Experiments on the MET OFFICE method of river runoffs implementation in NEMO

Clément Vic Anne-Marie Tréguier Henrick Berger

Laboratoire de Physique des Océans

August 17, 2012

Contents

This report is about the methods for prescribing river runoff in the Nucleus for European Modelling of the Ocean[1] (NEMO, Madec and the NEMO team, 2012) model.

In the NEMO ocean model as in most global circulation models, river runoffs used to be inserted on the top grid cells for precipitation. To better represent the reality of rivers where the fresh water flux has a nonzero depth, the vertical mixing is usually increased at the river mouth vicinity. From version 3.3 onwards it is possible to add river runoffs through many ocean levels. This new method has been coded and tested by R. Furner (U.K. Met Office). In this report, we will present how this new method is implemented and how to use it, then we will compare it with the classical top grid cell implementation method. The configuration on which is based the comparison has been developed by Vic et al. (2012) in the framework for studying the dynamics of the Congo river plume in the Gulf of Guinea. We do not deliberately detail the parameters of this configuration nore the dynamics of the plume to concentrate on the numerical aspects. Please refer to Vic et al. (2012) and to the namelist of the configuration in appendix A to get more details on the configuration and the dynamics.

1 Brief description of the method

1.1 How it works ?

The Met Office method of runoff implementation differs from the classical surface-prescribed runoff method by the fact that freshwater is prescribed on several layers on the vertical. This results in adding a mass of freshwater (zero-salinity by default but it can be parameterized) in each grid cell concerned by the runoff prescription. According to routine `sbc_rnf_div`, it enters the mass conservation equation this way :

$$\frac{\partial}{\partial t}\vec{\nabla}\cdot\vec{V} = -\frac{1}{\rho_{water} h_{\mathrm{RNF}}}\frac{\mathrm{RNF}+\mathrm{RNF_b}}{2}$$

where $\vec{V}$ is the velocity, ρ_{water} the reference volumic mass of freshwater, h_{RNF} the depth on which runoffs are prescribed and RNF (and RNF_b, b for *before*) is the mass of freshwater entering the grid cell section every second :

$$\mathrm{RNF} = \frac{\rho_{water} Q}{\sum_{i=1}^{n} S_{grid\ cell\ i}} \quad (\mathrm{kg/m^2/s})$$

where n is the number of grid cells where the runoffs are prescribed and $S_{grid\ cell\ i}$ is the surface of the i^{th} grid cell.

According to the first equation, the same quantity of freshwater enters each grid cell along the vertical, generating a pressure gradient and horizontal velocities at every level as illustrated in figure 1 (this certainly consists in the major difference with the surface prescription where

[1] http://www.nemo-ocean.eu

pressure gradient is only induced in the surface layer...We will confirm this in carrying out experiments later). Because of the varying size of the grid cells along the vertical, the pressure gradient is most important in the surface layers and velocities generated too.

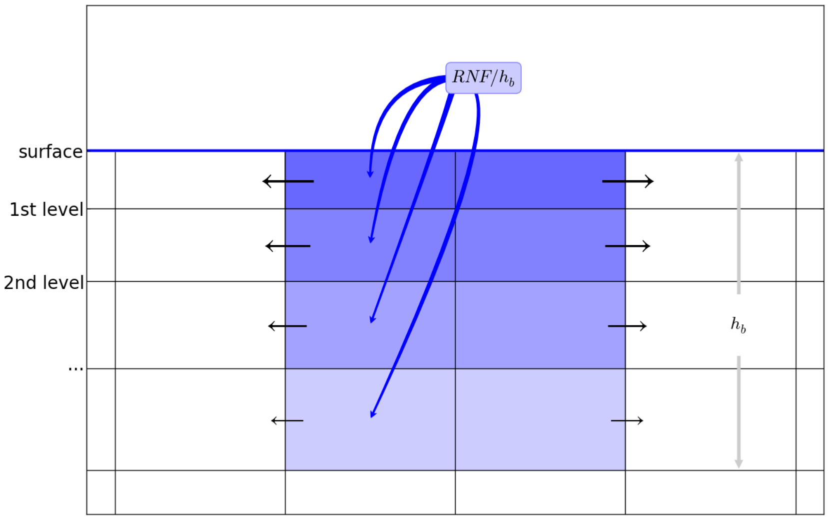

Figure 1: Schematic of the MET OFFICE method of multi-layer runoffs prescription. As suggested in the varying length of the arrows and in the gradient blue color of the cells, pressure induced by the freshwater flux and resulting velocities are more important at the surface.

1.2 How to use it ?

1.2.1 Namelist parameters

To prescribe the runoffs on multiple layers, one should set the namelist parameters this way (important parameters in red) :

```
!-----------------------------------------------------------------------
&namsbc_rnf    !   runoffs namelist surface boundary condition
!-----------------------------------------------------------------------
!              !  file name  ! frequency (hours) ! variable  ! time interp. !  clim   ! 'yearly'/ ! weights  ! rotation !
!              !             !  (if <0  months)  !   name    !   (logical)  !  (T/F)  ! 'monthly' ! filename ! pairing  !
   sn_rnf      = 'runoff',     -1     , 'sorunoff',    .true.      , .true. , 'yearly'  , ''        , ''
   sn_cnf      = 'runoff',      0     , 'socoefr',    .false.     , .true. , 'yearly'  , ''        , ''
   sn_s_rnf    = 'runoffs'            ,    -1     , 'rosaline',   .true.      , .true. , 'yearly'  , ''        , ''
   sn_t_rnf    = 'runoffs'            ,    -1     , 'rotemper',   .true.      , .true. , 'yearly'  , ''        , ''
   sn_dep_rnf  = 'runoffs'            ,     0     , 'rodepth' ,   .false.     , .true. , 'yearly'  , ''        , ''

   cn_dir       = './'      !  root directory for the location of the runoff files
   ln_rnf_emp   = .false.   !  runoffs included into precipitation field (T) or into a file (F)
   ln_rnf_mouth = .false.   !  specific treatment at rivers mouths
   rn_hrnf      =  10.e0    !  depth over which enhanced vertical mixing is used
   rn_avt_rnf   =   1.e-4   !  value of the additional vertical mixing coef. [m2/s]
   rn_rfact     =   1.e0    !  multiplicative factor for runoff
   ln_rnf_depth = .true.    !  read in depth information for runoff
   ln_rnf_tem   = .true.    !  read in temperature information for runoff
```

```
ln_rnf_sal    = .true.    !  read in salinity information for runoff
```

The specific treatment at rivers mouths must not be activated, because it could increase the vertical mixing (vertical mixing coefficient `rn_avt_rnf` to a depth of `rn_hrnf`) on layers where the fresh water flux is already prescribed. When using simultaneaously `ln_rnf_mouth = .true.` and one of `ln_rnf_depth`/`ln_rnf_tem`/`ln_rnf_sal = .true.`, one will get a warning in the `ocean.output`.

1.2.2 Input files

The method requires 4 input variables basically divided into 2 files :

- `runoff.nc` contains the variable `sorunoff` which contains the mass of fresh water per unity of surface per second at each point of the domain grid (`sorunoff` in kg/m^2/s). It can have a time-dependancy in hours (or it can be specified monthly if the parameter is negative), specified in the namelist.
- `runoffs.nc` contains the variables :
  1. `rosaline` is the salinity of the river runoff at each grid point. It is constant along the depth but can vary horizontally. If not specified, salinity is assumed to be fresh water (0 PSU).
  2. `rotemper` is the temperature of the river runoff at each grid point. It is constant along the depth but can vary horizontally. It not specified, temperature takes the SST value at each grid point along the depth.
  3. `rodepth` is the depth on which river runoff is prescribed. For the computation, it is adjusted to the bottom of the relevant grid box.

2 Comparison with the top grid cell implementation

To investigate the differences between the two methods, we use a 1/48° resolution idealized configuration based on the Congo river parameters (Vic et al., 2012); the namelist is provided in appendix A. The domain and bathymetry are shown on figure 2. The ocean is at rest and presents a constant salinity at each point of 35.5 PSU, and a homogenenous profile of temperature representative of the Congo estuary vicinity (based on Levitus data). There is no forcing except the Congo runoff (no wind stress, no boundary flux, . . .). The rate of flow Q is taken constant of 40000 m^3/s (mean climatological value). Runoffs RNF are prescribed on the half upstream part of the waterway (cf figure 3) and its value is constant on the domain (see section 1 to calculate RNF).

We have carried out three basic experiments to compare the two methods :

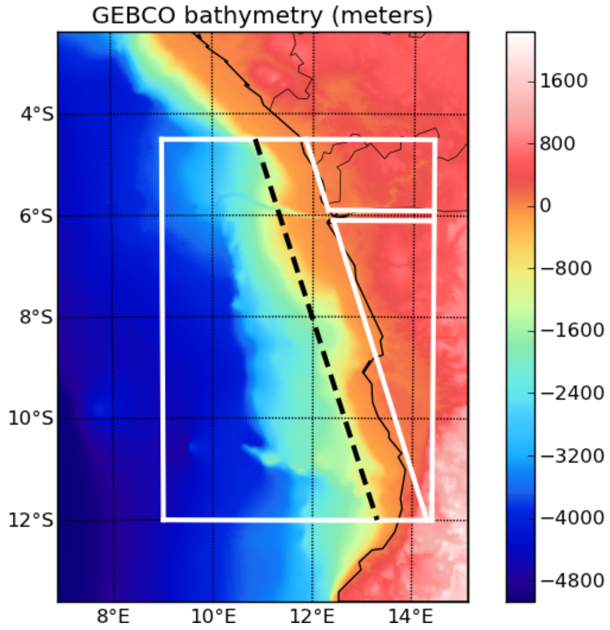

Figure 2: Domain and bathymetry of the idealized configuration. The Congo river estuary is represented as a zonal waterway at 6°S. The bathymetry is extremely simplified and has two levels : the waterway is at 30m depth and the ocean domain is at 2000m.

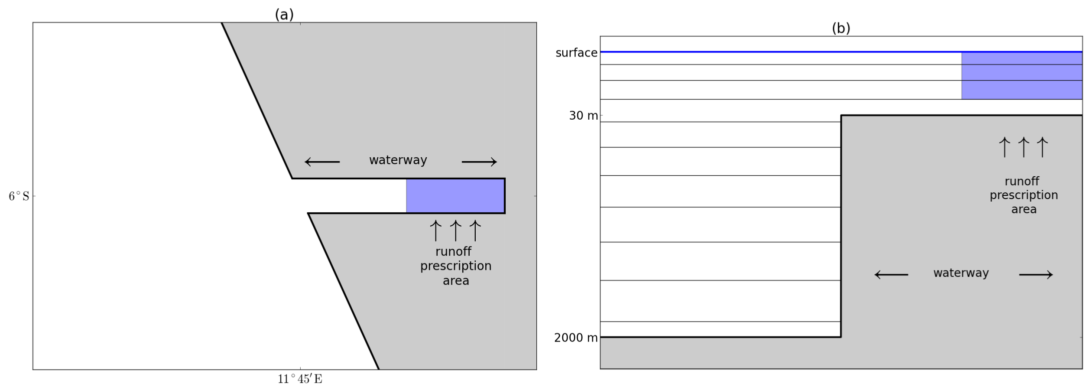

Figure 3: Schematic of the waterway and the runoff prescription area for (a) a plan view and (b) a section view. The Met Office method allows to select several vertical levels. The waterway is 3 grid points wide in the prescription area and widens to 9 grid points at the river mouth to better represent the Congo mouth, see Vic et al. (2012) for further information.

1. First experiment : runoffs prescribed on the first layer in the Met Office method must be compared with runoffs prescribed at the surface with the classical method. Both should give a thin layer of freshwater lying on the lower part of the ocean at rest.

2. Second experiment : runoffs prescribed on the first 10m depth in the Met Office method must be compared with runoffs prescribed at the surface with increased vertical mixing Kz = $1\text{m}^2/\text{s}$ (value used to simulate convection) on 10m. We expect the two methods to

give a uniform stratification on the prescribed area.

3. Third experiment : to be closest to realistic configurations where there is usually no waterway, runoffs are prescribed on a configuration with a straight coastline and no waterway, on a more or less symmetrical area around the coordinates of the river mouth.

2.1 First experiment

The first experiment confirms that the two methods are exactly the same in the case of a surface prescription for the multi-layer method. We do not show the figures because the salinity and velocity fields are namely the same.

We now expect the two methods to give different results for a deeper prescription.

2.2 Second experiment

The multi-layer prescription brings freshwater directly in the subsurface layers whereas in the surface layer prescription, increased vertical mixing coefficient tends to freshen these layers.

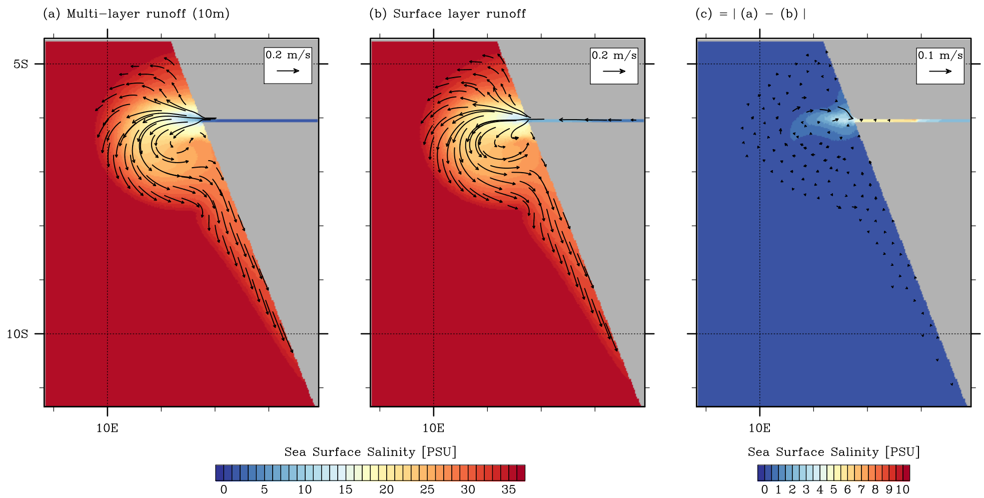

Figure 4: Sea surface salinity and surface velocity at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased dramatically to 1m^2/s in the first 10m. (c) is the difference between frame (a) and (b).

Figure 4 shows the sea surface salinity (SSS) and surface velocity for the two simulations at 30 days. Even if the SSS in the estuary reaches smaller values when runoff are prescribed on multiple layers (minimum of 1.0 PSU) than in the other case (minimum of 3.2 PSU), it does not seem to affect the velocity field in the ocean. Both methods show a classical anticyclonic circulation and a steady coastal current in the way of propagation of Kelvin waves (leaving the coast to the left because the experiment takes place in the southern hemisphere).

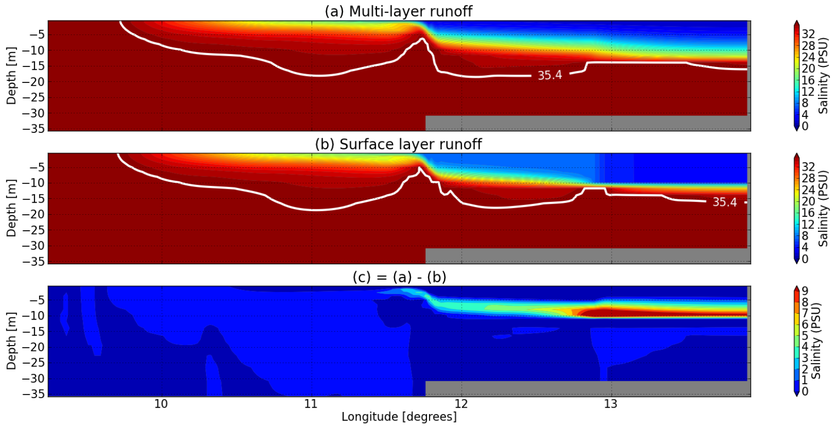

Figure 5: Salinity sections along the waterway at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased dramatically to 1m^2/s in the first 10m. The white line represents isohaline 35.4 PSU which is the limit of influence of fresh water. (c) is the difference between frame (a) and (b).

Figure 5 shows the salinity sections through the waterway at 30 days of simulation. The major difference concerns the area where the runoffs are prescribed : the increased vertical mixing (panel (b)) seems to be more «powerful» in homogenizing the water column on the first 10m than the runoffs prescription on multiple layers. The salinity increases smoothly in the multi-layer prescription, which is maybe more representative of the real stratification of estuaries. The salinity profiles in the basin do not present any stricking difference with each other. The influence of freshwater is represented by isohaline 35.4 PSU, and has the same shape in both cases. We could explain this by the fact that in the downstream part of the waterway the buoyancy adjustment is made rapidely : fresher waters reach the surface and the more common vertical diffusion ($\propto 10^{-5}$ m^2/s) tends to stabilize the flux. In the end, the difference between the two salinity fields vanishes along the waterway (panel (c)) and the salinity profiles are equivalent at the river mouth.

Doing the same experiment with a more classical vertical diffusion coefficient (10^{-4} m^2/s) at the river mouth (specific treatment) for the surface prescription gives similar results. Furthermore, it seems to be possible to get the same stratification and velocity fields in adjusting this coefficient.

2.3 Third experiment

The third experiment tests the methods in a more realistic case (without a waterway like in former academic shemes). The runoffs are prescribed on an area centered on the position of the river mouth (6°00'S ; 11°45'E), on 10m for the multi-layers method and at the surface method with increased Kz = 10^{-4} m^2/s in the other method.

2.3.1 Runoffs prescribed on a large area

Like in former experiments, in order to keep the same order of magnitude of the RNF value and to generate comparable velocities, runoffs are prescribed on 101 grid cells.

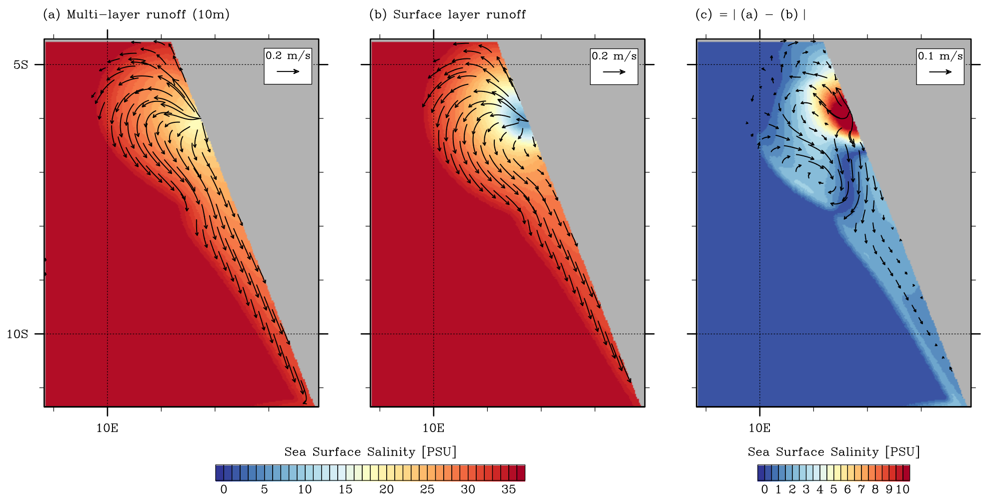

Figure 6: Sea surface salinity and surface velocity at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased to $10^{-4}\text{m}^2/\text{s}$ in the first 10m. (c) is the difference between frame (a) and (b).

The first comment that comes from the comparison of figures 4 and 6 is that the presence of the waterway does not affect dramatically the shape of the dynamics. We still have the anticyclonic circulation and the coastal current. Nonetheless, small differences concern the cross-shore extent compared with the width of the coastal current : on the one hand, the waterway controls the flux which suddenly reaches the basin at rest and preferentially goes offshore following its former direction and then is deflected to the left by the Coriolis force. On the other hand, with no waterway, the flux seems to spread more isotropically in all the directions (not preferentially to the offshore direction).

The most stricking difference is about the SSS : the two methods present a difference which is maximum at the bulge core and reaches 10 PSU (panel (c) of figure 6). According to figure 7 differences are confined to the prescription area and in the first 3 m, then they vanish with the flow. The multi-layer prescription actually better distribute the amount of freshwater with depth and it results a less desalinated water at the surface.

The other major difference concerns the variation of velocity (and salinity) with the depth. It derives from the fact that the multi-layer prescription brings a mass of water in every layer of the column until 10m, so that a pressure gradient is created in each layer. As a consequence, velocities are properly induced in each layer and the vertical shear is less important than with the surface layer prescription : we can see it in figure 8 where the velocity field at 5m depth is less attenuated in the multi-layer prescription (panel (a)) than in the surface layer prescription

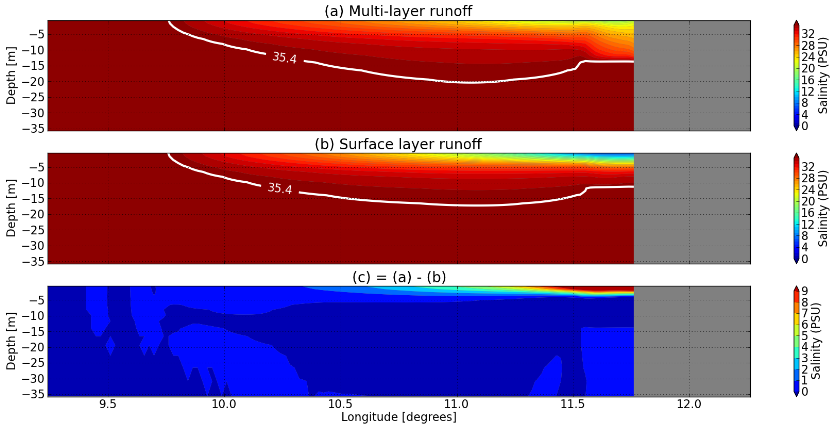

Figure 7: Salinity section at 6°S (runoffs are prescribed symmetrically along this axis) at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased to 10^{-4} m^2/s in the first 10m. The white line represents isohaline 35.4 PSU which is the limit of influence of fresh water. (c) is the difference between frame (a) and (b).

(panel (b)), compared to the surface velocity field. The difference of velocity between the two methods even reaches 10 cm/s in the bulge circulation (panel (c)).

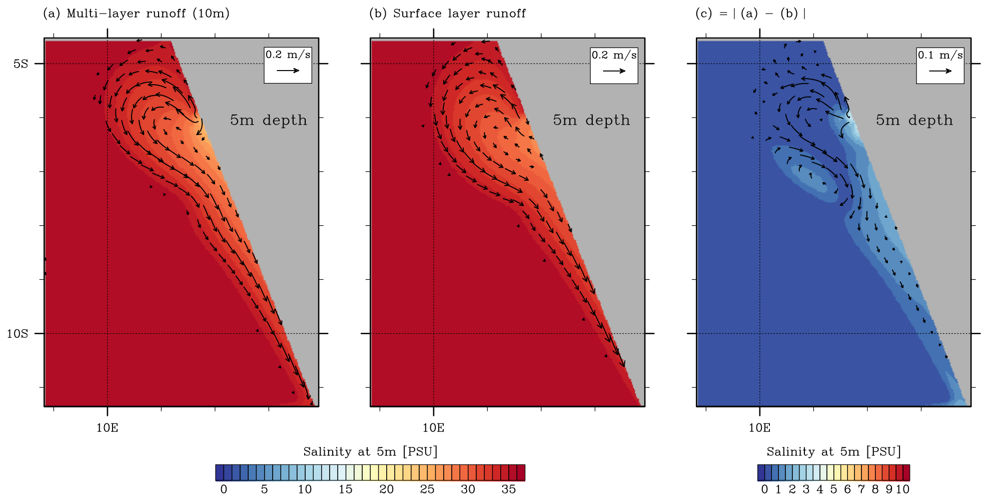

Figure 8: Salinity and velocity at 30 days at 5m depth for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased to $10^{-4} m^2/s$ in the first 10m. (c) is the difference between frame (a) and (b).

2.3.2 Runoffs prescribed on a few grid cells

In order to avoid prescription on ocean grid cells where there is no physical reason to bring freshwater (as we did in the former case with 101 grid cells affected by the prescription[2]),

[2] which represent a surface of ≈ 500 km^2

we confine the prescription on a small patch around the river mouth (here we take 6 points). Therefore, the value of RNF is increasing by two orders of magnitude (≈ 1 kg/m^2/s). Simulation with surface prescription aborted quickly as salinity could not be evacuated in the deeper layers and reached negative values.

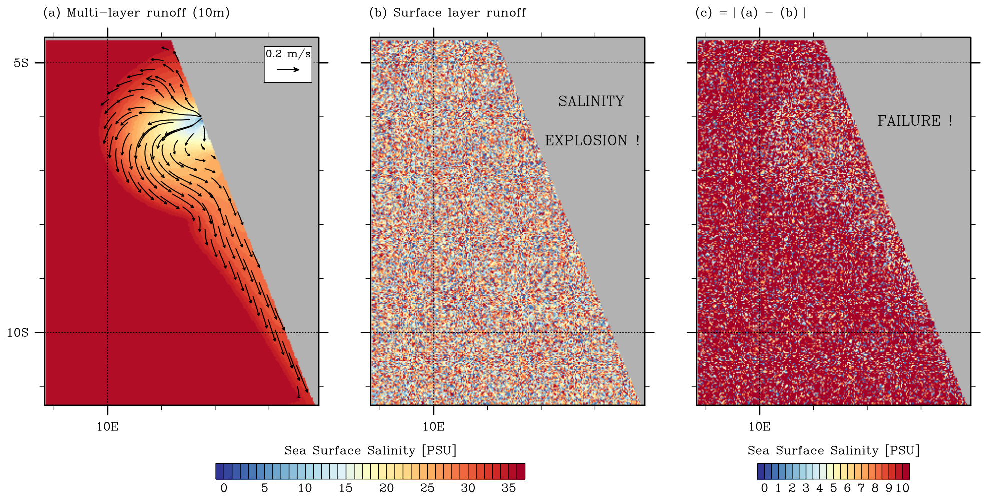

Figure 9: Sea surface salinity and surface velocity at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased to 10^{-4}m^2/s in the first 10m. (c) is the difference between frame (a) and (b). In case (b), the simulation aborts quickly as salinity becomes negative.

Figure 9 must be compared with figures 6 and 4. The dynamics of the surface layer seems to be closer to the one generated with the waterway than to the one without waterway with prescription on a large area. In fact, the anticyclonic bulge is less distorted to the coast and velocities are more comparable with a maximum value of 1.4 m/s at the river mouth and velocities in the bulge of 25 cm/s and of 20 cm/s in the coastal current. Moreover, the minimum of salinity reached at the river mouth is closer (≈ 9 PSU).

Salinity section in this case (Figure 10) differs from the usual ones (*e.g.* figure 7) by the depth reached by fresh water. The depth of isohaline 35.4 PSU only reaches 15 m whereas it goes deeper than 20 m in the waterwayless reference case.

2.3.3 Comparison with the waterway prescription

The two experiments carried out in the waterwayless cases must be compared with the waterway case in order to observe which type of patch ($\approx$ 100 points-prescription one or $\approx$ 1 point-prescription one) best fits with the reality. As we described sooner, the shape of the plume is more realistic in the 6-point-prescription experiment than in the large patch prescription one as illustrated in figure 11.

In terms of dynamics and salinity repartition, differences are minimum with the small patch prescription (panels (c) and (f)). The major differences concern the recirculation formed just

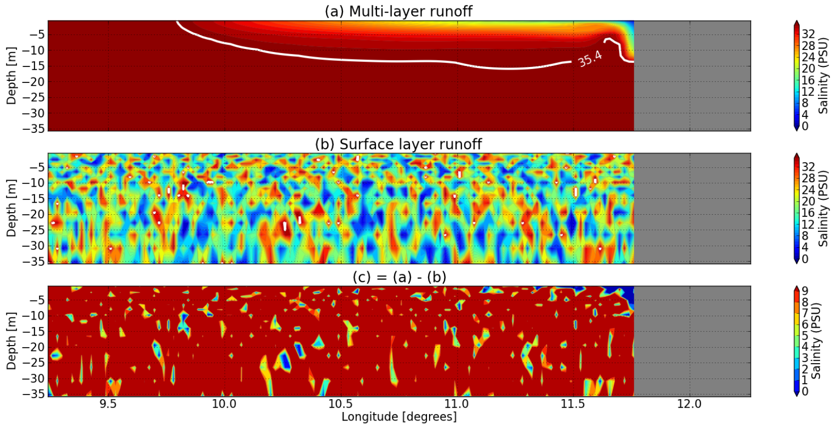

Figure 10: Salinity section at 6°S at 30 days for (a) runoffs prescribed on the first 10m (multiple ocean layers) and (b) runoffs prescribed at the surface and vertical mixing Kz is increased to 10^{-4} m^2/s in the first 10m. The white line represents isohaline 35.4 PSU which is the limit of influence of fresh water. (c) is the difference between frame (a) and (b). In case (b), the simulation aborts quickly as salinity becomes negative.

below the river mouth : it is more pronounced in the waterway prescription than in the patches ones. Nevertheless, it affects only the vicinity of the river mouth and differences of velocity vanish as we reach the frontier of the bulge.

Considering the stratification, it seems to be qualitatively the same, and it is difficult to say which one is better because it certainly depends on the river represented, and a comparison should be done with the observations to determine which profile of salinity best fits.

2.4 Remark on the temperature

We also did experiments on temperature but without precise sensibility tests. The effect observed is that decreasing the freshwater temperature (as it happens in reality where freshwater is often colder than the ocean) increases its density and tends to reduce the density gradient and the velocities induced.

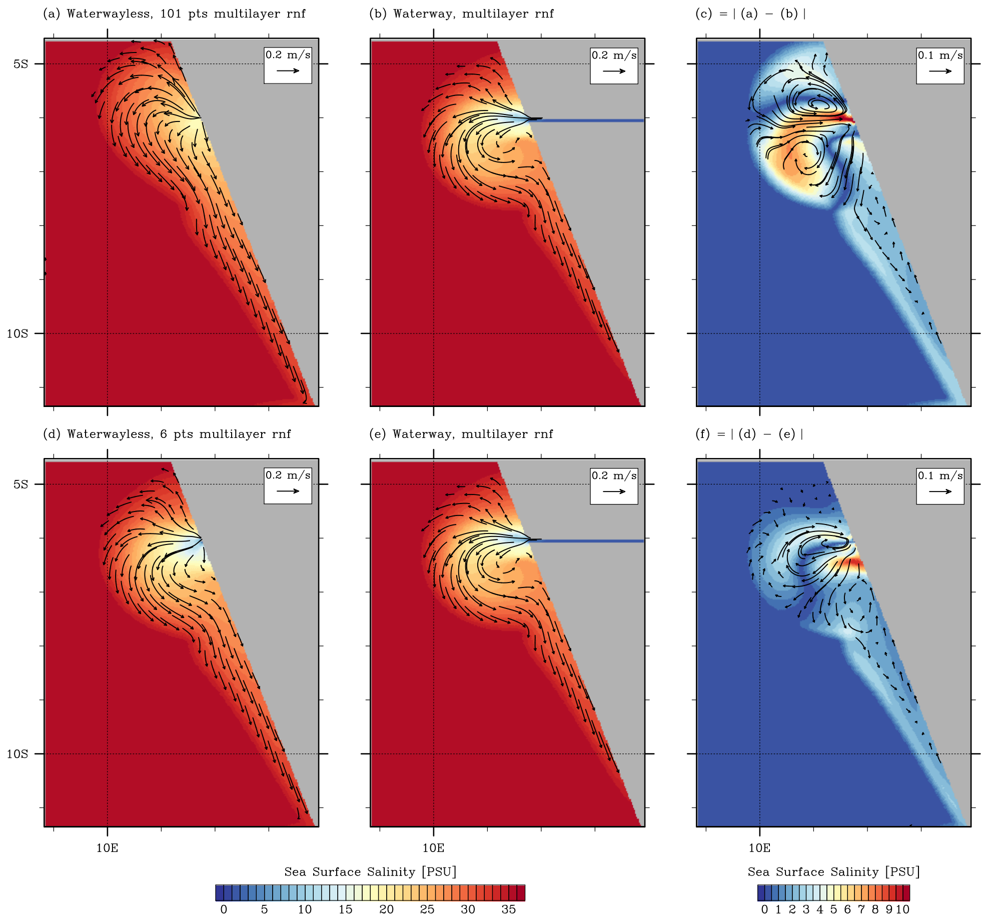

Figure 11: Salinity and velocity at 30 days at 5m depth for runoffs prescribed on multilayer (10 m) : without waterway on 101 grid cells (a) and 6 grid cells (d), with waterway (b)≡(e). (c) is the difference between frames (a) and (b). (f) is the difference between frames (d) and (e).

3 What should we remember about this new method ?

The Met Office method aims to prescribe runoffs on several vertical levels to be more realistic concerning rivers outflows as they usually enter ocean with a non-zero depth.

- The main difference with the surface prescription is that a pressure gradient is involved in each layer so that velocities are generated at each level.
- Concerning the stratification, one can obtain the same values of salinity along with the depth in the surface prescription method by tuning vertical mixing coefficient (from 10^{-4} m^2/s as with the original specific treatment at river mouth, to 1 m^2/s to get a constant salinity along with the depth). Nonetheless, even if the result is the same for the salinity stratification, the fact that we do not need to increase dramatically Kz is a gain in terms of representation of a more realistic physics.
- The most important breakthrough is that runoffs can be prescribed on a minimum number of grid cells, as freshwater is evacuated easier. As a result, one can divide by two orders of magnitude the number of grid cells in comparison with the surface prescription. It enables not to enhance vertical mixing as it was the case but for no other physical reason than homogenizing freshwater flux along with the depth, so that we can say that the new method is more physical. Furthermore, we obtain better results in terms of dynamics with smaller patches.

References

Madec, G. and the NEMO team (2012). *NEMO ocean engine*, volume 3.4. Pôle de modélisation de l'Institut Pierre-Simon Laplace.

Vic, C., Treguier, A.-M., and Berger, H. (2012). Dynamique d'un panache fluvial en zone équatoriale, modélisation numérique et application au cas du Congo. Master's thesis, ENSTA ParisTech and Paris VI UPMC.

A Namelist

```
!.!>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
!! NEMO/OPA  :  1 - run manager      (namrun)
!! namelists    2 - Domain           (namzgr, namzgr_sco, namdom, namtsd)
!!              3 - Surface boundary (namsbc, namsbc_ana, namsbc_flx, namsbc_clio, namsbc_core
!!                                    namsbc_cpl, namtra_qsr, namsbc_rnf,
!!                                    namsbc_apr, namsbc_ssr, namsbc_alb)
!!              4 - lateral boundary (namlbc, namcla, namobc, namagrif, nambdy, nambdy_tide)
!!              5 - bottom  boundary (nambfr, nambbc, nambbl)
!!              6 - Tracer           (nameos, namtra_adv, namtra_ldf, namtra_dmp)
!!              7 - dynamics         (namdyn_adv, namdyn_vor, namdyn_hpg, namdyn_spg, namdyn_ldf)
!!              8 - Verical physics  (namzdf, namzdf_ric, namzdf_tke, namzdf_kpp, namzdf_ddm, namzdf_tmx)
!!              9 - diagnostics      (namnc4, namtrd, namspr, namflo, namptr, namhsb)
!!             10 - miscellaneous    (namsol, nammpp, nammpp_dyndist, namctl)
!!             11 - Obs & Assim      (namobs, nam_asminc)
!!>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

!!======================================================================
!!                   ***  Run management namelists  ***
!!======================================================================
!!   namrun       parameters of the run
!!======================================================================
!
!-----------------------------------------------------------------------
&namrun        !   parameters of the run
!-----------------------------------------------------------------------
   nn_no       =       0   !  job number (no more used...)
   cn_exp      =  "3CONGO-BCV02"   !  experience name
   nn_it000    =       0   !  first time step
   nn_itend    =     25920 !  last  time step
   nn_date0    =  010101   !  date at nit_0000 (format yyyymmdd) used if ln_rstart=F or (ln_rstart=T and nn_rstctl=0 or 1)
   nn_leapy    =      30   !  Leap year calendar (1) or not (0)
   ln_rstart   = .false.   !  start from rest (F) or from a restart file (T)
   nn_rstctl   =       0   !  restart control => activated only if ln_rstart = T
                           !    = 0 nn_date0 read in namelist ; nn_it000 : read in namelist
                           !    = 1 nn_date0 read in namelist ; nn_it000 : check consistancy between namelist and restart
                           !    = 2 nn_date0 read in restart  ; nn_it000 : check consistancy between namelist and restart
   cn_ocerst_in  = "restart"   !  suffix of ocean restart name (input)
   cn_ocerst_out = "restart"   !  suffix of ocean restart name (output)
   nn_istate   =       1   !  output the initial state (1) or not (0)
   nn_stock    =    35000  !  frequency of creation of a restart file (modulo referenced to 1)
   nn_write    =      4320 !  frequency of write in the output file   (modulo referenced to nn_it000)
   ln_dimgnnn  = .false.   !  DIMG file format: 1 file for all processors (F) or by processor (T)
   ln_mskland  = .false.   !  mask land points in NetCDF outputs (costly: + ~15%)
   ln_clobber  = .false.   !  clobber (overwrite) an existing file
   nn_chunksz  =       0   !  chunksize (bytes) for NetCDF file (works only with iom_nf90 routines)
/

!!======================================================================
!!                      ***  Domain namelists  ***
!!======================================================================
!!   namzgr       vertical coordinate
```

```
!!   namzgr_sco   s-coordinate or hybrid z-s-coordinate
!!   namdom       space and time domain (bathymetry, mesh, timestep)
!!   namtsd       data: temperature & salinity
!!======================================================================
!
!-----------------------------------------------------------------------
&namzgr        !   vertical coordinate
!-----------------------------------------------------------------------
   ln_zco      = .false.   !  z-coordinate - full    steps   (T/F)      ("key_zco" may also be defined)
   ln_zps      = .true.    !  z-coordinate - partial steps   (T/F)
   ln_sco      = .false.   !  s- or hybrid z-s-coordinate    (T/F)
/
!-----------------------------------------------------------------------
&namzgr_sco    !   s-coordinate or hybrid z-s-coordinate
!-----------------------------------------------------------------------
   rn_sbot_min =  300.     !  minimum depth of s-bottom surface (>0) (m)
   rn_sbot_max = 5250.     !  maximum depth of s-bottom surface (= ocean depth) (>0) (m)
   rn_theta    =    6.0    !  surface control parameter (0<=rn_theta<=20)
   rn_thetb    =    0.75   !  bottom control parameter  (0<=rn_thetb<= 1)
   rn_rmax     =    0.15   !  maximum cut-off r-value allowed (0<rn_max<1)
   ln_s_sigma  = .false.   !  hybrid s-sigma coordinates
   rn_bb       =    0.8    !  stretching with s-sigma
   rn_hc       =  150.0    !  critical depth with s-sigma
/
!-----------------------------------------------------------------------
&namdom        !   space and time domain (bathymetry, mesh, timestep)
!-----------------------------------------------------------------------
   nn_bathy    =    1      !  compute (=0) or read (=1) the bathymetry file
   nn_closea   =    0      !  remove (=0) or keep (=1) closed seas and lakes (ORCA)
   nn_msh      =    1      !  create (=1) a mesh file or not (=0)
   rn_hmin     =   -3.     !  min depth of the ocean (>0) or min number of ocean level (<0)
   rn_e3zps_min=   25.     !  partial step thickness is set larger than the minimum of
   rn_e3zps_rat=    0.2    !  rn_e3zps_min and rn_e3zps_rat*e3t, with 0<rn_e3zps_rat<1
                           !
   rn_rdt      =  100.     !  time step for the dynamics (and tracer if nn_acc=0)
   nn_baro     =   60      !  number of barotropic time step           ("key_dynspg_ts")
   rn_atfp     =    0.1    !  asselin time filter parameter
   nn_acc      =    0      !  acceleration of convergence : =1      used, rdt < rdttra(k)
                                 !                           =0, not used, rdt = rdttra
   rn_rdtmin   = 7200.           !  minimum time step on tracers (used if nn_acc=1)
   rn_rdtmax   = 7200.           !  maximum time step on tracers (used if nn_acc=1)
   rn_rdth     =  800.           !  depth variation of tracer time step (used if nn_acc=1)
/
!-----------------------------------------------------------------------
&namtsd    !   data : Temperature  & Salinity
!-----------------------------------------------------------------------
!          ! file name ! frequency (hours)    ! variable ! time interp. ! clim  !'yearly' or ! weights  ! rotation !
!          !           !  (if <0  months)     !   name   !  (logical)   ! (T/F) ! 'monthly'  ! filename ! pairing  !
   sn_tem_ini = 'data_1m_potential_temperature_nomask', -1,'votemper',  .true.  , .true., 'yearly'   , ' '      , ' '
   sn_sal_ini = 'data_1m_salinity_nomask'             , -1,'vosaline',  .true.  , .true., 'yearly'   , ''       , ' '
   !
   cn_dir        = './'      !  root directory for the location of the runoff files
```

```
   ln_tsd_init   = .false.   !  Initialisation of ocean T & S with T &S input data (T) or not (F)
   ln_tsd_tradmp = .false.   !  damping of ocean T & S toward T &S input data (T) or not (F)
/
!!======================================================================
!!           ***  Surface Boundary Condition namelists  ***
!!======================================================================
!!   namsbc          surface boundary condition
!!   namsbc_ana      analytical         formulation
!!   namsbc_flx      flux               formulation
!!   namsbc_clio     CLIO bulk formulae formulation
!!   namsbc_core     CORE bulk formulae formulation
!!   namsbc_mfs      MFS  bulk formulae formulation
!!   namsbc_cpl      CouPLed            formulation                     ("key_coupled")
!!   namtra_qsr      penetrative solar radiation
!!   namsbc_rnf      river runoffs
!!   namsbc_apr      Atmospheric Pressure
!!   namsbc_ssr      sea surface restoring term (for T and/or S)
!!   namsbc_alb      albedo parameters
!!======================================================================
!
!-----------------------------------------------------------------------
&namsbc        !   Surface Boundary Condition (surface module)
!-----------------------------------------------------------------------
   nn_fsbc     = 1         !  frequency of surface boundary condition computation
                           !     (also = the frequency of sea-ice model call)
   ln_ana      = .true.    !  analytical formulation                    (T => fill namsbc_ana )
   ln_flx      = .false.   !  flux formulation                          (T => fill namsbc_flx )
   ln_blk_clio = .false.   !  CLIO bulk formulation                     (T => fill namsbc_clio)
   ln_blk_core = .false.   !  CORE bulk formulation                     (T => fill namsbc_core)
   ln_blk_mfs  = .false.   !  MFS bulk formulation                      (T => fill namsbc_mfs )
   ln_cpl      = .false.   !  Coupled formulation                       (T => fill namsbc_cpl )
   ln_apr_dyn  = .false.   !  Patm gradient added in ocean & ice Eqs.   (T => fill namsbc_apr )
   nn_ice      = 0         !  =0 no ice boundary condition   ,
                           !  =1 use observed ice-cover      ,
                           !  =2 ice-model used                         ("key_lim3" or "key_lim2)
   ln_dm2dc    = .false.   !  daily mean to diurnal cycle on short wave
   ln_rnf      = .true.    !  runoffs                                   (T => fill namsbc_rnf)
   ln_ssr      = .false.   !  Sea Surface Restoring on T and/or S       (T => fill namsbc_ssr)
   nn_fwb      = 0         !  FreshWater Budget: =0 unchecked
                           !     =1 global mean of e-p-r set to zero at each time step
                           !     =2 annual global mean of e-p-r set to zero
                           !     =3 global emp set to zero and spread out over erp area
   ln_cdgw = .false.       !  Neutral drag coefficient read from wave model (T => fill namsbc_wave)
/
!-----------------------------------------------------------------------
&namsbc_ana    !   analytical surface boundary condition
!-----------------------------------------------------------------------
   nn_tau000   =   0       !  gently increase the stress over the first ntau_rst time-steps
   rn_utau0    =   0.      !  uniform value for the i-stress
   rn_vtau0    =   0.      !  uniform value for the j-stress
   rn_qns0     =   0.      !  uniform value for the total heat flux
   rn_qsr0     =   0.      !  uniform value for the solar radiation
   rn_emp0     =   0.      !  uniform value for the freswater budget (E-P)
/
!-----------------------------------------------------------------------
&namsbc_flx    !   surface boundary condition : flux formulation
!-----------------------------------------------------------------------
!              !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!              !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_utau     = 'utau'      ,        24         , 'utau'    , .false.      , .false., 'yearly'  , ''       , ''
   sn_vtau     = 'vtau'      ,        24         , 'vtau'    , .false.      , .false., 'yearly'  , ''       , ''
   sn_qtot     = 'qtot'      ,        24         , 'qtot'    , .false.      , .false., 'yearly'  , ''       , ''
   sn_qsr      = 'qsr'       ,        24         , 'qsr'     , .false.      , .false., 'yearly'  , ''       , ''
   sn_emp      = 'emp'       ,        24         , 'emp'     , .false.      , .false., 'yearly'  , ''       , ''

   cn_dir      = './'      !  root directory for the location of the flux files
/
!-----------------------------------------------------------------------
&namsbc_clio   !   namsbc_clio  CLIO bulk formulae
!-----------------------------------------------------------------------
!              !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!              !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_utau     = 'taux_1m'   ,       -1          , 'sozotaux',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_vtau     = 'tauy_1m'   ,       -1          , 'sometauy',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_wndm     = 'flx'       ,       -1          , 'socliowi',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_tair     = 'flx'       ,       -1          , 'socliot2',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_humi     = 'flx'       ,       -1          , 'sociohu',   .true.      , .true. , 'yearly'  , ''       , ''
   sn_ccov     = 'flx'       ,       -1          , 'sociocl',   .false.     , .true. , 'yearly'  , ''       , ''
   sn_prec     = 'flx'       ,       -1          , 'sociopl',   .false.     , .true. , 'yearly'  , ''       , ''

   cn_dir      = './'      !  root directory for the location of the bulk files are
/
!-----------------------------------------------------------------------
&namsbc_core   !   namsbc_core  CORE bulk formulae
!-----------------------------------------------------------------------
!              !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!              !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_wndi = 'u_10.15JUNE2009_orca2'      ,  6  , 'U_10_MOD',   .false.    , .true. , 'yearly'  , ''       , 'Uwnd'
   sn_wndj = 'v_10.15JUNE2009_orca2'      ,  6  , 'V_10_MOD',   .false.    , .true. , 'yearly'  , ''       , 'Vwnd'
   sn_qsr  = 'ncar_rad.15JUNE2009_orca2'   , 24  , 'SWDN_MOD',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_qlw  = 'ncar_rad.15JUNE2009_orca2'   , 24  , 'LWDN_MOD',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_tair = 't_10.15JUNE2009_orca2'      ,  6  , 'T_10_MOD',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_humi = 'q_10.15JUNE2009_orca2'      ,  6  , 'Q_10_MOD',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_prec = 'ncar_precip.15JUNE2009_orca2', -1  , 'PRC_MOD1',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_snow = 'ncar_precip.15JUNE2009_orca2', -1  , 'SNOW'    ,   .false.    , .true. , 'yearly'  , ''       , ''
   sn_tdif = 'taudif_core'                , 24  , 'taudif'  ,   .false.    , .true. , 'yearly'  , ''       , ''

   cn_dir      = './'      !  root directory for the location of the bulk files
   ln_2m       = .false.   !  air temperature and humidity referenced at 2m (T) instead 10m (F)
   ln_taudif   = .false.   !  HF tau contribution: use "mean of stress module - module of the mean stress" data
   rn_pfac     = 1.        !  multiplicative factor for precipitation (total & snow)
/
!-----------------------------------------------------------------------
&namsbc_mfs   !   namsbc_mfs  MFS bulk formulae
```

```
!-----------------------------------------------------------------------
!            !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!            !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_wndi     =   'ecmwf'   ,        6          , 'u10'     ,    .true.    , .false. , 'daily'  ,'bicubic.nc' , ''
   sn_wndj     =   'ecmwf'   ,        6          , 'v10'     ,    .true.    , .false. , 'daily'  ,'bicubic.nc' , ''
   sn_clc      =   'ecmwf'   ,        6          , 'clc'     ,    .true.    , .false. , 'daily'  ,'bilinear.nc', ''
   sn_msl      =   'ecmwf'   ,        6          , 'msl'     ,    .true.    , .false. , 'daily'  ,'bicubic.nc' , ''
   sn_tair     =   'ecmwf'   ,        6          , 't2'      ,    .true.    , .false. , 'daily'  ,'bicubic.nc' , ''
   sn_rhm      =   'ecmwf'   ,        6          , 'rh'      ,    .true.    , .false. , 'daily'  ,'bilinear.nc', ''
   sn_prec     =   'ecmwf'   ,        6          , 'precip'  ,    .true.    , .true.  , 'daily'  ,'bicubic.nc' , ''

   cn_dir      = './ECMWF/'      !  root directory for the location of the bulk files
/
!-----------------------------------------------------------------------
&namsbc_cpl    !   coupled ocean/atmosphere model                       ("key_coupled")
!-----------------------------------------------------------------------
!                    !     description       !  multiple  !    vector   !      vector          ! vector !
!                    !                       ! categories !  reference  !    orientation       ! grids  !
! send
sn_snd_temp   =       'weighted oce and ice' ,    'no'    ,     ''      ,         ''           ,   ''
sn_snd_alb    =       'weighted ice'         ,    'no'    ,     ''      ,         ''           ,   ''
sn_snd_thick  =       'none'                 ,    'no'   ,     ''      ,         ''           ,   ''
sn_snd_crt    =       'none'                 ,    'no'    , 'spherical' , 'eastward-northward' ,  'T'
sn_snd_co2    =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
! receive
sn_rcv_w10m   =       'none'                 ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_taumod =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_tau    =       'oce only'             ,    'no'    , 'cartesian' , 'eastward-northward',  'U,V'
sn_rcv_dqnsdt =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_qsr    =       'oce and ice'          ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_qns    =       'oce and ice'          ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_emp    =       'conservative'         ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_rnf    =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_cal    =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
sn_rcv_co2    =       'coupled'              ,    'no'    ,     ''      ,         ''           ,   ''
/
!-----------------------------------------------------------------------
&namtra_qsr    !   penetrative solar radiation
!-----------------------------------------------------------------------
!            !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!            !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_chl      ='chlorophyll',        -1         , 'CHLA'    ,   .true.     , .true. , 'yearly'  , ''       , ''

   cn_dir      = './'      !  root directory for the location of the runoff files
   ln_traqsr   = .false.    !  Light penetration (T) or not (F)
   ln_qsr_rgb  = .false.   !  RGB (Red-Green-Blue) light penetration
   ln_qsr_2bd  = .true.    !  2 bands              light penetration
   ln_qsr_bio  = .false.   !  bio-model light penetration
   nn_chldta   =      0    !  RGB : Chl data (=1) or cst value (=0)
   rn_abs      =   0.58    !  RGB & 2 bands: fraction of light (rn_si1)
   rn_si0      =   0.35    !  RGB & 2 bands: shortess depth of extinction
   rn_si1      =   23.0    !  2 bands: longest depth of extinction
/
!-----------------------------------------------------------------------
&namsbc_rnf    !   runoffs namelist surface boundary condition
!-----------------------------------------------------------------------
!            !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!            !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_rnf      = 'runoff',    -1    , 'sorunoff',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_cnf      = 'runoff',     0    , 'socoefr',   .false.    , .true. , 'yearly'  , ''       , ''
   sn_s_rnf    = 'runoffs'           ,    -1    , 'rosaline',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_t_rnf    = 'runoffs'           ,    -1    , 'rotemper',   .true.     , .true. , 'yearly'  , ''       , ''
   sn_dep_rnf  = 'runoffs'           ,     0    , 'rodepth' ,   .false.    , .true. , 'yearly'  , ''       , ''

   cn_dir       = './'      !  root directory for the location of the runoff files
   ln_rnf_emp   = .false.   !  runoffs included into precipitation field (T) or into a file (F)
   ln_rnf_mouth = .false.   !  specific treatment at rivers mouths
   rn_hrnf      =  10.e0    !  depth over which enhanced vertical mixing is used
   rn_avt_rnf   =   1.e-4   !  value of the additional vertical mixing coef. [m2/s]
   rn_rfact     =   1.e0    !  multiplicative factor for runoff
   ln_rnf_depth = .true.    !  read in depth information for runoff
   ln_rnf_tem   = .false.   !  read in temperature information for runoff
   ln_rnf_sal   = .true.    !  read in salinity information for runoff
/
!-----------------------------------------------------------------------
&namsbc_apr    !   Atmospheric pressure used as ocean forcing or in bulk
!-----------------------------------------------------------------------
!            ! file name ! frequency (hours) ! variable ! time interpol. !  clim   ! 'yearly'/ ! weights  ! rotation !
!            !           !  (if <0  months)  !   name   !    (logical)   !  (T/F)  ! 'monthly' ! filename ! pairing  !
   sn_apr      = 'patm'    ,         -1        ,'somslpre',    .true.      , .true.  , 'yearly'  ,  ''      ,  ''

   cn_dir      = './'      !  root directory for the location of the bulk files
   ln_ref_apr  = .false.   !  ref. pressure: global mean Patm (T) or a constant (F)
/
!-----------------------------------------------------------------------
&namsbc_ssr    !   surface boundary condition : sea surface restoring
!-----------------------------------------------------------------------
!            !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!            !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_sst      = 'sst_data'  ,        24         ,  'sst'    ,    .false.   , .false., 'yearly'  , ''       , ''
   sn_sss      = 'sss_data'  ,        -1         ,  'sss'    ,    .true.    , .true. , 'yearly'  , ''       , ''

   cn_dir      = './'      !  root directory for the location of the runoff files
   nn_sstr     =     0     !  add a retroaction term in the surface heat       flux (=1) or not (=0)
   nn_sssr     =     0     !  add a damping     term in the surface freshwater flux (=2)
                           !  or to SSS only (=1) or no damping term (=0)
   rn_dqdt     =   -40.    !  magnitude of the retroaction on temperature   [W/m2/K]
   rn_deds     =   -27.7   !  magnitude of the damping on salinity   [mm/day]
   ln_sssr_bnd =   .false. !  flag to bound erp term (associated with nn_sssr=2)
   rn_sssr_bnd =   4.e0    !  ABS(Max/Min) value of the damping erp term [mm/day]
/
!-----------------------------------------------------------------------
&namsbc_alb    !   albedo parameters
!-----------------------------------------------------------------------
```

```
   rn_cloud    =    0.06   !  cloud correction to snow and ice albedo
   rn_albice   =    0.53   !  albedo of melting ice in the arctic and antarctic
   rn_alphd    =    0.80   !  coefficients for linear interpolation used to
   rn_alphc    =    0.65   !  compute albedo between two extremes values
   rn_alphdi   =    0.72   !  (Pyane, 1972)
/

!!======================================================================
!!              ***  Lateral boundary condition  ***
!!======================================================================
!!   namlbc        lateral momentum boundary condition
!!   namcla        cross land advection
!!   namobc        open boundaries parameters                         ("key_obc")
!!   namagrif      agrif nested grid ( read by child model only )      ("key_agrif")
!!   nambdy        Unstructured open boundaries                       ("key_bdy")
!!   namtide       Tidal forcing at open boundaries                   ("key_bdy_tides")
!!======================================================================
!
!-----------------------------------------------------------------------
&namlbc        !   lateral momentum boundary condition
!-----------------------------------------------------------------------
   rn_shlat    =    0.     !  shlat = 0  !  0 < shlat < 2  !  shlat = 2  !  2 < shlat
                           !  free slip  !   partial slip  !   no slip   ! strong slip
   ln_vorlat   = .false.   !  consistency of vorticity boundary condition with analytical eqs.
/
!-----------------------------------------------------------------------
&namcla        !   cross land advection
!-----------------------------------------------------------------------
   nn_cla      =    0      !  advection between 2 ocean pts separates by land
/
!-----------------------------------------------------------------------
&namobc        !   open boundaries parameters                         ("key_obc")
!-----------------------------------------------------------------------
   ln_obc_clim = .false.   !  climatological obc data files (T) or not (F)
   ln_vol_cst  = .true.    !  impose the total volume conservation (T) or not (F)
   ln_obc_fla  = .false.   !  Flather open boundary condition
   nn_obcdta   =    1      !  = 0 the obc data are equal to the initial state
                           !  = 1 the obc data are read in 'obc.dta' files
   cn_obcdta   = 'annual'  !  set to annual if obc datafile hold 1 year of data
                           !  set to monthly if obc datafile hold 1 month of data
   rn_dpein    =    1.     !  damping time scale for inflow at east  open boundary
   rn_dpwin    =    1.     !     -           -         -       west    -      -
   rn_dpnin    =    1.     !     -           -         -       north   -      -
   rn_dpsin    =    1.     !     -           -         -       south   -      -
   rn_dpeob    = 3000.     !  time relaxation (days) for the east  open boundary
   rn_dpwob    =   15.     !     -           -         -     west    -      -
   rn_dpnob    = 3000.     !     -           -         -     north   -      -
   rn_dpsob    =   15.     !     -           -         -     south   -      -
   rn_volemp   =    1.     !  = 0 the total volume change with the surface flux (E-P-R)
                           !  = 1 the total volume remains constant
/
!-----------------------------------------------------------------------
```

```
&namagrif      !  AGRIF zoom                                          ("key_agrif")
!-----------------------------------------------------------------------
   nn_cln_update =    3    !  baroclinic update frequency
   ln_spc_dyn    = .true.  !  use 0 as special value for dynamics
   rn_sponge_tra = 2880.   !  coefficient for tracer   sponge layer [m2/s]
   rn_sponge_dyn = 2880.   !  coefficient for dynamics sponge layer [m2/s]
/
!-----------------------------------------------------------------------
!       nam_tide       tide parameters (#ifdef key_tide)
!-----------------------------------------------------------------------
!  ln_tide_pot    = use tidal potential forcing
!  nb_harmo    = number of constituents used
!  name(1)     = 'M2', 'K1', etc name of constituent

&nam_tide
   ln_tide_pot          = .true.
   nb_harmo    = 11
   clname(1)     =   'M2'
   clname(2)     =   'S2'
   clname(3)     =   'N2'
   clname(4)     =   'K1'
   clname(5)     =   'O1'
   clname(6)     =   'Q1'
   clname(7)     =   'M4'
   clname(8)     =   'K2'
   clname(9)     =   'P1'
   clname(10)    =   'Mf'
   clname(11)    =   'Mm'
/
!-----------------------------------------------------------------------
&nambdy        !  unstructured open boundaries                        ("key_bdy")
!-----------------------------------------------------------------------
    nb_bdy = 1                            !  number of open boundary sets
    ln_coords_file = .true.               !  =T : read bdy coordinates from file
    cn_coords_file = 'coordinates.bdy.nc' !  bdy coordinates files
    ln_mask_file = .false.                !  =T : read mask from file
    cn_mask_file = ''                     !  name of mask file (if ln_mask_file=.TRUE.)
    nn_dyn2d      =  2                    !  boundary conditions for barotropic fields
    nn_dyn2d_dta  =  3                    !  = 0, bdy data are equal to the initial state
                                          !  = 1, bdy data are read in 'bdydata   .nc' files
                                          !  = 2, use tidal harmonic forcing data from files
                                          !  = 3, use external data AND tidal harmonic forcing
    nn_dyn3d      =  0                    !  boundary conditions for baroclinic velocities
    nn_dyn3d_dta  =  0                    !  = 0, bdy data are equal to the initial state
                               !  = 1, bdy data are read in 'bdydata   .nc' files
    nn_tra        =  1                    !  boundary conditions for T and S
    nn_tra_dta    =  1                    !  = 0, bdy data are equal to the initial state
                               !  = 1, bdy data are read in 'bdydata   .nc' files
    nn_rimwidth  = 10                      !  width of the relaxation zone
    ln_vol     = .false.                  !  total volume correction (see nn_volctl parameter)
    nn_volctl  = 1                        !  = 0, the total water flux across open boundaries is zero
/
```

```
!-----------------------------------------------------------------------
&nambdy_dta      !  open boundaries - external data           ("key_bdy")
!-----------------------------------------------------------------------
!            !   file name    ! frequency (hours) !  variable  ! time interpol. !  clim   ! 'yearly'/ ! weights  ! rotation !
!            !                !  (if <0  months)  !    name    !    (logical)   !  (T/F)  ! 'monthly' ! filename ! pairing  !
   bn_ssh =     'amm12_bdyT_u2d' ,         24        , 'sossheig' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_u2d =     'amm12_bdyU_u2d' ,         24        , 'vobtcrtx' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_v2d =     'amm12_bdyV_u2d' ,         24        , 'vobtcrty' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_u3d  =    'amm12_bdyU_u3d' ,         24        , 'vozocrtx' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_v3d  =    'amm12_bdyV_u3d' ,         24        , 'vomecrty' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_tem  =    'amm12_bdyT_tra' ,         24        , 'votemper' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   bn_sal  =    'amm12_bdyT_tra' ,         24        , 'vosaline' ,     .true.     , .false. ,  'daily'  ,    ''    ,   ''
   cn_dir  =    'bdydta/'
   ln_full_vel = .false.
/
!-----------------------------------------------------------------------
&nambdy_tide     ! tidal forcing at open boundaries
!-----------------------------------------------------------------------
   filtide      = 'bdydta/amm12_bdytide_'        !  file name root of tidal forcing files
    tide_cpt(1)   ='Q1'  !  names of tidal components used
    tide_cpt(2)   ='O1'  !  names of tidal components used
    tide_cpt(3)   ='P1'  !  names of tidal components used
    tide_cpt(4)   ='S1'  !  names of tidal components used
    tide_cpt(5)   ='K1'  !  names of tidal components used
    tide_cpt(6)   ='2N2' !  names of tidal components used
    tide_cpt(7)   ='MU2' !  names of tidal components used
    tide_cpt(8)   ='N2'  !  names of tidal components used
    tide_cpt(9)   ='NU2' !  names of tidal components used
    tide_cpt(10)   ='M2'  !  names of tidal components used
    tide_cpt(11)   ='L2'  !  names of tidal components used
    tide_cpt(12)   ='T2'  !  names of tidal components used
    tide_cpt(13)   ='S2'  !  names of tidal components used
    tide_cpt(14)   ='K2'  !  names of tidal components used
    tide_cpt(15)   ='M4'  !  names of tidal components used
    tide_speed(1)   = 13.398661 !  phase speeds of tidal components (deg/hour)
    tide_speed(2)   = 13.943036 !  phase speeds of tidal components (deg/hour)
    tide_speed(3)   = 14.958932 !  phase speeds of tidal components (deg/hour)
    tide_speed(4)   = 15.000001 !  phase speeds of tidal components (deg/hour)
    tide_speed(5)   = 15.041069 !  phase speeds of tidal components (deg/hour)
    tide_speed(6)   = 27.895355 !  phase speeds of tidal components (deg/hour)
    tide_speed(7)   = 27.968210 !  phase speeds of tidal components (deg/hour)
    tide_speed(8)   = 28.439730 !  phase speeds of tidal components (deg/hour)
    tide_speed(9)   = 28.512585 !  phase speeds of tidal components (deg/hour)
    tide_speed(10)   = 28.984106 !  phase speeds of tidal components (deg/hour)
    tide_speed(11)   = 29.528479 !  phase speeds of tidal components (deg/hour)
    tide_speed(12)   = 29.958935 !  phase speeds of tidal components (deg/hour)
    tide_speed(13)   = 30.000002 !  phase speeds of tidal components (deg/hour)
    tide_speed(14)   = 30.082138 !  phase speeds of tidal components (deg/hour)
    tide_speed(15)   = 57.968212 !  phase speeds of tidal components (deg/hour)
    ln_tide_date = .true.              !  adjust tidal harmonics for start date of run
/
!!======================================================================
```

```
!!                 ***  Bottom boundary condition  ***
!!======================================================================
!!   nambfr        bottom friction
!!   nambbc        bottom temperature boundary condition
!!   nambbl        bottom boundary layer scheme                          ("key_trabbl")
!!======================================================================
!
!-----------------------------------------------------------------------
&nambfr     !   bottom friction
!-----------------------------------------------------------------------
   nn_bfr      =    2      !  type of bottom friction :   = 0 : free slip,  = 1 : linear friction
                           !                              = 2 : nonlinear friction
   rn_bfri1    =    4.e-4  !  bottom drag coefficient (linear case)
   rn_bfri2    =    1.e-3  !  bottom drag coefficient (non linear case)
   rn_bfeb2    =    2.5e-3 !  bottom turbulent kinetic energy background  (m^2/s^2)
   ln_bfr2d    = .false.   !  horizontal variation of the bottom friction coef (read a 2D mask file )
   rn_bfrien   =    50.    !  local multiplying factor of bfr (ln_bfr2d = .true.)
   ln_bfrimp   = .true.    !  implicit bottom friction (requires ln_zdfexp = .false. if true)
/
!-----------------------------------------------------------------------
&nambbc        !   bottom temperature boundary condition
!-----------------------------------------------------------------------
   ln_trabbc   = .false.   !  Apply a geothermal heating at the ocean bottom
   nn_geoflx   =    0      !  geothermal heat flux: = 0 no flux
                           !     = 1 constant flux
                           !     = 2 variable flux (read in geothermal_heating.nc in mW/m2)
   rn_geoflx_cst = 86.4e-3 !  Constant value of geothermal heat flux [W/m2]
/
!-----------------------------------------------------------------------
&nambbl        !   bottom boundary layer scheme
!-----------------------------------------------------------------------
   nn_bbl_ldf  =  0      !  diffusive bbl (=1)   or not (=0)
   nn_bbl_adv  =  0      !  advective bbl (=1/2) or not (=0)
   rn_ahtbbl   =  1000.  !  lateral mixing coefficient in the bbl  [m2/s]
   rn_gambbl   =  10.    !  advective bbl coefficient                 [s]
/

!!======================================================================
!!                       Tracer (T & S ) namelists
!!======================================================================
!!   nameos        equation of state
!!   namtra_adv    advection scheme
!!   namtra_ldf    lateral diffusion scheme
!!   namtra_dmp    T & S newtonian damping
!!======================================================================
!
!-----------------------------------------------------------------------
&nameos        !   ocean physical parameters
!-----------------------------------------------------------------------
   nn_eos      =   2       !  type of equation of state and Brunt-Vaisala frequency
                           !     = 0, UNESCO (formulation of Jackett and McDougall (1994) and of McDougall (1987) )
                           !     = 1, linear: rho(T)   = rau0 * ( 1.028 - ralpha * T )
```

```
                        !     = 2, linear: rho(T,S) = rau0 * ( rbeta * S - ralpha * T )
   rn_alpha    =   2.0e-4  !  thermal expension coefficient (nn_eos= 1 or 2)
   rn_beta     =   7.7e-4  !  saline  expension coefficient (nn_eos= 2)
/
!-----------------------------------------------------------------------
&namtra_adv    !   advection scheme for tracer
!-----------------------------------------------------------------------
   ln_traadv_cen2   =  .false.  !  2nd order centered scheme
   ln_traadv_tvd    =  .true.   !  TVD scheme
   ln_traadv_muscl  =  .false.  !  MUSCL scheme
   ln_traadv_muscl2 =  .false.  !  MUSCL2 scheme + cen2 at boundaries
   ln_traadv_ubs    =  .false.  !  UBS scheme
   ln_traadv_qck    =  .false.  !  QUICKEST scheme
/
!----------------------------------------------------------------------------------
&namtra_ldf    !   lateral diffusion scheme for tracers
!----------------------------------------------------------------------------------
   !                        !  Operator type:
   ln_traldf_lap    =  .false.  !  laplacian operator
   ln_traldf_bilap  =  .true.   !  bilaplacian operator
   !                        !  Direction of action:
   ln_traldf_level  =  .false.  !  iso-level
   ln_traldf_hor    =  .true.  !  horizontal (geopotential)   (needs "key_ldfslp" when ln_sco=T)
   ln_traldf_iso    =  .false.   !  iso-neutral                 (needs "key_ldfslp")
   !           !  Griffies parameters              (all need "key_ldfslp")
   ln_traldf_grif   =  .false.  !  use griffies triads
   ln_traldf_gdia   =  .false.  !  output griffies eddy velocities
   ln_triad_iso     =  .false.  !  pure lateral mixing in ML
   ln_botmix_grif   =  .false.  !  lateral mixing on bottom
   !                        !  Coefficients
   ! Eddy-induced (GM) advection always used with Griffies; otherwise needs "key_traldf_eiv"
   ! Value rn_aeiv_0 is ignored unless = 0 with Held-Larichev spatially varying aeiv
   !                                  (key_traldf_c2d & key_traldf_eiv & key_orca_r2, _r1 or _r05)
   rn_aeiv_0        =     0.    !  eddy induced velocity coefficient [m2/s]
   rn_aht_0         =   -1.e9    !  horizontal eddy diffusivity for tracers [m2/s]
   rn_ahtb_0        =     0.    !  background eddy diffusivity for ldf_iso [m2/s]
   !                                           (normally=0; not used with Griffies)
/
!-----------------------------------------------------------------------
&namtra_dmp    !   tracer: T & S newtonian damping
!-----------------------------------------------------------------------
   ln_tradmp   =  .false.  !  add a damping termn (T) or not (F)
   nn_hdmp     =   -1      !  horizontal shape =-1, damping in Med and Red Seas only
                        !                   =XX, damping poleward of XX degrees (XX>0)
                        !                      + F(distance-to-coast) + Red and Med Seas
   nn_zdmp     =    1      !  vertical   shape =0    damping throughout the water column
                        !                   =1 no damping in the mixing layer (kz  criteria)
                        !                   =2 no damping in the mixed  layer (rho crieria)
   rn_surf     =   50.     !  surface time scale of damping   [days]
   rn_bot      =  360.     !  bottom  time scale of damping   [days]
   rn_dep      =  800.     !  depth of transition between rn_surf and rn_bot [meters]
   nn_file     =    1      !  create a damping.coeff NetCDF file (=1) or not (=0)
/

!!======================================================================
!!                       ***  Dynamics namelists  ***
!!======================================================================
!!   namdyn_adv    formulation of the momentum advection
!!   namdyn_vor    advection scheme
!!   namdyn_hpg    hydrostatic pressure gradient
!!   namdyn_spg    surface pressure gradient                            (CPP key only)
!!   namdyn_ldf    lateral diffusion scheme
!!======================================================================
!
!-----------------------------------------------------------------------
&namdyn_adv    !   formulation of the momentum advection
!-----------------------------------------------------------------------
   ln_dynadv_vec = .true.  !  vector form (T) or flux form (F)
   ln_dynadv_cen2= .false. !  flux form - 2nd order centered scheme
   ln_dynadv_ubs = .false. !  flux form - 3rd order UBS      scheme
/
!-----------------------------------------------------------------------
&namdyn_vor    !   option of physics/algorithm (not control by CPP keys)
!-----------------------------------------------------------------------
   ln_dynvor_ene = .false.  !  enstrophy conserving scheme
   ln_dynvor_ens = .false. !  energy conserving scheme
   ln_dynvor_mix = .false. !  mixed scheme
   ln_dynvor_een = .true. !  energy & enstrophy scheme
/
!-----------------------------------------------------------------------
&namdyn_hpg    !   Hydrostatic pressure gradient option
!-----------------------------------------------------------------------
   ln_hpg_zco  = .false.    !  z-coordinate - full steps
   ln_hpg_zps  = .true.   !  z-coordinate - partial steps (interpolation)
   ln_hpg_sco  = .false.   !  s-coordinate (standard jacobian formulation)
   ln_hpg_djc  = .false.   !  s-coordinate (Density Jacobian with Cubic polynomial)
   ln_hpg_prj  = .false.   !  s-coordinate (Pressure Jacobian scheme)
   ln_dynhpg_imp = .false. !  time stepping: semi-implicit time scheme  (T)
                                  !           centered      time scheme  (F)
/
!-----------------------------------------------------------------------
!namdyn_spg    !   surface pressure gradient   (CPP key only)
!-----------------------------------------------------------------------
!                          !  explicit free surface                   ("key_dynspg_exp")
!                          !  filtered free surface                   ("key_dynspg_flt")
!                          !  split-explicit free surface             ("key_dynspg_ts")

!-----------------------------------------------------------------------
&namdyn_ldf    !   lateral diffusion on momentum
!-----------------------------------------------------------------------
   !                       !  Type of the operator :
   ln_dynldf_lap    =  .false.  !  laplacian operator
   ln_dynldf_bilap  =  .true.   !  bilaplacian operator
   !                       !  Direction of action  :
```

```
   ln_dynldf_level  =  .false.  !  iso-level
   ln_dynldf_hor    =  .true.   !  horizontal (geopotential)            (require "key_ldfslp" in s-coord.)
   ln_dynldf_iso    =  .false.  !  iso-neutral                          (require "key_ldfslp")
   !                        !  Coefficient
   rn_ahm_0_lap     = 100000.   !  horizontal laplacian eddy viscosity   [m2/s]
   rn_ahmb_0        =     0.    !  background eddy viscosity for ldf_iso [m2/s]
   rn_ahm_0_blp     =     -1.e9    !  horizontal bilaplacian eddy viscosity [m4/s]
/

!!======================================================================
!!             Tracers & Dynamics vertical physics namelists
!!======================================================================
!!    namzdf        vertical physics
!!    namzdf_ric    richardson number dependent vertical mixing         ("key_zdfric")
!!    namzdf_tke    TKE dependent vertical mixing                       ("key_zdftke")
!!    namzdf_kpp    KPP dependent vertical mixing                       ("key_zdfkpp")
!!    namzdf_ddm    double diffusive mixing parameterization            ("key_zdfddm")
!!    namzdf_tmx    tidal mixing parameterization                       ("key_zdftmx")
!!======================================================================
!
!-----------------------------------------------------------------------
&namzdf        !   vertical physics
!-----------------------------------------------------------------------
   rn_avm0     =   1.2e-4  !  vertical eddy viscosity   [m2/s]          (background Kz if not "key_zdfcst")
   rn_avt0     =   1.2e-5  !  vertical eddy diffusivity [m2/s]          (background Kz if not "key_zdfcst")
   nn_avb      =    0      !  profile for background avt & avm (=1) or not (=0)
   nn_havtb    =    0      !  horizontal shape for avtb (=1) or not (=0)
   ln_zdfevd   = .true.    !  enhanced vertical diffusion (evd) (T) or not (F)
   nn_evdm     =    1      !  evd apply on tracer (=0) or on tracer and momentum (=1)
   rn_avevd    =  100.     !  evd mixing coefficient [m2/s]
   ln_zdfnpc   = .false.   !  Non-Penetrative Convective algorithm (T) or not (F)
   nn_npc      =    1            !  frequency of application of npc
   nn_npcp     =  365            !  npc control print frequency
   ln_zdfexp   = .false.   !  time-stepping: split-explicit (T) or implicit (F) time stepping
   nn_zdfexp   =    3            !  number of sub-timestep for ln_zdfexp=T
/
!-----------------------------------------------------------------------
&namzdf_ric    !   richardson number dependent vertical diffusion       ("key_zdfric" )
!-----------------------------------------------------------------------
   rn_avmri    = 100.e-4   !  maximum value of the vertical viscosity
   rn_alp      =   5.      !  coefficient of the parameterization
   nn_ric      =   2       !  coefficient of the parameterization
   rn_ekmfc    =   0.7     !  Factor in the Ekman depth Equation
   rn_mldmin   =   1.0     !  minimum allowable mixed-layer depth estimate (m)
   rn_mldmax   =1000.0     !  maximum allowable mixed-layer depth estimate (m)
   rn_wtmix    =  10.0     !  vertical eddy viscosity coeff [m2/s] in the mixed-layer
   rn_wvmix    =  10.0     !  vertical eddy diffusion coeff [m2/s] in the mixed-layer
   ln_mldw     = .true.    !  Flag to use or not the mized layer depth param.
/
!-----------------------------------------------------------------------
&namzdf_tke    !   turbulent eddy kinetic dependent vertical diffusion  ("key_zdftke")
!-----------------------------------------------------------------------
```

```
   rn_ediff    =   0.1     !  coef. for vertical eddy coef. (avt=rn_ediff*mxl*sqrt(e) )
   rn_ediss    =   0.7     !  coef. of the Kolmogoroff dissipation
   rn_ebb      =  67.83    !  coef. of the surface input of tke (=67.83 suggested when ln_mxl0=T)
   rn_emin     =   1.e-6   !  minimum value of tke [m2/s2]
   rn_emin0    =   1.e-4   !  surface minimum value of tke [m2/s2]
   nn_mxl      =   2       !  mixing length: = 0 bounded by the distance to surface and bottom
                           !                 = 1 bounded by the local vertical scale factor
                           !                 = 2 first vertical derivative of mixing length bounded by 1
                           !                 = 3 as =2 with distinct disspipative an mixing length scale
   nn_pdl      =   1       !  Prandtl number function of richarson number (=1, avt=pdl(Ri)*avm) or not (=0, avt=avm)
   ln_mxl0     = .true.    !  surface mixing length scale = F(wind stress) (T) or not (F)
   rn_mxl0     =   0.04    !  surface  buoyancy lenght scale minimum value
   ln_lc       = .true.    !  Langmuir cell parameterisation (Axell 2002)
   rn_lc       =   0.15    !  coef. associated to Langmuir cells
   nn_etau     =   0       !  penetration of tke below the mixed layer (ML) due to internal & intertial waves
                           !        = 0 no penetration
                           !        = 1 add a tke source below the ML
                           !        = 2 add a tke source just at the base of the ML
                           !        = 3 as = 1 applied on HF part of the stress    ("key_coupled")
   rn_efr      =   0.05    !  fraction of surface tke value which penetrates below the ML (nn_etau=1 or 2)
   nn_htau     =   1       !  type of exponential decrease of tke penetration below the ML
                           !        = 0  constant 10 m length scale
                           !        = 1  0.5m at the equator to 30m poleward of 40 degrees
/
!-----------------------------------------------------------------------
&namzdf_kpp    !   K-Profile Parameterization dependent vertical mixing  ("key_zdfkpp", and optionally:
!------------------------------------------------------------------------ "key_kppcustom" or "key_kpplktb")
   ln_kpprimix = .true.    !  shear instability mixing
   rn_difmiw   =  1.0e-04  !  constant internal wave viscosity [m2/s]
   rn_difsiw   =  0.1e-04  !  constant internal wave diffusivity [m2/s]
   rn_riinfty  =  0.8      !  local Richardson Number limit for shear instability
   rn_difri    =  0.0050   !  maximum shear mixing at Rig = 0    [m2/s]
   rn_bvsqcon  = -0.01e-07 !  Brunt-Vaisala squared for maximum convection [1/s2]
   rn_difcon   =  1.       !  maximum mixing in interior convection [m2/s]
   nn_avb      =  0        !  horizontal averaged (=1) or not (=0) on avt and amv
   nn_ave      =  1        !  constant (=0) or profile (=1) background on avt
/
!-----------------------------------------------------------------------
&namzdf_gls                !   GLS vertical diffusion                  ("key_zdfgls")
!-----------------------------------------------------------------------
   rn_emin       = 1.e-6   !  minimum value of e   [m2/s2]
   rn_epsmin     = 1.e-12  !  minimum value of eps [m2/s3]
   ln_length_lim = .true.  !  limit on the dissipation rate under stable stratification (Galperin et al., 1988)
   rn_clim_galp  = 0.53    !  galperin limit
   ln_crban      = .true.  !  Use Craig & Banner (1994) surface wave mixing parametrisation
   ln_sigpsi     = .true.  !  Activate or not Burchard 2001 mods on psi schmidt number in the wb case
   rn_crban      = 100.    !  Craig and Banner 1994 constant for wb tke flux
   rn_charn      = 70000.  !  Charnock constant for wb induced roughness length
   nn_tkebc_surf =   1     !  surface tke condition (0/1/2=Dir/Neum/Dir Mellor-Blumberg)
   nn_tkebc_bot  =   1     !  bottom tke condition (0/1=Dir/Neum)
   nn_psibc_surf =   1     !  surface psi condition (0/1/2=Dir/Neum/Dir Mellor-Blumberg)
   nn_psibc_bot  =   1     !  bottom psi condition (0/1=Dir/Neum)
```

```
   nn_stab_func  =   2     !  stability function (0=Galp, 1= KC94, 2=CanutoA, 3=CanutoB)
   nn_clos       =   1     !  predefined closure type (0=MY82, 1=k-eps, 2=k-w, 3=Gen)
/
!-----------------------------------------------------------------------
&namzdf_ddm    !   double diffusive mixing parameterization             ("key_zdfddm")
!-----------------------------------------------------------------------
   rn_avts     = 1.e-4     !  maximum avs (vertical mixing on salinity)
   rn_hsbfr    = 1.6       !  heat/salt buoyancy flux ratio
/
!-----------------------------------------------------------------------
&namzdf_tmx    !   tidal mixing parameterization                        ("key_zdftmx")
!-----------------------------------------------------------------------
   rn_htmx     = 500.      !  vertical decay scale for turbulence (meters)
   rn_n2min    = 1.e-8     !  threshold of the Brunt-Vaisala frequency (s-1)
   rn_tfe      = 0.333     !  tidal dissipation efficiency
   rn_me       = 0.2       !  mixing efficiency
   ln_tmx_itf  = .false.   !  ITF specific parameterisation
   rn_tfe_itf  = 1.        !  ITF tidal dissipation efficiency
/

!!======================================================================
!!                  ***  Miscelaneous namelists  ***
!!======================================================================
!!   nammpp           Massively Parallel Processing                     ("key_mpp_mpi)
!!   namctl           Control prints & Benchmark
!!   namsol           elliptic solver / island / free surface
!!======================================================================
!
!-----------------------------------------------------------------------
&namsol        !   elliptic solver / island / free surface
!-----------------------------------------------------------------------
   nn_solv     =      1    !  elliptic solver: =1 preconditioned conjugate gradient (pcg)
                           !                   =2 successive-over-relaxation (sor)
   nn_sol_arp  =      0    !  absolute/relative (0/1) precision convergence test
   rn_eps      =  1.e-6    !  absolute precision of the solver
   nn_nmin     =    210    !  minimum of iterations for the SOR solver
   nn_nmax     =    800    !  maximum of iterations for the SOR solver
   nn_nmod     =     10    !  frequency of test for the SOR solver
   rn_resmax   =  1.e-10   !  absolute precision for the SOR solver
   rn_sor      =  1.96     !  optimal coefficient for SOR solver (to be adjusted with the domain)
/
!-----------------------------------------------------------------------
&nammpp        !   Massively Parallel Processing                        ("key_mpp_mpi)
!-----------------------------------------------------------------------
   cn_mpi_send =  'I'      !  mpi send/recieve type   ='S', 'B', or 'I' for standard send,
                           !  buffer blocking send or immediate non-blocking sends, resp.
   nn_buffer   =   0       !  size in bytes of exported buffer ('B' case), 0 no exportation
   ln_nnogather=  .false.  !  activate code to avoid mpi_allgather use at the northfold
   jpni        =   4       !  jpni   number of processors following i (set automatically if < 1)
   jpnj        =   8       !  jpnj   number of processors following j (set automatically if < 1)
   jpnij       =   32      !  jpnij  number of local domains (set automatically if < 1)
/
!-----------------------------------------------------------------------
&namctl        !   Control prints & Benchmark
!-----------------------------------------------------------------------
   ln_ctl      = .true.   !  trends control print (expensive!)
   nn_print    =    0      !  level of print (0 no extra print)
   nn_ictls    =    0      !  start i indice of control sum (use to compare mono versus
   nn_ictle    =    0      !  end   i indice of control sum        multi processor runs
   nn_jctls    =    0      !  start j indice of control               over a subdomain)
   nn_jctle    =    0      !  end   j indice of control
   nn_isplt    =    1      !  number of processors in i-direction
   nn_jsplt    =    1      !  number of processors in j-direction
   nn_bench    =    0      !  Bench mode (1/0): CAUTION use zero except for bench
                           !     (no physical validity of the results)
   nn_timing   =    1      !  timing by routine activated (=1) creates timing.output file, or not (=0)
/

!!======================================================================
!!                  ***  Diagnostics namelists  ***
!!======================================================================
!!   namnc4       netcdf4 chunking and compression settings            ("key_netcdf4")
!!   namtrd       dynamics and/or tracer trends                        ("key_trddyn","key_trdtra","key_trdmld")
!!   namflo       float parameters                                     ("key_float")
!!   namptr       Poleward Transport Diagnostics
!!   namhsb       Heat and salt budgets
!!======================================================================
!
!-----------------------------------------------------------------------
&namnc4        !   netcdf4 chunking and compression settings           ("key_netcdf4")
!-----------------------------------------------------------------------
   nn_nchunks_i=   4       !  number of chunks in i-dimension
   nn_nchunks_j=   4       !  number of chunks in j-dimension
   nn_nchunks_k=   31      !  number of chunks in k-dimension
                           !  setting nn_nchunks_k = jpk will give a chunk size of 1 in the vertical which
                           !  is optimal for postprocessing which works exclusively with horizontal slabs
   ln_nc4zip   = .false.    !  (T) use netcdf4 chunking and compression
                           !  (F) ignore chunking information and produce netcdf3-compatible files
/
!-----------------------------------------------------------------------
&namtrd        !   diagnostics on dynamics and/or tracer trends        ("key_trddyn" and/or "key_trdtra")
!              !       or mixed-layer trends or barotropic vorticity    ("key_trdmld" or     "key_trdvor")
!-----------------------------------------------------------------------
   nn_trd      = 365       !  time step frequency dynamics and tracers trends
   nn_ctls     =   0       !  control surface type in mixed-layer trends (0,1 or n<jpk)
   rn_ucf      =   1.      !  unit conversion factor (=1 -> /seconds ; =86400. -> /day)
   cn_trdrst_in      = "restart_mld"   ! suffix of ocean restart name (input)
   cn_trdrst_out     = "restart_mld"   ! suffix of ocean restart name (output)
   ln_trdmld_restart = .false.         !  restart for ML diagnostics
   ln_trdmld_instant = .false.         !  flag to diagnose trends of instantantaneous or mean ML T/S
/
!-----------------------------------------------------------------------
&namflo       !   float parameters                                      ("key_float")
!-----------------------------------------------------------------------
```

```
   jpnfl        = 1         !  total number of floats during the run
   jpnnewflo    = 0         !  number of floats for the restart
   ln_rstflo    = .false.   !  float restart (T) or not (F)
   nn_writefl   =      75   !  frequency of writing in float output file
   nn_stockfl   =    5475   !  frequency of creation of the float restart file
   ln_argo      = .false.   !  Argo type floats (stay at the surface each 10 days)
   ln_flork4    = .false.   !  trajectories computed with a 4th order Runge-Kutta (T)
                            !  or computed with Blanke' scheme (F)
   ln_ariane    = .true.    !  Input with Ariane tool convention(T)
   ln_flo_ascii = .true.    !  Output with Ariane tool netcdf convention(F) or ascii file (T)
/
!-----------------------------------------------------------------------
&namptr      !   Poleward Transport Diagnostic
!-----------------------------------------------------------------------
   ln_diaptr  = .false.   !  Poleward heat and salt transport (T) or not (F)
   ln_diaznl  = .true.    !  Add zonal means and meridional stream functions
   ln_subbas  = .true.    !  Atlantic/Pacific/Indian basins computation (T) or not
                          !  (orca configuration only, need input basins mask file named "subbasins.nc"
   ln_ptrcomp = .true.    !  Add decomposition : overturning
   nn_fptr    =  1        !  Frequency of ptr computation [time step]
   nn_fwri    =  15       !  Frequency of ptr outputs [time step]
/
!-----------------------------------------------------------------------
&namhsb      !  Heat and salt budgets
!-----------------------------------------------------------------------
   ln_diahsb  = .false.   !  check the heat and salt budgets (T) or not (F)
/
!-----------------------------------------------------------------------
&namdct        ! transports through sections
!-----------------------------------------------------------------------
    nn_dct      = 60      !  time step frequency for transports computing
    nn_dctwri   = 60      !  time step frequency for transports writing
    nn_secdebug = 0       !      0 : no section to debug
                          !     -1 : debug all section
                          !  0 < n : debug section number n
/

!!======================================================================
!!           ***  Observation & Assimilation namelists ***
!!======================================================================
!!   namobs       observation and model comparison                     ('key_diaobs')
!!   nam_asminc   assimilation increments                              ('key_asminc')
!!======================================================================
!
!-----------------------------------------------------------------------
&namobs       !  observation usage switch                              ('key_diaobs')
!-----------------------------------------------------------------------
   ln_t3d     = .false.   ! Logical switch for T profile observations
   ln_s3d     = .false.   ! Logical switch for S profile observations
   ln_ena     = .false.   ! Logical switch for ENACT insitu data set
   !                      !     ln_cor                  Logical switch for Coriolis insitu data set
   ln_profb   = .false.   ! Logical switch for feedback insitu data set
   ln_sla     = .false.   ! Logical switch for SLA observations

   ln_sladt   = .false.   ! Logical switch for AVISO SLA data

   ln_slafb   = .false.   ! Logical switch for feedback SLA data
                          !     ln_ssh                  Logical switch for SSH observations

   ln_sst     = .false.   ! Logical switch for SST observations
                          !     ln_reysst               Logical switch for Reynolds observations
                          !     ln_ghrsst               Logical switch for GHRSST observations

   ln_sstfb   = .false.   ! Logical switch for feedback SST data
                          !     ln_sss                  Logical switch for SSS observations
                          !     ln_seaice               Logical switch for Sea Ice observations
                          !     ln_vel3d                Logical switch for velocity observations
                          !     ln_velavcur             Logical switch for velocity daily av. cur.
                          !     ln_velhrcur             Logical switch for velocity high freq. cur.
                          !     ln_velavadcp            Logical switch for velocity daily av. ADCP
                          !     ln_velhradcp            Logical switch for velocity high freq. ADCP
                          !     ln_velfb                Logical switch for feedback velocity data
                          !     ln_grid_global          Global distribtion of observations
                          !     ln_grid_search_lookup   Logical switch for obs grid search w/lookup table
                          !     grid_search_file        Grid search lookup file header
                          !     enactfiles              ENACT input observation file names
                          !     coriofiles              Coriolis input observation file name
!                         ! profbfiles: Profile feedback input observation file name
profbfiles = 'profiles_01.nc'
                          !     ln_profb_enatim         Enact feedback input time setting switch
                          !     slafilesact             Active SLA input observation file name
                          !     slafilespas             Passive SLA input observation file name
!                         ! slafbfiles: Feedback SLA input observation file name
slafbfiles = 'sla_01.nc'
                          !     sstfiles                GHRSST input observation file name
!                         ! sstfbfiles: Feedback SST input observation file name
sstfbfiles = 'sst_01.nc' 'sst_02.nc' 'sst_03.nc' 'sst_04.nc' 'sst_05.nc'
                          !     seaicefiles             Sea Ice input observation file name
                          !     velavcurfiles           Vel. cur. daily av. input file name
                          !     velhvcurfiles           Vel. cur. high freq. input file name
                          !     velavadcpfiles          Vel. ADCP daily av. input file name
                          !     velhvadcpfiles          Vel. ADCP high freq. input file name
                          !     velfbfiles              Vel. feedback input observation file name
                          !     dobsini                 Initial date in window YYYYMMDD.HHMMSS
                          !     dobsend                 Final date in window YYYYMMDD.HHMMSS
                          !     n1dint                  Type of vertical interpolation method
                          !     n2dint                  Type of horizontal interpolation method
                          !     ln_nea                  Rejection of observations near land switch
nmsshc     = 0            ! MSSH correction scheme
                          !     mdtcorr                 MDT  correction
                          !     mdtcutoff               MDT cutoff for computed correction
ln_altbias = .false.      ! Logical switch for alt bias
ln_ignmis  = .true.       ! Logical switch for ignoring missing files
                          !     endailyavtypes   ENACT daily average types
```

```
   ln_grid_global = .true.
   ln_grid_search_lookup = .false.
/
!-----------------------------------------------------------------------
&nam_asminc   !   assimilation increments                               ('key_asminc')
!-----------------------------------------------------------------------
    ln_bkgwri = .false.    !  Logical switch for writing out background state
    ln_trjwri = .false.    !  Logical switch for writing out state trajectory
    ln_trainc = .false.    !  Logical switch for applying tracer increments
    ln_dyninc = .false.    !  Logical switch for applying velocity increments
    ln_sshinc = .false.    !  Logical switch for applying SSH increments
    ln_asmdin = .false.    !  Logical switch for Direct Initialization (DI)
    ln_asmiau = .false.    !  Logical switch for Incremental Analysis Updating (IAU)
    nitbkg    = 0          !  Timestep of background in [0,nitend-nit000-1]
    nitdin    = 0          !  Timestep of background for DI in [0,nitend-nit000-1]
    nitiaustr = 1          !  Timestep of start of IAU interval in [0,nitend-nit000-1]
    nitiaufin = 15         !  Timestep of end of IAU interval in [0,nitend-nit000-1]
    niaufn    = 0          !  Type of IAU weighting function
    nittrjfrq = 0          !  Frequency of trajectory output for 4D-VAR
    ln_salfix = .false.    !  Logical switch for ensuring that the sa > salfixmin
    salfixmin = -9999      !  Minimum salinity after applying the increments
    nn_divdmp = 0          !  Number of iterations of divergence damping operator
/
!-----------------------------------------------------------------------
```

```
&namsbc_wave   ! External fields from wave model
!-----------------------------------------------------------------------
!              !  file name  ! frequency (hours) ! variable  ! time interp. !  clim  ! 'yearly'/ ! weights  ! rotation !
!              !             !  (if <0  months)  !   name    !   (logical)  !  (T/F) ! 'monthly' ! filename ! pairing  !
   sn_cdg      =  'cdg_wave' ,        1          , 'drag_coeff' , .true.   , .false. , 'daily'  ,''         , ''
!
   cn_dir_cdg  = './'  !  root directory for the location of drag coefficient files
/
!-----------------------------------------------------------------------
&namdyn_nept  !   Neptune effect (simplified: lateral and vertical diffusions removed)
!-----------------------------------------------------------------------
   ! Suggested lengthscale values are those of Eby & Holloway (1994) for a coarse model
   ln_neptsimp       = .false.  ! yes/no use simplified neptune

   ln_smooth_neptvel = .false.  ! yes/no smooth zunep, zvnep
   rn_tslse          =  1.2e4   ! value of lengthscale L at the equator
   rn_tslsp          =  3.0e3   ! value of lengthscale L at the pole
   ! Specify whether to ramp down the Neptune velocity in shallow
   ! water, and if so the depth range controlling such ramping down
   ln_neptramp       = .false.  ! ramp down Neptune velocity in shallow water
   rn_htrmin         =  100.0   ! min. depth of transition range
   rn_htrmax         =  200.0   ! max. depth of transition range
/
```