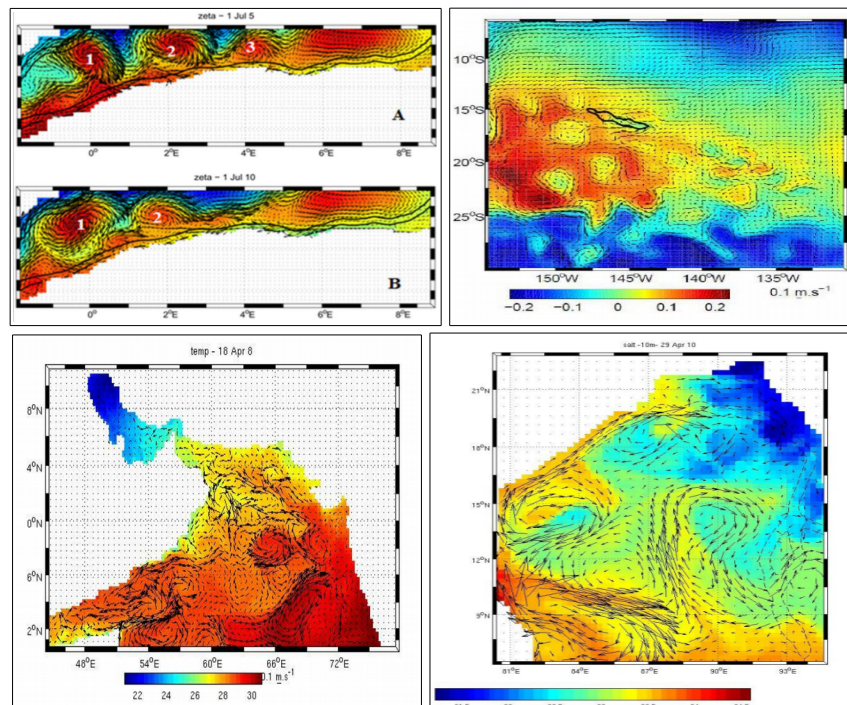


Andrea M. Doglioli

Lecture Notes and Tutorials

3D Ocean Modelling



last revision: 14 December 2022

Acknowledgements

I would like to thank all my students and colleagues for their comments, questions, corrections and suggestions. Specifically, this document benefited from contributions by M. Arancio, C. Assassi, J.M. Beckers, A.C. Blaizot, R. Campbell, N. Daniault, G. Dirberg, C. Dufau-Julliand, J. Gatti, Z. Hu, Y. José, M. Kersalé, S. Morrisset, A. Petrenko, L. Perrot, R. Rainaud, M. Simon, M. Supplien, and S. Zemenzer.

The tutorials in this course would not have been possible without the freely available *ROMS* model and the *ROMS_tools*; my thanks go therefore to the entire *ROMS* community and especially to G. Cambon, P. Penven, and P. Marchesiello.

Doglioli, A.M. (2015), *Lecture Notes and Tutorials in 3D Ocean Modelling*, University of Aix-Marseille, Marseille, France.

www.mio.univ-amu.fr/~doglioli/Doglioli_NotesCoursTD_Modelisation3DOceanique.pdf 

This material is distributed under the Creative Commons license [<http://creativecommons.org/>]

You are free to:

- reproduce, distribute and communicate this creation to the public
- modify this creation

under the following conditions:



Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



This work has been produced with the free software LibreOffice <http://www.libreoffice.org>

Table of Contents

Course Notes	6
Introduction.....	7
Fundamentals.....	17
The primitive equations.....	17
The Boussinesq approximation.....	18
The shallow water equations.....	19
Heat and salt conservation equations.....	20
“Density”.....	20
Turbulence “closure”.....	21
Boundary conditions.....	34
Model grids.....	41
Explicit and implicit approaches.....	42
Staggered grids.....	44
The CFL condition.....	47
Horizontal coordinates and transformation of horizontal grids.....	52
Vertical grids and coordinates.....	57
Advection schemes.....	61
Initial and open boundary conditions.....	65
Classification of partial differential equations.....	65
Models with open boundaries.....	66
Practical approaches to calculating initial conditions.....	68
Open boundary conditions for models solving the primitive equations.....	73
Tutorials	75
T 1 (2h): ROMS and ROMS_tools user registration and download.....	77
T 2 (2h): Unpacking and compiling.....	78
T 3 (2h): <i>Benguela example</i> : setting up the grid.....	84
T 4 (2h): <i>Benguela example</i> : forcings and boundary conditions.....	85
T 5 (2h): <i>Benguela example</i> : simulations and graphical presentation of results.....	86
T 6 (2h): <i>Benguela example</i> : multi-annual simulations and diagnostic tools.....	87
T 7 (4h): Implementing a new model for an oceanographic region of choice.....	88
T 8 (6h): Posting the results of the new model online.....	89
T 9 (4h): Writing the report and the presentation.....	90
Exam topics from previous years	91
List of projects from previous years	95

Bibliography and Useful Links

Jean-Marie Beckers and colleagues, **Notes diapositive des cours à l'Université de Liège**
<http://modb.oce.ulg.ac.be/courses.html>

J. Coiffier (2000), **Un demi siècle de prévision numérique du temps**. La Météorologie, 8^e série, 30, 11-31.
<http://www.smf.asso.fr/Ressources/Coiffier30.pdf>

Ocean Models (web book)

http://www.phy.ornl.gov/csep/CSEP/PS_FILES/OM.PS

N. Daniault (2005) **Océanographie Physique pour l'École Navale**. Cours en ligne, LPO - Université de Bretagne Occidentale, Brest.

http://stockage.univ-brest.fr/~daniault/oceano_physique.pdf

P. Penven, G. Cambon, T.-A. Tan, P. Marchesiello and L. Debreu (2010), **ROMS_AGRIF / ROMTOOLS User's guide**, IRD technical report, Mars, France.

http://roms.mpl.ird.fr/user_manual/doc_roms_agrif_v2.1_19_07_2010.pdf

G. L. Mellor (2004), **POM user guide**.

<http://www.aos.princeton.edu/WWWPUBLIC/htdocs/pom/>

A Semtner (1995), **Modeling Ocean Circulation**, Science 269,5229,1379-1385

<http://www.mbari.org/staff/braccio/Science/semtner.html>

ROMS Group, **User's Manual for the Regional Ocean Modeling System and the Terrain - following Ocean Modeling System (ROMS/TOMS)**

<http://www.myroms.org/documentation/>

Copin-Montégut G., **Le Courant Géostrophique**.

<http://www.obs-vlfr.fr/Enseignement/enseignants/copin/Geostro.pdf>

E.Blayo **Nesting Ocean Models** présentation à l'école d'été GODAE

http://www.meom.hmg.inpg.fr/Web/GODAE_school/Pdf/E.Blayo.pdf

Steven K. Baum, « **Ocean circulation models** » **Center for Coastal and Estuarine Science Web Page**

http://stommel.tamu.edu/~baum/ocean_models.html

A. Delcros (1998) **Les commandes fondamentales de Linux**

<http://www.linux-france.org/article/debutant/debutant-linux.html>

How to create a web page

<http://www.make-a-web-site.com>

Wikipedia

https://en.wikipedia.org/wiki/Main_Page

Lecture Notes

Introduction

The numerical modelling of ocean circulation is a very young discipline that was developed mostly during the second half of the 20th century, benefiting greatly from the simultaneous technological progress and the development of numerical methods especially in the field of meteorology and weather prediction. In fact, the equations that describe the behaviour of the ocean and the atmosphere are very similar.

Before 1900, meteorology and oceanography were mainly descriptive.

This first map of the Gulf Stream was printed in 1769-1770 and rediscovered by P. Richardson (1980). It shows a remarkably precise drawing by B. Franklin for the British Postal Service and was based on information provided by the whaling captain Timothy Folger.

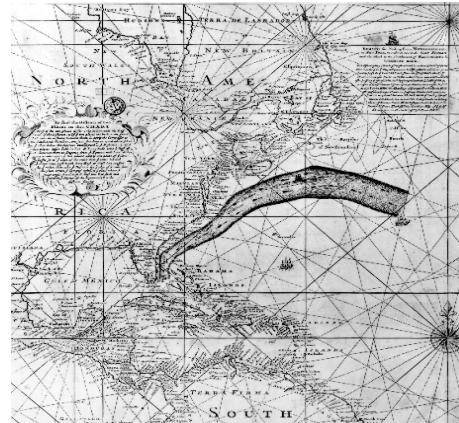


Image courtesy R. Peterson et al. (1996) in Progress in Oceanography.
http://sam.ucsd.edu/sio210/gifimages/Franklin_folger.gif

Coiffier (2000) presented a very interesting historical overview of the techniques used to construct numerical weather prediction models at the time, based on the numerical solution of the governing equations, i.e., to determine the future values of characteristic quantities based on the initial values known from observations. Numerical models of the atmosphere were built on this principle and have become indispensable tools for weather forecasting, gradually replacing methods based on observational approaches to predict the movement and evolution of visually identifiable atmospheric structures (such as large atmospheric centres of action and fronts).

“The equations used are general equations of fluid mechanics, established at the beginning of the 20th century, that a simplified based on order-of-magnitude comparisons of the different terms of the equations, particularly for the case of the Earth’s atmosphere and according to the scales to be described. The use of numerical simulations is essential to solve these systems of **non-linear equations** as they cannot usually be solved analytically.

The construction of a numerical model of the atmosphere and ocean consists of two steps: (i) establish a system of equations, and (ii) replace the equations relating to continuous variables with their “discretized” equivalents, the solutions to which are obtained via an appropriate algorithm. The results of a numerical forecast therefore depend on the general properties of the numerical solutions to the equations, which forms part of the field of dynamic meteorology, and on the effects due to the particular approach taken during the discretization, which must be examined in detail.

The implementation of any such algorithm implies having a calculation tool at one’s disposal that is sufficiently powerful while still easy to use. This is why advances in numerical weather prediction have been closely linked to the phenomenal technological advances in the field of computers since their inception at the end of World War II.” (translated from Coiffer, 2000).

In the 20th century: modelling became part of mathematical physics.

The first conceptual models:

- 1904 Bjerknæs: weather forecasting based on initial conditions and physical laws
- 1908 Poincaré: difficulties linked to non-linearities, notion of spectral windows
- 1914 Bjerknæs: identification of the practical problem of numerical computation
- 1922 Richardson: simplification of the mathematical problem through physical approximations

As early as 1904, the Norwegian Vilhelm Bjerknæs recognized that weather forecasting was fundamentally a deterministic problem with initial values in the mathematical sense of the term

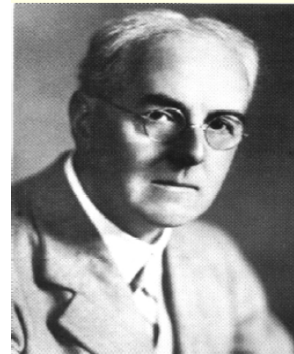
(Bjerknes, 1904): “If it is true, as any scientist believes, that subsequent states of the atmosphere develop from preceding ones according to physical laws, one will agree that the necessary and sufficient conditions for a rational solution of the problem of meteorological prediction are the following:

1. One has to know with sufficient accuracy the state of the atmosphere at a given time.
2. One has to know with sufficient accuracy the laws according to which one state of the atmosphere develops from another.”

However, Bjerknes realised that the difficulty of the problem comes from the need to solve a system of non-linear partial differential equations for which one does not have, in the general case, analytical solutions.

Richardson was therefore trying to show that weather forecasts were possible using computational methods, at a time when electronic computers did not yet exist. He published his method in 1922 in a book called *Weather Prediction by Numerical Process*.

Richardson advocated the use of computational methods to simplify the equations and indicated that it was necessary to start from a meteorological situation with a large amount of land and altitude data from all over the globe.



https://en.wikipedia.org/wiki/Lewis_Fry_Richardson

Using standardized forms, to speed up the calculation by dividing up the work, he estimated that 64,000 people were needed to be able to actually issue the forecast ahead of the forecast events.

Richardson divided the globe into a rectangular grid with dimensions of 230 km (latitudinal) by 200 km (longitudinal) which divided the global atmosphere into 3,200 vertical columns. Furthermore, he divided the atmosphere into horizontal layers at 4, 7, and 12 km altitude and proposed a time step for the simulation of 3 hours.

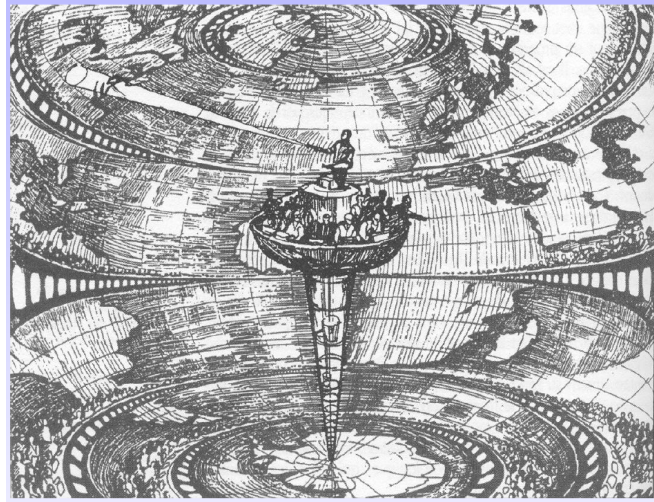
This huge system was dubbed the “**weather forecast factory**”.

Excerpt from: Richardson, *Weather Prediction by Numerical Process*, Chap. 11/2, pp. 219-220, Cambridge University Press, 1922.

“It took me the best part of six weeks to draw up the computing forms and to work out the new distribution in two vertical columns for the first time. My office was a heap of hay in a cold rest billet. With practice the work of an average computer might go perhaps ten times faster. If the time-step were 3 hours, then 32 individuals could just compute two points so as to keep pace with the weather, if we allow nothing for the very great gain in speed which is invariably noticed when a complicated operation is divided up into simpler parts, upon which individuals specialize. If the co-ordinate chequer were 200 km square in plan, there would be 3200 columns on the complete map of the globe. In the tropics the weather is often foreknown, so that we may say 2000 active columns. So that $32 \times 2000 = 64,000$ computers would be needed to race the weather for the whole globe. That is a staggering figure. Perhaps in some years' time it may be possible to report a simplification of the process. But in any case, the organization indicated is a central forecast-factory for the whole globe, or for portions extending to boundaries where the weather is steady, with individual computers specializing on the separate equations. Let us hope for their sakes that they are moved on from time to time to new operations.

After so much hard reasoning, may one play with a fantasy?

Imagine a large hall like a theatre, except that the circles and galleries go right round through the space usually occupied by the stage. The walls of this chamber are painted to form a map of the globe. The ceiling represents the north polar regions, England is in the gallery, the tropics in the upper circle, Australia on the dress circle and the Antarctic in the pit. A myriad computers are at work upon the weather of the part of the map where each sits, but each computer attends only to one equation or part of an equation. The work of each region is coordinated by an official of higher rank.



The model by Richardson, 1922

Imagining a human calculator (of 64,000 individuals) automated and parallelized with a calculation power of 1 Flops (Floating point operations per second).

Design by Alf Lannerbaeck, published by the Swedish magazine *Dagens Nyheter* on 22 September 1984.

Numerous little "night signs" display the instantaneous values so that neighbouring computers can read them. Each number is thus displayed in three adjacent zones so as to maintain communication to the North and South on the map. From the floor of the pit a tall pillar rises to half the height of the hall. It carries a large pulpit on its top. In this sits the man in charge of the whole theatre; he is surrounded by several assistants and messengers. One of his duties is to maintain a uniform speed of progress in all parts of the globe. In this respect he is like the conductor of an orchestra in which the instruments are slide-rules and calculating machines. But instead of waving a baton he turns a beam of rosy light upon any region that is running ahead of the rest, and a beam of blue light upon those who are behindhand.

Four senior clerks in the central pulpit are collecting the future weather as fast as it is being computed, and despatching it by pneumatic carrier to a quiet room. There it will be coded and telephoned to the radio transmitting station. Messengers carry piles of used computing forms down to a storehouse in the cellar. In a neighbouring building there is a research department, where they invent improvements. But there is much experimenting on a small scale before any change is made in the complex routine of the computing theatre. In a basement an enthusiast is observing eddies in the liquid lining of a huge spinning bowl, but so far the arithmetic proves the better way. In another building are all the usual financial, correspondence and administrative offices. Outside are playing fields, houses, mountains and lakes, for it was thought that those who compute the weather should breathe of it freely."

However, a small test to predict the evolution of atmospheric pressure using this method, but without all the personnel, gave a very disappointing result: the calculated variation was 145 hPa in 6 hours, a value quite impossible when one considers that a variation of 20 hPa is considered to be an extreme variation for this same time. In fact, the actual variation was almost zero.

Richardson's error was to discretize the derivatives of the primitive atmospheric equations without taking into account the experimental variability of the data.

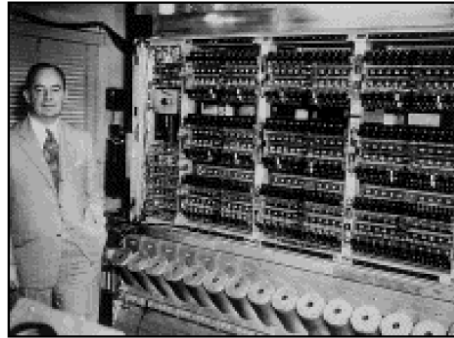
These derivatives involve divisions of numbers that are very close to each other (e.g., variation of pressure or temperature over time). However, the variation of meteorological variables can be of the same order as the measurement uncertainty which can also be temporarily very large. It is therefore necessary to use values that have been smoothed over a certain period to obtain realistic results. Iwe now know that the time step in the calculations should not have exceeded half an hour and that the initial data, not having undergone quality control, introduced instabilities to the calculation. Nevertheless, Richardson demonstrated that computational weather forecasting was possible and many problems of human and financial logistics were later solved by the advent of computers.

In 1928, the German mathematicians Courant, Friedrichs, and Levy studied in a systematic manner how the partial differential equations could be solved by using “finite differences” and specified the constraints that need to be taken into account during their discretization (Courant et al., 1928).

In 1939, the Swedish scientist Carl-Gustav Rossby showed that the equation for the conservation of absolute vorticity allows for the accurate interpretation of the movement of atmospheric centres of action (Rossby et al., 1939).

Appearance of electronic calculators

In 1946, the first electronic calculator, the Eniac (*Electronic Numerical Integrator Analyser and Computer*), was created at the University of Pennsylvania in Philadelphia (USA), due to the decisive impetus of an American mathematician of Hungarian origin, John von Neumann, who proposed to use this machine for weather forecasting.



In 1948, the American Jule Charney proposed a simplification of the general system of equations, known as the “quasi-geostrophic approximation”, and found as a special case the equation studies by Rossby (Charney, 1948).

Finally, in 1950, J. Charney, the Norwegian R. Fjørtoft, and J. von Neumann carried out the first numerical weather forecast (Charney et al., 1950): for their experiment they used the conservation of absolute vorticity equation and ran their calculation on the Eniac machine, installed at Aberdeen (Maryland, USA). The results obtained for the prediction of the 500hPa geopotential height, characteristic of the middle atmosphere, were sufficiently accurate to make this historical experiment the starting point of modern numerical prediction (Platzmann, 1979). In response to Charney, who sent him the article describing the experiment, Richardson wrote in 1952: “*Allow me to congratulate you and your associates on the remarkable success you have had at Princeton and on the prospects for improvement you indicate*” (Ashford, 1985).

Performance improvements

- 1955 baroclinic model by Charney on IBM701 (10,000 Flops)
- But until 1962 a barotropic model is used to supply 3-day forecasts (32×23 grid)
- Identification of problems due to numerical instabilities

New model concepts

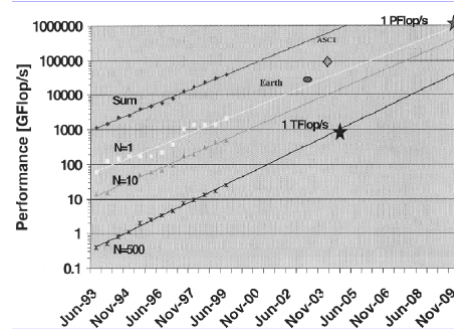
- 1956 Phillips: numerical verifications of the validity of the simplifying physical assumptions: non-linear quasi-geostrophic model verifying that a linear analytical model is valid for a given parameter range
- Linear and non-linear numerical instabilities
- Discussions on the advantages of robustness vs precision
- 1966 Arakawa: discretisation that preserves the non-linear properties

Available resources

- Increase in computational power used mostly to achieve higher spatial and temporal resolutions

Computational power

- Exponential growth: Moore's Law: computational power doubles every 18 months.
- A supercomputer from 1992 that cost €1M then had the same computing power as an ordinary PC in 2001.
- Expected limitations?
 - 1GHertz limits physical distances to 30cm (based on the speed of light)
 - Using light-based circuits introduces limitations by the wavelengths of visible light
 - Heating dissipation problem



New Architectures

- SISD single instruction, single data: classical processor
- SIMD single instruction, multiple data: vector processor
- MIMD multiple instruction, multiple data: parallel processor
 - shared memory: typically 16 processors efficiently operated
 - distributed memory: massive parallelization possible but recoding (1024 processors)

Programming always in FORTRAN ...

First appearance of oceanographic models

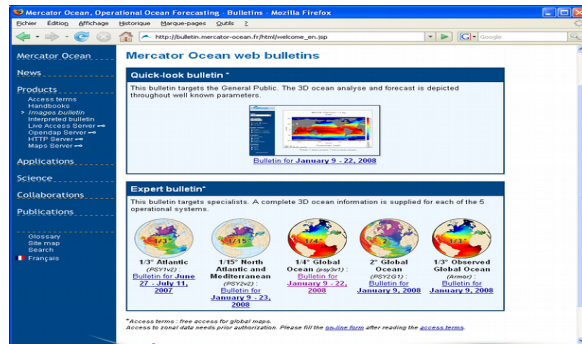
- 1963 Bryan: barotropic model
- 1969 Bryan: baroclinic model that accounts for topography and uses turbulence closure
- In meteorology, 3D primitive equation models exist since 1963 (Smagorinsky).

And since then?

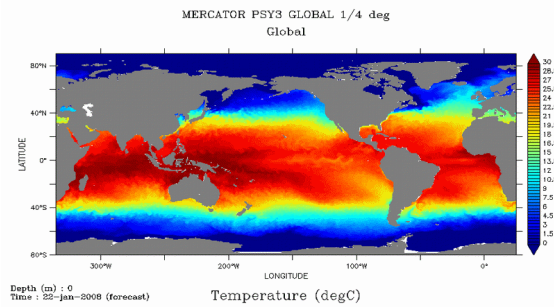
- The model by Bryan (GFDL, MOM) remains nearly unchanged and is still the most widely used
- New developments often originate from meteorological models (assimilation, closures, spectral methods, semi-Lagrangians)
- To rebuild a model from scratch is a major time investment
- Additions to models: data assimilation, improved numerical schemes, more complex closures

Examples of operational models in oceanography

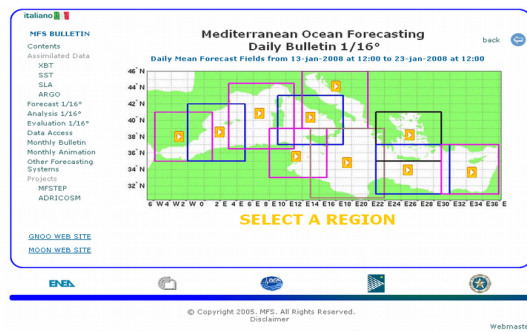
MERCATOR OCEAN



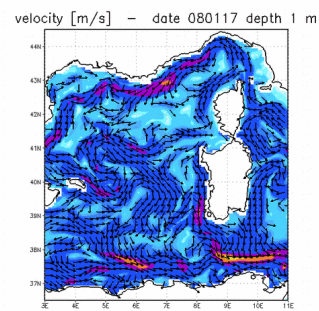
http://bulletin.mercator-ocean.fr/html/welcome_en.jsp



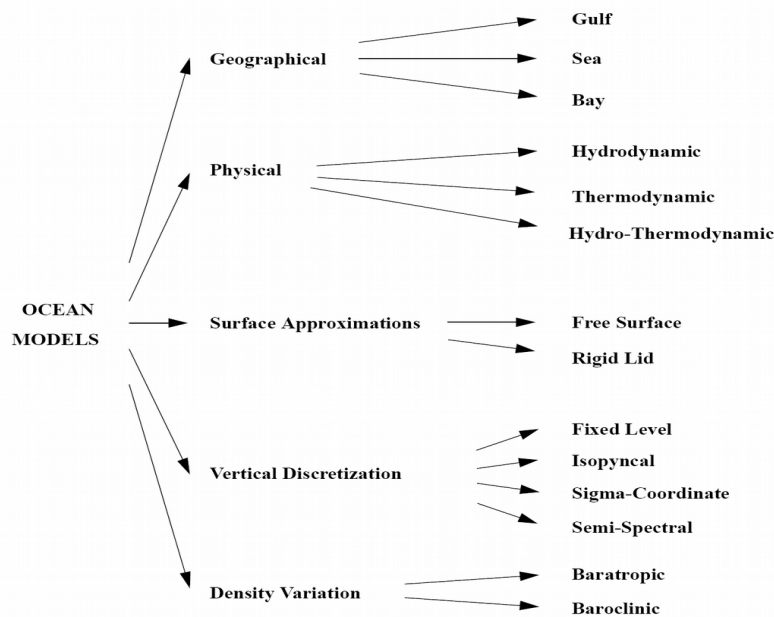
MFS



<http://www.bo.ingv.it/mfs/>



Tentative model classification



Ocean Models (web book)
http://www.phy.ornl.gov/csep/CSEP/PS_FILES/OM.PS

Presentation by P.Penven at the 2006 University of Toulon Summer School on Modelling http://lseet.univ-tln.fr/ecolette/Documents/ROMS_TOOLS.pdf

Introduction to ROMSTOOLS

TOULON – September 2006

Pierick.Penven@ird.fr

2.1- Introduction: ROMS

Equations to solve : the primitive equations (PE)

$$\frac{\partial u}{\partial t} + u \cdot \nabla u - fv = -\frac{1}{\rho_0} \frac{\partial P}{\partial x} + A_h \nabla_h^2 u + A_v \frac{\partial^2 u}{\partial z^2}$$

$$\frac{\partial v}{\partial t} + u \cdot \nabla v + fu = -\frac{1}{\rho_0} \frac{\partial P}{\partial y} + A_h \nabla_h^2 v + A_v \frac{\partial^2 v}{\partial z^2}$$

$$0 = \frac{\partial P}{\partial z} + \rho g$$

$$0 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$$

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = K_h \nabla_h^2 T + K_v \frac{\partial^2 T}{\partial z^2}$$

$$\frac{\partial S}{\partial t} + u \cdot \nabla S = K_h \nabla_h^2 S + K_v \frac{\partial^2 S}{\partial z^2}$$

$$\rho = \rho(T, S, z)$$

+ Boundary conditions:

Surface boundary conditions (z=η):

$$\frac{d\eta}{dt} = w$$

$$A_v \frac{\partial u}{\partial z} = \tau_x$$

$$A_v \frac{\partial v}{\partial z} = \tau_y$$

$$K_v \frac{\partial T}{\partial z} = \frac{Q}{\rho_0 C_p}$$

$$K_v \frac{\partial S}{\partial z} = \frac{S(E-P)}{\rho_0}$$

Evaporation – Rain : Salt flux

Bottom boundary conditions (z=-H):

$$w = -u \cdot \nabla H$$

$$A_v \frac{\partial u}{\partial z} = -ru$$

$$A_v \frac{\partial v}{\partial z} = -rv$$

$$K_v \frac{\partial T}{\partial z} = 0$$

$$K_v \frac{\partial S}{\partial z} = 0$$

Bottom friction

Principle:

Discretization : horizontal grid

× - u points
 □ - v points
 ○ - ρ points

Horizontal Arakawa "C" grid :

Vertical, topography following, "S" coordinate :

Regional Ocean Modelling System (ROMS), 3 versions:

- Rutgers University: <http://marine.rutgers.edu/po/index.php?page=&model=roms>
- UCLA: http://www.atmos.ucla.edu/cesr/ROMS_page.html
- IRD: http://www.brest.ird.fr/Roms_tools/

ROMS

- Solves the **Primitive Equations: Boussinesq approximation** + **hydrostatic** vertical momentum balance.
- Discretized in coastline- and **terrain-following curvilinear coordinates**.
- Split-explicit, **free-surface** oceanic model: short time steps to advance the barotropic momentum equations, and a much larger time step for temperature, salinity, and baroclinic momentum.
- Special 2-way time-averaging procedure for the barotropic mode, which satisfies the 3D continuity equation.
- Baroclinic mode discretized using a **third-order accurate** predictor (Leap-Frog) and corrector (Adams-Molton) time-stepping algorithm: substantial increase in the permissible time-step size.
- Designed to be optimized on shared-memory **parallel computer** architectures.
- Parallelization by two-dimensional subdomain partitioning.
- Multiple subdomains can be assigned to each processor to optimize the use of processor cache memory.
- Super-linear scaling: performance growth even faster than the number of CPUs.
- Ported successfully to distributed-memory platforms (clusters, earth simulator)
- **Third-order, upstream-biased advection scheme**: allows the generation of steep gradients.
- Improved calculation of the horizontal pressure gradients.
- Subgrid-scale vertical mixing processes: **K-Profile Parametrization (KPP) boundary-layer scheme**.
- **Open boundaries**: active, implicit, upstream-biased radiation conditions.
- **Nesting capability**: AGRIF (Adaptive Grid Refinement in Fortran) library. Arbitrary number of fixed grids.

- Sediment module.
- Biogeochemical module.
- Float tracking module.

- ROMSTOOLS: Matlab scripts to help with the generation and visualization of ROMS configurations.

ROMS References:

- Shchepetkin, A., and J.C. McWilliams, 1998: Quasi-monotone advection schemes based on explicit locally adaptive dissipation. *Monthly Weather Review*, 126, 1541-1580.
- Haidvogel D. B., H. Arango, K. Hedstrom, A. Beckman, P. Malanotte-Rizzoli, and A. Shchepetkin, 2000: Model evaluation experiments in the North Atlantic basin: simulations in nonlinear terrain following coordinates. *Dyn. Atmos. Oceans*, 32, 239-281.
- Marchesiello, P., J.C. McWilliams, and A. Shchepetkin, 2001: Open boundary conditions for long-term integration of regional oceanic models. *Ocean Modelling*, 3, 1-20.
- Ezer, T., H.G. Arango and A.F. Shchepetkin, 2002: Developments in terrain-following ocean models: intercomparisons of numerical aspects., *Ocean Modelling*, 4, 249-267
- Marchesiello P., J.C. McWilliams, and A. Shchepetkin, 2003: Equilibrium structure and dynamics of the California Current System. *Journal of Physical Oceanography*, 33, 753-783.
- Shchepetkin, A.F., and J.C. McWilliams, 2003: A method for computing horizontal pressure-gradient force in an ocean model with a non-aligned vertical coordinate. *J. Geophys. Res.*, 108, C3, 3090, doi:10.1029/2001JC001047.
- Shchepetkin, A., and J.C. McWilliams, 2005: The Regional Oceanic Modeling System: A split-explicit, free-surface, topography-following-coordinate ocean model. *Ocean Modelling*, 9, 347-404.
- Penven P., L. Debreu, P. Marchesiello, and J.C. McWilliams, 2006: Evaluation and application of the ROMS 1-way embedding procedure to the central california upwelling system. *Ocean Modelling*, 12, 157-187.

<h3>2.2 - Why ROMSTOOLS ?</h3> <p>What do we need to run an experiment ?</p>	<p>What do we need to run an experiment ? A computer with the model on it + data (netcdf files) to run the model.</p> <p>Necessary data:</p> <p>Grid data (roms_grd.nc):</p> <ul style="list-style-type: none"> - positions of the grid points. - size of the grid cells. - bottom topography. - land mask. <p>Surface forcing data (roms_frc.nc):</p> <ul style="list-style-type: none"> - wind stress. - surface heat flux. - surface freshwater flux. <p>Initial conditions (roms_ini.nc):</p> <ul style="list-style-type: none"> - Temperature, Salinity, Currents, Sea surface elevation. <p>Lateral boundary conditions (roms_clm.nc):</p> <ul style="list-style-type: none"> - Temperature, Salinity, Currents, Sea surface elevation. <p style="text-align: right;">} Provided by ROMSTOOLS</p>
<p>Strategy:</p>	<h3>2.3 - System</h3> <p style="text-align: center;">UNIX or LINUX !!!</p> <p>ROMS: FORTRAN programme (~55,000 lines of FORTRAN code!).</p> <p>ROMSTOOLS: Matlab scripts + datasets to prepare the files for ROMS.</p> <p>PC configuration:</p> <ul style="list-style-type: none"> Minimum hardware: 256 RAM, P3, 2 G of free disc space. Operating system: Linux (Good success with Mandrake and Fedora). Fortran compiler: g77 or Intel Fortran Compiler: http://www.intel.com/cd/software/products/asmo-na/eng/compilers/flin/index.htm Matlab for Linux.
<h3>2.4 – How to get the tools ?</h3> <p>http://www.brest.lird.fr/Roms_tools/ http://sea.uct.ac.za/perven/Roms_tools/</p>	

Fundamentals

The equations solved by ROMS, SYMPHONIE, POM, and other models are the so-called primitive equations.

The primitive equations

The primitive equations are described using a Cartesian coordinate system (x, y, z) , where the x-axis points toward the east, the y-axis toward the north, and the z-axis vertically upward. The origin is usually situated at the sea surface (at rest).

The equations of motion

The change in mean horizontal velocity (in the sense of turbulence) of particles in a geophysical fluid is described by the Navier-Stokes equations:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} &= -\frac{1}{\rho_o} \frac{\partial P}{\partial x} + f v - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z} \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} &= -\frac{1}{\rho_o} \frac{\partial P}{\partial y} - f u - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z} \end{aligned}$$

u , v , and w are the non-turbulent or “mean” horizontal and vertical velocity components;

u' , v' , and w' are the turbulent velocity components;

f is the Coriolis parameter;

P is pressure;

ρ_o is the reference density of seawater in the Boussinesq sense;

The first term corresponds to the local change in velocity as a function of time. The second, third, and fourth terms represent the horizontal advection. The fifth term is the pressure gradient.

The sixth term is the Coriolis term to account for the Earth’s rotation; if the flow is at a sufficiently small scale, the Earth’s surface can be approximated by its tangent plane and the Coriolis force can be considered i) constant: f -plane approximation, $f = f_o$, coastal dynamics; ii) to vary linearly with latitude: β -plane approximation, $f = f_o + \beta y$, for both regional dynamics and large scale.

The remaining three terms account for turbulence. In what follows, we will look at some examples of so-called turbulence closure schemes in more detail (i.e., to write the terms containing the turbulence velocities u' , v' , and w' as a function of the velocity averages to avoid having to add new variables for which we do not know the equations). At the base of these closure schemes is the theory known as “Newtonian closure” which states that just as for molecular viscosity, one can introduce coefficients of turbulence viscosity and rewrite the means of the products of the turbulent velocity components in terms of the average velocities:

$$\begin{aligned} \overline{u'u'} &= -A_x \frac{\partial u}{\partial x} ; & \overline{u'v'} &= -A_y \frac{\partial u}{\partial y} ; & \overline{u'w'} &= -A_z \frac{\partial u}{\partial z} ; \\ \overline{v'u'} &= -A_x \frac{\partial v}{\partial x} ; & \overline{v'v'} &= -A_y \frac{\partial v}{\partial y} ; & \overline{v'w'} &= -A_z \frac{\partial v}{\partial z} . \end{aligned}$$

For scales that are typical for a large part of motions in the ocean, the equation for the vertical velocity component reduces to the hydrostatic equation which describes the balance between pressure and gravitational forces. It gives the pressure as a function of depth:

$$P(z) = P_a + g \int_z^\eta \rho \cdot dz$$

where P_a is the atmospheric pressure, g the acceleration due to gravity, and η the sea surface elevation

with respect to the $z=0$. $z = \eta(x, y, t)$ therefore constitutes the free surface elevation of the ocean while $z = -h(x, y)$ corresponds to the sea bed. ρ density.

The Boussinesq approximation

Since vertical density differences in the ocean are of the order of 0.1%, we can replace ρ with a reference density, ρ_0 , without incurring too large an error. In contrast, the horizontal density differences ($\partial \rho / \partial x_i$) are not negligible. Boussinesq suggested to write density as

$$\rho \equiv \rho_0 + \rho'(x, y, z, t) \text{ avec } \rho' \ll \rho_0 \quad O(10^{-3})$$

and to retain in the *buoyancy* (Archimedes force) term only those quantities that have this order of magnitude.

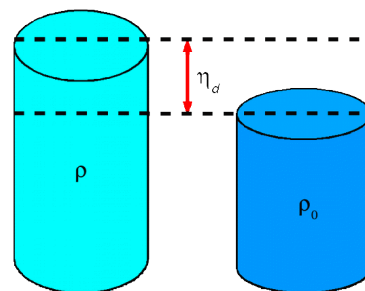
The horizontal derivatives become

$$\frac{1}{\rho_0} \frac{\partial P}{\partial x} = \frac{1}{\rho_0} \frac{\partial P_a}{\partial x} + g \frac{\partial \eta}{\partial x} + \frac{g}{\rho_0} \int_z^\eta \frac{\partial \rho'}{\partial x} dz$$

$$\frac{1}{\rho_0} \frac{\partial P}{\partial y} = \frac{1}{\rho_0} \frac{\partial P_a}{\partial y} + g \frac{\partial \eta}{\partial y} + \frac{g}{\rho_0} \int_z^\eta \frac{\partial \rho'}{\partial y} dz$$

The second and third terms on the RHS are the barotropic and baroclinic contributions to the pressure gradient, respectively.

The last term containing the integral is often written as η_d , as it is in fact a length that corresponds to the height difference between a water column of density ρ_0 and a column of “real” density ρ . In summary, the Boussinesq approximation allows replacing the exact density by its reference value, ρ_0 , everywhere except in the gravity acceleration term. In the pressure terms, the only remaining part is due to η_d , the dynamic height, as it is the principal driver of large-scale flows.



Often, the reference density is taken to be the maximum density such that the dynamic height is zero or positive.

If the pressure gradient $-\partial p / \partial x_i$ vanishes at a sufficiently great depth $z = -h$, Equation (1.12) becomes

$$(1.13) \quad \eta = \eta_d(-h)$$

by neglecting the constant of integration, i.e., by making a clever choice of the zero level of the free surface. In this case, the sea level topography can be directly determined from the density distribution (Sverdrup 1942).

In numerical models, the calculation of the dynamic height is not really interesting since we can resolve the complete equations numerically now (Sverdrup didn't have a computer!). However, the use of the Boussinesq approximation makes it possible to increase the precision of our calculations in baroclinic terms.

As an example, we can take a look at how these equations are included in the POM model, written in an x, y, σ coordinate system (see GL Mellor, 2004, **POM user guide**):

$$\begin{aligned} \frac{\partial UD}{\partial t} + \frac{\partial U^2 D}{\partial x} + \frac{\partial UVD}{\partial y} + \frac{\partial U\omega}{\partial \sigma} - fVD + gD \frac{\partial \eta}{\partial x} \\ + \frac{gD^2}{\rho_o} \int_{\sigma}^{\sigma'} \left[\frac{\partial \rho'}{\partial x} - \frac{\sigma'}{D} \frac{\partial D}{\partial x} \frac{\partial \rho'}{\partial \sigma'} \right] d\sigma' = \frac{\partial}{\partial \sigma} \left[\frac{K_M}{D} \frac{\partial U}{\partial \sigma} \right] + F_x \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{\partial VD}{\partial t} + \frac{\partial UVD}{\partial x} + \frac{\partial V^2 D}{\partial y} + \frac{\partial V\omega}{\partial \sigma} + fUD + gD \frac{\partial \eta}{\partial y} \\ + \frac{gD^2}{\rho_o} \int_{\sigma}^{\sigma'} \left[\frac{\partial \rho'}{\partial y} - \frac{\sigma'}{D} \frac{\partial D}{\partial y} \frac{\partial \rho'}{\partial \sigma'} \right] d\sigma' = \frac{\partial}{\partial \sigma} \left[\frac{K_M}{D} \frac{\partial V}{\partial \sigma} \right] + F_y \end{aligned} \quad (4)$$

The shallow water equations

If we neglect the baroclinic and atmospheric pressure contributions, the horizontal pressure gradient can be expressed as: $\nabla_H P = \rho_o g \nabla_H \eta$

This equation states that the forces acting on the fluid are purely horizontal, so we can assume that the horizontal components of the velocity will be independent of z . Thus, the fourth terms in the Navier-Stokes equations are zero and the vertical turbulent viscosity term is replaced by the conditions at the vertical boundaries, namely wind forcing $\vec{F} \equiv (F_x, F_y)$ and bottom friction $\vec{B} \equiv (B_x, B_y)$.

The equations of motion become

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -g \frac{\partial \eta}{\partial x} + f v + A_h \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] + F_x + B_x \quad (1) \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -g \frac{\partial \eta}{\partial y} - f u + A_h \left[\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right] + F_y + B_y \quad (2) \end{aligned}$$

Note that we now have two equations with three unknowns (the two velocity components and the sea surface elevation).

We therefore need to add a third equation, but which one? The continuity equation written as a function of those three variables.

We start with the continuity equation for an incompressible fluid:

$$\nabla \cdot \vec{u} = \frac{\partial w}{\partial z} + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0$$

By integrating from the sea bed to the free surface and given that the velocities and their derivatives are independent of z , we obtain:

$$w_{z=\eta} - w_{z=-h} + (h+\eta) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0 .$$

At the surface

$$w_{z=\eta} = \frac{d\eta}{dt} = \frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} + v \frac{\partial \eta}{\partial y} .$$

At the sea bed, considering that the bathymetry does not change with time

$$w_{z=-h} = -\frac{dh}{dt} = -u \frac{\partial h}{\partial x} - v \frac{\partial h}{\partial y} .$$

Through substitution we obtain:

$$\frac{\partial \eta}{\partial t} + \frac{\partial(Hu)}{\partial x} + \frac{\partial(Hv)}{\partial y} = 0 \quad \text{avec } H = h + \eta \quad (3)$$

The ensemble of these three equations (1-3) constitutes the shallow water equations (SWE). Like the hydrostatic approximation, they are valid whenever the water depth is small compared to the horizontal scales one wishes to study.

We will see later on how these equations are used in a 3D model that uses so-called *mode splitting*.

Conservation equations for heat and salt

They describe the transport of temperature, T , and salinity, S :

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \vec{\nabla} T = - \frac{\partial(\overline{T'u'})}{\partial x} - \frac{\partial(\overline{T'v'})}{\partial y} - \frac{\partial(\overline{T'w'})}{\partial z} + \frac{H_c}{\rho_o C_p} \frac{\partial I}{\partial z}$$

$$\frac{\partial S}{\partial t} + \vec{v} \cdot \vec{\nabla} S = - \frac{\partial(\overline{S'u'})}{\partial x} - \frac{\partial(\overline{S'v'})}{\partial y} - \frac{\partial(\overline{S'w'})}{\partial z}$$

We will see later on that the turbulent fluxes of heat and salt (first, second, and third terms on the RHS) are modelled using the concept of turbulence diffusivity.

H_c is the incident solar radiation.

C_p is the specific heat: $C_p = 3950 \text{ J/(kg K)}$

$I(z)$ is the fraction of this flux that reaches the depth z : $I(z) = \exp(-\lambda z)$ with $\lambda = 0.05 \text{ m}^{-1}$, the attenuation coefficient, chosen from the tabulated values for clear water in Jerlov (1976).

“Density”

Density is the most important thermodynamic property of seawater when studying ocean circulation. Small horizontal variations in density (generated, for instance, by differences in incident solar radiation) can induce large ocean currents. Strictly speaking,

density = (mass of 1 m^3 of seawater)/(mass of 1 m^3 of distilled water at 4°C) [no units].

IES80: International Equation of State of Seawater

$$\rho(S, t, p) = \rho(S, t, 0) \left[1 - p/K(S, t, p) \right]$$

with

$$\begin{aligned} \rho(S, t, 0) = & 999.842594 + 6.793952 \times 10^{-2} t - 9.095290 \times 10^{-2} t^2 + 1.001685 \times 10^{-4} t^3 - 1.120083 \times 10^{-6} t^4 \\ & + 6.536332 \times 10^{-9} t^5 + 8.24493 \times 10^{-1} S - 4.0899 \times 10^{-2} tS + 7.6438 \times 10^{-5} t^2 S - 8.2467 \times 10^{-7} t^3 S \\ & + 5.3875 \times 10^{-9} t^4 S - 5.72466 \times 10^{-3} S^2 + 1.0227 \times 10^{-4} tS^2 - 1.6546 \times 10^{-6} t^2 S^2 \\ & + 4.8314 \times 10^{-4} S^3 \end{aligned}$$

and

$$\begin{aligned} K(S, t, p) = & 19652.21 + 148.4206 t - 2.327105 t^2 + 1.360447 \times 10^{-2} t^3 - 5.155288 \times 10^{-5} t^4 + 3.239908 p \\ & + 1.43713 \times 10^{-3} tp + 1.16092 \times 10^{-4} t^2 p - 5.77905 \times 10^{-7} t^3 p + 8.50935 \times 10^{-5} p^2 \\ & - 6.12293 \times 10^{-6} tp^2 + 5.2787 \times 10^{-8} t^2 p^2 + 54.6746 S - 0.603459 tS + 1.09987 \times 10^{-2} t^2 S \\ & - 6.1670 \times 10^{-5} t^3 S + 7.944 \times 10^{-2} S^2 + 1.6483 \times 10^{-2} tS^2 - 5.3009 \times 10^{-4} t^2 S^2 + 2.2838 \times 10^{-3} pS \\ & - 1.0981 \times 10^{-5} tpS - 1.6078 \times 10^{-6} t^2 pS + 1.91075 \times 10^{-4} pS^2 - 9.9348 \times 10^{-7} p^2 S \\ & + 2.0816 \times 10^{-8} tp^2 S + 9.1697 \times 10^{-10} t^2 p^2 S \end{aligned}$$

<http://lecalve.univ-tln.fr/oceano/ies80/index.html>

However, in oceanography a different definition has been introduced, mostly for practical reasons:

$$\text{density} = \text{in situ density} [\text{kg m}^{-3}] - 1000.$$

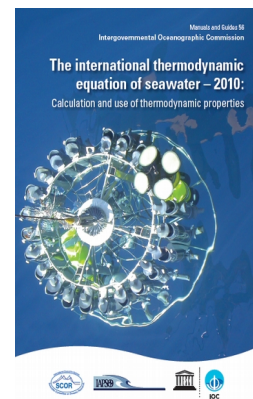
The *in situ* density (also known as specific mass), ρ , of seawater depends on salinity, S , temperature, T , and pressure p .

This dependence of ρ on S , T , and p is the so-called equation of state of seawater. It is an empirical relation that resulted from numerous laboratory studies. The first such equation was established in 1902 by Knudsen and Ekman which was replaced by the “International Equation of State of Seawater (1980)” (EOS-80).

A few years ago, more refined formulations appeared in the literature (Anati, 1999; Feistel 2003, 2005), as well as formulations taking into account the geographic variability linked to the shape of the geoid (for the concept of “*neutral density*” see McDougall, 1987).

In June 2009, the Intergovernmental Oceanographic Commission (IOC), with the support of the Scientific Committee on Oceanographic Research (SCOR) and the International Association for the Physical Sciences of the Oceans (IAPSO), adopted the Thermodynamic Equation of Seawater - 2010 (TEOS-10) as the official definition of properties of seawater and sea ice in marine science.

Oceanographers are now strongly advised to use the algorithms and variables defined by TEOS-10 to report their results.



The fundamental differences between TEOS-10 and EOS-80 are:

(1) TEOS-10 uses Absolute Salinity (SA) to describe the salinity of seawater; SA takes into account the spatial variability in the composition of seawater. In the open ocean, the use of this new salinity has a non-negligible effect on the horizontal density gradients, and thus on the velocities calculated using the “thermal wind” equation.

(2) TEOS-10 uses Conservative Temperature (Q) instead of the potential temperature q . Both temperatures are determined from what amounts to a thought experiment (i.e., their in-situ values are adiabatically and isohalinely normalised to surface pressure). Q can account for variations in the heat capacity of seawater with a precision that is two orders of magnitude better than with q .

(3) The properties of seawater defined by TEOS-10 all derive mathematically from a Gibbs function (through differentiation) and are thus compatible with each other (unlike the now obsolete EOS-80 approach in which different mutually incompatible polynomials defined each thermodynamic variable). To allow any oceanographer to use the new TEOS-10 formalism, two software programs are available:

(i) the Gibbs SeaWater (GSW) oceanographic toolbox (for MATLAB and FORTRAN),

(ii) the Sea-Ice-Air (SIA) library (for Fortran and Visual Basic).

Both are freely available from <http://www.TEOS-10.org>.

In this course, we will make certain simplifications.

“*Potential density*” is defined as the density of a particle that has been adiabatically moved to a reference pressure, for example, the pressure at the sea surface. If the pressure variations are small, around ten atmospheres, we can neglect the contribution by pressure altogether and adopt a simpler version:

$$\rho = \rho(T, S)$$

For instance, in the SYMPHONIE model density is calculated from S and T only (note the absence of p) in the diagnostic equation of state:

$$\rho = \rho_o [1 - \alpha(T - T_o) + \beta(S - S_o)]$$

where α and β are the thermal expansion and haline contraction coefficients, respectively, with values of $2.41 \cdot 10^{-4} \text{ K}^{-1}$ and $7.45 \cdot 10^{-4} \text{ psu}^{-1}$.

T_o and S_o are the reference temperature and salinity, respectively, with $T_o = 19.7 \text{ }^\circ\text{C}$, $S_o = 35 \text{ }^\circ\text{o}$, with respect to $\rho_o = 1024.8 \text{ kg/m}^3$.

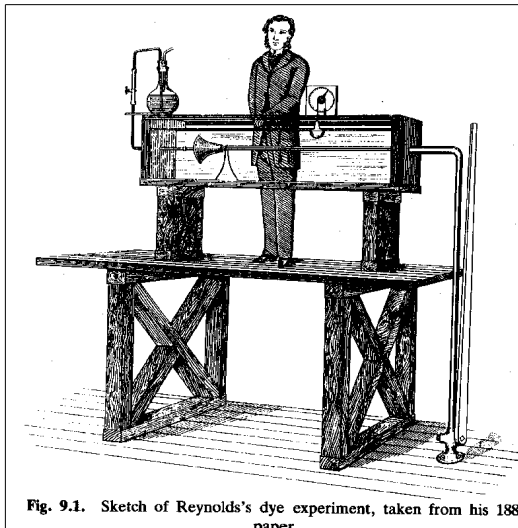


Fig. 9.1. Sketch of Reynolds's dye experiment, taken from his 1883 paper

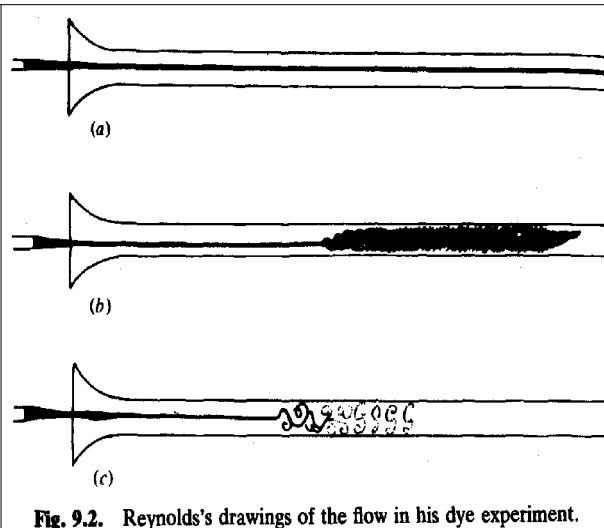


Fig. 9.2. Reynolds's drawings of the flow in his dye experiment.

His results were as follows:

1. At low velocities the streak of dye extended in a straight line along the tube.
2. If the water in the tank was not at rest, the streak would shift about the tube.
3. As the velocity increased, at some point in the tube, the color band would all at once mix up with the surrounding waters. When viewing the tube with an electric spark, this mixed fluid actually looked like a bunch of coherent eddies.

http://www.marine.maine.edu/~eboss/classes/SMS_491_2003/Week_5.htm

Turbulence “closure”

The indeterminacy that resides in the turbulent flow terms of the primitive equations does not allow the system of equations to be solved. It is necessary to add equations to “close” the system, i.e., make it solvable. Using these equations, we express the indeterminate terms as a function of variables calculated by the model. This is achieved by using a relation between the turbulent flows and the vertical gradients of the non-turbulent parameters (Newtonian closure hypothesis). Flow in the ocean is generally turbulent, especially in the surface and bottom layers, and the timescales of these movements are typically greater than the scales that are typical of turbulent fluctuations. In these circumstances it is possible to use the Reynolds approach and think of the equations of continuity and motion as relating to a suitably defined mean flow.

Mean flow equations (Reynolds)

For a turbulent flow, rather than looking for the instantaneous velocity, given by the Navier-Stokes equations seen previously, we are looking for a velocity that is smoothed over time, i.e., averaged over a period of time the length of which depends on the type of phenomenon studied.

At the same time, for each of the variables (velocity and pressure components) we do the following decomposition $u = \bar{u} + u'$, where \bar{u} is the mean and u' the variability about \bar{u} such that $\overline{u'} = 0$:

$$\bar{u} = \frac{1}{T} \int_0^T u dt \quad \overline{u'} = \frac{1}{T} \int_0^T u' dt = 0$$

This technique was developed by Osborne Reynolds (http://en.wikipedia.org/wiki/Osborne_Reynolds). For example, let us look at the mean of the product of two independent components u and v :

$$\overline{uv} = \frac{1}{T} \int_0^T uv dt = \frac{1}{T} \int_0^T \bar{u}\bar{v} dt + \frac{1}{T} \int_0^T \bar{u}v' dt + \frac{1}{T} \int_0^T u'\bar{v} dt + \frac{1}{T} \int_0^T u'v' dt$$

hence

$$\overline{uv} = \overline{u}\overline{v} + \overline{u'v'} + \overline{u'}\overline{v'} + \overline{u'v'}$$

but since $\overline{u'} = \overline{v'} = 0$

$$\overline{uv} = \overline{u}\overline{v} + \overline{u'v'}$$

T represents a period of time long enough for the mean values to be independent of time. There are analogous definitions for v , w , p , etc. Even if $\overline{u'} = 0$ it should be noted that these variabilities themselves can be of the same order of magnitude as u . Furthermore, the fluctuations superimposed on the mean velocity vector are three-dimensional, i.e., u' , v' , w' are always present even if the mean flow itself is one- or two-dimensional.

We can show that all the linear terms in the Navier-Stokes equations keep the same shapes for the mean and instantaneous flows. However, the advection terms (to be shown in an exercise) become:

$$\overline{\vec{V} \cdot \nabla u} = \overline{\vec{V}} \cdot \nabla \overline{u} + \underbrace{\overline{u' \frac{\partial u'}{\partial x}} + \overline{v' \frac{\partial u'}{\partial y}} + \overline{w' \frac{\partial u'}{\partial z}}}_{\text{termes turbulents}}$$

The Reynolds equations thus differ from the Navier-Stokes equations by the addition of these “turbulence” terms. They are referred to as *RANS (Reynolds-averaged Navier-Stokes) equations*.

Reynolds stress and turbulence viscosity

It can be shown that the continuity for an incompressible fluid

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

satisfies for the mean flow (to be shown in an exercise):

$$\frac{\partial \overline{u}}{\partial x} + \frac{\partial \overline{v}}{\partial y} + \frac{\partial \overline{w}}{\partial z} = 0$$

Based on the definitions of \overline{u} and u' we obtain:

$$\frac{\partial \overline{u}}{\partial x} = \frac{1}{T} \int_0^T \frac{\partial u}{\partial x} dt = \frac{\partial}{\partial x} \left[\frac{1}{T} \int_0^T u dt \right] = \frac{\partial}{\partial x} \left[\frac{1}{T} \int_0^T (\overline{u} + u') dt \right] = \frac{\partial \overline{u}}{\partial x}$$

By subtraction ($u' = u - \overline{u}$) we obtain

$$\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{\partial w'}{\partial z} = 0$$

The turbulent terms can therefore be written in the following form, without changing their value (selon Ox):

$$\overline{u' \frac{\partial u'}{\partial x}} + \overline{v' \frac{\partial u'}{\partial y}} + \overline{w' \frac{\partial u'}{\partial z}} + \overline{u' \left(\frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} + \frac{\partial w'}{\partial z} \right)} = \frac{\partial \overline{u'u'}}{\partial x} + \frac{\partial \overline{u'v'}}{\partial y} + \frac{\partial \overline{u'w'}}{\partial z}$$

and the Reynolds equation for the u component becomes:

$$\frac{\partial \overline{u}}{\partial t} + \overline{u} \frac{\partial \overline{u}}{\partial x} + \overline{v} \frac{\partial \overline{u}}{\partial y} + \overline{w} \frac{\partial \overline{u}}{\partial z} = -\frac{1}{\rho_0} \frac{\partial \overline{P}}{\partial x} + f_v + \nu \left[\frac{\partial^2 \overline{u}}{\partial x^2} + \frac{\partial^2 \overline{u}}{\partial y^2} + \frac{\partial^2 \overline{u}}{\partial z^2} \right] - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z}$$

The last three terms are called the Reynolds stress terms.

J.V. Boussinesq (1842 – 1929) then proposed to link these Reynolds stress terms to the components of the mean velocity gradient as:

$$\overline{u'u'} = -A_x \frac{\partial \bar{u}}{\partial x} ; \quad \overline{u'v'} = -A_y \frac{\partial \bar{u}}{\partial y} ; \quad \overline{u'w'} = -A_z \frac{\partial \bar{u}}{\partial z} ;$$

where A are the so-called turbulent exchange coefficients (A for *Austausch* = exchange in German) or *eddy viscosity*.

If we neglect the spatial variations of these coefficients, the turbulent terms take an identical form to the molecular friction terms, e.g., for the x component:

$$\frac{d\bar{u}}{dt} = -\frac{1}{\rho_o} \frac{\partial \bar{P}}{\partial x} + f\bar{v} + \nu \left[\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} + \frac{\partial^2 \bar{u}}{\partial z^2} \right] + A \left[\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} + \frac{\partial^2 \bar{u}}{\partial z^2} \right]$$

and since the latter are much smaller (in the ocean, 10^{-5} to 10^{-6} in the vertical, 10^{-10} to 10^{-12} in the horizontal) they are neglected in what follows.

On the other hand, care must be taken with these simplifications, because, unlike ν , the A 's are not properties of the fluid but of the flow. They vary from place to place and depend on the chosen “smoothing” scale. In oceanography, a distinction is made between horizontal and vertical coefficients.

Now we need to find a way to calculate the value of these turbulent coefficients.

Horizontal coefficients

In the equations of motion we can introduce the following substitutions:

$$\begin{aligned} -\frac{\partial(\overline{u'u'})}{\partial x} - \frac{\partial(\overline{u'v'})}{\partial y} &= A_x \frac{\partial^2 \bar{u}}{\partial x^2} + A_y \frac{\partial^2 \bar{u}}{\partial y^2} = A_h \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} \right) \\ -\frac{\partial(\overline{v'u'})}{\partial x} - \frac{\partial(\overline{v'v'})}{\partial y} &= A_x \frac{\partial^2 \bar{v}}{\partial x^2} + A_y \frac{\partial^2 \bar{v}}{\partial y^2} = A_h \left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} \right) \end{aligned}$$

with $A_x = A_y = A_h$ as there is no reason to consider horizontal mixing by turbulence differently in the two horizontal directions (hypothesis of horizontal isotropy).

The POM model in the literature, the coefficient is often denoted using A_M , where M stands for *momentum*) uses the formulation by Smagorinski [1963] based on velocity gradients:

$$A_h = C \Delta x \Delta y \frac{1}{2} |\nabla \vec{u} + (\nabla \vec{u})^T| \quad \text{with} \quad |\nabla \vec{u} + (\nabla \vec{u})^T| = \left[\left(\frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right]^{1/2}$$

The coefficient C generally ranges from 0.10 to 0.20, but if the grid is sufficiently fine, it can also be zero. The advantages of the Smagorinski formulation is that C is dimensionless and A_h decreases with increasing resolution and remains small if the velocity gradients are small.

In the literature on the SYMPHONIE model, the horizontal eddy viscosity is often written using the symbol ν_h . From a physics point of view, it does not depend on the properties of the fluid but on the type of flow and, thus, from a modelling point of view, on the mesh size of the model grid. It is often set to be constant, about $5 \text{ m}^2 \text{ s}^{-1}$ for a mesh size of $1 \text{ km} \times 1 \text{ km}$.



The **Prandtl number** is a dimensionless number, named after the German physicist Ludwig Prandtl, defined as the ratio of momentum diffusivity (or kinematic viscosity) to thermal diffusivity. It provides a measure of the efficiency of diffusive transport through the velocity and thermal boundary layers.

$$\text{Pr} = \frac{\nu}{\alpha} = \frac{\text{momentum diffusivity}}{\text{thermal diffusivity}} = \frac{\mu/\rho}{k/(c_p\rho)} = \frac{c_p\mu}{k}$$

where:

- ν : momentum diffusivity (kinematic viscosity), $\nu = \mu/\rho$, (SI units: m²/s)
- α : thermal diffusivity, $\alpha = k/(\rho c_p)$, (SI units: m²/s)
- μ : dynamic viscosity, (SI units: Pa s = N s/m²)
- k : thermal conductivity, (SI units: W/(m·K))
- c_p : specific heat, (SI units: J/(kg·K))
- ρ : density, (SI units: kg/m³).

The **Schmidt number** is the salinity equivalent of Pr (or for any given property), so we have to replace the thermal diffusivity with that of salt, etc.

https://en.wikipedia.org/wiki/Prandtl_number

Recently, a new numerical advection-diffusion scheme has been implemented for this model [Hu et al., 2009] using a coefficient that changes with direction, mesh size, and time:

$$\tilde{A}_x = \delta |u| \frac{\Delta x}{2} \quad \text{et} \quad \tilde{A}_y = \delta |v| \frac{\Delta y}{2}$$

In the ROMS model, it is possible to chose between the Smagorinski scheme and a constant coefficient (via the option *Laplacian horizontal mixing of momentum*). For the latter, if the horizontal mesh size is variable, the viscosity gets multiplied by an additional factor that accounts for the mesh size using a harmonic or bi-harmonic formulation [Holland et al. (1998); Webb et al. (1998); Griffies et al. (2000ab); Lee et al. (2002)].

For more details, see the advection schemes below (Chapter 3).

For the temperature and salinity equations we obtain in a similar manner:

$$\begin{aligned} -\frac{\partial(\overline{T'u'})}{\partial x} - \frac{\partial(\overline{T'v'})}{\partial y} &= A_h^T \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = \frac{A_h}{\text{Pr}_{\text{urb}}} \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \\ -\frac{\partial(\overline{S'u'})}{\partial x} - \frac{\partial(\overline{S'v'})}{\partial y} &= A_h^S \left(\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \right) = \frac{A_h}{\text{Sc}_{\text{urb}}} \left(\frac{\partial^2 S}{\partial x^2} + \frac{\partial^2 S}{\partial y^2} \right) \end{aligned}$$

given that the eddy diffusivity for temperature (salinity) can be expressed as the ratio of the eddy viscosity to the turbulent Prandtl (Schmidt) number.

Note: Ocean circulation models always simulate turbulent flows, so you have to replace the kinematic viscosity with the turbulent one, thermal diffusivity with eddy diffusivity of temperature, etc. and use the correct terminology of a turbulent Prandtl (Schmidt) number.

Finally, ocean circulation models do not usually account for possible differences between the eddy diffusivities for temperature and salt and one simply uses an eddy diffusivity for tracers (mass):

$$A_h^T = A_h^S = A_h^{tr}$$

The horizontal eddy diffusivity is of the same order of magnitude as the horizontal eddy viscosity. Mellor [POM user's guide, 2004] suggested to use a small turbulent Prandtl number of about 0.2, in order to avoid an excessive diffusion due to the discretisation using sigma coordinates.

Vertical coefficients

In the equations of motion and mass conservation we can introduce the following substitutions:

$$-\frac{\partial}{\partial z}(\overline{u'w'}, \overline{v'w'}) = A_z \frac{\partial^2(u, v)}{\partial z^2}$$

$$-\frac{\partial}{\partial z}(\overline{T'w'}, \overline{S'w'}) = A_z^{tr} \frac{\partial^2(T, S)}{\partial z^2} \quad \text{with } tr = T, S$$

A vast number of publications exists on ways to determine the vertical eddy diffusivity, this being one of the major problems in ocean (and atmospheric) modelling. We can summarize the different approaches that we will discuss below as follows:

Approach	Reference paper	Model
Kinetic energy	Gaspar et al. (1990)	SYMPHONIE
	Mellor and Yamada (1974)	POM, ROMS (vs. Rutgers)
K-profile	Pacanowski and Philander (1981)	ROMS
	Large et al. (1994)	ROMS

First, let us take a look at the turbulence closure models based on calculating the turbulent kinetic energy using a dedicated prognostic equation.

SYMPHONIE model: Gaspar et al. (1990)

In the SYMPHONIE code, in a manner similar to what we have seen for the horizontal, the eddy diffusivity A_z^{tr} for temperature (salt) is given by the ratio between the vertical eddy viscosity, A_z , and the turbulent Prandtl (Schmidt) number:

$$-\frac{\partial}{\partial z}(\overline{T'w'}, \overline{S'w'}) = A_z^{tr} \frac{\partial^2(T, S)}{\partial z^2} = \frac{A_z}{(Pr_{turb}, Sc_{turb})} \frac{\partial^2(T, S)}{\partial z^2}$$

What remains to be found is the value of the vertical eddy viscosity.

NB: in the literature on the SYMPHONIE model the eddy viscosity and diffusivity are often written using the symbols ν_t and Γ_t , respectively; the kinetic energy is denoted with K and the empirical coefficient with C_k .

Turbulence acts at spatial scales much smaller than the mesh sizes used in models. Turbulence viscosity is linked to the turbulent kinetic energy, Q , via:

$$A_z = C_Q L Q^{1/2}$$

C_Q is an empirical constant; L is the mixing length which is a characteristic length scale related to the size of the eddies creating the turbulent flow. Q can be expressed as a function of the turbulent velocity components

$$Q = \frac{1}{2} (u'^2 + v'^2 + w'^2)$$

and can be calculated from the following prognostic equation:

$$\frac{\partial Q}{\partial t} + \vec{v} \cdot \vec{\nabla} Q = A_z \underbrace{\left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right]}_{(1)} + \underbrace{\frac{g}{\rho_0} A_z \frac{\partial \rho}{\partial z}}_{(2)} + \underbrace{\frac{\partial}{\partial z} \left(A_z \frac{\partial Q}{\partial z} \right)}_{(3)} + \underbrace{\frac{\partial}{\partial x} \left(A_h \frac{\partial Q}{\partial x} \right) + \frac{\partial}{\partial y} \left(A_h \frac{\partial Q}{\partial y} \right)}_{(4)} - \varepsilon$$

In this equation, the production of turbulent kinetic energy by velocity shear is represented by (1), the production of buoyancy by (2), the redistribution of energy in the vertical by eddy diffusion is represented by (3) and in the horizontal by (4).

The dissipation of turbulent kinetic energy, ε , is parametrised as a function of the dissipation length L_ε :

$$\varepsilon = \frac{C_\varepsilon Q^{3/2}}{L_\varepsilon}$$

The constants C_ε and C_Q were determined by Gaspar et al. [1990] as: $C_\varepsilon = 0.7$ and $C_Q = 0.1$.

To determine the mixing and dissipation lengths, the SYMPHONIE model uses the approach by Bougeault and Lacarrère [1989] which is interesting because their formulation takes the local fluid stability into consideration and thus allows a better representation of the turbulent exchanges, especially at the level of the thermocline.

The lengths L and L_ε are usually explained in the following fashion: let a fluid particle be located at level z . It possesses as kinetic energy the mean turbulent kinetic energy at this level. It can therefore be displaced upward or downward as long as this kinetic energy remains larger than any change in potential energy associated with such a vertical displacement. This displacement thus has certain length scales for the possible up- and downward displacement, written as L_{up} and L_{down} . These lengths can be calculated from

$$Q(z) = \frac{g}{\rho_0} \int_z^{z+L_{up}} (\rho(z) - \rho(z')) dz' \quad \text{and} \quad Q(z) = \frac{g}{\rho_0} \int_{z-L_{down}}^z (\rho(z) - \rho(z')) dz'$$

and then we can define

$$L = \min(L_{up}, L_{down}) \quad \text{and} \quad L_\varepsilon = \sqrt{L_{up} \cdot L_{down}}$$

POM model: Mellor-Yamada (1974)

The POM model uses another type of approach that is still based on the idea of solving a prognostic equation for the turbulent kinetic energy. A second equation is used for the mixing length. Below, we have written both equations in an x - y - σ coordinate system (Mellor, 2004, *POM user's guide*).

NB: In the literature on the POM model, the vertical eddy diffusivity and viscosity are often written using the letter K : K_M for the diffusivity of momentum, K_H for temperature (*heat*) and salinity, K_q for the turbulent kinetic energy.

$$\begin{aligned} \frac{\partial Q^2 D}{\partial t} + \frac{\partial U Q^2 D}{\partial x} + \frac{\partial V Q^2 D}{\partial y} + \frac{\partial \omega Q^2}{\partial \sigma} &= \\ &= \frac{\partial}{\partial \sigma} \left[\frac{A_z^Q}{D} \frac{\partial Q^2}{\partial \sigma} \right] + \frac{2A_z}{D} \left[\left(\frac{\partial U}{\partial \sigma} \right)^2 + \left(\frac{\partial V}{\partial \sigma} \right)^2 \right] + \frac{2g}{\rho_0} A_z^r \frac{\partial \tilde{\rho}}{\partial \sigma} - \frac{2DQ^3}{B_1 L} + F_{Q^2} \end{aligned}$$

$$\frac{\partial Q^2 L D}{\partial t} + \frac{\partial U Q^2 L D}{\partial x} + \frac{\partial V Q^2 L D}{\partial y} + \frac{\partial \omega Q^2 L}{\partial \sigma} =$$

$$= \frac{\partial}{\partial \sigma} \left[\frac{A_z^Q}{D} \frac{\partial Q^2 L}{\partial \sigma} \right] + E_1 L \left(\frac{A_z}{D} \left[\left(\frac{\partial U}{\partial \sigma} \right)^2 + \left(\frac{\partial V}{\partial \sigma} \right)^2 \right] + E_3 \frac{g}{\rho_o} A_z^{tr} \frac{\partial \tilde{\rho}}{\partial \sigma} \right) - \frac{D Q^3}{B_1} \tilde{W} + F_{Q^2 L}$$

where $F_\Phi \equiv \frac{\partial}{\partial x} \left(H A_h^{tr} \frac{\partial \Phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(H A_h^{tr} \frac{\partial \Phi}{\partial y} \right)$ with $\Phi = Q^2$ ou $Q^2 L$

The vertical diffusivities A_z, A_z^{tr}, A_z^Q are defined as

$$A_z = Q L S_M$$

$$A_z^{tr} = Q L S_H$$

$$A_z^Q = Q L (0.41 S_H) \quad \text{!?!?!SH in pom2k, SM in pom98 !?!?!} \quad \text{☺}$$

and the coefficients S_M and S_H are calculated using the gradient Richardson number G_H

$$S_H [1 - (3A_2 B_2 + 18A_1 A_2) G_H] = A_2 [1 - 6A_1 / B_1]$$

$$S_M [1 - 9A_1 A_2 G_H] - S_H [(18A_1^2 + 9A_1 A_2) G_H] = A_1 [1 - 3C_1 - A_1 / B_1]$$

with

$$G_H = \frac{L^2 g}{Q^2 \rho_o} \left[\frac{\partial \rho}{\partial z} - \frac{1}{C_s^2} \frac{\partial p}{\partial z} \right]$$

Empirically, Mellor and Yamada [1974] determined the values for the five constants:

$$(A_1, B_1, A_2, B_2, C_1) = (0.98, 16.6, 0.74, 10.1, 0.08) .$$

C_s^2 is the sound speed squared, and the hydrostatic approximation is used to find

$$\frac{\partial p}{\partial z} = -\rho_o g .$$

Kantha and Clayson [1994] criticised the Mellor-Yamada type closure schemes, as they underestimate the entrainment due to convection and thus the amount of mixing across stable density gradients.

The ROMS model, being a more recent development, allows the user to choose between different closure schemes: either the Mellor and Yamada [1974] or two schemes [Pacanowski and Philander, 1981 or Large et al., 1994] that employ a so-called *K-profile* type parametrization. This name is derived from the fact that the eddy diffusivity and viscosity are often written using the letter K. The main difference between this K-profile approach and previously mentioned schemes is that the former, instead of solving a prognostic equation for turbulent kinetic energy, assumes a certain vertical profile for the vertical eddy diffusivity and viscosity. The goal is to find the most general (i.e., *non-local*) profile possible.

ROMS model: Pacanowski and Philander (1981)

NB: In this paper the eddy viscosity and diffusivity are written using ν and κ , respectively.

Measurements by Crawford and Osborn (1979) and Osborn and Bilodean (1980) suggest that mixing processes are greatly influenced by current shear. Empirical studies (Robinson, 1966; Jones, 1973) found the following dependencies of eddy viscosity and diffusivity:

$$A_z = \frac{\nu_o}{(1 + \alpha Ri)^n} + \nu_b \quad \text{et} \quad A_z^{tr} = \frac{A_z}{(1 + \alpha Ri)} + \kappa_b$$

with the Richardson number $Ri = \frac{\beta g T(z)}{u^2(z) + v^2(z)}$

and the thermal expansion coefficient $\beta \approx 8.75 \times 10^{-6} (T + 9)$

The Richardson number and Brunt-Väisälä frequency

$$Ri = \frac{N^2}{\left(\frac{\partial u}{\partial z}\right)^2}$$

Ri compares the static stability given by the Brunt-Väisälä frequency, N , to the dynamic instability represented by the vertical velocity gradient. If $Ri > 0.25$ the water column is stable; if $Ri < 0.25$ a vertical velocity gradient increases turbulence.

If the fluid is incompressible

$$N^2 = -\frac{g}{\rho} \frac{\partial \rho}{\partial z}$$

and a certain stratification is called “stable” if $N^2 > 0$.

The oscillatory behaviour of stratified fluids can be understood by thinking of a fluid particle whose density increases with depth. When it is displaced vertically out of its equilibrium position, its density is lower or higher than that of the ambient fluid and a restoring force, gravity or Archimedes’ thrust, tries to return the fluid particle back to its equilibrium position. Typically due to the fluid particle’s inertia, it overshoots the equilibrium depth and oscillates about the equilibrium position.

For instance, the Brunt-Väisälä frequency, among other parameters, controls the height and spacing between the streets of [cumulus clouds](#) or the [lenticular clouds](#) in the lee of mountains, and also between the ridges in oceanic [swell](#).

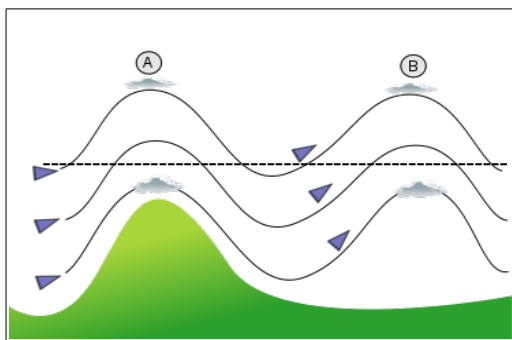
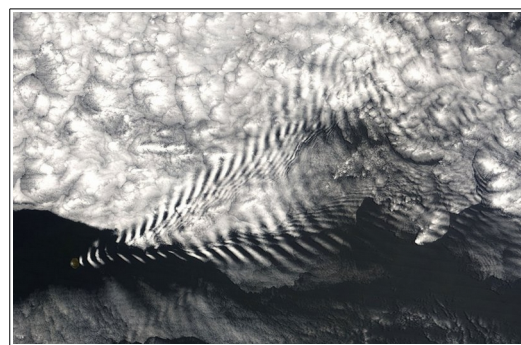


Diagram showing [orographic lift](#) caused by a mountain and the resulting lee wave. Clouds form in the wave crests if the raised air becomes saturated (points A and B) [image [Dake](#)].

https://en.wikipedia.org/wiki/Orographic_lift



A (gravity) lee wave in the downwind (leeward) zone of the Île Amsterdam, in the far southern Indian Ocean. The spacing between wave crests is governed by the Brunt-Väisälä frequency [Image [NASA](#), MODIS satellite].

https://en.wikipedia.org/wiki/Brunt-Väisälä_frequency

In the above equations, T is the temperature in $^{\circ}\text{C}$, u and v are the horizontal velocity components, g the acceleration due to gravity, ν_b and κ_b are the background viscosity and diffusivity, respectively, and ν_o, α , and n are calibration parameters.

Following a series of simulations of the equatorial ocean testing different wind regimes (east and west wind), Pacanowski and Philander found the best match between model and *in situ* measurements when using the following values for the calibration parameters:

$$\begin{aligned} \nu_b &= 1 \text{ cm}^2 \text{ s}^{-1} \\ \kappa_b &= 0.1 \text{ cm}^2 \text{ s}^{-1} \\ n &= 2 \\ \alpha &= 5 \\ \nu_o &= O(50 \text{ cm}^2 \text{ s}^{-1}) \end{aligned}$$

At the same time, they pointed out that it was much more difficult to obtain the same results with the Mellor and Durbin (1975) scheme, hence concluding that their model was more efficient from a numerical point of view.

ROMS model: KPP, Large et al. (1994)

The heart of this vertical mixing scheme is the parametrization of the surface boundary layer, the so-called KPP or K-profile parametrization. Large et al. (1994) also proposed a parametrization for the mixing in the ocean's interior and specified the conditions for coupling the surface and interior regimes. Due to the importance of mixing in the surface boundary layer this mixing scheme is usually referred to as the *KPP model*.

<p>COUCHE LIMITE ATMOSPHERIQUE</p> <p>$H = \text{HAUTEUR DE LA COUCHE}$</p> <p>In meteorology, the <i>Atmospheric or Planetary Boundary Layer (ABL or BPL)</i> is the part of the atmosphere between the Earth's surface, where friction retards air flow, and the free atmosphere where friction becomes negligible. Its thickness varies between 0.5 and 3 km depending on its stability and surface roughness.</p>	<p>https://en.wikipedia.org/wiki/Boundary_layer</p> <p>Its mean thickness is about 1500 m. Its theoretical treatment assumes that it is made up of two rather different layers:</p> <ul style="list-style-type: none"> - the Ekman layer (after the Swedish physicist Vagn Walfrid Ekman) in which the wind results from a balance between the pressure gradient, the Coriolis force which results from the rotation of the Earth, and a friction that gradually decreases with increasing altitude. The wind speed and direction at the top of this layer correspond approximately to those of the geostrophic wind, as we approach the Earth's surface the wind speed gradually weakens and its direction turns toward the lowest pressure; - the surface layer or atmospheric turbulent boundary layer is in direct contact with the surface and has a thickness of typically no more than 10% of the entire ABL thickness. The wind speed in this layer is due to convection resulting from temperature differences and from topographic effects. The flow is turbulent. One also speaks of a rough sub-layer very close to the surface whose thickness varies from a few centimetres to tens of metres depending on the surface roughness. The velocity tends to zero as we approach the surface.
---	--

<p>OCEANIC BOUNDARY LAYER</p> <p>The upper region of the ocean typically exhibits of a surface mixed layer with a thickness of a few to several hundreds meters (<i>OBL, Ocean Boundary Layer</i>). This mixed layer is a key component in studies of climate, biological productivity and marine pollution. It is the link between the atmosphere and deep ocean and directly affects the air-sea exchange of heat, momentum and gases. Moreover, turbulent flows in the mixed layer affect biological productivity by controlling both the supply of nutrients to the upper sunlit layer and the light exposure of phytoplankton.</p> <p>Several processes contribute to turbulent mixing in the mixed layer. Thermal convection can be generated by the ocean losing heat through longwave back radiation or evaporative cooling.</p>	<p>http://hpl.umces.edu/ocean/sml_main.htm</p> <p>The shear generated in wind-driven currents can produce Kelvin-Helmholtz billows. The interaction between surface waves and wind-driven shear current also produces Langmuir circulation, consisting of counter-rotating vortices with their axes aligned roughly in the wind direction.</p>
---	--

NB: In Large et al. (1994), the letter X is used to represent the time-averaged quantities (u, v, T, S) while x are the turbulent variations (u', v', t', s'). K_x are the vertical eddy diffusivities for the different properties in the boundary layer, while ν_x refer to the fluid's interior. The authors also adopted a notation that is based on a vertical coordinate, d , which measures the distance from the boundary layer, thus within the atmospheric boundary layer we have $d = z$ and in the OBL $d = -z$, with $d=0$ at the air-sea interface. We have adopted this convention below and for simplicity limited the equations to the u velocity component.

A generalised (*non-local*) model using a vertical profile of the vertical eddy diffusivities was developed by Troen and Mahrt [1986] for large scale models of the atmosphere. The two main ingredients of a non-local model are:

- a transport term is added to the RHS of the closure equation; e.g., for the u velocity component:

$$\overline{u'w'} = -A_z \frac{\partial u}{\partial z} + ??$$

- the eddy diffusivity in the boundary layer depends on forces acting at the surface and on the thickness, h , of the boundary layer; again using the example of the u velocity component we get:

$$A_z \equiv A_z([\overline{w'u'}]_{surf}, h)$$

The theory by Prandtl [1925, *non-linear mixing length*] proposes that the characteristic size of vortices in the boundary layer, called mixing length, L_m , is proportional to the standard deviation of the movement of the fluid particles.

Heisenburg [1938] hypothesized that $A_z \propto L_m^{4/3}$ and also $A_z \propto \nu^{1/3}$ where ν is the molecular diffusion

Blackadar [1962] reformulated the last relationship in terms of vertical shear to arrive at the expression

$$A_z = L_m^2 [(\partial_z U)^2 + (\partial_z V)^2]^{1/2} \quad \text{with} \quad L_m = \kappa d / (1 + \kappa d / L_o)$$

where L_o is an asymptotic value of the empirical L_m , obtained from observations in the atmosphere.

In the ocean, this equation yields good results only when we are close to the surface where L_m approaches $\kappa d = 0.4$ (Von Karman's constant) in agreement with the similarity theory for turbulent and molecular viscosity.

This formulation corresponds to the preceding definition because it contains a non-local transport term and A_z depends on the local shear and the BL thickness.

There are many other parametrisations of L_m ; e.g., in models where the grid size, Δ , is smaller than the mixing length, the mixing-length theory is used to parametrise the sub-grid scale processes. For this reason, it is commonly assumed that $L_m \propto \Delta$ as with Smagorinsky [1963] who developed his equation (see above) for a general circulation model of a 2-layer atmosphere. *LES (Large Eddy Simulation)* models use the same kind of formulation for the vertical dimension and always used idealised domains to compute a large part of the turbulent flow field in 3D [Lilly, 1967; Deardorff, 1970].

Large et al. [1994] adapted the KPP model by Troen and Mahrt [1986] for an OBL (*oceanic boundary layer*) model.

The problem of determining turbulent velocity components and turbulent fluxes of active and passive scalars is resolved by adding a term γ ; for the example of the u velocity component this yields:

$$\overline{w'u'}(d) = -A_z \left(\frac{\partial u}{\partial z} - \gamma \right)$$

In practise, the idea is to first prescribe an external forcing and then calculate the layer thickness, h , as well as the vertical viscosity and diffusivity profiles A_z and A_z^tr and the non-local transport γ .

The viscosity/diffusivity profile is the product of h , the OBL thickness, w_x , the turbulent velocity scale, and the shape function G (dimensionless):

$$A_z(\sigma) = h w_x(\sigma) G(\sigma)$$

w_x and G depend on the depth, given here by $\sigma = d/h$, which varies between 0 and 1 in the BL.

At every depth, A_z is directly proportional to h , which represents the fact that deeper boundary layers can contain larger and more energetic eddies.

The OBL thickness depends on the forcing at the surface, on the buoyancy B due to stratification (one often uses the Richardson number to express the layer stability), and on the velocity profiles V :

$$h \equiv h([\overline{w'u'}]_{surf}, B(d), V(d))$$

The term γ is zero except for the scalars (T, S, etc.) when the forcing conditions create instabilities, and when there are certain parametrisations, e.g., for salinity:

$$\gamma_s = C^* \frac{[\overline{w'u'}]_{surf}}{w^* h}$$

• Large et al.: OCEANIC VERTICAL MIXING

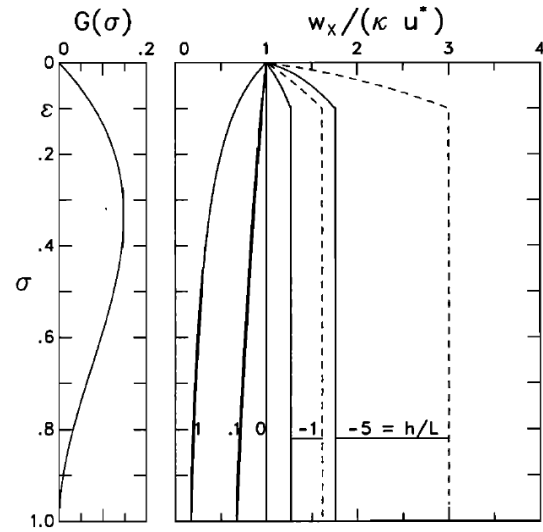


Figure 2. (left) Vertical profile of the shape function $G(\sigma)$, where $\sigma = d/h$, in the special case of $G(1) = \partial_\sigma G(1) = 0$. (right) Vertical profiles of the normalized turbulent velocity scale, $w_x(\sigma)/(\kappa u^*)$, for the cases of $h/L = 1, 0.1, 0, -1$, and -5 . In unstable conditions, $w_s(\sigma)$ (dashed traces) is greater than $w_m(\sigma)$ (solid traces) at all depths, but for stable forcing $h/L \geq 0$, the two velocity scales are equal at all depths.

In the ocean's interior, we do no longer have a non-local transport term and, to emphasize that the processes are different, a lower case "a" is used to represent the mixing coefficients:

$$\overline{w'u'}(d) = -a_z \left(\frac{\partial u}{\partial z} \right)$$

Mixing is considered the sum of three processes:

- instabilities as indicated by the local Richardson number linked to vertical shear;
- the breaking of internal waves;
- double diffusion.

Each process is parametrised in terms of the local vertical diffusivity:

$$a_z(d) = a_z^{inst}(d) + a_z^{wave}(d) + a_z^{double}(d)$$

Other processes can be added or certain simplifications can be applied if there is a clearly dominant process and the others can be neglected.

These last two equations also give the boundary conditions $a(h)$ and $w_x(1)$ for the shape function G . At the interface between the OBL and ocean's interior, G takes the following form

$$G(1) = \frac{a_z(h)}{h w_x(1)}$$

The diffusivity which results from instabilities linked to vertical velocity shear is expressed as the *local gradient Richardson number*

$$Ri_g = \frac{N^2}{(\partial_z U)^2 + (\partial_z V)^2}$$

The amount of mixing caused by breaking internal waves can be calculated from the *Garrett-Munk internal wave model* [Gregg, 1989] and Osborn's [1980] equations for diffusion and energy dissipation. However, the diffusivity due to internal wave breaking appears to be rather small and one often prescribes fixed values:

$$a_z^{wave} = 1.0 \times 10^{-4} \text{ m}^2 \text{ s}^{-1} \quad \text{for momentum and}$$

$$a_z^{wave} = 0.1 \times 10^{-4} \text{ m}^2 \text{ s}^{-1} \quad \text{for other tracers.}$$

Lastly, double diffusion occurs when the density gradient is stable but either the vertical salinity (*salt fingering*) or temperature gradients (*diffusive convection*) are unstable in their contribution to the overall density gradient [Turner, 1973]. Another cause of *double diffusion* in the ocean is the fact that the molecular diffusion of T is higher than for S, which reduces the potential energy in the vertical profile with a preferential mixing of one component (T or S) which acts destabilising.

Here, we also have functions describing the variation of a_z^{double} based on the ratio

$$R_p = \frac{\alpha \partial_z T}{\beta \partial_z S}$$

• Large et al.: OCEANIC VERTICAL MIXING

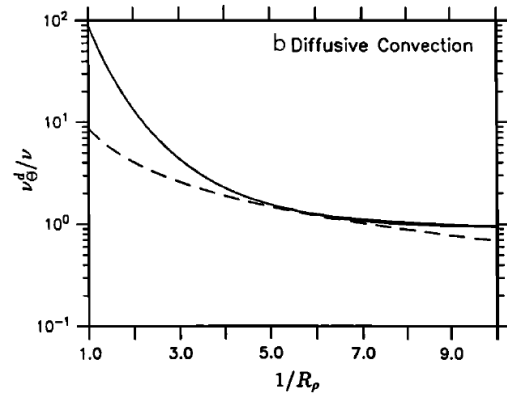
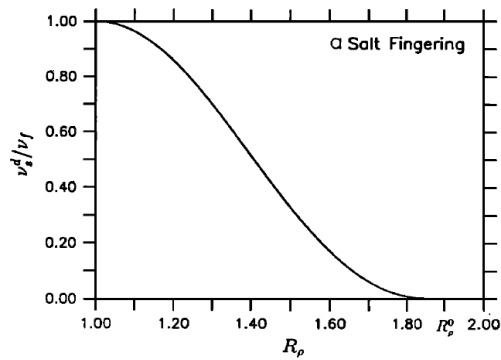


Figure 4. (a) Interior diffusivity for double-diffusive mixing due to salt fingering, $v_s^d = v_m^d$ (equation (31)), normalized by its maximum, $v_f = 10 \times 10^{-4} \text{ m}^2 \text{ s}^{-1}$, as a function of the density ratio R_p (equation (30)). (b) Interior diffusivity of potential temperature for double-diffusive mixing due to diffusive convection, v_θ^d , normalized by the molecular diffusivity $\nu (= 1.5 \times 10^{-6} \text{ m}^2 \text{ s}^{-1})$, as a function of R_p^{-1} . The solid trace represents (32); the dashed trace, (33).

Boundary conditions

Boundary conditions refer to the ocean surface, the sea bed, as well as the lateral boundaries, i.e., with the coast (closed boundaries) or the ocean (open boundaries).

At the free surface

As for the current velocity and kinetic energy, the free-surface boundary conditions at $z=\eta$ in the SYMPHONIE model are given by:

$$A_z \left(\frac{\partial u}{\partial z}, \frac{\partial v}{\partial z} \right) = (\tau_{sx}, \tau_{sy}) \quad \text{for } z=\eta$$

$$Q = \frac{\|\vec{\tau}_s\|}{\rho_o \sqrt{C_\epsilon \cdot C_Q}} \quad \text{for } z=\eta$$

The surface tension, $\vec{\tau}_s = (\tau_{sx}, \tau_{sy})$, is calculated from the air density, $\rho_{air} = 1.266 \text{ kg m}^{-3}$, the wind velocity 10m above the ground, \vec{V}_{wind} (e.g., in the Gulf of Lion, the ALADIN model by Météo-France is used to calculate this velocity), and the grad coefficient C_v :

$$\vec{\tau}_s = \rho_{air} C_v \|\vec{V}_{wind}\| \vec{V}_{wind}.$$

For the POM model:

$$\frac{K_M}{D} \left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma} \right) = - (\langle wu(0) \rangle, \langle wv(0) \rangle), \quad \sigma \rightarrow 0 \quad \sigma = \frac{z-\eta}{H+\eta}$$

For temperature, the boundary condition at the surface results from the heat budget:

$$A_z^T \frac{\partial T}{\partial z} = H_s + H_e + H_l + (1 - Tr) H_c \quad \text{for } z=\eta$$

where:

H_s is the sensible heat flux

H_e is the latent heat flux

H_l is the longwave radiative heat flux

H_c is the shortwave radiative heat flux

Tr is the transmissivity.

REMINDER: **Sensible heat** is heat exchanged by a body or thermodynamic system in which the exchange of heat changes the temperature of the body or system, and some macroscopic variables, but leaves unchanged certain other macroscopic variables, such as volume or pressure (e.g., no phase transition). The term is used in contrast to **latent heat**, which is the amount of heat exchanged that is hidden, meaning it occurs without a change of temperature (e.g., during a phase change).

The fluxes H_c and H_s are calculated from the sea surface temperature provided by the model and from atmospheric parameters provided by an atmospheric model (air temperature and humidity at 2 m altitude, temperature and pressure at sea level, and wind at 10 m altitude). This calculation is based on a so-called “*bulk*” parametrization of air-sea fluxes (see Appendix 1, thesis by C. Dufau-Julliard). The long- and shortwave radiative heat fluxes are directly supplied by the atmospheric models.

For salinity, the surface boundary condition is based on the following mass balance:

$$A_z^S \frac{\partial S}{\partial z} = (E - P) S \quad \text{for } z=\eta$$

where P and E are the precipitation and evaporation rates, respectively, written in terms of the latent heat flux and the latent heat of condensation.

For the POM model:

$$\frac{K_H}{D} \left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma} \right) = - \langle w\theta(0) \rangle, \quad \sigma \rightarrow 0 \quad \sigma = \frac{z - \eta}{H + \eta}$$

At the sea bed

The boundary conditions at $z = -h$ in the SYMPHONIE model are:

$$\begin{aligned} A_z \left(\frac{\partial u}{\partial z}, \frac{\partial v}{\partial z} \right) &= (\tau_{bx}, \tau_{by}) & \text{for } z = -H \\ Q &= \frac{\|\vec{\tau}_b\|}{\rho_0 \sqrt{C_\varepsilon \cdot C_Q}} & \text{for } z = -H \\ \left(\frac{\partial T}{\partial z}, \frac{\partial S}{\partial z} \right) &= 0 & \text{for } z = -H \end{aligned}$$

The bottom shear stress $\vec{\tau}_b = (\tau_{bx}, \tau_{by})$ depends on the square of the horizontal velocity near that boundary, V_b , taken at the first model grid level z_1 above the bed:

$$\vec{\tau}_b = \rho_0 C_d \|\vec{V}_b\| \vec{V}_b$$

C_d is again the drag coefficient and can be expressed in terms of the roughness length and Von Karman's constant, κ , using a logarithmic law:

$$C_d = \left(\frac{\kappa}{\log(Z_1/Z_0)} \right)^2$$

The roughness length is used to characterize the amount of bottom friction and is usually set to 0.01m. The preceding relations assume that the vertical velocity profile is logarithmic near the sea bed. Z_1 is the distance between the first sigma level and the sea bed.

For the POM model:

$$\frac{K_M}{D} \left(\frac{\partial U}{\partial \sigma}, \frac{\partial V}{\partial \sigma} \right) = C_z [U^2 + V^2]^{1/2} (U, V), \quad \sigma \rightarrow -1$$

where

$$C_z = \text{MAX} \left[\frac{\kappa^2}{[\ln\{(1 + \sigma_{kb-1})H/z_o\}]^2}, 0.0025 \right] \quad (14e)$$

$\kappa = 0.4$ is the von Karman constant and z_o is the roughness parameter. Equations (14c,d,e) can be derived by matching the numerical solution to the "law of the wall". Numerically, they are applied to the first grid points nearest the bottom. Where the bottom is not well resolved, $(1 + \sigma_{kb-1})H/z_o$ is large and (14e) reverts to a constant drag coefficient, 0.0025. The boundary conditions on (5) and (6) are

$$\begin{aligned} \frac{K_H}{D} \left(\frac{\partial T}{\partial \sigma}, \frac{\partial S}{\partial \sigma} \right) &= 0, \quad \sigma \rightarrow -1 \\ (q^2(-1), q^2 \ell(-1)) &= (B_1^{2/3} u_\tau^2(-1), 0) \end{aligned} \quad (16c,d)$$

where B_1 is one of the turbulence closure constants and u_τ is the friction velocity at the top or bottom as denoted in (16a) and (16c). In pom97.f and later versions, (16a) has been replaced by $q^2 \ell(\sigma_1) = q^2(\sigma_1) \kappa D \sigma_1$ where σ_1 is the value of σ corresponding to $k=1$, it is believed that this averts some numerical noise in some applications.

Freshwater input

If there are rivers in the model domain, they need to be taken into account. For example, in our simulation of the Gulf of Lion we account for the (Great and Little) Rhône, as well as the Aude, Hérault, and Orb rivers. They are modelled as channels, few model grid points in length in either of the horizontal directions. At the source points of these rivers, the horizontal velocity component in the channel is a function of the flow, D , and the channel width, L , and depth h_c :

$$U_{canal} = \frac{D}{L \cdot h_c}$$

The vertical current profile in the channel has a logarithmic shape.

A fixed temperature (e.g., 8°C) is prescribed at the river's source point that remains constant through the simulation. As the river water is fresh, the source point salinity is set to zero.

For large rivers such as the Amazon or Mississippi rivers, some modellers may also add conditions about the conservation of mass in the model domain that removes any excess waters via the open boundaries.

At the lateral boundaries

At closed lateral boundaries, a simple zero-flux condition is applied as there cannot be any cross-boundary fluxes. In contrast, open boundaries represent a major challenge and we will discuss them in more detail later on once we have introduced the different types of model grids.

Parametrisation of atmospheric fluxes in the SYMPHONIE model

Text from the thesis by C. Dufau-Julliand, Appendix 1

Annexes

Les formulations aérodynamiques de type « bulk » simplifient l'estimation des flux atmosphériques à l'interface océan-atmosphère. Ces relations s'appuient à la fois sur des coefficients déterminés expérimentalement et des informations météorologiques de type « bulk ». Les flux de surface (τ la tension du vent, H le flux de chaleur sensible et E le flux de chaleur latente) s'expriment de la manière suivante :

$$\begin{aligned}\tau/\rho &= u_*^2 = -\langle u'w' \rangle = C_D U^2 \\ H/\rho C_p &= u_* T_* = \langle w'T' \rangle = C_H U (T - T_0) \\ E/\rho L_v &= u_* q_* = \langle w'q' \rangle = C_E U (q - q_0)\end{aligned}$$

U , T et q sont respectivement la vitesse du vent, la température potentielle de l'air et l'humidité relative de l'air à l'altitude z . T_0 et q_0 se rapportent à la température et à l'humidité de l'air à proximité de la surface de l'océan. ρ est la densité de l'air, C_p la chaleur spécifique de l'air à pression constante et L_v la chaleur latente de vaporisation. u_* , T_* et q_* constituent des échelles caractéristiques pour la vitesse, la température et l'humidité de l'air. C_D est le coefficient de traînée, C_H le nombre de Stanton et C_E le nombre de Dalton.

Ces trois derniers coefficients, aussi appelés « coefficients bulk », dépendent de l'altitude, de la stratification de l'air et de l'état de vague à la surface de l'océan. La détermination de ces coefficients est le cœur de la paramétrisation des flux atmosphériques. Pour cela, on calcule les équivalents de ces coefficients dans le cas d'une stratification neutre C_{DN} , C_{HN} et C_{EN} .

La relation entre C_D et C_{DN} s'obtient en considérant que pour la même vitesse de vent U le profil de la vitesse peut s'écrire, dans le cas général

$$U = \frac{u_*}{k} \left(\ln(z/z_0) - \psi_m \right) = C_D^{-1/2} u_*$$

où k est la constante de Karman ($k=0.4$), z_0 est la longueur de rugosité et ψ_m est une fonction de stratification dépendant de l'altitude z .

et à la neutralité

$$U = \frac{u_{*N}}{k} \left(\ln(z/z_{0N}) \right) = C_{DN}^{-1/2} u_{*N}$$

où z_{0N} est la longueur de rugosité à la neutralité.

En combinant les deux équations précédentes, on obtient la relation entre les deux coefficients :

$$C_D = \left(C_{DN}^{-1/2} - k^{-1} \psi_m + k^{-1} \ln(z_{0N}/z_0) \right)^{-2}$$

Dans cette équation, on peut substituer les longueurs de rugosité par les échelles caractéristiques de la vitesse du vent, grâce à la relation de Charnock $z_0 = a u_*^2/g$, a étant le coefficient de Charnock, pris constant.

On obtient finalement la relation suivante entre les deux coefficients de traînée :

$$C_D = \left(C_{DN}^{-1/2} - k^{-1} \psi_m + \underbrace{k^{-1} \ln(C_{DN}/C_D)}_{\substack{\text{négligeable quand la} \\ \text{stratification diffère} \\ \text{peu de la neutralité}}} \right)^{-2}$$



Annexes

Un raisonnement identique à partir des profils de température et d'humidité près de la surface permet d'exprimer le nombre de Stanton et celui de Dalton en fonction de leur équivalent à la neutralité.

$$C_H = C_{HN} (1 - k^{-1} \psi_m C_{DN}^{1/2} - k^{-1} \psi_h C_{HN} C_{DN}^{-1/2} + k^{-2} C_{HN} \psi_m \psi_h)^{-1}$$

$$C_E = C_{EN} (1 - k^{-1} \psi_m C_{DN}^{1/2} - k^{-1} \psi_q C_{EN} C_{DN}^{-1/2} + k^{-2} C_{EN} \psi_m \psi_q)^{-1}$$

Les coefficients à la neutralité sont donnés par

$$C_{DN} = 10^{-3} (0.61 + 0.063U)$$

$$C_{HN} = \begin{cases} 1.2 \cdot 10^{-3} & \text{pour } z/L < 0 \\ 0.7 \cdot 10^{-3} & \text{pour } z/L > 0 \end{cases}$$

$$C_{EN} = 1.2 \cdot 10^{-3}$$

Les fonctions de stratification ψ dépendent de l'altitude z et de la stabilité de l'atmosphère calculée à partir de la longueur de Monin-Obukhov L :

$$z/L = - \frac{g \cdot k \cdot z \cdot \langle w' T_v' \rangle}{T_v u_*^3}$$

où $\langle w' T_v' \rangle$ est le flux turbulent de la température virtuelle tel que

$$\langle w' T_v' \rangle = \langle w' T' \rangle + 0.61 T \langle w' q' \rangle \quad \text{avec} \quad \begin{cases} H = \rho C_p \langle w' T' \rangle \\ E = \rho L_v \langle w' q' \rangle \end{cases}$$

Ces fonctions de stratifications ont été paramétrées par différents auteurs et s'expriment de la manière suivante:

$$\psi_m = \begin{cases} 2 \ln \left(\frac{1 + \varphi_m^{-1}}{2} \right) + \ln \left(\frac{1 + \varphi_m^{-2}}{2} \right) - 2 \tan^{-1} (\varphi_m^{-1}) + \pi/2 & \text{pour } z/L < 0 \\ 1 - \varphi_m & \text{pour } z/L > 0 \end{cases}$$

$$\psi_q = \psi_h = \begin{cases} 2 \ln \left(\frac{1 + \varphi_h^{-1}}{2} \right) & \text{pour } z/L < 0 \\ 1 - \varphi_h & \text{pour } z/L > 0 \end{cases}$$

Les fonctions ϕ sont les fonctions universelles suivantes, établies par *Paulson* [1970], *Businger et al.* [1971] :

$$\varphi_m = \begin{cases} (1 - \alpha z/L)^{-1/4} & \text{pour } z/L < 0 \\ 1 + \beta z/L & \text{pour } z/L > 0 \end{cases}$$

$$\varphi_q = \varphi_h = \begin{cases} \varphi_m^{1/2} & \text{pour } z/L < 0 \\ \varphi_m & \text{pour } z/L > 0 \end{cases}$$

α et β sont pris égaux à 16 et 7 [*Large and Pond*, 1982].

Annexes

On remarque donc que les fonctions de stratification dépendent des flux de chaleur et de vitesse. Il faut donc calculer ces flux avec une méthode itérative. L'état initial est choisi par exemple à la neutralité et l'état final lorsque les échelles caractéristiques u , T et q convergent.

Par ailleurs les coefficients C_{DN} , C_{HN} et C_{EN} sont établis pour 10 mètres d'altitude tandis que les paramètres atmosphériques tels que la température et l'humidité de l'air sont fournis par Météo-France à une altitude de 2 mètres. Il est donc nécessaire à chaque itération de calculer les valeurs à 10 mètres à partir de celles à 2 mètres d'altitude en combinant les équations décrivant les profils de température et d'humidité. Pour la température, cela donne par exemple

$$T_{10} = T_2 + T_*/k \left(\ln(10/2) - \psi_h(10/L) + \phi_h(2/L) \right)$$

Remarque: la température T_0 provient à chaque itération du calcul de la température océanique au niveau sigma situé le plus proche de la surface libre de l'océan.

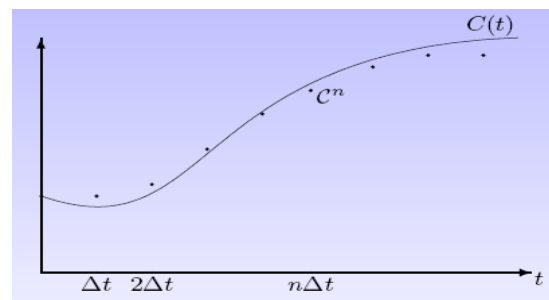
Une présentation détaillée de ces formulations est fournie par *Geenaert* [1990].

Model grids

Reminder:

the derivative $f'(x)$ of a function $f(x)$ is defined as

$$f'(x) = \frac{df}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$



Notation:

Discretisation $t = n\Delta t, n=1,2,\dots$

Real solution $C(t) = C(n\Delta t)$

At discrete times t_n we approximate the solution using $C(n\Delta t) = C^n$
(NB: Be careful not to confuse with power!)

Sometimes, when C is used without index, we imply C^n

If there is no possible confusion use $Q^n = Q(n\Delta t, C^n)$

Properties of a numerical finite-difference scheme:

Consistency
A finite difference scheme is consistent (or compatible) if the solution of the discretized equations approaches the solution of the analytical equations as the time step, Δt , and spatial resolution, Δx , tend to zero.

Convergence
A finite difference scheme is convergent if the solution of the discretized equation converges to the solution of the analytical equation.

Stability
A finite difference scheme is considered stable if small perturbations are not systematically amplified.

Error classification

- **Modelling errors:** differences between the physical system and the exact solution to our mathematical model:
 - Formulation of the law governing the system (turbulence closure or parametrisations of unresolved effects)
 - Poorly constrained parameters (grazing rate, sedimentation rate)
 - Treatment of open boundaries
 - Initial conditions and boundary conditions
 - Using simplified geometries of the problem

• **Discretisation errors:** differences between the exact solution of our mathematical model and the exact numerical solution of the discretized equations:

- Truncation errors in the numerical scheme used to solve the mathematical equations
- Errors in the conservation of masse, concentrations, or energy

• **Convergence errors:** differences between the exact numerical solution of the discretized equations and the solution obtained using iterative methods:

- errors in iterations used to find zeros in non-linear functions
- errors in Gauss-Seidel iterations
- errors in obtaining iterative solutions of the discrete Poisson equations

• **Rounding errors:** errors due to rounding by the computer:

- representation of real numbers and arithmetic operations on an electronic machine/compiler
- errors obtained when calculating the scalar product of large vectors
- errors obtained when subtracting very similar numbers from one another

We can thus define a “good” model as a model where

Modelling errors >> Discretization errors >> Convergence errors >> Rounding errors

Each type of error must be significantly smaller than the expected error in the solution without this error. However, one should not go overboard and employ a very precise numerical method if the model itself is imprecise.

While the above errors are often unavoidable, some errors are unacceptable:

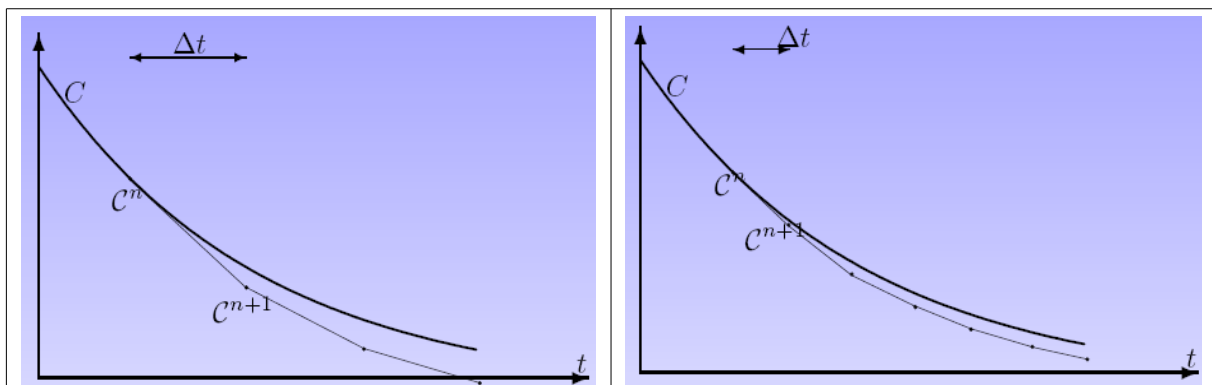
- Choosing a model that is ill-suited
- Errors in the data
- Using a discretisation that is too coarse
- Programming errors
- Using values in incompatible units

In general: anything that prevents a good model from functioning properly.

Explicit methods: the forward Euler method

$$\frac{dC}{dt} = Q(t, C) \text{ becomes } \frac{C(t+\Delta t) - C(t)}{\Delta t} = Q(n\Delta t, C^n) \text{ or } C^{n+1} = C^n + \Delta t Q^n$$

Explicit method (*forward differencing*): C^{n+1} follows directly (and exclusively) from C^n



Implicit schemes

While the Euler scheme is simple, it often becomes quickly unstable as it is based on extrapolating to

the future.

To resolve this problem, we can evaluate Q at a point in time in the future, either by already having more updated information or by shifting by a point in time

$$\frac{C(t+\Delta t) - C(t)}{\Delta t} = Q(t+\Delta t, C(t+\Delta t)) \quad \frac{C(t) - C(t-\Delta t)}{\Delta t} = Q(t, C(t))$$

Practical problem: this discretisation gives rise to an equation that is non-linear in C_{n+1}

$$C^{n+1} = C^n + \Delta t Q(t_{n+1}, C^{n+1}) = f(C^{n+1})$$

Except for very particular expressions of Q , this equation will need to be solved numerically to locate where it is zero. One such numerical method is the Newton-Raphson method, which is a root-finding algorithm that produces successively better approximations to the roots (or zeroes) of a real-valued function f . In general, the convergence is quadratic: as the method converges on the root, the difference between the root and the approximation is squared (the number of accurate digits roughly doubles) at each step (see https://en.wikipedia.org/wiki/Newton%27s_method).

Leapfrog scheme

Instead of introducing intermediate points, why not reuse some points? Example: the *Leapfrog* scheme allows us to formulate a second order scheme that is easy to implement:

$$\frac{C(t+\Delta t) - C(t-\Delta t)}{2\Delta t} = Q(t, C(t)) \quad C^{n+1} = C^{n-1} + 2\Delta t Q(t_n, C^n)$$

Parametric filtering

It is used to smooth the solution and reduce digital noise

$$\tilde{C}^n = C^n + \gamma [C^{n-1} - 2C^n + C^{n+1}]$$

Thus approach requires storing C^n and \tilde{C}^n and does not modify C^n .

A special case is the **Asselin filter**, which uses the parametric filtering and replaces C^{n-1} with the filtered value:

$$C^{n+1} = \tilde{C}^{n-1} + 2\Delta t Q(t_n, C^n)$$

$$\tilde{C}^n = C^n + \gamma [\tilde{C}^{n-1} - 2C^n + C^{n+1}]$$

In practise, this amounts to replacing C^n by \tilde{C}^n .

However, it also filters out physically correct waves, especially when the variations are rapid.

Example: SYMPHONIE model (Text taken from the thesis by C. Dufau-Julliard)

3.4.2 La discrétisation temporelle

La discrétisation temporelle se fait à l'aide d'un schéma « Leapfrog » ou Saute-Mouton explicite. Le calcul de la valeur des variables au temps $(t+1)$ nécessite la connaissance des valeurs aux temps (t) et $(t-1)$:

$$u^{t+1} = u^{t-1} + 2\Delta t f^n(u^t) \quad (3.23)$$

La solution numérique d'une équation discrétisée sur un schéma temporel de ce type comporte deux type de modes : un mode physique qui constitue la solution vraie et un mode numérique qui est en fait une source d'erreur. En effet, les solutions aux temps pairs et impairs peuvent diverger. Pour réduire ces erreurs numériques, il est fait appel à chaque pas de temps à un filtre d'Asselin diffusif [Asselin, 1972] qui consiste à calculer une valeur corrigée :

$$u_t^i = u^t + \frac{\alpha}{2}(u^{t+1} - 2u^t + u^{t-1}) \quad (3.24)$$

Ce filtre agit donc comme une force de rappel empêchant la divergence des solutions à des pas de temps consécutifs (pair et impair). α est le coefficient du filtre d'Asselin.

Staggered grids

- The finite-volume approach with fluxes across the interfaces requires defining (and possibly

predicting) the diffusion coefficients, A , at those interfaces.

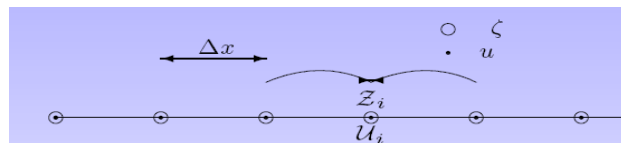
- A finite-volume approach for advection requires defining the (normal) velocities at the interfaces
 - In a 3D model, the velocities and eddy diffusivities are usually predicted by the model
- It is therefore advisable not to define the variables at the exact same locations.

Let us consider the equations of a highly simplified hydrodynamic model (2 variables, 1 spatial dimension):

$$\frac{\partial \zeta}{\partial t} + h \frac{\partial u}{\partial x} = 0 \quad \text{et} \quad \frac{\partial u}{\partial t} = -g \frac{\partial \zeta}{\partial x}$$

and the associated numerical model is formulated on a classical and a staggered grid.

On a classical grid:



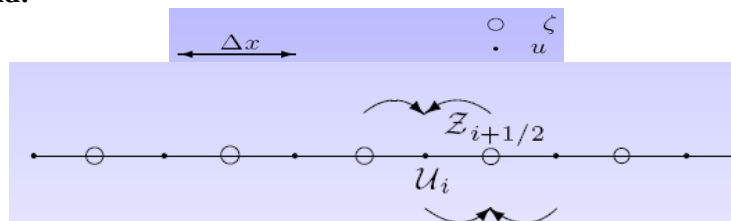
If we consider a classical second-order derivative with a distance between the points of $2\Delta x$ we have what we call a “forward-backward direct discretization”:

$$\frac{z_i^{n+1} - z_i^n}{\Delta t} + h \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -g \frac{z_{i+1}^{n+1} - z_{i-1}^{n+1}}{2\Delta x}$$

In this case, it can be shown that the approximations are of the order $O(\Delta t)$ and $O(\Delta x^2)$.

On a staggered grid:



$$\frac{z_{i+1/2}^{n+1} - z_{i+1/2}^n}{\Delta t} + h \frac{u_{i+1}^n - u_i^n}{\Delta x} = 0$$

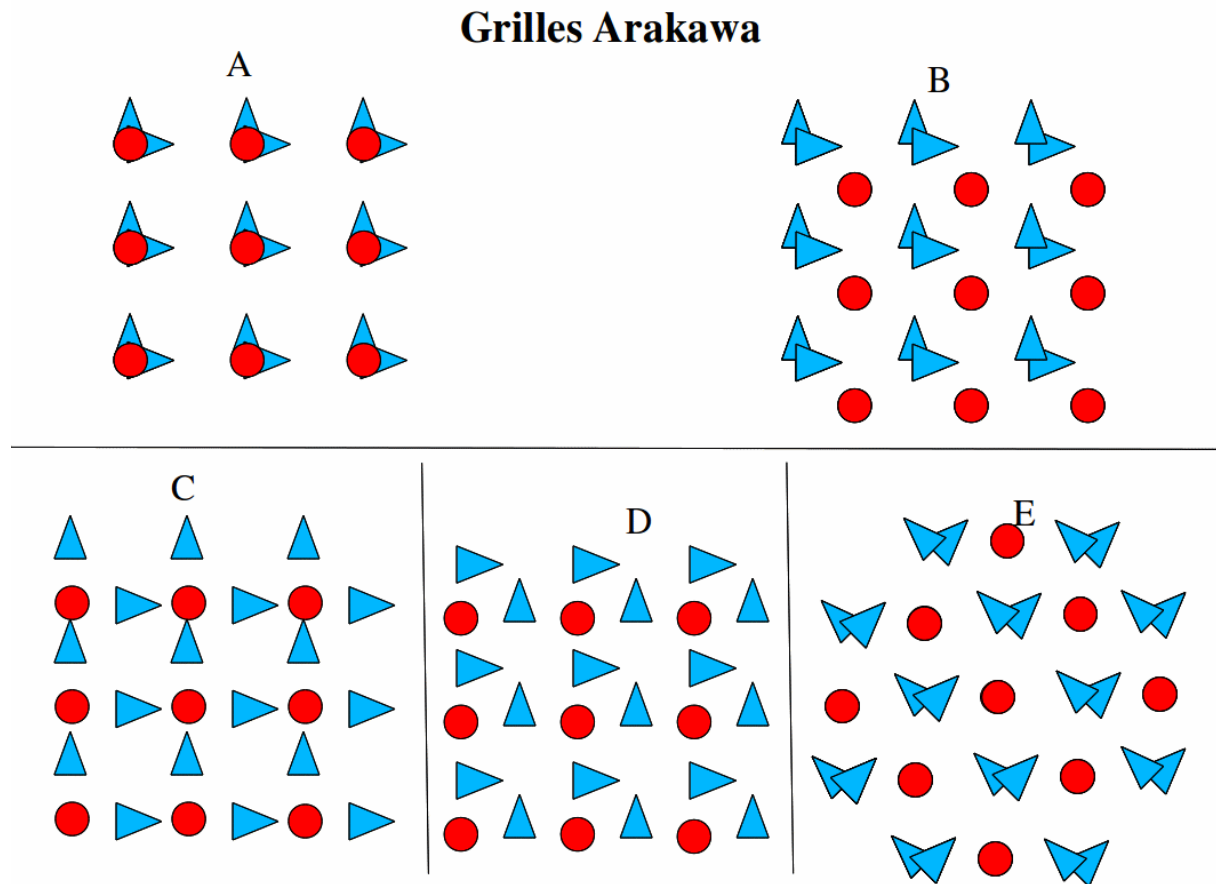
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -g \frac{z_{i+1/2}^{n+1} - z_{i-1/2}^{n+1}}{\Delta x}$$

This scheme is still of the “forward-backward direct discretization” type and the approximations are still of the order $O(\Delta t)$ and $O(\Delta x^2)$, but the distance between points is Δx and no longer $2\Delta x$, i.e., we have increased the precision.

From a practical point of view, in the numerical code we range $z_{i-1/2}$ in ZETA(i).

Arakawa classification of horizontal grids

Arakawa and Lamb (1977) proposed a classification consisting of 5 different possibilities of arranging scalar and vector variables on a numerical grid.



The Arakawa B and E grids have the two velocity components in the same locations and are therefore called “*semi-staggered*”. The E grid can be considered a B grid that has been rotated by 45°.

Choosing an Arakawa grid type

In practice, only the Arakawa grids B, C, and E are used. It can be shown that the B grid is better for models with a low spatial resolution and short time steps, while the C grid performs better with higher spatial resolutions.

In oceanography:

B-grid: MOM (*GFDL Modular Ocean Model*) <http://www.gfdl.noaa.gov/ocean-model>

C-grid: ROMS, POM, SYMPHONIE, OPA http://www.nemo-ocean.eu/content/download/259/1665/file/Doc_OPA8.1.pdf

E-grid: HOPE (*Hamburg Ocean Primitive Equation Model*) <http://meteora.ucsd.edu/~pierce/elnino/hope.html>

Discretizations

Arakawa A

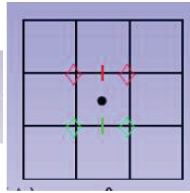
$$\frac{z_{i,j}^{n+1} - z_{i,j}^n}{\Delta t} + h \left[\frac{U_{i+1,j}^n - U_{i-1,j}^n}{2\Delta x} + \frac{V_{i,j+1}^n - V_{i,j-1}^n}{2\Delta y} \right] = 0$$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} - fV_{i,j}^{n+s} = -g \frac{z_{i+1,j}^{n+1} - z_{i-1,j}^{n+1}}{2\Delta x}$$

$$\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta t} + fU_{i,j}^{n+1-s} = -g \frac{z_{i,j+1}^{n+1} - z_{i,j-1}^{n+1}}{2\Delta y}$$

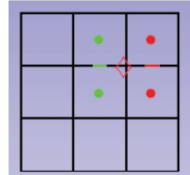
Arakawa B

$$\frac{\partial \zeta}{\partial t} + h \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) = 0$$



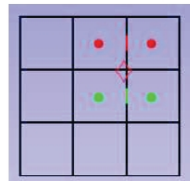
$$\frac{z_{i,j}^{n+1} - z_{i,j}^n}{\Delta t} + h \left(\frac{(U_{i+1/2,j+1/2}^n + U_{i+1/2,j-1/2}^n) - (U_{i-1/2,j+1/2}^n + U_{i-1/2,j-1/2}^n)}{2\Delta x} \right) + h \left(\frac{(V_{i-1/2,j+1/2}^n + V_{i+1/2,j+1/2}^n) - (V_{i-1/2,j-1/2}^n + V_{i+1/2,j-1/2}^n)}{2\Delta y} \right)$$

$$\frac{\partial u}{\partial t} = fv - g \frac{\partial \zeta}{\partial x}$$



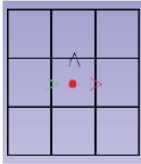
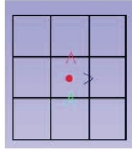
$$\frac{U_{i+1/2,j+1/2}^{n+1} - U_{i+1/2,j+1/2}^n}{\Delta t} = fV_{i+1/2,j+1/2}^{n+s} - g \frac{(z_{i+1,j}^{n+1} + z_{i+1,j+1}^{n+1}) - (z_{i,j}^{n+1} + z_{i,j+1}^{n+1})}{2\Delta x}$$

$$\frac{\partial v}{\partial t} = -fu - g \frac{\partial \zeta}{\partial y}$$

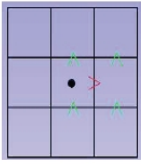
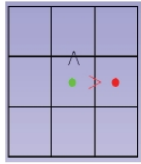


$$\frac{V_{i+1/2,j+1/2}^{n+1} - V_{i+1/2,j+1/2}^n}{\Delta t} = -fU_{i+1/2,j+1/2}^{n+1-s} - g \frac{(z_{i,j+1}^{n+1} + z_{i+1,j+1}^{n+1}) - (z_{i,j}^{n+1} + z_{i+1,j}^{n+1})}{2\Delta y}$$

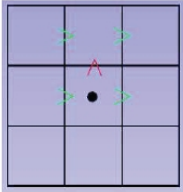
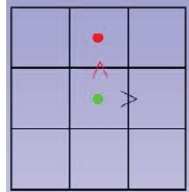
Arakawa C

$$(\delta_t \zeta)_{n_t+1/2, n_x, n_y} + h \left[\underbrace{(\delta_x u)_{n_t, n_x, n_y}} + \underbrace{(\delta_y v)_{n_t, n_x, n_y}} \right] = 0$$

$$(\delta_t u)_{n_t+1/2, n_x+1/2, n_y} - \underbrace{f \bar{v}^{xy}_{n_t+s, n_x+1/2, n_y}} = \underbrace{-g (\delta_x \zeta)_{n_t+1, n_x+1/2, n_y}}$$

$$(\delta_t v)_{n_t+1/2, n_x, n_y+1/2} + \underbrace{f \bar{u}^{xy}_{n_t+1-s, n_x, n_y+1/2}} = \underbrace{-g (\delta_y \zeta)_{n_t+1, n_x, n_y+1/2}}$$

The CFL condition

The spatial and temporal resolution need to meet the Courant-Friedrichs-Levy (CFL) stability (or convergence) condition to prevent that certain processes (current, wave, etc.) advance by more than one grid point in a single time step. The time step Δt and grid size Δx are therefore linked by the speed of propagation, c , of this event according to an equation of the type: $\Delta x > c \Delta t$. More generally, we can first choose the spatial resolution and then calculate the necessary time step

$$\Delta t \leq \frac{1}{c} \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right]^{-\frac{1}{2}}$$

In this way, any increase in spatial resolution will automatically be accompanied by a decrease in time step, which in turn leads to higher computing costs.

In oceanography, the fastest processes one usually models are long gravity waves and it is therefore necessary to calculate their propagation velocity.

The linearized equations for waves propagating in an ocean of constant depth without wind or bottom friction can be derived from the shallow water equations by assuming that the bathymetry is constant (i.e., flat bottom with $h = \text{const}$), the domain is non-rotating ($f = 0$) and that the motion depends only on x , i.e., is one-dimensional.

It is then necessary to solve the system

$$\frac{\partial U}{\partial t} = -gh \frac{\partial \eta}{\partial x} \quad \text{with} \quad U = \int_{-H}^0 u dz \quad \text{and} \quad H = h + \eta$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial U}{\partial x} = 0$$

By differentiation the first equation with respect to x and the second equation with respect to t and then subtracting one from the other we obtain:

$$\frac{\partial^2 \eta}{\partial t^2} - gh \frac{\partial^2 \eta}{\partial x^2} = 0$$

This is a classical wave equation the solution of which is a travelling wave

$$\eta = F(x - ct) \quad \text{which propagates at a speed} \quad c = \sqrt{gh}$$

Hence, the disturbance η propagates without changing its shape at the velocity c .

The temporal resolution of an oceanographic model thus not only depends on the spatial resolution, but also on the bathymetry in the model domain. In general, as a precaution, one evaluates the CFL condition using the maximum depth in the model domain, hence the condition becomes:

$$\Delta t \leq \frac{1}{\sqrt{gh_{max}}} \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right]^{-\frac{1}{2}}$$

Taking gravity waves into account has some advantages:

- takes into account the geostrophic adjustment and surface waves
- direct comparison with/assimilation of altimetry data

but also drawbacks:

- stability condition associated with a fast propagation speed
- parasitic waves due to an inaccurate initialization and/or incorrect boundary conditions as well as digital diffusion associated with the oscillating vertical velocity as the wave propagates, considering that $w \simeq \partial_t \eta$.

To resolve these drawbacks, we can adopt different technical solutions:

- Classical discretization using an adequate Δt (computationally expensive and therefore in practise limited to 2D applications)
- Implicit methods using linear systems that can be resolved (coupling in two directions)
- Time step separation (time splitting): follow the fast processes with an adequate Δt and the other processes with a larger Δt
- A priori elimination of inertia-gravity waves by modifying the equations using the so-called “rigid-lid” approximation

These last two solutions are the most commonly used.

The rigid-lid approach

We calculate the orders of magnitude of the continuity equation in e.p.p. in the open ocean:

$$\frac{\partial \eta}{\partial t} + \nabla_h \vec{V} = 0 \quad \frac{E}{T} = \frac{HU}{L} \quad \frac{1}{T} = \frac{10^3 10^{-1}}{10^6} = 10^{-4}$$

based on this time scale of $T=10^4$ s, or about 10 hours, any change in η in the conservation of mass becomes negligible

$$\frac{\partial \eta}{\partial t} + \nabla_h \vec{V} \simeq \nabla_h \vec{V} = 0 .$$

One assumes that the interface does not move very rapidly (which is the case with gravity waves).

Then, we set $w_{z=0} = 0$ and the unknown becomes the surface pressure.

The *rigid-lid* eliminates any fast processes associated with the movement of the free surface, η , but also the means to determine it directly which is required for the pressure calculation. The equation for η thus turns into a constraint that the transport be non-divergent.

This constraint is satisfied by the following flow equations (for transport):

$$U = -\frac{\partial \psi}{\partial y} \quad V = +\frac{\partial \psi}{\partial x}$$

The difference between two ψ isolines thus immediately gives the water flow between these two lines and instead of having to resolve two equations, one for U and one for V , it suffices to resolve a single equation for ψ .

Advantages

- Fast modes are already filtered out
- Fast processes are eliminated

Drawbacks

- The interpretation of η is ambiguous
- The computational cost grows quickly with increasing resolution
- Difficulties in the function describing the flow field around islands and when adding rivers
- Inability to take atmospheric pressure into account
- Impossibility to take into account the loss of water by evaporation
- Deformation of the propagation of long planetary waves

Mode splitting approach

We keep as a condition

$$w_{z=0} = \frac{\partial \eta}{\partial t} + u \cdot \nabla_h \eta$$

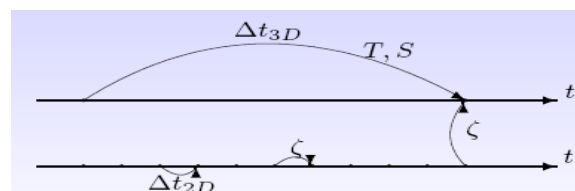
The unknowns in this equation are the position of the free surface and the surface (atmospheric) pressure (neglecting surface tension).

If a system contains processes that operate at very different time scales, the restrictions on the allowed time steps are very different. One can isolate fast processes and model them using a smaller time step. This is the case with external gravity waves in a 3D model.

The CFL condition for this process is

$$\Delta t_E \leq \frac{1}{\sqrt{g h_{max}}} \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right]^{-\frac{1}{2}}$$

and only applies to a 2D process (a so-called external mode with time step Δt_E). We then integrate a 2D model (unknowns η, U, V) within a 3D model (unknowns u, v, T, S, ρ).



Obviously, care must be taken that the vertical integration of the 3D model is consistent with the 2D transport.

In addition, a stability condition needs to be formulated for the internal mode

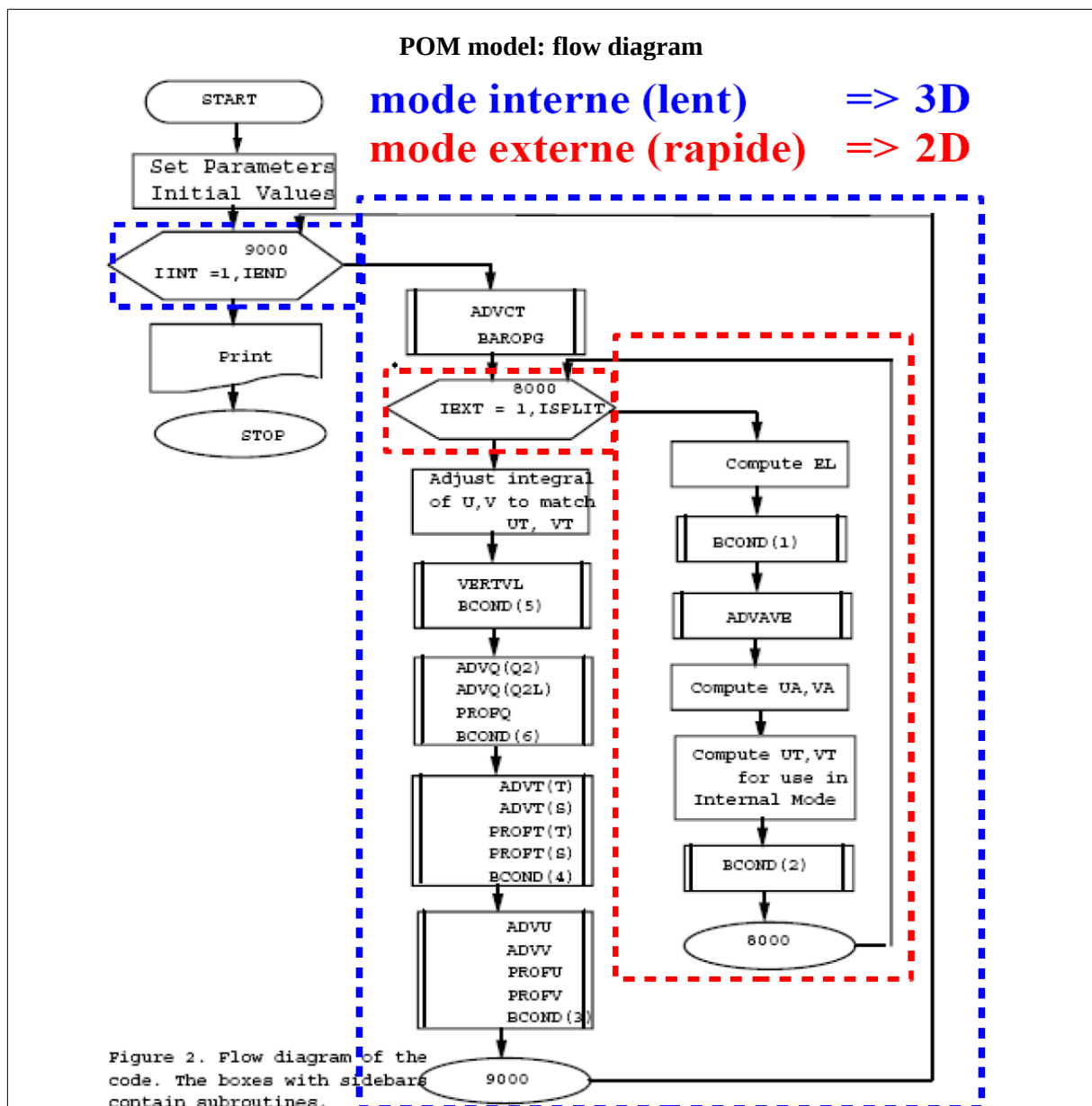
$$\Delta t_I \leq \frac{1}{C_I} \left[\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right]^{-\frac{1}{2}}$$

with $C_I = \sqrt{g'h'} + u_{max}$ where $g' = g \frac{\rho'}{\rho_o}$ is the reduced gravity, experienced by a water parcel sitting in a layer of thickness h' and density ρ' , while u_{max} is the maximum current velocity.

The condition for the internal time step is less restrictive than for the external time step, yielding a typical ratio in these ranges

$$30 < \frac{\Delta t_I}{\Delta t_E} < 80$$

In the ROMS model, to set the time parameters, one first choses the internal time step and then the ratio $\Delta t_I/\Delta t_E$, which is called *NDTFAST* in the file *roms.in*.



SYMPHONIE model

Text taken from the thesis of C. Dufau-Juliland

CHAPITRE 3 - Le modèle numérique SYMPHONIE

3.4.3 La technique de séparation des pas de temps

Les ondes de gravité externes se propagent beaucoup plus rapidement que les ondes de gravité internes. Elles nécessitent donc une résolution temporelle plus fine et conditionnent les critères de stabilité numérique du modèle. Afin de ne pas alourdir les temps de calcul, le traitement de ces ondes (équation (3.6)) s'effectue séparément. Les gradients horizontaux de pression présents dans les équations (3.1) sont décomposés en deux parties liées au mode externe et au mode interne. Suivant la direction x , le gradient de pression s'écrit en effet :

$$\frac{1}{\rho_0} \frac{\partial P}{\partial x} = \frac{1}{\rho_0} \frac{\partial}{\partial x} \left(g \int_z^\eta \rho dz' \right) = \frac{1}{\rho_0} \frac{\partial}{\partial x} \left(g \int_z^\eta \left(\frac{\rho - \rho_0}{\rho'} + \rho_0 \right) dz' \right) = \frac{g}{\rho_0} \frac{\partial}{\partial x} \left(\int_z^\eta \rho' dz' \right) + g \frac{\partial \eta}{\partial x} \quad (3.25)$$

Le terme de pression du mode externe est donné par l'intégration sur la verticale de ce gradient :

$$\Pi_E^x = \underbrace{g \rho_0 H \frac{\partial \eta}{\partial x}}_1 + g \underbrace{\int_{-h}^\eta \left(\frac{\partial}{\partial x} \int_z^\eta \rho' dz' \right)}_2 dz \quad (3.26)$$

Le terme (1) est calculé avec un pas de temps plus court (6 secondes) que le terme (2) qui se calcule avec le même pas de temps que le mode interne (180 secondes)

Le terme de pression du mode interne correspond quand à lui au gradient de pression débarrassé de sa valeur moyennée sur la verticale :

$$\Pi_I^x = \frac{\partial P}{\partial x} - \frac{\Pi_E^x}{H} \quad (3.27)$$

Le mode interne et le mode externe se calculent donc séparément de manière explicite. Cette technique permet de représenter correctement la propagation rapide des ondes externes de gravité avec un pas de temps plus court sans alourdir le temps de calcul global du modèle.

Concernant le mode externe, il faut résoudre simultanément l'équation (3.6) et les équations du mouvement intégrées sur la verticale (3.28) :

$$\begin{aligned} \frac{\partial H \bar{u}}{\partial t} + \frac{\partial}{\partial x} (H \bar{u} \bar{u}) + \frac{\partial}{\partial y} (H \bar{v} \bar{u}) - f H \bar{v} &= -\frac{1}{\rho_0} \Pi_E^x - \int_0^i \left[\frac{\partial H u' u'}{\partial x} \Big|_k + \frac{\partial H v' u'}{\partial y} \Big|_k \right] d\sigma + \frac{\partial}{\partial x} (H v \frac{\partial \bar{u}}{\partial x}) + \frac{\partial}{\partial y} (H v \frac{\partial \bar{u}}{\partial y}) + \frac{v_i}{\rho_0} \left[\frac{\partial u}{\partial z} \right]_0^i \\ \frac{\partial H \bar{v}}{\partial t} + \frac{\partial}{\partial x} (H \bar{u} \bar{v}) + \frac{\partial}{\partial y} (H \bar{v} \bar{v}) + f H \bar{u} &= -\frac{1}{\rho_0} \Pi_E^y - \int_0^i \underbrace{\left[\frac{\partial H u' v'}{\partial x} \Big|_k + \frac{\partial H v' v'}{\partial y} \Big|_k \right]}_{\text{forçage du mode externe par le mode interne}} d\sigma + \frac{\partial}{\partial x} (H v \frac{\partial \bar{v}}{\partial x}) + \frac{\partial}{\partial y} (H v \frac{\partial \bar{v}}{\partial y}) + \frac{v_i}{\rho_0} \left[\frac{\partial v}{\partial z} \right]_0^i \end{aligned}$$

Le mode interne est calculé quant à lui à partir des équations (3.1) en suivant la méthode de *Blumberg and Mellor* [1987]: la partie moyenne du courant est remplacée par le résultat obtenu pour le mode externe. Pour garantir la conservation des quantités scalaires lors des calculs d'advection, cette substitution est réalisée selon une méthode inspirée de *Blumberg and Mellor* [1987]. Le mode externe, résolu avec un pas de temps plus court, conditionne les critères de stabilité numérique du modèle. Une condition CFL (Courant-Friedrichs-Lévy) contraint ainsi le pas de temps du mode externe:

$$dt \leq \frac{1}{2c} \left(\frac{1}{\Delta x^2 + \Delta y^2} \right)^{-1/2} \quad (3.29)$$

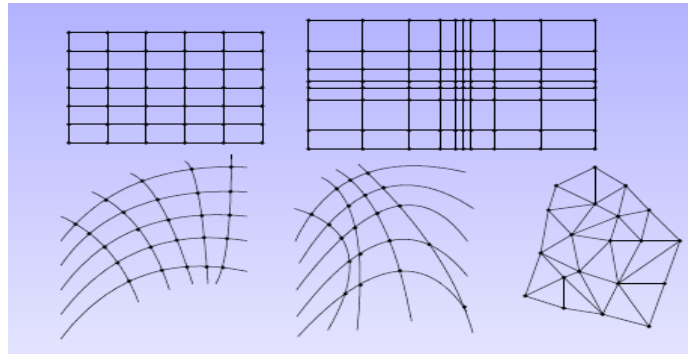
dt , Δx et Δy sont respectivement le pas de temps et d'espace du modèle.

La vitesse de propagation des ondes ($c = \sqrt{gH}$) est celle des ondes de gravité barotropes.

Horizontal coordinates and horizontal grid transformations

Examples of different types of grids:

- Cartesian,
- non-uniform Cartesian,
- curvilinear orthogonal,
- curvilinear non-orthogonal,
- non-structured



Cartesian coordinates.

Mathematical operators take their usual simple form and the numerical discretization is intuitive and quick to implement.

In contrast, it is difficult to adapt such a grid to complex topographies (sometimes a simple rotation can help, see SYMPHONIE NWMed model, Hu et al., 2009) and one has to limit the model domain to a small geographic area.

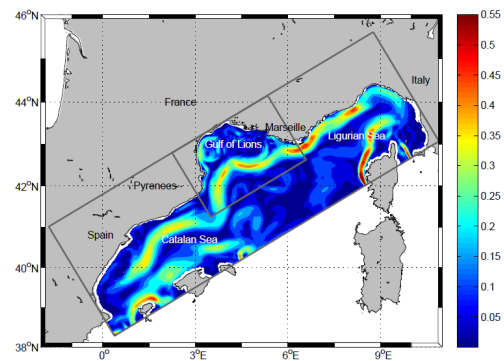
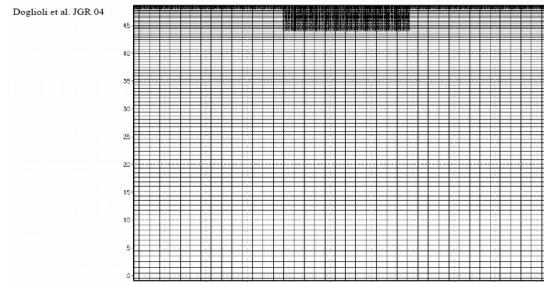
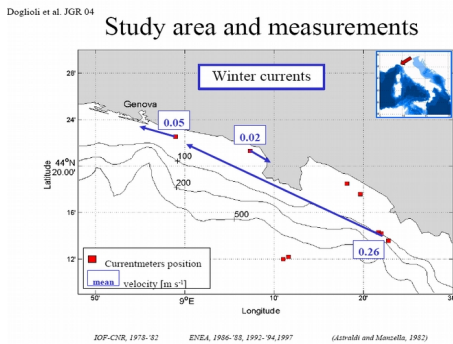


Fig. 2. Model domains. The larger (smaller) rectangle represents the model domain of the 3-km (1-km) resolution. Shaded color represents the intensity of the modelled horizontal current [m s⁻¹] at 20-m depth on July 25, 2001.

Example of a non-uniform Cartesian grid (POM model, Doglioli et al., 2004)



Numerical domain: 202 x 81 grid points ↔ Physical domain: 100 x 50 km
Horizontal resolution: Δx = 500 m 200m < Δy < 1000 m
Vertical resolution: 31 σ-layers

Spherical coordinates

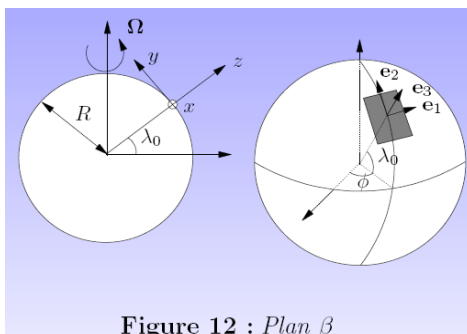


Figure 12 : Plan β

$$\frac{\partial w}{\partial r} + 2\frac{w}{r} + \frac{1}{r \cos \lambda} \frac{\partial v \cos \lambda}{\partial \lambda} + \frac{1}{r \cos \lambda} \frac{\partial u}{\partial \phi} = 0,$$

$$\frac{du}{dt} + \frac{uw}{r} - \frac{uv}{r} \tan \lambda - 2\Omega v \sin \lambda + 2\Omega w \cos \lambda = -\frac{1}{\rho r \cos \lambda} \frac{\partial p}{\partial \phi} + \frac{F_\phi}{\rho}$$

$$\frac{dv}{dt} + \frac{vw}{r} + \frac{u^2}{r} \tan \lambda + 2\Omega u \sin \lambda = -\frac{1}{\rho r} \frac{\partial p}{\partial \lambda} + \frac{F_\lambda}{\rho}$$

$$\frac{dw}{dt} - \frac{u^2 + v^2}{r} - 2\Omega u \cos \lambda = -\frac{1}{\rho} \frac{\partial p}{\partial r} - g + \frac{F_r}{\rho}$$

$$\text{ou } \frac{d}{dt} \equiv \frac{\partial}{\partial t} + \frac{u}{r \cos \lambda} \frac{\partial}{\partial \phi} + \frac{v}{r} \frac{\partial}{\partial \lambda} + w \frac{\partial}{\partial r}$$

As with the Cartesian coordinates, if a uniform grid is used one loses the possibility to adapt the grid

to the geometry or to a specific process. We also encounter a problem at the poles since the spherical coordinates have a mathematical pole at this point. Also, at the poles we find that the lines of longitude are getting closer and closer to one another so that a uniform latitude/longitude grid is a very distorted physical grid in Euclidean space with a much higher resolution near the poles compared to the equator.

Since the depth of the oceans is very small compared to the distance, r , from the centre of the Earth, we generally replace r by the mean radius, a , of the Earth in these equations, Likewise, the factor w/r before $\partial w/\partial r$ can be neglected. We then denote by z the local vertical coordinate measured positively upwards from the sea surface.

We thus obtain the simplified spherical coordinates:

$$\begin{aligned} \frac{\partial w}{\partial z} + \frac{1}{a \cos \lambda} \frac{\partial v \cos \lambda}{\partial \lambda} + \frac{1}{a \cos \lambda} \frac{\partial u}{\partial \phi} &= 0, \\ \frac{du}{dt} + \frac{uw}{a} - \frac{uv}{a} \tan \lambda - 2\Omega v \sin \lambda + 2\Omega w \cos \lambda &= -\frac{1}{\rho a \cos \lambda} \frac{\partial p}{\partial \phi} + \frac{F_\phi}{\rho} \\ \frac{dv}{dt} + \frac{vw}{a} + \frac{u^2}{a} \tan \lambda + 2\Omega u \sin \lambda &= -\frac{1}{\rho_0 a} \frac{\partial p}{\partial \lambda} + \frac{F_\lambda}{\rho} \\ \frac{dw}{dt} - \frac{u^2 + v^2}{a} - 2\Omega u \cos \lambda &= -\frac{1}{\rho_0} \frac{\partial q}{\partial z} - b + \frac{F_r}{\rho} \end{aligned}$$

ou $\frac{d}{dt} \equiv \frac{\partial}{\partial t} + \frac{u}{a \cos \lambda} \frac{\partial}{\partial \phi} + \frac{v}{a} \frac{\partial}{\partial \lambda} + w \frac{\partial}{\partial z}$

In global ocean models, one usually tries to ensure that the poles are on land. While this is a given for the South Pole, as it lies on the Antarctic continent, the North Pole lies in the middle of the Arctic Ocean.

For example, the OPA model distorts the grid in such a way that the two numerical poles comes to lie over Canada and Russia.

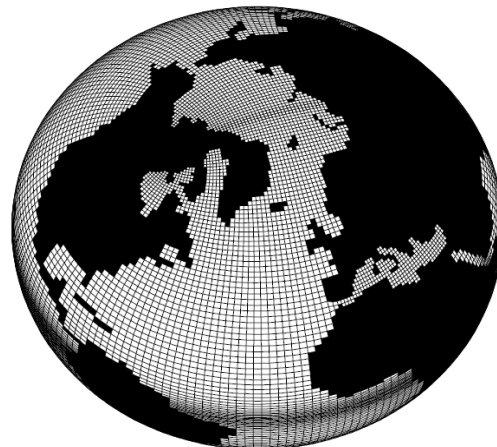
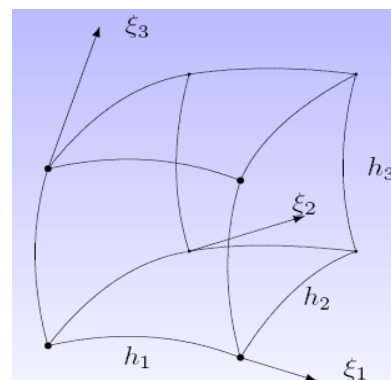


Figure taken from http://www.nemo-ocean.eu/content/download/259/1665/file/Doc_OPA8.1.pdf

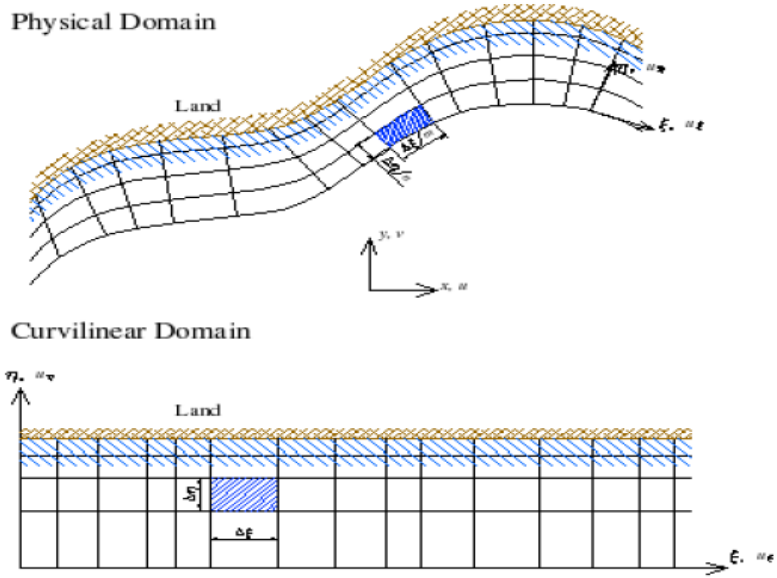
Curvilinear orthogonal coordinates

Transformation of a 3D element

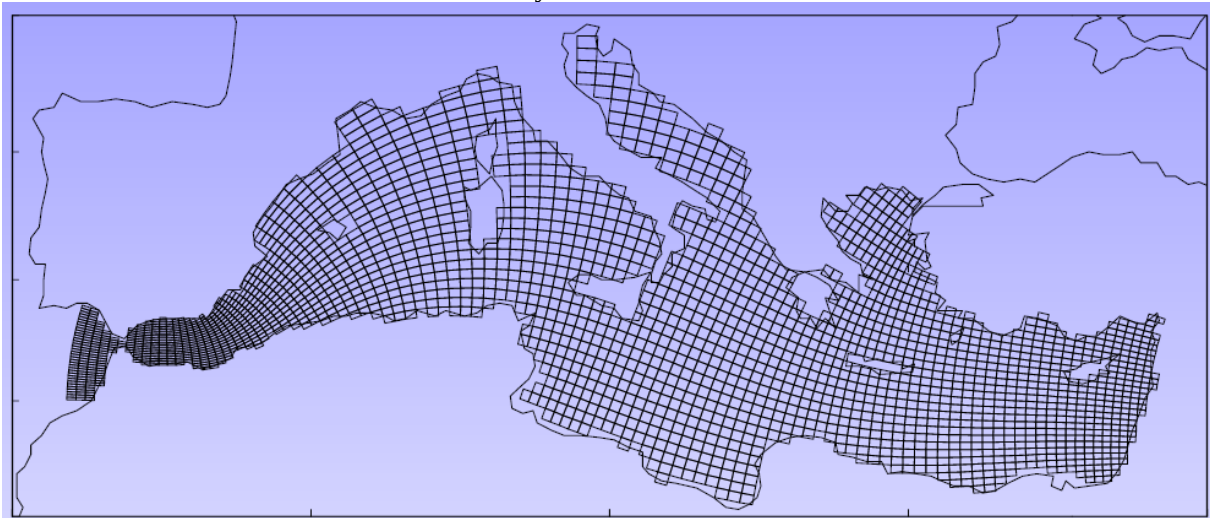
$$\begin{aligned} h_1^2 &= \left(\frac{\partial x}{\partial \xi_1}\right)^2 + \left(\frac{\partial y}{\partial \xi_1}\right)^2 \\ h_2^2 &= \left(\frac{\partial x}{\partial \xi_2}\right)^2 + \left(\frac{\partial y}{\partial \xi_2}\right)^2 \\ h_3 &= 1 \end{aligned}$$



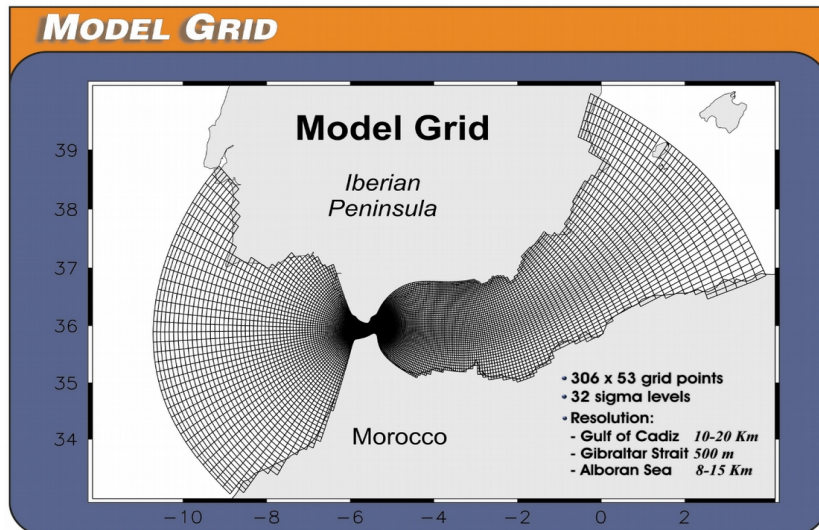
Horizontal curvilinear grid



Numerical model of the Mediterranean Sea by J.-M. Beckers



Sannino et al., POM model



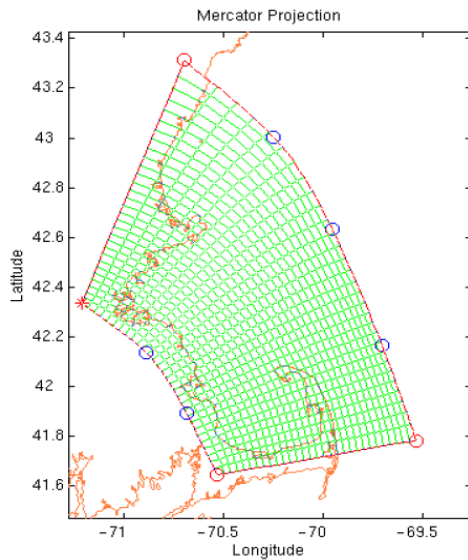
SeaGrid: a Matlab toolbox to create curvilinear orthogonal grids
<http://woodshole.er.usgs.gov/operations/modeling/seagrid/seagrid.html>



These pages are no longer updated.

SeaGrid Orthogonal Grid Maker For Matlab

Dr. Charles R. Denham, U.S. Geological Survey, Woods Hole, MA 02543




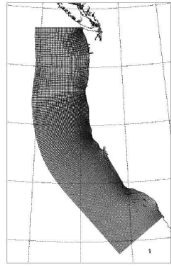

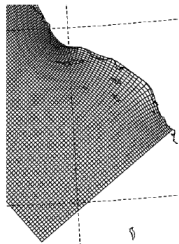


SeaGrid is a Matlab 5.2+ application for generating an orthogonal grid within a curved perimeter, suitable for oceanographic modeling.

With SeaGrid, one can:

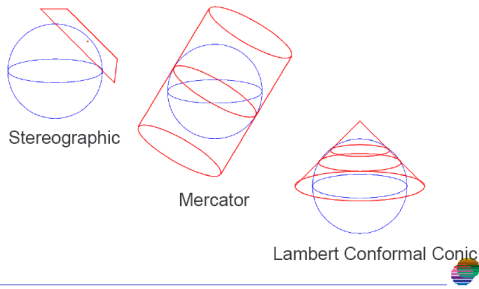
- Select a conformal map projection.
- Manually adjust grid corner points and curved boundaries.
- Modify the number of grid-cells and grid-line spacings.
- Compute depths, land-masking, and orthogonality.
- Save, then later reload a grid for further work.
- Port a grid to Matlab on another computer.
- Create SCRUM/ROMS-like and ECOM-like output files.
- Get comprehensive built-in help.

Starting from a conformal projection (such as Mercator) of the targeted area, SeaGrid uses the Ives-Zacharias scheme to conformally map the curved perimeter to a rectangle, after which, a Poisson solver fills the interior with orthogonally distributed grid points.

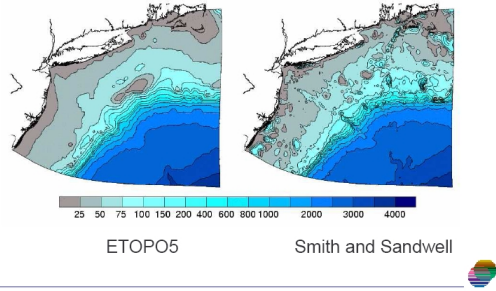
Slides taken from a presentation of SeaGrid at <http://www.myroms.org/Workshops/TOMS2001/presentations/>

<p>ARCTIC REGION SUPERCOMPUTING CENTER</p> <p>Grid Generation with SeaGrid</p> <p>Kate Hedstrom Chuck Denham</p> 	<p>ARCTIC REGION SUPERCOMPUTING CENTER</p> <p>Curvilinear orthogonal grids</p> <ul style="list-style-type: none"> • Define the boundary, interior filled in • Fine resolution at capes, coarse resolution in bays • Best with four 90 degree corners • Control spacing on two adjacent edges  
<p>ARCTIC REGION SUPERCOMPUTING CENTER</p> <p>Curvilinear grids, continued</p> <ul style="list-style-type: none"> • Algorithm uses complex math, requires flat geometry • Masking is a useful feature  	<p>ARCTIC REGION SUPERCOMPUTING CENTER</p> <p>Orthogonal grids on a sphere</p> <ul style="list-style-type: none"> • Find the domain boundary in latitude, longitude • Use conformal map projection on boundary • Create grid in x,y Euclidean space • Use inverse map projection • Recompute grid metrics for spherical geometry 

Common conformal map projections



Bathymetry datasets

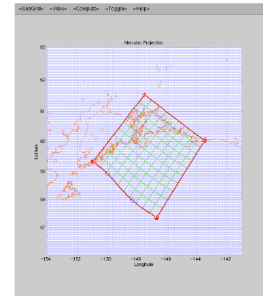


Using SeaGrid

- Extract coastline
 - Pick one of five resolutions from GSHHS
 - Pick latitude, longitude range
- Extract bathymetry
 - Pick latitude, longitude range of ETOPO5 or find something better for your domain
- Run SeaGrid

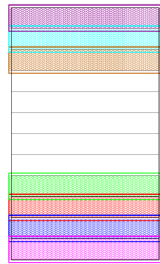
Running SeaGrid

- Load coastline
- Load bathymetry
- Set four corners
- Fuss with boundary
- Set number of gridpoints
- Compute mask and bathymetry
- Export to ROMS/ROMS

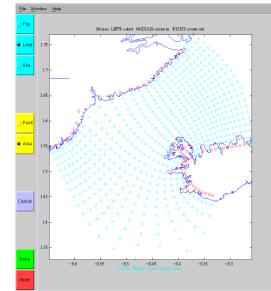


Considerations

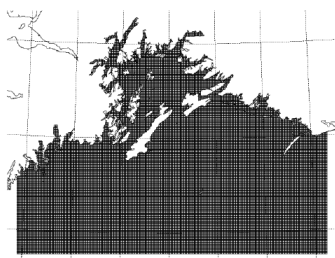
- Know the oceanography
- Parallel tiling
- (Lm+2, Mm+2)
- Lines are through rho points, outermost are image points



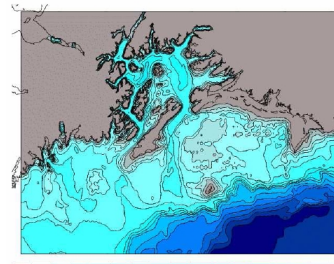
Still need mask editing



Another Prince William Sound grid



Bathymetry



Vertical coordinates and grids

Cartesian z-grid

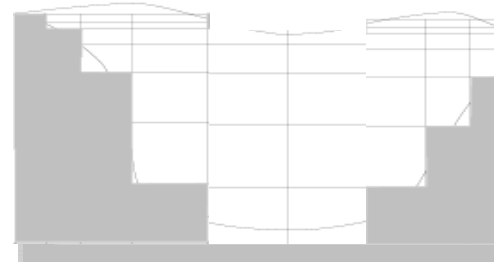
This is the simplest possible vertical grid.

Advantage:

Robust with a numerical diffusion (horizontal) that tends to a stratification without thermal wind

Disadvantage:

Poorly resolved topography (step-wise) especially weak slopes at great depths where Δz is much greater than variations in h between two grid points. Many grid points are effectively unused.



Terrain-following coordinates

The desire to improve the resolution of the topography has led to the use of numerical grids that follow the irregular shape of the model domain. This has led, both in meteorology and in oceanography, to models with curvilinear vertical coordinates. As seen for the horizontal, these coordinate transformations result in a rectangular computational domain which ensures efficient use of computer resources, since all grid points are in water.

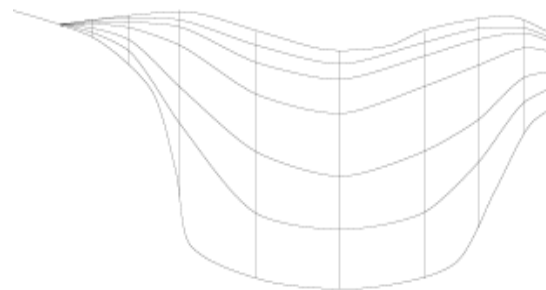
The *terrain-following* coordinates were initially introduced in atmospheric modelling (Phillips, 1957), and have since become the standard in ocean modelling. An example of an ocean model using terrain following coordinates is SPEM (*S-coordinate Primitive Equation Model*), initially developed by Haidvogel et al. (1991) to study processes at high resolution.

Advantages

Good representation of the real topography, good vertical resolution and representation of the free surface

Disadvantages

Quick changes in Δz and very steep grid lines leading to intersects with the density gradients (numerical diffusion of the thermocline and appearance of errors in the pressure gradient – which is horizontal)



Sigma coordinates, e.g., POM model

coordinate equations which are based on the transformation,

$$x^* = x, y^* = y, \sigma = \frac{z - \eta}{H + \eta}, t^* = t \quad (1a, b, c, d)$$

where x, y, z are the conventional cartesian coordinates; $D \equiv H + \eta$ where $H(x, y)$ is the bottom topography and $\eta(x, y, t)$ is the surface elevation. Thus, σ ranges from $\sigma = 0$ at $z = \eta$ to $\sigma = -1$ at $z = H$. After conversion to sigma coordinates and deletion of the asterisks, the basic equations may be written (in horizontal cartesian coordinates),

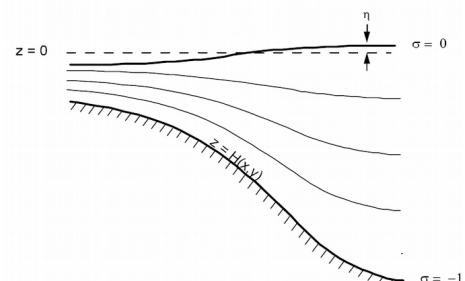


Figure 1. The sigma coordinate system.

Generalised sigma coordinates, e.g., ROMS model.

Figures taken from

Song, Y.T., Haidvogel, D., 1994. *A semi-implicit ocean circulation model using a generalized topography following*

coordinate system. J. Comp. Phys. 115, 228–248.

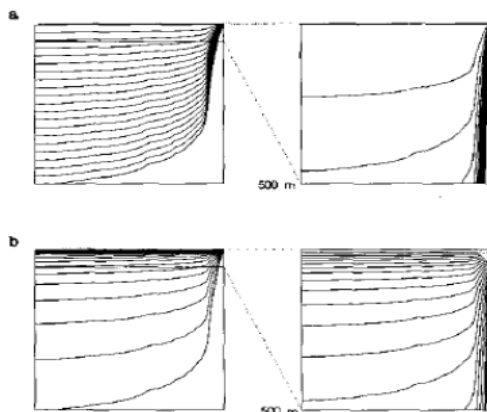
however, there is no coordinate system in use which successfully permits uniformly high resolution near the surface like the z -coordinate, yet preserves the bottom following character of the σ -coordinate. In this paper, we will introduce such a general vertical coordinate (s -coordinate) system. Our s -coordinate consists of three terms:

$$z = \zeta(1 + s) + h_c s + (h - h_c)C(s), \quad -1 \leq s \leq 0, \quad (2.16)$$

where $C(s)$ is a set of s -curves, defined by

$$C(s) = (1 - b) \frac{\sinh(\theta s)}{\sinh \theta} + b \frac{\tanh[\theta(s + 1/2)] - \tanh((1/2)\theta)}{2 \tanh((1/2)\theta)},$$

where θ and b are the surface and bottom control parameters. Their ranges are $0 \leq \theta \leq 20$ and $0 \leq b \leq 1$, respectively. h_c is a constant chosen to be the minimum depth of the bathymetry or a width of surface or bottom boundary layer in which a higher resolution is required.



The first term in (2.16) is used to follow the free surface $\zeta(x, y, t)$, the second is chosen to prevent possible linear instability, and the third is designed to stretch the interior coordinate lines and to follow the bottom $h(x, y)$. There are three attractive features of this s -coordinate system:

- It is a generalized σ -coordinate system. Letting θ approach zero, by L'Hopital's rule, the s -coordinate becomes the σ -coordinate

$$z = (\zeta + h)(1 + s) - h. \quad (2.17)$$

Numerically, taking $\theta \ll 1$ is sufficient to recover the σ -coordinate system.

- It has a simple linear functional form of $\zeta(x, y, t)$ and $h(x, y)$, and z is an infinitely differentiable function of s . Therefore, the formulation of primitive equations in the s -coordinate is similar to that in the σ -coordinate.

- With an equal grid size in s , and by choosing the parameter θ appropriately, the highest resolution is achieved near the surface layer, independently of the varying bottom topography.

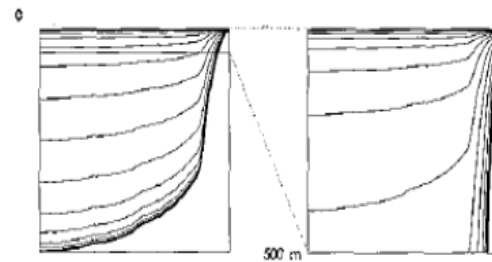


FIG. 1. Vertical computational levels for the California coastal shelf in S -coordinate, left column for the full depth and right column for the upper 500 m: (a) $\theta = 0.0001$, $b = 0$; (b) $\theta = 8$, $b = 0$; (c) $\theta = 8$, $b = 1$.

Figures taken from

A.F. Shchepetkin, J.C. McWilliams: *The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model*. Ocean Modelling 9 (2005) 347–404

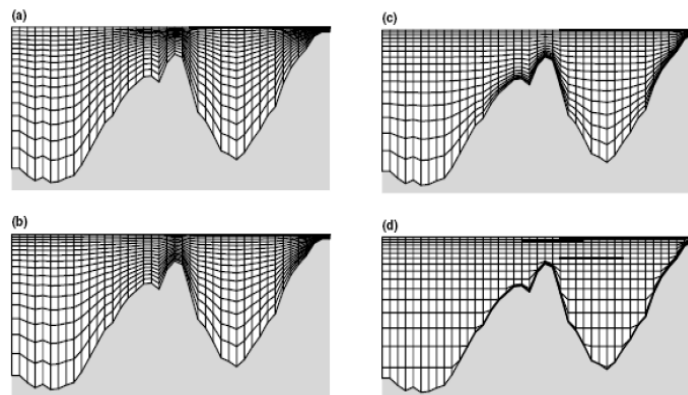


Fig. 1. Examples of vertical coordinate systems: (a) S -coordinate of Song and Haidvogel, 1994 with $\theta_s = 3$ and $\theta_b = 0.01$; (b,c) hybrid $z - \sigma$ coordinate systems obtained by relaxing (d) the z -coordinate system toward the S -coordinate.

Generalised sigma coordinates, e.g., SYMPHONIE model

taken from the thesis by C. Dufau-Julliard

3.4.1.2 La grille du modèle

Les équations décrites aux paragraphes précédents sont discrétisées par la méthode des différences finies sur la grille du modèle SYMPHONIE. Sur ce maillage de type C étagé [Arakawa and Suarez, 1983] les variables sont définies un point sur deux et de façon alternée à la fois sur la verticale et l'horizontale comme illustré sur la Figure 3-2. Les variables turbulentes et la vitesse verticale sont ainsi définies sur les niveaux verticaux entiers tandis que la température, la salinité, la masse volumique et les vitesses horizontales sont définies sur les demi-niveaux.

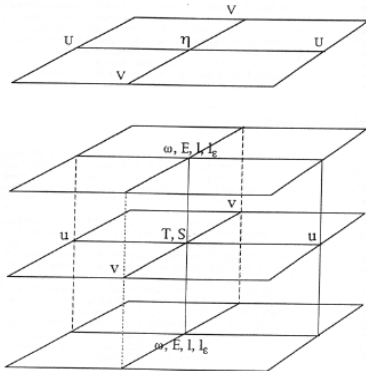


Figure 3-2 Grille du modèle océanique

Selon la verticale, le modèle utilise par ailleurs 25 niveaux de coordonnées sigma généralisées. Les coordonnées de type sigma permettent de suivre la bathymétrie, la résolution verticale étant proportionnelle à l'épaisseur de la colonne d'eau. Le passage de la coordonnée z à la coordonnée sigma généralisée s'écrit :

$$\sigma(i, j, k) = \frac{z(i, j, k) + h(i, j)}{\eta(i, j) + h(i, j)} \tag{3.19}$$

σ vaut donc 1 à la surface et 0 sur du fond.

Dans un modèle à coordonnée sigma classique, comme la version du modèle POM (Princeton Ocean Model) donnée par Blumberg and Mellor [1987], la variable sigma est une grandeur 1D verticale : pour un niveau donné, la valeur sigma est constante quelque soit le point de grille (i,j). Cette coordonnée présente l'inconvénient de diminuer la résolution verticale en surface et près du fond dans les régions de forts gradients bathymétriques. L'augmentation du nombre de niveau permettrait de pallier ce défaut mais aurait pour conséquence un accroissement du coût de calcul. La généralisation du système de coordonnées sigma à une dépendance aux trois indices de la grille permet une meilleure description des phénomènes qui ont lieu au fond et en surface. Elle limite en effet les variations d'épaisseurs des couches sigma dans les régions sensibles. Elle permet ainsi de réduire les erreurs de troncature (représentation erronée du gradient horizontal de pression) dans les régions où le gradient vertical de masse volumique est fort [Auclair et al., 2000a]. Elle réduit aussi la diffusion diapycnale liée aux coordonnées topographiques [Beckman and Haidvogel, 1993]

Isopycnic coordinates, e.g., MICOM model

Instead of following the topography, these coordinates follow the isopycnals

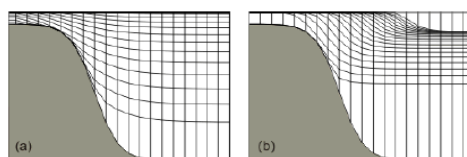
isopycnic coordinate system

http://www.ocean-modeling.org/docs_main.php?page=isopycnic

In an isopycnic coordinate system with potential density as the vertical coordinate, $Z \equiv \sigma_\theta$, and the hydrostatic equation becomes

$$\frac{\partial p}{\partial \sigma_\theta} = -g \sigma_\theta \frac{\partial z}{\partial \sigma_\theta}$$

Figure 4: Vertical discretization for an isopycnic coordinate model and 20 layers: (a) typical time-mean location of coordinate lines, approximately equidistant in density; (b) a less uniform discretization.



Horizontally, MICOM is discretized on the Arakawa "C" grid, with curvilinear coordinates like SPEM/SCRUM. Arbitrary topography can be included.

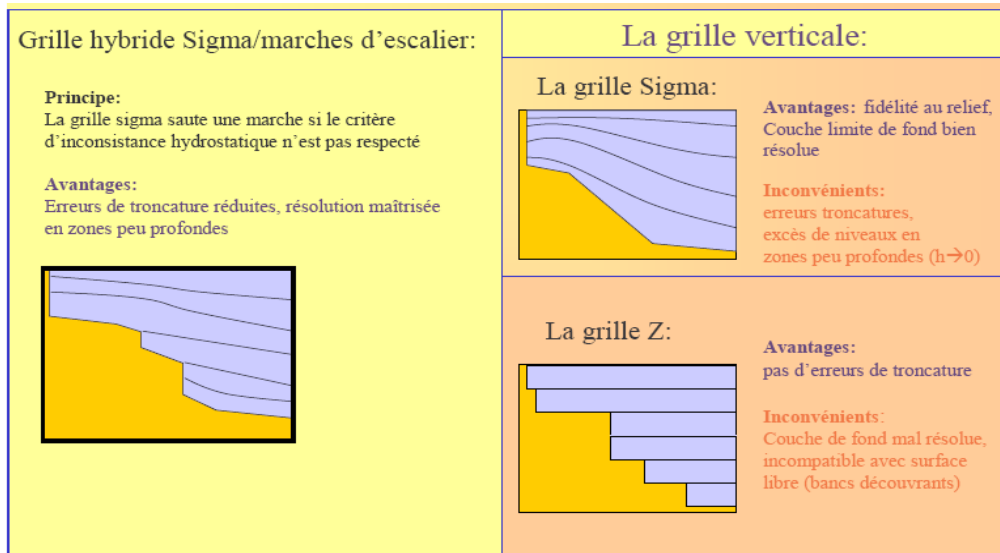
Hybrid coordinate systems

These are “mixtures” between different types of the previously introduced coordinate systems.

SYMPHONIE model: generalised sigma – z - hybrid

Taken from P. Marsaleix

http://www.mercator-ocean.fr/documents/science/ao/ao_2003/gmmc/marsaleix_presentation.ppt



HYCOM model: hybrid between isopycnic, sigma, and z

This model uses isopycnic coordinates in the open ocean where we find strong vertical stratification, terrain following coordinates near the coast, and z coordinates in the mixing layer or in other areas without stratification. This approach allows isopycnic models like MICOM (*Miami Isopycnic Coordinate Ocean Model*) or NLOM (*Navy Layered Ocean Model*) to extend their range of application to coastal areas and non-stratified parts of the open ocean.

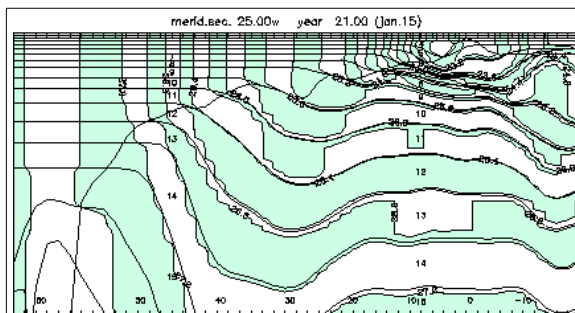


Fig. 1. HYCOM vertical section at 25°W in January of year 21. Shaded field: density. Thin solid lines; layer interfaces. Thick line: mixed-layer depth. Depth range: 500 m. Numbers along bottom indicate latitude. Tick marks at the top and bottom indicate horizontal mesh size.

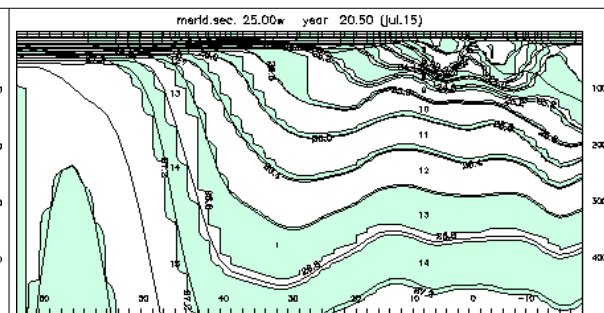


Fig. 2. As in Fig. 1, but for July of year 21.

Figures taken from <http://hycom.rsmas.miami.edu/hycom-model/overview.html>

Advection schemes

The equation

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} = 0$$

can be discretized in several different ways.

Upwind or donor-cell scheme

Simple *forward* discretization in time and *backward* in space

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + u \frac{C_i^n - C_{i-1}^n}{\Delta x} = 0$$

$$C_i^{n+1} = C_i^n - u \frac{\Delta t}{\Delta x} (C_i^n - C_{i-1}^n)$$

Centred differences

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + u \frac{C_{i+1}^n - C_{i-1}^n}{2\Delta x} = 0$$

$$C_i^{n+1} = C_i^n - u \frac{\Delta t}{2\Delta x} (C_{i+1}^n - C_{i-1}^n)$$

This scheme is unstable. It can also be written as:

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} = -\frac{u^2 \Delta t}{2} \frac{\partial^2 C}{\partial x^2} + O(\Delta x^2)$$

Lax-Wendroff (or Leith) scheme (1D)

It is a second-order scheme that is based on the idea of adding as much numerical diffusion as required to compensate for the negative numerical diffusion of the centred scheme. The Symphonie model uses it in 2D.

$$\frac{C_i^{n+1} - C_i^n}{\Delta t} + u \frac{C_{i+1}^n - C_{i-1}^n}{2\Delta x} = \frac{u^2 \Delta t}{2} \frac{C_{i+1}^n - 2C_i^n + C_{i-1}^n}{\Delta x^2}$$

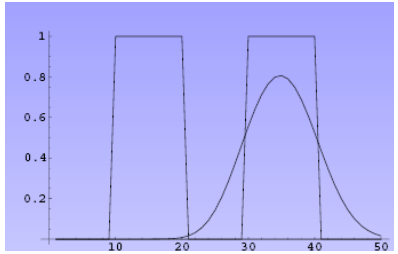
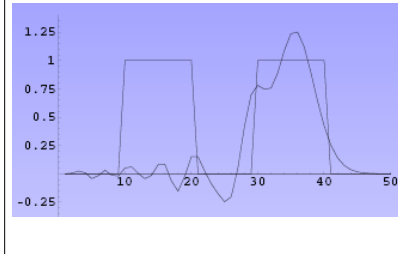
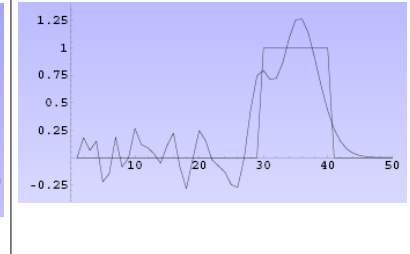
$$C_i^{n+1} = C_i^n - u \frac{\Delta t}{2\Delta x} (C_{i+1}^n - C_{i-1}^n) + u^2 \frac{\Delta t^2}{2\Delta x^2} (C_{i+1}^n - 2C_i^n + C_{i-1}^n)$$

Advection with Leapfrog

$$\frac{C_i^{n+1} - C_i^{n-1}}{2\Delta t} + u \frac{C_{i+1}^n - C_{i-1}^n}{2\Delta x} = 0$$

$$C_i^{n+1} = C_i^{n-1} - u \frac{\Delta t}{\Delta x} (C_{i+1}^n - C_{i-1}^n)$$

Comparing different schemes (see course by JM Beckers)

		
<p>1D upwind advection scheme. Initial condition ("hat" to the left), numerical solution (Gaussian) and exact solution ("hat" to the right) after 200 time steps with CFL=0.1.</p>	<p>1D 2nd-order advection scheme. Initial condition ("hat" to the left), numerical solution (oscillating curve) and exact solution ("hat" to the right) after 200 time steps with CFL=0.1</p>	<p>Leapfrog advection scheme</p>

To choose one scheme over another there are certain criteria one can consider, particularly the so-called *total variation diminishing* (TVD) which was introduced by A. Harten (1983).

Total variation (TV) is defined as

$$TV = \int \left| \frac{\partial u}{\partial x} \right| dx, \text{ or in discretized form } TV = \sum_j |u_{j+1} - u_j|$$

This property is interesting because Harten could prove that


- a numerical scheme that is monotone is TVD, and
- a TVD scheme is monotonicity preserving.

Monotone schemes are attractive to resolve scientific and engineering problems as they never deliver non-physical solutions.

The second-order Lax-Wendroff scheme is dispersive and yields non-physical extremes. The Leapfrog scheme is also dispersive and non-monotonic.

Godunov's theorem proves that only linear first-order schemes (like the upwind scheme) preserve monotonicity and are therefore TVD. Linear higher-order schemes, although more accurate in that they produce solutions that are "smooth", are not TVD and tend to introduce parasitic oscillations (wiggles) where discontinuities or disturbances occur.

To overcome these drawbacks, various high resolution and non-linear schemes have been developed, often using slope or flow limiters.

A non-linear scheme can contain weighted versions of both  upwind (monotonic) and second-order (precise) scheme depending on the solution itself.

Another solution is to adopt the upwind scheme, which is TVD, but whose main flaw is that it is too diffusive. James (1996) has shown that this scheme can be considered a combination of a centred advection scheme and a Laplacian type dissipation term, in which the diffusion coefficient is

$$A = |u| \frac{\Delta x}{2}. \text{ Combined with the Leapfrog we get the following scheme:}$$

$$\frac{\phi_i^{t+\Delta t} - \phi_i^{t-\Delta t}}{2\Delta t} = -\frac{u_{i+1/2}^t}{\Delta x} \frac{\phi_i^t + \phi_{i+1}^t}{2} + \frac{u_{i-1/2}^t}{\Delta x} \frac{\phi_i^t + \phi_{i-1}^t}{2} + \frac{A_{i+1/2}}{\Delta x} \frac{\phi_{i+1}^{t-\Delta t} - \phi_i^{t-\Delta t}}{\Delta x} - \frac{A_{i-1/2}}{\Delta x} \frac{\phi_i^{t-\Delta t} - \phi_{i-1}^{t-\Delta t}}{\Delta x}$$

However, this numerical diffusion can be used like a physical diffusivity: one omits the eddy diffusivity term from the momentum equations and introduces a coefficient $0 < \delta < 1$ which allows

the amount of diffusivity to be regulated $A = \delta |u| \frac{\Delta x}{2}$.

The value of δ is determined via sensitivity analysis and by comparisons with available data (see Hu et al., 2009).

Table 1
Summary of numerical experiment setups and results.

Experiment code	Resolution (km)	δ (non-dimensional)	Eddy area mean \pm SD (km^2)	Tracking duration (days)
<i>exp0</i>	3	$\bar{A} = 15^a$ ($\text{m}^2 \text{s}^{-1}$)	1035 ± 407	42
<i>exp1</i>	3	0.2	950 ± 398	33
<i>exp2</i>	3	0.8	1853 ± 1753	45
<i>exp3</i>	1	0.2	694 ± 446	70
<i>exp4</i>	1	0.8	1193 ± 543	56

^a N.B.: here, the horizontal viscosity coefficient is taken as a fixed value and, hence, does not contain the attenuation coefficient δ .

Z.Y. Hu et al. / Ocean Modelling 28 (2009) 203–208

205

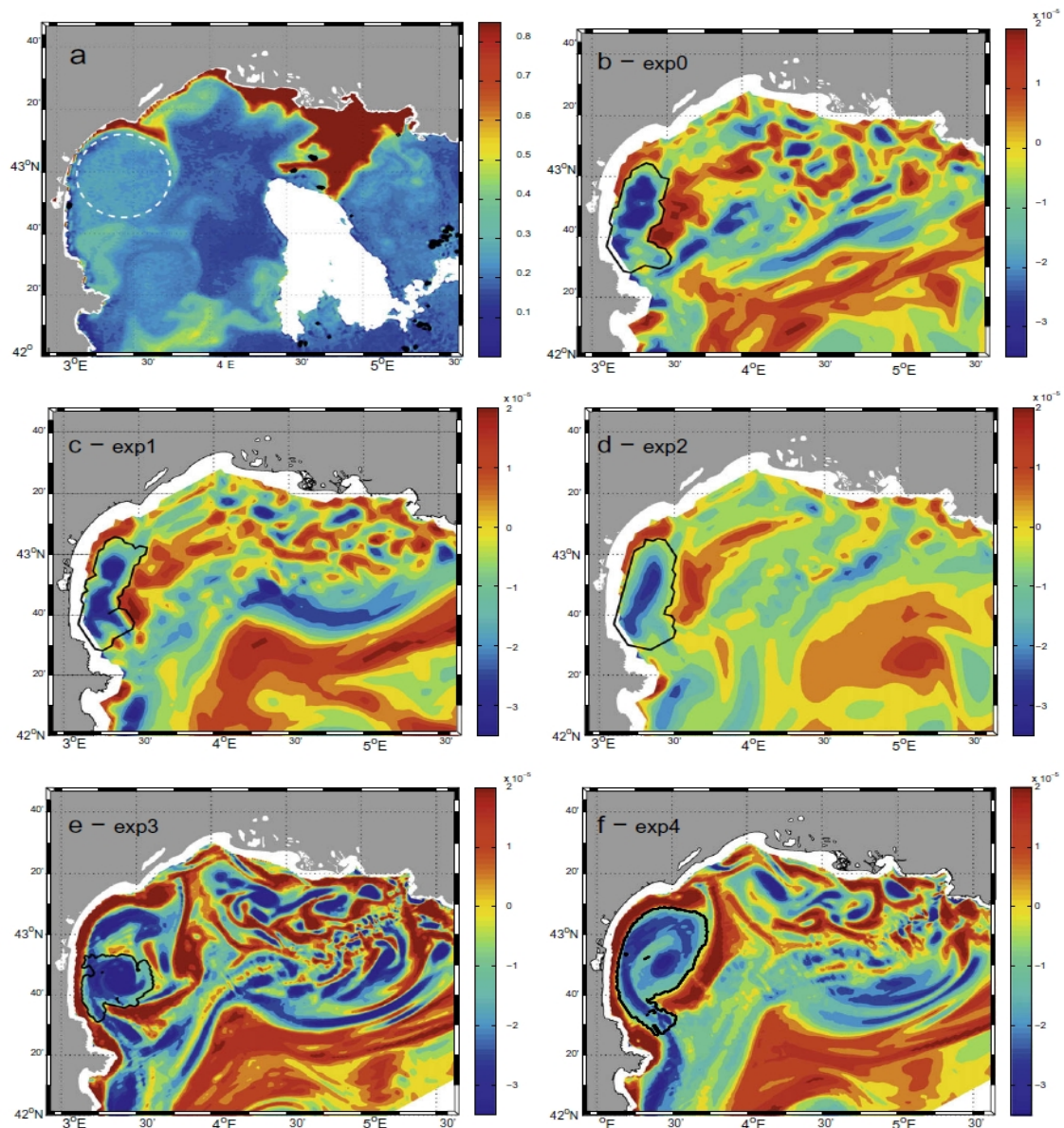


Fig. 3. Comparison of the signature of the anticyclonic eddy on a chlorophyll a [mg m^{-3}] map and the eddy, as identified by the wavelet analysis (contour black) of simulated relative vorticity [s^{-1}] on July 25, 2001. (a) SeaWiFS image processed with OC4 (courtesy E.Bosc); the dashed white circle represents the eddy area. Relative vorticity at 20-m depth from simulations (b) *exp0*, (c) *exp1*, (d) *exp2*, (e) *exp3* and (f) *exp4*. See text for more explanations.

FROM: <http://www.myroms.org/>

ROMS has various options for advection schemes: second- and forth-order centered differences; and third-order, upstream biased. The later scheme is the model default and it has a velocity-dependent hyper-diffusion dissipation as the dominant truncation error ([Shchepetkin and McWilliams, 1998](#)). These schemes are stable for the predictor-corrector methodology of the model. In addition, there is an option for conservative parabolic spline representation of vertical advection which has dispersion properties similar to an eight-order accurate conventional scheme.

There are several subgrid-scale parameterizations in ROMS. The horizontal mixing of momentum and tracers can be along vertical levels, geopotential (constant depth) surfaces, or isopycnic (constant density) surfaces. The mixing operator can be harmonic (3-point stencil) or biharmonic (5-point stencil). See [Haidvogel and Beckmann \(1999\)](#) for an overview of all these operators.

FROM DONG & McWILLIAMS 2007

A third-order, upstream-biased advection operator allows the generation of steep gradients in the solution, enhancing the effective resolution of the solution for a given grid size when the explicit viscosity is small. The numerical diffusion implicit in the third-order upstream-biased operator allows the explicit horizontal viscosity to be set to zero without excessive computational noise or instability (The impact of different forms of the horizontal viscosity on numerical solutions can be found in [Shchepetkin and O'Brien, 1996](#))

TEST MARION May 2011

Different possible choices for the coefficients of viscosity and horizontal diffusivity in ROMS:

- `cppdefs.h`

```
define UV_VIS2
define TS_DIF2
```

The coefficients are constants that are set in `roms.in` or `roms_inter.in`:

```
lateral_visc: VISC2, VISC4 [m^2/sec for all]
              10.    0.
```

```
tracer_diff2: TNU2(1:NT) [m^2/sec for all]
              0.d0 0.d0 0.d0 0.d0 0.d0 0.d0 0.d0 0.d0 0.d0
```

- `cppdefs.h`

```
define UV_VIS2
define TS_DIF2
define VIS_COEF_3D
define DIF_COEF_3D
```

The coefficients are automatically initialized and conserved = 0

- `cppdefs.h`

```
define UV_VIS2
define TS_DIF2
define VIS_COEF_3D
define DIF_COEF_3D
defined VIS_SMAGO
defined DIF_SMAGO
defined UV_SPLIT_UP3
```

Smagorinsky viscosity (harmonic)
Combined Smagorinsky+Peclet hyperviscosity (physical and numerical closure)
Smagorinsky diffusivity (harmonic)

- cppdefs.h

```
define UV_VIS2  
define TS_DIF2  
define VIS_COEF_3D  
define DIF_COEF_3D  
defined VIS_SMAGO  
defined DIF_SMAGO  
defined UV_SPLIT_UP3  
defined TS_SPLIT_UP3
```

Smagorinsky viscosity (harmonic)
Combined Smagorinsky+Peclet hyperviscosity (physical and numerical closure)
Tracer mixing coefficient based on Peclet number

Initial and open boundary conditions

The partial differential equations in ocean circulation models only provide information locally in space and time; to find a global and unique solution for the model domain we require knowledge of the values at the initial time and at the boundaries of the model domain. The fact that we require the initial conditions simply reminds us that the ocean obeys the principle of causality. The boundary conditions provide information about the exchanges with the surrounding environment. For example, the Dirichlet conditions reflect the fact that the system considered is in contact with a reservoir so large that it is not disturbed by the exchanges. The Neumann conditions are imposed if a domain is completely isolated and there are no diffusive exchanges.

From a numerical point of view, to start the iteration that makes up a numerical model, we must know the initial condition $C^0_i, i = 1, \dots, N_x$. On the other hand, to be able to calculate changes from the first and last point in our domain ($i = 1$ and $i = N_x$), we need to know the values at the virtual points C^0_0 and $C^0_{N_x+1}$, i.e., at the points just outside the model domain. This is why we need to specify the boundary conditions.

Classification of partial differential equations

Partial differential equations (PDEs) can be divided into hyperbolic, parabolic, and elliptic types. From a physical point of view, the type of equation dictates the number and type of required initial and boundary conditions; while from a numerical point of view, it guides the numerical methods used. In practise, a precise classification is difficult. Often only a subset of the processes is analysed.

Hyperbolic equations

$$\frac{\partial C}{\partial t} - u \frac{\partial C}{\partial x} = 0$$

Requires an initial condition and a single boundary condition (input) at $x = 0$ if $u > 0$, $x > 0$. The signal is propagated in the direction of the velocity.

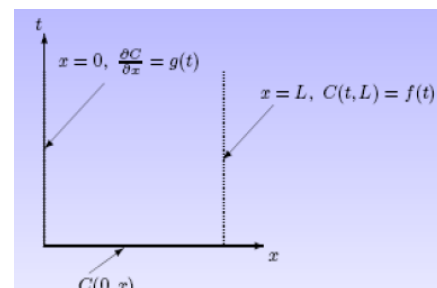
$$\frac{\partial^2 C}{\partial t^2} = u^2 \frac{\partial^2 C}{\partial x^2}$$

Requires two initial conditions (C and $\frac{\partial C}{\partial t}$) and one boundary condition (input) at $x = 0$ and $x = L$. The signal propagates in two directions.

Parabolic equations

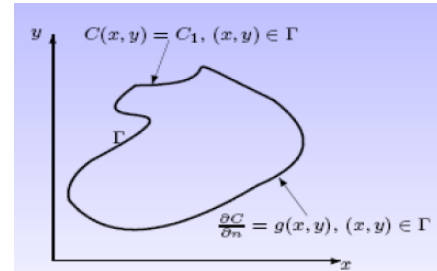
$$\frac{\partial C}{\partial t} = A \frac{\partial^2 C}{\partial x^2}$$

Require one initial condition and one boundary condition (input) at $x = 0$ and $x = L$.



Elliptic equations

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} = -Q \quad \text{Require one boundary condition}$$



Sometimes, a scheme requires more boundary or initial conditions than the mathematical formulation allows. This is due to the fact that we do not solve the mathematical equation but an approximation of it, containing derivatives of higher order and requiring additional auxiliary conditions (see truncation error in the equivalent equation). The effect of these additional boundary or initial conditions should normally disappear as the resolution increases.

Models with open boundaries

text translated from the Ifremer site

To protect oneself from the risk of retaining certain conditions that are completely devoid of physical meaning, the mathematical concept of a well-posed problem has been introduced. A problem is said to be well-posed if:

- the solution exists and
- the solution's behaviour changes continuously with the initial conditions.

However, according to this mathematical definition, a well-posed problem can have an infinite number of solutions, which is not acceptable in modelling. This is why Courant and Hilbert (1962) added the following condition:

- the solution is only determined by the data.

Determining the mathematically correct boundary conditions requires certain assumptions. Without continuity and convergence, the case of a coast (closed border) being the easiest to demonstrate as the normal and tangential velocities become zero.

It has been shown that the open boundary conditions used during the first numerical simulations in fluid mechanics by Charney et al. (1950) with a single-layer quasi-geostrophic model were adequate (constant current at the borders and vorticities prescribed at inflow points). In fact, the problem of the quasi-geostrophic model is well-posed as long as there are no isolated points for which the flow is not tangential at the boundaries.

Bennett (1992) analysed the case of the linear shallow water model for which he identified satisfactory open boundary conditions; they also showed that the problem is ill-posed in the non-linear case. Then he addresses the problem of stratified models. Open boundary conditions can be exhibited in a certain linear case, although rather complex calculations are required in the non-linear case which means that a result may not always be found. They also showed that the number of boundary conditions to be imposed at an open border depends on the local flow regime and direction, i.e., the number of boundary conditions to be added in order to be able to solve the system of primitive equations can differ along the same border and even vary over time. Of course, in a numerical simulation, one does not perform a point-by-point analysis of the flow but imposes the same set of boundary conditions along the entire border. This generally results in an over-determination of the boundary conditions at certain points, in particular the points where the flow is leaving the domain. If the specified conditions were 100% accurate, this over-determination would not cause any problems, but in practice the values are never 100% accurate, already because of the inherent rounding errors in computers.

The conclusion is therefore that the open boundary problem is undoubtedly ill-posed for primitive equation models (see also Yakimiw and Robert 1986 and Robert and Yakimiw 1990).

After these theoretical considerations, let us look at a practical problem: the lack of oceanographic observations. Even if the theory could give us an explicit and perfect solution to the boundary problem, there is still the issue of having to go to sea in order to measure the values that will eventually be fed into the model. It is never possible to have all the necessary observations, unless the modelled domain is quite small. Experiments of this kind have been attempted with quasi-geostrophic models (De Mey and Robinson 1987, De Mey and Ménéard 1989, Rienecker and Miller 1991). Dedicated measurement campaigns have been devised to sample domain borders as well as possible. However, the simulation results showed that the sampling error that is always associated with any kind of field observations quickly make their way from the model boundary to the interior of the model domain, more so if the domain is small (see also Miller and Bennett (1988) for numerical instability problems at open boundaries).

Since the mathematical problem of open boundaries is ill-posed for primitive equation models, it is necessary to remedy this problem. Usually, this is achieved by adding additional conditions that the solution needs to satisfy. These additional conditions are based on a priori knowledge of certain aspects of the solution, e.g., the regularity of the solution (bounded derivatives), available statistics (climatology), or information resulting from observations. We then search for a solution that satisfies all constraints. Physically, this corresponds to removing the indeterminacy of the initial problem by seeking among all the possible solutions the one that is most regular in the case of cross-validation or the one that is closest to the observations in the case of data assimilation. Mathematically, we are faced with an inversion problem.

The classical inversion methods (Tarantola 1987), initially developed for oil drilling, are rather ineffective for the four-dimensional (space + time) problems encountered in meteorology and oceanography where the number of variables is simply too large. New methods resulting from variational calculus have therefore been introduced into these disciplines. These variational methods (LeDimet and Talagrand 1986, Thacker and Long 1988) make it possible to solve the inversion problems encountered in large ocean or atmospheric models at a reasonable computational cost. In principle, they make it possible to find physical fields which use all the information provided by both the digital model and the observations. Variational methods have been used successfully for some time in meteorology to identify the initial conditions and are beginning to be used for estimating the open boundary conditions as well. This promising approach is currently the subject of intense research. The theoretical results of Bennett (1992) show that it is possible, using an inverse variational formulation, to resolve problem of open boundary conditions in primitive equation models. However, this encouraging result is theoretical and its application poses certain problems. Due to the large model domain, only iterative algorithms can be used to solve the problem of minimization on which the variational formulation depends. However, during the iterative process of integration using poorly specified boundary conditions we may find that the model is numerically unstable. This can be avoided by combining the variational methods with a simple and robust solution method. Thus the variational formulation will no longer be applied to the initial unstable problem, but to a modified form which makes it numerically stable.

The preceding remarks show the difficulty operating ocean models in limited domains and explains the small number of publications that really address this problem. They are often associated with meteorological models (Yakimiw and Robert 1986, Robert and Yakimiw 1990, Errico and Bauhefner 1987) and with ocean models where physical problems have been simplified, e.g., shallow water equations (Oliger and Sunström 1978) or reduced gravity monolayers (Camerlango and O'Brien 1980). Of course, it can also be found in studies on quasi-geostrophic models (De Mey and Robinson 1987, De Mey and Ménéard 1989, Rienecker and Miller 1991). Seiler (1993), in particular, identified the boundary conditions using the adjoint method. While this work is purely theoretical and does not involve observations, the method is very interesting and the results encouraging. Marotzke and Wunsch (1993) worked with a primitive equation model using a variational method to study a stationary circulation of the North Atlantic that was compatible with climatology.

Primitive equation models are more complicated. In the absence of theoretical results, practical solutions have been proposed to work around the problem in numerical simulations. The most rudimentary solution is to either move the open boundaries far enough from the region of interest (or until it hits a coast) so that the boundary problem does not have time to propagate to the centre of the model domain during the simulation period. This solution is computationally costly and is not used

very often. Other solutions (radiation conditions, sponge-type conditions) that have been used to date are a little more sophisticated. However, although based on certain theoretical foundations, they are empirical and their effectiveness depends very much on the situation to be simulated.

Practical approaches to find the initial conditions

ROMS model (IRD version ROMS_tools)

The initial conditions of the current field are calculated in *make_clim.m* using the dynamic method and data from the *World Ocean Atlas*.

Below you can find a short reminder of the dynamic method, which permits to calculate a 3D current field in geostrophic equilibrium from *in situ* T and S. For more details see: Copin-Montégut, Le Courant Géostrophique, <http://www.obs-vlfr.fr/Enseignement/enseignants/copin/Geostro.pdf>

The geostrophic current at the isobar p is given by:

$$u_p = \frac{1}{f} g \frac{dh}{dn} \approx \frac{1}{f} g \frac{z^A - z^B}{\Delta n}$$

where dh/dn is the local slope in isobars and z_A and z_B are the depths of the same isobar, p , at two locations A and B that are separated by a distance Δn .

We obtain the depth of the p isobar by integrating the hydrostatic equation.

$$z_p = \frac{1}{g_0} \int_0^p \frac{1}{\rho} dp = \frac{1}{g_0} \int_0^p \alpha dp$$

where we have replaced the density with the mass volume. In practise, the discretized version is used:

$$z_p = \frac{1}{g} \sum_0^p \bar{\alpha} \Delta p$$

The geopotential is defined as

$$\Phi_p \equiv g \times z = \frac{1}{g_0} \int_0^p \alpha dp \approx \sum_0^p \bar{\alpha} \Delta p$$

which finally yields the geostrophic velocity

$$u_p = \frac{1}{f} \frac{\Phi_p^A - \Phi_p^B}{\Delta n}$$

The geopotential of an isobar can be calculated from:

$$\Phi_p = \sum_0^p \bar{\alpha}_{(T,S,p)} \Delta p = \sum_0^p \bar{\alpha}_{(0,35,p)} \Delta p + \sum_0^p \bar{\delta} \Delta p$$

so if we define $D_p = \sum_0^p \bar{\delta} \Delta p$ as the geopotential anomaly, we can rewrite the geostrophic velocity equation as :

$$u_p = \frac{1}{f} \frac{D_p^A - D_p^B}{\Delta n}$$

There exist tabulated values of D for combinations of T and S at a certain isobar p .


Finally, assuming that the isobars at depth are parallel to each other but not the sea surface, we have to level the surface based on a deep isobar (this is **zref** in *make_clim.m*!!!).

Symphonie model

(text from the paper by Estournel et al., 2003)

The purpose of calculating the initial state of the model is to start the simulation with a large-scale circulation already established. In the Gulf of Lion this circulation is quasi-geostrophic and corresponds to the Northern Current [Millot, 1990]. Different methods have already been proposed to model circulations whose horizontal extent exceeds the size of the numerical domain. The model can start from a state at rest by imposing external forcings at the open boundaries establishing a geostrophic current [Pinazo et al., 1996]. Nevertheless, the time for this circulation to establish itself can be long and, because the problem with the primitive equations is ill-posed, the final state is

unpredictable. A different solution is to initialize the coastal model with interpolated dynamic fields based on results provided by a larger scale or even ocean global circulation models (OGCMs) for the Mediterranean [Beckers et al., 1997] and/or *in situ* data. However, Auclair et al. [2000] have shown that this method can sometimes give unsatisfactory results if strong inconsistencies appear between the interpolated bottom currents and the local bathymetry gradients. The start of a model run can be characterized by unrealistic transient processes, such as gravity waves emanating from the coast or from areas with strong gradients in topography. They concluded that a dynamic adjustment to the topographic constraint must be introduced into the large-scale field interpolation scheme in order to avoid conflicts between the bathymetry used by the large-scale model and the bathymetry used by the finer-resolution model. The Gulf of Lion with its steep continental slope, indented by numerous submarine canyons, is an example of an area where the initialization of a coastal model is particularly complicated.

 The solution proposed by Auclair et al. [2000b] is to apply an optimal interpolation method based on the linearized model equations. If the full set of equations and variables of the model is used, this inverse method leads to solve an eventually large linear equations system. An interesting compromise between c.p.u. and equilibrium of the initial state can be found if the set of equations and variables is for instance limited to the external mode of the model. In the present paper, the initialization is calculated from a set of linearized equations deduced from the model under the following assumptions:

1. Rigid-lid approximation

$$U = -\frac{\partial \Psi}{\partial y}, \quad V = \frac{\partial \Psi}{\partial x} \quad (\text{A1})$$



with U and V the transport components and Ψ the barotropic stream function.

2. Parametrization of the vertical geostrophic current shear

Let us assume the following form for the pressure gradient

$$\nabla p(z) = \nabla p_s + g \int_z^0 \nabla \rho dz \quad (\text{A2})$$

where ∇p_s is the surface pressure gradient and $\nabla \rho$ the horizontal density gradient.

We followed the approach by Cooper and Haines [1996] to simplify (A2). Considering $\rho(z)$ a horizontally homogeneous density field, we may assume that  horizontal density gradients result from homogeneous vertical displacements h of the water columns.  The pressure gradient may then be approximated by

$$\nabla p(z) = \nabla p_s + g(\rho(0) - \rho(z))\nabla h \quad (\text{A3})$$

the gradient of vertical displacement ∇h may be given using (A3) and a condition of no motion at a prescribed depth z_{ref} , which may be the depth of the bottom as in the work of Cooper and Haines [1996]:

$$\nabla h = -\frac{\nabla p_s}{g(\rho(0) - \rho(z_{ref}))} \quad (\text{A4})$$

Inserting (A4) into (A3) yields:

$$\nabla p(z) = R(z)\nabla p_s = R(z)g\rho_o\nabla\eta \quad (\text{A5})$$

with

$$R(z) = \frac{\rho(z) - \rho(z_{ref})}{\rho(0) - \rho(z_{ref})} \quad (\text{A6})$$

$R(z)$ was first determined using (A6) and the CTD density data. We can see from (A5) that $R(z)$ can be considered as a non-dimensional form of the vertical shear of the geostrophic current. The validity of our basic assumption (∇h is vertically uniform) was then examined thanks to a comparison of $R(z)$ with geostrophic calculations performed from hydrographic data. Finally, we concluded that $R(z)$ was better estimated by:

$$R(z) = \frac{V_g(z)}{V_g(0)} \quad (\text{A7})$$

where V_g is the mean geostrophic current modulus. Geostrophic currents were calculated from CTD data located in the region of the Liguro-Provençal current, using a classical level of no motion at the bottom [Durrieu de Madron et al., 1992].

3. Equilibrated Linearized Transport Equations

Neglecting advective terms and taking (A5) into account, the equilibrated equations of the external mode are given by:

$$-f\frac{\partial\psi}{\partial x} = -g\bar{R}\frac{\partial\eta}{\partial x} + \frac{\tau_{sx}-\tau_{bx}}{\rho_o} - H\left[\frac{\partial}{\partial x}A_h\frac{\partial}{\partial x}\left(\frac{1}{H}\frac{\partial\psi}{\partial y}\right) + \frac{\partial}{\partial y}A_h\frac{\partial}{\partial y}\left(\frac{1}{H}\frac{\partial\psi}{\partial x}\right)\right] \quad (\text{A8})$$

$$-f\frac{\partial\psi}{\partial y} = -g\bar{R}\frac{\partial\eta}{\partial y} + \frac{\tau_{sy}-\tau_{by}}{\rho_o} + H\left[\frac{\partial}{\partial x}A_h\frac{\partial}{\partial x}\left(\frac{1}{H}\frac{\partial\psi}{\partial x}\right) + \frac{\partial}{\partial y}A_h\frac{\partial}{\partial y}\left(\frac{1}{H}\frac{\partial\psi}{\partial y}\right)\right] \quad (\text{A9})$$

where τ_s is the wind stress, τ_b the bottom stress, H is the bottom depth, A_h the horizontal viscosity coefficient, and $\bar{R} = \int_{-H}^0 R dz$.

The bottom stress, usually parametrized by a quadratic law, is here given by a linear relation $(\tau_{bx}, \tau_{by}) = \rho_o c_d (u_b, v_b)$, the bottom current is estimated from the transport and vertical shear of the geostrophic current using the relationship:

$$(u_b, v_b) = \frac{R_{(z=-H)}}{\bar{R}}(U, V) = \frac{R_{(z=-H)}}{\bar{R}}\left(-\frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial x}\right) \quad (\text{A10})$$

If (A8) and (A9) are divided by \bar{R} , then differentiated with respect to y and x , respectively, and then (A8) subtracted from (A9), the surface elevation can be eliminated. Using (A10) an equation for the stream function can be finally obtained (see thesis by Dufau-Juillard, Eq. 3.48).

$$-\frac{\partial}{\partial y} \left(\frac{f}{\bar{R}} \right) \frac{\partial \Psi}{\partial x} + \frac{\partial}{\partial x} \left(\frac{f}{\bar{R}} \right) \frac{\partial \Psi}{\partial y} = r \frac{\partial}{\partial x} \left(\frac{R_{(z=-h)}}{\bar{R}^2} \frac{\partial \Psi}{\partial x} \right) + r \frac{\partial}{\partial y} \left(\frac{R_{(z=-h)}}{\bar{R}^2} \frac{\partial \Psi}{\partial y} \right) - \left[\frac{\partial}{\partial x} \left(\frac{\theta(\partial \Psi / \partial x)}{\bar{R}} \right) - \frac{\partial}{\partial y} \left(\frac{\theta(\partial \Psi / \partial y)}{\bar{R}} \right) + \frac{1}{Tr} \left[\frac{\partial}{\partial y} \left(\frac{U_g + (\partial \Psi / \partial y)}{\bar{R}} \right) - \left(\frac{V_g - (\partial \Psi / \partial x)}{\bar{R}} \right) \right] \right]$$

In practise, a linear equation system of the following type:

$$A\Psi = B \quad (A11)$$

is numerically solved on the model grid. Ψ is a column vector containing values of the stream function at each grid point. The matrix A contains the operators of the stream function that deduced from the linearized schemes of the external mode of the model. The column vector

contains the forcing terms, namely the curl of $\frac{\tau_s}{\rho_o \bar{R}}$ and the boundary conditions.

4. Boundary conditions

The stream function vanishes at the coast. The problem of the open boundary was rather tricky insofar as we did not have outputs of a large scale model to specify the boundary values of the stream function. General characteristics of the northwestern Mediterranean which have been reported by numerous authors over the last 20 years (see, for instance, Millot [1990]) were nevertheless introduced at the open boundaries in a simplified manner. The assumption that the Liguro-Provençal current mainly followed the continental slope led us to prescribe a bottom depth dependent relationship. The boundary stream function has been set to:

$$\Psi_{pb} = \alpha H \quad (A12)$$

calculated by ADCP measurements.

The intensity of the slope current obtained by (A11) is strongly dependent on the value of α (instead of the wind stress value, especially in the Gulf of Lion region) and Estournel et al. [2003] set its value equal to 400 s^{-1} (OK for PSI if not CHECK DIMENSIONS ???) in order to obtain values comparable to the observed currents and they solved numerically (A11) over a domain larger than the model domain in order to reduce the influence of the border.

5. Initialisation of the model

Let Ψ_{t0} be a solution of (A11). The component of transport, (U_{t0}, V_{t0}) are deduced from Ψ_{t0} using (A1). The components of the current are deduced from the transport and the vertical shear parametrization R :

$$(u_{t0}, v_{t0}) = \frac{R(z)}{\bar{R}} (U_{t0}, V_{t0}) \quad (A13)$$

Transport and current values are then introduced in (A8) and (A9) in order to calculate the surface elevation. Similarly to (A11), (A8) and (A9) are used to form an overdetermined system of linear equations solved on the model grid:

$$C X = D \quad (A14)$$

where X is a column vector containing the surface elevation value at each grid point. The matrix C contains the horizontal gradient operators of the surface elevation, and D the remaining terms of (A8) and (A9), i.e., the Coriolis term, the bottom and wind stresses and the horizontal viscosity. The solution of (A14) is

$$X = (C^T C)^{-1} C^T D \quad (A15)$$

A perturbation of the density field can be deduced from the surface elevation, using the hydrostatic equilibrium assumption and (A5) differentiated respect to z :

$$\rho_{t0} = -\rho_o R' \eta_{t0} \quad (A16)$$

where R' is the derivative of R with respect to z . The total density field is given by

$$\rho = -\rho(z) + \rho_{t0} \quad (A17)$$

where the horizontally homogeneous density field, $\rho(z)$, is chosen to be representative of the simulated case, namely determined from the FETCH CTD data. The initial salinity field, S_{t0} is given by a function of density empirically determined from the set of CTD data. The temperature field, T_{t0} , is finally computed using the model equation of state. The complete field is then used as an initial state for the simulations. In agreement with conclusions of Auclair et al. [2000b], the initial state is stable and does not produce significant transient adjustment processes at the beginning of the simulation.

Stability analyses on the duration of the spin-up period and comparisons between the model and data led Estournel et al. [2003] to consider 8 days a period that was sufficient for the local forcings by wind and the Rhône river plume to initiate the local circulation.

The numerical schemes to solve (A11) and (A14) derive directly from the external mode of the model. This aspect makes it possible to obtain an initial state in accordance with the dynamics of the model and therefore to be able to reduce the spin-up time. The Northern Current is well represented in the model, particularly its intensity and the three-dimensional velocity distribution. These two characteristics are easily controlled via the open boundary conditions and through the function $R(z)$. However, Dufau-Juilland et al. [2004] pointed out how the choice of a single $R(z)$ for the whole model domain limits the level of realism of the initial state. It can be expected that there is a spatial variability in the vertical current shear between the coast, the shelf, and the open sea.

Auclair et al. [2003] showed that intrusions of the slope current onto the continental shelf are more likely when $R(z)$ decreases rapidly with depth (shallow current) and that barotropic currents ($R(z) = 1$) tend to follow the continental slope.

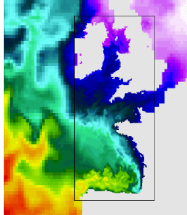
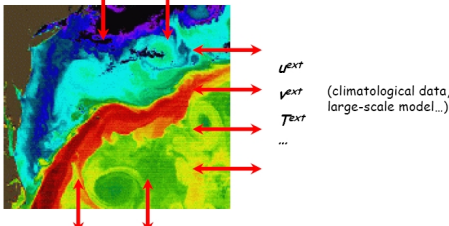
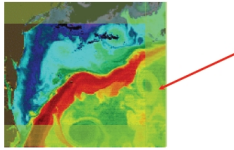
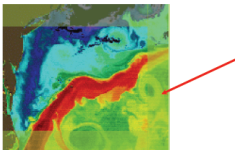
Dufau-Juilland et al. [2004] proposed to initialize the coastal model using a linear combination of N solutions Ψ^i corresponding to several vertical profiles $R^i(z), i=1, N$, i.e.,

$$(\Psi, u, v, \nabla P_s, \nabla \rho) = \sum_{i=1, N} (\Psi^i, u^i, v^i, \nabla P_s^i, \nabla \rho^i)$$

in order to reproduce the spatial variability of the three-dimensional circulation in the Gulf of Lion with a limited number of profiles. In particular, three profiles were chosen from the data of the large-scale MOM model: one for the open sea, one for the slope area, and one for the continental shelf area.

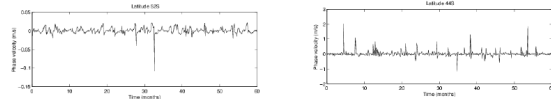
Moreover, the method of Dufau-Juilland et al. [2004] also differs from that of Estournel et al. [2003] by their choice of boundary conditions, which were provided by the MOM model (run for the entire Mediterranean), and not by a schematic representation of the Northern Current.

Open boundary conditions for models solving the primitive equations

<p>Selection of slides from a presentation by E.Blayo</p> <p><i>Nesting Ocean Models</i></p> <p>http://www-meom.hmg.inpg.fr/Web/GODAE_school/Pdf/E.Blayo.pdf</p>	<h3 style="text-align: center;">Nesting ocean models</h3> <p style="text-align: center;">E. Blayo LMC-IMAG, University of Grenoble and INRIA Rhône-Alpes</p>  <p style="text-align: right;"><i>Collaboration with :</i> L. Debreu, V. Fedorenko, C. Vouland (LMC-IMAG) B. Barnier, G. Broquet, S. Cailleau (LEGI, Grenoble) L. Halpern, C. Japhet, V. Martin (LAGA Paris 13) P. Marchesiello (IRD Brest) F. Vandermeirsch (IFREMER Brest)</p>
<p><u>The open boundary problem / one-way nesting</u></p>  $\begin{cases} L_{loc}u_{loc} = f_{loc} & \text{in } \Omega_{loc} \times [0, T] \\ Bu_{loc} = ?? & \text{on } \Gamma \times [0, T] \end{cases}$ <p>Goal : choose the partial differential operator B in order to</p> <ul style="list-style-type: none"> ➢ evacuate the outgoing information ➢ bring some external knowledge on incoming information 	<p><u>The open boundary problem (2)</u></p> <ul style="list-style-type: none"> ➢ OBCs (data + operator) have a strong influence on the model results. ➢ Designing OBCs is an old problem : abundant literature, numerous conditions proposed, often no clear conclusions. ➢ Some OBCs are often recommended in comparative studies : radiation methods with relaxation term, Flather condition, relaxation (+ sponge layer), Hedström condition. <ul style="list-style-type: none"> ➢ give an overview of usual OBCs ➢ performances of OBCs : fact and fiction ➢ How can we design OBCs for primitive equations ?
<p><u>Usual open boundary conditions : relaxation methods</u></p>  <p style="text-align: center;">relaxation layer $(1 - \alpha(x)) \phi + \alpha(x) \phi^{ext}$</p> <ul style="list-style-type: none"> ➢ This is equivalent to adding a nudging term: $\frac{\partial \phi}{\partial t} + F(\phi) + K(\phi - \phi^{ext}) = 0$ ➢ Relaxation is often used together with a sponge layer (i.e. increased viscosity within the relaxation area) ➢ Limit case : Clamped boundary conditions $\phi = \phi^{ext}$ 	<p><u>Usual open boundary conditions : relaxation methods (2)</u></p>  <p style="text-align: center;">$(1 - \alpha(x)) \phi + \alpha(x) \phi^{ext}$</p> <ul style="list-style-type: none"> ➢ Clamped boundary conditions $\phi = \phi^{ext}$ should be avoided : the outgoing information does not depend on the interior solution. ➢ Relaxation + sponge layer gives good results in all comparative studies (e.g. Röed & Cooper, 87; Palma & Matano, 98; Nycander & Döös, 03). ➢ Drawbacks: - increase of computational cost - solution in the relaxation layer ? ➢ Future methods : Perfectly Matched Layers (Berenger, 1994; Hu, 2001; Navon et al, 2004)
<p><u>Usual open boundary conditions : radiation methods</u></p> <ul style="list-style-type: none"> ➢ Radiation conditions are based on the Sommerfeld condition : $\frac{\partial \phi}{\partial t} + c \frac{\partial \phi}{\partial x} = 0$ <ul style="list-style-type: none"> ➢ For complex flows: adaptive evaluation of c (Orlanski-type methods) <p>$c = c(x, t)$ is computed at each gridpoint on the boundary and at each timestep, using inner values at the same or at previous timesteps.</p> $c(x, t) = - \frac{\partial \phi / \partial t}{\partial \phi / \partial x} \quad \text{gives for example} \quad c_B^n = - \frac{\Delta x}{\Delta t} \frac{\phi_{B-1}^{n-1} - \phi_{B-1}^{n-2}}{\phi_{B-1}^{n-1} - \phi_{B-2}^{n-1}}$ <p>Numerous variants : Orlanski (76), Camerlengo & O'Brien (80), Miller & Thorpe (81), Raymond & Kuo (84), Barnier et al. (98), Marchesiello et al. (01)</p> <ul style="list-style-type: none"> ➢ Results and recommendations are split. 	<p><u>Usual open boundary conditions : radiation methods (2)</u></p> <p>Orlanski = Sommerfeld + an arbitrary hypothesis e.g. $\frac{\partial c}{\partial x} = 0$</p> <ul style="list-style-type: none"> ➢ This additional hypothesis makes the condition nonlinear (Nycander & Döös, 03). $\frac{\partial^2 \phi}{\partial t \partial x} \frac{\partial \phi}{\partial x} - \frac{\partial \phi}{\partial t} \frac{\partial^2 \phi}{\partial x^2} = 0$ <ul style="list-style-type: none"> ➢ Orlanski-type methods have no justification as soon as there is a combination of several waves with different velocities : <p style="text-align: center;">$F(x, t) = f(x - ct)$ satisfies any Orlanski-type condition, but $F(x, t) = f_1(x - c_1 t) + f_2(x - c_2 t)$ with $c_1 \neq c_2$ does not.</p>

Usual open boundary conditions : radiation methods (3)

➤ The adaptive estimation of c has no physical meaning, but is more or less a white noise (Tréguier et al., 2001; Durran, 2001).



➤ Their apparent efficiency in some studies is mostly due to the addition of a **relaxation term**.

$$\begin{cases} \phi = \phi^{ext} & \text{or} & \frac{\partial \phi}{\partial t} = -\frac{\phi - \phi^{ext}}{\tau_{in}} & \text{if } c \text{ is inward} \\ \frac{\partial \phi}{\partial t} + c \frac{\partial \phi}{\partial n} = -\frac{\phi - \phi^{ext}}{\tau_{out}} & & & \text{if } c \text{ is outward} \end{cases}$$

$\frac{\partial(\phi - \phi^{ext})}{\partial t} = -\frac{\phi - \phi^{ext}}{\tau}$ or $\phi = \phi^{ext}$ would give quite similar results.

Usual open boundary conditions : Flather condition

For 2-D barotropic flows :

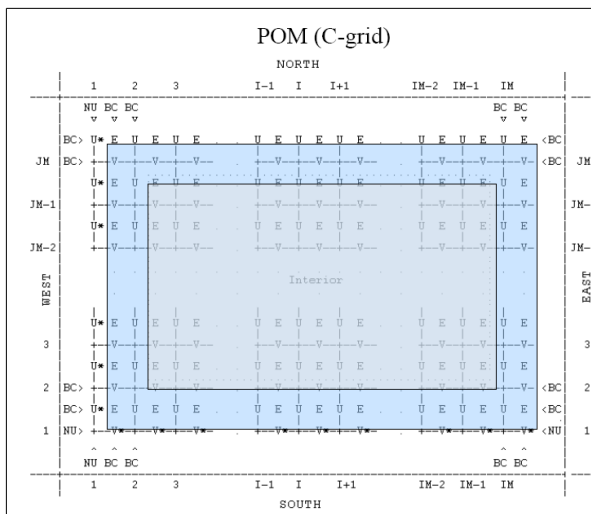
Sommerfeld condition for surface elevation : $\frac{\partial \eta}{\partial t} + \sqrt{gh} \frac{\partial \eta}{\partial n} = 0$

1-D approximation of the continuity equation : $\frac{\partial \eta}{\partial t} + h \frac{\partial \bar{v}_n}{\partial n} = 0$

Combination + integration through Γ : $\bar{v}_n - \sqrt{\frac{g}{h}} \eta = \bar{v}_n^{ext} - \sqrt{\frac{g}{h}} \eta^{ext}$

- Used with additional conditions like $\frac{\partial h}{\partial n} = 0$
- Gives good results in all comparative studies (e.g. Palma & Matano, 98; Marchesiello et al, 01; Nycander & Döös, 03).

Example coding



Clamped Condition

```

!*** NORTH ***
      j=JM
      ELF(i,JM)=? VAF(i,JM)=? UAF(i,JM)=?

!*** WEST ***
      i=1
      ELF(1,j)=?
      UAF(2,j)=?
      VAF(1,j)=?

!*** EAST ***
      i=IM
      ELF(IM,j)=?
      UAF(IM,j)=?
      VAF(IM,j)=?

!*** SOUTH ***
      j=1
      ELF(i,1)=? VAF(i,2)=? UAF(i,1)=?
    
```

$$\begin{aligned} \eta &= \eta_{BC} \\ U &= U_{BC} \\ V &= V_{BC} \end{aligned}$$

Zero Gradient Condition

```

!*** NORTH ***
      j=JM
      ELF(i,JM)=ELF(i,JM-1) VAF(i,JM)=? UAF(i,JM)=?

!*** WEST ***
      i=1
      ELF(1,j)=ELF(2,j)
      UAF(2,j)=UAF(3,j)
      VAF(1,j)=VAF(2,j)

!*** EAST ***
      i=IM
      ELF(IM,j)=ELF(IM-1,j)
      UAF(IM,j)=UAF(IM-1,j)
      VAF(IM,j)=VAF(IM-1,j)

!*** SOUTH ***
      j=1
      ELF(i,1)=ELF(i,2) VAF(i,2)=VAF(i,3) UAF(i,1)=UAF(i,2)
    
```

$$\begin{aligned} \frac{\partial \eta}{\partial n} &= 0 \\ \frac{\partial U}{\partial n} &= 0 \\ \frac{\partial V}{\partial n} &= 0 \end{aligned}$$

Flather Condition

```

!*** NORTH ***
      j=JM
      ELF(i,JM)=ELF(i,JM-1) VAF(i,JM)=? UAF(i,JM)=?

!*** WEST ***
      i=1
      ELF(1,j)=ELF(2,j)
      UAF(2,j)=UAF(3,j)
      VAF(1,j)=VAF(2,j)

!*** EAST ***
      i=IM
      ELF(IM,j)=ELF(IM-1,j)
      UAF(IM,j)=? + sqrt(G/H(IM,j))
      VAF(IM,j)=VAF(IM-1,j)

!*** SOUTH ***
      j=1
      ELF(i,1)=ELF(i,2) VAF(i,2)=VAF(i,3) UAF(i,1)=UAF(i,2)
    
```

$$\begin{aligned} \frac{\partial \eta}{\partial n} &= 0 \\ U &= U_{BC} + \sqrt{\frac{g}{H}} (\eta - \eta_{BC}) = 0 \end{aligned}$$

Tutorials

TD 1: Familiarise yourself with Linux; register as a ROMS user and download the model and data

After logging into one of the Linux machines in the informatics room, open a *Terminal* window.

Try the following Linux commands:

pwd to know which directory you are in

cd *name_of_directory* change to a different directory (**cd** by itself returns you to your home directory)

ls list the contents of a directory (useful options: **ls -l**, **ls -lh**, **ls -lrth**)

man *name_of_command* displays the man (help) page on a particular command

mkdir *name_of_directory* create a new directory

rm -i *name_of_file* delete a file (**rmdir** for directories)

& at the end of a command (or **Ctrl+Z** followed by **bg**) executes the command in the background (**fg**, **top**)

Open a second terminal window and connect to the computing cluster using

ssh *login@cluster.osupytheas.fr*

Compare the contents of your local home directory and the home directory on the cluster: the directory-link named *modelisation* should be in both.

Create a subdirectory *modelisation/ROMS* which will serve as your working directory for this course.

Go to <https://www.croco-ocean.org>

Read the homepage and open the link *History* for additional info: a new model, CROCO, has been born!

Click on *Download* and *ROMS_AGRIF* and read the text, then register as a ROMS user by opening the following link in a new window: [official ROMS website](#). 

Come back to the *Download* page and download the following files into your working directory (right click and “save link as ...”):

- from the *ROMS_AGRIF* page, *Source code* tab

Roms_Agrif_v3.1.1_07_07_2014.tar.gz;

- from the *ROMS_AGRIF* page, *Roms tools* tab:

ROMSTOOLS_v3.1.1_07_07_2014.tar.gz ;

- from the *Utilities* page:

Utilities_ROMSTOOLS_v3.0_21_12_2012.tar.gz

- from the *Datasets* page:

COADS05/ COADS05_2006_10_25.tar.gz

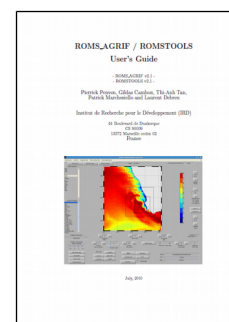
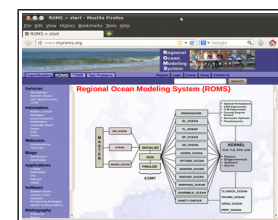
Topo/ Topo_26_05_2004.tar.gz

WOA2009_24_06_2011.tar.gz

***m_map1.4f/private/**.b GSHHS_coastline_05_02_2013.tar.gz**. 

Open the page <https://www.croco-ocean.org/documentation-2/> and download from the *Archives* tab the PDF

ROMS_AGRIF v2.1 & ROMSTOOLS v2.1 User Guide.



Then, verify that there are 8 files in your working directory.

SOME ACRONYMS

NCEP



National Weather Service



The Office of the Director at the National Centers for Environmental Prediction gives overarching management to the nine centers, which include the:

Aviation Weather Center provides aviation warnings and forecasts of hazardous flight conditions at all levels within domestic and international air space.

Climate Prediction Center monitors and forecasts short-term climate fluctuations and provides information on the effects climate patterns can have on the nation.

Environmental Modeling Center develops and improves numerical weather, climate, hydrological and ocean prediction through a broad program in partnership with the research community.

Hydrometeorological Prediction Center provides nationwide analysis and forecast guidance products out through seven days.

NCEP Central Operations sustains and executes the operational suite of numerical analyses and forecast models and prepares NCEP products for dissemination.

Ocean Prediction Center issues weather warnings and forecasts out to five days for the Atlantic and Pacific Oceans north of 30 degrees North.

Space Environment Center provides space weather alerts and warnings for disturbances that can affect people and equipment working in space and on earth.

Storm Prediction Center provides tornado and severe weather watches for the contiguous United States along with a suite of hazardous weather forecasts.

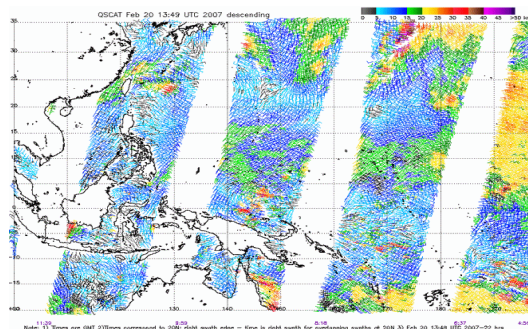
Tropical Prediction Center includes the National Hurricane Center and provides forecasts of the movement and strength of tropical weather systems and issues watches and warnings for the U.S. and surrounding areas.

<http://www.ncep.noaa.gov/>

QuikSCAT Ocean Page

This page contains the most recent wind vector images covering the Atlantic Ocean, Indian Ocean, East Pacific and West Pacific Ocean. It is updated shortly after receipt of new QuikSCAT orbit data files.

QuikSCAT is a polar orbiting satellite with an 1800 km wide measurement swath on the earth's surface. Generally, this results in twice per day coverage over a given geographic region.



http://manati.orbit.nesdis.noaa.gov/cgi-bin/qscat_ocean.pl

ECMWF re-analysis

From Wikipedia

The ECMWF re-analysis project is in two phases. The first, ERA-15 generated re-analyses from December 1978 to February 1994. The second, ERA-40, begins in 1957 (the IGY).

Numerical weather prediction models generate an "analysis" to begin their forecasts from. These analyses themselves are a valuable tool for meteorological and climatological analysis. However, they suffer from inconsistency, because the analysis system is being constantly improved. Hence, the idea developed to re-analyse all the old data, including much archived data that was not available to the original analyses, using a consistent modern analysis system.

ICOADS: The International Comprehensive Ocean-Atmosphere Data Set Project

Is the Earth's ocean surface getting warmer? Are wind and pressure patterns changing over the global oceans and do these changes relate to recent extreme weather events?

To answer questions such as these, data collected today must be compared to data taken over a sufficiently long time span in the past. The historical surface marine record, presently covering more than 200 years, is maintained through the ICOADS project. The original COADS project, and the continuing US contribution toward the new international database, ICOADS, is the result of a cooperative effort between NOAA -- its NESDIS/National Climatic Data Center (NCDC) and its OAR/Earth System Research Laboratory (ESRL) -- and

the National Science Foundation's National Center for Atmospheric Research (NCAR). ICOADS provides a standard of comparison and gives meaning and context to the environmental measurements being collected daily around the globe.

Global marine data observed between 1784 and 2004 (the currently available period-of-record), primarily from ships of opportunity, have been collected, edited, and summarized statistically for each month of each year of the period. These measurements of temperatures, humidity, winds, pressures, waves, and clouds have been applied to global and regional studies of heat balance and heat exchange at the air-sea interface, ENSO, North Atlantic Oscillation, satellite data calibration, sustainable fisheries, 4-dimensional ocean circulation, shelf dynamics, coastal morphology, sea level, and more.

<http://www.ncdc.noaa.gov/oa/climate/coads/>

*****S

SODA Simple Ocean Data Assimilation

The algorithms explored in SODA are sequential estimation with empirically modeled error covariances. The basic algorithm is described in two papers (Carton et al., 2000a,b: see <http://www.meto.umd.edu/~carton/carton/ref.html>)[1]. We have recently implemented an improved two-stage bias correction algorithm (Chepurin et al., 2004; submitted, see <http://www.atmos.umd.edu/~carton/bias>). The latest release of the MOM2-based moderate resolution version is beta23, for which a brief description is provided below.

We are now exploring an eddy-permitting reanalysis based on the Parallel Ocean Program POP-1.4 model with 40 levels in the vertical and a 0.4x0.25 degree displaced pole grid (25 km resolution in the western). The first version of this we will release is SODA1.2, a reanalysis driven by ERA-40 winds covering the period 1958-2001 (extended to the current year using available altimetry).

Data Access

The standard way to access the SODA reanalysis data sets is through the Live Access Server (LAS) mechanism (link to the left). The LAS makes it relatively easy to create basic graphics and to download subsets. The available data sets are described in detail in the History section. An expanding set of LAS sites offer many interesting data sets for comparison.

We also offer OPeNDAP formerly known as DODS (Distributed Oceanographic Data Server, link to the left) .

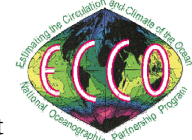
Ancillary Data Sets

We maintain a number of related data sets used in the reanalysis effort (e.g. quality-controlled hydrographic data, surface drifter data, winds) as well as datasets maintained for comparison purposes (e.g. gridded surface currents, sea level), and datasets collected as part of the assimilation procedure (e.g. observation-forecast and observation-analysis differences). Some of these we will make available through the LAS mechanism. Others will be available upon request.

<http://www.atmos.umd.edu/~ocean/>

Estimating the Circulation and Climate of the Ocean (ECCO)

ECCO was established in 1999 as part of the World Ocean Circulation Experiment (WOCE) with the goal of combining a general circulation model (GCM) with diverse observations in order to produce a quantitative depiction of the time-evolving global ocean state. The importance of such an endeavor is recognized by numerous national and international organizations, such as the WMO's World Climate Research Programme (WCRP) (and within it the Climate Variability and Predictability, CLIVAR) programme), UNESCO's Intergovernmental Oceanographic Commission (IOC) (and within it the Global Ocean Data Assimilation Experiment project, GODAE). These programs have all noted the necessity of synthesizing the diverse remotely-sensed and in-situ observations with known dynamics and thermodynamics through a GCM. ECCO was initially formed under the National Ocean Partnership Program (NOPP) with funding provided by the National Science Foundation (NSF), the National Aeronautics and Space Administration (NASA), and the Office of Naval Research (ONR). Funding now also comes from directly from NASA and the National Oceanic and Atmospheric Administration (NOAA).



The long term goal of ECCO is to provide a high resolution coupled ocean/sea-ice/biochemical (and ultimately consistent atmospheric) state estimate to a wide community. In contrast to so-called numerical weather prediction, these estimates will include the ocean's history as well as predictions. ECCO's efforts toward this goal now include several projects, each of which is bringing ECCO closer to its long term goal, while providing significant scientific contributions. These projects include efforts at ECCO-SIO, ECCO-JPL, ECCO-GODAE, ECCO2, and are further described here.

An early vision, ca. 1982 (taken from "A Celebration in Geophysics and Oceanography 1982. In Honor of Walter Munk on his 65th birthday." C. Garrett and C. Wunsch, Eds., SIO Reference Series 84-5, March 1984)

<http://www.ecco-group.org/>

Pathfinder SST

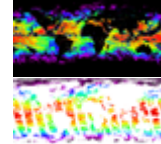
The NOAA/ NASA AVHRR Oceans Pathfinder sea surface temperature data are derived from the 5-channel Advanced Very High Resolution Radiometers (AVHRR) on board the NOAA -7, -9, -11, -14, -16 and -17 polar orbiting satellites. Daily, 8-day and monthly averaged data for both the ascending pass (daytime) and descending pass (nighttime) are available on equal-angle grids of 8192 pixels/360 degrees (nominally referred to as the 4km resolution), 4096 pixels/360 degrees (nominally referred to as the 9km resolution), 2048 pixels/360 degrees (nominally referred to as the 18km resolution), and 720 pixels/360 degrees (nominally referred to as the 54km resolution or 0.5 degree resolution).

AVHRR Oceans Pathfinder Global 4km Equal-Angle All SST V5 (NOAA) (#216)

Version 5.0 has significant improvements over 4.1. Version 5.0 is available globally at 4km and has improved ice and land masks, which allow for more retrievals along the coasts and in Lakes.

AVHRR Oceans Pathfinder Global Equal-Angle Best SST (NOAA/NASA) (#102)

Version 4.1. Data high in resolution contains 2 bands (sst,num of obs)

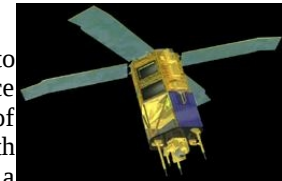


http://podaac.jpl.nasa.gov/cgi-bin/dcatalog/fam_summary.pl?sst+pfsst

Voir aussi: http://www.rsmas.miami.edu/groups/rsl/pathfinder/Processing/proc_index.html

Background of the SeaWiFS Project

The purpose of the Sea-viewing Wide Field-of-view Sensor (SeaWiFS) Project is to provide quantitative data on global ocean bio-optical properties to the Earth science community. Subtle changes in ocean color signify various types and quantities of marine phytoplankton (microscopic marine plants), the knowledge of which has both scientific and practical applications. The SeaWiFS Project will develop and operate a research data system that will process, calibrate, validate, archive and distribute data received from an Earth-orbiting ocean color sensor. A detailed description of the objectives, organization and operations as well as the current status of the SeaWiFS Project is available.



The SeaWiFS Mission is a part of NASA's Earth Science Enterprise, which is designed to look at our planet from space to better understand it as a system in both behavior and evolution.

The concentration of microscopic marine plants, called phytoplankton, can be derived from satellite observation and quantification of ocean color. This is due to the fact that the color in most of the world's oceans in the visible light region, (wavelengths of 400-700 nm) varies with the concentration of chlorophyll and other plant pigments present in the water, i.e., the more phytoplankton present, the greater the concentration of plant pigments and the greener the water.

Ocean color data have been deemed critical by the oceanographic community for the study of ocean primary production and global biogeochemistry.

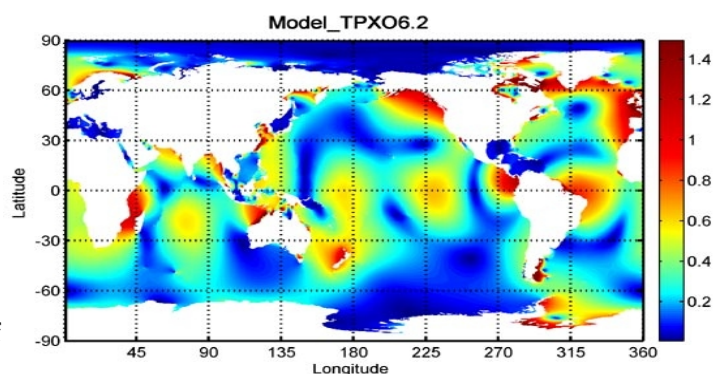
Since an orbiting sensor can view every square kilometer of cloud-free ocean every 48 hours, satellite-acquired ocean color data constitute a valuable tool for determining the abundance of ocean biota on a global scale and can be used to assess the ocean's role in the global carbon cycle and the exchange of other critical elements and gases between the atmosphere and the ocean. SeaWiFS will operate as a follow-on sensor to the Coastal Zone Color Scanner (CZCS), which ceased operations in 1986. In the first arrangement of its kind, the Government will procure space-based environmental remote sensing data for research purposes from a commercial operator. Orbital Sciences Corporation (OSC) will integrate SeaWiFS into its SeaStar spacecraft and will market the data for commercial and operational use following launch.

<http://oceancolor.gsfc.nasa.gov/SeaWiFS/BACKGROUND/>

TPXO6.2: Global Inverse Tide Model

TPXO6.2 is a medium-resolution, 1/4o x 1/4o global model developed by Gary Egbert and coworkers at Oregon State University. The model domain includes ocean cavities under the floating ice shelves.

The principal assimilated data set is TOPEX/Poseidon (T/P) satellite radar altimetry, between +/-66o latitude. However, the model also includes patches for various "coastal" oceans, including the Antarctic (assimilating tide gauge data used in circum-Antarctic inverse model CADA00.10 and the Ross Sea height-based inverse model Ross_Inv_2002, and the Arctic (cf. the Arctic inverse model AOTIM-5). TPXO6.2 is one of the most accurate global tidal solutions,



particularly for high latitudes since it utilizes recent Antarctic grounding line information and Antarctic and Arctic tide height data.

TPXO6.2 is distributed with a Matlab Graphical User Interface ("GUI") called "TMD" (the Tide Model Driver). TMD can be used to quickly access and browse the model, and to make tide height and current (velocity component) predictions. The TMD package also contains scripted functions for use in batch-mode Matlab processing. For an overview of the GUI and scripts, view or download the README PDF file. For FORTRAN access, please go to the Oregon State "OTIS" web page.

Please reference Egbert and Erofeeva (2002) for the use of TPXO6.2.

Egbert, G.D., and S.Y. Erofeeva, 2002: *Efficient inverse modeling of barotropic ocean tides*, *J. Atmos. Oceanic Technol.*, 19(2), 183-204.

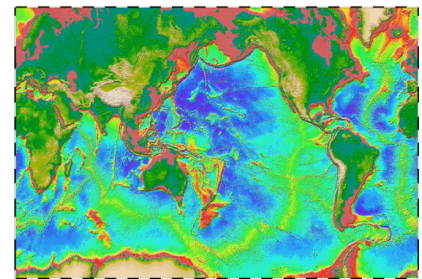
http://www.esr.org/polar_tide_models/Model_TPXO62.html

Global Bathymetric Prediction for Ocean Modelling and Marine Geophysics

David T. Sandwell and Walter H. F. Smith

ABSTRACT

We are constructing a complete bathymetric map of the oceans at a 3-10 km resolution by combining all of the available depth soundings collected over the past 30 years with high resolution marine gravity information provided by the Geosat, ERS-1/2, and Topex/Poseidon altimeters. Detailed bathymetry is essential for understanding physical oceanography and marine geophysics. Currents and tides are controlled by the overall shapes of the ocean basins as well as the smaller sharp ocean ridges and seamounts. Because erosion rates are low in the deep oceans, detailed bathymetry reveals the mantle convection patterns, the plate boundaries, the cooling/subsidence of the oceanic lithosphere, the oceanic plateaus, and the distribution of off-ridge volcanoes. Current global digital bathymetry maps (e.g. ETOPO-5) lack many important details such as a 400 km-long ridge that rises to within 135 m of sea level (Figure 1). Moreover, they are contaminated by long-wavelength errors (~2000 km) which prevent accurate identification of seafloor swells associated with mantle plumes [Smith, 1993].



The tasks include:

- * Accumulate all available depth soundings collected over the past 30 years. (funded by NSF)
- * Use the short wavelength (< 160 km) satellite gravity information to interpolate between sparse ship soundings.
- * Improve the resolution of the marine gravity field using enhanced estimates along repeat altimeter profiles together with the dense altimeter measurements.
- * Refine/Improve bathymetric predictions using the improved resolution gravity field and also by investigating computer-intensive methods for bathymetric prediction such as inverse theory.
- * Produce a Globe of the Earth similar to the globe of Venus prepared by the NASA Magellan investigation. This will also include the best available digital land data.

Smith, W. H. F. and D. T. Sandwell, *Bathymetric prediction from dense satellite altimetry and sparse shipboard bathymetry*, *J. Geophys. Res.*, 99, 21803-21824, 1994.

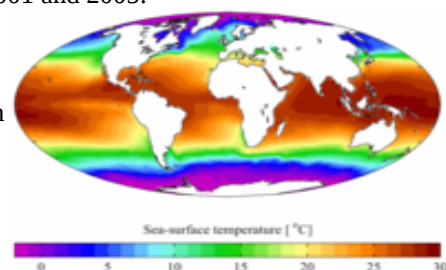
http://topex.ucsd.edu/marine_topo/text/topo.html

World Ocean Atlas

From Wikipedia

The World Ocean Atlas (WOA) is a data product of the Ocean Climate Laboratory of the National Oceanographic Data Center (USA). The WOA consists of a climatology of fields of in situ ocean properties for the World Ocean. It was first produced in 1994 (based on the earlier Climatological Atlas of the World Ocean[1]), with later editions at roughly four year intervals in 1998, 2001 and 2005.

The fields that make up the WOA dataset consist of objectively-analysed global grids at 1° spatial resolution. The fields are three-dimensional, and data are typically interpolated onto 33 standardised vertical intervals[2] from the surface (0 m) to the seafloor (5500 m). In terms of temporal resolution, averaged fields are produced for annual, seasonal and monthly time-scales. The WOA fields include ocean temperature, salinity, dissolved oxygen, apparent oxygen utilisation (AOU), percent oxygen saturation, phosphate, silicic acid, and nitrate. Early editions of the WOA additionally included fields such as mixed layer depth and sea surface height.



In addition to the averaged fields of ocean properties, the WOA also contains fields of statistical information concerning the constituent data that the averages were produced from. These include fields such as the number

of data points the average is derived from, their standard deviation and standard error. A lower horizontal resolution (5°) version of the WOA is also available.

Levitus, S. (1982) *Climatological Atlas of the World Ocean*, NOAA Professional Paper No. 13, pp. 191
<http://www.nodc.noaa.gov/OC5/indprod.html>

DASILVA Atlas of Surface Marine Data 1994

DASILVA Atlas of Surface Marine Data 1994: a five-volume atlas series depicting the seasonal and yearly variations of the surface marine atmosphere over the global oceans.

Documents

an outline showing sub-datasets of this dataset dataset documentation GCMDGlobal Change Master Directory entry

Datasets and variables

anomalies DASILVA SMD94 anomalies[tau3 fvcubed u3 latent3 ua v3 vq w3 taux3 evaprate netheat sst-sat sensib3 uq airdens va shortrad zdl cloud q vappress achi longrad precip qs-qa qsea sat ac qair slp ae precip6 sst rh]

climatology DASILVA SMD94 climatology[qs-qa w3 va taux3 evaprate latent3 precip6 v3 sst sst-sat airdens precip7 qair u3 slp ua tau3 netheat cloud qsea fvcubed shortrad vq zdl longrad rh sat sensib3 uq]

halfbyhalf DASILVA SMD94 halfbyhalf[climatology]

Grids

Time grid: /T (months since 1960-01-01) ordered (Jan 1945) to (Dec 1993) by 1. N= 588 pts :grid

Longitude grid: /X (degreeE) periodic (0.5E) to (0.5W) by 1. N= 360 pts :grid

Latitude grid: /Y (degreeN) ordered (89.5S) to (89.5N) by 1. N= 180 pts :grid

Other Info

GCMD_idNOAA_ATLAS_SURFACE_MARINE_DATA iridl:hasSemantics iridl:NOAA

References

A. da Silva, A. C. Young, and S. Levitus, Atlas of Surface Marine Data 1994, Volume 1: Algorithms and Procedures. NOAA Atlas NESDIS 6, U.S. Department of Commerce, Washington, D.C., 1994.

Last updated: Fri, 26 May 2006 20:03:09 GMT

Seaflux from Space is an information system designed to devise, display, and disseminate spacebased ocean-atmosphere fluxes of momentum, heat and water.

QuikSCAT winds

Seaflux provides, at near-real-time, twice-daily maps of ocean surface winds over global oceans, derived from the observations by the scatterometer SeaWinds on space mission QuikSCAT, of the National Aeronautics and Space Administration (NASA). From these maps, emerging marine weather systems can be monitored. Improved scientific products will be available in delayed mode.

The twice daily gridded wind u and v components over global ocean are available [online](#).

You may also browse wind field images created routinely to present: Quikscat wind streamlines for Pacific, Atlantic, and Indian Oceans Quikscat wind zonal, meridional components and wind speed over global ocean surface

The Physical Oceanography Distributed Active Archive Center (PO.DAAC) at the Jet Propulsion

Laboratory/Caltech distributes several sea surface temperature products (SST) at different spatial and temporal resolutions. The start and stop times for these data products varies. The following is a summary of information relevant to each product that should facilitate the choosing, by the user, of the appropriate product to meet their needs. Each summary contains relevant links for more detailed information on ordering, FTP access, andHeide Costantini <costant@c ppm.in2p3.fr> software.

TD 3: Benguela example: preparation of a model grid (*make_grid*)

Slides from a presentation by P. Penven given at the University of Toulon modelling summer school in 2006

http://lseet.univ-tln.fr/ecoletee/Documents/ROMS_TOOLS.pdf

2.6.1: Pre-processing data
2.6.1.1: ~/Roms_tools/Preprocessing_tools/make_grid.m

```

title = 'Southern Benguela Test Model';
config = 'Benguela';
%
% Grid dimensions:
lonmin = 12; % Minimum longitude [degree east]
lonmax = 21; % Maximum longitude [degree east]
latmin = -35.5; % Minimum latitude [degree north]
latmax = -26.5; % Maximum latitude [degree north]
%
% Grid resolution [degree]
di = 1/4;
%
% Number of vertical Levels
N = 32;
%
% Vertical grid parameters
theta_s = 6;
theta_b = 0;
hc = 10;
% Minimum depth [m]
hmin = 10;

```

File to edit: ~/Roms_tools/Run/romstools_param.m

```

%
% Topography netcdf file name (ETOPO 2)
topofile = './Topoetopo2.nc';
%
% Slope parameter (rigrad(h)/h) maximum value for topography smoothing
rtarget = 0.2;
%
% Open boundaries switches
obc = [1 1 1]; % open boundaries (I=open, [S E N W])
%
% Level of reference for geostrophy calculation
zref = -500;
%
% GSHHS user defined coastline (see m_map)
%
% XXX_f_mat Full resolution data
% XXX_h_mat High resolution data
% XXX_i_mat Intermediate resolution data
% XXX_l_mat Low resolution data
% XXX_c_mat Crude resolution data
%
coastfileplot = 'coastline_l.mat';

```

Horizontal grid parameters

Vertical grid parameters

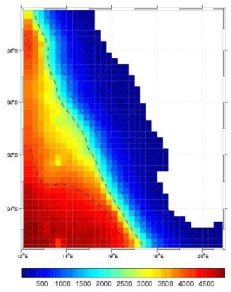
Open boundary conditions

Result of >>make_grid : roms_grd.nc

```

Making the grid: roms_grd.nc
Title: Southern Benguela Test Model
Resolution: 1/4 deg
Create the grid file...
LLm = 35
MMm = 42
Fill the grid file...
Compute the metrics...
Min dx=22.6187 km - Max dx=24.9066 km
Min dy=22.6473 km - Max dy=24.834 km
...

```



Open a Terminal window and go to the directory $\$Roms_tools/Run$
Launch Matlab using the command `matlab2012a -nodesktop`.

Open a second Terminal window and go to the directory $\$Roms_tools/Run$. There, open the file `start.m` using the command `gedit start.m`.
Save the file under the name `startup.m` so it will be automatically execute each time Matlab is launched (verify by quitting Matlab and relaunching it).

Open the file `romstools_param.m` and read its contents, trying to identify the different parameters for creating the grid.

In Matlab, execute the command `help romstools_param`; and see what text appears in the Matlab window. Compare this text with the contents of the file `romstools_param.m`.

Open the file `make_grid.m` in the directory $\$Roms_tools/Preprocessing_tools/$ and try to understand its contents. Then, execute it in Matlab.

During the execution chose the option `edit_mask` by replying with `y`.

In order to understand how these scripts work, please consult the ROMS_tools guide.

Save the figure in PNG and PDF format using the menu in the figure window: `File/Save As...`

Transcribe values of the grid size (variables `LLm` and `MMm`), of the spatial resolution (`Dxmin`, `Dymin`, `Dxmax`, `Dymax`) displayed in the Matlab Terminal window as they will be used in TD5.

TD 4: Benguela example: forcings and boundary conditions (make_forcing and make_clim)

Slides from a presentation by P. Penven given at the University of Toulon modelling summer school in 2006
http://seet.univ-tln.fr/ecoleete/Documents/ROMS_TOOLS.pdf

<p>2.6.1.2: ~/Roms_tools/Preprocessing_tools/make_forcing.m</p> <pre> %%% USERS DEFINED VARIABLES %% % % Title - Grid file name - Forcing file name % title='Forcing (COADS)'; grdname='roms_grd.nc'; frcname='roms_frc.nc'; % % Wind stress % taux_file='COADS05taux.cdf'; taux_name='taux'; tauy_file='COADS05tauy.cdf'; tauy_name='tauy'; smat = (15:30:345); % time for wind stress [days] smac = 360; % cycle for wind stress [days] ... </pre> <p>Result of >>make_forcing : roms_frc.nc</p>	<p>2.6.1.3: ~/Roms_tools/Preprocessing_tools/make_clim.m</p> <pre> % % Common parameters % romstools_param % % Title % title='Climatology'; % % Switches for selecting what to process (1=ON) % makeclim=1; %1: process boundary data makeoa=1; %1: process oa data makeini=1; %1: process initial data % % Grid file name - Climatology file name % % Initial file name - OA file name % grdname='roms_grd.nc'; frcname='roms_frc.nc'; climname='roms_clm.nc'; iname='roms_ini.nc'; oaname='roms_oa.nc'; </pre> <pre> % Day of initialisation % tini=0; % % Set times and cycles: monthly climatology for all data % time=[15:30:345]; % time cycle=360; % cycle % % Data climatologies file names: % temp_month_data : monthly temperature climatology temp_ann_data : annual temperature climatology salt_month_data : monthly salinity climatology salt_ann_data : annual salinity climatology % temp_month_data='WOA2001/temp_month.cdf'; temp_ann_data='WOA2001/temp_ann.cdf'; % % transform in-situ temperature to potential temperature % %1: transform in-situ temperature to potential temperature salt_month_data='WOA2001/salt_month.cdf'; salt_ann_data='WOA2001/salt_ann.cdf'; </pre>
<p>Result of >>make_clim : roms_in.nc AND roms_clm.nc</p>	

Open a Terminal window and go to the directory `$Roms_tools/Run`. Launch Matlab with the command `matlab2012a -nodesktop`.

Open the file `make_forcing.m` in the directory `Preprocessing_tools`, understand how it works and then execute it in Matlab.

Do the same with the file `make_clim.m`.
!!!IMPORTANT!!! in line 258 of the file `romstools_param.m` chose the climatology **WOA2009** instead of **CARS2009**

Save the figures produced in *pdf* format.

TD 5: Benguela example: monthly simulation and visualizing the results

Slides from a presentation by P. Penven given at the University of Toulon modelling summer school in 2006
http://seet.univ-tln.fr/ecoleete/Documents/ROMS_TOOLS.pdf

<p>2.6.2.2: ~/Roms_tools/Run/cppdefs.h</p> <p>cppdefs.h:</p> <ul style="list-style-type: none"> - Define CPP keys used by the C-preprocessor when compiling the model. - Reduce the code to its minimal size: fast compilation. - Avoid FORTRAN logical statements: efficient coding. <pre> /* Configuration Name */ #define BENGUELA /* Parallelization */ /* undef OPENMP /* undef MPI /* Embedding */ /* undef AGRIP /* Open Boundary Conditions */ /* undef TIDES /* define OBC_EAST /* define OBC_WEST /* define OBC_NORTH /* define OBC_SOUTH </pre>	<p>2.6.2: Preparing the model</p> <p>2.6.2.1: ~/Roms_tools/Run/param.h</p> <p>param.h defines the size of the arrays in ROMS:</p> <pre> #ifdef REGIONAL /* defined USWC3 parameter (LLmD=52, MMmD=126, N=40) l <- US_West grid15 L0 /* defined USWC1 parameter (LLmD=60, MMmD=96, N=40) l <- US_West grid15 L1 /* defined USWC2 parameter (LLmD=60, MMmD=120, N=40) l <- US_West grid15 L2 /* defined CANARY parameter (LLmD=97, MMmD=159, N=32) l <- Canary 25km /* defined FINISTERE parameter (LLmD=78, MMmD=100, N=16) l <- Finistere /* defined RIA parameter (LLmD=77, MMmD=96, N=28) l <- RIA /* defined PERU parameter (LLmD=39, MMmD=32, N=20) l <- Peru test /* defined SAFE parameter (LLmD=111, MMmD=96, N=32) l <- SAFE /* defined BENGUELA parameter (LLmD=35, MMmD=42, N=32) l <- Southern Benguela Test Model /* else parameter (LLmD=39, MMmD=32, N=20) #endif </pre>
<p>2.6.3: Compiling the model</p> <pre> ~/Roms_tools/Run/jobcomp % ./jobcomp </pre> <ul style="list-style-type: none"> - Automatic selection of compilation options according to the platform. - Set the library path. - Use Makefile: <ol style="list-style-type: none"> 1 C-preprocessing: file.F → file.f 2 Compiling: file.f → file.o 3 Link the file.o files and the libraries → executable: roms 	<p>2.6.4: Running the model</p> <p>2.6.4.1: ~/Roms_tools/Run/roms.in</p> <p>roms.in provides the run time parameters for ROMS:</p> <pre> #file: SOUTHERN BENGUELA TEST MODEL time_stepping: NTIMES dt[sec] NDTFAST NINFO 1080 2400 45 1 S-coord THETA_S THETA_B Hz (m) 6.040 0.040 10.040 grid: filename rome_grd.nc forcing: filename rome_frc.nc bulk_forcing: filename rome_bulk.nc climatology: filename rome_clm.nc boundary: filename rome_bry.nc initial: NRREC filename 1 restart: NRST, NRPRST / filename 1080 -1 rome_rst.nc history: LDEFHIS, NWRT, NRPFHIS / filename T 180 0 rome_his.nc averages: NTSAVG, NAVG, NRPFVAVG / filename 1 180 0 rome_avg.nc </pre> <p>Warning! These should be identical to the ones in romstools_param.m</p> <pre> primary_history_fields: zeta UBAR VBAR U V wrt(1:NT) T T T T 10^F auxiliary_history_fields: rho Omega W Akv Aks HBL Bostr F F F F F F F primary_averages: zeta UBAR VBAR U V wrt(1:NT) T T T T 10^T auxiliary_averages: rho Omega W Akv Aks HBL Bostr F T T F T F T T rho0: 1025 d0 lateral_visc: VISC2, VISC4 [m^2/sec for all] 0. 0. tracer_dif2: TNU2(1:NT) [m^2/sec for all] 10^0 d0 bottom_drag: RDRG [nvis], RDRG2, Zob [m], Cdb_min, Cdb_max 3.0d-04 0.d-3 0.d-3 1.d-4 1.d-1 gamma2: 1.d0 sponge: X_SPONGE [m], V_SPONGE [m^2/sec] 150 a3 1000. nudg_cof: TauT_in, TauT_out, TauM_in, TauM_out [days for all] 1. 360. 10. 360. </pre>
<pre> % ./roms.roms.in SOUTHERN BENGUELA TEST MODEL 1080 ntimes Total number of timesteps for 3D equations. 2400 00 dt Timestep [sec] for 3D equations 45 ndtfast Number of 2D timesteps within each 3D step. ... lmin lmax grdmn grdmax Cu_min Cu_max 10.00000 4938.03195 2.2632072E+05 2.48702715E+05 0.05337490 0.73143622 volume=1.020096442479109875000E+15 open_cross=6.628549416812264442444E+09 ... MAIN: started time-stepping. STEP time(DAYS) KINETIC_ENRG POTEN_ENRG TOTAL_ENRG NET_VOLUME trd 0 0.00000 0.000000000E+00 2.2861580E+01 2.2861580E+01 1.0200964E+15 0 1 0.02778 2.730158895E-05 2.2862365E+01 2.2862363E+01 1.0200962E+15 0 2 0.05556 5.998408374E-05 2.2863172E+01 2.2863232E+01 1.0200947E+15 0 3 0.08333 1.070776171E-04 2.2863976E+01 2.2864083E+01 1.0200783E+15 0 </pre> <p>Courant number: evaluation of the CFL criterion: dt/dt > fastest waves (here gravity waves), Cu_max < 1 !!!</p>	<p>2.6.5: Visualizing the results</p> <pre> % matlab >> roms_gui </pre>

Open the file *cppdefs.h* and check the configuration type and name as well as the location of the open boundaries (see TD3 and the file *Romstools_param.m*).

Open the file *param.h* and check the grid size (values *MMm*, *LLm* and *N*: see TD3).

Open the file *roms.in* and check the internal and external time steps (variables *dt[sec]* and *NDTFAST*) against the CFL condition and set the total time for the simulation (variable *NTIMES*) and the frequency with which to save the instantaneous and mean outputs (variables *NWRT* and *NAVG*).

Connect to the cluster and go to the working directory. Compile the model with *./jobcomp* and run it with the command *./roms & roms.in*.

On the local machine use Matlab to plot the results stored in the file *Run/ROMS_FILES* using the script *roms_gui.m*.

TD 6: Benguela example: multi-annual simulation and diagnostic tools

Slides from a presentation by P. Penven given at the University of Toulon modelling summer school in 2006
http://seet.univ-tln.fr/ecoleete/Documents/ROMS_TOOLS.pdf

2.7 – Long simulations	2.8 – Analysing the results
<pre>~/Roms_tools/Run/run_roms.csh In many studies, there is a need for long simulations: spin-up, statistical equilibrium. For regional models, 10 years appears to be a reasonable model length. In this case, it is easier to do several simulations of 1 month. run_roms.csh do it automatically: - It gets the grid file, the initial file and the boundary file. - It runs the model for 1 month. - It stores the files in a specific form: roms_avg_Y4M3.nc (i.e. march of year 4). - It replaces the initial file by the restart file (roms_rst.nc) which has been generated at the end of the month. - It relaunch the model for next month. set MODEL=roms # set SCRATCHDIR=pwd/SCRATCH # Time Schedule - TIME_SCHED=0 -> yearly files set INPUTDIR=pwd # TIME_SCHED=1 -> monthly files set MSSDIR=pwd # set MSSOUT=pwd # set TIME_SCHED=1 set CODFILE=roms # set AGRIF_FILE=AGRIF_FixedGrids.in # set NY_START=1 # set NY_END=10 # number total of grid levels # set NM_START=1 # set NM_END=12 set NLEVEL=1 To run in batch mode: %nohup ./run_roms.csh > exp1.out &</pre>	<p>A few tools can help to do basic statistical analysis of the model outputs:</p> <pre>roms_diags.m : Compute surface and volume averaged variables (temperature, salinity, ke). plot_diags.m : Plot the integrated variables (done after roms_diags.m). get_Mmean.m : Get the monthly mean. get_Smean.m : Get the seasonal mean (done after get_Mmean.m). get_Meddy.m : Get the monthly eddy kinetic energy (eke) (done after get_Smean.m). get_Seddy.m : Get the seasonal eke (done after get_Meddy.m). get_Meddy_ns.m : Get the monthly non-seasonal eke (done after get_Smean.m). get_Seddy_ns.m : Get the seasonal non-seasonal eke (done after get_Meddy_ns.m). roms_anim.m : Make a movie (ppm2fll must be installed).</pre>

Open the file *roms_inter.in* and, as you have done already for *roms.in* (see TD5), check the settings for the vertical grid, the internal and external time step (variables *dt[sec]* and *NDTFAST*) against the CFL condition and set the duration of the simulation (variable *NTIMES*) and the frequency which with to save the instantaneous and mean outputs (variables *NWRT* and *NAVG*). Also check that a restart file *roms_rst.nc* will be created at the end of the month (variable *NRST*).

Open the file *run_roms.csh* and check the time step, the number of days per simulated month, as well as the starting month and year and the end. The variable *NLEVEL* represents the number of nested grids, currently there is just one. To launch the multi-annual simulation in a way that it keeps running even if you log out of the cluster and also to optimize the cluster's computational capacity, you need to use the command *oarsub* and a new script that executes the script to launch the model *run_roms.csh*.

Create a new script with *gedit* *oarsub_run_roms.sh* and use the instructions on the right. Connect to the cluster, go to the working directory and make the script you just created executable with the command *chmod* *a+x* *oarsub_run_roms.sh*

```
#!/bin/sh
## Name of the job
#OAR -n MY_RUN_ROMS
## Number of processors to use and estimated time [h] for the job
#OAR -l core=1,walltime=200:00:00
## Name of the log file collecting all messages
#OAR -O result.%jobid%.log
## Name of the log file containing all errors
#OAR -E result.%jobid%.log
## email address
#OAR --notify mail:your.email@univ-amu.fr

export LD_LIBRARY_PATH=/export/apps/netcdf-4.1.2/lib:/export/apps/intel-v15/mpi/5.0.3.048/intel64/lib:/export/apps/intel-v15/lib/intel64/
ulimit -s unlimited

echo "Start of Job ${OAR_JOB_ID}"
cat $OAR_NODE_FILE

echo "**** Start ****"
date
./run_roms.csh
echo "**** Finish ****"
date

exit $?
```

Launch the model with the command *oarsub* *-S* *./oarsub_run_roms.sh*. Check that the model is running with the command *oarstat* (if the model is running correctly you get *state=R*). Check the file *oarsub_run_roms.log* and the contents of your *SCRATCH/* directory. If necessary, you can stop the model run with the command *oardel* *job-ID*. You can obtain the *job-ID* of your run using the command *oarstat*.

Edit the MATLAB script *roms_diags.m* (directory *Diagnostic_tools*) and insert at lines 44 and 46 the month and year at which the model has arrived, then execute the script. Use the script *plot_diags.m* to plot the averaged diagnostic variables.

TD 7: Implementation of a new model for a specific part of the ocean

!!! IMPORTANT !!!

Before starting a new configuration, save the existing configuration into a copy of the *Run* directory. Rename the *~/Run/* directory to *Run_Benguela*, create a new *Run* directory with subdirectories *~/Run/SCRATCH/* and *~/Run/ROMS_FILES*. Then copy all the files (not the directories!) from *Run_Benguela* to the new *Run* directory. Do this every time you start a new configuration.

Download the file www.mio.univ-amu.fr/~doglioli/OPB205/ad_tools.tar.gz which contains additional tools. Save the files in the *MYROMS* directory and unpack the archive using the command `tar xvfz ad_tools.tar.gz`. Then, add the following instructions to the end of the *startup.m* file:
`disp('add path of ad-tools');addpath(' ../ad_tools')`

Use the Matlab script *ad_findgeocoord.m* to set the coordinates of your model domain.


Modify the geographic coordinates and the horizontal resolution and check the number of vertical levels in *romstools_param.m* (to avoid too long simulations, limit the grid size to about 10^5 grid points) and launch *make_grid.m*. Using the option *editmask* modify the mask of the model. Be very careful with your open boundaries which need to be specified in *romstools_param.m* at line 218.

Once *make_grid.m* has executed successfully, save the figure showing the bathymetry and the grid layout to a file called *memo.mat* so you can check your setup against the CFL condition. Use these commands in Matlab:

```
LLm=L-1;MMm=M-1;
```

```
save memo LLm MMm dxmin dymin dxmax dymax grdname
```

Run the scripts *make_forcing.m* and *make_clim.m* to calculate the forcings and the initial and boundary conditions. Don't forget to save the figures so you can use the later in your report.

Modify the files *param.h* and *cppdefs.h* by defining a new key  the new configuration:



fichier *param.h* lines 102-105

<pre># elif defined BENGUELA_VHR parameter (LLm0=167, MMm0=170, N=32) # else parameter (LLm0=94, MMm0=81, N=40)</pre>	<pre># elif defined BENGUELA_VHR parameter (LLm0=167, MMm0=170, N=32) # elif defined NEW_NAME parameter (LLm0=VAL, MMm0=VAL, N=VAL) # else parameter (LLm0=94, MMm0=81, N=40)</pre>
---	---

fichier *cppdefs.h* line 49

<pre># define BENGUELA_LR</pre>	<pre># undef BENGUELA_LR # define NEW_NAME</pre>
---------------------------------	--

Still in *cppdefs.h*, activate/deactivate the keys corresponding to the open boundaries in the new configuration (lines 60-64).

Calculate the internal and external time steps for the simulation  using the CFL condition and the MATLAB script *ad_cfl.m*. Check the result by hand with the instructions  the course handout.

Edit the files *roms.in* and *roms_inter.in* setting the internal and external time steps (variables *dt[sec]* and *NDTFAST*) using the CFL condition, selecting the duration of the simulation (variable *NTIMES*) and the frequency with which to save the instantaneous and mean outputs (variables *NWRT* and *NAVG*), and the restart files to relaunch the model from (variable *NRST*).

Connect to the cluster and compile the model with *.jobcomp*.

After checking the file *run_roms.csh*, launch a multi-annual simulation with the command
`oarsub □ -S □ ./oarsub_run_roms.sh`

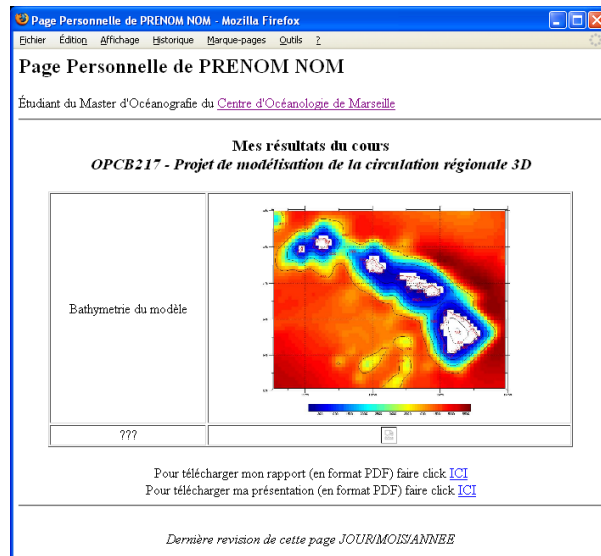
TD 8: Put the results of the new model online

To put the model results online, you need to edit the file *index.html* in the *\$YOURHOME/public_html/* directory. This file is your personal *homepage* and can be reached at <http://FIRSTNAME.NAME.etu.perso.luminy.univ-amu.fr> (if you haven't done so already, you need to activate your personal homepage via l'ENT).

Below, the source code for the page shown on the right, with the following HTML commands to insert:

- a link to an external web page or a local PDF document ``
- a table `<TABLE>` with rows `<TR>` and columns `<TD>`
- a JPEG figure ``
- a horizontal line `<HR>`
- a line break `
`
- to centre text `<CENTER>`

Each command ends with the same command preceded by /



```
<html>
<!-- Header -->
<head>
  <title>Page Personnelle de PRENOM NOM</title>
</head>
<!-- Main body of page -->
<body bgcolor=white>
<h2>Page Personnelle de PRENOM NOM</h2>
  &Eacute;tudiant du Master de Sciences de la Mer de l'
  <A href="http://www.pytheas.univ-amu.fr" target="new">Institut Pyth&eacute;as de
  Marseille</A>
  <BR><HR>
<CENTER>
  <H3>Mes r&eacute;sultats du cours <BR> <I>OPCB205 - Mod&eacute;lisation 3D
  Oc&eacute;anique</I></H3>
  <TABLE BORDER=1 width=90%>
    <TR>
      <TD width=30% ALIGN="CENTER"> Bathymetrie du mod&egrave;le</TD>
      <TD width=70% ALIGN="CENTER"><IMG width=90% src="./bathy.jpg"></TD>
    </TR>
    <TR>
      <TD width=30% ALIGN="CENTER"> ??? </TD>
      <TD width=70% ALIGN="CENTER"><IMG src="./nom_du_fichier2.jpg"></TD>
    </TR>
  </TABLE>
  <BR>
  Pour t&eacute;charger mon rapport (en format PDF) faire click <A
  href="./nom_du_fichier.pdf"> ICI </A>
  <BR>
  Pour t&eacute;charger ma pr&eacute;sentation (en format PDF) faire click <A
  href="./nom_du_fichier.pdf"> ICI </A>
  <BR>
  <HR><BR>
  <I>Derni&egrave;re r&eacute;vision de cette page JOUR/MOIS/ANNEE</I><BR>
</CENTER>
</body>
</html>
```


TD 9 (4h): Preparing and presenting your report

Following the typical outline of a research article (introduction, method, results, discussion, conclusion), the report on the work carried out should contain:

- a summary in French and an *abstract* in English
- a presentation of the study area and numerical model
- a description of the ROMS model and its implementation for the chosen oceanic region (resolution, size of the domain, initial and boundary conditions, parameters for the time integration, etc.) with a sufficient level of detail that will allow other modellers to reproduce your results;
- a presentation of the model results obtained (model diagnostics, seasonal circulations, temperature and salinity profiles, etc.);
- a comparison with literature data (see the paper provided to each of you and possibly others);
- a conclusion on the performance of the model with suggestions for possible improvements and perspectives.

This document must consist of 8 to 10 pages (from introduction to conclusion) using single line spacing and at least 2 cm margins. Use Times New Roman or Nimbus Roman font with a minimum size of 12pt. The number of pages of the report must be strictly adhered to. Bibliographical references and any digital programs should go into an appendix.

Layout instructions:

- on the first page (cover) must appear: year, name of the University, the Master program and of the University, LAST NAME and first name of the student, the title of the report, the date;
- all figures and tables must be numbered in order of appearance and have an informative, exhaustive, and accurate legend. They must be cited in the text using “Fig. NUMBER” or “Tab. NUMBER”;
- only equations that you cite in the text should be numbered in order of appearance. In the text refer to them as “Eq. NUMBER”;
- bibliographic references should be cited according to the “author-date” style, i.e., the name of the author, followed by the year of publication of the paper. If there are two authors, both names should be given; if there are more than 2 authors, only give the first name followed by “et al.”.

Examples:

- In-text citation: (Smith et al., 2011) showed ...”
- Reference citation: ...previous studies (Smith et al., 2011)...”
- If citing several articles published by the same author in the same year, add a letter

after the year depending on the order you are citing them in (or ordering them in your reference list, e.g., alphabetical based on co-authors, (Assassi et al., 2011a)... (Assassi et al., 2011b) ...”

- If you have more than one paper, cite them in chronological order: “Several studies (Kersalé 2009, Rainaud and Dutheil 2011) have confirmed...”

- all citations must appear in the reference list in alphabetical order, and if you have two papers by the same author in chronological order. Examples:

NAME, Initial(s). *Title of the article*. Name of the journal, date of publication, volume, number.

NAME, Initial(s). *Title of the contribution and/or name of website*. Date of publication. Date when the page was last update [date when you viewed the page]. Available at <URL address>

Your slide presentation should follow the same outline as your report and should last between 10 and 12 minutes. Times are strictly enforced.

On your first slide should appear: year, name of the UE, the Master program, and of the University, NAME and first name of the student, title of the report, the date;

Choose dark colours on a light background or light colours on a dark background.

Use a font size of at least 24 pt.

Don't forget to number your slides.

The report (use as file name *NameOfStudent_rapportOPB205.pdf*) and the presentation (use as file name *NameOfStudent_presentationOPB205.pdf*) should be put online on your personal home page in PDF format, respectively one week (report) and two days (presentation) before the exam.

The revised and corrected version of the report should be put online before the jury

!!! TIP !!!

Consider drafting your report and presentation using free software like TeXworks (a more user friendly interface for the TeX language) or LibreOffice, installed on the machines in the informatics room or downloadable from the addresses <http://www.tug.org/texworks/> and <http://www.libreoffice.org> for Windows and Linux. In addition, a report template in .tex format is available on the course web page.

These programs can save the files directly to PDF format. Otherwise, if using MSOffice (Word or PowerPoint) you can obtain a PDF by following the procedure at the following address:

<http://www.eleves.ens.fr/wintuteurs/tutorials/ms-office2ps/index.html>

Exam topics from previous years

Partial Examination

Wednesday 27 February 2008

DURATION: 1h30

*You are not allowed to use any document.
Each question is worth 4 points.
Answers must be as concise and precise as possible.*

- 1) Describe the meaning, from the physics point of view, of the different terms of the equations of motion

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho_o} \frac{\partial P}{\partial x} + f v - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z},$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho_o} \frac{\partial P}{\partial y} - f u - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z}.$$

- 2) Explain the differences between the turbulence closure schemes used in the SYMPHONIE, POM, and ROMS models.
- 3) Write down and explain the surface condition for the conservation equation for salt.
- 4) Explain the differences between the Arakawa B and C grids.
- 5) Explain how the *time splitting* technique works.

Partial Examination

Monday 15 March 2010

DURATION: 1h30

*You are not allowed to use any document.
Each question is worth 5 points.
Answers must be as concise and precise as possible.*

- 1) Write down and explain the conservation equations for temperature and salt.
- 2) Describe the meaning, from the physics point of view, of the different terms of the turbulent kinetic energy equation in the SYMPHONIE model

$$\frac{\partial Q}{\partial t} + \vec{v} \cdot \vec{\nabla} Q = A_z \underbrace{\left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 \right]}_{(1)} + \underbrace{\frac{g}{\rho_0} A_z \frac{\partial \rho}{\partial z}}_{(2)} + \underbrace{\frac{\partial}{\partial z} \left(A_z \frac{\partial Q}{\partial z} \right)}_{(3)} + \underbrace{\frac{\partial}{\partial x} \left(A_h \frac{\partial Q}{\partial x} \right) + \frac{\partial}{\partial y} \left(A_h \frac{\partial Q}{\partial y} \right)}_{(4)} - \varepsilon$$

- 3) Explain the purpose of the preceding equation in the system of equations solved by the SYMPHONIE model and analyse the differences with other approaches used in the POM and ROMS models.
- 4) Write down and explain the conditions at the sea bed for the equations of a thermo-hydrodynamic model of the ocean.

Partial Examination

Friday 18 March 2011

DURATION: 1h30

*You are not allowed to use any document.
Each question is worth 5 points.
Answers must be as concise and precise as possible.*

- 1) What is the “time factory”? What physical ideas are behind this concept?
- 2) Write down and explain the shallow water equations.
- 3) Describe the physical meaning of the different terms in the equation

$$\frac{\partial T}{\partial t} + \vec{v} \cdot \vec{\nabla} T = -\frac{\partial \overline{(T'u')}}{\partial x} - \frac{\partial \overline{(T'v')}}{\partial y} - \frac{\partial \overline{(T'w')}}{\partial z} + \frac{H_c}{\rho_o C_p} \frac{\partial I}{\partial z}$$

- 4) Explain the operation of the vertical mixing scheme called “KPP, K-profile parametrization” used in the ROMS model.

Partial Examination

RETAKE EXAM  March 2011

DURATION: 1h30

*You are not allowed to use any document.
Each question is worth 5 points.
Answers must be as concise and precise as possible.*

- 1) Write down and explain the conservation equations for temperature.
- 2) Describe the physical meaning of the different terms in the turbulent kinetic energy equation used in the SYMPHONIE model.
- 3) Explain the purpose of the preceding equation in the system of equations solved by the SYMPHONIE model and analyse the differences with the different types of approaches used in the POM model.
- 4) Write down and explain the conditions at the sea bed for the equations of a thermo-hydrodynamic model of the ocean.

Oral Examination

RETAKE EXAM 20 June 2012

DURATION: 45'

*You are not allowed to use any document.
Each question is worth 5 points.
Answers must be as concise and precise as possible.*

- 1) Describe the physical meaning of the different terms in the equations of motion below:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho_0} \frac{\partial P}{\partial x} + f v - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z},$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho_0} \frac{\partial P}{\partial y} - f u - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z}.$$

- 2) Explain the differences between the turbulence closure schemes used in the SYMPHONIE, POM, and ROMS models.
- 3) Explain the differences between Arakawa B and C grids.
- 4) Explain how the *time splitting* technique works.

Partial Examination

Friday 10 April 2015

DURATION: 1h30'

*You are not allowed to use any document.
Each question is worth 5 points.
Answers must be as concise and precise as possible.*

- 1) Describe the physical meaning of the different terms in the equations of motion below:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = -\frac{1}{\rho_0} \frac{\partial P}{\partial x} + f v - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z},$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = -\frac{1}{\rho_0} \frac{\partial P}{\partial y} - f u - \frac{\partial \overline{v'u'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z}.$$

- 2) Explain the CFL condition and give an example of its use.
- 3) Explain the Arakawa C grid.
- 4) What is the difference between a vertical z-coordinate grid and a vertical terrain following grid?

List of projects from previous years

- [Arancio, M.](#) (2008), *Modélisation de la Mer Adriatique avec ROMS*. [[PDF rapport](#); [PDF presentation](#)]
- [Assassi, C.](#) (2011), *Le courant Algérien*. [[PDF rapport](#); [PDF presentation](#)]
- [Ayache, M.](#) (2012), *Application du modèle ROMS dans le bassin Algérien*. [[PDF rapport](#); [PDF presentation](#)]
- Bachelier, C. (2007), *Modélisation de la Méditerranée Nord Occidentale* . [[PDF rapport](#)]
- [Beghoura, H.](#) (2015), *Étude de la dynamique océanique dans le bassin Algérien*. []
- Beloucif, N. (2007), *Modélisation de la circulation dans le Golfe du Lion : application du modèle Romstools et comparaison avec le modèle Symphonie* . [[PDF rapport](#)]
- [Bergthold, V.](#) (2015), *Modélisation de la circulation dans le Golfe du Mexique*. []
- [Blaizot, A.-C.](#) (2008), *Modélisation océanographique dans la Mer du Japon* . [[PDF rapport](#) ; [PDF presentation](#)]
- [Bourgeois, T.](#) (2012), *Application du modèle ROMS aux Îles Marquises*. [[PDF rapport](#); [PDF presentation](#)]
- [Bouvier, C.](#) (2013), *Modélisation de la circulation dans le Golfe du Lion*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Caffin, J.](#) (2014), *Application du modèle ROMS à la Manche*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Campbell, R.](#) (2007), *Introduction à ROMS: étude de la mer Ligure* . [[PDF rapport](#)]
- [Chauvin, V.](#) (2014), *Modélisation du Courant Est Australien*. []
- [Desbiolles, F.](#) (2010), *Comparaison entre deux schémas de fermeture de la turbulence avec ROMS en Adriatique* . [[PDF rapport](#) ; [PDF presentation](#)]
- [Dorel, M.](#) (2010), *Étude du bassin levatin avec le modèle ROMS* . [[PDF rapport](#) ; [PDF presentation](#)]
- [Dutheil, C.](#) (2011), *Modélisation du Courant Est Australien avec ROMS et son utilitaire ROMSTOOLS*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Giannesini, L.](#) (2011), *Etudes du courant Oyashio par le modèle ROMS*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Grangis, E.](#) (2011), *Application de ROMS au courant de Kuroshio avec ROMSTOOLS*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Guillon, B.](#) (2014), *Modélisation du bassin de Sorell et mise en évidence du courant de Zeehan par le modèle océanique régional ROMS*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Huguet, J.-R.](#) (2013), *Le courant Caraïbe*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Ienny, R.](#) (2015), *Modélisation de la circulation océanique autour du Plateau des Kerguelen*. []
- [José, Y.](#) (2008), *Application de ROMS-Romstools pour étudier les aspects hydrodynamiques du canal du Mozambique* . [[PDF rapport](#) ; [PDF presentation](#)]
- [Lacour, L.](#) (2012), *Modélisation de la circulation océanique en Mer d'Arabie*. [[PDF rapport](#); [PDF presentation](#)]
- [Le Clec'H, S.](#) (2013), *étude de la circulation océanique autour des îles Malouines en fonction de différents types de mélange verticale*. [[PDF rapport](#) ; [PDF presentation](#)]
- [Lucet, M.](#) (2012), *Modélisation de la circulation océanique de l'Argentine*. [[PDF rapport](#); [PDF presentation](#)]

[presentation\]](#)

[Kersalé, M.](#) (2009), *Modélisation de la circulation autour des îles Hawaii* . [[PDF rapport](#) ; [PDF presentation](#)]

[Martin, S.](#) (2015), *Sensibilité d'un modèle de l'upwelling californien à la résolution de la grille et du forçage du vent*. []

[Martinache, M.](#) (2013), *Modélisation du Courant d'Irmingier avec étude de la sensivité aux conditions aux bords ouverts*. [[PDF rapport](#) ; [PDF presentation](#)]

[Mereu, J.](#) (2014), *Etude de la circulation au large des côtes d'Amérique du sud et de l'impact de l'upwelling du Pérou*. [[PDF rapport](#) ; [PDF presentation](#)]

[Morisset, S.](#) (2007), *Modélisation du plateau de Kerguelen à l'aide du modèle RomsTools* . [[PDF rapport](#)]

[Navarro, J.](#) (2014), *Modélisation de la mer des Caraïbes avec le modèle ROMS*. [[PDF rapport](#) ; [PDF presentation](#)]

[Pagès, R.](#) (2015), *Étude de sensibilité aux modèles de fermeture turbulente dans le code ROMS. Application à l'Atlantique*. []

[Perrot, L.](#) (2011), *Introduction au modèle ROMS : Le courant des Aiguilles*. [[PDF rapport](#) ; [PDF presentation](#)]

[Rainaud, R.](#) (2011), *Modélisation 3D du Golfe du Lion*. [[PDF rapport](#) ; [PDF presentation](#)]

[Rammou, A.-M.](#) (2012), *Ocean Circulation Modeling: The Bay of Bengal*. [[PDF rapport](#); [PDF presentation](#)]

[Riousset, R.](#) (2015), *Modélisation de la circulation océanique dans la passe de Drake*. []

[Rocchietta, J.](#) (2014), *Modélisation de la circulation océanique de la Mer des Salomon*. [[PDF rapport](#) ; [PDF presentation](#)]

[Rousselet, L.](#) (2014), *Etude des caractéristiques du Pacifique Sud-Ouest avec le modèle ROMS*. [[PDF rapport](#) ; [PDF presentation](#)]

[Rimaud, J.](#) (2010), *Modélisation régionale 3D: application de ROMS à l'Atlantique Nord avec ROMSTOOLS* . [[PDF rapport](#) ; [PDF presentation](#)]

[Si-Bachir, M.-A.](#) (2012), *Application du model ROMS dans le golfe de Gascogne*. [[PDF rapport](#); [PDF presentation](#)]

[Simon, M.](#) (2008), *Introduction au modèle ROMS: Application de ce modèle au cas du Golfe du Mexique et étude du «Loop Current»* . [[PDF rapport](#) ; [PDF presentation](#)]

[Supplien, M.](#) (2011), *Modélisation du courant de Californie*. [[PDF rapport](#) ; [PDF presentation](#)]

[Susini, M.](#) (2014), *Modélisation et étude de l'upwelling de Somalie*. [[PDF rapport](#) ; [PDF presentation](#)]

[Tehami, I.](#) (2012), *Application du modèle ROMS à la Mer Rouge*. [[PDF rapport](#); [PDF presentation](#)]

[Yefsah, H.](#) (2015), *Étude de la circulation océanique au large du Pérou et rôle de l'upwelling*. []

[Zemenzer, S.](#) (2008), *Application du modèle ROMS dans la méditerranée occidentale; Circulation de LIW dans le bassin Algérien* . [[PDF rapport](#) ; [PDF presentation](#)]