

Tutorial 07 CROCO: Encadenamiento (anidamiento offline)

1. Propósito

En este documento describiremos los pasos a seguir para generar una simulación encadenada al caso Benguela de bajo resolución (BENGUELA_LR). Con este propósito, explicaremos el procedimiento para generar los archivos de entrada y la configuración requerida para correr una simulación con estas características.

Información Previa

Para los siguientes ejercicios asumiremos que en su cuenta ya se encuentra los directorios croco y croco_tools (que fueron obtenidos en el Tutorial 01).

2. Caso 1: Simulación encadenada para Benguela

Para este primer ejemplo, realizaremos una simulación para la región entre 12 - 19°E y 29 - 33.5 °S y de resolución horizontal de 1/9 °(simulación hijo, L1), la cual estará encadenada a la simulación de Benguela de baja resolución (BENGUELA_LR; simulación padre, L0) (Figura 1). A continuación describimos los pasos a realizar.

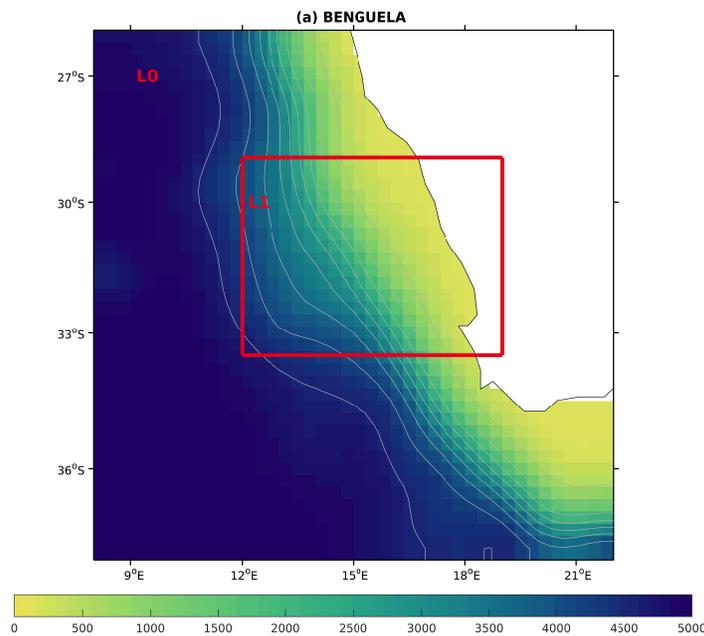


Figura 1: Topografía de la simulación padre (L0). Líneas roja limitan el área de la simulación hijo (L1)

2.1. Simulación padre

Primero necesitamos simular el caso Benguela_LR durante 90 días, guardando las salidas cada 3 días y almacenando 5 datos temporales por cada archivo avg/his. Esta simulación corresponderá a la simulación padre (L0).

2.1.1. Creación del directorio de trabajo

Crearemos un nuevo directorio donde trabajaremos en este ejercicio

```

1 cd croco
2 nano create_run.bash

```

Verificamos que los directorios de croco y croco_tools (CROCO_DIR, SOURCES_DIR y TOOLS_DIR) se encuentran correctamente definidos (ver Tutorial 01) y editamos el nombre del directorio en donde trabajaremos (MY_CONFIG_NAME):

```

1 #=====
2 # BEGIN USER SECTION
3 #
4 # Get CROCO directory
5 #
6 CROCO_DIR="/home/courses/student11/croco"
7 #
8 SOURCES_DIR="/home/courses/student11/croco"
9 #
10 TOOLS_DIR="/home/courses/student11/croco/croco_tools"
11 #
12 MY_CONFIG_PATH=${CROCO_DIR}
13 #
14 # Name of the configuration directory defined by the user
15 #
16 MY_CONFIG_NAME='BENGUELA_LO'
17 #
18 #
19 # END USER SECTION
20 #=====

```

Guardamos los cambios y ejecutamos create_run.bash

```

1 ./create_run.bash

```

y nos dirigimos hacia la carpeta

```

1 cd BENGUELA_LO

```

Este directorio será en donde trabajaremos para generar los archivos de entradas y correr la simulación. Por lo tanto, los comandos que se mencionarán, ya sea en la terminal de MATLAB o de linux, deben ser ejecutados en este directorio (a menos que se indique lo contrario).

2.1.2. Generando archivos de entrada

Para generar los archivos de entrada de la simulación padre, se debe seguir los pasos descritos en tutoriales anteriores. Como recordatorio, desde el directorio de trabajo **BENGUELA_LO** iniciamos MATLAB:

```

1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash

```

y ejecutamos las siguientes rutinas en la terminal de MATLAB:

```

1 start
2 make_grid
3 make_forcing
4 make_clim
5 exit

```

(en make_grid no realizamos modificaciones a la grilla). Al finalizar, en el directorio CROCO_FILES encontrarán los archivos croco_grd.nc, croco_frc.nc, croco_clm.nc y croco_ini.nc. En caso de tener previamente estos archivos, pueden copiarlos del directorio BENGUELA_LR/CROCO_FILES (o en el directorio en que estén guardados) al directorio BENGUELA_L0/CROCO_FILES:

```

1 cp ../BENGUELA_LR/croco_grd.nc ./CROCO_FILES
2 cp ../BENGUELA_LR/croco_frc.nc ./CROCO_FILES
3 cp ../BENGUELA_LR/croco_clm.nc ./CROCO_FILES
4 cp ../BENGUELA_LR/croco_ini.nc ./CROCO_FILES

```

Estos archivos también estarán disponibles en:

```

1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R1A/croco_grd.nc
2 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R1A/croco_frc.nc
3 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R1A/croco_ini.nc
4 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R1A/croco_clm.nc

```

Recordatorio

Si ejecuta start.m, verifique que la variable `tools_path` esté correctamente definida (ver Tutorial 03), donde `tools_path='/home/courses/student11/croco/croco_tools/';`

2.1.3. Compilando y corriendo simulación

Para correr la simulación en el servidor, debemos modificar el archivo `cppdefs.h`

```

1 nano cppdefs.h

```

donde definimos la opción MPI

```

1          /* Parallelization */
2 # undef  OPENMP
3 # define  MPI

```

Guardamos los cambios, y en la terminal de linux, compilamos `jobcomp`:

```

1 ml purge
2 ml intel/2018.04
3 ml netCDF-Fortran/4.4.4
4 ./jobcomp

```

posteriormente se habrá generado el ejecutable `croco`. A continuación modificaremos `croco.in` para configurar los días de simulación y a cada cuanto guardar.

```

1 nano croco.in

```

Al inicio de `croco.in` se encuentra la variable `NTIMES`, que corresponde al número de pasos de tiempo a simular. Para estimar `NTIMES` consideramos el tiempo en segundos a simular, es decir, si deseamos simular 90

días corresponderá a $3600 * 24 * 90 = 7,776 * 10^6$ segundos. Además, necesitamos definir la cantidad de segundos equivalente a un paso de tiempo, el cual es definido en croco.in con la variable *dt*. Para este ejemplo usaremos un $dt = 3600$. Entonces, para simular 90 días necesitamos $NTIMES = 7,776 * 10^6 / 3600 = 2160$, por lo tanto editamos en croco.in:

```
1 time_stepping: NTIMES dt[sec] NDTFAST NINFO
2                 2160   3600     60     1
```

Por otra parte, para que guarde a cada 3 días y 5 datos de tiempo por archivo de salida (his o avg), en croco.in se debe modificar las siguientes líneas

```
1 history: LDEFHIS, NWRT, NRPFHIS/filename
2           T      72    5
3           CROCO_FILES/croco_his.nc
4 averages: NTSAVG, NAVG, NRPF AVG / filename
5           1      72    5
6           CROCO_FILES/croco_avg.nc
```

donde NWRT y NAVG señala la cantidad de pasos a cada cuanto guardar. Considerando el *dt* definido, entonces un día equivale a $24 * 3600[seg]/dt = 24$ pasos. Si deseamos guardar a cada 3 días, entonces la cantidad de pasos a cada cuanto guardar es $24 * 3 = 72$.

NRPF AVG (NRPFHIS) señala el número de registro de tiempo a guardar en cada archivo avg (his). Si NRPF AVG (NRPFHIS)=0, solo se creará un solo archivo avg (his) donde se guardarán las salidas. En cambio, si NRPF AVG (NRPFHIS)=N, un archivo avg (his) solo contendrá N datos temporal.

Ahora corremos la simulación padre. Para esto, copiamos el script *run_nlhpc.bash*, el cual ha sido utilizado en tutoriales previos, en el directorio actual

```
1 cp ../BENGUELA_LR/run_nlhpc.bash ./
```

y enviamos la tarea al servidor

```
1 sbatch run_nlhpc.bash
```

Al finalizar encontrarás en el directorio de CROCO_FILES, 6 (7) archivos avg (his) con el formato croco_avg.00000.nc, croco_avg.00005.nc, ..., etc (croco_his.00000.nc, croco_his.00005.nc, ..., etc) y un archivo rst. Un archivo his/avg, como por ejemplo, croco_avg.00000.nc contiene 5 datos de tiempo, donde cada dato es el promedio de 3 días. Esto implicará el archivo croco_avg.00000.nc comprende un periodo de 15 días.

En caso de no haber logrado estos archivo, estarán disponibles en

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00000.nc
2 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00005.nc
3 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00010.nc
4 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00015.nc
5 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00020.nc
6 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L0/croco_avg.00025.nc
```

2.2. Simulación hijo

La simulación hijo es la simulación que comprende un área menor al dominio padre, y presenta una mayor resolución horizontal. Usualmente se recomienda que la resolución sea mayor entre 3 y 5 veces más que la simulación padre.

2.2.1. Creación del directorio de trabajo

A continuación crearemos otra carpeta para trabajar la simulación hijo, para lo cual debemos realizar los pasos que ya han sido explicados en la sección anterior. Si nuestro directorio actual es BENGUELA_L0 entonces,

```
1 cd ..
```

luego modificamos el archivo create_run.bash

```
1 nano create_run.bash
```

donde cambiamos el nombre del directorio que se creará (y verificamos que las variables CROCO_DIR, SOURCES_DIR y TOOLS_DIRS presentan la misma configuración mostrada la sección 2.1.1):

```
1 MY_CONFIG_NAME='BENGUELA_L1'
```

guardamos los cambios y ejecutamos

```
1 ./create_run.bash
```

Ahora, nos dirigimos a la carpeta creada

```
1 cd BENGUELA_L1
```

Ahora el directorio **BENGUELA_L1** será donde trabajaremos para generar los archivos de entradas y correr la simulación. Por lo tanto, los comandos que se mencionarán, ya sea en la terminal de MATLAB o de linux, deben ser ejecutada en este directorio, a lo menos que se mencioné directamente otro directorio.

2.2.2. Generación de archivos de entrada

2.2.2.1 Grilla: GRD

Comenzamos por editar crocotools_param.m, para configurar la región del dominio hijo (que debe ser un área menor y contenida al dominio padre, y debe evitar la capa esponja, como también, accidentes geográficos importantes). Entonces, en la terminal modificamos crocotools_param.m

```
1 nano crocotools_param.m
```

Acá, editamos las latitudes y longitudes de la simulación hijo. Una propuesta es (el área se muestra en la Figura 1):

```
1 % Grid dimensions:
2 %
3 lonmin = 12; % Minimum longitude [degree east]
4 lonmax = 19; % Maximum longitude [degree east]
5 latmin = -33.5; % Minimum latitudeF [degree north]
6 latmax = -29; % Maximum latitude [degree north]
```

También en crocotools_param.m definimos la resolución espacial (dl), la cual será 3 veces mayor a la simulación padre ($1/3^\circ$). Entonces la simulación hijo tendrá un resolución de $1/9^\circ$:

```
1 %
2 % Grid resolution [degree]
3 %
4 dl = 1/9;
```

Guardamos los cambios de crocotools_param.m e iniciamos MATLAB

```
1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/fort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash
```

y en la terminal ejecutamos start.m y make_grid.m

```
1 start
2 make_grid
```

Recordatorio

Si ejecuta start.m, verifique que la variable `tools_path` esté correctamente definida (ver Tutorial 03), donde `tools_path='/home/courses/student11/croco/croco-tools/';`

Nos preguntará si deseamos hacer modificaciones (como rotar la grilla) y para este caso respondemos no (n)

```
1 Title: Benguela Model
2
3 Resolution: 1/9 deg
4
5 Do you want to use interactive grid maker ?
6 (e.g., for grid rotation or parameter adjustments) : y,[n] n
```

Luego nos preguntará si deseamos editar la máscara y decimos no (n)

```
1 Do you want to use editmask ? y,[n] n
```

Al finalizar la rutina, tendremos la grilla hijo en CROCO_FILES. Es importante destacar que a medida que se ejecute MATLAB nos irá enviando información sobre la grilla. Entre eso, las dimensiones de la grilla **LLm** y **MMm**, con los cuales reemplazaremos en **param.h**. Esta información se encuentra entre la primera y segunda pregunta que realiza MATLAB/Octave al ejecutar make_grid.

```
1 Do you want to use interactive grid maker ?
2 (e.g., for grid rotation or parameter adjustments) : y,[n] n
3
4 Create the grid file...
5 LLm = 62
6 MMm = 47
7
8 .
9 ..
10 ....
11
12 Do you want to use editmask ? y,[n] n
13
```

Entonces para esta grilla los valores de LLm y MMm son 62 y 47 respectivamente. Esta información la utilizaremos posteriormente. Por ahora cerramos MATLAB.

```
1 exit
```

Con el fin de que no ocurran errores durante la simulación, los datos de la grilla aún deben ser modificados. Para esto, debemos hacer que los bordes de la topografía hijo coincida con los bordes de la topografía padre. Con este propósito utilizaremos las rutinas de ROMS2ROMS (descrito en Mason et al.,2010; código originalmente disponible en ftp://ftp.ifremer.fr/ifremer/cersat/users/sjullien/CROCO/Ucla_tools_June2017.tar). Primero, debemos descargar las rutinas almacenadas en:

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/R2RV2.tar
2 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/INIV2.tar
```

Y luego descomprimos los archivos:

```

1 tar -xvf R2RV2.tar
2 tar -xvf INIV2.tar

```

Esto generará dos directorios: (i) R2RV2 donde se encuentran las rutinas para las condiciones de borde y (ii) INIV2 donde se encuentran las rutinas para las condiciones iniciales. Ambos directorios deben encontrarse en el directorio actual de trabajo.

A continuación, modificamos la rutina `make_h.m` que se encuentra en R2RV2

```

1 nano R2RV2/make_h.m

```

En esta rutina, debemos indicar la dirección de los directorios donde se encuentra la grilla padre (`pdir`) y la grilla hijo (`cdir`) y el nombre de los archivos de la grilla padre e hijo (`pgrid` y `cgrid`, respectivamente).

```

1 % ROMS parent and child grid directories
2 pdir = '/home/courses/student11/croco/BENGUELA_LO/CROCO_FILES/';
3 cdir = '/home/courses/student11/croco/BENGUELA_L1/CROCO_FILES/';
4 pgrid = 'croco_grd.nc';
5 cgrid = 'croco_grd.nc';

```

Luego, definimos los bordes abiertos de nuestra dominio hijo. Esto se define en la variable `obcflag`, que es un vector donde la primera, segunda, tercera y cuarta posición representa el borde sur, este, norte y oeste, respectivamente. Si en la simulación hijo tendrá uno de estos bordes abierto (cerrado), entonces se reemplaza por 1 (0). Debido a que el dominio hijo, el borde este se encuentra la costa (ver Figura 1), entonces estará cerrado y `obcflag` será definido como

```

1 obcflag = [1 0 1 1];      % open boundaries flag (1=open , [S E N W])

```

Por último indicamos los valores de $r - target$ y $hmin$, donde $r - target$ es el valor máximo del parámetro de pendiente de la topografía y $hmin$ es la profundidad mínima del dominio. Estos valores se encuentran definidos en `croco_tools_param.m` (donde $r - target = rmax$).

```

1 rmax = 0.25;
2 hmin = 75;

```

Guardamos los cambios, e iniciamos nuevamente MATLAB:

```

1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash

```

donde ejecutamos `start.m`, añadimos el directorio R2RV2, y ejecutamos `make_h`

```

1 start
2 addpath('./R2RV2')
3 make_h

```

Recordatorio

Si ejecuta `start.m`, verifique que la variable `tools_path` esté correctamente definida (ver Tutorial 03), donde `tools_path='/home/courses/student11/croco/croco_tools/';`

Al finalizar, la rutina `make_h` habrá modificado la topografía (`h`) del archivo `croco_grd.nc` del dominio hijo y se obtendrá una figura que muestra la topografía padre, topografía hijo, la diferencia entre ambas topografías y la zona de transición en donde coincide la topografía hijo y padre (Figura 2). Posteriormente cerramos MATLAB

```

1 exit

```

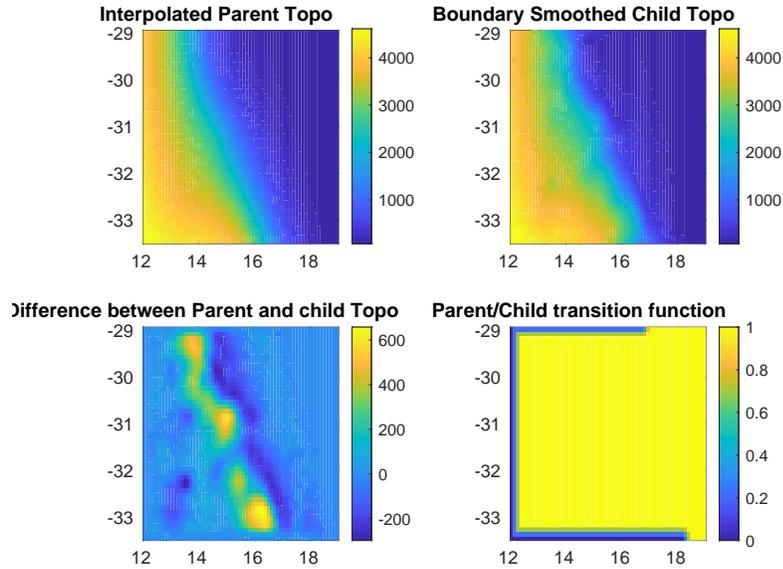


Figura 2: Figura resultante al ejecutar make_h

En caso de no haber generado este archivo, puede descargarlo en:

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L1/croco_grd.nc
```

2.2.2.2 Condiciones de borde e iniciales: BRV y INI

Para crear las condiciones de bordes (archivos BRV) e iniciales (INI) primero modificaremos las rutinas con la información de la configuración que deseamos.

Comenzamos modificando la rutina make_r2r.m (ubicado en ./R2RV2/), con la cual generaremos las condiciones de bordes:

```
1 nano ./R2RV2/make_r2r.m
```

En la primera sección de la rutina, entregaremos la información sobre la simulación padre. Para esto completamos con la siguiente información:

```
1 par_dir = '/home/courses/student11/croco/BENGUELA_L0/CROCO_FILES/';
2 par_grd = 'croco_grd.nc';
3 par_file = 'croco';
4 par_tind = 1;           % frame number in parent file
5 par_thetas = 7;
6 par_thetab = 2;
7 par_hc = 200.0;
8 par_N = 32;
9 parscoord = 'new2008'; % parent 'new' or 'old' type scoord
```

En esta parte, indicamos el directorio donde se encuentra la grilla padre y sus salidas avg y/o his (par_dir), el nombre del archivo de la grilla padre (par_grd) y el nombre con que comienza los archivos avg de la simulación padre (par_file) (por ejemplo, si el archivo es croco_avg.00000.nc, entonces el nombre es par_file=croco). Además par_tind es el tiempo de inicio, par_thetas, par_thetab, par_hc y par_N es theta_s, theta_b, hc, N. Estas últimas variables se encuentran en crocotools_param.m ubicado el directorio de la simulación padre. Por último parscoord es la versión vtransform (definido en crocotools_param.m, ubicado en el directorio de la simulación padre). Si en crocotools_param, vtransform=2 (1) entonces parscoord= 'new2008' ('old1994').

Luego entregamos la información sobre la simulación hijo, donde las variables las definiremos como:

```

1  % Child...
2  chd_dir   = '/home/courses/student11/croco/BENGUELA_L1/CROCO_FILES/';
3  chd_grd   = 'croco_grd.nc';
4  chd_file  = 'croco_ini.nc';           % name of new ini file
5  chd_thetas = 7;
6  chd_thetab = 2;
7  chd_hc    = 200.0;
8  chd_N     = 32;
9  chdscoord = 'new2008';              % child 'new' or 'old' type scoord

```

donde chd.dir es directorio donde se encuentra el archivo grid de la simulación hijo y donde se guardarán las condiciones de borde, chd_grd es el nombre de la archivo grd hijo, chd_file es el nombre del archivo de las condiciones iniciales, chd_thetas, chd_thetab, chd_hc y chd_N es theta_s, theta_b, hc, N de la simulacion hijo (que se encuentra en crocotools_param.m del directorio hijo).

Y por último modificamos la configuración respecto a las salidas bry, con la siguiente información:

```

1  -----
2  % 2. BOUNDARY FILE
3  %-----
4  obcflag   = [1 0 1 1];           % open boundaries flag (1=open , [S E N W])
5  bry_cycle = 0;                   % 0 means no cycle
6  bry_fname = 'croco_bry.nc';      % bry filename
7  bry_type  = 'avg';               % 'avg', 'his' or 'rst'
8  total_num_files = 3;             % number of files to read
9  first_file = 00010;              % first avg/his file, eg roms_avg.0282.nc gives 282
10 first_record = 1;                % desired record no. from first avg/his file
11 last_record  = 5;                % desired record no. from last avg/his file
12 num_records_per_file = 5;        % number of records per parent output file

```

donde en obcflag indicaremos los bordes que estarán abiertos en la simulación (esta variable debe coincidir con obcflag definido en make_h.m). bry_cycle indica si las condiciones de bordes tiene un ciclo climatológico (bry_cycle=1) o no (bry_cycle=0), bry_fname es el nombre del archivo de salida de las condiciones de borde, bry_type es el tipo de archivo de salidas de la simulacion padre que utilizaremos para obtener las condiciones de borde, en este caso, trabajaremos con avg (pero también puede ser his). total_num_files es la cantidad de archivos de salida de la simulación padre que se usarán, first_file es el número del nombre del primer archivo avg que se utilizará, first_record (last_record) es la primera (ultima) posición temporal del archivo avg/his inicial (termino), num_records_per_file es el número de datos por cada archivo.

Por ejemplo, para este caso utilizaremos los archivos de salida de la simulación padre desde croco_avg.00010.nc hasta croco_avg.00020.nc, esto implica que en total serán 3 archivos que usaremos (total_num_files=3), donde first_file=00010. Utilizaremos el primer registro temporal de croco_avg.00010.nc (archivo inicial), entonces first_record=1 y el último registro temporal a usar como condición de borde será el registro número 5 (last_record=5) del archivo croco_avg.00020. Basado en este ejemplo, copiamos el recuadro anterior en la configuración de make_r2r.

Continuamos con la configuración de la rutina que genera las condiciones iniciales. Modificamos el archivo r2r_ini.m que se encuentra en el directorio INIV2

```

1  nano INIV2/r2r_ini.m

```

Modificamos la información de la ubicación de los datos de la simulación padre e hijo. Nuevamente especificamos el directorio donde se encuentra la grilla padre (par_dir) y el nombre de la grilla (par_grd) y el nombre del archivo donde se va extraer los datos para las condiciones iniciales (par_ini). Si para realizar las condiciones de bordes (bry) se comenzó en el archivo croco_avg.00010.nc, entonces utilizaremos este archivo para las condiciones iniciales y usaremos el registro temporal número 2 (par_tind). No utilizamos el primer registro porque necesitamos tener datos de las condiciones de bordes y forzantes previos al día de inicio de la simulación. Por lo tanto, en la sección de la simulación padre de r2r_ini reemplazamos:

```

1  par_dir    = '/home/courses/student11/croco/BENGUELA_LO/CROCO_FILES/';
2  par_grd   = 'croco_grd.nc';
3  par_ini   = 'croco_avg.00010.nc';
4  par_tind  = 2;                % frame number in parent file
5  par_thetas = 7;
6  par_thetab = 2;
7  par_hc    = 200.0;
8  par_N     = 32;
9  parscoord = 'new2008';      % parent 'new' or 'old' type scoord

```

Posteriormente especificamos la información para simulación hijo, que es la misma configuración mencionada en la sección anterior (ver Sección 2.2.2.2)

```

1  % Child...
2  chd_dir   = '/home/courses/student11/croco/BENGUELA_L1/CROCO_FILES/';
3  chd_grd   = 'croco_grd.nc';
4  chd_file  = 'croco_ini.nc';      % name of new ini file
5  chd_thetas = 7;
6  chd_thetab = 2;
7  chd_hc    = 200.0;
8  chd_N     = 32;
9  chdscoord = 'new2008';          % child 'new' or 'old' type scoord

```

Guardamos los cambios realizados en r2r_ini.m.

A continuación, abrimos MATLAB, para ejecutar las rutinas configuradas (make_r2r.m y r2r_ini.m)

```

1  ml purge
2  ml Matlab/2017
3  LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4  compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5  matlab -nodesktop -nosplash

```

Agregamos los directorios de croco_tools y las rutinas de R2R en MATLAB

```

1  start
2  addpath('R2RV2')
3  addpath('INIV2')

```

y comenzamos con la creación de las condiciones de borde, ejecutando la rutina make_r2r.m en MATLAB

```

1  make_r2r

```

Si no hubieron errores, el mensaje que aparecerá en MATLAB

```

1  ===== Boundary file done =====

```

Ahora ejecutamos la rutina r2r_ini en MATLAB, para crear las condiciones iniciales:

```

1  r2r_ini

```

Al finalizar el mensaje que aparecerá en MATLAB

```

1  --- barotropic velocity
2  Writing ini file

```

Si no hubieron errores, y las condiciones de bordes e iniciales fueron creados con éxito, entonces en directorio CROCO_FILES (ubicado en el directorio de la simulación hijo), encontrarán los archivos croco_bry.nc y croco_ini.nc

En caso de no haber logrado estos archivos, se puede descargar en

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L1/croco_bry.nc
2 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L1/croco_ini.nc
```

2.2.2.3 Forzantes

Por último, crearemos el archivo de los forzantes (croco_frc.nc). Este paso es el más sencillo con respecto a los anteriores, ya que solo se requiere interpolar los datos padres o de la fuente original de datos. Entonces, en MATLAB ejecutamos

```
1 make_forcing
2 exit
```

Al finalizar, se habrá creado en CROCO_FILES los forzantes para la grilla hijo. En caso de no haber generado este archivo, puede descargar el archivo:

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L1/croco_frc.nc
```

2.2.3. Compilar

Para correr la simulación hijo debemos hacer los siguientes cambios en cppdefs.h

```
1 nano cppdefs.h
```

Primero, para correr la simulación en el servidor NLHPC debemos definir la opción MPI

```
1 /* Parallelization */
2 # undef OPENMP
3 # define MPI
```

Adicionalmente, para el dominio de L1 propuesto en este tutorial, observamos que el borde este no entregará información (como fue mencionamos anteriormente) por lo tanto debemos desactivar el borde este en cppdefs.h

```
1 # undef OBC_EAST
```

También desactivamos la opción CLIMATOLOGY para el modelo no solicite forzantes tipo .clm.nc

```
1 # undef CLIMATOLOGY
```

Y activaremos la opción de forzantes del tipo .bry.nc)

```
1 # define FRC_BRY
```

Guardamos los cambios realizados en cppdefs.h. Ahora, editamos el archivo param.h para actualizar las dimensiones del nuevo dominio hijo

```
1 nano param.h
```

Reemplazaremos LLm0 y MMm0 de la sección de BENGUELA_LR en param.h con los valores emitidos durante make_grid (sección 2.2.2.1)

```
1 # elif defined BENGUELA_LR
2 parameter (LLm0=62, MMm0=47, N=32) ! BENGUELA_LR
```

Guardamos los cambios, y ahora compilamos

```

1 ml purge
2 ml intel/2018.04
3 ml netCDF-Fortran/4.4.4
4 ./jobcomp

```

2.2.4. Corriendo la simulación hija

A continuación modificamos el archivo `croco.in`. Dado que se aumento la resolución, entonces dt debe ser menor al utilizado en la simulación padre, para evitar que en la simulación se produzca un “blow up”. En este caso, un valor seguro es $dt = 1600$. Esto implicará que se necesitará 54 pasos para simular un día. Entonces si simulamos 30 días necesitamos `NTIMES=1620`, es decir.

```

1 title:
2     BENGUELA TEST MODEL
3 time_stepping: NTIMES  dt[sec]  NDTFAST  NINFO
4                 1620    1600     60      1

```

Ya que se modificó dt , también debemos definir el tiempo a cada cuanto guardar en los archivos `avg/his`. Si deseamos guardar cada 3 días entonces (`NWRT/NAVG/NRST=3*54=162`)

```

1 restart:      NRST, NRPFST / filename
2              162   -1
3 CROCO_FILES/croco_rst.nc
4 history: LDEFHIS, NWRT, NRPFHIS / filename
5           T    162   0
6 CROCO_FILES/croco_his.nc
7 averages: NTSAVG, NAVG, NRPF AVG / filename
8           1    162   0

```

Ahora corremos la simulación hijo. Para esto, copiamos en el directorio actual el script `run_nlhpc.bash`, el cual ha sido utilizado en tutoriales previos:

```

1 cp ../BENGUELA_LR/run_nlhpc.bash ./

```

y enviamos la tarea al servidor

```

1 sbatch run_nlhpc.bash

```

Al finalizar tendremos un solo archivo de las salida `avg/his` en `CROCO_FILES`. Si no logró generar estos archivos entonces los podrá encontrar en:

```

1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial07/BENGUELA_L1/croco_avg.nc

```

Un ejemplo del impacto del aumento de resolución es presentado en Figura 3, donde se muestra el campo de temperatura superficial para la simulación padre e hijo para un tiempo similar.

Puede visualizar las salidas del modelo a través de `ncview`

```

1 ml purge
2 ml icc/2019.2.187-GCC-8.2.0-2.31.1 impi/2019.2.185 ncview/2.1.7
3 ncview CROCO_FILES/croco_avg.nc

```

3. Trabajo Avanzado

Realice una simulación encadenada a su elección y:

1. Analice la simulación padre e hijo y describa sus diferencia: Obtenga Figuras de la temperatura, nivel del mar, corrientes en superficie de la simulación padre e hijo y describa su diferencia

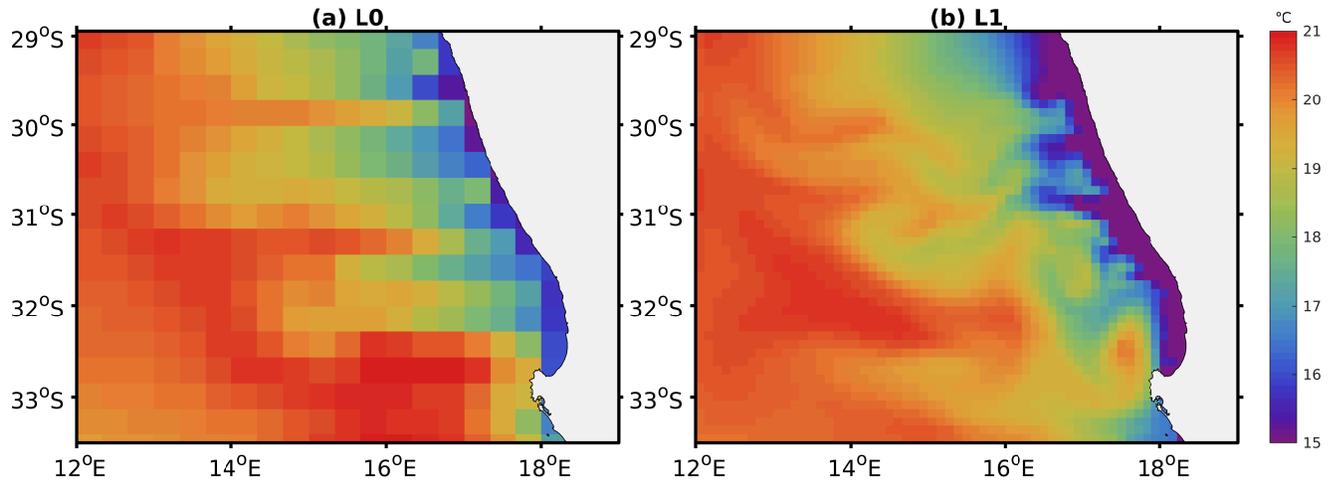


Figura 3: Caso 1: TSM de simulación L0 y L1.

2. Compare el tiempo de simulación de un 1 año de la simulación padre e hijo.

4. Conclusiones

Al finalizar esta guía, podrán generar simulaciones encadenadas (anidaciones offline) utilizando las rutinas de ROMS2ROMS.

Para más información:

Andrés Sepúlveda (asepulveda@dgeo.udec.cl)

Desarrollado por:

Marcela Contreras (marcela.contreras@legos.obs-mip.fr)

Contribuciones de:

Mauro Santiago

5. Referencias

Mason, E., Molemaker, J., Shchepetkin, A. F., Colas, F., McWilliams, J. C., & Sangrà, P. (2010). Procedures for offline grid nesting in regional ocean models. *Ocean modelling*, 35(1-2), 1-15.

ANEXOS

A. Caso 2: Benguela encadenada, modificaciones

En este ejercicio repetiremos el caso de la sección 2, pero ahora realizaremos dos modificaciones: niveles verticales y rotaremos la grilla. En simulaciones encadenadas se suele realizar estos cambios, aprovechandose de la flexibilidad de este método de “anidación“. A continuación describimos los pasos a seguir.

A.1. Simulación padre

Utilizaremos la misma simulación padre, por lo cual no es necesario hacer cambios.

A.2. Simulación hijo

A continuación describimos los pasos a seguir para esta simulación.

A.2.1. Creación del directorio de trabajo

Crearemos una segunda carpeta para esta simulación hijo, para lo cual debemos realizar los pasos que ya han sido explicados en la sección anterior. Si estamos en el directorio BENGUELA.L1 entonces

```
1 cd ..
```

modificamos el archivo create_run.bash

```
1 nano create_run.bash
```

Verificamos que los directorios de croco y croco.tools (CROCO_DIR, SOURCES_DIR y TOOLS_DIR) se encuentran correctamente definidos (ver Tutorial 01) y editamos el nombre del directorio en donde trabajaremos (MY_CONFIG_NAME):

```
1 #=====
2 # BEGIN USER SECTION
3 #
4 # Get CROCO directory
5 #
6 CROCO_DIR="/home/courses/student11/croco"
7 #
8 SOURCES_DIR="/home/courses/student11/croco"
9 #
10 TOOLS_DIR="/home/courses/student11/croco/croco_tools"
11 #
12 MY_CONFIG_PATH=\${CROCO_DIR}
13 #
14 # Name of the configuration directory defined by the user
15 #
16 MY_CONFIG_NAME='BENGUELA\L1R'
17 #
18 #
19 # END USER SECTION
20 #=====
```

Guardamos los cambios y ejecutamos create_run.bash

```
1 ./create_run.bash
```

y nos dirigimos hacia la carpeta

```
1 cd BENGUELA_L1R
```

Ahora el directorio BENGUELA_LR será en donde trabajaremos para generar los archivos de entradas y correr la simulación. Por lo tanto los comandos que se mencionarán, ya sea en la terminal de MATLAB o de linux, deben ser ejecutada en este directorio, a lo menos que se mencionó directamente otro directorio.

A.2.2. Grilla

Modificamos crocotools_param,

```
1 nano crocotools_param.m
```

Para definir las longitudes y latitudes del área del dominio y la resolución espacial, las cuales fueron propuesta en la sección 2, pero, adicionalmente modificamos los niveles verticales (N) a 42.

```
1 %
2 % CROCO title names and directories
3 %
4 CROCO_title = 'Benguela Model';
5 CROCO_config = 'Benguela_LR';
6 %
7 % Grid dimensions:
8 %
9 lonmin = 12; % Minimum longitude [degree east]
10 lonmax = 19; % Maximum longitude [degree east]
11 latmin = -33.5; % Minimum latitudeF [degree north]
12 latmax = -29; % Maximum latitude [degree north]
13 %
14 % Grid resolution [degree]
15 %
16 dl = 1/9;
17 %
18 % Number of vertical Levels (! should be the same in param.h !)
19 %
20 N = 42;
21 %
```

Guardamos los cambios, y en abrimos MATLAB

```
1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/fort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash
```

Y ejecutamos make_grid

```
1 start
2 make_grid
```

Recordatorio

Si ejecuta start.m, verifique que la variable *tools_path* esté correctamente definida (ver Tutorial 03), donde *tools_path*='/home/courses/student11/croco/croco-tools/';

Al igual que en el ejercicio anterior nos preguntará si deseamos hacer modificaciones al dominio y con el propósito de rotar la grilla, respondemos si (y)

```

1 Title: Benguela Model
2
3 Resolution: 1/9 deg
4
5 Do you want to use interactive grid maker ?
6 (e.g., for grid rotation or parameter adjustments) : y,[n] y

```

En esta ocasión aparecerá una ventana interactiva donde podemos configurar varios aspectos de la grilla (Figura A1). Nosotros solo modificaremos la variable rotación (rotation) rotando la grilla en 20°. Actualizamos la grilla (Update) y si estamos conforme con el cambio guardamos (Apply).

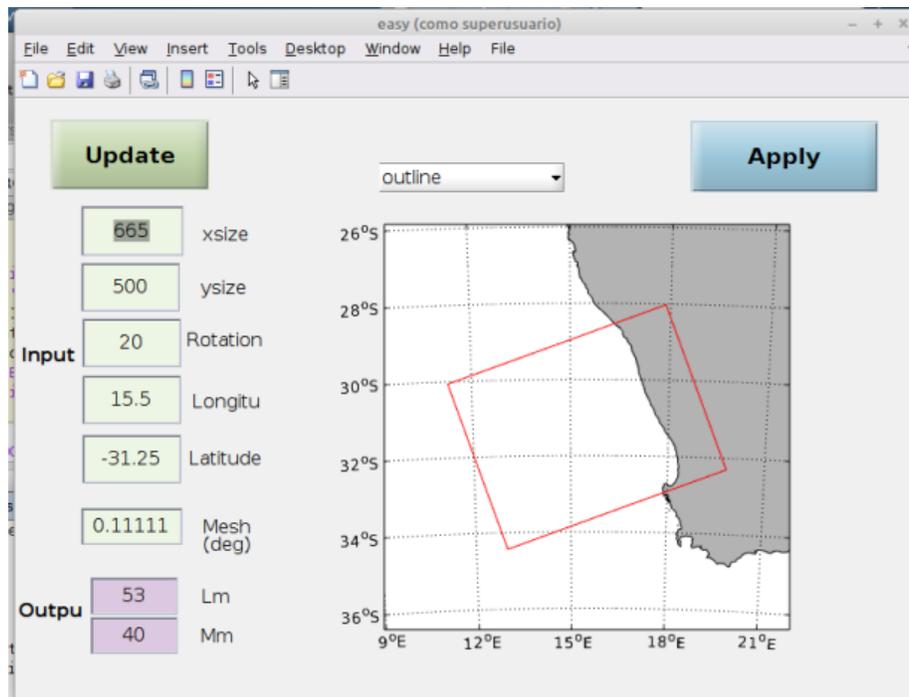


Figura A1: Ventana interactiva para la rotación de la grilla

Posteriormente, presionamos enter, y make_grid continuará con el procesamiento, donde nos preguntará si deseamos editar máscara y decimos no (n)

```

1 Do you want to use editmask ? y,[n] n

```

Recordar que MATLAB entregará la información de los valores de **LLm=53** y **MMm=40**

```

1 Do you want to use interactive grid maker ?
2 (e.g., for grid rotation or parameter adjustments) : y,[n] y
3
4 Use Easy interactive grid maker:
5 Update grid and click "Apply" in "Easy" window
6 (-> new parameters will be saved in easy_grid_params.mat)
7 ... then press a key to finalize make_grid
8 LLm = 53
9 MMm = 40
10

```

Estos valores, que tendrán que ser reemplazados en param.h, son diferentes con respecto al Caso 1 (sección 2) debido a la rotación de la grilla. Cuando finalice la rutina, cerramos MATLAB

```
1 exit
2
```

y repetimos los pasos de la sección 2, es decir, modificamos la rutina make.h.m que se encuentra en R2RV2. Para esto copiamos las rutinas de ROMS2ROMS al actual directorio. Si ya descargo estas rutinas en el ejercicio anterior :

```
1 cp -r ../BENGUELA_L1/R2RV2 ./
2 cp -r ../BENGUELA_L1/INIV2 ./
```

Y a continuación, modificamos la rutina make.h.m que se encuentra en R2RV2

```
1 nano R2RV2/make_h.m
```

Lo único que cambiaremos con respecto a la sección 2 es el directorio de la simulación hijo (cdir), para actualizar el directorio de la simulación hijo en que estamos trabajando

```
1 % ROMS parent and child grid directories
2 % ROMS parent and child grid directories
3 pdir = '/home/courses/student11/croco/BENGUELA_LO/CROCO_FILES/';
4 cdir = '/home/courses/student11/croco/BENGUELA_L1R/CROCO_FILES/';
5 pgrid = 'croco_grd.nc';
6 cgrid = 'croco_grd.nc';
```

Mantenemos la configuración de los bordes abierto de la sección anterior:

```
1 obcflag = [1 0 1 1]; % open boundaries flag (1=open , [S E N W])
```

Como también mantenemos los valores de r-target(o rmax) y hmin (ambos valores se encuentra en crocootools_param.m)

```
1 rmax = 0.25;
2 hmin = 75;
```

Iniciamos MATLAB

```
1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash
```

donde ejecutamos start.m, añadimos el directorio R2RV2, y ejecutamos make.h

```
1 start
2 addpath('./R2RV2')
3 make_h
```

Al finalizar, la rutina habrá modificado la topografía (h) del archivo croco_grd.nc y se obtendrá una figura similar a la presentada en la sección 2.2.2.1 (Figura A2).

A.2.2.1 Condiciones de borde: BRY

El procedimiento para crear las condiciones de bordes son similares al ejercicio anterior, solo cambiaremos el número de niveles verticales y el directorio de la simulación hijo. Por lo tanto modificamos la rutina make_r2r.m (./R2RV2/)

```
1 nano ./R2RV2/make_r2r.m
```

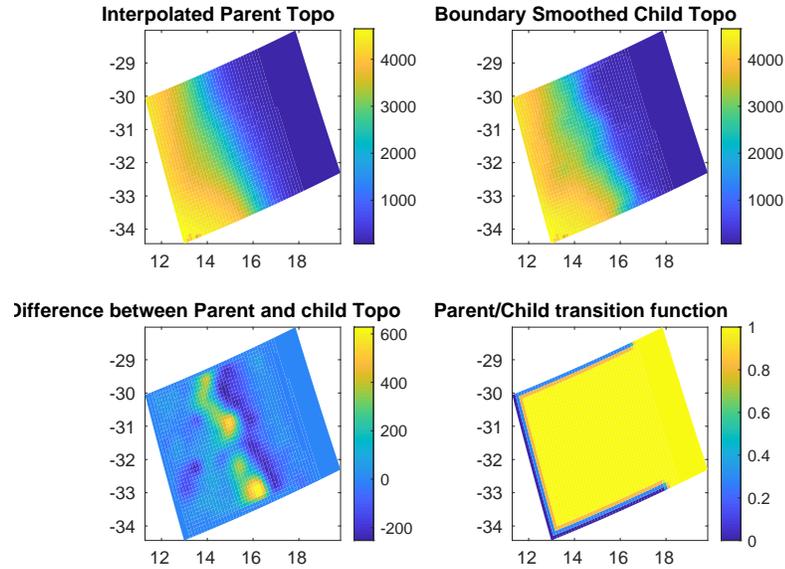


Figura A2: Figura resultante al ejecutar make_h

En la configuración de la simulación padre, mantenemos la configuración la primera sección de la rutina, lo que implica que completamos con la siguiente información.

```

1  par_dir = '/home/courses/student11/croco/BENGUELA_L0/CROCO_FILES/';
2  par_grd  = 'croco_grd.nc';
3  par_file = 'croco';
4  par_tind = 1;           % frame number in parent file
5  par_thetas = 7;
6  par_thetab = 2;
7  par_hc   = 200.0;
8  par_N    = 32;
9  parscoord = 'new2008'; % parent 'new' or 'old' type scoord

```

(Ver sección 2.2.2.2 para recordar la definición de las variables a completar). Ahora entregamos la información sobre la simulación hijo, donde utilizamos la misma configuración de la sección anterior a excepción de las variables chd_dir y N , las cuales reemplazaremos por:

```

1  % Child...
2  chd_dir  = '/home/courses/student11/croco/BENGUELA_L1R/CROCO_FILES/';
3  chd_grd  = 'croco_grd.nc';
4  chd_file = 'croco_ini.nc';           % name of new ini file
5  chd_thetas = 7;
6  chd_thetab = 2;
7  chd_hc   = 200.0;
8  chd_N    = 42;
9  chdscoord = 'new2008';              % child 'new' or 'old' type scoord

```

(Ver sección 2.2.2.2 para recordar la definición de las variables a completar)
La configuración de los archivos de salidas bry se mantiene igual a la sección 2.2.2.2.

```

1 -----
2 % 2. BOUNDARY FILE
3 %-----
4   obcflag          = [1 0 1 1];    % open boundaries flag (1=open , [S E N W])
5   bry_cycle        = 0;            % 0 means no cycle
6   bry_fname        = 'croco_bry.nc'; % bry filename
7   bry_type         = 'avg';        % 'avg', 'his' or 'rst'
8   total_num_files  = 4;            % number of files to read
9   first_file       = 00040;        % first avg/his file, eg roms_avg.0282.nc gives 282
10  first_record     = 1;            % desired record no. from first avg/his file
11  last_record      = 5;            % desired record no. from last avg/his file
12  num_records_per_file = 5;        % number of records per parent output file

```

(Ver sección 2.2.2.2 para recordar la definición de las variables a completar).

Una vez realizado los cambios en make_r2r.m, iniciamos nuevamente MATLAB

```

1 ml purge
2 ml Matlab/2017
3 LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5 matlab -nodesktop -nosplash

```

Agregamos los directorios de croco_tools y las rutinas de R2R en MATLAB

```

1 start
2 addpath('R2RV2')

```

y ejecutamos la rutina en MATLAB

```

1 make_r2r

```

Si no hubieron errores, el mensaje que aparecerá en MATLAB

```

1 ===== Boundary file done =====

```

Y cerramos de MATLAB

```

1 exit

```

Y habrá creado en CROCO_FILES, un archivo croco_bry.nc.

A.2.2.2 Condiciones iniciales: INI

Para la generación de condiciones iniciales, mantenemos la configuración mencionada en la sección 2.2.2.3 a excepción del directorio de la simulación hijo y niveles verticales. Por lo tanto editamos la rutina INIV2/r2r_ini.nc

```

1 nano ./INIV2/r2r_ini.m

```

En la sección de la configuración de la simulación padre no realizamos modificaciones

```

1  par_dir   = '/home/usuario/CROCOV3/BENGUELA_LO/CROCO_FILES/';
2  par_grd   = 'croco_grd.nc';
3  par_ini   = 'croco_his.00040.nc';
4  par_tind  = 2;           % frame number in parent file
5  par_thetas = 7;
6  par_thetab = 2;
7  par_hc    = 200.0;
8  par_N     = 32;
9  parscoord = 'new2008';  % parent 'new' or 'old' type scoord

```

Y en la sección de la configuración de la simulación hijo modificamos `chd_dir`(=' /home/courses/student11/croco/BENGUELA_L1R/CROCO_FILES/' y `N` (=42)

```

1  % Child...
2  chd_dir   = '/home/courses/student11/croco/BENGUELA_L1R/CROCO_FILES/';
3  chd_grd   = 'croco_grd.nc';
4  chd_file  = 'croco_ini.nc';      % name of new ini file
5  chd_thetas = 7;
6  chd_thetab = 2;
7  chd_hc    = 200.0;
8  chd_N     = 42;
9  chdscoord = 'new2008';          % child 'new' or 'old' type scoord

```

Guardamos los cambios en `r2r_ini`, e iniciamos MATLAB

```

1  ml purge
2  ml Matlab/2017
3  LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
4  compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
5  matlab -nodesktop -nosplash

```

Agregamos los directorios de `croco_tools` y `Roms2roms` en MATLAB

```

1  start
2  addpath('INIV2')
3  addpath('R2RV2')

```

y ejecutamos la rutina en MATLAB

```

1  r2r_ini

```

Al finalizar, si no hubieron errores, el mensaje que aparecerá en MATLAB

```

1  --- barotropic velocity
2  Writing ini file

```

Y habrá creado en `CROCO_FILES`, un archivo `croco_ini.nc`.

A.2.2.3 Forzantes

En Matlab/Octave creamos los forzantes para la simulación, siguiendo los mismo pasos de la sección 2.2.2.4, es decir ejecutamos

```

1  make_forcing
2  exit

```

Al finalizar (sin error) se habrá creado en `CROCO_FILES` los forzantes para la grilla hijo

A.2.3. Compilar

Verificamos que `cppdefs.h` presente la misma configuración descrita en sección

```
1 nano cppdefs.h
```

Esto quiere decir que debemos comprobar que los siguientes item esten definido de la siguiente manera

```
1          /* Parallelization */
2 # undef  OPENMP
3 # define MPI
4 .
5 ..
6 ...
7 # undef  OBC_EAST
8 .
9 ..
10 ...
11 # undef  CLIMATOLOGY
12 .
13 ..
14 ...
15 # define  FRC_BRY
```

Guardamos los cambios en `cppdefs.h`. También se puede copiar el `cppdefs.h` que se encuentra en el directorio de la simulación hijo del ejercicio anterior (`BENGUELA_L1`)

```
1 cp ../BENGUELA_L1/cppdefs.h ./
```

A continuación, para actualizar las dimensiones del nuevo dominio hijo (ya que no tendrá las mismas dimensiones que en el caso anterior, debido a la rotación de la grilla y al aumento de niveles verticales), debemos editar el archivo `param.h`.

```
1 nano param.h
```

En este caso la grilla hijo tiene los valores `LLm0=53`, `MMm0=40` y `N=42`

```
1 # elif defined BENGUELA_LR
2     parameter (LLm0=53, MMm0=40, N=42) ! BENGUELA_LR
```

Guardamos los cambios, y ahora compilamos

```
1 ml purge
2 ml intel/2018.04
3 ml netCDF-Fortran/4.4.4
4 ./jobcomp
```

A.2.4. Corriendo la simulación hija

A continuación modificamos el archivo `croco.in`, utilizando la misma configuración del caso anterior, es decir, $dt = 1600$ y corremos 30 días la simulación ($NTIMES = 1620$)

```
1 title:
2     BENGUELA TEST MODEL
3 time_stepping: NTIMES dt[sec] NDTFAST NINFO
4                 1620    1600    60    1
```

Y modificamos para guardar a cada 3 días (al igual que en el caso anterior)

```

1 restart:      NRST, NRPFIRST / filename
2              162   -1
3      CROCO_FILES/croco_rst.nc
4 history: LDEFHIS, NWRT, NRPFHIS / filename
5          T    162   0
6      CROCO_FILES/croco_his.nc
7 averages: NTSAVG, NAVG, NRPF AVG / filename
8          1    162   0

```

Corremos la simulación hijo, para lo cual copiamos el script *run_nlhpc.bash*, el cual ha sido utilizado en tutoriales previos, en el directorio actual:

```

1 cp ../BENGUELA_LO/run_nlhpc.bash ./

```

y enviamos la tarea al servidor

```

1 sbatch run_nlhpc.bash

```

Al finalizar tendremos archivos *croco_avg.nc* y *croco_his.nc*.

En este caso, también podemos ver el impacto del aumento de la resolución (Figura A3). Debido a la rotación y aumento de los niveles verticales, las salidas no serán exactamente igual con la simulación hijo del caso anterior.

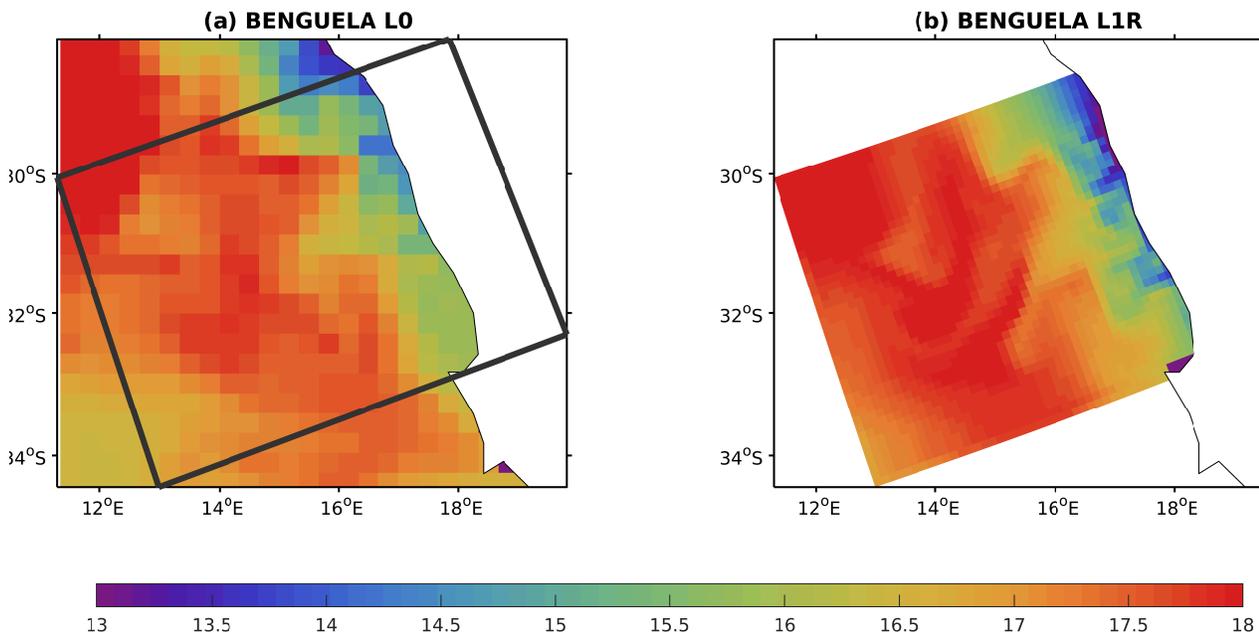


Figura A3: Caso 2: TSM de simulación padre e hijo, con rotación de grilla y aumento niveles verticales