

Tutorial 05 CROCO: Biogeoquímica

1. Propósito

En este tutorial revisaremos como realizar una simulación del dominio de Benguela incluyendo un módulo con biogeoquímica simple, los archivos importantes para este tipo de simulaciones, y la validación de esta.

2. Creando el directorio de trabajo BGQ

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).

Asumiremos que ya tiene una copia local del código de CROCO y de CROCO_TOOLS en su directorio raíz. Esto se puede lograr siguiendo los pasos del Tutorial 01.

El primer paso es editar el archivo `create_run.bash` con las instrucciones para crear un nuevo directorio de trabajo que llamaremos **BGQ**

```
1 cd andres/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/andres/croco"
6 #
7 SOURCES_DIR="/home/courses/student11/andres/croco"
8 #
9 TOOLS_DIR="/home/courses/student11/andres/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='BGQ'
16 #
17 #
18 # END USER MODIFICATIONS
19 #=====
```

y después ejecute la instrucción

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

3. Configuración REGIONAL con Biogeoquímica

Para este tutorial agregaremos la opción de biogeoquímica a la física del modelo.

3.1. Activando la Biogeoquímica

Nos cambiamos al directorio **BGQ**

```
1 cd BGQ
```

En el archivo **cppdefs.h** activamos los módulos biogeoquímicos modificando el archivo a

```
1          /* Applications */
2 # define  BIOLOGY
3 # undef   FLOATS
4 # undef   STATIONS
```

activar esta opción tiene mas consecuencias, como se puede ver más abajo en el código

```
          /* Choice of Biology models */
# ifdef  BIOLOGY
# undef  PISCES
# undef  BIO_NChlPZD
# undef  BIO_N2ChlPZD2
# define BIO_BioEBUS
```

usamos por ahora el modelo más simple, el **BIO_NChlPZD** que tiene cinco componentes: Nitrato, Clorofila, Fitoplancton, Zooplankton, y Detritus.

```
          /* Choice of Biology models */
# ifdef  BIOLOGY
# undef  PISCES
# define BIO_NChlPZD
# undef  BIO_N2ChlPZD2
# undef  BIO_BioEBUS
```

y notemos que eso activa un cálculo de oxígeno

```
# ifdef  BIO_NChlPZD
# define OXYGEN
# endif
```

3.2. Paralelización

No olvide activar la paralelización para su simulación.

```
          /* Parallelization */
# undef  OPENMP
# define  MPI
```

4. Compilando CROCO

A continuación compilamos el modelo

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

5. Los archivos de entrada

Los archivos de entrada que leerá el ejecutable **croco** los crearemos con la herramienta **CROCO_TOOLS**. Esta depende principalmente de dos archivos, el **start.m** y el **crocotools_param.m**

5.1. start.m

Recuerde modificar la siguiente línea en su archivo *start.m*

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

5.2. Biogeoquímica - Usando Matlab

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

```
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
```

Dentro de Matlab las instrucciones a usar, desde el directorio de trabajo **BGQ** son:

```
1 start
2 make_grid
3 make_forcing
4 make_clim
```

en este caso, agregaremos las variables biogeoquímicas a la condición inicial con la instrucción

```
1 make_ini_npzd
```

y en el terminal veremos

```
1 >> make_ini_npzd
2 > In make_ini_npzd (line 27)
3 mkdir: cannot create directory '/home/courses/student11/andres/croco/BGQ/CROCO_FILES/': File exists
4
5 Adding NPZD data in file : /home/courses/student11/andres/croco/BGQ/CROCO_FILES/croco_ini.nc
6
7 Interpolations / extrapolations
8
9 Nitrate ...
10
11 Ext tracers: ro = 0 km - default value = NaN
12   ext_tracers_ini: time index: 1 of total: 4
13   ext_tracers_ini: horizontal interpolation of seasonal data
14   ext_tracers_ini: vertical interpolation
15
16 Oxygen ...
17
18 Ext tracers: ro = 0 km - default value = NaN
19   ext_tracers_ini: time index: 1 of total: 4
20   ext_tracers_ini: horizontal interpolation of seasonal data
21   ext_tracers_ini: vertical interpolation
22
23 CHla...
24 Add_ini_chla: creating variable and attribute
25 time index: 1 of total: 1
26 temporal interpolation
27 Initialisation time: 0 - Time 1: -45 - Time 2: 45
28 Add_ini_chla: horizontal extrapolation of surface data
29 Add_ini_chla: vertical extrapolation of chlorophyll
30
31 Phyto...
32 Add_ini_phyto: creating variable and attribute
33 time index: 1 of total: 1
34
35 Zoo...
36 Add_ini_zoo: creating variable and attribute
37 time index: 1 of total: 1
38 >>
```

Esto agrega las variables asociadas a la biogeoquímica en el archivo **croco_ini.nc**.

Ahora, agregaremos las variables biogeoquímicas a la condición de borde oceánica con la instrucción

```
1 make_clim_npzd
```

y en el terminal veremos

```
1 >> make_clim_npzd
2 > In make_clim_npzd (line 27)
3 mkdir: cannot create directory '/home/courses/studen11/andres/croco/BGQ/CROCO_FILES/': File exists
4 =====
5 => You need the croco_oa.nc file created by make_clim.m
6 => with makeoa=1 from crocotools_param.m
7 =====
8 Add_no3: creating variables and attributes for the OA file
9 Add_no3: creating variables and attributes for the Climatology file
10 Add_o2: creating variables and attributes for the OA file
11 Add_o2: creating variables and attributes for the Climatology file
12
13 Ext tracers: Roa = 0 km - default value = NaN
14 Ext tracers: horizontal interpolation of the annual data
15 Ext tracers: horizontal interpolation of the seasonal data
16 time index: 1 of total: 4
17 time index: 2 of total: 4
18 time index: 3 of total: 4
19 time index: 4 of total: 4
20
21 Ext tracers: Roa = 0 km - default value = NaN
22 Ext tracers: horizontal interpolation of the annual data
23 Ext tracers: horizontal interpolation of the seasonal data
24 time index: 1 of total: 4
25 time index: 2 of total: 4
26 time index: 3 of total: 4
27 time index: 4 of total: 4
28
29 Vertical interpolations
30
31 O2...
32 Time index: 1 of total: 4
33 Time index: 2 of total: 4
34 Time index: 3 of total: 4
35 Time index: 4 of total: 4
36
37 N03...
38 Time index: 1 of total: 4
39 Time index: 2 of total: 4
40 Time index: 3 of total: 4
41 Time index: 4 of total: 4
42
43 Phyto...
44 Add_phyto: creating variable and attribute
45 time index: 1 of total: 4
46 time index: 2 of total: 4
47 time index: 3 of total: 4
48 time index: 4 of total: 4
49
50 Zoo...
51 Add_zoo: creating variable and attribute
52 time index: 1 of total: 4
53 time index: 2 of total: 4
54 time index: 3 of total: 4
55 time index: 4 of total: 4
56 >>
```

Esto agrega las variables asociadas a la biogeoquímica en el archivo **croco_clm.nc**. Como puede observar, sólo se agregan valores de estas variables para cuatro pasos de tiempo, estos son los valores estacionales. Recuerde que CROCO realiza una interpolación lineal entre estos valores para los pasos de tiempo intermedios.

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

```
1 http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial05/ArchivosIniciales/
```

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

6. Simulación Biogeoquímica

Para estudiar una simulación biogeoquímica tenemos que forzar el modelo durante varios años. Para esto modificaremos el **croco.in** para que el número de pasos de tiempo sea 8640

```
1 time_stepping: NTIMES dt[sec] NDTFAST NINFO
2                 8640   3600     60     1
```

Una vez hecho esto, lanzamos la simulación copiando el script *run_nlhpc.bash* de los tutoriales anteriores

```
1 cp ../CLIMATOLOGIA/run_nlhpc.bash .
```

y lanzando la simulación con

```
1 sbatch run_nlhpc.bash
```

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
```

Copie el archivo **RST** al **INI**

```
1 mv croco_rst.nc croco_ini.nc
```

y renombre los archivos de salida

```
1 mv croco_avg.nc croco_avg_Y1.nc
2 mv croco_his.nc croco_his_Y1.nc
```

y repita la simulación dos veces más para alejarse de la condición inicial. Después de cada simulación acuerdese de renombrar los archivos **HIS** y **AVG** y mover el **RST** al **INI**.

El archivo de nuestro interés será el **croco_avgY1.nc** puesto que este contiene valores promedio de las variables de interés.

7. Validación de la simulación biogeoquímica

Compararemos las salidas del modelo con datos de clorofila satelital

7.1. Análisis de la simulación

Analizaremos nuestra simulación y usando información de clorofila satelital.

7.2. Análisis de resultados del modelo

Para comparar los resultados del modelo con las observaciones use las herramientas NCO para generar promedios estacionales y

```
1 ml purge
2 ml icc/2018.5.274-GCC-8.2.0-2.31.1 impi/2018.4.274 NCO/4.7.6
3 ncks -F -d time,1,30 croco_avg_Y1.nc croco_avg_EFM.nc
4 ncks -F -d time,31,60 croco_avg_Y1.nc croco_avg_AMJ.nc
5 ncks -F -d time,61,90 croco_avg_Y1.nc croco_avg_JAS.nc
6 ncks -F -d time,91,120 croco_avg_Y1.nc croco_avg_OND.nc
```

el promedio anual del último año calculado.

```
1 ncra croco_avg_Y1.nc croco_avg_anual.nc
```

7.3. Análisis de observaciones

Del NEO (Nasa Earth Observation), Fig. 1 :

https://neo.sci.gsfc.nasa.gov/view.php?datasetId=MY1DMM_CHLORA

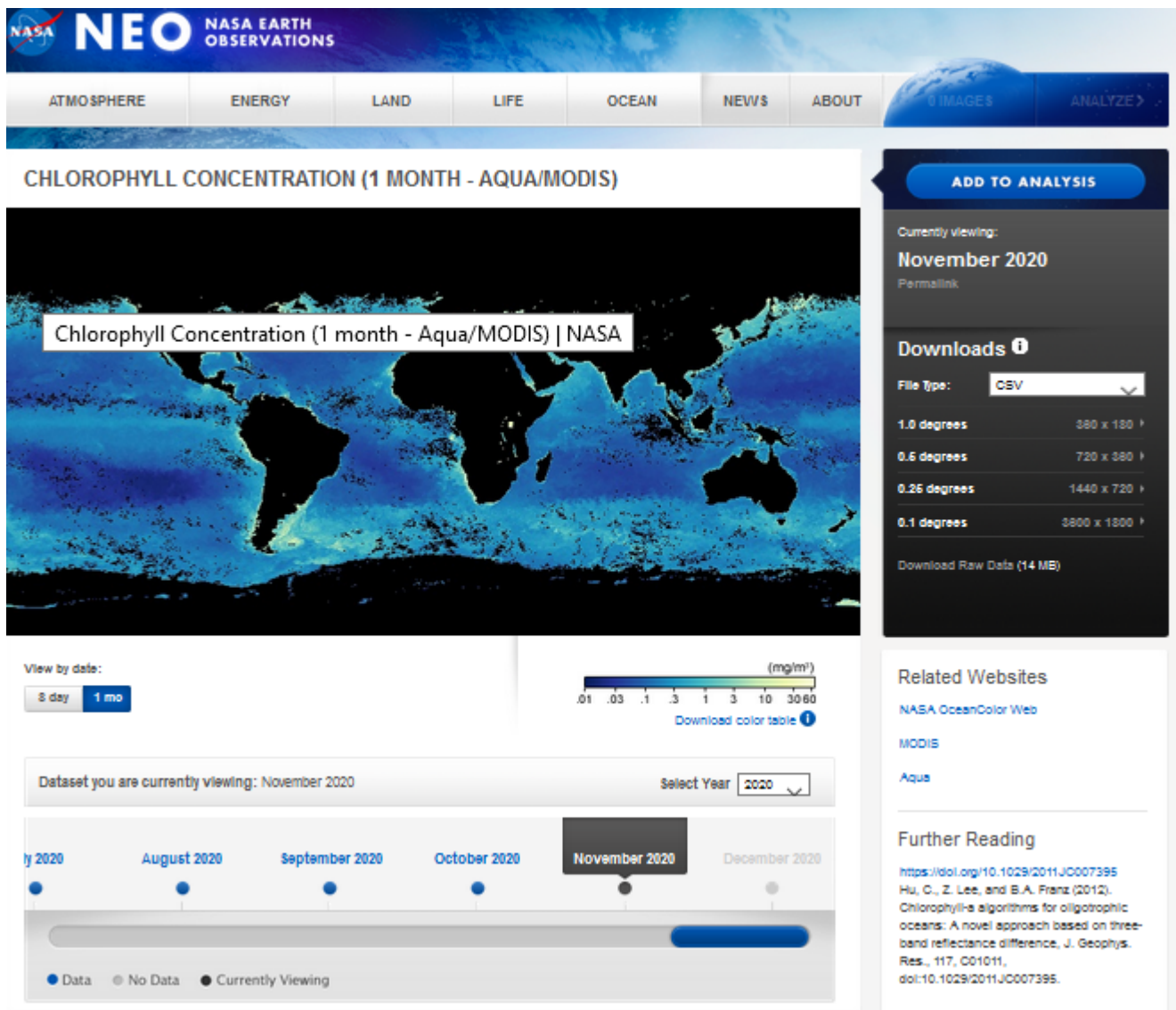


Figura 1: Base de datos para Clorofila satelital. Sensor Aqua-MODIS

bajamos los datos del periodo 2015-2019 con una resolución de 0.25° y promediamos los archivos para obtener una climatología mensual. Puede encontrar los archivos en formato CSV en:

1 <http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial05/ClimatologiaChla/>

Ingrese cada uno de los archivos mensuales en la función `chl_aqua_v2.m` (Anexo A) para poder compararlos con las salidas del modelo.

8. Trabajo Avanzado

Realice una simulación para un dominio de su interés y analice la clorofila superficial.

9. Conclusión

En este tutorial aprendió más detalles de los archivos `cppdefs.h`, `param.h`, y `croco.in`, y las modificaciones que hay que hacer para hacer una simulación de biogeoquímica, así como aspectos de la

validación de los resultados.

Para más información:
Andrés Sepúlveda (asepulveda@dgeo.udec.cl)
Contribuciones de:
Ignacio Acuña
Marcela Contreras
Daniela Henríquez Mauro Santiago

10. 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.

11. Anexo A: Código Matlab para análisis de clorofila superficial

```

clear all
close all

%%
% Enero 2021 - Ignacio Acuna N. (ignacioacu2016@udec.cl)
% EL script genera graficas de resultados de clorofila superficial de
% croco, de datos satelitales (AQUA/MODIS) para la misma zona con la misma
% resolución de croco, y la diferencias entre datos y modelo.
%
%

data=csvread('archivo.csv'); %Ingresar datos de climatología mensual
filename='croco_his.nc'; %Nombre de archivo de CROCO (.his o .avg)
STEP=1; %Paso de tiempo de archivo croco con el que se quiere comparar

%% CARGA DE ARCHIVOS Y GRILLA
modelo=ncread(filename,'CHLA');
modelo=modelo(:,:,32,STEP)';
modelo(find(modelo==0))=NaN;
LON_CROCO=ncread(filename,'lon_rho');
LAT_CROCO=ncread(filename,'lat_rho');
[X_CROCO,Y_CROCO]=size(LON_CROCO);

[Y_DATA,X_DATA,NDATA]=size(data);
LON_DATA=linspace(-180,180,X_DATA);
LAT_DATA=linspace(-90,90,Y_DATA);

%% POSICIONES DE GRILLA DE DATOS CORRESPONDIENTES A DOMINIO
difx=abs(min(min(LON_CROCO))-LON_DATA);
dify=abs(min(min(LAT_CROCO))-LAT_DATA);
minx=min(abs(min(min(LON_CROCO))-LON_DATA));
miny=min(abs(min(min(LAT_CROCO))-LAT_DATA));
poslon1=find(difx == minx);
poslat1=find(dify == miny);
difx=abs(max(max(LON_CROCO))-LON_DATA);
dify=abs(max(max(LAT_CROCO))-LAT_DATA);
minx=min(abs(max(max(LON_CROCO))-LON_DATA));
miny=min(abs(max(max(LAT_CROCO))-LAT_DATA));
poslon2=find(difx == minx);
poslat2=find(dify == miny);
%% DATOS A RESOLUCIÓN DE MODELO
CORTE=data(poslat1:poslat2,poslon1:poslon2);
[Y_CORTE,X_CORTE]=size(CORTE);
[x,y]=meshgrid(1:X_CORTE,1:Y_CORTE);
[u,v]=meshgrid(linspace(1,X_CROCO,X_CROCO),linspace(1,Y_CROCO,Y_CROCO));
NUEVA=interp2(x,y,CORTE,u,v);
resta=NUEVA-modelo;

%% FIGURAS

figure()
pcolor(LON_CROCO',LAT_CROCO',NUEVA)
set(gca,'FontSize',12,'LineWidth',2)
colorbar

```

```
ylabel(colorbar,'[mg/mm3]','FontSize',16)
xlabel('Lon','FontSize',20)
ylabel('Lat','FontSize',20)
title('DATOS','FontSize',18)
%saveas(gcf,'path','png')
```

```
figure()
pcolor(LON_CROCO',LAT_CROCO',modelo)
set(gca,'FontSize',12,'LineWidth',2)
colorbar
ylabel(colorbar,'[mg/mm3]','FontSize',16)
xlabel('Lon','FontSize',20)
ylabel('Lat','FontSize',20)
title('MODELO','FontSize',18)
%saveas(gcf,'path','png')
```

```
figure()
pcolor(LON_CROCO',LAT_CROCO',resta)
set(gca,'FontSize',12,'LineWidth',2)
colorbar
ylabel(colorbar,'[mg/mm3]','FontSize',16)
xlabel('Lon','FontSize',20)
ylabel('Lat','FontSize',20)
title('DATOS-MODELO','FontSize',18)
%saveas(gcf,'path','png')
```