

# Tutorial 08 CROCO: Ríos

## 1. Propósito

El objetivo de este tutorial es explicar la implementación de ríos en una simulación realizada con CROCO. En este documento revisaremos los casos de ríos de flujo constante y de flujo variable.

### 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 1a: Ríos de flujo constante

En este ejercicio simularemos el caso Benguela de baja resolución (BENGUELA\_LR; caso que ha sido analizado en tutoriales previos), donde incluiremos dos ríos de flujo constante, y definiremos la temperatura y salinidad que advecta el flujo. Para este ejemplo se propone que la ubicación de estos ríos sea  $34.65^{\circ}\text{S}$  y  $19.83^{\circ}\text{E}$ , y  $28.7^{\circ}\text{S}$  y  $16.16^{\circ}\text{E}$  (Figura 1). Notar que estos ejemplos son irreales. A continuación describimos los pasos a seguir.

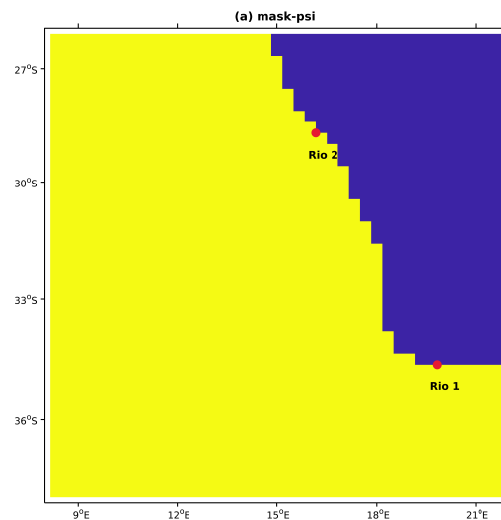


Figura 1: Posición de los ríos para el Caso 1a.

### 2.1. Creación del directorio de trabajo

Crearemos el 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_R1A'
17 #
18 #
19 # END USER SECTION
20 #=====

```

Guardamos los cambios y ejecutamos create\_run.bash

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

y nos dirigimos hacia el directorio de trabajo.

```
1 cd BENGUELA_R1A
```

Es importante mencionar que este es el directorio 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 linux o MATLAB, deben ser ejecutados en este directorio (a menos que se indique lo contrario).

## 2.2. Creando los archivos de entrada

Continuamos con la generación de archivos de entrada. Estos archivos son los mismos archivos netcdf creados en el caso BENGUELA\_LR. Como recordatorio, debemos iniciar 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 ejecutar 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\_R1A/CROCO\_FILES:

```
1 cp ../BENGUELA_LR/CROCO_FILES/croco_grd.nc ./CROCO_FILES
2 cp ../BENGUELA_LR/CROCO_FILES/croco_frc.nc ./CROCO_FILES
3 cp ../BENGUELA_LR/CROCO_FILES/croco_clm.nc ./CROCO_FILES
4 cp ../BENGUELA_LR/CROCO_FILES/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/';`

## 2.3. Modificando las condiciones físicas: cppdefs.h

Para incluir un río (o varios ríos) de flujo constante es necesario modificar `cppdefs.h`:

```
1 nano cppdefs.h
```

Primero, definimos la opción MPI, para así correr la simulación en el servidor NLHPC.

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

Para incluir ríos de flujo constante debemos modificar la sección “Point Sources - Rivers“ (línea 240) donde definiremos la opción PSOURCE

```
1 /* Point Sources - Rivers */
2 # define PSOURCE
3 # undef PSOURCE_NCFILE
4 # ifdef PSOURCE_NCFILE
5 # undef PSOURCE_NCFILE_TS
6 # endif
```

Guardamos los cambios y compilamos

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

### Nota

En caso de incluir más de 100 celdas que representan los ríos (psource), habrá que modificar el código *param.h*, en la sección *! Point sources, Floast, Stations* (en particular en línea 315 o línea 10 en recuadro de abajo.)

```

1  !-----
2  ! Point sources, Floast, Stations
3  !-----
4  !
5  #if defined PSOURCE || defined PSOURCE_NCFILE
6      integer Msrc                ! Number of point sources
7  # ifdef RIVER
8      parameter (Msrc=2)          ! ===== == =====
9  # else
10     parameter (Msrc=100)         ! ===== == =====
11 # endif
12 #endif

```

## 2.4. Propiedades de los ríos

A continuación configuraremos las variables: posición, dirección, sentido, caudal, y valores de trazadores de los ríos. Estas propiedades deben ser ingresadas en el archivo *croco.in*, el cual modificaremos

```
1 nano croco.in
```

cerca del final de este archivo (entre las líneas 256 y 259), se presenta una sección especial para *psource*, donde se nos mostrará : (no confundir con la sección *psource.ncfile* que aparece más abajo).

```

1 psource:  Nsrc  Isrc  Jsrc  Dsrc  Qbar [m3/s]    Lsrc      Tsrc
2           2
3           3    54    1    200.          T T      20. 15.
4           3    40    0    200.          T T      20. 15.

```

donde *Nsrc* es el número de celdas de fuentes (o ríos) a incluir, *Isrc* y *Jsrc* son la posición del río en dirección *xi* y *eta*, respectivamente (o en este caso, zonal y meridional), *Dsrc* es la dirección río, donde 0 es zonal y 1 meridional, *Qbar* es el caudal del río, donde positivo indica sentido de sur/norte (o oeste/este) y negativo un sentido de norte/sur (este/oeste), *Lsrc* indica si los trazadores advectado por el río son estimado de manera analítica ("F") o son entregados por el usuario ("T"), y *Tsrc* los valores de los trazadores. En este caso en *Lsrc* y *Tsrc* la primera y segunda columna corresponde a los trazadores de temperatura y salinidad, respectivamente.

Como mencionamos al inicio del ejercicio, incluiremos dos ríos ( $Nsrc = 2$ ) de caudal de 20000 m<sup>3</sup>/s, donde la boca del primer (R1) y segundo (R2) río se encuentra en 34.65 °S y 19.83 °E, y 28.7°S y 16.16 °E, respectivamente (Figura 1). A partir de la ubicación de los ríos sugerimos que R1 fluye desde el norte hacia el sur, por lo tanto  $Dsrc = 1$  y  $Qbar = -20000$ . En el caso de R2 fluirá de el este al oeste, entonces  $Dsrc = 0$  y  $Qbar = -20000$ . Como deseamos definir los trazadores en la advección de los flujos mantenemos *Lsrc* como aparece originalmente. *Tsrc* muestra los valores de temperatura y salinidad en su primera y segunda columna, respectivamente. Ambos ríos tendrán la misma temperatura y salinidad es decir: 20 (°C) y 2 (psu). Un aspecto un poco más complejo de definir es la posición de los ríos (*Isrc*, *Jsrc*). La explicación en detalle de como escoger la posición es explicada en el Anexo A. En esta parte nos limitaremos a mencionar que la posición del Río 1 (Río 2) es  $Isrc = 36$  y  $Jsrc = 13$  ( $Isrc = 24$  y  $Jsrc = 35$ ). Por lo tanto reemplazando con los valores mencionados anteriormente en *croco.in*:

```

1 psource:  Nsrc  Isrc  Jsrc  Dsrc  Qbar [m3/s]  Lsrc      Tsrc
2           2
3           36   13   1   -20000.      T T      20. 2.
4           24   35   0   -20000.      T T      20. 2.

```

Guardamos los cambios y a continuación enviamos la tarea al servidor

## 2.5. Correr simulación

Con las modificaciones realizadas, corremos la simulación. 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 la simulación, tendremos un archivo *his/avg* y *rst* en *CROCO\_FILES*. Si hubo un error o algún problema, podrá descargar el archivo *avg* en

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R1A/croco_avg.nc
```

El impacto de los ríos podrá ser observado a través del campo de salinidad superficial. Utilizando *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

```

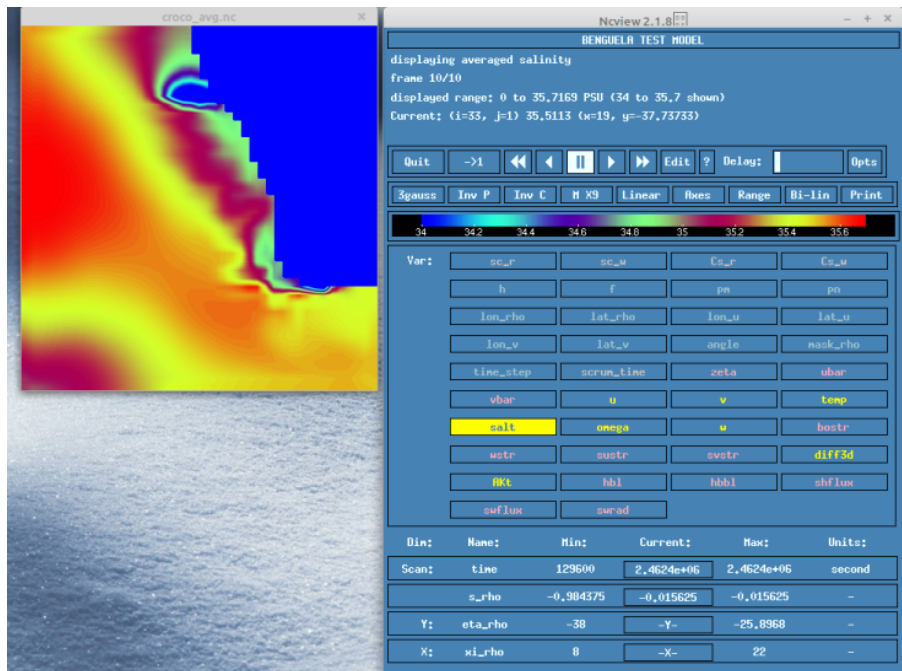


Figura 2: Caso 1a: Salinidad superficial en  $t=10$  obtenido de *croco\_avg.nc* (rango 34-35.7 psu)

Seleccionamos la profundidad más superficial y en la opción *range* (se encuentra en la parte superior del panel) modificamos los límites del gráfico entre 34 y 35.7 psu. En el último registro temporal observamos claramente el impacto de los dos ríos (Figura 2). Sin embargo, se debe recordar que esta situación no es realista ya que se sobreestimó el caudal de estos ríos.

### 3. Caso 2a: Ríos con flujo variable

Continuando con el caso Benguela, realizaremos una simulación que incluye ríos con flujo variable, donde definiremos los valores de los trazadores que advecta los ríos. La información de la variabilidad del flujo del río se obtendrá la base de datos de Dai & Trenberth (2002) y los valores de trazadores se obtienen de los datos `croco_clim.nc`. A continuación describimos los pasos a seguir.

#### 3.1. Creación del directorio de trabajo

Crearemos un nuevo directorio donde trabajaremos en este ejercicio. Si es que en la terminal se encuentran en el directorio de la simulación anterior (BENGUELA\_R1A):

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

Y editamos el nombre del directorio (verificando que las variables `CROCO_DIR`, `SOURCES_DIR` y `TOOLS_DIRS` presentan la misma configuración mostrada la sección 2.1) :

```
1 # Name of the configuration directory defined by the user
2 #
3 MY_CONFIG_NAME='BENGUELA_R2A'
```

y nos dirigimos hacia al directorio

```
1 cd BENGUELA_R2A
```

Este es el directorio en que trabajaremos, por lo tanto, las indicaciones que se describen a continuación (en la terminal de linux o MATLAB) deben ser ejecutadas en este directorio.

#### 3.2. Creando archivos de entrada

Para este ejercicio los archivos de entrada `croco_grd.nc`, `croco_frc.nc`, `croco_clm.nc` y `croco_ini.nc` son los mismos que hemos utilizado en los ejercicios anteriores (ver Sección 2.2), y podemos copiar estos archivos al directorio `CROCO_FILES`:

```
1 cp ../BENGUELA_R1A/CROCO_FILES/croco_grd.nc ./CROCO_FILES
2 cp ../BENGUELA_R1A/CROCO_FILES/croco_frc.nc ./CROCO_FILES
3 cp ../BENGUELA_R1A/CROCO_FILES/croco_clm.nc ./CROCO_FILES
4 cp ../BENGUELA_R1A/CROCO_FILES/croco_ini.nc ./CROCO_FILES
```

Sin embargo, en este caso necesitamos un nuevo archivo de entrada: **croco\_runoff.nc**. Este archivo contiene la información sobre los ríos a incluir en la simulación. Para generar este archivo utilizaremos la rutina **make\_runoff** para MATLAB. Esta rutina forma parte de los `croco.tools`, particularmente en el directorio `Rivers`.

En `crocotools.param.m`, podemos modificar el nombre del archivo de `netcdf` con la información de los ríos (línea 178 o línea 6 en el recuadro inferior)

```

1 % CROCO file names (grid, forcing, bulk, climatology, initial)
2 %
3 ..
4 ..
5 ...
6 rivname = [CROCO_files_dir, 'croco_runoff.nc'];

```

Y definir el directorio donde se encuentra la base de datos de los caudales de los ríos (*global\_clim\_riverdir*) y el nombre del archivo (*global\_clim\_rivername*) (líneas 275 - 278):

```

1 % Runoff monthly seasonal climatology (Dai and Trenberth)
2 %
3 global_clim_riverdir=[DATADIR, 'RUNOFF_DAI/'];
4 global_clim_rivername=[global_clim_riverdir, 'Dai_Trenberth_runoff_global_clim.nc'];

```

El archivo *Dai\_Trenberth\_runoff\_global\_clim.nc* está incluido en los *croco\_tools*, y será la base de datos que utilizaremos para este ejercicio. Por lo tanto no realizamos modificaciones en *crocotools\_param.m*

### IMPORTANTE

Antes de continuar, es necesario modificar *make\_runoff.m*, debido a un error asociado a la generación de figuras de salidas (línea 666 en la rutina). Para esto nos dirigiremos al directorio donde se encuentra los *croco\_tools/Rivers*

```
1 cd ../croco_tools/Rivers
```

Removemos *make\_runoff.m*

```
1 rm make_runoff.m
```

Y reemplazamos por la nueva versión

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/make_runoff.m
```

Notar que en la nueva versión de la rutina simplemente se comento la línea 666.  
Regresamos al directorio en que estamos trabajando

```
1 cd ../../BENGUELA_R2A
```

Y continuamos con el tutorial

A continuación, modificaremos *make\_runoff* con la configuración que deseamos:

```
1 nano ../croco_tools/Rivers/make_runoff.m
```

En la primera sección, podemos informar si la simulación es anidada online (*gridlevel=1*), para así generar los archivos de la simulación hijo (e.g. *croco\_runoff.nc.1*). Como no es una simulación anidada, entonces en esta sección no realizamos cambios (*gridlevel=0*):

```

1 % Choose the grid level into which you ant to set up the runoffs
2
3 gridlevel=0

```

En la segunda parte, seleccionamos si el caudal de los ríos es una serie de tiempo climatológica (`clim_run=1`) o no (`clim_run=0`). Para este caso elegimos definimos la opción climatológica

```
1 % Choose the monthly runoff forcing time and cycle in days
2 clim_run=1
```

A continuación configuramos la opción para definir los valores/concentración de los trazadores (`psource_ncfile_ts=1`, en caso de no definir los valores de los trazadores `psource_ncfile_ts=0`). También podemos definir si queremos que los trazadores sean obtenido de manera automática (`psource_ncfile_ts_aut=1`), es decir, que los datos de trazadores se extraen del archivo de entrada `croco_clm.nc`. Esta es la opción que escogeremos. La otra opción es que los datos de los trazadores sean ingresados manualmente (`psource_ncfile_ts_aut=0`). Por lo tanto

```
1 % Choose if you process variable tracer concentration(temp, salt, NO3, ...)
2
3 psource_ncfile_ts=1;
4
5 if psource_ncfile_ts
6     psource_ncfile_ts_auto=1 ;
7     psource_ncfile_ts_manual=0;
8 end
```

Existen otras opciones, como por ejemplo si queremos que realicen otras figuras, como también definir manualmente el sentido y dirección,

```
1 %=====
2 % Fancy plots
3
4 plotting=1;
5 plotting_zoom=1;
6 %
7 %=====
8 % Add biogeochemical variables
9 if (makenpzd | makepisces | makebioebus)    makebio = 1;
10 else
11     makebio = 0;
12 end
13 %%
14 disp(' ')
15 disp(['Create runoff forcing from Dai and Trenberth's global monthly climatological run-off dataset'])
16 disp(' ')
17 title_name='runoff forcing file (Dai and Trenberth, 2002 dataset)'
18
19 %=====
20 define_dir=0 ; %%->flag to define directly the orientation / direction of the runoff
21 %
```

pero por ahora mantenemos las opciones por default. Adicionalmente en el script, modificaremos la definición de la salinidad. El script `make_runoff.m` original, obtiene la salinidad de `croco_clm.nc` en la posición del río y le sustrae 10 psu ( $S$ , línea 27 del recuadro inferior; en `make_runoff.m` ver la sección donde se procesa los trazadores definidos automáticamente (`psource_ncfile_ts_auto=1`), línea 500).



```

1      if psource_ncfile_ts_auto
2          %%=====
3          ncclim=netcdf(clmname);
4          N=length(ncclim('s_rho'));
5
6          for k0=1:number_rivertoprocess
7              k=rivertoprocess(k0);
8              %%
9              %% For temperature, use the closest surface point in the clim file
10             %% to reduce any heat flux induced by the rivers
11             %%
12             T=squeeze(ncclim{'temp'}(:,N,J(k)+1,I(k)+1));
13             my_temp_src0(k,:)=T';
14             %%
15             %% For salinity ... ?
16             %%
17             S=squeeze(ncclim{'salt'}(:,N,J(k)+1,I(k)+1))-10; % hum...
18             S(S<2)=2; % to prevent negative salinities in the
19                 % equation of state
20             disp([' Use psource_ncfile_ts_auto using S = scim -10 '])
21             disp([' Check line 464 in make_runoff.m to change ' ...
22                 'this arbitrary runoff salinity'])
23             %%S=2;
24             my_salt_src0(k,:)=S';
25             ...

```

Con el propósito de incrementar el impacto de los ríos re-definiremos la salinidad, la cual corresponderá a la salinidad de croco\_clm.nc pero sustraeremos 30 psu (línea 27 del recuadro inferior).

```

1      if psource_ncfile_ts_auto
2          %%=====
3          ncclim=netcdf(clmname);
4          N=length(ncclim('s_rho'));
5
6          for k0=1:number_rivertoprocess
7              k=rivertoprocess(k0);
8              %%
9              %% For temperature, use the closest surface point in the clim file
10             %% to reduce any heat flux induced by the rivers
11             %%
12             T=squeeze(ncclim{'temp'}(:,N,J(k)+1,I(k)+1));
13             my_temp_src0(k,:)=T';
14             %%
15             %% For salinity ... ?
16             %%
17             S=squeeze(ncclim{'salt'}(:,N,J(k)+1,I(k)+1))-30; % hum...

```

En caso de querer modificar manualmente los valores de la temperatura y salinidad ver Anexo B. Guardamos los cambios realizados 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

```

y ejecutamos el archivo start.m y posteriormente make\_runoff.m:

```

1 start
2 make_runoff

```

### 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/'`;

Esta rutina es interactiva. Al inicio, te mostrarán dos figuras: (i) La ubicación de los ríos disponibles en la base de datos de Dai,& Trenberth (2002) para la zona del dominio (Figura 5) y (ii) series de tiempo del caudal de los ríos presente obtenidos de la base de datos de Dai,& Trenberth (2002) (Figura 6). Cuando aparezca estas figuras, presionamos 'enter' en la terminal de MATLAB. Luego te preguntará que ríos deseas incluir en la simulación. En este caso incluiremos los dos ríos que están presente en la zona, para lo cual confirmamos escribiendo 1 y presionando .enter" por cada río que queremos incluir

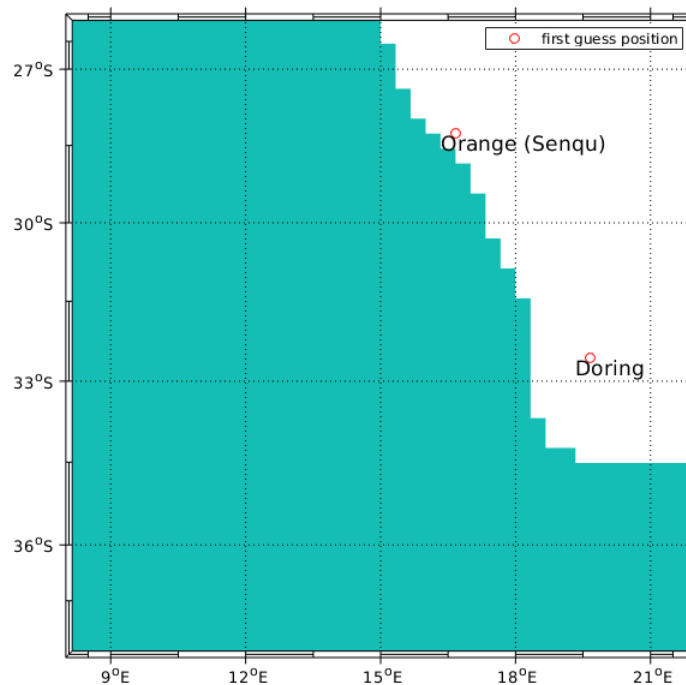


Figura 3: Ubicación de los ríos disponible en el dominio. Figura emitada al inicio de ejecutar make\_runoff.

```

1 Do you want to use river (Yes[1], No[0]) ? Orange (Senqu)      1
2 Do you want to use river (Yes[1], No[0]) ? Doring           1
3 =====

```

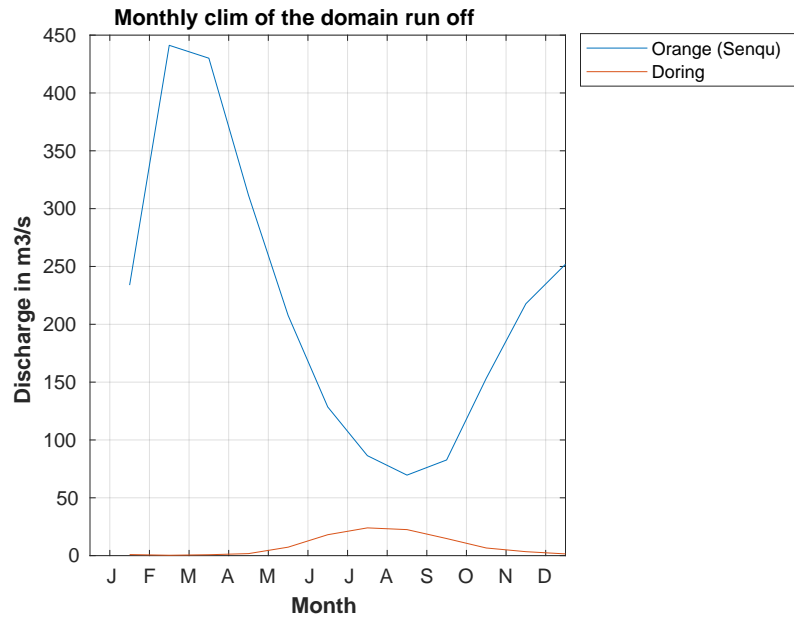


Figura 4: Serie de tiempo del caudal de los ríos disponible en el dominio. Figura emitada durante make\_runoff.

De inmediato te preguntarán sobre la orientación del flujo del río (0 para zonal, 1 para meridional).

```

1 =====
2 River Orange (Senqu)
3 Choose the orientation of the flow
4 0=zonal or 1=meridional. 0

```

Por la ubicación del río Orange seleccionamos zonal (ver Figura 5), es decir 0 y presionamos enter. Ahora te preguntarán por el sentido del río Orange, si este/oeste (-1) u oeste/este (1), donde seleccionamos la opción -1

```

1 Choose the direction of the flow.
2 1 is positive [S-N or W-E], -1 negative [N-S or E-W]. -1
3 k=1
4 k0=1
5 dir(k,:)=0 -1

```

Como seleccionamos dos ríos, ahora nos falta completar la información para el segundo (Doring), donde escogemos la opción zonal (0) y flujo este/oeste (-1)

```

1 River Doring
2 Choose the orientation of the flow
3 0=zonal or 1=meridional. 0
4 Choose the direction of the flow.
5 1 is positive [S-N or W-E], -1 negative [N-S or E-W]. -1
6 k=2
7 k0=2
8 dir(k,:)=0 -1

```

A continuación se creará en CROCO\_FILES el archivo croco\_runoff.nc, el cual contiene la información de la serie de tiempo del caudal, temperatura y salinidad. En caso de tener problemas para generar este archivo, podrá ser descargado en

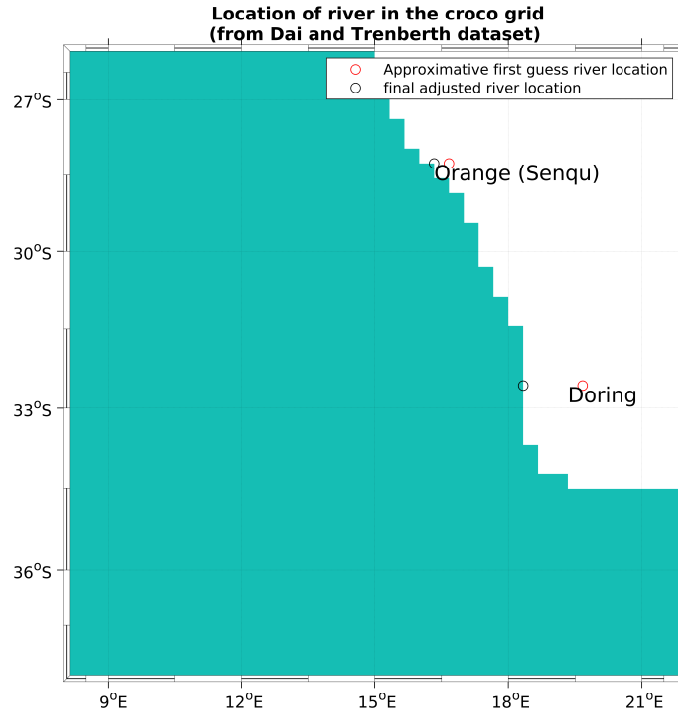


Figura 5: Ubicación de los ríos disponible en el dominio. Figura emitada al finalizar de ejecutar make\_runoff.

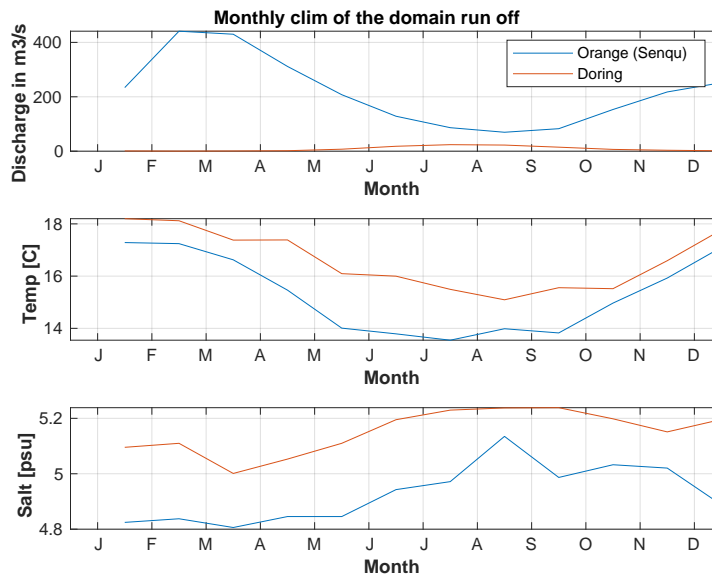


Figura 6: Serie de tiempo del caudal, temperatura y salinidad de los ríos disponible en el dominio. Figura emitada al finalizar de ejecutar make\_runoff.

```
1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R2A/croco_runoff.nc
```

Además, al finalizar la rutina, la figura del dominio con la ubicación de los ríos se habrá actualizado con la posición final de los ríos (Figura 7), y a la vez te mostrará otra figura con la serie de tiempo del caudal, temperatura y salinidad (Figura 8). Si no hubieron errores, el mensaje final debería ser

```
1 psource_ncfile:  Nsrc  Isrc  Jsrc  Dsrc  qbardir  Lsrc  Tsrc  runoff file name
2                  CROCO_FILES/croco_runoff.nc(.#nestlevel)
3                  2
4                  25  34  0  -1  30*T   15.3139  4.928
5                  31  19  0  -1  30*T   16.5999  5.1513
```

Este mensaje debe ser copiado y pegado en la sección de psource\_ncfile en croco.in. Por lo tanto, (si es que aún se encuentra en la terminal de MATLAB), cerramos MATLAB:

```
1 exit
```

Y modificamos croco.in

```
1 nano croco.in
```

y pegamos (o reemplazamos) en croco.in la información entregada en MATLAB (mostrada en el recuerdo anterior)

```
1 psource_ncfile:  Nsrc  Isrc  Jsrc  Dsrc  qbardir  Lsrc  Tsrc  runoff file name
2                  CROCO_FILES/croco_runoff.nc
3                  2
4                  25  34  0  -1  30*T   15.3139  4.928
5                  31  19  0  -1  30*T   16.5999  5.1513
```

Note, que en croco.in no debe aparecer (.#nestlevel) que se muestra en el mensaje de MATLAB. Mencionar que Nsrc es el número de ríos a incluir, Isrc y Jsrc es la posición de los ríos, Dsrc es el sentido, qbardir es la dirección, Lsrc define si los trazadores son entregados por el usuario o analíticamente (True o False, respectivamente), Tsrc son las magnitudes promediadas en el tiempo de los trazadores (temperatura y salinidad en la primera y segunda columna, respectivamente).

### 3.3. Compilando

Previo a compilar, debemos modificar cppdefs.h

```
1 nano cppdefs.h
```

A diferencia del ejercicio anterior, en el cppdefs.h NO debemos definir la opción PSOURCE y definimos la opción PSOURCE\_NCFILE (para que considere un flujo variable) y PSOURCE\_NCFILE\_TS (para que considere la definición de los trazadores).

```
1          /* Point Sources - Rivers */
2  # define PSOURCE_NCFILE
3  # ifdef PSOURCE_NCFILE
4  #   define PSOURCE_NCFILE_TS
5  # endif
```

Recuerde que para correr la simulación en el servidor NLHPC, es necesario que la opción MPI este definida en cppdefs.h

```

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

```

Guardamos los cambios y compilamos

```

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

```

### 3.4. Corriendo la simulación

En la sección 3.2 ya habíamos modificados el archivo croco.in, sin embargo, a continuación volvemos a modificar,

```

1 nano croco.in

```

con el propósito de ver el impacto de la variabilidad anual del flujo de los ríos, correremos la simulación por 1 año climatológico (equivalente a 360 días en el modelo). Dado que  $dt=3600$ , entonces necesitamos 8640 ( $= 360*24$ ) pasos para simular un año

```

1 title:
2       BENGUELA TEST MODEL
3 time_stepping: NTIMES  dt[sec]  NDTFAST  NINFO
4               8640      3600      60      1

```

Ahora, corremos la simulación por lo cual copiamos run\_nlhpc.bash desde el directorio de trabajo anterior al actual

```

1 cp ../BENGUELA_R1A/run_nlhpc.bash

```

Y enviamos la tarea al servidor:

```

1 sbatch run_nlhpc.bash

```

El resultado será la creación de los archivos de salida (croco\_avg.nc y croco\_his.nc) en el directorio CROCO\_FILES. En caso de no haber generado los archivos, se podrán descargar el archivo avg:

```

1 wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial08/BENGUELA_R2A/croco_avg.nc

```

El impacto de los ríos en la región se puede observar a través de ncview (Figura 9):

```

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

```

A diferencia de los ejercicios presentado anteriormente (Figuras 2 y 3), en este caso el impacto no se observa claramente, debido al bajo caudal. Figura 9 muestra la salinidad superficial para la fecha de mayor caudal del río Orange (marzo), y se muestra su relativo impacto. Río Doring, por su menor caudal, no es posible observar su impacto.

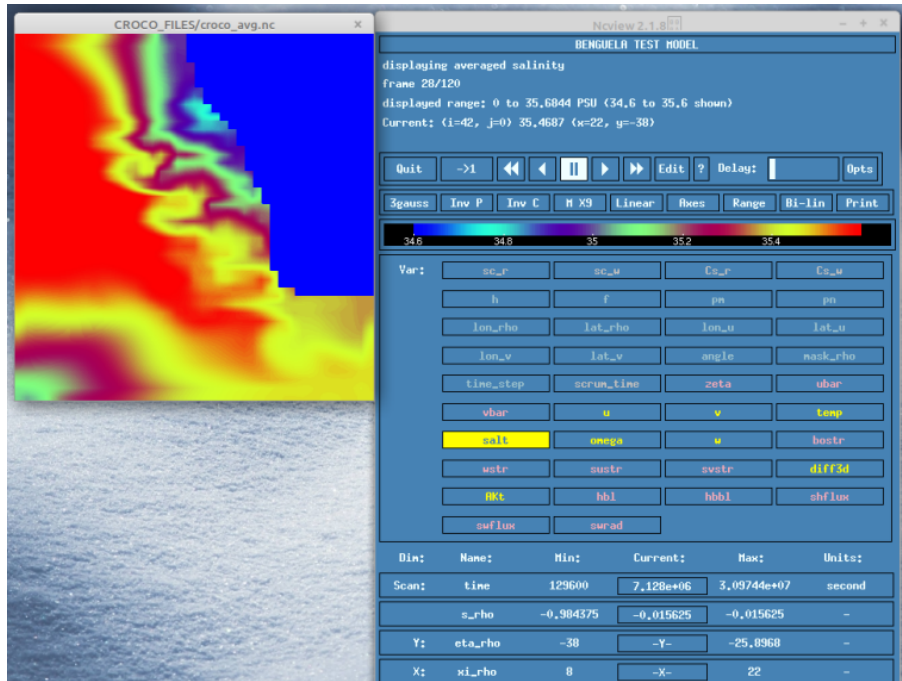


Figura 7: Caso 2a: Salinidad superficial en  $t=28$  obtenido de croco\_avg.nc (rango 34.6-35.6 psu)

## 4. Trabajo Avanzado

Realice una simulación para un dominio de su interés, el cual incluya por lo menos un río de flujo constante y flujo variable. Analice las variables de salinidad, temperatura, velocidad y anomalía del nivel del mar.

## 5. Conclusiones

Al finalizar esta guía, podrán implementar en sus simulaciones ríos, y modificar sus principales propiedades

Para más información:

Andrés Sepúlveda (asepulveda@dgeo.udec.cl)

Desarrollado por:

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

Contribuciones de:

Mauro Santiago

Ornella Silvestri



## 6. Referencias

Dai, A., & Trenberth, K. E. (2002). Estimates of freshwater discharge from continents: Latitudinal and seasonal variations. *Journal of hydrometeorology*, 3(6), 660-687.

# ANEXOS

## A. Posición de los ríos

Un aspecto un poco más complejo de definir es la posición de los ríos en la grilla (Isrc,Jsrc). La posición debe estar en la cara U/V (según el flujo) y en el borde entre tierra y mar (es decir, la línea de costa). Entonces, para encontrar la posición es recomendable observar la máscara en coordenadas psi (*mask\_psi* en *grd.nc*) ya que su posición corresponde al borde la celda y no al centro (como es el caso de coordenadas rho).

Para ejemplificar la búsqueda de la posición: Si tenemos un flujo meridional de norte a sur en  $34.65^{\circ}\text{S}$  y  $19.83^{\circ}\text{E}$  (como es el caso de Río 1), entonces, primero buscando la posición de la longitud en la grilla psi (*mask\_psi* en *croco\_grd.nc*). Para el dominio de BENGUELA\_LR es la posición 36. Dado que el flujo es meridional, entonces atravesará en el borde mar/tierra de la cara v. Entonces debemos buscar la posición más hacia el sur de la celda que corresponde a tierra (*mask\_psi*=0). Esta posición corresponde a 13 (Ver Figura A.1)

En caso del Río 2 el flujo fluye de este a oeste, entonces buscamos la posición de la latitud (posición 34). Como fluye zonalmente, entonces el flujo ingresa por la cara-u buscamos la celda que se encuentre más hacia el oeste y corresponda a la máscara de tierra (posición 35; ver Figura A.1).

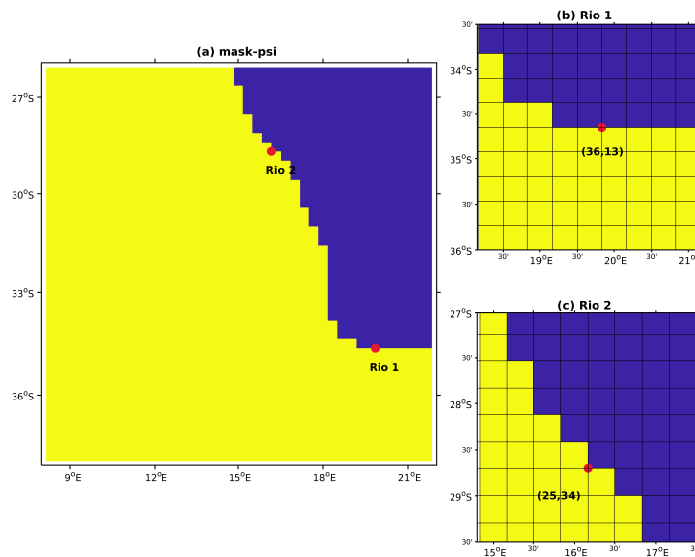


Figura A1: (a) Máscara en coordenadas psi donde se muestra la posición de los ríos 1 y 2. (b) y (c) Zoom de Río 1 y 2, respectivamente

A continuación se presenta un ejemplo de rutina para MATLAB para encontrar la posición de los ríos presentado en este documento.

```

1 clear all
2 cd croco/BENGUELA_R1A
3 crocotools_param
4
5 maskp=ncread(grdname,'mask_psi')';
6 latp=ncread(grdname,'lat_psi')';
7 lonp=ncread(grdname,'lon_psi')';
8
9 posR1=[19.83 -34.65 ];
10 posR2=[16.16 -28.7]
11
12 %%
13 % RIO 1: Flujo de norte a sur. Se debe encontrar la cara v por donde fluye
14 % el flujo
15
16 % % Buscando Longitud
17 posxR1=find(lonp(1,:)>=posR1(1)-.1 & lonp(1,:)<=posR1(1)+.1)
18
19 % % Buscando latitud en que se encuentra el limite de la linea de costa, es
20 % decir el punto de las mascara mas al sur. Si el flujo fuera de norte a sur
21 % entonces se buscaría la posición hacia el norte, entonces
22 posyR1=min(find(maskp(:,posxR1)==0))
23
24 %%
25 % RIO 2: Flujo de este a oeste. Se debe encontrar la cara u por donde
26 % fluye el flujo
27 posyR2=find(latp(:,1)>=posR2(2)-.1 & latp(:,1)<=posR2(2)+.1)
28
29 % % Buscando latitud en que se encuentra el limite de la linea de costa, es
30 % decir el punto de las mascara mas al oeste. Si el flujo fuera de oeste a este,
31 % entonces se buscaria la posición más hacia el este, y entonces posxR2=max(find(...
32 posxR2=min(find(maskp(posyR2,:)==0))

```

## B. Caso 1b: Ríos de flujo constante con cambio de distribución del flujo vertical

Hasta este momento no hemos mencionado sobre la configuración de la distribución del flujo vertical (*Qshape*). En este ejercicio correremos el mismo ejercicio anterior, pero modificando *Qshape*. Para esto, crearemos un nuevo directorio pero mantendremos la misma configuración utilizada en la sección anterior.

### B.1. Creación del directorio de trabajo

Para ese caso, utilizaremos la misma configuración y archivos de entrada que en el caso de la sección 2. Para simplificar, copiaremos el directorio de dicho caso

```
1 cp -r BENGUELA_R1A BENGUELA_R1B
```

### B.2. Modificando *Qshape* y Compilando

A continuación configuramos la distribución vertical del flujo. Para esto modificamos el código *analytical.F* que se encuentra en el directorio del código de CROCO (i.e., *croco/OCEAN*). Dado que este archivo es parte del código original, antes de modificar, es necesario respaldar:

```
1 cp OCEAN/analytical.F OCEAN/analytical_respaldo.F
```

Ahora, estamos en condiciones de modificar analytical.F

```
1 nano OCEAN/analytical.F
```

Este archivo es un conjunto de subrutina en las cuales se proporcionar varios campos analíticos al modelo cuando son requerido, como por ejemplo la distribución de vertical del flujo. Buscamos la subrutina ana\_psourc (línea 1759), donde se define la parte analítica de los ríos. Esta rutina comienza de la siguiente forma:

```
1 !=====
2 !           subroutine ana_psourc
3 !=====
4 !
5 #if defined PSOURCE && defined ANA_PSOURCE
6 !
7 !-----
8 ! Set analytical tracer and mass point sources and sinks
9 !-----
10 !
11     subroutine ana_psourc_tile (Istr,Iend,Jstr,Jend)
12     implicit none
```

Buscamos donde se muestra la configuración para el caso regional (línea 1883) y encontraremos:

```
1 # elif defined REGIONAL
2 # define CST_SHAPE
3 # ifdef CST_SHAPE
4     cff=1./float(N)
5     do k=1,N                ! Uniform vertical
6         do is=1,Nsrc        ! distribution
7             Qshape(is,k)=cff
8         enddo
9     enddo
10 # elif defined EXP_SHAPE
11     do is=1,Nsrc
```

La opción por default es CST\_SHAPE. (# define CST\_SHAPE; línea 2 en el recuadro superior). Esta opción distribuye el flujo en la vertical de manera uniforme. La otra opción disponible para casos regionales es EXP\_SHAPE, donde el flujo es distribuido exponencialmente. Esta última opción es con la cual trabajaremos y definiremos en analytical.F:

```
1 # elif defined REGIONAL
2 # define EXP_SHAPE
3 # ifdef CST_SHAPE
4     cff=1./float(N)
5     do k=1,N                ! Uniform vertical
6         do is=1,Nsrc        ! distribution
7             Qshape(is,k)=cff
8         enddo
9     enddo
10 # elif defined EXP_SHAPE
11     do is=1,Nsrc
```

Guardamos los cambios y nos dirigimos al directorio BENGUELA\_R1B:

```
1 cd BENGUELA_R1B
```

y compilamos

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

### B.3. Corriendo la simulación

Enviamos la tarea al servidor:

```
1 sbatch run_nlhpc.bash
```

Y similar al ejercicio, anterior, en CROCO\_FILES encontraremos los archivos de salida croco\_his/avg.nc. El impacto del cambio de la distribución vertical es posible observarlo 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
```

donde seleccionamos la salinidad superficial (utilizando un rango entre 34-35.7 psu; Figura A2), donde se muestran diferencia con respecto al caso anterior (Figura 2)

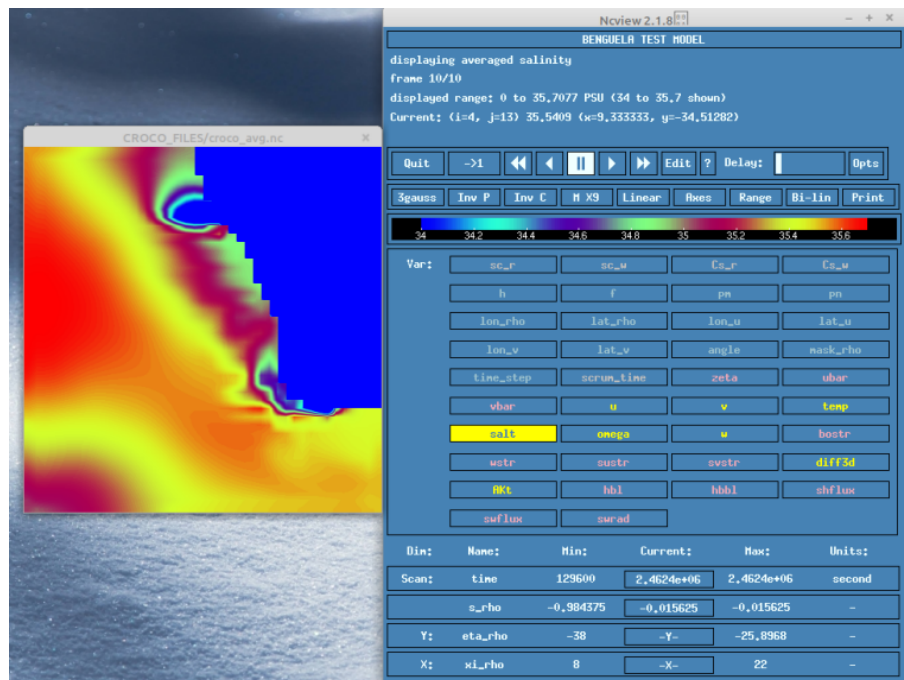


Figura A2: Caso 1b: Salinidad superficial en  $t=10$ , obtenido de croco\_avg.nc (rango 34-35.7 psu)

Adicionalmente, Figura A3 muestra secciones verticales de la salinidad, perpendicular a la boca de los ríos 1 y 2, donde se observa las diferencias entre usar una distribución constante en la vertical (CST\_SHAPE) y exponencial (EXP\_SHAPE).

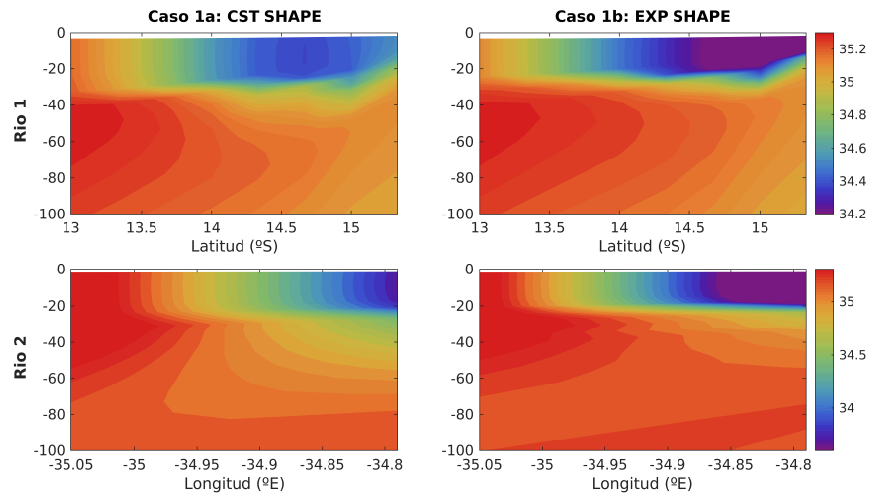


Figura A3: Secciones verticales de la salinidad, perpendicular a la boca de Rio 1 (arriba) y 2 (abajo). Izquierda (Derecha), Sección utilizando configuración CST\_SHAPE (EXP\_SHAPE)

### Nota

Se recomienda modificar analytical.F para mantener la configuración original:

```
1 nano ../OCEAN/analytical.F
```

Buscamos la subrutina ana\_psource, y en la sección para casos regionales (línea 1883) modificamos # define EXP\_SHAPE a # define CST\_SHAPE,

```
1 # elif defined REGIONAL
2 # define CST_SHAPE
3 # ifdef CST_SHAPE
4     cff=1./float(N)
5     do k=1,N                ! Uniform vertical
6         do is=1,Nsrc        ! distribution
7             Qshape(is,k)=cff
8         enddo
9     enddo
10 # elif defined EXP_SHAPE
11     do is=1,Nsrc
```

## C. Caso 2b: Ríos con flujo variable y modificaciones a trazadores

En el caso de utilizar otra base de datos para los trazadores advectados por los ríos (y no utilizar datos de croco\_clm.nc) podemos modificar make\_runoff para ingresar la información que deseamos.

### C.1. Creación del directorio de trabajo

Crearemos un nuevo directorio donde trabajaremos en este ejercicio. Si estamos en el directorio anterior de trabajo:

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

Y editamos el nombre del directorio (verificando que las variables CROCO\_DIR, SOURCES\_DIR y TOOLS\_DIRS presentan la misma configuración mostrada la sección 3.1) : :

```
1 # Name of the configuration directory defined by the user
2 #
3 MY_CONFIG_NAME='BENGUELA_R2B'
```

y nos dirigimos hacia la carpeta

```
1 cd BENGUELA_R2B
```

## C.2. Creando archivos de entrada

En este caso utilizamos los archivos de entrada de la sección anterior (croco\_grd.nc, croco\_frc.nc, croco\_clm.nc y croco\_ini.nc), por lo cual copiamos los archivos a la carpeta actual:

```
1 cp ../BENGUELA_R1A/croco_grd.nc ./CROCO_FILES
2 cp ../BENGUELA_R1A/croco_frc.nc ./CROCO_FILES
3 cp ../BENGUELA_R1A/croco_clm.nc ./CROCO_FILES
4 cp ../BENGUELA_R1A/croco_ini.nc ./CROCO_FILES
```

Ahora, modificamos make\_runoff.m, para poder ingresar los valores de temperatura y salinidad que deseamos

```
1 nano make_runoff
```

Mantemos la misma configuración de las primeras líneas escogidas en la sección anterior, es decir:

```

1 % Choose the grid level into which you ant to set up the runoffs
2
3 gridlevel=0
4 if ( gridlevel == 0 )
5     % #-> Parent / zoom #0
6     grdname = [CROCO_files_dir,'croco_grd.nc'];
7     rivname = [CROCO_files_dir,'croco_runoff.nc'];
8     clmname = [CROCO_files_dir,'croco_clm.nc']; % <- climato file for runoff
9 else
10    % # -> Child / zoom #XX
11    grdname = [CROCO_files_dir,'croco_grd.nc.',num2str(gridlevel)];
12    rivname = [CROCO_files_dir,'croco_runoff.nc.',num2str(gridlevel)];
13    clmname = [CROCO_files_dir,'croco_clm.nc.',num2str(gridlevel)]; % <- climato file for runoff
14 end
15
16 % Choose the monthly runoff forcing time and cycle in days
17 clim_run=1
18
19 if (clim_run == 1)
20     qbar_time=[15:30:365];
21     qbar_cycle=360;
22 else
23     qbar_time=[15.2188:30.4375:350.0313];
24     qbar_cycle=365.25;
25 end
26
27 % - times and cycles for runoff conditions:
28 % - clim_run = 1 % climato forcing experiments with climato calendar
29 %     qbar_time=[15:30:365];
30 %     qbar_cycle=360;
31 %
32 % - clim_run = 0 % interannual forcing experiments with real calendar
33 %     qbar_time=[15.2188:30.4375:350.0313];
34 %     qbar_cycle=365.25;

```

Pero a continuación desactivamos `psorce_ts_auto` y activamos `psorce_ts_manual`, es decir:

```

1 % Choose if you process variable tracer concentration(temp, salt, NO3, ...)
2
3 psource_ncfile_ts=1;
4
5 if psource_ncfile_ts
6     psource_ncfile_ts_auto=0 ;
7     psource_ncfile_ts_manual=1;
8 end

```

Luego, en el mismo archivo buscamos la sección del código donde comienza el procesamiento de los datos de los trazadores (línea 484) para ser considerado en el flujo del ríos (o de los ríos). En la línea 494 el código mostrará la opción cuando `psource_ncfile_ts_auto=1` (es decir la temperatura y salinidad se obtienen del archivo de entrada `croco_clm.nc`).



```

1  if psource_ncfile_ts_auto
2      %=====
3      ncclim=netcdf(clmname);
4      N=length(ncclim('s_rho'));
5      .
6      ..
7      ...

```

Pero, más abajo (línea 528) comienza el procesamiento para valores de temperatura (*my\_temp\_src0*) y salinidad (*my\_salt\_src0*) a nuestra elección (*psource\_ncfile\_ts\_manual=1*).

```

1  elseif psource_ncfile_ts_manual
2
3      %% Alternativly : Define all manually the tracer
4      %% t, s, and eventually biogeochemical tracer concentration
5
6      %% my_temp/salt/_src0 : line are rivers / rows are month
7      %% size(my_temp_src0)=[number_rivertoprocess 12]
8      %% my_temp/salt/_src : line are rivers effectivelly processed / rows are month
9
10     %% Example for GIGATL FAMILY
11     my_temp_src0(:, :)=ones(rivernumber,12)*10;
12     .
13     ..
14     ...
15     ..
16     .
17     my_salt_src0(:, :)=ones(rivernumber,12)*5;
18

```

Reemplazamos los valores que muestra el recuadro superior(o pueden crear sus propios valores). Note que *my\_temp\_src0* son los datos de temperatura, donde cada fila muestra la información de un río en específico y las columnas muestra los datos temporales. En este caso *my\_temp\_src0* tiene dimensiones  $2 \times 12$ , donde ya dos ríos y hay 12 datos temporales (1 dato por mes). La salinidad (*my\_salt\_src0*) se mantendrá constante con una concentración de 5 psu.

### Recordatorio

La salinidad no debe ser menor de 2 psu, para así evitar que la ecuación de estado entregue valores negativos!.

Guardamos los cambios y he 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 cargamos las rutinas de *croco\_tools* y ejecutamos *make\_runoff*

```

1  start
2  make_runoff

```

Se comenzará a ejecutar la rutina y realizará las mismas preguntas de la sección anterior, donde incluiremos los mismos ríos y el mismo sentido y dirección que en la sección 2a:

```

1 Do you want to use river (Yes[1], No[0]) ? Orange (Senqu)      1
2 Do you want to use river (Yes[1], No[0]) ? Doring             1
3 =====
4 River Orange (Senqu)
5 Choose the orientation of the flow
6 0=zonal or 1=meridional. 0
7 Choose the direction of the flow.
8 1 is positive [S-N or W-E], -1 negative [N-S or E-W]. -1
9 k=1
10 k0=1
11 dir(k,:)=0 -1
12 =====
13 River Doring
14 Choose the orientation of the flow
15 0=zonal or 1=meridional. 0
16 Choose the direction of the flow.
17 1 is positive [S-N or W-E], -1 negative [N-S or E-W]. -1
18 k=2
19 k0=2
20 dir(k,:)=0 -1
21
22 Create the runoff file...

```

Y al término de la rutina tendremos en CROCO\_FILES el archivo croco\_runoff.nc junto con las figuras finales. En caso de no haber errores, aparecerá

```

1 psource_ncfile:  Nsrc  Isrc  Jsrc  Dsrc  qbardir  Lsrc  Tsrc  runoff file name
2                  CROCO_FILES/croco_runoff.nc(.#nestlevel)
3
4                  2
5                  25  34  0  -1   30*T   10  5
6                  31  19  0  -1   30*T   10  5

```

(si usted utilizó otros valores para temperatura y salinidad entonces es de esperar que los valores de las últimas dos columnas sean diferentes). Modificamos croco.in

```

1 nano croco.in

```

y copiamos el mensaje entregado en MATLAB (cerramos MATLAB) y lo copiamos en croco.in (sin incluir (.#nestlevel))

```

1 psource_ncfile:  Nsrc  Isrc  Jsrc  Dsrc  qbardir  Lsrc  Tsrc  runoff file name
2                  CROCO_FILES/croco_runoff.nc
3
4                  2
5                  25  34  0  -1   30*T   10  5
6                  31  19  0  -1   30*T   10  5

```

### C.3. Compilando

Utilizaremos la misma configuración mencionadas para el caso de la sección 3, es decir, modificamos cppdefs.h

```

1 nano cppdefs.h

```

y definimos la opción MPI

```

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

```

Además, la sección "Point Sources - Rivers" debe estar configurada como

```

1      /* Point Sources - Rivers */
2 # undef  PSOURCE
3 # define  PSOURCE_NCFILE
4 # ifdef  PSOURCE_NCFILE
5 #   define  PSOURCE_NCFILE_TS
6 # endif

```

Guardamos los cambios y compilamos

```

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

```

## C.4. Corriendo simulación

Tal como se hizo en la sección 3.4, modificamos nuevamente croco.in con el objetivo de simular un año climático:

```

1 title:
2     BENGUELA TEST MODEL
3 time_stepping: NTIMES  dt[sec]  NDTFAST  NINFO
4                 8640      3600      60      1

```

Guardamos los cambios y corremos la simulación. Copiamos en el directorio la rutina run\_nlhpc.bash

```

1 cp ../BENGUELA_R1A/run_nlhpc.bash

```

y ejecutamos la rutina

```

1 sbatch run_nlhpc.bash

```

Al finalizar se habrá generados los archivos croco\_avg.nc y croco\_his.nc.

El impacto de los ríos en el océano es posible observarlo principalmente en el río Orange (por su mayor caudal) en marzo (Figura 10). Esto es posible de visualizarlo 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

```

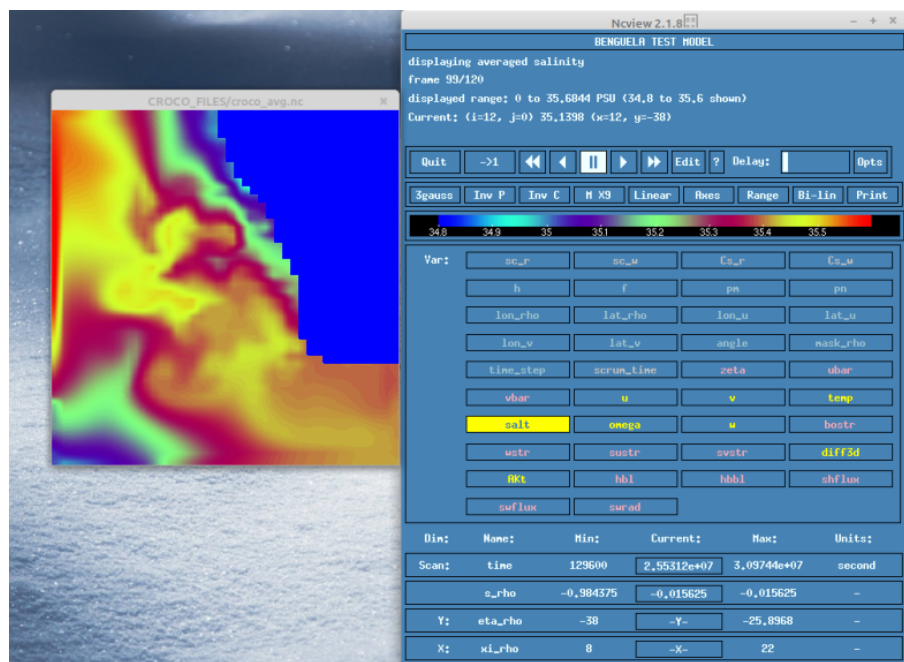


Figura A4: Caso 2b: Salinidad superficial en  $t=28$  obtenido de croco\_avg.nc (rango 34.6-35.6 psu)