PIANO Reference Manual Python Interface for Assimilation in NemO

Claire Chauvin

PIANO Reference Manual
Python Interface for Assimilation in NemO

Claire CHAUVIN

3 décembre 2010

Table des matières

1 First steps with PIANO

PIANO intends to provide a user-friendly interface to a full run of assimilation. For the moment, PIANO is only about one full cycle of variationnal assimilation. Its aims is to manage runs and communications between the model NEMO and the Incremental minimization of the cost function available in NEMOVAR.

1.1 Description of the directory piano

The software PIANO is available in the VODA development repository in directory UTIL :

```
>cd ~/VODA/UTIL/python/piano
```

This directory contains all the python sources (files *.py), and two directories called `exp` and `config`

```
>ls
```

```
block.py          ! Python class defining one namelist block
blocklist.py      ! Python class defining the whole namelist
char_handling.py  ! Extract strings from a files
comments.py       ! Python class defining a comment in the namelist
config/           ! Directory containing the available NEMO configuration
descrlist.py      ! Python class defining the description file
exp/              ! Directory containing the user experiments
init_run.py       ! Method for initialization of PIANO
piano.py          ! Main
param.py          ! Python class defining a parameter (name, value, comment)
ps_handling.py    ! Manages processes
run_main.py       ! Methods that manage the run of one executable
hlt_handling.py   ! Method to compute RMS, coeff. corr of lin.tgt hyp test
npz2plot          ! Method to convert hlt_handling output to plot
tlmdiagmrg.py     ! Method to merge outputs files from tangent tests
```

Details on the python code architecture are given in section **??**. The directory `config` contains the following configurations :

```
>ls config/

GYRE_Z31/
ORCA2_Z31/
POMME_Z46/
SQB_Z11/
```

Each directory contains two default namelists

```
>ls GYRE_Z31/

namelist_nemo
namelist_nemovar
```

In the following, we make a distinction between the *namelist* - the file that contains all the definition of all the parameters needed by a run -, and a *namelist block* - a section of namelist defined by its name `&nam_block`. Recall that a NEMO run needs a namelist called `namelist` in the outer directory, and NEMOVAR asks for the namelist `namelist.nemovar` in the inner directory. These two namelists contain an *exhaustive* list of all the existing namelist blocks, needed by a run of NEMO (or NEMOTAM / NEMOVAR). The directory `exp` contains examples of user experiments :

```
>ls exp/

GYRE-ENACT
GYRE-SGL
ORCA2
POMME
```

An experiment is described by three files :

```
>ls GYRE-ENACT/

descr
namelist_nemo
namelist_nemovar
```

The description file `descr` contains all the parameters needed by a run *that are not in the namelists* :

```
> cat exp/GYRE-ENACT/descr

#
#  Description file for the Experiment
#
ndays=6                           ! to compute nitend and nwrite
grid=GYRE_Z31                     ! Configuration
compile=mac_intel                 ! Compiler
npx=1                             ! Parallel run:
npy=1                             !     not implemented
tdir=/tmp                         ! Contains output directories
exe_dir=/tmp                      ! Contains nemofcm_build dir
time_dir=/path_to_forcing_dir     ! For forcing files
common_dir=/path_to_common_dir    ! For data files
obs_dir=/path_to_obs_dir          ! Observation files
restart_file=/path_to/restart.nc  ! If ln_rstart
#bkgnorm_file=/path_to/background.normalization.nc
#run_solo=nemovar                 ! Run only one executable
#ndone=1                          ! Start assim cycle from a previous one
#trj_dir=/path_to/tam_trajectories! When only nemovar run
```

This example contains an exhaustive list of all the arguments defined in file `descr`. Some are not implemented yet (parallel architecture not managed).
The following parameters of final namelist are computed from `ndays` : `nitend`, `nstock, nwrite`, `nstockfl`, `nwritfl` and `nitiaufin`.
Parameter `grid` stands for the configuration name : piano loads the namelists defined in `config/grid`.
Four directories have to be declared in the description file, namely `tdir, exe_dir`, `obs_dir,` and `forcing_dir`.
If needed, the name of the restart file, and the background normalization file, can be provided.
The two last parameters can be usefull to : if the user wants to run only one executable, the parameter `run_solo` has to be specified. Also, it is possible to ask for piano to start the assimilation cycle from a previous one, stopped at iteration `ndone`.

Files `namelist_nemo` and `namelist_nemovar` contains a subset of the full namelist of a NEMO/NEMOVAR/NEMOTAM run. Indeed, they contain only the modifications made by the user to the default namelists located in `config/grid`.

For instance :

```
> cat exp/GYRE-ENACT/namelist_nemo

&nam_asminc
    ln_bkgwri = .true.
    nitbkg    = 0
    nitdin    = 0
&namtam
    ln_trjwri = .true.
&namobs
    ln_t3d = .true.
    ln_s3d = .true.
    ln_profb = .true.
    profbfiles='profb_01_fdbk_0000.nc'
```

piano merges `namelist_nemo` files loaded from `exp/GYRE-ENACT` and `config/GYRE_Z31`, and generates the `namelist` file that will be used for the NEMO run. In this example, the experiment asks for NEMO to assimilate an increment from a NEMOVAR run (`ln_bkgwri`), to generate the new trajectory (`ln_trjwri`), and to write the model equivalent of T and S profile observations (parameters set to true in `&namobs` and name of the observation file).

```
> cat exp/GYRE-ENACT/namelist_nemovar

&namobs
   ln_prf=.true.
   ln_t3d=.true.
   ln_s3d=.true.
&namalg
   noutmax = 8
   noutit  = 1
   nvarex  = 1
```

Here the experiment parameters specify the kind of assimilated observations, the number of outer loop, of the current loop (`noutit` that can eventually be updated by `ndone`), the kind of assimilation (4D-var in this case, since `nvarex=1`).

Files `namelist_nemo` and `namelist_nemovar` in the `exp` directory can be empty : in this case, the namelist used by the run is the one defined in the asso-

ciated configuration directory (for instance, GYRE_Z31 in the case of GYRE-ENACT experiment).

1.2 Running piano

1.2.1 Running piano - general features

In directory piano, run the command

```
>python piano.py exp_name
```

where exp_name is the name of an experiment. Depending on your experiment, several directories are created in tdir. If a full assimilation cycle is asked, then an outer and an inner directory are created, namely :

```
> cd $tdir

inner_GYRE-ENACT_mac_intel_1x1
outer_GYRE-ENACT_mac_intel_1x1
outer_tam_GYRE-ENACT_mac_intel_1x1
```

If tangent and adjoint model tests are done (parameters ln_tst=ln_tst_tam = .true. in namelist_nemovar, for instance) then the output is located in the outer_tam directory. Else, if run_solo=nemo (resp. nemovar), only the outer (resp. inner) directory is created, and modified. If a full assimilation cycle is done (run_solo deactivated), then both inner and outer directories are created (or overwritten).

A run is described by two parameters of description file : ndone and run_solo :

- run_solo=nemo : If ndone not defined in descr or ndone=1 : Delete outer directory. Outer loop counter noutit=ndone.
- run_solo=nemotam : Delete outer_tam directory. Executable model_tam.exe is used to run tests on tangent and adjoint model of NEMO : check that logical for such tests are activated in namelist namtst : ln_tst=.true., and at least one of the following flags has to be set to .true. :

  ln_tst_cpd_tam

  ln_tst_stp_tam

  ln_tst_tan

- run_solo=nemovar : If ndone not defined in descr or ndone=1 : Delete inner directory. Outer loop counter noutit=ndone.

- `run_solo=full_assim` (default) : if `ndone` not defined clean all directories. Else set `noutit=ndone`.

In the three last cases, the NEMOVAR namelist is initialized. For nemovar or full_assim, check that `nvarex=1` if you want a 4D-Var assimilation.

1.2.2 Running piano for Tangent Linear Hypothesis (HLT) test

HLT stands for Tangent Linear Hypothesis test.
The user needs first t define its experiments through *namelist_nemo* and *namelist_nemovar* and *descr* in the experiment. (see NEMOTAM User's Manual Reference) More particularly, blocks *namhlt* and *tl_tamtrj*. The HLT test is activated if *run_solo* is defined as *nemohlt*.

For instance :

```
> cat exp/GYRE-HLT/descr
compile=mac_g95
tdir=MY_PATH
exe_dir=MY_EXE_PATH
npx=1
npy=1
grid=GYRE_Z31
run_solo=nemohlt
nittrjfrq_hlt=75
increments_hlt=MY_INCREMENT_PATH
restart_file=MY_RESTART_PATH
#restart_hlt=MY_2ND_RESTART_PATH
#ndone=1
time_dir=MY_TIME_PATH
common_dir=MY_COMMON_PATH
```

The increments δx is read from an increments file *increments_hlt* or computed from two restart files *restart_file* and *restart_hlt*. The choice is done through the namelist parameter *ln_incdx* from *namhlt* namelist block.
The response time frequency for HLT is controlled with *nittrjfrq_hlt* (Note : it will orverwrite *nittrjfrq* when computing $\mathrm{M}(x+\delta x)$ and *nittrjfrq_tan* when computing $\mathrm{L}(\delta x)$.
If *ndone* is equal to 1, the driver assumes that $\mathrm{M}(x)$ is already available in *outer*

directory.
At the end of the process, a python compressed file, *hlt_output.npz* is saved in *outer_tam* directory. For graphical analysis you may use the standalone python routine *hlt_npz2plot.py* (python library *matplotlib* is required). It saves a *png* file, *hlt_output.png* in *outer_tam*.

1.2.3 Running piano for tangent test

The tangent test refers to the analysis of the tangent behaviour with respect to the direct module when the perturbation evolves as $p_n \delta X$. If p_n has a geometric progression, PIANO is able to perfom the loops on n and concatenate the final results into a single file *tan_diag.global_0000* .
To do so, the user must provide the element to compute $p_n = p_0 r^{-n}$:

- the initial value p_0 (***tlm_start***)
- the common ratio r (***tlm_fact***)
- the number of loops on n (***tlm_loop***)

The tangent test is activated if *run_solo* is defined as *nemotam* and *ln_tst_tan* is set to .true. (plus *ln_tst_tan_cpd* for instance) in *namtst* block namelist.
For instance :

```
> cat exp/GYRE-XXX/descr
compile=mac_g95
tdir=MY_PATH
exe_dir=MY_EXE_PATH
npx=1
npy=1
grid=GYRE_Z31
run_solo=nemotam
restart_file=MY_RESTART_PATH
time_dir=MY_TIME_PATH
common_dir=MY_COMMON_PATH
tlm_fact=0.1
tlm_start=1.
tlm_loop=1
```

2 Managing the input files

2.1 Forcing files (time_dir)

The Forcing files are usually time dependent data. They will be linked or copy to the outer and inner directories (and if needed test directories).
The directory is identified in the **descr** file under the name **time_dir**. The forcing files (surface boundary conditions, ice cover, fluxes, surface wind) are defined in the NEMO namelist (namelists `&namsbc_xxx`, activated by the logical `ln_xxx` in namelist block `&namsbc`). Once the `time_dir` is specified in `descr`, the loading of all the files activated in these namelist blocks should be transparent.
Note that for the moment *only the forcing files for ORCA2 has been implemented : the use of other files (by then other parts of the namelist) has not been tested.*
Since information on files is given in the namelist, note that it is possible to load different forcing files for NEMO and NEMOVAR.

2.2 Data files (common_dir)

We call Data files, files related to the configuration and general forcing file. The are usualyy used for realistic configuration such as ORCA2 and their file names are hardcoded into NEMO. The directory is identified in the **descr** file under the name **common_dir**. For the moment, they are automatically loaded for the ORCA2 grid, by mean of the method `init_real` in module `init_run.py`.

```
def init_orca(descrlist,common_files):
 grid=descrlist.get_value('grid')
 if grid=="ORCA2_Z31":
  forcing_files.append("data_1m_potential_temperature_nomask.nc")
  forcing_files.append("data_1m_salinity_nomask.nc")
  forcing_files.append("dist.coast.nc")
  forcing_files.append("bathy_level.nc")
  forcing_files.append("bathy_meter.nc")
  forcing_files.append("ahmcoef")
  forcing_files.append("coordinates.nc")
  forcing_files.append("ssh.nc")
  forcing_files.append("geothermal_heating.nc")
 forcing_files.append("slaReference.nc")
```

Note 1 : treatment of "slaReference.nc" does not rely on a specific configuration.

Note 2 : For the moment, piano works only for one cycle of assimilation : no specific treatment is done with respect to the date management, and so the storage is done in a very simple way. This question will be addressed when the implementation of several cycles of assimilation is done.

2.3 Observation files (obs_dir)

It is possible to load observation files of different format, as specified in the namelist `&namobs`, but the communication between NEMO and NEMOVAR is done only with feedback files.

Note that observations such as sla needs an additional set of data (that can be called *slaReference.nc*) to work.

3 Specific features

3.1 NEMOVAR tests

By default feedback files needed by nemovar (inner loop) is the output of a direct simulation (outer loop). In a inner loop test context it is possible to specify profiles and/or sla feedback files (key word : **profile_fdbk_file** and **sla_fdbk_file** respectively) when the key word `nemovar` is used for `run_solo` experiment. Then, your `descr` may contain the following lines :

```
run_solo=nemovar
profile_fdbk_file=$MY_PATH_TO_PROF_FB_FILE.nc
sla_fdbk_file=$MY_PATH_TO_SLA_FB_FILE.nc
```

3.2 Change Assimilation method between outer loops (nvarex_list)

For testing purpose we offer the possibility to change the `nvarex` value (means 3DFGAT or 4DVAR assimilation method for nvar equals 0 or 1 respectively). It can be done adding the parameter **nvarex_list** and the corresponding list in your `descr` file.

Parameter `nvarex` is a list a maximum `noutmax` (maximum number of outer

loops) elements. Each element is separated by a comma (,). If the number elements is less than `noutmax`, `nvarex` is set to the default value specified in namelist_nemovar (or by default in the default configuration).

Then, if you want to perform 4 outer loops in your assimilation process using 3DFGAT method for the first and the second set of inner loops and then 4DVAR for the thrids and the fourth sets, your `descr` may contain the following lines :

```
nvarex_list=0,0,1,1
```

4 Installation New Configuration

4.1 Input files

The management of input files for NEMO/NEMOVAR depends on the namelist, and on convention name defined in the code. There are several kind of input files :

Observation files : their name can either be defined by default in case of feedback files for profiles, or sla observations, or be present as arguments in the namelist block `namobs`.

Forcing files : surface boundary condition, ice cover, defined in the namelists `namsbc_xxx` : Their names are defined in the corresponding namelist block, when the flag in `nam_sbc` is activated. These kind of files are thus automatically treated in PIANO.

Data files (for instance the file `slaReferences.nc` needed when there is some SLA observation)

Files that are not SBC files, or that are about specific data, need a specific treatment in PIANO. Examples are available for ORCA2.

5 Available options in 'descr' file

Tables 1 and 2 give a list of available options for the *descr* file.

Références

Name	Description	Remark - Value
grid	configuration name	
compile	compiler name	
npx	x axis subdomain	! No Parallel run
npy	y axis subdomain	! No Parallel run
tdir	output directories path	
exe_dir	nemofcm_build dir	
time_dir	path to forcing files	
common_dir	path to common files	
obs_dir	path to observation files	
restart_file	path to restart.nc file	optional
bkgnorm_file	path to background.normalization.nc	optional
run_solo	run only one executable	optional, nemo, nemovar, nemotam, nemohlt
ndone	Start assim cycle from a previous one	optional, with nemohlt skip outer run (=1)
trj_dir	path to tam_trajectories	optional
restart_hlt	path to restart2.nc file	optional, with nemohlt if ln_incdx=.false.
increments_hlt	path to increments file	optional, with nemohlt if ln_incdx=.true.
profile_fdbk_file	path to profile fbdk file	optional, with run_solo=nemovar
sla_fdbk_file	path to sla fbdk file	optional, with run_solo=nemovar
nvarex_list	nvarex list	optional, = 0,0,1,1 for instance

TABLE 1 – Available options in *descr* file. Part 1/2

Name	**Description**	**Remark - Value**
tlm_fact	is r in $p_n = p_0 r^{-n}$	optional, with nemotam if ln_tst_tan=.true.
tlm_start	is p_0 in $p_n = p_0 r^{-n}$	optional, with nemotam if ln_tst_tan=.true.
tlm_loop	is n in $p_n = p_0 r^{-n}$	optional, with nemotam if ln_tst_tan=.true.

TABLE 2 – Available options in *descr* file. Part 2/2