



**Issue 2009**  
November 2009

*ANS – Numerical  
Applications and  
Scenarios*

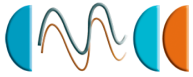
# IMPLEMENTATION OF NEMO-OPA IN CONFIGURATION ORCA-R025

*By* **Giuseppe Grieco**  
Centro Euro-Mediterraneo per i  
Cambiamenti Climatici (CMCC)  
*giuseppe.grieco@cmcc.it*

**Simona Masina**  
Centro Euro-Mediterraneo per i  
Cambiamenti Climatici (CMCC)  
Istituto Nazionale di Geofisica e  
Vulcanologia (INGV)  
*masina@bo.ingv.it*

**SUMMARY** This document describes the NEMO-OPA (Nucleus for European Modelling of the Ocean, Ocèan PARallèlisè) Ocean General Circulation Model (OGCM) in the configuration ORCA-R025 implemented at CMCC. In the first part it gives a description of the most important technical aspects of the model, the physical parameterization adopted and the forcing used.

In the second part, the results of the benchmarks on the vector and scalar systems of the CMCC Computer Center are presented and compared.

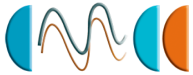


# 02

Centro Euro-Mediterraneo per i Cambiamenti Climatici

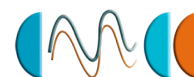
# Contents

Introduction . . . . .	4
Basic Settings . . . . .	4
Horizontal grid . . . . .	5
Vertical grid . . . . .	5
Bathymetry . . . . .	5
Initial conditions . . . . .	6
Forcings . . . . .	6
Run-offs . . . . .	6
ENACT Dataset . . . . .	6
Sea surface height adjustment . . . . .	6
Surface relaxation . . . . .	6
Parametrizations . . . . .	7
Lateral mixing . . . . .	7
Vertical mixing . . . . .	7
Bottom friction and lateral boundary condition . . . . .	8
Numerical schemes . . . . .	9
Advection schemes . . . . .	9
Time discretization . . . . .	9
Model output . . . . .	10
Archived fields . . . . .	10
Storage . . . . .	11
Code Structure . . . . .	11
Compiling environment . . . . .	11



# 04

Running environment . . . . .	11
Numerics . . . . .	12
Domain Decomposition . . . . .	12
Benchmark on vector system . . . . .	12
Benchmark on scalar system . . . . .	15
Auxiliary files . . . . .	16
Initial and boundary conditions . . . . .	16
Restart files . . . . .	16
Output files . . . . .	16
<b>Bibliography</b> . . . . .	<b>17</b>
Bibliography . . . . .	17



## INTRODUCTION

This report describes the implementation of ORCA-R025 at the Numerical Applications and Scenarios Division of CMCC. The ORCA-R025 implemented at CMCC is a global  $1/4^\circ$  configuration of NEMO-OPA ocean circulation model (Madec, 2008) [6] and the code is based on version NEMO 2.3. The model has been implemented on three different supercomputers available at CMCC:

- A vector system of 4 nodes for a total of 30 NEC SX-8R processors (3 nodes equipped with 8 CPUs and 1 node with 6 CPUs) located at CIRA in Capua with a peak performance of 1 TFlops.
- A vector system of 112 NEC SX-9 processors distributed on 7 nodes with a peak performance of 11,47 TFlops situated inside the Campus "Ecotekne" in Lecce.
- A scalar system of 30 nodes for a total of 960 IBM Power 6 CPUs with a peak performance of 18 TFlops situated inside the Campus "Ecotekne" in Lecce.

A first benchmark of the code has been carried out on the cluster of NEC SX-8R in order to find the best choice of the domain decomposition. Afterward, the code has been imported on the cluster of NEC SX-9 and the benchmark of the code has been done again by varying the number of CPUs while keeping the same best domain decomposition found on the vector system located in Capua.

An additional benchmark has been carried out to assess the performance of the code on the cluster of IBM Power 6 processors.

All the simulations have been done without the ice model and with the same forcing dataset. It consists of the climatological year derived by the ENACT dataset (Bellucci et al., 2007)

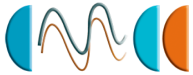
[2] More details will be given in section *ENACT Dataset*.

## BASIC SETTINGS

The list of the CPP keys used for the simulations in configuration ORCA-R025 is found in table 1

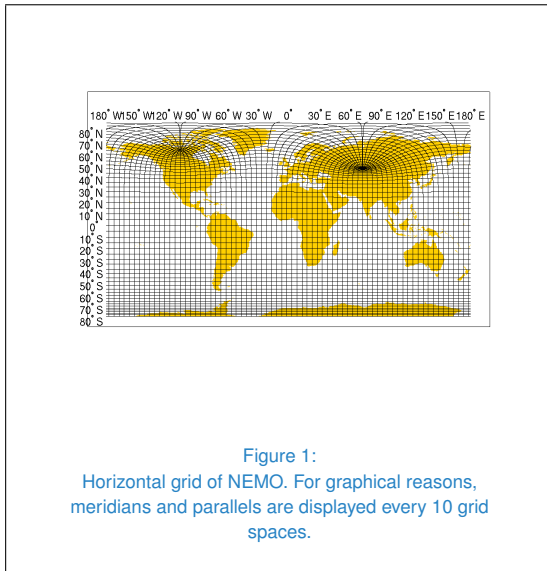
**Table 1**  
CPP keys for ORCA-R025

ORCA-R025 key-	Description
orca_r025	Configuration name
dynspgflt	Filtered free surface
zdfcke	TKE vertical mixing scheme
zdfddm	Double diffusive mixing
zrefsurf	Increment of the vertical background eddy viscosity for dynamic and vertical background eddy diffusivity for tracers near the surface (former 10 layers) and uses a different vertical layer distribution
dtatem	Use temperature data for initial conditions
dtasal	Use salinity data for initial conditions
dtasst	Use sea surface temperature data for damping
traldfc2d	2D lateral diffusion for tracers (depends on $\delta x$ )
limfdd	Sea/Ice interaction
dynldfc3d	3D lateral diffusion for dynamics (depends on $\delta x^3$ )
ldfslp	Need to calculate isopycnal slope
trabbl_dif	Diffusive bottom boundary layer parameterization
trabbc	Tracer bottom boundary condition
partial_steps	Partial cell topography
tau_daily	Establishes that the stress is updated every day
flx_forced_daily	Establishes that the fluxes are updated every day
diagap	Model data gap diagnostic
diahth	Thermocline diagnostic
dynhpa	Hydrostatic pressure anomaly diagnostic
mpp_mpi	Parallelization through MPI
vectopt_loop	Computational optimization on vector machines
vectopt_memory	Memory optimization for vector machines



### HORIZONTAL GRID

The horizontal grid (Fig. 1) has 3 poles (2 poles in the northern hemisphere, one over Canada and the other over Siberia). The horizontal grid resolution is 1/4 deg at the equator for both latitude and longitude, leading to horizontal dimensions of 1442 X 1021. The minima of the scale factors (ocean points) are around 5.6 Km (zonal direction) and 3.2 Km (meridional direction). The maximum is 27.8 Km at the equator.



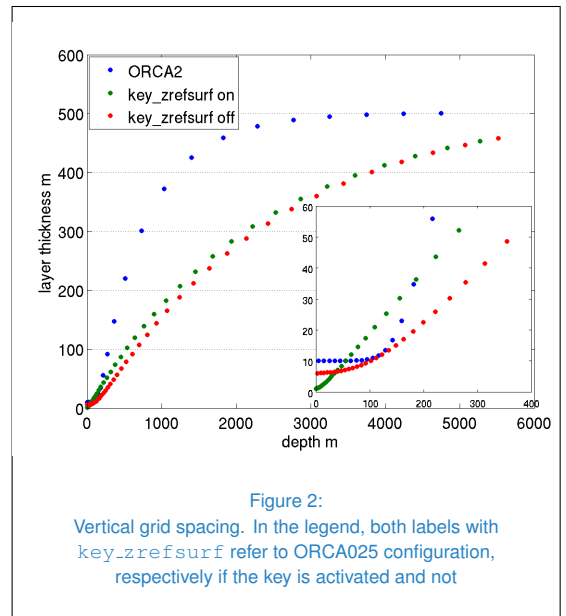
In Fig. 1, the closed sea are not present because we are running without them. This option may be set in the namelist as follows:

```
!-----
!...
! nclosea = 0 no closed sea
!           = 1 closed sea (Black
Sea, Caspian Sea, Great US
Lakes...) 5760
!...
nclosea = 0
```

### VERTICAL GRID

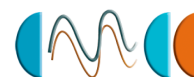
There are 50 levels on the vertical, with a grid spacing depending on the `key_zrefsurf` CPP key. If the key is used, the grid spacing ranges from 1 m near the surface to 450 m at 5750 m. In the other case, the grid spacing ranges from 6 m near the surface to 450 m at 6000 m. Figure 2 compares the vertical distribution of the layers between the two configuration of ORCA-R025 (with and without `key_zrefsurf`) and a version of the OPA 8.2 in configuration ORCA2 also used at CMCC (Di Pietro et al.) [3] (Scocimarro et al., 2007) [10].

It is important to stress here that the vertical spacing of ORCA2 configuration is more resolved than ORCA-R025 at depths between 50 m and 200 m when the `key_zrefsurf` is used.



### BATHYMETRY

The bathymetry file has been constructed and provided by MERCATOR OCEAN. It is derived from two input files:



- The 2-minute bathymetry file (etopo2) of NGDC, combination of (Smith & Sandwell, 1997) [11] satellite-based bathymetry (8.2 version), IBCAO (in the Arctic region) and other data in the Antarctic region. The MERCATOR project team has applied additional corrections near the Antarctic (Remy et al., 2003) [9]. This file is used for the deep ocean (below 300m).
- The GEBCO 1mn bathymetry file provided between 88S and 88N. This file is used for bathymetry on the shelf because it is found to be more accurate there (above 200m).

## INITIAL CONDITIONS

The initial conditions for temperature and salinity were derived from the Levitus98 (Levitus S. et al, 1998) [5] data set. The zonal and the meridional components of the velocity field start from zero value.

## FORCINGS

### RUN-OFFS

The runoff file has been provided by MERCATOR-OCEAN. It includes 99 major rivers and coastal runoffs. It is a monthly climatology.

The namelist variable `nrunoff=2` has been set. In that case, a special treatment is applied in runoff regions, where the array `upsrnfh` is non zero. This array is read in the runoff file (netcdf variable `socoefr`). It has a value of zero where there is no runoff and 0.5 at runoff points. Since the centered scheme has not been used, there is no need to revert to upstream near runoffs (see Section *Numerical schemes*). The special treatment consists in an enhanced vertical diffusion at the interface of layers 1 and 2 applied in `step.F90`:

```
avt (:,:,2) = avt (:,:,2) + 2.e-3
* upsrnfh (:, :)
```

The vertical mixing is then set to  $10^{-3}$  below the top level at runoff points.

`upsrnfh` is also used to avoid SSS restoring in the vicinity of river mouths.

## ENACT DATASET

The wind stress, solar heat flux, absorbed solar radiation and evaporation minus precipitation fields come from the ENACT data set. The available fields are all daily mean values and therefore they cannot represent the diurnal cycle. They are derived from 0-24 hour averages from the ERA-40 (Uppala et al., 2005) [13] reanalyses. The main difference between the ERA-40 data set and the ENACT data set concerns the precipitation field which has been corrected for a bias (Troccoli & Källberg, 2004) [12].

The model has been forced with a climatological year (1958-2001) with daily values.

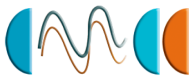
The original ENACT data set at the resolution of T106, i.e., the full resolution of ERA-40 has been interpolated onto the ORCA-R025 horizontal grid and rotated to take into account its distortion north of  $40^{\circ}$  N. The OASIS3 package (Valcke, 2006) [14] with a simple bilinear interpolation has been used.

## SEA SURFACE HEIGHT ADJUSTMENT

A SSH adjustment has been used in order to take into account the fact that the fresh water budget, i.e., E-P-R is not close and can be the cause of a spurious drift of the SSH. `ln_fwb = .true.` has been set in the namelist. The code has been modified to allow a daily adjustment instead of a yearly one.

## SURFACE RELAXATION

SST has been relaxed to the ERA-40 SST with a restoring time of 12 days corresponding to a



damping term of  $200 \text{ W/m}^2/\text{C}$  in the heat flux. No relaxation has been applied either to SSS or T and S on the whole water column.

### PARAMETRIZATIONS

All runs are performed with the free surface, constant volume formulation (`key_dynspgflt` defined).

### LATERAL MIXING

A laplacian isopycnal diffusion scheme is used for tracers with  $ah_{t0}=300\text{m}^2/\text{s}$  at the equator. The coefficient decreases poleward proportionally to the grid size. Associated keys are `traldf_c2d` and `ldfslp`.

A biharmonic horizontal viscosity is used for dynamics with  $ah_{m0}=-1.5 \cdot 10^{11} \text{ m}^4/\text{s}$ . The viscosity depends on the grid size at the cubic power according to the equation 1

$$ahm = ahm0 * \frac{\delta x^3}{\delta x_{max}^3} \quad (1)$$

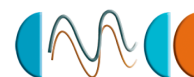
### VERTICAL MIXING

There is double-diffusive mixing `zdfddm`. Turbulent Kinetic Energy scheme is used to compute the vertical mixing `zdf_tke` with the following namelist parameters

```
!-----
! namtke turbulent eddy kinetic
! dependent vertical diffusion
! ( #ifdef "key_zdf_tke" )
!-----
! 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
! (m2/s2)
! emin0 surface minimum value of
! tke (m2/s2)
! 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=pd1*avm)
! = 2 same as = 1 but a shapiro
! filter is applied on pdl
! nave = horizontal averaged
! (=1/2) or not (=0) of avt (default
! =1)
!&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 = 2
npdl = 1
navb = 0
```





```
nave = 1
```

The enhanced vertical diffusion is used when convection is diagnosed. The corresponding namelist block is:

```
!-----
! 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)
! nevdn = 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-4
avt0 = 1.e-5
avevd = 10.
nevdn = 1
ln_zdfexp = .false.
n_zdfexp = 3
```

## BOTTOM FRICTION AND LATERAL BOUNDARY CONDITION

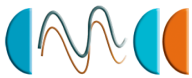
The namelist for the bottom friction is the following.

```
!-----
! 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 (m2/s2)
! ln_bfr2d: flag for 2d coef
enhancement read in file
! bfrien: enhancement
coefficient (integer > 1)
! bfrien = 30
! ln_bfr2d = .TRUE.
&nambfr
nbotfr = 2
bfri1 = 4.e-4
bfri2 = 1.e-3
bfeb2 = 2.5e-3
```

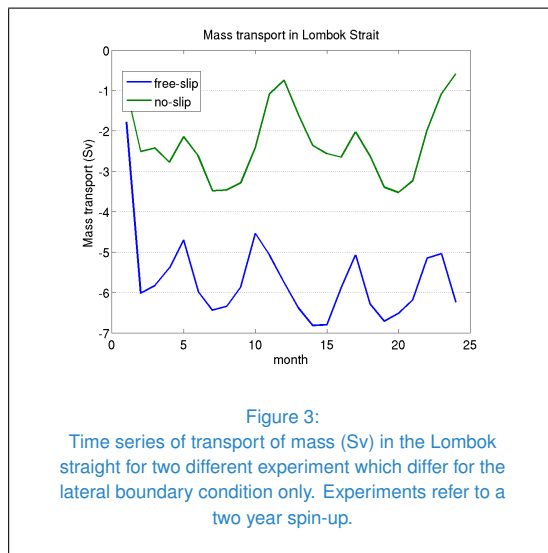
A diffusive bottom boundary layer is activated using `trabbl_dif` with `atrbbl=3000` m<sup>2</sup>/s.

The lateral boundary conditions can be set to *no-slip*, *free-slip*, *partial free-slip* or *strong no-slip*. The namelist parameter to be set is `shlat`. For more details, please refer to (Madec, 2008) [6].

The free slip condition has been previously



used but it gives strong currents in the Indonesian Through-Flow (ITF) straits. Therefore we decided to set the no slip lateral boundary condition. Fig 3 shows the transport of mass in the Lombok strait for two experiments which differ just for the lateral boundary conditions. The experiment with no-slip lateral condition is much more consistent with experimental observed data. In fact, the average value is 2.3 Sv which is close to the value of 1.7 Sv stated by (Murray & Arief., 1988) [7] while the free-slip condition experiment gives an average of 6 Sv.



In the Technical report (Dussin et al., 2009) [8] they use the free slip condition because in a previous simulation with the no slip condition they were not satisfied about the results. However, they indicate that with the free slip condition the transports in the ITF are too strong so they modified the condition locally in order to decrease them. At the light of our initial results and the results reported in Dussin et al., (2009) a final decision on the best lateral conditions to be used with the present version of ORCA-R025 is therefore still under evaluation.

## NUMERICAL SCHEMES

### ADVECTION SCHEMES

The Total Variance Dissipation (TVD) scheme for tracers is used (`ln_traadv_tvd=.true.`) together with the energy enstrophy advection scheme suitable for partial step topography (`ln_dynvor_eeen=.true.`).

The other possible advection schemes for tracers are the 2<sup>nd</sup> order centred scheme (`ln_traadv_cen2=.true.`), the Monotone Upstream Scheme for Conservative Laws (MUSCL)

(`ln_traadv_muscl=.true.`)

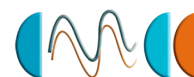
and a second MUSCL scheme (`ln_traadv_muscl2=.true.`).

The other possible advection scheme for dynamics are the enstrophy conserving scheme (`ln_dynvor_ens=.true.`), the energy conserving scheme (`ln_dynvor_ene=.true.`) and the mixed scheme (`ln_dynvor_mix=.true.`).

The reader interested in more details can refer to the manual of NEMO (Madec 2008) [6].

### TIME DISCRETIZATION

The model uses a Leapfrog scheme for non-diffusive processes with an Asselin (Asselin, 1972) [1] time filter with the Asselin coefficient set to 0.1. For the non-diffusive part, two different schemes are available: a) a forward time differencing scheme using a time splitting technique (`ln_zdfexp=.true.` in the namelist) or b) a backward time differencing scheme (`ln_zdfexp=.false.`). Here, we use the second option. The time step is set to 1080 s (80 steps/day).

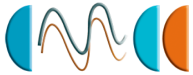


## MODEL OUTPUT

### ARCHIVED FIELDS

Files are stored every 5 days (`nwrite = 300 steps`) with 5 days averaged fields. 3D variables are

- `vozocrtx` Zonal velocity (in `grid.U.nc`)
  - `vomecrtx` Meridional velocity (in `grid.V.nc`)
  - `votemper` Temperature (in `grid.T.nc`)
  - `vosaline` Salinity (in `grid.T.nc`)
  - `vovecrtz` Vertical velocity (in `grid.W.nc`)
  - `votkeavm` Vertical eddy viscosity (in `grid.W.nc`)
  - `votkeavt` Vertical eddy diffusivity (in `grid.W.nc`)
  - `votkeevm` Enhanced vertical viscosity (in `grid.W.nc`)
  - `votkeevd` Enhanced vertical diffusivity (in `grid.W.nc`)
  - `voddmavs` Salt vertical eddy diffusivity (in `grid.W.nc`)
- Two dimensional fields are `sozotaux` (zonal wind stress), `sozotauy` (meridional wind stress) and `soleahtw` (lateral eddy diffusivity) (in `grid.U`, `grid.V` and `grid.W` files respectively), and in `grid.T` files:
- `sossh eig` Sea surface height (meters)
  - `somxl010` Mixed layer depth (meters) based on a  $\sigma_0$  difference of 0.01 with the surface
  - `sohefldo` Net downward heat flux ( $\text{Wm}^{-2}$ )
  - `soshfldo` Short wave downward radiative flux ( $\text{Wm}^{-2}$ )
  - `sowafgup` Net upward water flux ( $\text{Kg m}^{-2}\text{s}^{-1}$ )
  - `sostsst` Sea surface temperature (K)
  - `so saline` Sea surface salinity (PSU)
  - `so runoff` Runoffs ( $\text{Kg m}^2\text{s}^{-1}$ )
  - `sowafgcd` Concentration/Dilution water flux ( $\text{Kg m}^{-2}\text{s}^{-1}$ )
  - `so salflx` Surface salt flux ( $\text{Kg m}^{-2}\text{s}^{-1}$ )
  - `so mixhgt` Turbocline depth (meters)
  - `so icecov` Ice cover [0,1]
  - `sohefldp` Surface heat flux damping ( $\text{Wm}^{-2}$ )
  - `sowafldp` Surface water flux damping ( $\text{Kg m}^{-2}\text{s}^{-1}$ )
  - `so salfldp` Surface salt flux damping ( $\text{Kg m}^{-2}\text{s}^{-1}$ )
  - `so bowl in` Bowl index
  - `so thedep` Thermocline depth (m)
  - `so20chgt` Depth of 20C isotherm (m)
  - `so28chgt` Depth of 28C isotherm (m)
  - `sohtc300` Heat content 300 m (W)
  - `so hpa` Hydrostatic pressure anomaly (m)



## STORAGE

The output is written by each processor in direct access files. All these partial files are merged by means of the `flio_rblld` tool provided by IOIPSL. The outputs are netcdf files written for each month and the filenames are of kind `ORCA025_spinup_5d_yyyyymm01-yyyymmld_gridX.nc`, where `5d` says that the output is stored every 5 days, `yyyy` is the year, `mm` is the month, `ld` is the last day of the month and `X` stands for T, U, V or W. All files are stored in `ans007@ulysses.cmcc.it`.

## CODE STRUCTURE

### COMPILING ENVIRONMENT

The NEMO code is located under the directory `../nemo025`. Table 2 shows the directory tree of NEMO.

The code is designed to run on the NEC-SX8R/9 vector supercomputer and on massive parallel machines.

In order to compile the code, the following prerequisite are needed:

- the MPI library
- the NetCDF library
- the GNU make

According to the NEC supercomputer, the compilation of the code can be done using the NEC cross-compiler on a front-end machine.

### RUNNING ENVIRONMENT

The running environment is divided in two sections:

- experiment management (`../nemo025/config/ORCA025_LIM/EXPID`)
- experiment directory tree (`../EXPID`)

Table 2  
Directory tree of NEMO

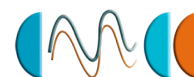
```
nemo025
|-- bin
|-- config
|   |-- ORCA025_LIM
|   |   |-- EXPID
|-- doc
|-- lib
|-- modeles
|   |-- AGRIF
|   |-- IOIPSL
|   |-- NEMO
|   |-- UTIL
|-- scripts
|-- tmp
-- util
```

where `EXPID` is the name of the generic experiment.

Note that the experiment management directory tree `../nemo025/config/ORCA025_LIM/EXPID` is located in the `../nemo025` main directory and that the `EXPID` subdirectories need to be created for every new experiment. The `../EXPID` directory is usually placed on a fast filesystem.

The shell script `JOB_recur`, located in the `../nemo025/config/ORCA025_LIM/EXPID` directory, is the one used to run the experiment. The script needs to have specified the environment variables of the experiment to be run.

A chain submission procedure is applied to avoid too long jobs on the computing platform. In our current configuration, the model is designed to run 1 month of simulation for each job submission. The information regarding the month to integrate after the first submission is stored in a text file named `time.txt` which is generated by the `JOB_recur` script during the



first submission. The `time.txt` file contains year and month in execution.

The `JOB_recur` script is submitted to the NEC-SX NQS (IBM lsf) batch system through the `qsub (bsub)` command and executes the following steps:

1. Set the environment variable related to:
  - experiment name
  - paths of the directory tree
  - number of processes to be used
2. Modify the namelist according to the job to run
3. Get the executables, the namelist and create the links to the forcings and the restart files in the scratch directory where the job will run
4. Launch the job
5. Save the output and the restart files in the storage directory
6. Update the `time.txt` file
7. Resubmit itself to perform the next month (until the end of the experiment)

## NUMERICS

### DOMAIN DECOMPOSITION

A different domain decomposition has been chosen for the two available supercomputer architectures. The criterion that drives this choice is a compromise between the efficiency of the code and the available computing resources. The available computing resources refer not only to the potential of the machine but also to the requests of all the users which access the supercomputer. Our best choice is therefore a domain decomposition of 8 zonal domain for

the vector system while for the scalar system there is not any particular choice because as it will be shown in Section *Benchmark on scalar system*, the efficiency of the code is almost the same for a number of CPUs greater than 25.

Section *Benchmark on vector system* describes the benchmarks done on the cluster of SX-8R and on the cluster of SX-9.

Section *Benchmark on scalar system* describes the benchmark on the scalar system. In this case we did not assess the most performant domain decomposition, given a number of available CPUs. This kind of analysis would have requested an exaggerate CPU time with respect to the benefits that could have been derived.

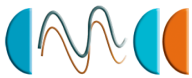
## BENCHMARK ON VECTOR SYSTEM

### BRIEF DESCRIPTION OF THE VECTOR SYSTEM

Two different benchmarks of the code have been carried out, respectively on the cluster of NEC SX-8R and on the cluster of NEC SX-9. The vector system located at CIRA consists of 4 nodes for a total of 30 NEC SX-8R vector CPUs (3 nodes equipped with 8 CPUs and 1 node with 6 CPUs) and a peak performance of 1 TFlop. Here we assessed the most convenient geometry for the domain decomposition and afterward we made another benchmark on the cluster of SX-9 just varying the number of CPUs while keeping the most convenient geometry. The vector system located at Ecotekne consists of 112 NEC SX-9 vector CPUs for a total peak performance of 11.47 TFlop.

### DESCRIPTION OF THE SIMULATION

The simulations done on the cluster of SX-8R and on the cluster of SX-9 in order to assess the performance of the code have a duration of one month. The forcing used and the physical parametrizations are identical to what has been



## 14

largely described in the section *Forcings*.

## PERFORMANCE

A first analysis of the performance has been conducted on the cluster NEC SX8-R. In order to do that, a series of runs have been made with a different number of CPUs ranging from 4 to 16 by evaluating the efficiency of the code. The domain decomposition used in these runs is of the kind  $2 \times Y$  where  $2$  is the number of domain along the zonal direction and  $Y$  is along the meridional direction. For example, a domain decomposition  $2 \times 8$  means 2 meridional domains and 8 zonal domains.

Figures 4, 5 and 6 show respectively the user and the elapsed time for all the simulations on the cluster of SX-8R, the speed-up with respect to the simulation with domain decomposition  $2 \times 2$  and the efficiency with respect to the same domain decomposition. The first thing to note is that the code loses lots of efficiency when using 8 CPU. It seems that this domain decomposition is particularly disadvantageous. We investigated on the possible causes of this result and we found that the code produces a high bank conflict time when employing 8 CPUs. This time doubles with respect to the other simulations, but it does not depend on the particular domain decomposition. In fact, figure 7 shows that elapsed and user time of the simulations performed with all possible domain decompositions with 8 CPUs do not vary in a significant way. Anyway, this trend is not found on the cluster NEC SX9 (figure 10). More experiments should be done to investigate on the possible causes.

All the possible combinations of domain decomposition with 8 CPUs were tested and as figure 7 shows, the simulations with domain decomposition  $1 \times 8$  is most performant if we consider the elapsed time while is equivalent to domain decompositions  $2 \times 4$  and  $4 \times 2$  if we consider

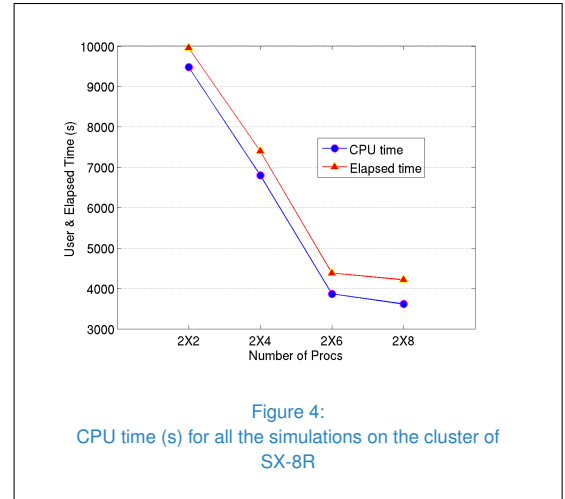


Figure 4:  
CPU time (s) for all the simulations on the cluster of SX-8R

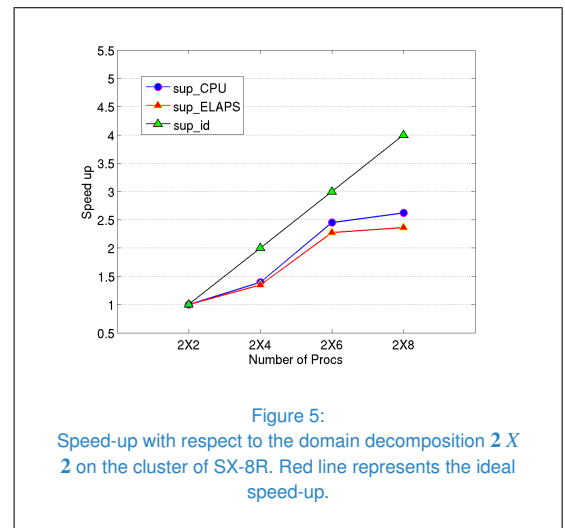


Figure 5:  
Speed-up with respect to the domain decomposition  $2 \times 2$  on the cluster of SX-8R. Red line represents the ideal speed-up.

the user time.

Figures 8, 9 and 10 show respectively the elapsed and the user time for all the simulations, the speed-up with respect to the simulation with domain decomposition  $1 \times 4$  and the efficiency with respect to the same domain decomposition on the cluster of NEC SX9.

Figures 9 and 10 show that the code loses lots of efficiency when using more than 8 CPUs. In particular, the efficiency (with respect to 4 CPUs) drops to 70% (if we consider the user time and 50% if we consider the elapsed time) when employing 16 CPUs while it is 90% for 8

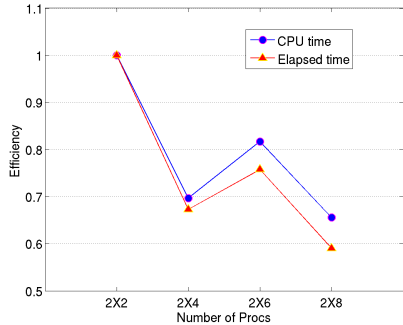
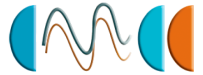


Figure 6:  
Efficiency of the code. One is the efficiency for the domain decomposition 2 X 2 on the cluster of SX-8R.

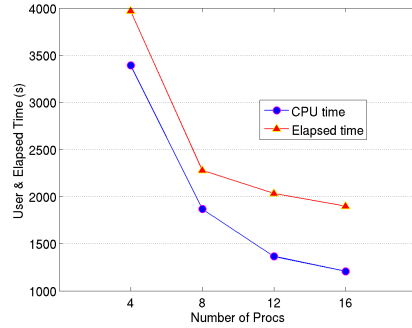


Figure 8:  
Elapsed and CPU time (s) for all the simulations on the cluster of SX-9

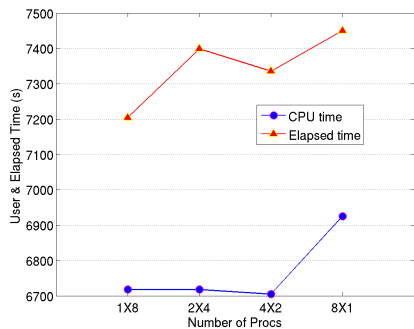


Figure 7:  
Elapsed time (s) for all the possible combination of domain decomposition with 8 CPUs.

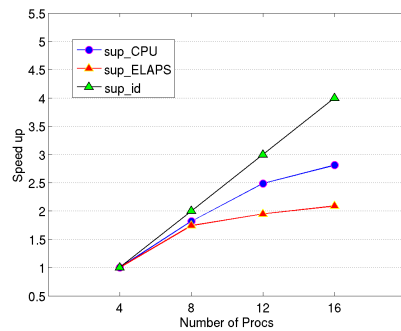


Figure 9:  
Speed-up with respect to the domain decomposition 1 X 4 on the cluster of SX-9. Green line represents the ideal speed-up.

CPUs. Therefore, 8 CPUs seems to be the best compromise between the efficiency of the code and the availability of the computing resources. The last consideration is referred to the NEC SX-8R and SX-9 CPUs. The peak performance of one CPU is 35.2 GF and 102.4 GF respectively, so the speed-up should be around 2.9, when moving from SX-8R to SX-9. Table 3 shows the elapsed time and the user time for the simulation with domain decomposition 1 X 4 on both clusters, and the speed-up between of them.

Table 3 shows that the real speed-up is similar

**Table 3**  
Performance comparison between the clusters SX-8R and SX-9 for the domain decomposition 1 X 4

	elapsed (s)	user (s)
<b>SX-8R</b>	9761.1	9324
<b>SX-9</b>	3845.9	3220.6
<b>speed up</b>	2.53	2.89

to what expected.

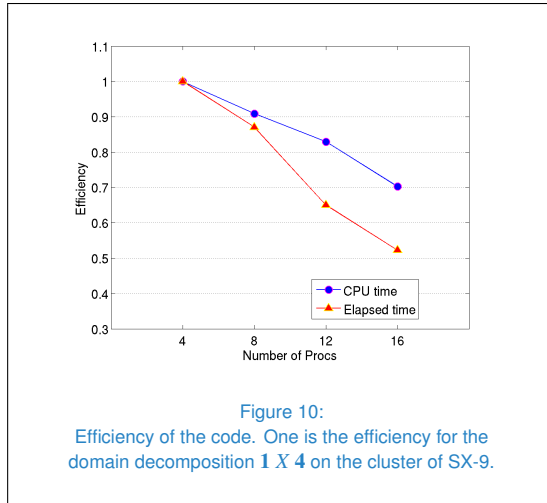
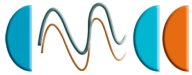


Figure 10: Efficiency of the code. One is the efficiency for the domain decomposition 1 X 4 on the cluster of SX-9.

### BENCHMARK ON SCALAR SYSTEM

#### BRIEF DESCRIPTION OF THE SCALAR SYSTEM

The scalar system on which the benchmark has been carried out is located at Campus "Ecotekne" in Lecce. It consists of 960 IBM Power 6 CPUs distributed on 30 multicore nodes, for a total peak performance of 18 TFlops.

#### DESCRIPTION OF THE SIMULATION

The simulation done to assess the performance of the code has a duration of one month. The forcing used for this simulation and the physical parametrizations are identical to what has been largely described in the section *Forcings*.

#### PERFORMANCE

A different strategy for the domain decomposition has been adopted for the benchmark on the scalar system with respect to the vector system case. In the other case, we choose to evaluate the performance for a domain decomposition of the kind 2 X Y where 2 is the number of domain along the zonal direction and Y is along the

meridional direction. In this case we adopted a domain decomposition of the kind N X N where N varies between 2 and 13. It could be simpler to assess the performance for a domain decomposition for a multiple of 4 or 16 or 32 and have a linear axis for the number of CPUs employed. However, we thought that two different performances are better comparable if the geometry of their domain decomposition is similar. In the scalar system case we had much more available CPUs with respect to the vector system case and this was possible. Figures 11, 12 and 13 show respectively the elapsed and the user time for all the simulations, the speed-up with respect to the simulation with domain decomposition 2 X 2 and the efficiency with respect to the same domain decomposition.

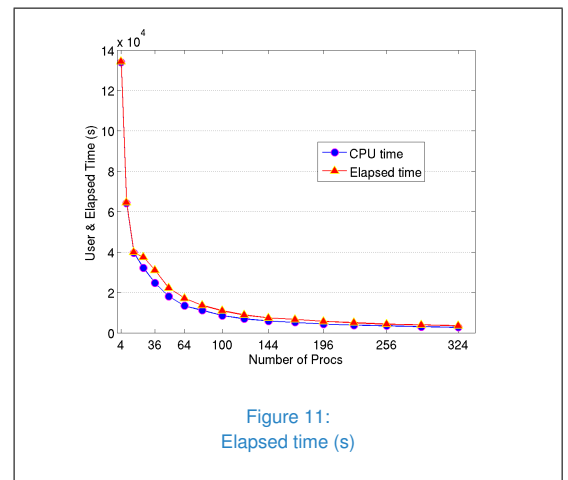


Figure 11: Elapsed time (s)

Figures 12 and 13 show that the code loses dramatically efficiency when using more than 16 CPUs and stabilizes on 50% if we consider the elapsed time or the 60% if we consider the user time. Running with less than 64 CPUs is not workable because this would imply more than 5 hours (the time required when 64 CPUs are used) per month of simulation, so, we have to consider that, for practical use, the code has an efficiency of about 50%.

As said before, we did not investigate the best



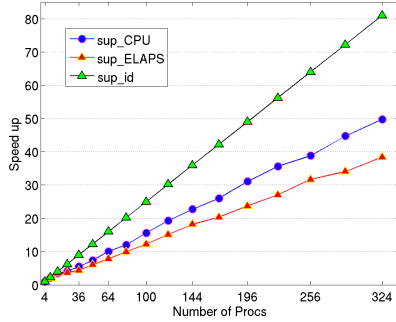
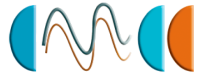


Figure 12:  
Speed-up with respect to the domain decomposition 2 X 2. Green line represents the ideal speed-up.

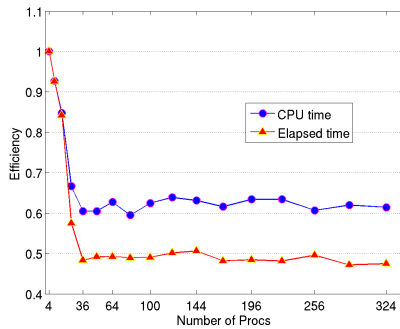


Figure 13:  
Efficiency of the code. One is the efficiency for the domain decomposition 2 X 2.

geometry for the domain decomposition. This analysis would make sense for a large number of CPUs which is plausible for a realistic operational use. On the other hand, this would request a strong effort in terms of CPU time considering the large number of combination of different domain decomposition for a number of order 100.

## AUXILIARY FILES

## INITIAL AND BOUNDARY CONDITIONS

### Input files:

coordinates_ORCA025_LIM.nc	Grid coordinate file
bathy_ORCA025_LIM.nc	Ocean bathymetry file
EMPave_old.dat	Initial Average EMP
Levitus98_PHC21_Sal_ORCA_R025.nc (Levitus98_Sal_nodirect.nc)	Salinity climatology from Levitus
Levitus98_PHC21_Tem_ORCA_R025.nc (Levitus98_Tem_nodirect.nc)	Potential Temperature climatology from Levitus
runoff_ORCA025_LIM.nc	Climatological run-off
Gr_SST_clim_58_01.nc	ERA-40 SST

## RESTART FILES

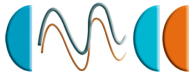
### Restart file:

ORCA025_EXPID_YYYYMMDD_restart.nc	Restart file
-----------------------------------	--------------

## OUTPUT FILES

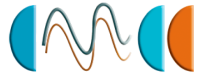
### Output files: ORCA025\_EXPID\_5d\_YYYYMMID\_YYYYMMFD\_

grid.T	Output frequency: monthly; Format: NetCDF
grid.U.nc	Output frequency: monthly; Format: NetCDF
grid.V.nc	Output frequency: monthly; Format: NetCDF
grid.W.nc	Output frequency: monthly; Format: NetCDF



## Bibliography

- [1] Asselin R., 1972: Frequency filter for time integrations. *Mon. Wea. Rev.*, **118** (10), 1960-1969.
- [2] Bellucci A., Masina S., Di Pietro P., Navarra A., 2007: Using Temperature Salinity Relations in a Global Ocean Implementation of a Multivariate Data Assimilation Scheme. *Mon. Wea. Rev.*, **135**, 3785-3807.
- [3] Di Pietro P., Masina S., 2009: The CMCC-INGV Global Ocean Data Assimilation System (CIGODAS). CMCC Research Papers, RP0071.
- [4] Gent P.R., McWilliams J.C., 1990: Isopycnal Mixing in Ocean Circulation Models. *J. Phys. Oceanogr.*, **20**, 150-155.
- [5] Levitus, S., and Coauthors, 1998: World Ocean Database 1998. NOAA Atlas NESDIS 1, 346 pp.
- [6] Madec, G., 2008: NEMO ocean engine, Note du Pole de modelisation. Institut Pierre-Simone Laplace (IPSL), France, No 27, ISSN 1288-1619.
- [7] Murray S.P. & Arief D., 1988: Through-flow into the Indian Ocean through the Lombok Strait, January 1985-January 1986. *Nature*, **333**, 444 - 447, doi:10.1038/333444a0.
- [8] Dussin R., Treguier A.M., Molines J.M., Barnier B., Penduff T., Brodeau L., Madec G., 2009: Definition of the interannual experiment ORCA025-B83, 1958-2007. LPO Report 09-02, January 2009.
- [9] Remy E. et al., 2003: Construction de la bathymetrie pour la configuration model ORCA025. Rapport CERFACS May 2003.
- [10] Scoccimarro E., Gualdi S., Fogli P. G., Manzini E., Grezio A., Navarra A., 2007: A Coupled Atmosphere Ocean Sea-Ice General Circulation Climate Model. RP0015-INGV-SXG, Technical Report, Centro Euro-Mediterraneo per i Cambiamenti Climatici, February 2007.
- [11] Smith W. and Sandwell D., 1997: Global Sea Floor Topography from Satellite Altimetry and Ship Depth Soundings. *Science*, **277**, No 5, 896-914.
- [12] Troccoli A., Källberg P., 2004: Precipitation correction in the ERA-40 reanalysis. ERA-40 Project Rep. Series 13, 6 pp.
- [13] Uppala S., and Coauthors, 2005: The ERA-40 reanalysis. *Quart. J. Roy. Meteor. Soc.*, **131**, 2961-3012.
- [14] Valcke, S., 2006: Oasis3 user guide (prism 2-5). Tech. rep., PRISM Support Initiative. Report No 3, CERFACS, Toulouse, France, 64 pp.



© Centro Euro-Mediterraneo per i Cambiamenti Climatici 2009

Visit [www.cmcc.it](http://www.cmcc.it) for information on our activities and publications.

The Euro-Mediterranean Centre for Climate Change is a Ltd Company with its registered office and administration in Lecce and local units in Bologna, Venice, Capua, Sassari and Milan. The society doesn't pursue profitable ends and aims to realize and manage the Centre, its promotion, and research coordination and different scientific and applied activities in the field of climate change study.

