M2R Internship

Air-sea interaction at the synoptic and the meso-scale

Aimie Moulin

February to June 2012

Supervisor : Achim Wirth, chargé de recherche CNRS

Joseph Fourier University, M2R ECE/ OSUG

LEGI Laboratory/ MEIGE team

Acknowledgment

Firstly, I wish to thank everyone in the LEGI laboratory and the director M. Christophe Baudet, for their warm welcome during my M2R internship. I want to thank especially computer specialists Gabriel Moreau and Olivier De Marchi, always available, and who were very helpful for solving my computer problems. I also thank Quam Akuetevi for his various explanations on programming methods and the use of the visualization. Of course I want to thank Achim Wirth for his availability, his kindness and for all the knowledge he helped me deepen during these five months of internship. Finally I wish to thank all teachers and leaders of the M2R ECE.

Résumé

L'énergie mécanique apportée à l'océan par les vents atmosphériques est connue pour être la source d'énergie principale entrainant la circulation océanique. Cependant, aucune étude à notre connaissance, n'a été réalisé sur l'influence directe des courants océaniques sur l'atmosphère. Le but de ce stage était d'étudier ce phénomène, à l'échelle synoptique et méso-échelle, grâce à un modèle numérique idéalisé mais dynamiquement consistant.

La première partie de mon stage a été consacré à la conception des modèles physique, mathématique et numérique. Tout d'abord il m'a fallu élaborer un modèle physique approprié et choisir un modèle mathématique adapté, les équations de Saint-Venant bicouches en gravité réduite. J'ai ensuite construit un modèle numérique en discrétisant ces équations avec la méthode des différences finies d'ordre deux. J'ai enfin mis en application ce modèle et l'ai résolu en mode "batch" sur les machines multiprocesseurs du LEGI.

La deuxième partie de mon stage a été dédié à la vérification et la validation du modèle numérique. En effet des analyses de simulations simples en une dimension, comparées à des calculs analytiques m'ont permis de valider le modèle. Dans cette partie j'ai aussi identifié la génération d'ondes inertielles de gravité.

La dernière partie de mon stage a été destiné à l'analyse de l'interaction air-mer en deux dimensions. Les résultats obtenus, pour une loi de friction quadratique, ont révélé une influence non négligeable de la dynamique océanique sur la circulation atmosphérique. En effet, des tourbillons atmosphériques sont générés par la dynamique océanique bien qu'il n'y ai pas de transfert d'énergie de l'océan vers l'atmosphère. Ces tourbillons sont ensuite déplacés par le courant moyen atmosphèrique. Le tourbillon cyclonique dans l'océan attire le tourbillon cyclonique de l'atmosphère periodiquement, et le capture pour un peu plus de 2 mois. Cela entraine une longue variabilité de la dynamique atmosphérique avec une périodicité typique d'environ 6 mois. Ce phénomène n'est à ce jour pas pris en compte dans les modèles océan-atmosphère couplés.

Abstract

The mechanical energy input to the ocean by atmospheric winds is known to be the dominant source of energy for driving the ocean circulation. However, no study has, to the best of our knowledge, been realized on the direct influence of the ocean currents on the atmosphere. The purpose of this internship was to study this phenomenon, at the synoptic- and the meso-scale, thanks to an idealized but dynamically consistent numerical model.

The first part of my internship was devoted to the construction of the physical, mathematical and numerical models. First I had to envision a suitable physical model and choose an adapted mathematical model, the 2-layer reduced gravity shallow water equations. I then constructed a numerical model by discretizing the mathematical equations using the finite difference method of second order. I implemented this model and solved it in "batch" mode on the multiprocessor machines of LEGI. The analysis of the data and visualization was done with Scilab.

The second part of my internship was dedicated to the verification and the validation of the numerical model. Indeed, analyzes of simple one dimension simulations, compared to analytical calculations have ensured the validity of the model. In this part I also identified the generation of inertia-gravity waves.

The last part of my internship was intended for the analysis of the air-sea interaction in two dimension. The results obtained, for a quadratic friction law, revealed a significant influence of the ocean dynamics on the atmospheric circulation. In fact atmospheric eddies are generated by the ocean dynamics although there is no transfer of energy from the ocean to the atmosphere. These eddies are then advected by the mean wind. The cyclonic ocean eddy attracts the atmospheric cyclonic eddy periodically and captures it for up to two months. This leads to a longtime variability of atmospheric dynamics with a typical period of about 6 months. This phenomenon is not taken into consideration in today's ocean-atmosphere coupled models.

Contents

1 Introduction

Air-sea interactions are key to understand both the oceanic and the atmospheric circulation, and is therefore important for weather forecasting and for the study of climate variability. The air-sea interface is of great interest to the physics of the ocean and the atmosphere. Air-sea interaction mainly consists in the exchange of heat, momentum, fresh water and other chemical substances. Surface stress is one of the fundamental quantitative indicators of the coupling between the atmosphere and the ocean. Indeed the mechanical energy input to the ocean by atmospheric winds is known to be the dominant source of energy for driving the ocean circulation. Heating-cooling and precipitation-evaporation at the ocean surface are also important, because they create density differences influencing the ocean and atmosphere dynamics.

Since 2003, satellite scatterometer observations have shown that current induced features are visible in the wind stress curl (Chelton et al. 2004). It is an important consideration for ocean circulation modeling. Indeed few studies (Duhaut and Straub 2006, Scott and Xu 2009) have found that calculations of wind power input to the ocean should depend on the relative motion between the atmosphere and the ocean, not the velocity of the atmosphere alone. The surface shear stress τ, should be typically parametrized as a function of the difference between wind and ocean velocities. Because they did not account for the effects of ocean currents on the wind stress, numerical weather prediction models until few years ago, did not provide the true wind stress that drives the ocean circulation (Chelton et al. 2004).

Until recent years the influence of the ocean on the atmosphere has only been studied at large scales. The surface winds response to sea surface temperature (SST) at smaller scales have become evident over the past decade from simultaneous satellite measurements of SST and surface winds. They have revealed that ocean-atmosphere interaction is fundamentally different on oceanic mesoscales (order of hundred kilometers) than at synoptic-scale (order of thousand kilometers) (Chelton and Xie, 2010). At large scales (synoptic-scale) an anomaly of positive temperature in the atmosphere is usually connected to a heat flux from the atmosphere into the ocean. At smaller scales (mesoscale) such anomaly is associated with a heat flux from the ocean to the atmosphere. Much of the small scale variability in the wind stress field are assigned to SST modification of low level winds through the influence of air sea-heat flux on the marine atmospheric boundary layer (Chelton et al. 2004).

Currently, studies concerned with the influence of the ocean on the atmosphere are made considering SST induced processes and not the involvement of surface ocean currents, that remains unexplored and unknown while they may have an important role in the air-sea interaction, as I demonstrate in this work. Following the studies performed on the SST, its influence is now

included in ocean-atmosphere coupled models whereas the ocean is considered motionless and bottom friction of the atmosphere is calculated from atmospheric velocities only (M. Herrmann and P. Marchesiello, LEGOS, private communication).

One of the difficulties of this study is the difference in characteristic time scales between the atmosphere and ocean. Indeed it takes a long time to alter the ocean circulation while the atmospheric circulation moves rapidly. Moreover the typical horizontal scales of the ocean (order of tens of kilometers) are small compared to the typical horizontal scales of the atmosphere (order of hundreds of kilometers). Ocean-atmosphere coupled models must have a high spatial and temporal resolutions. Models with high resolutions including more and more components are designed to provide the best representation of the system and its dynamics, but they are limited by their high computational cost. An idealized model is optimum for answering to specific questions (IPCC report, 'The Physical Science Basis', 2007).

The aim of this internship is to construct an idealized model to study the air-sea interaction, due to momentum flux only, between ocean and atmosphere at the synoptic and the meso-scale.

2 Model:

2.1 Physical model

The model consist in two superposed homogeneous fluid layers, a shallow layer of the atmosphere above an ocean surface layer.

The physical model is illustrated by figure 1. For the following, the exponent "o" refers to the ocean layer and the exponent "a" refers to the atmosphere layer.

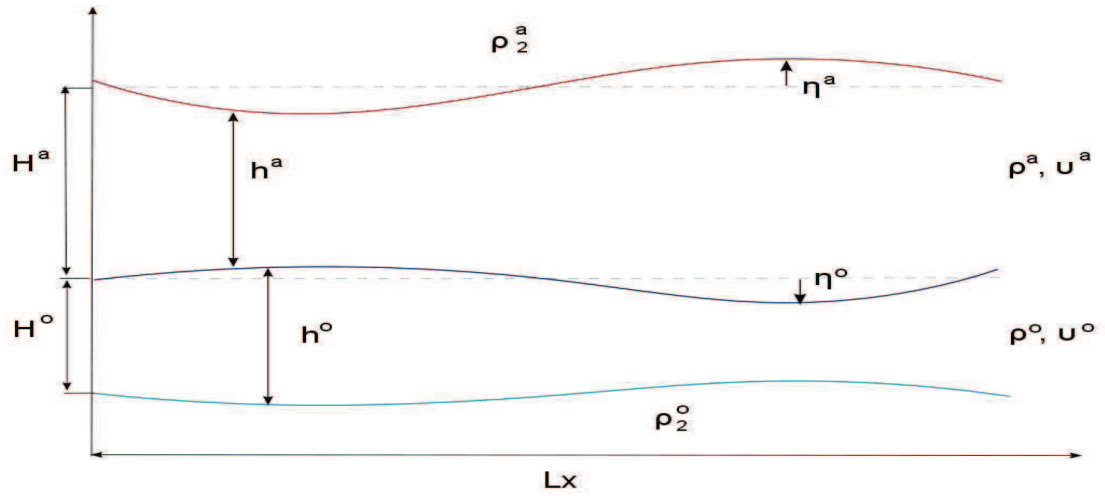


Figure 1: Scheme of the physical model

The average local thicknesses of the layers are $H^o = 200m$ for the ocean and $H^a = 500m$ for the atmosphere. The actual thicknesses can vary over time and are denoted by $h^o(x, y, t)$, $h^a(x, y, t)$ (Fig 1):

$$h^a(x, y, t) = H^a + \eta^a(x, y, t) \quad \text{and} \quad h^o(x, y, t) = H^o + \eta^o(x, y, t). \tag{1}$$

Where η^a, η^o represent only height perturbations of the top of the atmosphere layer and of the sea surface respectively. They depend on space and time.

The layers have an average density of $\rho^a = 1kg/m^3$ and $\rho^o = 1000kg/m^3$.

The ocean surface layer superposes a motionless layer of higher density ρ_2^o of infinite depth. Similarly, a motionless layer of atmosphere of density ρ_2^a superposes the shallow atmosphere layer. Thereby the two layers considered can be represented by reduced gravity models. The gravity g has just to be replaced by the reduced gravity of the ocean g^o and the atmosphere g^a in the dynamic equations:

$$g^o = \frac{\rho_2^o - \rho^o}{\rho_2^o} g = 2.10^{-2} m.s^{-2}, \tag{2}$$

where ρ_2^o is the density of the deep ocean layer and ρ^o the density of the ocean surface layer. In the same way : $g^a = 0.8m.s^{-2}$.

The fluid motion considered extends over many days and so, the model must take into account the Earth's rotation. The Earth's rotation vector $\overrightarrow{\Omega}$ points northward along the south-north axis, and has a magnitude $\Omega = 2\pi/T = 7,45.10^{-5}s^{-1}$, where T is the Earth's rotation period. At each point of the surface of the Earth, Ω can be described by a vertical and an horizontal component. For large scale motion as here, the horizontal component of the rotation vector $\overrightarrow{\Omega}$ is usually neglected; this is called the traditional approximation. Thus, at latitude θ, the vertical component of the rotation vector is denoted by $f = 2\Omega \sin\theta$ and called the Coriolis parameter. Here I set $f = 10^{-4}s^{-1}$ a typical value for mid latitudes.

The Rossby radius can be calculated for the ocean and the atmosphere:

$$Rd^k = \frac{\sqrt{g^k h^k}}{f} \qquad (\text{so} \quad Rd^a = 200km \quad \text{and} \quad Rd^o = 20km). \tag{3}$$

It represents the length scale at which rotational effects become as important as buoyancy effects in the evolution of the flow.

In this idealized model, the two layers are only linked by frictional forces at the interface, as atmospheric pressure variations have a negligible effect on the ocean dynamics and as height variations of the ocean surface have a negligible effect on the atmosphere. It is this friction term that I am going to vary between simulations.

2.2 Mathematical model based on the reduced gravity shallow water equations

The physical model presented above can be described by a mathematical model based on the reduced gravity shallow water equations. The shallow water equations are the most widely used equations in geophysical and environmental fluid dynamics.

To use this mathematical model, I assume that (Achim Wirth, 2011):
- The atmosphere and the ocean are composed of two incompressible fluids.
- The atmosphere and the ocean are very flat because height scales are very small compared to length scales ($H^a = 500m,\ H^o = 200m$ whereas $Rd^a = 200km$ and $Rd^o = 20km$).
- Density variations within one fluid are very low.

The shallow water equations which describe the dynamics of the ocean and atmosphere layers are written:

$$\partial_t u^k + u^k \partial_x u^k + v^k \partial_y u^k - f v^k + g^k \partial_x \eta^k = \nu^k \nabla^2 u^k + F_x^k \quad (4)$$

$$\partial_t v^k + u^k \partial_x v^k + v^k \partial_y v^k + f u^k + g^k \partial_y \eta^k = \nu^k \nabla^2 v^k + F_y^k \quad (5)$$

$$\partial_t h^k + \partial_x [h^k u^k] + \partial_y [h^k v^k] = 0, \quad (6)$$

where k=a or k=o.

The friction force applied to the atmosphere is the opposite of the friction force applied to the ocean:

$$\begin{pmatrix} f_x^o \\ f_y^o \end{pmatrix} = - \begin{pmatrix} f_x^a \\ f_y^a \end{pmatrix}.$$

The friction force F between the two layers in equations (4) and (5) can be defined by:

$$\begin{pmatrix} F_x^k \\ F_y^k \end{pmatrix} = \pm \frac{1}{\rho^k h^k} \begin{pmatrix} f_x \\ f_y \end{pmatrix}.$$

Where f_x and f_y are the surface forces depending on x and y axes.

I am going to study the dynamics with four different types of friction forces as defined below:

- The first is the linear Rayleigh friction:

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \rho^a \tau \begin{pmatrix} u^o - u^a \\ v^o - v^a \end{pmatrix} = \rho^a h^a \lambda^a \begin{pmatrix} u^o - u^a \\ v^o - v^a \end{pmatrix},$$

where the units of τ are in m/s. The friction time t^a for the atmospheric layer is defined by:

$$t^a = \frac{1}{\lambda^a} = \frac{h^a}{\tau} = \frac{h^a}{C_d u}, \tag{7}$$

where u is a typical velocity for the atmosphere.

The ocean friction time t^o can be calculated according to the atmospheric friction time:

$$t^o = \frac{1}{\lambda^o} = \frac{\rho^o h^o}{\rho^a h^a} \frac{1}{\lambda^a}. \tag{8}$$

- The second is a quadratic friction law. Such law parametrizes the turbulent dynamics of the boundary layers. It is based on the turbulent law-of-the-wall stating that the average velocity of a turbulent flow at a certain point is proportional to the logarithm of the distance from that point to the "wall", or the boundary of the fluid region.

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \rho^a C_d |u| \begin{pmatrix} u^o - u^a \\ v^o - v^a \end{pmatrix},$$

with $|u| = \sqrt{(u^o - u^a)^2 + (v^o - v^a)^2}$.

C_d is the drag coefficient it can be calculated with an empirical formula [Wu, 1982 and Smith, 1988]:

$$C_d = (0.8 + 0.065 \times u_{10}) \times 10^{-3} = 8.10^{-4}.$$

where u_{10} is the typical atmospheric velocity at 10 meters height.

Finally, Ekman veering is incorporated in the linear and in the quadratic friction.

- the linear Ekman friction is defined by:

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \rho^a \tau \begin{pmatrix} 1 & -\alpha \\ \alpha & 1 \end{pmatrix} \begin{pmatrix} u^o - u^a \\ v^o - v^a \end{pmatrix}.$$

- and the quadratic Ekman friction is equal to:

$$\begin{pmatrix} f_x \\ f_y \end{pmatrix} = \rho^a C_d |u| \begin{pmatrix} 1 & -\alpha \\ \alpha & 1 \end{pmatrix} \begin{pmatrix} u^o - u^a \\ v^o - v^a \end{pmatrix}.$$

Where $\alpha = \sin\theta$, with θ the veering angle which is 45° for a laminar Ekman spiral and around 20° for turbulent Ekman layers.

2.3 Dimensional analysis of the mathematical model

I am starting from the shallow water equations obtained in the previous section ((4) to (6)). Each parameter is adimensionalized:

$$x' = \frac{x}{L} \quad \text{and} \quad y' = \frac{y}{L}$$

$$
\begin{aligned}
u^{k\prime} &= \frac{u^k}{u_0^k} \quad \text{and} \quad v^{k\prime} = \frac{v^k}{u_0^k} \\
t' &= tf \\
\eta^{k\prime} &= \frac{\eta^k}{H^k}
\end{aligned}
$$

The first equation adimensionalized (4) becomes:

$$u_0^k f\partial'_t u^{k\prime} + \frac{u_0^{k2}}{L}u^{k\prime}\partial'_x u^{k\prime} + \frac{u_0^{k2}}{L}v^{k\prime}\partial'_y u^{k\prime} - f u_0^k v^{k\prime} + \frac{g^k H^k}{L}\partial'_x \eta^{k\prime} = \frac{\nu^k u_0^k}{L^2}\nabla^2 u^{k\prime} + F_x^k \quad (9)$$

$$\partial'_t u^{k\prime} + \frac{u_0^k}{fL}u^{k\prime}\partial'_x u^{k\prime} + \frac{u_0^k}{fL}v^{k\prime}\partial'_y u^{k\prime} - v^{k\prime} + \frac{g^k H^k}{u_0^k fL}\partial'_x \eta^{k\prime} = \frac{\nu^k}{fL^2}\nabla^2 u^{k\prime} + F_x^k \quad (10)$$

$$\partial'_t u^{k\prime} + Ro^k(u^{k\prime}\partial'_x u^{k\prime} + v^{k\prime}\partial'_y u^{k\prime}) - v^{k\prime} + \frac{Ro^k}{Fr^{k2}}\partial'_x \eta^{k\prime} = \frac{Ro^k}{Re^k}\nabla^2 u^{k\prime} + F_x^k. \quad (11)$$

The second equation adimensionalized (5) gives:

$$\partial'_t v^{k\prime} + Ro^k(u^{k\prime}\partial'_x v^{k\prime} + v^{k\prime}\partial'_y v^{k\prime}) - u^{k\prime} + \frac{Ro^k}{Fr^{k2}}\partial'_y \eta^{k\prime} = \frac{Ro^k}{Re^k}\nabla^2 v^{k\prime} + F_y^k. \quad (12)$$

The third equation (6) becomes:

$$H^k f\partial'_t h^{k\prime} + \frac{H^k u_0^k}{L}(\partial'_x u^{k\prime} h^{k\prime} + \partial'_y v^{k\prime} h^{k\prime}) = 0 \quad (13)$$

$$\partial'_t h^{k\prime} + \frac{u_0^k}{fL}(\partial'_x u^{k\prime} h^{k\prime} + \partial'_y v^{k\prime} h^{k\prime}) = 0 \quad (14)$$

$$\partial'_t h^{k\prime} + Ro^k(\partial'_x u^{k\prime} h^{k\prime} + \partial'_y v^{k\prime} h^{k\prime}) = 0 \quad (15)$$

$Ro = u/fL$ is the Rossby number, $Fr = u/\sqrt{gH}$ is the Froude number, and $Re = uL/\nu$ is the Reynolds number.

The friction term adds other dimensionless numbers:

- for the Rayleigh friction: $\frac{\lambda}{f}$.
- for the Quadratic friction: $C_d u/fH^k$.

Another non dimensional parameters is the ratio of densities: ρ^a/ρ^o.

2.4 Numerical model

2.4.1 Numerical grid

The numerical frame has only two horizontal dimensions x and y, and there are two layers in the vertical. The velocities u and v are z-independent, and obviously the thickness of the layers $H^k + \eta$ does not depend on z, thus there is no need to work with the vertical axis. The ocean and the atmosphere basins are represented by a square, of $Lx \times Ly = 1000km \times 1000km$. Periodic boundary conditions are used in both horizontal directions. The numerical grid is regular and contains $nx \times ny$ points. Each grid point is characterized by a couple of integers ix, iy, with ix=

[1:nx] and iy= [1:ny]. Spacial resolutions are the same in both directions, $\Delta x = Lx/nx = \Delta y = Ly/ny$. The horizontal components of the velocity u^k, v^k, and height variations η^k are calculated on each grid points. u^k is positive going eastwards and v^k is positive going northward. Variables u^k, v^k, η^k at time t and $t + \Delta t$ will respectively write u_o^k, v_o^k, η_o^k and u_n^k, v_n^k, η_n^k. The layers have kinematic viscosity coefficients ν^a and ν^o, which are constants in space and time. (Appendix 1)

2.4.2 Discretization of the mathematical model with second order schemes

Numerical schemes used here are based on the Taylor series. If a function f(x,y) has continuous derivatives up to the order n+1 then:

$$\begin{aligned} f(x,y) &= f(x_0,y) + \frac{f'(x,y)}{1!}(x-x_o) + \frac{f''(x,y)}{2!}(x-x_0)^2 \\ &+ \ ... + \frac{f^{(n)}(x,y)}{n!}(x-x_0)^n + O((x-x_0)^{n+1}). \end{aligned}$$

First I have to discretize all equations in space. To improve accuracy, the centered finite difference method is used for the space discretization. Thanks to the Taylor series, I calculate first and second order space derivatives:

$$\begin{aligned} f'(x,y) &= \frac{f(x+\Delta x,y) - f(x-\Delta x,y)}{2\Delta x} + O(\Delta x^2) \\ f''(x,y) &= \frac{f(x+\Delta x,y) - 2f(x,y) + f(x-\Delta x,y)}{\Delta x^2} + O(\Delta x^2) \end{aligned}$$

A second-order Runge-Kutta time scheme is used for the time discretization. The Runge-Kutta 2 method consist in evaluating the function derivative at the middle time $t + \Delta t/2$ of the iteration,

$$\begin{aligned} f(t+\frac{\Delta t}{2}) &= f(t) + \frac{\Delta t}{2}F(t,f(t)) \qquad\qquad With \quad F(t,f(t)) = f'(t) \\ f(t+\Delta t) &= f(t) + \Delta t F(t+\frac{\Delta t}{2}, f(t)+\frac{\Delta t}{2}) \end{aligned}$$

All values (u^k, v^k, η^k) are calculated in each points of the area. For that, I realize for each equation two "do" loops : the first from ix=1 to nx-1 and the second from iy=1 to ny-1. Then I introduce a loop for the time. (Appendix 1, pages iv-vii)

2.4.3 Discretization of periodic boundary conditions

As in the second order scheme, values at points i+1 and i-1 are involved to calculate values at point i. At the boundaries values have to be provided. For this model, periodic boundary conditions are used (Appendix 1, pages v-viii). Every point that comes out of the field at a boundary reappears at its opposite side (Fig 2).

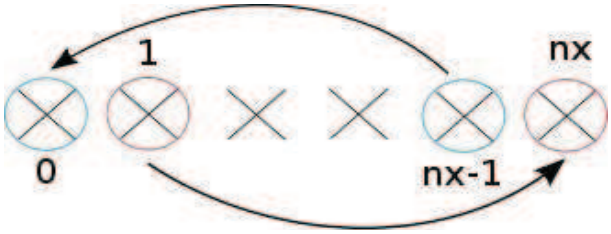


Figure 2: Scheme of the periodic boundary conditions in the x-direction

2.4.4 Stability of the numerical scheme

To ensure the stability of the numerical scheme, the Courant Friedrich Levy condition (CFL) must be complied. In order to avoid numerical instabilities, the time step Δt of the simulation must be inferior to the characteristic advection time scales. Indeed, the distance of wave propagation during a time step Δt must be smaller than the mesh size. Finally the model has to satisfy the condition:

$$\frac{c^k \Delta t}{\Delta x} \leq 1 \qquad \text{where} \quad c^k = \sqrt{g^k h^k} \qquad \text{(wave velocity)}$$

To improve accuracy I choose:

$$\sqrt{g^k h^k}\frac{\Delta t}{\Delta x} \leq 0.2$$

Since the thicknesses and the reduced gravities of the atmosphere and the ocean are very different, it is necessary to take a time step which respect the stability condition for both layers. In the atmospheric layer, gravity wave propagation c^a is ten times faster than in the ocean :

$$\begin{aligned} c^a &= \sqrt{g^a h^a} = \sqrt{0.8 \cdot 500} = 20 m/s \\ c^o &= \sqrt{g^o h^o} = \sqrt{2.10^{-2} \cdot 200} = 2 m/s \end{aligned}$$

so it is the atmospheric dynamics that sets the time-step.

2.4.5 Numerical implementation

I wrote the code in FORTRAN 77 starting from nothing (Appendix 1). The Fortran program was compiled with the Intel Ifort compiler. Then the compiled file was submitted on one of the LEGI calculators (servcalcul3) thanks to a job performed by the batch system OAR. All calculation are made in double precision. The runs presented have lasted up to 8 days. The simulation results were finally analyzed thanks to the SCILAB software (Appendix 2).

3 Validation of the model with a basic simulation:

In this part I use the linear Rayleigh friction:

$$F_x^k = \pm\frac{\rho^a\tau}{\rho^k(H^k+\eta^k)}(u^o-u^a) \qquad \text{where } \tau = C_d u_0 = 8.10^{-5} m.s^{-1} \tag{16}$$

$$F_y^k = \pm\frac{\rho^a\tau}{\rho^k(H^k+\eta^k)}(v^o-v^a) \tag{17}$$

I want to impose a sinusoidal velocity $u^o(y)$ independent of x for the ocean. I consider a rotating frame with a flow that is initially in a geostrophic equilibrium. A cosine initial condition, which only depends on the y direction, is imposed on the sea surface height:

$$\eta^o(x,y) = -Acos(2\pi y) \qquad \text{with } A = 100m. \tag{18}$$

The initial velocity field is calculated using geostrophic equilibrium:

$$v^o = \frac{g^o}{f}\partial_x\eta^o = 0\ m/s \tag{19}$$

$$u^o = -\frac{g^o}{f}\partial_y\eta^o = -\frac{g^o}{f}2\pi Asin(2\pi y) = -4\pi.10^{-2}sin(2\pi y)\ \ m/s. \tag{20}$$

Thereby I obtain the sinusoidal velocity for the zonal velocity $u^o(y)$ and a vanishing meridional velocity $v^o(y)$. The atmosphere is initially motionless. There is no x-dependence for all variables.

The system evolution is studied during ten months with a spatial resolution of 3.9 kilometers (256×256points). Values are saved every hour.

3.1 Spatial evolution

Firstly, the ocean and the atmosphere sections along the y axis are plotted.

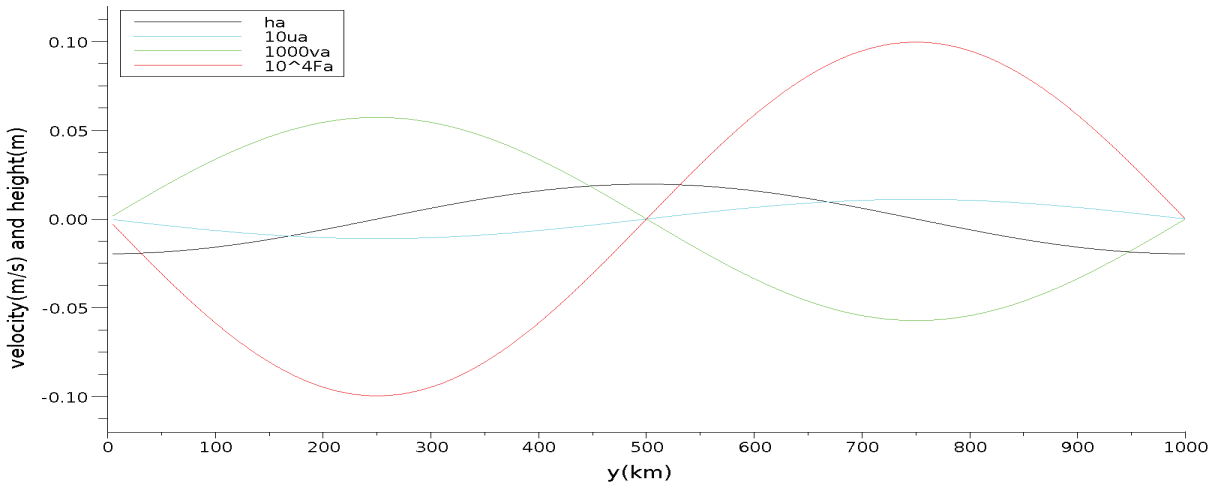


Figure 3: Atmospheric section after one day

After few hours the atmospheric height variation is a cosine and atmospheric velocities look like a sine (Fig 3). As expected,due to the friction forces the atmosphere adopts the same dynamics as the ocean.

Sections after one day show that the meridional velocity is not really sinusoidal (Fig 3). Indeed, when the velocity $v^k(y)$ increases, the non-linear terms $v^k\partial_y u^k$ and $v^k\partial_y v^k$ (equations (4) and (5)) become significant. There is a superposition of two harmonics so velocities $u^k(y)-u_g^k(y)$ and $v^k(y)-v_g^k(y)$ (u_g^k and v_g^k are the geostrophic velocities) are no longer sinusoidal. The harmonic superposition is not obvious for the zonal velocity because the zonal geostrophic velocity is not insignificant.

Section evolutions for the atmosphere and the ocean show wave propagation for the meridional velocities $v^k(y)$ and the zonal velocity $u^k(y)-u_g^k(y)$. These waves are out of phase.

3.2 Inertia-gravity waves

To identify more precisely wave propagation, velocity and height variations are plotted for the ocean and the atmosphere (Fig 4) at a function of time, at coordinates where each value is maximal.

Wave generation is emphasized for v^o and v^a because it is respectively multiply by 10^5 and 10^4. If I calculate wave periods I find: $T^o = 27.5j/38 = 17.3h \quad and \quad T^a = 30j/66 = 10.9h$.

As the Rossby number $\epsilon^k = u^k/(fLy)$ is small ($\epsilon^a \leq 6.10^{-3}$; $\epsilon^o \leq 1,25.10^{-3}$) the nonlinear terms can be neglected and the shallow water linear system can be considered:

$$\partial_t u^k - fv^k = F_x^k \tag{21}$$

$$\partial_t v^k + fu^k + g^k\partial_y \eta^k = F_y^k \tag{22}$$

$$\partial_t \eta^k + H^k\partial_y v^k = 0. \tag{23}$$

If I write the velocities u^k,v^k and the elevation η^k in complex form:

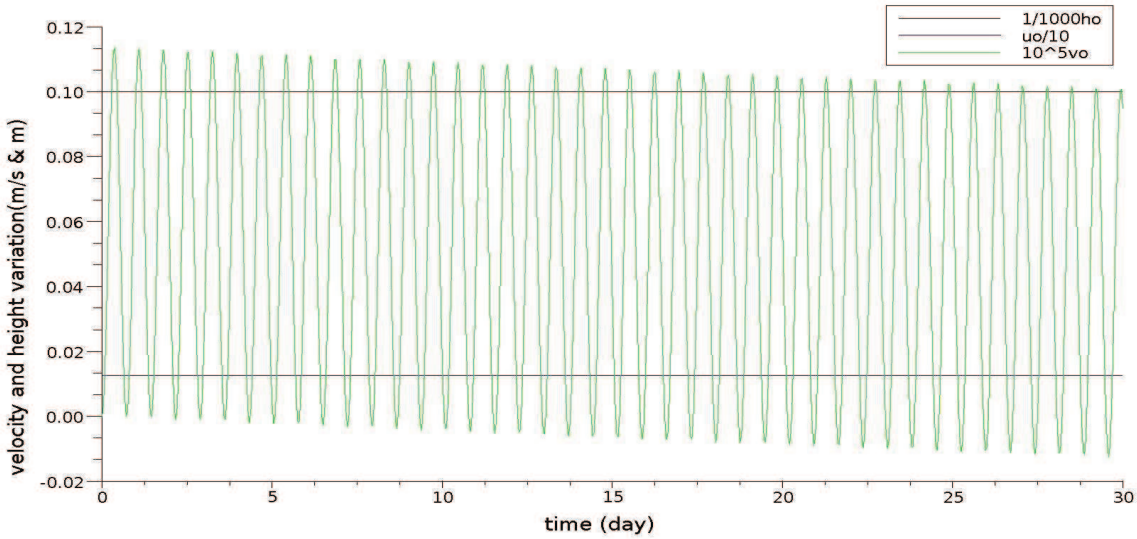
$$(u^k, v^k, \eta^k)(y,t) = (\hat{u}^k, \hat{v}^k, \hat{\eta}^k)\, exp\, i(ly - \omega^k t) \quad where\ l = \frac{2\pi}{Ly},$$

the system can be rewritten as:

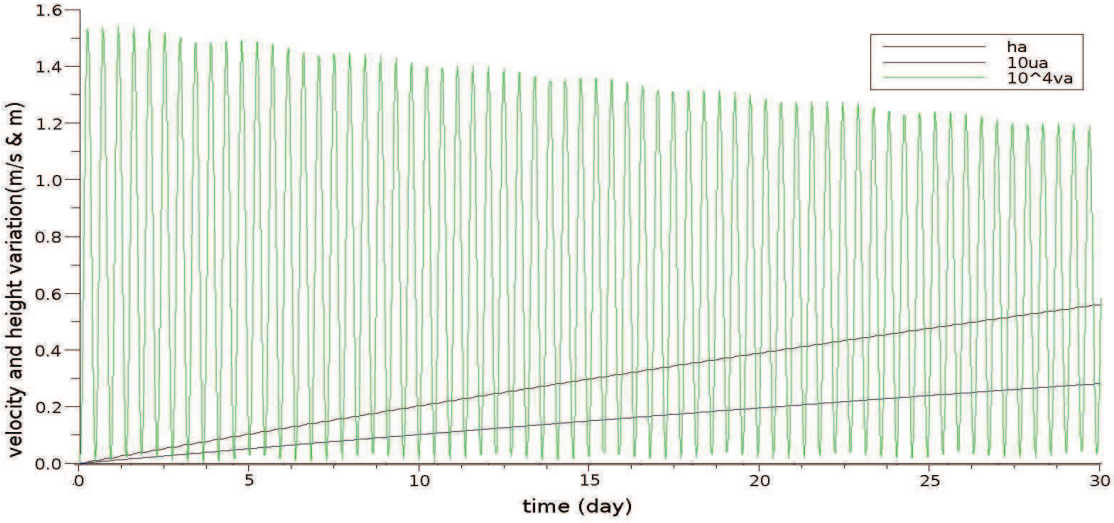
$$\begin{pmatrix} -i\omega^k & -f & 0 \\ +f & -i\omega^k & lig^k \\ 0 & liH^k & -i\omega^k \end{pmatrix} \begin{pmatrix} \hat{u}^k \\ \hat{v}^k \\ \hat{\eta}^k \end{pmatrix} = \begin{pmatrix} F_x^k \\ F_y^k \\ 0 \end{pmatrix}.$$

This system proves that inertia-gravity waves are generated and have periods:

$$T^k = Ly/c^k = \pm\frac{Ly}{\sqrt{\frac{f^2}{l^2} + g^k H^k}} \tag{24}$$



(a) Ocean



(b) Atmosphere

Figure 4: Temporal evolution of the ocean and the atmosphere characteristics

Substituting all values I found:

$$T^a \simeq 10.87\ h \quad and \quad T^o \simeq 17.32\ h \tag{25}$$

Indeed wave periods are the same that I have obtained from my simulations so they are inertia-gravity waves.

High coupling frequency is an important issue for air-sea interaction (IPCC report, 'The Physical Science Basis', 2007). However, it can bring new technical issues. As internal gravity waves are excited in the ocean which are poorly resolved in realistic ocean models some smoothing is necessary to damp this numerical problem (IPCC report, 'The Physical Science Basis', 2007). Here the high time resolution of the model allows to observe and study inertia-gravity waves with a good accuracy.

3.3 Energy

Frictions between the ocean and the atmosphere conserve the total momentum but induce an energy dissipation. To calculate the loss of energy due to friction, I firstly study a non rotating model.

I can not find u^o and v^o with geostrophic conditions as f=0 so I impose directly velocities:

$$\begin{aligned} u^o(y) &= -4\pi.10^{-2} sin(2\pi y)\ m/s \\ v^o &= 0\ m/s. \end{aligned}$$

At the initial step atmospheric velocity is null. Over time both fluids approach the same average velocity $\overline{u}$. According to inertial conservation:

$$(m^o + m^a)\overline{u} = m^o u^o \quad \Leftrightarrow \quad \overline{u} = \frac{m^o}{m^o + m^a} u^o. \tag{26}$$

Where m^o and m^a are respectively the mass of the ocean and atmosphere, per surface area unit $\rho^o h^o$ and $\rho^a h^a$.

The relative variation of total energy is:

$$\frac{E_f - E_i}{E_i} = \frac{\frac{1}{2}(m^o + m^a)\overline{u}^2}{\frac{1}{2} m^o u^{o2}} - 1 = \frac{m^o + m^a}{m^o} \frac{\overline{u}^2}{u^{o2}} - 1.$$

Hence, I replace $\overline{u}$ thanks to equation (26):

$$\frac{E_f - E_i}{E_i} = \frac{-m^a}{m^o + m^a} \simeq \frac{-\rho^a H^a S}{\rho^o H^o S} \simeq -2,5.10^{-3}$$

The numerical results (Fig 5), check well that the potential energies are zero, and that mechanical energy of one fluid is equal to its kinetic energy. They also show a growth of kinetic energy of the atmosphere layer and a decrease of kinetic energy of the ocean. Indeed, as I already know, kinetic energy is transferred from the ocean to the atmosphere as initially atmosphere is at rest.

At the beginning of the simulation, the friction is high ($u^o - u^a \gg 0$), it leads to a significant dissipation of energy, that's why the relative total energy decreases rapidly during the first two

months. After the velocity difference reduces, so the friction is lower and there is no or a little loss of energy. According to the graph (Fig 5), the relative total energy approaches $-2,5.10^{-3}$, which is the analytical obtained value. Please note that this value is independent of the friction.

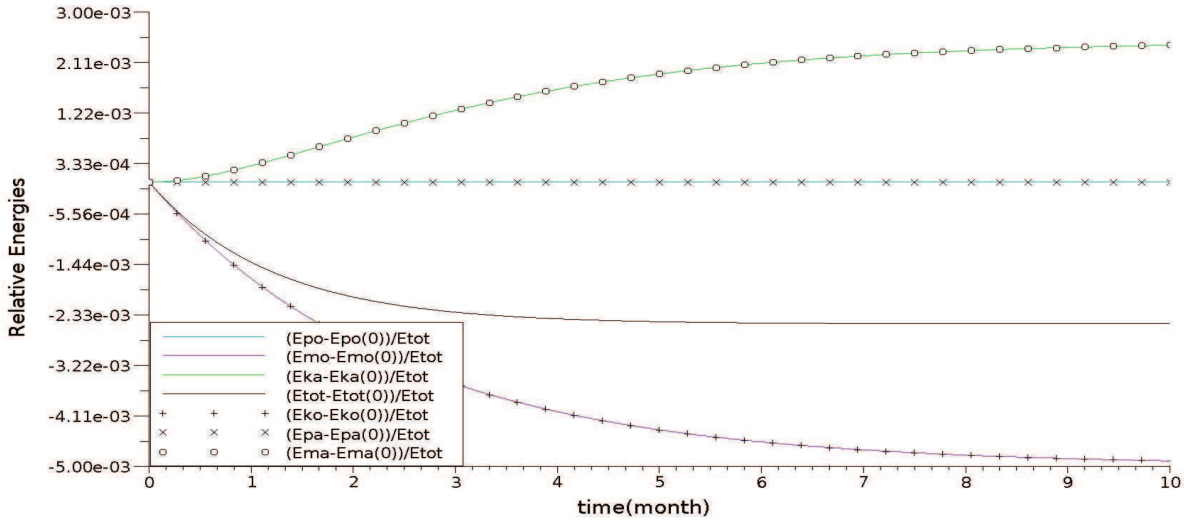


Figure 5: Temporal evolution of energies in a non rotating frame

When the Coriolis force is included some kinetic energy is transformed in potential energy so it is difficult to analytically calculate the energy dissipation. However simulations show a loss of relative energy forty times smaller ($\frac{Ef-Ei}{Ei} = -6.10^{-5}$) (Fig 6). The ocean kinetic energy is now constant, this is balanced with the decrease of the ocean potential energy.

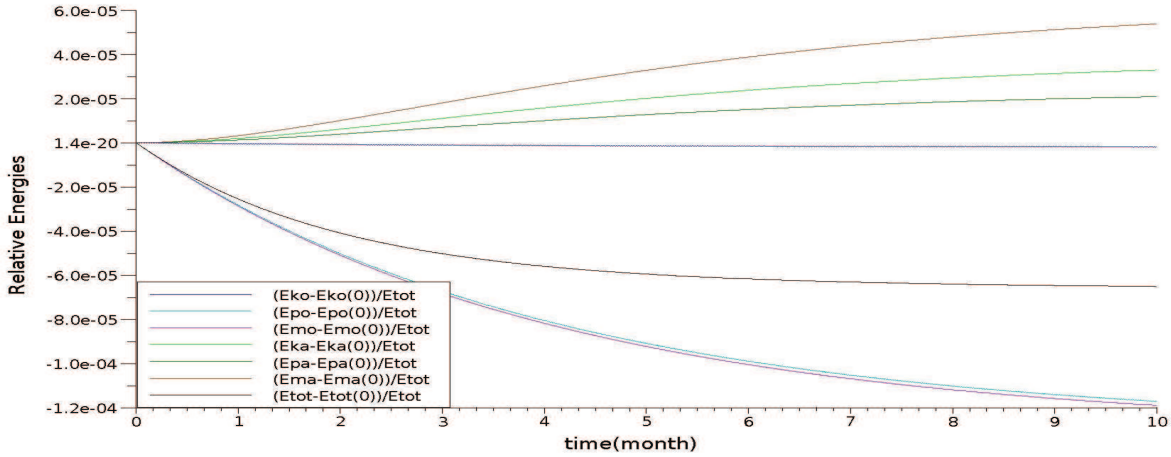


Figure 6: Temporal evolution of energies in a rotating frame

3.4 Energy Fluxes

It is interesting to study the temporal evolution of the kinetic energy flux at a particular point. Here the study is made at the point $y = 750$ kilometers where velocities are maximal in absolute values (Fig 7). Results show well the transfer of kinetic energy from the ocean to the atmosphere.

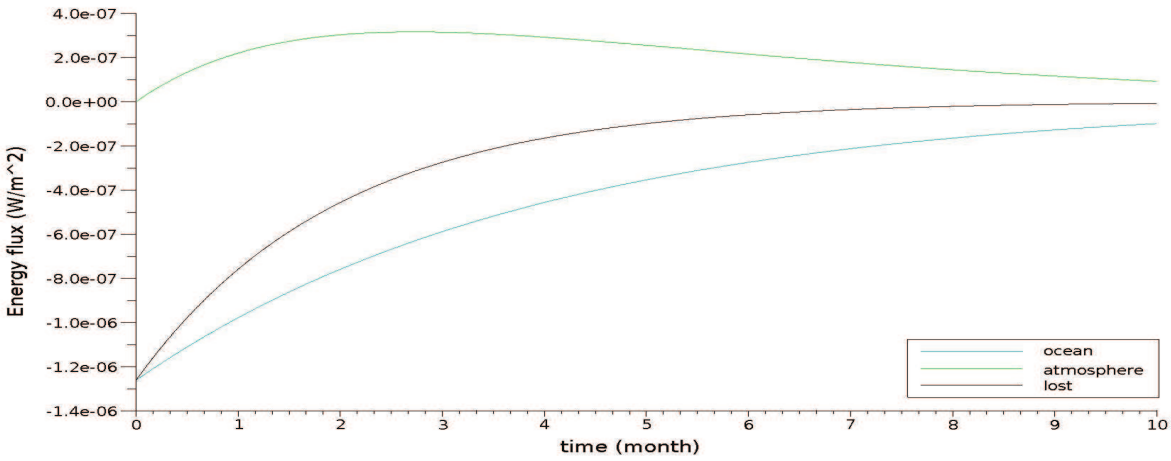


Figure 7: Temporal evolution of energy flux

The simplify system can be described by:

$$\partial_t u^o - f v^o = -\frac{\rho^a \tau (u^o - u^a)}{\rho^o h^o} \tag{27}$$

$$\partial_t v^o + f u^o = -\frac{\rho^a \tau (v^o - v^a)}{\rho^o h^o} \tag{28}$$

$$\partial_t u^a - f v^a = \frac{\tau (u^o - u^a)}{h^a} \tag{29}$$

$$\partial_t v^a + f u^a = \frac{\tau (v^o - v^a)}{h^a} \quad . \tag{30}$$

Kinetic energy fluxes can be calculated thanks to these equations:

$$\partial_t(Eco) = \partial_t(\frac{\rho^o h^o}{2}(u^{o2} + v^{o2})) = \rho^o h^o (u^o \partial_t u^o + v^o \partial_t v^o) \tag{31}$$

$$= -\rho^a \tau (u^{o2} - u^a u^o - v^o v^a + v^{o2}) \tag{32}$$

$$\partial_t(Eca) = \partial_t(\frac{\rho^a h^a}{2}(u^{a2} + v^{a2})) = \rho^a h^a (u^a \partial_t u^a + v^a \partial_t v^a) \tag{33}$$

$$= -\rho^a \tau (u^{a2} - u^a u^o - v^o v^a + v^{a2}). \tag{34}$$

At the beginning, oceanic fluxes are higher in absolute values because atmospheric velocities u^a and v^a are very low, so $\partial_t(Eco) = -\rho^a \tau (u^{o2} - u^a u^o - v^o v^a + v^{o2}) \simeq -\rho^a \tau (u^{o2} + v^{o2}) \ll 0$ (equation (32)). Over time, atmospheric velocities increase so $-u^o u^a - v^o v^a$ rises and ocean energy fluxes

decrease. Initially atmospheric energy fluxes are $0W/m^2$ because $u^a = v^a = 0$. Then u^a and v^a increase, while remaining low, thereby $u^{a2} + v^{a2} \simeq 0$ and $-u^o u^a - v^o v^a$ decreases so atmospheric energy fluxes increase. Over time $u^{a2} + v^{a2}$ rises faster than $-u^o u^a - v^o v^a$ so inflows decrease until oceanic and atmospheric velocities are equals (equation (34)).

The loss of energy is equal to the sum of atmospheric inflows and oceanic outflows:

$$\partial_t(Eco + Eca) = -\rho^a \tau[(u^o - u^a)^2 + (v^o - v^a)^2] = -\rho^a \tau |u|^2, \tag{35}$$

where $|u| = \sqrt{(u^o - u^a)^2 + (v^o - v^a)^2}$.

Similarly, the dissipation of energy decreases in time as it depends on the ocean and atmospheric velocity difference (equation (35)). In fact analytical calculations show that the energy flux lost is proportional to the square of the velocity difference, for a Rayleigh friction.

4 Influences of Ekman veering and quadratic drag law:

In this part I am going to describe evolution differences when I include the Ekman veering and the quadratic drag law to the friction.

4.1 Spatial evolution

The Ekman veering does not change the velocity shape because it just modifies velocity directions.

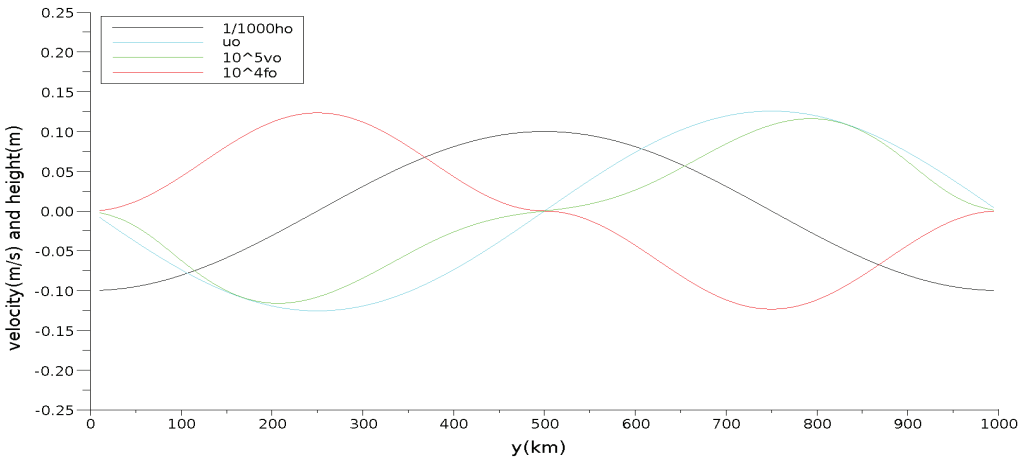


Figure 8: Section of ocean after one day

The quadratic friction leads to an inflexion point at the center of the field (Fig 8), due to the non linear quadratic term. In the meridional direction geostrophic velocity initially vanishes

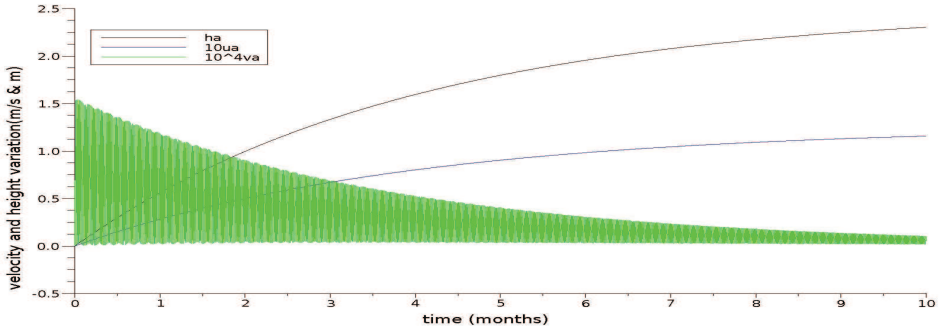
thereby the meridional velocity $v^o(y)$ adopts the same shape as the friction force. As for the linear friction, velocities in the atmosphere generated by the friction adopt the same shape as ocean velocities.

4.2 Temporal evolution and wave generation

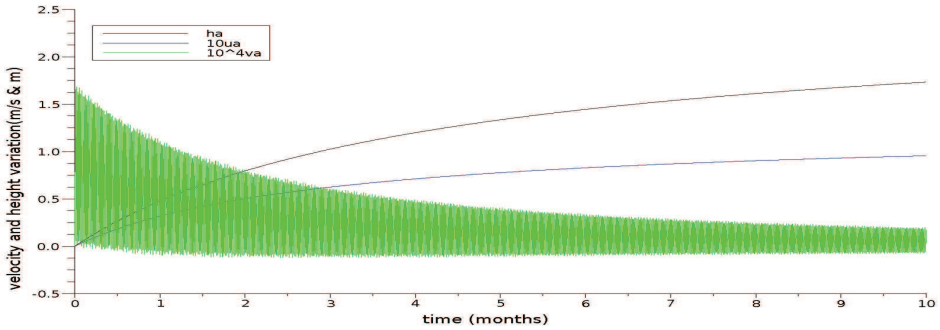
As noted previously, the Ekman veering does not change the standard velocity but only the direction. Temporal evolution of velocities and height variations with and without the Ekman veering have so only one difference in the wave amplitudes (Fig 9 b and c).

Inertia-gravity waves are generated in the ocean and atmosphere and decay exponentially in time for all frictions. After 10 months waves amplitude are very small, lower than 0.2×10^{-4}m/s for all friction (Fig 9). In the ocean and in the atmosphere Ekman veering and quadratic friction increase the wave amplitudes.

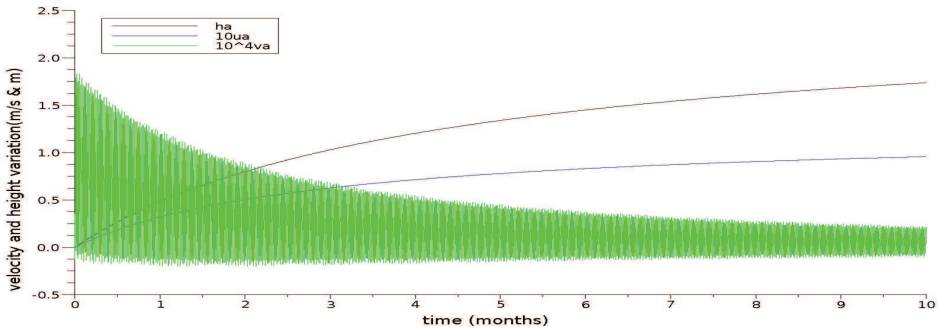
For the quadratic friction, velocities and height variation are slower and smaller than for a linear Rayleigh friction (Fig 9, a and b).



(a) Rayleigh friction



(b) Quadratic friction



(c) Ekman Quadratic friction

Figure 9: Profiles of the atmosphere for different frictions

4.3 Energy Fluxes

The Ekman veering does not change the standard velocity but only the direction, so this term has no influence on fluxes. For quadratic cases the term $|u| = \sqrt{(u^o - u^a)^2 + (v^o - v^a)^2}$ comes into play.

I do the same computation as in the paragraph (3.4), just changing the friction term to:

$$F_x{}^k = -\frac{\rho^a C_d |u|(u^o - u^a)}{\rho^k h^k} \quad . \tag{36}$$

Energy fluxes are equals to:

$$\partial_t(Eco) = -\rho^a C_d |u|(u^{o2} - u^a u^o - v^o v^a + v^{o2}) \tag{37}$$

$$\partial_t(Eca) = -\rho^a C_d |u|(u^{a2} - u^a u^o - v^o v^a + v^{a2}). \tag{38}$$

Inflow in the atmosphere and outflow of the ocean are multiplicated by $|u|$. At the beginning, u^a and v^a are very low so $|u| \simeq \sqrt{u^{o2} + v^{o2}}$ while in the Rayleigh friction ($\tau = Cdu^o$) I have not considered the v^o. That's why energy fluxes are more important initially (Fig 10 and Fig 7).

Then energy is given by the ocean to the atmosphere thereby atmospheric velocities u^a and v^a increase, and $Cd|u|$ decreases whereas the term τ in the Rayleigh friction is constant. This induces a decrease of fluxes more important for the quadratic friction(Fig 10 and Fig 7).

The energy flux dissipated is equal to:

$$\partial_t(Eco + Eca) = -\rho^a C_d |u|[(u^o - u^a)^2 + (v^o - v^a)^2] = -\rho^a C_d |u|^3. \tag{39}$$

It is proportional to $|u|^3$ while it is proportional to $|u|^2$ with a Rayleigh friction. Initially, velocity difference is high so $Cd|u|^3 \gg \tau|u|^2$ and the energy flux lost is higher for quadratic

friction. When the velocity difference becomes very low ($|u| < 10^{-1} m/s$), $Cd|u|^3 \ll \tau|u|^2$, and the energy flux lost is much smaller for the quadratic friction.

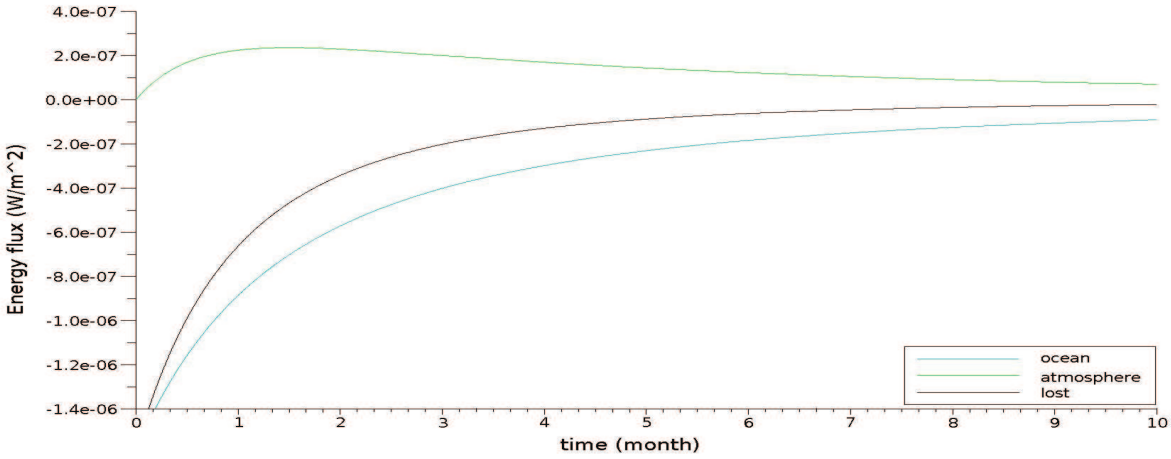


Figure 10: Energy flux for a quadratic friction

4.4 Energy

For the energy mass balance there is no difference when I include the Ekman veering term to the friction as velocities and height variations are the same that for the Rayleigh friction.

For quadratic friction, the potential energy in the ocean and energies in the atmosphere respectively decrease and increase faster during the first two months (Fig 11, Fig 6). In contrast, they decrease and increase slower after this period. This is directly due to the difference of energy fluxes with a quadratic friction.

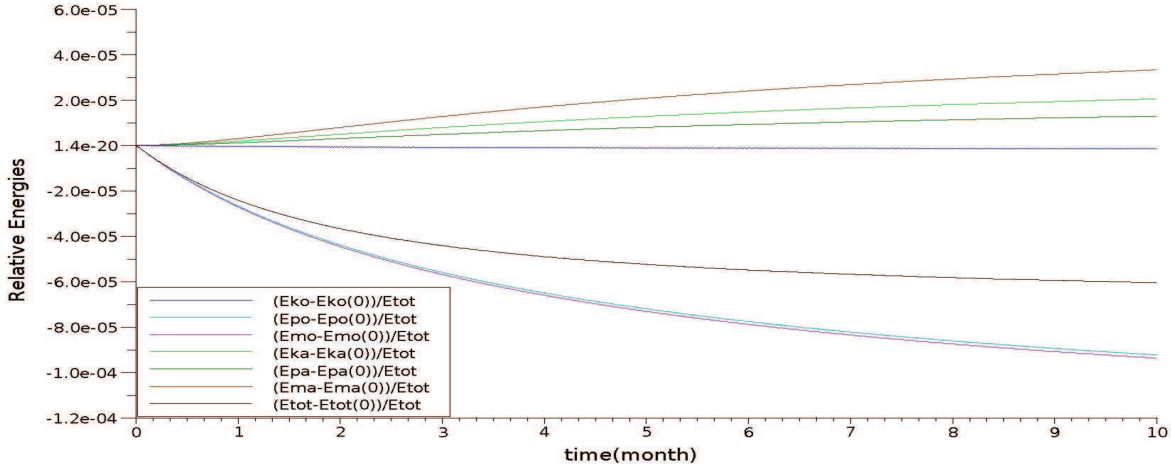


Figure 11: Balance relative energy for a quadratic friction

5 Air-sea interaction due to the friction:

The aim of these simulations is to ease the instability development in the ocean for study the air-sea interaction in two dimensions, in presence of eddies. The resolution is increased to 1.95 kilometers, there are 512×512 points.

First it is necessary to lower the ocean dynamic viscosity. I reduced the ocean dynamic viscosity to have Reynolds numbers ($Re = U^k L/\nu^k$) of the same magnitude in the ocean and atmosphere ie for $\nu^o = 1m^2/s$.

Then narrow oceanic and atmospheric currents must be imposed in different direction from the outset. In the atmosphere it only depend on y-direction unlike in the ocean it only depends on x-direction, so they are perpendicular.

They are initially defined by the height variations which represent the first terms in the Fourier series of a sawtooth function (Fig 12):

$$\begin{aligned} \eta_o^o(x,y) &= 100m \times (sin(2\pi x) - \frac{1}{3}sin(4\pi x) + \frac{1}{5}sin(6\pi x) - \frac{1}{7}sin(8\pi x) + \frac{1}{9}sin(10\pi x)) \\ \eta_o^a(x,y) &= 300m \times (sin(2\pi y) - \frac{1}{3}sin(4\pi y) + \frac{1}{5}sin(6\pi y) - \frac{1}{7}sin(8\pi y)), \end{aligned}$$

and the geostrophic velocity field which is associated (ie u_o^o and v_o^a are 0).

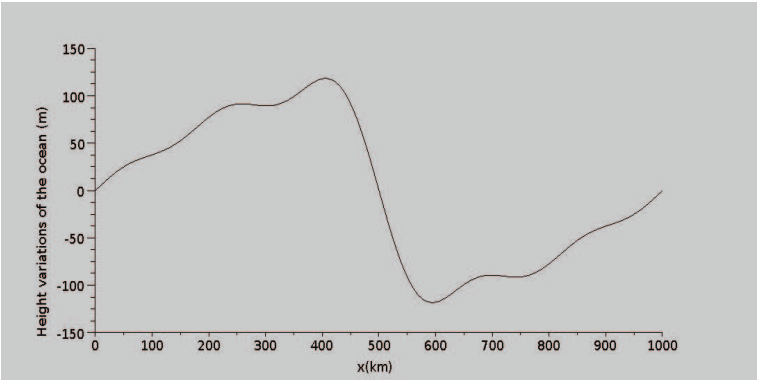


Figure 12: Sawtooth function of ocean height variations

A restoring in the atmosphere to the initial (sawtooth) layer thickness is essential. The restoring acts to force the average (in the x-direction) of the layer thickness projected on the sawtooth profile towards its initial value. This is accomplished by projecting the observed average in the x-direction on the sawtooth profile followed by an adding or subtracting of the sawtooth profile on the whole domain to restore the projection towards a constant level. The restoring time is two days. Such kind of (large-scale) restoring affects the large scale dynamics without directly influencing the small scales which can evolve freely.

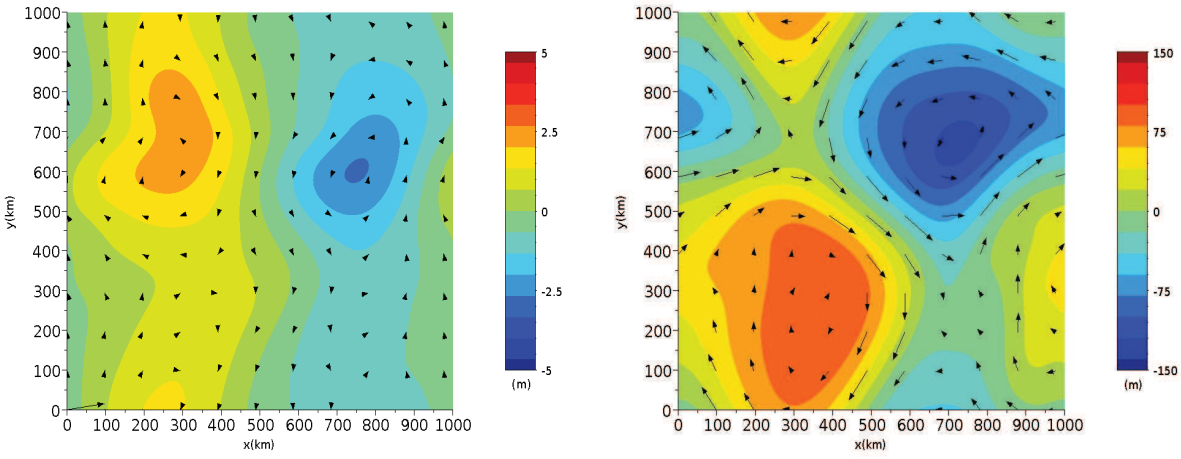
Another restoring is done in the ocean to avoid the explosion of the simulation. Indeed in these simulations there is no dissipation of energy in the ocean whereas in the reality continents and exchanges with the deep water layer dissipate mechanical energy. In order to not disturb the air-sea interaction, the restoring time is very long (1000 days).

5.1 Atmosphere influence on the ocean for a linear friction

In this part the friction is considered as linear.

5.1.1 Global evolution of the system

After few days, little instabilities ($-5m < \eta^a < 5m$) appear in the atmosphere (Fig 13,a). They have typical velocities of about 6×10^{-2}m/s. They can be observed when the average height variation along the x-direction for each y is removed to each height variation at the y coordinate. In the same way, mean current along the x-direction for one y is removed to each velocity at the y coordinate. This allows to observe instabilities hidden by the mean current which is sixty times faster(about 3.7m/s) and leads to height variations ten times higher (about 70m). After several hundred days two oceanic instabilities with diameters about 500 kilometers are set up (Fig 13,b). These two eddies rotate in opposite directions and are located at each extremity of the field, to better develop. Indeed if rotations were in the same direction at this scale (few hundred kilometers), they would form a single eddy.



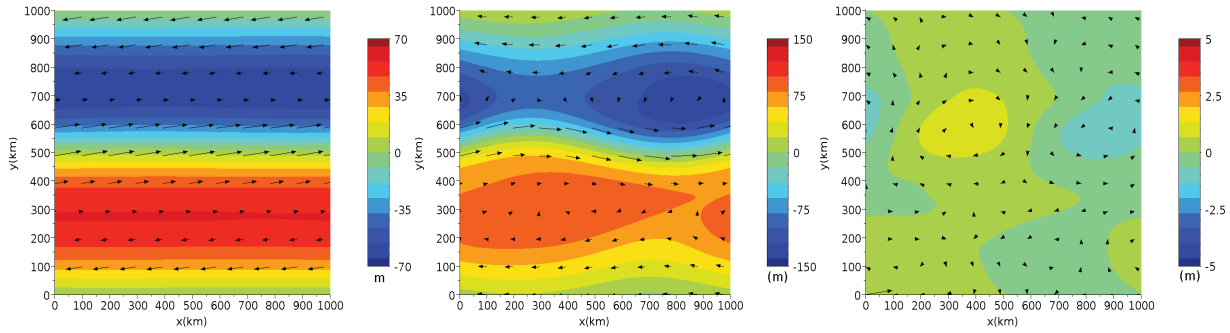
(a) for the atmosphere without the mean current

(b) for the ocean

Figure 13: Maps of height variations and velocity field after 2 years

After few years for example 4 years and 10 months (Fig 14), the dynamics of the ocean

depends almost only on the y-direction, ie it is nearly aligned with the large scale atmospheric dynamics. At the same time, little atmospheric eddies start to disappear. If the simulation is continued the ocean will have a dynamics similar to the atmospheric dynamics.



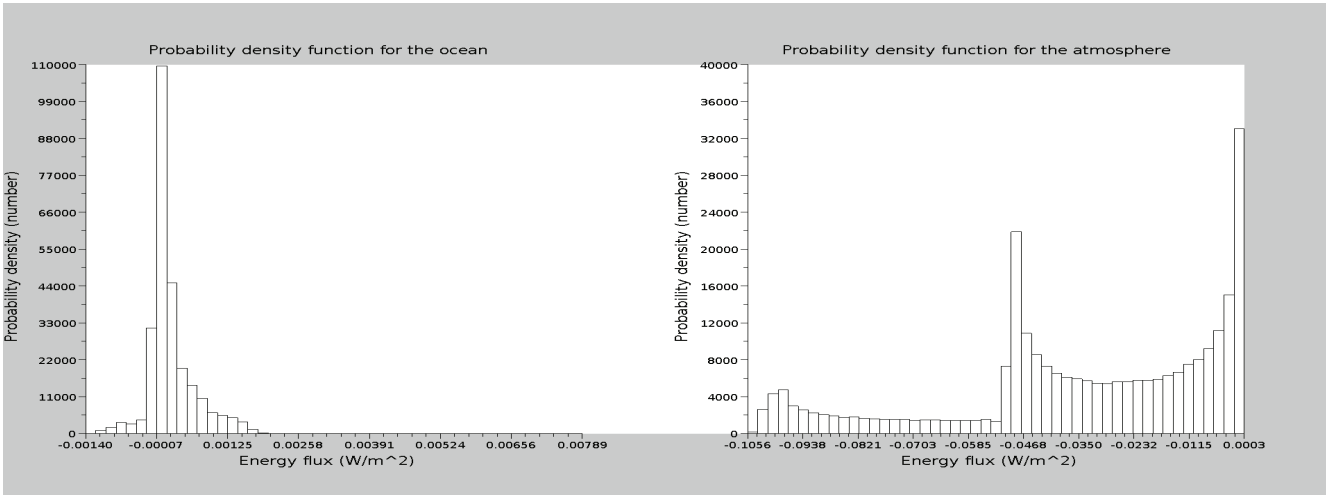
(a) for the atmosphere (b) for the ocean (c) for the atmosphere without the mean current

Figure 14: Maps of height variations and velocity field after 4 years and 10 months

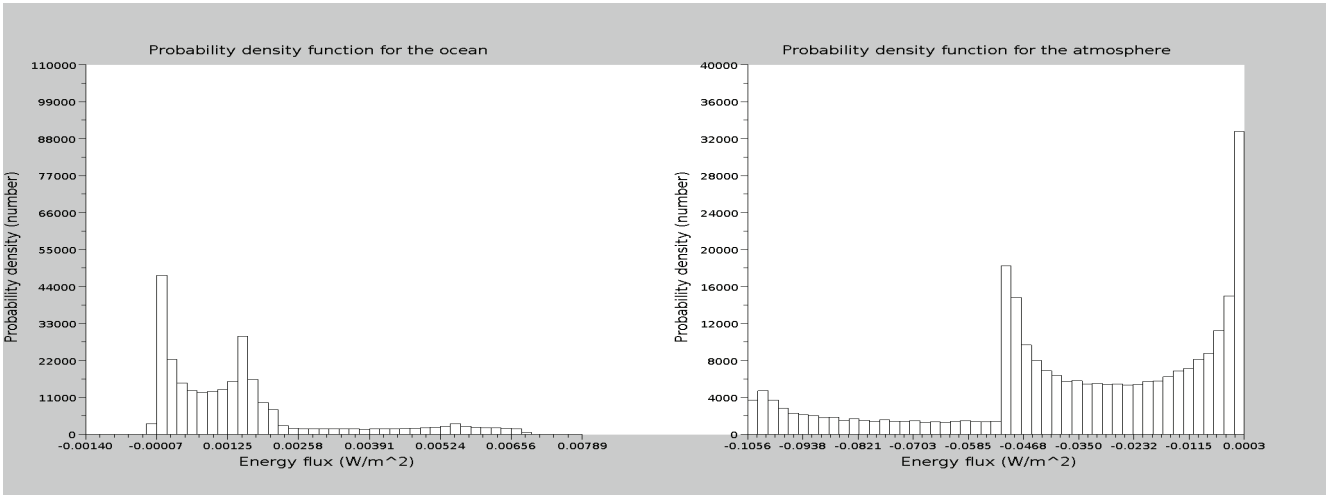
5.1.2 Energy fluxes

The evolution of probability density function of energy fluxes (Fig 15) confirms that the ocean adjusts to the atmosphere. Indeed the same trimodal form is observed in the atmosphere and in the ocean after several years (Fig 15, b), with a third peak much smaller. Energy fluxes in the ocean are the symmetrical of atmospheric energy fluxes. Globally the atmosphere losses energy and transfers it to the ocean. However, absolute energy fluxes outgoing of the atmosphere are higher than the absolute energy fluxes incoming into the ocean. As ocean mass is very big, energy given by the atmosphere and absorbed by the ocean is very inefficiently, so a lot of energy is dissipated into heat fluxes.

The integral of the probability density function for one time step gives the dissipated energy flux in the field. The dissipate energy flux is strong initially ($2.40\times10^6 Watts$) and decreases rapidly to reach a value of $3.59\times10^4 Watts$ after one week. After 4 years and 10 months the dissipate energy flux in all the field is equal to $3.55\times10^4 Watts$. The average total dissipated energy, in all the field, during 4 years and 10 months is equal to $2.63\times10^{11} Joules$.



(a) after 3 months



(b) after 4 years and 10 months

Figure 15: Probability density function of energy fluxes

The probability density function of energy fluxes can be linked to the energy flux maps (Fig 15 b and Fig 16) to determine where the energy fluxes the most numerous are located on the field.

In the atmosphere, the main part of energy fluxes is between 3×10^{-4} and -1.7×10^{-3} W/m^2 so they are very slightly negative or extremely slightly positive. These fluxes are located between 200 and 350 kilometers and between 650 and 800 kilometers in the y-direction (Fig 16, blue-green color). The second major part of energy fluxes is moderately negative about $-4.9 \times 10^{-2} W/m^2$. They are mainly located at the north between 900 and 1000 kilometers and at the south of the field between 0 and 100 kilometers (Fig 16, azure color). The third major part corresponds to an

important loss of energy about $-1.0 \times 10^{-1} W/m^2$ located at the center of the field between 500 and 550 kilometers (Fig 16, navy blue color).

In the same way, the main part of energy fluxes of the ocean is almost zero, and is located at the south between 150 and 400 kilometers and at the north between 650 and 800 kilometers (Fig 16, blue-green color). The second corresponds to incoming energy fluxes of about $1.5 \times 10^{-3} W/m^2$ mainly located at the north between 850 and 1000 kilometers and at the south between about 0 and 70 kilometers (Fig 16, yellow-green and yellow color). The third peak is hardly visible but it represents strong incoming fluxes about $5.5 \times 10^{-3} W/m^2$ in the center of the field around 500-550 kilometers (Fig 16, orange color). The main energy fluxes incoming in the ocean is located in the same areas where main energy fluxes go out of the atmosphere.

This energy flux repartition can be explained by the oceanic and atmospheric structures. Because of oceanic eddy locations, atmospheric and oceanic velocities are slow and substantially perpendicular between 150-350 kilometers and 650-800 kilometers (Fig 13), so energy fluxes are very low (equations (32)-(34)). Unlike at the north, the south and at the center of the field, oceanic and atmospheric velocities are in the same direction, and ocean velocities are much slower than atmospheric velocities, which leads to important energy fluxes incoming on the ocean and outgoing of the atmosphere (equations (32)-(34)). Moreover at the center, atmospheric velocities are much faster so energy fluxes are the highest in this area.

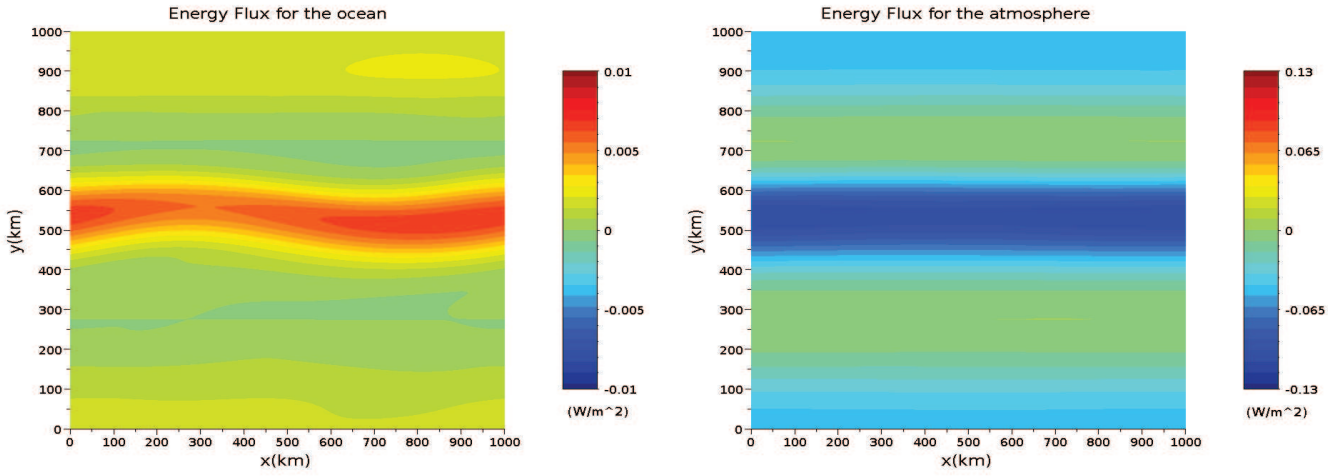


Figure 16: Maps of energy fluxes after 4 year and 10 months

Few very small areas in the atmosphere, gain some energy. Obviously in these areas the ocean energy fluxes are negative because it is the ocean that provides the energy going to the atmosphere. These areas are located, around 280 kilometers in the y-direction and between 0 to 100 kilometers in the x-direction and around 725 kilometers in the y-direction and between 580

and 800 kilometers in the x-direction (Fig 16, green color). They are located where atmospheric velocities (about -0.01 m/s) are lightly slower than oceanic velocities (about -0.05 m/s). With a linear friction, there is an exchange from the ocean to the atmosphere but it is highly localized and very low.

It is also interesting to look for energy fluxes which the maximum of energy is lost for the atmosphere and won for the ocean (Fig 17). They can also be linked to the energy flux maps (Fig 16) to determine where energy is lost and gained in the oceanic and atmospheric field. The trimodal form is still observed for the atmosphere whereas the form is bimodal in the ocean. In the atmosphere most of energy is lost, for energy fluxes around $5\times10^{-2}W/m^2$ located in the south and in the north of the field where the outgoing energy fluxes are medium but rather numerous. There is also a significant loss in the middle of the area where energy fluxes are the strongest about $1 \times 10^{-1}W/m^2$ but less numerous. For the ocean most of energy is gained in the north and in the south of the field where incoming energy fluxes are medium about $1.6 \times 10^{-3}W/m^2$ but rather numerous. An other part of energy is gained in the center of the field where incoming energy fluxes are the strongest about $5.6 \times 10^{-3}W/m^2$. Finally in the main part of the field energy fluxes are nearly zero so there is no energy exchange between the atmosphere and the ocean.

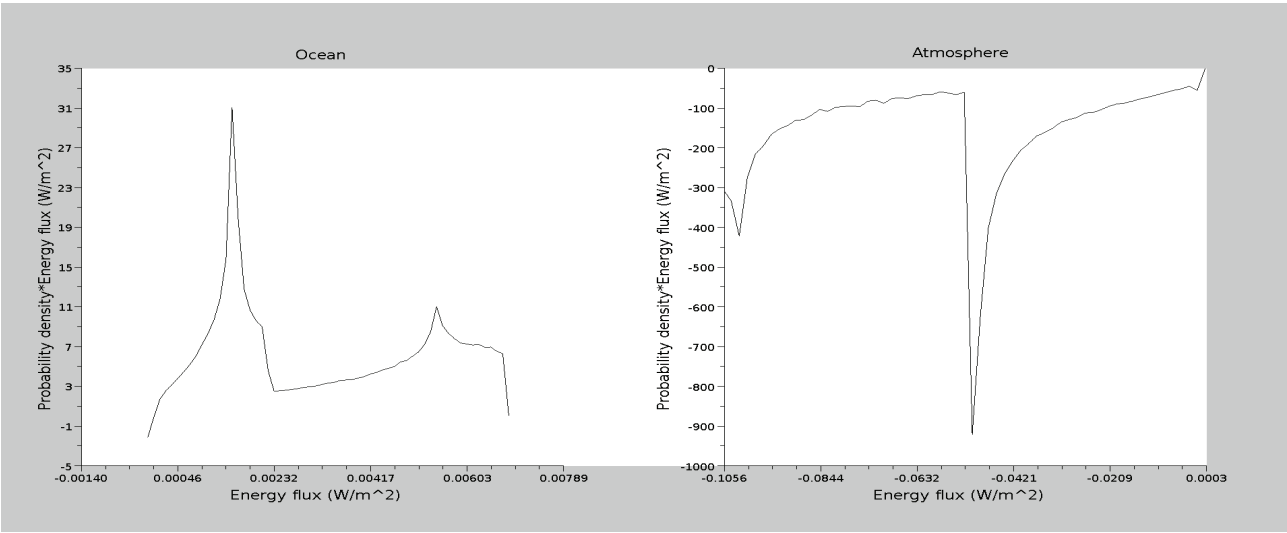


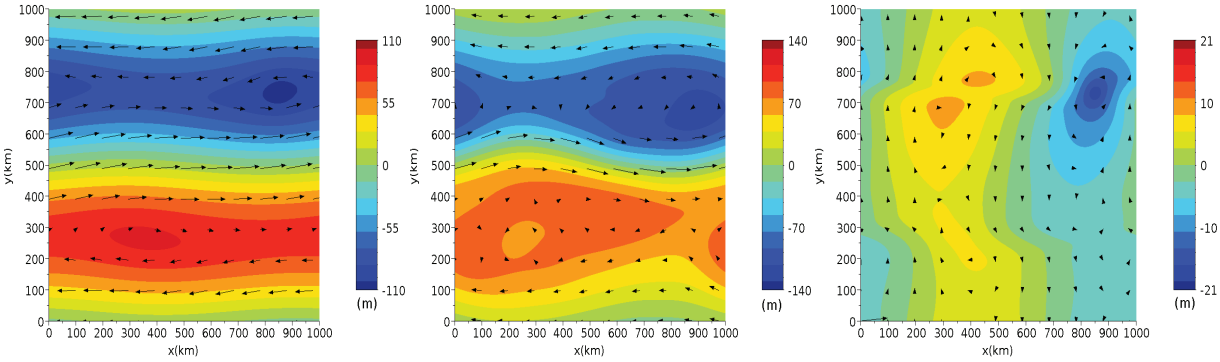
Figure 17: Energy lost or gained in function of energy flux values after 4 years and 10 months

5.2 Ocean influence on the atmosphere for a quadratic drag law

In this part the friction is considered as quadratic.

5.2.1 Global evolution of the system

As in the linear case, the ocean tries to align with the atmosphere and adopt its dynamics (Fig 18, a and b). Atmospheric perturbation are more important ($-21m < \eta^a < 21m$), that is why they are also apparent in the map of atmospheric height variations (Fig 18, a). They have typical velocities of 0.95 m/s, which is four times speeder than typical velocities of ocean eddies.



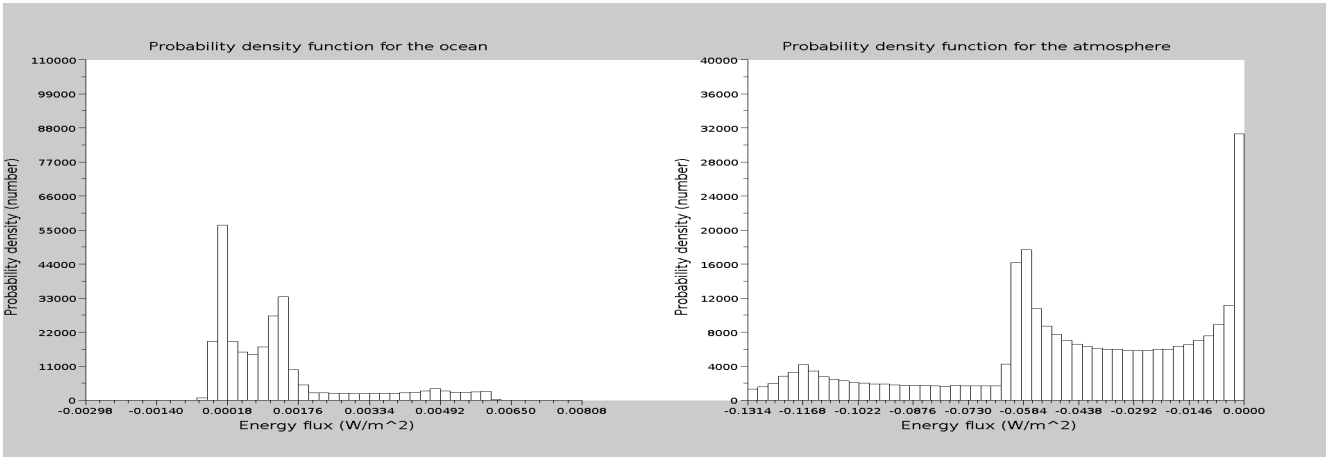
(a) for the atmosphere (b) for the ocean (c) for the atmosphere without the mean current

Figure 18: Maps of height variations and velocity field after 4 years and 10 months with a quadratic drag law

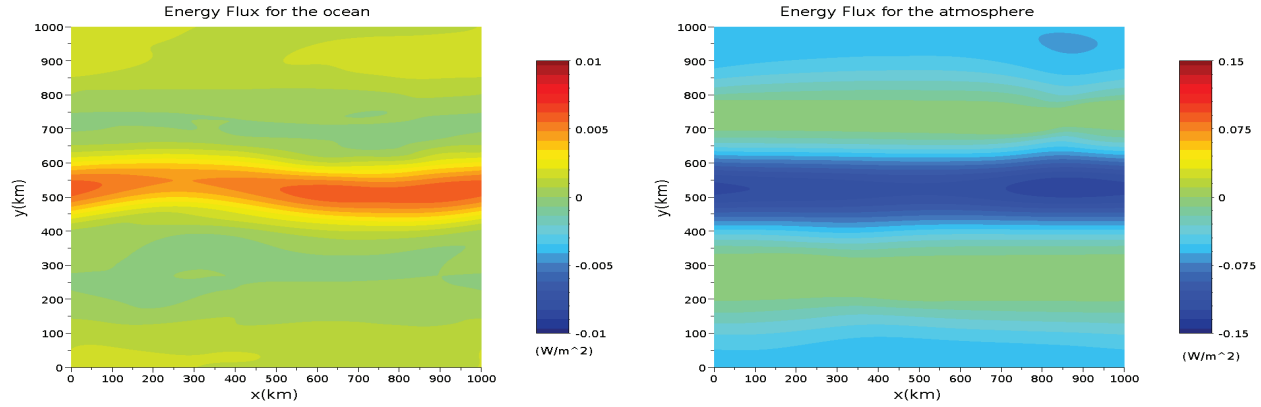
5.2.2 Energy fluxes

Energy fluxes are studied in the same way as in the previous subsection.

The same shape is observed for the probability density function of energy fluxes, for the linear and the quadratic friction, although the third peak is more pronounced with a quadratic friction (Fig 19 a and Fig 15 b). The number of energy fluxes null or outgoing of the ocean is higher, 55000 with the quadratic law friction versus 46000 with the linear friction. Globally these fluxes are still localized in the same areas. Indeed atmospheric eddies modify the atmospheric velocity direction at the north between 650 and 800 kilometers and at the south between 150 and 300 kilometers. In these areas atmospheric and oceanic velocities are more opposed hence the increase of negative flow in the ocean. In the central area, atmospheric instabilities lead to atmospheric velocities better aligned with the oceanic velocities. High energy fluxes in absolute values in the ocean and in the atmosphere are so more numerous in this area that is why the trimodal form is more pronounced.



(a) Probability density function



(b) Maps of energy flux

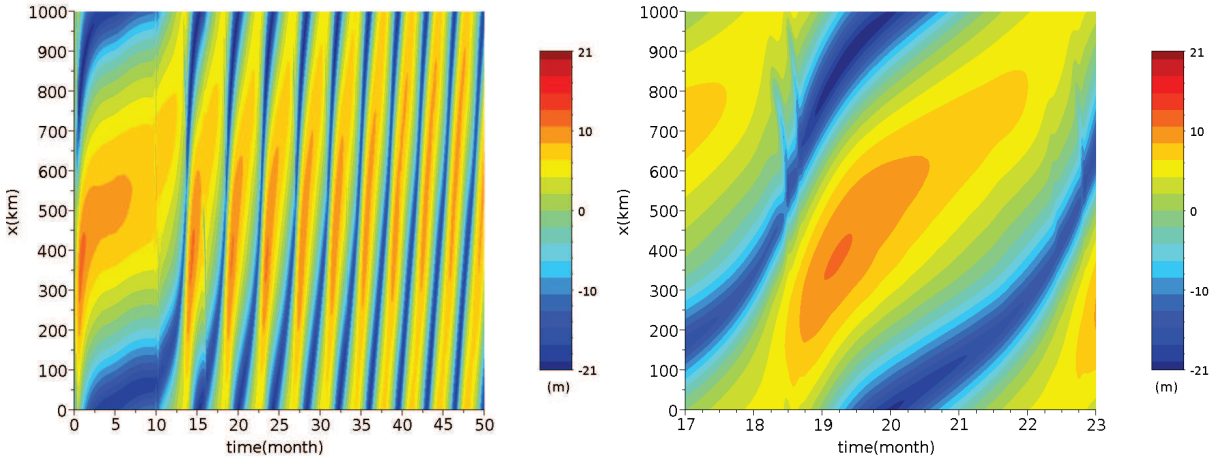
Figure 19: Probability density function and maps of energy flux at 4 years and 10 months

5.2.3 Hovmöller diagram

A Hovmöller diagram is used to plot the time evolution of the height variation profile at y=750 kilometers along the x-direction in the atmosphere (Fig 20). The axes of the Hovmöller diagram are the time in abscissa, the x-direction in ordinate, and the value of the height variation field represented through colors. A zoom of the Hovmöller diagram between month 17 and 23 (Fig 20 b) allows to calculate the typical displacement of the perturbation.

It shows a displacement of atmospheric instabilities over time (Fig 20). The quadratic drag law friction generates stronger atmospheric eddies and also a higher mean wind of the order of several meters by seconds. Indeed these eddies are advected by the mean wind. The atmospheric cyclonic eddy (negative height) moves slowly (4.37×10^{-2} m/s) when it is above the ocean cyclonic eddy because this is a stable configuration(Fig 20 b). Then after two months, the mean current succeeds in moving the eddies, so the atmospheric eddy is disconnected from the ocean eddy

and moves in the x-direction with a velocity of 1.35×10^{-1}m/s. The atmospheric anticyclonic perturbation (positive height) moves quickly when it is above the ocean cyclonic eddy and slowly when it exceeds the oceanic eddy. Simultaneously, perturbations become stronger when they move slowly. In fact atmospheric instabilities move slowly and grow when they are superposed with ocean eddies rotating in the same direction.



(a) Hovmöller diagram (b) Zoom of Hovmöller diagram

Figure 20: Hovmöller diagrams of height variations

Just before the acceleration of the cyclonic atmospheric disturbance, there are two little structures moving very quickly (4.9×10^{-1}m/s) in the opposite direction of the eddy movement.

These phenomenon become less and less pronounced after several years. Indeed, after 2 years the ocean dynamics tries to adopt the atmospheric dynamics so oceanic instabilities are less strong and less well localized. The capture, of the atmospheric eddy by the oceanic eddy with the same rotating direction, during a finite period is less efficient and the horizontal velocity becomes homogeneous.

Although there is only an exchange of mechanical energy from the atmosphere to the ocean, the cyclonic ocean eddy attracts the atmospheric cyclonic eddy periodically and captures it for up to two months. This phenomena has a typical periodicity of about 6 months which is much longer than the restoring time of the atmosphere and much shorter than the restoring time of the ocean. Atmospheric dynamics has thereby a longtime variability.

6 Conclusion

My internship consisted in constructing an idealized but dynamically consistent model for air-sea exchange of inertia, taking into account the velocity difference between the atmosphere and the ocean. As far as we know, today, none of the models used for weather prediction or climates studies takes into consideration the effect of ocean currents on the atmosphere (Achim Wirth). Indeed the only air-sea action of the ocean on the atmosphere considered in models is the SST. Thereby, it is of a great importance, for atmospheric and oceanic circulation models, to improve the knowledge on this neglected phenomenon.

The first part of my internship was devoted to the elaboration of the physical, mathematical and numerical models. The numerical model is implemented as a Fortran code and it executed on the multiprocessor machines of the LEGI in "batch" mode. Then I validated the model with idealized simulations and analytical calculations. I showed, that inertial gravity waves are generated in the ocean and atmosphere which decay in time.

The last part allowed to discover the importance of the friction in the air-sea interaction in two dimensions. As we known the atmosphere transfers mechanical energy to the ocean and drives the ocean circulation. I find that atmospheric eddies are generated by the ocean dynamics although there is no transfer of energy from the ocean to the atmosphere. This phenomena is stronger for the quadratic friction law. These eddies are then advected by the mean wind. The cyclonic ocean eddy attracts the atmospheric cyclonic eddy periodically and captures it for up to two months. This lead to a longtime variability of atmospheric dynamics with a typical period of about 6 months.

For the continuation of this work, it would be interesting to increase the spatial resolution of the model to less than 1 kilometer. This would allow to observe the air-sea interaction in the presence of structures at smaller scales with a stronger turbulence. I could not do it during my internship because the increase of the number of points to 1024×1024 (ie multiplying the number of points by 4 and decreasing the time step by a factor of 2, CFL condition), leads to simulation times of several months, that was far too long for an internship.

References

[1] D. B. Chelton, M. G. Schlax, M. H. Freilich, R. F. Milliff, *(2004), Satellite Measurements Reveal Persistent Small-Scale Features in Ocean Winds.* Science Mag, VOL 303, p 978-983.

[2] D. B. Chelton, S. P. Xie, *(2010), Coupled Ocean-Atmosphere Interaction at Oceanic Mesoscales.* Oceanography, VOL 23 No.4, p 52-69.

[3] T. H. A. Duhaut and D. N. Straub, *(2006), Wind Stress Dependence on Ocean Surface Velocity: Implications for Mechanical Energy input to Ocean circulation.* Journal of physical oceanography, VOL 36, p 202-211.

[4] IPCC report, *(2007), Climate Change 2007: The Physical Science Basis.* $http://www.ipcc.ch/publications\_and\_data/ar4/wg1/en/contents.html$, CH 8

[5] F. Roquet and C. Wunsch, G. Madec, *(2011), On the Patterns of Wind-Power Input to the Ocean Circulation.* Journal of physical oceanography, VOL 41, p 2328-2342.

[6] R. B. Scott and Y. Xu *(2009), An update on the wind power input to the surface geostrophic flow of the World Ocean.* Deep Sea Research Part I: Oceanographic Research Papers, VOL 56, Issue 3, p 295-304.

[7] S. D. Smith, *(1988), Coefficients for Sea Surface Wind stress, Heat Flux, and Wind Profiles as a Function of Wind Speed and Temperature* Journal of geophysical research, VOL 93, p 15467-15472

[8] A. Wirth, *(2011), A Guided Tour Trough Physical Oceanography*

[9] Wu Jin *(1982), Wind-Stress Coefficients Over Sea Surface From Breeze to Hurricane.* Journal of geophysical research, VOL 87, p 9704-9706.

[10] Zhai, X., H. L. Johnson, D. P. Marshall and C. Wunsch, *(2012), On the Wind Power Input to the Ocean General Circulation.* Journal of Physical Oceanography, in press, 2012.

A) APPENDIX 1: Numerical model in Fortran

```
c234567891234567893123456789412345678951234567896123456789123456789
c     ****************************************************************
c     **                                                            **
c     **                PROGRAM                                     **
c     **                                                            **
c     **             by AIMIE                                       **
c     **                                                            **
c     **  date : 13.03.2012                                         **
c     **                                                            **
c     **  version : 008                                             **
c     **                                                            **
c     **    code:                                                   **
c     **                                                            **
c     **                                                            **
c     **  all units sgi                                             **
c     **                                                            **
c     ****************************************************************
c
c     ****************************************************************
      PROGRAM Fquadratique
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,nxm2,ny,nym1,nym2
      PARAMETER   (nx=513,nxm1=nx-1,nxm2=nx-2,
     &            ny=513,nym1=ny-1,nym2=ny-2)
      INTEGER ::  ix,iy,nt,nti
      INTEGER ::  it,step,expnum,step0
      REAL(8) ::  dt,dx,dy,lx,ly,x,y,rdx2,rdy2,r2dx,r2dy,dtd2
     &           ,ho0,ha0,nuo,nua,go,ga,tau,kappaa,u,pi,ffa,hfa
     &           ,kappao,f,rhoa,rhoo,twopi,month,Epo,Epa,Eca,Eco
     &           ,Ema,Emo,Etot,rrhoo,rrhoa,CFLa,CFLo,lambdaf,Cd
     &           ,lambdao
      REAL(8) ::  hon(0:nx,0:ny),hoo(0:nx,0:ny),uon(0:nx,0:ny)
     &           ,uoo(0:nx,0:ny),von(0:nx,0:ny),voo(0:nx,0:ny)
     &           ,han(0:nx,0:ny),hao(0:nx,0:ny),uan(0:nx,0:ny)
     &           ,uao(0:nx,0:ny),van(0:nx,0:ny),vao(0:nx,0:ny)
     &           ,fa(0:nx,0:ny)
      COMMON / newcom  /  hon,uon,von,han,uan,van
      COMMON / encom  /  Eco,Eca,Epo,Epa,Ema,Emo,Etot
c     ****************************************************************
c     **                                                            **
c     **                          Def of the variables              **
c     **                                                            **
c     ****************************************************************
      expnum=000118
      nt=500   !500jr
      nti=4*60*24 !24h
      step=0
      lx=1.D6  ! m
      ly=1.D6  ! m
      dx=lx/DFLOAT(nxm1)
      dy=ly/DFLOAT(nym1)
      r2dx=1.D+0/(2.D+0*dx)
      r2dy=1.D+0/(2.D+0*dy)
      rdx2=dx**(-2)
```

```
      rdy2=dy**(-2)
      dt=15.  ! s
      dtd2=dt/2.0D0
c      kappao=1.D-3  ! m**2 /s
c      kappaa=1.D-5  ! m**2 /s
      kappao=1  ! m**2 /s
      kappaa=100  ! m**2 /s
      nuo=kappao
      nua=kappaa
      rhoo=1000. ! kg/m**3
      rhoa=1.  ! kg/m**3
      rrhoo=1.
      rrhoa=1.D-3
      go=2.D-2 !m/s**2
      ga=0.8   !m/s**2
      ho0=200. !m
      ha0=500. !m
      CFLa=sqrt(ga*ha0)*(dt/dy)
      CFLo=sqrt(go*ho0)*(dt/dy)
      u=10  !vitesse caracteristique atmo
      f=1.D-4 !s**-1
      pi=3.141592653589793
      twopi=3.141592653589793*2.0D+0
      tau=8.D-4*u
      Cd=(0.8+0.065*u)*(1.D-3)
      WRITE(26,*) lx,ly,nx,ny,CFLo,CFLa
      lambdaf=1./(2.*24.*3600.)
      lambdao=1./(1000.*24.*3600.)
c     ****************************************************************
c     **                                                            **
c     **                      Initial conditions                    **
c     **                                                            **
c     ****************************************************************
      hon=0.0
      hoo=0.0
      uon=0.
      uoo= 0.
      von=0.
      voo= 0.
      han=0.0
      hao=0.0
      uan=0.
      uao= 0.
      van=0.
      vao= 0.
      Eco=0.
      Eca=0.
      Epo=0.
      Epa=0.
c     ****************************************************************
c     **                                                            **
c     **                      h initial                             **
c     **                                                            **
c     ****************************************************************
      DO  ix=0,nx
      DO  iy=0,ny
      x=(ix)/DFLOAT(nxm1)
      y=(iy)/DFLOAT(nym1)
      hoo(ix,iy)=100*(Dsin(twopi*x)-1./3.*Dsin(2*twopi*x)
     &+1./5.*Dsin(3*twopi*x)-1./7.*Dsin(4*twopi*x)
     &+1./9.*Dsin(5*twopi*x))
      hao(ix,iy)=300*(Dsin(twopi*y)-1./3.*Dsin(2*twopi*y)
```

```
     &+1./5.*Dsin(3*twopi*y)-1./7.*Dsin(4*twopi*y))
      ENDDO
      ENDDO
      hon=hoo
      han=hao
c     ****************************************************************
c     **                                                            **
c     **                       forcing atmo                         **
c     **                                                            **
c     ****************************************************************
      ffa=0.0
      DO  ix=0,nx
      DO  iy=0,ny
      x=(ix)/DFLOAT(nxm1)
      y=(iy)/DFLOAT(nym1)
      fa(ix,iy)=300*(Dsin(twopi*y)-1./3.*Dsin(2*twopi*y)
     &+1./5.*Dsin(3*twopi*y)-1./7.*Dsin(4*twopi*y))
      ffa= ffa+fa(ix,iy)**2
      ENDDO
      ENDDO
c     ****************************************************************
c     **                                                            **
c     **                    geostrophic equilibrium                 **
c     **                                                            **
c     ****************************************************************
      DO  ix=1,nxm1
      DO  iy=1,nym1
      uoo(ix,iy)=-go/f*(hon(ix,iy+1)-hon(ix,iy-1))*(r2dy)
      voo(ix,iy)=go/f*(hon(ix+1,iy)-hon(ix-1,iy))*(r2dx)
      uao(ix,iy)=-ga/f*(han(ix,iy+1)-han(ix,iy-1))*(r2dy)
      vao(ix,iy)=ga/f*(han(ix+1,iy)-han(ix-1,iy))*(r2dx)
      ENDDO
      ENDDO
c     ****************************************************************
c     **                          Boundary conditon                 **
c     ****************************************************************
      DO  ix=1,nxm1
      hoo(ix,ny)=hoo(ix,1)
      hoo(ix,0)=hoo(ix,nym1)
      uoo(ix,ny)=uoo(ix,1)
      uoo(ix,0)=uoo(ix,nym1)
      voo(ix,ny)=voo(ix,1)
      voo(ix,0)=voo(ix,nym1)
      hao(ix,ny)=hao(ix,1)
      hao(ix,0)=hao(ix,nym1)
      uao(ix,ny)=uao(ix,1)
      uao(ix,0)=uao(ix,nym1)
      vao(ix,ny)=vao(ix,1)
      vao(ix,0)=vao(ix,nym1)
      ENDDO

      DO  iy=1,nym1
      hoo(nx,iy)=hoo(1,iy)
      hoo(0,iy)=hoo(nxm1,iy)
      uoo(nx,iy)=uoo(1,iy)
      uoo(0,iy)=uoo(nxm1,iy)
      voo(nx,iy)=voo(1,iy)
      voo(0,iy)=voo(nxm1,iy)
      hao(nx,iy)=hao(1,iy)
      hao(0,iy)=hao(nxm1,iy)
      uao(nx,iy)=uao(1,iy)
      uao(0,iy)=uao(nxm1,iy)
```

```
      vao(nx,iy)=vao(1,iy)
      vao(0,iy)=vao(nxm1,iy)
      ENDDO
      uan=uao
      van=vao
      uon=uoo
      von=voo
c     ****************************************************************
c     **                                                            **
c     **                       READ RESTART                         **
c     **                                                            **
c     ****************************************************************
      step0=0
c      CALL INRESTSUB(step0,expnum)
c     ****************************************************************
c     **                                                            **
c     **                       INITIAL ENERGY                       **
c     **                                                            **
c     ****************************************************************
      Do ix=1,nxm1
      Do iy=1,nym1
      Eco=Eco+(ho0+hon(ix,iy))*(uon(ix,iy)**2+von(ix,iy)**2)
      Eca=Eca+(ha0+han(ix,iy))*(uan(ix,iy)**2+van(ix,iy)**2)
      Epo=Epo+hon(ix,iy)**2
      Epa=Epa+han(ix,iy)**2
      ENDDO
      ENDDO
      Eco=Eco*(1./2)*rhoo*dx*dy
      Eca=Eca*(1./2)*rhoa*dx*dy
      Epo=Epo*(1./2)*rhoo*go*dx*dy
      Epa=Epa*(1./2)*rhoa*ga*dx*dy
      Ema=Epa+Eca
      Emo=Epo+Eco
      Etot=Eca+Eco+Epa+Epo
      CALL OUTSUB(0,expnum)
c      CALL OUTSUB2(0,expnum)
c      CALL OUTSUB3(0,expnum)
      DO  step=step0+1,step0+nt
      WRITE(26,*) step
      DO  it=1,nti
      hfa=0.0
      DO  ix=1,nxm1
      DO  iy=1,nym1
      hfa=hfa+han(ix,iy)*fa(ix,iy)
      ENDDO
      ENDDO
      DO  ix=1,nxm1
      DO  iy=1,nym1
c     ****************************************************************
c     **                       SW OCEAN                             **
c     ****************************************************************
      uoo(ix,iy)=uon(ix,iy)+dtd2*(
     &-uon(ix,iy)*r2dx*(uon(ix+1,iy)-uon(ix-1,iy))
     &-von(ix,iy)*r2dy*(uon(ix,iy+1)-uon(ix,iy-1))
     &+nuo*(rdx2)*(uon(ix+1,iy)-2.*uon(ix,iy)+uon(ix-1,iy))
     &+nuo*(rdy2)*(uon(ix,iy+1)-2.*uon(ix,iy)+uon(ix,iy-1))
     &-go*(hon(ix+1,iy)-hon(ix-1,iy))*r2dx
     &+f*von(ix,iy)
     &-Cd*rhoa
     &*sqrt((uon(ix,iy)-uan(ix,iy))**2+(von(ix,iy)-van(ix,iy))**2)
     &*(uon(ix,iy)-uan(ix,iy))/(rhoo*(ho0+hon(ix,iy)))
c     &-tau*rhoa*(uon(ix,iy)-uan(ix,iy))/(rhoo*(ho0+hon(ix,iy)))
```

```
     &                    )

      voo(ix,iy)=von(ix,iy)+dtd2*(
     &-uon(ix,iy)/(2*dx)*(von(ix+1,iy)-von(ix-1,iy))
     &-von(ix,iy)*r2dy*(von(ix,iy+1)-von(ix,iy-1))
     &+nuo*(rdx2)*(von(ix+1,iy)-2.*von(ix,iy)+von(ix-1,iy))
     &+nuo*(rdy2)*(von(ix,iy+1)-2.*von(ix,iy)+von(ix,iy-1))
     &-go*(hon(ix,iy+1)-hon(ix,iy-1))*r2dy
     &-f*uon(ix,iy)
     &-Cd*rhoa
     &*sqrt((uon(ix,iy)-uan(ix,iy))**2+(von(ix,iy)-van(ix,iy))**2)
     &*(von(ix,iy)-van(ix,iy))/(rhoo*(ho0+hon(ix,iy)))
c    &-tau*rhoa*(von(ix,iy)-van(ix,iy))/(rhoo*(ho0+hon(ix,iy)))
     &                    )

      hoo(ix,iy)=hon(ix,iy)+dtd2*(
     &+(kappao*(hon(ix+1,iy)-2.*hon(ix,iy)+hon(ix-1,iy)))*rdx2
     &+(kappao*(hon(ix,iy+1)-2.*hon(ix,iy)+hon(ix,iy-1)))*rdy2
     &-(uon(ix+1,iy)*(hon(ix+1,iy)+ho0)
     & -uon(ix-1,iy)*(hon(ix-1,iy)+ho0))*r2dx
     &-(von(ix,iy+1)*(hon(ix,iy+1)+ho0)
     & -von(ix,iy-1)*(hon(ix,iy-1)+ho0))*r2dy
     &-lambdao*hon(ix,iy)
     &                    )
c    ***************************************************************
c    **                        SW ATMOSPHERE                      **
c    ***************************************************************
      uao(ix,iy)=uan(ix,iy)+dtd2*(
     &-uan(ix,iy)*r2dx*(uan(ix+1,iy)-uan(ix-1,iy))
     &-van(ix,iy)*r2dy*(uan(ix,iy+1)-uan(ix,iy-1))
     &+nua*(rdx2)*(uan(ix+1,iy)-2.*uan(ix,iy)+uan(ix-1,iy))
     &+nua*(rdy2)*(uan(ix,iy+1)-2.*uan(ix,iy)+uan(ix,iy-1))
     &-ga*(han(ix+1,iy)-han(ix-1,iy))*r2dx
     &+f*van(ix,iy)
     &+Cd*sqrt((uon(ix,iy)-uan(ix,iy))**2+(von(ix,iy)-van(ix,iy))**2)
     &*(uon(ix,iy)-uan(ix,iy))/(ha0+han(ix,iy))
c    &+tau*(uon(ix,iy)-uan(ix,iy))/(ha0+han(ix,iy))
c    &+lambdaf*(fa(ix,iy)-uan(ix,iy))
     &                    )

      vao(ix,iy)=van(ix,iy)+dtd2*(
     &-uan(ix,iy)*r2dx*(van(ix+1,iy)-van(ix-1,iy))
     &-van(ix,iy)*r2dy*(van(ix,iy+1)-van(ix,iy-1))
     &+nua*(rdx2)*(van(ix+1,iy)-2.*van(ix,iy)+van(ix-1,iy))
     &+nua*(rdy2)*(van(ix,iy+1)-2.*van(ix,iy)+van(ix,iy-1))
     &-ga*(han(ix,iy+1)-han(ix,iy-1))*r2dy
     &-f*uan(ix,iy)
     &+Cd*sqrt((uon(ix,iy)-uan(ix,iy))**2+(von(ix,iy)-van(ix,iy))**2)
     &*(von(ix,iy)-van(ix,iy))/(ha0+han(ix,iy))
c    &+tau*(von(ix,iy)-van(ix,iy))/(ha0+han(ix,iy))
     &                    )

      hao(ix,iy)=han(ix,iy)+dtd2*(
     &+(kappaa*(han(ix+1,iy)-2.*han(ix,iy)+han(ix-1,iy)))*rdx2
     &+(kappaa*(han(ix,iy+1)-2.*han(ix,iy)+han(ix,iy-1)))*rdy2
     &-(uan(ix+1,iy)*(han(ix+1,iy)+ha0)
     & -uan(ix-1,iy)*(han(ix-1,iy)+ha0))*r2dx
     &-(van(ix,iy+1)*(han(ix,iy+1)+ha0)
     & -van(ix,iy-1)*(han(ix,iy-1)+ha0))*r2dy
     &+lambdaf *(ffa-hfa)/ffa*fa(ix,iy)
     &                    )
      ENDDO
```

```
      ENDDO
c     ****************************************************************
c     **                        Boundary conditon                     **
c     ****************************************************************
      DO  ix=1,nxm1
      hoo(ix,ny)=hoo(ix,1)
      hoo(ix,0)=hoo(ix,nym1)
      uoo(ix,ny)=uoo(ix,1)
      uoo(ix,0)=uoo(ix,nym1)
      voo(ix,ny)=voo(ix,1)
      voo(ix,0)=voo(ix,nym1)
      ENDDO

      DO  iy=1,nym1
      hoo(nx,iy)=hoo(1,iy)
      hoo(0,iy)=hoo(nxm1,iy)
      uoo(nx,iy)=uoo(1,iy)
      uoo(0,iy)=uoo(nxm1,iy)
      voo(nx,iy)=voo(1,iy)
      voo(0,iy)=voo(nxm1,iy)
      ENDDO

      DO  ix=1,nxm1
      hao(ix,ny)=hao(ix,1)
      hao(ix,0)=hao(ix,nym1)
      uao(ix,ny)=uao(ix,1)
      uao(ix,0)=uao(ix,nym1)
      vao(ix,ny)=vao(ix,1)
      vao(ix,0)=vao(ix,nym1)
      ENDDO

      DO  iy=1,nym1
      hao(nx,iy)=hao(1,iy)
      hao(0,iy)=hao(nxm1,iy)
      uao(nx,iy)=uao(1,iy)
      uao(0,iy)=uao(nxm1,iy)
      vao(nx,iy)=vao(1,iy)
      vao(0,iy)=vao(nxm1,iy)
      ENDDO


      DO  ix=1,nxm1
      DO  iy=1,nym1
c     ****************************************************************
c     **                        RK OCEAN                              **
c     ****************************************************************
      uon(ix,iy)=uon(ix,iy)+dt*(
     &-uoo(ix,iy)*r2dx*(uoo(ix+1,iy)-uoo(ix-1,iy))
     &-voo(ix,iy)*r2dy*(uoo(ix,iy+1)-uoo(ix,iy-1))
     &+nuo*(rdx2)*(uoo(ix+1,iy)-2.*uoo(ix,iy)+uoo(ix-1,iy))
     &+nuo*(rdy2)*(uoo(ix,iy+1)-2.*uoo(ix,iy)+uoo(ix,iy-1))
     &-go*(hoo(ix+1,iy)-hoo(ix-1,iy))*r2dx
     &+f*voo(ix,iy)
     &-Cd*rhoa
     &*sqrt((uoo(ix,iy)-uao(ix,iy))**2+(voo(ix,iy)-vao(ix,iy))**2)
     &*(uoo(ix,iy)-uao(ix,iy))/(rhoo*(ho0+hoo(ix,iy)))
c     &-tau*rhoa*(uoo(ix,iy)-uao(ix,iy))/(rhoo*(ho0+hoo(ix,iy)))
     &                  )

      von(ix,iy)=von(ix,iy)+dt*(
     &-uoo(ix,iy)*r2dx*(voo(ix+1,iy)-voo(ix-1,iy))
     &-voo(ix,iy)*r2dy*(voo(ix,iy+1)-voo(ix,iy-1))
```

```
     &+nuo*(rdx2)*(voo(ix+1,iy)-2.*voo(ix,iy)+voo(ix-1,iy))
     &+nuo*(rdy2)*(voo(ix,iy+1)-2.*voo(ix,iy)+voo(ix,iy-1))
     &-go*(hoo(ix,iy+1)-hoo(ix,iy-1))*r2dy
     &-f*uoo(ix,iy)
     &-Cd*rhoa
     &*sqrt((uoo(ix,iy)-uao(ix,iy))**2+(voo(ix,iy)-vao(ix,iy))**2)
     &*(voo(ix,iy)-vao(ix,iy))/(rhoo*(ho0+hoo(ix,iy)))
c    &-tau*rhoa*(voo(ix,iy)-vao(ix,iy))/(rhoo*(ho0+hoo(ix,iy)))
     &               )

      hon(ix,iy)=hon(ix,iy)+dt*(
     &+(kappao*(hoo(ix+1,iy)-2.*hoo(ix,iy)+hoo(ix-1,iy)))*rdx2
     &+(kappao*(hoo(ix,iy+1)-2.*hoo(ix,iy)+hoo(ix,iy-1)))*rdy2
     &-(uoo(ix+1,iy)*(hoo(ix+1,iy)+ho0)
     & -uoo(ix-1,iy)*(hoo(ix-1,iy)+ho0))*r2dx
     &-(voo(ix,iy+1)*(hoo(ix,iy+1)+ho0)
     & -voo(ix,iy-1)*(hoo(ix,iy-1)+ho0))*r2dy
     &-lambdao*hoo(ix,iy)
     &                    )
c    ****************************************************************
c    **                          RK ATMOSPHERE                     **
c    ****************************************************************
      uan(ix,iy)=uan(ix,iy)+dt*(
     &-uao(ix,iy)*r2dx*(uao(ix+1,iy)-uao(ix-1,iy))
     &-vao(ix,iy)*r2dy*(uao(ix,iy+1)-uao(ix,iy-1))
     &+nua*(rdx2)*(uao(ix+1,iy)-2.*uao(ix,iy)+uao(ix-1,iy))
     &+nua*(rdy2)*(uao(ix,iy+1)-2.*uao(ix,iy)+uao(ix,iy-1))
     &-ga*(hao(ix+1,iy)-hao(ix-1,iy))*r2dx
     &+f*vao(ix,iy)
     &+Cd*sqrt((uoo(ix,iy)-uao(ix,iy))**2+(voo(ix,iy)-vao(ix,iy))**2)
     &*(uoo(ix,iy)-uao(ix,iy))/(ha0+hao(ix,iy))
c    &+tau*(uoo(ix,iy)-uao(ix,iy))/(ha0+hao(ix,iy))
c    &+lambdaf*(fa(ix,iy)-uao(ix,iy))
     &                    )

      van(ix,iy)=van(ix,iy)+dt*(
     &-uao(ix,iy)*r2dx*(vao(ix+1,iy)-vao(ix-1,iy))
     &-vao(ix,iy)*r2dy*(vao(ix,iy+1)-vao(ix,iy-1))
     &+nua*(rdx2)*(vao(ix+1,iy)-2.*vao(ix,iy)+vao(ix-1,iy))
     &+nua*(rdy2)*(vao(ix,iy+1)-2.*vao(ix,iy)+vao(ix,iy-1))
     &-ga*(hao(ix,iy+1)-hao(ix,iy-1))*r2dy
     &-f*uao(ix,iy)
     &+Cd*sqrt((uoo(ix,iy)-uao(ix,iy))**2+(voo(ix,iy)-vao(ix,iy))**2)
     &*(voo(ix,iy)-vao(ix,iy))/(ha0+hao(ix,iy))
c    &+tau*(voo(ix,iy)-vao(ix,iy))/(ha0+hao(ix,iy))
     &                    )

      han(ix,iy)=han(ix,iy)+dt*(
     &+(kappaa*(hao(ix+1,iy)-2.*hao(ix,iy)+hao(ix-1,iy)))*rdx2
     &+(kappaa*(hao(ix,iy+1)-2.*hao(ix,iy)+hao(ix,iy-1)))*rdy2
     &-(uao(ix+1,iy)*(hao(ix+1,iy)+ha0)
     & -uao(ix-1,iy)*(hao(ix-1,iy)+ha0))*r2dx
     &-(vao(ix,iy+1)*(hao(ix,iy+1)+ha0)
     & -vao(ix,iy-1)*(hao(ix,iy-1)+ha0))*r2dy
     &+lambdaf*(ffa-hfa)/ffa*fa(ix,iy)
     &                    )
      ENDDO
      ENDDO
c
c    ****************************************************************
c    **                        Boundary conditon                   **
c    ****************************************************************
```

```
      DO  ix=1,nxm1
      hon(ix,ny)=hon(ix,1)
      hon(ix,0)=hon(ix,nym1)
      uon(ix,ny)=uon(ix,1)
      uon(ix,0)=uon(ix,nym1)
      von(ix,ny)=von(ix,1)
      von(ix,0)=von(ix,nym1)
      ENDDO

      DO  iy=1,nym1
      hon(nx,iy)=hon(1,iy)
      hon(0,iy)=hon(nxm1,iy)
      uon(nx,iy)=uon(1,iy)
      uon(0,iy)=uon(nxm1,iy)
      von(nx,iy)=von(1,iy)
      von(0,iy)=von(nxm1,iy)
      ENDDO


      DO  ix=1,nxm1
      han(ix,ny)=han(ix,1)
      han(ix,0)=han(ix,nym1)
      uan(ix,ny)=uan(ix,1)
      uan(ix,0)=uan(ix,nym1)
      van(ix,ny)=van(ix,1)
      van(ix,0)=van(ix,nym1)
      ENDDO

      DO  iy=1,nym1
      han(nx,iy)=han(1,iy)
      han(0,iy)=han(nxm1,iy)
      uan(nx,iy)=uan(1,iy)
      uan(0,iy)=uan(nxm1,iy)
      van(nx,iy)=van(1,iy)
      van(0,iy)=van(nxm1,iy)
      ENDDO
      ENDDO
c     ****************************************************************
c     **                          ENERGIES                          **
c     ****************************************************************
      Eco=0.
      Eca=0.
      Epo=0.
      Epa=0.
      Do ix=1,nxm1
      Do iy=1,nym1
      Eco=Eco+(ho0+hon(ix,iy))*(uon(ix,iy)**2+von(ix,iy)**2)
      Eca=Eca+(ha0+han(ix,iy))*(uan(ix,iy)**2+van(ix,iy)**2)
      Epo=Epo+hon(ix,iy)**2
      Epa=Epa+han(ix,iy)**2
      ENDDO
      ENDDO
      Eco=Eco*(1./2)*rhoo*dx*dy
      Eca=Eca*(1./2)*rhoa*dx*dy
      Epo=Epo*(1./2)*rhoo*go*dx*dy
      Epa=Epa*(1./2)*rhoa*ga*dx*dy
      Ema=Epa+Eca
      Emo=Epo+Eco
      Etot=Emo+Ema
      CALL OUTSUB(step,expnum)
c     CALL OUTSUB2(step,expnum)
c     CALL OUTSUB3(step,expnum)
```

```
      ENDDO
      CALL OUTRESTSUB(step,expnum)
      END
c     ****************************************************************
c     **                                                            **
c     **                        END of program                      **
c     **                                                            **
c     ****************************************************************
c     ****************************************************************
c     ****************************************************************
c     ****************************************************************
c     **                                                            **
c     **                        SUBROUTINE:                         **
c     **          Enregistrement h,u,v en tous point à chaque step  **
c     **                                                            **
c     ****************************************************************
      SUBROUTINE OUTSUB(step,expnum)
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,ny,nym1
      PARAMETER   (nx=513,nxm1=nx-1,ny=513,nym1=ny-1)
      INTEGER ::  ix,iy,nt
      INTEGER  :: step,expnum,rhoo,ho0
      REAL(8) ::  Eco,Eca,Epo,Epa,Ema,Emo,Etot,dx,dy
      REAL(8) ::  hon(0:nx,0:ny),uon(0:nx,0:ny),von(0:nx,0:ny),
     &          han(0:nx,0:ny),uan(0:nx,0:ny),van(0:nx,0:ny),
     &          zo(0:nx,0:ny)
      REAL(4) ::  aa(0:nx,0:ny),xx
      COMMON / newcom  /  hon,uon,von,han,uan,van
      COMMON / encom  /  Eco,Eca,Epo,Epa,Ema,Emo,Etot
      CHARACTER(4) :: filename
      CHARACTER(6) :: expnumber
      CHARACTER(4) :: stepnumber
      CHARACTER(5) :: fileext
      CHARACTER(10) :: filename1
      CHARACTER(11) :: filename2
      CHARACTER(15) :: filename3
      CHARACTER(20) :: filenamef
c     ****************************************************************
c     **                                                            **
c     **  nameing of output file OCEAN(ho,uo,vo)                    **
c     **                                                            **
c     ****************************************************************
      dx=3981.
      dy=3981.
      filename="ho__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // stepnumber
      filenamef = filename3 // fileext
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      aa=0.D+0
      aa=REAL(hon,4)
      DO  ix=0,nx
      DO  iy=0,ny
      WRITE(13,*)  aa(ix,iy)
      enddo
      enddo
      CLOSE(13)
```

```
      filename="uo__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // stepnumber
      filenamef = filename3 // fileext
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      aa=0.D+0
      aa=REAL(uon,4)
      DO  ix=0,nx
      DO  iy=0,ny
      WRITE(13,*)  aa(ix,iy)
      enddo
      enddo
      CLOSE(13)
      filename="vo__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // stepnumber
      filenamef = filename3 // fileext
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      aa=0.D+0
      aa=REAL(von,4)
      DO  ix=0,nx
      DO  iy=0,ny
      WRITE(13,*)  aa(ix,iy)
      enddo
      enddo
      CLOSE(13)
c     filename="zo__"
c     fileext=".data"
c     WRITE(expnumber,FMT='(I6)') 100000+ expnum
c     WRITE(stepnumber,FMT='(I4)') 1000+ step
c     filename1 = filename // expnumber
c     filename2 = filename1 // "_"
c     filename3 = filename2 // stepnumber
c     filenamef = filename3 // fileext
c     OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
c    &FILE=filenamef)
c     zo=0.
c     DO  ix=1,nxm1
c     DO  iy=1,nym1
c     zo(ix,iy)=(von(ix+1,iy)-von(ix-1,iy))/dx
c    &-(uon(ix,iy+1)-uon(ix,iy-1))/dy
c     enddo
c     enddo
c     aa=0.D+0
c     aa=REAL(zo,4)
c     DO  ix=0,nx
c     DO  iy=0,ny
c     WRITE(13,*)  aa(ix,iy)
c     enddo
c     enddo
c     CLOSE(13)
c     ***************************************************************
```

```
c    **                                                              **
c    **  nameing of output file ATMOSPHERE(ha,ua,va)                 **
c    **                                                              **
c    ***************************************************************
     filename="ha__"
     fileext=".data"
     WRITE(expnumber,FMT='(I6)') 100000+ expnum
     WRITE(stepnumber,FMT='(I4)') 1000+ step
     filename1 = filename // expnumber
     filename2 = filename1 // "_"
     filename3 = filename2 // stepnumber
     filenamef = filename3 // fileext
     OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
    &FILE=filenamef)
     aa=0.D+0
     aa=REAL(han,4)
     DO  ix=0,nx
     DO  iy=0,ny
     WRITE(13,*)  aa(ix,iy)
     enddo
     enddo
     CLOSE(13)
     filename="ua__"
     fileext=".data"
     WRITE(expnumber,FMT='(I6)') 100000+ expnum
     WRITE(stepnumber,FMT='(I4)') 1000+ step
     filename1 = filename // expnumber
     filename2 = filename1 // "_"
     filename3 = filename2 // stepnumber
     filenamef = filename3 // fileext
     OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
    &FILE=filenamef)
     aa=0.D+0
     aa=REAL(uan,4)
     DO  ix=0,nx
     DO  iy=0,ny
     WRITE(13,*)  aa(ix,iy)
     enddo
     enddo
     CLOSE(13)
     filename="va__"
     fileext=".data"
     WRITE(expnumber,FMT='(I6)') 100000+ expnum
     WRITE(stepnumber,FMT='(I4)') 1000+ step
     filename1 = filename // expnumber
     filename2 = filename1 // "_"
     filename3 = filename2 // stepnumber
     filenamef = filename3 // fileext
     OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
    &FILE=filenamef)
     aa=0.D+0
     aa=REAL(van,4)
     DO  ix=0,nx
     DO  iy=0,ny
     WRITE(13,*)  aa(ix,iy)
     enddo
     enddo
     CLOSE(13)
     RETURN
     END SUBROUTINE OUTSUB
c    ***************************************************************
c    ***************************************************************
```

```
c     **                                                              **
c     **                        SUBROUTINE 2:                         **
c     **                   Enregistrement des Energies:               **
c     **                cinetiques, potentielles, mecaniques, totale  **
c     **                                                              **
c     ****************************************************************
      SUBROUTINE OUTSUB2(step,expnum)
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,ny,nym1
      PARAMETER   (nx=513,nxm1=nx-1,ny=513,nym1=ny-1)
      INTEGER ::  ix,iy,nt
      INTEGER  :: step,expnum,dx,dy,rhoo,ho0
      REAL(8) ::  Eco,Eca,Epo,Epa,Ema,Emo,Etot
      REAL(8) ::  hon(0:nx,0:ny),uon(0:nx,0:ny),von(0:nx,0:ny),
     &           han(0:nx,0:ny),uan(0:nx,0:ny),van(0:nx,0:ny)
      REAL(4) ::  aa(0:nx,0:ny),xx
      COMMON / newcom  /  hon,uon,von,han,uan,van
      COMMON / encom  /  Eco,Eca,Epo,Epa,Ema,Emo,Etot
      CHARACTER(4) :: filename
      CHARACTER(6) :: expnumber
      CHARACTER(4) :: stepnumber
      CHARACTER(5) :: fileext
      CHARACTER(10) :: filename1
      CHARACTER(11) :: filename2
      CHARACTER(15) :: filename3
      CHARACTER(20) :: filenamef
c     ****************************************************************
c     **                                                              **
c     **   subroutine ENERGIE cin                                     **
c     **                                                              **
c     ****************************************************************
      filename="Eco__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Eco,4)
      WRITE(13,*)  xx
      CLOSE(13)
      filename="Eca__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
```

```
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Eca,4)
      WRITE(13,*)  xx
      CLOSE(13)
c     ****************************************************************
c     **                                                            **
c     **   subroutine ENERGIE pot                                   **
c     **                                                            **
c     ****************************************************************
      filename="Epo__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Epo,4)
      WRITE(13,*)  xx
      CLOSE(13)
      filename="Epa__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Epa,4)
      WRITE(13,*)  xx
      CLOSE(13)
c     ****************************************************************
c     **                                                            **
c     **   subroutine ENERGIE mec                                   **
c     **                                                            **
c     ****************************************************************
      filename="Emo__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
```

```
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Emo,4)
      WRITE(13,*)  xx
      CLOSE(13)
      filename="Ema__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Ema,4)
      WRITE(13,*)  xx
      CLOSE(13)
c     ****************************************************************
c     **                                                            **
c     **   subroutine ENERGIE tot                                   **
c     **                                                            **
c     ****************************************************************
      filename="Etot__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      xx=0.D+0
      xx=REAL(Etot,4)
      WRITE(13,*)  xx
      CLOSE(13)
      RETURN
      END SUBROUTINE OUTSUB2
c     ****************************************************************
c     ****************************************************************
c     **                                                            **
c     **   SUBROUTINE 3:                                            **
c     ** Enregistrement de h,u,v en fonction du tps pour un point   **
```

```
c     **                                                              **
c     **************************************************************
      SUBROUTINE OUTSUB3(step,expnum)
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,ny,nym1
      PARAMETER   (nx=513,nxm1=nx-1,ny=513,nym1=ny-1)
      INTEGER ::  ix,iy,nt
      INTEGER  :: step,expnum,dx,dy,rhoo,ho0
      REAL(8) ::  Eco,Eca,Epo,Epa,Ema,Emo,Etot
      REAL(8) ::  hon(0:nx,0:ny),uon(0:nx,0:ny),von(0:nx,0:ny),
     &           han(0:nx,0:ny),uan(0:nx,0:ny),van(0:nx,0:ny)
      REAL(4) ::  aa(0:nx,0:ny),xx
      COMMON / newcom  /  hon,uon,von,han,uan,van
      COMMON / encom  /  Eco,Eca,Epo,Epa,Ema,Emo,Etot
      CHARACTER(4) :: filename
      CHARACTER(6) :: expnumber
      CHARACTER(4) :: stepnumber
      CHARACTER(5) :: fileext
      CHARACTER(10) :: filename1
      CHARACTER(11) :: filename2
      CHARACTER(15) :: filename3
      CHARACTER(20) :: filenamef
c     **************************************************************
c     **                                                          **
c     **   subroutine3  Ocean(t)                                  **
c     **                                                          **
c     **************************************************************
      filename="hot__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(hon,4)
      DO ix=1,1
      DO iy=100,100
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
      CLOSE(13)
      filename="uot__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
```

```
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(uon,4)
      DO ix=1,1
      DO iy=150,150
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
      CLOSE(13)
      filename="vot__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(von,4)
      DO ix=1,1
      DO iy=165,165
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
c     ***************************************************************
c     **                                                           **
c     **   subroutine3  Atmo(t)                                    **
c     **                                                           **
c     ***************************************************************
      CLOSE(13)
      filename="hat__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(han,4)
      DO ix=1,1
      DO iy=100,100
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
      CLOSE(13)
```

```
      filename="uat__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(uan,4)
      DO ix=1,1
      DO iy=150,150
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
      CLOSE(13)
      filename="vat__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // "00000"
      filenamef = filename3 // fileext
      IF(step .EQ. 0) THEN
      OPEN(UNIT=13,STATUS='NEW',FORM='FORMATTED',
     &FILE=filenamef)
      ELSE
      OPEN(UNIT=13,STATUS='OLD',FORM='FORMATTED',ACCESS='APPEND',
     &FILE=filenamef)
      ENDIF
      aa=0.D+0
      aa=REAL(van,4)
      DO ix=1,1
      DO iy=50,50
      WRITE(13,*)  aa(ix,iy)
      ENDDO
      ENDDO
      CLOSE(13)

      RETURN
      END SUBROUTINE OUTSUB3
c     ****************************************************************
c     **                                                            **
c     ** END                                                        **
c     **                                                            **
c     ****************************************************************
c     ****************************************************************
c     ****************************************************************
c     **                                                            **
c     **   SUBROUTINE:                                              **
c     **      restart ecriture                                      **
c     **                                                            **
c     ****************************************************************
      SUBROUTINE OUTRESTSUB(step,expnum)
```

```
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,ny,nym1
      PARAMETER   (nx=513,nxm1=nx-1,ny=513,nym1=ny-1)
      INTEGER ::  ix,iy,nt
      INTEGER  :: step,expnum
      REAL(8) ::  hon(0:nx,0:ny),uon(0:nx,0:ny),von(0:nx,0:ny),
     &           han(0:nx,0:ny),uan(0:nx,0:ny),van(0:nx,0:ny)
      COMMON / newcom  /  hon,uon,von,han,uan,van
      CHARACTER(4) :: filename
      CHARACTER(6) :: expnumber
      CHARACTER(4) :: stepnumber
      CHARACTER(5) :: fileext
      CHARACTER(10) :: filename1
      CHARACTER(11) :: filename2
      CHARACTER(15) :: filename3
      CHARACTER(20) :: filenamef
c     ****************************************************************
c     **                                                            **
c     **  restart                                                   **
c     **                                                            **
c     ****************************************************************
      filename="re__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step-1
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // stepnumber
      filenamef = filename3 // fileext
      OPEN(UNIT=13,STATUS='NEW',FORM='UNFORMATTED',
     &FILE=filenamef)
      WRITE(13)  step-1
      WRITE(13)  hon
      WRITE(13)  uon
      WRITE(13)  von
      WRITE(13)  han
      WRITE(13)  uan
      WRITE(13)  van
      CLOSE(13)
      RETURN
      END SUBROUTINE OUTRESTSUB
c     ****************************************************************
c     ****************************************************************
c     **                                                            **
c     **   SUBROUTINE:                                              **
c     **      restart lecture                                       **
c     **                                                            **
c     ****************************************************************
      SUBROUTINE INRESTSUB(step,expnum)
      IMPLICIT NONE
      INTEGER ::  nx,nxm1,ny,nym1
      PARAMETER   (nx=513,nxm1=nx-1,ny=513,nym1=ny-1)
      INTEGER ::  ix,iy,nt
      INTEGER  :: step,expnum
      REAL(8) ::  hon(0:nx,0:ny),uon(0:nx,0:ny),von(0:nx,0:ny),
     &           han(0:nx,0:ny),uan(0:nx,0:ny),van(0:nx,0:ny)
      COMMON / newcom  /  hon,uon,von,han,uan,van
      CHARACTER(4) :: filename
      CHARACTER(6) :: expnumber
      CHARACTER(4) :: stepnumber
      CHARACTER(5) :: fileext
      CHARACTER(10) :: filename1
```

```
      CHARACTER(11) :: filename2
      CHARACTER(15) :: filename3
      CHARACTER(20) :: filenamef
c     ************************************************************
c     **                                                        **
c     **  restart                                               **
c     **                                                        **
c     ************************************************************
      filename="re__"
      fileext=".data"
      WRITE(expnumber,FMT='(I6)') 100000+ expnum
      WRITE(stepnumber,FMT='(I4)') 1000+ step
      filename1 = filename // expnumber
      filename2 = filename1 // "_"
      filename3 = filename2 // stepnumber
      filenamef = filename3 // fileext
      OPEN(UNIT=13,STATUS='OLD',FORM='UNFORMATTED',
     &FILE=filenamef)
      READ(13)  step
      READ(13)  hon
      READ(13)  uon
      READ(13)  von
      READ(13)  han
      READ(13)  uan
      READ(13)  van
      CLOSE(13)
      RETURN
      END SUBROUTINE INRESTSUB
```

B) APPENDIX 2:
Scilab programs (simulation analysis)

1. Spatial evolution (Section):

```
clearglobal;
chemin=""
chdir ("/home/users/moulin9a/PROG/Fquadratique/PROGRAMME")
tau=8.*10^(-5)
Cd=(0.8+0.065*0.01)*(1*10^(-3))
a=3.141592653589793*20./180.
y=[1:200]
for i =24
 t=1000+i
 cpt = sprintf('%d',t)
 nomfig = ['Atmo_'+cpt]
 nomfichier1= ['ha__100003_'+cpt+'.data']
 nomfichier2= ['ua__100003_'+cpt+'.data']
 nomfichier3= ['va__100003_'+cpt+'.data']
 nomfichier4= ['uo__100003_'+cpt+'.data']
 nomfichier5= ['vo__100003_'+cpt+'.data']
 fid1=file('open',nomfichier1,"old");
 fid2=file('open',nomfichier2,"old");
 fid3=file('open',nomfichier3,"old");
 fid4=file('open',nomfichier4,"old");
 fid5=file('open',nomfichier5,"old");
 ha=read(fid1,1,201)
 ua=read(fid2,1,201)
 va=read(fid3,1,201)
 uo=read(fid4,1,201)
 vo=read(fid5,1,201)
 //Flin
 //F=tau*(uo-ua)
 //Fquadr
 F=Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo-ua)
 //Fekmanlin
 //F=tau.*((uo-ua)-a.*(vo-va))
 y=y'
 ha=ha'
 ua=ua'
 va=va'
 F=F'
 ha=ha([2:1:(201)])
 ua=ua([2:1:(201)])
 va=va([2:1:(201)])
 F=F([2:1:(201)])
file('close',fid1)
file('close',fid2)
file('close',fid3)
file('close',fid4)
file('close',fid5)
chdir ("/home/users/moulin9a/PROG/Fquadratique")
clf()
plot2d(y*5,[ha 10*ua va*1000 F*10^4],style=[1,4,3,5],rect=[0,-0.12,1000,0.12])
xset('font',6,5)
xlabel ('y(km)','fontsize',6)
ylabel ('velocity(m/s) and height(m)','fontsize',6)
```

2. Temporal evolution (Profiles):

```
clearglobal;
chemin=""
chdir ("/home/users/moulin9a/PROG/Frayleigh/PROGRAMME")
t=[0:720]
nomfig = ['Atmo30j']
nomfichiera1= ['hat_100003_0000.data']
nomfichiera2= ['uat_100003_0000.data']
nomfichiera3= ['vat_100003_0000.data']
fida1=file('open',nomfichiera1,"old");
fida2=file('open',nomfichiera2,"old");
fida3=file('open',nomfichiera3,"old");
za1=read(fida1,1,721)
za2=read(fida2,1,721)
za3=read(fida3,1,721)
t=t'
za1=za1'
za2=za2'
za3=za3'
file('close',fida1)
file('close',fida2)
file('close',fida3)
clf()
plot2d(t/(24*30),[za1 za2*10 za3*10^4])
xset('font',6,5)
xlabel ('time (day)','fontsize',6)
ylabel ('velocity and height variation(m/s & m)','fontsize',6)
title('','fontsize',5);
legend(['ha';'10ua';'10^4va'],5)
chdir ("/home/users/moulin9a/PROG/Frayleigh")
xs2jpg(0,nomfig)
close()
```

3. Map of height variations with the velocity field:

```
clearglobal;
chemin=""
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job117')
stacksize('max')
x=[0:513]
y=[0:513]
x1=[0:50:509]
y1=[0:50:509]
//x=[0:201]
//y=[0:201]
for i =1760
 t=1000+i
 cpt = sprintf('%d',t)
 nomfig = ['Carteva'+cpt]
 nomfichier1= ['ha__100117_'+cpt+'.data']
 nomfichier2= ['ua__100117_'+cpt+'.data']
 nomfichier3= ['va__100117_'+cpt+'.data']
 fid1=file('open',nomfichier1,"old");
 fid2=file('open',nomfichier2,"old");
 fid3=file('open',nomfichier3,"old");
 ho=read(fid1,514,514)
 uo=read(fid2,514,514)
 vo=read(fid3,514,514)
file('close',fid1)
file('close',fid2)
```

```
file('close',fid3)
vo1=vo(1:50:510,1:50:510)
uo1=uo(1:50:510,1:50:510)
figure("figure_size",[1000 950]);
xset('font',6,5)
xlabel ('x(km)','fontsize',4)
ylabel ('y(km)','fontsize',4)
title('','fontsize',4);
xset('colormap',jetcolormap(16)); // definit la carte des couleurs
//Sgrayplot(x*1.953,y*1.953,ho,rect=[0,0,1000,1000],zminmax=[-200 170]);
//colorbar(-200,170,colorminmax=[1 100]);
Sgrayplot(x*1.953,y*1.953,vo,rect=[0,0,1000,1000]);
colorbar(min(vo),max(vo),colorminmax=[1 100]);
//champ(x1*1.953,y1*1.953,uo1,vo1,rect=[0,0,1000,1000])
chdir ('/home/users/moulin9a/PROG/instab/Job117')
xs2gif(0,nomfig)
close()
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job117')
end
```

4. Map of height variations without the mean current in the x-direction:

```
clearglobal;
chemin=""
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job117')
stacksize('max')
x=[0:513]
y=[0:513]
x1=[0:50:509]
y1=[0:50:509]
//Carte Atmo:
for i =1760
   t=1000+i
   cpt = sprintf('%d',t)
   nomfig = ['Carteha'"+cpt]
   nomfichier1= ['ha__100117_'+cpt+'.data']
   nomfichier2= ['ua__100117_'+cpt+'.data']
   nomfichier3= ['va__100117_'+cpt+'.data']
   fid1=file('open',nomfichier1,"old");
   fid2=file('open',nomfichier2,"old");
   fid3=file('open',nomfichier3,"old");
   ha=read(fid1,514,514)
   ua=read(fid2,514,514)
   va=read(fid3,514,514)
   file('close',fid1)
   file('close',fid2)
   file('close',fid3)
   for j=1:514
     ham(j)=0
     uam(j)=0
     vam(j)=0
   end
   for j=1:514
     for i=1:514
       ham(j)=ham(j)+ha(i,j)
       uam(j)=uam(j)+ua(i,j)
       vam(j)=vam(j)+va(i,j)
     end
   end
   ham=ham./514
   uam=uam./514
```

```
    vam=vam./514
    for j=1:514
      for i=1:514
        ha(i,j)=ha(i,j)-ham(j)
        ua(i,j)=ua(i,j)-uam(j)
        va(i,j)=va(i,j)-vam(j)
      end
    end
    va1=va(1:50:510,1:50:510)
    ua1=ua(1:50:510,1:50:510)
//Figure atmo
figure("figure_size",[1000 950]);
xset('font',6,5)
xlabel ('x(km)','fontsize',5)
ylabel ('y(km)','fontsize',5)
title('','fontsize',4);
xset('colormap',jetcolormap(16));
Sgrayplot(x*1.953,y*1.953,ha,rect=[0,0,1000,1000],zminmax=[-5 5]);//min max fixé
colorbar(-5,5,colorminmax=[1 100]);//min max fixé
fig = gcf();
a1 = fig.children(1);
a1.font_size = 5;
//Sgrayplot(x*1.953,y*1.953,ha,rect=[0,0,1000,1000]);
//colorbar(min(ha),max(ha),colorminmax=[1 100]);
champ(x1*1.953,y1*1.953,ua1,va1,rect=[0,0,1000,1000])
chdir ('/home/users/moulin9a/PROG/instab/Job117/')
xs2gif(0,nomfig)
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job117'
close()
end
```

5. Map of energy fluxes:

```
clearglobal;
chemin=""
chdir ("/home/users/moulin9a/PROG/Fekman");
month=29*24*3600.0
tau=1/month
Cd=(0.8+0.065*0.01)*(1*10^(-3))
a=3.141592653589793*20/180
nx=2
ny=202;
x=[1:nx];
y=[1:ny];
for i =96
  cpt1 = sprintf('%d',i)
  t=1000+i
  cpt = sprintf('%d',t)
//ouverture et lecture fichier (uo,vo,ua,va)
nomfichieruo = ['uo__100003_'+cpt+'.data']
nomfichiervo = ['vo__100003_'+cpt+'.data']
nomfichierua = ['ua__100003_'+cpt+'.data']
nomfichierva = ['va__100003_'+cpt+'.data']
fiduo=file('open',nomfichieruo,"old")
fidvo=file('open',nomfichiervo,"old")
fidua=file('open',nomfichierua,"old")
fidva=file('open',nomfichierva,"old")
uo=read(fiduo,2,202)
ua=read(fidua,2,202)
vo=read(fidvo,2,202)
va=read(fidva,2,202)
```

```
file('close',fiduo)
file('close',fidua)
file('close',fidvo)
file('close',fidva)
//DEco,DEca linéaire
//DEco=-tau*(uo.^2-ua.*uo-vo.*va+vo.^2)
//DEca=-tau*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca quadratique
DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2)
DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca ekman
DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2+a*va.*uo-a*vo.*ua)
DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2+a*vo.*ua-a*va.*uo)
xset("window",0)
clf()
xsetech([0 0 0.5 1])
xset('font',6,4)
xlabel ('x(m)','fontsize',5)
ylabel ('y(m)','fontsize',5)
title('Energy Flux for the ocean','fontsize',5);
xset('colormap',jetcolormap(16)); // definit la carte des couleurs
//colorbar(min(DEco),max(DEco),colorminmax=[1 100]);
//colorbar(-6*10^-14 ,6*10^-14,colorminmax=[1 100]);
//Sgrayplot(x*5000,y*5000,DEco);
Sgrayplot(x*5000,y*5000,DEco,zminmax=[-6*10^-16 6*10^-16]);
xsetech([0.5 0 0.5 1])   // dessin en haut a droite
xset('font',6,4)
xlabel ('x(m)','fontsize',5)
ylabel ('y(m)','fontsize',5)
title('Energy Flux for the atmosphere','fontsize',5);
xset('colormap',jetcolormap(16)); // definit la carte des couleurs
//colorbar(min(DEca),max(DEca),colorminmax=[1 100]);
colorbar(-6*10^-16 ,6*10^-16,colorminmax=[1 300]);
//Sgrayplot(x*5000,y*5000,DEca);
Sgrayplot(x*5000,y*5000,DEca,zminmax=[-6*10^-16 6*10^-16]);
xs2gif(0,'DEc'+cpt1)
//DEct
//DEct=DEco+DEca
//    figure();
//    xset('colormap',jetcolormap(16)); // definit la carte des couleurs
//    colorbar(min(DEct),max(DEct),colorminmax=[1 100]);
//    Sgrayplot(x*5000,y*5000,DEct);
//    legend(['dtEct'],3)
//    xtitle('Energy Flux','x(m)','y(m)')
//    xs2gif(0,'DEct'+cpt)
//    close()
end
```

6. Energy fluxes along y-direction for one x:

```
clearglobal;
chemin=""
chdir ("/home/users/moulin9a/PROG/Fquadratique/PROGRAMME");
tau=8*10^(-4)*0.1
Cd=(0.8+0.065*0.01)*(1*10^(-3))
a=3.141592653589793*20/180
nx=2
ny=202;
x=[1:nx];
y=[1:ny];
dt=[0:7200]
```

```
Eco=file('open','DEco',"unknow")
Eca=file('open','DEca',"unknow")
for i =0:7200
    t = sprintf('%03d',i)
    t=1000+i
  cpt = sprintf('%d',t)
//ouverture et lecture fichier (uo,vo,ua,va)
nomfichieruo = ['uo__100003_'+cpt+'.data']
nomfichiervo = ['vo__100003_'+cpt+'.data']
nomfichierua = ['ua__100003_'+cpt+'.data']
nomfichierva = ['va__100003_'+cpt+'.data']
fiduo=file('open',nomfichieruo,"old")
fidvo=file('open',nomfichiervo,"old")
fidua=file('open',nomfichierua,"old")
fidva=file('open',nomfichierva,"old")
uo=read(fiduo,2,202)
ua=read(fidua,2,202)
vo=read(fidvo,2,202)
va=read(fidva,2,202)
file('close',fiduo)
file('close',fidua)
file('close',fidvo)
file('close',fidva)
//DEco,DEca linéaire
//DEco=-tau*(uo.^2-ua.*uo-vo.*va+vo.^2)
//DEca=-tau*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca quadratique
DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2)
DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca ekman quadratique
//DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2+a*va.*uo-a*vo.*ua)
//DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2+a*vo.*ua-a*va.*uo)
//DEco,DEca ekman linéaire
//DEco=-tau*(uo.^2-ua.*uo-vo.*va+vo.^2+a*va.*uo-a*vo.*ua)
//DEca=-tau*(ua.^2-ua.*uo-vo.*va+va.^2+a*vo.*ua-a*va.*uo)
X=DEco(1,150)
Y=DEca(1,150)
write(Eco,X)
write(Eca,Y)
end
file('close',Eco)
file('close',Eca)
Eco=file('open','DEco',"old")
Eca=file('open','DEca',"old")
DEco=read(Eco,7201,1)
DEca=read(Eca,7201,1)
file('close',Eco)
file('close',Eca)
clf()
xset('font',6,5)
plot2d(dt/(24*30),[DEco DEca DEco+DEca],style=[18,3,1],rect=[0,-1.4*10^(-6),10,4.0*10^(-7)],nax=[5,11,1,10])
legend(['ocean';'atmosphere';'lost'],5)
xlabel ('time (month)','fontsize',6)
ylabel ('Energy flux (W/m^2)','fontsize',6)
//title('Energy flux for a quadratic friction','fontsize',6);
xs2jpg(0,'DEc2(1,150)')
```

7. Histogram of probability density function of energy fluxes:

```
clearglobal;
chemin=""
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job118');
//chdir ('/home/users/moulin9a/PROG/instab/Job110/donne')
stacksize('max')
u=10
tau=8.D-4*u
Cd=(0.8+0.065*10)*(1*10^(-3))
a=3.141592653589793*20/180
nx=513;
ny=513;
x=[0:nx];
y=[0:ny];
for i=1760
  cpt1 = sprintf('%d',i)
  t=1000+i
  cpt = sprintf('%d',t)
//ouverture et lecture fichier (uo,vo,ua,va)
nomfichieruo = ['uo__100118_'+cpt+'.data']
nomfichiervo = ['vo__100118_'+cpt+'.data']
nomfichierua = ['ua__100118_'+cpt+'.data']
nomfichierva = ['va__100118_'+cpt+'.data']
fiduo=file('open',nomfichieruo,"old")
fidvo=file('open',nomfichiervo,"old")
fidua=file('open',nomfichierua,"old")
fidva=file('open',nomfichierva,"old")
uo=read(fiduo,514*514,1)
ua=read(fidua,514*514,1)
vo=read(fidvo,514*514,1)
va=read(fidva,514*514,1)
file('close',fiduo)
file('close',fidua)
file('close',fidvo)
file('close',fidva)
//DEco,DEca linéaire
//DEco=-tau*(uo.^2-ua.*uo-vo.*va+vo.^2)
//DEca=-tau*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca quadratique
DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2)
DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2)
//DEco,DEca ekman
//DEco=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(uo.^2-ua.*uo-vo.*va+vo.^2+a*va.*uo-a*vo.*ua)
//DEca=-Cd*((uo-ua).^2+(vo-va).^2).^(0.5).*(ua.^2-ua.*uo-vo.*va+va.^2+a*vo.*ua-a*va.*uo)

chdir ('/home/users/moulin9a/PROG/instab/Job118/pdf')
write('DEco'+cpt+'.data',DEco)
write('DEca'+cpt+'.data',DEca)
//chdir ('/home/users/moulin9a/PROG/instab/Job110/donne')
chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job118')
close()
end
clearglobal;
chemin=""
chdir ('/home/users/moulin9a/PROG/instab/Job118/pdf')
for
 i =1760
 t=1000+i
 cpt = sprintf('%d',t)
 nomfig1 = ['pdf50'+cpt]
```

```
 nomfichier1= ['DEca'+cpt+'.data']
   nomfichier2= ['DEco'+cpt+'.data']
   fid1=file('open',nomfichier1,"old");
   fid2=file('open',nomfichier2,"old");
   DEca=read(fid1,264196,1)
   DEco=read(fid2,264196,1)
   //DEca=read(fid1,512,512)
   //DEco=read(fid2,512,512)
   file('close',fid1)
   file('close',fid2)
   Ma=max(DEca)
   ma=min(DEca)
   Mo=max(DEco)
   mo=min(DEco)
//    o=linspace(-1.5*10^(-3),6.06*10^(-3),61) //Job116
//    a=linspace(-3.17*10^(-1),0.,61) //Job116
   o=linspace(-0.00298,0.00808,50) //Job110 16 classes
   a=linspace(-0.1314,0,50) //Job110 16 classes
   figure("figure_size",[1920 1200]);
xset("window",0)
clf()
xsetech([0 0 0.5 1])
xset('font',6,4)
xlabel ('Energy flux (W/m^2)','fontsize',5)
ylabel ('Probability density (number)','fontsize',5)
title('Probability density function for the ocean','fontsize',5);
histplot(o,DEco,normalization=%f,rect=[-0.00298,0,0.00808,110000],nax=[4,8,1,11])//intervals egaux
//histplot(40,DEco,rect=[-2.*10^(-3),0,6.06*10^(-3),225000],nax=[3,6,1,11],normalization=%f)
//histplot(40,DEco,rect=[mo,0,Mo,225000],nax=[3,6,1,11],normalization=%f)
xsetech([0.5 0 0.5 1])
xset('font',6,4)
xlabel ('Energy flux (W/m^2)','fontsize',5)
ylabel ('Probability density (number)','fontsize',5)
title('Probability density function for the atmosphere','fontsize',5);
histplot(a,DEca,normalization=%f,rect=[-0.1314,0,0,40000],nax=[5,10,1,11])//intervals egaux
//histplot(40,DEca,rect=[-1.317*10^(-1),0,0,200000],nax=[3,6,1,11],normalization=%f)
//histplot(40,DEca,rect=[ma,0,Ma,200000],nax=[3,6,1,11],normalization=%f)
xs2gif(0,nomfig1)
close()
clear
end
```

8. Hovmöller diagram of height variations:

```
clearglobal;
chemin=""
chdir ('/home/users/moulin9a/PROG/instab/Job122/moyHov')
//chdir ('/fsnet/data/legi/calcul9/data/moulin9a/instab/Job122')
stacksize('max')
x=[0:513]
y=[0:513]
//tps=[500:700]
tps=[0:1500]
for i=0:1500
   t=1000+i
   //   k=i-499
   k=i+1
   cpt = sprintf('%d',t)
   nomfig = ['Hovmoller.data']
  nomfichier1= ['hamoy'+cpt+'.data']
//    nomfichier1= ['ha__100122_'+cpt+'.data']
```

```
    fid1=file('open',nomfichier1,"old");
    ha=read(fid1,514,514)
    file('close',fid1)
    h(:,k)=ha(:,385)
  end
  h=h'
  chdir ('/home/users/moulin9a/PROG/instab/Job122/')
  write('Hovmollermoy.data',h)
//  figure("figure_size",[1000 950]);
//  xset('font',6,5)
//  xset('colormap',jetcolormap(25));
//  Sgrayplot(tps,x,h,rect=[500,0,700,513],zminmax=[-21 12])
//  colorbar(-21,-12,colorminmax=[1 100])
//  xs2gif(0,nomfig)
//  clear
figure("figure_size",[1000 950]);
  xset('font',6,5)
  xset('colormap',jetcolormap(20));
  xlabel ('time(month)','fontsize',5)
  ylabel ('x(km)','fontsize',5)
  title('','fontsize',4);
  //  Sgrayplot(tps,x*1.95,h,rect=[500,0,700,513],,zminmax=[-115 -75])
  Sgrayplot(tps/30,x*1.95,h,rect=[17,0,23,1000],zminmax=[-21 21])
  colorbar(-21,21,colorminmax=[1 100])
  fig = gcf();
a1 = fig.children(1);
a1.font_size = 4;
  xs2gif(0,nomfig)
close()
```