

WN/CMGC/08/7

**Portage ARPEGE/NEMO
sur IBM Blue Gene/L**

Eric Maisonnave

Table of Contents

| | |
|--|----|
| Caractéristiques de la machine..... | 4 |
| Portage et performances du modèle ARPEGE..... | 5 |
| Portage et performances du modèle OPA..... | 6 |
| Portage et performances de la configuration couplée..... | 8 |
| Conclusion..... | 10 |

Au cours des mois de décembre 07 – janvier 08, nous avons procédé au portage du modèle couplé haute résolution ARPEGE-NEMO sur la machine IBM Blue Gene L du Cerfacs. Dans cette note technique, nous décrirons les différentes étapes de la mise en place de chaque modèle pris individuellement (mode forcé), puis couplés avec la version 3 d'OASIS.

1. Caractéristiques de la machine

L'IBM Blue Gene est configuré de la manière suivante :

- 1 noeud de login quadri processeur Power5+ 1.5GHz 8 GO de mémoire
- 1 noeud de service quadri processeur Power5 2GHz 4 GO de mémoire
- 1024 noeuds de calcul soit 2048 processeurs Power PC440
- 16 noeuds d'IO (soit 1 noeud d'IO pour 64 noeuds de calcul)
- 2 serveurs GPFS des répertoires utilisateurs.

Les noeuds de calcul sont répartis dans 2 "midplanes". Chaque midplane" contient 16 "node cards".

Chaque "node card" contient 32 noeuds et 0 à 2 cartes IO.

Le noeud possède les caractéristiques suivantes :

- 2 processeurs PowerPC440 700 Mhz (4 Flops x 700 Mhz =2.8 GFlops de performance crête par processeur, adressage mémoire 32 bits)
- 512 MO de mémoire par processeur soit 1GO par noeud
- Caches L1 (instruction et données) 32 KB par processeur, cache L2 4KB par processeur, cache L3 4MB par noeud

Trois réseaux d'interconnexion sont utilisés pour les calculs sur Blue Gene :

- un tore 3D entre les noeuds de calcul, 2.1 GO/s de bande passante, entre 4 et 10 μ s de latence, pour les communications MPI point à point
- un arbre entre noeuds de calcul et noeuds IO, 700 MO/s de bande passante, 5 μ s de latence pour les IO, les communications globales, opérations de réduction
- un réseau pour les synchronisations globales (MPI_BARRIER) et interruptions, 1.3 μ s de latence

Deux modes de calcul sont disponibles sur blue gene :

- Coprocesseur mode (CO) : le CPU0 gère le calcul, le CPU1 gère les communications point à point, La communication recouvre le calcul.
Performance crête de $5.6/2=2.8$ Gflops

- Virtual node mode (VN) : CPU0 et CPU1 gère des tâches indépendantes « virtuelles », le calcul et la communication ne se recouvrent pas. La performance est de 5.6 Gflops mais on divise par 2 : la mémoire disponible (512 MO au lieu de 1GO), le cache L3 (2MO au lieu de 4), la bande passante réseau.

Le mode coprocesseur bénéficie aux applications communiquant beaucoup, alors que le mode virtuel convient aux applications plus CPU intensive, n'ayant pas besoin de plus de 512 MO de mémoire.

2. Portage et performances du modèle ARPEGE

Le modèle atmosphérique testé est issu des versions mises en place au Cerfacs depuis plusieurs années dans le cadre des travaux d'adaptation nécessaires au calcul sur plateformes scalaires.

Ce code est, pour la physique comme pour la dynamique, en tout point similaire à la version 4.6 d'ARPEGE Climat utilisée au CNRM sur la machine vectorielle NEC SX8R de Météo-France. La résolution choisie est relativement élevée: c'est celle de la troncature t1159, grille globale non étirée, avec un maillage régulier d'environ 1 degré de côté (320x160) et 31 niveaux sur la verticale.

La paramétrisation physique utilisée est celle du modèle couplé basse-résolution. Une nouvelle paramétrisation plus adaptée à la résolution et prenant en compte les spécificités du modèle en mode couplé devra être implémentée pour donner une représentation plus vraisemblable des calculs qui seront nécessaires à la conduite d'une expérience à cette résolution.

Options de compilation: -qarch=440d -qrealsize=8 -qhot=simd -qxlf90=nosignedzero -O3

| Nb proc | Mode calcul | Temps restitution |
|---------|-------------|-------------------|
| | | (1 mois) en min |
| 32 | VN | 114 |
| 32 | CO | 102 |
| 48 | CO | 72 |
| 64 | CO | 57 |
| 96 | CO | 48 |
| 96 | VN | 49 |
| 110 | CO | 45 |

Tableau 1: Temps de restitution pour une simulation ARPEGE d'un mois (en minutes)

On note que, bien que proche des meilleures performances, notre configuration ne peut les atteindre car la version 4 d'ARPEGE que nous utilisons ne permet pas une parallélisation par boîtes mais seulement par bandes de latitudes, c'est à dire

seulement en Y alors qu'un découpage des domaines en X et en Y serait nécessaire.

A la troncature 159, le nombre de domaines autorisés ne peut dépasser 110. Cette limitation aurait des conséquences encore plus négatives pour des troncatures plus élevées. Fort heureusement, la future version 5 du modèle rendra possible la parallélisation par boîtes et permettra donc de tester plus largement la scalabilité du modèle pour des configurations de plus haute résolution.

Pour le moment, dans le meilleur des cas, notre modèle peut donc terminer **une simulation atmosphérique forcée de 10 ans en un peu plus de 3 jours**, ce qui satisfait aux critères de faisabilité de nombreux types d'expériences (AMIP, tests de paramétrisation ...)

3. Portage et performances du modèle OPA

La configuration océanique utilisée est OPA9 (NEMO) v2, telle que délivrée en juin 2007, avec la namelist jointe en annexe. A noter que dans tous les cas, nitend=nstock=nwrite ce qui limite beaucoup la production d'IO, accélère la simulation (voir la comparaison des chiffres pour 5 jours et 30 jours de run) et donne de meilleurs résultats en terme de reproductibilité des performances.

Le modèle comprend la glace de mer. Son maillage est celui d'ORCA05 (722x511), avec 31 niveaux verticaux.

Options de compilation: -qarch=440d -qhot=simd -qxl90=nosignedzero -O3 -qrealsize=8 -qfltrap=overflow.

Clefs CPP: key_trabbl_dif key_vectopt_loop key_vectopt_memory key_orca_r05 key_ice_lim key_lim_cp2 key_lim_fdd key_ldfslp key_dynspgflt key_traldf_c2d key_traldf_eiv key_dynldf_c2d key_dtatem key_dtasal key_trabbc key_zdfcke key_zdfddm key_diahth key_mpp_mpi key_flx_core key_tradmp

Les options d'optimisation maximales sont utilisées (-qarch=440d -qhot=simd -O3). L'option -qfltrap=overflow est rendue nécessaire du fait de la présence d'une erreur non identifiée se manifestant par l'écrasement d'un tableau dans la routine dynspgflt, qui conduit à des valeurs de courants de surface supérieures à 20 m/s.

| Nb proc | Jpni x jpnj | Mode calcul | MPI mode | Temps restitution | Temps restitution |
|---------|-------------|-------------|----------|---------------------------------|---------------------------------|
| | | | | Pour 480 ts (5 jours) en min | pour 2880 ts (1 mois) en min |
| 30 | 5x6 | CO | I | 25 | |
| 64 | 8x8 | CO | I | 20 | 80 |
| 128 | 8x16 | CO | I | 15 | 48 |
| 128 | 8x16 | VN | I | 15 | 45 |
| 128 | 8x16 | VN | S | | 63 |
| 256 | 16x16 | VN | I | | 39 |
| 512 | 32x16 | VN | I | | 41 |

Tableau 2: Temps de restitution pour une simulation OPA de 5 jours ou de 1 mois (en minutes)

Le portage de NEMO de la machine vectorielle NEC SX8R à la machine scalaire IBM BG/L se solde par un surcoût de 150% environ. Alors qu'il faut 15 min sur 5 processeurs vectoriels pour réaliser une simulation d'un mois (30 heures pour 10 ans), la même simulation sur un nombre maximum de 256 processeurs sur IBM BG/L prendra 39 minutes (**3 jours pour 10 ans**).

Un traçage des temps de chaque routine a été mené en se servant de l'utilitaire gmon. Le tableau ci-dessous classe les routines par pourcentage de temps utilisé durant l'exécution. Le test a été fait avec la configuration à 64 processeurs.

| % time | cumulative seconds | self seconds | calls | self s/call | total s/call | name |
|--------|--------------------|--------------|-------|-------------|--------------|---------------------------------|
| 22.72 | 159.79 | 159.79 | | | | BGLML Messenger_CMadvance |
| 9.56 | 227 | 67.21 | 60 | 1.12 | 1.12 | Limrhg NMOD_lim_rhg |
| 6.89 | 275.46 | 48.46 | 180 | 0.27 | 0.27 | Solsor NMOD_sol_sor |
| 4.92 | 310.08 | 34.62 | 180 | 0.19 | 0.19 | traldf_iso NMOD_tra_ldf_iso |
| 4.44 | 341.3 | 31.22 | | | | BGLML Messenger_tree_advance |
| 4.3 | 371.53 | 30.23 | 180 | 0.17 | 0.18 | Ldfslp NMOD_ldf_slp |
| 3.67 | 397.31 | 25.78 | 180 | 0.14 | 0.15 | Zdftke NMOD_zdf_tke |
| 3.26 | 420.24 | 22.93 | 180 | 0.13 | 0.34 | traadv_ubs NMOD_tra_adv_ubs |
| 3.22 | 442.86 | 22.62 | 360 | 0.06 | 0.06 | traadv_ubs NMOD_nonosc_z |

| % time | cumulative seconds | self seconds | calls | self s/call | total s/call | name |
|-----------|-----------------------|-----------------|-------|----------------|-----------------|--------------------------------|
| 3.06 | 464.41 | 21.55 | | | | BGLML Messenger_advance |
| 2.62 | 482.83 | 18.42 | 180 | 0.1 | 0.1 | trazdf_imp NMOD_tra_zdf_imp |
| 2.22 | 498.48 | 15.65 | 180 | 0.09 | 0.09 | dynzdf_imp NMOD_dyn_zdf_imp |

On peut constater la place prépondérante des routines de communication (BGLML Messenger), place qui croit naturellement avec le nombre de processeurs utilisés. Les temps d'IO ne sont pas compris dans ce décompte.

4. Portage et performances de la configuration couplée

Pour terminer, un couplage utilisant les deux composantes précédemment décrites a été implémenté.

C'est la version OASIS3 "parallèle par champs de couplage" (utilisé sur le Earth Simulator) qui a été choisie pour assembler le modèle. Elle utilise deux processeurs seulement, étant donné le faible nombre de champs de couplage utilisés dans cette configuration de test.

A noter que le mode MPMD de MPI n'est pas disponible sur BG/L. Un utilitaire de lancement a donc été proposé par IBM pour les besoins d'OASIS.

La mise en place de la configuration couplée exige des temps de développement et d'optimisation bien plus importants que ceux requis par l'installation des composantes océaniques et atmosphériques séparées. Ce problème, hélas trop souvent négligé à chaque nouveau benchmark, requiert une coopération plus étroite avec les spécialistes IBM du fonctionnement de la machine Blue Gene.

La présentation des performances du modèle couplé fera donc l'objet d'une communication ultérieure.

Conclusion

Avant toute modification préalable des codes sources des modèles ARPEGE et NEMO, il s'avère que leur utilisation à une résolution relativement élevée (t159 dans l'atmosphère, 1/2 degré dans l'océan) donne des temps de restitution acceptables sur la machine scalaire Blue Gene/L disponible au Cerfacs.

Ces résultats sont observés pour les configuration forcées des deux modèles. Ils n'ont pu être reproduits pour l'instant en mode couplé.

Les résultats obtenus sur notre machine scalaire nous encouragent à mettre en chantier le développement de configurations massivement parallèles. Cependant, lors du choix des plateformes appelées à accueillir ces futures configurations, il sera important de ne pas limiter les essais de qualification aux seules composantes isolées de notre CGCM mais d'y associer explicitement la configuration couplée.

[illegible]

```

ln_dimgnnn = .FALSE.
/
!-----
!   nam_ctl    Control prints & Benchmark
!-----
! ln_ctl    trends control print (expensive!)
! nprint    level of print (0 no print)
! nictls    start i indice to make the control SUM (very usefull to compare mono-
! nictle    end  i indice to make the control SUM (-versus multi processor runs)
! njctls    start j indice to make the control SUM (very usefull to compare mono-
! njctle    end  j indice to make the control SUM (-versus multi processor runs)
! nisplt    number of processors following i
! njsplt    number of processors following j
! nbench    Bench parameter (0/1): CAUTION it must be zero except for bench
!           for which we don't care about physical meaning of the results
! nbit_cmp  bit comparison mode parameter (0/1): enables bit comparison between
!           single and multiple processor runs.
&namctl
  ln_ctl = .false.
  nprint =    0
  nictls =    0
  nictle =    0
  njctls =    0
  njctle =    0
  isplt =    1
  jsplt =    1
  nbench =    0
  nbit_cmp =    0
/
!-----
!   nam_mpp    Massively Parallel Processing
!-----
! c_mpi_send    mpi send/recieve type
!               = 'S' : standard blocking send
!               = 'B' : buffer blocking send
!               = 'T' : immediate non-blocking send
&nam_mpp
  c_mpi_send = 'T'
/
!-----
!   nam_zgr    vertical coordinate
!-----
! ln_zco    z-coordinate - full steps    (T/F)
! ln_zps    z-coordinate - partial steps (T/F)
! ln_sco    s- or hybrid z-s-coordinate (T/F)
&nam_zgr
  ln_zco = .true.
  ln_zps = .false.

```

```

ln_sco = .false.
/
!-----
!   nam_zgr_sco  s-coordinate or hybrid z-s-coordinate
!-----
! sbot_min  minimum depth of s-bottom surface (>0) (m)
! sbot_max  maximum depth of s-bottom surface (= ocean depth) (>0) (m)
! theta     surface control parameter (0<=theta<=20)
! thetb     bottom control parameter (0<=thetb<= 1)
! r_max     maximum cut-off r-value allowed (0<r_max<1)
&nam_zgr_sco
  sbot_min = 300.
  sbot_max = 5250.
  theta    = 6.0
  thetb    = 0.75
  r_max    = 0.15
/
!-----
!   nam_traadv  advection scheme for tracer (option not control by CPP keys)
!-----
! ln_traadv_cen2  2nd order centered scheme (default T)
! ln_traadv_tvd   TVD scheme (default F)
! ln_traadv_muscl MUSCL scheme (default F)
! ln_traadv_muscl2 MUSCL2 scheme (default F)
! ln_traadv_ubs   UBS scheme (default F)
&nam_traadv
  ln_traadv_cen2 = .false.
  ln_traadv_tvd  = .false.
  ln_traadv_muscl = .false.
  ln_traadv_muscl2 = .false.
  ln_traadv_ubs  = .true.
  ln_traadv_qck  = .false.
/
!-----
!   nam_traldf  lateral diffusion scheme for tracer (option not control by CPP keys)
!-----
! Type of the operator :
! ln_traldf_lap  laplacian operator (default T)
! ln_traldf_bilap bilaplacian operator (default F)
! Direction of action :
! ln_traldf_level iso-level (default F)
! ln_traldf_hor  horizontal (geopotential) (default F)^**
! ln_traldf_iso  iso-neutral (default T)^*
! Coefficient
! aht0  horizontal eddy diffusivity for tracers (m2/s)
! ahtb0 background eddy diffusivity for isopycnal diffusion (m2/s)
! aeiv0 eddy induced velocity coefficient (m2/s)
! ^* require key_ldfslp to compute the direction of the lateral diffusion

```

```

! ^** require key_ldfslp in s-coordinate
&nam_traldf
  ln_traldf_lap = .true.
  ln_traldf_bilap = .false.
  ln_traldf_level = .false.
  ln_traldf_hor = .false.
  ln_traldf_iso = .true.
  aht0 = 500.
  ahtb0 = 0.
  aeiv0 = 500.
/
!-----
!   nam_dynldf  lateral diffusion on momentum
!-----
! Type of the operator :
!   ln_dynldf_lap  laplacian operator      (default T)
!   ln_dynldf_bilap bilaplacian operator   (default F)
! Direction of action :
!   ln_dynldf_level iso-level              (default F)
!   ln_dynldf_hor  horizontal (geopotential) (default F)^**
!   ln_dynldf_iso  iso-neutral            (default T)^*
! Coefficient
! ahm0  horizontal eddy viscosity for the dynamics (m2/s)
! ahmb0 background eddy viscosity for isopycnal diffusion (m2/s)
&nam_dynldf
  ln_dynldf_lap = .false.
  ln_dynldf_bilap = .true.
  ln_dynldf_level = .true.
  ln_dynldf_hor = .false.
  ln_dynldf_iso = .false.
  ahm0 = -8.5e+11
  ahmb0 = 0.
/
!-----
!   namflg  algorithm flags (algorithm not control by CPP keys)
!-----
! ln_dynhpg_imp  hydrostatic pressure gradient: semi-implicit time scheme (T)
!               centered time scheme (F)
! nn_dynhpg_rst add dynhpg implicit variables in restart or not (1/0)
&namflg
  ln_dynhpg_imp = .true.
  nn_dynhpg_rst = 0
/
!-----
!   nam_dynhpg  Hydrostatic pressure gradient option
!-----
! type of pressure gradient scheme (choose one only!)
!   ln_hpg_zco  z-coordinate - full steps          (default T)

```

```

! ln_hpg_zps  z-coordinate - partial steps (interpolation)
! ln_hpg_sco  s-coordinate (standard jacobian formulation)
! ln_hpg_hel  s-coordinate (helsinki modification)
! ln_hpg_wdj  s-coordinate (weighted density jacobian)
! ln_hpg_djc  s-coordinate (Density Jacobian with Cubic polynomial)
! ln_hpg_rot  s-coordinate (ROTated axes scheme)
! parameters
! gamm       weighting coefficient (wdj scheme)
&nam_dynhpg
  ln_hpg_zco = .true.
  ln_hpg_zps = .false.
  ln_hpg_sco = .false.
  ln_hpg_hel = .false.
  ln_hpg_wdj = .false.
  ln_hpg_djc = .false.
  ln_hpg_rot = .false.
  gamm      = 0.e0
/
!-----
!   nam_dynadv  option of physics/algorithm (not control by CPP keys)
!-----
! ln_dynadv_vec  vector form flag
! ln_dynadv_cen2  flux form - 2nd order centered scheme  (default T)
! ln_dynadv_ubs  flux form - 3rd order UBS scheme      (default F)
&nam_dynadv
  ln_dynadv_vec = .TRUE.
  ln_dynadv_cen2 = .FALSE.
  ln_dynadv_ubs = .FALSE.
/
!-----
!   nam_dynvor  option of physics/algorithm (not control by CPP keys)
!-----
! ln_dynvor_ens  vorticity trends: enstrophy conserving scheme (default T)
! ln_dynvor_ene  "      " : energy conserving scheme  (default F)
! ln_dynvor_mix  "      " : mixed scheme              (default F)
! ln_dynvor_een  "      " : energy & enstrophy scheme (default F)
&nam_dynvor
  ln_dynvor_ene = .FALSE.
  ln_dynvor_ens = .TRUE.
  ln_dynvor_mix = .FALSE.
  ln_dynvor_een = .FALSE.
/
!-----
!   namtau  surface wind stress
!-----
! ntau000  gently increase the stress over the first ntau_rst time-steps
! tau0x    uniform value used as default surface heat flux
! tau0y    uniform value used as default solar radiation flux

```

```

&namtau
  ntau000 = 0
  tau0x = 0.e0
  tau0y = 0.e0
/
!-----
!   namflx  surface fluxes
!-----
! q0      uniform value used as default surface heat flux
! qsr0    uniform value used as default solar radiation flux
! emp0    uniform value used as default surface freshwater budget (E-P)
! dqdt0   feedback coefficient for SST damping (W/m2/K)
! deds0   feedback coefficient for SSS damping (mm/day)
&namflx
  q0      = 0.e0
  qsr0    = 0.e0
  emp0    = 0.e0
  dqdt0   = -40.
  deds0   = 0.
/
!-----
!   namalb  albedo parameters
!-----
! cgren   correction of the snow or ice albedo to take into account
! albice  albedo of melting ice in the arctic and antarctic
! alphd   coefficients for linear interpolation used to compute albedo
!         between two extremes values (Pyane, 1972)
! alphc   "
! alphdi  "
&namalb
  cgren   = 0.06
  albice  = 0.5
  alphd   = 0.80
  alphc   = 0.65
  alphdi  = 0.72
/
!-----
!   namdom  space and time domain (bathymetry, mesh, timestep)
!-----
! ntopo   = 0/1 ,compute/read the bathymetry file
!         (mbathy, nb of T-ocean levels)
! e3zps_min the thickness of the partial step is set larger than the
! e3zps_rat the minimum of e3zps_min and e3zps_rat * e3t
!         (N.B. 0<e3zps_rat<1)
! nmsh    =1 create a mesh file (coordinates, scale factors, masks)
! nacc     the acceleration of convergence method
!         = 0, no acceleration, rdt = rdttra
!         = 1, acceleration used, rdt < rdttra(k)

```

```

! atfp    asselin time filter parameter
! rdt     time step for the dynamics (and tracer if nacc=0)
! rdtmin  minimum time step on tracers
! rdtmax  maximum time step on tracers
! rdth    depth variation of tracer time step
! rdtbt   barotropic time step (for the time splitting algorithm)
! nfice   frequency of ice model call
! nfbulk  frequency of bulk formula call (not used if ice used)
! nclosea = 0 no closed sea
!         = 1 closed sea (Black Sea, Caspian Sea, Great US Lakes...)
&namdom
  ntopo   = 1
  e3zps_min = 50.
  e3zps_rat = 0.1
  nmsh    = 0
  nacc     = 0
  atfp    = 0.1
  rdt     = 2400.
  rdtmin  = 2400.
  rdtmax  = 2400.
  rdth    = 1200.
  rdtbt   = 90.
  nfice   = 3
  nfbulk  = 3
  nclosea = 0
/
!-----
!   namfwb  freshwater budget correction
!-----
! ln_fwb   logical flag for freshwater budget correction (0 annual mean)
&namfwb
  ln_fwb   = .false.
/
!-----
!   namptr  Poleward Transport Diagnostic
!-----
! ln_diaptr logical flag for Poleward transport computation
! ln_subbas logical flag for Atlantic/Pacific/Indian basins computation
!           need input basins mask file named "subbasins.nc"
! nf_ptr   Frequency of computation
&namptr
  ln_diaptr = .true.
  ln_subbas = .true.
  nf_ptr    = 1116
/
!-----
!   namcro  cross land advection
!-----

```



```

! n_cla  advection between 2 ocean pts separates by land
&namcla
  n_cla  = 0
/
!-----
!   namzdf  vertical physics
!-----
! ln_zdfevd  enhanced vertical diffusion      (default T)
! ln_zdfnpc  Non-Penetrative Convection      (default T)
! avm0      vertical eddy viscosity for the dynamic (m2/s)
! avt0      vertical eddy diffusivity for tracers (m2/s)
! avevd     vertical coefficient for enhanced diffusion scheme (m2/s)
! nevdm     = 0  apply enhanced mixing on tracer only
!           = 1  apply enhanced mixing on both tracer and momentum
! ln_zdfexp  vertical physics: (=T) time splitting (T)  (Default=F)
!           (=F) euler backward (F)
! n_zdfexp  number of sub-timestep for time splitting scheme
&namzdf
  ln_zdfevd = .true.
  ln_zdfnpc = .false.
  avm0      = 1.e-5
  avt0      = 1.e-6
  avevd     = 100.
  n_evdm    = 1
  ln_zdfexp = .false.
  n_zdfexp  = 3
/
!-----
!   namnpc  vnon penetrative convection
!-----
! nnpc1  non penetrative convective scheme frequency
! nnpc2  non penetrative convective scheme print frequency
&namnpc
  nnpc1 = 1
  nnpc2 = 365
/
!-----
!   nambb1  bottom boundary layer scheme
!-----
! atrbb1  lateral tracer coeff. for bottom boundary layer scheme(m2/s)
&nambb1
  atrbb1 = 10000.
/
!-----
!   namtke  turbulent eddy kinetic dependent vertical diffusion
!           ( #ifdef "key_zdftke" )
!-----
! ln_rstke flag to restart with tke from a run without tke (default F)

```

```

! ediff  coef. to compute vertical eddy coef. (avt=ediff*mxl*sqrt(e) )
! ediss  coef. of the Kolmogoroff dissipation
! ebb    coef. of the surface input of tke
! efave  coef. to applied to the tke diffusion ( avtke=efave*avm )
! emin   minimum value of tke (m^2/s^2)
! emin0  surface minimum value of tke (m^2/s^2)
! nitke  number of restart iterative loops
! ri_c   critic richardson number
! nmxl   flag on mixing length used
!        = 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
! npdl   flag on prandtl number
!        = 0 no vertical prandtl number (avt=avm)
!        = 1 prandtl number function of richarson number (avt=pdl*avm)
!        = 2 same as = 1 but a shapiro filter is applied on pdl
! nave   = horizontal averaged (=1) or not (=0) of avt (default =1)
! navb   = 0 cst background avt0, avm0 / =1 profile used on avtb
&namtke
  ln_rstke = .false.
  ediff =    0.1
  ediss =    0.7
  ebb  =    60.
  efave =     1.
  emin =  1.e-6
  emin0 = 1.e-4
  nitke =    50
  nmxl =     3
  npdl =     1
  navb =     1
/
!-----
!   namddm  double diffusive mixing parameterization
!-----
! avts  maximum avs for dd mixing
! hsbfr heat/salt buoyancy flux ratio
&namddm
  avts = 1.e-4
  hsbfr = 1.6
/
!-----
!   namlbc  lateral momentum boundary condition
!-----
! shlat  lateral boundary condition on velocity
!        shlat = 0 , free slip
!        0 < shlat < 2 , partial slip
!        shlat = 2 , no slip
!        2 < shlat   , strong slip

```

```

&namlbc
  shlat = 2.
/
!-----
!   nambfr  bottom friction
!-----
! nbotfr type of bottom friction
!       nbotfr = 0 , no slip
!       nbotfr = 1 , linear friction
!       nbotfr = 2 , nonlinear friction
!       nbotfr = 3 , free slip
! bfri1  bottom drag coefficient (linear case)
! bfri2  bottom drag coefficient (non linear case)
! bfeb2  bottom turbulent kinetic energy (m^2/s^2)
&nambfr
  nbotfr = 2
  bfri1 = 4.e-4
  bfri2 = 1.e-3
  bfeb2 = 2.5e-3
/
!-----
!   nambbc  bottom temperature boundary condition
!-----
! ngeo_flux = 0 no geothermal heat flux
!           = 1 constant geothermal heat flux
!           = 2 variable geothermal heat flux (read in geothermal_heating.nc)
!           ( C A U T I O N : flux in mW/m2 in the NetCDF file )
! ngeo_flux_const  Constant value of geothermal heat flux (W/m2)
&nambbc
  ngeo_flux = 0
  ngeo_flux_const = 86.4e-3
/
!-----
!   namqsr  penetrative solar radiation
!-----
! ln_traqsr : penetrative solar radiation (T) or not (F)   (Default=T)
! rabs      fraction of qsr associated with xsi1
! xsi1      first depth of extinction
! xsi2      second depth of extinction
&namqsr
  ln_traqsr = .true.
  rabs      = 0.58
  xsi1      = 0.35
  xsi2      = 23.0
/
!-----
!   namtdp  tracer newtonian damping ('key_tradmp')
!-----

```

```

! ndmp  type of damping in temperature and salinity
!      ('latitude', damping poleward of 'ndmp' degrees and function
!      of the distance-to-coast. Red and Med Seas as ndmp=-1)
!      (= -1 damping only in Med and Red Seas)
! ndmpf  =1 create a damping.coeff NetCDF file (the 3D damping array)
! nmldmp type of damping in the mixed layer
!      (=0 damping throughout the water column)
!      (=1 no damping in the mixed layer defined by avt >5cm2/s )
!      (=2 no damping in the mixed layer defined rho<rho(surf)+.01 )
! sdmp  surface time scale for internal damping (days)
! bdmp  bottom time scale for internal damping (days)
! hdmp  depth of transition between sdmp and bdmp (meters)
&namtdp
  ndmp  = -1
  ndmpf = 0
  nmldmp = 1
  sdmp  = 50.
  bdmp  = 360.
  hdmp  = 800.
/

!-----
!  nameos  ocean physical parameters
!-----
! neos  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 )
!      with rau0=1020 set in parsst routine
! ralpha thermal expansion coefficient (linear equation of state)
! rbeta  saline expansion coefficient (linear equation of state)
&nameos
  neos  = 0
  ralpha = 2.e-4
  rbeta  = 0.001
/

!-----
!  namsol  elliptic solver / island / free surface
!-----
! nsolv  elliptic solver (=1 preconditioned conjugate gradient: pcg)
!      (=2 successive-over-relaxation: sor)
!      (=3 FETI: fet, all require "key_feti" defined)
!      (=4 sor with extra outer halo)
! nsol_arp absolute/relative (0/1) precision convergence test
! nmin  minimum of iterations for the SOR solver
! nmax  maximum of iterations for the SOR solver
! nmod  frequency of test for the SOR solver
! eps  absolute precision of the solver

```

```

! resmax  absolute precision for the SOR solver
! sor     optimal coefficient for SOR solver
! epsisl  absolute precision on stream function solver
! nmisl   maximum pcg iterations for island
! rnu     strength of the additional force used in free surface b.c.
&namsol
  nsolv   = 2
  nsol_arp = 0
  nmin    = 300
  nmax    = 800
  nmod    = 5
  eps     = 1.E-6
  resmax  = 1.E-11
  sor     = 1.964
  epsisl  = 1.e-10
  nmisl   = 4000
  rnu     = 1.
/
!-----
!   namspr surface pressure diagnostic
!-----
! nmaxp   maximum of iterations for the solver
! epsp    absolute precision of the solver
! niterp  number of iteration done by the solver
&namspr
  nmaxp   = 1000
  epsp    = 1.e-3
  niterp  = 400
/
!-----
!   namcore CORE
!-----
!
! In this version there are 8 files ( jpfle = 8)
! THE ORDER OF THE FILES MATTER:
! 1 - precipitation total (rain+snow) in kg/m2/s
! 2,3 - u10,v10 -> scalar wind at 10m in m/s - ON 'T' GRID POINTS!!!
! 4 - specific humidity          in %
! 5 - solar radiation (short wave) in W/m2
! 6 - thermal radiation (long wave) in W/m2
! 7 - temperature at 10m         in degrees K
! 8 - precipitation (snow only)   in kg/m2/s
!
! cname    file names (256 char max for each)
! clvarname name of variable in netcdf file (32 char max)
! freqh    frequency of fields in the file
!          it is in hours (6 hourly, daily) if positive.
!          if freqh = -12 the file contains 12 monthly data.

```

```
&namcore
  ln_2m      = .TRUE.
  alpha_precip = 1.
  clname    = 'precip.nc' 'u10.nc' 'q2.nc' 'v10.nc' 'radsw.nc' 'radlw.nc' 't2.nc' 'snow.nc'
  clvarname = 'precip' 'u10' 'q2' 'v10' 'radsw' 'radlw' 't2' 'snow'
  freqh     = -12      24      24      24      24      24      24      -12

/
```