

Rapport de stage

GESHKOVSKI Borjan et RODRIGUEZ Charlotte

Table des matières

I	Introduction	3
1	Notions : EEG et VCN	6
1.1	EEG	6
1.2	VCN	6
2	Expérience vigilance-musique	8
2.1	But	8
2.2	Protocol expérimental	8
3	Stockage des données de l'expérience	11
II	Interface Fixe	13
4	??	14
5	Utilisation de l'interface Fixe	15
5.1	Les entrées	15
5.2	Les sorties	17
5.3	Les liens vers d'autres fenêtres	17
III	Interface Defixe	24
6	Utilisation de l'interface Defixe	25
6.1	Interface0	25
6.2	Interface1	28
7	Code de l'interface Defixe	40
7.1	Prog_load()	40
7.2	Charger()	42
7.3	Recup()	43
7.4	Creation_interface()	46
7.5	Callback_load()	47
7.6	Callback_run(detache)	47
7.7	Calcul_aire(numéro_option,matrice,moy)	50
7.8	Plan_casque()	50
7.9	Chargement_triangles_coor(nouveau)	51
7.10	Topo_defixe(defixe)	52

7.11	Callback_menu_moy()	54
7.12	Callback_menu_sujet()	55
7.13	Callback_imagesc()	56
7.14	Appliquer_echelle()	57
7.15	Callback_play()	58
7.16	Callback_vitesse_play()	58
7.17	Callback_3d(tag)	58
7.18	Perspective_oiseaux_depart()	59
7.19	Perspective_oiseaux_arrivee(detache_graphe_global)	59
7.20	Affiche_aire_global(tableau)	64

Première partie

Introduction

Deux expériences mettant en jeu l'analyse de données EEG ont été conduites par Frédérique Faita Ainseba. Dans ce rapport nous les appelons 'expérience sur la musique' (ou 'expérience vigilance-musique') et 'expérience sur la relaxation'. Ce sont des études que nous avons d'abord connues en tant que sujets une année auparavant. Puis nous nous y sommes impliqués lors d'un stage de un mois et demi.

Avant le début de notre stage, et Pierrick Legrand avaient créé une interface graphique codée en Matlab, permettant la visualisation des données récoltées lors de ces expériences sous forme de courbes et de calculs d'aires. Cette interface, d'abord créée pour l'expérience sur la relaxation, avait ensuite été en partie adaptée pour l'expérience sur la musique.

Notre travail se présentait de la façon suivante. Nous devions dans un premier temps réparer le lien entre l'interface créé par Pierrick Legrand (nous l'appellerons 'interface1') et l'interface créée par Julien Clauzel (nous l'appellerons 'topo', pour 'topographique'). Découvrant pour la première fois le langage Matlab, la première démarche a été d'apprendre ce langage et de décortiquer le code pour comprendre son fonctionnement. Puis nous avons pu réparer le 'lien' entre les deux interfaces.

Une fois cette tâche effectuée, nous devions adapter à l'expérience sur la musique, la partie de l'interface qui n'était encore faite que pour l'expérience sur la relaxation. Ces modifications nous ont conduit à contacter Arthur Devemy, étudiant en troisième année de licence MIASHS, dont le TER (Travaux Encadrés de Recherche) portait sur l'expérience musique, afin d'échanger des informations et de récupérer les dernières données de l'expérience.

Une fois le lien refait entre les deux interfaces, nous avons champ libre pour apporter les modifications qui nous semblaient utiles. Nous avons donc ajusté l'affichage des courbes de l'interface1 pour qu'elles soient intégralement affichées. Concernant l'affichage des données sous forme de carte topographique du crâne, nous l'avons modifié de sorte que l'évolution de la différence de potentiel mesurée à la surface du crâne puisse être vue sous forme d'un film, et de sorte que l'utilisateur puisse voir le scalp en trois dimensions. Nous avons également ajouté la visualisation de la carte sous forme d'un graphe en trois dimensions (temps, électrode, différence de potentiel). Puis, nous avons ajouté une fenêtre permettant à l'utilisateur d'avoir une vision globale des résultats, sous forme de graphiques affichant l'aire sous la courbe sur l'intervalle choisi en fonction de l'électrode, ou du sujet.

Enfin, il est apparu que malgré que nous aillions réparé et amélioré l'interface, celle-ci ne resterait utilisable que pour cette expérience, sans l'intervention d'une personne sachant programmer. L'objectif suivant a donc été de créer une nouvelle interface, inspirée de l'ancienne (concernant le travail qu'elle effectuait), mais étant adaptable à toute expérience impliquant l'analyse de données électroencéphalographiques.

Nous voyons ce rapport comme un bilan du travail effectué mais aussi comme un guide pour utiliser les interfaces et un guide pour une meilleure approche du code pour les personnes amenées à modifier, améliorer les interfaces.

Dans une première partie nous parlerons de l'expérience sur la musique et de notions importantes à connaître pour l'étude. Dans les parties suivantes nous décrirons les deux interfaces (Fixe et Defixe) sur lesquelles nous avons travaillé. Nous aborderons également l'interface Defixe du point de vue de la programmation, cela nous semblant utile pour futurs stagiaires devant à leur tour travailler sur cette interface et la modifier.

Chapitre 1

Notions : EEG et VCN

1.1 EEG

L'électroencéphalographie (EEG) est une méthode permettant d'enregistrer l'activité électrique à la surface du crâne. Il capte le champ électrique résultant des communications entre neurones à différents endroits du scalp. Le dispositif de recueil des données est un casque muni d'électrodes placées en contact avec le crâne. Entre la source des signaux électriques et les électrodes se trouvent des milieux hétérogènes ; de plus, les neurones sont orientés différemment, par conséquent l'EEG ne permet pas de localiser les sources d'activité. Par contre la résolution temporelle de l'EEG est très fine (de l'ordre de la milliseconde). On appelle électroencéphalogramme (EEG) le tracé obtenu par électroencéphalographie.

1.2 VCN

Lorsqu'un stimulus apparaît (par exemple un bruit, ou la préparation à une activité motrice), l'activité électrique cérébrale change brièvement pour donner un pic positif ou négatif sur quelques centaines de millisecondes (les neurones se synchronisent sur l'apparition du stimulus). On appelle ce pic un potentiel évoqué (PE).

Expérimentalement, pour voir le PE, il faut procéder à un moyennage qui est effectué en calculant la moyenne de tous les tracés pour chaque individu. On met donc plusieurs fois chaque individu dans la même situation face au même stimulus, afin d'avoir plusieurs enregistrements de la même 'scène'. En effet, les signaux électriques qui ne sont pas liés au stimulus tendent à s'annuler les uns les autres, n'étant ni synchrones ni du même signe, alors que les signaux liés au stimulus tendent à se sommer. Dans cette expérience, on a 50 enregistrements de la même 'scène' dont est calculée la moyenne. Le moyennage a déjà été effectué sur les signaux que nous traitons avec l'interface graphique ?

Dans cette expérience, nous nous intéressons au potentiel évoqué appelé variation contingente négative (VCN), qui est un grand plateau négatif. Pour la provoquer, il faut deux stimuli : un stimulus avertisseur (dans l'expérience, un

son) qui prévient de l'imminence d'un deuxième stimulus à l'apparition duquel le sujet doit fournir une réponse le plus rapide possible (dans l'expérience, il faut appuyer sur une touche du clavier lorsque la croix à l'écran se transforme en carré). La VCN traduit la façon dont le cerveau se prépare à effectuer la tâche.

La VCN est un marqueur électrophysiologique de la vigilance. Lorsque la vigilance du sujet diminue, l'amplitude de la VCN diminue également (la VCN devient plus positive) et vice versa. L'amplitude de la VCN est proportionnelle à l'état de vigilance de l'individu.

Chapitre 2

ExpÃ©rience vigilance-musique

2.1 But

La recherche d'un critère permettant de déterminer l'état de vigilance d'une personne a donné lieu à une expérience. Le but de cette expérience était d'obtenir un jeu de données réelles afin de rechercher ce critère. Ainsi cette expérience cherche à trouver des marqueurs de l'état de vigilance d'un individu ainsi qu'à savoir si la musique peut avoir une influence sur cet état de vigilance. Une application possible de ces recherches serait de pouvoir détecter une baisse de vigilance afin d'y remédier. Cela serait par exemple très bénéfique pour les chauffeurs routiers chez qui la baisse de vigilance est à l'origine d'accidents.

2.2 Protocol expérimental

2.2.1 Participants

Cette expérience était faite majoritairement avec des étudiants en première année de Licence MIASHS qui ont suivi l'UE 'Bases Biologiques de Comportement', enseignée par M. Frederique Faïta-Ainseba, chercheuse responsable de l'expérience. Pour notre travail, on avait traité des données de 48 sujets différents.

2.2.2 Mesures

Matériel

La première étape de l'expérience est la pose du casque. Le casque utilisé pour cette expérience était un casque 'Electrocap' à 62 électrodes, positionnées sur le scalp selon le système international 10/10. On précise que nous n'étions pas physiquement présents pendant les expériences.

Le logiciel Cohérence 3NT (Deltamed, <http://www.natus.com>) était utilisé dans ces expériences pour le recueil des signaux EEG. Un code MATLAB était développé (de la part de M. Pierrick Legrand) dans le but d'extraire ces signaux EEG et les rendre plus utilisables par un utilisateur. Ainsi, ce code d'extraction est donc une étape intermédiaire entre l'enregistrement des signaux et la visualisation via l'interface graphique.

Déroulement

Après la pose du casque, pendant le reste de l'expérience, le sujet se trouve assis face à un écran d'ordinateur, une main placée de façon à pouvoir presser la barre d'espace sans beaucoup bouger le bras.

Les enregistrements EEG ont lieu trois fois durant l'expérience : au début de l'expérience pour servir de données contrôle, après la première écoute de musique, et après la deuxième écoute de musique. Durant les écoutes de musiques, le sujet n'a pas de consignes particulières.

Durant chaque enregistrement le sujet devait effectuer 50 fois la tâche suivante (qui lui est expliquée par l'expérimentateur) : il fixe une croix affichée sur l'écran d'ordinateur, puis il entend un bip sonore lui annonçant qu'un carré va bientôt apparaître à la place de la croix (il ne doit plus cligner des yeux tant que le carré n'est pas apparu), il presse la barre espace du clavier dès qu'il voit apparaître le carré, puis il fixe de nouveau la croix en attendant le bip sonore.

Le protocole diffère selon les sujets au niveau du contenu des musiques 1 et 2. Une partie des sujets écoutaient en 1 de la musique relaxante, puis en 2 de la musique stimulante. Pour l'autre partie, la musique 1 était stimulante et la musique 2 relaxante.

L'expérience sur la relaxation est suivie le même protocole que l'expérience sur la musique mis à part que l'écoute de musique est remplacée par une séance de relaxation (et il n'y a pas deux mais une séance de relaxation). Le déroulement est donc : enregistrement EEG (contrôle), séance de relaxation, nouvel enregistrement EEG.

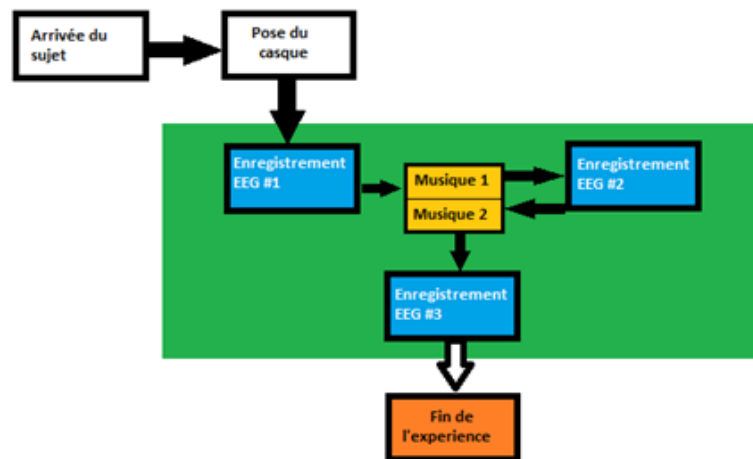


FIGURE 2.1 – Déroulement de l'expérience vigilance-musique

Chapitre 3

Stockage des données de l'expérience

Les matrices contenant les données de l'expérience

Les matrices contenant les données utilisées par l'interface graphique doivent avoir un format particulier.

Format des matrices :

$$\begin{matrix} & 1 & \dots & temps\ max \\ \begin{matrix} 1 \\ \vdots \\ nombre\ sujets \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

NB : ddp =différence de potentiel

$$MAT1 = \begin{matrix} & 1 & \dots & 1024 \\ \begin{matrix} 1 \\ \vdots \\ 48 \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Le nom de la matrice (`nom_matrice`) représente la nature des données qu'elle contient. Par exemple dans l'expérience musique-vigilance nous avons 3 noms : MAT (pour les données contrôle : enregistrements EEG avant toute écoute de musiques), MATS (enregistrement EEG après écoute de la musique stimulante), MATR (enregistrement EEG après écoute de la musique relaxante).

Dans cette expérience le casque contient 62 électrodes. Alors chaque `nom_matrice` donne 62 matrices (une matrice par électrode) : `nom_matrice1`, `nom_matrice2`, ..., `nom_matrice62`. Dans cette expérience on a donc MAT1, MAT2, ..., MAT62, MATS1, MATS2, ..., MATS62, MATR1, MATR2, ..., MATR62. Dans l'explication du code de l'interface Defixe, nous appelons 'sous matrices', les matrices obtenues à partir des `nom_matrice`.

Dans chaque matrice, les lignes correspondent aux numéros d'individus, les colonnes au temps. Pour cette expérience il y a 48 sujets. Et les enregistrements se font sur quatre secondes avec une fréquence de 256 Hertz (256 mesures par secondes), donc sur 4 secondes, $256 \times 4 = 1024$ mesures sont faites.

Pour résumer on a, dans cette expérience, pour chaque électrode (numero_elec) trois matrices MATnumero_elec, MATSnumero_elec et MATRnumero_elec. Chacune de ces matrices contient 48 lignes (sujets) et 1024 colonnes (temps).

Ce format de rangement des données est fait par le prétraitement réalisé par le programme écrit par P. Legrand.

Deuxième partie

Interface Fixe

Chapitre 4

??

Chapitre 5

Utilisation de l'interface Fixe

Nous décrivons l'interface Fixe dans sa version finale, c'est-à-dire réparée/-modifiée. Cette interface a été créée pour visualiser les résultats de l'expérience vigilance-musique.

Nous appelons 'interface 1' la première fenêtre de l'interface. A l'ouverture de cette fenêtre il faut tout d'abord cliquer sur le bouton 'load' et choisir, dans la boîte de dialogue qui s'ouvre, le fichier contenant les données de l'expérience (c'est-à-dire le fichier contenant les matrices dont nous avons décrit le format plus haut).

Les éléments constituant la fenêtre peuvent être séparés en trois groupes. Un premier groupe rassemble les 'entrées' : les informations que l'utilisateur communique au programme pour qu'il fasse le travail souhaité. Un second groupe rassemble les 'sorties' : le résultat du travail du programme. 'Entre' ces deux groupes nous avons :

- un bouton 'run' permettant de lancer le travail du programme,
- un bouton 'Close figure' pour fermer la fenêtre,
- le bouton 'load' que nous venons de mentionner,
- d'autres boutons qui envoient l'utilisateur dans de nouvelles fenêtres que nous décrirons plus bas.

5.1 Les entrées

L'utilisateur choisit, en tapant son numéro dans la case attitrée, l'électrode dont il souhaite visualiser les données.

Il coche des paramètres et options pour déterminer quelles données qu'il souhaite observer : Il choisit un à trois des paramètres suivants :

- Tous : données de tous les sujets (autant de courbes s'afficheront qu'il y a de sujets dans l'expérience),
- Sujet : données d'un sujet, sujet qu'il choisit en inscrivant le numéro dans la case attitrée,
- Moyenne : données moyennées sur tous les sujets.

Il choisit de zéro à trois des options suivantes :

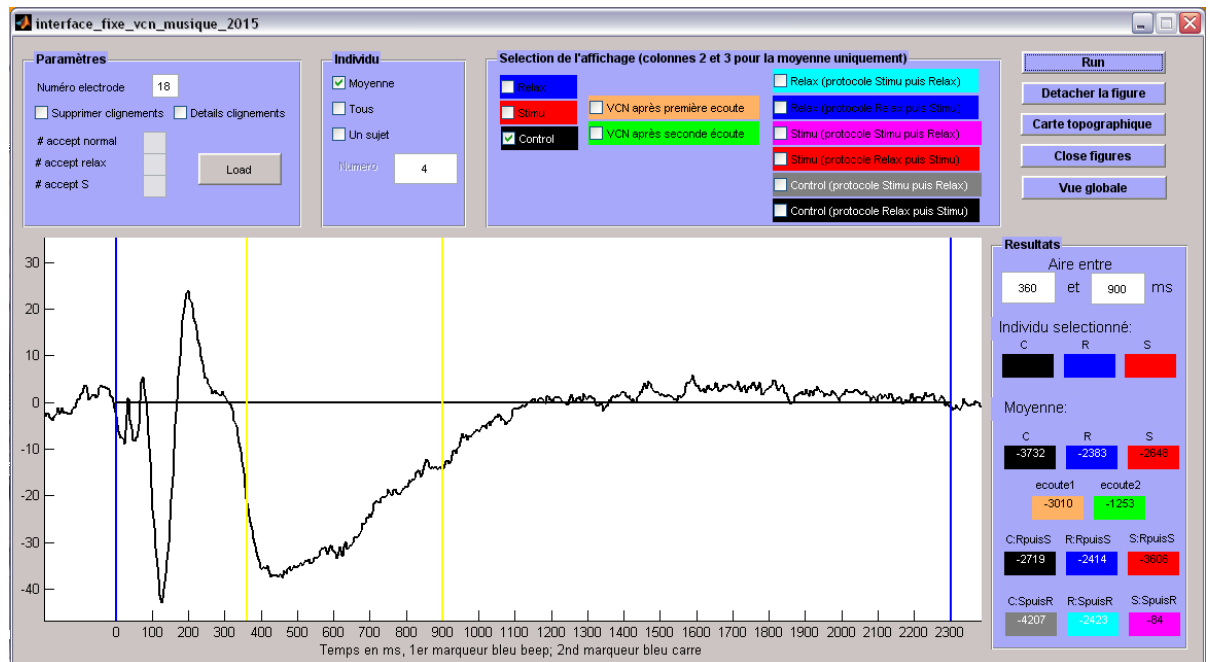


FIGURE 5.1 – Interface Fixe

- Control : données contrôles,
- Relax : données obtenues après écoute de musique relaxante,
- Stimu : données obtenues après écoute de musique stimulante.

Si l'utilisateur a choisi le paramètre Moyenne, alors il peut également choisir de zéro à huit des options suivantes :

- VCN après première écoute : données obtenues après la première écoute de musique,
- VCN après seconde écoute : données obtenues après la deuxième écoute de musique,
- Relax (protocole stimu puis relax) : données obtenues après écoute de musique relaxante en deuxième écoute,
- Relax (protocole relax puis stimu) : données obtenue après écoute de musique relaxante en première écoute,
- Stimu (protocole stimu puis relax) : données obtenues après écoute de musique stimulante en première écoute,
- Stimu (protocole relax puis stimu) : données obtenue après écoute de musique stimulante en deuxième écoute,
- Control (protocole stimu puis relax) : données obtenues avant toute écoute de musique lorsque le sujet a ensuite écouté de la musique stimulante suivie de musique relaxante,
- Control (protocole relax puis stimu) : données obtenues avant toute écoute de musique lorsque le sujet a ensuite écouté de la musique relaxante suivie de musique stimulante.

En cochant 'supprimer clignements', il choisit d'afficher les données pour lesquelles les moyennages des 50 'scènes' n'ont été réalisés qu'à partir des enregistrements durant lesquels le sujet n'a pas cligné des yeux. Rappel : nous appelons 'scène', la suite d'événements 'bip sonore-carré-appui sur clavier'. Cependant, pour cette expérience, cocher cette option semble rendre le signal EEG moins clair, sûrement à cause du fait que le moyennage est réalisé sur un plus petit nombre de 'scènes'. Il choisit sur quel intervalle de temps le calcul de l'aire sous la courbe sera effectué, en entrant les bornes de cet intervalle en millisecondes.

Le paramètre 'détails clignements' n'est pas utilisable.

5.2 Les sorties

Les sorties sont une ou plusieurs courbes représentant la différence de potentiel mesurée au cours du temps pour les données choisies par l'utilisateur (en cochant les paramètres et options que nous venons de décrire plus haut). Pour chacune de ces courbes, excepté celles affichées en cochant le paramètre 'tous', l'aire sous la courbe est calculée et affichée. La couleur et la forme du trait des courbes différencient les données affichées. Les couleurs des cases d'affichage des aires sont en adéquation avec les couleurs des options, et elles sont légendées avec les abréviations suivantes :

- C, R, S pour control, relax et stimu
- Ecoule1 pour VCN après première écoute
- Ecoule2 pour VCN après seconde écoute
- RpuisS pour protocole Relax puis Stimu
- SpuisR pour protocole Stimu puis Relax

Par ailleurs, si l'utilisateur a coché 'supprimer clignements', alors le nombre de 'scènes' conservées est indiqué pour chacun des trois types de données : contrôle, post-musique stimulante et post-musique relaxante.

5.3 Les liens vers d'autres fenêtres

Des liens vers trois autres fenêtres se font via trois boutons : 'Détacher la figure', 'Carte topographique' et 'Vue globale'.

5.3.1 Détacher la figure

Le bouton 'Détacher la figure' nous envoie vers une fenêtre qui affiche exactement le même graphe que celui affiché dans l'interface1 au moment où l'utilisateur clique sur le bouton. La fenêtre est munie de barres d'outils permettant à l'utilisateur de réaliser de nombreuses manipulations sur le graphique. Par exemple, il peut zoomer, ajouter une légende, ou encore modifier les tracés. Cela permettra notamment à l'utilisateur, en modifiant les tracés, d'avoir un graphe en noir et blanc.

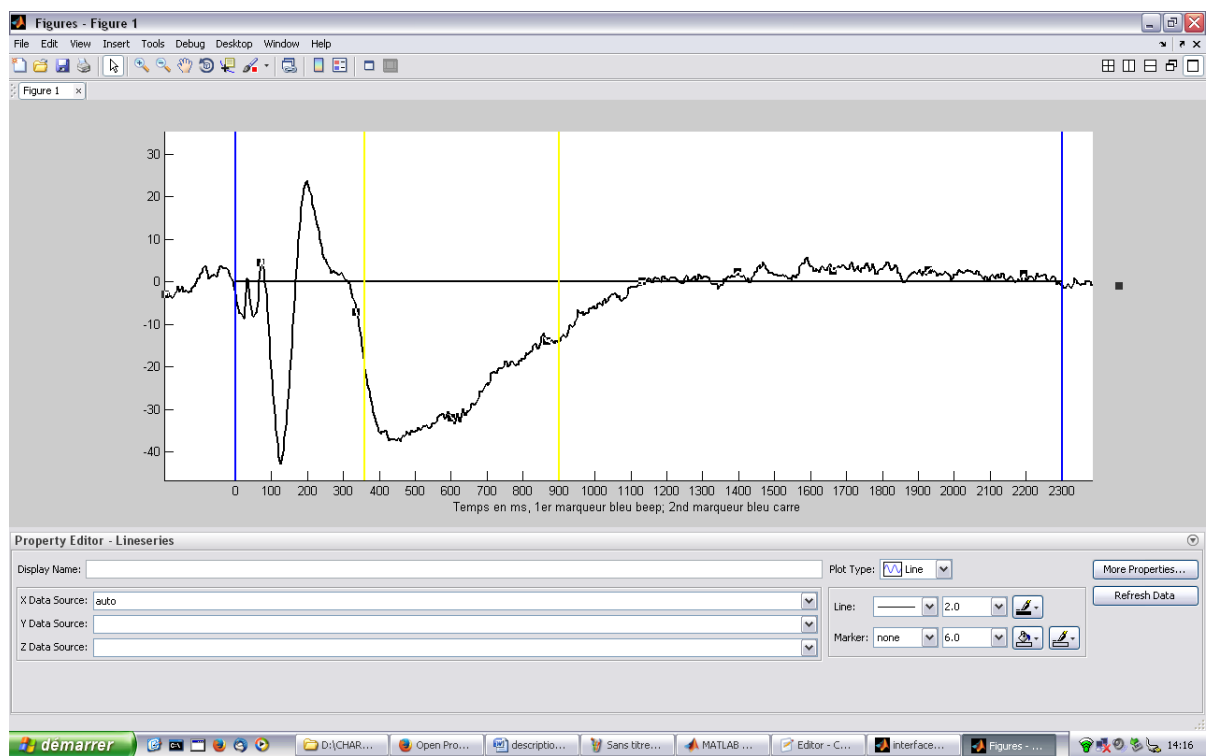


FIGURE 5.2 – Détacher la figure Fixe

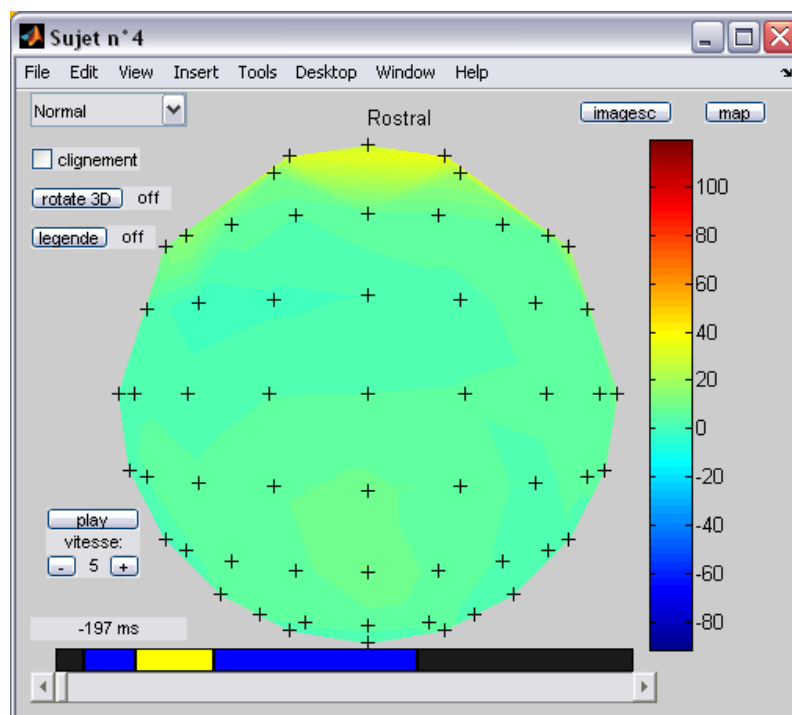


FIGURE 5.3 – Carte topographique Fixe

5.3.2 Carte topographique

La fenêtre ouverte par le bouton 'Carte topographique' contient les mêmes trois groupes (entrées, sorties, liens vers d'autres fenêtres) que la fenêtre interface1, à la différence qu'il n'y a pas de bouton 'run' pour lancer le travail du programme étant donné que la seule modification d'une entrée changera directement l'affichage en sortie.

La sortie est une carte de la surface du crâne où les électrodes sont représentées par des croix et où la couleur représente la différence de potentiel observée à la surface du crâne. Pour comprendre le code de couleur une barre de couleur légendée est située à droite de la carte. La carte est accompagnée d'une zone qui donne le temps (en millisecondes) auquel correspond l'affichage.

Les entrées sont d'une part un menu déroulant permettant à l'utilisateur de choisir les données qu'il veut voir : avant toute écoute (option 'control'), après écoute de musique relaxante (option 'relax'), après écoute de musique stimulante (option 'stimu'), la différence entre les données obtenues après écoute de musique relaxante et les données contrôles ('relax-control'), ou la différence entre les données obtenues après écoute de musique stimulante et les données contrôles ('stimu-control'). D'autre part les entrées consistent à choisir, comme pour la fenêtre principale, si on souhaite les données obtenues sans tenir compte des enregistrements dans lesquels le sujet a cligné des yeux, ou en en tenant compte. Dans le premier cas il ne faut pas cocher 'clignement', dans le second il faut cocher. L'utilisateur choisit à quel moment situer l'affichage de la carte en bougeant le slider. Il peut également voir comme un film l'évolution de la carte au cours du temps au moyen d'un bouton 'play/stop' et de boutons '+' et '-' changeant la vitesse du film. L'utilisateur peut voir la carte en trois dimensions en cliquant sur le bouton 'rotate 3D' pour activer la vue en 3D et en changeant l'angle de vue de la carte grâce à la souris (cliquer sur la carte, rester appuyé et faire glisser la souris). Il peut également afficher d'avantage de légende pour mieux visualiser l'orientation de la carte dans l'espace.

Les boutons 'imagesc' et 'map' renvoient vers de nouvelles fenêtres.

'Imagesc' ouvre deux fenêtres qui affichent la même information : la différence de potentiel en fonction de l'électrode et du temps. Cette différence de potentiel est codée par la couleur, comme sur la carte topographique. Une des deux fenêtres affiche la différence de potentiel non seulement par le code couleur, mais aussi par une coordonnée sur un troisième axe, l'affichage se fait donc en trois dimensions. Pour ce dernier affichage il y a possibilité de voir le graphique en 3D et changer d'angle de vue, comme précédemment avec le bouton 'rotate 3D' et la souris.

'Map' ouvre une fenêtre affichant pour 15 électrodes des miniatures des graphiques qui auraient été affichés dans la fenêtre interface1. Ces graphiques sont situés aux emplacements des électrodes correspondantes sur un cercle représentant la surface du crâne. Les données affichées dans ces graphiques sont celles choisies dans le menu de la carte topographique.

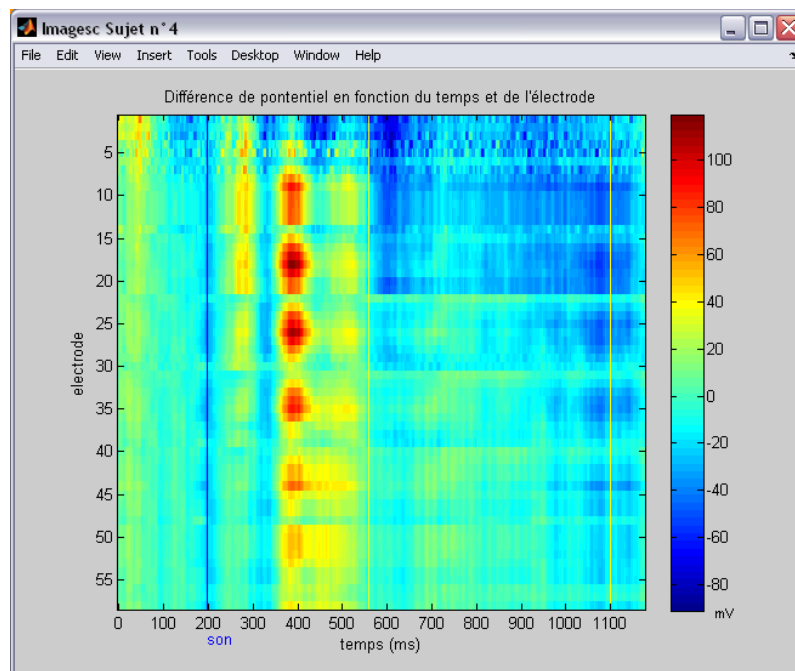


FIGURE 5.4 – Imagesc Fixe : fenêtre 1

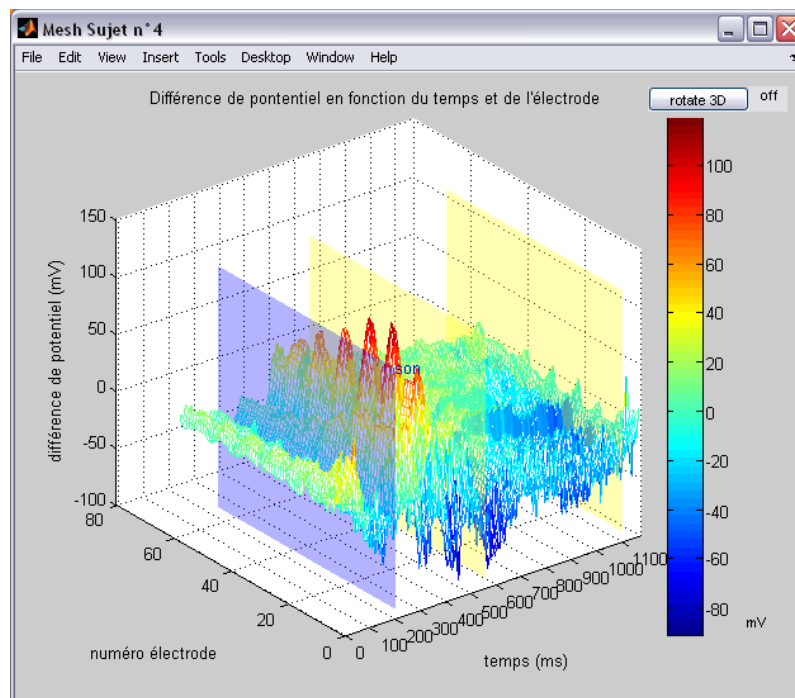


FIGURE 5.5 – Imagesc Fixe : fenêtre 2

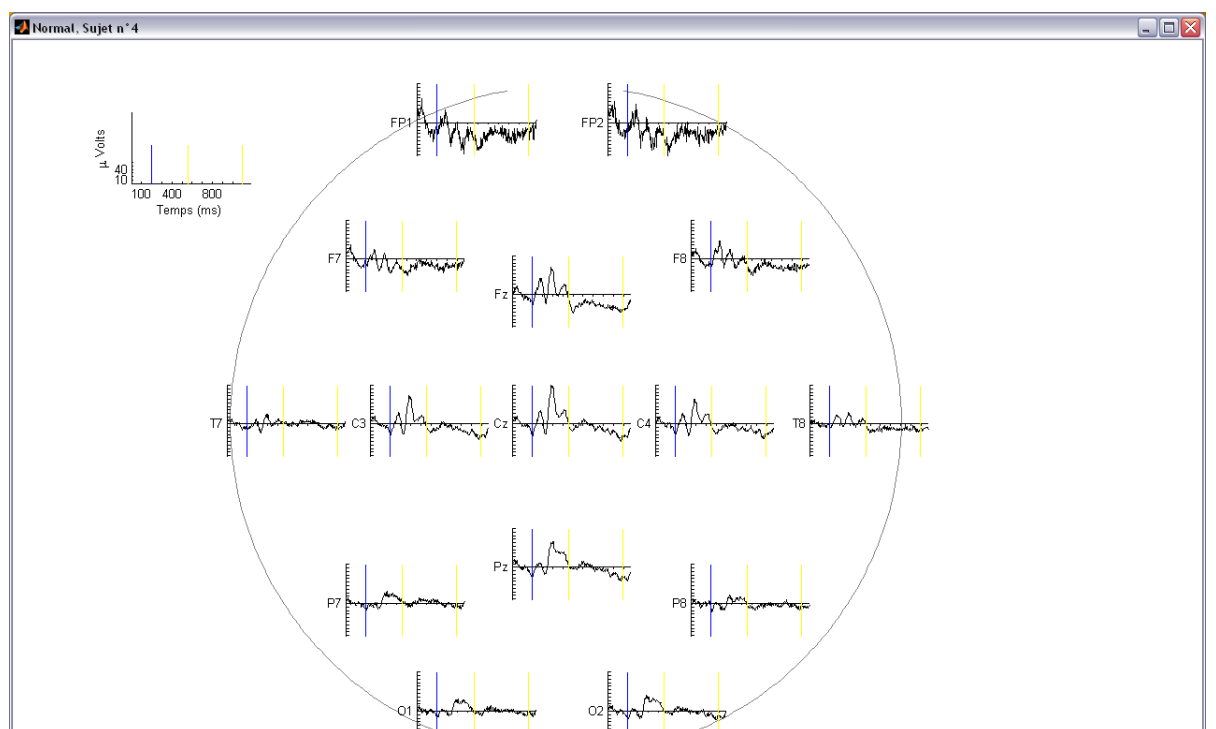


FIGURE 5.6 – Maps Fixe

5.3.3 Vue globale

'Vue globale' permet d'afficher le calcul d'aire sous la courbe entre les deux intervalles de temps choisis par l'utilisateur dans la fenêtre principale. Cette donnée peut être affichée de quatre façons différentes selon la case cochée par l'utilisateur :

- l'aire en fonction du numéro de sujet
 - pour l'électrode sélectionnée dans la fenêtre interface1 ;
 - moyennée sur toutes les électrodes ;
- l'aire en fonction du numéro d'électrode
 - pour le sujet sélectionné dans la fenêtre interface1 ;
 - moyennée sur tous les sujets.

Comme dans l'interface1, un bouton 'détacher figure' ouvre une nouvelle fenêtre contenant le même graphique que celui affiché au moment de l'appui sur le bouton, et cette fenêtre est munie d'outils permettant de modifier le graphique.

Troisième partie

Interface Defixe

Chapitre 6

Utilisation de l'interface Defixe

Nous décrirons l'interface Defixe. Cette interface permet la visualisation des données d'une expérience effectuant des enregistrements EEG.

6.1 Interface0

La première fenêtre que rencontre l'utilisateur, interface0, recueille des informations sur l'expérience et sur la façon dont sont organisées les données EEG dans les matrices.

6.1.1 Utilisateur n'ayant jamais utilisé l'interface Defixe

S'il n'a jamais utilisé l'interface Defixe, l'utilisateur doit renseigner des informations à partir desquelles sera construite une interface similaire à l'interface Fixe. On a séparé ces informations en quatre catégories : options, stimuli, électrodes et fréquence d'enregistrement.

Options

L'utilisateur donne des informations sur les options qu'il souhaite visualiser. Nous appelons 'options' les différentes natures de données qui seront affichables dans l'interface. Dans l'interface Fixe, ces options correspondent par exemple à 'control', 'relax', 'relax(protocol Stimu puis Relax)', ou encore 'VCN après première écoute'. Il faut entrer le nom de l'option qui est celui qui apparaîtra dans l'interface (il doit donc être parlant et unique). Il faut également écrire où se trouvent les données que concerne l'option. Pour cela nous demandons le(s) nom_matrice de la/des matrice(s) qui contien(nen)t les données. A chaque nom_matrice doit être associé le nom d'un fichier Excel (sans l'extension) contenant dans la première ligne les numéros de sujets (donc de lignes) à prendre en compte pour le nom_matrice, à raison de un numéro par colonne. Enfin, il faut dire si cette option concerne tous les sujets de l'expérience ou seulement un sous-ensemble des sujets, c'est-à-dire 'est ce que pour remplir les données de cette option on utilise les données de tous les sujets ou bien seulement les données



FIGURE 6.1 – Interface 0



FIGURE 6.2 – Interface 1 Fixe : options

d'un sous-ensemble des sujets ?'. Par exemple, l'option 'control' concerne tous les sujets. On a enregistré les données EEG de chaque sujet avant toute écoute de musique et on utilise toutes ces données pour cette option. Par contre l'option 'relax(protocol Stimu puis Relax)' contient les données obtenues après écoute de musique relaxante lorsque la musique relaxante a été présentée au sujet en deuxième écoute. Donc elle ne concerne pas tous les sujets. Seule une sous-partie des sujets a écouté la musique relaxante en second. Pour choisir, l'utilisateur coche soit la case tous, soit la case select. L'intérêt de donner cette information n'est pas visible immédiatement pour l'utilisateur mais c'est une donnée nécessaire pour le programme. Une conséquence visible est que les options marquées 'tous' pourront être observées pour un sujet et pour la moyenne des sujets alors que les options marquée 'select' ne seront observables que moyennées sur tous les sujets concernés par l'option (ces sujets sont ceux dont les numéros sont listés dans les fichiers Excels associés aux nom__matrices de l'option).

Stimuli

L'interface permet de visualiser, sur les différents affichages (graphiques, carte topographique), des repères temporels, par exemple pour l'expérience sur la musique, le moment d'apparition du son et la transformation de la croix en carré. Nous appelons ces repères temporels des stimuli. Pour chaque stimulus, l'utilisateur doit donner un nom (qui servira de légende au stimulus dans l'interface), ainsi que le moment de son apparition en millisecondes.

Electrodes

L'utilisateur doit donner le nombre d'électrodes que possède le casque EEG utilisé pour l'expérience. Il doit également fournir la liste des numéros des électrodes à prendre en compte pour l'exploitation des données. En effet on n'exploite pas forcément les données de toutes les électrodes. Dans l'expérience musique -vigilance par exemple, il n'y a pas d'intérêt à afficher les données recueillies par les quatre électrodes situées autour des yeux et derrière les oreilles (à des fins de contrôles et de calculs). La liste d'électrodes doit être fournie en donnant le nom d'un fichier Excel (sans extension). Ce fichier doit contenir sur la première ligne la liste des numéros d'électrodes, à raison de un numéro par colonne.

Fréquence d'enregistrement

Il est demandé de donner la fréquence, en Hertz, d'enregistrement du dispositif qui a recueilli les données EEG.

Ainsi, l'utilisateur doit placer, aux côtés du programme, les fichiers Excel dont il a donné les noms au cours de ces manipulations.

Une fois toutes les informations remplies et les fichiers Excel placés aux côtés du programme, l'utilisateur clique sur le bouton 'charger et sauvegarder ces données'. S'ouvre alors une boîte de dialogue permettant de choisir le nom et l'emplacement du fichier de sauvegarde. Ce fichier portera l'extension '.mat'.

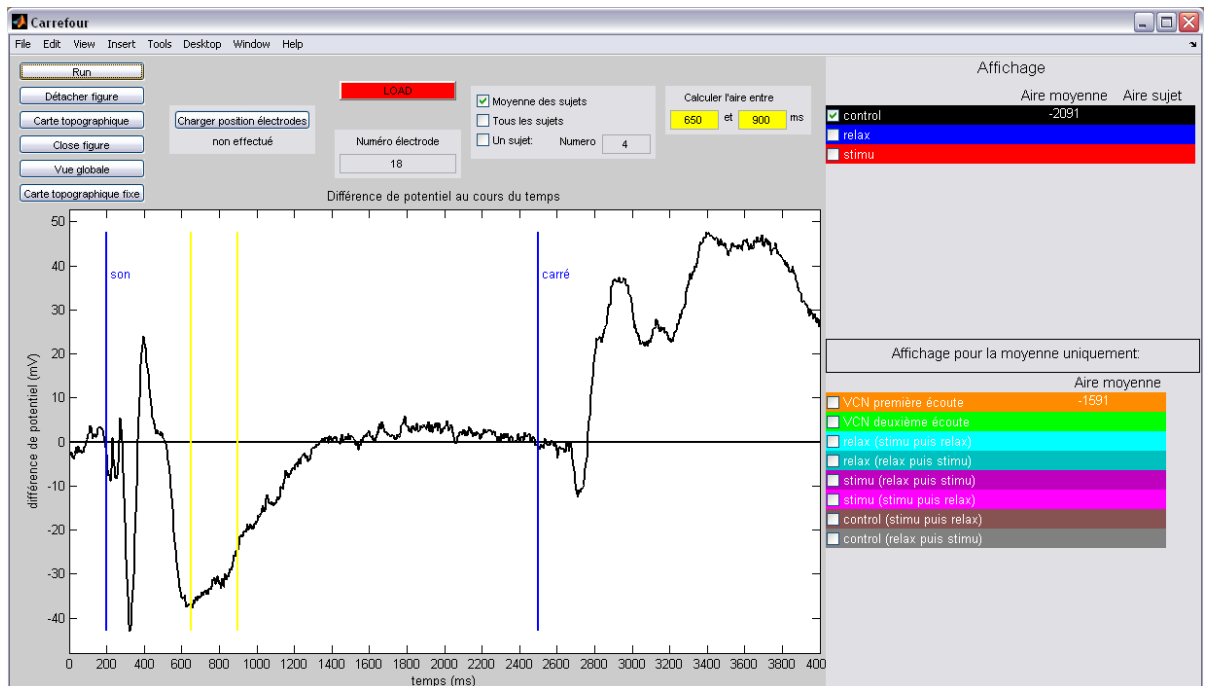


FIGURE 6.3 – Interface 1 Defixe

6.1.2 Utilisateur ayant déjà utilisé l'interface Defixe

Si l'utilisateur avait déjà effectué la procédure de renseignement des informations sur l'expérience, alors il les a normalement enregistrées dans un fichier .mat. Il lui suffit de cliquer sur le bouton 'charger anciennes données'. Une boîte de dialogue est ouverte. Elle permet de sélectionner le fichier .mat en question.

Si l'utilisateur n'avait pas enregistré ce fichier .mat, il doit reprendre la méthode décrite dans le paragraphe précédent.

6.2 Interface1

Nous arrivons sur une interface similaire à l'interface Fixe, que nous appelons 'interface 1'.

La première manipulation à effectuer quand la fenêtre s'ouvre est de cliquer sur le bouton 'load'. Une boîte de dialogue est ouverte, via laquelle il faut sélectionner le fichier contenant les données. Ce fichier est le .mat contenant

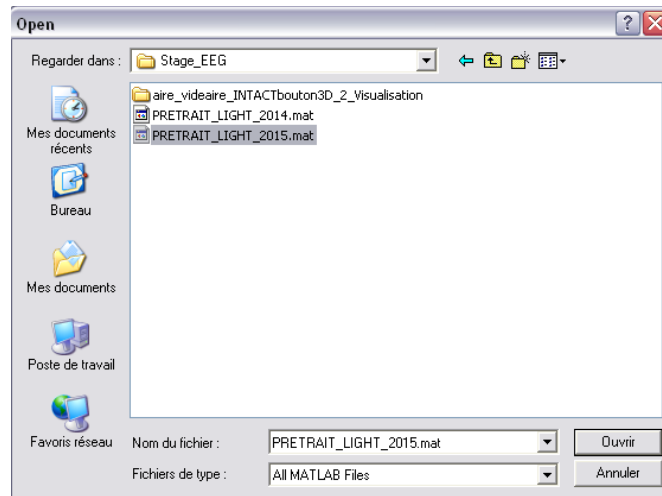


FIGURE 6.4 – Boîte de dialogue load Defixe

les matrices 'nom_matricenuméro_electrode' (par exemple, pour l'expérience musique vigilance-musique, ce sont les matrices MAT1, MAT2, etc.).

Comme pour l'interface Fixe, on peut regrouper les éléments de cette fenêtre interfacel en trois parties : les entrées, les sorties et les liens vers d'autres fenêtres.

6.2.1 Les entrées

L'utilisateur choisi l'électrode et le sujet dont il va exploiter les données.

Il peut choisir un intervalle de temps sur lequel le programme calculera l'aire sous la courbe. Pour que cet intervalle de temps soit valide, il faut évidemment que la première valeur soit inférieure à la deuxième, mais il faut aussi que les deux valeurs temporelles appartiennent à l'intervalle de temps sur lequel ont été réalisées les mesures (dans l'expérience vigilance-musique, elles doivent appartenir aux 4 secondes affichées dans le graphique). Cette dernière contrainte peut poser problème au tout début et à la toute fin de l'enregistrement des données. Par exemple pour l'expérience vigilance-musique, le programme, du fait des conversions et du procédé d'enregistrement des données, ne permet de calculer l'aire qu'à partir de 4 millisecondes.

L'utilisateur choisit quelles sont les options qu'il souhaite observer. Ces options sont celles qu'il a créées dans la fenêtre interface0.

Pour les options qu'il a cochées, il choisit s'il souhaite les visualiser moyennées sur tous les sujets, s'il souhaite les voir pour chaque sujet, ou s'il souhaite les voir pour un unique sujet (le sujet dont il aura donné le numéro). La seule contrainte dans ce choix est qu'une option qui ne concerne pas tous les sujets ne pourra pas

être visualisées en cochant 'voir pour tous les sujet', ni en cochant 'voir pour un seul sujet'. Elle ne sera visible que moyennée sur tous les sujets concernés par l'option. Ce cas concerne donc les options pour lesquelles l'utilisateur a coché la case 'select' dans l'interface0.

6.2.2 Les sorties

Les sorties sont de deux types : un graphique présentant la différence de potentiel au cours du temps mesurée par l'électrode sélectionnée pour les données choisies par l'utilisateur ; et des calculs d'aire sous la courbe. Les calculs d'aires sous la courbe sont réalisés pour chaque courbe qui est affichée dans l'axe, à l'exception des courbes qui ont été affichées en cochant la case permettant de voir les options 'pour tous les sujets'. Dans le graphique, on visualise l'intervalle de temps sur lequel l'aire est calculée grâce à deux lignes verticales jaunes.

6.2.3 Les liens vers d'autres fenêtres

Chacun des boutons suivants provoque l'ouverture d'une fenêtre supplémentaire : 'détacher figure', 'vue globale', 'carte topographique' et 'carte topographique fixe'.

Détacher figure

La fenêtre ouverte par le bouton 'détacher figure' contient le même graphique que celui affiché dans l'interface1. A cela s'ajoute, aux côtés du graphique, des barres d'outils permettant de modifier les courbes (couleur, trait, etc.) en cliquant avant sur la courbe à modifier, d'ajouter une légende, de zoomer, etc. Cela permettra notamment à l'utilisateur, en modifiant les courbes, d'avoir un graphe en noir et blanc.

Vue globale

'Vue globale' permet d'afficher le calcul d'aire sous la courbe entre les deux intervalles de temps choisis par l'utilisateur dans la fenêtre principale. Cette fenêtre ne sera donc utilisable que si l'utilisateur a choisi un intervalle de temps valide pour le calcul de l'aire. L'aire sous la courbe peut être affichée de quatre façons différentes selon la case cochée par l'utilisateur :

- l'aire en fonction du numéro de sujet
 - pour l'électrode sélectionnée dans la fenêtre interface1 ;
 - moyennée sur toutes les électrodes ;
- l'aire ne fonction du numéro d'électrode
 - pour le sujet sélectionné dans la fenêtre interface1 ;
 - moyennée sur tous les sujets.

Comme dans l'interface1, un bouton 'détacher figure' ouvre une nouvelle fenêtre contenant le même graphique que celui affiché au moment de l'appui sur le bouton, et cette fenêtre est munie d'outils permettant de modifier le graphique.

Carte topographique

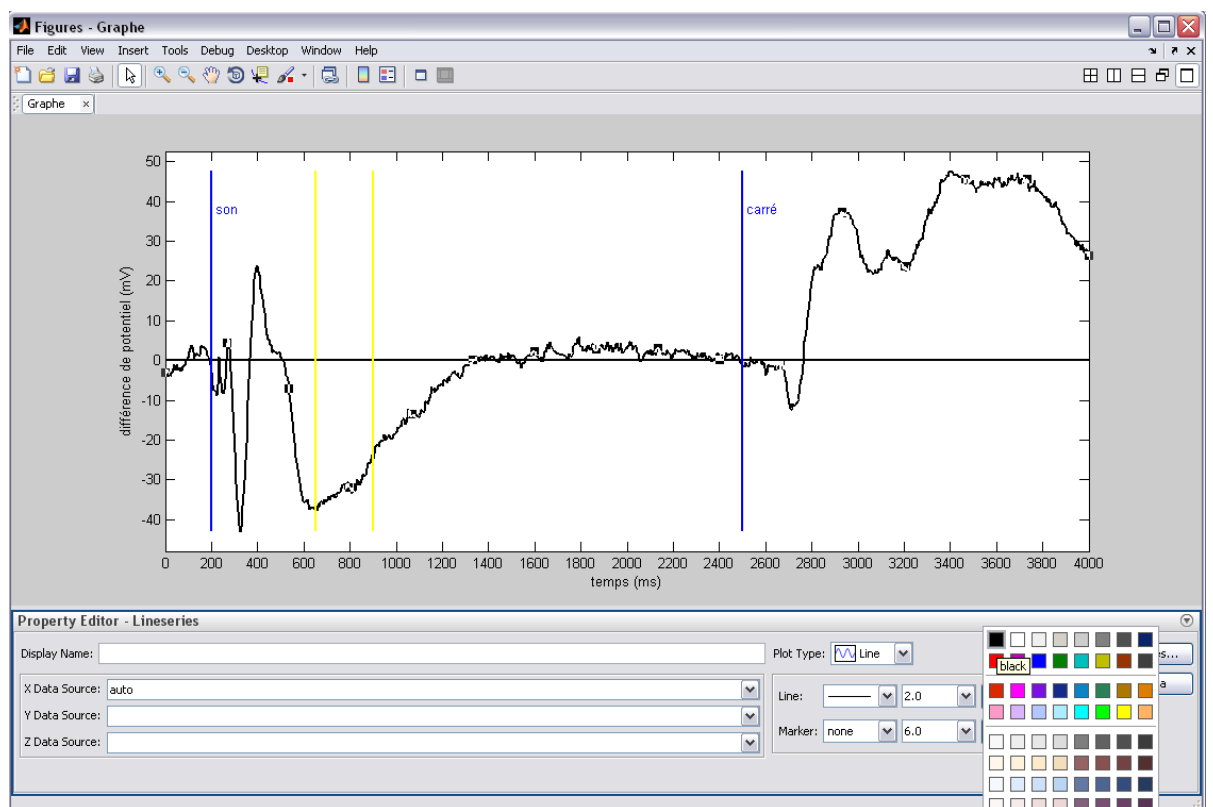


FIGURE 6.5 – Détacher la figure Defixe

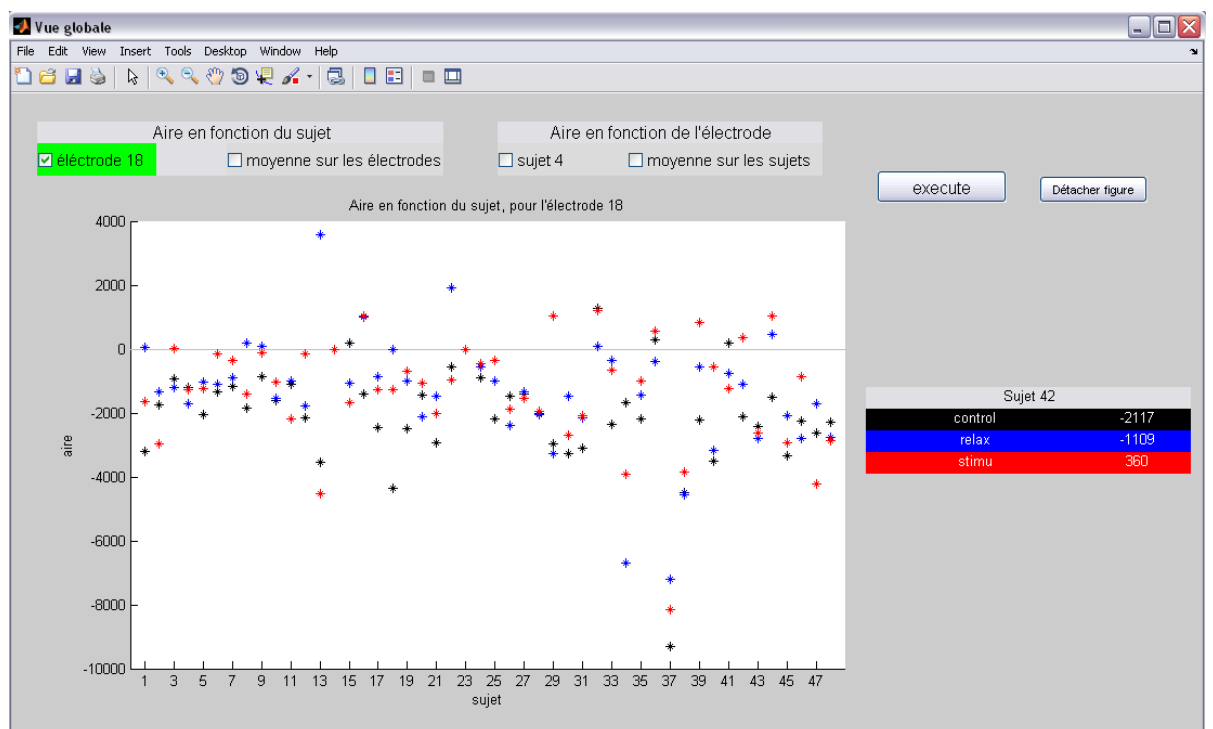


FIGURE 6.6 – Vue globale Defixe : Aire sous la courbe en fonction du sujet

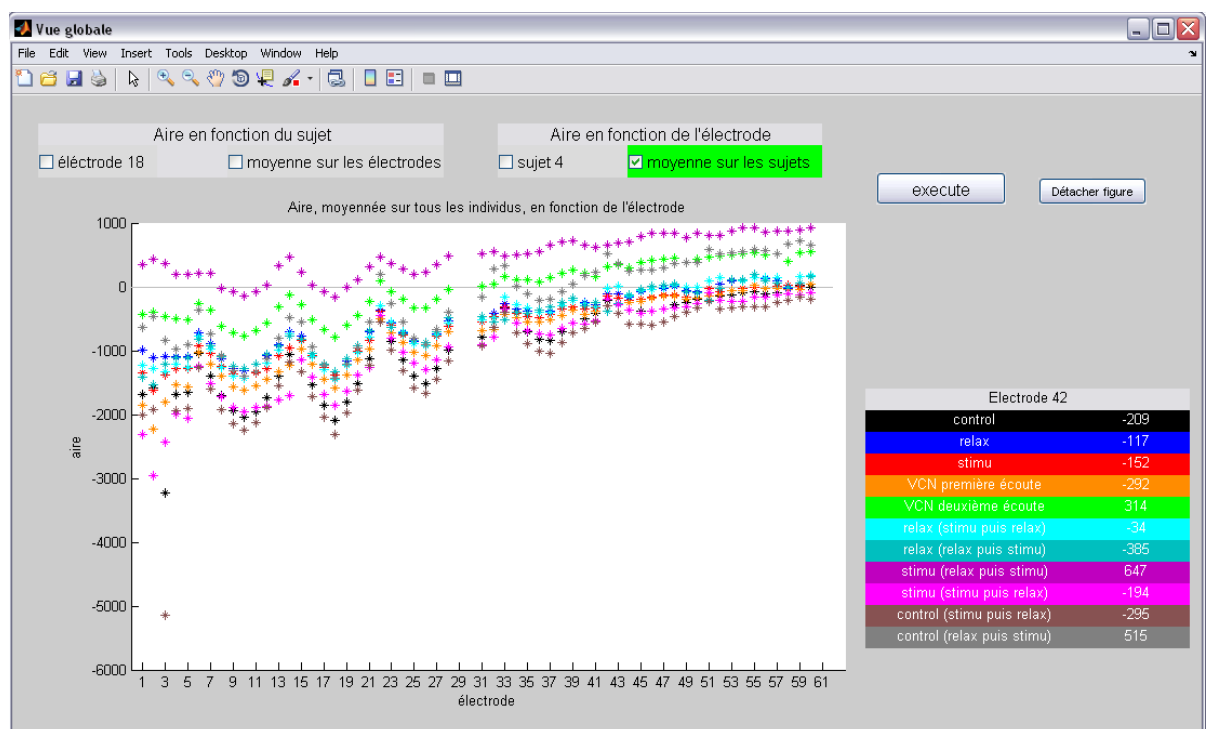


FIGURE 6.7 – Vue globale Defixe : Aire sous la courbe en fonction de l'électrode

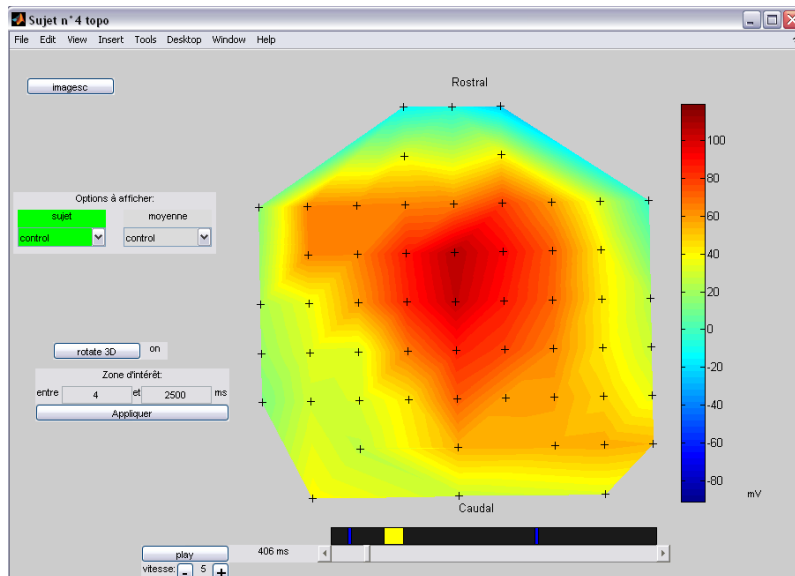


FIGURE 6.8 – Carte topographique Defixe

Les boutons 'carte topographique' et 'carte topographique fixe' ouvrent tous deux la même fenêtre qui affiche une carte de la surface du crâne sur laquelle les électrodes sont représentées par des croix. Les couleurs de la carte représentent la différence de potentiel mesurée. On l'appelle carte topographique. Pour pouvoir afficher cette carte il est nécessaire pour le programme d'avoir comme données les positions des électrodes (coordonnées (x,y,z) de chaque électrodes).

Le bouton 'carte topographique fixe' utilise les coordonnées des électrodes du casque utilisé lors de l'expérience vigilance-musique.

Le bouton 'carte topographique' permet de voir la carte pour n'importe quel type de casque. Pour pouvoir l'utiliser, l'utilisateur doit d'abord avoir donner des renseignements sur le casque EEG. Pour faire cela il faut cliquer sur le bouton 'charger position électrodes'. Nous expliquerons plus bas la démarche à suivre une fois que l'on a cliqué sur ce bouton. Un défaut du programme est qu'il ne fait pas un travail satisfaisant pour calculer la position des électrodes à partir des données fournies par l'utilisateur sur le casque EEG. La visualisation est donc pour l'instant meilleure via le renseignement direct des coordonnées (x,y,z) des électrodes. Par conséquent, pour l'expérience vigilance-musique, nous laissons la possibilité d'utiliser ces positions via le bouton 'carte topographique fixe'. Lorsque le programme sera à même de calculer les coordonnées, ce bouton sera retiré de l'interface.

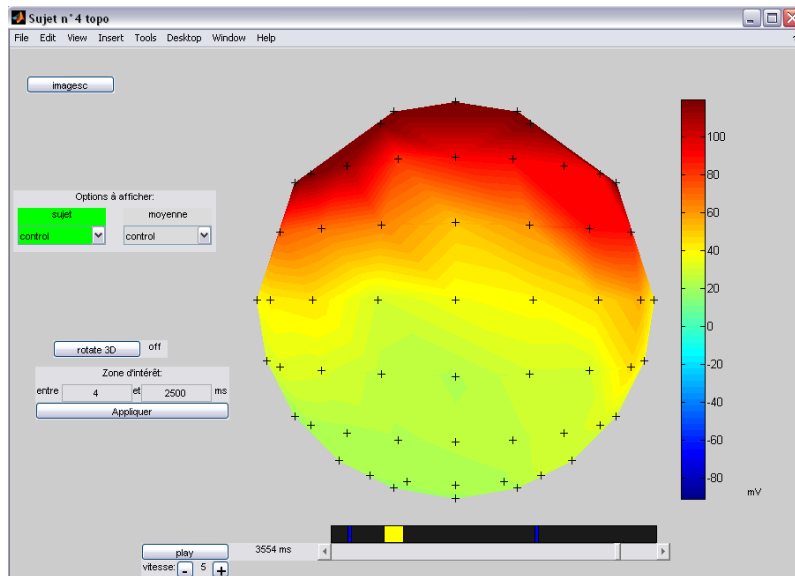


FIGURE 6.9 – Carte topographique Defixe pour l'expérience vigilance-musique

Décrivons maintenant la fenêtre ouverte par les deux boutons 'carte topographique' et 'cartes topographique fixe'. Elle contient les mêmes trois groupes que l'interface1 (entrées, sorties et liens vers d'autres fenêtres), à la différence qu'il n'y a pas de bouton 'run' pour lancer le travail du programme étant donné que la seule modification d'une entrée changera directement l'affichage en sortie.

La sortie est la carte topographique. Comme on l'a dit, les couleurs représentent la différence de potentiel. Pour comprendre le code de couleur, une barre de couleur légendée est située à droite de la carte. Le temps correspondant à l'affichage de la carte est donné à gauche de la carte.

Les entrées permettent de jouer sur l'affichage. Deux menus déroulants permettent de choisir l'option à visualiser sur la carte : un menu contient les options à voir pour un seul sujet (choisi dans l'interface1), un menu pour contient les options à voir moyennées sur les sujets. Comme dans l'interface1, seule une option concernant tous les sujets est visualisable pour un seul sujet. Les options disponibles dans les menus sont celles qui ont été cochées dans l'interface1. Pour résumer, les courbes affichées dans l'interface1 en cochant sujet moyenne, sont visualisables sous forme de carte topographique. Un slider permet à l'utilisateur de choisir à quel moment situer l'affichage de la carte. Il peut également voir, comme un film, l'évolution de la carte au cours du temps grâce à un bouton 'play/stop' et aux boutons '+' et '-' qui changent la vitesse du film. Cliquer sur le bouton 'rotate 3D' affiche la carte en trois dimensions et alors avec la souris il est possible de changer l'angle de vue de la carte (cliquer sur la carte, rester appuyé et faire glisser la souris). Enfin, l'utilisateur peut choisir de changer la zone sur laquelle il souhaite faire l'échelle de couleur afin de mieux visualiser les variations de la différence de potentiel sur un intervalle de temps qui l'intéresse. Pour cela,

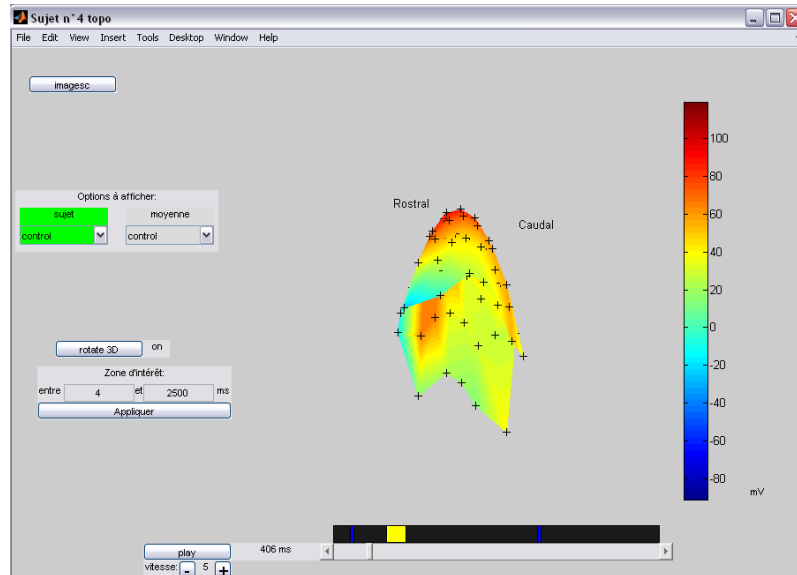


FIGURE 6.10 – Vue 3D Carte topographique Defixe

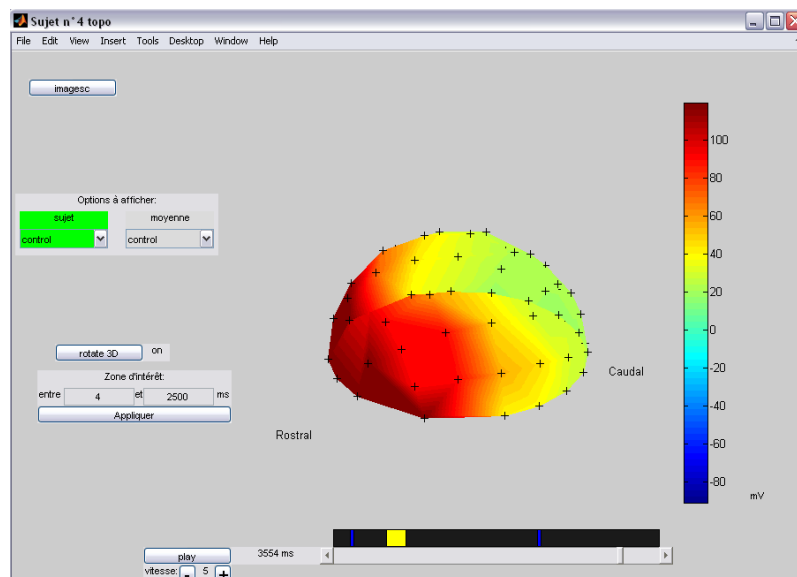


FIGURE 6.11 – Vue 3D Carte topographique Defixe pour l'expérience vigilance-musique

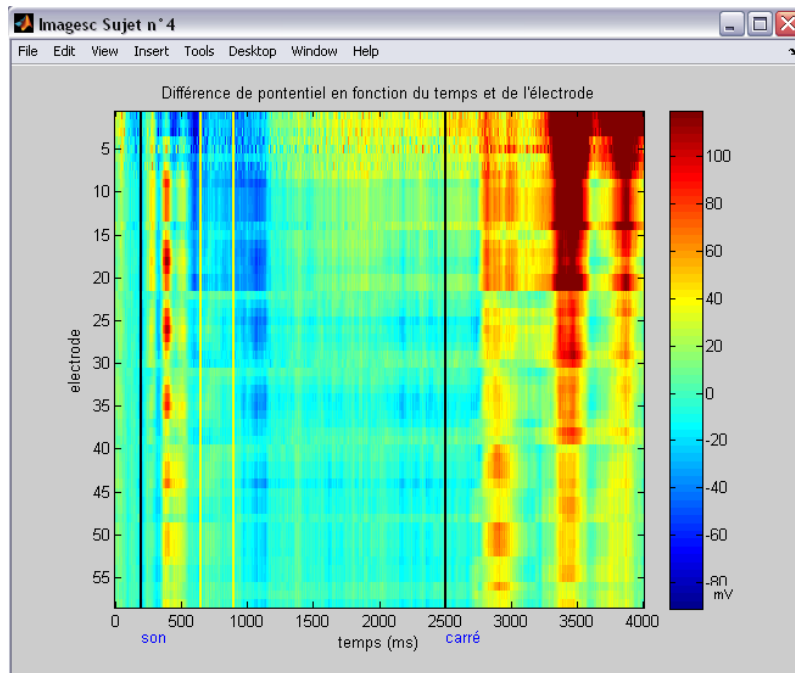


FIGURE 6.12 – Imagesc Defixe : fenêtre 1

il lui faut modifier ce que nous appelons la 'zone d'intérêt' en inscrivant (en millisecondes) les bornes de l'intervalle de temps qui l'intéresse, puis en cliquant sur le bouton 'appliquer'.

Le bouton 'imagesc' renvoie vers deux nouvelles fenêtres. Celles-ci affichent la même information : la différence de potentiel en fonction de l'électrode et du temps. Cette différence de potentiel est codée comme précédemment par la couleur. Une des deux fenêtres affiche la différence de potentiel non seulement par le code couleur, mais aussi par l'ajout d'une coordonnée sur un troisième axe, l'affichage se fait donc en trois dimensions. Pour ce dernier affichage il y a possibilité de voir le graphique en 3D et changer d'angle de vue, avec le bouton 'rotate 3D' et la souris. L'échelle appliquée dans la fenêtre précédente (contenant la carte topographique) est également appliquée dans les deux fenêtres que nous venons de décrire.

Charger position électrodes

La fenêtre ouverte en cliquant sur le bouton 'charger position électrodes' contient une représentation schématique du casque EEG sous forme d'un tableau. La taille de ce schémas est proportionnelle au nombre d'électrodes que contient le casque. L'utilisateur, s'il fait cette démarche pour la première fois, doit placer les électrodes sur le schéma en inscrivant leurs numéros dans les

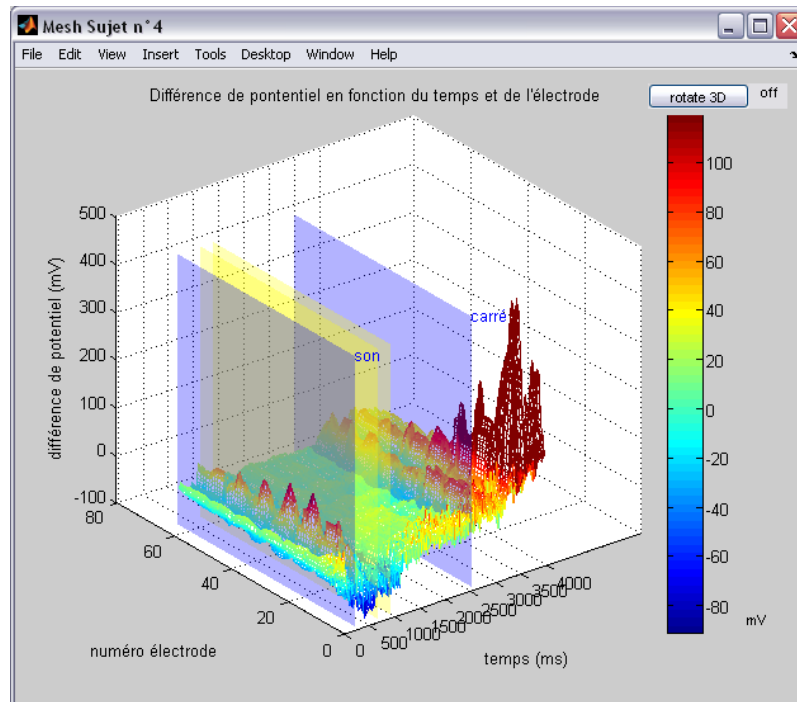


FIGURE 6.13 – Imagesc Defixe : fenêtre 2

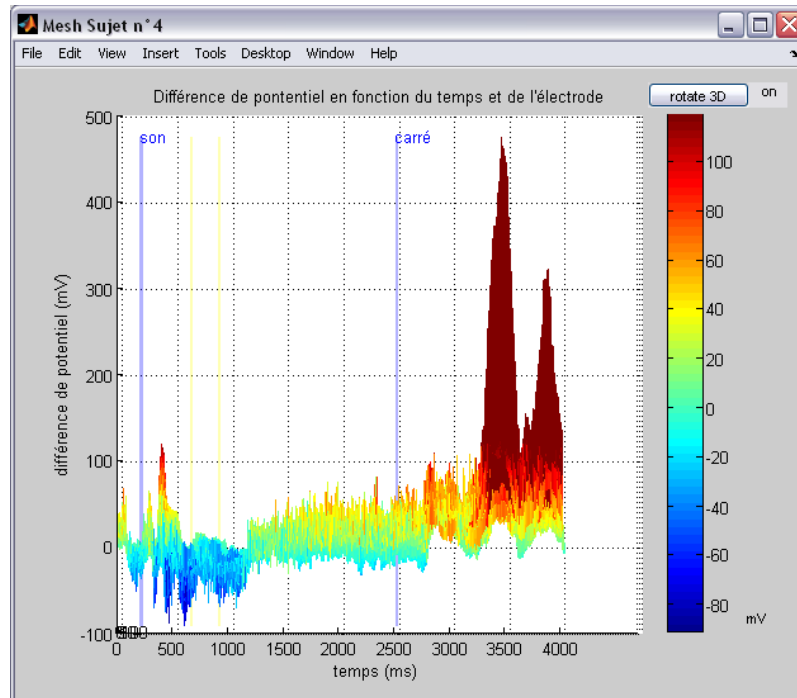


FIGURE 6.14 – Rotation 3D Imagesc Defixe : fenêtre 2

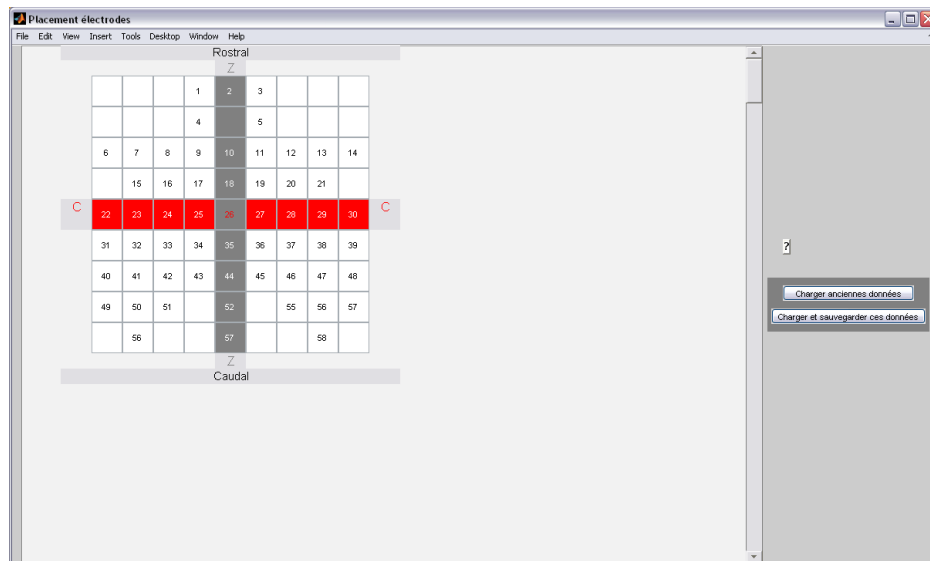


FIGURE 6.15 – Renseignement positions électrodes

cases du tableau. Une fois cela fait il lui reste à cliquer sur le bouton 'charger et sauvegarder ces données' et il a alors la possibilité d'enregistrer les informations qu'il vient de donner dans un fichier .mat dont il choisit le nom et l'emplacement via la boîte de dialogue qui s'ouvre à l'appui sur le bouton. Si cette démarche avait déjà été réalisée dans une utilisation précédente de l'interface, l'utilisateur clique directement sur le bouton 'charger anciennes données' et choisit le fichier .mat contenant les données via la boîte de dialogue qui apparaît.

Une fois la position des électrodes chargée correctement, il est possible d'utiliser le bouton 'carte topographique'.

Chapitre 7

Code de l'interface Defixe

Nous expliquerons le code de l'interface de façon à ce qu'une personne externe ayant des bases en Matlab puisse rapidement comprendre le fonctionnement du programme.

Chaque fonction du programme a été écrite dans un fichier portant le nom de la fonction. Un schéma résume l'architecture de ces fichiers/fonctions. Dans la description du code, nous suivrons ce schéma en allant de fichier en fichier, donc en expliquant le code fonction par fonction.

Précision : lorsque nous ferons référence au 'contenu d'un tag', nous parlerons en fait du contenu du champ 'string' ou du champ 'value' de l'uicontrol marqué par le tag en question.

7.1 Prog_load()

Prog_load() est la fonction qu'il faut lancer pour ouvrir l'interface. Son appel ouvre la fenêtre que nous appelons 'interface0'. L'objectif d'interface0 est de récolter des informations sur l'expérience. Des uicontrols de style edit récupèrent ces informations. Leurs tags sont choisis de façon à pouvoir ensuite, dans une autre fonction, récupérer les informations afin de les stocker dans des variables. Nous ne décrivons pas ici le code qui a servi à la mise en page de la fenêtre.

L'utilisateur peut créer jusqu'à 15 options et 5 stimuli.

On entre donc dans une boucle de 15 itérations dans laquelle on crée, pour chaque option **numéro_option**, les uicontrols avec les tags suivants (les tags en Gris seront créés mais leurs uicontrols ne seront pas forcément remplis) :

- 'signin**numéro_option**' : nom_signification de l'option
- 'nom_mat1.**numéro_option**' : nom_matrice de la première matrice contenant les données
- 'nom_mat2.**numéro_option**' : nom_matrice de la deuxième matrice contenant les données
- 'nom_mat3.**numéro_option**' : nom_matrice de la troisième matrice contenant les données

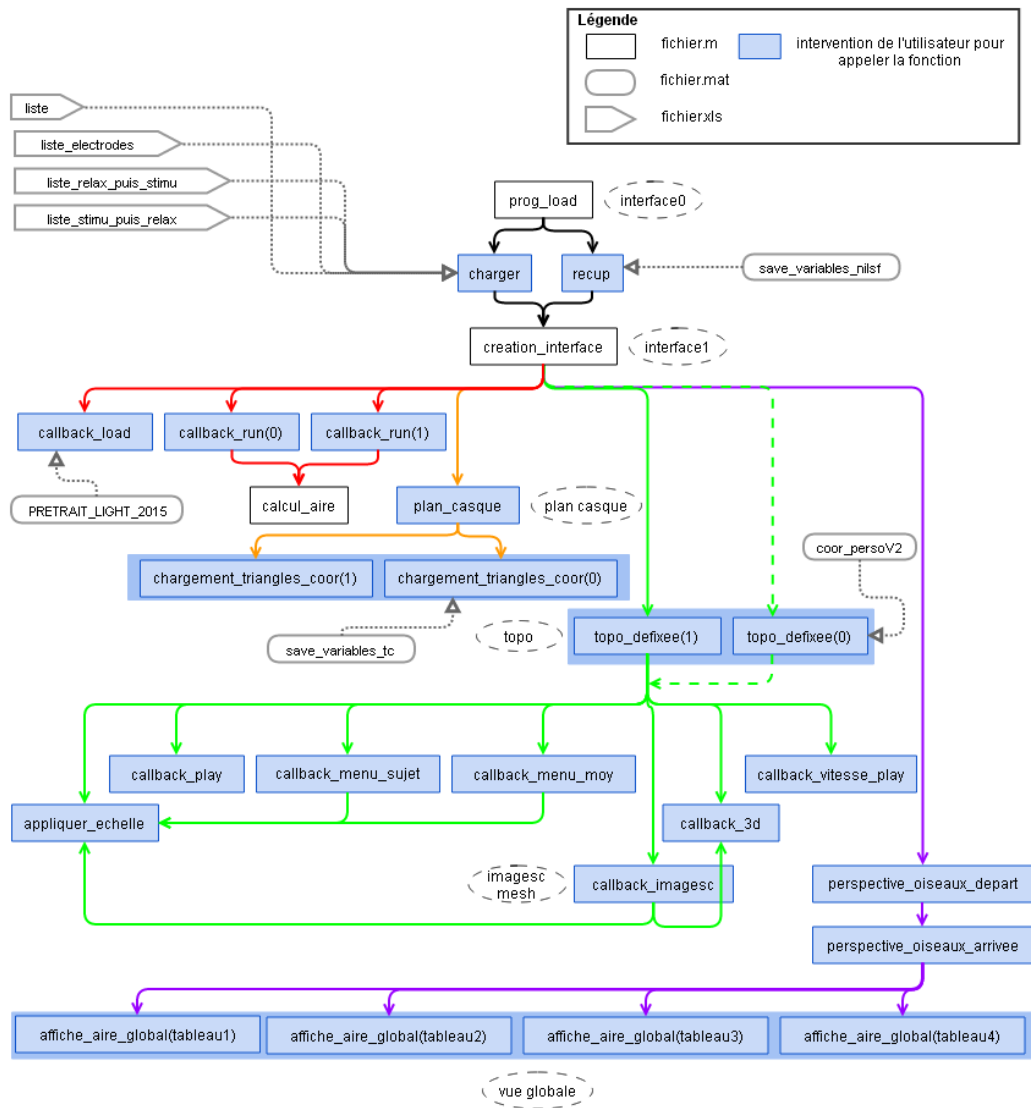


FIGURE 7.1 – Architecture des fichiers/fonctions de l'interface Defixe

- '**nom_mat4.numero_option**' : nom_matrice de la quatrième matrice contenant les données
- '**nom_mat5.numero_option**' : nom_matrice de la cinquième matrice contenant les données
- '**nom_excel1.numero_option**' : nom du fichier Excel contenant les lignes à prendre en compte dans la première matrice
- '**nom_excel2.numero_option**' : nom du fichier Excel contenant les lignes à prendre en compte dans la deuxième matrice
- '**nom_excel3.numero_option**' : nom du fichier Excel contenant les lignes à prendre en compte dans la troisième matrice
- '**nom_excel4.numero_option**' : nom du fichier Excel contenant les lignes à prendre en compte dans la quatrième matrice
- '**nom_excel5.numero_option**' : nom du fichier Excel contenant les lignes à prendre en compte dans la cinquième matrice
- '**tousnumero_option**' : 1 si est coché, 0 sinon
- '**selectnumero_option**' : 1 si est coché, 0 sinon

Puis on entre dans une boucle de 5 itérations, dans laquelle on crée, à chaque itération **numéro_itération**, les uicontrols avec les tags suivants :

- '**stimuli_nomnuméro_itération**' : nom/signification du stimulus
- '**stimuli_tempsnuméro_itération**' : moment d'apparition du stimulus en millisecondes

Pour récupérer les informations concernant le casque EEG, sont créés les uicontrols avec les tags suivants :

- '**edit_nombre_electrodes**' : nombre d'électrodes total sur le casque EEG
- '**edit_liste_electrodes**' : nom du fichier Excel contenant la liste des électrodes

La fréquence d'enregistrement est récupérée dans l'uicontrol de tag '**edit_frequence**'.

Deux uicontrols de style pushbutton permettent de passer à la suite. 'Charger et sauvegarder ces données' est utilisé pour se servir des informations remplies dans les différents uicontrols cité ci-dessus. Son callback envoie sur la fonction charger(). 'Charger anciennes données' est à utiliser dans le cas où l'utilisateur avait auparavant rempli les uicontrols cités ci-dessus et sauvegardé ces informations. Son callback renvoie sur la fonction recup().

7.2 Charger()

La fonction charger() créé 5 variables globales dans lesquelles sont stockées les informations renseignées dans interface0.

Pour récupérer les informations concernant les options, on entre dans une boucle de 15 itérations. Cette boucle peut s'arrêter avant la réalisation des 15 itérations. En effet, dès que le nom d'une option n'est pas rempli, on sort de la boucle.

Les informations seront placées dans la variable globale `info`. `Info` est une cellule. A chaque itération, c'est-à-dire, pour chaque option, une cellule est placée dans `info`. On accède alors à cette cellule en écrivant '`info{numéro_option}`'. `Info` contient donc, elle-même, une cellule par option. Pendant l'itération, une fois `info{numéro_option}` créée, elle est remplie de la façon suivante :

- Dans sa colonne 1, `info{numéro_option}1`, est placé le contenu du tag '`signuméro_option`'.
- Dans sa colonne 2, `info{numéro_option}2`, est créée une cellule. Dans cette dernière, via une boucle de 5 itérations, sont placés les contenus des tags :
 - '`nom_mat1.numéro_option`' colonne 1 (`info{numéro_option}{2}{1}`)
 - '`nom_mat2.numéro_option`' colonne 2 (`info{numéro_option}{2}{2}`)
 - '`nom_mat3.numéro_option`' colonne 3
 - '`nom_mat4.numéro_option`' colonne 4
 - '`nom_mat5.numéro_option`' colonne 5
 Sur le même principe que pour les options, la taille de la cellule `info{numéro_option}{2}` est le nombre de '`nom_mat[...].numéro_option`' remplis d'affilée.
- Dans sa colonne 3, `info{numéro_option}3`, est créée une cellule remplie de la même façon que `info{numéro_option}2`, à la différence que, dans la cellule, ne sont pas placés directement le(s) contenu(s) du/des tag(s) '`nom_excel1.numéro_option`', ... On y place le(s) contenu(s) du/des fichiers Excel(s) sous forme d'une/de liste(s) de nombres.
- Dans sa colonne 4, `info{numéro_option}4`, est placé un 1 si l'utilisateur a coché tous, 0 sinon.

Pour les informations concernant les stimuli, on entre dans une boucle de 5 itérations (une itération par stimulus). Cette boucle, toujours comme pour les options, s'arrêtera dès qu'un stimulus n'est pas rempli. C'est la variable globale `stimuli` qui récupère les informations. Cette variable est une cellule contenant autant de lignes qu'il y a de stimuli et deux colonnes. La première colonne récupère les contenus des tags '`stimuli_nomnuméro_stimulus`' et la deuxième récupère les contenus des tags '`stimuli_tpsnuméro_stimulus`'.

La variable globale `nombre_electrodes` récupère le contenu du tag '`edit_nombre_electrodes`'. La variable globale `liste_electrodes` récupère, sous forme d'une liste de nombres, le contenu du fichier Excel dont le nom se trouve au tag `edit_liste_electrodes`'.

La variable globale `frequence` récupère le contenu du tag '`edit_frequence`'.

Ensuite est appelée la fonction `creation_interface()` qui va créer la fenêtre que nous appelons `interface1`. Est également ouverte une boîte de dialogue permettant d'enregistrer les cinq variables globales que nous venons de décrire dans un fichier `.mat` à un emplacement souhaité. Une fois la sauvegarde faite par l'utilisateur, la fenêtre `interface0` est fermée.

7.3 Recup()

La fonction `recup()` ouvre une boîte de dialogue permettant de charger le fichier `.mat` contenant les cinq variables globales décrites dans la fonction `char-`

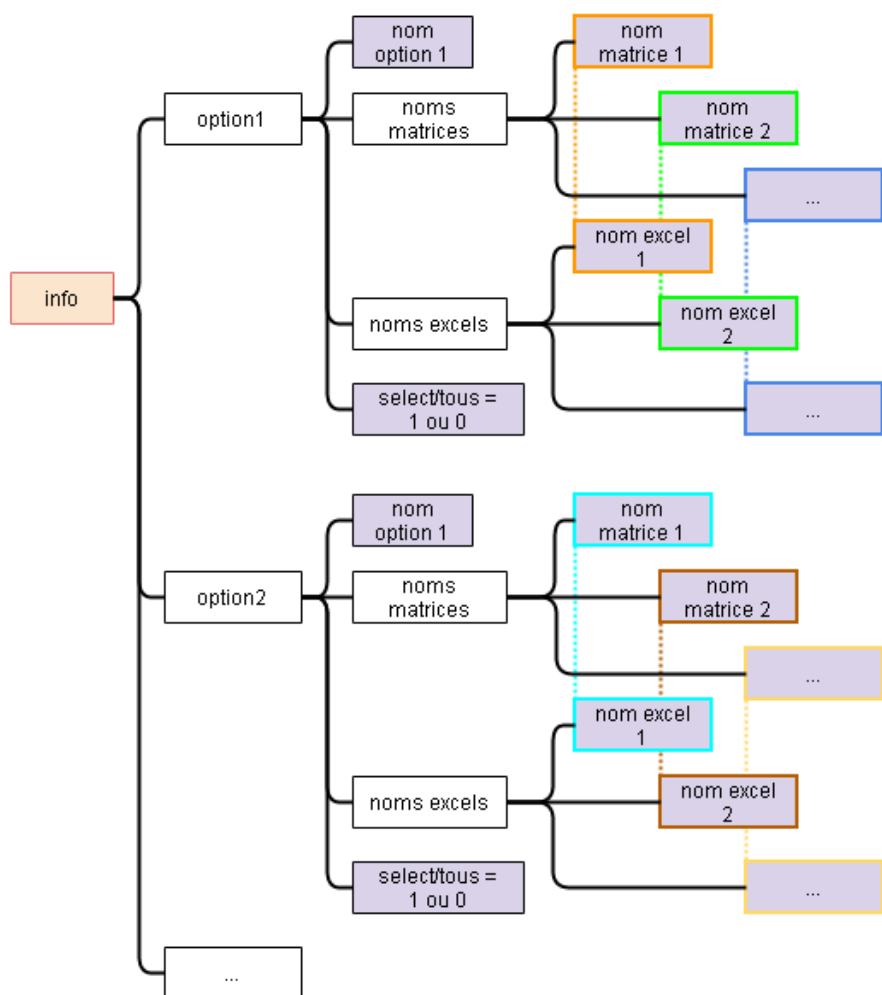


FIGURE 7.2 – Architecture de la cellule info

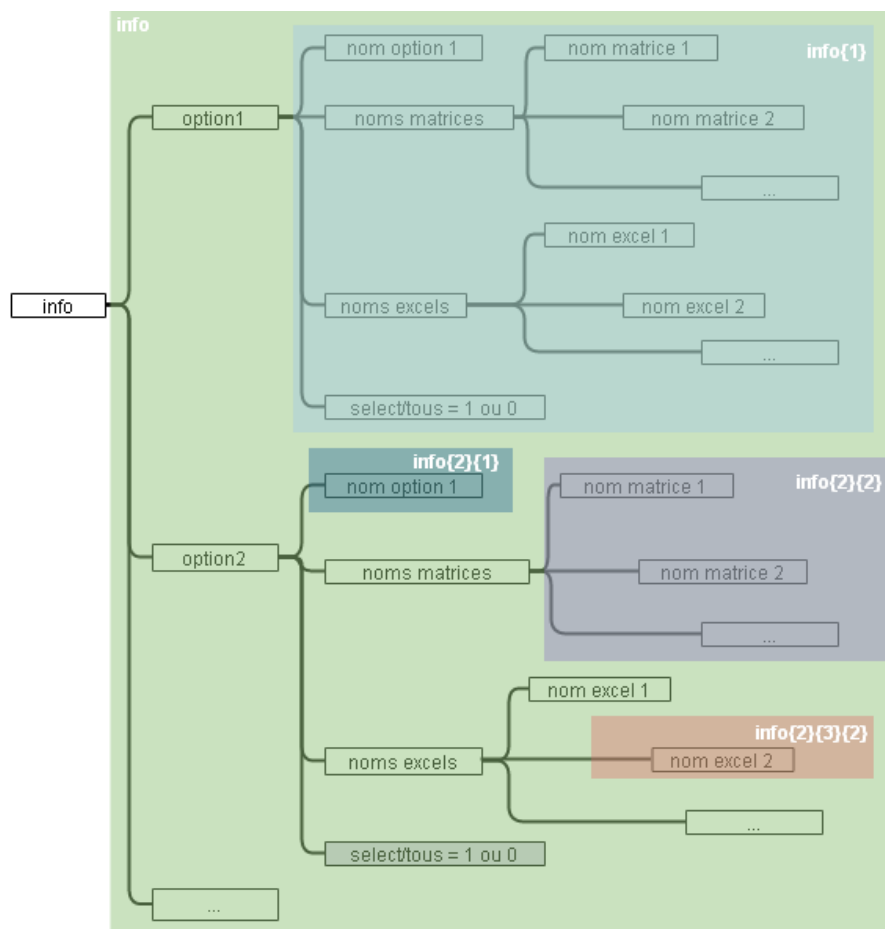


FIGURE 7.3 – Cellule info : accès aux différents champs

ger()). Une fois ces variables chargées, comme la fonction charger(), il appelle la fonction creation_interface() et ferme la fenêtre interface0.

7.4 Creation_interface()

Nous ne décrivons pas ici les détails de la mise en page.

Dans cette fonction sont créés la figure (ou fenêtre) et l'axe (ou graphique).

Sont créés des uicontrols de style pushbutton que nous pouvons séparer en trois groupes. Un uicontrol a pour effet de modifier l'affichage dans la figure de l'interface1. C'est le pushbutton nommé 'run' dont le callback est la fonction callback_run(0). Un uicontrol a pour simple effet de fermer la figure. Six autres uicontrols ouvrent de nouvelles figures :

- 'Détacher figure' dont le callback est *callback_run(1)*
- 'Carte topographique' dont le callback est *topo_defixe(1)*
- 'Carte topographique fixe' dont le callback est *topo_defixe(0)*
- 'Vue globale' dont le callback est *perspective_oiseaux_depart()*
- 'Charger position électrodes' dont le callback est *plan_casque()*. Ce pushbutton est accolé à un uicontrol de style text qui sera modifié par son callback, initialisé à 'non effectué' pour indiquer que le callback n'a pas été appelé.
- 'LOAD' dont le callback est *callback_load()*

Des uicontrols de style edit et checkbox récupèrent des informations qui serviront de paramètres à l'affichage dans la figure. Nous listerons ici leurs tags et l'information qu'ils contiennent :

tag	information
Edit_num_elec	Numéro de l'électrode dont on affiche les données dans la figure
Edit_bas	Borne inférieure de l'intervalle de calcul de l'aire sous la courbe
Edit_haut	Borne supérieure de l'intervalle de calcul de l'aire sous la courbe
Moyenne	Valeur=1 si on veut afficher les options avec les données moyennées sur les sujets qu'elles concernent respectivement
Tous	Valeur=1 si on veut afficher les options pour chaque sujet de l'expérience
Unique	Valeur=1 si on veut afficher les options pour un unique sujet dont on choisi le numéro
Edit_num_indiv	Numéro du sujet dont on affiche les informations

Chaque option est affichée sous forme de checkbox dans la couleur qui lui est attribuée. A côté de chacune de ces checkbox sont placés un ou deux uicontrols de style text qui servent à l'affichage de l'aire sous la courbe de chacune des options. Les options pour lesquelles le tag 'tous' contient 1 auront deux affichages d'aires (un pour le sujet et un pour la moyenne des sujets) et les autre

n'auront qu'un affichage d'aire (pour la moyenne des sujets). Pour chaque option, le tag de la checkbox sera '`nom_optionnuméro_option`', les tags pour l'aire seront '`aire_moynuméro_option`' et '`aire_uniquenuméro_option`'.

Les couleurs sont stockées dans une cellule (nommée `couleur_fond`) contenant une ligne et 15 colonnes (une colonne par option). Dans le programme, à chaque fois que l'affichage d'une option nécessitera une légende de couleur, cette cellule sera utilisée.

7.5 Callback_load()

La fonction `callback_load` crée des variables globales qui seront utiles tout au long du programme et charge les matrices contenant les données de l'expérience. Les variables globales créées sont les suivantes :

- `Nb_stimuli` : nombre de stimuli
- `Nb_matrice_par_option` : liste de nombre (un nombre par option). Elle contient pour chaque option, le nombre de matrices contenant ses données. C'est-à-dire le nombre d'éléments dans `info{numéro_option}{2}`
- `Liste_sous_matrices` : liste sans doublons des `nom_matrices` des matrices contenant les données des options

Nous appelons 'sous matrices', les matrices obtenues à partir des `nom_matrices` contenus dans `liste_sous_matrices`. Les sous matrices sont déclarées comme variables globales. Puis il y a ouverture d'une boîte de dialogue permettant à l'utilisateur de charger le fichier contenant ces sous matrices.

- `Nb_sujets` : nombre de sujets dans l'expérience. Pour le trouver, on prend dans une option concernant tous les sujets, le numéro de sujet le plus grand qu'elle concerne (les numéros des sujets qu'elle concernent se trouvent dans ses listes Excel de sujet, dans `info{numéro_option}{3}`)
- `Tps_max` : indice maximal de temps des matrices (vérifier pour une matrice suffit).

7.6 Callback_run(detache)

La fonction `callback_run(detache)` modifie l'affichage dans l'interface1 : dans l'axe et les uicontrols affichant les calculs d'aires. Elle prend en paramètre la variable '`detache`' qui contient un 0 ou un 1. Dans les deux cas, le travail de `callback_run(detache)` est le même : l'affichage des courbes et calculs d'aires en fonction des paramètres choisis par l'utilisateur. Lorsque `callback_run(detache)` est appelée en appuyant sur le pushbutton 'run', `detache=0`. Alors, la fonction place les courbes et calculs d'aire dans l'axe de l'interface1 (la figure courante est celle du tag '`fig_interface1`'). Lorsque `callback_run(detache)` est appelée en appuyant sur le pushbutton 'détacher figure', `detache=1`. Alors, on crée une nouvelle figure munie d'un axe dans lequel les courbes seront tracées (les calculs d'aires sont quand même placés dans l'interface1). On donne à cette figure une barre d'outils supplémentaire permettant à l'utilisateur de modifier le graphique.

Dans les trois cas que nous allons voir, nous ne travaillons que sur les données de l'électrode sélectionnée dans l'interface1. Ainsi, lorsque l'on récupère les don-

nées d'une option dans les 'sous-matrices' dont les `nom_matrice` sont contenus dans `info{numéro_option}{2}`, on choisi les matrices contenant les données de l'électrodes en question : '`nom_matricenuméro_électrode`'.

Cas 1

Si le tag 'tous' contient 1, le programme fera le travail suivant pour chaque option ayant été cochée et concernant tous les individus :

```

1  for j_mat_ds_option=1:nb_matrice_par_option(i_option) %pour ←
    chaque matrice qui remplit cette option
2  for sous_ligne=info{i_option}{3}{j_mat_ds_option} %pour ←
    chaque ligne (sujet) de cette matrice a prendre en ←
    compte (indiqué par info{i_option}{3}
3  eval(['matrice_tous' num2str(i_option) '(sous_ligne,:)=' ←
    info{i_option}{2}{j_mat_ds_option} valeur_num_elec ←
    '(sous_ligne,:);']);
4  end
5  end
6  eval(['matrice_transposee' num2str(i_option) '=transpose(' ←
    matrice_tous' num2str(i_option) ');']);

```

Il créé la matrice `matrice_tousnuméro_option` et place à l'intérieur les données de l'option. Les données que doit contenir l'option sont indiquées par `info{numéro_option}{2}` et `info{numéro_option}{3}` contenant le nom de la/des matrice(s) contenant les données de l'option, ainsi que les lignes à récupérer dans chacune de ces matrices. On obtient donc la matrice suivante :

$$matrice_tousnuméro_option = \begin{matrix} & 1 & \dots & tps_max \\ \begin{matrix} 1 \\ \vdots \\ nb_sujets \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Il créé sa transposée :

$$matrice_transposenuméro_option = \begin{matrix} & 1 & \dots & nb_sujets \\ \begin{matrix} 1 \\ \vdots \\ tps_max \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Il l'affiche.

Cas 2

Si le tag 'unique' contient 1, le programme fera le travail suivant pour chaque option ayant été cochée et concernant tous les individus :

```

1  for j_mat_ds_option=1:nb_matrice_par_option(i_option)
2  for sous_ligne=info{i_option}{3}{j_mat_ds_option}

```

```

3   eval([ 'matrice_unique' num2str(i_option) '(sous_ligne,:) ←
      =' info{i_option}{2}{j_mat_ds_option} ←
      valeur_num_elec '(sous_ligne,:); ']);
4   end
5 end
6 eval([ 'matrice_unique_sujet' num2str(i_option) '= ←
      matrice_unique' num2str(i_option) '(valeur_num_indiv,:); ←
      ']);

```

Il crée la matrice `matrice_unique`**numéro_option** et place à l'intérieur les données de l'option, de la même façon qu'il a créé `matrice_tous`**numéro_option**. On obtient donc la matrice suivante :

$$matrice_unique\text{numéro_option} = \begin{matrix} & & 1 & \dots & tps_max \\ & 1 & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ nb_sujets & ddp & \dots & ddp \end{pmatrix} \\ & \vdots & \\ & nb_sujets \end{matrix}$$

Il récupère la ligne correspondant au sujet sélectionné :

$$matrice_unique_sujet\text{numéro_option} = \begin{matrix} & 1 & \dots & tps_max \\ 1 & \begin{pmatrix} ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Il affiche et il appelle la fonction `calcul_aire(numéro_option,'matrice_unique_sujet`**numéro_option**',0)

 qui va calculer l'aire sous cette courbe et l'inscrire au bon emplacement dans l'interface1.

Cas 3

Si le tag 'moyenne' contient 1, le programme fera le travail suivant pour chaque option ayant été cochée :

```

1 ligne=1;
2 for j_mat_ds_option=1:nb_matrice_par_option(i_option)
3   for sous_ligne=info{i_option}{3}{j_mat_ds_option}
4     eval([ 'matrice_moy' num2str(i_option) '(ligne,:)= ' info{←
      i_option}{2}{j_mat_ds_option} valeur_num_elec '(←
      sous_ligne,:); ']);
5     ligne=ligne+1;
6   end
7 end
8 eval([ 'matrice_moy_mean' num2str(i_option) '=nanmean(←
      matrice_moy' num2str(i_option) '(:,:); ']);

```

Il crée une matrice `matrice_moy`**numéro_option**. Dans cette matrice il place les lignes (toujours indiquées dans `info{numéro_option}{3}`) des matrices (toujours indiquées dans `info{numéro_option}{2}`) contenant les données de l'option. Il place ces lignes l'une après l'autre au fur et à mesure qu'elles sont lues, sans se soucier des numéros de sujets. Ainsi dans `matrice_moy`**numéro_option**,

le numéro de ligne ne correspond pas au numéro de sujet. On obtient la matrice suivante :

$$matrice_moy_{numéro_option} = \begin{matrix} & 1 & \dots & tps_max \\ \begin{matrix} 1 \\ \vdots \\ nombre\ sujets\ concernés\ par\ l'option \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Il fait la moyenne sur les lignes de cette matrice :

$$matrice_moy_mean_{numéro_option} = \begin{matrix} & 1 & \dots & tps_max \\ 1 & (ddp & \dots & ddp) \end{matrix}$$

Il affiche et il appelle la fonction `calcul_aire(numéro_option, 'matrice_moy_mean'numéro_option', 1)`

Donc à chaque courbe à afficher correspond une matrice (la matrice que l'on affiche). Juste après création de chacune de ces matrices, le minimum et le maximum de la matrice est calculé. L'objectif de cette démarche est de récupérer un minimum et un maximum valables pour toutes les courbes affichées dans l'axe. Nous les appelons `y_min` et `y_max`.

Pour cela, on initialise ces deux variables à 0 au début de la fonction. On les compare aux minimum et maximum de chaque matrice nouvellement créée (pour une courbe), et on ne garde respectivement que le plus petit et le plus grand des deux (dans `y_min` et `y_max`). Cela nous assure ensuite de pouvoir afficher toutes les courbes intégralement dans l'axe en choisissant le minimum et le maximum de l'axe des ordonnées à la fin de la fonction.

On trace en jaune et en bleu les lignes verticales correspondant respectivement aux bornes du calcul d'aire et aux stimuli.

7.7 Calcul_aire(numéro_option,matrice,moy)

La fonction `calcul_aire(numéro_option, matrice,moy)` calcule l'aire, sur l'intervalle choisi dans l'interface1, sous la courbe correspondant à la matrice 'matrice'. Puis elle affiche cette aire au bon emplacement dans l'interface1, emplacement indiqué par le numéro de l'option 'numéro_option' et la variable 'moy' qui dit s'il faut placer le résultat au tag 'aire_moy'numéro_option' ou au tag 'aire_uniquenuméro_option'.

7.8 Plan_casque()

La fonction `plan_casque` ouvre une nouvelle figure de tag 'fig_casque'.

On va créer un tableau constitué d'uicontrols de style edit dans lesquels l'utilisateur entrera les numéros d'électrodes. Le tableau contient autant de lignes que de colonnes. Le nombre de lignes (et colonnes) est calculé et majoré de façon à ce que toutes les électrodes puissent être placées dans le tableau. Ce nombre est placé dans la variable `bn`.

Le tableau représente schématiquement le casque EEG. Alors il doit contenir un nombre impair de cases afin d'avoir deux lignes centrales : une horizontale et une verticale.

On crée une matrice à trois dimensions 'carre' telle que :

- `Carre(:, :, 1)` est une matrice de `bn` lignes et `bn` colonnes contenant la coordonnée x des électrodes
- `Carre(:, :, 2)` est une matrice de `bn` lignes et `bn` colonnes contenant la coordonnée y des électrodes
- `Carre(:, :, 3)` est une matrice de `bn` lignes et `bn` colonnes contenant la coordonnée z des électrodes

Imaginons que nous sommes sur un repère orthonormé (i,j) et que nous ne regardons que les entiers. Nous plaçons le carré dans ce repère de sorte que sa case centrale ait les coordonnées (0,0). Chaque case du carré a pour coordonnées les deux entiers (l'abscisse x et l'ordonnée y) dont elle se trouve à l'intersection. Grâce à deux boucles imbriquées, pour chaque case du carré, on remplit `carre(i,j,1)` avec la valeur en abscisse de la case, on remplit `carre(i,j,2)` avec la valeur en ordonnée de la case, et on remplit `carre(i,j,3)` avec la valeur $abscisse^2 + ordonnée^2$.

A chaque fois que ces trois valeurs sont remplies, on crée l'uicontrol correspondant à la case. Son tag est 'case*i_j*'.

Si l'utilisateur n'avait jamais rempli ce tableau de placement des électrodes, il le remplit et clique sur l'uicontrol 'charger et sauvegarder ces données' dont le callback est la fonction `chargement_triangles_coor(1)`. S'il avait déjà réalisé cette démarche et enregistré les positions il clique sur l'uicontrol 'charger anciennes données' dont le callback est la fonction `chargement_triangles_coor(0)`.

7.9 Chargement_triangles_coor(nouveau)

La fonction `chargement_triangles_coor(nouveau)` crée ou charge les deux variables globales, `triangles` et `coor`, qui seront nécessaires pour l'affichage de la carte topographique.

Si `nouveau=1`, il faut créer les deux variables globales et placer à l'intérieur les données écrites par l'utilisateur dans le tableau d'uicontrols de la façon suivante :

Dans 'coor', on place les coordonnées (x,y,z) de chaque électrode placée dans le tableau d'uicontrols. Le nom du tag 'case*i_j*' de chaque case correspond à un emplacement dans carré (matrice 3D qui contient les coordonnées (x,y,z) de toutes les cases du tableau). Donc pour chaque case qui contient un nombre

(ce nombre étant un numéro d'électrode), on récupère ses coordonnées (x,y,z) que l'on place dans la ligne de coor qui correspond au numéro d'électrode. Coor possède donc autant de lignes qu'il y a d'électrodes sur le casque et possède trois colonnes (pour x, y et z).

A partir des abscisses et ordonnées des électrodes, la fonction delaunay crée des triangles dont les trois côtés correspondent à trois numéros d'électrodes. Elle crée ces triangles de façon à ce qu'ils 'remplissent' tout l'espace entre les électrodes sans superpositions de triangles. Elle renvoie alors une matrice contenant autant de lignes qu'elle a créé de triangles, et contenant trois colonnes (une pour chaque sommet de triangle). On stocke cette matrice dans la variable globale 'triangles'.

On déclenche l'ouverture d'une boîte de dialogue permettant à l'utilisateur de sauvegarder triangles et coor dans un fichier .mat dont il choisit le nom et l'emplacement.

Si nouveau=0, une boîte de dialogue est ouverte, permettant à l'utilisateur de charger le fichier contenant triangles et coor qui avaient été créés auparavant.

On ferme la figure et l'uicontrol text placé sous le pushbutton 'charger position électrodes' affiche 'effectué'.

7.10 Topo__defixee(defixe)

La fonction topo_defixee(defixe) crée une nouvelle figure, de tag 'lafigure', contenant un axe de tag 'laxe'.

Si defixe=0, on utilise les variables coor et triangles créées spécifiquement pour l'expérience vigilance-musique, en chargeant le fichier qui les contient. Sinon coor et triangles doivent avoir été créés via les fonctions plan_casque() et chargement_triangles_coor(nouveau).

Deux menus permettront de choisir les options à afficher sur la carte topographique. Ils seront créés à partir des options cochées dans l'interface1.

- Le menu concernant les données du sujet choisi propose les options qui concernent tous les sujets.
- Le menu concernant les données moyennées sur les sujets propose toutes les options cochées.

Pour chaque menu on crée son champ 'string' ainsi qu'une cellule contenant les noms des options (info{numéro_option}{1}) qu'il propose. Les deux cellules ainsi créées sont les variables globales liste_mat_menu_sujet et liste_mat_menu_moy.

Vit, valeurs, point

On crée les trois variables globales vit, valeurs et point. Vit détermine la vitesse à laquelle se joue le film de la carte topographique. Valeurs est la matrice

contenant les données qui sont affichées sur la carte topographique, ses lignes sont les numéros d'électrodes et ses colonnes le temps.

$$valeurs = \begin{matrix} & 1 & \dots & tps_max \\ \begin{matrix} 1 \\ \vdots \\ nombre_electrodes \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Point est le temps auquel est affichée la carte topographique (c'est l'indice qui indique quelle colonne de valeurs doit être 'affichée' sur la carte topographique).

Si aucune option n'a été cochée dans l'interface1 un message est affiché pour en informer l'utilisateur et la fonction s'arrête là. Si des options ont été cochées, on poursuit avec les étapes ci-dessous.

Patch

La carte topographique est créé par la fonction `patch()` qui prend notamment en paramètres `coor`, `triangles` et la colonne de la matrice `valeurs` indiquée par `point`.

Menus

Les menus sont créés avec les tags '`menu_sujet`' et '`menu_moy`'. Les callbacks de chacun de ces menus effectuent les tâches suivantes : ils changent les couleurs de certains uicontrols pour mettre en avant l'option qui a été sélectionnée, ils appellent respectivement la fonction `callback_menu_sujet()` et la fonction `callback_menu_moy()` qui modifient le contenu de la matrice `valeurs`, et ils mettent à jour l'affichage de la carte topographique avec la nouvelle matrice `valeurs`.

Echelle

Les tags '`min_ech`' et '`max_ech`' récupèrent les bornes de l'intervalle de temps sur lequel l'utilisateur souhaite que l'échelle de couleur soit calculée. A l'appui sur le pushbutton '`appliquer`', la fonction `appliquer_echelle()` est appelée.

On initialise la matrice `valeurs` en appelant la fonction `callback_menu_moy()`.

Temps

On crée un uicontrol de style text qui donne le temps auquel est affichée la carte topographique en convertissant '`point`' en millisecondes. Son tag est '`edit_tps`'.

Slider

On crée un slider dont le callback modifie `point` selon la valeur du slider, puis met à jour l'affichage de la carte topographique ainsi que le contenu du tag '`edit_tps`' avec la nouvelle valeur de `point`.

La légende placée au dessus du slider est créée par une série d'uitcontrols de style frame superposés. Pour les créer, on utilise un coefficient de proportionnalité calculé en divisant la longueur du slider en pixels par la durée qu'il parcourt en millisecondes. Ce coefficient permet de calculer où placer les frames et quelle longueur leur donner. On place un frame gris foncé sur toute la longueur du slider, un frame jaune pour représenter l'intervalle de temps sur lequel est calculée l'aire. Les stimuli sont représentés par des frame bleus sous forme de traits.

Film

Des uicontrols de style pushbutton permettent de faire jouer la carte topographique sous forme d'un film et d'en modifier la vitesse. Le pushbutton 'play/stop' a pour tag 'video' et pour callback `callback_play()`. Les callbacks des pushbutton 'plus' et 'moins' respectivement incrémente et décrémente vit de 1 et appellent la fonction `callback_vitesse_play()`.

3D

Le pushbutton 'rotate 3D' est accolé à l'uitcontrol text de tag 'edit_troisd' dont le champ 'string' est initialisé à 'off'. Ce pushbutton a pour callback la fonction `callback3d('edit_troisd')`.

Imagesc

Le pushbutton 'imagesc' a pour callback la fonction `callback_imagesc()`.

L'échelle de couleur est initialisée en appelant la fonction `applique_echelle()` alors que l'intervalle de temps de calcul de l'échelle à été initialisé à la durée totale de l'enregistrement EEG.

7.11 Callback_menu_moy()

La fonction `callback_menu_moy()` appelée dès que le menu est utilisé, remplit la matrice valeur en fonction de l'option choisie dans le menu.

La valeur du menu indique, dans la cellule `liste_mat_menu_moy`, quelle option a été choisie. On cherche cette option parmi toutes les options pour récupérer son numéro. Puis on fait le travail suivant pour chaque numéro d'électrode :

```

1  sujet=1;
2  for j_mat_ds_option=1:nb_matrice_par_option(i_option) %dans ←
    chaque sous-matrice de l'option
3      for num_sujet=info{i_option}{3}{j_mat_ds_option} %pour ←
        chaque sujet de cette sous matrice a prendre en compte
4      eval(['matr' num2str(elec) '(sujet,:)=' info{i_option}←
            '{2}{j_mat_ds_option} num2str(elec) '(num_sujet,:) ; '←
            ']);

```

```

5     sujet=sujet+1;
6     end
7 end
8 eval(['valeurs(elec,:)=nanmean(matr' num2str(elec) '(:,:));'←
    ]);

```

On crée la matrice **matrnuméro_électrode**. Dans cette matrice on place les lignes (info{numéro_option}{3}) des matrices (info{numéro_option}{2}) contenant les données de l'option pour l'électrode considérée. On place ces lignes l'une après l'autre au fur et à mesure qu'elles sont lues, sans se soucier des numéros de sujets. Ainsi dans **matrnuméro_électrode**, le numéro de ligne ne correspond pas au numéro de sujet. On obtient la matrice suivante :

$$\text{matrnuméro_électrode} = \begin{matrix} & 1 & \dots & tps_max \\ & ddp & \dots & ddp \\ & \vdots & \ddots & \vdots \\ \text{nombre sujets concernés par l'option} & ddp & \dots & ddp \end{matrix}$$

On fait la moyenne sur les lignes de cette matrice et on place le vecteur ligne obtenu dans la ligne de valeurs correspondant au numéro d'électrode.

$$\text{valeurs(numéro_électrode,:)} = \begin{matrix} & 1 & \dots & tps_max \\ 1 & ddp & \dots & ddp \end{matrix}$$

On appelle `appliquer_echelle()` pour mettre à jour l'échelle de couleur avec la nouvelle matrice valeurs.

7.12 Callback_menu_sujet()

La fonction `callback_menu_sujet()` appelée dès que le menu de tag 'menu_sujet' est utilisé, remplit la matrice valeur en fonction de l'option choisie dans le menu. Le travail est similaire à celui effectué dans `callback_menu_moy()`.

La valeur du menu indique, dans la cellule liste `_mat_menu_sujet`, quelle option a été choisie. On cherche cette option parmi toutes les options pour récupérer son numéro. Puis on fait le travail suivant pour chaque numéro d'électrode :

```

1 for j_mat_ds_option=1:nb_matrice_par_option(i_option) %pour ←
   chaque sous matrice de l'option
2     %si l'indiv est dans la sous matrice (normalement cela n←
   'arrive qu'une fois entre toutes les sous matrices ←
   de l'option)
3     if any(nindividu==info{i_option}{3}{j_mat_ds_option})
4         eval(['valeurs(elec,:)=info{i_option}{2}{←
           j_mat_ds_option} num2str(elec) '(nindividu,:);'←
           ]);
5     end
6 end

```


On cherche dans les matrices contenant les données de l'option (info{numéro_option}{2}) pour l'électrode considérée, parmi les lignes qu'il faut prendre en compte dans ces matrices (info{numéro_option}{3}), la ligne qui correspond au numéro du sujet qui a été choisi dans l'interface1. On place cette ligne dans valeur au numéro de ligne correspondant au numéro d'électrode.

On appelle appliquer_echelle() pour mettre à jour l'échelle de couleur avec la nouvelle matrice valeurs.

7.13 Callback_imagesc()

La fonction callback_imagesc() donne la même information que nous aurions en regardant le film de la carte topographique : la différence de potentiel observée en fonction de l'électrode et du temps sur un graphique. Elle donne cette information sous deux formes dans deux fenêtres. La première représentation est donnée par la fonction imagesc(), la deuxième par la fonction mesh(). Toutes deux prennent en paramètre la matrice valeurs. Mesh(), comme imagesc(), code la différence de potentiel par la couleur, mais en plus la code en volume en lui donnant une troisième coordonnée.

Dans les deux représentations, les bornes de l'intervalle de temps sur lequel est calculée l'aire sous la courbe sont indiquées en jaune, et les stimuli en bleu. La différence est qu'alors que dans imagesc() ces indications sont sous forme de traits, dans mesh() elle sont sous forme de plans transparents.

Ces plans sont créés avec la fonction patch comme pour la carte topographique. Pour construire ces carrés/plans, deux variables doivent être construites. Une variable 'sommet_carre' contiendra les quatre sommets de chaque carré à construire, à raison d'un quatuor par ligne. Sommet_carre est donc une matrice de 2+nb_stimuli lignes et 4 colonnes. Une variable 'face_carre' contiendra les coordonnées (x,y,z) de chaque sommet de carre. Etant donné qu'il y a 2+nb_stimuli carrés, il y a (2+nb_stimuli)*4 sommets. Face_carre est donc une matrice de (2+nb_stimuli)*4 et trois colonnes (pour les trois coordonnées). Les sommets des carrés sont repérés par le numéro de la ligne à laquelle ils se trouvent dans face_carre et ce sont ces numéros de lignes qui sont inscrits dans la matrice sommet_carre.

Voici les significations des trois coordonnées de chaque point dans mesh : x est le temps, y est le numéro d'électrode, z est la différence de potentiel. La coordonnée x de 'face_carre' doit donc contenir la valeur temporelle de la borne de l'intervalle de calcul d'aire ou de l'apparition du stimulus. Cette valeur temporelle doit être convertie des millisecondes vers l'indice de temps (qui permet d'accéder aux colonnes de la matrice valeurs). Pour que les carrés/plans s'étendent dans tout l'axe, on place leurs sommets aux valeurs extrêmes d'y et

z.

$$\begin{aligned}
 \text{face_carre} = & \begin{matrix} & 1 & 2 & 3 & 4 \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ 2 + nb_stimuli \end{matrix} & \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & (2 + nb_stimuli) * 4 \end{pmatrix} \end{matrix} \\
 \\
 \text{sommet_carre} = & \begin{matrix} & 1 & 2 & 3 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ 9 \\ 10 \\ 11 \\ 12 \\ \vdots \\ (2 + nb_stimuli) * 4 \end{matrix} & \begin{pmatrix} borne_inf & 0 & min\ de\ valeurs \\ borne_inf & nombre_electrodes & min\ de\ valeurs \\ borne_inf & 0 & max\ de\ valeurs \\ borne_inf & nombre_electrodes & max\ de\ valeurs \\ \vdots & \vdots & \vdots \\ tps_stimulus_1 & 0 & min\ de\ valeurs \\ tps_stimulus_1 & nombre_electrodes & min\ de\ valeurs \\ tps_stimulus_1 & 0 & max\ de\ valeurs \\ tps_stimulus_1 & nombre_electrodes & max\ de\ valeurs \\ \vdots & \vdots & \vdots \\ \dots & \dots & \dots \end{pmatrix} \end{matrix}
 \end{aligned}$$

Les deux représentations graphiques sont graduées pour le temps toutes les 500 millisecondes, ce qui est adapté pour un enregistrement de 4000 millisecondes. Mais plus l'enregistrement sera long, plus la graduation sera dense, ce qui gênera la visibilité de l'affichage.

Pour les deux fenêtres on appelle la fonction `appliquer_echelle()`.

7.14 Appliquer_echelle()

La fonction `appliquer_echelle()` modifie l'échelle de couleur de la carte topographique, d'imagesc et de mesh. Ces trois affichages sont basés sur le contenu de la matrice `valeurs`.

L'échelle est créée en calculant le minimum et le maximum des différences de potentiels. Ces deux valeurs sont recherchées dans la matrice `valeurs` parmi les colonnes qui correspondent à l'intervalle de temps choisi par l'utilisateur.

On récupère cet intervalle de temps avec les tags '`min_ech`' et '`max_ech`'. S'il est valide, c'est-à-dire si la borne inférieure est plus petite que la borne supérieure et si toutes deux (après conversion des millisecondes à l'indice) permettent d'accéder à une colonne de la matrice `valeurs`, alors on calcule les nouveaux minimums et maximum. Si l'intervalle n'est pas valide, alors on le réinitialise pour le faire correspondre à l'intégralité de l'enregistrement et on calcule également de nouveaux minimum et maximum.

Puis on crée l'échelle avec les minimum et maximum calculés.

7.15 Callback_play()

La fonction `callback_play()` affiche l'évolution de la carte topographique au cours du temps, sous forme d'un film. Le temps auquel est affichée la carte est déterminé par la variable globale 'point'. Chaque modification de cette variable sera donc suivie d'une mise à jour des contenus des tags 'slider', 'edit_tps', 'patch'.

La fonction est appelée à l'appui sur le pushbutton de tag 'video'. Dès son appel elle regarde ce que contenait le champ 'string' du pushbutton juste avant que l'utilisateur ait cliqué dessus. Si ce champ contenait 'play', alors on le remplace par 'stop' et le champ 'value' prend la valeur 1. Si ce champ contenait 'stop', alors on le remplace par 'play' et le champ 'value' prend la valeur 0. Ainsi, le fait que l'utilisateur ait cliqué sur le pushbutton affichant le champ 'string' 'play' (plus simplement : le fait que l'utilisateur ait appuyé sur play), est codé par le fait que la valeur du pushbutton soit égale à 1. Et lorsque l'utilisateur appuie sur stop, la valeur du pushbutton est égale à 0.

L'effet de film est donné en faisant tourner une boucle tant que le pushbutton a pour valeur 1 et tant que nous ne sommes pas à la fin de l'enregistrement (ie. `point==tps_max`). Cette boucle incrémente la variable `point` de 1 et met alors à jour le patch (qui crée la carte) avec la colonne de la matrice valeurs qui est indiquée par la variable `point`. Avec cette nouvelle valeur de `point`, elle met également à jour la position du slider et l'affichage du temps.

Le changement de vitesse est réalisé en utilisant la fonction `pause()` qui force la fonction à cesser de travailler pendant un certain temps, temps que nous faisons dépendre de la variable `vit`. Le fait que l'utilisateur mette la vitesse à 0 a le même effet que s'il avait appuyé sur stop.

La boucle se termine, soit lorsque utilisateur a cliqué sur stop, soit lorsque la valeur de `point` a atteint `tps_max`.

Le fait que `point` ait atteint `tps_max` aura le même effet que si l'utilisateur avait appuyé sur stop. De plus, cela entraîne la remise au début du film en donnant à `point` la valeur 1 et en mettant à jour les contenus des tags 'patch', 'edit_tps' et 'slider' avec cette nouvelle valeur de `point`.

7.16 Callback_vitesse_play()

La fonction `callback_vitesse_play()` met à jour l'uicontrol text de tag 'vitesse' qui affiche la valeur de la variable `vit`. Cette fonction contrôle également la valeur de cette variable qui doit rester entre 0 et 10.

7.17 Callback_3d(tag)

La fonction `callback_3d()` active la possibilité de voir le contenu de l'axe en trois dimensions et de changer l'angle de vue à l'aide de la souris. Il modifie

également l'uicontrol du tag passé en paramètre qui affiche si oui ou non la vue en 3D est activée.

7.18 Perspective_oiseaux_depart()

La fonction `perspective_oiseaux_depart()` crée la figure et l'axe où seront affichées les données permettant d'avoir une vision globale des données de l'expérience. Y sont aussi créés les checkbox permettant à l'utilisateur de choisir comment il souhaite afficher ces données. Le principe de cette 'vision globale' est d'afficher l'aire sous la courbe dans l'intervalle choisi par l'utilisateur dans l'interface1. Cette aire est affichée de quatre façons différentes, donnant donc lieu à la création de quatre checkbox pour choisir un de ces affichages :

tag	affichage
Indiv_aire_pasmoy	Aire en fonction du sujet, pour l'électrode choisie dans l'interface1
Indiv_aire_moy	Aire en fonction du sujet, moyennée sur les électrodes de liste_electrodes
Elec_aire_pasmoy	Aire en fonction de l'électrode, pour le sujet choisi dans l'interface1
Elec_aire_moy	Aire en fonction de l'électrode, moyennée sur les sujets

Sont créés deux pushbutton 'Détacher_figure' et 'Execute' dont les callbacks appellent la fonction `perspective_oiseaux_arrivée()` avec respectivement les paramètres 1 et 0.

7.19 Perspective_oiseaux_arrivee (detache_graphe_global)

La fonction `perspective_oiseaux_arrivee(detache_graphe_global)` affiche dans un axe les données choisies par l'utilisateur lorsqu'il a sélectionné l'une des checkbox créées dans la fonction `perspective_oiseaux_depart()`.

Selon la valeur du paramètre `detache_graphe_global`, l'axe dans lesquels se fait l'affichage ne sera pas le même. Si `detache_graphe_global` est égal à 1, alors on crée une nouvelle figure avec un axe et des barres d'outils permettant la modification du contenu de cet axe. Sinon, l'axe est celui créé dans la fonction `perspective_oiseaux_depart()` et cet axe est vidé de son contenu au début de la fonction. Dans les deux cas les uicontrols qui seront créés par la suite dans cette fonction seront placés dans la figure créée dans `perspective_oiseaux_depart()` (cette figure sera leur 'parent').

Les bornes de l'intervalle de calcul de l'aire choisi par l'utilisateur dans l'interface1 sont récupérées. Elles sont converties des millisecondes à l'indice, et sont placées dans les variables E1 et E2 (pour respectivement la borne inférieure et la borne supérieure).

Tous les uicontrols éventuellement présents dans la figure créée dans `perspective_oiseaux_depart()` sont supprimés. S'ils existent, c'est qu'ils ont été créés dans un appel précédent de `perspective_oiseau_arrivée(detache_graphe_global)`.

Modifier la couleur des quatre checkbox permet de repérer quelle checkbox est 'affichée dans l'axe', et de repérer si la fonction est en cours de travail ou si elle a terminé l'affichage. Ces modifications de couleur sont réalisées au début et à la fin de la fonction : au début toutes les checkbox sont sur fond gris, à la fin la checkbox sélectionnée pour l'affichage est mise sur fond vert.

La variable `tab_check` est créée. Cette variable est une liste de `nb_option` chiffres : un chiffre par option. Chacun de ces chiffre est un 1 si l'option à été cochée dans l'interface1, un 0 sinon. Elle permettra, dans la suite du programme, de dire rapidement si une option est cochée ou non.

Le reste du code de la fonction ne sera exécuté que si l'aire choisie par l'utilisateur est valide (bornes dans l'ordre croissant et appartiennent à l'enregistrement).

Nous ne décrivons pas ici la mise en forme des axes.

Pour chacun des quatre affichages, la variable `pos_couleurs` est créée, contenant une liste de `nb_option` chiffres (un chiffre par option). Chacun de ces chiffres sera un 1 si l'option doit être affichée, un 0 sinon. Ils sont initialisés à 0 au début de la fonction et seront modifiés pendant la construction de la matrice qui contiendra les données à afficher.

Nous allons maintenant décrire les quatre cas.

Cas 1

Si le tag `'indiv_aire_pasmoy'` contient 1 : On crée la variable globale `tableau1`. Parmi toutes les options, si une option est cochée dans l'interface1 et qu'elle concerne tous les sujets de l'expérience, on réalise les étapes suivantes :

```

1 for j_mat_ds_option=1:nb_matrice_par_option(i_option)
2   for sous_ligne=info{i_option}{3}{j_mat_ds_option} %↵
      sous_ligne est le numero de sujet
3   eval([ 'tableau1(i_option,sous_ligne)=nansum(' info{↵
      i_option}{2}{j_mat_ds_option} valeur_num_elec '(↵
      sous_ligne,E1:E2)); ']);
4   end
5 end

```

Pour chacune des matrices contenant les données de l'option (`info{numéro_option}{2}`) on récupère chaque ligne à prendre en compte (`info{numéro_option}{3}`), en prenant les données concernant l'électrode choisie dans l'interface1.

Dans chacune de ces lignes (c'est-à-dire pour chaque sujet), on réalise le travail suivant : On calcule la somme des valeurs contenues des colonnes E1 à E2 de la ligne. Le résultat de cette somme est le calcul de l'aire dans l'intervalle choisi par l'utilisateur pour le sujet considéré, pour l'option `numéro_option`.

On place ce résultat dans `tableau1` à la ligne `numéro_option`, à la colonne `numéro_sujet`. Tableau 1 sera donc une matrice contenant les calculs d'aire. Ses lignes correspondent aux numéros d'options, ses colonnes correspondent aux numéros de sujets.

$$tableau1 = \begin{matrix} & \begin{matrix} 1 & \dots & nb_sujets \end{matrix} \\ \begin{matrix} 1 \\ \dots \\ nb_option \end{matrix} & \begin{pmatrix} aire\ ou\ NaN & \dots & aire\ ou\ NaN \\ \vdots & \ddots & \vdots \\ aire\ ou\ NaN & \dots & aire\ ou\ NaN \end{pmatrix} \end{matrix}$$

Cas 2

Si le tag `'indiv_aire_moy'` contient 1 : On crée la variable globale `tableau2`. Parmi toutes les options, si une option est cochée dans `l'interface1` et qu'elle concerne tous les sujets de l'expérience, on réalise les étapes suivantes :

```

1 for j_mat_ds_option=1:nb_matrice_par_option(i_option)
2   for sous_ligne=info{i_option}{3}{j_mat_ds_option} %↔
      sous_ligne est le numero de sujet
3   for elec = liste_electrodes
4     eval(['INDI',num2str(sous_ligne),'(' num2str(elec) ',:)' ↔
          =' info{i_option}{2}{j_mat_ds_option} num2str(elec) ↔
          '(sous_ligne,:)');'])
5   end
6
7   eval(['MINDI',num2str(sous_ligne),'=nanmean(INDI',num2str(↔
      sous_ligne),')');'])
8
9   eval(['tableau2(i_option,' num2str(sous_ligne) ') = nansum(↔
      (MINDI',num2str(sous_ligne),'(E1:E2));')]);
10  end
11 end

```

Les données de l'option sont indiquées par des `nom_matrice` auxquels sont associées les lignes à prendre en compte dans les `nom_matrice`. Ces lignes correspondent aux numéros de sujets. Chaque `nom_matrice` renvoie à `nombre_electrodes` matrices. Donc pour une ligne/sujet on a les données de chaque électrode : pour chaque ligne/sujet on a `nombre_electrodes` lignes de données. On place ces `nombre_electrodes` lignes de données dans la matrice `INDI` **numéro_sujet** à raison d'une ligne de données par ligne de la matrice `INDI` **numéro_sujet**.

$$INDI_numéro_sujet = \begin{matrix} & \begin{matrix} 1 & \dots & tps_max \end{matrix} \\ \begin{matrix} 1 \\ \dots \\ nombre_electrodes \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Une fois que cette matrice est remplie avec les données de chaque électrode, on en calcule la moyenne sur les lignes/électrodes. On récupère donc un vecteur

ligne que l'on place dans la variable `MINDInuméro_sujet`.

$$MINDI_numéro_sujet = \begin{matrix} & 1 & \dots & tps_max \\ 1 & (ddp & \dots & ddp) \end{matrix}$$

Puis on somme les valeurs de `MINDInuméro_sujet` comprises entre E1 et E2. Et on place le résultat de cette somme dans `tableau2` à la ligne `numéro_option`, à la colonne `numéro_sujet`. `Tableau2` est composé de façon similaire à `tableau1`.

$$tableau2 = \begin{matrix} & 1 & \dots & nb_sujets \\ 1 & \left(\begin{matrix} aire\ ou\ NaN & \dots & aire\ ou\ NaN \\ \vdots & \ddots & \vdots \\ aire\ ou\ NaN & \dots & aire\ ou\ NaN \end{matrix} \right) \\ \dots & & & \\ nb_option & & & \end{matrix}$$

Cas 3

Si le tag '`elec_aire_pasmoy`' contient 1 : On crée la variable globale `tableau3`. Parmi toutes les options, si une option est cochée dans l'interface1 et qu'elle concerne tous les sujets de l'expérience, on réalise les étapes suivantes :

```

1 for j_mat_ds_option=1:nb_matrice_par_option(i_option)
2   if any(str2num(valeur_num_indiv)==info{i_option}{3}){←
      j_mat_ds_option})==1 %si le sujet sélectionné est dans←
      la liste de sujets de la sous-matrice
3   for elec = liste_electrodes
4     eval(['tableau3(i_option, ' num2str(elec) ')=nansum(' ←
      info{i_option}{2}{j_mat_ds_option} num2str(elec) ' ←
      (' valeur_num_indiv ',E1:E2));'])
5   end
6 end
7 end

```

Les données de l'option sont indiquées par des `nom_matrice` auxquels sont associées les lignes à prendre en compte dans les `nom_matrice`. Ces lignes correspondent aux numéros de sujets. Chaque `nom_matrice` renvoie à `nombre_electrodes` matrices. Donc pour une ligne/sujet on a les données de chaque électrode : pour chaque ligne/sujet on a `nombre_electrodes` lignes de données. On s'intéresse à la ligne/sujet qui correspond au sujet sélectionné dans l'interface1. On prend les `nombre_electrodes` lignes de données de cette ligne/sujet. Pour chacune d'entre elles (c'est-à-dire pour chaque électrode), on calcule la somme des valeurs comprise des colonnes E1 à E2 et on place le résultat dans `tableau3` à la ligne `numéro_option`, à la colonne `numéro_électrode`.

`Tableau 3` sera donc une matrice contenant les calculs d'aire. Ses lignes correspondent aux numéros d'options, ses colonnes correspondent aux numéros d'électrodes.

$$tableau3 = \begin{matrix} & 1 & \dots & nombre_electrodes \\ 1 & \left(\begin{matrix} aire\ ou\ NaN & \dots & aire\ ou\ NaN \\ \vdots & \ddots & \vdots \\ aire\ ou\ NaN & \dots & aire\ ou\ NaN \end{matrix} \right) \\ \dots & & & \\ nb_option & & & \end{matrix}$$

Cas 4

Si le tag 'elec_aire_moy' contient 1 : On crée la variable globale tableau4. Parmi toutes les options, si une option est cochée, on réalise les étapes suivantes :

```

1 for elec = liste_electrodes
2     ligne=1;
3     for j_mat_ds_option=1:nb_matrice_par_option(i_option) %↔
4         pour chaque sous matrice de cette option
5             for sous_ligne=info{i_option}{3}{j_mat_ds_option} %pour ↔
6                 chaque sujet (sous_ligne) à récupérer dans cette ↔
7                 matrice
8                 eval(['matrice' num2str(i_option) '_' num2str(elec) '('↔
9                     ligne,:)=' info{i_option}{2}{j_mat_ds_option} ↔
10                     num2str(elec) '(sous_ligne,:)');
11                 ligne=ligne+1;
12             end
13         end
14     end
15
16     eval(['matrice_moy' num2str(i_option) '_' num2str(elec) '('↔
17         nanmean(matrice' num2str(i_option) '_' num2str(elec) '↔
18         (:,:));']);
19
20     eval(['tableau4(i_option,elec)=nansum(matrice_moy' num2str↔
21         (i_option) '_' num2str(elec) '(E1:E2));']);
22 end

```

Pour chaque électrode on réalise le travail suivant : On prend les données de l'option qui concernent cette électrode via les étapes qui suivent. Pour chaque nom_matrice de l'option (info{numéro_option}{2}) : à partir du nom_matrice on choisit la matrice contenant les données de l'électrode, et dans cette matrice on récupère les lignes/sujets à prendre en compte pour ce nom_matrice (info{numéro_option}{3}) que l'on place dans la matrice matricenuméro_option_numéro_électrode. matricenuméro_option_numéro_électrode contient donc autant de lignes qu'il y a de lignes/sujets à récupérer dans les nom_matrice de l'option (sans se soucier de garder la correspondance entre ses numéro de ligne et les numéro de sujet). matricenuméro_option_numéro_électrode=

$$\begin{matrix} & 1 & \dots & tps_max \\ \begin{matrix} 1 \\ \dots \\ \text{nombre sujets concernés par l'option} \end{matrix} & \begin{pmatrix} ddp & \dots & ddp \\ \vdots & \ddots & \vdots \\ ddp & \dots & ddp \end{pmatrix} \end{matrix}$$

Une fois toutes les données récupérées pour l'électrode, on calcule la moyenne sur les lignes de matricenuméro_option_numéro_électrode. On obtient donc un vecteur ligne que l'on place dans la variable matrice_moynuméro_option_numéro_électrode.

$$\text{matrice_moynuméro_option_numéro_électrode} = \begin{matrix} & 1 & \dots & tps_max \\ 1 & (ddp & \dots & ddp) \end{matrix}$$

Puis on fait la somme des valeurs de cette matrice contenues dans les colonnes E1 à E2. On place le résultat obtenu dans tableau4 à la ligne nb_option, à la

colonne numéro_électrode. Tableau4 est composé de façon similaire à tableau3.

$$tableau4 = \begin{matrix} & 1 & \dots & nombre_electrodes \\ \begin{matrix} 1 \\ \dots \\ nb_option \end{matrix} & \begin{pmatrix} aire\ ou\ NaN & \dots & aire\ ou\ NaN \\ \vdots & \ddots & \vdots \\ aire\ ou\ NaN & \dots & aire\ ou\ NaN \end{pmatrix} \end{matrix}$$

Dans les quatre cas : A chaque fois que les aires ont été calculées pour une option et donc qu'une ligne de tableau1, 2, 3 ou 4 a été remplie : cette ligne est affichée dans l'axe. De plus, un 1 est inscrit (à la place du 0) dans la colonne numéro_option du vecteur pos_couleur. Et deux uicontrols text de tags 'legendnuméro_option' et 'affichenuméro_option' sont créés. Une fois qu'un tableau 1, 2, 3 ou 4 est terminé, le champ 'ButtonDownFcn' de l'axe est rempli de sorte que la fonction affiche_aire_global() soit appelée lorsque l'utilisateur clique dans l'axe avec la souris. Cette fonction prend en paramètre la matrice qui vient d'être créée : tableau1, tableau2, tableau3 ou tableau4.

Remarque également valable pour les matrices tableaux 2, 3 et 4 : Tableau1, à son initialisation, est rempli de NaN. Puis, lorsque nous remplissons tableau1 avec les aires, certaines lignes (options) sont ignorées. Par exemple si l'option numéro 2 doit être affichée mais pas l'option numéro 1, alors la ligne 1 de tableau1 n'est pas remplie par la fonction mais la ligne 2 l'est. Ces lignes ignorées contiennent donc des NaN qui ne seront pas affichés par la fonction plot. Pour poursuivre l'exemple précédent, la ligne 1 contiendra des NaN. Cela est aussi valable pour les colonnes (sujets ou électrodes) qui ne seraient pas remplies.

7.20 Affiche_aire_global(tableau)

Lorsque l'utilisateur clique sur l'axe, la fonction affiche_aire_global(tableau) affiche les valeurs des points ayant la même 'abscisse' que l'endroit où il a cliqué, 'l'abscisse' de l'endroit où il a cliqué étant arrondi à un entier.

Il affiche ces valeurs dans l'uicontrol de tag 'affichenb_option'. Le programme retrouve ces valeurs dans la matrice tableau. Il sait quelles options regarder (quelles lignes regarder dans tableau) grâce au vecteur pos_couleurs créé dans la fonction perspective_oiseaux_arrivee(detache_graphe_global) qui indique quelles options sont affichées dans l'axe. La colonne de tableau à choisir lui quant à elle, est indiquée par 'l'abscisse' de l'endroit où l'utilisateur a cliqué.

Table des figures

2.1	Déroulement de l'expérience vigilance-musique	10
5.1	Interface Fixe	16
5.2	Détacher la figure Fixe	18
5.3	Carte topographique Fixe	19
5.4	Imagesc Fixe : fenêtre 1	21
5.5	Imagesc Fixe : fenêtre 2	21
5.6	Maps Fixe	22
6.1	Interface 0	26
6.2	Interface 1 Fixe : options	26
6.3	Interface 1 Defixe	28
6.4	Boîte de dialogue load Defixe	29
6.5	Détacher la figure Defixe	31
6.6	Vue globale Defixe : Aire sous la courbe en fonction du sujet . .	32
6.7	Vue globale Defixe : Aire sous la courbe en fonction de l'électrode	33
6.8	Carte topographique Defixe	34
6.9	Carte topographique Defixe pour l'expérience vigilance-musique .	35
6.10	Vue 3D Carte topographique Defixe	36
6.11	Vue 3D Carte topographique Defixe pour l'expérience vigilance- musique	36
6.12	Imagesc Defixe : fenêtre 1	37
6.13	Imagesc Defixe : fenêtre 2	38
6.14	Rotation 3D Imagesc Defixe : fenêtre 2	38
6.15	Renseignement positions électrodes	39
7.1	Architecture des fichiers/fonctions de l'interface Defixe	41
7.2	Architecture de la cellule info	44
7.3	Cellule info : accès aux différents champs	45

Liste des tableaux