SYMPHONIE short notice

Short description of the S.26 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 S.26

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, 2012 Ocean Sciences

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 excited 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).

What do you need to use symphonie

Symphonie is an open-source software programmed in Fortran 90 using MPI, NETCDF and PNETCDF libraries. .
We are using a regular basis, Intel and Gnu fortran compilers and both are working correctly.

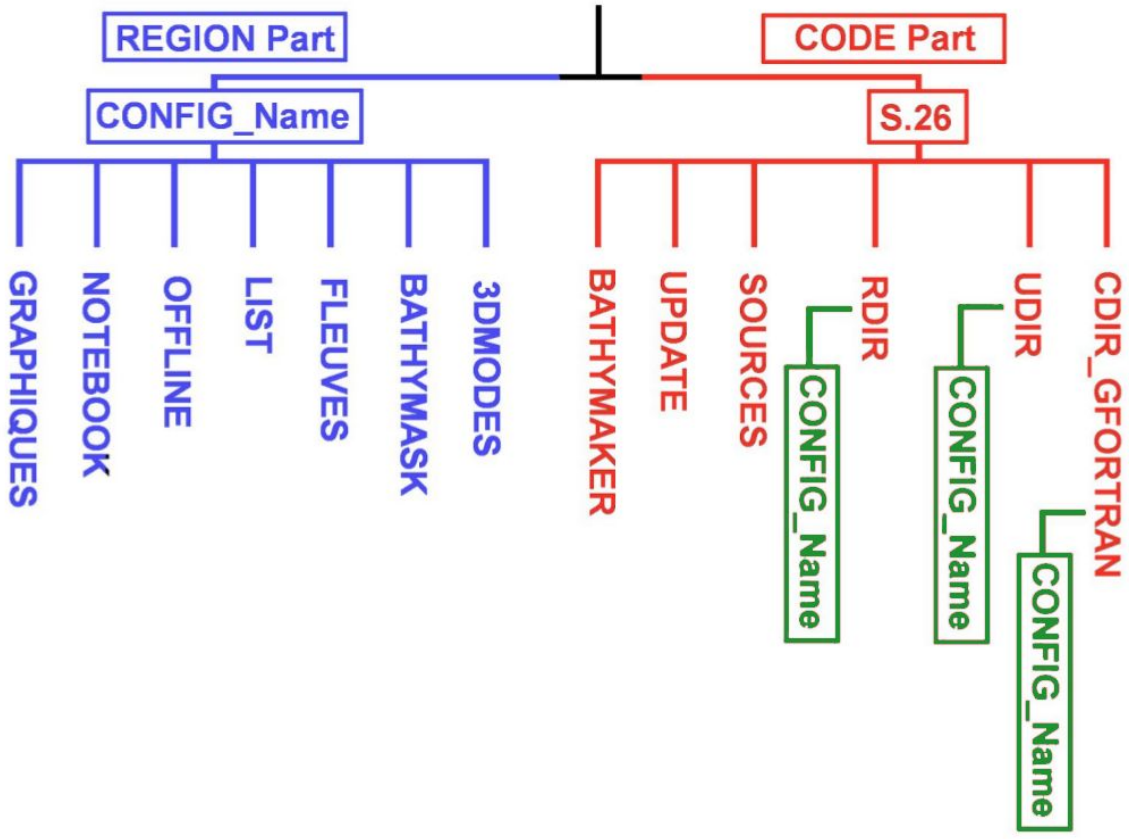
So you will need to install those software to be able to run Symphonie:

- A fortran compiler : Ifort or Gfortran
- A MPI compiler : IntelMPI, OpenMPI, MPICH
- Netcdf 4 version
- Pnetcdf (compiled of course with the mpi you had installed already).

The S.26 model directories environment

You have first download the tar file Symphonie.tgz from the website.
You will have to extract this archive, let says you done it from your “home” (/home/users/) directory.
Then you will create a SYMPHONIE directory and several subdirectories

Here is a global description of a typical SYMPHONIE environment tree as seen from the root ; **/work/users/SYMPHONIE** in the present case :
from the root directory (not exhaustively):
/work/users/SYMPHONIE/----------|



In a first approach, you can see that the code is splitted in two sections : the Region part and the Code part.

The **Region** part is the place where you are going to give the material for your simulation, by material we mean : the geographical zone, set the time, the different forcing sources, the graphical output of your simulation and a couple of less obvious parameters.

The **Code** part is the engine part, it is the place where you will find :

- the SOURCES directory : it is the location you will find the sources code of the program,

- The UDIR/CONFIG_NAME directory : here you will put the specific routines you may want to modify/adjust (or maybe even create). It is also the place where you are going to compile the fortran code in order to **generate** the **executable program**.
- The RDIR/CONFIG_NAME directory : It is the directory where you will find this **executable program** you have compiled from the UDIR directory. It is also the location where you will launch your simulation and set the dashboard to control it.
- The CDIR_GFORTRAN directory : It is the place where the sources code from SOURCES routines and the modified routines from UDIR are being compiled to create the executable program. This place can be ignored by beginners.

We continue by going under the S.26 directory.
Among the several files and directories contained in the S.26 directory, the first relevant items are:

CDIR_GFORTRAN
configbox
mkregiondir
RDIR
SOURCES
compress_config
mkconfdir
removeconfdir
UDIR

The SOURCES directory contains the fortran sources of the SYMPHONIE 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_GFORTRAN (*.o* and *.f90* files) and to RDIR (*S.26.exe* file), the execution directory. During the execution of the run, output netcdf files will be sent to the GRAPHIQUES directory in the **Region Part** of the code tree.

Create a "simulation" sub-level environment

To start, 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 this first application SIM1. You can create the SIM1 environment with the *mkconfdir* command (the command ***mkconfdir*** is under the directory S.26) :

mkconfdir SIM1

The result of this operation is to create several subdirectories under : CDIR_GFORTRAN, RDIR and UDIR directories and create altered copies of files from the configbox to the newly created subdirectories linked to the new simulation environment **SIM1**.
Noticed that 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:

A safe way to remove a "simulation" sub-level environment

To Avoid the proliferation of unused applications, clean your workspace once in awhile. Remove quickly and safely the old directories (without collateral damaging) with the *removeconfdir* command. So to remove the simulation SIM1 you just need to use the command :

removeconfdir SIM1

Compilation

How to : compilation

From now on, we work on the SIM1 simulation. The compilation is done with the *makefile* file in the *UDIR/SIM1* directory.

In that directory you will find a file : ***makefile***
makefile gives the instructions to the fortran compiler in order to produce the executable program.
This "***makefile***" deals with a file : ***makefile.inc*** for user and system dependent information. This means we have to know :

- the compiler
- the name of the config (SIM1, etc...)
- the optimisation option for compiler
- and the location of netcdf and pnetcdf libraries and include files.

The standard procedure explained below will help you to be able to set the compilation properly for any kind of linux system (as long as they were correctly set with the good libraries).

- The compiler for intelmpi is for mpif90 (mpif90 -show => to see the compiler)
- The name of the config here is SIM1
- For the optimisation options the default parameters are set for a intel fortran compiler. You could found some example of others parameters in the S.26/configbox directory (*makefile.inc_gnu_gen for gfortran example)* .
- Set the LIB and NETINC : you can get information for libraries and includes for netcdf by using the nf-config command :
  - nf-config --fflags => the NETINC for netcdf
  - nf-config --flibs => the LIB for netcdf

  But for the pnetcdf part you will need to know where you had it installed

Now how to compile SYMPHONIE, nothing easier if everything else is set properly, just type the command :

make

When the compilation is finished correctly, the executable file **S26.exe** must have been created under the **/home/users/Symphonie/S.26/RDIR/SIM1/** directory.

The code is run from those “RDIR” directory (also named “Run directory”). To launch it, a simple way would be using the “**s**” bash script you will found in the RDIR/SIM1 directory. The script need to give the number of processors you want to use. For running the code on 2 processors you can simply use the command :

```
s  2
```

At this time the code will start to read the **notebook_list.f** nameliste located in the local directory.
This namelist is the dashboard of your simulation. All the informations for the code to run properly will be read from this notebook_list.f .

This nameliste gives the location and name of other namelistes used by the code. Those namelist are stored in a separate directory which is given as the first parameter of the notebook_list.f :

```
&notebook_list
!main points to check before starting the simulation:
!https://docs.google.com/document/d/1TRF8uYjsVen8EiY0rERHw3YxiofZmpZ6cH_sEhWiMwA/edit?usp=sharing

! DIRECTORY
directory='../../../SIM1/NOTEBOOK/'              ! Directory of the notebooks

! TIME
nomfichier(1)='notebook_time.f'                  ! Departure/End time of the runs, time steps,...

! GRID
nomfichier(2) ='notebook_grid.f'                 ! Dimensions, projection, mpi, ...
nomfichier(3) ='notebook_bathy.f'                ! Land/Sea mask, bathymetry, wetdrying,...
nomfichier(13)='notebook_vertcoord.f'            ! Vertical coordinate, sigma stretching,...

! FORCING
nomfichier(4) ='notebook_rivers'                         ! RIVERS
nomfichier(7)='notebook_airseaflux_ecmwf_s26.f'          ! METEO
nomfichier(8) ='notebook_obcforcing_nemo.f'  ! OGCM
nomfichier(22)='notebook_wave.f'                         ! WAVES
nomfichier(11)='notebook_tide'                           ! TIDES

! I/O
nomfichier(20)='notebook_offline.f'              ! Offline files
nomfichier(21)='notebook_graph'                  ! Outputs files for graph

! PHYSIC
nomfichier(5)='notebook_advection.f'             ! Advection schemes
nomfichier(9)='notebook_visco.f'                 ! Turbulence schemes
nomfichier(15)='notebook_optical.f'              ! Light attenuation
nomfichier(17)='notebook_eqstate.f'              ! Equations of state
nomfichier(14)='notebook_spongelayer.f'          ! OBC schemes, nudging layer
nomfichier(34)='notebook_nh.f'                   ! m0v0m

! TRACERS
nomfichier(10)='notebook_tracer.f'               ! Eulerian (passive)
nomfichier(12)='notebook_bio'                    ! Eulerian (bio)
nomfichier(16)='notebook_drifter'                ! Lagrangian

! BIO
nomfichier(23)='notebook_light'
nomfichier(24)='notebook_zooplankton'
nomfichier(25)='notebook_phytoplankton'
nomfichier(26)='notebook_bacteria'
```

```
nomfichier(27)='notebook_remineralisation'
nomfichier(28)='notebook_initpelagic'
nomfichier(29)='notebook_biobcforcing'
nomfichier(30)='notebook_benthic'
nomfichier(31)='notebook_oxygen'

! OTHERS
nomfichier(18)='notebook_dateoutput'
nomfichier(19)='notebook_atlas'
nomfichier(33)='notebook_sedim.f'

! OASIS COUPLER
nomfichier(32)='notebook_oasis'

/
```

All the namelists above are stored in directory “DIRECTORY” by default as the relative path to the “Region Part” directory.

Create the “notebooks” directory : The Region Part

Now that the *model engine (or Code part)* 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. From the S.26 directory, use the command *mkregiondir*:

mkregiondir SIM1

A new directory has been created next to the *S.26* directory:

SIM1 S.26 etc….

Go under the newly created *SIM1* directory and list the content :

BATHYMASK GRAPHIQUES LIST NOTEBOOK OFFLINE

The NOTEBOOK directory is containing all the namelists and notebooks declared in the global nameliste : **notebook_list.f**, you can found under the “RUN directory” (here RDIR/SIM1).