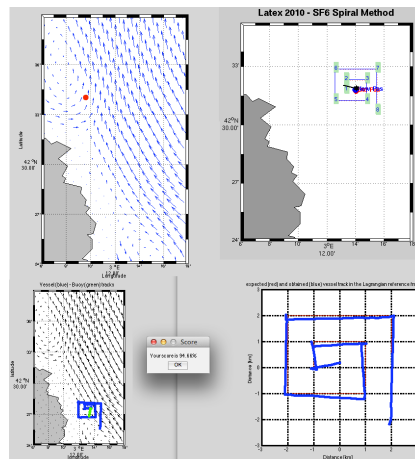

Rapport de stage : Développement d'un programme informatique de navigation Lagrangienne

Louise Rousselet

Maître de stage : Mr Andrea M. DOGLIOLI

Université d'Aix-Marseille, Mediterranean Institute of Oceanography (MIO) ;
Université du Sud Toulon-Var ; CNRS-INSU/IRD UM 110 ;
France.



7 mai 2013

Table des matières

1	Introduction	3
1.1	Objectifs	4
2	Matériel et méthode	4
2.1	Les logiciels utilisés	4
2.2	Les fonctions	5
2.3	Les données	5
3	Résultats	6
3.1	Mis à jour du package	6
3.2	Mis à jour de la Demo	8
3.3	Parution du package sur Internet	10
3.3.1	Les dernières modifications	10
3.3.2	Installation du paquet	11
3.3.3	Faire tourner la démo : Lagrangian_Navigation_Demo	12
4	Discussion	15
5	Conclusion	17
6	Annexes	20
A	Fichier Contents.m mis à jour	20
B	Brouillon de script	22
C	Script spiral_calculation	23
D	Fichier readme.txt	25

Résumé

Pour étudier la dispersion d'un traceur passif à la surface d'une masse d'eau, il est nécessaire d'échantillonner la surface dans un référentiel Lagrangien en mouvement avec la structure étudiée. En effet la masse d'eau étant en mouvement constant, il faut naviguer en la suivant tout en l'échantillonnant. Les chercheurs du MIO ont donc développé un programme qui permet à l'utilisateur de calculer le cap que doit suivre le bateau pour échantillonner le plus précisément possible la zone où le traceur se disperse. Pour cela ils définissent le centre du référentiel Lagrangien par une bouée Lagrangienne relâchée au centre de la masse d'eau à étudier, ensuite le bateau doit suivre une spirale théorique. Le package LATEXtools_0.0 , créé pour MATLAB et qui regroupent tous les fichiers nécessaires au calcul de cette trajectoire, a été conçu en 2010. Cependant cette version n'était pas aboutie et de nombreuses modifications devaient être apportées pour le rendre utilisable par n'importe quel personne qui souhaiterait échantillonner une masse d'eau. Il a donc fallu corriger toutes les erreurs de fonctionnement et les imperfections, modifier des calculs pour les rendre plus précis ou encore épurer le package afin de le rendre plus simple et compréhensible. Toutes les étapes de cette mise à jour, jusqu'à la publication sur internet, sont détaillées dans ce rapport.

Abstract

To study dispersion of a passive tracer, it is necessary to sample the water surface in a Lagrangian reference frame moving with the structure. Indeed the waterbody is constantly moving, so the boat must follow it and sample it at the same time. MIO searchers have developed a program which allows users to calculate the cap the boat has to follow to sample the most precisely the area where tracer has been released. To do that, they define the center of the Lagrangian reference frame with a Lagrangian buoy released in the middle of the waterbody, then the boat has to follow a theoretical spiral. The LATEXtools_0.0 package, designed for MATLAB which includes all essential files to calculate this trajectory, has been created in 2010. Meanwhile this version is not complete and many changes needed to be made to make it usable for any one who would sample a waterbody. Correct performance mistakes and imperfections, modify calculations to make them more precise or even remove every things which are not necessary to make the package easier and more understandable, had to be done. Each update step, until the publication on internet, are detailed in this report.

1 Introduction

Des études pionnières ont montré l'importance d'une stratégie Lagrangienne d'échantillonnage pour analyser la dispersion d'un traceur. Le projet IronEx (Law et al., 1998; Stanton et al., 1998; Coale et al., 1998) a été le premier suivi de nombreux autres comme PRIME (Law et al., 2001), SOIREE (Boyd and Law, 2001), CYCLOPS (Law et al., 2005), SEEDS (Tsumune et al., 2005), SERIES (Law et al., 2006) et SEEDS II (Tsumune et al., 2009). En effet dans le cas de dispersion, seules les mesures réalisées dans un référentiel Lagrangien, c'est à dire qui est en mouvement avec l'élément, permettent de corriger les effets d'advection de l'eau et de pouvoir estimer correctement la dispersion. Par exemple lors de la campagne IronEx, le système de navigation Lagrangienne développé par les chercheurs permettait de corriger la trajectoire du bateau pendant la dispersion du traceur dans l'eau par rapport à la dérive du courant. Pour cela le centre du référentiel Lagrangien de référence devait être défini. Ils ont utilisé une bouée Lagrangienne munie d'un Système de Position Globale (GPS) relié à une radio présente à bord du bateau. Un logiciel a été créé pour cette campagne afin d'afficher simultanément à bord la position du bateau et de la bouée superposée à la grille d'échantillonnage.

Toutes les études suivantes ont été menées en utilisant la technique développée lors du projet IronEx avec parfois quelques modifications. Cependant les recherches sur le développement d'une technique d'échantillonnage dans un référentiel Lagrangien restent assez pauvres malgré la nécessité d'une telle méthode pour le succès de telles expériences sur le terrain.

Le but du projet LATEX était de marquer "une caractéristique dynamique de mésoéchelle" à l'aide d'un traceur passif (Sulfure Hexafluoride-SF6) associé à des bouées lagrangiennes afin d'étudier les dynamiques physiques et biogéochimiques couplées à méso et subméso échelle sur les transfères de matière et de chaleur entre les zones côtières et le large. Pour suivre le mélange turbulent et la dispersion du traceur il était nécessaire de constamment réajuster la trajectoire du bateau pour rester le plus proche possible du cadre de référence Lagrangien en mouvement avec la structure étudiée. Les chercheurs du MIO ont donc cherché à créer un logiciel plus performant que celui développé lors des précédentes études sur la dispersion d'un traceur, qui permettrait de calculer la trajectoire que doit suivre le bateau. Pour cela les chercheurs ont mené quatre campagnes en mer dans la partie Ouest du Golf du Lion entre 2007 et 2008. Deux des quatre campagnes ont été dédiées à l'étude de dispersion du traceur (SF6). Elles ont abouti au développement du logiciel 'LATEXtools_0.0' dont la dernière version datait de 2010.

1.1 Objectifs

Le but de ce stage était donc de tester cette version afin de la rendre utilisable par tous ainsi que de développer la démo du logiciel qui n'était ni fonctionnelle ni abouti à l'époque. Après avoir effectué ce travail il était nécessaire de créer un formulaire afin que chaque utilisateur puisse comprendre le fonctionnement du logiciel et le prendre en main rapidement : Userguide.

2 Matériel et méthode

Comme il s'agit d'un logiciel informatique développé pour MATLAB, la totalité du travail s'est effectué sur ordinateur et sur ce logiciel. Une deuxième partie du travail a été réalisé sur le logiciel LaTeX pour la création du 'Userguide'. De nombreuses données expérimentales ont été utilisées pour la création des cartes pour avoir une démo conforme à la réalité.

2.1 Les logiciels utilisés

L'intégralité du travail a été effectué sur un ordinateur portable personnel MacBook Pro. Il était en effet intéressant de tester cette démo et d'arranger le paquet sur différentes plateformes afin de le rendre utilisable par tous. Le paquet a donc été créé sous Linux, re-travaillé sous Linux et Mac OS et testé a posteriori sous Windows.

Cependant pour chacune de ces trois plateformes le logiciel MATLAB reste indispensable dont les versions testées vont du R2010a au R2011b. L'environnement de développement choisi était donc MATLAB (matrix laboratory) car il permet la création de "boîtes à outils" dont LATEXtools fait aujourd'hui parti. MATLAB est donc un langage de quatrième génération qui permet des calculs numériques plus rapides et un encodage simplifié. Pour créer LATEXtools les séquences de commandes ont été sauvegardés dans un fichier texte, qui avec l'éditeur MATLAB prend la forme d'un "script", ou dans une fonction. Ces scripts et fonctions sont réunis dans un dossier et l'ensemble forme le "package LATEXtools_0.0" .

Une seule boîte à outils de MATLAB a été utilisé : `m_map`. Cet "toolbox" permet, simplement à partir de données analysées sous MATLAB, de faire différents type de graphiques et notamment des cartes en coordonnées latitude/longitude. Cet outils contient aussi de nombreuses fonctions qui sont utilisées dans LATEXtools.

Le deuxième logiciel utilisé est un système de composition de document, qui facilite l'utilisation de mise en page de texte, LaTeX. Sous Mac OS X la distribution est plus précisément MacTeX. Il a été choisi pour sa relative simplicité dans l'écriture et la mise en page de documents scientifiques. Pour créer un document sous LaTeX, il faut créer un fichier *.tex dans lequel l'utilisateur utilise des commandes pour mettre en forme son texte. Les commandes sont généralement de la forme : `\nomdelacommande{texte sur lequel la commande doit s'appliquer}`. Par exemple pour mettre un texte en gras il faut taper la commande `\textbf{texte à mettre en gras}`. LaTeX possède donc son propre système d'encodage.

De plus il permet la gestion automatique d'une bibliographie grâce à l'outil auxiliaire BibTeX. La description des ouvrages - titre, auteurs, année de parution...- est écrite dans un fichier extérieur unique auquel se réfèrent les documents LaTeX. Cela limite les erreurs dues à une réalisation "à la main".

2.2 Les fonctions

De nombreuses fonctions ont été créées par les concepteurs de LATEXtools cependant les fonctions intégrées dans MATLAB et dans m_map ont aussi été très utiles. Celles qui ont le plus servi sont principalement des fonctions qui permettent de passer d'une coordonnées à une autre. La première `m_xy2ll`, et son opposée `m_ll2xy`, permet de convertir des coordonnées cartésiennes en coordonnées géographiques, et inversement. La deuxième est la fonction `cart2pol`, et son opposé `pol2cart`, qui transforme des coordonnées cartésiennes en coordonnées polaires c'est-à-dire un angle et un rayon, et inversement.

Ensuite le calcul de distance en coordonnées géographiques étant compliqué, les fonctions `m_lldist`, `m_fdist` et `m_idist` ont permis de réaliser ces calculs rapidement. En effet la fonction `m_fdist` retourne les longitude, latitude et azimuts d'un point à la surface de la terre calculé à partir d'un autre point et de la distance entre les deux, en utilisant une version vectorisé de l'algorithme de Vincenty. `M_idist` effectue le même travail dans le sens inverse, elle retourne la distance entre les deux points et l'azimut "allé" et l'azimut "retour". Quant à `m_lldist`, elle permet le calcul de distance entre deux points de coordonnées géographiques.

Enfin la dernière fonction importante de la démo est la fonction `rand` qui permet le tirage aléatoire de nombre entre 0 et 1 suivant une loi uniforme.

2.3 Les données

Pour la création d'un tel logiciel il était nécessaire que des données réelles soit utilisées afin de reproduire un mouvement le plus proche de la réalité. C'est pourquoi durant la campagne Latex10 en 2010 des données de vitesse de courant

ont été mesurées par système ADCP (Acoustic Current Meter Profiler) un courantomètre acoustique. Les positions géographiques des bouées et du bateau ont été relevés à chaque expérience. La concentration de sulfure Hexafluoride (SF₆) a également été mesurées lors de chaque sorties. Des données de température de surface provenant de Météo France ainsi que la bathymétrie du Nord-Ouest de la Méditerranée, ont aussi été nécessaires à l'élaboration des cartes géographiques.

Enfin le champ de vitesse intégré au modèle de la démo provient d'une simulation numérique de Nord-Ouest de la Méditerranée (Hu et al., 2009).

3 Résultats

Le logiciel LATEXtools a été mis à disposition sur internet le 21 Février 2013. Cette version contient tous les fichiers mis à jour ainsi qu'une démo fonctionnelle. Elle constitue à ce jour la version officielle du logiciel.

Dans ce chapitre sont décrites toutes les étapes qui ont conduit à l'élaboration de ce package informatique ainsi que les informations nécessaires au bon fonctionnement de la démo développé lors de ce stage. Ces informations ont été développées dans les deux documents fournis avec le paquet lors de son téléchargement : un fichier 'readme.txt' utile lors de l'installation et le démarrage du paquet et un document 'Usersguide.pdf' qui relate toutes les informations sur le paquet ainsi qu'un tutorial de la démo. Ces fichiers ont également été réalisés lors de ce stage. La version rédigée en anglais du fichier 'readme.txt' est disponible en Annexe cependant le document 'Userguide.pdf' étant trop long n'est pas inclus dans ce rapport, il est toutefois disponible dans le paquet LATEXtools en téléchargement.

3.1 Mis à jour du package

La première étape fut la prise en main de tous les scripts contenus dans le package afin de comprendre leur utilité. Ce travail a été l'un des plus long car il a fallu lire des centaines de lignes de code. Les problèmes les plus fréquemment rencontrés, pour faire fonctionner les scripts, ont été des problèmes de compatibilité entre les systèmes Linux/Mac/windows notamment pour le chargement de données. Souvent pour ces trois systèmes les chemins peuvent être différemment écrit "/" ou "\" pour d'autres. Il a donc fallu automatiser cela pour que n'importe quel système d'exploitation soit capable de faire tourner les scripts de LATEXtools.

La deuxième étape a été le référencement de tous les fichiers de données nécessaires au bon fonctionnement du logiciel, afin de les intégrer dans le package pour

que toutes les figures puissent s'afficher sans erreur. En effet tous les scripts n'ont pas été fait par la même personne ce qui a posé quelques problèmes car certains chargeaient des fichiers présent sur leur disque dur sans les intégrer dans le paquet. Il était donc impossible de faire tourner le scripts sur un autre ordinateur.

Une fois tous les fichiers manquants ajoutés, les scripts fonctionnaient plus ou moins bien. Le travail à ce moment là consistait à lister tous les scripts et fonctions disponibles dans le paquet pour mettre à jour le fichier "Contents.m" , disponible en annexe, qui permet à chaque utilisateur d'avoir un petit descriptif de chaque script. Ce fichier est en quelque sorte le sommaire du package, il était donc très important que celui ci soit à jour afin que chaque personne qui utilisait LATEXtools sache à quoi correspondent les scripts et ce qu'ils font. C'est ici que la première étape du travail a été la plus utile. Le tout devait être rédigé en anglais pour une compréhension de tous à travers le monde. A cela s'est ajouté le changement de tous les commentaires et noms de variables en français en anglais. En effet lorsqu'un script est rédigé il est toujours plus facile de commencer en donnant des noms, aux variables, dans la langue natale pour avoir un descriptif de la variable immédiat. Travail long et fastidieux car après chaque changement de nom de variables il fallait vérifier que le script fonctionnait, ce qui n'était pas toujours le cas généralement à cause de mêmes variables inchangées. De plus comme de nombreux scripts font appel à d'autres dans leur code il a fallu vérifier que les variables étaient bien les mêmes dans les autres scripts où elles étaient utilisées.

Un script 'load.AVHRR' a du être créé pour charger les fichiers AVHRR.png afin de respecter la méthode du paquet (Fig. 1). En effet chaque fichiers nécessaires à l'utilisation sont chargés grâce à un script de la forme 'load.nom du fichier à charger'. Il était important, même si le package n'utilise qu'un seul fichier AVHRR, qu'il soit chargé de la même façon que les autres fichiers et non pas chargé au milieu d'un script. Tout cela toujours dans le but de faciliter la compréhension de l'utilisateur.

```
%Script for load AVHRR .png file
function [p_png,f_png,imc4] = load_AVHRR(dirin,filename)

f_png=filename ; p_png=dirin ;
imc4=imread([p_png,f_png]) ; imc4=double(imc4) ;
end
```

FIGURE 1 – Script pour charger les fichiers AVHRR.

Une fois ce travail de prise en main et de mise en forme du paquet l'arborescence des répertoires était très différente de la précédente et est devenue la

suivante :

```
LATEXtools_0.0
|- ADCP
|- ARGOS
|- AVHRR
|- BATHYMETRY_NWMED
|- Documentation
|   |- Userguide
|-figures
|- FSLE
|   |-MatFiles_*
|- Log_Position
|- m_map
|   |-private
|- other_stuffs
|- SF6
|- Tethys
|- VESSEL
```

3.2 Mis à jour de la Demo

Le principe d'une démo pour un tel logiciel est de simuler une navigation Lagrangienne dans un champ de courant quelconque en prenant comme point de référence une bouée Lagrangienne dont le mouvement a été simulé grâce à un modèle de "marche aléatoire". La première idée a été de créer un champ de vitesse très simple et de faire dériver la bouée simplement avec le champ de vitesse. Cependant la représentation sur une carte a été assez compliqué étant donné qu'il fallait constamment passer des coordonnées cartésiennes (x,y) pour effectuer les calculs, aux coordonnées géographiques (longitude , latitude) pour la représentation graphique. Un bon nombre d'erreurs pour des petits détails de fonctionnement de fonction et de taille de matrices incompatibles ont fait perdre beaucoup de temps. Pour éviter une encore plus grande perte de temps une simulation numérique dans le Nord Ouest de la Méditerranée (Symphonie model, Hu et al., 2009) a été utilisé afin d'obtenir un champ de vitesse qui se rapproche de la réalité. Le brouillon du premier script est disponible en annexe.

Une fois cette étape réglée, le travail a consisté à tester la nouvelle version de la démo améliorée par Mr Doglioli afin de la rendre la plus simple et compréhensible possible, en notant chaque erreur et en les modifiant. Cette nouvelle version était déjà bien aboutie, seuls quelques petits bug persistaient notamment lorsque

le système d'équation ballistique n'avait aucune solution le programme s'arrêtait. Il a donc fallu ajouter des messages d'erreurs pour informer les utilisateurs du problème et continuer la navigation. De plus lorsque l'utilisateur entrait une valeur invalide pour le programme, la démo cessait automatiquement. Il fallait donc mettre en place une boucle 'while' pour que le programme continue à demander une valeur à l'utilisateur tant qu'elle n'était pas comprise dans la gamme de valeur par défaut du programme.

Un morceau de script a aussi été ajouté pour calculer le score de l'utilisateur (Fig. 2). Ce score est un pourcentage de réussite à la navigation, c'est-à-dire que la trajectoire du bateau, géré par l'utilisateur pendant la simulation, doit être la plus proche de la trajectoire théorique que le bateau doit suivre pour échantillonner la masse d'eau dans laquelle le traceur est relâché. Cette trajectoire théorique correspond à une spirale carrée à plusieurs coins. Plus la masse d'eau à échantillonner s'étalera sur une grande distance plus la spirale sera grande et possèdera un nombre de coins élevé.

```

% Calculation of distance made by the boat and theoretical spirale distance

% Spirale
cumulative_distance=totdist/1000; % [km]

% Boat
size_VesselLonMemo=size(VesselLonMemo);
cumulative_distance_vessel=0;
for i=1 :size_VesselLonMemo(2)-1;
cumulative_distance_vessel=cumulative_distance_vessel...
...+m_lldist(VesselLonMemo(i :i+1),VesselLatMemo(i :i+1));
end

if target_number==size_lonspir(2)
% Lagrangian navigation score : 100% = boat follow exactly the spiral.
% The greater the cumulative distance of the boat,the lower the score.
score=100-abs(cumulative_distance-cumulative_distance_vessel)/cumulative_distance*100;
waitfor(msgbox(['Your score is ', num2str(score,'%2.2f' ),'%'], 'Score' ));

figure(4) text(max(xx)+1,1,' Score', 'FontSize',20)
text(max(xx)+1,0,[' ',num2str(score,'%2.2f' ),'%'], 'FontSize',20)
else
waitfor(msgbox(['I can not calculate your score since you do not navigate until the last spiral point'], 'Score' ));
end

saveas(3,[figdir,'lagrangian_navigation_demo_lonlat.png'])

saveas(4,[figdir,'lagrangian_navigation_demo_lagframe.png'])

break

```

FIGURE 2 – Calcul du score du pilote Lagrangien inclus dans le script Lagrangian_reconstitution_demo.

Enfin la dernière étape lorsque la démo fonctionnait correctement fut la suppression des fichiers inutiles au roulement du logiciel afin de le rendre moins 'lourd' au téléchargement. Deux scripts qui étaient quasi-identiques, `spiral_calculator.m` et `spiral_calculation.m`, ont été réunis en un seul afin d'éviter la surcharge de scripts dans le paquet.

3.3 Parution du package sur Internet

Avant la parution du package sur internet tout un protocole doit être mis en place pour éviter des problèmes de fonctionnements et pour donner les dernières retouches à la démo. Ensuite dans ce chapitre sera développé le principe d'installation et de fonctionnement mis en place.

3.3.1 Les dernières modifications

Avant le dépôt du programme sur internet toutes les personnes ayant participé de près ou de loin à l'élaboration du package ont été invitées à tester la démo pour vérifier qu'elle était fonctionnelle sur des appareils différents et pour apporter un regard un peu plus extérieur afin de faire toutes les remarques susceptibles d'améliorer la démo. Plusieurs relectures du Usersguide ont permis de corriger les petites fautes d'anglais qui persistaient ainsi que les inexactitudes et imperfections dans les noms des dossiers ou fichiers. En effet il fallut être très rigoureux dans l'écriture pour respecter à la majuscule près les noms des différents objets du package.

De plus une remarque intéressante a permis d'ajouter une fonction à une fenêtre de "dialogue" afin que la valeur entrée par l'utilisateur, dans la fenêtre, devienne la valeur par défaut à chaque réouverture de la fenêtre de "dialogue". En effet cette fenêtre demande à l'utilisateur la valeur du coin de la spirale visé par le "navigateur". En effet pour que le bateau ait une trajectoire la plus proche possible de la spirale, il doit prendre un certain cap calculé à partir de la position géographique du prochain coin de la spirale théorique que le bateau doit atteindre. Il est nécessaire de maintenir ce cap jusqu'à ce que le bateau atteigne le point de la spirale car il ne rejoint pas ce point en une fois. En effet la position du bateau est calculée toutes les 10 secondes. Il est donc admis que le bateau ne peut rejoindre le coin visé en seulement 10 secondes. Il faut donc, qu'à chaque fois qu'une nouvelle position du bateau est calculée, c'est-à-dire toutes les 10 secondes, l'utilisateur entre la valeur du coin de la spirale à atteindre. Or pour faire toute une simulation de navigation il est nécessaire de passer par tous les coins de la spirale (ici 9) et chaque coin n'est atteint qu'au bout de x itérations, proportionnellement à la distance entre le point précédent et le point visé. Ce changement fait donc gagner du temps à l'utilisateur qui n'est pas obligé de retaper à chaque itération le chiffre

du point visé mais peu seulement changer le chiffre lorsque le point visé a été atteint et qu'il faut viser le coin suivant. La figure suivante éclaircit ce mécanisme (Fig. 3)

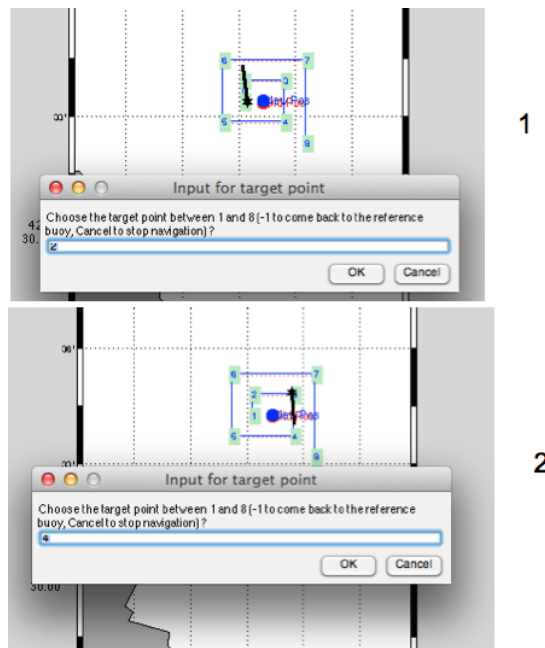


FIGURE 3 – La valeur entrée par l'utilisateur devient la valeur par défaut. Sur la figure 1, le point visé par l'utilisateur est la point 2, la fenêtre de dialogue propose à la réouverture le point 2 par défaut. Le principe est le même pour la figure 2. L'utilisateur a choisi le point 4 au tour précédent, la fenêtre lui propose alors le point 4 par défaut. Il ne changera la valeur de la boîte de dialogue que lorsque le bateau (en noir) aura atteint le point 4. Il visera alors le point 5 etc...

3.3.2 Installation du paquet

Ce programme a donc été conçu pour MATLAB et nécessite au moins 32Mbytes d'espace disque. Il a été testé sur des versions de MATLAB allant de R2010a à R2011b mais il peut être opérationnel sur des versions antérieures et plus récentes. Il fonctionne sous Linux, Windows et Mac. Pour éviter des problèmes de fonctionnement le paquet 'm_map' nécessaire a été inclu dans LATEXtools_0.0 afin que chaque utilisateur ne soit pas obligé de le télécharger sur internet si il ne le possède pas déjà. Ainsi l'utilisation est simplifiée et directe.

Pour installer le paquet il suffit de le télécharger gratuitement à partir du site

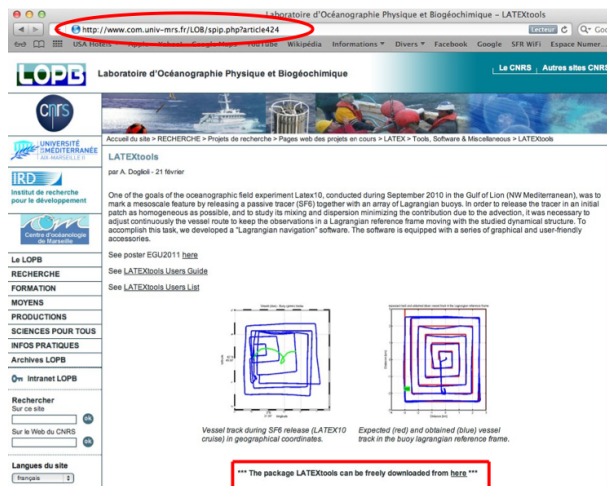


FIGURE 4 – Site du projet LATEX

internet du projet LATEX : <http://www.com.univ-mrs.fr/LOB/spip.php?article424> (Fig. 4). Un dossier compressé est téléchargé. La décompression fait apparaître le dossier LATEXtools_0.0 qui contient tous les programmes, données et figures du paquet. Pour l'utiliser, il faut ouvrir une session MATLAB et se placer dans le répertoire contenant le dossier LATEXtools_0.0 : Current folder : `*/LATEXtools_0.0`. Le logiciel est fin prêt à être utilisé.

Un 'Usersguide' ('Guide d'utilisation') est mis à disposition dans le paquet si l'utilisateur rencontre un quelconque problème. De plus MATLAB contient une grande section d'aide appelé 'Help' qui peut aider chaque utilisateur.

3.3.3 Faire tourner la démo : Lagrangian_Navigation_Demo

Lorsque la démo démarre une première figure s'affiche à l'écran représentant une petite portion de la côte de l'Ouest du Golf du Lion et des vecteurs vitesse des courants présents dans cette zone (Fig. 5). L'utilisateur doit alors choisir l'endroit de mise à l'eau de la bouée de référence en cliquant sur la carte. A cet instant, il est recommandé à l'utilisateur de choisir un point proche du centre de la carte afin d'éviter que la bouée ne sorte de la carte, emportée par le courant, pendant la démo.

La boîte de dialogue suivante demande de choisir le point visé de la spirale, entre 1 et le nombre maximum de point que compte la spirale. Dans cette démo la spirale présente 8 points par défaut. De plus la valeur -1 permet de voir les instructions de direction pour revenir à la bouée de référence. Le bouton "Cancel" (ou Tab+Entrée) arrête la démo. Pour commencer la simulation de navigation il faut choisir le premier point de la spirale et cliquer sur "OK". A ce stade deux

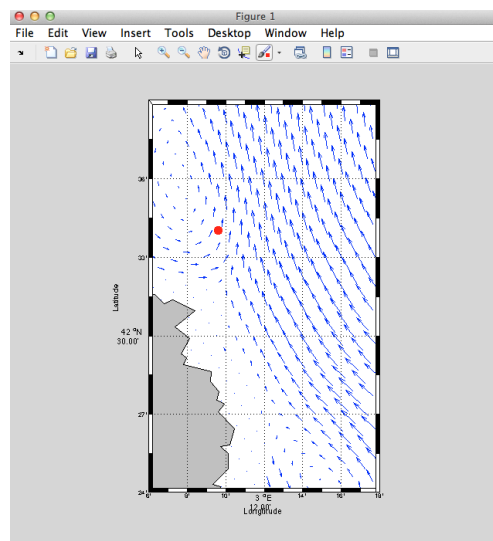


FIGURE 5 – Première figure de la démo. Les vecteurs du champ de vitesse du courant et la côte Ouest du Golf du Lion sont représentés. Le point rouge représente le point choisi par l'utilisateur pour la mise à l'eau de la bouée.

figures apparaissent à l'écran (Fig. 6). Sur la première figure sont représentés : i) la dernière et avant-dernière position de la bouée de référence (points bleu et rouge, respectivement) ; ii) la trajectoire de la spirale, et les différents coins associés aux points visés par le bateau ; iii) la position du bateau (étoile noire) ; iv) le vecteur, représentant la direction que le bateau doit suivre (ligne noire), calculé pour atteindre le point visé. La deuxième figure permet d'afficher les informations concernant la dérive de la bouée et les instructions de navigation du bateau. Enfin, une nouvelle boîte de dialogue apparaît pour choisir le prochain point de la spirale à atteindre.

La navigation Lagrangienne se poursuit jusqu'à ce que l'utilisateur arrête la démo ("Cancel"). Une fois la démo arrêtée, deux nouvelles figures apparaissent ainsi qu'une nouvelle boîte de dialogue (Fig. 7). Ces figures sont issues du programme `lagrangian_reconstitution_demo.m`. La 3ème figure MATLAB représente les trajectoires du bateau (en bleu) et de la bouée de référence (en vert) dans les coordonnées géographiques. La figure 4 dessine la trajectoire théorique (en rouge) et obtenue (en bleu) du bateau dans le référentiel Lagrangien. La boîte de dialogue affiche le score de l'utilisateur comme "Pilote Lagrangien". Ce score est calculé en comparant la longueur de la spirale théorique et celle de la route du bateau. Plus le score se rapproche de 100.00% plus la route du bateau "colle" celle de la spirale théorique. Cependant si l'utilisateur arrête la navigation avant d'avoir atteint le

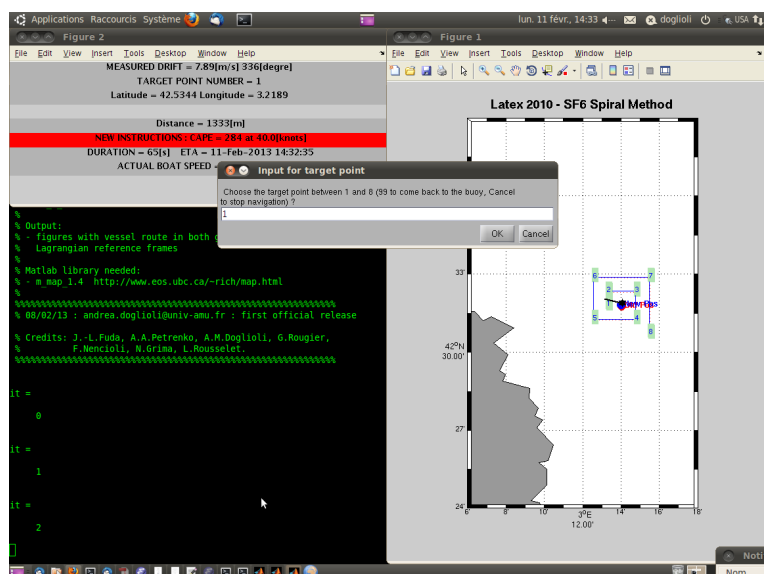


FIGURE 6 – Ecran de navigation Lagrangienne

dernier point de la spirale, le score n'est pas calculé. Lorsque l'utilisateur clique sur le bouton "OK" de la boîte de dialogue, les figures disparaissent de l'écran et sont sauveées dans le dossier "figures" du paquet dont le chemin par défaut est le suivant : LATEXtools_0.0/figures.

La démo dure environ 5 à 6 minutes en fonction de la rapidité de l'utilisateur à répondre aux boîtes de dialogue et des performances de son ordinateur. Il faut aussi noter que la position de la fenêtre MATLAB a été optimisée pour l'affichage des figures. Si l'utilisateur change cette optimisation et remet l'affichage par défaut, le programme tourne plus rapidement.

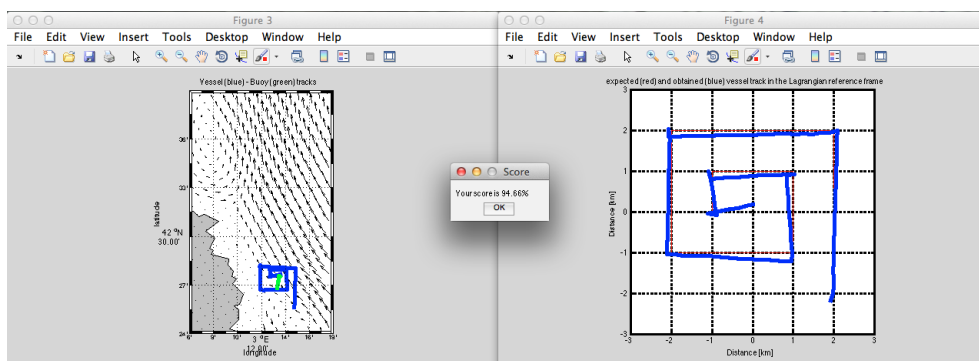


FIGURE 7 – Dernière figures de la démo.

4 Discussion

La première version de ce logiciel a été conçu en 2010, il s'est donc déroulé plus de deux ans depuis. Les scripts ont donc été re-travailler pour un fonctionnement plus réel. L'avantage de ce paquet est que tous les utilisateurs ont accès aux codes de chaque script. Ainsi chacun peut ajouter, enlever ou modifier un morceau de script afin qu'il colle le plus possible à l'utilisation que la personne veut en faire. En effet la démo a été faite avec une carte et des données concernant la côte Ouest du Golf du Lion. Cependant si l'utilisateur souhaite tester cette démo avec une carte différente et qu'il possède les informations nécessaires à la réalisation de cette carte, par exemple les champs de vitesse des courants, il peut le faire. Si une autre personne change un calcul car il ne le trouve pas assez précis, il peut ensuite partager le logiciel contenant des modifications sur internet. Ce logiciel est donc un logiciel libre et c'est aussi ce qui fait son utilité auprès de chercheur du monde entier.

Le deuxième atout de ce logiciel est qu'il a été réalisé le plus simplement possible. Ainsi même quelqu'un qui n'est pas expert en physique océanographique peut le prendre en main facilement suite à la découverte du logiciel grâce à la démo. Cependant cette démo n'est pas aussi réaliste qu'une navigation réalisée en mer. En effet dans la démo, la spirale Lagrangienne théorique, représentant la masse d'eau dans laquelle se situe la bouée de référence, reste la même tout au long de la navigation alors qu'en réalité la masse d'eau subie des déformations dues aux courants.

Il est donc envisageable d'ajouter un programme qui calcul les déformations subies par la masse d'eau et de les appliquer à cette même spirale. Les déformations sont du type compression, dilatation, rotation et déformation due au cisaillement de la vitesse. Celles-ci engendreraient des changements sur la forme de la spirale (Fig. 8).

Pour pouvoir utiliser les formules de déformations classiques suivantes, il serait nécessaire d'obtenir des axes x et y dans la masse d'eau. Pour cela il suffirait d'ajouter deux bouées dans la masse d'eau. Les trois bouées formeraient donc un système de coordonnées cartésiennes à deux dimensions.

– Rotation :

$$\omega = \nabla \times \vec{u} = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \quad (1)$$

Pour notre cas la matrice de rotation serait la suivante :

$$R_{ij} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2}(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}) \\ \frac{1}{2}(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}) & 0 \end{bmatrix} \quad (2)$$

– Compression / Dilatation :

La matrice de compression/dilatation est la suivante :

$$C_{ij} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & 0 \\ 0 & \frac{\partial v}{\partial y} \end{bmatrix} \quad (3)$$

– Cisaillement : Le cisaillement correspond au tenseur de contraintes suivant :

$$S_{ij} = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2}(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ \frac{1}{2}(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) & 0 \end{bmatrix} \quad (4)$$

Les dérivées partielles seraient approximées par des distances ou par des différences de composantes de la vitesse. Par exemple ∂u serait égal à $u_b - u_a$ où u est la composante horizontale de la vitesse de la bouée b ou a . Les dérivées partielles ∂x et ∂y seraient équivalentes aux distances entre les bouées a et b et b et c .

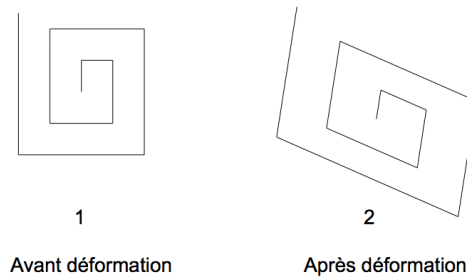


FIGURE 8 – La spirale avant et après les déformations de la masse d'eau.

Enfin après quelques tests sur le code, l'équipe s'est rendu compte que le déplacement des bouées n'était pas conforme à la direction du courant lorsque les bouées ont été placées dans un autre champ de vitesse. En effet les bouées se déplacent toujours vers le Nord quelque soit le courant dans lequel elles se trouvent. Cela provient d'un problème, dans le code de déplacement des bouées, qu'il est impératif de corriger. En effet le code utilise la fonction `cart2pol` pour changer des coordonnées cartésiennes en une distance radiale et une distance angulaire. Ensuite ces distances sont utilisées pour calculer les nouvelles latitudes et longitudes de la bouées comme suit :

```
[angle,radius]=cart2pol(dx,dy);
```

```
[lon1_new,lat1_new,azimuth21] = m_fdist(lon1,lat1,angle,radius);
```

Cependant le fait que les bouées se dirigent vers le Nord au lieu de suivre le courant vient du fait que l'angle que retourne `cart2pol` est en radian et que l'angle utilisé par `m_fdist` doit être en degré. Il faut donc faire ce changement. De plus l'angle que doit utiliser `m_fdist` doit être un azimut c'est-à-dire un angle calculé par rapport au Nord. Or ici l'angle est calculé de `dx` à `dy`, `dx` étant l'axe horizontal et `dy` l'axe vertical. Si on inverse ces deux données dans le calcul de `cart2pol` l'angle trouvé devrait être un angle calculé par rapport au Nord. Le code devrait donc être changé comme suit :

```
[angle,radius]=cart2pol(dy,dx); %angle [rad]
angle=rad2deg(angle); %convert angle from radians to degrees
[lon1_new,lat1_new,azimuth21] = m_fdist(lon1,lat1,angle,radius); %angle must
be degrees
```

Après ces quelques ajouts le package devrait être bien abouti cependant rien n'exclu que d'autres modifications puissent intervenir dans les prochains mois voire les prochaines années.

5 Conclusion

Ce stage s'est terminé avec la première parution sur Internet de la démo qui est aujourd'hui disponible en téléchargement gratuitement. Le travail effectué fut rigoureux et précis car les scripts et le Userguide doivent être sans fautes et compréhensibles par tous. Il a donc fallu noter chaque petits changements ou modifications ce qui demande une grande rigueur. D'un point de vue informatique les progrès acquis pendant ce stage ont été notables de part une meilleur compréhension du logiciel MATLAB et du langage informatique. Il aura aussi fallu beaucoup réfléchir et se débrouiller par soi même pour trouver des solutions à des problèmes informatiques. En effet l'application d'un problème physique sur un outils informatique n'est pas toujours simple car parfois même si le calcul est théoriquement correct le logiciel ne l'applique pas comme l'utilisateur l'aurait voulu. Il a donc fallu faire preuve de patience et de réflexion pour résoudre certains problèmes.

Le projet de développer une telle boîte à outils n'avait jamais été réalisé auparavant il a donc fallu avancer parfois à tâtons en essayant beaucoup de choses, parfois sans succès mais c'est aussi ce qui fait avancer les choses. Il faut aussi être très pointilleux dans le travail pour ne laisser aucune erreur se glisser un code déjà très long.

Enfin l'objectif a été atteint car la démo peut aujourd'hui être utilisée par qui le souhaite. Elle a été améliorée en tout point et quelques ajouts sont aussi venu enrichir le package. En effet un programme qui simule la dispersion d'un traceur passif dans un tourbillon aide l'utilisateur à planifier la région dans laquelle une quantité donnée de traceur va se disperser. Ensuite un deuxième programme aide l'utilisateur à planifier la route d'un bateau qui doit réaliser un échantillonnage.

Il n'y a pas de doute que ce package continuera à être développer et à s'enrichir au cours des prochaines années, c'est aussi le but d'un tel projet , que des gens de tout horizon puissent participer à l'amélioration de cet outils.

Références

- Law, C., Watson, A., Liddicoat, M., and Stanton, T. (1998). Sulphur hexafluoride as a tracer of biogeochemical and physical processes in an open-ocean iron fertilisation experiment. *Deep-Sea Res. II*, 45(6) :977 – 994.
- Law, C., Martin, A., Liddicoat, M., Watson, A., Richards, K., and Woodward, E. (2001). A Lagrangian SF_6 tracer study of an anticyclonic eddy in the North Atlantic : patch evolution, vertical mixing and nutrient supply to the mixed layer. *Deep-Sea Res. II*, 48 :705–724.
- Boyd, P. and Law, C. (2001). The Southern Ocean Iron RElease Experiment (SOIREE) - introduction and summary. *Deep-Sea Res. II*, 48(11–12) :2425 – 2438.
- Law, C., Abraham, E., Woodward, E., Liddicoat, M., Fileman, T., Thingstad, T., Kitidis, V., and Zohary, T. (2005). The fate of phosphate in an in situ Lagrangian addition experiment in the Eastern Mediterranean. *Deep-Sea Res. II*, 52(22–23) :2911 – 2927.
- Law, C. S., Crawford, W. R., Smith, M. J., Boyd, P. W., Wong, C. S., Nojiri, Y., Robert, M., Abraham, E. R., Johnson, W. K., Forsland, V., and Arychuk, M. (2006). Patch evolution and the biogeochemical impact of entrainment during an iron fertilisation experiment in the subArctic Pacific. *Deep-Sea Res. II*, 53 :2012–2033.
- Hu, Z. Y., Doglioli, A. M. Petrenko, A. A., Marsaleix, P., and Dekeyser, I. (2009). Numerical simulations of eddies in the Gulf of Lion. *Ocean Model.*, 28(4) :203 – 208.
- Tsumune, D., Nishioka, J., Shimamoto, A., Takeda, S., and Tsuda, A. (2005). Physical behavior of the SEEDS iron-fertilized patch by sulphur hexafluoride tracer release. *Prog. Oceanogr.*, 64(2–4) :111 – 127.
- Tsumune, D., Nishioka, J., Shimamoto, A., Watanabe, Y. W., Aramaki, T., Nojiri, Y., Takeda, S., Tsuda, A., and Tsubono, T. (2009). Physical behaviors of the iron-fertilized patch in SEEDS II. *Deep-Sea Res. II*, 56(26) :2948 – 2957.
- Stanton, T., Law, C., and Watson, A. (1998). Physical evolution of the IronEx-I open ocean tracer patch. *Deep-Sea Res. II*, 45(6) :947 – 975.
- Coale, K. H., Johnson, K. S., Fitzwater, S. E., Blain, S. P., Stanton, T. P., and Coley, T. L. (1998). IronEx-I, an in situ iron-enrichment experiment : Experimental design, implementation and results. *Deep-Sea Res. II*, 45(6) :919 – 945.

6 Annexes

A Fichier Contents.m mis à jour

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%%                               LATEXtools                               %%%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 %% This file is free software ; it is distributed in %%
5 %% the hope that it will be useful , but without any %%
6 %% warranty. You can redistribute it and/or modify it %%
7 %% under the terms of the GNU General Public License %%
8 %% as published by the Free Software Foundation at %%
9 %%      http://www.gnu.org/copyleft/gpl.html      %%
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %% This package contains software developed during
12 %% LATEX projet 2008–2011
13 %% www.com.univ-mrs.fr/LOPB/LATEX
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %% 21/02/13 : andrea.doglioli@univ-amu.fr : first official release
16 %%
17 %% 02/12/2012 LR : translations and comments
18 %% 03/09/2012, FN+AD: bathymetry data added
19 %% 24/08/2012 AD: Submission of the paper to JTECH and beta release of LATEXtools
20 %%      merging the package for EGU2011 poster and software developed
21 %%      during
22 %%      Latex10 cruise
23 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
24 %%
25 %% LR : louise.rousselet@etu.univ-amu.fr
26 %% FN : francesco.nencioli@univ-amu.fr
27 %% AD : andrea.doglioli@univ-amu.fr
28 %% AP : anne.petrenko@univ-amu.fr
29 %%
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32 %%
33 %%% LIST of SCRIPTS and FUNCTIONS %%%
34 %%
35 %%% LAGRANGIAN NAVIGATION %%%
36 %%
37 %% lagrangian_navigation_demo
38 %%
39 %% lagrangian_reconstitution_demo
40 %%
41 %% lagrangian_release_demo
42 %%
43 %% lagrangian_sampling_spiral
44 %%
45 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GRAPHICS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46 %%
47 %% map_ADCP          Plots ADCP vectors
48 %% plot_ADCP         Help to get all the necessary infos from the ADCP file
```

```

49 % load_ADCP Load data from ADCP file
50 %
51 %% map_ARGOS Plots trajectories of ARGOS buoys
52 % plot_ARGOS Help to get all the necessary infos from the Argos file -
   *.txt
53 % load_ARGOS Load data from Argos file
54 %
55 %% map_AVHRR Plots map of SST
56 % plot_AVHRR Helps to draw AVHRR frame
57 % load_AVHRR Load AVHRR data
58 %
59 %% map_BATHY Add high resolution isobath for NW Mediterranean
60 % plot_BATHY Help to draw high res isobaths
61 % load_BATHY data from BATHY_NWMED directory
62 %
63 %% map_FSLE Plots FSLE manifolds
64 % plot_FSLE Help to get FSLE infos
65 % load_FSLE Load data from FSLE
66 %
67 %% map_FSLE_ADCP Plot geographical map with FSLE-ADCP manifolds
68 %
69 %% map_FSLE_ARGOS Plot geographical map with FSLE-ARGOS manifolds
70 %
71 %% map_SF6 Plots map of concentration
72 % plot_SF6 Help to get infos to plot data of the SF6 concentration
73 % load_SF6 Load data of the SF6 validated data
74 %
75 %% map_VESSEL Plots Vessel positions
76 % plot_VESSEL Help to get vessel infos to plot vessel positions
77 % load_VESSEL Load vessel data
78 %
79 %%%%%%%%%%% UTILITIES %%%%%%%%%%%
80 %
81 %% make_route Draws a route and calculate time between stations
82 %
83 %% make_clean_ADCP Helps in positioning the cuts in ADCP data
84 % clean_ADCP Script for cutting ADCP data
85 % regroup_ADCP Script for matching data from different ADCP transects of a
   same cruise
86 %
87 % decideg_to_degmin Convert geographical coordinate from decimal
   degrees to degrees minutes
88 % degmin_to_decideg Convert geographical coordinate from degrees
   minutes to decimal degrees
89 % degminsec_to_decideg Convert geographical coordinate from degrees
   minutes secondes to decimal degrees
90 %
91 %
92 %
93 %%%%%%%%%%%

```

B Brouillon de script

Il s'agit ici du brouillon du script simulant la dérive d'un bouée dans un champ de vitesse, il n'est donc pas fonctionnel car inachevé.

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%                               Test Demo
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4
5 clear all;
6 close all;
7
8 %xdir=[1:1:200]*10^3 % x direction [m]
9 %ydir=[1:1:100]*10^3 % y direction [m]
10 [lat ,lon] = meshgrid(41:0.1:45,0:0.1:10);
11
12
13 %- Map boundaries -%
14 lonmin=3;
15 lonmax=4.3;
16 latmin=42;
17 latmax=43.2;
18 %
19 %- Geographic projection for m_map -%
20 projection = 'mercator';
21 %
22 %- User coast in dirin -%
23 UserCoast='GOL.mat'; % in "sub_functions" directory
24 %
25 %- Plot the figure
26 figure(1); hold on;
27 m_proj('mercator','lon',[lonmin lonmax],'lat',[latmin latmax])
28 %
29
30
31 [x,y]=m_ll2xy(lon,lat); % convert lon,lat in x,y coordinates
32 u=zeros(size(x));
33 u(1:101,1:41)=-0.25; % x component of current speed [m.s-1]
34 v=zeros(size(y));
35 v(1:101,1:41)=+0.00; % y component of current speed [m.s-1]
36
37 m_quiver(lon,lat,u,v)
38 m_usercoast(UserCoast,'patch',[.7 .7 .7],'edgecolor','k');
39 m_grid
40
41
42 DX=1
43 distanceX=[1:DX:200];
44 distanceY=[1:DX:100];
45
46
47 deltat=10 % [s]
48 xbuoy_old_lon=3.5 %initial lon position
49 ybuoy_old_lat=42.5 %initial lat position
50 [xbuoy_old,ybuoy_old]=m_ll2xy(xbuoy_old_lon,ybuoy_old_lat)
```

```

51
52 %xbuoy_old=100 % initial x position
53 %ybuoy_old=50 % initial y position
54
55
56 for it=1:10
57 xbuoy=xbuoy_old+u( fix(xbuoy_old), fix(ybuoy_old))*deltat % buoy x displacement
    with current
58 ybuoy=ybuoy_old+v( fix(xbuoy_old), fix(ybuoy_old))*deltat % buoy y displacement
    with current
59
60 xbuoy_old=xbuoy
61 ybuoy_old=ybuoy
62
63 [long , lati]=m_xy2ll(xbuoy ,ybuoy , 'clip' , 'patch') % convert x,y buoy coordinates in
    lon , lat
64
65 figure(1);
66
67 hold on
68 %quiver(xdir(1:1:end),ydir(1:10:end),u(1:10:end,1:10:end),v(1:10:end,1:10:end))
69 quiver(distanceX(1:10:end),distanceY(1:10:end),u(1:10:end,1:10:end),v(1:10:end
    ,1:10:end));
70 m_plot(long , lati , '—rs' , 'LineWidth' , 2 , ... % plot(
    xbuoy , ybuoy ..
71         'MarkerEdgeColor' , 'k' , ...
72         'MarkerFaceColor' , 'g' , ...
73         'MarkerSize' , 10)
74         axis([1 200 1 100])
75
76 pause
77 end % for it

```

C Script spiral_calculation

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%%                               LATEXtools                               %%%
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 %% This file is free software ; it is distributed in %
5 %% the hope that it will be useful , but without any %
6 %% warranty. You can redistribute it and/or modify it %
7 %% under the terms of the GNU General Public License %
8 %% as published by the Free Software Foundation at %
9 %% http://www.gnu.org/copyleft/gpl.html %
10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
11 %% spiral_calculation.m (Matlab script)
12 %%
13 %% Computes a spiral with a number of corners defined by coil_number ,
14 %% and a longitudinal and latitudinal spacing defined by step_x and step_y
    respectively;
15 %%
16 %% There are a number of user defined switches that can be set to change
17 %% the sense and type of rotation of the spiral.
18 %%

```



```

19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 %% 21/02/13 : andrea.doglioli@univ-amu.fr : first official release
21 %% Credits : F.Nencioli
22 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
23 function [ coord_spiral , totdist ]= spiral_calculation ( coil_number , step_x , step_y )
24 %-----
25 % User defined parameters: %
26 %-----
27 % 1) Center of the spiral
28 acoord(1)=0;
29 bcoord(1)=0;
30 % 2) direction of the first step:
31 % the first step from the origin can be set to be
32 % along y ('y') or along x ('x')
33 first_step='x';
34 % 3) sense of direction of the first 2 steps:
35 % - switch1 determines if the first step is towards positive (1) or negative (-1)
    direction
36 % - switch2 determines if the second step is towards positive (1) or negative
    (-1) direction
37 % (The rest of the spiral is computed iteratively)
38 switch1=-1;
39 switch2=1;
40 %=====
41
42 if first_step=='x'
43     steпа=step_x;
44     stepb=step_y;
45 elseif first_step=='y'
46     steпа=step_y;
47     stepb=step_x;
48 else
49     error('First_step can only be ''x'' or ''y'' ')
50 end
51
52 if abs(switch1)~=1 || abs(switch2)~=1
53     error('switch1 or switch2 can only be +1 or -1')
54 end
55
56 % Start to iteratively compute the spiral vertices.
57 % For each loop there are four points that are computed
58 % until the number of loops reach the maximum defined by nombre_de_creneaux
59 for i=1:coil_number
60     % compute position of first corner of the loop
61     acoord((4*i)+1-3)=acoord(((4*i)+1-3)-1)+(switch1*steпа*(i*2-1));
62     bcoord((4*i)+1-3)=bcoord(((4*i)+1-3)-1);
63     switch1=-switch1;
64     % compute position of second corner of the loop
65     acoord((4*i)+1-2)=acoord((4*i)+1-3);
66     bcoord((4*i)+1-2)=bcoord((4*i)+1-3)+(switch2*stepb*(i*2-1));
67     switch2=-switch2;
68     % compute position of third corner of the loop
69     acoord((4*i)+1-1)=acoord((4*i)+1-2)+(switch1*steпа*(i*2));
70     bcoord((4*i)+1-1)=bcoord((4*i)+1-2);
71     switch1=-switch1;

```

```

72         % compute position of fourth corner of the loop
73         acoord((4*i)+1)=acoord((4*i)+1-1);
74         bcoord((4*i)+1)=bcoord((4*i)+1-1)+(switch2*stepb*(i*2));
75         switch2=-switch2;
76     end
77 % acoord and bcoord are saved as xcoord or ycoord depending on the values of
       first_step
78 if first_step=='x'
79     coord_spiral=[acoord;bcoord];
80 elseif first_step=='y'
81     coord_spiral=[bcoord;acoord];
82 end
83
84 % enleve le point central (bouee) de la spirale
85 coord_spiral=coord_spiral(:,2:end);
86
87
88
89
90 %[ totdist , xx , yy ]
91 %if first_step=='x'
92     %xx=acoord;
93     %yy=bcoord;
94 %elseif first_step=='y'
95     %xx=bcoord;
96     %yy=acoord;
97 %end
98 % compute tot dist
99 i=1:coil_number*2;
100 totdist=sum((stepa+stepb)*i);
101
102
103 return

```

D Fichier readme.txt

Table of Contents

1. System Requirements
2. Installation
3. Help
4. Contact Information

1. System Requirements
-

This toolbox has been designed for Matlab. It needs at least 32 Mbytes of disk space. It has been tested for Matlab versions ranging from R2010a to R2011b but may be operational with other version. It can be used on Linux workstations, but also used on Windows and Mac computers. At the moment no special workstation version incompatible with LATEX-tools_0.0 are known.

The necessary Matlab toolbox `m_map` (vs 1.4, <http://www.eos.ubc.ca/rich/map.html>) and other free programs are included.

2. Installation

No special installation is needed a simple uncompressing and untaring file (gunzip and tar -xvf) command is useful to extract the file.

To use this software, open a Matlab session and make sure to be placed in the right directory : Current folder : `_*/LATEX-tools_0.0` . The software is ready to be used.

3. Getting Help

If you are encountering any troubles to run programs, you can find more help in the users guide. It is available in the "Documentation" file. Problems might also come from your different version of Matlab. Matlab includes an extensive online help system. Use the Help menu in the Matlab menu bar to access Help.

5. Contact Information

For contact information visits :

<http://www.com.univ-mrs.fr/doglioli/>

This file is free software ; it is distributed in the hope that it will be useful, but without any warranty. You can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation at <http://www.gnu.org/copyleft/gpl.html>

Date : 2013/02/15
