CROCO-PISCES Training Session South Africa 2022

Olivier Aumont, Christian Éthé, Vincent Echevin, Odette Vergara and Renaud Person

---

The objective of this session is to learn the fundamentals of the PISCES marine biogeochemistry model, one of the biogeochemical components implemented in the ocean modeling platform CROCO. You will explore some of the features of PISCES using the Benguela LR configuration. A presentation will be given on marine biogeochemistry modeling and on the architecture and main features of PISCES. You will first install the Benguela_LR configuration, run the code and explore some of the physical and biogeochemical variables produced by the model. In a second session, you will perform sensitivity tests of the model to a set of biogeochemical parameters. In order to improve the material and the course of future sessions for beginners, we welcome any positive or negative feedback from you.

Please note that PISCES training sessions are organized every year around September and October. These training sessions are announced around May on the PISCES community website: https://www.pisces-community.org/

The PISCES training team

---

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.

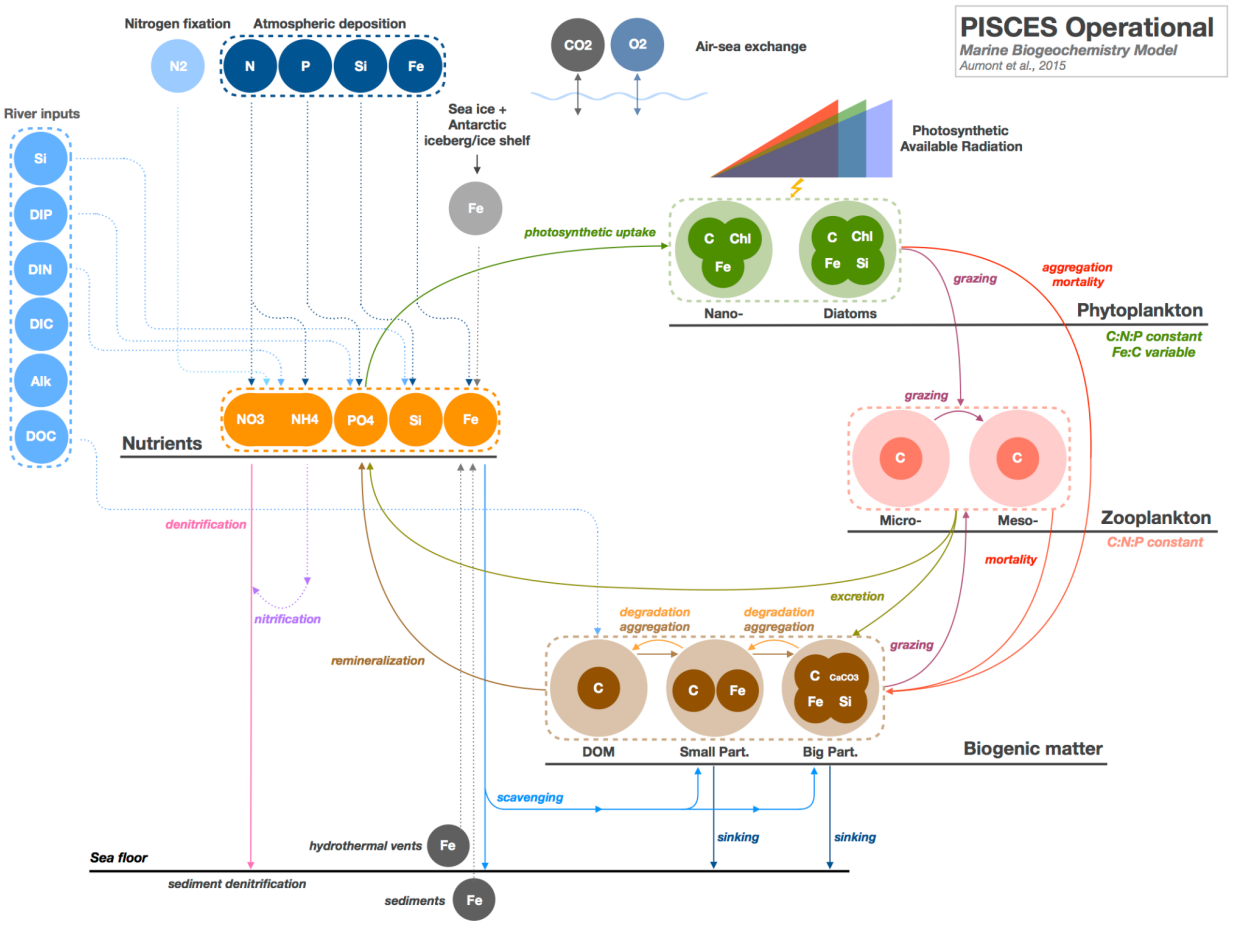


Figure 1. schematic of PISCES

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)…

PRACTICAL SESSION

The objective is to simulate a 1-month run of CROCO-PISCES with the Benguela configuration and perform several sensitivity tests to observe the changes that occur in the study area when some parameters are modified.

We will first set up the Benguela domain test configuration, run a one month experiment, which will serve as a control run and explore some physical and biogeochemical variables generated in the output file.

We will first set up the Benguela LR configuration, run a one-month experiment used after as a control run and explore some of the physical and biogeochemical variables generated in the output file.

In the second session, you will perform the sensitivity experiments, compare them to the control run and interpret the differences.

Session 1: Model set up

Warning: avoid copying / pasting the command lines indicated in the pdf document of the TP in your terminal because, depending on the environment, this may introduce errors.

1. Log-on onto the cluster

- open a terminal
- connect to the cluster

```
ssh -X yourlogin@scp.chpc.ac.za
yourlogin@scp.chpc.ac.za's password::
```

2. Setting the environment

```
cp /mnt/lustre/users/cethe/SpringSchool2022/bashrc .bashrc

source .bashrc
```

Then reserve 1 interactive processor

```
qsubi1
```

3. Get and install CROCO-PISCES in the BENGUELA_LR configuration

Copy the code and create the run directory for CROCO-PISCES

```
cd lustre/
mkdir PISCES
cd PISCES/
```

Get the code

```
cp /mnt/lustre/users/cethe/SpringSchool2022/Sources/croco.tgz .
tar zxvf croco.tgz
rm -f croco.tgz
```

Get the croco tools

```
cp /mnt/lustre/users/cethe/SpringSchool2022/Sources/croco_tools.tgz .
tar zxvf croco_tools.tgz
rm -f croco_tools.tgz
```

Set the run configuration

```
cd croco
```

Open the configuration setup file

```
vi create_config.bash
```

Set the configuration name

```
# Configuration name
# ------------------
MY_CONFIG_NAME=Run_PISCES
```

Create your run configuration

```
./create_config.bash
```

Answer “Yes” to the question on creating configuration
A directory named Run_PISCES has been created.

Now, we will activate the PISCES model by enabling the cpp key dedicated to compile the PISCES code

Go in your run directory:

```
cd Run_PISCES/
```

Open the `cppdefs.h` file

```
vi cppdefs.h
```

Activate the MPI cpp keys for code computing parallelisation on the cluster

```
                  /* Parallelization */
# undef  OPENMP
# define  MPI
```

Activate the cpp key for Biology applications in CROCO

```
                  /* Applications */
# define  BIOLOGY
```

Different biogeochemical models can be activated in the Choice of Biology models section of the cppdefs.h. Here we will activate the cpp key for `PISCES` and deactivate the `BIO_BioEBUS` model.

```
                  /*   Choice of Biology models   */
# ifdef BIOLOGY
#  define  PISCES
#  undef  BIO_NChlPZD
#  undef  BIO_N2ChlPZD2
#  undef BIO_BioEBUS
```

Now, you can compile the CROCO-PISCES code:

```
./jobcomp
```

4. Create the input files for PISCES

The CROCO-PISCES code is now compiled. Before running the model you will have to create the input files for initial and boundary biogeochemical conditions of the BENGUELA_LR configuration.

Open the `crocotools_param.m` file

```
vi crocotools_param.m
```

Choose the parameter to pre-process PISCES files

```
makebry    = 0;   % lateral boundary data
makepisces = 1;   % initial and boundary data for PISCES model
```

Launch Matlab

```
matlab -nodesktop
```

```
>> start
```

Create the Benguela configuration grid

```
>> make_grid
```

Create the surface forcing files

```
>> make_forcing
```

Create initial and boundary conditions for CROCO

```
>> make_clim
```

Create the initial and boundary conditions for PISCES

```
>> make_biol
```

Exit from Matlab

```
>> exit
```

5. Running

Copy the `namelist_pisces_cfg` in your Run directory

```
cp /mnt/lustre/users/cethe/SpringSchool2022/Namelist/namelist_pisces_cfg .
```

Copy the job file in your run directory

```
cp /mnt/lustre/users/cethe/SpringSchool2022/Scripts/run_pisces.pbs .
```

Edit the `run_pisces.pbs` file to set your email address

```
vi run_pisces.pbs
```

```
#!/bin/bash
#PBS -l select=1:ncpus=4:mpiprocs=4
#PBS -P WCHPC
#PBS -q express
#PBS -l walltime=0:10:00
######PBS -l walltime=2:10:00
#PBS -m abe
#PBS -M Your.email@address.com
```

Run the model configuration

```
qsub run_pisces.pbs
```

5. Explore the results

Copy the following ferret scripts:

```
cp /mnt/lustre/users/cethe/SpringSchool2022/Scripts/*.jnl .
cp /mnt/lustre/users/cethe/SpringSchool2022/Scripts/sc_r32_thetas7.a.dat .
```

Explore the model output surface fields using the following ferret scripts:

Launch ferret

```
ferret
```

```
yes? go plot_xy_dyn.jnl
yes? go plot_xy_nut_surf.jnl
yes? go plot_xy_bio_surf.jnl
yes? go plot_xz_nut.jnl
yes? go plot_xz_bio.jnl
```

For information : You can modify the zonal sections by changing the following parameters in the ferret scripts (*.jnl):

- "j1" is the latitude
- "lev1" is the colorbar range (min, max,delta)
- "zz1" is the depth range
- "i1i2" is the zonal (longitudinal) range
- "l1" and "l2" are the two time indices shown in some of the figures.

Session 2: Model tuning

In this session, we will explore the sensitivity of the model to some parameters and see how they impact the solution. To do this, you will modify the values of some biogeochemical parameters in the configuration namelist of PISCES and compare the results to the run performed yesterday as a control run. To plot the results, you will use the Ferret visualization tool. Of course, you can use your favorite tool for this. You will find information about Ferret at the end of this document.

---

1. Log-on onto the cluster

- open a terminal
- connect to the cluster

```
ssh -X yourlogin@scp.chpc.ac.za
yourlogin@scp.chpc.ac.za's password::
```

Then reserve a calcul node

```
qsubi1
```

Go in your run directory

```
cd lustre/PISCES/croco/Run_PISCES
```

2. Sensitivity tests

To perform the sensitivity experiments, you will for each experiment modify the parameters in the namelist_pisces_cfg file where only the parameters to be modified were copied. All the parameters of PISCES can be found in the namelist_pisces_ref file. This latter file **should not be modified** as it served as a reference file.

2.1 Suppression of the source of iron from sediment

In this fist sensitivity test, we investigate the impact of turning off the sediment iron source

open file “namelist_pisces_cfg” and set the flag of the sedimentary Fe input to false

```
vi namelist_pisces_cfg
```

```
&nampissbc   !   parameters for inputs deposition
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   ln_ironsed =  .false.   ! boolean for dust input from atmosphere
/
```

Edit the `run_pisces.pbs` file to change the name of your experiment

```
vi run_pisces.pbs
```

```
EXPNAME='BENGUELA_NOIRONSED'
```

Run the model

```
qsub run_pisces.pbs
```

Run the following ferret scripts (change the name of the input netcdf file at the beginning of the ferret scripts you use):

```
go plot_xy_diff_Fe_run1_run2.jnl
go plot_xy_diff_CHL_NO3surf_run1_run2.jnl
go plot_xy_diff_PO4_Sisurf_run1_run2.jnl
```

Important : when the run is complete, reset the modified parameter to its original setting.

2.2 Suppression of the dust input of nutrient

Here, we investigate the impact of turning off the atmospheric dust deposition of nutrient:.

open file “namelist_pisces_cfg” and set the flag of the dust input to false:

```
vi namelist_pisces_cfg
```

```
&nampissbc  !   parameters for inputs deposition
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   ln_dust   =  .false.   ! boolean for dust input from atmosphere
/
```

Edit the `run_pisces.pbs` file to change the name of your experiment

```
vi run_pisces.pbs
```

```
EXPNAME='BENGUELA_NODUST'
```

Run the model

```
qsub run_pisces.pbs
```

Run the following ferret scripts (change the name of the input netcdf file at the beginning of the ferret scripts you use):

```
go plot_xy_diff_Fe_run1_run2.jnl
go plot_xy_diff_CHL_NO3surf_run1_run2.jnl
go plot_xy_diff_PO4_Sisurf_run1_run2.jnl
```

Important : when the run is complete, reset the modified parameter to its original setting.

2.3 Modification of the P-I slope parameters

In this sensitivity test, we investigate changes in the P-I slope parameters for diatoms and nanophytoplankton on chlorophyll and nutrient concentrations.

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

First both parameters are increased (+50%):
Open file “namelist_pisces_cfg” and set the pi-slope parameters to 3:

```
vi namelist_pisces_cfg
```

```
&namp4zprod !   parameters for phytoplankton
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   pislopen    =  3.        ! P-I slope
   pisloped    =  3.        ! P-I slope  for diatoms
/
```

Edit the `run_pisces.pbs` file to change the name of your experiment

```
vi run_pisces.pbs
```

```
EXPNAME='BENGUELA_PISLOPEP50'
```

Run the model

```
qsub run_pisces.pbs
```

Run the following ferret scripts (change the name of the input netcdf file at the beginning of the ferret scripts you use):

```
go plot_xy_diff_CHL_NO3surf_run1_run2.jnl
go plot_xy_diff_PO4_Sisurf_run1_run2.jnl
go plot_xz_diff_Chl_NO3_run1_run2.jnl
```

Important : when the run is complete, reset the modified parameter to its original setting.

Here, the P-I slope parameters for diatoms and nanophytoplankton are decreased by 50%: Open file “namelist_pisces_cfg” and set the pi-slope parameters to 1:

```
vi namelist_pisces_cfg
```

```
&namp4zprod !   parameters for phytoplankton
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   pislopen   =  1.        ! P-I slope
   pisloped   =  1.        ! P-I slope  for diatoms
/
```

Edit the `run_pisces.pbs` file to change the name of your experiment

```
EXPNAME='BENGUELA_PISLOPEM50'
```

Run the model and repeat the same steps as before. Don’t forget to rename the output file.

Important : when the run is complete, reset the modified parameter to its original setting.

2.4 Modification of both zooplankton grazing rate (50% decrease)

We investigate the impact of decreasing the grazing pressure of micro and mesozooplankton on phytoplankton.

```
!----------------------------------------------------------------------
&namp4zzoo     !   parameters for microzooplankton for PISCES std      -
!----------------------------------------------------------------------
   grazrat    =  3.0      ! maximal zoo grazing rate
/
```

```
!----------------------------------------------------------------------
&namp4zmes     !   parameters for mesozooplankton for PISCES std       -
!----------------------------------------------------------------------
   grazrat2   =  0.75     ! maximal mesozoo grazing rate
/
```

Open file “namelist_pisces_cfg” and change the grazrat and grazrat2 parameters as follow :

```
vi namelist_pisces_cfg
```

```
&namp4zzoo   !  parameters for microzooplankton for PISCES
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   grazrat   =  1.5      ! maximal zoo grazing rate
/
```

```
&namp4zmes  !  parameters for mesozooplankton for PISCES
!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
   grazrat2   =  0.35    ! maximal zoo grazing rate
/
```

Edit the `run_pisces.pbs` file to change the name of your experiment

```
EXPNAME='BENGUELA_GRAZINGM50'
```

Run the model configuration :

Run the following ferret scripts (change the name of the input netcdf file):

```
go plot_xy_diff_CHL_NO3surf_run1_run2.jnl
go plot_xy_diff_PO4_Sisurf_run1_run2.jnl
go plot_xy_zoo_surf.jnl
```

Outputs visualization

To visualize the results of the simulations that you will produce throughout this session, you can use the Ferret software, which can be launched directly in the cluster. This visualization environment is particularly convenient and fast for exploring the variables in the generated netcdf files. Of course, you can also use your favorite software. 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 BENGUELA_REF_avg.nc
yes? use BENGUELA_NODUST_avg.nc
```

To display the variables in the imported files

```
yes? show data
```

To unload all the files

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

plot NO3 concentration at the surface (k=32) at the time l=10

```
yes? shade/NO3[d=1,k=32,l=10]
```

To compare two simulations

```
yes? shade/z=0:200/l=10/k=32 NO3[d=1] - NO3[d=2]
```

To plot the total chlorophyll at surface

```
yes? Let CHL = DCHL + NCHL
yes? shade/k=32/l=10 CHL[k=32,d=1]
```

To compute a variable integrated over depth

```
yes? shade/l=10 CHL[d=1,k=20:32@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

Session 3: Adding a new tracer in PISCES

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

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.

Practical work :

Based on the **ligand tracer 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

In this equation:

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

Complementary information :

In CROCO/PISCES code:

- `tra(:,:,:)` is the trend and `trb(:,:,:)` the concentration value of the biogeochemical tracer
- `xstep` is the time step duration for biology relative to a day
- `nyear_len` is the length in days of the year

The practical work will be done by creating a new BENGUELA_CDOM configuration.

```
cd croco-dev_2022_PISCES ; vim ./create_config.bash
MY_CONFIG_NAME='Run_BENGUELA_LR_CDOM'
./create_config.bash
```

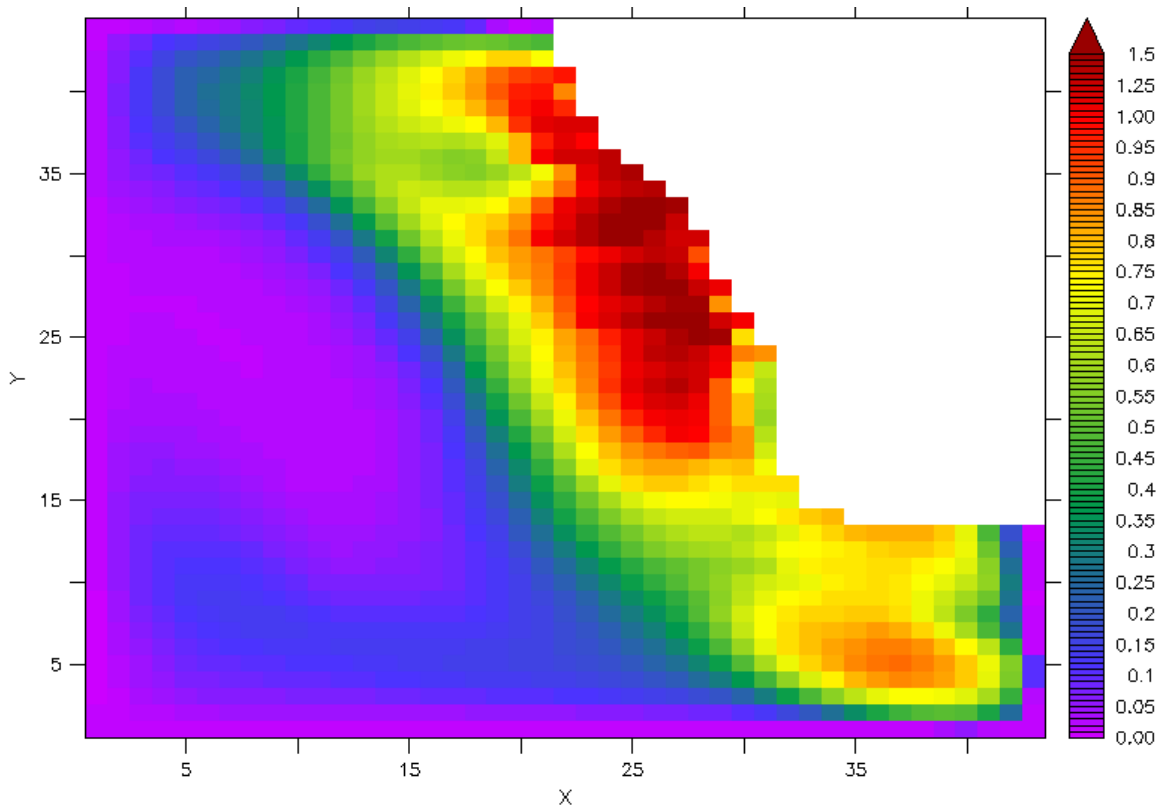
You will copy the routines to be modified in the run-directory `Run_BENGUELA_LR_CDOM`

```
- croco-dev_2022_PISCES/OCEAN : cppdefs.h, ncscrum.h, param.h ,
  ana_initial.F, analytical.F, get_bry_bio.F, get_psource_ts.F,
  get_tclima.F , init_scalars.F, Makefile

- croco-dev_2022_PISCES/PISCES : par_pisces.F90, sms_pisces.F90
  trcini_pisces.F90, p4zbio.F90, p4zprod.F90
```

You will also create a module computing the sources and sinks of the CDOM tracer, `p4zcdom.F90` and add this module within the list of PISCES routines in the `Makefile`.

Here is the result you should get for CDOM in CROCO (monthly mean concentration at surface)



CDOM concentration (mmol/m3)

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

First, define a new cpp key, as shown for the key_ligand, in:
../CROCO/croco-dev_2022_PISCES/OCEAN/cppdefs.h:

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

In the source code, define the identifiers in the same way as those of the ligand tracer.

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

- Increase the total number of tracers (jptra = 25 with ligand)

vim ../croco-dev_2022_PISCES/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 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
/
```