

# The control print option in NEMO for debugging a non MPP .vs. MONO reproducibility

Claude Talandier (LOCEAN-IPSL)  
ESOPA team  
October 2005  
Version 2.

Technical report

- I. Introduction
- II. How to use it ?
- III. The control sum organization
- IV. Necessary conditions for the reproducibility

## I. Introduction

Ensuring the reproducibility - i.e. getting exactly the same results bit to bit of the elliptic solver - for each simulation done with NEMO components is very important. This is particularly true when doing MPP runs which represent the major part of performed runs by users for CPU time and computation speed issues.

Under specific conditions it is possible to obtain exactly the same results between a simulation done using one processor and the same one using many processors.

The only way to know if the reproducibility is ensured is to compare solver.stat files (which include the solver's iterations, the residual part and the right hand side member). The reproducibility is assumed when they are identical else something is wrong in the MPP case ; but there is no usefull information to detect the location of the reproducibility problem. To be able to find it we developed a simple option in the code which should help the user.

Since trend terms of the momentum and tracer equations are computed successively, the idea is to make control sums of trend terms over each sub-domain i.e. processor in the MPP case and control sums over strictly the same sub-areas in the mono-processor case. In doing that it is possible to compare prints over one specific processor area from a MPP run to the equivalent reference mono-processor prints over the same « virtual » area ; so the very first detected difference will indicate the possible subroutine induced in the non reproducibility error.

This option is implemented in most usual parameterizations within the ocean and the sea-ice components. It could be easily added in others component of NEMO or to a new implemented module.

## II. How to use it ?

Notice that this option is useful only if the *solver.stat* files resulting from the MPP and MONO runs are different. The control print is controlled by parameters in the ocean namelist. Prints are done in ASCII files for each sub-domain i.e. for each processor area.

### In the namelist :

The *In\_ctl* logical parameter allow to active the sum control print without having to recompile the whole code; it is associated to the following parameters in the ocean namelist *namrun* :

- **nictls** = starting indice along i- direction
- **nictle** = ending indice along i- direction
- **njctls** = starting indice along j- direction
- **njctle** = ending indice along j-direction
- **nisplt** = number of processors following i
- **njsplt** = number of processors following j

# The parameters **nictl[se]** **njctl[se]** are the indices used to bound each 2D or 3D array for the sum control over a sub-domain. For instance if **qt** is a 2D array the sum will be done so that : total = SUM ( qt(nictls : nictle, njctls : njctle) ).

- If setting these parameters to zero their default computed values will be respectively  
- for a given processor area – the first and the last point effectively computed by this processor in both direction, i.e without taking into account the overlap strips (See figure just below).

- If these parameters are different from zero, the control print will be done over the corresponding bounded area of the global domain and it will be useful only for debugging a mono-processor run not a MPP one. *Doing a MPP run in that case is a non sense.*

# **nisplt** and **njsplt** correspond to the splitted domain applied in both direction.

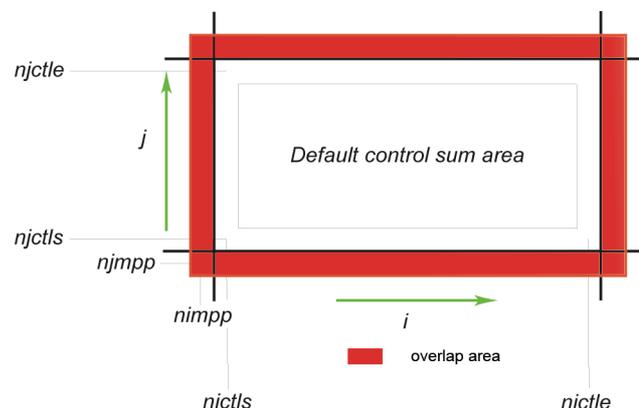
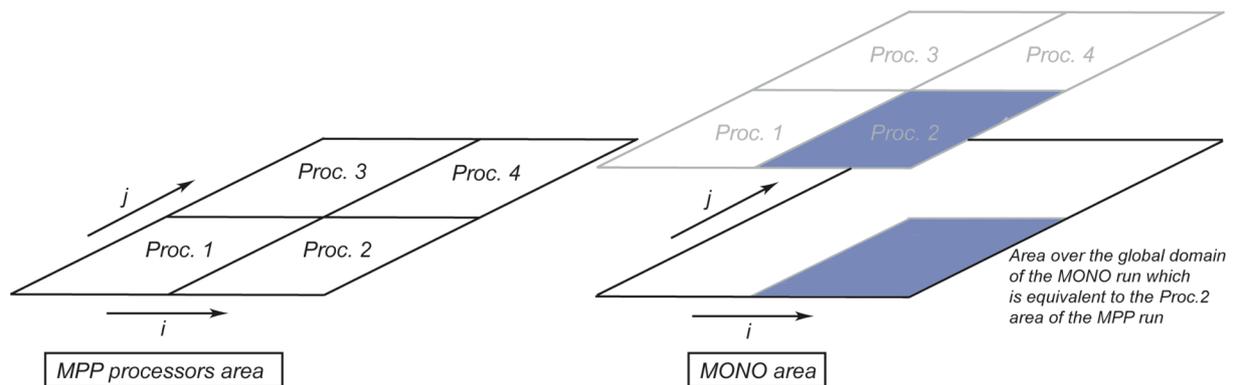


Figure above shows the location of main indices associated to each processor area and the default control sum area in white (overlap strips considered here correspond respectively to the parameters value **jpreci** and **jprej** equal to one).

The table below show how to set parameters to be able to compare control sums from a mono-processor run (**MONO** hereafter) to those of a 4 processors run (**MPP** hereafter), 2 along i- and 2 along j- :

Parameters \ Case	MONO	MPP
In_ctl =	.true.	.true.
nictls =	0	0
nictle =	0	0
njctls =	0	0
njctle =	0	0
nisp1t =	2	2
njspl1t =	2	2

In MPP, the control sum will be done over each processor area,  
 In MONO, the control sum will be done over each MPP corresponding area. See figure below.



### Output files :

The output ASCII files generated using this option are :

For the **MONO** run:  
 mono.output\_000  
 mono.output\_001  
 mono.output\_002  
 mono.output\_003

For the **MPP** run:  
 mpp.output\_000  
 mpp.output\_001  
 mpp.output\_002  
 mpp.output\_003

Processor number :  
 First processor area  
 Second processor area  
 Third processor area  
 Fourth processor area

the file *mono.output\_000* is comparable to the *mpp.output\_000* one since control prints are done over exactly the same area. Making a difference between these 2 files will give something which looks like this :

```

L O D Y C - I P S L
O P A model
Ocean General Circulation Model
version OPA 9.0 (2005)

PROC number: 1

MONO processor run

prt_ctl : Sum control indices
~~~~~
nlej = 0075
njctle = 0075
nictls = 0001
nldi = 0001
nicle = 09
nlei = 09
njmpp = 0001
nimpp = 0001
njctls = 0001
nldj = 0001

kt = 1 d/m/y = 1 1 1
w**2 - : 0,000000000000000D+00
eos-p: 0,1422026115876752D+04 pot : 0,1526138687
bn2 : 0,6756023350159674D+01
rrau : 0,2382205841097513D+19
eos2d: 0,1166993074037692D+03
eos2d: 0,1176157679160260D+03
kt = 1 d/m/y = 1 1 1
declin : iday 1 nfbulk= 5
Ice Forcings
qsr_oce : 0,1113357076005600D+07 qsr_ice : 0,108065
qnsr_oce : -0,592738556997037D+06 qnsr_ice : 0,247157
evap : 0,1351226604525612D+00
precip : 0,1890809328960061D+00 Snow : 0,117559
u-stress : 0,9249561064007982D+02 v-stress : -0,681945
sst : 0,1525696896992797D+07 sss : 0,183479
u_io : 0,000000000000000D+00 v_io : 0,000000
hsnif 1 : 0,961999999999881D+02 hicif : 0,962000
frld 1 : 0,579419999999982D+04 sist : 0,180675
lim_dyn : i_j1 = 1 i_j_pj = 149
lim_rhg : res = 0,1786155962060082D-03 iter = 101
lim_rhg : u_ice : 0,6782111034099104D+01 v_ice : 0
lim_dyn : u_oce : 0,000000000000000D+00 v_oce : 0
lim_dyn : u_ice : 0,6782111034099104D+01 v_ice : 0
lim_dyn : tio_u : -0,9313815445839674D-01 tio_v : 0
lim_dyn : ust2s : 0,3256460470857356D+00
hsnif 2 : 0,961999999999881D+02 hicif : 0,962000
frld 2 : 0,579419999999982D+04 sist : 0,180675
stdin

L O D Y C - I P S L
O P A model
Ocean General Circulation Model
version OPA 9.0 (2005)

PROC number: 1

MULTI processor run

prt_ctl : Sum control indices
~~~~~
nlej = 0075
njctle = 0075
nictls = 0001
nldi = 0001
nicle = 09
nlei = 09
njmpp = 0001
nimpp = 0001
njctls = 0001
nldj = 0001

kt = 1 d/m/y = 1 1 1
w**2 - : 0,000000000000000D+00
eos-p: 0,1422026115876752D+04 pot : 0,1526138687
bn2 : 0,6756023350159674D+01
rrau : 0,2382205841097513D+19
eos2d: 0,1166993074037692D+03
eos2d: 0,1176157679160260D+03
kt = 1 d/m/y = 1 1 1
declin : iday 1 nfbulk= 5
Ice Forcings
qsr_oce : 0,1113357076005600D+07 qsr_ice : 0,108065
qnsr_oce : -0,592738556997037D+06 qnsr_ice : 0,247157
evap : 0,1351226604525612D+00
precip : 0,1890809328960061D+00 Snow : 0,117559
u-stress : 0,9249561064007982D+02 v-stress : -0,681945
sst : 0,1525696896992797D+07 sss : 0,183479
u_io : 0,000000000000000D+00 v_io : 0,000000
hsnif 1 : 0,961999999999881D+02 hicif : 0,962000
frld 1 : 0,579419999999982D+04 sist : 0,180675
lim_dyn : i_j1 = 1 i_j_pj = 76
lim_rhg : res = 0,1786155962060082D-03 iter = 101
lim_rhg : u_ice : 0,6782111034099104D+01 v_ice : 0
lim_dyn : u_oce : 0,000000000000000D+00 v_oce : 0
lim_dyn : u_ice : 0,6782111034099104D+01 v_ice : 0
lim_dyn : tio_u : -0,9313815445839674D-01 tio_v : 0
lim_dyn : ust2s : 0,3256460470857356D+00
hsnif 2 : 0,961999999999881D+02 hicif : 0,962000
frld 2 : 0,579419999999982D+04 sist : 0,180675

```

On the left (right) side the *mono.output\_000* (*mpp.output\_000*) file sum control print; in the head, values of different parameters, further the control sums of all arrays with their associated names.

In making the difference for each processor, the user will be able to find the time step and the subroutine which is (normaly) implied in the non reproductibility problem. The first difference will not appear at the same place and time step for all processors area, this is why it is recommended to do the difference for each processor area.

### III. The control sum organization

The *prtctl.f90* module has been created with 3 public and 1 private subroutines :

- *prt\_ctl\_init* : this one is called inside the *opa.F90* module to read the *namrun* namelist and to compute control sum indices (They are computed in the *mppini.F90* module by the subroutine *mpp\_pdbo*),
- *prt\_ctl\_info* : to print only characters
- *prt\_ctl* : to print the control sum of 1 (or 2) 2D (or 3D) arrays with associated information and mask.
- *mpp\_pdbo*: contains the part which compute all necessary indices associated to each MPP sub-domain.

We describe succinctly arguments of the subroutines *prt\_ctl* and *prt\_ctl\_info* :

**SUBROUTINE *prt\_ctl\_info*** (*clinfo1, ivar1, clinfo2, ivar2*) :

- All arguments are optional,
- *clinfo*[12] are characters
- *ivar*[12] are integers only

**SUBROUTINE *prt\_ctl*** (*tab2d\_1, tab3d\_1, mask1, clinfo1, tab2d\_2, tab3d\_2, mask2, & clinfo2, overlap, kdim, clinfo3*) :

- All arguments are optional,
- *tab2d*\_[12] are the first and second 2D arrays,
- *tab3d*\_[12] are the first and second 3D arrays,
- *mask*[12] are the mask associated to the first and second 2D or 3D arrays,
- *clinfo*[12] are the information associated to each 2D or 3D arrays,
- *overlap* is to take into account the overlap band in the control sum (zero is the default value),
- *kdim* is the level until which is done the control sum for 3D arrays (*jpkm1* is the default value)

To use one of it in any other module, the user has just to add in the head of the target module " USE prtctl " and then add as many CALLs he wants for any arrays.

Notice : Since all arguments are optional, the generic name should be used in the calling sequence in any order.

### IV. Necessary condition for the reproductibility

Before launching the mono- and multi- processors runs using the control sum option describes above, some specific actions in the namelist, in the source code and at the compilation step are required to ensure the reproductibility behavior.

In the ocean nameilst :

- the Red/Black **S.O.R** (Successive Over Relaxation) elliptic solver must be used for all runs. With that solver, the elliptic equations resolution is done identically in mono- and multi- processors run. In the namelist the **nsolv** parameter must be set to 2.

- the minimum number of iteration (**nitend** parameter) for test must be at least 200 ; when the reproducibility error is founded, it should be tested over 400 iterations to be sure.

In the source code :

# All 'CALL mpp\_sum()' statements in the code do not allow the reproducibility since sums are not performed in the same order in multi- and mono- processors runs. To skip that, scalars used in the mpp\_sum() CALLs must be fixed to a constant in both cases. One usual module concerned by this point is ocesbc.F90 where for instance, **aplus** and **aminus** scalars could be fixed to 1.e0 just after their mpp\_sum() calling sequence. We give hereafter a non exhaustive list of modules using mpp\_sum() function : obcvol.F90, obcini.F90 when using open boundaries conditions, flxfwb.F90 for the freshwater budget ....

# If using the sea-ice model, the computation area of the rheology must be the same in mono- and multi- processors case. For that, the CALL lim\_rhg ( i\_j1, i\_jpj ) statement in limdyn.F90 module must be done with the same indices i\_j1 = 1 and i\_jpj = jpj in both mono- and multi- processors runs. *Doing that will not avoid differences on the correponding control print since jpj will not have the same value in mono- and multi- processor runs ; this has no consequences for the reproducibility.* Current control print differences between a mono- and a multi- control print appear for *nbpb*, *nbpac* scalars if using the sea-ice component.

At the compilation process :

The user should use the lower compiler optimization level to avoid different order of computation due to the re-organization of some loops by the compiler. For instance on the IBM SP4 (IDRIS) the lower optimization level is **-O2** ; on the NEC-SX5 (IDRIS) it is recommended to use the **-C vsafe** optimization level (which corresponds in fact to no optimization).