

SIROCCO Training Course Tutorial

Session: **SIROCCO**, **S**tructured grid, **S** coordinate, ocean model

Toulouse March 2012

1	A short description of the S25 release of the SIROCCO ocean model.....	2
2	Untar the release2010.25_20120307.tar.gz archive	3
3	The S25 model directories environment	3
3.1	General environment	3
3.2	Create a “simulation” sub-level environment	3
3.3	A safe way to remove a “simulation” sub-level environment.....	4
4	Compilation.....	4
4.1	My first compilation.....	4
4.2	Import special routines in UDIR and compile again.....	5
5	Create the “notebooks” directory	5
5.1	The standard procedure	5
5.2	The particular case of this training course	5
6	First execution	6
6.1	Make the model properly stop at the end of the initialisation stage.....	6
6.2	Declare the grid dimensions in module_parameter.F90.....	6
7	Looking at the first outputs	7
7.1	<i>grid.nc</i>	7
7.2	<i>z2dv_OE_xxx.txt</i> and <i>z2dv_SN_xxx.txt</i>	7
7.3	The « <i>messages</i> » file	7
8	Vertical grid and notebook_vertcoord.....	7
8.1	S “attracted” coordinate system	8
8.2	The <i>H0</i> parameter.....	8
9	Rivers input	8
10	OGCM forcing	9
11	Meteorological forcing.....	10
11.1	Turn the meteorological forcing on.....	10
11.2	<i>notebook_graph</i> : ask to plot the air/sea fluxes terms	10
12	Wave forcing.....	10
12.1	Turn the wave forcing on	11
12.2	Makefile: select the “stokes” compilation key.....	11
13	A first simulation.....	11
13.1	The settings of <i>notebook_time</i>	11
13.2	Is my simulation properly working? First answers	12
13.3	Job interruption	12
14	Make the model work faster.....	12
14.1	Compilation option “O2”	12
14.2	<i>mpi</i> parallel computing.....	13
15	Offline simulations	14
15.1	Step 1: offline files production.....	14
15.2	Step 2: read the files	14
16	Passive tracers	14
16.1	The settings in <i>notebook_tracer</i>	15
16.2	Other settings (<i>notebook_graph</i>).....	16
17	Lagrangian drifters	16
17.1	Initialisation using the “notebook” key.....	16
17.2	Initialisation using the “fortran” key.....	17
18	I make my own netcdf files!.....	18
19	References	19

1 A short description of the S25 release of the SIROCCO ocean model

S is a Boussinesq hydrostatic ocean circulation model. Momentums and tracers are computed on an Arakawa curvilinear C-grid using an energy conserving finite difference method described in Marsaleix et al. (2008). The time stepping method consists of a Leap Frog scheme combined to a Laplacian filter (Marsaleix et al, 2012). A generalized terrain following coordinate preserves the vertical resolution within the bottom boundary layer and ensures the continuity of the fields near the bottom boundary. On the other hand, the well known "sigma coordinate errors" reported in Auclair et al (2000a) have been reduced through the use of a suitable pressure gradient scheme (Marsaleix et al, 2009, 2011). Radiative conditions are applied at the lateral open boundaries (Marsaleix et al, 2006). The large scale forcing terms, included in the radiation conditions formulation, are generally provided by the daily outputs of the MERCATOR system. The relevant questions related to the nested models are discussed in Estournel et al, 2009 and Auclair et al, 2006, 2000b. The high frequency barotropic forcing is provided by the TUGO tidal model. The astronomical tide potential has been implemented in the momentum equations according to Pairaud et al (2008). The air/sea fluxes are computed by the bulk formulae detailed in Estournel et al 2009. The river discharge is introduced through a lateral volume and salt conserving condition (Reffray et al, 2004). The turbulence closure, based on a TKE prognostic equation and two diagnostic turbulent length scales, covers a wide range of applications including convective processes (Bougeault and Lacarrère, 1989, Gaspar et al, 1990). Note that the S25 release now offers the possibility to use also the K-epsilon turbulent scheme (Michaud et al, submitted).

TABLE: Summary of the numerical schemes used in S25

Numerical method	C-grid, s coordinate, energy conserving	Marsaleix et al., 2008 http://dx.doi.org/10.1016/j.ocemod.2007.07.005
Time Stepping	Leap-Frog+Laplacian Filter	Marsaleix et al., 2012 http://dx.doi.org/10.1016/j.ocemod.2011.11.002
Pressure Gradient	Pressure Jacobian	Marsaleix et al., 2009 http://dx.doi.org/10.1016/j.ocemod.2009.06.011
Equation of state	McDougall 2003	Marsaleix et al., 2011 http://dx.doi.org/10.1016/j.ocemod.2011.07.004
Open boundary conditions	Radiation conditions	Marsaleix et al., 2006 http://dx.doi.org/10.1175/JTECH1930.1
Sea surface conditions	Bulk formulae, Craig & Banner TKE boundary conditions	Estournel et al, 2009 Ocean Science, 5, 73-90
Turbulence closure	Gaspar (JGR 1990) or K-epsilon	Michaud et al, submitted
Tides	Tide potential & TUGO nesting	Pairaud et al, 2008 http://dx.doi.org/10.1016/j.csr.2008.03.004
T,S advection	QUICK	
River input	Lateral condition	Estournel et al, 2001 http://dx.doi.org/10.1006/ecss.2000.0685

This model has been widely used to understand the Mediterranean Sea circulation at different spatial and time scales, leading to a series of processes oriented papers generally including a validation section based on in situ data. Among these processes, we may cite the river plume dynamics (Reffray et al, 2004), the dense water formation over the continental shelves (Estournel et al, 2005, Herrmann et al, 2008), the dense water cascading (Ulses et al, 2008), the eddy formation (Rubio et al, 2009, Hu et al 2009), the northern current (Bouffard et al, 2008). Most of the other applications concern the Bay of Biscay. This region is notably known for the strength of the tidal currents (Pairaud et al 2008) and the related internal waves exited at the continental shelf break (Pairaud et al, 2010). The general circulation along the slope (Herbert et al, 2011) is eventually dominated by the Navidad current. Errors developed by the model in relation with the uncertainties surrounding the wind field have been studied by Le Hénaff et al (2009).

2 Untar the release2010.25_20120307.tar.gz archive

The S25 can be downloaded from the SIROCCO web site:

http://sirocco.omp.obs-mip.fr/outils/Symphonie/Reserve/release2010.25_20120307.tar.gz

login: *sirocco* password: *Smodel*

If the web site is not available, you will find the archive in the */home/sirocco/session_prodelta_HELP* directory.

Copy the *release2010.25_20120307.tar.gz* archive in the *session_Smodel_student* directory and untar it:

```
cd /home/sirocco/session_Smodel_student
tar xvfz release2010.25_20120307.tar.gz
```

3 The S25 model directories environment

3.1 General environment

Let's have a look to the result of the previous action with the *ls* command:

```
ls
    GRAPHIQUES
    S2010.25
```

```
cd S2010.25
ls
```

Among the several files and directories contained in the S2010.25 directory, the relevant items (for the purpose of our training course) are:

```
CDIR_IFORT
configbox
mkregiondir
RDIR
SOURCES
compress_config
mkconfdir
removeconfdir
UDIR
```

The SOURCES directory contains the fortran sources of the S model. UDIR is the compilation directory. **Never touch to the routines of the SOURCES directory. Copy in the UDIR directory the routines that you want to modify.** The compilation with *makefile* will give the priority to these files. The result of the compilation process is sent to CDIR_IFORT (*.o* and *.f90* files) and to RDIR (*S25.exe* file), the execution directory. Output netcdf files will be sent to the GRAPHIQUES directory.

3.2 Create a “simulation” sub-level environment

Now, let's include a new level (your own simulation environment) in the current environment. This new level will be dedicated to a particular application, a special configuration, a new region, etc... Let's name our first application SIM1. Create the SIM1 environment with the *mkconfdir* command:

```
cd
cd session_Smodel_student/S2010.25
mkconfdir SIM1
```

What is the result? The answer is in directories CDIR_IFORT, RDIR, UDIR. Several distinct “simulation” environments can be present in the general environment, enabling a rational multi-use of the same code. Create a second “simulation” environment:

```
mkconfdir SIM2
```

Have a look to directories CDIR_IFORT, RDIR, UDIR.

Last, please create the PRODELTA environment (our favourite application today!).

```
mkconfdir PRODELTA
```

3.3 A safe way to remove a “simulation” sub-level environment

Avoid the proliferation of unused applications. Clean your work space once in a while. Remove quickly and safely the old directories (without collateral damaging) with the *removeconfdir* command. Let’s consider that SIM1 and SIM2 have become useless:

```
removeconfdir SIM1
removeconfdir SIM2
```

Then check the content of CDIR_IFORT, UDIR and RDIR directories.

4 Compilation

4.1 My first compilation

From now on, we work on the PRODELTA simulation. The compilation is done with the *makefile* file in the UDIR/PRODELTA directory.

```
cd /home/sirocco/session_Smodel_student/S2010.25/UDIR/PRODELTA
```

Different options are available in *makefile*. The *O2* option enables a faster execution of your code. On the other hand, bugs are better detected with special options like *check bounds* or *check all*. Always choose safety options when you start with a new case, or when you just modified something in the code. Edit *makefile* and go to line 40:

```
#
# Options de compilations: parametres F77FLAGS
#
#F77FLAGS= -check bounds -traceback
#F77FLAGS= -check all -traceback
#F77FLAGS= -O2 -traceback
#F77FLAGS=
#
```

The current option is *O2*. Replace by the *check bound* option. Execute the compilation command:

```
make ifortpar=on
```

When the compilation is done, explore your environment and find the execution file (S25.exe), as well as the .o and .f90 files.

4.2 Import special routines in UDIR and compile again

A few modified routines are necessary to continue with the PRODELTA case. I have put them for you in the /home/sirocco/data_prodelta_HELP directory. Copy *my_outputs.F90*, *obc_river.F90* and *river_upd.F90* in /home/sirocco/session_Smodel_student/UDIR/PRODELTA. Compile again:

```
make ifortpar=on
```

What is the difference with the previous compilation? Answer: *makefile* did not recompile all the code but only the 3 added routines.

What should I do if I want to recompile all the code? Answer: use the *clean* option and then recompile:

```
make ifortpar=on clean  
make ifortpar=on
```

5 Create the “notebooks” directory

5.1 The standard procedure

Now that the *model engine* space is correctly configured, let's create the input files environment. The latter mainly contains the notebook files (see the *notebook* document for a detailed explanation of each notebook) and the “sea-land grid mask” file. Use the command *mkregiondir*:

```
cd  
cd session_Smodel_student/S2010.25  
mkregiondir PRODELTA
```

A new directory has been created next to the *model engine* directory:

```
cd  
cd session_Smodel_student  
ls
```

```
GRAPHIQUES  
PRODELTA  
S2010.25
```

Check that a subdirectory PRODELTA has also been added in GRAPHIQUES. Check that the notebooks have been added in the main PRODELTA directory:

```
cd  
cd session_Smodel_student/PRODELTA  
ls
```

A default standard package of notebooks has been installed in PRODELTA.

5.2 The particular case of this training course

We now need to adapt this new space to the specificities of our new application. Normally, you should carefully check (and in some cases modify) the settings of each file of the “notebooks” directory. This step is eventually time consuming. As we don’t have so much time today, we will proceed differently. Exceptionally, you will import in `home/sirocco/session_Smodel_student/PRODELTA` the files already prepared for this training course.

First of all, import the `notebook_grid` file as well as the “sea-land grid mask” file (`bathycote_in.dat`) created during the “xscan grid generator” session:

```
cd
cd session_Smodel_student/PRODELTA
cp /home/sirocco/data_prodelta_HELP/notebook_grid .
cp /home/sirocco/data_prodelta_HELP/bathycote_in.dat .
```

and while at it, also import the files containing the list of the external forcing fields:

```
cp /home/sirocco/data_prodelta_HELP/*list* .
```

Make sure that the following files have been imported: `file_list_dir` `file_list_hs` `file_list_l` `file_list_taw` `file_list_uss` `list_ogcm_grid` `file_list_foc` `file_list_j` `file_list_t` `file_list_two` `file_list_wch` `list_ogcm_uvts`.

If you are not in a hurry, look at the detailed description of `notebook_grid` and `notebook_bathy` in the notebooks document.

6 First execution

6.1 Make the model properly stop at the end of the initialisation stage

At this stage, I will recommend you to open several windows on your screen, since we will regularly have to move from one directory to another. Let’s go first to the “notebooks” space. Import `notebook_time` from `/home/sirocco/data_prodelta_HELP` into the “notebooks” directory `/home/sirocco/session_Smodel_student`. Go there and edit `notebook_time`. Go to line 16:

```
-----
0 ! 0=Standard procedure -1= Stop the simulation at the end of the initialization
-----
```

The “stop or go” key is currently 0. Replace it by -1 so that the model will stop when the initialisation stage is finished.

Go to the `model engine` space, in the execution `RDIR/PRODELTA` directory. Start the simulation:

```
rm tmp/*      (clean the temporary files directory)
S25.exe
```

What happens? The simulation is prematurely interrupted, an “error” message appearing on your window. It tells you that the grid size declaration made in `module_parameters.F90` does not correspond to that specified in `notebook_grid`.

6.2 Declare the grid dimensions in `module_parameter.F90`

Where is `module_parameter.F90`? In the `SOURCES` directory? In the `UDIR/PRODELTA` directory? Which one should be modified? Go to the `UDIR/PRODELTA` directory. Edit `module_parameter.F90`. Goto line 16:

```
! _____!  
! Global domain dimensions:  
!   parameter(iglb=402,jglb=270,kmax=30)  
! _____!
```

Replace the current dimensions with the dimensions of `notebook_grid`. Now, the model should be entirely recompiled. So recompile all the code:

```
make ifortpar=on clean  
make ifortpar=on
```

Go to RDIR/PRODELTA and start the simulation again:

```
rm tmp/*  
S25.exe
```

7 Looking at the first outputs

7.1 *grid.nc*

A netcdf file, *grid.nc*, containing the principal grid fields has been created in the “temporary files” directory, RDIR/PRODELTA/tmp. Use `xscan` or `ncview` to visualize it. The PRODELTA horizontal grid is a curvilinear grid based on a MERCATOR projection combined to a rotation of the main axis in order to move the grid pole close to our region of interest. As a consequence, the horizontal resolution is highly variable within the numerical domain. The `prodelta` (the area in front of the Rhône river mouth) is particularly well resolved. Check this by yourself by plotting the `dx` and `dy` fields.

7.2 *z2dv_OE_xxx.txt* and *z2dv_SN_xxx.txt*

Ascii format files containing the level depth of grid points in 2D vertical along axis sections have been created in the “temporary files” directory, RDIR/PRODELTA/tmp. Use `gnuplot` to visualize them.

```
gnuplot  
plot "z2dv_OE_xxx.txt" u 1:2 w lp  
...
```

These files have been generated by the `sigma_levels_figure` subroutine (file `sigma_levels.F90`, line 808). The cross axis location is indicated lines 818 and 849. Copy `sigma_levels.F90` in directory UDIR/PRODELTA in order to select another location. Mind staying within the grid size declaration and don't forget to recompile!

7.3 The « *messages* » file

The “messages” text file is created in the “execution” directory during the initialisation stage of the simulation. This file contains useful information like the time step or constants involved in some of the model parametrizations. It also contains the grid indexes conversion of (latitude, longitude) locations listed in *notebook_atlas*. Edit *notebook_atlas* and note by yourself that it mostly contains a series of river mouth location. Then check that their corresponding grid indexes are present in the “messages” file. We will soon use this useful functionality.

8 Vertical grid and `notebook_vertcoord`

8.1 S “attracted” coordinate system

We will now use the visualisation of *z2dv_OE_xxx.txt* and *z2dv_SN_xxx.txt* to check the settings in *notebook_vertcoord*. This notebook is used to parameterize the *terrain-following* coordinate. Go to the “notebook” directory `/home/sirocco/session_Smodel_student/PRODELTA`. List the different examples of *notebook_vertcoord* that we have prepared for you.

```
ls notebook_vertcoord*
```

Notebook_vertcoord_case1: A regular sigma coordinate is applied when $H < H_0$ (H_0 defined line 5). A “s attracted” coordinate is used when $H > H_0$. Line 6 indicates that 80% of the vertical levels are *attracted* by the sea surface, the remaining 20% being *attracted* by the sea bottom.

Notebook_vertcoord_case2 is as *Notebook_vertcoord_case1* except that 100% of the vertical levels are attracted by the sea surface.

Select one of these 2 cases using *notebook_list*. **WARNING:** unlike the other notebooks, *notebook_list* is not located in the “notebooks” directory. You will find it in the “execution” directory, that is `RDIR/PRODELTA`. Go there, edit *notebook_list* and replace the current setting (*notebook_vertcoord*) by *notebook_vertcoord_case1*. Run the model. Create a new directory (for instance *tmp1*) where you will save the *z2_OE_xxx.txt* and *z2dv_SN_xxx.txt* files.

Now select *notebook_vertcoord_case2*. Run the model again. Save the *z2_OE_xxx.txt* and *z2dv_SN_xxx.txt* files in *tmp2*.

Compare the results of these 2 cases using gnuplot. What are the main differences? Answer: the bottom layer contains more vertical levels in Case1 than in Case2. A first impression is that Case1 could be a better option for sediment transport and Case2 a better option for air/sea fluxes driven processes.

8.2 The H_0 parameter

H_0 is a critical bathymetry value. A regular sigma distribution is applied when $H < H_0$. The “attracted” coordinate is used when $H > H_0$. The primary idea behind this method was to control the slope of the s levels over steep topography, in order to limit the well know sigma truncation errors on the pressure gradient force. A possible setting is consequently to choose a value that is representative of the shelf break ($H_0=150m$). In our case, strong topography slopes are rather found in the Rhône prodelta area, over 20m ($H_0=20m$).

Using a gnuplot visualisation of *z2_OE_xxx.txt* and *z2dv_SN_xxx.txt* files, test yourself the sensitivity of the vertical grid to this key parameter (see for instance *notebook_vertcoord_case3*).

On the other hand, a regular sigma coordinate is applied everywhere if the IGESIG parameter is equal to 0 (see for instance *notebook_vertcoord_case4*).

Last, make a choice for *notebook_vertcoord* (case1 is a good option) and let’s go to the next step.

9 Rivers input

Go to the “notebooks” directory `/home/sirocco/session_Smodel_student/PRODELTA`. Edit *notebook_rivers*. The number of river inputs is currently 0. Note that one river can have several inputs (here understand several grid points with a freshwater discharge boundary condition), especially if the river mouth is larger than the size of a cell box. This is the case here with the *Grand Rhone* river for which fives input grids are needed. The *Petit Rhone* river mouth is small compared to the size of a cell

grid box. It requires only one input grid point. The number of river inputs is consequently 5+1=6. Modify the first line of *notebook_rivers* in consequence.

On the other hand, note that the (i,j) location of the *Petit Rhone* is missing in *notebook_rivers*. Complete it, using the output file "messages" created by your previous run in the RDIR/PRODELTA directories. Somewhere in this file, you will find a first estimate of the (i,j) coordinates of the *Petit Rhone*, computed by S25.exe from *notebook_atlas* where the geographical coordinates (lat,lon) of the *Petit Rhone* have been mentioned. WARNING: these numbers have necessarily an "integer" type.

The grid point must be a "land" grid point contiguous to a "sea" point. Check that this constraint is verified thanks to the land sea mask and bathymetry file, PRODELTA/bathycote_in.dat:

```
cd /home/sirocco/session_Smodel/PRODELTA
```

Indeed, open it with the "vi" editor (*vi bathycote_in.dat*) having in mind that the (line, column) numbers appearing in the "vi" window actually correspond to the (i,j) indexes of the S model grid. The code mask is as follows: "land"=0, "sea"=1. Are our first estimate of the (i,j) *Petit Rhone* river mouth correct? What happens if these (i,j) indexes found in the "messages" file are entered in *notebook_rivers*. Try...

Does not work in fact...

Re-open /home/sirocco/session_Smodel/PRODELTA/bathycote_in.dat and find a better choice, i.e. consistent with your land sea mask. Complete *notebook_rivers*. Try again....

It works....

10 OGCM forcing

We will now introduce the OGCM forcing. The latter will provide the initial fields and drive our model at its open lateral boundaries. Go to the "notebooks" directory and edit *notebook_obcforcing*:

```
*****
Large scale OGCM files in the Open Boundary Conditions:
*****
Set the following parameter to 1 to add the OGCM forcing (0 otherwise):
0          ! 1=on 0=off          IOBC_AF
2          ! 2="online interpolation"
sympa      ! OGCM key
../../PRODELTA/list_ogcm_uvts
../../PRODELTA/list_ogcm_grid
none
6.         ! Periodicity of the ogcm fields          OBCINFO(1)
.....
Date of the first field in the lists:
Y M D H M S
2007 11 03 06 00 00
Add the Inverse Barometer if not contained in the ssh ogcm field:
1          ! 1=add the BI 0=otherwise          BI_ONOFF
```

The "on/off" key is currently 0. Switch on the OGCM forcing: replace 0 by 1. The next key indicates that the OGCM field will be interpolated on our model grid during the initialisation stage. The OGCM key is "sympa", meaning that we will use the outputs of the SIROCCO system. Note here that our interpolator is also compatible with the MERCATOR-OCEAN outputs. Two "list" files are then given. The access path (../../PRODELTA) is relative to the execution directory. The absolute path (/home/sirocco/session_Smodel_student/PRODELTA) could have been indifferently chosen here. Edit *list_ogcm_grid*. The content is the complete name (i.e. including the access path) of the OGCM netcdf grid file. Edit *list_ogcm_uvts*. The content is a list of netcdf files containing the principal physical

variables (U,V,T,S,ssh,Kz) of the OGCM outputs. The time sampling of the OGCM outputs is indicated in *notebook_obcforcing* (6 hours in our case) as well as the date (Y/M/D/H/M/S) of the first OGCM file in *list_ogcm_uvts*.

Before going ahead, check the OGCM fields. Plot one (or more) of them using *Xscan* or *Ncview*.

Then execute the model:

```
cd
cd session_Smodel_student/S2010.25/RDIR/PRODELTA
rm tmp/*
S25.exe
```

The simulation has been interrupted at the end of the initialisation step. A netcdf output file has been created just before the model stopping. Go to `/home/sirocco/session_Smodel/GRAPHIQUES/PRODELTA` and plot the result of the OGCM fields interpolation on our “prodelta” grid using *Xscan* or *Ncview*.

11 Meteorological forcing

11.1 Turn the meteorological forcing on

We will now introduce the meteorological forcing. The latter is provided by the 3 hourly outputs of a meteorological model. Practically the air/sea fluxes are computed using the Bulk formulae described in Estournel et al, 2009. Go to the “notebooks” directory and edit *notebook_airseaflux*. The first line of this file is:

```
0 ! 4=bulk method as in Estournel et al, Ocean Science 2009.
```

The “on/off” key is currently 0 (the meteorological forcing is turned off). Turn it on and select the Estournel et al (2009) bulk formulae by replacing 0 with 4:

```
4 ! 4=bulk method as in Estournel et al, Ocean Science 2009.
```

The following lines indicate the name and the location of the “list” files (containing the name and the location of the meteorological files), the time sampling (3 hours in our case) and the (y/m/d/h/m/s) date of the first file in each list.

11.2 Notebook_graph: ask to plot the air/sea fluxes terms

You can choose the variables to be stored in the output netcdf files with *notebook_graph*. Edit *notebook_graph* and select the longwave, shortwave and heat fluxes as well as the wind stress. The “yes or no” key is currently 0. Replacing 0 by 1 will insert the variables into the output netcdf files.

Execute the model. Go to the “outputs” directory: `/home/sirocco/session_Smodel_student/GRAPHIQUES/PRODELTA`. Plot the netcdf file with *xscan* or *ncview*.

12 Wave forcing

12.1 Turn the wave forcing on

Go to the “notebooks” directory and edit *notebook_wave*. The principle of this notebook is very similar to that of *notebook_obcforcing* and *notebook_airseaflux*. A “on/off” key is currently 0. Replacing 0 by 1 turns the wave forcing on: replace 0 by 1. Check the content of the 10 “list” files. Where are the ww3 netcdf files? (answer: /home/sirocco/data_ww3). Go there and plot one of these files with ncview. Note that each daily ww3 file contains 24 hourly fields.

12.2 Makefile: select the “stokes” compilation key

Execute the model. What happens? The simulation is prematurely interrupted. An error message tells you that the selection of the wave forcing requires a particular compilation key that is presently missing.

Go to the makefile directory:

```
cd /home/sirocco/session_Smodel_student/S2010.25/UDIR/PRODELTA
```

Edit *makefile* and select the “stokes” key:

```
# KEY1 = -Dstokes -Dsimple_precision  
KEY1 = -Dstokes  
# KEY1 = -Dzeroadvection  
# KEY1 =
```

Recompile all the code:

```
make ifortpar=on clean  
make ifortpar=on
```

Select the “wave” variables in *notebook_graph*. Execute the model. Go to the “output” directory. Plot the netcdf file in order to check the interpolation of the WW3 model on our prodelta S grid.

13 A first simulation

13.1 The settings of notebook_time

You are now ready for your first simulation. Edit *notebook_time*. Line 16: The “stop or go” key is currently -1. Replace it by 0 so that the model will not stop at the end of the initialisation stage.

```
-----  
0 ! 0=Standard procedure -1= Stop the simulation at the end of the initialization  
-----
```

Define the start and end dates of the simulation as follows:

```
-----  
Enter the time for the start and the end of the simulation:  
2007 11 18 12 00 00 ! Start time (yyyy mm dd hh mm ss)  
2007 11 25 12 00 00 ! End time (yyyy mm dd hh mm ss)  
-----
```

Check the time step of the barotropic and baroclinic mode. The barotropic time step is automatically computed using a robust CFL stability condition. You can nevertheless lower this time

step through the CFL_REDUCE factor. The optimal time step is for instance divided by 2 if CFL_REDUCE=0.5. The theoretical maximum value for CFL_REDUCE is 1. The baroclinic time step is defined by the ratio $\Delta t^{BAROCLINIC} / \Delta t^{BAROTROPIC}$. The higher the ratio, the larger the baroclinic time step. The current setting is 12, that is $\Delta t^{BAROCLINIC} = 12 \times \Delta t^{BAROTROPIC}$.

Go to the run directory:

```
cd
cd /home/sirocco/session_Smodel_student/S2010.25/RDIR/PRODELTA
```

Launch the mode:

```
rm tmp/*
S25.exe
```

13.2 Is my simulation properly working? First answers

Let the simulation move a little bit forward. Check that the calendar date is regularly updated in the execution window. You can also check that the time counter file, *kount*, is regularly updated. This file is created in the execution directory. Open it:

```
vi kount
```

Last, check the cpu and memory resources used by your computer. Open a new window and launch the *top* command:

```
top
```

Find your job (S25.exe) in the list.

13.3 Job interruption

For one reason or another, you may want to stop the model before its scheduled end in *notebook_time*. A radical way is to use the *top* command to identify a job number (say 1958) and cancel it with the *kill -9* command (example: *kill -9 1958*).

Another way is to turn on the “stop” key in the interactive file, *interrupteur*. Edit this file (you will find it in the execution directory). The current setting is:

```
0      0      0
```

The replacement of the first key by 1 will stop the model. Warning: *interrupteur* is a formatted file so don't change the number of characters between each key. Optionally, an output netcdf file will be created if the second key is set to 1. A restart file will be created if the third key is set to 1.

Using one of these different options, stop your simulation.

14 Make the model work faster

14.1 Compilation option “O2”

Now that you are sure that the model is correctly working, use an optimized compilation to make your code go faster. Go to the compilation directory and edit *makefile*. Line 40, select the *O2* option.

Recompile the code:

```
make ifortpar=on clean
make ifortpar=on
```

Re-launch the code and check that the progression of the simulation is actually faster.

14.2 mpi parallel computing

An efficient way to improve the cpu efficiency of your code is to use the mpi parallel computing functionality of the S model. Go to the compilation directory. Edit *module_parameter.F90*.

```
cd
cd /home/sirocco/session_Smodel_student/S2010.25/UDIR/PRODELTA
vi module_parameter.F90
```

Go to the “PARALLELISATION” section of the file (line 20). Select the number of sub-domains in each horizontal direction. Practically speaking, the computers of the present training course allow the use of 4 sub domains. A logical distribution is thus 2 sub-domains in each direction. We will now modify *module_parameter.F90* in consequence. The current setting is:

```
integer, parameter :: nbdom_imax=1,nbdom_jmax=1
```

Replace by:

```
integer, parameter :: nbdom_imax=2,nbdom_jmax=2
```

Warning: the computation of *imax* and *jmax* (the next lines in *module_parameter.F90*) has to remain consistent with their “integer” type declaration. In other words, *nbdom_imax* is necessarily a multiple of *iglb-2* and *nbdom_jmax* a multiple of *jglb-2*.

Recompile the model:

```
make ifortpar=on clean
make ifortpar=on
```

Launch the model:

```
rm tmp/*
time mpirun -np 4 S25.exe &
```

Note that in the previous command, the setting “4” corresponds to the number of sub-domains.

Check that the simulation is correctly progressing (and faster). Ask interactively the creation of an output netcdf file (without stopping the run) through the *interruptionneur* file using the following settings:

```
0 1 0
```

Warning: *interruptionneur* is a formatted file, so do not change the number of characters between each key. Go to the “output” directory (GRAPHIQUES/PRODELTA) and plot the last netcdf file. Check that every thing is OK.

Launch the *top* command in a new window. How many jobs in the list?

Eventually kill the current simulation if you are ready to move to the next section of this training course.

Plot the mpi sub-domains distribution using the grid.nc file created in the “temporary files” directory:

```
cd
cd session_Smodel_student/S2010.25/RDIR/PRODELTA/tmp
ncview grid.nc
```

Select the mpi_grid variables. You should see for zones. Check that the field value is either 0,1,2 or 3.

15 Offline simulations

15.1 Step 1: offline files production

Some applications like biogeochemical modelling, passive tracers, lagrangian drifters can be computed using an offline procedure (understand *in addition to the standard usual online procedure*). The offline procedure consists of feeding the biogeochemical, or passive tracers, or lagrangian drifters routines with physical fields to be read from the files created by a previous run. The interest is to save the cost of the physical equations computation. In order to use this option, we first need to perform the “physical” run that will create the offline physical files. Go to the “notebook” directory and edit *notebook_offline*.

The offline functions are turned off if the first key is 0. The current setting should be 1, meaning that the physical run will periodically store output netcdf files in the /home/sirocco/session_Smodel_student/PRODELTA/OFFLINE directory. Check the period sampling (in hours) of these files and the date of the first output.

WARNING: avoid backdating! In other words, be consistent with the departure date of the simulation: don’t order the first output file before the departure date.

If every thing is OK then go to the execution directory and launch the model:

```
rm tmp/*
time mpirun -np 4 S25.exe &
```

Check that the “OFFLINE” directory is periodically receiving the expected output files.

15.2 Step 2: read the files

Step 1 is finished. Go to the “offline” directory:

```
cd /home/sirocco/session_Smodel_student/PRODELTA/OFFLINE
```

Eventually using xscan or ncview, check that the output files have been created normally and look nice. Go to the “notebooks” directory. Edit *notebook_offline*. Turn the procedure key on “2”: the model will read the offline files instead of computing the physical equations. Don’t touch to the other settings!

WARNING: here again avoid backdating. Don’t start the simulation before the first available offline file. You must consequently change the departure date in *notebook_time*. A simple option (among many other choices) is to take the date of the first offline file as a departure time in *notebook_time*.

16 Passive tracers

16.1 The settings in notebook_tracer

The passive tracer simulation can be performed with the offline or the online mode. As we don't have so much time today, we will use the offline mode which is less cpu consuming. The two previous steps of section 15 must have been successfully performed. If not, use the online mode.

Edit *notebook_offline* and make sure that the procedure key is "2".

Edit *notebook_tracer*. The "on/off" key is currently off. Turn it on.

```
***** Do not touch this line please 1
0 ! Passive tracers model: 1=on 0=off          IMODELTRC
```

Define the number of passive tracer:

```
3 ! Number of independent 3D tracers          KBMAX
```

For each tracer, define the settling vertical velocity, having in mind that downward velocities are negative:

```
Settling velocity for each tracer in m/s (sign convention: <0):
-0.          ! Tracer n°1          WSED(:)
-0.          ! Tracer n°2          WSED(:)
-0.00001    ! Tracer n°3          WSED(:)
```

The next lines concern the source terms. Two kinds of sources are available: river input or accidental spillage. First of all, define the river inputs. The tracer concentration must be specified for all the rivers and for each tracer (even if the source is zero). Rivers are considered in the same order as in *notebook_rivers*:

```
River 1:
100. ! Tracer 1
0.   ! Tracer 2
0.   ! Tracer 3
River 2:
100. ! Tracer 1
0.   ! Tracer 2
0.   ! Tracer 3
...
...etc...
```

Then give the number of accidental spillages:

```
***** Do not touch this line please 4
Additional source points:
2          ! Sources number          KSOMAX
.....
```

For each source, give the horizontal location, using **grid indexes** or **latitude and longitude**. Then define the vertical location through a lower bound and an upper bound. They can be expressed in **meters** or with vertical **grid indexes**. Indicate which tracer is concerned by each source. Give the production term. The latter is homogeneous to the variation of tracer concentration per second. Last, give the beginning and the end dates of the production term.

```
source 1:
0 140 70      ! (a,b,c) if a=0 then b=i c=j / if a=1 then b=lat(°) c=lon(°)
1 -50. -0.    ! (a,b,c) if a=0 then b=kmin c=kmax / if a=1 then b=zmin c=zmax
2 3          ! tracers concerned (min & max number)
0.001       ! Production term (per second)
```



```

2007 11 19 00 00 00 Beginning of the production term
2007 11 23 00 00 00 End of the production term
source 2:
1 43.3 4.7 ! (a,b,c) if a=0 then b=i c=j / if a=1 then b=lat(°) c=lon(°)
0 15 20 ! (a,b,c) if a=0 then b=kmin c=kmax / if a=1 then b=zmin c=zmax
3 3 ! tracers concerned (min & max number)
0.002 ! Production term (per second)
2007 11 20 00 00 00 Beginning of the production term
2007 11 24 00 00 00 End of the production term

```

16.2 Other settings (notebook_graph)

Edit *notebook_graph* and ask for storing the passive tracers in the output netcdf files.

```

20 3D scalar variables on '_t' points      GRH_NB
1 Potential temperature                   1
1 Salinity                                2
0 Potential density                       3
1 Passive tracers                         4

```

Eventually, choose the “regular outputs” mode and the **periodicity** of the output files:

```

1 ! 1> regular in time 2> dates defined in notebook_dateoutput IDATE_OUTPUT
1 ! gives the periodicity (in days) if the previous line = 1

```

Edit *notebook_time* and make sure that the departure and end dates are compatible with the settings of *notebook_offline* and *notebook_tracer*.

Launch the model.

17 Lagrangian drifters

The lagrangian drifters routine can indifferently work in online or offline modes. Because of the lack of time, it is convenient to choose the offline mode today. Consequently, the settings in *notebook_offline* should be the same as in the previous section. Edit *notebook_drifter*.

The “on/off” key is currently off. Turn it **on**:

```

1 ! 1=Lagrangian drifters routine is on (=0 otherwise) ! DRIFTER_ONOFF

```

17.1 Initialisation using the “notebook” key

The initialisation mode key is either “notebook” or “fortran”. The “notebook” is easier for beginners but is not really handy if you plan to use thousand of drifters. Let’s define a few drifters with the “notebook” mode:

```

notebook ! initial mode: "notebook" or "fortran" ! DRIFTER_INIT_MODE

```

Trajectories will be stored in individual ascii files. In order to avoid huge files, you can request to store the drifters position once in a while. Define the **time sampling** (in seconds), noting that a negative number will cancel this ascii file option:

```

180. ! sampling (seconds) of the outputs (individual file, ascii format) (<0 if no ascii output)

```

Then give a list of drifter initial location. The location can be expressed in **latitude and longitude** or with **grid indexes**.

```
-- if "notebook" mode: give now the drifter list --- check that nbomax is big enough in parameter ----
43.3 4.7 -5. 0 ! arg4=1: i j k, arg4=0: lat lon z of the drifter
43.2 4.6 -10. 0 ! arg4=1: i j k, arg4=0: lat lon z of the drifter
193 112 20 1 ! arg4=1: i j k, arg4=0: lat lon z of the drifter
193 110 20 1 ! arg4=1: i j k, arg4=0: lat lon z of the drifter
193 108 20 1 ! arg4=1: i j k, arg4=0: lat lon z of the drifter
```

Check that the number of drifters does not exceed the declaration size parameter *nbomax* in module_parameters.F90. Modify *nbomax* if necessary (recompile in this case).

Launch the model.

The drifter ascii file (*drifter0000001*, *drifter0000002*, *drifter0000003*, *drifter0000004* *drifter0000005*) are created in the "temporary files" directory:

```
/home/sirocco/session_Smodel_student/S2010.25/RDIR/PRODEKTA/tmp
```

Use gnuplot to visualise them:

```
gnuplot
gnuplot> set xrange [3:6]
gnuplot> set yrange [42:44]
gnuplot> plot "/home/sirocco/data_prodelta_HELP/bathycote.source" u 2:1 w d, "drifter0000005" u 5:6 w d, "drifter0000004" u 5:6 w d, "drifter0000003" u 5:6 w d, "drifter0000002" u 5:6 w d, "drifter0000001" u 5:6 w d
```

17.2 Initialisation using the "fortran" key

Edit notebook_drifter. Turns the drifters **on**:

```
1 ! 1=Lagrangian drifters routine is on (=0 otherwise) ! DRIFTER_ONOFF
```

Request the "fortran" initialisation mode:

```
fortran ! initial mode: "notebook" or "fortran" ! DRIFTER_INIT_MODE
```

As we plan to use hundreds of drifters, do not request the "ascii files" option:

```
-999. ! sampling (seconds) of the outputs (individual file, ascii format) (<0 if no ascii output)
```

The visualisation remains possible but through the netcdf output files. Thus, edit *notebook_graph* and **select** the drifters in the list of available variables:

```
1 horizontal location of the lagrangian drifters 16
```

The "fortran" initialisation mode is based on fortran programming in the drifter.F90 routine. Never modify the routines of the SOURCES directory. Import them in the compilation directory first. Thus, go to the compilation directory and import *drifter.F90*:

```
cd
cd session_Smodel_student/S2010.25/UDIR/PRODELTA
cp ../../SOURCES/drifter.F90 .
```

Edit *drifter.F90* and go to the subroutine *drifter_initial* fortran line 426. The drifter arrays have two indexes (*N*,*Q*). *N* corresponds to a drifter counter. *Q*=1,2,3 corresponds to the *i,j,k* grid location and *Q*=4 corresponds to a drifter identifier (>0). In the following example, the drifter are initialised on **the surface level** (*k*=*kmax*) and horizontally regularly distributed (**one drifter every 3 grid points** in both horizontal directions). Note that *par%timax(1)* *par%tjmax(1)* are related to the mpi parallel computing: they are used here to convert local sub-domain indexes into absolute global grid indexes.

On the other hand `par%rank` is the sub-domain counter (actually the `mpi_grid` variable in the `grid.nc` file).

```

kbu=0
do j=1,jmax
do i=1,imax
if(mask_t(i,j,kmaxp1)==1) then
if(mod(i+par%timax(1),3)==0.and.mod(j+par%tjmax(1),3)==0) then
kbu=kbu+1
drifter_1(kbu,1)=real(i)+par%timax(1)
drifter_1(kbu,2)=real(j)+par%tjmax(1)
drifter_1(kbu,3)=real(kmax)
drifter_1(kbu,4)=(real(i)-1)*jmax+real(j)+(par%rank )*1000000
endif
endif
enddo
enddo

```

Suggestion: replace this “surface” initialisation by a bottom initialisation. Compile the model:

```
make ifortpar=on
```

Launch the model.

Plot the drifter variable of the output netcdf files. Colours are function of the drifter depth.

18 I make my own netcdf files!

OGCM, meteorological or wave external forcing terms are interpolated “on line” by the S model. The primary idea is to avoid preprocessing data treatments and thus to simplify the use of the model. As far as the OGCM forcing is concerned, *the interp_ogcm.F90* routine is compatible with the current MERCATOR-OCEAN files. What should I do if I use an other OGCM? You may adapt the existing `interp_ogcm.F90` routine or convert your files into a netcdf “MERCATOR like” format. Generating a netcdf file with a fortran routine is not difficult. Try with this simple example:

```

program toto
integer,parameter :: imax=100,jmax=50

real*4 lon_t(imax,jmax),lat_t(imax,jmax),var1_t(imax,jmax),var2_t(imax,jmax),filval

integer i,j,status,ncid,iddimx,iddimy,count_netcdfvar,varid(2),varid(4)

include 'netcdf.inc'

! Variables:

filval=-999.

do j=1,jmax
do i=1,imax
lon_t(i,j)= 10.*real(i)/real(imax)
lat_t(i,j)=40.+10.*real(j)/real(jmax)
var1_t(i,j)=real(i)*real(j)
var2_t(i,j)=real(i)+real(j)
enddo
enddo

! Step 1 ! Header of the file:

```

```

count_netcdfvar=0
status=nf_create('toto.nc',nf_clobber,ncid)
status=nf_def_dim(ncid,'imax_t',imax,iddimx)
status=nf_def_dim(ncid,'jmax_t',jmax,iddimy)
vardim(1)=iddimx ; vardim(2)=iddimy
count_netcdfvar=count_netcdfvar+1
status=nf_def_var(ncid,'lon_t',nf_real,2,vardim,varid(count_netcdfvar))
count_netcdfvar=count_netcdfvar+1
status=nf_def_var(ncid,'lat_t',nf_real,2,vardim,varid(count_netcdfvar))
count_netcdfvar=count_netcdfvar+1
status=nf_def_var(ncid,'var1_t',nf_real,2,vardim,varid(count_netcdfvar))
status=nf_put_att_real(ncid,varid(count_netcdfvar),'_FillValue',nf_real,1,filval)
count_netcdfvar=count_netcdfvar+1
status=nf_def_var(ncid,'var2_t',nf_real,2,vardim,varid(count_netcdfvar))
status=nf_put_att_real(ncid,varid(count_netcdfvar),'_FillValue',nf_real,1,filval)
status=nf_enddef(ncid)

```

! Step 2 ! Writing the data:

```

count_netcdfvar=0
count_netcdfvar=count_netcdfvar+1
status=nf_put_var_real(ncid,varid(count_netcdfvar),lon_t(1:imax,1:jmax))
count_netcdfvar=count_netcdfvar+1
status=nf_put_var_real(ncid,varid(count_netcdfvar),lat_t(1:imax,1:jmax))
count_netcdfvar=count_netcdfvar+1
status=nf_put_var_real(ncid,varid(count_netcdfvar),var1_t(1:imax,1:jmax))
count_netcdfvar=count_netcdfvar+1
status=nf_put_var_real(ncid,varid(count_netcdfvar),var2_t(1:imax,1:jmax))
status=nf_close(ncid)

```

end

This program *toto.F90* is compiled as follows:

```

ifort -I/home/sirocco/softs/ifort-icc/include/ toto.F90 /home/sirocco/softs/ifort-icc/lib/libnetcdf.a -
lnetcdf -lnetcdf -lgfortran -lstdc++ -o toto.exe

```

Launch *toto.exe* and then plot *toto.nc* using *ncview*. Look at the content of *toto.nc* using the *ncdump -h* command.

19 References

- Auclair F., Marsaleix P. and Estournel C., 2000a Sigma coordinate pressure gradient errors : evaluation and reduction by an inverse method. *Journal of Atmospheric and Oceanic Technology*, 17, 1348-1367. doi:10.1175/1520-0426(2000)017<1348:SCPGEE>2.0.CO;2
- Auclair F., Casitas S., Marsaleix P., 2000b. Application of an inverse method to coastal modelling. *Journal of Atmospheric and Oceanic Technology*. 17, 1368-1391. doi:10.1175/1520-0426(2000)017<1368:AOAIMT>2.0.CO;2
- Auclair F., Estournel C., Marsaleix P., Pairaud I. 2006. On coastal ocean embedded modeling. *Geophysical Research Letters*, 33, L14602. <http://dx.doi.org/10.1029/2006GL026099>
- Bouffard, J., S. Vignudelli, M. Herrmann, F. Lyard, P. Marsaleix, Y. Ménard, and P. Cipollini, 2008, Comparison of ocean dynamics with a regional circulation model and improved altimetry in the North-western Mediterranean. *Terrestrial, Atmospheric and Oceanic Sciences*, 19, 1-XXX, doi: 10.3319/TAO.2008.19.1-2.117(SA)

- Bougeault, P. and P. Lacarrère, 1989 : Parameterization of orography-induced turbulence in a meso-beta scale model, *Mon. Wea. Rev.*, 117, 1870-1888.
- Estournel C., Zervakis V., Marsaleix P., Papadopoulos A., Auclair F., Perivoliotis L., Tragou E., 2005. Dense water formation and cascading in the Gulf of Thermaikos (North Aegean) from observations and modelling, *Continental Shelf Research*, 25, 2366-2386 doi:10.1016/j.csr.2005.08.014
- Estournel C., Auclair F, Lux M., Nguyen C., Marsaleix P., 2009. "Scale oriented" embedded modeling of the North-Western Mediterranean in the frame of MFSTEP. *Ocean Science*, 5, 73-90
- Gaspar, P., Gregoris, Y., Lefevre, J.M., 1990. A simple eddy kinetic energy model for simulations of the oceanic vertical mixing: tests at station Papa and long-term upper ocean study site. *Journal of Geophysical Research* 95, 16179–16193.
- Herrmann M., Estournel C., Déqué M. , Marsaleix P., Sevaut F., Somot S., 2008, Dense water formation in the Gulf of Lions shelf: Impact of atmospheric interannual variability and climate change, *Continental shelf research*, 28, 2092-2112 doi:10.1016/j.csr.2008.03.003
- Herbert G., N. Ayoub, P. Marsaleix and F. Lyard, 2011. Signature of the coastal circulation variability in altimetric data in the southern Bay of Biscay during winter and fall 2004, *J. Mar. Syst.* , doi:10.1016/j.jmarsys.2011.03.004
- Hu Z.Y., Doglioli A.M., Petrenko A.A., Marsaleix P., Dekeyser I., 2009. Numerical simulations of eddies in the Gulf of Lion. *Ocean Modelling*, 28, 203-208, <http://dx.doi.org/10.1016/j.ocemod.2009.02.004>
- Le Hénaff M., De Mey P., Marsaleix P., 2009. Assessment of observational networks with the Representer Matrix Spectra method-application to a 3D coastal model of the Bay of Biscay. *Ocean Dynamics*, 59, 3-20 <http://dx.doi.org/10.1007/s10236-008-0144-7>
- Marsaleix P., Auclair F., Estournel C., 2006, Considerations on Open Boundary Conditions for Regional and Coastal Ocean Models. *Journal of Atmospheric and Oceanic Technology*, 23, 1604-1613, <http://dx.doi.org/10.1175/JTECH1930.1>
- Marsaleix P., Auclair F., Floor J. W., Herrmann M. J., Estournel C., Pairaud I., Ulses C., 2008. Energy conservation issues in sigma-coordinate free-surface ocean models. *Ocean Modelling*. 20, 61-89. <http://dx.doi.org/10.1016/j.ocemod.2007.07.005>
- Marsaleix P., Auclair F., Estournel C., 2009. Low-order pressure gradient schemes in sigma coordinate models: The seamount test revisited. *Ocean Modelling*, 30, 169-177. <http://dx.doi.org/10.1016/j.ocemod.2009.06.011>
- Marsaleix P., Auclair F., Estournel C., Nguyen C., Ulses C., 2011. An accurate implementation of the compressibility terms in the equation of state in a low order pressure gradient scheme for sigma coordinate ocean models. *Ocean Modelling*, 40, 1-13. <http://dx.doi.org/10.1016/j.ocemod.2011.07.004>
- Marsaleix P., Auclair F., Estournel C., Nguyen C., Ulses C., 2012. Alternatives to the Robert-Asselin filter. *Ocean Modelling*, 41, 53-66 <http://dx.doi.org/10.1016/j.ocemod.2011.11.002>
- Pairaud I. L., Lyard F., Auclair F., Letellier T., Marsaleix P., 2008, Dynamics of the semi-diurnal and quarter-diurnal internal tides in the Bay of Biscay. Part 1: Barotropic tides, *Continental Shelf Research*, 28, 1294-1315 doi:10.1016/j.csr.2008.03.004
- Pairaud I. L., Auclair F., Marsaleix P., Lyard F., Pichon A., 2010. Dynamics of the semi-diurnal and quarter-diurnal internal tides in the Bay of Biscay. Part 2: Baroclinic tides, *Continental Shelf Research*, 30, 253-269, <http://dx.doi.org/10.1016/j.csr.2009.10.008>
- Reffray G., Fraunié P. and P. Marsaleix, 2004 Secondary flows induced by wind forcing in the Rhône region of freshwater influence. *Ocean Dynamics*, 54, 179-196. doi:10.1007/s10236-003-0079-y
- Rubio A., Barnier B., Jorda G., Espino M., Marsaleix P., 2009 Origin and dynamics of mesoscale eddies in the Catalan Sea (NW Mediterranean): Insight from a numerical model study. *Journal of Geophysical Research*, 114, C06009, doi:10.1029/2007JC004245

Ulses, C., C. Estournel, P. Puig, X. Durrieu de Madron, and P. Marsaleix, 2008, Dense shelf water cascading in the northwestern Mediterranean during the cold winter 2005. Quantification of the export through the Gulf of Lion and the Catalan margin. *Geophysical Research Letters* 35, L07610 doi:10.1029/2008GL033257