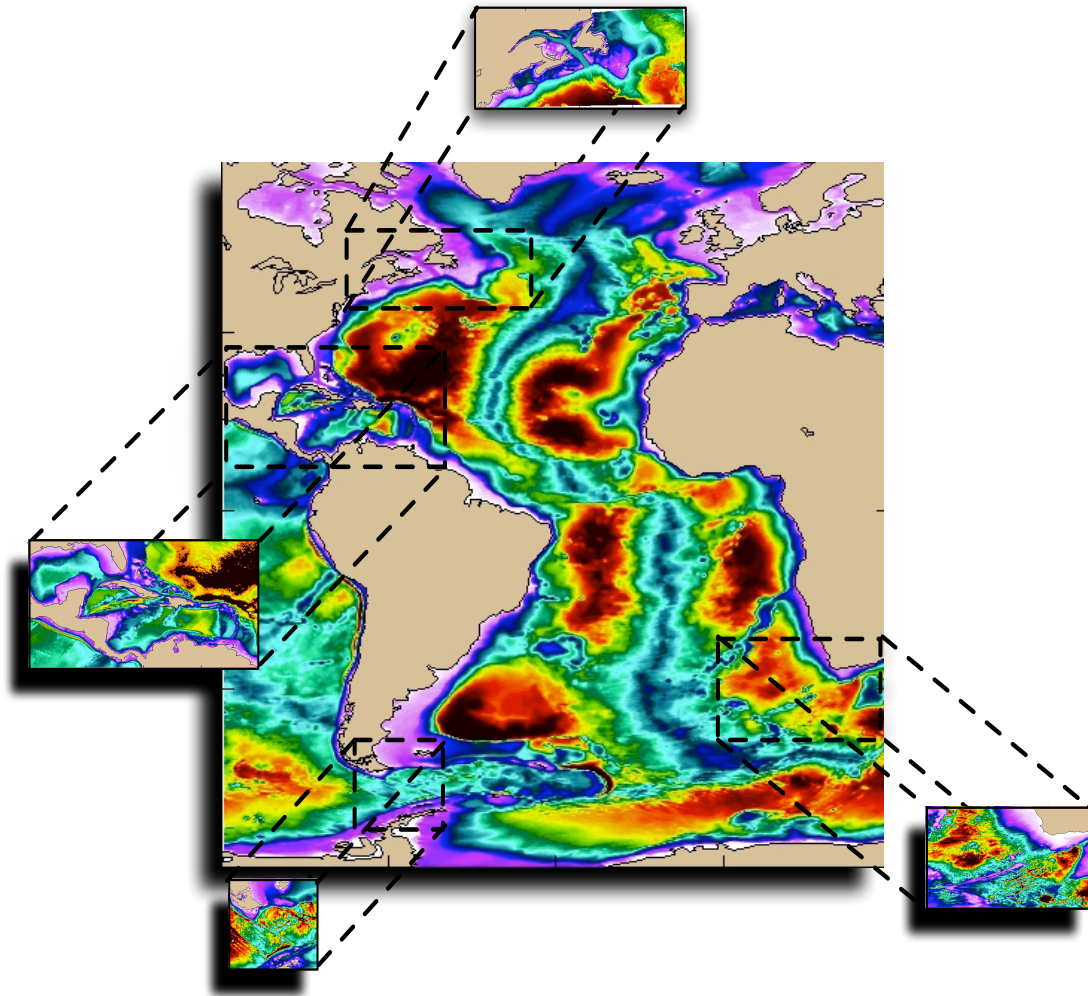


NEMO/AGRIF Nesting tools



User's Guide

January 30, 2006

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 4 |
| 2 | Installation | 6 |
| 3 | Input file | 8 |
| 4 | Running | 16 |
| 5 | Technical aspects | 18 |
| 5.1 | Coordinates interpolation | 18 |
| 5.2 | Bathymetry building | 18 |
| 5.2.1 | Interpolation method | 19 |
| 5.2.2 | Connection with parent grid bathymetry | 20 |
| 5.2.3 | Smoothing method | 20 |
| 5.2.4 | Linear connection method | 22 |
| 5.2.5 | Adding an external high resolution database | 24 |
| 5.3 | Forcing files creation | 24 |
| 5.3.1 | Remapping method | 24 |
| 5.3.2 | Conservation at the boundaries | 25 |
| 5.3.3 | Extrapolation process | 25 |
| 5.4 | Summary of Nesting tools possibilities | 27 |
| 5.4.1 | What kind of external database for bathymetry | 27 |
| 5.4.2 | What kind of tracers, flux and momentum fields | 27 |
| 5.4.3 | What kind of input files are supported | 28 |

Chapter 1

Introduction

This guide presents a series of Fortran 95 procedures that could be useful for the pre-processing of OPA/NEMO ¹ ocean model when running an embedded model.

The version 9 (called "NEMO") of the code makes use of the AGRIF² grid refinement procedure developed at the LMC-IMAG. AGRIF is a Fortran 95 package for the inclusion of adaptive mesh refinement (AMR) features within a finite difference numerical model.

To implement an embedded model using AGRIF, the user needs to provide a whole set of data containing the surface forcing fields and the initial conditions for each child domain of the grid hierarchy. Indeed for a NEMO/AGRIF run each grid must have his own input files namely bathymetry, coordinates and forcing fields files.

This package provides procedures that have been designed to perform these tasks, including bathymetry interpolation and smoothing as well as forcing data interpolation.

The next chapter describes system requirements and installation process, while the following sections describe how to use nesting tools and the various capabilities provided.

¹<http://www.lodyc.jussieu.fr/NEMO/>

²<http://www-lmc.imag.fr/IDOPT/AGRIF/>

Chapter 2

Installation

The distribution file is a gzipped tarfile, so you must uncompress the file using "gunzip" and then extract package from the tar file using "tar -xvf nesting_tools.tar". The extraction process will create a directory called Nesting_tools

System requirement

In order to compile nesting tools, you need a Fortran 95 compiler and a netCDF library (<http://my.unidata.ucar.edu/content/software/netcdf/index.html>) with F90 enabled.

This toolbox has been tested on Linux workstations with g95, pgf90 (version 5.2-4 or later) and ifort9.0 compilers, as well as on Alpha platform with f95 compiler. In the src/ directory, you must edit the makefile corresponding to your compiler in order to insert the appropriate path to netCDF library.

Once the makefile has been properly edited, type "make" and let it work. By default, the makefile builds in bin/ directory four executables named :

- create_coordinates.exe : computation of child grid coordinates
- create_bathy.exe : bathymetry interpolation, smoothing and connection
- create_data.exe : interpolation of the parent surface fluxes and tracers on child grid
- create_restart.exe : interpolation of a parent restart file on a given child grid

Chapter 3

Input file

The different options of the Nesting Tools are passed to the program through an entry file which has the structure of a namelist file.

Namelist Input

The namelist input file name can be given as an argument to various executables. By default this file is called 'namelist.input' and contains a group of namelist described in Fig. 3.1.

Figure 3.1: Required input namelist.

```
&coarse_grid_files
  parent_coordinate_file = 'coordinates.nc'
  parent_bathy_level     = 'bathy_level.nc'
/
&bathymetry
  new_topo = true
  elevation_database = 'etopo2.nc'
  smoothing = true
  smoothing_factor = 0.6
  nb_connection_pts = 3
  removeclosedseas = true
  type_bathy_interp = 1
/
&nesting
  imin = 56           ! i-direction lower left point indice
  imax = 76           ! i-direction upper right point indice
  jmin = 30           ! j-direction lower left point indice
  jmax = 50           ! j-direction upper right point indice
  rho  = 3            ! space refinement ratio
  rhot = 3            ! time refinement ratio (only for restart)
/
&vertical_grid
!
! Vertical grid parameter for domzgr ( see par_ORCA.h90 or par_NATL.h90 files )
!
  ppkth = 21.4333619793800
  ppacr = 3
  ppdzmin = 0
  pphmax  = 0
  psur = -4762.96143546300
  pa0  = 255.58049070440
  pa1  = 245.58132232490
  N    = 31
/
```

```

&partial_cells
  partial_steps = true
  parent_bathy_meter = 'bathy_meter.nc'
  e3zps_min = 50.
  e3zps_rat = 0.25
/

&nemo_coarse_grid
  jpizoom = 1           ! left bottom (i,j) indices of the zoom
  jpjzoom = 1         ! in data domain indices
/

&wind_stress
  taux_file = 'taux_1m.nc'
  tauy_file = 'tauy_1m.nc'
/

&forcing_files
  FLX_FILES =
  'data_1m_salinity_nomask.nc',
  'data_1m_potential_temperature_nomask.nc',
  'flx.nc',
  'geothermal_heating.nc'
/

&interp
  VAR_INTERP =
  'votemper/bilinear/conservative',
  'vosaline/bilinear/conservative',
  'heatflow/bilinear/conservative'
/

&restart
  restart_file = '/NULL'
  shlat = 0
  ding = false
  ding_output_file = 'test_ding'
  adatrj = 360,25
  interp_type = 'bilinear'
/

```

Namelist coarse_grid_files

The first part of this namelist contains path to files needed for bathymetry and coordinates interpolation process. *parent_bathy_level* and *parent_coordinate_file* are names of parent grid bathymetry file in levels and parent grid coordinates file.

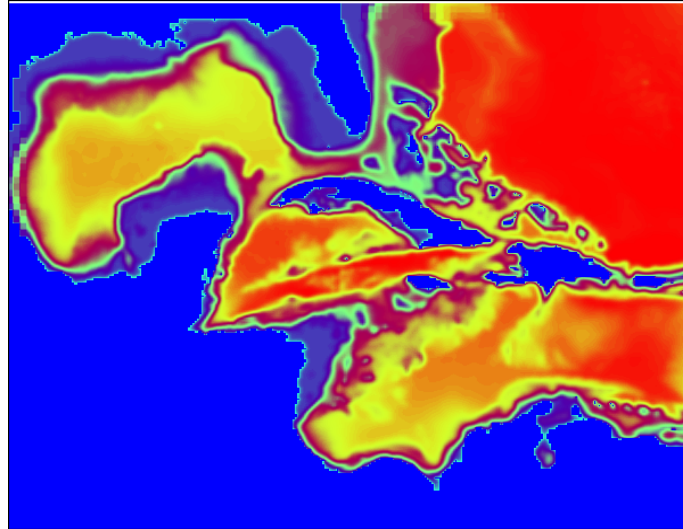
Namelist bathymetry

The bathymetry on the child grid can either be interpolated from parent grid values (if flag *new_topo* is set to false) or be recomputed from an external database (etopo2 or OPA format file currently) in that case *elevation_database* should contain the path to this fine bathymetry file. To avoid too important bathymetry gradients a smoothing algorithm is available (if flag *smoothing* is set to true) and is carried out according to a criteria called *smoothing_factor* whose values are typically between 0.6 and 0.8. Actually, *smoothing_factor* is slope parameter (factor = $\text{grad}(h)/h$) maximum value for topography smoothing.

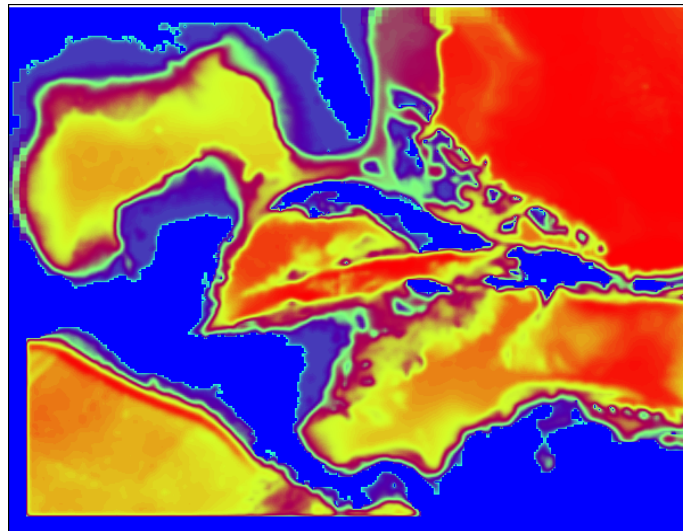
One can choose to avoid applying smoothing (in the case of tuned bathymetry) then *smoothing* must be false. In order to ensure the compatibility between parent grid and fine grid bathymetry, a linear connection is made on the boundaries of fine domain on *nb_connection_pts* points.

Thanks to *removeclosedseas* parameter it is possible to remove closed seas from your fine grid domain. This option can be useful when using for example etopo2 database which can add some extra seas in fine domain. Figure 3.2 illustrates this problem with a zoom on the gulf of Mexico from NATL3, we see that when using etopo2 a part of Pacific Ocean is incorporated whereas large scale domain doesn't contain it.

If a high resolution bathymetry is provided as input, one can choose between median average (*type_bathy_interp* = 1), arithmetic average (*type_bathy_interp* = 0) interpolation methods as well as bilinear interpolation (*type_bathy_interp* = 2). Without any external bathymetry dataset, a bilinear interpolation is carried out. When user aims at using mesh refinement in two way grid nesting mode it's possible to update parent domain bathymetry in order to ensure the compatibility between coarse and fine grids masks for interior points of the nest. This action is carried out when *parent_bathy_update* variable is true, then one can choose the name of the updated parent bathymetry file thanks to *updated_parent_file* variable.



removeclosedseas = true



removeclosedseas = false

Figure 3.2: Example of the influence of removeclosedseas option in a zoom on the gulf of Mexico from NATL3 configuration.

Namelist nesting

The third section is very important to implement an embedded model. Here is the specification of fine grid position on its parent grid and its space and time refinement factor.

$imin,imax,jmin$ and $jmax$ refer to the positions of the fine grid corners, relatively to its parent starting with the value 1 in each space direction. When the user writes these positions, one has to take care that fine grid is contained in the parent grid. $(imin,jmin)$ are the lower left corner coordinates and $(imax,jmax)$ are the upper right corner position.

Those values must be identical to the ones given in the "Agrif.FixedGrid.in" when running the model. rho and rho_t are space and time refinement ratio (time refinement ratio is used only when creating a restart file).

Namelist vertical_grid

Vertical_grid section contains various parameters related to vertical grid structure in z-coordinates.

N refers to the number of vertical levels and $ppacr$ is the stretching factor for the grid while $ppkth$ is the approximate layer number above which stretching will be maximum. Values of those variables can be found in NEMO source code (for example in file *par_ORCA_R2.h90* for *ORCA2_LIM* configuration), note that as AGRIF doesn't yet support vertical refinement, the number N of vertical levels must be the same for each grid of the hierarchy.

$ppdzmin$ is the vertical thickness of the top layer in meters and $pphmax$ is the maximum depth of the ocean. If in NEMO .h90 file $ppdzmin$ and $pphmax$ are initialized with character string 'pp_not_used', instead of those parameters you have to specify $pa0,pa1$ and $psur$ also defined in this configuration file.

Namelist partial_cells

The NEMO code offers the option to include partial cells at the bottom in order to better resolve topographic variations. If this option is chosen an additional bathymetry file in meters is required by NEMO. In this case *partial_steps* must be set to true and you have to add the path to parent bathymetry file in meters.

$e3zps_min$ is the minimum thickness of the partial step and $e3zps_rat$ is the minimum percentage of the partial step regarding to corresponding level thickness. Those both parameters should stay the same than in NEMO namelists .

The *jpizoom* and *jjpzoom* variables are parent grid lower left position on a large model (eg ORCA2).

Namelist files

The *wind_stress* and *forcing_file* sections enables the user to specify which parent fluxes files to interpolate on child grid. Then, in *interp* section user can choose what type of interpolation will be used for each variable among 'bilinear' and 'bicubic'. It's also possible to choose a correction of child grid values in order to ensure conservation on cells at the boundaries. When one doesn't have any wind stress files to interpolate (in case of coupled run), just set *taux_file* = '/NULL'.

Namelist restart

If restart file interpolation is required, *restart_file* variable determines the location of the parent file. *shlat* defines the lateral friction : *shlat* = 0, free slip (no shear along the coast); *shlat* = 2, no slip (specified zero velocity at the coast). The value *shlat* is used for the computation of the relative vorticity.

One can choose the format of the output restart file; by default a file in netcdf format is created but dimg format file can be built when *dimg* value is true, in this case the name of the output file is *dimg_output_file* and user must specify *adatrj* which corresponds to number of elapsed days since the beginning of the run given by ocean.output file (not needed for netcdf format file). Then the type of interpolation applied to restart file variables is *type_interp* and one can choose bicubic or bilinear interpolation. When using dimg format compiler options like "-assume byterecl and -convert big_endian" (used by ifort compiler) can be required.

Chapter 4

Running

Once namelist input file is properly filled in, it's possible to run the code. For basic NEMO run using AGRIF capabilities you have to execute following programs, *create_coordinates.exe* must be execute first :

1. `./create_coordinates.exe namelist_file`
2. `./create_bathy.exe namelist_file`
3. `./create_data.exe namelist_file`

If your namelist file is called 'namelist.input', you don't need to specify `namelist_file` name as argument. Additional executable named `create_restart.exe` interpolates any given parent grid restart file on child grid.

If your grid hierarchy comprises n levels you just have to create a namelist file for each level and fill in `data_file` section with files created at level $n-1$ in that case you should give value 1 for *jpizoom* and *jjzoom*.

Chapter 5

Technical aspects

5.1 Coordinates interpolation

This part aims at describing the work carried out by create_coordinates.exe. This executable creates coordinates file for child grid according to his position into coarser grid. The type of interpolation used for this task is currently 4th order polynomial interpolation. The general operations are described below. Let (λ, ϕ, z) be the geographical coordinates system.

1. **Interpolation of latitude and longitude coordinates on child grid at T,U,V and F points**

2. **Scale factors computation on T,U,V and F points**

$$- e_1 = (a + z) \left[\left(\frac{\partial \lambda}{\partial i} \cos \phi \right)^2 + \left(\frac{\partial \phi}{\partial i} \right)^2 \right]^{\frac{1}{2}}$$

$$- e_2 = (a + z) \left[\left(\frac{\partial \lambda}{\partial j} \cos \phi \right)^2 + \left(\frac{\partial \phi}{\partial j} \right)^2 \right]^{\frac{1}{2}}$$

It's possible to easily change directly in the code if user prefers to apply an interpolation process for e_1 and e_2 computation.

5.2 Bathymetry building

This is a crucial part when running an embedded model. Various options are offered for the creation of child grid bathymetry. As mentioned in chapter 3 users can choose between full steps

and partial steps, the code also offers the possibility to use external high resolution database in order to better resolve topographic variations. This section presents some key components of bathymetry construction, namely smoothing and interpolation method as well as connection with parent grid bathymetry.

5.2.1 Interpolation method

Interpolation from parent bathymetry

The method used to interpolate bathymetry on child grid is bilinear interpolation based on SCRIP¹ (Spherical Coordinates Remapping Interpolation Package). The first step is to compute remapping weights, then interpolation process is similar to a matrix-vector multiplication.

$$(c_1 \ c_2 \ \dots \ c_n) \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{pmatrix} = (f_1 \ f_2 \ \dots \ f_m)$$

The $1 \times n$ matrix C contains the 2D coarse grid field to interpolate unrolled into a single vector. The $1 \times m$ matrix F is the result of matrix multiplication so F contains the interpolated field on fine grid. Bathymetry is mapped with weights calculated using SCRIP's bilinear interpolation scheme in a quadrilateral grid. This scheme is quite simple to explain : let's consider point P on destination grid, we search for a cell formed by four points on source grid containing P and then the remapping weights must be assigned to each of those four points.

Interpolation from external high resolution bathymetry

When using a high resolution bathymetry the relevance of bilinear interpolation is not obvious. In this way an interpolation method based on an average of all high resolution points contained in a given child grid cell has been implemented. This method is quite simple, the value assigned to every child grid points is the average of all high resolution points present in its cell. There is two different way to carry out this average indeed user have the possibility to choose between arithmetic and median average.

¹<http://climate.lanl.gov/Software/SCRIP/>

5.2.2 Connection with parent grid bathymetry

The bathymetry thus interpolated will be connected to the parent topography before being smoothed .

The connection is made on three coarse grid cells at the boundaries of child grid as well as on both ghost cells. This connection is not exactly the same between full cells and partial cells. Then the way to connect this connection zone to the rest of the interior high resolution domain is a linear connection.

Full cells

In full cells configuration only bathymetry in levels is required. So the connection is simply made by copying parent grid values on corresponding child grid cells as shown in figure 5.1. Then bathymetry in levels is converted into meters in order to implement smoothing process. This smoothing is carried out by using interior domain points. To ensure connection between interior domain and boundaries a linear connection is made. The bathymetry is then reconverted in levels.

Partial cells

In partial cells the process is quite similar. But in this case we work directly on bathymetry in meters. For the three coarse cells at the boundaries, it's no need to directly copy parent value, we just have to take care about the fact that child grid value remain at the same level than corresponding parent grid value. The interpolation process (1D in each space directions) is based on Van Leer slope limiters with an additional criterion to check that the interpolated points remain at the same level.

5.2.3 Smoothing method

In order to prevent horizontal pressure gradients errors, interpolated bathymetry requires some smoothing. This is not the values of bathymetry but the values of its logarithm that are smoothed. This is a usual method which ensures that the smoothing won't be too strong in deep areas. The logarithm of the bathymetry (whose derivative is $\frac{dh}{h}$) will be smoothed until it fits the slope parameter called *smoothing_factor* in nesting tools namelist file. Thus the stop criterion of the smoothing algorithm is when $\frac{dh}{h} \leq \textit{smoothing_factor}$ on all points. The method used to carry out this smoothing makes use of hanning filter, if we note $\ln h$ the logarithm of h hanning filter is defined by :

$$\ln h(i, j) = \frac{1}{8} [\ln h(i - 1, j) + \ln h(i + 1, j) + \ln h(i, j - 1) + \ln h(i, j + 1) + 4 * \ln h(i, j)]$$

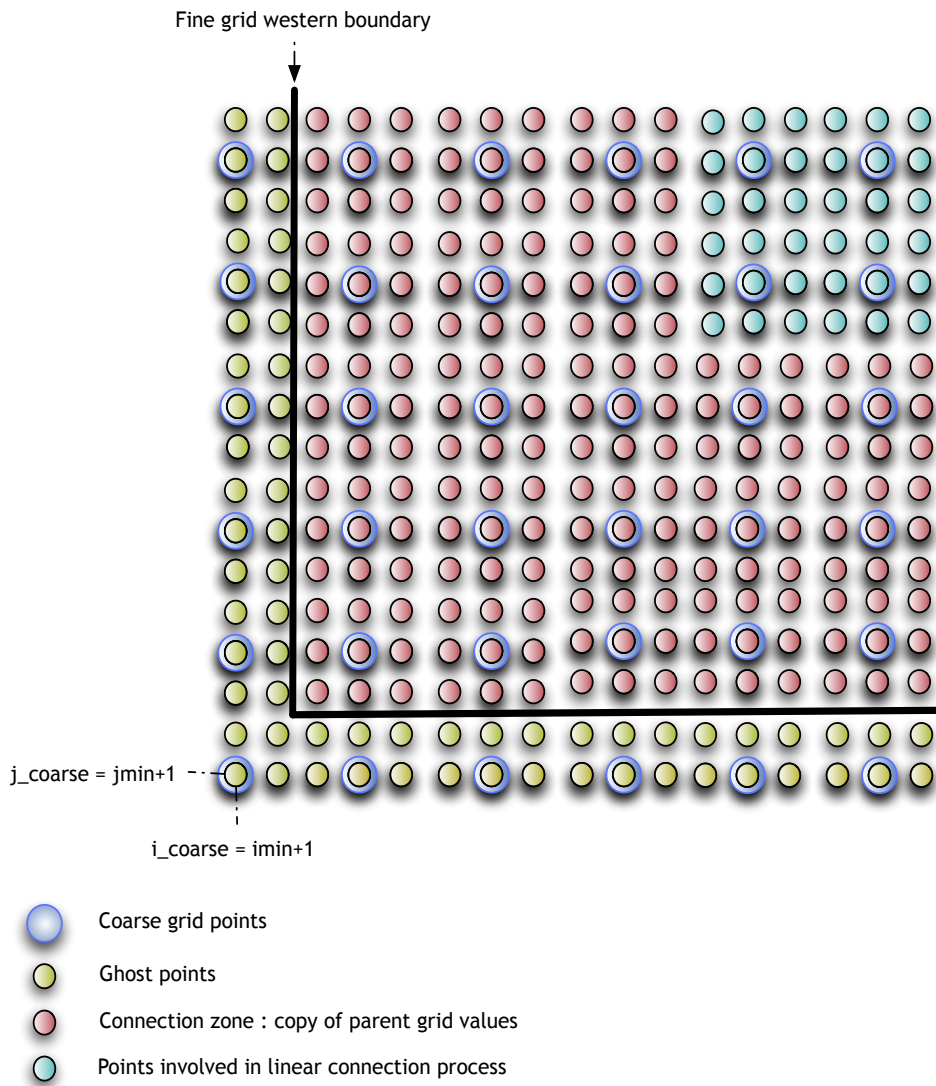


Figure 5.1: Bathymetry connection process for full cells (example with space refinement ratio = 3)

Then the following convolution kernel is applied to resulting bathymetry

$$\mathbf{K} = \frac{1}{64} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 5 & -9 & 74 & 1 & 0 \\ 1 & -4 & -9 & -2 & 0 \\ 0 & 1 & 5 & 1 & 0 \end{bmatrix}$$

we can note that the sum of the elements of this 5x5 kernel is equal to 1, so this kernel is normalized. If the kernel has m rows and n columns, mathematically we can write the convolution as:

$$lnh_{smoothed}(i, j) = \sum_{k=1}^m \sum_{l=1}^n lnh(i + k - 1, j + l - 1)K(k, l)$$

Smoothing algorithm iterates those two steps as long as smoothing criterion isn't satisfy..

5.2.4 Linear connection method

In this case, fine grid bathymetry and interpolated coarse grid bathymetry fields are blended across the *nb_connection_pts* (mentioned in namelist.input file) outer rows and columns of the fine grid. The blending is a simple linear weighting between the interpolated coarse grid values and the fine grid values from the input file. The weighting scheme for a 5 point connection is given as :

- row/column 1 : 100% interpolated coarse grid, 0% fine grid
- row/column 2 : 75% interpolated coarse grid, 25% fine grid
- row/column 3 : 50% interpolated coarse grid, 50% fine grid
- row/column 4 : 25% interpolated coarse grid, 75% fine grid
- row/column 5 : 0% interpolated coarse grid, 100% fine grid

This relaxation zone takes place between the three coarse grid at the boundaries and the interior domain.

```
netcdf etopo_file {
dimensions:
    lon = x ;
    lat = y ;
variables:
    float lon(lon) ;
    float lat(lat) ;
    float topo(lat, lon) ;
}

netcdf OPA_format_file {
dimensions:
    x = 182 ;
    y = 149 ;
variables:
    float nav_lon(y, x) ;
float nav_lat(y, x) ;
float Bathymetry(y, x) ;
}
```

Figure 5.2: sample input high resolution bathymetry file

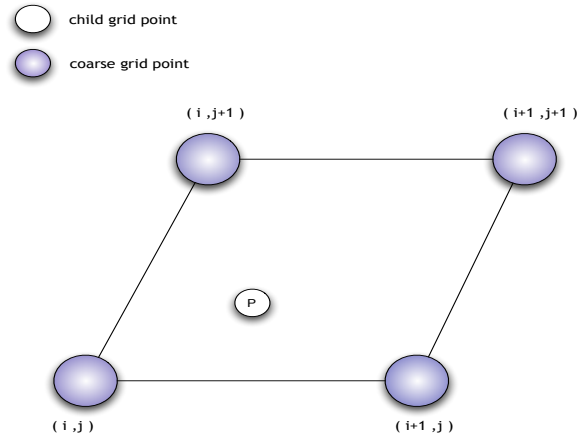


Figure 5.3: Coarse grid points box around a child grid point to interpolate

5.2.5 Adding an external high resolution database

The current code of nesting tools is designed to include etopo2 global elevation data base gridded at 2-minute (latitude-longitude) resolution as well as various OPA format bathymetry file. The user can choose to add his own database for bathymetry interpolation. In that way the code is flexible and is able to properly work if this external database is in netcdf format with the structure presented in figure 5.2. If for any reasons user doesn't want to change the structure of his file to match the structure required by the code, it's of course possible to edit directly in the code the routine responsible for external bathymetry file reading.

5.3 Forcing files creation

5.3.1 Remapping method

As introduced in section 5.2.1 we use remapping to interpolate various fields between coarse domain grid and fine domain grid. Remapping method are well adapted in our case because it accelerates interpolation process. Indeed remapping matrix is calculated once for all and can be re-used for all fields and also for all levels if fields to interpolate are not masked. At this point we have the choice between bilinear and bicubic interpolation schemes. We can quickly present those both methods. Let f be the function to approximate on child grid point P . A search algorithm on coarse grid finds in which box the point P resides. If we use indices defined in figure 5.3, bilinear scheme gives

$$f_P = w_1 f(i, j) + w_2 f(i + 1, j + 1) + w_3 f(i + 1, j + 1) + w_4 f(i, j + 1)$$

whereas bicubic scheme can be written

$$\begin{aligned} f_P = & w_1 f(i, j) + w_2 f(i + 1, j + 1) + w_3 f(i + 1, j + 1) + w_4 f(i, j + 1) + \\ & w_5 \frac{\partial f}{\partial i}(i, j) + w_6 \frac{\partial f}{\partial i}(i + 1, j) + w_7 \frac{\partial f}{\partial i}(i + 1, j + 1) + w_8 \frac{\partial f}{\partial i}(i, j + 1) \\ & w_9 \frac{\partial f}{\partial j}(i, j) + w_{10} \frac{\partial f}{\partial j}(i + 1, j) + w_{11} \frac{\partial f}{\partial j}(i + 1, j + 1) + w_{12} \frac{\partial f}{\partial j}(i, j + 1) \\ & w_{13} \frac{\partial^2 f}{\partial i \partial j}(i, j) + w_{14} \frac{\partial^2 f}{\partial i \partial j}(i + 1, j) + w_{15} \frac{\partial^2 f}{\partial i \partial j}(i + 1, j + 1) + w_{16} \frac{\partial^2 f}{\partial i \partial j}(i, j + 1) \end{aligned}$$

For each field to interpolate user can choose between the both schemes. You can note that bicubic remapping can create some local extrema whereas values interpolated thanks to bilinear scheme are limited by parent grid values.

5.3.2 Conservation at the boundaries

For two way interactive grid nesting user can choose to ensure conservation mainly for tracers. This conservation is implemented on the two coarse grid cells at the boundary of child grid, namely the zone where coarse grid points will be updated. The process is quite simple a correction is applied to each fine grid points covered by the same coarse grid cell. If we note α this correction, ρ_x and ρ_y space refinement ratio in x and y direction, for a field F the correction is given by :

$$\alpha = \rho_x \cdot \rho_y \cdot F_{coarse} - \sum_{i=1}^{\rho_x} \sum_{j=1}^{\rho_y} F_{fine}(i, j)$$

5.3.3 Extrapolation process

This part is useful when the field to interpolate on child grid is already masked (e.g. when interpolating restart files fields). Indeed the main drawback of SCRIP package is the fact that it doesn't manage extrapolations. During search algorithm if one of the four points surrounding one given child grid point to interpolate is masked then no interpolation is done and child grid point receive the value zero even if this point is not masked. So we have to preprocess

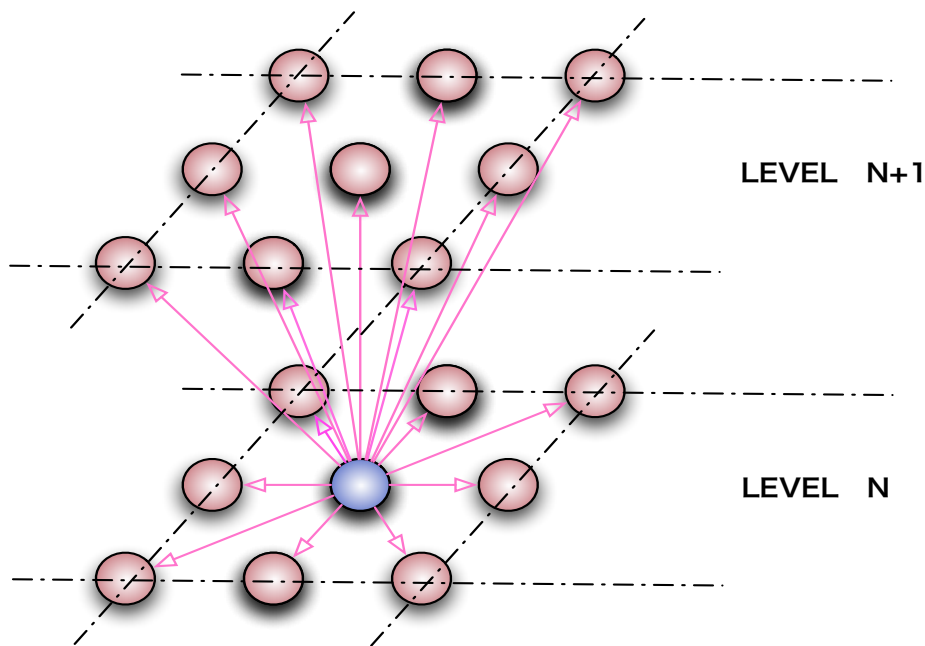


Figure 5.4: Directions investigated around a point where extrapolation is required.

interpolation step to avoid such problems by correcting mask on coarse grid and giving value to points where mask has been changed. The principle is to construct a box around the point to be extrapolated (as shown in figure 5.4) and to search for all no masked points candidates for this extrapolation. Then between all possible points we calculate the error made by replacing the value of the point to be extrapolated with this new possible value. Using Taylor expansion, the error is given by

$$|f(x + h_x, y + h_y) - f(x, y)| = |h_x \frac{\partial f}{\partial x} + h_y \frac{\partial f}{\partial y}|$$

So at each no masked points around a point to be extrapolated we compute gradients in x and y directions as well as distances h_x and h_y . Of course the point giving minimum error is chosen for extrapolation.

5.4 Summary of Nesting tools possibilities

5.4.1 What kind of external database for bathymetry

The choice of high resolution database for bathymetry is linked to the resolution of child grid. The use of etopo2 is not completely relevant in case of an ORCA2 zoom, it would be better to use a higher resolution ORCA bathymetry. For example, for an ORCA2 zoom with a space refinement ratio of 3 the best choice could consist in using ORCA05 or ORCA025 bathymetry. If one already owns a tuned fine bathymetry mapped on the fine grid points, the smoothing option isn't relevant indeed the smoothing algorithm could modify the values of this improved bathymetry. If no smoothing option is activated, outside the connection zone the interior domain bathymetry stay the same than the one given by the external file.

The type of interpolation used is closely linked to the type of external bathymetry used. Indeed median and arithmetic average are relevant when the resolution of external bathymetry is much higher than the one of fine grid (for example when using etopo2). A bilinear interpolation is more accurate and fast when resolutions are quite similar.

5.4.2 What kind of tracers, flux and momentum fields

This problem is quite similar to the one of the previous section. One can choose to interpolate the flux, tracers and momentum fields from parent grid files. The code also offer the possibility to use ORCA025 or ORCA05 forcing files to interpolate on child grid even if one is implementing ORCA2 zoom.

This is a quite easy stuff to do, just replace in the namelist file the coordinate file by the one corresponding to the new fields, for example ORCA05 coordinate file, and then user have to mention ORCA05 forcing files names in FLX_FILES and wind_stress sections. This is relevant only when forcing files are not masked and when the vertical level position (for 3D fields) is exactly the same (this is the case between ORCA05 and ORCA2 configurations) because no vertical interpolation is currently implemented.

The default version of the code doesn't carry out extrapolation during forcing files interpolation because NEMO standard files are not masked. If user wants to activate extrapolation during interpolation process on files mentioned by FLX_FILES it's possible to activate this option by editing agrif_interpolation.f90 file, (in routine Interp_Extrap_var) the logical variable Extrapolation must be initialize to true value (this option is not available for wind stress files interpolation).

| namelist variable name | variable position | netcdf file dimensions | variable type | variable name |
|------------------------|-------------------|---------------------------------------|---|--|
| taux_file | U | x,y,time_counter, depthu(o),deptht(o) | 3D (x,y,time_counter) | 'taux' or 'sozotaux' |
| tauy_file | V | x,y,time_counter, depthu(o),deptht(o) | 3D (x,y,time_counter) | 'tauy' or 'sometauy' |
| FLX_FILES | T | x,y,time_counter, deptht or z | 4D or 3D (x,y,time_counter) (x,y,z,time_counter) (x,y,deptht,time_counter) | interpolate any name different from 'nav_lat', 'nav_lon', 'nav_lev', 'deptht', 'time_steps', 'time_counter', 'tmask' |

(o) : optional dimensions

Figure 5.5: Netcdf files structure supported by nesting tools

5.4.3 What kind of input files are supported

The current version of Nesting tools have been fully tested with NATL3 and ORCA2_LIM configurations (tests with NATL4 and ORCA025 are currently carried out). In the case of tuned input files, user can refer to figure 5.5 to check the current kind of netcdf file structure currently supported by nesting tools. Developments are planed to enable the interpolation of U and V type 4D variables, this feature can be useful when using TOP (ocean tracer and biochemistry model) or other additional option of NEMO.

Bibliography

- [1] G. Madec : OPA Ocean General Circulation Model, Reference Manual *http : //www.lodyc.jussieu.fr/NEMO*
- [2] Philip W. Jones : A User's Guide for SCRIP : A Spherical Coordinate Remapping and Interpolation Package *http : //climate.lanl.gov/Software/SCRIP/SCRIPusers.pdf*
- [3] Van Leer B., 1976 : Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection, *J. Comput. Phys.*, **23**, 276-299
- [4] P. Penven and P. Marchesiello : A Regional Oceanic Modeling System with mesh refinement, pre- and post-processing tools *http : //www.brest.ird.fr/Roms_tools/*