

# Tutorial 08 LiveCROCO: Simulación con Partículas

## 1. Propósito

En este tutorial revisaremos cómo realizar una simulación del dominio de Benguela incluyendo la liberación simultánea (*on line*) de partículas pasivas, los archivos importantes para este tipo de simulaciones, y la validación de ésta.

## 2. Creando el directorio de trabajo PARTÍCULAS

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*). Ésta 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_config.bash` con las instrucciones para crear un nuevo directorio de trabajo que llamaremos **PARTICULAS**

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

Ahora tiene que modificar esta sección para colocar los directorios correctos

```
1 #
2 # Name of the configuration directory defined by the user
3 #
4 MY_CONFIG_NAME='PARTICULAS'
5 #
6 #
7 # END USER MODIFICATIONS
8 #=====
```

y después ejecute la instrucción

```
1 ./create_config.bash
```

## 3. Revisando la física - cppdefs.h

El archivo `cppdefs.h` es uno de los más importantes a la hora de realizar una simulación con CROCO. Es muy conveniente ir entendiendo la estructura de este archivo. Usted tendrá una copia del archivo `cppdefs.h` en su directorio de trabajo, y es el que debe modificar. Algunos aspectos que destacaremos ahora son:

### 3.1. Configuración REGIONAL

En esta sección se define que usaremos la configuración tipo REGIONAL, que es adecuada para simulaciones realistas, asociadas a un área geográfica en particular.

En el archivo `cppdefs.h` corresponde a ésta parte del código:

```
#if defined REGIONAL
/*
!=====
!           REGIONAL (realistic) Configurations
!=====
!
```

```

!-----
! BASIC OPTIONS
!-----
!
*/
/* Configuration Name */
# define BENGUELA_LR

```

Dentro de la configuración REGIONAL, lo primero que definimos es una palabra clave para identificar nuestra configuración. En este caso, esa palabra clave es **BENGUELA\_LR**. Esta palabra clave nos sirve para conectar los archivos **cppdefs.h** y **param.h**, como veremos más adelante.

### 3.2. Partículas - cppdefs.h

En esta sección se activa la aplicación o módulo FLOATS definiéndolo como *define FLOATS*. La advección de partículas se hará en simultáneo con el cálculo de la hidrodinámica de nuestro dominio.

```

/* Applications */
# undef BIOLOGY
# define FLOATS
# undef STATIONS
# undef PASSIVE_TRACER
# undef SEDIMENT
# undef BBL

```

### 3.3. Modelo Lagrangiano

Otra sección relevante es el modelo Lagrangiano. Por defecto, la configuración está definida como advección únicamente por corrientes. El esquema de advección-difusión se puede activar definiendo: *define RANDOM\_WALK*, pero por ahora las dejaremos sin definir (*undef*), es decir, desactivadas.

```

/* Lagrangian floats model */
# ifdef FLOATS
# undef FLOATS_GLOBAL_ATTRIBUTES
# undef IBM
# undef RANDOM_WALK
# ifdef RANDOM_WALK
# define DIEL_MIGRATION
# define RANDOM_VERTICAL
# define RANDOM_HORIZONTAL
# endif
# endif

```

## 4. Compilando CROCO

Una vez que revisamos que el **cppdefs.h** tiene las opciones físicas que queremos, y que el archivo **param.h** define de forma correcta las dimensiones de nuestro dominio, compilamos el ejecutable de CROCO usamos las siguientes instrucciones.

```

1 cd PARTICULAS
2 ./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 **oct\_start.m** y el **crocotools\_param.m**

## 5.1. PARTICULAS - usando Octave

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

```
1 octave-cli
```

Dentro de Octave las instrucciones a usar, desde el directorio de trabajo **PARTICULAS** son:

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

Se RECOMIENDA usar la condición inicial generada a partir del tercer o cuarto año de la corrida climatológica, lo que implica que la circulación en el dominio ya está desarrollada (no inicia en reposo). Usted puede descargar el archivo de la condición inicial que corresponde al cuarto año de simulación climatológica. El archivo debe ser colocado en el directorio `/PARTICULAS/CROCO_FILES/`.

```
1 wget http://mosa.dgeo.udec.cl/LiveCROCO/Tutorial09/Particulas/croco_ini.nc
```

## 5.2. Generando el archivo de partículas - floats.in

En esta sección se definen las posiciones de las partículas  $x$  ( $x,y,z,t$ ) a liberar usando una subrutina de Octave (*i.e.* `initfloats.m`) que se descarga en el directorio de trabajo (PARTICULAS) mediante la siguiente instrucción,

```
1 wget http://mosa.dgeo.udec.cl/LiveCROCO/Tutorial09/initfloats.m
```

Carge nuevamente el programa Octave usando,

```
1 octave-cli
```

Ejecute el programa `initfloats.m` y verifique que se haya creado el archivo de salida `floats.in` en el directorio de trabajo (PARTICULAS). La información de la estructura del archivo `floats.in` lo puede encontrar en el archivo `floats.in_README` que se descarga con el comando,

```
1 wget http://mosa.dgeo.udec.cl/LiveCROCO/Tutorial09/floats.in_README
```

Estructura general del archivo `floats.in`.

```
1 1 Ftitle (a80)
2 ROMS 1.0 - Initial Drifters Locations
3 2 Ft0, Fx0, Fy0, Fz0, Fgrd,Fcoor,Ftype,Fcount, Fdt, Fdx, Fdy, Fdz
4 0.0 15.0000 -26.7918 -10.0 0 1 0 1 0.00 0.000 0.000 0.0
```

A continuación se describen algunos parámetros,

```

1 Ftitle='ROMS 1.0 - Initial Drifters Locations';
2 Fgrd =0;      Fgrd      Nested grid number (ng)
3 Fcoor =1;    Fcoor      Initial horizontal location coordinate type:
4              Fcoor = 0,  grid units:  1 =< Fx0 =< Lm(ng)+1,
5              1 =< Fy0 =< Mm(ng)+1
6
7              Fcoor = 1,  Fx0 is longitude (west values are negative).
8              Fy0 is latitude (south values are negative).
9 Ftype =0;    Float trajectory type:
10            Ftype = 0,  neutral density 3D Lagrangian particles.
11
12            Ftype = 1,  isobaric (constant depth) floats.
13 Fcount=1;   Number of floats to be released at the (Fx0,Fy0,Fz0)
14            location. It must be equal or greater than one.
15            If Fcount > 1, a cluster distribution of floats
16            centered at (Fx0,Fy0,Fz0) is activated.
17
18            NOTE: The total number of floats trajectories to compute
19            ===== must add to NFLOATS.

```

## 6. El archivo de opciones - croco.in

En el archivo `croco.in` se define el período de registro de las posiciones de las partículas, por defecto está definida cada 6 h. También se definen los nombres de archivos de entrada y salida,

```

1 floats: LDEFFLT, NFLT, NRPFFLT / inpname, hisname
2           T      6      0
3           floats.in
4 CROCO_FILES/floats.nc

```

Una vez hecho esto, lanzamos la simulación

```

1 ./croco croco.in

```

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
4 floats.nc

```

## 7. Archivo de salida-floats.nc

A continuación se muestra la estructura de algunas variables del archivo de salida (`/CROCO_FILES/floats.in`). Es importante notar que la trayectoria de cada partícula es guardada en cada columna de la matriz (lon, lat, temp, salt).

```
1 double lon(ftime, drifter) ;
2     lon:long_name = "longitude of floats trajectories" ;
3     lon:units = "degree_east" ;
4     lon:field = "lon, scalar, series" ;
5 double lat(ftime, drifter) ;
6     lat:long_name = "latitude of floats trajectories" ;
7     lat:units = "degree_north" ;
8 double temp(ftime, drifter) ;
9     temp:long_name = "temperature" ;
10    temp:units = "degrees Celsius" ;
11 double salt(ftime, drifter) ;
12    salt:long_name = "salinity" ;
13    salt:units = "PSU" ;
```

Para visualizar las trayectorias de las partículas, usted puede adaptar el siguiente programa,

```
1 wget http://mosa.dgeo.udec.cl/LiveCROCO/Tutorial09/graph_particles.m
```

## 8. Resultados Esperados

Usted debe poder generar

1. Los gráficos de las trayectorias de las partículas, así como las propiedades de temperatura y de salinidad asociadas a su trayectoria.
2. Realice una corrida con 100 partículas liberadas en una determinada posición (fuente puntual)
3. Active el modulo RANDOM\_WALK y compare las trayectorias de la fuente puntual (100 partículas)

## 9. Trabajo Avanzado

Realice una simulación con partículas para un dominio de su interés.

## 10. Conclusión

En este tutorial aprendió más detalles de los archivos **cppdefs.h** y las modificaciones que hay que hacer para realizar una simulación con partículas lagrangianas.

Desarrollo:

Mauro Santiago

Para más información:

Andrés Sepúlveda (asepulveda@dgeo.udec.cl)

Contribuciones de:

Marcela Contreras

Si le sirvió este tutorial mande una postal a:

Dr. Andrés Sepúlveda

Departamento de Geofísica

Casilla 160-C

Correo 3

Concepción

Chile

## 11. Enlaces útiles

- Publicaciones recomendadas

Marinone, S. G., Lavín, M. F., & Parés-Sierra, A. (2011). A quantitative characterization of the seasonal Lagrangian circulation of the Gulf of California from a three-dimensional numerical model. *Continental Shelf Research*, 31(14), 1420-1426. doi:10.1016/j.csr.2011.05.014.

Maslo, A., Azevedo Correia de Souza, J. M., Andrade-Canto, F., & Rodríguez Outerelo, J. (2019). Connectivity of deep waters in the Gulf of Mexico. *Journal of Marine Systems*, 103267. doi:10.1016/j.jmarsys.2019.103267.

Rivas, D., & Samelson, R. M. (2011). A Numerical Modeling Study of the Upwelling Source Waters along the Oregon Coast during 2005, *Journal of Physical Oceanography*, 41(1), 88-112.

Mantovanelli, A., Heron, M. L., Prytz, A., Steinberg, C. R., & Wisdom, D. (2011). Validation of radar-based Lagrangian trajectories against surface-drogued drifters in the Coral sea, Australia. *OCEANS'11 MTS/IEEE KONA*. doi:10.23919/oceans.2011.6107233.