

Tutorial 01 CROCO:

Creando el ambiente de trabajo

1. Propósito

En este tutorial revisaremos las instrucciones básicas ingresar al servidor de cálculo que usaremos en este curso, configurar y lanzar una simulación simple del dominio de Benguela, además de algunas instrucciones básicas del cluster NLHPC.

2. Ingresando al NLPHC

2.1. Desde una máquina Linux

Para conectarse al NLHPC debe escribir, desde un terminal la siguiente instrucción

```
1 ssh -XC student11@leftrarar.nlhpc.cl
```

Reemplace *student11* por el número de cuenta que le corresponde. La clave de la cuenta será enviada por Zoom. Al ingresar verá en su terminal algo parecido a esto

```
1 Laboratorio Nacional de Computacion de Alto Rendimiento (NLHPC)
2 Centro de Modelamiento Matematico (CMM)
3 Universidad de Chile
4
5 IMPORTANTE: NO EJECUTAR PROCESOS EN ESTE NODO POR T > 30 min
6 PARA ESO DEBEN DE USARSE LAS COLAS DE EJECUCION
7 .....
8
9 *****
10 EN CASO DE TENER DUDAS CON SU SCRIPT, LO INVITAMOS A USAR NUESTRO GENERADOR
11 AUTOMÁTICO EN EL SIGUIENTE LINK: https://wiki.nlhpc.cl/Generador_Scripts
12 *****
13
14 *****
15 ESTIMADO USUARIO, A CONTINUACIÓN SE LISTAN LOS NODOS LIBRES PARA SER UTILIZADOS:
16 *****
17
18 PARTICION  NODO  ESTADO
19 slims*     41    idle
20 debug      4     idle
21 general    10    idle
22 largemem   1     idle
23 student11@leftrarar3:~$
```

Cambie su clave siguiendo estas instrucciones

```
1 https://wiki.nlhpc.cl/Tutorial_de_acceso_a_Leftrarar_via_SSH#Cambiar_contrase.C3.B1a
```

3. Copiando el código de CROCO

Primero vamos a obtener una copia del código de CROCO en nuestro directorio. Para copiar el código que usaremos hay que utilizar la instrucción

```
1 cp -r /home/lmod/software/MPI/intel/2018.5.274-GCC-8.2.0-2.31.1/impi/2018.4.274/croco/1.2beta/croco .
```

4. Copiando el código de CROCO_TOOLS

Para copiar el código de CROCO_TOOLS que usaremos hay que utilizar la instrucción

```
1 cp -r /home/dbs/croco/croco_tools-master/* ./croco/croco_tools
```

5. Creando el directorio de trabajo BENGUELA_LR

El ejemplo más sencillo de CROCO es la configuración llamada BENGUELA_LR que corresponde a un dominio de la zona de surgencia de Benguela de baja resolución (de ahí el LR : *Low Resolution*). Esta configuración es la que viene por defecto en el código de CROCO y lo que haremos es similar a lo descrito en Penven et al. (2001).

El primer paso es editar el archivo **create_run.bash** con las instrucciones

```
1 cd croco
2 nano create_run.bash
```

Ahora tiene que modificar esta sección para colocar los directorios correctos

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

y después ejecute la instrucción

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

lo que le dará

```

1 student11@leftraru3:~/croco$ ./create_run.bash
2
3 Your choices :
4 - SOURCES_DIR   : /home/courses/student11/croco
5 - TOOLS_DIR     : /home/courses/student11/croco/croco_tools
6 - CONFIG_DIR    : /home/courses/student11/croco
7 - CONFIG_NAME   : BENGUELA_LR
8 Do you want to proceed ? [Y/n]

```

y al apretar la tecla **Y** aparece

```

1 Creating configuration ...
2
3 => Copy the source files from /home/courses/student11/croco
4     needed to setup your own simulations
5
6 => Copy from /home/courses/student11/croco done
7
8 => Copy the tools from and /home/courses/student11/croco/croco_tools
9     needed to setup your own simulations
10
11 => Copy from /home/courses/student11/croco/croco_tools done
12 /home/courses/student11/croco
13 student11@leftraru3:~/croco$

```

Ese script crea un directorio con el nombre que Ud. definió en **CONFIG_NAME** con todos los códigos necesarios para realizar su simulación, este será nuestro directorio de trabajo. El contenido de esta carpeta debe ser parecido al siguiente:

```

1 student11@leftraru3:~/croco$ ls BENGUELA_LR/
2 cppdefs.h          domain_def.xml      namelist_pisces_ref   SCRATCH
3 create_run.bash.BCK field_def.xml_full  namelist_pisces_ref.1 sediment.in
4 CROCO_FILES        iodef.xml          oct_start.m           start.m
5 croco.in           jobcomp            param.h               TEST_CASES
6 croco.in.1         Misc               README_XIOS           xios_launch.file
7 croco_inter.in     NAMELIST_OANALYSIS run_croco.bash
8 crocotools_param.m namelist_pisces_cfg run_croco_forecast.bash
9 DATA              namelist_pisces_cfg.1 run_croco_inter.bash

```

6. Compilando CROCO

Primero vamos a compilar el ejecutable de CROCO usamos las siguientes instrucciones

```

1 cd BENGUELA_LR
2 ml purge
3 ml intel/2019b
4 ml netCDF-Fortran/4.4.4
5 ./jobcomp

```

y la compilación del código comienza con los siguientes mensajes en pantalla

```

1 student11@leftraru3:~/croco/BENGUELA_LR$ ./jobcomp
2 OPERATING SYSTEM IS: Linux
3 cp: omitting directory '/home/courses/student11/croco/OCEAN/../../PISCES/SED'
4 file namelist_pisces exists in Run directory
5 Checking COMPILEAGRIF...
6 Checking COMPILEMPI...
7 Checking COMPILEXIOS...
8 Checking COMPILEOASIS...
9 Checking COMPILEOMP...
10 cpp -traditional -DLinux -Difort -P -I/home/lmod/software/MPI/intel/
11 2018.5.274-GCC-8.2.0-2.31.1/impi/2018.4.274/netCDF-Fortran/4.4.4/
12 include -ICROCOFILES/AGRIF_INC mpc.F > mpc.f
13 mpiifort -O3 -f2 -fno-alias -i4 -r8 -fp-model precise
14 -mmodel=medium -axCORE-AVX512,AVX,SSE4.2 -o mpc mpc.f

```

Ahora hay que esperar un par de minutos a que el compilador **mpiifort** genere el archivo ejecutable. Cuando la compilación termina exitosamente, verá las siguientes líneas en su pantalla

```

1 re/MPI/intel/2018.5.274-GCC-8.2.0-2.31.1/impi/2018.4.274/imkl/2018.4
2 .274/lib -L/home/lmod/software/MPI/intel/2018.5.274-GCC-8.2.0-2.31.1
3 /impi/2018.4.274/imkl/2018.4.274/mkl/lib/intel64
4 -L/home/lmod/software/MPI/intel/2018.5.274-GCC-8.2.0-2.31.1/impi/
5 2018.4.274/imkl/2018.4.274/lib
6 -L/home/lmod/software/MPI/intel/2018.5.274-GCC-8.2.0-2.31.1/impi/
7 2018.4.274/netCDF/4.6.2/lib64 -lnetcdf -lnetcdf -liomp5 -lpthread
8 mv a.out croco
9 student11@leftraru3:~/croco/BENGUELA_LR$

```

En la última línea se ve cómo el archivo compilado, cuyo nombre por defecto es **a.out** es renombrado como **croco**. Ese es el ejecutable que usaremos.

7. Creando los archivos de entrada

Para crear los archivos de entrada que leerá el ejecutable **croco** usaremos una serie de funciones llamadas en su conjunto **CROCO_TOOLS**. Estas funciones fueron escritas en Matlab (Penven et al., 2008) y han sido adaptadas para funcionar en Octave, usando el paquete OCTCDF.

7.1. El código de CROCO_TOOLS

Para poder usar el código de CROCO_TOOLS debe editar el archivo **start.m**

```
1 nano start.m
```

y modificar la siguiente línea

```
1 tools_path='/home/courses/student11/croco/croco_tools/';
```

7.2. Usando Matlab

Para crear los archivos de entrada usando Matlab, las instrucciones a usar, desde el directorio de trabajo **BENGUELA_LR** son

Para crear los archivos de entrada usando Matlab, primero tenemos que cargar el programa usando

```
1 ml purge
2 ml Matlab/2017
3
4 LD_PRELOAD=/home/lmod/software/Core/ifort/2019.2.187-GCC-8.2.0-2.31.1/
5 compilers_and_libraries_2019.6.324/linux/compiler/lib/intel64/libirc.so
6
7 matlab -nodesktop -nosplash
```

Dentro de Matlab las instrucciones a usar, desde el directorio de trabajo **BENGUELA_LR** son:
Primero definir los caminos de búsqueda (*path*) de las herramientas que usa **CROCO_TOOLS**

```
1 start
```

lo que entrega

```
1 >> start
2 Add the paths of the different toolboxes
3 Arch : x86_64 - Matlab version : 2017a
4 Use of mexnc and loaddap in 64 bits.
```

Luego escribimos la instrucción para generar la grilla del modelo, la que queda descrita en el archivo **cro-co_grd.nc** que se generará en el directorio **CROCO_FILES**

```
1 >> make_grid
2 mkdir: cannot create directory '/home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/':
3 File exists
4
5 Making the grid: /home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/croco_grd.nc
6
7 Title: Benguela Model
8
9 Resolution: 1/3 deg
10
11 Do you want to use interactive grid maker ?
12 (e.g., for grid rotation or parameter adjustments) : y,[n]
```

apretamos **n** y luego aparece

```
1 Create the grid file...
2 LLm = 41
3 MMm = 42
4
5 Fill the grid file...
6
7 Compute the metrics...
8
9 Min dx=29.1913 km - Max dx=33.3244 km
10 Min dy=29.2434 km - Max dy=33.1967 km
11
12 Fill the grid file...
13
14 Add topography...
15   CROCO resolution : 31.3 km
16   Topography data resolution : 3.42 km
17   Topography resolution halved 4 times
18   New topography resolution : 54.6 km
19 Processing coastline_l.mat ...
20
21 Do you want to use editmask ? y,[n]
22
```

acá veremos la Fig. 1, y nuevamente escogemos la opción **n**

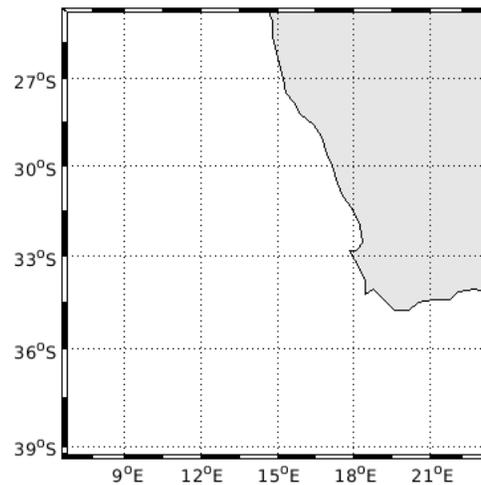


Figura 1: Mapa aproximado del dominio Benguela

finalmente vemos los siguientes mensajes

```
1 Filter topography ...
2 Apply a filter on the Deep Ocean to reduce isolated seamounts :
3   4 pass of a selective filter.
4 Apply a selective filter on log(h) to reduce grad(h)/h :
5   20 iterations - r_max = 0.27931
6   29 iterations - r_max = 0.24975
7 Smooth the topography a last time to prevent 2DX noise:
8   2 pass of a hanning smoother.
9
10 Write it down...
```

La siguiente instrucción es **make_forcing** que genera el archivo **croco_frc.nc** con la información del forzante atmosférico.

```
1 >> make_forcing
2 mkdir: cannot create directory '/home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/':
3 File exists
4
5 Benguela Model
6
7 Read in the grid...
8
9 Create the forcing file...
10 Getting tau_x for time index 1
11 Getting tau_y for time index 1
12 Getting tau_x for time index 2
13 Getting tau_y for time index 2
14 Getting tau_x for time index 3
15 Getting tau_y for time index 3
16 Getting tau_x for time index 4
17 ....
18 Getting shortrad for time index 7
19 Getting shortrad for time index 8
20 Getting shortrad for time index 9
21 Getting shortrad for time index 10
22 Getting shortrad for time index 11
23 Getting shortrad for time index 12
24 >>
```

Y finalmente la instrucción **make_clim** que crea el archivo **croco_clm.nc** con la condición de borde oceánica

```

1 make_clim
2 >>mkdir: cannot create directory '/home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/':
3 File exists
4
5 Making the clim: /home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/croco_clm.nc
6
7 Title: Benguela Model
8
9 Read in the grid...
10
11 Create the climatology file...
12
13 Creating the file : /home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/croco_clm.nc
14
15 VTRANSFORM = 2
16 ...
17 PSI: 72 iterations
18 Flux correction : 5.2674e-15
19 =====
20 Initial
21
22 Creating the file : /home/courses/student11/croco/BENGUELA_LR/CROCO_FILES/croco_ini.nc
23 VTRANSFORM = 2
24
25 Temperature...
26 Time index: 1
27
28 Salinity...
29 Time index: 1
30
31 Compute potential temperature from in-situ...
32 getpot: Time index: 1 of total: 1
33 >>

```

Notemos que esta instrucción también implica la creación del archivo de condiciones iniciales **croco_ini.nc**, el cual puede ser creado independientemente usando la instrucción **make_ini**

Es importante que primero llamemos la función **make_forcing** y después **make_clim**. Estas instrucciones crearán los archivos de entrada, en formato NetCDF. Estos archivos quedarán dentro del directorio **CROCO_FILES** y son

```
1 croco_clm.nc croco_frc.nc croco_grd.nc croco_ini.nc croco_oa.nc
```

Los archivos que obtenga deben ser iguales a los que se encuentran en

```
1 http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/ArchivosIniciales/
```

si tuvo problemas con esta etapa, copie esos archivos en el directorio **CROCO_FILES** para avanzar a la siguiente sección usando las instrucciones

```

1 cd CROCO_FILES
2 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/ArchivosIniciales/croco_grd.nc
3 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/ArchivosIniciales/croco_frc.nc
4 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/ArchivosIniciales/croco_clm.nc
5 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/ArchivosIniciales/croco_ini.nc

```

8. Lanzando la simulación

Para lanzar la simulación hay que usar el programa **nano** para crear un script bash que llamaremos **run_nlhpc.bash**.
Escriba

```
1 nano run_nlhpc.bash
```

y llene el archivo con las siguientes instrucciones

```
1 #!/bin/bash
2 #SBATCH --job-name=test01
3 #SBATCH --partition=slims
4 #SBATCH --mail-user=hectorhsepulveda@udec.cl
5 #SBATCH -n 4
6 #SBATCH --ntasks-per-node=4
7 #SBATCH --reservation=curso_udec
8
9 ml purge
10 ml intel/2019b
11 ml netCDF-Fortran/4.4.4
12 srun ./croco croco.in
```

Recuerde definir su propio correo en la línea

```
1 #SBATCH --mail-user=hectorhsepulveda@udec.cl
```

Puede encontrar un ejemplo de este archivo en el siguiente enlace

```
1 http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/run_nlhpc.bash
```

Para lanzar la simulación hay que escribir, desde el directorio de trabajo **BENGUELA_LR**

```
1 sbatch run_nlhpc.bash
```

y en pantalla veremos algo como

```
1 student11@leftraru3:~/croco/BENGUELA_LR$ sbatch run_nlhpc.bash
2 Submitted batch job 21125780
3 student11@leftraru3:~/croco/BENGUELA_LR$
```

Ahí aparece el número del proceso de cálculo asignado a esta simulación. Podemos ver el estado de avance de esta simulación con la instrucción

```
1 squeue
```

También nos llegará al correo que definimos anteriormente, indicando el inicio y el fin, exitoso o no, de la simulación.

8.1. Terminando anticipadamente una simulación

Si es necesario terminar una simulación anticipadamente, hay que usar la instrucción

```
1 scancel 21125780
```

donde debe reemplazar **21125780** con el número que le fue asignado a su proceso.

9. Archivos de salida

Una vez que la simulación termine exitosamente, encontraremos en el directorio **CROCO_FILES** los siguientes archivos de salida

```
1 croco_avg.nc
2 croco_his.nc
3 croco_rst.nc
```

Los archivos que obtenga deben ser iguales a los que se encuentran en

```
1 http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/Resultados/
```

si tuvo problemas con esta etapa, copie esos archivos en el directorio **CROCO_FILES** para avanzar a la siguiente sección usando las instrucciones

```
1 cd CROCO_FILES
2 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/Resultados/croco_avg.nc
3 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/Resultados/croco_his.nc
4 wget http://mosa.dgeo.udec.cl/CROCO2021/Tutorial01/Resultados/croco_rst.nc
```

10. Visualización de resultados

10.1. ncdump

El programa **ncdump** es muy útil para mirar el contenido de un archivo NetCDF. Recuerden que tanto los archivos de entrada como los archivos de salida de CROCO están en el formato NetCDF.

La instrucción

```
1 ml purge
2 ml netCDF-Fortran/4.4.4
3 ncdump -h CROCO_FILES/croco_avg.nc | less
```

nos mostrará información del contenido del archivo **croco_avg.nc**

```
1 netcdf croco_avg {
2 dimensions:
3     xi_rho = 43 ;
4     xi_u = 42 ;
5     eta_rho = 44 ;
6     eta_v = 43 ;
7     s_rho = 32 ;
8     s_w = 33 ;
9     time = UNLIMITED ; // (10 currently)
10    auxil = 4 ;
11 variables:
12     char spherical ;
13         spherical:long_name = "grid type logical switch" ;
14         spherical:option_T = "spherical" ;
15         spherical:option_F = "cartesian" ;
16     float xl ;
17         xl:long_name = "domain length in the XI-direction" ;
18         xl:units = "meter" ;
```

De esta forma podemos ver detalles como las dimensiones del dominio y el número de pasos de tiempo ahí grabados. Esto lo podemos comparar con nuestras estimaciones del número de pasos de tiempo que debería grabar, por ejemplo. Para salir de **ncdump** hay que apretar la tecla **q**.

10.2. ncview

El programa **ncview** es muy útil para hacer una visualización preliminar de los archivos, en formato NetCDF, que obtuvimos en nuestra simulación para esto hacemos

```

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

```

esto nos muestra la siguiente interfaz

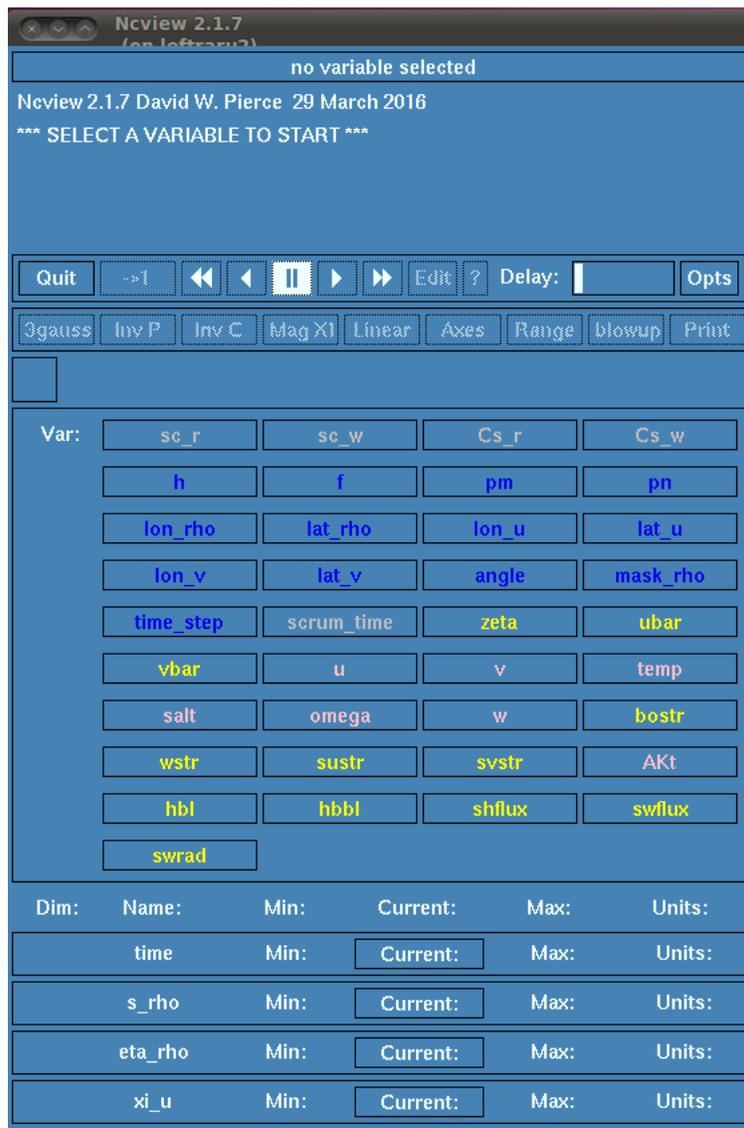


Figura 2: Interfaz gráfica de ncview

al apretar en la variable **temp** obtenemos

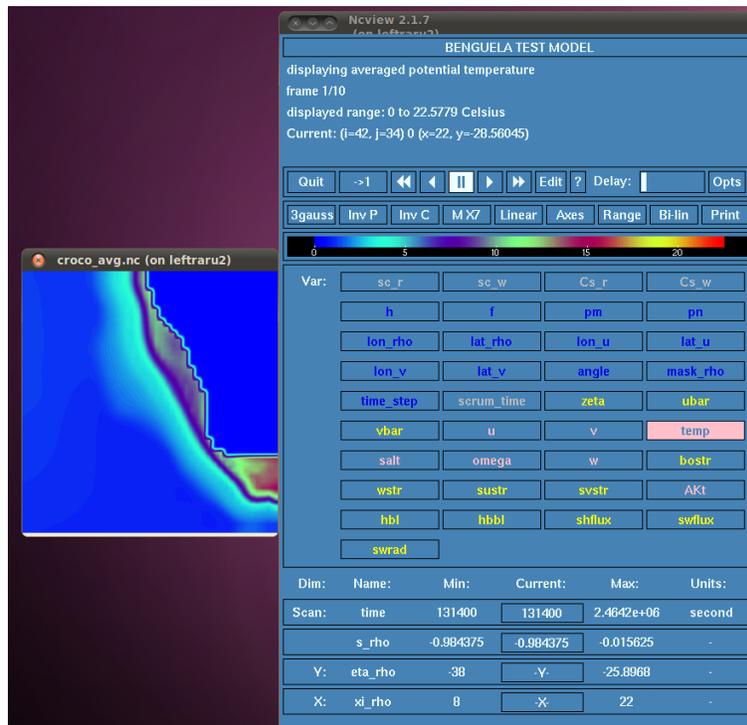


Figura 3: Temperatura del mar

Esta figura nos muestra la temperatura del mar en la campa sigma del modelo mas profunda. Si queremos ver los valores de la temperatura superficial del mar apretamos el botón derecho del ratón sobre la caja que tiene el valor **-0.984375** que corresponde al nivel vertical **s_rho** que queremos analizar. Al hacer esto el valor de esa caja cambia a **-0.015625**. Además conviene apretar en el botón que dice **Bi-lin**, para que ncview no interpole los valores que fueron calculados en cada celda. Una vez hecho esto obtenemos la siguiente figura

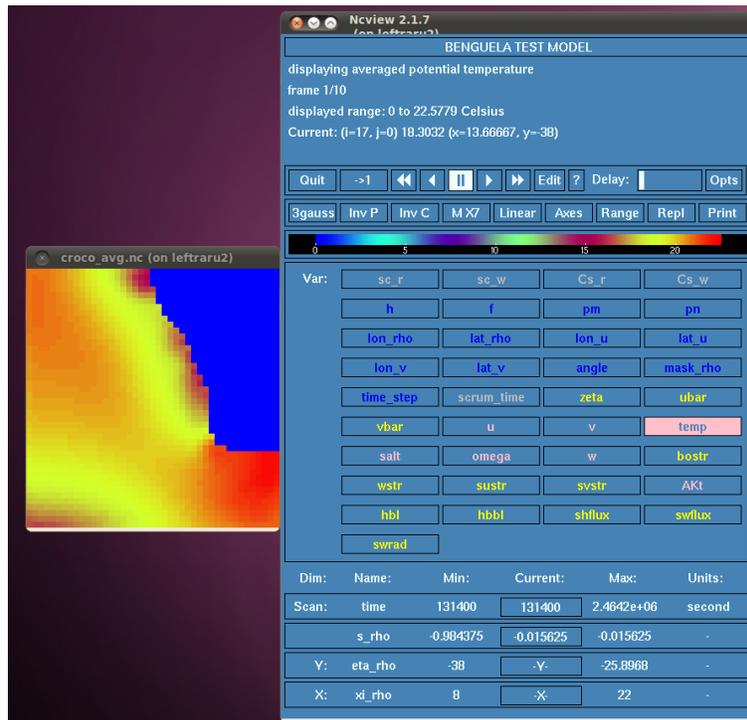


Figura 4: Temperatura superficial

Si hacemos ahora click con el ratón en algún punto del mar, **ncview** nos muestra la serie de tiempo, en superficie, de esa variable.

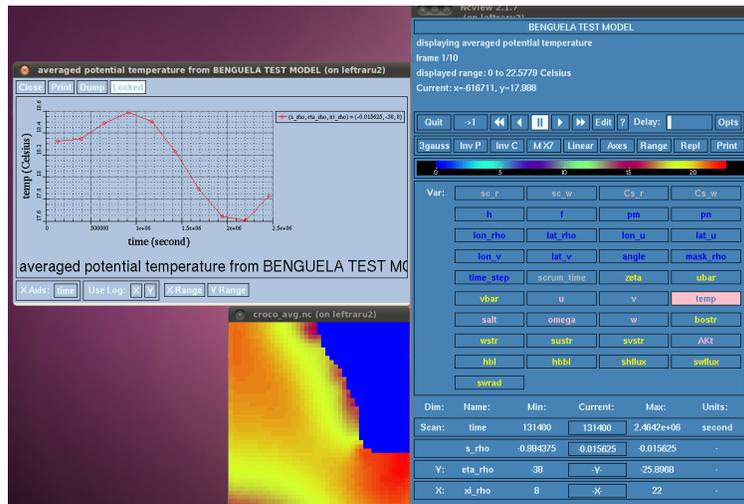


Figura 5: Serie de tiempo

En resumen, **ncview** es una herramienta muy rápida y sencilla que nos permite tener una visualización inmediata de los archivos obtenidos. Es muy útil usarla para observar si los resultados son, a primera vista, razonables.

10.3. CROCO_TOOLS

La herramienta **CROCO_TOOLS** contiene varias funciones que son útiles para visualizar los resultados de nuestra simulación. Estas funciones nos permiten hacer varios tipos de gráficos como secciones verticales, mapas superficiales, perfiles en profundidad, series de tiempo y otros. El detalle de su uso está descrito en otro tutorial.

11. Detalles avanzados

Hay aspectos importantes para realizar una simulación usando **CROCO** que dependen de cómo configuremos los archivos

```

1 crocotools_param.m
2 cppdefs.h
3 param.h
4 croco.in

```

En este caso, todo funcionó por que estos archivos están preconfigurados para el caso de ejemplo **BENGUELA_LR**. En el siguiente tutorial discutiremos qué modificar de esos archivos para estudiar la zona que sea de su interés.

12. Conclusión

En este tutorial aprendió a ingresar al centro de cómputo NLHPC, a copiar los archivos para compilar el modelo **croco** y preparar los archivos de entrada con el código **croco_tools**. Además logró lanzar la simulación básica para la zona de Benguela usando instrucciones típicas de un cluster y visualizó los resultados usando **ncview**.

Para más información:

Andrés Sepúlveda (asepulveda@dgeo.udec.cl)

Contribuciones de:

Marcela Contreras

Mauro Santiago

13. Referencias

Penven, P., Roy, C., Brundrit, G. B., De Verdière, A. C., Fréon, P., Johnson, A. S., Lutjeharms J. R. E. & Shillington, F. A. (2001). A regional hydrodynamic model of upwelling in the Southern Benguela. *South African Journal of Science*, 97(11-12), 472-475.

Penven, P., Marchesiello, P., Debreu, L., & Lefèvre, J. (2008). Software tools for pre-and post-processing of oceanic regional simulations. *Environmental Modelling & Software*, 23(5), 660-662.

14. Enlaces útiles

14.1. El modelo CROCO

<http://www.croco-ocean.org>

14.2. El foro de usuarios de CROCO

<https://forum.croco-ocean.org/questions/>

14.3. El paquete OctCDF

<http://modb.oce.ulg.ac.be/mediawiki/index.php/Octcdf>