOKE M.H.. - Electrical imaging surveys for environmental and engineering studies. - A practical guide to 2-D and 3-D surveys, 2000, 60 p.

ROBAIN H. & BOBACHEV A.. - X2ipi : Tool box for 2D DC measurements with Syscal equipment. - User Manual, 2002, 25 p.

ZONGE. – Transient electromagnetism program. – section 12, pp 1-51, GDP32II INSTRUCTION MANUEL, 2001.

SOMMAIRE DES ANNEXES

SOMMAIRE DES ANNEXES	i
ANNEXE I : Principales fonctions de Matlab	ii
ANNEXE II : Liens et ouvrages utiles.....	vi
ANNEXE III : Fichier head.m	vii
ANNEXES III : Fichier montpellier.m	viii
ANNEXE IV: Centre des fenêtres temporelles ZeroTem.....	xxii
ANNEXE IV : Centre des fenêtres temporelles ZeroTem.....	xxiii
ANNEXE IV : Centre des fenêtres Nanotem.....	xxiv
ANNEXE IV : Centre des fenêtres NanoTem	xxv
ANNEXE V : Fichier Z2tem.m.....	xxvi

ANNEXE I : Principales fonctions de Matlab

General Purpose Commands

Managing Commands and Functions

help Online help for MATLAB functions and M-files
helpdesk Display Help Desk page in Web browser, giving access to extensive help
help for all commands

Managing Variables and the Workspace

clear Remove items from memory
disp Display text or array
length Length of vector
load Retrieve variables from disk
pack Consolidate workspace memory
save Save workspace variables on disk
saveas Save figure or model using specified format
size Array dimensions
who, whos List directory of variables in memory
workspace Display the Workspace Browser, a GUI for managing the workspace.

Controlling the Command Window

clc Clear command window
echo Echo M-files during execution
format Control the output display format

Working with Files and the Operating Environment

cd Change working directory
copyfile Copy file
delete Delete files and graphics objects
dir Directory listing
ls List directory on UNIX
mkdir Make directory
pwd Display current directory
! Execute operating system command

Operators and Special Characters

+ Plus
- Minus
* Matrix multiplication
. * Array multiplication
^ Matrix power
.^ Array power
kron Kronecker tensor product.1-4
\ Backslash or left division
/ Slash or right division
./ and \ Array division, right and left
: Colon

() Parentheses
[] Brackets
{ } Curly braces
. Decimal point
... Continuation
, Comma
; Semicolon
% Comment
! Exclamation point
' Transpose and quote
' Nonconjugated transpose
= Assignment
== Equality
< > Relational operators
& Logical AND
| Logical OR
~ Logical NOT
xor Logical EXCLUSIVE OR

Logical Functions

all Test to determine if all elements are nonzero
any Test for any nonzeros
exist Check if a variable or file exists
find Find indices and values of nonzero elements
*is** Detect state
isa Detect an object of a given class
logical Convert numeric values to logical

Language Constructs and Debugging

MATLAB as a Programming Language

eval Interpret strings containing MATLAB expressions
evalc Evaluate MATLAB expression with capture.
evalin Evaluate expression in workspace
feval Function evaluation
function Function M-files
global Define global variables
nargchk Check number of input arguments

Control Flow

break Terminate execution of for loop or while loop
case Case switch
catch Begin catch block
else Conditionally execute statements
elseif Conditionally execute statements
end Terminate for, while, switch, try, and if statements or indicate last index
for Repeat statements a specific number of times

if Conditionally execute statements
otherwise Default part of switch statement
return Return to the invoking function
switch Switch among several cases based on expression
try Begin try block
warning Display warning message
while Repeat statements an indefinite number of times

Interactive Input

input Request user input
keyboard Invoke the keyboard in an M-file
menu Generate a menu of choices for user input
pause Halt execution temporarily

Object-Oriented Programming

double Convert to double precision
int8, *int16*, *int32*
 Convert to signed integer
uint8, *uint16*, *uint32*
 Convert to unsigned integer

Elementary Matrices and Matrix Manipulation

Elementary Matrices and Arrays

eye Identity matrix
ones Create an array of all ones
rand Uniformly distributed random numbers and arrays
randn Normally distributed random numbers and arrays
zeros Create an array of all zeros
 : (colon) Regularly spaced vector

Special Variables and Constants

ans The most recent answer
eps Floating-point relative accuracy
flops Count floating-point operations
i Imaginary unit.
Inf Infinity
j Imaginary unit
NaN Not-a-Number
nargin, *nargout*
 Number of function arguments
pi Ratio of a circle's circumference to its diameter.
varargin,
varargout Pass or return variable numbers of arguments

Time and Dates

calendar Calendar
clock Current time as a date vector
cputime Elapsed CPU time
date Current date string
etime Elapsed time
now Current date and time

tic, *toc* Stopwatch timer

Matrix Manipulation

cat Concatenate arrays
diag Diagonal matrices and diagonals of a matrix
fliplr Flip matrices left-right
flipud Flip matrices up-down
 repmat Replicate and tile an array
reshape Reshape array
rot90 Rotate matrix 90 degrees
tril Lower triangular part of a matrix
triu Upper triangular part of a matrix
 : (colon) Index into array, rearrange array.

Elementary Math Functions

abs Absolute value and complex magnitude
acos, *acosh* Inverse cosine and inverse hyperbolic cosine
acot, *acoth* Inverse cotangent and inverse hyperbolic cotangent
acsc, *acsch* Inverse cosecant and inverse hyperbolic cosecant
angle Phase angle
asec, *asech* Inverse secant and inverse hyperbolic secant
asin, *asinh* Inverse sine and inverse hyperbolic sine
atan, *atanh* Inverse tangent and inverse hyperbolic tangent
atan2 Four-quadrant inverse tangent
ceil Round toward infinity
complex Construct complex data from real and imaginary components
conj Complex conjugate
cos, *cosh* Cosine and hyperbolic cosine
cot, *coth* Cotangent and hyperbolic cotangent
csc, *csch* Cosecant and hyperbolic cosecant
exp Exponential
fix Round towards zero
floor Round towards minus infinity
gcd Greatest common divisor
imag Imaginary part of a complex number
lcm Least common multiple
log Natural logarithm
log2 Base 2 logarithm and dissect floating-point numbers into exponent and mantissa
log10 Common (base 10) logarithm
mod Modulus (signed remainder after division)
nchoosek Binomial coefficient or all combinations.
real Real part of complex number
rem Remainder after division
round Round to nearest integer
sec, *sech* Secant and hyperbolic secant
sign Signum function
sin, *sinh* Sine and hyperbolic sine
sqrt Square root
tan, *tanh* Tangent and hyperbolic tangent

Eigenvalues and Singular Values

eig Eigenvalues and eigenvectors
gsvd Generalized singular value decomposition
svd Singular value decomposition

Data Analysis and Fourier Transform Functions

Basic Operations

max Maximum elements of an array
mean Average or mean value of arrays
median Median value of arrays
min Minimum elements of an array
perms All possible permutations
prod Product of array elements
sort Sort elements in ascending order
sortrows Sort rows in ascending order
std Standard deviation
sum Sum of array elements
var Variance
voronoi Voronoi diagram

Finite Differences

del2 Discrete Laplacian
diff Differences and approximate derivatives.
gradient Numerical gradient

Correlation

corrcoef Correlation coefficients
cov Covariance matrix

Filtering and Convolution

conv Convolution and polynomial multiplication
conv2 Two-dimensional convolution
deconv Deconvolution and polynomial division
filter Filter data with an infinite impulse response (IIR) or finite impulse response (FIR) filter
filter2 Two-dimensional digital filtering

Fourier Transforms

abs Absolute value and complex magnitude
angle Phase angle
fft One-dimensional fast Fourier transform
fft2 Two-dimensional fast Fourier transform
ifft Inverse one-dimensional fast Fourier transform
ifft2 Inverse two-dimensional fast Fourier transform
unwrap Correct phase angles

Polynomial and Interpolation Functions

Polynomials

conv Convolution and polynomial multiplication
deconv Deconvolution and polynomial division

Sound Processing Functions

General Sound Functions

sound Convert vector into sound

soundsc Scale data and play as sound

SPARCstation-Specific Sound Functions

aread Read NeXT/SUN (.au) sound file
awrite Write NeXT/SUN (.au) sound file

.WAV Sound Functions

wavread Read Microsoft WAVE (.wav) sound file
wavwrite Write Microsoft WAVE (.wav) sound file.

Character String Functions

General

abs Absolute value and complex magnitude
eval Interpret strings containing MATLAB expressions
real Real part of complex number
strings MATLAB string handling

String to Number Conversion

char Create character array (string)
int2str Integer to string conversion
mat2str Convert a matrix into a string
num2str Number to string conversion
sprintf Write formatted data to a string
sscanf Read string under format control
str2double Convert string to double-precision value
str2num String to number conversion

Low-Level File I/O Functions

File Opening and Closing

fclose Close one or more open files
fopen Open a file or obtain information about open files

Unformatted I/O

fread Read binary data from file
fwrite Write binary data to a file

Formatted I/O

fgetl Return the next line of a file as a string without line terminator(s)
fgets Return the next line of a file as a string with line terminator(s)
fprintf Write formatted data to file
fscanf Read formatted data from file

File Positioning

feof Test for end-of-file
ferror Query MATLAB about errors in file input or output
frewind Rewind an open file
fseek Set file position indicator
ftell Get file position indicator

String Conversion

sprintf Write formatted data to a string
sscanf Read string under format control

Specialized File I/O

imfinfo Return information about a graphics file
imread Read image from graphics file.
imwrite Write an image to a graphics file
textread Read formatted data from text file

Multidimensional Array Functions

reshape Reshape array

Plotting and Data Visualization

Basic Plots and Graphs

bar Vertical bar chart
barh Horizontal bar chart
hist Plot histograms
hold Hold current graph
loglog Plot using log-log scales
plot Plot vectors or matrices.
semilogx Semi-log scale plot
semilogy Semi-log scale plot
subplot Create axes in tiled positions

Three-Dimensional Plotting

plot3 Plot lines and points in 3-D space

Plot Annotation and Grids

grid Grid lines for 2-D and 3-D plots
gtext Place text on a 2-D graph using a mouse
legend Graph legend for lines and patches
plotyy Plot graphs with Y tick labels on the left and right
title Titles for 2-D and 3-D plots
xlabel X-axis labels for 2-D and 3-D plots
ylabel Y-axis labels for 2-D and 3-D plots
zlabel Z-axis labels for 3-D plots

Surface, Mesh, and Contour Plots

contour Contour (level curves) plot
meshe Combination mesh/contourplot
mesh 3-D mesh with reference plane
peaks A sample function of two variables
surf 3-D shaded surface graph
surface Create surface low-level objects
surfc Combination surf/contourplot
surfll 3-D shaded surface with lighting

Domain Generation

griddata Data gridding and surface fitting
meshgrid Generation of X and Y arrays for 3-D plots

Color Operations

colormap Set the color look-up table

hsv2rgb Hue-saturation-value to red-green-blue conversion
rgb2hsv RGB to HSV conversion
rgbplot Plot color map

Colormaps

bone Gray-scale with a tinge of blue color map
contrast Gray color map to enhance image contrast
cool Shades of cyan and magenta color map
copper Linear copper-tone color map
flag Alternating red, white, blue, and black color map
gray Linear gray-scale color map
hot Black-red-yellow-white color map
hsv Hue-saturation-value (HSV) color map
spring Shades of magenta and yellow color map
summer Shades of green and yellow colormap
winter Shades of blue and green color map

Printing

print Print graph or save graph to file
printopt Configure local printer defaults
saveas Save figure to graphic file

Handle Graphics, Object Creation

axes Create Axes object
figure Create Figure (graph) windows
image Create Image (2-D matrix)
line Create Line object (3-D polylines)
text Create Text object (character strings)

Handle Graphics, Figure Windows

capture Screen capture of the current figure
clc Clear figure window
clf Clear figure
clg Clear figure (graph window)
close Close specified window
gcf Get current figure handle
newplot Graphics M-file preamble for NextPlot property
refresh Refresh figure
saveas Save figure or model to desired output format

Handle Graphics, Axes

axis Plot axis scaling and appearance
cla Clear Axes
gca Get current Axes handle

Interactive User Input

ginput Graphical input from a mouse or cursor
zoom Zoom in and out on a 2-D plot

Region of Interest

drawnow Complete any pending drawing

ANNEXE II : Liens et ouvrages utiles

OUVRAGES :

D.M.Etter, D.C. Kentucky, "Introduction to Matlab", Prentice Hall, 1999.

W.J.Palm, McGraw-Hill, "Introduction to Matlab for Engineers", 1998.

J.R. Buck, M.M. Daniel, A.C. Singer, "Computer explorations in signals and systems using Matlab", Prentice Hall, 1997.

The MathWorks Inc, "The Student Edition of MATLAB", Edition Prentice-Hall, 1992.

Marvin Marcus, "Matrices and Matlab", Edition Prentice-Hall, 1993.

Duane Hanselman, Bruce Littlefield, "The Student Editor of Matlab: "High performance Numerical Computation and Vizualisation Software", Version 4. Edition Prentice-Hall, 1995.

Duane Hanselman, Bruce Littlefield, "MASTERING Matlab: a comprehensive tutorial and reference", Edition Prentice-Hall, 1996.

LIENS INTERNET :

<http://www.mathworks.com>

<http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml>

<http://www.mathworks.fr>

<http://www.mathworks.fr/products/matlab/description2.shtml>

<http://www.math.utah.edu/lab/ms/matlab/matlab.html>

<http://users.ece.gatech.edu/~bonnie/book/TUTORIAL/tutorial.html>

<http://www.site.uottawa.ca/~bouchard/elg3520/elg3520plan02.pdf>

<http://www.sciences.univ-nantes.fr/physique/perso/aloui/matlab/matladex.htm>

<http://www.ann.jussieu.fr/~postel/matlab>

<http://www.indiana.edu/~statmath/math/matlab/links.html>

<http://www.gel.ulaval.ca/~lehuy/intromatlab/intromat.pdf>

http://www.unilim.fr/pages_perso/paul.armand/matlab/doc/d.pdf

<http://www-syscom.univ-mlv.fr/~bianchi/ig2000/polymatlab.pdf>

<http://www-prima.imag.fr/Prima/jlc/Courses/2000/RICM.TS/RICM.TS.S2.TD.text>

ANNEXE III : Fichier head.m

```
% HEAD.in fichier de commande pour executer la lecture des différents fichiers de mesure, nécessite les quatres fichiers .m suivant : lec3Dpp.m ;  
% lec3Ddpp.m ; lec3Ddpdp.m ; ltrim.m (fonction qui permet de récupérer différentes variables partir du fichier texte  
true=0;  
while true==0  
    arraytype = input( 'choix du dispositif (2 pour pole-pole,3 pour le dipole-dipole, 6 pour le pole-dipole) : \n'); % entrée interactive  
    if arraytype==2  
        disp('Lecture et écriture du fichier au format res3dinv pour le dispositif pole-pole')  
        montpellier  
        true=1  
    elseif arraytype==3  
        disp('Lecture et écriture du fichier au format res3dinv pour le dispositif dipole-dipole')  
        lec3Ddpdp  
        true=1;  
    elseif arraytype==6  
        true1=0;  
        while true1==0  
            dd=input('AMN (1) ou MNB (2) ? \n');  
            if dd==1  
                lec3Dpdpa  
                true1=1;  
            elseif dd==2  
                lec3Dpdpa  
                true1=1;  
            else  
                disp('choix incorrect')  
                true1==0;  
            end  
        end  
    else  
        disp('choix incorrect')  
        true==0;  
    end  
end  
end
```

ANNEXES III : Fichier montpellier.m

```
% MONTPELLIER.m. Lecture de fichier de données .txt exporter d'ELECTRE pour le dispositif pole-pole avec 64 électrodes (65-128)
headfich = input('nom du fichier à lire (avec son extension): \n','s'); % entrée interactive nom du fichier avec son extension
closefich = input('nom de sauvegarde (avec son extension) : \n','s');
disp(closefich) %affiche la valeur de la variable
xgrid=input('x grid size : \n','s');
c=str2num(xgrid);
ygrid=input('y grid size : \n','s');
d=str2num(ygrid);
xelec=input('distance x inter-électrodes : \n', 's');
a=str2num(xelec);
yelec=input('distance y inter-électrodes : \n', 's');
b=str2num(yelec);
FID3=fopen(headfich,'rt');
i=1;
while i<4
    ch=fgetl(FID3);
    i=i+1;
end
ch4=fgetl(FID3);
%-----lecture du nombre de mesure-----%
ch4=strrep(ch4,',',''); %strrep:remplace les occurrences d'une chaine ch1, dans ch2, par ch3
[aa, nbmesure]=strtok(ch4); %strtok:retourne le 1er mot d'une chaine et le reste séparé par délimiteur
nbmesure=ltrim(nbmesure); %ltrim supprime les espaces à gauche d'une chaine
nbmesure=str2num(nbmesure);
disp(['Le fichier contient ',num2str(nbmesure),' mesures '])
%-----%
n=4;
for i=1:n,
    ch=fgetl(FID3);
end
ch9=fgetl(FID3);
%-----lecture du commentaire-----%
ch9=strrep(ch9,',','');
[nom, commentaire] = strtok(ch9);
commentaire=ltrim(commentaire);
n=4;
for i=1:n,
```

```

    ch=fgetl(FID3);
end
coornodes=fscanf(FID3,'%f',[4,128]);
nu = coornodes(1,65:128);
X = coornodes(2,65:128);
Y= coornodes(3,65:128);
Z = coornodes(4,65:128);
n=5;
for i=1:n,
    ch=fgetl(FID3);
end
n=nbmesure;
for i=1:n,
    ch=fgetl(FID3);
end
[nums A B M N K] = textread (headfich,'%d %f %s %f %s %f' , nbmesure , 'headerlines',145);
B1=zeros(size(B));
N1=zeros(size(N));
%------%
% find and replace coordonates for 65-72 or 65-80
replace=input ('Le carré de 64 électrodes a-t-il pour première colonne les électrodes :\n65 puis 72 (1) ou 65 puis 80 (2)? \n');
if replace==1
    Ax = A;
    for n=65:8:121
        i=find(Ax==n);
        Ax(i)=0;
    end
    for n=66:8:122
        i=find(Ax==n);
        Ax(i)=1*a;
    end
    for n=67:8:123
        i=find(Ax==n);
        Ax(i)=2*a;
    end
    for n=68:8:124
        i=find(Ax==n);
        Ax(i)=3*a;
    end
end

```

```

for n=69:8:125
    i=find(Ax==n);
    Ax(i)=4*a;
end
for n=70:8:126
    i=find(Ax==n);
    Ax(i)=5*a;
end
for n=71:8:127
    i=find(Ax==n);
    Ax(i)=6*a;
end
for n=72:8:128
    i=find(Ax==n);
    Ax(i)=7*a;
end
Mx = M;
for n=65:8:121
    i=find(Mx==n);
    Mx(i)=0;
end
for n=66:8:122
    i=find(Mx==n);
    Mx(i)=1*a;
end
for n=67:8:123
    i=find(Mx==n);
    Mx(i)=2*a;
end
for n=68:8:124
    i=find(Mx==n);
    Mx(i)=3*a;
end
for n=69:8:125
    i=find(Mx==n);
    Mx(i)=4*a;
end
for n=70:8:126
    i=find(Mx==n);

```

```
    Mx(i)=5*a;
end
for n=71:8:127
    i=find(Mx==n);
    Mx(i)=6*a;
end
for n=72:8:128
    i=find(Mx==n);
    Mx(i)=7*a;
end
Ay = A;
for n=65:1:72
    i=find(Ay==n);
    Ay(i)=0;
end
for n=73:1:80
    i=find(Ay==n);
    Ay(i)=1*b;
end
for n=81:1:88
    i=find(Ay==n);
    Ay(i)=2*b;
end
for n=89:1:96
    i=find(Ay==n);
    Ay(i)=3*b;
end
for n=97:1:104
    i=find(Ay==n);
    Ay(i)=4*b;
end
for n=105:1:112
    i=find(Ay==n);
    Ay(i)=5*b;
end
for n=113:1:120
    i=find(Ay==n);
    Ay(i)=6*b;
end
```

```

    for n=121:1:128
        i=find(Ay==n);
        Ay(i)=7*b;
    end
My = M;
    for n=65:1:72
        i=find(My==n);
        My(i)=0;
    end
    for n=73:1:80
        i=find(My==n);
        My(i)=1*b;
    end
    for n=81:1:88
        i=find(My==n);
        My(i)=2*b;
    end
    for n=89:1:96
        i=find(My==n);
        My(i)=3*b;
    end
    for n=97:1:104
        i=find(My==n);
        My(i)=4*b;
    end
    for n=105:1:112
        i=find(My==n);
        My(i)=5*b;
    end
    for n=113:1:120
        i=find(My==n);
        My(i)=6*b;
    end
    for n=121:1:128
        i=find(My==n);
        My(i)=7*b;
    end
elseif replace==2
c0=[65 80 81 96 97 112 113 128];

```

```
c1=[66 79 82 95 98 111 114 127];
c2=[67 78 83 94 99 110 115 126];
c3=[68 77 84 93 100 109 116 125];
c4=[69 76 85 92 101 108 117 124];
c5=[70 75 86 91 102 107 118 123];
c6=[71 74 87 90 103 106 119 122];
c7=[72 73 88 89 104 105 120 121];
l0=[65:1:72];
l1=[80:-1:73];
l2=[81:1:88];
l3=[96:-1:89];
l4=[97:1:104];
l5=[112:-1:105];
l6=[113:1:120];
l7=[128:-1:121];
```

```
Ax = A;
for n=c0
    i=find(Ax==n);
    Ax(i)=0;
end
for n=c1
    i=find(Ax==n);
    Ax(i)=1*a;
end
for n=c2
    i=find(Ax==n);
    Ax(i)=2*a;
end
for n=c3
    i=find(Ax==n);
    Ax(i)=3*a;
end
for n=c4
    i=find(Ax==n);
    Ax(i)=4*a;
end
for n=c5
```

```
i=find(Ax==n);
Ax(i)=5*a;
end
for n=c6
i=find(Ax==n);
Ax(i)=6*a;
end
for n=c7
i=find(Ax==n);
Ax(i)=7*a;
end
Mx = M;
for n=c0
i=find(Mx==n);
Mx(i)=0;
end
for n=c1
i=find(Mx==n);
Mx(i)=1*a;
end
for n=c2
i=find(Mx==n);
Mx(i)=2*a;
end
for n=c3
i=find(Mx==n);
Mx(i)=3*a;
end
for n=c4
i=find(Mx==n);
Mx(i)=4*a;
end
for n=c5
i=find(Mx==n);
Mx(i)=5*a;
end
for n=c6
i=find(Mx==n);
Mx(i)=6*a;
```

```
end
for n=c7
    i=find(Mx==n);
    Mx(i)=7*a;
end
Ay = A;
for n=l0
    i=find(Ay==n);
    Ay(i)=0;
end
for n=l1
    i=find(Ay==n);
    Ay(i)=1*a;
end
for n=l2
    i=find(Ay==n);
    Ay(i)=2*a;
end
for n=l3
    i=find(Ay==n);
    Ay(i)=3*a;
end
for n=l4
    i=find(Ay==n);
    Ay(i)=4*a;
end
for n=l5
    i=find(Ay==n);
    Ay(i)=5*a;
end
for n=l6
    i=find(Ay==n);
    Ay(i)=6*a;
end
for n=l7
    i=find(Ay==n);
    Ay(i)=7*a;
end
My = M;
```

```

for n=10
    i=find(My==n);
    My(i)=0;
end
for n=11
    i=find(My==n);
    My(i)=1*a;
end
for n=12
    i=find(My==n);
    My(i)=2*a;
end
for n=13
    i=find(My==n);
    My(i)=3*a;
end
for n=14
    i=find(My==n);
    My(i)=4*a;
end
for n=15
    i=find(My==n);
    My(i)=5*a;
end
for n=16
    i=find(My==n);
    My(i)=6*a;
end
for n=17
    i=find(My==n);
    My(i)=7*a;
end
end
t=[A Ax Ay M Mx My];
%-----%
quadripole= [nums A B1 M N1 K];
n=5;
for i=1:n,
    ch=fgetl(FID3);

```

```

end
nbmesurel=nbmesure+149;
[nums1 V I Rho q] = textread (headfich,'%d%f%f%f%f' ,-1 , 'headerlines',nbmesurel);
dat=[Ax Ay Mx My Rho];
%-----Ecriture dans un fichier pour res3dinv-----%
fid = fopen(closefich,'wt');
fprintf(fid,commentaire);
fprintf(fid,'\n %s\t',xgrid);
fprintf(fid,'\n %s\t',ygrid);
fprintf(fid,'\n %s\t',xelec);
fprintf(fid,'\n %s\t',yelec);
fprintf(fid,'\n%hd\n',arraytype);
fprintf(fid,' %hd\n',nbmesure);
fprintf(fid,'%6.2f\t%6.2f\t%6.2f\t%6.2f\t%6.2f\t\n',dat);
true=0;
while true==0
n=input('Topographie ? (Oui(0)/Non(1)): \n ');
if n==0
true1=0;
while true1==0
m=input('Est-elle contenue dans le fichier de données brutes ? oui(0)/non(1): \n');
if m==0
topog=Z([1:8 ; 9:16; 17:24 ; 25:32 ; 33: 40 ; 41:48 ; 49 :56 ; 57 :64]);
topog
fprintf(fid,'Topography');
fprintf(fid,'\n2\n');
fprintf(fid,'%5.2f%5.2f%5.2f%5.2f%5.2f%5.2f%5.2f\n',topog');
true3=0;
while true3==0
g=input('La topographie est-elle correcte ? (oui(0)/non(1)): \n');
if g==0
true4=0;
while true4==0
h=input('Voulez vous visualiser la topographie ? oui(0)/non(1): \n');
if h==0
x=0:a:(c-1)*a;
y=0:b:(d-1)*b;
[X,Y]=meshgrid(x,y);
Z=topog;

```

```

    % représentation en 'fil de fer' : mesh
    figure (1)
    mesh(X,Y,Z)
    colormap(cool)
    title('Visualisation de la topographie')
    xlabel('x'), ylabel('y'), zlabel('altitude')
    true4=1;
    true3=1;
    true2=1;
    true1=1;
elseif h==1
    true4=1;
    true3=1;
    true2=1;
    true1=1;
else
    disp('Choix incorrect')
    true4=0;
end
end
disp('Le bloc topo a été crée !')
true3=1;
true2=1;
true1=1;
true=1;
elseif g==1
    true3=1;
    true2=1;
    true1==0;
else
    disp('choix incorrect')
    true3==0;
end
end

elseif m==1
    true2=0;
    while true2==0
        t=input ('Voulez vous l'encoder maintenant ? (oui(0)/non(1)): \n');
    end
end

```

```

if t==0
    topog= input ('Ecrire une matrice de la forme :\n[8 valeurs de la 1ere ligne ";"8 valeurs de la 2eme ligne";"etc jusqu"à la 8eme ligne] \navec les valeurs
lues sur la mire données en mm :\n');
    topog=-topog./1000;
    topog
    fprintf(fid,'Topography');
    fprintf(fid,'\n2\n');
    fprintf(fid,'%5.2ft%5.2ft%5.2ft%5.2ft%5.2ft%5.2ft%5.2ft\n',topog);
    true3=0;
    while true3==0
        g=input('La topographie est-elle correcte ? (oui(0)/non(1)): \n');
        if g==0
            true4=0;
            while true4==0
                h=input('Voulez vous visualiser la topographie ? oui(0)/non(1): \n');
                if h==0
                    x=0:a:(c-1)*a;
                    y=0:b:(d-1)*b;
                    xi=0:a/2:(c-1)*a;
                    yi=0:b/2:(d-1)*b;
                    Z=topog;
                    [X,Y]=meshgrid(x,y);
                    [Xi,Yi]=meshgrid(xi,yi);
                    Zi1=interp2(X,Y,Z,Xi,Yi,'nearest');
                    Zi2=interp2(X,Y,Z,Xi,Yi,'bilinear');
                    Zi3=interp2(X,Y,Z,Xi,Yi,'bicubic');
                    figure(1)
                    subplot(3,1,1)
                    contour(Xi,Yi,Zi1)
                    subplot(3,1,2)
                    contour(Xi,Yi,Zi2)
                    subplot(3,1,3)
                    surf(Xi,Yi,Zi3)
                    shading interp
                    figure(3)
                    subplot(3,1,1)
                    mesh(Xi,Yi,Zi1)
                    subplot(3,1,2)
                    mesh(Xi,Yi,Zi2)

```

```

subplot(3,1,3)
mesh(Xi,Yi,Zi3)
% représentation en 'fil de fer' : mesh
figure (2)
meshc(X,Y,Z)
colormap(cool)
title('Visualisation de la topographie')
xlabel('x'), ylabel('y'), zlabel('altitude')
true4=1;
true3=1;
true2=1;
true1=1;
elseif h==1
true4=1;
true3=1;
true2=1;
true1=1;
else
disp('Choix incorrect')
true4=0;
end
end
disp('Le bloc topo a été crée !')
true3=1;
true2=1;
true1=1;
true=1;
elseif g==1
true3=1;
true2==0;
else
disp('choix incorrect')
true3==0;
end
end
elseif t==1
disp('Le fichier crée ne comporte pas de topographie')

```

```

        true2=1;
        true1=1;
        true=1;
    else
        disp('Choix incorrect')
        true2==0;
    end
end
true1=1;
true=1;
else
    disp('Choix incorrect')
    true1==0;
end
end

elseif n==1
    disp('Le fichier crée ne comporte pas de topographie')
    true=1;
else
    disp('Choix incorrect')
    true==0;
end
end
fprintf(fid,'0');
n=4;
for i=1:n,
    fprintf(fid,'\n0');
end
fclose (fid);
fclose (FID3);

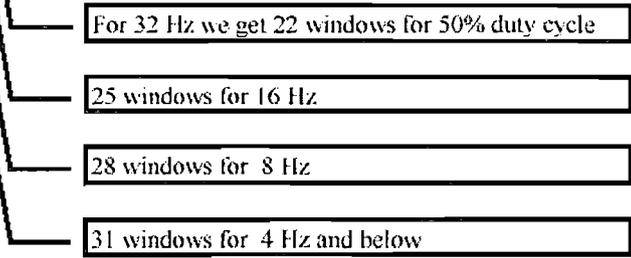
disp(['Le fichier ',closefich,' a été crée'])

```

ANNEXE IV: Centre des fenêtres temporelles ZeroTem

12.8 TEM WINDOW CENTERS FOR ZERO DELAY

WINDOW #S	32, 16, 8 and 4 Hz	2 Hz	1 Hz	.5 Hz	.25 Hz	.125 Hz	.0625 Hz
1	0.0 μ s	0.0 μ s	0.0 μ s	0.0 μ s	0.0 μ s	0.0 ms	0.0 ms
2	30.5	61.0	122.0	244.1	488.3	976.6	1.953
3	61.0	122.1	244.1	488.3	976.6	1.953	3.906
4	91.6	183.1	366.2	732.4	1.465 ms	2.930	5.859
5	122.1	244.1	488.3	976.6	1.953	3.906	7.813
6	152.6	305.2	610.4	1.221 ms	2.441	4.883	9.766
7	197.8	395.6	791.1	1.582	3.164	6.329	12.66
8	259.0	517.9	1.036 ms	2.072	4.143	8.287	16.57
9	320.1	640.2	1.280	2.561	5.121	10.24	20.48
10	395.6	791.1	1.582	3.164	6.329	12.66	25.32
11	487.3	974.7	1.949	3.899	7.797	15.59	31.19
12	607.3	1.215 ms	2.429	4.858	9.717	19.43	38.87
13	774.5	1.549	3.098	6.196	12.39	24.78	49.57
14	972.3	1.945	3.889	7.778	15.56	31.11	62.23
15	1.215 ms	2.429	4.858	9.717	19.43	38.87	77.73
16	1.518	3.037	6.073	12.15	24.29	48.58	97.17
17	1.911	3.821	7.643	15.29	30.57	61.14	122.3
18	2.426	4.852	9.704	19.41	38.82	77.63	155.3
19	3.064	6.128	12.26	24.51	49.02	98.05	196.1
20	3.852	7.704	15.41	30.82	61.63	123.3	246.5
21	4.838	9.676	19.35	38.71	77.41	154.8	309.6
22	6.094	12.19	24.38	48.75	97.50	195.0	390.0
23	7.687	15.37	30.75	61.50	123.0	246.0	492.0
24	9.659	19.32	38.64	77.28	154.6	309.1	618.2
25	12.14	24.29	48.57	97.15	194.3	388.6	777.2
26	15.30	30.59	61.18	122.4	244.7	489.5	978.9
27	19.25	38.51	77.02	154.0	308.1	616.1	1232.3
28	24.24	48.48	96.96	193.9	387.8	775.6	1551.3
29	30.53	61.06	122.1	244.2	488.5	977.0	1954.0
30	38.42	76.83	153.7	307.3	614.7	1229.3	2458.6
31	48.38	96.76	193.5	387.1	774.1	1548.2	3096.4



ANNEXE IV : Centre des fenêtres temporelles ZeroTem

12.9 WINDOW CENTERS AND WIDTHS

32, 16, 8 and 4 HZ REPETITION RATES

WINDOW #S	POINTS PER WINDOW	WINDOW CENTERS	WINDOW WIDTHS	
		32, 16, 8 and 4 Hz	BEGINNING	END
1	1	0.0 μ sec	0.0 μ sec	0.0 μ sec
2	1	30.5	30.5	30.5
3	1	61.0	61.0	61.0
4	1	91.6	91.6	91.6
5	1	122.1	122.1	122.1
6	1	152.6	152.6	152.6
7	2	197.8	183.1	213.6
8	2	259.0	244.1	274.7
9	2	320.1	305.2	335.7
10	3	395.6	366.2	427.3
11	3	487.3	457.8	518.8
12	5	607.3	549.3	671.4
13	6	774.5	701.9	854.5
14	7	972.3	885.0	1.068 m sec
15	9	1.215 m sec	1.099 m sec	1.343
16	11	1.518	1.373	1.678
17	15	1.911	1.709	2.136
18	19	2.426	2.167	2.716
19	23	3.064	2.747	3.418
20	29	3.852	3.449	4.303
21	36	4.838	4.334	5.402
22	47	6.094	5.432	6.836 — 32 Hz
23	58	7.687	6.867	8.606
24	72	9.659	8.637	10.803
25	92	12.14	10.834	13.611 — 16 Hz
26	116	15.30	13.642	17.151
27	145	19.25	17.182	21.576
28	184	24.24	21.607	27.192 — 8 Hz
29	231	30.53	27.222	34.241
30	289	38.42	34.272	43.061
31	369	48.38	43.091	54.322 — 4 Hz

ANNEXE IV : Centre des fenêtres Nanotem

12.23 NanoTEM WINDOW CENTERS

12.23.1 Sample Period: 1.6 Microseconds
Zero Delay After Current Turnoff

WINDOW #S	POINTS PER WINDOW	WINDOW CENTERS	WINDOW WIDTHS BEGINNING	WINDOW WIDTHS END
1	1	0.0 μ sec	0.0 μ sec	0.0 μ sec
2	1	1.6	1.6	1.6
3	1	3.2	3.2	3.2
4	1	4.8	4.8	4.8
5	1	6.4	6.4	6.4
6	1	8.0	8.0	8.0
7	2	10.4	9.6	11.2
8	2	13.6	12.8	14.5
9	2	16.8	16.1	17.7
10	3	20.8	19.3	22.5
11	3	25.6	24.1	27.3
12	5	32.0	28.9	35.3
13	6	40.8	36.9	45.0
14	7	51.2	46.6	56.2
15	9	63.9	57.8	70.7
16	11	79.9	72.3	88.3
17	15	100.6	89.9	112.4
18	18	127.7	114.0	142.9
19	23	161.2	144.5	179.9
20	29	202.7	181.5	226.4
21	36	254.6	228.1	284.3
22	47	320.7	285.9	359.7
23	58	404.5	361.4	452.9
24	72	508.3	454.5	568.5
25	92	639.0	570.1	716.3
26	116	804.9	717.9	902.6
27	145	1.013 m sec	904.2	1.135 m sec
28	184	1.276	1.137 m sec	1.431
29	231	1.607	1.433	1.802
30	289	2.022	1.804	2.266
31	369	2.546	2.268	2.859

ANNEXE IV : Centre des fenêtres NanoTem

12.23.2 Sample Period: 1.2 Microseconds Zero Delay After Current Turnoff

WINDOW #S	POINTS PER WINDOW	WINDOW CENTERS	WINDOW WIDTHS	
			BEGINNING	END
1	1	0.0 μ sec	0.0 μ sec	0.0 μ sec
2	1	1.2	1.2	1.2
3	1	2.4	2.4	2.4
4	1	3.6	3.6	3.6
5	1	4.8	4.8	4.8
6	1	6.0	6.0	6.0
7	2	7.8	7.2	8.4
8	2	10.2	9.6	10.8
9	2	12.6	12.1	13.3
10	3	15.6	14.5	16.9
11	3	19.2	18.1	20.5
12	5	24.0	21.7	26.5
13	6	30.6	27.7	33.7
14	7	38.4	34.9	42.2
15	9	48.0	43.4	53.0
16	11	59.9	54.2	66.3
17	15	75.4	67.5	84.4
18	19	95.8	85.6	107.2
19	23	121.0	108.5	135.0
20	29	152.1	136.2	169.9
21	36	191.0	171.1	213.3
22	47	240.6	214.5	269.9
23	58	303.5	271.1	339.8
24	72	381.4	341.0	426.6
25	92	479.5	427.8	537.4
26	116	604.0	538.6	677.2
27	145	760.2	678.4	851.9
28	184	957.1	853.1	1,074
29	231	1,205	1,075	1,352
30	289	1,517	1,353	1,700
31	369	1,910	1,701	2,145

ANNEXE V : Fichier Z2tem.m

```
clear all
% Version provisoire (au 16-01-03), de visualisation et transformation des données TDEM Zonge % au format d'entré dans SiTem, et grapher
%(Nordine Bouzid, 2003)

chemin=input('Donnez le chemin du répertoire dans lequel \n se trouve vos data depuis la racine : ','s');
nom_fich=input('Nom du fichier (.txt): ','s');
%-----Ouverture du fichier mesures-----%
headfich = [chemin,'\',nom_fich];
%-----Lecture du fichier-----%
t = dir(headfich);
filesize =t.bytes;
fid = fopen(headfich,'rt');
lec_fich = headfich;
compt = 0;
while ~feof(fid)
    line = fgetl(fid);
    if isempty(line)| strcmp(line,'% ',1)
        continue
    end
    compt = compt + 1;
    lec_fich = strvcat(lec_fich,line);
end
fclose (fid)
fclose('all')
disp(lec_fich);
disp(sprintf('%d lignes contenues dans le fichier',compt))
lec_fich = strvcat(lec_fich,'0000'); % ajoute dans la variable lec_fich '0000', pour recuperation du dernier bloc
t
```

```

%-----Extraction des indices des blocs de lec_fich-----%
nb = lec_fich(2,1);
indbloc=strmatch(nb,lec_fich);    %lit les '0'string' des numéros de bloc et retourne les indices dans une matrice
num_bloc=[];
num_bloc=[num_bloc; (lec_fich(indbloc,:))];    %lit la ligne qui correspond à l'indice
num_bloc=str2num (num_bloc);
ind_min=(indbloc(1,:));
ind_max=(indbloc(end,:));
%-----Calcul de la différence entre les indices pour déterminer quel type de bloc-----%
indbloc2=indbloc([2:end]);
indbloc3=indbloc2([1:end-1]);
indbloc3=indbloc3(end:-1:1);
indbloc4=ones(size(indbloc2));
indbloc4([1:end-1])=indbloc3;
indbloc4=indbloc4(end:-1:1);
diff=indbloc2-indbloc4;
diff1=ones(size(num_bloc));
diff1([1:end-1])=diff;
diff1(1)=10;
b=length(diff1);
diff1(b)=37;
diff=diff1;
%-----Test sur la nature du bloc (head or data) et récupération des données-----%
a=length(diff);
a=a-1;
for n=1:a
    if diff(n)==diff(1)
        bmin=(indbloc(n,:));
        bmax=(indbloc(n+1,:));
        comment_bloc=[];
        comptcomment = 0;
    end
end

```

```

for i=bmin:bmax-1
A = lec_fich(i,:);
comment_bloc= strvcats(comment_bloc,A);
comment = ['comment',int2str(num_bloc(n)), '= comment_bloc'];
comptcomment = comptcomment + 1;
end
eval(comment);
disp(sprintf('%d lignes',comptcomment))
else %if diff(n)==37
cmin=(indbloc(n,:));
cmax=(indbloc(n+1,:));
data_bloc=[];
donne=[];
data_bloc1=[];
comptdata = 0;
for i=cmin:cmax-1
B = lec_fich(i,:);
data_bloc= strvcats(data_bloc,B);
data_bloc=char(data_bloc);
data_bloc1=data_bloc;
C=[(data_bloc(7:end,1:7))];
C = regexprep(C,'m','e-03','preserve'); %regexprep remplace chaine de caractere par une expression definie
C = char(deblank(regexprep(C,'u','e-06','preserve')));
W = ['W',int2str(num_bloc(n)), '= (str2num(C))'];
D=[(data_bloc(7:end,8:16))];
D = regexprep(D,'-','+', 'preserve');
D = regexprep(D,'m','e-03','preserve'); %regexprep remplace chaine de caractere par une expression definie
D = char(deblank(regexprep(D,'u','e-06','preserve')));
V = ['V',int2str(num_bloc(n)), '= (str2num(D))'];
E=[(data_bloc(7:end,17:end))];
E = regexprep(E,'K','e+03','preserve'); %regexprep remplace chaine de caractere par une expression definie
E = char(deblank(regexprep(E,'M','e+06','preserve')));

```

```

rho = ['rho',int2str(num_bloc(n)),','+(str2num(E))'];
data_cbloc = regexprep(data_bloc,'-','+','preservecase');
data_cbloc = regexprep(data_cbloc,'m','e-03','preservecase'); %regexprep remplace chaine de caractere par une expression definie
data_cbloc = regexprep(data_cbloc,'u','e-06','preservecase');
data_cbloc = regexprep(data_cbloc,' 0 ','0.e-06'); %regexprep remplace chaine de caractere par une expression definie
data_cbloc = regexprep(data_cbloc,'k','e+03');
data_cbloc = regexprep(data_cbloc,'M','e+06');
data_cbloc = regexprep(data_cbloc,'K','e+03');
data_cbloc = char(data_cbloc);
donne = ['donne',int2str(num_bloc(n)),','+(str2num(data_cbloc(7:end,1:end)))']; % enleve les 7 premières lignes de W309 qui est un char
array et recupère les 10er caractère qui correspond 1ere col
comptdata = comptdata + 1;
end
disp(sprintf('%d lignes',comptdata))
eval(W)
eval(V)
eval(rho)
eval(donne)
end

end
%-----Test sur la nature du bloc (head or data) et récupération des paramètres-----%
%La taille des matrices est différente par rapport au test précédent
for n = 1:a
    if diff(n)==diff(1)
        bmin=(indbloc(n,:));
        bmax=(indbloc(n+1,:));
        comment_bloc=[];
        for i=bmin:bmax-1
            A = lec_fich(i,:);
            comment_bloc= strvcats(comment_bloc,A);
            site=['site',int2str(num_bloc(n)),','+(comment_bloc(3,5:15))']; % récupération du site dans le champs oper
            car_boucle=['car_boucle',int2str(num_bloc(n)),','+(comment_bloc(5,5:end))']; % récupération des caractéristique des boucles

```

```

turn_off = ['turn_off',int2str(num_bloc(n)),'= (comment_bloc(6,1:33))']; % récupération du tx delay et antenna delay
Rxm = ['Rxm',int2str(num_bloc(n)),'= str2num((comment_bloc(5,9:15)))']; % moment du récepteur
Pc = ['Pc',int2str(num_bloc(n)),'= strvcat ((comment_bloc(5,5:end)),(comment_bloc(6,1:34)))'];
end
eval(turn_off);
eval(car_boucle);
eval(Pc)
eval(Rxm)
eval(site)
else
cmin=(indbloc(n,:));
cmax=(indbloc(n+1,:));
data_bloc=[];
for i=cmin:cmax-1
    B = lec_fich(i,:);
    data_bloc= strvcat(data_bloc,B);
    data_bloc=char(data_bloc );
    date_heure_disp=['date_heure_disp',int2str(num_bloc(n)),'= (data_bloc(2,1:39))'];
    freq_curr_alias=['freq_curr_alias',int2str(num_bloc(n)),'= (data_bloc(4,1:end))'];
    Pdleg = ['Pdleg',int2str(num_bloc(n)),'= strcat ((data_bloc(2,1:4)),(data_bloc(2,9:28)),(data_bloc(4,2:32)))'];
    mode = ['mode',int2str(num_bloc(n)),'= strcat ((data_bloc(2,1:4)))'];

    freqce=['freqce',int2str(num_bloc(n)),'= str2num ((data_bloc(4,1:4)))'];
    cadence=['cadence',int2str(num_bloc(n)),'= str2num ((data_bloc(4,9:14)))'];
    I=['I',int2str(num_bloc(n)),'= str2num ((data_bloc(4,30:32)))'];

    Pdleggh = ['Pdleggh',int2str(num_bloc(n)),'= strvcat ((data_bloc(2,9:28)),(data_bloc(4,2:32)))'];
end
eval(freq_curr_alias)
eval(date_heure_disp)
eval (Pdleg)
eval(Pdleggh)

```

```

        eval(mode)
        eval(freqce)
        eval(cadence)
        eval(I)
    end
end
%-----Calcul de Db/dt normalisé par le courant I et Rxm (sitem)-----%
for n = 1:a
    Vn=[];
    if diff(n)==diff(1)
        Rxcoil = eval(['Rxm',int2str(num_bloc(n))]);
    else
        Vn=['Vn',int2str(num_bloc(n)),'= ((eval(["V",int2str(num_bloc(n))])).*eval(num2str(["I",int2str(num_bloc(n))]))./Rxcoil)'];
        eval(Vn)
    end
end
%-----Graphique pour chacun des blocs de données brutes-----%
for n = 1:a
    if diff(n)==diff(1)
        turnoff = eval(['turn_off',int2str(num_bloc(n))]);
        siteL = eval(['site',int2str(num_bloc(n))]);
    else
        figure(num_bloc(n))
        subplot(2,1,1) % partage de la fenetre graphique en deux
        loglog (eval(['W',int2str(num_bloc(n))]),eval(['V',int2str(num_bloc(n))]),'*r',eval(['W',int2str(num_bloc(n))]),...
            eval(['Vn',int2str(num_bloc(n))]),'.b');
    % documentation du graphique
    ylabel('\partial\partial B\partial (V/A) ou (V/m^2)','FontWeight','bold','FontSize',6)
    title('Décroissance du voltage dans la bobine de réception','FontWeight','bold','fontSize',10)
    legend(['\partial\partial en (V/A) bloc ' mat2str((num_bloc(n))) ' ' (eval (strcat ('mode', num2str(num_bloc(n)))))]...
        ,'\partial\partial normalisé (V/m^2)',(eval (strcat ('Pdleg', num2str(num_bloc(n))))),turnoff)
    hold on

```

```

grid on
subplot(2,1,2)
loglog (eval(['W',int2str(num_bloc(n))]),eval(['rho',int2str(num_bloc(n))]),'*b')
grid on
hold on
xlabel ('Temps en secondes','FontWeight','bold','FontSize',8)
ylabel('\fontsize{11}\rho_{a} {\(\fontsize{11}\Omega\fontsize{11}.m)}','FontWeight','bold','FontSize',8)
title ('Courbe de résistivité apparente ','FontWeight','bold','FontSize',10)
legend ('\rho_{a} obtenue avec \partial B/\partial t en (V/A),siteL)
end
end
%-----Graphique avec l'ensemble des données brutes-----%
for n = 1:a
    if diff(n)==diff(1)
    else
    figure(1)
    subplot(2,1,1)
    loglog (eval(['W',int2str(num_bloc(n))]),eval(['V',int2str(num_bloc(n))]),'+r')
    ylabel ('\fontsize{11}\partial\fontsize{10}B/\fontsize{11}\partial t \fontsize{8}(V/A) ou (V/m^2)','FontWeight','bold','FontSize',8)
    title ('Décroissance du voltage dans la bobine de réception','FontWeight','bold','fontSize',10)
    legend ('\partial B/\partial t en (V/A)')
    grid on
    hold on
    subplot(2,1,2)
    loglog (eval(['W',int2str(num_bloc(n))]),eval(['rho',int2str(num_bloc(n))]),'*b')
    grid on
    hold on
    xlabel ('Temps en secondes','FontWeight','bold','FontSize',8)
    ylabel('\fontsize{11}\rho_{a} {\(\fontsize{11}\Omega\fontsize{11}.m)}','FontWeight','bold','FontSize',8)
    title ('Courbe de résistivité apparente ','FontWeight','bold','FontSize',10)
    legend ('\rho_{a} obtenue avec \partial B/\partial t en (V/A)')
    end
end
end

```