

MEMOIRE DE STAGE EN LABORATOIRE DE RECHERCHE

MODELISATION DES REJETS DE L'AQUACULTURE ET  
MISE A JOUR DU MODELE LAMP3D

PRESENTÉ PAR : BRUN VICTORIA

RESPONSABLE DU STAGE : DOGLIOLI ANDREA

ORGANISME D'ACCUEIL : MIO



## TABLE DES MATIERES

<b>1. Introduction</b>	<b>4</b>
1.1 Objectifs . . . . .	4
<b>2. Matériels et méthodes</b>	<b>6</b>
2.1 Les logiciels utilisés . . . . .	6
2.2 Les données . . . . .	7
<b>3. Résultats</b>	<b>8</b>
3.1 Mise à jour du package . . . . .	8
3.2 Mise à jour de la démo . . . . .	10
3.3 Parution du package sur Internet . . . . .	11
3.3.1 Les dernières modifications . . . . .	12
3.3.2 Installation du paquet . . . . .	12
3.3.3 Faire tourner la démo . . . . .	13
<b>4. Discussion</b>	<b>18</b>
<b>5. Conclusion</b>	<b>19</b>
<b>6. Annexes</b>	<b>21</b>
Annexe I . . . . .	21
Annexe II . . . . .	22
Annexe III . . . . .	23

## Résumé

Depuis un bon nombre d'années, il a été possible d'observer une expansion massive de l'aquaculture. Cette industrie, bien qu'indispensable à certains pays, connaît ses défauts. En effet, il a été observé que les fermes aquacoles avaient un effet nocif sur l'environnement qui les entoure. Mr Doglioli et son équipe ont créé un programme informatique afin de modéliser l'impact des rejets de l'aquaculture dans l'ouest de la Méditerranée : en Mer de Ligurie. Le package LAMP3D a été créé il y a plus de dix ans, et utilisait donc des logiciels anciens. Il a donc fallu mettre à jour tous les scripts afin qu'ils fonctionnent sur les systèmes d'exploitation actuels. Il a aussi fallu rendre ces scripts utilisables par tous et créer un Userguide, c'est à dire un guide d'utilisation rédigé en anglais, afin d'expliquer le déroulement du logiciel. Toutes ces modifications jusqu'à la publication sur Internet sont décrites dans ce rapport.

Mots clés : Aquaculture, informatique, LAMP3D, Méditerranée.

## Abstract

For a number of years we have noticed a massive growth of aquaculture. This industry, while useful to some countries, shows some severe drawbacks. As a matter of fact aquaculture farms have shown negative impacts on their close environment. Mr Doglioli and his team have created a software in order to build a model simulating the impacts of aquaculture in the western part of the Mediterranean sea : the Ligurian sea. The LAMP3D package has been created over 10 years ago and as such was using old softwares. As a consequence there was a need to rewrite all the scripts so that they can run with current operating systems. There was a need to make all these scripts usable by everyone and we created an English userguide in order to explain in details how the software was working. All these adjustments up until they are published on internet are described in this report.

Key Words : Aquaculture, computer science, LAMP3D, Mediterranean.

## 1) Introduction

L'aquaculture est connue maintenant depuis le IVème millénaire avant Jésus-Christ. Des le Moyen-Age cette dernière connaît une expansion fulgurante dans toute l'Europe. Outre le fort apport économique mondial de l'aquaculture, cette technique de production intensive de poissons et plantes aquatiques connaît aussi ces défauts. En effet, depuis de nombreuses années maintenant les fermes aquacoles peuvent présenter des dangers pour les écosystèmes environnants. Il existe plusieurs raisons telles l'usage fréquent de médicaments, nocif pour les eaux côtières à proximité de ces fermes, la fuite de poissons des cages aquacoles non désirée dans l'environnement pouvant être porteurs de maladies ou de souches OGM, etc.

Du fait de l'expansion massive de l'aquaculture, cela a engendré un grand nombre d'études afin d'être capable d'estimer le cout de cette industrie aquacole. Egalement, de nombreuses études se consacrent à un travail plus en amont : estimer les éventuels impacts d'une ferme aquacole avant qu'elle ne soit créée.

Le projet étudier ici est le suivant : LAMP3D (« Lagrangian Assessment for Marine Pollution 3D model »). Cette étude a été réalisée dans le but de simuler la dispersion de polluants rejetés par une ferme marine aquacole dans la mer de Ligurie, en Méditerranée Occidentale. Des études pionnières ont déjà été réalisées mais ces dernières modélisaient la dispersion par un courant constant dans le temps et dans l'espace. De ce fait, ces études utilisaient des équations à deux dimensions, ne considéraient aucune variation de courant en fonction de la profondeur, et enfin, ne faisaient aucune différence quant aux différents types de polluants (azote, phosphore, carbone organique, etc.), puisque chaque polluant se disperse différemment en fonction du milieu où il est présent.

L'étude LAMP3D a donc développé un modèle basé sur des équations à trois dimensions, en couplant un modèle lagrangien (c'est à dire en mouvement avec les particules) à trois dimensions avec un modèle hydrodynamique afin d'être capable de décrire à la fois les flux hydrodynamiques en 3D ainsi que la dispersion en 3D des polluants rejetés par la ferme aquacole.

### 1.1) Objectifs

Le but de ce stage a été de tester les différents programmes déjà créés et de les mettre à jour car certains n'étaient pas fonctionnels. Il a également fallu créer un script Matlab pour sortir

les figures car l'ancienne version du package utilisait un ancien logiciel, bien moins performant. Après avoir effectué ce travail, il a été nécessaire de créer un formulaire d'utilisateur afin que chaque personne désirant télécharger le package LAMP3D puisse l'utiliser sans aucune difficulté. C'est ce que l'on appelle un Userguide.

## 2) Matériels et méthodes

Ce projet a été écrit sous Linux grâce au langage Fortran 77. Les scripts ont été écrits avec Kate, un éditeur de texte adapté à la programmation. La totalité du travail a donc été réalisée sur un ordinateur personnel, grâce à une « Virtual Box » préalablement installée. Il a aussi été nécessaire d'utiliser le logiciel Matlab. De plus, la deuxième partie du stage a été réalisée grâce au logiciel LaTeX afin d'écrire le Userguide.

### 2.1) Les logiciels utilisés

L'intégralité du stage a été réalisée sur un ordinateur personnel MacBook Air. La partie Fortran a été réalisée grâce à une Virtual Box Linux installée sur cet ordinateur, et la partie Matlab a été testée sous le système d'exploitation Mac OS X (version 10.6.8) et la version Matlab R2012b. D'autres tests ont aussi été réalisés sur des ordinateurs utilisant Linux comme principal système d'exploitation, et sur Matlab R2010a.

L'environnement de développement principal a été la Virtual Box Linux car les scripts sont écrits en langage Fortran. Ils ont donc été ouverts grâce à l'éditeur de texte Kate qui est spécialisé dans la programmation. Ces scripts contiennent tout le corps des commandes à réaliser, c'est eux qui, une fois compiler, donnent les commandes au terminal.

Tous ces scripts sont réunis dans un seul et même dossier et forment le package « LAMP3D ».

La partie concernant Matlab a été entièrement réalisée lors du stage car les figures qui étaient créées auparavant utilisaient le logiciel GrADS, bien moins performant que ceux que l'on peut trouver actuellement. Matlab est ce que l'on appelle un langage de quatrième génération et permet des calculs numériques très simplifiés ce qui est un avantage conséquent au vu du nombre de valeurs que contenaient les différents fichiers.

Enfin, la dernière partie du travail a été réalisée avec LaTeX qui est un logiciel de composition de documents qui permet une mise en page plus aisée qu'avec d'autres logiciels pour des articles scientifiques. Pour le système d'exploitation Mac OS X, ce logiciel s'appelle MacTeX, il présente exactement les mêmes possibilités. Pour créer un document sous LaTeX, il faut créer un fichier .tex dans lequel l'utilisateur doit écrire des commandes précises qui seront appliquées lors de la mise en page. Généralement, le nom d'une commande s'écrit : \nom de la commande {texte qui va subir la commande}. Par exemple pour écrire un texte en

italique il faut taper la commande \textit{ {texte à mettre en italique}}. Comme un bon nombre de logiciels, LaTeX a son propre langage d'encodage.

## 2.2) Les données

Afin de modéliser au mieux la dispersion des particules dans cette région de la Méditerranée, il a fallu se baser sur des données réelles. Il n'y a pas eu de campagne en mer réalisée pour cette étude, les auteurs se sont basés sur des données déjà mesurées. Cependant il a fallu prendre en compte un bon nombre de paramètres. En effet, les caractéristiques hydrodynamiques utilisées proviennent d'une étude réalisée par Astraldi et Manzella (1983). Ces données montrent un flux quasi-permanent de toute la masse d'eau vers le Nord-Ouest, parallèlement à la côte.

Il a aussi fallu prendre en compte la bathymétrie. Ces données ont été récoltées grâce à l'appariement de deux cartes bathymétriques appartenant à l'Institut Hydrographique de la Marine italienne. Il a été indispensable de coupler ces deux cartes car seules, elles n'apportaient pas une résolution suffisante. De plus, les cartes ne présentaient des données que jusqu'à deux cents mètres. Grâce à une interpolation linéaire, des valeurs ont été affectées jusqu'à la profondeur de trois-cent-dix mètres. D'autres données bathymétriques ont été arrangées pour obtenir la direction principale de l'écoulement, on pourra notamment retenir l'inclinaison de l'ensemble de la grille bathymétrique à douze degrés vers la gauche par rapport au nord géographique grâce à des programmes adaptés.

Il était également indispensable de considérer le forçage du vent. La encore, les auteurs se sont basés sur les données de Ravasco (2000). Ce sont des données récoltées par l'Armée de l'Air italienne entre 1963 et 1996. Grâce à des calculs statistiques, il a pu être dégagé trois vents importants distincts : un dans la zone Nord-Est, un autre au Sud-Est et enfin un dernier au Sud-Ouest, avec chacun leurs caractéristiques saisonnières.

Même si ces données ne donnent pas de réponse pour un moment précis, elles permettent tout de même de dégager des tendances générales qui restent suffisamment exactes dans un contexte général.

### 3) Résultats

Le package LAMP3D a été mis à disposition le 1<sup>er</sup> Mai 2014 sur internet. Cette version contient tous les fichiers nécessaires au bon fonctionnement du package. Il s'agit de la version officielle du logiciel.

Nous verrons dans cette partie toutes les étapes qui ont du être réalisées afin que le package mis à disposition sur internet soit utilisable par tous. En effet, pour respecter cette utilisation universelle, il a fallu mettre à disposition deux documents supplémentaires lors du téléchargement du paquet afin de fournir des informations précises d'utilisation aux différents usagers. Dans le répertoire LAMP3D sont donc présents deux fichiers : « readme.txt » qui explique très brièvement l'installation et les conditions d'utilisation, et « Userguide.pdf » qui contient toutes les informations nécessaires à l'utilisation du paquet ainsi qu'un tutoriel du logiciel. Ces deux fichiers ont été créés lors du stage.

L'intégralité du fichier « readme.txt » est disponible en annexe, contrairement au fichier « Userguide.pdf » qui lui est trop long, mais qui est disponible en téléchargeant le paquet LAMP3D.

#### 3.1) Mise à jour du package

La première partie de ce stage a été de prendre en main l'intégralité des scripts présents dans l'ancienne version du package. Plusieurs problèmes se sont posés : les scripts avaient été écrits plus de dix ans auparavant, ce qui signifie que les auteurs utilisaient le langage Fortran 77 (F77) alors que depuis, le langage le plus couramment utilisé est le langage Fortran 90 (F90). Même si ces langages d'encodages restent très proches, il est important de noter que certains codes de F77 ne sont pas compatibles avec le F90. Il a donc fallu exécuter chaque programme afin de réaliser des « traductions » pour que les fichiers s'exécutent correctement. Parfois ces « traductions » n'existaient pas, il a alors fallu supprimer certaines lignes de code, qui heureusement n'affectaient pas l'ensemble du programme. Par exemple, la commande « allocate », pour une variable « Z » à deux dimensions s'écrit, en F77 : Allocate(Z( :, :)), alors qu'en F90 il faut l'écrire de la manière suivante : Allocate :: Z( :, :). Toutes ces petites modifications ont du être réalisées pour chaque programme qui contenait des commandes trop anciennes, et donc non exécutables par le F90. Ce fut un travail très minutieux puisqu'il fallait noter chaque modification de chaque programme pour pouvoir revenir en arrière si, après modification le programme n'était pas exécutable. Il faut avancer petit à petit, en

modifiant seulement quelques lignes, puis compiler le programme pour voir s'il tourne, et ainsi de suite.

En plus de ce changement de langage à effectuer, des problèmes de compatibilité se sont posés. En effet, les programmes ont été écrits sur un ordinateur utilisant Linux comme système d'exploitation principal. L'ordinateur utilisé pendant le stage est un MacBook Air contenant une Virtual Box Linux. Cet outil est une machine virtuelle qui permet d'émuler un autre système d'exploitation sur un même ordinateur. Cependant, cela inflige des contraintes supplémentaires car ce n'est pas aussi efficace qu'un ordinateur Linux à proprement parlé. Egalement, quelques programmes étaient au format « ifortran », il a donc fallu modifier ces programmes car la Virtual Box utilisée compile uniquement un langage « gfortran ».

Une fois ces modifications apportées et fonctionnelles, il a fallu comprendre à quoi servait chaque programme. En effet, ce logiciel nécessite une grande rigueur puisque les scripts doivent s'exécuter dans un ordre bien précis afin de fonctionner correctement. Cependant, les scripts présents dans le paquet de téléchargement étaient écrits en italien. Un deuxième travail a donc été de traduire toutes les lignes de commentaires. Ces lignes permettent à n'importe quel utilisateur de comprendre les différentes étapes d'un code informatique. Puisque ce paquet peut être téléchargé partout dans le monde il a fallu réécrire les lignes de commentaires dans une langue universelle : l'anglais. Une fois la traduction réalisée, il a été plus aisé de comprendre le fonctionnement des codes même si certains avoisinaient les cinq cents lignes.

A la suite de ce travail, il a fallu comprendre quels scripts étaient utilisés et quels scripts ne l'étaient pas. En effet, le package disponible à l'époque contenaient certains fichiers inutiles par rapport à la modélisation désirée lors de ce stage. Par exemple, un répertoire nommé « GRADS » était présent lors du téléchargement. Ce répertoire permettait de sortir des figures grâce au logiciel GrADS. Cependant, ce logiciel étant trop vieux, des programmes Matlab ont été créés afin d'avoir des figures plus modernes. Il a donc fallu analyser chaque répertoire et observer quels scripts étaient utiles pour la version finale du paquet.

Une fois ces différentes étapes réalisées, l'arborescence du paquet avait bien changée par rapport à l'originale. En téléchargeant le paquet LAMP3D, voici l'arborescence obtenue :

- LAMP3D
  - o CONVERT
  - o POM\_DATA
  - o POMLAMPFIN
  - o pomMATLAB

### 3.2) Mise à jour de la démo

Le principe de ce paquet est de modéliser les rejets d'une ferme aquacole dans l'ouest de la Méditerranée : dans la mer de Ligurie. Grâce à la démo, il est possible d'obtenir différentes figures. En premier lieu, il a été modélisé le courant moyen de la région sur plusieurs jours et ensuite, il est possible d'observer la concentration des particules, sur une zone plus précise que le jeu de figures précédent.

Grâce à toutes ces étapes de corrections et de manipulations de lignes de codes, les programmes tournaient convenablement. Lors de l'exécution des différents fichiers, des fichiers textes sont créés, et sont remplis de données grâce à la simulation. Cependant ces fichiers sont au format F90, c'est à dire inexploitables pour un script Matlab. L'ancienne version du paquet contenait un script pour traduire les données Fortran en Matlab pour le premier jeu de figure. Néanmoins, la partie de conversion des concentrations entre Fortran et Matlab n'existe pas. En effet, un programme avait été fait par d'anciens élèves de Mr Doglioli, afin de convertir les données de concentration de Fortran à GrADS. Un des objectifs de ce stage était de ne plus utiliser ce logiciel car trop ancien, il ne sortait pas des figures efficaces et bien lisibles. La première solution qui s'est présentée a été d'intégrer les lignes du programme de concentration dans le programme du courant moyen. Le but de cette solution était en fait d'obtenir un seul et même programme qui convertissait dans un même temps le courant moyen et la concentration de particules. Cependant, les lignes de codes pour la concentration n'avaient pas été écrites par Mr Doglioli, il a donc été très difficile d'essayer de comprendre des lignes de codes écrites par quelqu'un d'autre dix ans auparavant. Après quelques tentatives, il a été décidé d'opter pour la deuxième solution : modifier le programme des concentrations pour une conversion vers Matlab, et devoir exécuter deux fichiers différents pour obtenir les fichiers convertis. Ici aussi, ce fut un travail très minutieux puisqu'il fallait reprendre des « morceaux » d'un script pour le remettre dans l'autre tout en modifiant le nom des variables qui changeait selon le programme.

De plus, les différentes variables de chaque programme n'étaient pas à la même dimension. En effet, les variables présentes dans le programme du courant moyen sont à trois dimensions et modélisent : l'axe x, l'axe y, la hauteur, la surélévation, la vitesse horizontale moyenne selon u, et la vitesse horizontale moyenne selon v. Toutes ces variables sont également fonction du temps ce qui en fait des variables 3D. Alors que, les variables présentes dans le programme de la concentration sont à quatre dimensions et correspondent à : l'axe i, l'axe j, l'axe k, la concentration, la vitesse horizontale selon u, la vitesse horizontale selon v et la vitesse verticale selon w. La aussi, toutes les variables sont fonction du temps. Il fallait donc

adapter un texte fonctionnel pour des variables 3D, à des variables 4D. Au fur et à mesure, il a fallu faire différentes modifications afin d'obtenir un résultat satisfaisant. Ci-joint une partie du script pour la conversion des données Matlab en Fortran, pour le courant moyen.

```

!2D+t matrix
write(*,*) 'scrittura file '//fileOUT2//'.txt'

open(98,file=datadirOUT2//fileOUT2//'.txt',status='new') !,form='formatted',recordType='stream',status='new')

DO it=1,iTMAX
  do i=1,IM
    do j=1,JM
      write(98,'(3i5,4F16.6)') i,j,it,H(i,j,it),
      &                           ELB(i,j,it),UAB(i,j,it),VAB(i,j,it)
    enddo
  enddo
ENDDO

close(98)

```

Figure 1: Ecriture du fichier de sortie pour Matlab de la matrice 3D

Enfin, une fois les deux fichiers fonctionnels, il a fallu créer le fichier Matlab qui utilise les deux fichiers de sortie du courant moyen et de la concentration.

Afin de créer ce fichier il a tout d'abord bien fallu comprendre le fonctionnement des matrices obtenues. En effet, il a fallu utiliser certaines commandes afin de redimensionner les matrices pour qu'elles soient correctement utilisables, notamment grâce à la commande « reshape ».

Une autre difficulté a été de bien comprendre les différentes dimensions de chaque matrice afin de pouvoir imaginer une représentation des différents paramètres dans l'espace. En effet, il a été plus difficile de trouver une solution de représentation pour la matrice 4D, puisque cette dimension n'est pas modélisable. La solution a donc été de représenter la concentration en fonction des axes *i* et *j* pour une certaine profondeur et non pour toutes les profondeurs. L'utilisateur est donc libre de modifier le programme Matlab afin de choisir à quelle profondeur il veut observer les valeurs de concentrations.

### 3.3) Parution du package sur Internet

Avant que le package soit publier sur Internet, il faut être sûr de son fonctionnement puisqu'il sera par la suite téléchargeable par tous. On verra donc dans cette partie les dernières modifications apportées au package puis son fonctionnement.

### 3.3.1) Les dernières modifications

Avant le dépôt du package sur internet, il était indispensable de le tester sur d'autres machines car certains bugs persistaient. En effet, le programme qui se charge de remplir les fichiers textes était très lent, il fallait parfois attendre plusieurs jours avant d'obtenir des fichiers entièrement remplis. De plus, certaines erreurs graphiques survenaient lors de la compilation du script Matlab. Par exemple, lors de l'utilisation de la commande « pcolor », qui permet de colorer une carte grâce à un panel de couleur qui est fonction des valeurs de la variable considérée. Lors de l'exécution du script le code couleur n'était pas respecté. Après de nombreuses recherches sur internet, des modifications et des recherches sur des programmes écrits par Mr Doglioli afin de comparer les lignes de codes, il était évident que l'erreur pouvait venir de l'ordinateur personnel utilisé. C'est ici que les tests sur différentes machines deviennent très utiles, car ils ont apporté une grande satisfaction. En effet, d'une part, toute la partie sur Linux s'exécute très rapidement et sans erreur, et d'autre part, la sortie des figures sur Matlab ne présentait pas de problème de coloration. Ce sont grâce à ces derniers tests qu'il faut se rendre à l'évidence que l'utilisation d'un système Mac OS X/Virtual Box Linux constitue un système de contrainte assez conséquent pour le bon fonctionnement du paquet.

### 3.3.2) Installation du paquet

Ce package a donc été conçu pour un système d'exploitation Linux, et nécessite également le logiciel Matlab. Il nécessite 32 Mo d'espace libre sur la mémoire de l'ordinateur. Il a été testé sur une Virtual Box Linux installée sur un MacBook Air, sur un ordinateur utilisant Linux comme système d'exploitation principal, et sur des versions de Matlab allant de R2010a à R2012b, mais il peut être fonctionnel sur des versions plus anciennes ou récentes. Quelques petites modifications pourraient cependant être nécessaires.

Pour installer le paquet, il suffit de se rendre sur la page web où il est disponible : <http://moi.pytheas.univ-amu.fr/~doglioli/>. Il faut se rendre dans l'onglet LAMP3D qui se trouve sur la page principale, un dossier compressé est alors téléchargé. Après décompression le dossier LAMP3D apparaît, et contient tous les fichiers nécessaires à la bonne utilisation du paquet. Dans un premier il est nécessaire d'ouvrir un terminal de contrôle afin d'être capable d'exécuter les différents programmes, et il faut ensuite se placer dans le bon répertoire pour que cette exécution soit fonctionnelle : Répertoire parent: ~/LAMP3D. Les programmes sont alors prêts à être exécutés.

Un « Userguide » est disponible dans le paquet de téléchargement afin de guider au mieux l'utilisateur pour les étapes d'exécution et lui expliquer l'utilité des différents programmes. Pour la partie concernant Matlab, l'utilisateur peut se référer au Userguide mais il existe également une aide aux commandes Matlab très simple d'utilisation. Il suffit de taper, dans la fenêtre principale de Matlab : help nom de la commande. Cet aide explique à l'utilisateur à quoi correspond la commande tapée, et lui explique comment elle doit être utilisée.

### 3.3.3) Faire tourner la démo

Le premier programme à exécuter est le fichier makepomlamp.sh qui se trouve dans le répertoire POMLAMPFIN. Il s'agit d'un fichier .sh c'est à dire un fichier Shell. Ces fichiers sont en fait une succession de commandes qui vont s'exécuter selon l'écriture contenue dans le fichier .sh.

```
#! /bin/sh

#####
echo
echo "----- compilazione pom -----"
echo
echo "Number of simulation?"
read NSIM

gfortran -O -o pomlamp$NSIM.exe pom98_lamp.f mysr3d_lamp.f icond_lamp$NSIM.f bcond_lamp$NSIM.f lamp3d_3_2.f

echo "backup source files"
tar cvzf temp.tgz pom98_lamp.f mysr3d_lamp.f icond_lamp$NSIM.f bcond_lamp$NSIM.f lamp3d_3_2.f params98_lamp.fi comblk98_lamp.fi

mv -i temp.tgz pomlampsources$NSIM.tgz

echo
echo "-----      fine      -----"
####
```

Figure 2 : Contenu du fichier makepomlamp.sh

Grâce à ce fichier, il est possible d'observer les différents fichiers qui vont être utilisés et créés.

A la suite de cette exécution, il faut compiler le fichier exécutable qui a été préalablement créé : le fichier pomlamp240.exe. Cette exécution va permettre d'assigner des valeurs dans les différents fichiers à remplir. Cette étape peut être plus ou moins longue selon les paramètres voulus par l'utilisateur. En effet plusieurs critères entrent en jeu. Grâce au fichier params98\_lamp.fi, il est possible de changer les valeurs de différents paramètres. Par exemple, il est possible de changer la durée de la simulation en changeant le nombre de jour. Il est également envisageable de changer le pas de temps pour chaque mesure ou encore la vitesse de sédimentation des particules. Le fichier params\_98.fi est disponible en annexe. Il y aussi possibilité de modifier la dispersion des particules en faisant varier le paramètres « Sigma » qui se trouve dans le fichier icond\_lamp240.f. Il est possible de faire varier Sigma

entre 5 et 10, 5 correspond à une dispersion faible tandis que 10 correspond à une grande dispersion.

Le terminal doit afficher une liste de valeurs pendant un certain temps (dépendant du nombre de jour et du pas de temps) après la compilation de l'exécutable :

```
begin numerical integration
MODE,IINT,TIME      2      1      0.00
MODE,IINT,TIME      2      2      0.00
MODE,IINT,TIME      2      3      0.00
MODE,IINT,TIME      2      4      0.00
MODE,IINT,TIME      2      5      0.00
MODE,IINT,TIME      2      6      0.00
MODE,IINT,TIME      2      7      0.00
MODE,IINT,TIME      2      8      0.00
MODE,IINT,TIME      2      9      0.00
MODE,IINT,TIME      2     10      0.00
MODE,IINT,TIME      2     11      0.00
MODE,IINT,TIME      2     12      0.00
MODE,IINT,TIME      2     13      0.00
MODE,IINT,TIME      2     14      0.00
MODE,IINT,TIME      2     15      0.01
MODE,IINT,TIME      2     16      0.01
MODE,IINT,TIME      2     17      0.01
MODE,IINT,TIME      2     18      0.01
MODE,IINT,TIME      2     19      0.01
MODE,IINT,TIME      2     20      0.01
MODE,IINT,TIME      2     21      0.01
MODE,IINT,TIME      2     22      0.01
MODE,IINT,TIME      2     23      0.01
MODE,IINT,TIME      2     24      0.01
MODE,IINT,TIME      2     25      0.01
MODE,IINT,TIME      2     26      0.01
MODE,IINT,TIME      2     27      0.01
MODE,IINT,TIME      2     28      0.01
MODE,IINT,TIME      2     29      0.01
MODE,IINT,TIME      2     30      0.01
MODE,IINT,TIME      2     31      0.01
MODE,IINT,TIME      2     32      0.01
MODE,IINT,TIME      2     33      0.01
MODE,IINT,TIME      2     34      0.01
MODE,IINT,TIME      2     35      0.01
MODE,IINT,TIME      2     36      0.01
MODE,IINT,TIME      2     37      0.01
MODE,IINT,TIME      2     38      0.01
MODE,IINT,TIME      2     39      0.01
MODE,IINT,TIME      2     40      0.01
```

Figure 3 : Intégration numérique des différentes valeurs.

Une fois l'intégration numérique terminée, il faut alors convertir les données Fortran en données Matlab. L'utilisateur doit alors se placer dans le répertoire CONVERT afin d'exécuter le fichier makeconvert.sh. Cette compilation va alors créer l'exécutable convert.exe, qui doit lui aussi être compilé. Lors de cette exécution le terminal demande à l'utilisateur quelques paramètres graphiques et techniques

```

etudiant@linuxSM05:~/pom_lamp_fin_lavagna/CONVERT$ ./makeconvert.sh
----- compilazione convert -----

etudiant@linuxSM05:~/pom_lamp_fin_lavagna/CONVERT$ ./convert.exe

*****
*****      CONVERT      vs 3D 05      *****
*****



DX?
200
DY?
200
codice simulazione?
AB
numero simulazione?
240

```

Figure 4 : Paramètres que doit entrer l'utilisateur lors de la compilation de l'exécutable.

DX et DY correspondent au zoom que l'ordinateur effectuera lors de la mise en forme du premier jeu de figures. Le code de simulation correspondant va être utilisé lors de la nomination du fichier de sortie. En effet, à la suite de cette exécution, un fichier de sortie compatible avec Matlab est créé. Si un utilisateur veut effectuer ce travail plusieurs fois mais souhaite garder ses fichiers de valeurs, il aura simplement à modifier le code de simulation lors de l'exécution, un nouveau fichier sera alors créé, sous un nom différent cette fois-ci, il pourra donc être correctement sauvegarder dans le bon répertoire. En considérant les paramètres de la figure, le fichier sera alors nommé : r2DAB240.txt.

Le numéro de simulation reste le même pendant toute l'exécution du logiciel : 240.

Une fois le premier fichier de sortie créé, il faut alors composer le deuxième. Pour cela il faut exécuter le fichier makeconvertlamp.sh qui se trouve toujours dans le répertoire CONVERT. Le terminal demande à nouveau à l'utilisateur quelques paramètres :

```

etudiant@linuxSM05:~/pom_lamp_fin_lavagna/CONVERT$ ./makeconvertlamp.sh
----- compilazione convert -----

etudiant@linuxSM05:~/pom_lamp_fin_lavagna/CONVERT$ ./convertlamp.exe
*****
*****      CONVERTLAMP      (FA & PO)      *****
*****



numero simulazione?
240
IOLAMP?
20
JOLAMP?
20

```

Figure 5 : Paramètres que doit entrer l'utilisateur lors de la compilation du deuxième exécutable.

Le numéro de simulation reste donc toujours le même. Les paramètres IOLAMP et JOLAMP signifient la même chose que DX et DY mais il faut cette fois-ci leur assigner la valeur de 20 car il s'agit ici des paramètres graphiques pour la concentration. Le premier jeu de figures permet d'observer les courants moyens présents dans la mer de Ligurie, alors que le deuxième jeu de figures permet lui d'observer la concentration en particules. Cette concentration s'observe donc à proximité de la ferme aquacole, en assignant les valeurs de 20 à IOLAMP et JOLAMP, la figure sera donc extrêmement zoomée sur la zone de la ferme.

A la suite de cette exécution le deuxième et dernier fichier texte sera créé et est nommé lamp\_240.txt.

A ce stade tous les fichiers nécessaires ont été créés, et la partie d'exécution sous Fortran90 est terminée.

Il faut à présent que l'utilisateur ouvre le logiciel Matlab en se plaçant toujours dans le bon répertoire afin que le fichier LAMP240.m soit exécutable. Ce fichier est disponible en annexe. L'utilisateur doit simplement exécuter ce programme en s'assurant que les deux fichiers de sorties ont bien été créés et sont disponibles dans le répertoire pomMATLAB.

Ci-joint, le résultat d'une simulation de cinq jours, avec un pas de temps de 0.5, une vitesse de sédimentation de 0 m/s, c'est à dire caractéristique des pelotes fécales et enfin une dispersion (c'est à dire le paramètre sigma) de 5.

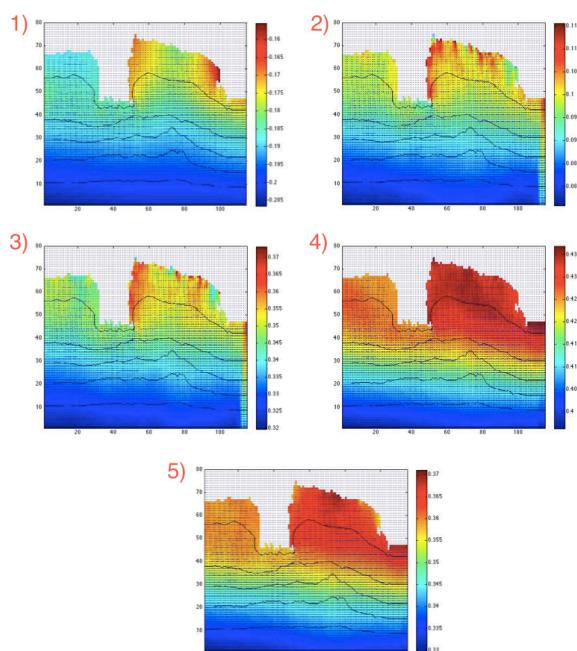


Figure 6 : Variation de la surélévation et représentation du courant moyen durant 5 jours.

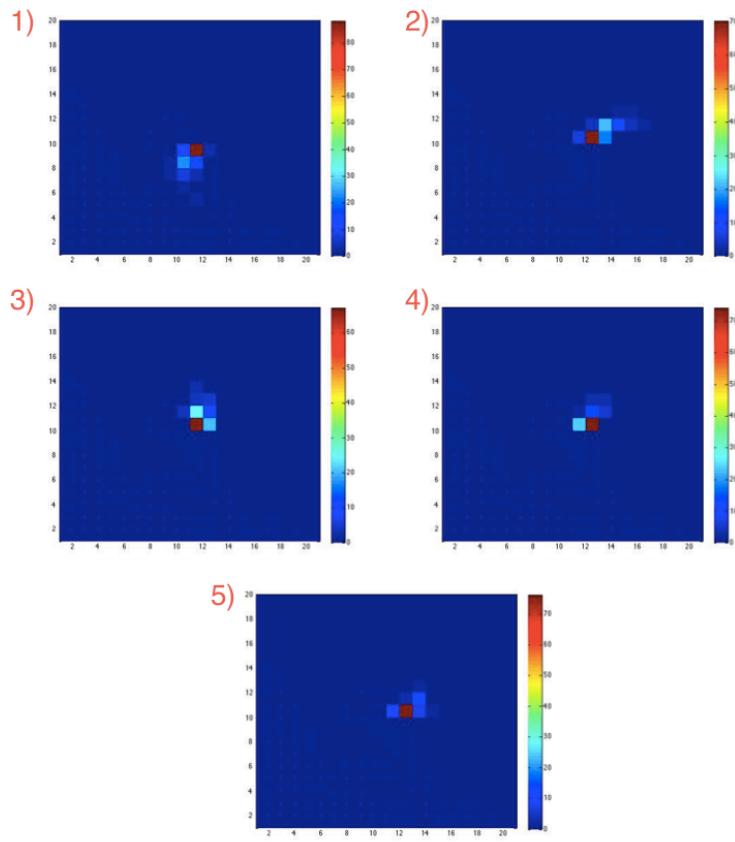


Figure 7: Variation de la concentration des particules pour une durée de 5 jours.

## 4) Discussion

La dernière modification officielle de ce package avait été réalisée en 2004. Dix ans se sont donc écoulés depuis. Il a fallu prendre en main des scripts écrits en italien et anglais afin de les améliorer pour une utilisation universelle. L'avantage d'un tel package est que tous les scripts sont disponibles à n'importe quel utilisateur lors du téléchargement du paquet. Ainsi, les scripts sont modifiables par chacun, selon le but final désiré. Ces modifications peuvent être par la suite publiées sur internet et partagées à travers le monde, ce qui en fait un logiciel libre.

Le deuxième avantage de ce logiciel est qu'il est maintenant fourni avec un Userguide. Ainsi, un utilisateur s'y connaissant peu en informatique n'a qu'à suivre le bon déroulement de ce document et il pourra obtenir sans problème les figures désirées, sans avoir à aller lire des lignes de codes.

Pour des utilisateurs plus expérimentés, il pourrait être possible d'observer une autre zone géographique. En effet, l'obtention de telles cartes est due aux données entrées dans la base du logiciel. Ainsi, en utilisant d'autres données de surélévation et de courant par exemple, il serait possible d'étudier l'impact d'une autre ferme aquacole.

## 5) Conclusion

Ce stage a été conclu par la parution sur Internet de cette mise à jour qui est disponible gratuitement au téléchargement. Le travail réalisé durant ce stage a été très minutieux car la moindre modification dans un script peut engendrer l'arrêt total du fonctionnement du logiciel. Egalement, il faut que les documents fournis soient sans faute et les plus compréhensibles possibles afin que n'importe quel utilisateur puisse faire fonctionner le package dans son intégralité.

Concernant l'aspect informatique, des progrès notables ressortent de ce stage puisque de nombreuses commandes à modifier ou même à créer, jusque là inconnues ont pu être utilisées. Il a fallu également de la patience puisque les contraintes de bases pour utiliser ce logiciel étaient tout de même conséquentes : l'utilisation d'une Virtual Box sur un système d'exploitation OS X tout en compilant avec du Fortran 90. Les erreurs lors de la compilation des programmes étaient donc très fréquentes, il fallait donc trouver la meilleure solution à chaque fois.

Ce stage s'est tout de même révélé efficace puisqu'il a abouti à la parution du package sur internet. Ce package a été amélioré, puisqu'il utilise maintenant le logiciel Matlab afin de sortir les figures de la modélisation, il est fourni avec un Userguide qui permet à n'importe quel utilisateur d'utiliser le paquet, et enfin tous les scripts ont été traduits de l'italien à l'anglais afin qu'ils puissent être « déchiffrés » par tous.

Il est évident que ce paquet sera plus abordable à l'utilisation, et peut-être que cela permettra une utilisation plus importante à travers le monde. De plus, puisque les programmes sont modifiables, cela pourrait faciliter l'amélioration du paquet LAMP3D.

## REFERENCES

- A.M Doglioli et al., 2004. Development of a numerical model to study the dispersion of wastes coming from a marine fish farm in the Ligurian Sea (Western Mediterranean). *Aquaculture, Volume 231, Issues 1-4, Pages 215-235.*
- De Gaetano et al., 2011. Impact of new measured Mediterranean mineralization rates on the fate of simulated aquaculture wastes. *Aquaculture Research, Volume 42, Issue 9, Pages 1359-1370.*
- De Gaetano et al., 2008. FOAM, a new simple benthic degradative module for the LAMP3D model : an application to a Mediterranean fish farm. *Aquaculture Research, Volume 39, Issue 11, Pages 1229-1242.*
- Vassallo et al., 2006. Determination of physical behaviour of feed pellets in Mediterranean water. *Aquaculture Research, Volume 32, Issue 2, Pages 119-126.*
- Oscar Pérez et al., 2014. Food and faeces settling velocities of meagre (*Argyrosomus regius*) and its application for modelling waste dispersion from sea cage aquaculture. *Aquaculture, Volumes 420-421, Pages 171-179.*
- Astraldi et Manzella 1983. Some observations on current measurements on the east ligurian shelf, Mediterranean Sea. *Continental Shelf Research, Volume 2, Issues 2-3, Pages 179-194.*
- Chelossi et al., 2003. Antibiotic resistance of benthic bacteria in fish-farm and control sediments of the Western Mediterranean. *Aquaculture, Volume 219, Issues 1-4, Pages 83-97.*
- Enell, 1995. Environmental impacts of nutrient from nordic fish farming. *Water Science and Technology, Volume 31, Issue 10, Pages 61-71.*

## **6) ANNEXES**

### **ANNEXE I**

#### Table of Contents

---

1. System Requirements

2. Installation

3. Help

4. Contact Information

#### 1. System Requirements

---

This package has been designed for Fortran and Matlab. It needs at least 32 Mbites of disk space. It has been tested for Matlab versions ranging from R2010a to R2012b but may be operational with other version. It can be used on Linux workstations, but also used on Windows and Mac computers with Virtual Box. At the moment no special workstation version incompatible with LAMP3D are known.

#### 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 Linux session and make sure to be placed in the right directory : Current folder : ~/LAMP3D . 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 userguide. It is available in the "Documentation" file. Problems might also come from your different version of Linux or 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 \$

---

## ANNEXE II

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!!!!!!      params98.fi      !!!!!!!  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!  
!*****  
!*          PARAMETERS FOR POM      *!  
!*****  
  
NUMsim=240      !Simulation number  
  
MODE=2          !2=2D,3=3D,4=3D diagnostic  
  
DTE=0.5          !external time step  
ISPLIT=60        !DTI=DTE*ISPLIT intern time step  
ISPADV=5  
  
DAYS=5  
PRTD1=1  
PERIOD=SMALL    !time of LAMP PERIOD must be greater than 0  
!  PERIOD=9.6  
!iEND=6000  
!iPRINT=600  
!iPERIOD=0  
!iPRINTCORR=600  
  
NWRITE=1 !NWRITE=1 Hotstart writes files for the next simulation  
  
SMOTH=0.10 !leapfrog filter  
  
AMcost=1. ! in MODE 2, if AMcost=0 then AAM2D is computed in sr advAVE  
BFlin=0. ! in MODE 2, if BFlin=1 then bottom stress is linear with vel  
horcon=0.1 !coeff Smagorynski  
tprni=1. !AH/AM Prandtl coeff vert diffusione  
umol=2.e-5 !numerical viscosity  
  
tbias=0.  
sbias=0.
```

```

!*****PARAMETERS FOR LAMP3D*****!
!*          PARAMETERS FOR LAMP3D          *!
!*****PARAMETERS FOR LAMP3D*****!

seme=2.
deltaZ=5.      ![m]
typepart=0     !1->decad.;0->cons.
Vs=0           !Sedimentation velocity [m/s]
tiporilascio=1 !2->pe;1->co;0->ist
dE=-50.        !Ekman depth
Fsc=1.          !Scale factor VEL within the water column
Crid=1.         !Reduction coefficient saltoRZ=g*sigmaPROF*Crid
!   LAMPperiod=1440 !release period in interior time step?????
!=(60min*60sec*ore)/(DTE*Isplit)

T90=-9999.     !decay time
Zeff=10.        !effective height of the plume [m]

iFLUXSORGENTE=5 !each particle IINT
!*****PARAMETERS FOR LAMP3D*****!

```

### ANNEXE III

```

clear all;
close all;

M=load('r2DAB240.txt');
M2=load('lamp_240.txt');

M(M>999.9)=NaN;

x=M(:,1);
y=M(:,2);
t=M(:,3);
H=M(:,4);
ELB=M(:,5);
UAB=M(:,6);
VAB=M(:,7);

i=M2(:,1);
j=M2(:,2);
k=M2(:,3);
it=M2(:,4);
CONC=M2(:,5);
ULAMP=M2(:,6);
VLAMP=M2(:,7);
WLAMP=M2(:,8);

```

```

ELB=reshape(ELB,80,115,6);
H=reshape(H,80,115,6);

CONC=reshape(CONC,20,21,20,6);

for t=1:6
figure;

pcolor(squeeze(ELB(:,:,t)));
colorbar
shading flat;
hold on;
cRange=caxis;
contour(squeeze(H(:,:,t)),'k');
caxis(cRange);
A=quiver(x,y,UAB,VAB);

end

for t=1:6
figure;
ULAMP=squeeze(ULAMP(:,:,1));
VLAMP=squeeze(VLAMP(:,:,1));

pcolor(squeeze(CONC(:,:,1,t)));
colorbar
shading flat;
hold on;

B=quiver(i,j,ULAMP,VLAMP);
end

```