

# Tutorial 04 CROCO: Simulación con Mareas

## 1. Propósito

En este tutorial revisaremos como realizar una simulación del dominio de Benguela incluyendo el efecto de la marea en los bordes, los archivos importantes para este tipo de simulaciones, y la validación de esta.

## 2. Creando el directorio de trabajo MAREAS

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 **MAREAS**

```
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='MAREAS'
16 #
17 #
18 # END USER MODIFICATIONS
19 #=====
```

y después ejecute la instrucción

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

## 3. Configuración REGIONAL con MAREAS

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

### 3.1. MAREAS

Nos cambiamos al directorio **MAREAS**

```
1 cd MAREAS
```

En el archivo **cppdefs.h** activamos las mareas modificando el archivo a

```
1 /* Open Boundary Conditions */
2 # define TIDES
```

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

```

/* Open Boundary Conditions */
# ifdef TIDES
# define SSH_TIDES
# define UV_TIDES
# define POT_TIDES
# undef TIDES_MAS
# ifndef UV_TIDES
# define OBC_REDUCED_PHYSICS
# endif
# define TIDERAMP
# endif
# define OBC_M2CHARACT
# undef OBC_M2ORLANSKI
# define OBC_M3ORLANSKI
# define OBC_TORLANSKI
# undef OBC_M2SPECIFIED
# undef OBC_M3SPECIFIED
# undef OBC_TSPECIFIED
```

Para comenzar, destacaremos la opción

```
1 # define TIDERAMP
```

esta implica que la onda de marea será atenuada por los primeros tres días, con el objetivo de evitar inestabilidades numéricas en la simulación.

### 3.2. Paralelización

No olvide activar la paralelización para su simulación.

```

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

## 4. Mareas - param.h

Dentro del *param.h* el definir la opción **TIDES** implica que

```

!-----
! Tides
!-----
!
#if defined SSH_TIDES || defined UV_TIDES
    integer Ntides          ! Number of tides
```

```

! ===== == =====

# if defined IGW
    parameter (Ntides=1)
# else
    parameter (Ntides=8)
# endif
#endif

```

es decir que nuestro modelo usará 8 componentes de marea. Estas están definidos en la sección 5 de **croco-tools\_param.m**

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %
3  % 5 - Parameters for tidal forcing
4  %
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  %
7  % TPX0 file name (TPX06 or TPX07)
8  %
9  tidename=[DATADIR, 'TPX07/TPX07.nc'];
10 %
11 % Self-Attraction and Loading GOT99.2 file name
12 %
13 sal_tides=1;
14 salname=[DATADIR, 'GOT99.2/GOT99_SAL.nc'];
15 %
16 % Number of tides component to process
17 %
18 Ntides=10;
19 %
20 % Chose order from the rank in the TPX0 file :
21 % "M2 S2 N2 K2 K1 O1 P1 Q1 Mf Mm"
22 % " 1  2  3  4  5  6  7  8  9 10"
23 %
24 tidalrank=[1 2 3 4 5 6 7 8 9 10];
25 %
26 % Compare with tidegauge observations
27 %
28 lon0 = 18.37; % Example:
29 lat0 = -33.91; % Cape Town location
30 Z0 = 1; % Mean depth of tide gauge
31 %
32 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Es decir, para el forzamiento de mareas estaremos usando la solución globales TPX07 (Egbert y Erofeeva, 2002).

La opción `sal_tides=1` implica una corrección del 5% de la amplitud, con una desfase de 180° con la marea local o aún más en zonas costeras.

1 [https://www.esr.org/data-products/antarctic\\_tg\\_database/ocean-tide-and-ocean-tide-loading/](https://www.esr.org/data-products/antarctic_tg_database/ocean-tide-and-ocean-tide-loading/)

## 5. Compilando CROCO

A continuación compilamos el modelo

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

## 6. 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**

### 6.1. start.m

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

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

### 6.2. MAREAS - 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 **MAREAS** son:

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

en este caso, agregaremos la instrucción

```
1 make_tides
```

y en el terminal veremos

```

1 mkdir: cannot create directory '/home/courses/instructor01/andres/croco/MAREAS/CROCO_FILES/':
2 File exists
3 Start date for nodal correction : 1-Jan-2005
4 Reading CROCO grid parameters ...
5 Tidal components : M2 S2 N2 K2 K1 O1 P1 Q1 Mf Mm
6 Processing tide : 1 of 10
7   ssh...
8   Getting ssh_r for time index 1
9   Getting ssh_i for time index 1
10  u...
11  Getting u_r for time index 1
12  Getting u_i for time index 1
13  v...
14  Getting v_r for time index 1
15  Getting v_i for time index 1
16 Convert to tidal ellipse parameters...
17 Process equilibrium tidal potential...
18 Process tidal loading and self-attraction potential...
19 Get total tidal potential...

```

Obteniendo gráficos de la amplitud y fase de las componentes usadas, por ejemplo para K1 Figs 1 y 2

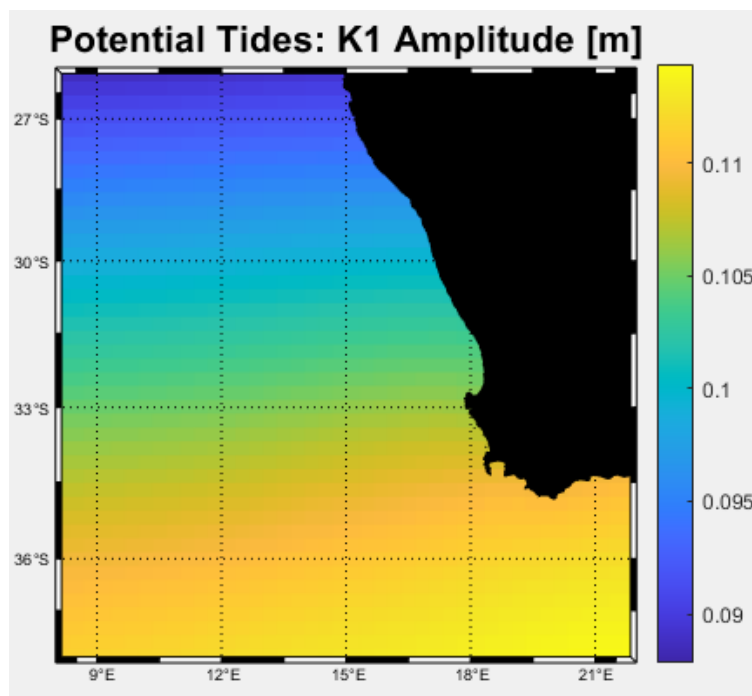


Figura 1: Amplitud de la componente K1

y una serie de tiempo dentro del dominio (Fig. 3

Esto agrega las variables asociadas a la marea en el archivo **croco\_frc.nc**

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

```

1 http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial04/ArchivosIniciales/

```

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

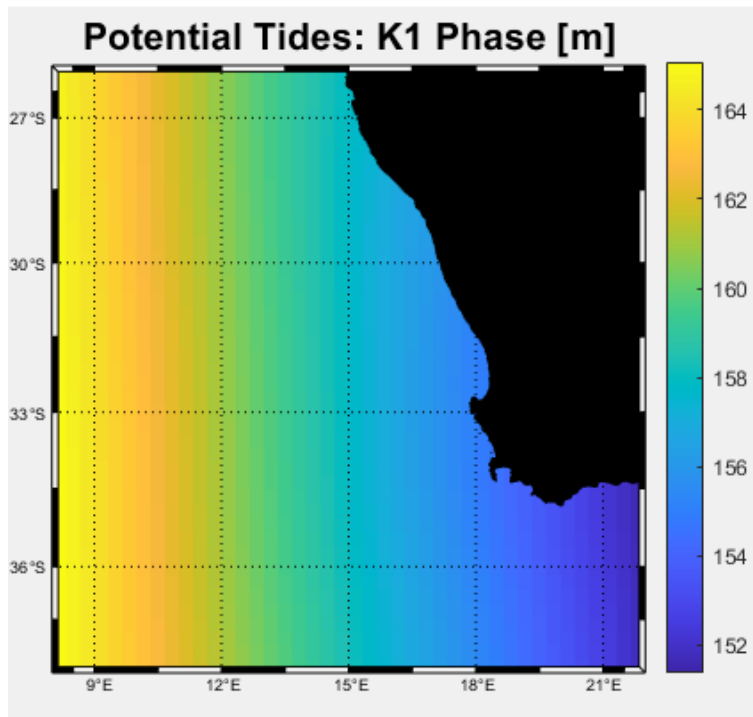


Figura 2: Fase de la componente K1

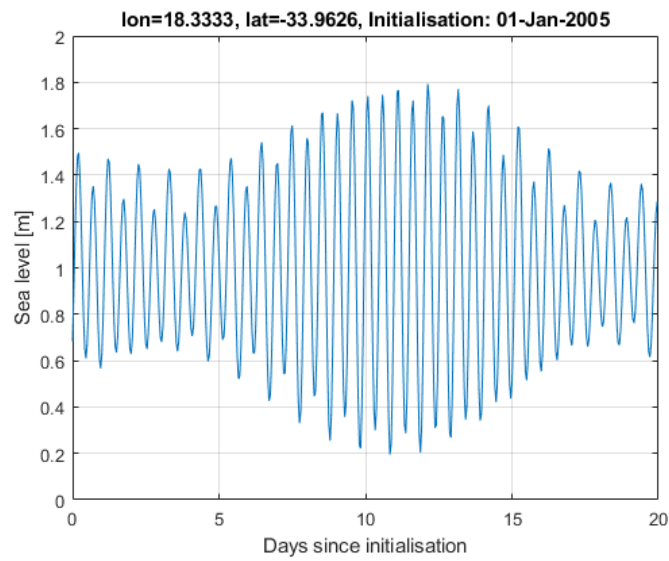


Figura 3: Serie de tiempo en un punto del dominio

## 7. MAREAS - croco.in

Para el análisis de las mareas en nuestro modelo es importante modificar el **croco.in** para grabar las salidas del archivo **HIS** cada 1 hora. Como el paso de tiempo de esta configuración es de 3600 segundos, grabaremos las salida entonces cada 1 paso de tiempo

```
1 history: LDEFHIS, NWRT, NRPFHIS / filename
2         T      1      0
3         CROCO_FILES/croco_his.nc
```

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

```
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
```

donde el archivo de nuestro interés será el **croco\_his.nc**. Sin embargo, como usamos la opción **TIDERAMP** en el **cppdefs.h** el nivel del mar de los primeros tres días es irreal.

Para corregir esto edite de nuevo el **cppdefs.h** y configure

```
1 # undef TIDERAMP
```

luego recompila el modelo con

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

reemplace el archivo **INI**

```
1 cp croco_rst.nc croco_ini.nc
```

y lance de nuevo la simulación

```
1 sbatch run_nlhpc.bash
```

## 8. Validación de la simulación de mareas

### 8.1. Análisis de la simulación

Analizaremos nuestra simulación y observaciones usando el programa **t\_tide** (Pawlowicz et al., 2002). Una versión de **t\_tide** viene ya incluida en **CROCO\_TOOLS**. Para esto usaremos el siguiente código

```
1 https://raw.githubusercontent.com/AndresSepulveda/CROCO_plots/master/t_tide/analyze_t_tide.m
```

que fue adaptado de

```
1 https://isabeljalonrojas.com/analisis-de-armonicos-de-marea-astronomica-con-t_tide/
```

## 8.2. Análisis de observaciones

Primero obtendremos mediciones para su zona del Centro de Variación del Nivel del Mar de la Universidad de Hawaii

<https://uhslc.soest.hawaii.edu/>

o en el sitio web antiguo

<http://uhslc.soest.hawaii.edu/data/>

Por ejemplo, usamos los datos horarios de Simon's Town, Sud Africa (latitud -34.18300 , longitud 18.43300) válidos entre 1959-04-24 y 2020-08-31.

```
1 wget http://uhslc.soest.hawaii.edu/data/nc/fdh/OS_UH-FDH221_20170628_R.nc
```

El archivo **analyze\_t\_tide.m** está configurado para usar sólo 720h, pues es el mismo periodo que tendremos disponible del modelo numérico.

```
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
6 >> analyze_t_tide
```

que debería entregar



```
1 Add the paths of the different toolboxes
2 Arch : x86_64 - Matlab version : 2017a
3 Use of mexnc and loaddap in 64 bits.
4 Points used: 721 of 721
5 percent of var residual after lsqfit/var original: 2.72 %
6 Phases at central time
7 Using nonlinear bootstrapped error estimates
8 Generating prediction without nodal corrections, SNR is 2.000000
9 percent of var residual after synthesis/var original: 2.82 %
10 Points used: 721 of 721
11 percent of var residual after lsqfit/var original: 2.72 %
12 Phases at central time
13 Using nonlinear bootstrapped error estimates
14 Generating prediction without nodal corrections, SNR is 2.000000
15 percent of var residual after synthesis/var original: 2.82 %
16
17 TOTAL =
18
19 4x30 char array
20
21 'M2 0.0805 0.502 250.504'
22 'S2 0.0833 0.295 215.455'
23 'N2 0.079 0.093 36.242'
24 'K1 0.0418 0.054 310.811'
25
26 Nombre Frecuencia Amplitud Fase
27 Factor de Forma (F)
28
29 F =
30
31 0.0941
32
33 Marea Semidiurna
34 >>
```

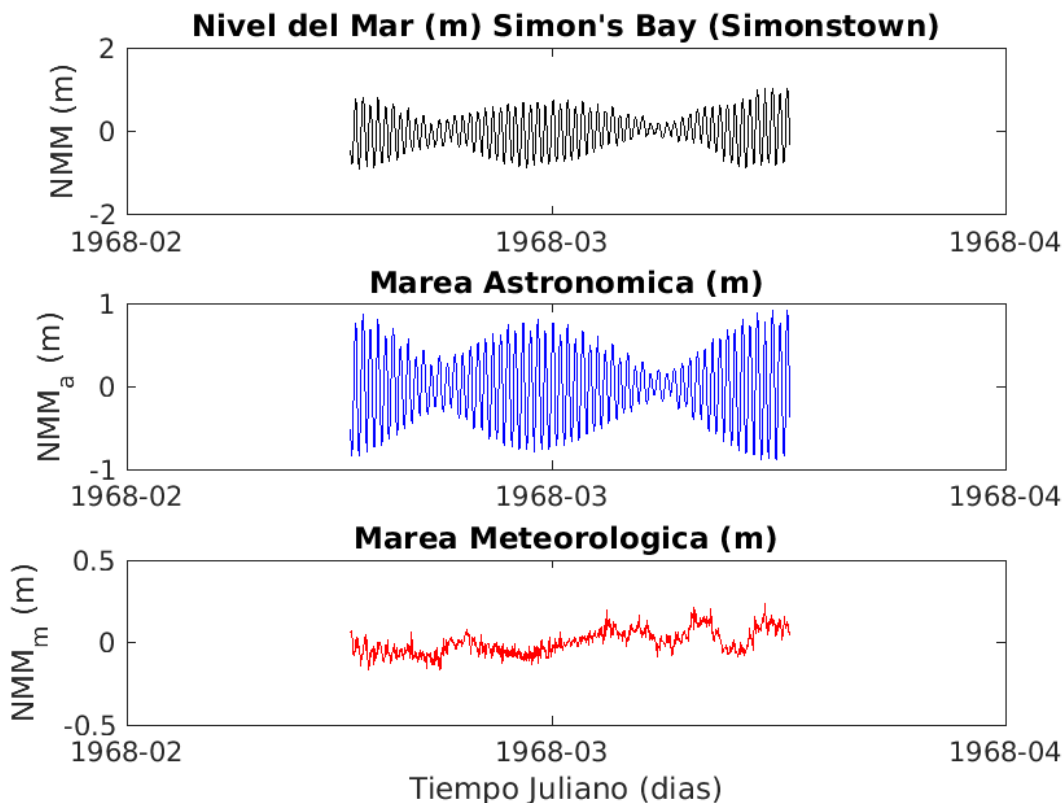


Figura 4: Serie de tiempo para Simon's Town, África de Sur. Observaciones.

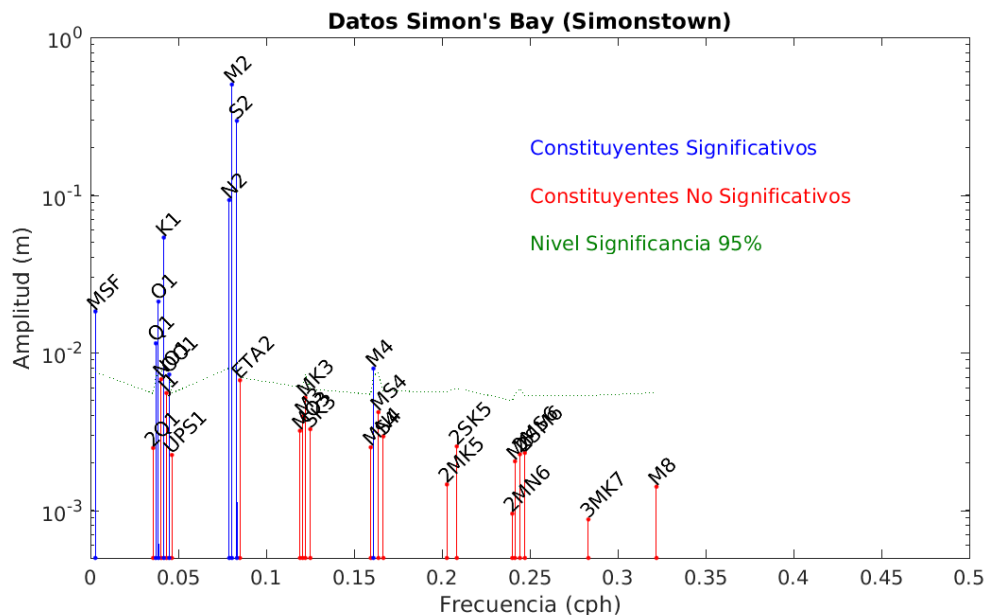


Figura 5: Componentes de marea para Simon's Town, África de Sur. Observaciones

Compare lo resultados del modelo con los datos del UHSLC mediante una tabla. Para esto necesitará ubicar la coordenada (i,j) del punto de la grilla mas cercano a la estación de nivel del mar estudiada y extraer la serie de tiempo de la variable **zeta** en el archivo **croco\_his.nc**. Un ejemplo de eso se hace en el archivo

```
wget http://mosa.dgeo.udec.cl/CROCO2022/CursoBasico/Tutorial04/Tides_Modelo_Benguela.m
```

Al usarlo, debería obtener figuras parecidas a las Figs. 6-7

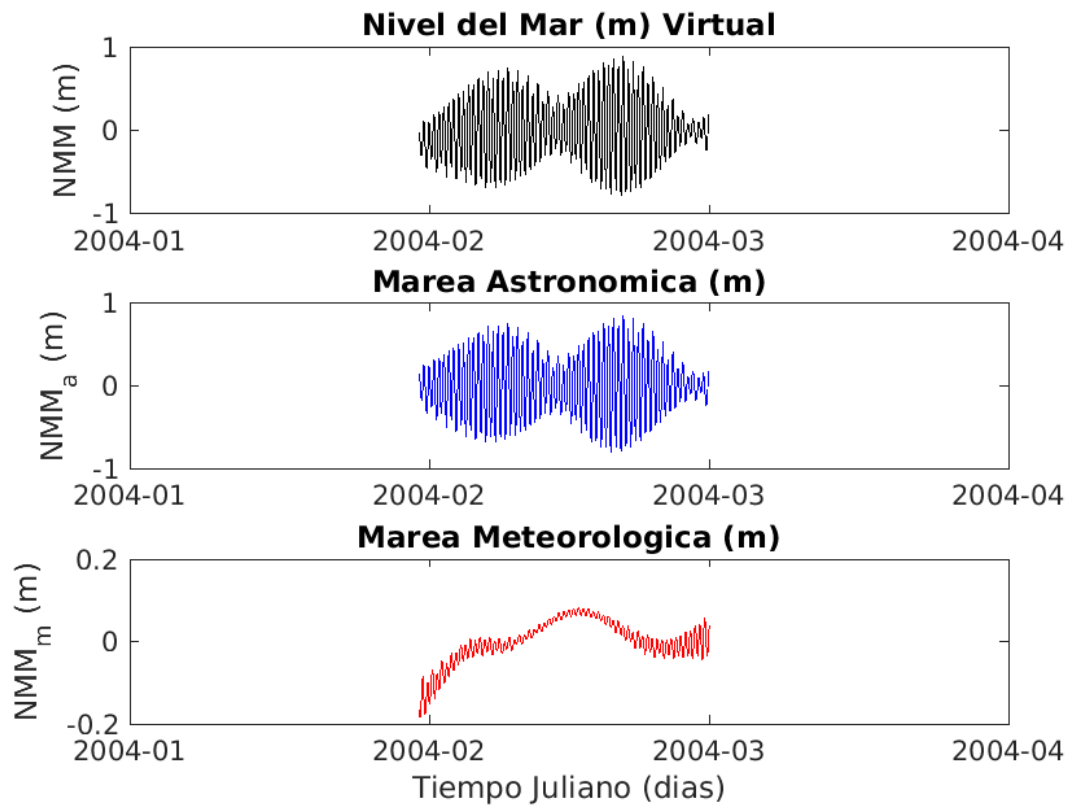


Figura 6: Serie de tiempo para Simon's Town, África de Sur. Modelo

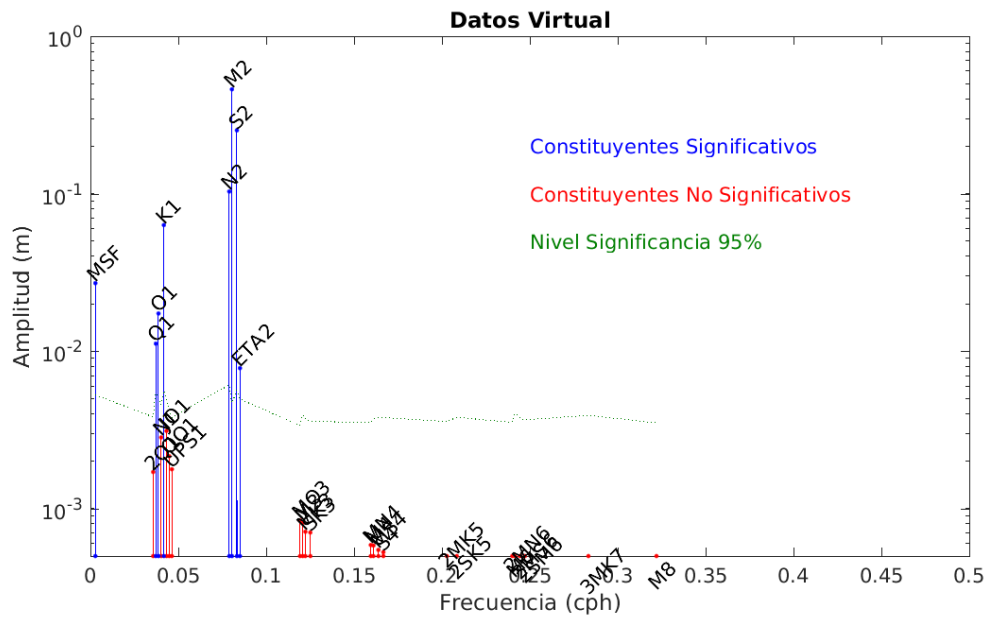


Figura 7: Componentes de marea para Simon's Town, África de Sur. Modelo

y el texto

```

1 Add the paths of the different toolboxes
2 Arch : x86_64 - Matlab version : 2017a
3 Use of mexnc and loaddap in 64 bits.
4   Points used:721 of 721
5   percent of var residual after lsqfit/var original: 0.28 %
6   Phases at central time
7   Using nonlinear bootstrapped error estimates
8   Generating prediction without nodal corrections, SNR is 2.000000
9   percent of var residual after synthesis/var original: 0.29 %
10  Points used: 721 of 721
11  percent of var residual after lsqfit/var original: 0.28 %
12  Phases at central time
13  Using nonlinear bootstrapped error estimates
14  Generating prediction without nodal corrections, SNR is 2.000000
15  percent of var residual after synthesis/var original: 0.29 %
16
17 TOTAL =
18
19 4×30 char array
20
21 'M2 0.0805 0.461 290.979'
22 'S2 0.0833 0.252 88.38'
23 'N2 0.079 0.104 272.562'
24 'K1 0.0418 0.065 87.148'
25
26 Nombre Frecuencia Amplitud Fase
27 Factor de Forma (F)
28
29 F =
30
31 0.1164
32
33 Marea Semidiurna
34 >>

```

Construyan una tabla con la amplitud y fase de las componentes de marea significativas del modelo y de las observaciones, incluyendo la incerteza respectiva. Use a lo más 2 cifras significativas. Si analiza varias estaciones puede hacer un gráfico como la Figura 3 de Artal et al. (2019).

Si le interesan los problemas inversos puede intentar usar el código de Leffler & Jay (2009) ó Codiga (2011), que contienen una estimación de parámetros más robusta. Otros códigos (NS\_TIDE) permiten comparar armónicos no estacionarios (Matte et al. 2013).

## 9. Trabajo Avanzado

Realice una simulación para un dominio de su interés y analice la señal de marea.

## 10. Conclusión

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

Para más información:

Andrés Sepúlveda (asepulveda@dgeo.udec.cl)

Contribuciones de:

Marcela Contreras  
Mauro Santiago

## 11. Referencias

- Artal, O., Pizarro, O., & Sepúlveda, H. H. (2019). The impact of spring-neap tidal-stream cycles in tidal energy assessments in the Chilean Inland Sea. *Renewable Energy*, 139, 496-506.
- Codiga, D.L., 2011. Unified Tidal Analysis and Prediction Using the UTide Matlab Functions. Technical Report 2011-01. Graduate School of Oceanography, University of Rhode Island, Narragansett, RI. 59pp. <https://github.com/wesleybowman/UTide>
- Egbert, Gary D., and Svetlana Y. Erofeeva. "Efficient inverse modeling of barotropic ocean tides." *Journal of Atmospheric and Oceanic Technology* 19.2 (2002): 183-204.
- Leffler, K. E., & Jay, D. A. (2009). Enhancing tidal harmonic analysis: Robust (hybrid L1/L2) solutions. *Continental Shelf Research*, 29(1), 78-88.
- Matte, P., Jay, D.A., Zaron, E.D., 2013. Adaptation of classical tidal harmonic analysis to nonstationary tides, with application to river tides. *J. Atmos. Ocean. Technol.* 30, 569{589. doi:10.1175/JTECH-D-12-00016.
- Pawlowicz, R., Beardsley, B., & Lentz, S. (2002). Classical tidal harmonic analysis including error estimates in MATLAB using T\_TIDE. *Computers & Geosciences*, 28(8), 929-937.
- 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.

## 12. ANEXO A: Mapas de Marea

Es posible generar mapas espaciales de la amplitud y fase de la marea como en

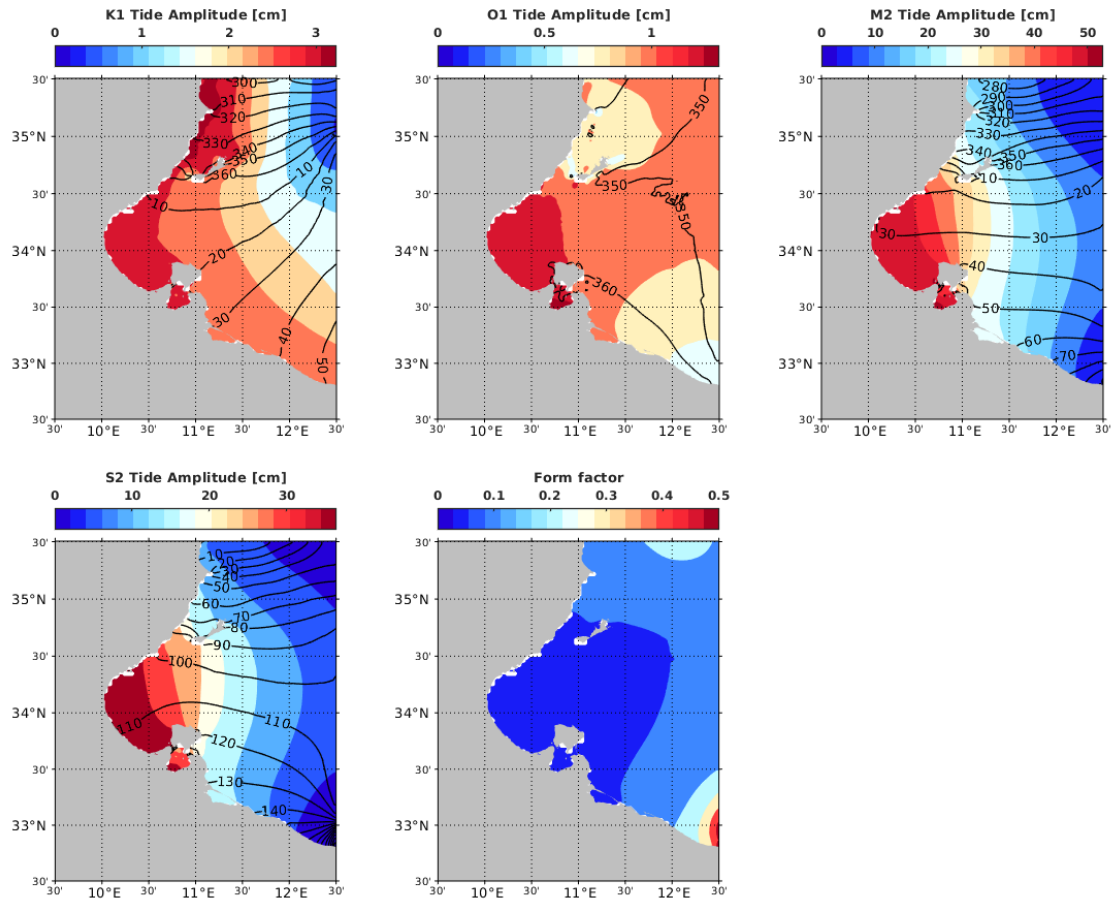


Figura 8: Amplitud y fase de componentes de marea en el Golfo de Gabes, Tunicia

calculando el valor de la amplitud y la fase en cada punto de la grilla usando `t_tide`, guardando los valores en una matriz que después podemos graficar.