PISCES Training Autumn School 2020

Advanced training session, November 2020

Olivier Aumont, Christian Éthe, Vincent Echevin, Coralie Perruche, Renaud Person

---

The objective of this hands-on session is to explore the different functionalities and adjustment possibilities with low-cost configurations of PISCES, the biogeochemical modeling component implemented in the two ocean platforms NEMO and CROCO. A presentation will be given to show the modeling choices, the code architecture, and the main features of PISCES. You will first explore the sensitivity of the PISCES model to a set of parameters with NEMO-PISCES in a tuning session. Then, you will learn the different steps to add a new biogeochemical tracer in the ocean modeling platform of your choice, NEMO or CROCO, in a more autonomous session. Finally, an introduction to diagenesis and its modeling module will be given, followed by a relatively short practical session. At the end of each session, there will be an hour or so to discuss the blocking points and answer any questions that may have arisen. A Discord support group has been set up to accompany you during the different sessions. An invitation to join this group will be sent to you. This being the first practical session organized at the level of the PISCES community, any positive or negative feedback is welcome to improve the content and form of this training. We would like to thank you warmly for being our experimental guinea pigs.
The PISCO training team.

Accounts must first be created on NEMO & CROCO's websites in order to download the code versions used in this practical work.

NEMO : Bypass the following below since NEMO doesn't need a password anymore

- go to **https://forge.ipsl.jussieu.fr/nemo/wiki/Users**
- click on **register** in the right corner
- Fill the surname you want, the password and the email address
- click on **create account** on the bottom

CROCO :

- Login to https://gitlab.inria.fr/croco-ocean/croco (request an access to the project)
- generate a ssh key to get the code (if this does not exists in **$HOME/.ssh/id_rsa.pub**)
- **ssh-keygen -t rsa**
- login to https://gitlab.inria.fr/croco-ocean/croco
- go to **profile** (upper right corner)
- select **settings**
- left column click on **SSH Keys**
- copy/paste the ssh key generated
- click on **add key**

---

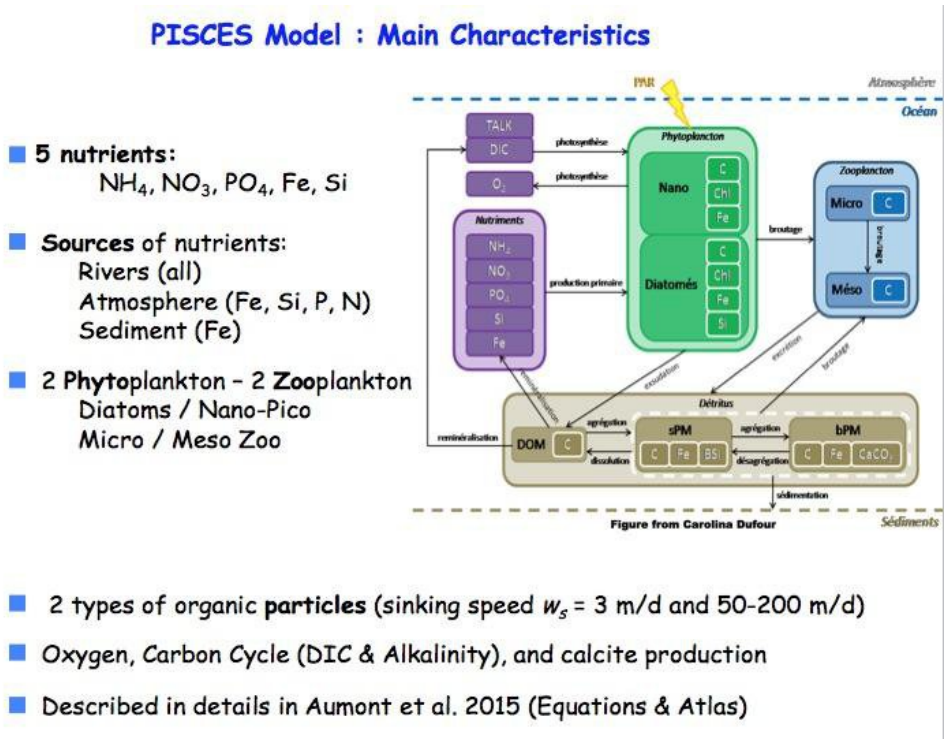
1. Brief description of PISCES

PISCES is constructed on the assumption that phytoplankton growth is directly limited by the external availability in nutrients [Monod, 1942]. This choice was mostly dictated by the computing cost as PISCES has been designed to suit a wide range of temporal and spatial scales, including quasi steady state simulations on the global scale.

The model has 24 compartments (Figure 1). Phytoplankton growth can be limited by five different nutrients: nitrate, ammonium, phosphate, silicate and iron. Four living pools are represented: two phytoplankton size classes/groups (nanophytoplankton and diatoms) and two zooplankton size classes (microzooplankton and mesozooplankton). Diatoms differ from nanophytoplankton by their need in Si, by higher requirements in Fe [Sunda and Huntsman, 1995] and by higher half-saturation constants because of their larger mean size. For all living compartments, the ratios between C, N and P are kept constant to the values proposed by Takahashi et al. [1985]. On the other hand, the internal contents in Fe of both phytoplankton groups and in Si of diatoms are prognostically simulated as a function of the external concentrations in nutrients and of the light level. The Chl/C ratio is modeled using a modified version of the photoadaptation model by Geider et al.[1998]. All the elemental ratios of zooplankton are kept constant.

There are three non-living compartments: semi labile dissolved organic matter (with timescales of several weeks to several years), small and big sinking particles. The two particle size classes differ by their sinking speeds (2 m/d for the small size class and 50 to 200 m/d for the large size class). As for the living compartments, constant Redfield ratios are imposed for C/N/P. However, the iron, silicon and calcite pools of the particles are fully simulated. As a consequence, their ratios relative to organic carbon are allowed to vary. The impact of ballast minerals on particles sinking speeds is not accounted for in the model [e.g., Armstrong et al., 2002].

Nutrients are supplied to the ocean from three different sources: atmospheric dust deposition, rivers and sediment mobilization. These sources are explicitly modeled and are extensively described in the supplementary material. Thus only the main aspects are presented here. Iron deposition from the atmosphere has been estimated from the climatological monthly maps of dust deposition simulated by the model of Tegen and Fung [1995] assuming constant values for the iron content and the solubility [e.g., Jickells and Spokes, 2001; Moore et al., 2004]. River discharge of carbon is taken from the Global Erosion Model (GEM) of Ludwig et al. [1996]. Fe, N, P and Si supplies are derived from the same model output by considering globally constant Fe/P/N/Si/C ratios in the rivers. Reductive mobilization of iron from marine sediments has been recognized as a significant source to the ocean [e.g., Johnson et al., 1999; de Baar and de Jong, 2001]. Unfortunately, almost no quantitative information is available to describe this potentially important source. In a way similar to Moore et al. [2004], we have very crudely parameterized this input of iron.



PISCES indices	Units	Description
jpdic	$mol\,C\,l^{-1}$	Dissolved inorganic carbon
jptal	$eq\,l^{-1}$	Total alkalinity
jpoxy	$mol\,O_2\,l^{-1}$	dissolved oxygen
jpcal	$mol\,C\,l^{-1}$	Calcite
jppo4	$mol\,C\,l^{-1}$	Phosphate
jppoc	$mol\,C\,l^{-1}$	Small particulate organic carbon
jpsil	$mol\,Si\,l^{-1}$	silicate
jpphy	$mol\,C\,l^{-1}$	Nanophytoplankton
jpzoo	$mol\,C\,l^{-1}$	Microzooplankton
jpdoc	$mol\,C\,l^{-1}$	Semi-labile dissolved organic carbon
jpdia	$mol\,C\,l^{-1}$	Diatoms
jpmes	$mol\,C\,l^{-1}$	Mesozooplankton
jpdsi	$mol\,Si\,l^{-1}$	Silicon content of the diatoms
jpfer	$mol\,Fe\,l^{-1}$	Dissolved iron
jpbfe	$mol\,Fe\,l^{-1}$	Iron in the big particles
jpgoc	$mol\,C\,l^{-1}$	Big particulate organic carbon
jpsfe	$mol\,Fe\,l^{-1}$	Iron in the small particles
jpdfe	$mol\,Fe\,l^{-1}$	Iron content of the diatoms
jpgsi	$mol\,Si\,l^{-1}$	Sinking biogenic silica
jpnfe	$mol\,Fe\,l^{-1}$	Iron content of the nanophytoplankton
jpnch	$g\,Chl\,l^{-1}$	Chlorophyll of the nanophytoplankton
jpdch	$g\,Chl\,l^{-1}$	Chlorophyll of the diatoms
jpno3	$mol\,C\,l^{-1}$	Nitrate
jpnh4	$mol\,C\,l^{-1}$	Ammonium

PISCES has been used, at the global scale, to study past climates (Bopp et al. 2003 Paleoceanography), to understand the mechanisms that explain interannual variability in marine productivity (Aumont et al. 2008 GRL) or ocean-atmosphere carbon fluxes (Rodgers et al. 2008 GBC), to assess the impact of climate change or ocean acidification on marine ecosystems and air-sea carbon fluxes (Bopp et al. 2001 GBC, Orr et al. 2005 Nature), to evaluate geo-engineering strategies to mitigate climate change (Aumont and Bopp, 2006 GBC, Dutreuil et al. 2009 BG)…

2. The 1D vertical configuration NEMO-PISCES

In the tuning session, we use a 1-D (vertical) configuration of NEMO-PISCES in NEMOv4 version.

In the 1-D configuration, the size of the domain is (3 x 3 x jpk) and computation is performed only at the central point (i=2,j=2,jpk). The user has to provide, in the namelist, the coordinates (longitude and latitude) of the studied point : ***rn_lat1d, rn_lon1d*** and the ocean depth at the given point **rn_bathy.** In the v4 version, one needs input data (initial state and forcings) at the considered location : a set of 3 x 3 x jpk values

Note that there is no specific treatment of boundary conditions: the four lateral bands are considered as artificial walls; they are treated as land.

The one-dimensional vertical model only considers the vertical – z, sigma or partial steps – coordinate and time as independent variables in the primitive equations, with no horizontal variations. All horizontal derivatives are set to zero. **Thus, in this one-dimensional vertical configuration, there is no (both lateral and vertical) advection, no lateral mixing on tracers nor dynamics. The parameterization of vertical mixing for momentum and tracers is done via a vertical mixing scheme (TKE, KPP, Richardson, constant mixing).**

Temperature and salinity are only controlled by vertical diffusion equations and the density, needed for the turbulent mixing closure scheme, is computed using an equation of state of sea water:

dT/dt = d(KdT/dz) + penetrating solar flux F(z)

+ surface boundary conditions for T (flux bulk)

(vertical advection term -d(w.T)/dz =0 as w=0)

dS/dt = d(KdS/dz)

+ surface boundary conditions for S (E-P)

For a biogeochemical tracer:

dC/dt = -d(K dC/dt) + SMS_bio (SMS= sources minus sinks)

K is computed from TKE or any other way depending on the option selected in the dynamical namelist.

For passive tracers, only biogeochemical sources and sinks are taken into account together with vertical diffusion processes.

We will focus on some JGOFS sites:

BATS : (64°W, 31.5°N))
http://bats.bios.edu/

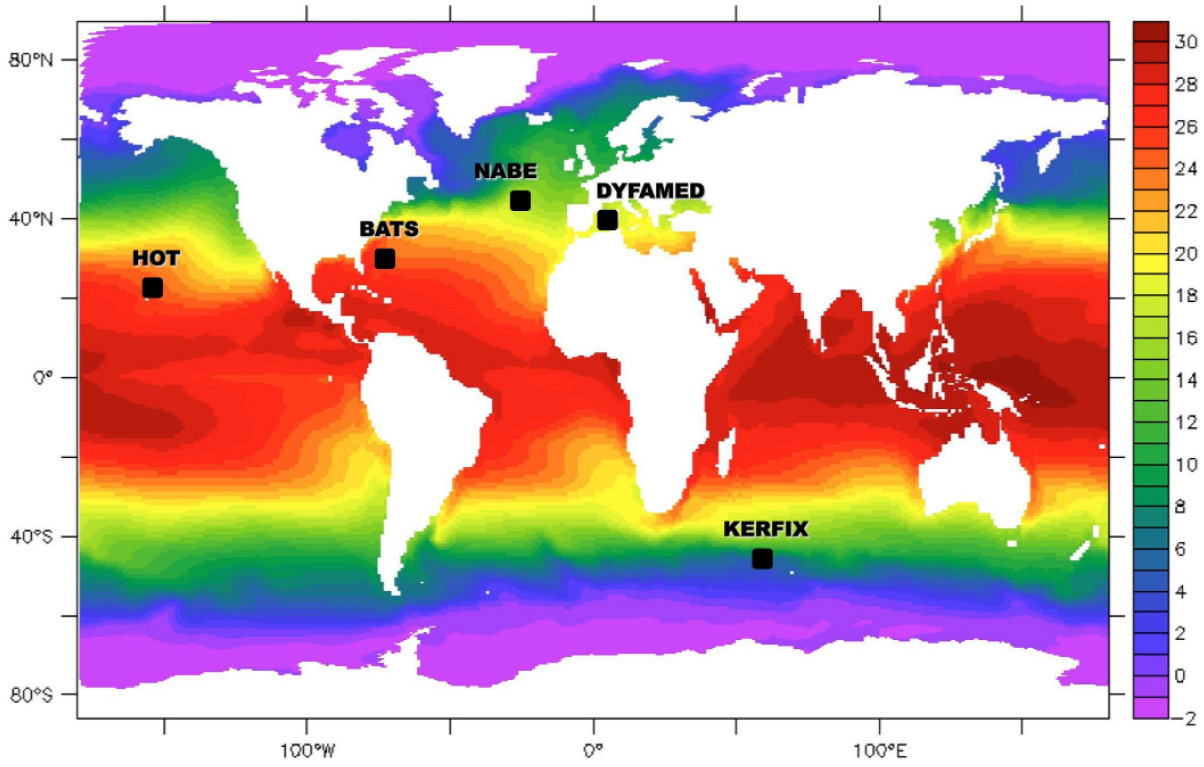
HOT : (158°W, 22.45°N)
http://hahana.soest.hawaii.edu/hot/

DYFAMED : (7.52°E, 43.27°N)
http://www.obs-vlfr.fr/cd_rom_dmtt/sodyf_main.htm

KERFIX (68.25°E, 50.40°S) :
http://www.obs-vlfr.fr/cd_rom_dmtt/OTHER/KERFIX/bacteries/kfx_bact_delille.htm.htm

NABE : (20°W,47°N)

http://usjgofs.whoi.edu/research/nabe.html



Outputs visualization

To visualize the results of the amazing simulations that you will produce throughout this session, you can use the Ferret product. This visualization environment is particularly convenient and fast for exploring the variables in the generated netcdf files, even if the plots are not of a quality approved for first-rank publications. Of course, you can also use your favorite product. Below is a short summary of some Ferret commands needed to explore the files within the framework of this training.

Launch Ferret

```
ferret
```

Read two netcdf files

```
yes? use BATS_1d_20100701_20130630_nemo.nc
yes? use BATS_1d_20100701_20130630_nemo.ctl.nc
```

To display the variables in the imported files

```
yes? show data
```

To remove all the files

```
yes? cancel data/all
```

1D visualization: plot NO3 concentration at the surface as a function of time

```
yes? plot/i=2/j=2/k=1 NO3[d=1]
yes? plot/i=2/j=2/k=1/over NO3[d=2]
```

k is vertical level (from 1 to N), d is the number of the file read by import order, the over command allows to draw NO3 of the second file above the first profile.

Plot surface NO3 concentrations of both files at once

```
yes? plot/i=2/j=2/k=1 NO3[d=1], NO3[d=2]
```

2D visualization: plot the temporal evolution of NO3 as a function of depth

```
yes? shade/i=2/j=2 NO3[d=1]
```

To compare two simulations

```
yes? shade/i=2/j=2 NO3[d=1] - NO3[d=2]
```

To plot the total chlorophyll

```
yes? Let CHL = DCHL + NCHL
yes? shade/i=2/j=2 CHL[d=1]
```

To compute a variable integrated over depth, here for silicate in sediments (cf # session 3)

```
yes? plot/i=2/j=2 sedsil[d=1,k=2:18@din]
```

To open a second window

```
yes? set window 2
```

A Ferret Tour introducing basics can be found here:
https://ferret.pmel.noaa.gov/Ferret/documentation/ferret-tutorial-script

PRACTICAL SESSION

Session 1: Model tuning

It takes about 3 hours to complete this session.

1. Your environment should first be properly set. Here is a bash_profile example of loaded modules

```
# ------- Librairies mpi    ------------
module load intel/15.0.6.233
module load openmpi/1.6.5-ifort
module load netcdf4/4.4.1.1-parallel-ifort
module load hdf5/1.8.18-parallel-ifort
```

2. Extract and install NEMOv4.0

Create a new directory in your `$workdir`.

```
mkdir -p $YOUR_WORKDIR/TP_PISCES/NEMO ; cd $YOUR_WORKDIR/TP_PISCES/NEMO
```

Download the code of NEMO V4

```
svn co http://forge.ipsl.jussieu.fr/nemo/svn/NEMO/releases/r4.0/r4.0.3 NEMOGCM
```

3. Compile and create NEMO executable

Set the arch file for compilation according to your computer environment

```
cd NEMOGCM  ; rm -rf doc

vi  arch/arch-your_fortran_comp.fcm
```

Create and compile the 1D configuration

```
./makenemo  -n  ORCA_1D_PISCES  -r  GYRE_PISCES  -m  your_fortran_comp  -j  0
add_key "key_c1d key_mpi2"

./makenemo -n ORCA_1D_PISCES -r GYRE_PISCES -m your_fortran_comp clean

cp cfgs/C1D_PAPA/MY_SRC/* cfgs/ORCA_1D_PISCES/MY_SRC/.

./makenemo -n ORCA_1D_PISCES -r GYRE_PISCES -m your_fortran_comp -j 4
```

4. Run the model

Create the run directory for the BATS station

```
cd $YOUR_WORKDIR/TP_PISCES/NEMO ; mkdir -p Runs/BATS ; cd Runs/BATS
```

Copy inputs and files needed to run the 1D configuration at BATS station

```
ln -sf $path_to_downloaded_ZenodoDir/TP_PISCES/inputs/BATS/* .

cp $path_to_downloaded_ZenodoDir/TP_PISCES/Runs/Shared/* .

cp $path_to_downloaded_ZenodoDir/TP_PISCES/Runs/BATS/* .
```

Explore the run directory

- **Changes in namelist_cfg**

The ocean depth at the station:

```
!-----------------------------------------------------------------------
&namusr_def    !   C1D user defined namelist
!-----------------------------------------------------------------------
   rn_bathy    =  4730.    ! depth in meters
/
!-----------------------------------------------------------------------
```

The geographical coordinates of the station:

```
!-----------------------------------------------------------------------
&namc1d        !   1D configuration options
!-----------------------------------------------------------------------
   rn_lat1d    =     31.5   !  Column latitude
   rn_lon1d    =    -64.    !  Column longitude
   ln_c1d_locpt =  .true.    ! Localization of 1D config in a grid
/
```

The initialisation files of temperature and salinity:

```
!-----------------------------------------------------------------------
&namtsd        !    Temperature & Salinity Data  (init/dmp)            (default: OFF)
!-----------------------------------------------------------------------
   !                     ! =T  read T-S fields for:
   ln_tsd_init = .true.          !  ocean initialisation

   cn_dir      = './'      !  root directory for the T-S data location
   !___________!_____________________________!___________________!___________!_____________!________!_____
   !           !  file name                  ! frequency (hours) ! variable  ! time interp.!  clim  ! 'year
   !           !                             !  (if <0  months)  !   name    !   (logical) !  (T/F) ! 'mont
   sn_tem      = 'InitTS_l75_BATS'    ,         -1.       ,'votemper',  .false.     , .true.,  'yearly'
   sn_sal      = 'InitTS_l75_BATS'    ,         -1.       ,'vosaline',  .false.     , .true.,  'yearly'
/
```

The forcing atmospheric files at the station:

```
!-----------------------------------------------------------------------
&namsbc_blk    !   namsbc_blk  generic Bulk formula                    (ln_blk =T)
!-----------------------------------------------------------------------
   !                   !  bulk algorithm :
   ln_NCAR     = .true.   ! "NCAR"      algorithm   (Large and Yeager 2008)
      !
   cn_dir = './'  !  root directory for the bulk data location
   !___________!_____________!____________________!___________!_____________!________!___________!___________
   !           !  file name   ! frequency (hours) ! variable  ! time interp.!  clim  ! 'yearly'/ !  weights f
   !           !              !  (if <0  months)  !   name    !   (logical) !  (T/F) ! 'monthly' !
   sn_wndi     = 'JRA_Forcings_3h_BATS',         3          , 'uas'      ,     .false.  ,  .false. , 'yearly'
   sn_wndj     = 'JRA_Forcings_3h_BATS',         3          , 'vas'      ,     .false.  ,  .false. , 'yearly'
   sn_qsr      = 'JRA_Forcings_3h_BATS',         3          , 'rsds'     ,     .false. ,  .false. , 'yearly'
   sn_qlw      = 'JRA_Forcings_3h_BATS',         3          , 'rlds'     ,     .false. ,  .false. , 'yearly'
   sn_tair     = 'JRA_Forcings_3h_BATS',         3          , 'tas'      ,     .false.  ,  .false. , 'yearly'
   sn_humi     = 'JRA_Forcings_3h_BATS',         3          , 'huss'     ,     .false.  ,  .false. , 'yearly'
   sn_prec     = 'JRA_Forcings_3h_BATS',         3          , 'prra'     ,     .false.  ,  .false. , 'yearly'
   sn_snow     = 'JRA_Forcings_3h_BATS',         3          , 'prsn'     ,     .false.  ,  .false. , 'yearly'
   sn_slp      = 'JRA_Forcings_3h_BATS',         3          , 'psl'      ,     .false.  ,  .false. , 'yearly'
/
```

The Chlorophyll for bio penetration:

```
!-----------------------------------------------------------------------
&namtra_qsr    !   penetrative solar radiation                        (ln_traqsr =T)
!-----------------------------------------------------------------------
   !                     !  type of penetration                        (default: NO selection)
   ln_qsr_rgb  = .true.      !  RGB light penetration (Red-Green-Blue)

   cn_dir      = './'      !  root directory for the chlorophyl data location
   !___________!_________________________!___________________!___________!_____________!_______!_______
   !           !  file name              ! frequency (hours) ! variable  ! time interp.!  clim  ! 'yearly
   !           !                         !  (if <0  months)  !   name    !   (logical) !  (T/F) ! 'monthl

   sn_chl      ='chloro_BATS'            ,        -1.        , 'CHLA'    ,   .true.    , .true. , 'yearly
/
```

The runoff if needed at the station (for example DYFAMED)

- **Changes in namelist_top_cfg**

PISCES initial data

```
&namtrc_dta      !    Initialisation from data input file
!-----------------------------------------------------------------------
!          !  file name  ! frequency (hours) ! variable  ! time interp. !  clim
!          !             !  (if <0  months)  !   name    !   (logical)  !  (T/F)
   sn_trcdta(1)  = 'Glodapv2.1_annual_l75_BATS.nc',    -12.       ,  'TDIC'
   sn_trcdta(2)  = 'Glodapv2.1_annual_l75_BATS.nc',    -12.       ,  'TALK',
   sn_trcdta(3)  = 'WOA2009_monthly_l75_BATS.nc',       -1.        ,  'O2'
   sn_trcdta(5)  = 'WOA2009_monthly_l75_BATS.nc',       -1.        ,  'PO4'
   sn_trcdta(7)  = 'WOA2009_monthly_l75_BATS.nc',       -1.        ,  'Si'
   sn_trcdta(10) = 'PISCES_monthly_l75_BATS.nc' ,        -1.       ,  'DOC'
   sn_trcdta(14) = 'PISCES_monthly_l75_BATS.nc' ,        -1.       ,  'Fer'
   sn_trcdta(23) = 'WOA2009_monthly_l75_BATS.nc',        -1.        ,  'NO3'
```

Flag to disable advection & lateral diffusion

```
!-----------------------------------------------------------------------
&namtrc_adv      !   advection scheme for passive tracer
!-----------------------------------------------------------------------
   ln_trcadv_OFF =  .true.  !  MUSCL scheme
/
!-----------------------------------------------------------------------
&namtrc_ldf      !   lateral diffusion scheme for passive tracer
!-----------------------------------------------------------------------
   ln_trcldf_OFF   = .true.      !  use active tracer setting
/
```

- **Changes in namelist_pisces_cfg**

PAR from a file

```
&nampisopt     !   parameters for optics
!-----------------------------------------------------------------
   sn_par      = 'par_fr_gewex_clim_BATS.nc'       ,     24.
   cn_dir      = './'      !  root directory for the location
   ln_varpar   =  .true.   ! boolean for PAR variable
   parlux      =  0.43      ! Fraction of shortwave as PAR
/
```

Nutrients from external inputs (dust, river etc …)

```
   sn_solub    = 'Solubility_Mahowald_BATS' ,    -12.          , 'solubility2' ,
   sn_riverdic = 'river_global_news_BATS'      ,    -1.          , 'riverdic'
   sn_riverdoc = 'river_global_news_BATS'      ,    -1.          , 'riverdoc'
   sn_riverdin = 'river_global_news_BATS'      ,    -1.          , 'riverdin'
   sn_riverdon = 'river_global_news_BATS'      ,    -1.          , 'riverdon'
   sn_riverdip = 'river_global_news_BATS'      ,    -1.          , 'riverdip'
   sn_riverdop = 'river_global_news_BATS'      ,    -1.          , 'riverdop'
   sn_riverdsi = 'river_global_news_BATS'      ,    -1.          , 'riverdsi'
   sn_ndepo    = 'Dustdep_BATS'        ,    -1.          , 'ndep'        ,  .tru
   sn_ironsed  = 'pmarge_etopo_BATS'          ,    -12.          , 'bathy'
```

Run the model :

Copy the executable in your run directory and run the model

```
cd $YOUR_WORKDIR/TP_PISCES/NEMO/Runs/BATS

cp $YOUR_WORKDIR/TP_PISCES/NEMO/NEMOGCM/cfgs/ORCA_1D_PISCES/BLD/bin/nemo.exe .

./nemo.exe &

exit  !   at the end of the run
```

5. Sensitivity tests (tuning)

The sensitivity experiments are performed at two stations: BATS and NABE. For this purpose, create two directories, one for each station (BATS run directory is already created), run the model and back up the outputs and the pisces namelist (ref and cfg) files (CTL runs).

We start with BATS. In this first part, we will explore the sensitivity of the model to some parameters and see how they impact the solution.

5.1 First, remove all the nutrient supplies. In the namelist_pisces_cfg, flag to disable in the appropriate block:

```
   ln_dust     =  .false.   ! boolean for dust input from the atmosphere
   ln_solub    =  .false.   ! boolean for variable solubility of atm. Iron
   ln_river    =  .false.   ! boolean for river input of nutrients
   ln_ndepo    =  .false.   ! boolean for atmospheric deposition of N
   ln_ironsed  =  .false.   ! boolean for Fe input from sediments
   ln_ironice  =  .false.   ! boolean for Fe input from sea ice
   ln_hydrofe  =  .false.   ! boolean for from hydrothermal vents
```

Run the model and compare to the control simulation.

Optional: From the run without nutrient inputs activate individually ln_dust and ln_ndepo and compare to both previous runs.

5.2 Second, increase and decrease the pislopen, pisloped by 50% and compare to CTL.

```
!-----------------------------------------------------------------------
&namp4zprod    !   parameters for phytoplankton growth for PISCES std  - ln_p4z
!-----------------------------------------------------------------------
   pislopen    =  2.       ! P-I slope
   pisloped    =  2.       ! P-I slope  for diatoms
```

Repeat that step at NABE *(Optional, if you have time or as a home work)*.

5.3 Increase and decrease the half-saturation constants of both NO3 and NH4 by 50% for both phytoplankton groups and compare to CTL.

```
!-----------------------------------------------------------------------
&namp4zlim     !   parameters for nutrient limitations for PISCES std  - ln_p4z
!-----------------------------------------------------------------------
   concnno3   =  1.e-6    ! Nitrate half saturation of nanophytoplankton
   concdno3   =  3.E-6    ! Nitrate half saturation for diatoms
   concnnh4   =  1.E-7    ! NH4 half saturation for phyto
   concdnh4   =  3.E-7    ! NH4 half saturation for diatoms
```

Repeat that step for diatoms only and compare to the CTL and previous tests.

5.4 Decrease by 50% the grazing rates of both zooplankton groups simultaneously.

```
!-----------------------------------------------------------------------
&namp4zzoo     !   parameters for microzooplankton for PISCES std      - ln_p4z
!-----------------------------------------------------------------------
   part       =  0.5      ! part of calcite not dissolved in microzoo guts
   grazrat    =  3.0      ! maximal zoo grazing rate
!-----------------------------------------------------------------------
&namp4zmes     !   parameters for mesozooplankton for PISCES std       - ln_p4z
!-----------------------------------------------------------------------
   part2      =  0.75     ! part of calcite not dissolved in mesozoo guts
   grazrat2   =  0.75     ! maximal mesozoo grazing rate
```

Compare to the CTL run.

Repeat that step by changing grazrat2 only and compare to the previous and CTL tests.
What would you suggest to increase the mesozooplankton biomass to levels similar to the CTL simulation?

5.5 Increase the scavenging rate by a factor of 10 at both stations (NABE and BATS).

```
!-----------------------------------------------------------------------
&nampisfer     !   parameters for iron chemistry
!-----------------------------------------------------------------------
   ln_ligvar =  .false.    ! variable ligand concentration
   xlam1     =  0.005     ! scavenging rate of Iron
```

Run and compare to the CTL simulations.

5.6 Divide by 2 the fecnm and fecdm values:

```
!-----------------------------------------------------------------------
&namp4zprod    !   parameters for phytoplankton growth for PISCES std  - ln_p4z
!-----------------------------------------------------------------------
   pislopen    =  2.        ! P-I slope
   pisloped    =  2.        ! P-I slope  for diatoms
   xadap       =  0.        ! Adaptation factor to low light
   excretn     =  0.05      ! excretion ratio of phytoplankton
   excretd     =  0.05      ! excretion ratio of diatoms
   bresp       =  0.033     ! Basal respiration rate
   chlcnm      =  0.033     ! Maximum Chl/C in nanophytoplankton
   chlcdm      =  0.05      ! Maximum Chl/C in diatoms
   chlcmin     =  0.004     ! Minimum Chl/c in phytoplankton
   fecnm       =  40E-6     ! Maximum Fe/C in nanophytoplankton
   fecdm       =  40E-6     ! Maximum Fe/C in diatoms
```

Run and compare to the CTL and previous simulations.
Try this at KERFIX *(Optional)*.

5.7 Copy the outputs from the sensitivity test performed at BATS in your run directory

```
cp
$path_to_downloaded_ZenodoDir/TP_PISCES/BATS_STD/BATS_1d_20100701_20130630
_nemo.ctl.nc .
```

Your objective is to tune the model to be the closest you can to this sensitivity test by tweaking parameters in the namelist. Hint: the parameters you will have to change are those that have been explored in the previous steps of this session.

6. Manage the outputs (file_def_nemo.xml): *Optional for beginners*

1. change the output frequency : 5d
2. output in another file the non solar heat flux, the net downward heat flux, phosphate at daily frequency
3. output sst and sss at daily frequency and monthly mean temperature

---

Setting the BENGUELA config of CROCO-PISCES for Session 2 **(if you want to do session 2 with CROCO)**

1. Download and compile

Create the code and run directory for CROCO

```
mkdir $YOUR_WORKDIR/TP_PISCES/CROCO ; cd $YOUR_WORKDIR/TP_PISCES/CROCO
```

Download the code

```
git clone git@gitlab.inria.fr:croco-ocean/croco.git croco
```

Set the run configuration

```
cd croco ; vim create_run.bash
```

Modify your path configuration

```
MY_CONFIG_PATH=$YOUR_WORKDIR/TP_PISCES/CROCO
```

Replace in the script the several:

```
cp -Rf
by
cp -f
```

Create the run directory:

```
./create_run.bash
```

https://croco-ocean.gitlabpages.inria.fr/croco_doc/tutos/tutos.04.config.html

A directory named Run_Benguela_LR had to be created

Go in the run directory and set the cpp keys

```
cd $YOUR_WORKDIR/TP_PISCES/CROCO/Run_BENGUELA_LR
vi cppdefs.h
```

Modify the following CPP keys:

```
# define MPI
# define XIOS
# define BIOLOGY
# define PISCES
# undef BIO_BioEBUS
```

You have to install and compile XIOS on your machine

```
svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/branchs/xios-2.5
cd xios-2.5
./make_xios --arch Your_Arch_configuration --full --prod --job 8
```

Set the compiler in the jobcomp script

```
vi jobcomp
```

Replace predefined variables by your fortran compiler. For instance:

```
FC=ifort
and
XIOS_ROOT_DIR=$Path_to_XIOS/xios-2.5
```

Compile the code:

```
./jobcomp
```

2. Running

Copy the input files

```
cd $YOUR_WORKDIR/TP_PISCES/CROCO/Run_BENGUELA_LR/CROCO_FILES

ln -sf $path_to_downloaded_ZenodoDir/TP_PISCES/inputs/Run_BENGUELA_LR/* .
```

Set the output file directory

```
cd .. ; mkdir HIS
vim file_def.xml
remove all the   << split_freq="5d" split_freq_format="%y-%mo-%d" >>
```

Run the model configuration

```
mpirun -np 4 ./croco > benguela.log &
exit  !   at the end of the run
```

Outputs files are in directory HIS

Session 2: Adding a new tracer in PISCES

It takes about 2 hours to complete this session in NEMO and at least half an hour more in CROCO.

The purpose here is to detail step by step how to add a new tracer in PISCES within the two ocean models NEMO and CROCO. To do this, we will take inspiration from a tracer already existing in PISCES, the ligand tracer.

In NEMO:

The addition of a new tracer is dynamical, meaning that there is no need to recompile the code whether you are in standard version or in the version modified with an added tracer. You can activate or deactivate the tracer with a namelist key parameter.

For the case of the ligand tracer, the different implementation steps in the code are as follows:

In the source code:

You have to define the identifiers of your tracer.

in the **par_pisces.F90** routine, we define the tracer id (number id), jplgw for ligand:

```
   INTEGER, PUBLIC ::   jplgw     !: Weak Ligands
```

Now, we have to define the logical flag which activates the tracer in **sms_pisces.F90**:

```
   LOGICAL  ::  ln_ligand         !: Flag to enable organic ligands
```

To activate the reading of this parameter in the pisces namelist, add the ligand logical flag in **trcnam_pisces.F90**:

```
  NAMELIST/nampismod/ln_p2z, ln_p4z, ln_p5z, ln_ligand, ln_sediment
```

In **trcini_pisces.F90**:

- Add in the p4z_ini subroutine

```
USE p4zligand       !  Remineralization of organic ligands
```

- Indexing the ligand tracer in trc arrays

```
   IF( cltra == 'LGW'      )   jplgw = jn      !: Weak ligands
```

- Initialize the ligand tracer

```
trn(:,:,:,jpnh4) = bioma0
IF( ln_ligand) THEN
   trn(:,:,:,jplgw) = 0.6E-9
ENDIF
```

- And finally add the call for the initialization of the ligand tracer

```
IF( ln_ligand ) &
   & CALL p4z_ligand_init  !  remineralisation of organic ligands
```

Now you can create a module where you compute the source-minus-sink terms of your tracer. In the ligand case: p4zligand.F90. The screenshot of the routine shown below has been stripped of these features for illustration. You have to be inspired by the original routine whose path is shown below the figure.

```
MODULE p4zligand
   !!======================================================================
   !!                        ***  MODULE p4zligand  ***
   !! TOP :   PISCES Compute remineralization/dissolution of organic ligands
   !!======================================================================

   IMPLICIT NONE
   PRIVATE

   PUBLIC   p4z_ligand         ! called in p4zbio.F90
   PUBLIC   p4z_ligand_init    ! called in trcsms_pisces.F90

CONTAINS

   SUBROUTINE p4z_ligand( kt, knt )
      !!---------------------------------------------------------------------
      !!                     ***  ROUTINE p4z_ligand  ***
      !! ** Purpose :   Compute remineralization/scavenging of organic ligands
      !!---------------------------------------------------------------------
      !
   END SUBROUTINE p4z_ligand

   SUBROUTINE p4z_ligand_init
      !!---------------------------------------------------------------------
      !!
      !! ** Purpose :   Initialization of remineralization parameters
      !
   END SUBROUTINE p4z_ligand_init

   !!======================================================================
END MODULE p4zligand
```

The original p4zligand.F90 routine is in the directory:
NEMOGCM_v4.0.3/src/TOP/PISCES/P4Z/

Now, you can call your ligand routine in **p4zbio.F90** for instance, in two steps:

- Add in the header routine:

```
USE p4zligand       !  Prognostic ligand model
```

- and add in p4z_bio subroutine:

```
IF( ln_ligand )  CALL p4z_ligand( kt, knt )
```

Eventually, you can add other sink and source processes in other pisces routines depending on your scientific objectives.

In the namelist parameters:

First, in **namelist_top_cfg**:

- increase the total number of tracers: jp_bgc = 25
- you have to define your tracer:

```
tracer(24)  = 'NH4     ' , 'Ammonium Concentration                ',  'mol-C/L'
tracer(25)  = 'LGW     ' , 'Ligands Concentration                 ',  'mol-C/L'
```

Second, in **namelist_pisces_cfg**, you have to define the logical flag to enable or disable the activation of the tracer:

```
!-----------------------------------------------------------------------
&nampismod     !  Model used
!-----------------------------------------------------------------------
  ln_ligand = .false.        !  Enable  organic ligands
/
```

You have to add the specific parameters if needed and define them for your tracer. For example, in the **namelist_pisces_ref**:

```
!-----------------------------------------------------------------------
&nampislig     !   Namelist parameters for ligands, nampislig
!-----------------------------------------------------------------------
   rlgw        =  100.     ! Lifetime (years) of weak ligands
   rlig        =  1.E-4    ! Remin ligand production per unit C
   prlgw       =  1.E-4    ! Photolysis of weak ligand
   rlgs        =  1.       ! Lifetime (years) of strong ligands
/
```

Now, modify the xml files in two steps:

- Define the tracer name and eventually your additional diagnostics in the **field_def_nemo-pisces.xml**

```
<field_group id="ligd_T" grid_ref="grid_T_3D">
 <!-- PISCES with ligand parametisation : variables available namelist paramter ln_ligand -->
  <field id="LGW"         long_name="Weak ligands concentration"               unit="mmol/m3" />
  <field id="LGW_e3t"     long_name="LGW * e3t"                                unit="mmol/m2"  > LGW * e3t </field >
  <field id="LPRODR"      long_name="OM remineralisation ligand production rate" unit="nmol-L/m3/s"     />
  <field id="LPRODP"      long_name="phytoplankton ligand production rate"    unit="nmol-L/m3/s"        />
  <field id="LIGREM"      long_name="Remineralisation loss of ligands"        unit="nmol-L/m3/s"        />
  <field id="LIGPR"       long_name="Photochemical loss of ligands"           unit="nmol-L/m3/s"        />
  <field id="LDETP"       long_name="Ligand destruction during phytoplankton uptake" unit="nmol-L/m3/s" />
</field_group>
```

- Choose the diagnostics you want to output in the **file_def_nemo.xml**

```
<file_group id="1m" output_freq="1mo" output_level="10" enabled=".TRUE."> <!-- real monthly files -->
 <file id="file40" name_suffix="_ligd_T" description="pisces sms variables" >
    <field field_ref="LGW"       name="LGW"      operation="average" freq_op="1mo" > @LGW_e3t / @e3t </field>
    <field field_ref="LIGREM"    name="LIGREM"   operation="average" />
 </file>
</file_group>
```

In CROCO:

Unlike NEMO, the addition of a new tracer is not dynamic, which means that it cannot be controlled only by a namelist flag but also by a CPP key.

For the case of the ligand tracer, the different implementation steps in the code are as follows.

Copy all the routines to be modified in the run directory:

- ***.h, *.F, *.F90**
- **OCEAN/Make***
- **If you need to create new modules/routines, add them to the list of routine to be compile in the Makefile**

First, define a key_ligand cpp in OCEAN/cppdefs.h:

```
#  ifdef PISCES
#   undef  DIURNAL_INPUT_SRFLX
#   define key_pisces
#   define key_ligand
#  endif
```

In the source code, define the identifiers of the ligand tracer.

In the following files under the PISCES CPP key, add:

- Increase the total number of tracers: jptra = 25

vim ./OCEAN/param.h

```
#        ifdef key_ligand
         parameter (ntrc_bio=25)
#        endif
```

vim ./PISCES/par_pisces.F90

```
#   if defined key_ligand
   INTEGER, PUBLIC, PARAMETER ::   jp_pisces     = 25
#   endif
```

vim ./OCEAN/ncscrum.h

```
#     ifdef key_ligand
       integer indxLGW
       parameter (indxLGW=indxDIC+24)
#      endif
```

vim ./OCEAN/param.h

```
#     ifdef key_ligand
       parameter (iLGW_=iDIC_+24)
#      endif
```

- Add the attributes of the ligand tracer (necessary only if not using XIOS server):

vim ./OCEAN/init_scalars.F

```
#     if defined key_ligand
      vname(1,indxLGW)='LGW                                          '
      vname(2,indxLGW)='Ligands                                      '
      vname(3,indxLGW)='umol L-1                                     '
      vname(4,indxLGW)='Ligands, scalar, series                      '
      vname(5,indxLGW)='                                             '
      vname(6,indxLGW)='lat_rho lon_rho                              '
      vname(7,indxLGW)='                                             '
#     endif
```

- setting up the initial conditions (analytical or climatological), boundary conditions, and external sources for the ligand tracer in

vim ./OCEAN/ana_initial.F

```
            if (.not.got_tini(iNH4_)) then
              t(i,j,k,1,iNH4_)=1.e-2
              t(i,j,k,2,iNH4_)=t(i,j,k,1,iNH4_)
            endif
#     if defined key_ligand
            if (.not.got_tini(iLGW_)) then
              t(i,j,k,1,iLGW_)=1.e-3
              t(i,j,k,2,iLGW_)=t(i,j,k,1,iLGW_)
            endif
#     endif
```

vim ./OCEAN/analytical.F

```
             tclm(i,j,k,iDCH_)=1.e-2*12./55.
             tclm(i,j,k,iNH4_)=1.e-2
#      if defined key_ligand
             tclm(i,j,k,iLGW_)=1.e-3
#      endif
```

vim ./OCEAN/get_tclima.F

```
          elseif (itrc.eq.iNH4_) then
            got_tclm(itrc)=.true.
            ierr=nf_inq_varid (ncidclm, 'nh4_time', tclm_tid(itrc))
            if (ierr .ne. nf_noerr) then
              got_tclm(itrc)=.false.
              write(stdout,3) 'nh4_time', clmname(1:lstr)
c              goto 99                                    !--> ERROR
            endif
#     if defined key_ligand
          elseif (itrc.eq.iLGW_) then
            got_tclm(itrc)=.true.
            ierr=nf_inq_varid (ncidclm, 'lgw_time', tclm_tid(itrc))
            if (ierr .ne. nf_noerr) then
              got_tclm(itrc)=.false.
                MPI_master_only write(stdout,3) 'lgw_time', clmname(1:lstr)
c              goto 99                                    !--> ERROR
            endif
#     endif
```

vim ./OCEAN/get_bry_bio.F

```
            elseif (itrc.eq.iNH4_) then
              got_tbry(itrc)=.true.
              ierr=nf_inq_varid (bry_id, 'nh4_time', bry_tid(itrc))
             ! ierr_all=ierr_all+ierr
              if (ierr .ne. nf_noerr) then
                got_tbry(itrc)=.false.
                MPI_master_only write(stdout,3) 'nh4_time', bry_file(1:lstr)
c     goto 99                                          !--> ERROR
              endif
#     if defined key_ligand
              elseif (itrc.eq.iLGW_) then
              got_tbry(itrc)=.true.
              ierr=nf_inq_varid (bry_id, 'lgw_time', bry_tid(itrc))
             ! ierr_all=ierr_all+ierr
              if (ierr .ne. nf_noerr) then
                got_tbry(itrc)=.false.
                MPI_master_only write(stdout,3) 'lgw_time', bry_file(1:lstr)
c     goto 99                                          !--> ERROR
              endif
#     endif
```

vim ./OCEAN/get_psource_ts.F

```
          elseif (itrc.eq.iNH4_) then
            got_tsrc(itrc)=.true.
            ierr=nf_inq_varid (ncidqbar, 'nh4_src_time',
     &           tsrc_tid(itrc))
            if (ierr .ne. nf_noerr) then
              got_tsrc(itrc)=.false.
              MPI_master_only write(stdout,3) 'nh4_src_time', qbarname(1:lstr)
c     goto 99                                     !--> ERROR
            endif
#   if defined key_ligand
          elseif (itrc.eq.iLGW_) then
            got_tsrc(itrc)=.true.
            ierr=nf_inq_varid (ncidqbar, 'lgw_src_time',
     &           tsrc_tid(itrc))
            if (ierr .ne. nf_noerr) then
              got_tsrc(itrc)=.false.
              MPI_master_only write(stdout,3) 'lgw_src_time', qbarname(1:lstr)
c     goto 99                                     !--> ERROR
            endif
#   endif
```

In ./PISCES/trcini_pisces.F90:

- Add under the logical flag “if defined key_pisces”

```
 USE p4zligand        !  Remineralization of organic ligands
```

- define the logical flag associated with the CPP key (to match with the PISCES code in NEMO, which is in dynamical allocation).

```
#if defined key_ligand
      ln_ligand = .true.
#else
      ln_ligand = .false.
#endif
```

- Initialize the ligand tracer

```
trn(:,:,:,jpnh4) = bioma0
IF( ln_ligand) THEN
   trn(:,:,:,jplgw) = 0.6E-9
ENDIF
```

- And finally add the call for the initialization of the ligand tracer

```
 IF( ln_ligand ) &
    & CALL p4z_ligand_init  !  remineralisation of organic ligands
```

in ./XIOS/field_def.xml_full_xios2.5, add:

```
        <field id="PFe"      long_name="Picophytoplankton Fe biomass"          unit="mmol/m3" grid_ref="rho_3D" />
        <field id="PCHL"     long_name="Picophytoplankton Chl biomass"         unit="mg/m3"   grid_ref="rho_3D" />
#endif
#if defined key_ligand
        <field id="LGW"      long_name="Weak ligands concentration"            unit="mmol/m3" grid_ref="rho_3D" />
#endif
# if defined key_trc_diaadd
        <field id="PH"       long_name="PH"                                    unit="-"           grid_ref="rho_3D" />
        <field id="CO3"      long_name="Bicarbonates"                          unit="mol/L"       grid_ref="rho_3D" />
```

In the source code (PISCES/), define the identifiers of the tracer:
in the par_pisces.F90 routine, we define the tracer id (number id), jplgw for ligand:

```
   INTEGER, PUBLIC ::   jplgw     !: Weak Ligands
```

Now, we have to define the logical flag which activates the tracer in sms_pisces.F90:

```
   LOGICAL  ::  ln_ligand         !: Flag to enable organic ligands
```

Now, you can create a module where you compute the source minus sink of your tracer, in the ligand case, p4zligand.F90. The screenshot of the routine shown below has been stripped of these features for illustration. You have to be inspired by the original routine whose path is shown below the figure:

```
MODULE p4zligand
   !!======================================================================
   !!                        ***  MODULE p4zligand  ***
   !! TOP :   PISCES Compute remineralization/dissolution of organic ligands
   !!======================================================================

   IMPLICIT NONE
   PRIVATE

   PUBLIC   p4z_ligand         ! called in p4zbio.F90
   PUBLIC   p4z_ligand_init    ! called in trcsms_pisces.F90

CONTAINS

   SUBROUTINE p4z_ligand( kt, knt )
      !!---------------------------------------------------------------------
      !!                     ***  ROUTINE p4z_ligand  ***
      !! ** Purpose :   Compute remineralization/scavenging of organic ligands
      !!---------------------------------------------------------------------
      !
   END SUBROUTINE p4z_ligand

   SUBROUTINE p4z_ligand_init
      !!---------------------------------------------------------------------
      !!
      !! ** Purpose :   Initialization of remineralization parameters
      !
   END SUBROUTINE p4z_ligand_init

   !!======================================================================
END MODULE p4zligand
```

The original p4zligand.F90 routine is in the PISCES source code directory.

Now, you can call your ligand routine in p4zbio.F90 for instance, in two steps:

- Add in the header routine:

```
USE p4zligand       !  Prognostic ligand model
```

- And add in p4z_bio subroutine:

```
IF( ln_ligand )  CALL p4z_ligand( kt, knt )
```

Eventually, you can add other sink and source processes in other pisces routines depending on your scientific objectives.

In the namelist parameters:
First, in namelist_pisces_ref:

- you have to define your tracer:

```
tracer(24)  = 'NH4     ' , 'Ammonium Concentration                ',  'mol-C/L'
tracer(25)  = 'LGW     ' , 'Ligands Concentration                 ',  'mol-C/L'
```

- You have to add the specific parameters if needed and defined for your tracer. For example, in the namelist_pisces_ref:

```
!-----------------------------------------------------------------------
&nampislig     !   Namelist parameters for ligands, nampislig
!-----------------------------------------------------------------------
   rlgw        =  100.     ! Lifetime (years) of weak ligands
   rlig        =  1.E-4    ! Remin ligand production per unit C
   prlgw       =  1.E-4    ! Photolysis of weak ligand
   rlgs        =  1.       ! Lifetime (years) of strong ligands
/
```

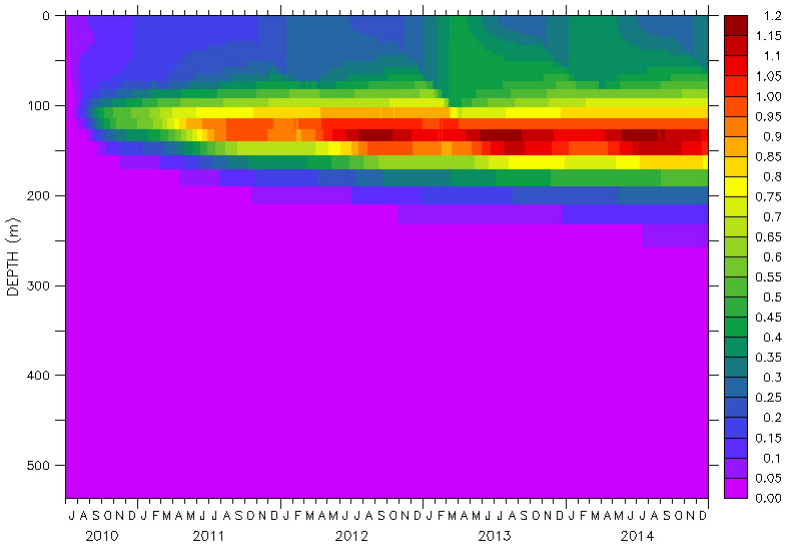
Practical work :

Based on the above example create a new tracer C whose behavior is controlled by the following equation:

$\partial C/\partial t$= alpha * (ppdiat + ppnano) - k * f(t) * Blim * C - Beta * PAR * C

ppnano and ppdiat are the primary production by respectively nanophytoplankton and diatoms (model variables zprorcan and zprorcad in p4zprod.F90). In this equation, alpha is the fraction of primary production that produces CDOM, beta the photochemistry rate constant, and k is the concentration lifetime and they should be respectively set to 5E-2, 1E-4 $(W\ m^{-2})^{-1}\ d^{-1}$, 1 yr^{-1}. f(t) is the temperature sensitivity used in phytoplankton growth, Blim is the bacterial production factor (blim) and PAR is the Photosynthetic Available Radiation (etot).

Here is the result you should get for CDOM in NEMO (BATS Station):



CDOM concentration (mmol/m3)

Complementary information: in the code tra is the trend and trb the concentration value of the biogeochemical tracer, xstep is the time step, and nyear_len is the length in days of the year (previous/current/next)

Session 3: SEDIMENT module

It takes about 1.5 hour to complete this session.

A rapid introduction to diagenetic processes and modelling will be provided and a brief explanation of the code will be given.

To run the sediment module online with PISCES, set ln_sediment to true in namelist_pisces_cfg. Here, we will run the sediment module offline with PISCES in a 1D configuration. We will create this new configuration from the ORCA2 offline configuration:

```
./makenemo -n ORCA_1D_SEDOFF -r ORCA2_OFF_PISCES -m your_fortran_comp -j 0
add_key "key_c1d key_mpi2 key_sed_off"

./makenemo -n ORCA_1D_SEDOFF -r ORCA2_OFF_PISCES -m your_fortran_comp
clean
```

copy routines to the MY_SRC directory of your configuration

```
cp $path_to_downloaded_ZenodoDir/TP_PISCES/SRC/BATS_SEDOFF/*
cfgs/ORCA_1D_SEDOFF/MY_SRC/
```

compile the code

```
./makenemo -n ORCA_1D_SEDOFF -r ORCA2_OFF_PISCES -m your_fortran_comp -j 8
```

Create your run directory

```
cd $YOUR_WORKDIR/TP_PISCES/NEMO ; mkdir -p Runs/BATS_SEDOFF

cd Runs/BATS_SEDOFF
```

Copy inputs, namelist, and xml files in your run directory:

```
ln -sf $path_to_downloaded_ZenodoDir/TP_PISCES/inputs/BATS_SEDOFF/*  .

rm -f restart_sed_in.nc

cp $path_to_downloaded_ZenodoDir/TP_PISCES/inputs/BATS_SEDOFF/restart_sed_in.nc  .

cp $path_to_downloaded_ZenodoDir/TP_PISCES/Runs/Shared/* .

cp $path_to_downloaded_ZenodoDir/TP_PISCES/Runs/BATS_SEDOFF/* .
```

Explore the namelist_cfg. There are two major changes relative to the previous 1D configuration:

1. The duration and the time step value:

```
!-----------------------------------------------------------------------
&namrun        !   parameters of the run
!-----------------------------------------------------------------------
   cn_exp      =  "PISCES"    !  experience name
   nn_it000    =       1      !  first time step
   nn_itend    =    14600     !  last  time step (std 5475)
   nn_date0    =  19600101    !  date at nit_0000 (format yyyymmdd) used if ln_rstart=F or (ln_rstart=T and nn_rstctl=0 or 1)
   nn_stock    =    14600     !  frequency of creation of a restart file (modulo referenced to 1)
   ln_mskland  = .true.       !  mask land points in NetCDF outputs (costly: + ~15%)
   ln_cfmeta   = .true.       !  output additional data to netCDF files required for compliance with the CF metadata standard
   ln_clobber  = .true.       !  clobber (overwrite) an existing file
/
!-----------------------------------------------------------------------
&namdom        !   time and space domain
!-----------------------------------------------------------------------
   ln_linssh   = .true.   !  =T  linear free surface  ==>>  model level are fixed in time
   rn_rdt      = 86400.     !  time step for the dynamics and tracer
/
```

2. The switch from an online to an offline dynamical simulation

```
!-----------------------------------------------------------------------
&namdta_dyn    !   offline ocean input files                       (OFF_SRC only)
!-----------------------------------------------------------------------
   ln_dynrnf       =  .false.    !  runoffs option enabled (T) or not (F)
   ln_dynrnf_depth =  .false.    !  runoffs is spread in vertical (T) or not (F)
   cn_dir      = './'      !  root directory for the ocean data location
   !___________!_________________________!___________________!___________!_____________!________!___________!__________________!__________!_______________!
   !           !  file name              ! frequency (hours) ! variable  ! time interp.!  clim  ! 'yearly'/ ! weights filename ! rotation ! land/sea mask !
   !           !                         !  (if <0  months)  !   name    !   (logical) !  (T/F) ! 'monthly' !                  ! pairing  !    filename   !
   sn_tem      = 'dyna_grid_T_BATS'          ,       -12.       , 'thetao'  ,  .true.    , .true. , 'yearly'  , ''                 , ''       , ''
   sn_sal      = 'dyna_grid_T_BATS'          ,       -12.       , 'so'  ,  .true.    , .true. , 'yearly'  , ''                 , ''       , ''
/
```

Explore the namelist_top_cfg. The tracers necessary to compute the boundary conditions at sediment interface are read in a file. In that specific case, the input file has been produced from the standard 1D configuration at BATS and used during the first training session.

```
!-----------------------------------------------------------------------
&namtrc_dta      !    Initialisation from data input file
!-----------------------------------------------------------------------
!          !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation ! land/sea mask !
!          !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  ! filename      !
   sn_trcdta(1)  = 'model_PISCES_BATS.nc',       -12.         , 'DIC'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(2)  = 'model_PISCES_BATS.nc',       -12.         , 'Alkalini',    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(3)  = 'model_PISCES_BATS.nc',       -12.         , 'O2'      ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(4)  = 'model_PISCES_BATS.nc',       -12.         , 'CaCO3'   ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(5)  = 'model_PISCES_BATS.nc',       -12.         , 'PO4'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(6)  = 'model_PISCES_BATS.nc',       -12.         , 'POC'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(7)  = 'model_PISCES_BATS.nc',       -12.         , 'Si'      ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(14) = 'model_PISCES_BATS.nc' ,      -12.         , 'Fer'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(15) = 'model_PISCES_BATS.nc' ,      -12.         , 'BFe'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(16) = 'model_PISCES_BATS.nc' ,      -12.         , 'GOC'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(17) = 'model_PISCES_BATS.nc' ,      -12.         , 'SFe'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(19) = 'model_PISCES_BATS.nc' ,      -12.         , 'GSi'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(23) = 'model_PISCES_BATS.nc',       -12.         , 'NO3'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
   sn_trcdta(24) = 'model_PISCES_BATS.nc',       -12.         , 'NH4'     ,    .true.    , .true. , 'yearly'  , ''        , ''    , ''
```

Explore the namelist_pisces_cfg:

```
!-----------------------------------------------------------------------
&nampismod     !  Model used
!-----------------------------------------------------------------------
  ln_sediment = .true.      !  Enable sediment module
/
```

Explore the namelist_sediment_ref and namelist_sediment_cfg.

Run the sediment model :

Copy the executable in your run directory

```
cp $YOUR_WORKDIR/TP_PISCES/NEMO/NEMOGCM/cfgs/ORCA_1D_SEDOFF/BLD/bin/nemo.exe .
```

Run the model

```
./nemo.exe &

exit  !   at the end of the run
```

Explore the outputs (solid and dissolved species). Back up the outputs of this CTL experiment.

Sensitivity tests:

- Set to the same value the remineralization rates of the 3 classes of POC. Choose for that test the value of the most labile class.

```
!-----------------------------------------------------------------------
&nam_poc     !   parameters for the redox reactions of the sediment module
!-----------------------------------------------------------------------
   redO2    =  140.     ! Redfield coef for Oxygen
   redNo3   =  16.      ! Redfield coef for Nitrate
   redPo4   =  1.       ! Redfield coef for Phosphate
   redC     =  122.     ! Redfield coef for Carbon
   redfep   =  0.05     ! Ration of Fe bound P
   rcorgl   =  10.      ! Reactivity for labile POC [an-1]
   rcorgs   =  0.1      ! Reactivity for semi-refractory POC [an-1]
   rcorgr   =  1.E-4    ! Reactivity for refractory POC [an-1]
```

Run the model for 2 cycles (equivalent to 100 years) and compare to CTL. To run the model for 2 cycles, use the available script script_job.1 (ksh script_job.1)

- Restore the standard values for the remineralization rates of the 3 classes of POC and set off irrigation in the sediments:

```
!-----------------------------------------------------------------------
&nam_btb     !   parameters for bioturbation and bioirrigation
!-----------------------------------------------------------------------
   dbiot       =  10.       ! Bioturbation coefficient (in cm2/yr)
   ln_btbz     =  .true.    ! Depth varying bioturbation
   dbtbzsc     =  6.0       ! coefficient for btb attenuation. If no variation, depth of the mixed layer
   adsnh4      =  1.3       ! Adsorption coefficient of NH4
   ln_irrig    =  .true.   ! Bioirrigation in sediment
   xirrzsc     =  1.5       ! Vertical scale of the attentuation of bioirrigation [cm]
/
```

Run the model and compare to CTL.

- Set dbiot to 0.1.

Run the model and compare to CTL.

In the next sensitivity experiment, we will explore the response of the sediments to a large decrease in the POC flux.

- Open the file namelist_pisces_cfg and set wsbio, wsbio2 and wsbio2max to 50 (10 times less than the initial configuration value).

Compare the outputs of solid and dissolved species to CTL and interprete.