Getting started with the IPSL tools: modipsl and libIGCM

Answers to questions in training_exercises_2024.pdf

Revised for 2024 training sessions.

1. Check your quota

Question 1a:

- *Is the quota individual? What happens to the other users if you exceed the quota?*

Quota is attributed for each project for the entire group and not individually, so be careful of your own practices to avoid blocking all the group.

- *What kind of quotas do you have?*

There are quota on space and quota on inodes.

- *What is the meaning of “non_files"?*

“Non_files” means directories or symbolic links.

- *Which type of files do you store in your HOME? your WORK? and your STORE?*

In your $HOME: small files, such as configuration files (`.bash_login` for example).

In your $WORK: your working space, i.e. where you download the files for your simulation and launch it.

In your $STORE: for archive, where outputs of your simulations will be copied. For big files.

Question 1b:

- *Is the quota individual? What happens to the other users if you exceed the quota?*

Quota is attributed for each project for the entire group and not individually, so be careful of your own practices to avoid blocking all the group.

- *What kind of quotas do you have?*

There are quota on space and quota on inodes.

- *What is your global score?*

This score reflects how close you are from the recommended usage of the STOREDIR filesystem.

- *What is the meaning of “non_files"?*

“Non_files” means directories or symboliks links.

- *Which type of files do you store in your HOME? your WORKDIR? and your STOREDIR?*

In your $HOME: small files, such as configuration files (`.bash_login` for example).

In your $CCCWORKDIR: your working space, i.e. where you download the files for your simulation and launch it.

In your $CCCSTOREDIR: for archive, where outputs of your simulations will be copied. For big files.

- *What is the size of the files that you are supposed to store in the STOREDIR?*

The expected size of files is between 10Gb and 1Tb.

2. Installing and compiling

2.1 Extract the ICOLMDZOR_v7 configuration to run LMDZOR experiment

Question 2a: *Using* `./model -h` *command, find which version of LMDZ, ORCHIDEE and libIGCM are currently defined in the configuration* **IPSLCM7_TP2024**. *Note the SVN revision number and SVN branch or tag name. Verify that you can find the same information in the* `mod.def` *file.*

Answer 2a:

	`./model -h IPSLCM7_TP2024`	`vim mod.def;  /IPSLCM7_TP2024`
LMDZ	Component: LMDZ6/trunkbranches/IPSL-CM6A-MR Revision: 4515	#-C- IPSLCM7_TP2024 LMDZ6/trunkbranches/IPSL-CM6A-MR 4515
ORCHIDEE	Component : branches/ORCHIDEE_2_2/ORCHIDEE Revision: 8378 trunk/ORCHIDEE Revision: 8378	#-C- IPSLCM7_TP2024 branches/ORCHIDEE_2_2/ORCHIDEE 8378 #-C- IPSLCM7_TP2024 trunk/ORCHIDEE 8378
libIGCM	Component: trunk/libIGCM Revision : 1623	#-#-C- IPSLCM7_TP2024 trunk/libIGCM 1623

Question 2b: *Have all the components of the IPSLCM7 model been extracted? If not, which ones are missing?*

Answer 2b: the NEMO, OASIS and INCA components have not been downloaded as they are not used in ICOLMDZOR.

2.2 Compile with the resolution 144x142x79

Question 2c: Open the compilation script and try to find all available options for the compilation. Find out which resolution is the default one, then launch the compilation for both grids "regular" (with the resolution "144x142x79") and "icosahedral" (in this case the resolution is choose when we launch the experiment) the resolution 144x142x79.

Answer 2c:

Compilation option available : "-parallel" "-arch" "-xios" "-debug" "-dev" "-prod" "-resol_atm" "CE0L" "-full" "-full_xios" "-full_lmdz" "-full_orch" "-netcdf_lib_seq"
Default resolution : resol_atm=144x142x79

Question 2d: *How can you recompile the whole code? Open the compilation script and check the different script options.*

Answer 2d:

Use the option “-full”

```
./compile_ipslcm7.sh -resol_atm 144x142x79 -full
```

3. Basic simulations

3.2 Define and launch your first simulation of 1 day

Question 3a: *Which files are produced and where are they stored? You did not find any files in the archive directory at `$STORE` (Jean Zay) or `$CCCSTOREDIR` (Irene)? Why not?*
Answer 3a: Files produced during the run are listed in *Script_Output_*.0001* (see below ***"DIR AFTER RUN EXECUTION").*** Some of them (restart files, output files, debug files) are stored on `$CCCSCRATCHDIR` (Irene) or `$SCRATCH` (JeanZay) and ordered by component, for example `$SCRATCH/IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/ATM/Output`
`$SCRATCH/IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/ATM/Restart`
`$SCRATCH/IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/ATM/Debug`

These files are stored on `$CCCSCRATCHDIR` (Irene) or `$SCRATCH` (JeanZay) rather than archive directory (*`$STORE` (Jean Zay)* or *`$CCCSTOREDIR` (Irene)*) because of `SpaceName=TEST`, which means it is not a "production" run (`SpaceName=PROD)` or a "development" run (`SpaceName=DEVT).`

3.3 Continue the simulation 4 more days

Question 3b: *How many times did the job go into the queue?*
Answer 3b:
To do 4 more days we change `DateEnd` in `config.card` as

```
DateEnd=1980-01-05
```

Your simulation will be submitted 4 times, because it's a succession of 4 simulations of 1 day. At the end of each **period** the simulation is submitted one more time to launch the next **period.**

Question 3c: *create a new simulation of 5 days, always with `PeriodLength=1D`, but with a different `NbPeriodsPerJob` parameter to submit the job only one time to the queue.*
Answer 3c:

```
NbPeriodsPerJob=5
```

in the Job file allows to perform 5 runs within the same Job submission instead of 1 run per Job submission (since PeriodLength=1D, 1 run means here 1D run).

Question 3d: *Look into your first simulation run.card file. How long did one day take? Did every day take the same time?*
Answer 3d: The 1st day (= 1st run) takes more time than the following days. That's because initial work (maps interpolations,...) is needed by ORCHIDEE to start a simulation with no restart files. Following days take less time than the 1st one, because restart files are used to initialize the state of the models.

3.4 Create another simulation with pack

Question 3e: *explore the output directories, can you understand what was done ?*
Answer 3e: On scratch directory (`$CCCSCRATCHDIR` (Irene) or `$SCRATCH` (JeanZay)) , there is no output files in Output directories:
`$SCRATCH/IGCM_OUT/LMDZOR/DEVT/clim/MyJobTest2/ATM/Output`
That's because a pack step is achieved to concatenate output files and store the result on the archive directory (`$STORE` (Jean Zay) or `$CCCSTOREDIR` (Irene)). This pack step is activated by defining *PackFrequency* in config.card. The concatenated output files are stored on archive directory thanks to *SpaceName=DEVT*. As *PackFrequency=2M,* the pack step is done every 2 months runs. As the period of the simulation is 4 months, Output directories of archive directory contains files over the period 19800101_19800228 and files for the period 19800301_19800430 (see
`$STORE/IGCM_OUT/LMDZOR/DEVT/clim/MyJobTest2/ATM/Output).`

3.5 Use different forcing files

Question 3f: *which files are used as start.nc, startphy.nc, sechiba_rest_in.nc ? Read the Script_output file to answer this question.*
Answer 3f: in Script_Output file of the 1st period (=1st day), you can see the files used as restart files by each component of the model :

```
IGCM_sys_GetBuffer :
…IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/ATM/Restart/MyJobTest_19800105_restart.nc start.nc
IGCM_sys_GetBuffer :
…IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/ATM/Restart/MyJobTest_19800105_restartphy.nc startphy.nc
IGCM_sys_GetBuffer :
…IGCM_OUT/LMDZOR/TEST/clim/MyJobTest/SRF/Restart/MyJobTest_19800105_sechiba_rest.nc sechiba_rest_in.nc
```

Question 3g: *Verify in the Script_output file you use the file you want.*

Answer 3g: in Script_Output files :

```
IGCM_sys_Cp                                                                    :
/ccc/work/cont003/igcmg/igcmg/IGCM/SRF/PFTMAPS/CMIP6/ESA-LUH2v2/historical/15P
FT.v1/PFTmap_1980.nc PFTmap.nc
```

3.6 CREATE_clim and CREATE_amip: Experiments to create initial state files and boundary conditions for LMDZ

Question 3h: *Where can you find the output? Which files are produced and where are they stored?*

Answer 3h: Output files are found in the directory ***IGCM_OUT/LMDZ/ELC-144x142x79/ATM/Output*** on the archive filesystem $STORE at IDRIS or $CCCSTOREDIR at TGCC.

Output directory contains 3 sub-directories as follows :

- Boundary : boundary conditions files ELC-144x142x79_clim_limit.nc and ELC-144x142x79_climoz_LMDZ.nc.
- Grid : grid file ELC-144x142x79_grilles_gcm.nc contains information related to the grid used by the model (lon, lat,...).
- Restart : initial state files for physics and dynamics of LMDZ : ELC-144x142x79_clim_start.nc ELC-144x142x79_clim_startphy.nc.

Question 3i: *What type of calendar is used? How many days contains a year? Check also the number of time steps in the output file limit.nc. Do you know how you can change the calendar that has been used?*

Answer 3i: The calendar type is defined in config.card as follows :

```
CalendarType=noleap
```

A year of noleap calendar contains 365 days as you can see by dumping limit.nc output file :

```
netcdf ELC-144x142x79_clim_limit {
dimensions:
        points_physiques = 20306 ;
        time = UNLIMITED ; // (365 currently)
```

The calendar can be changed in config.card by choosing leap, noleap or 360d.

Question 3j: *Now create a new experiment clim_pdControl using the boundaries files created by ELC-144x142x79. For this you will modify the path in COMP/lmdz.card for start.nc, startphy.nc, limit.nc and climoz_LMDZ.nc files.*

Answer 3j: You have to modify lmdz.card as follows to use boundaries files previously generated and stored on archive directory:

```
ListNonDel=
…
(...IGCM_OUT/LMDZ/ELC2022-144x142x79/ATM/Output/Boundary/ELC-144x142x79_climoz_LMDZ.nc, climoz_LMDZ.nc),\
(...IGCM_OUT/LMDZ/ELC-144x142x79/ATM/Output/Boundary/ELC-144x142x79_clim_limit.nc, limit.nc),\
…
```

and as follows to use initial state files previously generated and stored on archive directory:

```
[InitialStateFiles]
List=
(.../IGCM_OUT/LMDZ/ELC2022-144x142x79/ATM/Output/Restart/ELC-144x142x79_clim_start.nc, start.nc),\
(.../IGCM_OUT/LMDZ/ELC2022-144x142x79/ATM/Output/Restart/ELC-144x142x79_clim_startphy.nc, startphy.nc)
```

Check config.card contains:

```
[Restarts]
#D- by default: config.card describes no restart for all components
#D-                 ie start from Levitus or limit files
#D- If you want to restart all components from the same simulation,
#D-       put OveRule flag to 'y' and set the next 3 parameters
OverRule=n
```

in order to use restart files defined in [InitialStateFiles] part of lmdz.card.
You can check the files used by the run in Script_Output :

```
IGCM_sys_Get :
…/IGCM_OUT/LMDZ/ELC-144x142x79/ATM/Output/Restart/ELC2022-144x142x79_cli
m_start.nc start.nc
IGCM_sys_Get :
…/IGCM_OUT/LMDZ/ELC-144x142x79/ATM/Output/Restart/ELC2022-144x142x79_cli
m_startphy.nc startphy.nc
…
IGCM_sys_Get :
…/IGCM_OUT/LMDZ/ELC2022-144x142x79/ATM/Output/Boundary/ELC2022-144x142
x79_climoz_LMDZ.nc climoz_LMDZ.nc
IGCM_sys_Get :
…/IGCM_OUT/LMDZ/ELC2022-144x142x79/ATM/Output/Boundary/ELC2022-144x142
x79_clim_limit.nc limit.nc
```

4. Debug

4.1 Debug : setup error

Question 4a: *What was the error?*
Answer 4a:

In Script_Output file you can see that the problem is coming during the running phase.

```
=========================================================================
EXECUTION of : /usr/bin/time srun  --cpu-bind=none --distribution=arbitrary --multi-prog ./run_file
Return code of executable : 143
IGCM_debug_Exit :  EXECUTABLE

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!   ERROR TRIGGERED   !!
!!   EXIT FLAG SET           !!
!------------------------------------!

0 - IGCM_debug_Exit (_0_)
IGCM_sys_Mkdir : ****/modipsl/config/IPSLCM7/*****/Debug
IGCM_sys_Cp : out_execution ****/modipsl/config/IPSLCM7/***/Debug/***_***_***_out_execution_error

to debug your experience, you can analyze the various files (*.err, *.out, and parameters files) available in the Debug/ directory
these files are managed by the list [OutputText] defined in componant's card (COMP/*.card)

=========================================================================
```

As indicated in the message, you can open the file Debut/*out_lmdz.x.err where there is this message :

```
0 klon_omp =            57
0 klon_omp_begin =             116
0 klon_omp_end =      172
0Abort(1) on node 0 (rank 0 in comm 928): application called MPI_Abort(comm=0x84000005, 1) - process 0
```

This message is not explicit. You can try to find more informations in Debug/*out_lmdz.x.out file. In this file, there is this error message

```
0 in abort_physic
0 Stopping in limit_read_mod
```

```
0 Reason = Pb d'ouverture du fichier de conditions aux limites
0 Houston, we have a problem, ierr =           1
```

It means that the LMDZ model in routine `limit_read_mod` is waiting for a boundary file (*"fichier de conditions aux limites"*), and has a problem opening it.

Then if you don't know the name of this specific file you need to open the routine model/LMDZ/libf/phylmd/limit_read_mod.F90 and look, for example, to the word "ouverture".

```
    IF (is_mpi_root) THEN ! Only master processus
        ierr = NF90_OPEN ('limit.nc', NF90_NOWRITE, nid)
        IF (ierr /= NF90_NOERR) CALL abort_physic(modname,&
               'Pb d''ouverture du fichier de conditions aux limites',1)
```

These lines mean that the code will try to open file limit.nc, and if it's failed, it will write "Pb d'ouverture du fichier de conditions aux limites".

Now you need to understand why the simulation failed to open limit.nc. For this you can read to the file Script_Output of your simulation, and look for limit.nc to know the one used.

You will see that this file is not loaded by the simulation. And that's why there is a problem. To solve it you need to indicate in COMP/lmdz.card the path of a boundary file to be used as limit.nc.
For example:
${R_IN}/ATM/LIMIT/AMIP.v20180427/interpol/${RESOL_ATM_XY}_eORCA1.2_365d/limit_1979_2008_clim.nc

```
[BoundaryFiles]
List=()
#
# With CMIP6 aerosols version 1
# With 3D ozone
ListNonDel=
(${R_IN}/ATM/LIMIT/AMIP.v20180427/interpol/${RESOL_ATM_XY}_eORCA1.2_365d/limit_1979_2008_clim.nc, limit.nc ),\
(${R_IN}/ATM/AEROSOLS/CMIP6/v1/${RESOL_ATM_XY}/L${RESOL_ATM_Z}/aerosols1850_from_inca.nc,         aerosols.nat.nc),\
(${R_IN}/ATM/AEROSOLS/CMIP6/v1/${RESOL_ATM_XY}/L${RESOL_ATM_Z}/aerosols1995_from_inca.nc,   aerosols1980.nc),\
(...)
```

Question 4b: *What was the error?*
Answer 4b:

In the Script_Output_* file you can find this error message:

```
cp: cannot stat
'/gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/144x142_eORCA1.2_367d/limit_1979_2008_clim.nc': No such
file or directory
IGCM_debug_Exit :  IGCM_sys_Get

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!   ERROR TRIGGERED   !!
!!   EXIT FLAG SET           !!
!------------------------------------!
```

it means that there is a problem during the copy of the file $path/limit_1979_2008_clim.nc. If you check the path of this file you will see that it doesn't exist. There is a typographical error. To fix it you can test the path section by section to find where is the problem

```
$ ls /gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/144x142_eORCA1.2_367d/limit_1979_2008_clim.nc
ls: cannot access
'/gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/144x142_eORCA1.2_367d/limit_1979_2008_clim.nc': No such
file or directory

$ ls /gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/144x142_eORCA1.2_367d
ls: cannot access '/gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/144x142_eORCA1.2_367d': No such file or
directory

$ ls /gpfswork/rech/psl/commun/IGCM/ATM/LIMIT/AMIP.v20180427/interpol/
144x142_eORCA1.2_360d
144x142_eORCA1.2_gregorian
256x256_eORCA1.2_365d
nbp160_365d
nbp320_365d
nbp40_365d
nbp60_365d
144x142_eORCA1.2_365d
256x256_eORCA1.2_360d
256x256_eORCA1.2_gregorian
nbp160_gregorian  nbp40_360d
nbp40_gregorian  nbp80_365d
```

So the problem is because we write 144x142_eORCA1.2_367d instead of 144x142_eORCA1.2_365d in the path of the `limit.nc` file in `COMP/lmdz.card`.

Question 4c: *What was the error?*
Answer 4c:

In Script_Output file you can see that the problem is coming during the running phase.

```
===========================================================================
EXECUTION of : /usr/bin/time srun  --cpu-bind=none --distribution=arbitrary --multi-prog ./run_file
Return code of executable : 143
IGCM_debug_Exit :  EXECUTABLE
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!   ERROR TRIGGERED   !!
!!   EXIT FLAG SET           !!
!------------------------------------!

0 - IGCM_debug_Exit (_0_)
IGCM_sys_Mkdir : ****/modipsl/config/IPSLCM7/*****/Debug
IGCM_sys_Cp : out_execution ****/modipsl/config/IPSLCM7/***/Debug/***_***_***_out_execution_error

to debug your experience, you can analyze the various files (*.err, *.out, and parameters files) available in the Debug/ directory
these files are managed by the list [OutputText] defined in componant's card (COMP/*.card)

====================================================================
```

As indicated in the message, you can open the file Debug/*out_lmdz.x.err where there is this error message.

```
0 nf95_open tauswstrat.2D.nc:
0 No such file or directory
0forrtl: warning (528): IEEE_UNDERFLOW is signaling
```

As previously LMDZ model has a problem with an input file. In this case the message indicates that the file `tauswstrat.2D.nc` is not found. One more time you can check in Script_Output to confirm that this file wasn't copied. But if you open COMP/lmdz.card to add it to the boundary list, you can see that it's already here. The problem is difficult to understand: there's a space at the end of the line

```
(${R_IN}/ATM/LIMIT/AMIP.v20180427/interpol/${RESOL_ATM_XY}_eORCA1.2_365d/limit_1979_2008_clim.nc, limit.nc),\
```

and libIGCM understands this as the end of the game, so it doesn't copy the following lines. To solve this problem you need to remove this space after “\”.

The symbol “\” must be the last character of the line otherwise the following lines will be ignored.

4.3 Compilation in debug mode

Question4d: *What causes the bug ?*

the operation

```
day_end = day_ini - nday/(day_step-day_step)
```

will generate a division by zero because `day_step - day_step = 0`

4.4 Use of RUN_DIR directory to debug without libIGCM infrastructure

Question 4e: Do you understand what these files are?

script_lmdz.x.ksh, prog_lmdz.x.sh, script_xios.x.ksh, run_file, prog_xios.x.sh, hostlist, hostlist_template, components_binary.txt, compiler.txt, Job_****, arch.env → files to run the simulation
lmdz.x and xios.x are executables

other files are needed by models to run :

- parameters files (like gcm.def, vert.def, traceur.def, etc.)
- initial state files (like start.nc, startphy.nc, sechiba_rest_in.nc, etc.)
- definitions xml files for output (context*xml, file*xml, field*xml)
- other netcdf files are the model boundary condition files. They are listed by model component on the COMP/*card.

5. Post_processing : how to quickly visualise the main diagnostics of a simulation

5.1 Launch 2 years with default Time Series and default C-ESM-EP atlas

Question 5a: *What is different from previous experiment directories ?*
As explained in the statement, there is no Driver directory. And there is a new subdirectory : `cesmep_lite`

Question 5b: *Once the simulation and post-processing are complete (~15 minutes), check Time Series in the archive directory (see in the following directories: `IGCM_OUT/[...]/JobName/***/Analyse/TS_MO`).*
Answer 5a: Time Series files have been created and stored in the archive directory (`$STORE` (Jean Zay) or `$CCCSTOREDIR` (Irene)). Have a look at files in:
`[...]/IGCM_OUT/OL2/DEVT/test/MyPostExp/SRF/Analyse/TS_MO`
and
`[...]/IGCM_OUT/OL2/DEVT/test/MyPostExp/SBG/Analyse/TS_MO`

For details about how to manipulate output files, see exercise 9.

Note that TS files were done for 2 years (`***_19010101_19021231_***.nc`). If we had continued the simulation for more years, the TS files would have been calculated every 2 years (because we chose `TimeSeriesFrequency=2Y` in `config.card`).

As we used `PackFrequency=1Y` for Pack frequency, you can note that files in `RESTART`, `DEBUG`, `SBG/Output/MO` and `SRF/Output/MO` have been created every year.

5.2 Add variables to Time Series and relaunch with the TimeSeries_Checker.job

Question 5c: *How to check that “z0h” and “z0m” variables exist in the simulation output?*
Answer 5c: Go to the simulation output directory (in `$STORE` (Jean Zay) or `$CCCSTOREDIR` (Irene)) for SRF component:
`[...]/IGCM_OUT/OL2/DEVT/test/MyPostExp/SRF/Output/MO`

Then you can use, for instance, `ncdump` command (`|grep` is just to avoid seeing all variables):

`ncdump -h MyPostExp_19010101_19021231_1M_sechiba_history.nc | grep -i z0h`
and
`ncdump -h MyPostExp_19010101_19021231_1M_sechiba_history.nc | grep -i z0m`

You may also use `ncdump -v z0h …_sechiba_history.nc` to see the values of z0h variable.

You may also use `ncview` tool to have a look at the data.

More details can be found in exercise 9.

Question 5d: *How to use `TimeSeries_Checker.job`?*
Answer 5d: Don't forget to add `z0h` and `z0m` in `COMP/sechiba.card`. Then create a new `POST_REDO` directory and copy `config.card`, `COMP`, `POST`, and `run.card` in it, as explained in:
https://forge.ipsl.jussieu.fr/igcmg_doc/wiki/Doc/CheckDebug#RestarttheTimeSerieswithTimeSeries_checker.job
Copy `TimeSeries_Checker.job` script in `POST_REDO` directory:

```
cd POST_REDO/
cp ../../../../libIGCM/TimeSeries_Checker.job .
```

Before launching the `TimeSeries_Checker.job` script, you have to modify the `POST_REDO/run.card` file as follows to allow the creation of new Time Series over the whole simulation period:
`TimeSeriesCompleted=`

Run the script in interactive mode:

```
./TimeSeries_Checker.job
```

You have first to answer "**n**" to the question "`Run for real (y/n)?`" asked when you launch the script. This will allow you to see the two sechiba missing variables:

```
2023-01-04 18:05:53 --Debug1--> SRF
```

```
2023-01-04 18:05:53 --------Debug2--> 2D time series activated with 200Y chunck for
Post_1M_sechiba_history
2023-01-04 18:05:53 ------------Debug3--> Missing time series from 1M_sechiba_history in
/gpfsstore/rech/psl/rdzt910/IGCM_OUT/OL2/DEVT/test/MyPostExp/SRF/Analyse/TS_MO:
2023-01-04 18:05:53 --------------Debug3-->
MyPostExp_19010101_19021231_1M_z0h.nc
2023-01-04 18:05:53 --------------Debug3-->
MyPostExp_19010101_19021231_1M_z0m.nc
2023-01-04 18:05:53 --------Debug2--> 95% files OK. for period 19010101-19021231
```

Then, run `TimeSeries_Checker.job` again and answer “**y**” to create the Time Series for missing variables.

Question 5e: *Check that Time Series for z0h and z0m were created.*
Answer 5e:
`TimeSeriesChecker.job` had generated missing variables and had stored them in :
`[...]/IGCM_OUT/OL2/DEVT/test/MyPostExp/SRF/Analyse/TS_MO/MyPostExp_19010101_19021231_1M_z0h.nc`
and
`[...]/IGCM_OUT/OL2/DEVT/test/MyPostExp/SRF/Analyse/TS_MO/MyPostExp_19010101_19021231_1M_z0m.nc`

If you run `TimeSeriesChecker.job` again (answer “**n**”), it will indicate that 100% of Time Series requested have been generated:

```
2023-01-04 18:12:26 --Debug1--> SRF
2023-01-04 18:12:26 --------Debug2--> 2D time series activated for
Post_1M_sechiba_history
2023-01-04 18:12:26 --------Debug2--> 100% files OK. for period 19010101-19021231
```

8. Modify output using XIOS

The command grep will return two lines on the file src_sechiba/hydrol.f90

```
grep precip_ src_*/* | grep xios
src_sechiba/hydrol.f90:    CALL xios_orchidee_send_field("precip_rain",precip_rain/dt_sechiba)
src_sechiba/hydrol.f90:    CALL xios_orchidee_send_field("precip_snow",precip_snow/dt_sechiba)
```

8.1 Create a new output file for ORCHIDEE

Question 8a: *Verify that this new file is created and TimesSeries of two variables exist: since these variables are daily outputs, you have to search into ...SRF/Analyse/TS_DA/*
Answer 8a: The new file was packed and stored on archive directory $STORE at IDRIS or $CCCSTOREDIR at TGCC :

```
ncdump -h
…/IGCM_OUT/OL2/DEVT/test/MyPostExp2/SRF/Output/DA/MyPostExp2_19010101_19021231_1D_myoutput_orch.nc
…
        float rainfall(time_counter, lat, lon) ;
                rainfall:long_name = "Rainfall rate" ;
                rainfall:units = "mm/s" ;
                rainfall:online_operation = "average" ;
                rainfall:interval_operation = "1800 s" ;
                rainfall:interval_write = "1 d" ;
                rainfall:cell_methods = "time: mean (interval: 1800 s)" ;
                rainfall:_FillValue = 9.96921e+36f ;
                rainfall:missing_value = 9.96921e+36f ;
                rainfall:coordinates = "time_centered" ;
        float snowfall(time_counter, lat, lon) ;
                snowfall:long_name = "Snowfall rate" ;
                snowfall:units = "mm/s" ;
                snowfall:online_operation = "average" ;
                snowfall:interval_operation = "1800 s" ;
                snowfall:interval_write = "1 d" ;
                snowfall:cell_methods = "time: mean (interval: 1800 s)" ;
                snowfall:_FillValue = 9.96921e+36f ;
                snowfall:missing_value = 9.96921e+36f ;
                snowfall:coordinates = "time_centered" ;
…
```

and time series have been created and stored:

```
…/IGCM_OUT/OL2/DEVT/test/MyPostExp2/SRF/Analyse/TS_DA/MyPostExp2_19010101_19021231_1D_rainfall.nc
…/IGCM_OUT/OL2/DEVT/test/MyPostExp2/SRF/Analyse/TS_DA/MyPostExp2_19010101_19021231_1D_snowfall.nc
```

8.2 Enable a new output file in LMDZ

Question 8b : *Verify that this new file is created and that it contains the slp variable.*
Answer 8b: The new file is created and stored on scratch filesystem :
…/IGCM_OUT/LMDZOR/TEST/clim/MyJobTestLMDZ/ATM/Output/HF/**MyJobTestLMDZ_19800101_19800105_HF_myoutput_lmdz.nc**
and this file contains :

```
ncdump -h MyJobTestLMDZ_19800101_19800105_HF_myoutput_lmdz.nc
…
float slp(time_counter, lat, lon) ;
                slp:long_name = "Sea Level Pressure" ;
                slp:units = "Pa" ;
                slp:online_operation = "average" ;
                slp:interval_operation = "900 s" ;
                slp:interval_write = "1 h" ;
                slp:cell_methods = "time: mean (interval: 900 s)" ;
                slp:_FillValue = 9.96921e+36f ;
                slp:missing_value = 9.96921e+36f ;
                slp:coordinates = "time_centered" ;
…
```

9. Output files manipulations

9.1 Network Common Data Form (NetCDF) format

Question 9a: *Look at the file structure, how is it composed ? Explore other variables or components (SBG, MBG, OCE, ICE…). Are they structured in the same way ?*

NetCDF is a self-describing data format and all NetCDF files are structured in the same way. The ncdump command allows you to access information (*dimensions, description* and *variables* sections) about data the file contains.

```
netcdf MyJobTest_19800101_19800130_1D_histday {
dimensions:
    axis_nbounds = 2 ;
    lon = 144 ;
    lat = 143 ;
    presnivs = 79 ;
    time_counter = UNLIMITED ; // (30 currently)
variables:
    float lat(lat) ;
       lat:axis = "Y" ;
       lat:standard_name = "latitude" ;
       lat:long_name = "Latitude" ;
       lat:units = "degrees_north" ;
    float lon(lon) ;
       lon:axis = "X" ;
       lon:standard_name = "longitude" ;
       lon:long_name = "Longitude" ;
       lon:units = "degrees_east" ;
    float presnivs(presnivs) ;
       presnivs:name = "presnivs" ;
       presnivs:standard_name = "Vertical levels" ;
       presnivs:units = "Pa" ;
    double time_centered(time_counter) ;
       time_centered:standard_name = "time" ;
       time_centered:long_name = "Time axis" ;
       time_centered:calendar = "360_day" ;
       time_centered:units = "seconds since 1980-01-01 00:00:00" ;
       time_centered:time_origin = "1980-01-01 00:00:00" ;
       time_centered:bounds = "time_centered_bounds" ;
    double time_centered_bounds(time_counter, axis_nbounds) ;
    double time_counter(time_counter) ;
       time_counter:axis = "T" ;
       time_counter:standard_name = "time" ;
       time_counter:long_name = "Time axis" ;
       time_counter:calendar = "360_day" ;
       time_counter:units = "seconds since 1980-01-01 00:00:00" ;
       time_counter:time_origin = "1980-01-01 00:00:00" ;
       time_counter:bounds = "time_counter_bounds" ;
```

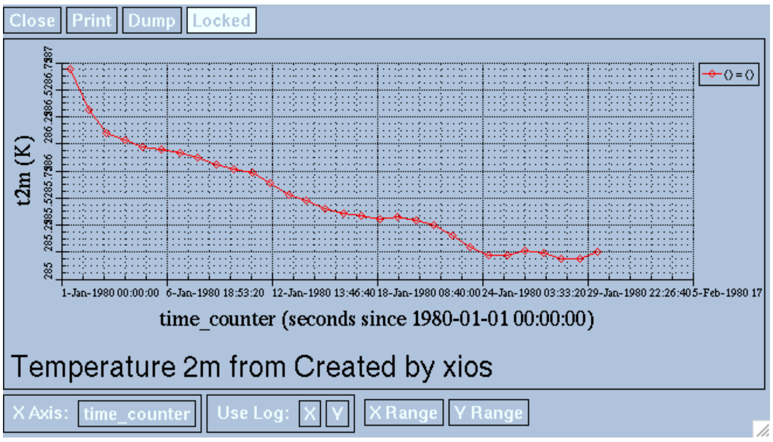
9.2 NetCDF Operator (NCO)

Question 9b: *Check the output file content using ncdump -h*
Only the variable “t2m” is in the output file with all its dimensions and metadata.

Question 9c and 9e: *Compare computing performances between NCO and CDO treatment.*
For this average computation, CDO treatment is faster than NCO treatment which needs to compute explicitly weights needed for the global and zonal means.

9.4 NetCDF Visual browser (NCView)

Use NCview with global t2m average.



NCview quick snapshot of global mean temperature file

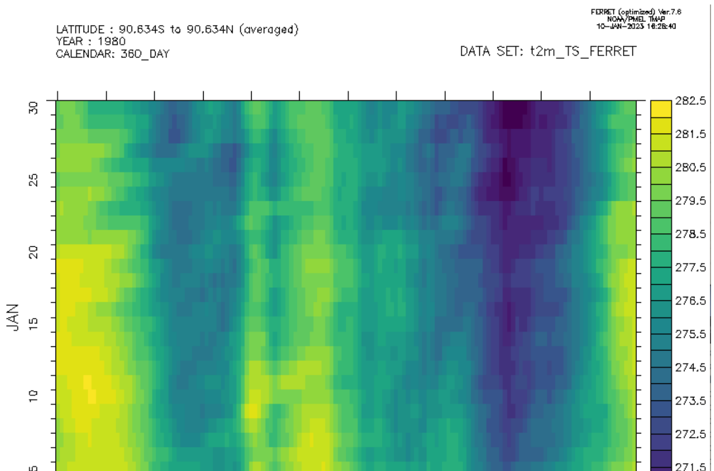
9.5 Ferret

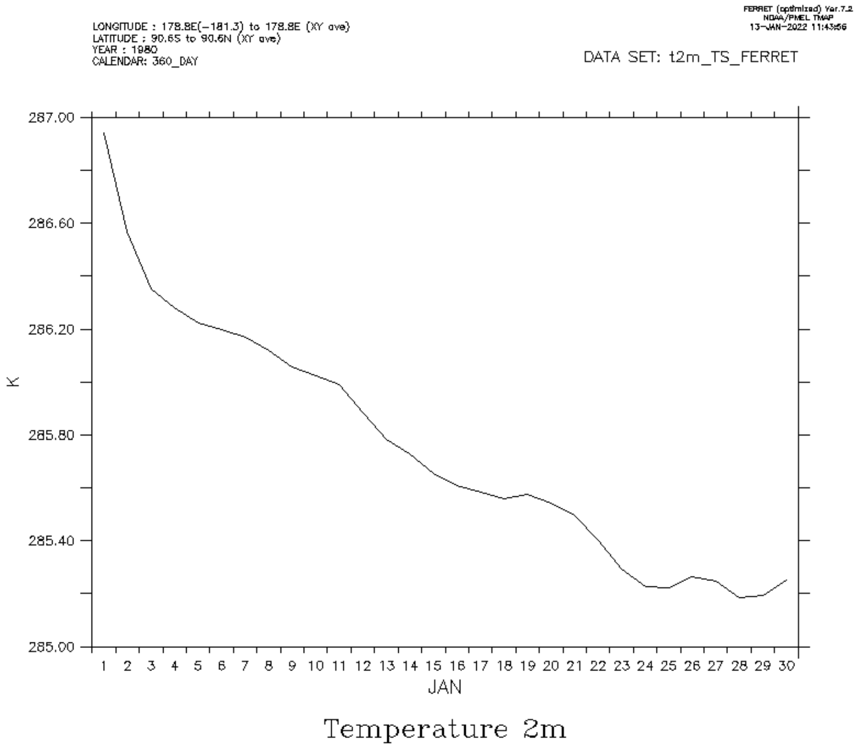
Ferret show infos.

```
yes? show data/f 2
      currently SET data sets:
      2> ./t2m_TS_FERRET.nc  (default)

 name title                            I      J      K      L
 T2M  Temperature 2m                   1:144  1:143  ...    1:30
            K on grid GDZ1 with 9.96921E+36 for missing data
            X=178.8E(-181.3):178.8E   Y=90.6S:90.6N

  time range: 01-JAN-1980 12:00 to 30-JAN-1980 12:00 Calendar:360_DAY
```





Ferret : plot global mean temperature

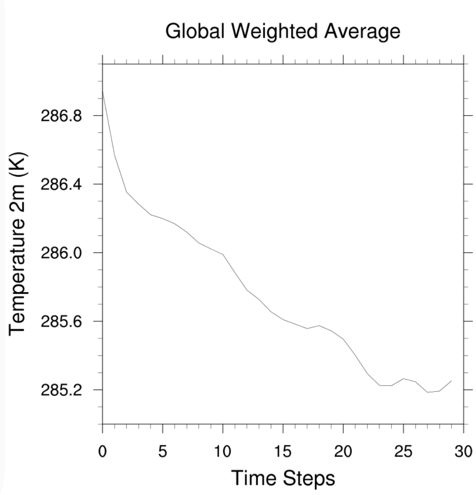
9.6 NCAR Command Language (NCL)

Question 9f: *Look with ncdump to the new NCL file.*

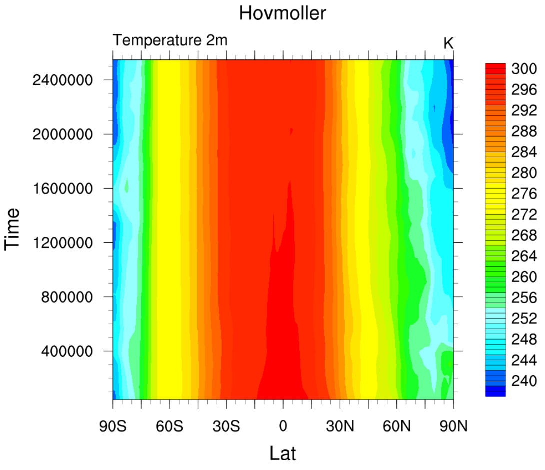
```
>> ncdump -h t2m_TS_NCL.nc
netcdf t2m_TS_NCL {
dimensions:
   time_counter = 30 ;
   lat = 143 ;
   lon = 144 ;
variables:
   float t2m(time_counter, lat, lon) ;
          t2m:long_name = "Temperature 2m" ;
          t2m:units = "K" ;
          t2m:online_operation = "average" ;
          t2m:interval_operation = "900 s" ;
          t2m:interval_write = "1 d" ;
          t2m:cell_methods = "time: mean (interval: 900 s)" ;
          t2m:_FillValue = 9.96921e+36f ;
          t2m:missing_value = 9.96921e+36f ;
          t2m:coordinates = "time_centered" ;
   double time_counter(time_counter) ;
          time_counter:axis = "T" ;
          time_counter:standard_name = "time" ;
          time_counter:long_name = "Time axis" ;
          time_counter:calendar = "360_day" ;
          time_counter:units = "seconds since 1980-01-01 00:00:00" ;
          time_counter:time_origin = "1980-01-01 00:00:00" ;
```

```
        time_counter:bounds = "time_counter_bounds" ;
    float lat(lat) ;
        lat:axis = "Y" ;
        lat:standard_name = "latitude" ;
        lat:long_name = "Latitude" ;
        lat:units = "degrees_north" ;
    float lon(lon) ;
        lon:axis = "X" ;
        lon:standard_name = "longitude" ;
        lon:long_name = "Longitude" ;
        lon:units = "degrees_east" ;
}
```

Question 9g & 9h: *Plot NCL output files ; ave.png for global mean and zonal.png for zonal one*



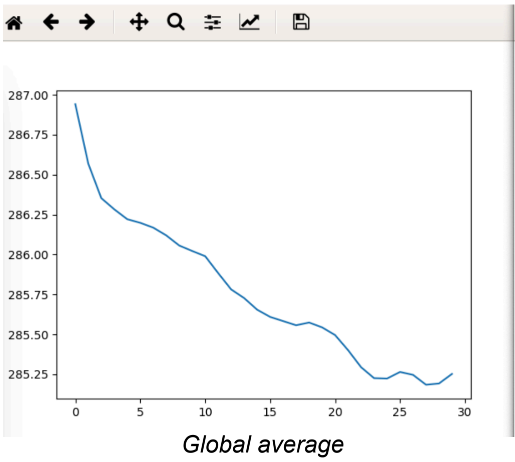
NCL : display of ave.png file



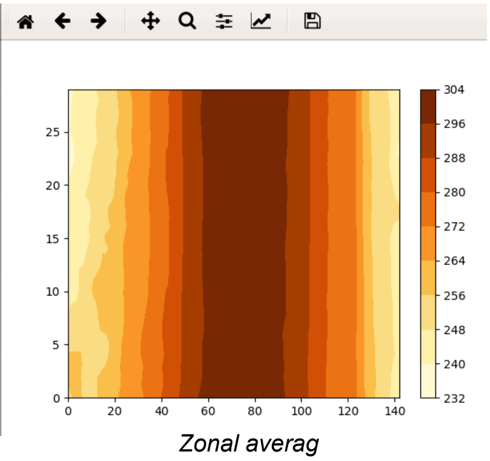
NCL: display of the zonal.png file

9.7 Python

Section 9.6 NetCDF4 & Xarray exercises: *Plot output files*



Global average



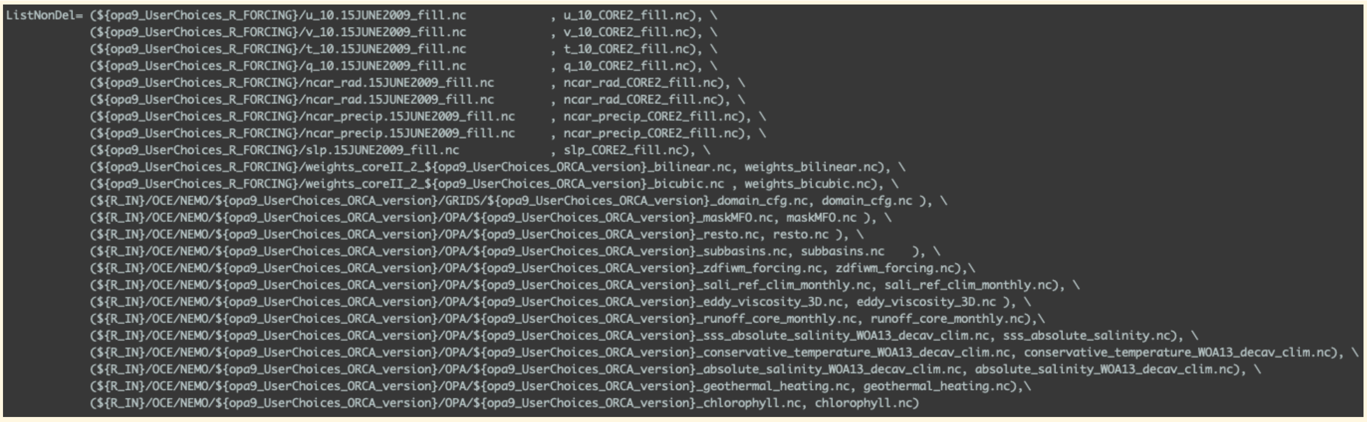
Zonal averag

10. Install and run NEMO-PISCES

10.1 Run a 1 month online experiment of NEMO-PISCES

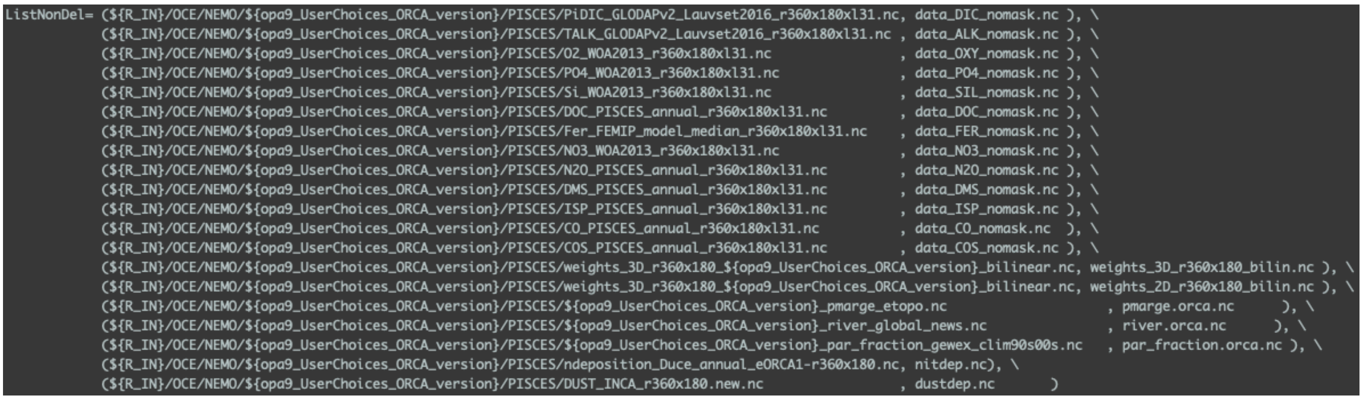
Question 10a: Explore the `COMP/opa9.card` and `COMP/pisces.card`

In the `opa9.card` file, you will find all the input files needed to run the dynamical core of NEMO such as atmospheric forcing, radiative forcing, and precipitation files etc…, as well as files related to the grid of the considered configuration etc.



```
ListNonDel= (${opa9_UserChoices_R_FORCING}/u_10.15JUNE2009_fill.nc          , u_10_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/v_10.15JUNE2009_fill.nc          , v_10_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/t_10.15JUNE2009_fill.nc          , t_10_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/q_10.15JUNE2009_fill.nc          , q_10_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/ncar_rad.15JUNE2009_fill.nc      , ncar_rad_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/ncar_rad.15JUNE2009_fill.nc      , ncar_rad_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/ncar_precip.15JUNE2009_fill.nc   , ncar_precip_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/ncar_precip.15JUNE2009_fill.nc   , ncar_precip_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/slp.15JUNE2009_fill.nc           , slp_CORE2_fill.nc), \
            (${opa9_UserChoices_R_FORCING}/weights_coreII_2_${opa9_UserChoices_ORCA_version}_bilinear.nc, weights_bilinear.nc), \
            (${opa9_UserChoices_R_FORCING}/weights_coreII_2_${opa9_UserChoices_ORCA_version}_bicubic.nc , weights_bicubic.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/GRIDS/${opa9_UserChoices_ORCA_version}_domain_cfg.nc, domain_cfg.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_maskMFO.nc, maskMFO.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_resto.nc, resto.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_subbasins.nc, subbasins.nc    ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_zdfiwm_forcing.nc, zdfiwm_forcing.nc),\
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_sali_ref_clim_monthly.nc, sali_ref_clim_monthly.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_eddy_viscosity_3D.nc, eddy_viscosity_3D.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_runoff_core_monthly.nc, runoff_core_monthly.nc),\
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_sss_absolute_salinity_WOA13_decav_clim.nc, sss_absolute_salinity.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_conservative_temperature_WOA13_decav_clim.nc, conservative_temperature_WOA13_decav_clim.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_absolute_salinity_WOA13_decav_clim.nc, absolute_salinity_WOA13_decav_clim.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_geothermal_heating.nc, geothermal_heating.nc),\
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/OPA/${opa9_UserChoices_ORCA_version}_chlorophyll.nc, chlorophyll.nc)
```

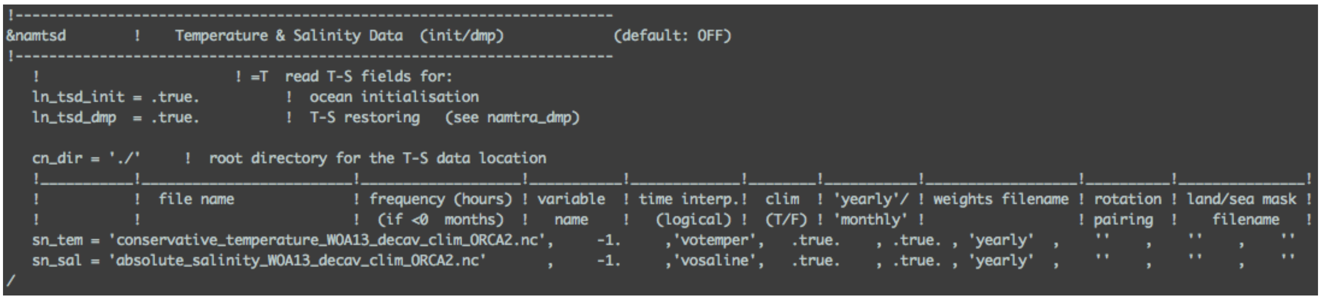
In the `pisces.card` file, you will find all the input files related to PISCES such as initial states of nutrients from the World Ocean Atlas 2009 observational climatologies, total DIC and alkalinity from the GLODAP V2.1 compiled data, biogeochemical inputs from rivers, atmospheric nitrate deposition, Fe inputs from dust and sediments etc.



```
ListNonDel= (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/PiDIC_GLODAPv2_Lauvset2016_r360x180xl31.nc, data_DIC_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/TALK_GLODAPv2_Lauvset2016_r360x180xl31.nc , data_ALK_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/O2_WOA2013_r360x180xl31.nc                , data_OXY_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/PO4_WOA2013_r360x180xl31.nc               , data_PO4_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/Si_WOA2013_r360x180xl31.nc                , data_SIL_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/DOC_PISCES_annual_r360x180xl31.nc         , data_DOC_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/Fer_FEMIP_model_median_r360x180xl31.nc    , data_FER_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/NO3_WOA2013_r360x180xl31.nc               , data_NO3_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/N2O_PISCES_annual_r360x180xl31.nc         , data_N2O_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/DMS_PISCES_annual_r360x180xl31.nc         , data_DMS_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/ISP_PISCES_annual_r360x180xl31.nc         , data_ISP_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/CO_PISCES_annual_r360x180xl31.nc          , data_CO_nomask.nc  ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/COS_PISCES_annual_r360x180xl31.nc         , data_COS_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/weights_3D_r360x180_${opa9_UserChoices_ORCA_version}_bilinear.nc, weights_3D_r360x180_bilin.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/weights_3D_r360x180_${opa9_UserChoices_ORCA_version}_bilinear.nc, weights_2D_r360x180_bilin.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/${opa9_UserChoices_ORCA_version}_pmarge_etopo.nc                  , pmarge.orca.nc      ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/${opa9_UserChoices_ORCA_version}_river_global_news.nc             , river.orca.nc      ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/${opa9_UserChoices_ORCA_version}_par_fraction_gewex_clim90s00s.nc  , par_fraction.orca.nc ), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/ndeposition_Duce_annual_eORCA1-r360x180.nc, nitdep.nc), \
            (${R_IN}/OCE/NEMO/${opa9_UserChoices_ORCA_version}/PISCES/DUST_INCA_r360x180.new.nc                 , dustdep.nc       )
```

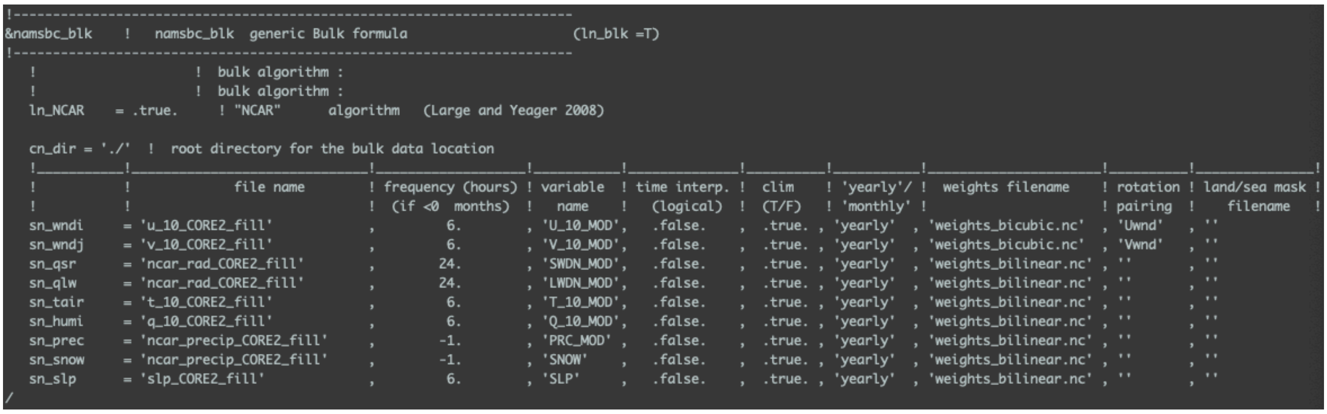
Question 10b: Explore in `PARAM/NAMELIST/ORCA2` the `namelist_core_clim_cfg` file

This file contains, among other parameters, the initial temperature and salinity states from the World Ocean Atlas 2013 climatologies (`&namstd`) as well as the names of the CORE 2 atmospheric forcing variables and their reading frequencies used for this configuration (`&namsbc_blk`).



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

   cn_dir = './'     !  root directory for the T-S 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 = 'conservative_temperature_WOA13_decav_clim_ORCA2.nc',     -1.     ,'votemper',    .true.    , .true. , 'yearly'  ,    ''    ,     ''    ,     ''
   sn_sal = 'absolute_salinity_WOA13_decav_clim_ORCA2.nc'       ,     -1.     ,'vosaline',    .true.    , .true. , 'yearly'  ,    ''    ,     ''    ,     ''
/
```



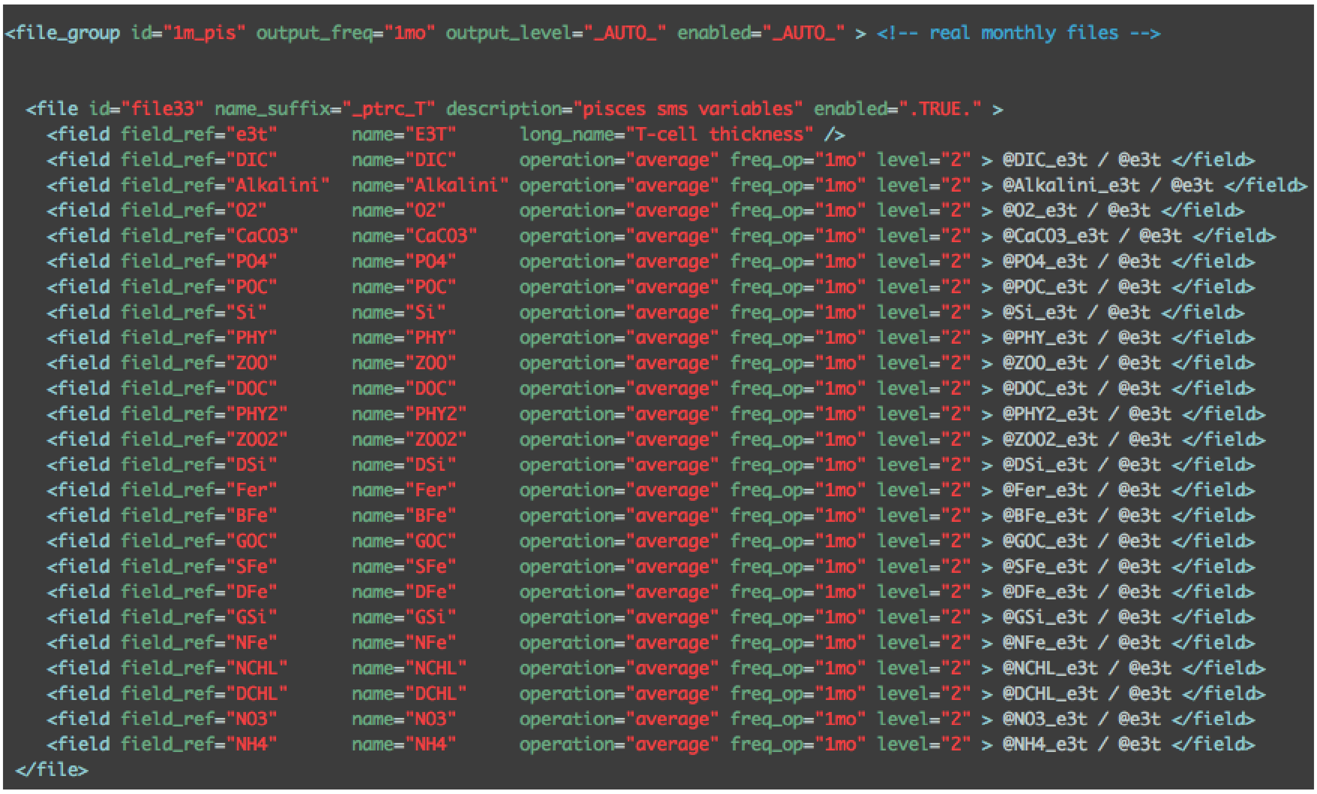
```
!-----------------------------------------------------------------------
&namsbc_blk    !   namsbc_blk  generic Bulk formula                    (ln_blk =T)
!-----------------------------------------------------------------------
   !                    !  bulk algorithm :
   !                    !  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 filename ! rotation ! land/sea mask !
   !           !                         !  (if <0  months)  !   name    !  (logical)  !  (T/F)  ! 'monthly' !                  ! pairing  !    filename   !
   sn_wndi     = 'u_10_CORE2_fill'       ,         6.        , 'U_10_MOD',  .false.    , .true. , 'yearly'  , 'weights_bicubic.nc'  , 'Uwnd'  , ''
   sn_wndj     = 'v_10_CORE2_fill'       ,         6.        , 'V_10_MOD',  .false.    , .true. , 'yearly'  , 'weights_bicubic.nc'  , 'Vwnd'  , ''
   sn_qsr      = 'ncar_rad_CORE2_fill'   ,        24.        , 'SWDN_MOD',  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_qlw      = 'ncar_rad_CORE2_fill'   ,        24.        , 'LWDN_MOD',  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_tair     = 't_10_CORE2_fill'       ,         6.        , 'T_10_MOD',  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_humi     = 'q_10_CORE2_fill'       ,         6.        , 'Q_10_MOD',  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_prec     = 'ncar_precip_CORE2_fill'  ,      -1.        , 'PRC_MOD' ,  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_snow     = 'ncar_precip_CORE2_fill'  ,      -1.        , 'SNOW'    ,  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
   sn_slp      = 'slp_CORE2_fill'        ,         6.        , 'SLP'     ,  .false.    , .true. , 'yearly'  , 'weights_bilinear.nc' , ''      , ''
/
```

Question 10c: Explore in `PARAM/XML/file_def_nemo*`

In `file_def_nemo-oce.xml`, `file_def_nemo-top.xml`, and `file_def_nemo-ice_ORCA2.xml` you will find all the variables for each model component, i.e., ocean, sea ice, and marine biogeochemistry, that are written to the output files for each output frequency.

Below is the list of the biogeochemical variables from the PISCES output file at a frequency of 1 month :



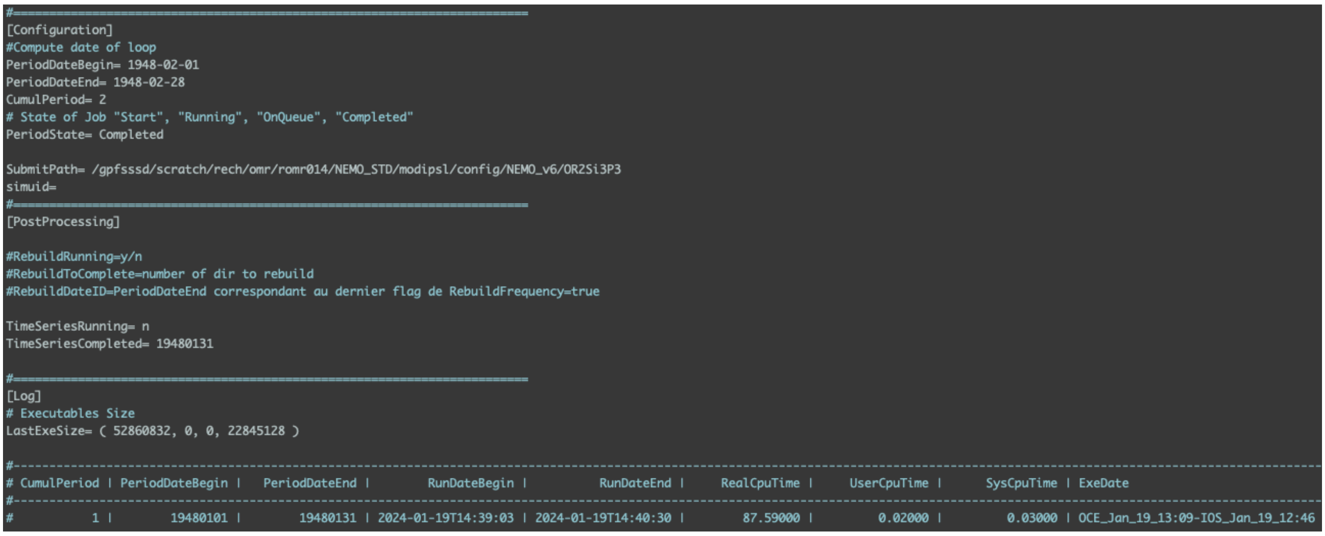
```
<file_group id="1m_pis" output_freq="1mo" output_level="_AUTO_" enabled="_AUTO_" > <!-- real monthly files -->


  <file id="file33" name_suffix="_ptrc_T" description="pisces sms variables" enabled=".TRUE." >
    <field field_ref="e3t"       name="E3T"      long_name="T-cell thickness" />
    <field field_ref="DIC"       name="DIC"      operation="average" freq_op="1mo" level="2" > @DIC_e3t / @e3t </field>
    <field field_ref="Alkalini"  name="Alkalini" operation="average" freq_op="1mo" level="2" > @Alkalini_e3t / @e3t </field>
    <field field_ref="O2"        name="O2"       operation="average" freq_op="1mo" level="2" > @O2_e3t / @e3t </field>
    <field field_ref="CaCO3"     name="CaCO3"    operation="average" freq_op="1mo" level="2" > @CaCO3_e3t / @e3t </field>
    <field field_ref="PO4"       name="PO4"      operation="average" freq_op="1mo" level="2" > @PO4_e3t / @e3t </field>
    <field field_ref="POC"       name="POC"      operation="average" freq_op="1mo" level="2" > @POC_e3t / @e3t </field>
    <field field_ref="Si"        name="Si"       operation="average" freq_op="1mo" level="2" > @Si_e3t / @e3t </field>
    <field field_ref="PHY"       name="PHY"      operation="average" freq_op="1mo" level="2" > @PHY_e3t / @e3t </field>
    <field field_ref="ZOO"       name="ZOO"      operation="average" freq_op="1mo" level="2" > @ZOO_e3t / @e3t </field>
    <field field_ref="DOC"       name="DOC"      operation="average" freq_op="1mo" level="2" > @DOC_e3t / @e3t </field>
    <field field_ref="PHY2"      name="PHY2"     operation="average" freq_op="1mo" level="2" > @PHY2_e3t / @e3t </field>
    <field field_ref="ZOO2"      name="ZOO2"     operation="average" freq_op="1mo" level="2" > @ZOO2_e3t / @e3t </field>
    <field field_ref="DSi"       name="DSi"      operation="average" freq_op="1mo" level="2" > @DSi_e3t / @e3t </field>
    <field field_ref="Fer"       name="Fer"      operation="average" freq_op="1mo" level="2" > @Fer_e3t / @e3t </field>
    <field field_ref="BFe"       name="BFe"      operation="average" freq_op="1mo" level="2" > @BFe_e3t / @e3t </field>
    <field field_ref="GOC"       name="GOC"      operation="average" freq_op="1mo" level="2" > @GOC_e3t / @e3t </field>
    <field field_ref="SFe"       name="SFe"      operation="average" freq_op="1mo" level="2" > @SFe_e3t / @e3t </field>
    <field field_ref="DFe"       name="DFe"      operation="average" freq_op="1mo" level="2" > @DFe_e3t / @e3t </field>
    <field field_ref="GSi"       name="GSi"      operation="average" freq_op="1mo" level="2" > @GSi_e3t / @e3t </field>
    <field field_ref="NFe"       name="NFe"      operation="average" freq_op="1mo" level="2" > @NFe_e3t / @e3t </field>
    <field field_ref="NCHL"      name="NCHL"     operation="average" freq_op="1mo" level="2" > @NCHL_e3t / @e3t </field>
    <field field_ref="DCHL"      name="DCHL"     operation="average" freq_op="1mo" level="2" > @DCHL_e3t / @e3t </field>
    <field field_ref="NO3"       name="NO3"      operation="average" freq_op="1mo" level="2" > @NO3_e3t / @e3t </field>
    <field field_ref="NH4"       name="NH4"      operation="average" freq_op="1mo" level="2" > @NH4_e3t / @e3t </field>
 </file>
```

Question 10d: Explore the `Script_Output` and `run.card` files

The `Script_Output` file provides you with all the information about the model configuration used in your experiment before and after the run. You will find there the different steps and files needed and produced during your experiment. If an error occurred in the execution of your run, this is where you should first look for information to understand what went wrong.

The run.card file provides you with information about the computing period, the status of the job and the real Cpu time.



```
#==========================================================================
[Configuration]
#Compute date of loop
PeriodDateBegin= 1948-02-01
PeriodDateEnd= 1948-02-28
CumulPeriod= 2
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState= Completed

SubmitPath= /gpfsssd/scratch/rech/omr/romr014/NEMO_STD/modipsl/config/NEMO_v6/OR2Si3P3
simuid=
#==========================================================================
[PostProcessing]

#RebuildRunning=y/n
#RebuildToComplete=number of dir to rebuild
#RebuildDateID=PeriodDateEnd correspondant au dernier flag de RebuildFrequency=true

TimeSeriesRunning= n
TimeSeriesCompleted= 19480131

#==========================================================================
[Log]
# Executables Size
LastExeSize= ( 52860832, 0, 0, 22845128 )

#-------------------------------------------------------------------------------------------------------------------------------------------------------------------
# CumulPeriod | PeriodDateBegin |   PeriodDateEnd |         RunDateBegin |           RunDateEnd |     RealCpuTime |     UserCpuTime |       SysCpuTime | ExeDate
#-------------------------------------------------------------------------------------------------------------------------------------------------------------------
#           1 |        19480101 |        19480131 | 2024-01-19T14:39:03 | 2024-01-19T14:40:30 |        87.59000 |         0.02000 |          0.03000 | OCE_Jan_19_13:09-IOS_Jan_19_12:46
```

Question 10e: Explore the output directories (`OCE/Output`, `ICE/Output`, `MBG/Output`)

The output directories are located on the scratch directory of your machine (Jean Zay or Irene) as the run was performed in the TEST case (cf. `config.card`):
`/$SCRATCH/IGCM_OUT/NEMO/TEST/clim/OR2Si3P1/`

To continue the simulation for 1 month, set in the `run.card` file the period state to "`OnQueue`":

```
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState= OnQueue
```

Change in the config.card file the date end of the job:

```
#-- Begin and end of Job
#-- "YYYY-MM-DD"
DateBegin=1948-01-01
DateEnd=1948-02-28
```

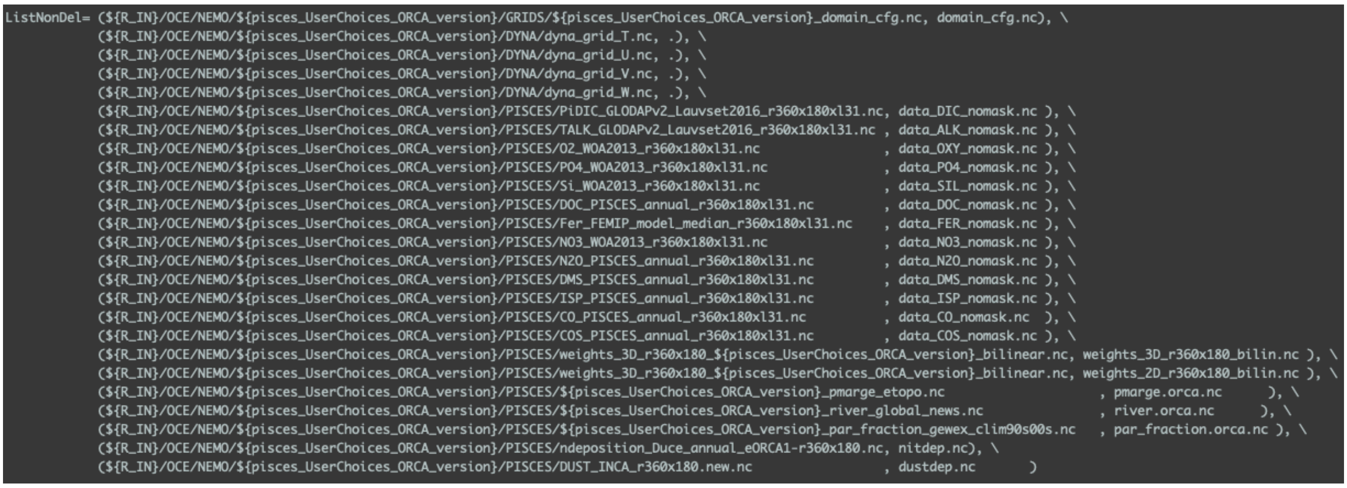
and submit the job:

```
sbatch Job_OR2Si3P1 / ccc_msub Job_OR2Si3P1
```

10.2 Run a 1-year offline experiment of NEMO-PISCES

Question 10f: Explore the `COMP/pisces.card`

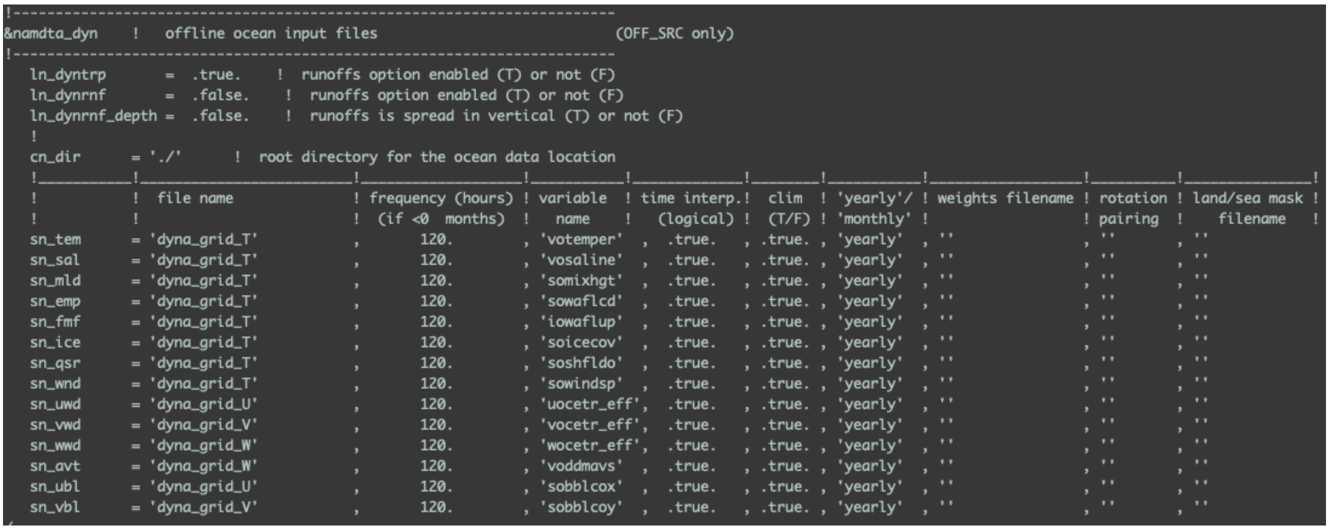
As with the previous configuration, you will find all the necessary input files for PISCES. In the *offline* case, the CORE atmospheric forcing files are no longer needed. Instead, you will find the oceanic forcing files (`dyna_grid_*.nc`) used to simulate the climatological state of the ocean over a year and used to compute the biogeochemical fields by the PISCES model.



```
ListNonDel= (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/GRIDS/${pisces_UserChoices_ORCA_version}_domain_cfg.nc, domain_cfg.nc), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/DYNA/dyna_grid_T.nc, .), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/DYNA/dyna_grid_U.nc, .), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/DYNA/dyna_grid_V.nc, .), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/DYNA/dyna_grid_W.nc, .), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/PiDIC_GLODAPv2_Lauvset2016_r360x180xl31.nc, data_DIC_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/TALK_GLODAPv2_Lauvset2016_r360x180xl31.nc , data_ALK_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/O2_WOA2013_r360x180xl31.nc                , data_OXY_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/PO4_WOA2013_r360x180xl31.nc               , data_PO4_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/Si_WOA2013_r360x180xl31.nc                , data_SIL_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/DOC_PISCES_annual_r360x180xl31.nc         , data_DOC_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/Fer_FEMIP_model_median_r360x180xl31.nc    , data_FER_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/NO3_WOA2013_r360x180xl31.nc               , data_NO3_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/N2O_PISCES_annual_r360x180xl31.nc         , data_N2O_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/DMS_PISCES_annual_r360x180xl31.nc         , data_DMS_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/ISP_PISCES_annual_r360x180xl31.nc         , data_ISP_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/CO_PISCES_annual_r360x180xl31.nc          , data_CO_nomask.nc  ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/COS_PISCES_annual_r360x180xl31.nc         , data_COS_nomask.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/weights_3D_r360x180_${pisces_UserChoices_ORCA_version}_bilinear.nc, weights_3D_r360x180_bilin.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/weights_3D_r360x180_${pisces_UserChoices_ORCA_version}_bilinear.nc, weights_2D_r360x180_bilin.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/${pisces_UserChoices_ORCA_version}_pmarge_etopo.nc                   , pmarge.orca.nc      ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/${pisces_UserChoices_ORCA_version}_river_global_news.nc              , river.orca.nc      ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/${pisces_UserChoices_ORCA_version}_par_fraction_gewex_clim90s00s.nc   , par_fraction.orca.nc ), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/ndeposition_Duce_annual_eORCA1-r360x180.nc, nitdep.nc), \
            (${R_IN}/OCE/NEMO/${pisces_UserChoices_ORCA_version}/PISCES/DUST_INCA_r360x180.new.nc                 , dustdep.nc       )
```

Question 10g: Explore in `PARAM/NAMELIST/ORCA2` the `namelist_offline_clim_cfg`

Here the `&namsbc_blk` section describing the atmospheric forcing variables (see question 10b) is now empty. Instead, it is the `&namdta_dyn` section that is filled with the variables read by PISCES to simulate the annual seasonal variations of the ocean and used to compute the biogeochemical variables in the *offline* mode.



```
!-----------------------------------------------------------------------
&namdta_dyn    !   offline ocean input files                            (OFF_SRC only)
!-----------------------------------------------------------------------
   ln_dyntrp       =  .true.    !  runoffs option enabled (T) or not (F)
   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'           ,        120.       , 'votemper'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_sal      = 'dyna_grid_T'           ,        120.       , 'vosaline'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_mld      = 'dyna_grid_T'           ,        120.       , 'somixhgt'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_emp      = 'dyna_grid_T'           ,        120.       , 'sowaflcd'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_fmf      = 'dyna_grid_T'           ,        120.       , 'iowaflup'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_ice      = 'dyna_grid_T'           ,        120.       , 'soicecov'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_qsr      = 'dyna_grid_T'           ,        120.       , 'soshfldo'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_wnd      = 'dyna_grid_T'           ,        120.       , 'sowindsp'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_uwd      = 'dyna_grid_U'           ,        120.       , 'uocetr_eff',  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_vwd      = 'dyna_grid_V'           ,        120.       , 'vocetr_eff',  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_wwd      = 'dyna_grid_W'           ,        120.       , 'wocetr_eff',  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_avt      = 'dyna_grid_W'           ,        120.       , 'voddmavs'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_ubl      = 'dyna_grid_U'           ,        120.       , 'sobblcox'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
   sn_vbl      = 'dyna_grid_V'           ,        120.       , 'sobblcoy'  ,  .true.    , .true. , 'yearly'  , ''               , ''       , ''
```

Question 10h: Explore the output directories where the output files are stored (`MBG/Output`)

Files can be found : `/$SCRATCH/IGCM_OUT/NEMO/TEST/clim/OR2OFFPIS1/`

Question 10i: Find where the reference namelist of PISCES is stored

To find the path where the namelist_pisces_ref is stored, open the `pisces.card` in the COMP directory of your configuration.

Question 10j: Explore the `namelist_pisces_ref`

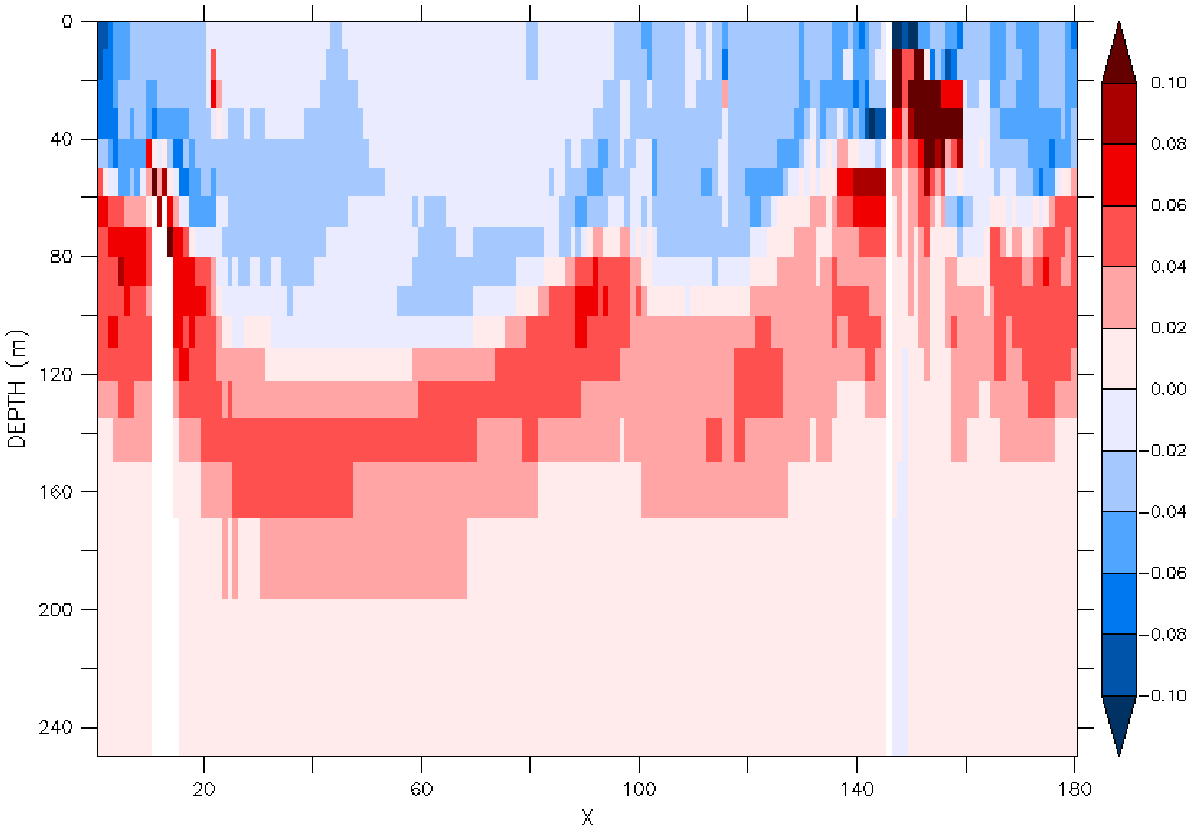
The `namelist_pisces_ref` file contains all the biogeochemical parameters that can be modified without recompiling the PISCES code. The values of all these parameters have been tuned for a global configuration at 2° resolution. They can be modified by copying the ones you want to modify into the `namelist_pisces_cfg` file. The values modified in `namelist_pisces_cfg` will then automatically overwrite the reference value in the code execution directory. Thus, you can always go back to the initial values of the modified parameters by opening the `namelist_pisces_ref` whose path is in the pisces.card file.

Question 10k: Explore the output directories (`MBG/Output`)

Files can be found: `/$SCRATCH/IGCM_OUT/NEMO/TEST/clim/OR2OFFPIS2/`

Question 10l: Compare the annual output files of the 2 *offline* configurations (OR2OFFPIS1, OR2OFFPIS2)

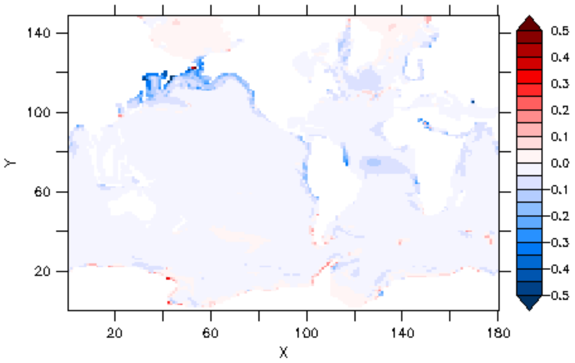
Here we plot the difference in annual means of total chlorophyll (DCHL+NCHL) and surface NO3, Si, and Fe (x1.e3) between the two experiments (`OR2OFFPIS2` minus `OR2OFFPIS1`). By increasing the sensitivity of phytoplankton to low light availability, the ability of phytoplankton to assimilate nutrients at depth increases. This is illustrated by the increase in the chlorophyll maximum at depth in the oligotrophic regions of tropical gyres, here in the North Atlantic associated with a deepening of the nutricline.



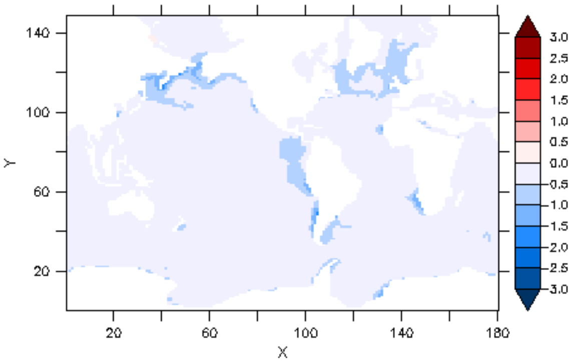
Chlorophyll Difference, Northern Hemisphere

The deepening of the nutricline in tropical regions results in a decrease in the supply of macronutrients to the surface, which is particularly visible in upwelling regions. This leads to a decrease in surface chlorophyll in tropical coastal regions for NO3 and Si.

This adaptation also leads to increased surface chlorophyll at high latitudes. The increase in primary production induces a decrease in NO3 concentrations near the coasts at high latitudes, a decrease even more pronounced for Si at high latitudes as diatoms predominate in these regions. In the tropical band, the increase in P-I slope leads to an increase in surface Fe concentrations. Here, the mechanisms that control this increase are more difficult to understand, but this illustrates the impact of a change in a parameter on the surface distribution of this trace metal.



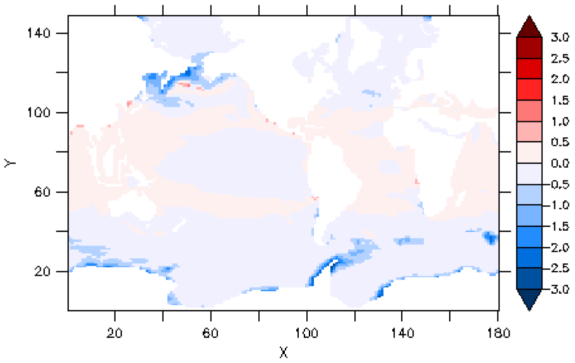
Total Chlorophyll Difference



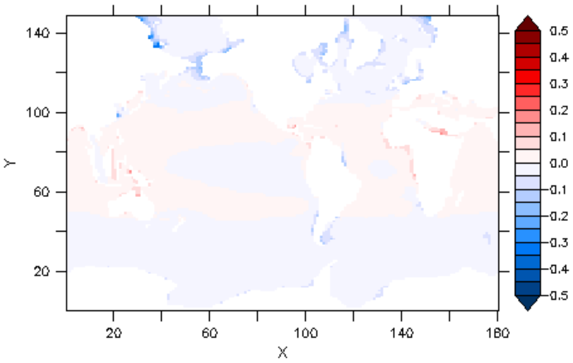
NO3 Difference

DEPTH (m) : 4.9999
TIME : 02-JUL-1850 12:00 NOLEAP

DEPTH (m) : 4.9999
TIME : 02-JUL-1850 12:00 NOLEAP



Si Difference



Fe Difference (nmol/L)

13. ICOLMDZOR configuration

Question13a: *use "-h" option to know all options of the compilation script `compile_icolmdzor.sh`. Which command will you launch to create executables for the regular grid and the icosaedral grid ? Which command will you launch to create only the executable for the icosaedral grid ?*

Answer 13a: options available for this script are "`-full, -clean_nemo, -regular_latlon, -resol_atm, -debug, -dev, -prod(default), -subconfig, -orch4, -orch22, -orchdriver`". You can find their meaning in the "-h" description.

To create the executable to run on icosaedral grid, and the executable to run on regular grid, you need to launch

```
./compile_ipslcm7.sh
```

The executables created will be

- `ce0l_144x142x79_prod.e`
- `gcm_144x142x79_orch22_prod.e`
- `orchideedriver.orch22_prod`
- `icosa_lmdz_orch22_prod.exe`
- `orchidee_ol.orch22_prod`
- `xios_server_prod.exe`

You can note that default regular resolution is 144x142x79, and default mode is prod.

To create only the executable to run on icosaedral grid, you need to launch

```
./compile_ipslcm7.sh -regular_latlon no
```

In this case you will create only executables for xios and ICOLMDZOR :

- `xios_server_prod.exe`
- `icosa_lmdz_orch22_prod.exe`
- `+ orchidee_ol exe`

To create only the executable to run on regular grid, you need to launch

```
./compile_ipslcm7.sh -regular_latlon yes -subconfig LMDZOR
```

In this case you will create only executables for xios and ICOLMDZOR :

- `xios_server_prod.exe`
- `gcm_144x142x79_orch22_prod.e`
- `+ orchidee_ol exe + create_etat0 exe`

Question 13b: *Do you see any differences with LMDZOR output files produced in part 2 ?*

Answer 13b: There is no difference in terms of grid the variables are written on. The model run on an icosaedral grid but the variables are interpolated (and written) by XIOS on a regular grid in output files in order to facilitate the analysis.