

# Generación automática de dominios regulares para la ejecución del modelo de olas WAVEWATCH III

Nota Técnica SMN 2024-172

**Matias De Oto<sup>1</sup>, Yanina García Skabar<sup>1</sup> y Paula Etala**

*<sup>1</sup>Dirección de Productos de Modelación Ambiental y de Sensores Remotos, Dirección Nacional de Ciencia e Innovación en Productos y Servicios*

Junio 2024

### *Información sobre Copyright*

*Este reporte ha sido producido por empleados del Servicio Meteorológico Nacional con el fin de documentar sus actividades de investigación y desarrollo. El presente trabajo ha tenido cierto nivel de revisión por otros miembros de la institución, pero ninguno de los resultados o juicios expresados aquí presuponen un aval implícito o explícito del Servicio Meteorológico Nacional.*

*La información aquí presentada puede ser reproducida a condición que la fuente sea adecuadamente citada.*

## Resumen

En esta Nota Técnica se presentan un conjunto de funciones desarrolladas en Python para la generación de batimetrías, máscaras y obstrucciones de subgrilla, elementos fijos necesarios para la ejecución del modelo de olas WAVEWATCH III® (Tolman, 2002b). Estas funciones siguen las ideas de los algoritmos desarrollados por Chawla y Tolman (2007) que fueron escritos en código MATLAB. Se evalúa la eficiencia temporal del código ante cambios en la resolución espacial de 3 dominios anidados compuestos por un dominio global, uno hemisférico y otro regional. Tanto el código Python como los ejemplos de evaluación del mismo se encuentran alojados en repositorio gitlab<sup>1</sup>.

## Abstract

This Technical Note presents a set of module functions developed in Python for the generation of bathymetry, masks and subgrid obstructions, necessary elements for the execution of the WAVEWATCH III® wave model (Tolman, 2002b). These functions follow the ideas of the algorithms developed by Chawla and Tolman (2007) that were written in MATLAB code. These sets of modular functions are evaluated against changes in the spatial resolution of 3 nested domains comprising a global, a hemispheric and a regional domain. Both the Python code and its evaluation examples are hosted provisionally in the gitlab repository<sup>1</sup>.

**Palabras clave:** Automatización, grillas, olas, WAVEWATCH III®

## Citar como:

De Oto, M., García Skabar, Y. y Etala P., 2024: Generación automática de dominios regulares para la ejecución del modelo de olas WAVEWATCH III®. Nota Técnica SMN 2024-172

---

<sup>1</sup><https://gitlab.smn.gov.ar/dmsr/automatedgrids>

## 1. INTRODUCCIÓN

WAVEWATCH III © (WWIII, Tolman, 2002b) es un modelo numérico que resuelve la ecuación de balance para la densidad de la acción de las olas, incorporando la partición del espectro bidimensional en mar de viento y distintos trenes de mar de fondo para todos los puntos del dominio. El modelo incluye opciones para aplicaciones en aguas poco profundas, así como la incorporación de puntos de grilla de humectación y desecación. También, permite resolver la propagación de un espectro de oleaje mediante grillas regulares (rectilíneas o curvilíneas) y no estructuradas (triangulares). Actualmente, WWIII se encuentra en su versión 7.14 y liberado a la comunidad científica<sup>2</sup>. El código abierto permite incluir los últimos avances científicos en el campo de la modelización y la dinámica de las olas. Así también corresponde el compromiso y la responsabilidad de estar en un estado del arte permanente.

La puesta en marcha del modelo WWIII requiere de la precompilación de la configuración de la dinámica de olas, y posteriormente, la compilación de los programas con las subrutinas que luego se ejecutarán de forma secuencial para obtener una simulación del oleaje<sup>3</sup>. Dentro de estos programas se encuentra el denominado **ww3\_multi** cuya función es calcular el pronóstico de olas en dominios anidados. La utilización de este programa requiere de la definición de los dominios. Los dominios corresponden a los campos de profundidad (batimetría), las máscaras y las obstrucciones de subgrilla, los cuales se consideran elementos fijos en la generación del pronóstico. Dentro de estos dominios, las obstrucciones de subgrilla (Tolman, 2003) buscan contemplar la energía de ola disipada por los elementos geográficos que son más pequeños que la resolución configurada.

Cuando se desea implementar el modelo WWIII de forma anidada, a priori, se desconoce cuál es la configuración de dominios que representa mejor el oleaje regional. Se desconoce, además, cuál es la configuración que tendría una mayor eficiencia temporal. Por esta razón, la búsqueda de los dominios se vuelve una tarea repetitiva en la optimización de la ejecución del modelo. Repetitiva, ya que involucra múltiples pruebas con distintas configuraciones de batimetría, máscaras y obstrucciones de subgrilla. Este proceso genera un gran consumo de horas de trabajo cuando la generación de los dominios es de forma manual. En este sentido, resulta obvia la formulación de los beneficios que puede otorgar un algoritmo automático. Chawla y Tolman (2007) fueron los que primero diseñaron un software de generación automática de dominios para WWIII en código MATLAB. Dicho software genera dominios de batimetría, máscaras, y los dominios de obstrucción de subgrilla. A pesar de que el software desarrollado por Chawla y Tolman (2007) es de licencia privada, lo que dificulta la libre utilización del mismo, en su trabajo se encuentra una rica descripción de los algoritmos utilizados y se exponen ideas sólidas sobre la necesidad de tener un proceso de generación de dominios automatizado.

En este trabajo, se materializaron algunas ideas del algoritmo de generación de dominios automatizado de Chawla y Tolman (2007) en lenguaje de código abierto Python. Particularmente, se explotó la utilización de las funciones internas de las librerías Shapely y Geopandas, la primera relacionada con la manipulación de elementos geométricos y la segunda, relacionada a la geolocalización de los mismos. También se hizo uso de la librería Cartopy, la cual contiene información de campos globales de batimetría y líneas de costas. Por el momento, los dominios de batimetría, máscaras y obstrucciones de subgrilla son del tipo regular, aunque sí existe la necesidad, el algoritmo puede extenderse a dominios no-estructurados. Finalmente, se puso a prueba dicho algoritmo a través del estudio del tiempo de cómputo de la generación de distintos dominios.

<sup>2</sup><https://github.com/NOAA-EMC/WW3>

<sup>3</sup>[https://gitlab.smn.gov.ar/dmsr/hindcast\\_config](https://gitlab.smn.gov.ar/dmsr/hindcast_config)

## 2. CÓDIGO

A continuación, se describen los módulos que contemplan el algoritmo desarrollado en lenguaje de código abierto Python y sus librerías (Tabla I).

**Tabla I:** Librerías Python que requieren los módulos para su funcionamiento

NUMPY
PANDAS
GEOPANDAS
SHAPELY
XARRAY
CARTOPY

### 2.1 Módulo *generate\_grids\_av*

- batimetria, mask = generate\_grid\_av(latitude, longitude, batimetria, umbral)

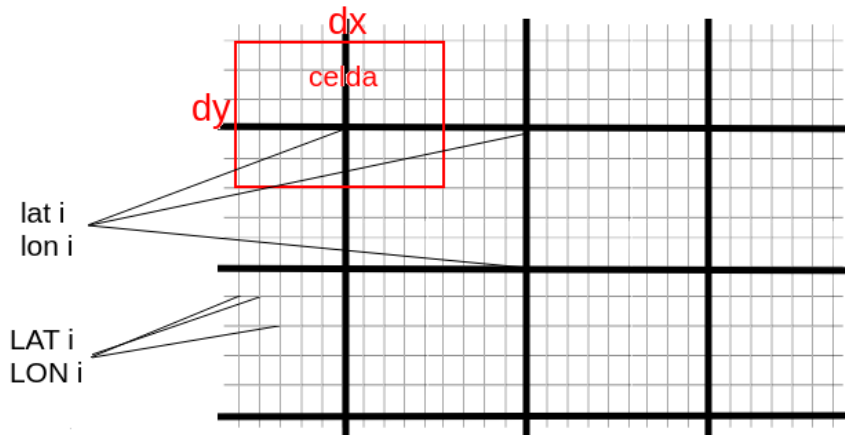
Parámetros de entrada:

1. latitude: Numpy 1D Array de discretización regular
2. longitude: Numpy 1D Array de discretización regular
3. batimetria: Batimetría regular Xarray de alta resolución
4. umbral: Es un valor entre 0 y 1 que representa el cociente aceptable entre los nodos de profundidades no negativas y los nodos totales para generar la máscara de baja resolución. Si el cociente es mayor al umbral establecido, se define un nodo seco (drycell) caso contrario es nodo húmedo (wetcell).

Parámetros de salida:

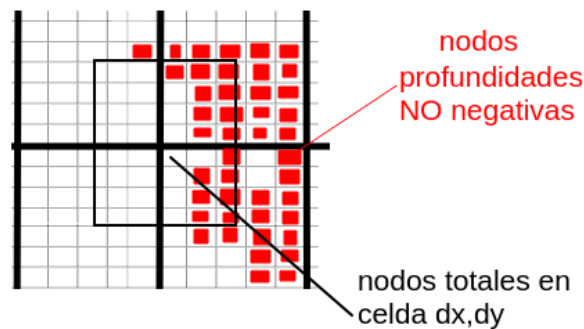
1. batimetría: Numpy 2D array que contiene la información de la batimetría de baja resolución
2. mask: Numpy 2D array de ceros (drycells) y unos (wetcells).

El módulo *generate\_grids\_av* genera una batimetría de baja resolución mediante un promedio en celdas regulares de una batimetría de alta resolución. Las celdas regulares se construyen alrededor de cada coordenada de latitud y longitud de baja resolución (Fig. 1), las que se le ingresan al módulo como parámetros de entrada. Dicha celda está caracterizada por un rectángulo definido por el espaciamiento regular de las coordenadas de longitud y latitud de baja resolución, dx y dy, respectivamente.



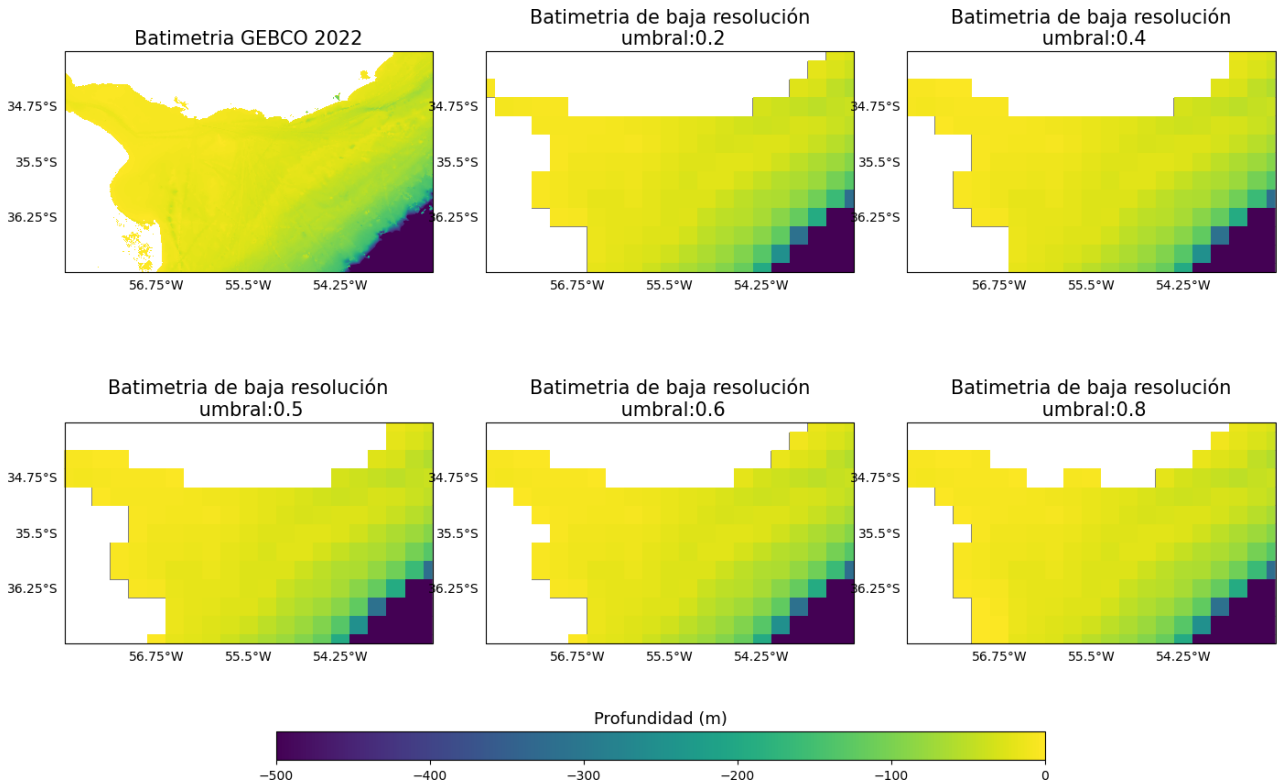
**Fig. 1:** Asignación de la celda regular ( $dx, dy$ ) alrededor de las coordenadas de baja resolución ( $lon i, lat i$ ) para computar el promedio de la batimetría de alta resolución ( $LON i, LAT i$ ).

El promedio de la batimetría de alta resolución sobre la celda regular (Fig. 1) se computará únicamente para las profundidades negativas. Según Chawla y Tolman (2007) realizar un promedio evita el submuestreo y la aparición del aliasing que potencialmente surge en la interpolación. Además, el módulo otorga una máscara de ceros y unos. Los ceros representan los nodos secos (drycells) y los unos los nodos húmedos (wetcells). Para determinar si el nodo es seco o húmedo se cuantifica la cantidad de nodos con profundidades no negativas y la cantidad de nodos totales que hay en una celda de la batimetría de alta resolución. En la Figura 2 se observa un ejemplo de una celda regular que contiene 48 nodos de alta resolución de los cuales 16 son nodos de profundidades no negativas (rojo). En este caso, el cociente entre los nodos con profundidades no negativas y los totales es de 0.33. En este caso, la máscara de baja resolución en dicha celda se considerará húmeda si el umbral definido como parámetro de entrada es superior a 0.33, caso contrario la celda se considerará seca.



**Fig. 2:** Celda regular que contiene 48 nodos de alta resolución, de los cuales 16 son nodos con profundidades no negativas (rojo).

La Figura 3 muestra la batimetría de alta resolución GEBCO 2022 (panel superior izquierdo) de la cual se han obtenido las profundidades promedio (sombreado) en celdas regulares de  $0.25^\circ$  a partir del módulo *generate\_grids\_av*. Los distintos paneles muestran la variación de los nodos secos (blanco) de la máscara para distintos umbrales (ver títulos). Notar que la exigencia para clasificar una celda como seca aumenta a medida que el umbral tiende a 1, por lo que la máscara presenta una importante dependencia del umbral seleccionado. Esta característica se observa en la geometría irregular de la costa, y puede mitigarse utilizando el módulo que se describe a continuación.



**Fig. 3:** Batimetría de alta resolución GEBCO 2022 (panel superior izquierdo) y profundidades promedio (sombreado - metros-) en los nodos húmedos generados por el módulo `generate_grids_av` para distintos umbrales (ver títulos). Los nodos secos se muestran en blanco, y el dominio considerado presenta una resolución espacial de  $0.25^\circ \times 0.25^\circ$ .

## 2.2 Módulo `generate_mask`

- `mask = generate_mask(latitude, longitude, mask*, boundaryes, umbral)`

Parámetros de entrada:

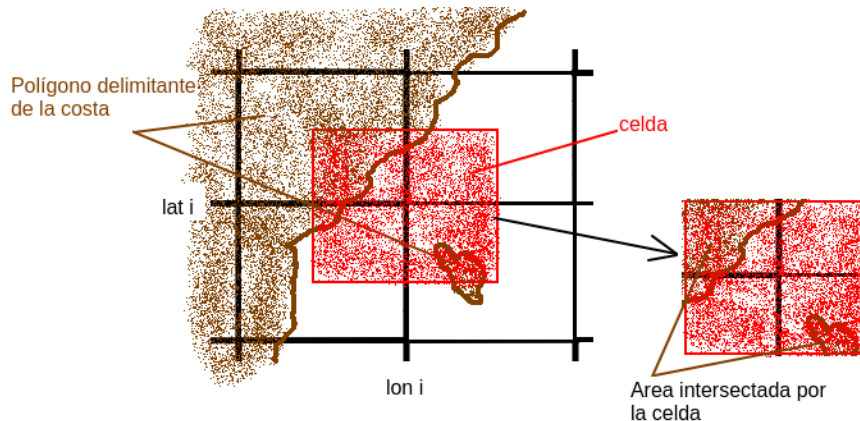
1. `latitude`: Numpy 1D Array de discretización regular
2. `longitude`: Numpy 1D Array de discretización regular
3. `mask*`: Numpy 2D array de ceros (drycells) y unos (wetcells) generado en el módulo `generate_grids_av` ó integrado con la batimetría.
4. `boundaryes`: Conjunto de polígonos geopandalizados que contienen regiones de tierra, y/o regiones a enmascarar
5. `umbral`: Es un valor entre 0 y 1 que representa el cociente aceptable entre el área encerrada por los polígonos (tierra) y el área de la wetcell de la máscara de entrada. Si el cociente es mayor al umbral establecido, se define un nuevo nodo drycell.

Parámetros de salida:

1. `mask`: Numpy 2D array de ceros (drycells) y unos (wetcells).

El módulo `generate_mask` busca mejorar la máscara en función de la identificación de las costas por medio de polígonos. A diferencia de una batimetría de alta resolución, los polígonos de costa presentan una mayor resolución en un menor tamaño de archivo. Estos archivos se presentan en formato “.shp”, aunque en nuestro

caso, y por default, utilizamos la librería de cartopy<sup>4</sup>, *NaturalEarthFeatures*. Independientemente de donde provengan los polígonos que delimitan las costas, los mismos deben “geopandalizarse”<sup>5</sup>, es decir deben localizarse en un sistema georeferencial. Previo a la geopandalización, existe la posibilidad de adjuntar polígonos que delimitan regiones de enmascaramiento arbitrarias y que no son útiles en la ejecución del modelo. La georeferenciación de los polígonos permite cuantificar el área de la intersección entre los polígonos y las celdas regulares de baja resolución (Fig. 4). El área de la intersección será un requisito para determinar si la celda regular es considerada seca o húmeda mediante la definición de un umbral en los parámetros de entrada de dicho módulo.



**Fig. 4:** Intersección entre los polígonos y la celda regular de baja resolución. El área de la intersección entre estos es  $\sim 0.3$  del área total de la celda. Si definimos un umbral mayor a 0.3 la celda se considerará húmeda, en cambio para un umbral menor se considerará seca.

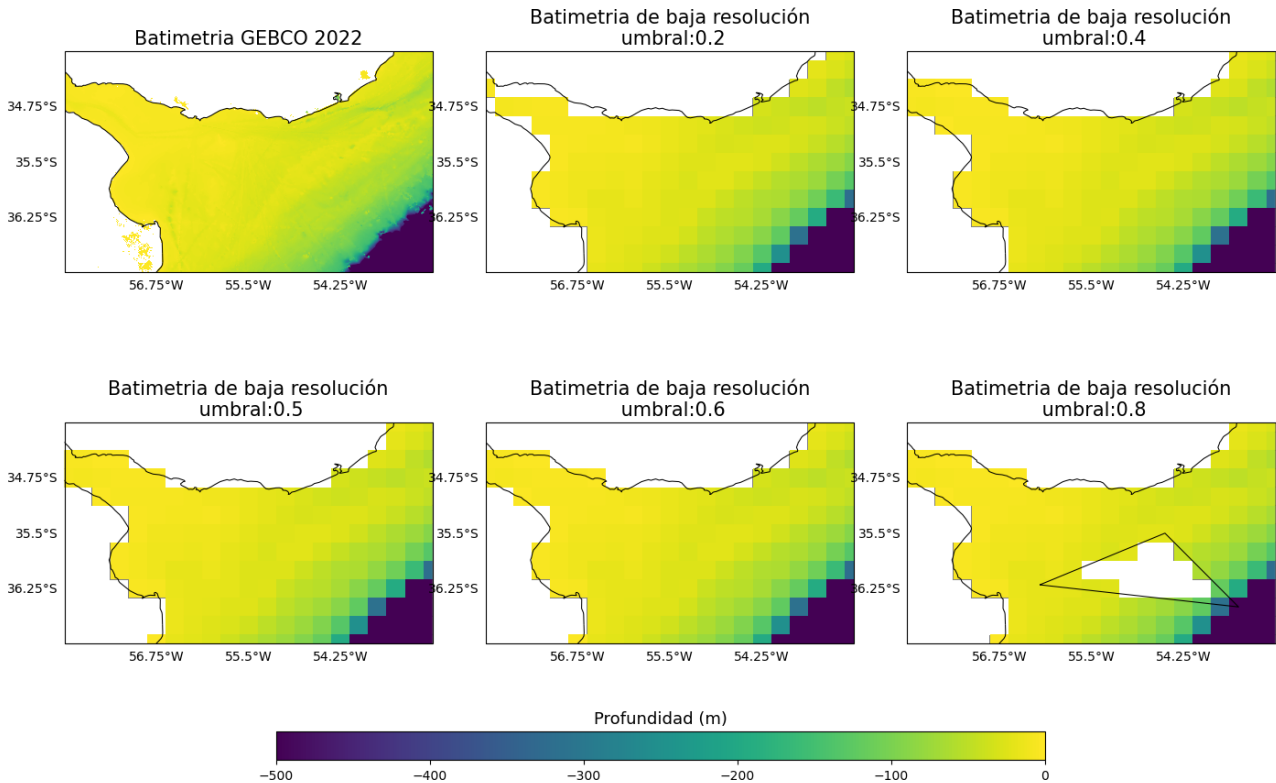
La Figura 5 muestra la batimetría de alta resolución GEBCO 2022 (panel superior izquierdo) en donde se han superpuesto los polígonos de costa de *NaturalEarthFeatures* (línea negra continua). El resto de los paneles muestran la batimetría de baja resolución (sombreado -metros-) obtenida por el módulo *generate\_grids\_av* y distintas máscaras generadas por el módulo *generate\_mask* con distintos umbrales (ver títulos). Para la construcción de las máscaras se utilizó la máscara de salida del módulo *generate\_grids\_av* con un umbral de 0.5 y los polígonos del módulo *NaturalEarthFeatures* de la librería *cartopy*. Al módulo *generate\_mask* se le pueden incorporar polígonos de enmascaramiento, estos polígonos se adhieren a los polígonos costeros<sup>6</sup>. En el panel inferior derecho se muestra un polígono de enmascaramiento con vértices (-56.2, -36.2), (-54.5, -35.5), (-36.5, -53.5).

<sup>4</sup> <https://scitools.org.uk/cartopy/docs/latest/index.html>.

<sup>5</sup> <https://geopandas.org/en/stable/>.

<sup>6</sup> Se recomienda seguir el tutorial de gitlab para comprender cómo se adhiere un polígono extra de enmascaramiento.

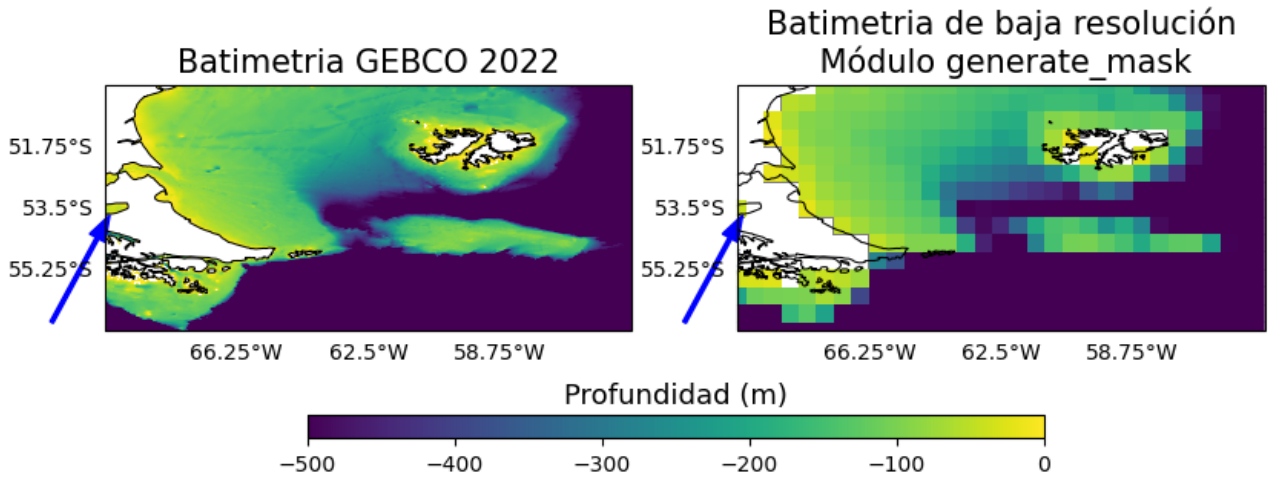




**Fig. 5:** Panel superior izquierdo. Batimetría de alta resolución GEBCO 2022 superpuesta a los polígonos costeros de NaturalEarthFeatures (trazo negro). El resto de los paneles muestran las profundidades promedio (sombreado -metros-) en las celdas húmedas generadas con un umbral de 0.5 del módulo `generate_grids_av`. Mientras que, las celdas secas (en blanco), fueron generadas con distintos umbrales (ver títulos) del módulo `generate_mask`. En el panel inferior derecho se muestra un polígono de enmascaramiento de vértices  $(-56.2, -36.2)$ ,  $(-54.5, -35.5)$ ,  $(-36.5, -53.5)$ . La resolución espacial del dominio es de  $0.25^\circ \times 0.25^\circ$ .

Nótese que el módulo `generate_mask` requiere de una máscara inicial en los parámetros de entrada. En este caso, la máscara inicial se obtiene del módulo `generate_grids_av` que implícitamente posee una batimetría asociada. El objetivo de esta máscara inicial es reducir el tiempo de iteración sobre las celdas húmedas, se debe recordar que el módulo busca transformar celdas húmedas en secas. En este sentido, el módulo no busca realizar la transformación inversa, de celdas secas a húmedas, debido a que no existe una profundidad asociada para dichas celdas (Chawla y Tolman, 2007). Un ejemplo de este caso se observa en las celdas cercanas a las coordenadas  $(-57, -35.5)$ , zona de Punta Indio, en donde el algoritmo no transforma dichas celdas secas en húmedas dado que no tienen asociada una profundidad.

La metodología de enmascaramiento a través de polígonos costeros tiende a suavizar la geometría irregular de la costa que presenta la máscara luego de la utilización del módulo `generate_grids_av` (comparar Fig. 3 con Fig. 5). No obstante, todavía pueden surgir inconsistencias como, por ejemplo, obtener áreas encerradas de profundidades no negativas que están asociadas a depresiones continentales y también áreas asociadas a espejos de agua encerrados que no tienen contacto abierto con el océano. Un ejemplo de esta problemática se observa en la Figura 6, en donde a pesar de utilizar polígonos de alta resolución costera existen celdas húmedas desvinculadas del océano abierto (flecha azul) que es la región en donde se ejecutará el modelo. Para eliminar las áreas encerradas desvinculadas al océano abierto, se recomienda el módulo que se explica a continuación.



**Fig. 6:** Problemática del enmascaramiento utilizando el módulo `generate_mask`. La flecha azul denota un ejemplo de celda húmeda desvinculada del océano abierto.

### 2.3 Módulo `generate_wetcell_mask`

- `mask = generate_wetcell_mask(latitude, longitude, mask*, nro_max)`

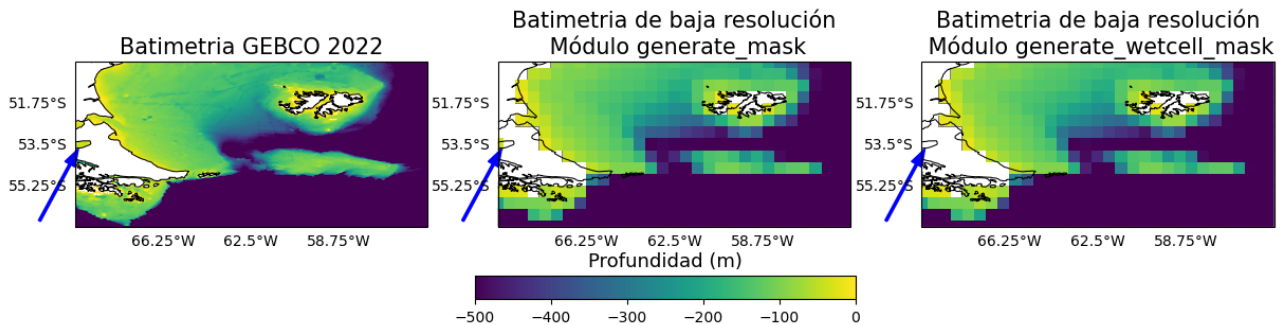
Parámetros de entrada:

1. `latitude`: Numpy 1D Array de discretización regular
2. `longitude`: Numpy 1D Array de discretización regular
3. `mask*`: Numpy 2D array de ceros (`drycells`) y unos (`wetcells`) generado en el módulo `generate_mask` (ó predefinida) que presente regiones cerradas desvinculadas a la región donde se ejecutará el modelo de olas
4. `nro_max`: Número entero que cuantifica la cantidad de regiones oceánicas que deseamos considerar.

Parámetros de salida:

1. `mask`: Numpy 2D array de ceros (`drycells`) y unos (`wetcells`) en donde se erradicaron las regiones cerradas desvinculadas.

El módulo `generate_wetcell_mask` tiene como objetivo disminuir los tiempos de cómputo del modelo mediante la eliminación de celdas húmedas desvinculadas del océano abierto puesto que no tienen incidencia en el mismo. Recordar que una celda húmeda es, por definición, una celda en donde el modelo computa las ecuaciones de olas. Para ello, el módulo poligoniza todas las celdas húmedas y concatena únicamente aquellas que están en contacto. La concatenación genera un conjunto de polígonos en donde cada uno contiene una región húmeda, luego se seleccionan aquellos polígonos que poseen la mayor área en función del parámetro de entrada `nro_max`. La Figura 7 muestra la batimetría GEBCO 2022 (panel izquierdo), la máscara luego de aplicar el módulo `generate_mask` (ver Fig. 6) y el módulo `generate_wetcell_mask`. Para este caso de ejemplo se utilizó `nro_max = 1` debido a que sólo se quiso mantener la región oceánica de mayor tamaño espacial. Obsérvese como la región húmeda desvinculada al océano abierto (flecha azul) es identificada por este último módulo y convertida en región seca.



**Fig. 7:** Batimetría de alta resolución GEBCO 2022 (panel izquierdo), dominio batimétrico y máscara de baja resolución luego de aplicar el módulo `generate_mask` (panel central) y el módulo `generate_wetcell_mask` (panel derecho) utilizando `nro_max = 1`. Este último módulo cataloga como región seca (flecha azul) una región húmeda encerrada que está desvinculada del océano abierto.

## 2.4 Módulo `generate_obs`

-  $S_x, S_y = \text{generate\_obs}(\text{latitude}, \text{longitude}, \text{mask}, \text{boundaries}, \text{displacementx}, \text{displacementy})$

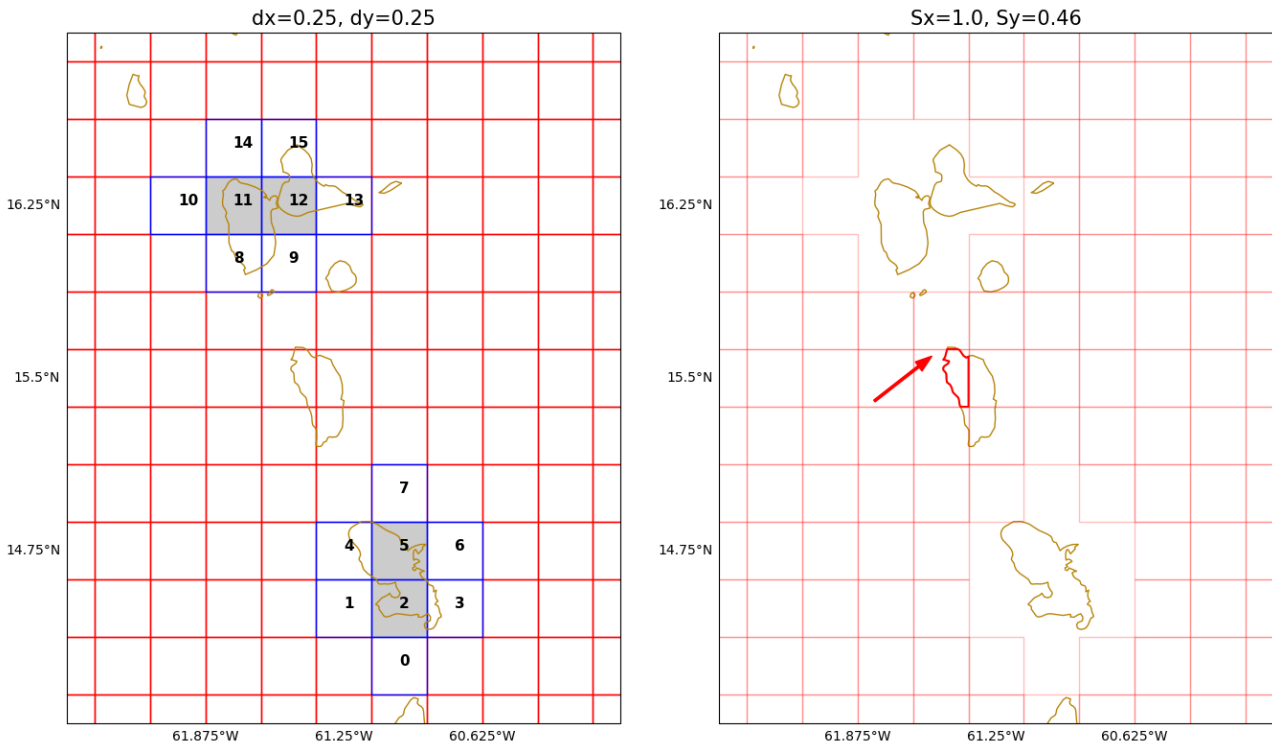
Parámetros de entrada:

1. *latitude*: Numpy 1D Array de discretización regular
2. *longitude*: Numpy 1D Array de discretización regular
3. *mask*: Numpy 2D array de ceros (*drycells*) y unos (*wetcells*) otorgada por el módulo `generate_wetcell_mask` (ó predefinida)
4. *boundaries*: Conjunto de polígonos geopandalizados con características geográficas
5. *displacementx*: Numpy 1D Array con las posiciones de las celdas vecinas en longitud que se tendrán en cuenta para la cuantificación de la obstrucción
6. *displacementy*: *idem displacementx* pero en latitud.

Parámetros de salida:

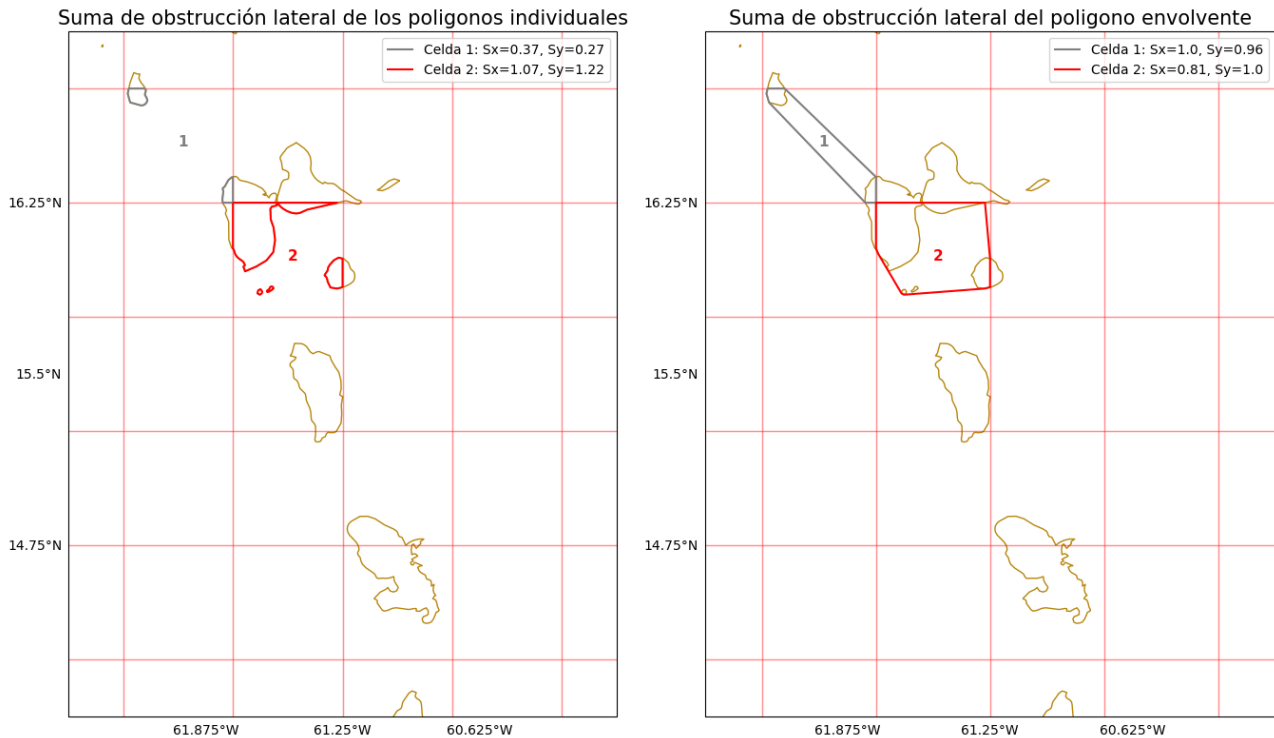
1. *Sx*: Numpy 2D Array que contiene la información de las obstrucciones de subgrilla en las longitudes. Los valores están en el rango de 0 (retícula sin obstrucción) y 1 (retícula con obstrucción total)
2. *Sy*: *Idem Sx* pero en las latitudes.

El módulo `generate_obs` cuantifica la obstrucción de energía del oleaje de aquellos elementos geográficos que son irresolubles (subgrilla) dada la resolución del dominio. En el panel izquierdo de la Figura 8 se muestra como el módulo identifica las celdas secas N° 2, 8 y 9 (sombreado gris) y las celdas húmedas originales (polígonos rojos) de una máscara de ceros y unos, por ejemplo, la máscara obtenida con el módulo `generate_wetcell_mask`. Siguiendo a Chawla y Tolman (2007), se definen como inactivas las celdas que se encuentran al norte, sur, este y oeste de la celda seca original, es decir las celdas N° 0, 1, 3, 4, 5, 6, 7, 10, 11 y 12 (polígonos azules). El paso siguiente es cuantificar el espacio lateral zonal y meridional que ocupan todos los polígonos de la costa (marrón) dentro de cada celda húmeda. En el panel derecho de la Figura 8 se encuentran las celdas húmedas (trazo rojo fino) y un ejemplo de intersección (flecha roja) de una celda húmeda con el sector SE de la Isla de Dominica (trazo rojo grueso). Esta intersección obstruye en un 69% y en un 49%, el flujo longitudinal y latitudinal in-situ, respectivamente.



**Fig. 8:** Panel izquierdo: celdas secas originales (N° 2, 8 y 9; sombreado gris), celdas húmedas inactivas (N° 0, 1, 3, 4, 5, 6, 7, 10, 11 y 12; polígonos azules) y celdas húmedas activas (polígonos rojos). Panel derecho: Ejemplo del cómputo de la obstrucción lateral zonal ( $S_x = 69\%$ ) y meridional ( $S_y = 49\%$ ) de la intersección (flecha, trazo rojo grueso) de un polígono costero y una celda húmeda activa (polígonos rojos). Los polígonos costeros de la librería de NaturalEarthFeatures se muestran en color marrón y la resolución espacial de las celdas es de  $0.25^\circ \times 0.25^\circ$ .

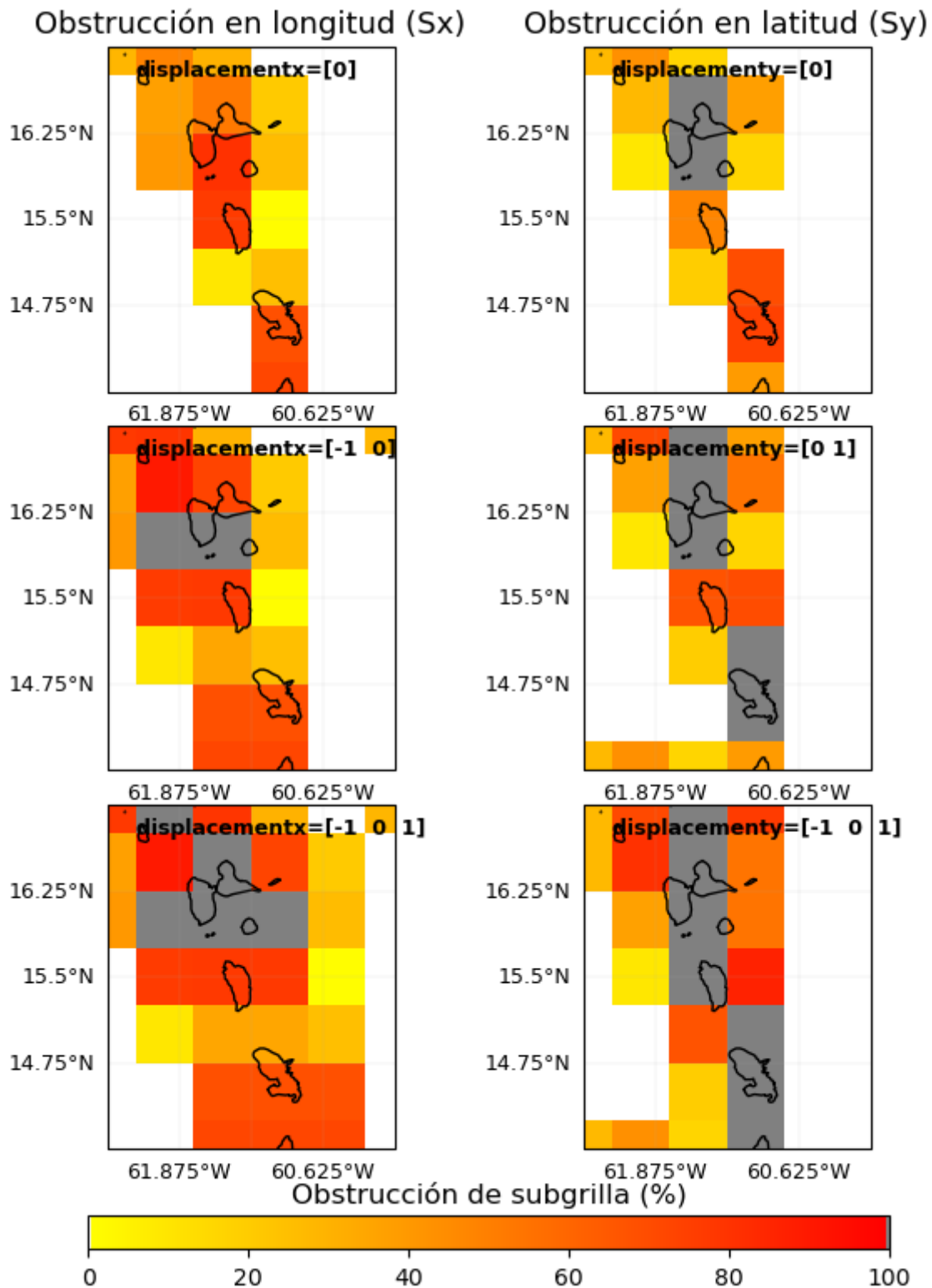
Un problema a resolver es la sobreestimación de la obstrucción lateral cuando dos o más polígonos costeros dentro de una celda se encuentran dentro de lo que se denomina la región de sombra (ver Chawla y Tolman, 2007). Para evitar que se sobreestime la obstrucción lateral, el módulo considera la obstrucción mínima entre las obstrucciones computadas a partir de dos métodos, la suma de la obstrucción lateral de los polígonos individuales (Figura 9, panel izquierdo) y la suma de la obstrucción lateral del polígono envolvente (Figura 9, panel derecho). En el ejemplo de la Figura 9, se muestran dos casos particulares identificados por el módulo, caso de celda N° 1 (gris) y caso de celda N° 2 (rojo). En la celda N° 1 el algoritmo determina que la obstrucción real, aquella que es mínima, es la suma de la obstrucción lateral de los polígonos individuales (panel izquierdo) siendo  $S_x=37\%$  y  $S_y=27\%$  ya que el polígono envolvente (panel derecho) sobreestima la obstrucción. En el caso de la celda N° 2 el algoritmo determina que la obstrucción real está dada por la obstrucción del polígono envolvente (panel derecho), siendo  $S_x=81\%$  y  $S_y=100\%$ . En este caso, el método de la suma individual sobreestima las obstrucciones laterales (panel izquierdo) en más del 100% ya que existen varias islas dentro de la zona de sombra.



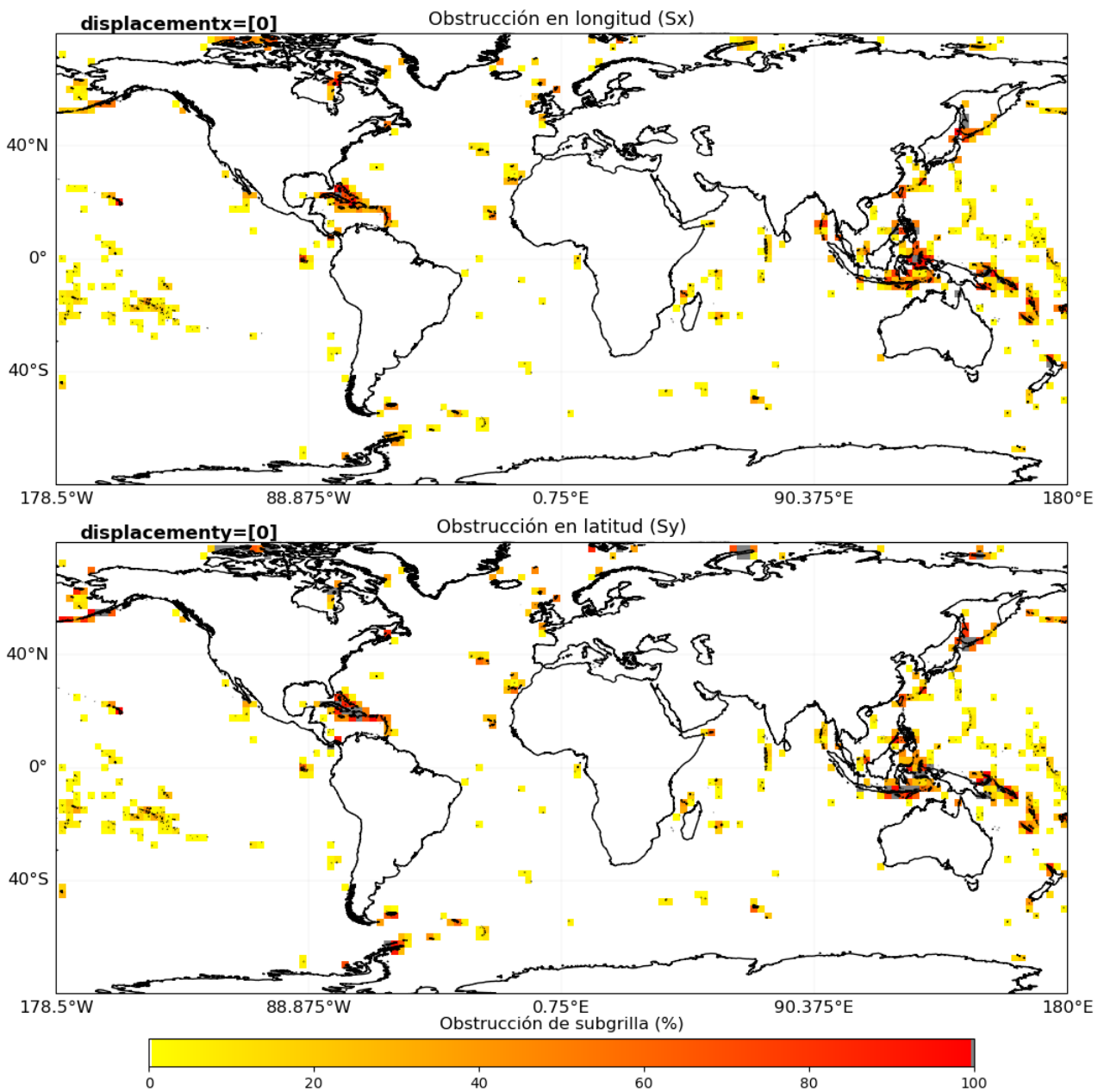
**Fig. 9:** Obstrucción lateral de la energía del oleaje mediante la suma de la obstrucción lateral de los polígonos individuales (panel izquierdo) y del polígono envolvente (panel derecho). Se muestran dos ejemplos para ambos métodos, celda N° 1 (gris) y celda N° 2 (rojo) en donde el módulo computa la obstrucción lateral (ver etiquetas) y selecciona aquella que es mínima, considerada como obstrucción real. La resolución espacial de las celdas es de  $0.5^\circ \times 0.5^\circ$ .

Otro problema que surge en el cómputo de la obstrucción de energía por elemento de subgrilla es lo que Chawla y Tolman (2007) denominaron “contabilización de la celda vecina”, es decir, que la obstrucción lateral se propague entre celdas vecinas. Para contabilizar cuál es la obstrucción lateral real que se propaga entre las celdas vecinas se debe contabilizar cuál es la región de sombra entre las mismas. Dado que no hay un acuerdo en común sobre qué es una celda vecina, y si existe alguna dependencia física entre las mismas, se consideró no formalizar este aspecto en el módulo Python *generate\_obs*. Sin embargo, el módulo acepta la consideración de obstrucción de las celdas vecinas en longitud y latitud mediante los parámetros de entrada *displacementx* y *displacementy*, respectivamente. El cómputo de la contabilización de la celda vecina es una suma directa entre las obstrucciones vecinas por lo que no contempla a la zona de sombra, diferenciándose del algoritmo descrito en Chawla y Tolman (2007). En consecuencia, se espera que la contabilización de la obstrucción de energía por la contabilización de las celdas vecinas sea sobreestimada. Aquellos casos en donde el algoritmo supera en porcentaje una obstrucción total de energía, el valor de obstrucción se fija en 100%.

La Figura 10 muestra una obstrucción lateral sin considerar celdas vecinas (paneles superiores), considerando sólo las celdas contiguas al oeste en longitud y al norte en latitud (paneles centrales) y considerando las celdas contiguas pero en ambas direcciones (paneles inferiores). Nótese cómo a medida que se consideran más celdas vecinas, mayor es la obstrucción zonal y meridional. En definitiva, el número de celdas como parámetro de entrada del módulo tiene como función amortiguar la propagación de energía por las olas. En este sentido, es tarea de una validación del oleaje in-situ para determinar cuál es el número óptimo de celdas vecinas a considerar, es decir cómo este parámetro ayuda a la disipación de la energía del oleaje propagante. Finalmente, y a modo de ejemplo, la Figura 11 muestra el cómputo de las obstrucciones de subgrilla para un dominio global con resolución de  $2.5^\circ$  en longitud y latitud.



**Fig. 10:** Obstrucción de subgrilla (% en sombreado) longitudinal ( $S_x$ , panel izquierdo) y meridional ( $S_y$ , panel derecho) sin considerar celdas vecinas (panel superior), considerando sólo las celdas contiguas al oeste en longitud y al norte en latitud (panel central) y considerando las celdas contiguas pero en ambas direcciones (panel inferior). El sombreado gris denota las celdas en donde la obstrucción es del 100%.



**Fig. 11:** Obstrucción local de energía por las subgrillas (% en sombreado) en longitud ( $S_x$ , panel superior) y latitud ( $S_y$ , panel inferior) para un dominio global de resolución  $2.5^\circ \times 2.5^\circ$ . El sombreado gris denota las celdas en donde la obstrucción es del 100%.

## 2.5 Módulo `modify_mask`

- `mask = modify_mask(latitude, longitud, mask*, active_region_pol)`

Parámetros de entrada:

1. *latitude*: Numpy 1D Array de discretización regular
2. *longitude*: Numpy 1D Array de discretización regular

3. *mask\**: *Numpy 2D array de ceros (drycells) y unos (wetcells) generado por alguno de los módulos mencionados anteriormente o integrado con la batimetría. Se recomienda aplicar este módulo luego de haber realizado aplicado el módulo de obstrucciones de subgrilla generate\_obs*
4. *active\_region\_pol*: *Polígono geopandalizado que contienen la región activa en donde se ejecutará el modelo.*

Parámetros de salida:

1. *mask*: *Numpy 2D array de ceros (drycells), unos (región activa), dos (borde de interacción) y tres (región inactiva).*

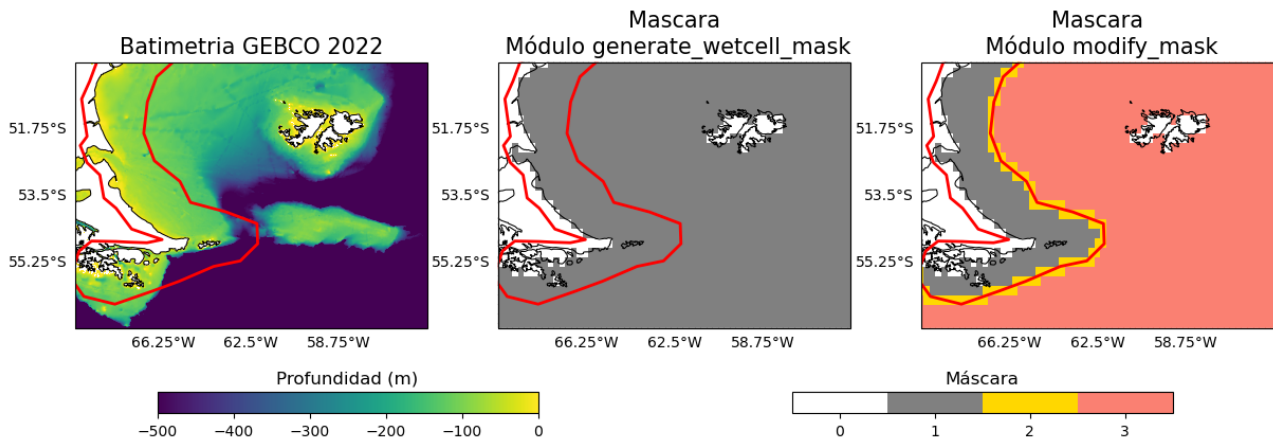
El módulo *modify\_mask*, tal como su nombre lo indica, modifica la codificación de la máscara incorporando nuevos valores (Tabla II). A los valores ya existentes de ceros y unos, se amplía la codificación con los valores dos y tres. Esta nueva codificación sólo está disponible en las versiones de WAVEWATCH III ® mayores a la 3.10, y tiene como objetivo reducir el tiempo de cómputo ejecutando el modelo en un nuevo dominio activo que es menor al dominio rectangular original. Esta nueva región activa trata de emular un dominio que no es estructurado a través de un polígono de entrada, denominado *active\_region*.

**Tabla II:** Codificación de los valores de la nueva máscara generada por el módulo *modify\_mask*.

Codificación	Valor
celda seca	0
celda húmeda activa	1
celda de condición de borde. Se realiza la interacción del dominio padre con el dominio hijo	2
celda húmeda inactiva. No se ejecuta el modelo	3

La Figura 12 muestra un ejemplo de aplicación de esta nueva máscara codificada para la región sur de la Plataforma Patagónica. Nótese que, a pesar de que este dominio sea del tipo rectangular, la nueva región activa en donde el modelo se ejecutará es una región regular pero no estructurada alrededor de la costa (gris). Esta región se definió a partir de un polígono como parámetro de entrada denominado *active\_region* (línea continua roja). A su vez, la interacción de esta nueva región con el dominio padre se realizará únicamente en los bordes de la misma (amarillo). A diferencia de la antigua codificación de la máscara, el resto de la región se considera una región inactiva (rojo salmón).





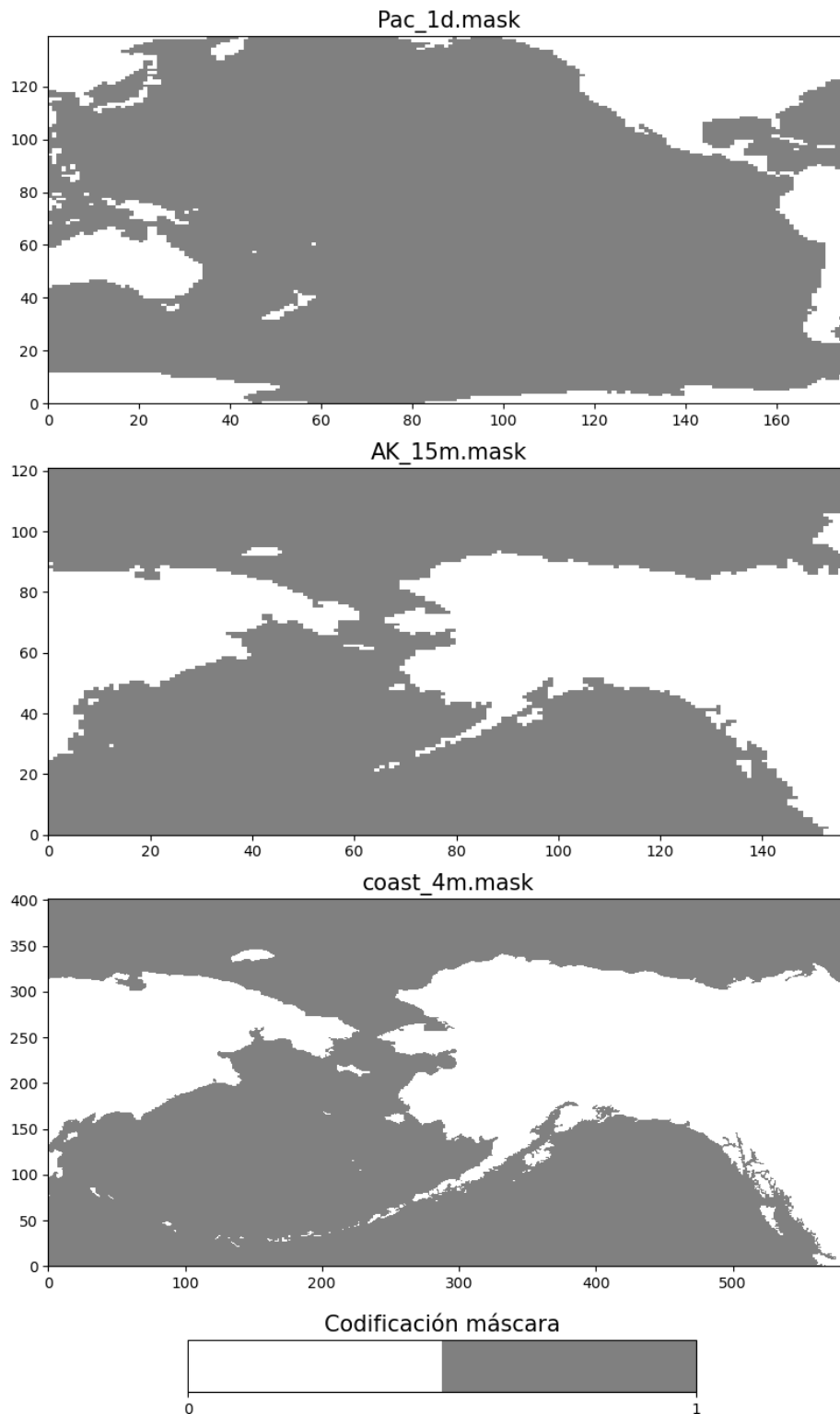
**Fig. 12:** Batimetría de alta resolución GEBCO 2022 (panel izquierdo), máscara generada por el módulo `generate_wetcell_mask` (panel central) para versión WAVEWATCH < 3.10 y el módulo `modify_mask` (panel derecho) en la que se amplía la codificación de la misma (Tabla II) para las nuevas versiones. La línea roja indica el polígono que delimita la región activa.

## 2.6 Módulos Misceláneos

En el sistema de proyecciones WGS84, el más utilizado para dominios globales, la coordenada periódica de longitud se define entre  $-180^\circ$  y  $180^\circ$ . Esta definición genera inconsistencias en el funcionamiento del algoritmo cuando los dominios se construyen en torno al antimeridiano, como por ejemplo un dominio con objetivo de la cuenca del pacífico. A continuación, se diseñaron una serie de módulos que tratan este tipo de dificultades.

- `longitude = checklon_anti(lat_s, lat_n, lon_w, lon_e, dx, polygons)`
- `narray = cuttinglongitude(narray, longitude, lon_w, lon_e, dx)`
- 

En donde los parámetros de entrada para la función `checklon_anti` están dados por los límites de los vectores de latitud (`lat_s`: longitud sur; `lat_n`: longitud norte) y longitud (`lon_w`: longitud occidental y `lon_e`: longitud oriental, ambos entre  $-180$  y  $180$ ), la discretización longitudinal (`dx`) y los polígonos costeros, extraídos de *NaturalEarthFeatures*. Como parámetro de salida, el módulo devuelve una longitud monótonicamente creciente y un conjunto de polígonos que serán utilizados por el resto de los módulos. Mientras que, el módulo `cuttinglongitude` no sólo incorpora como parámetros de entrada a los límites del vector de longitudes (`lon_w` y `lon_e`) sino que además se le ingresan las matrices Numpy 2D que pueden representar máscaras, batimetrías y/o obstrucciones de subgrilla y las longitudes monótonicamente creciente que otorga el módulo `checklon`. Como resultado, `cuttinglongitude` otorga una nueva matriz en “formato antimeridiano” y con las longitudes, monótonicamente crecientes de  $0$  a  $360^\circ$ , que serán requeridas por los módulos de los metadatos. A partir de estos módulos se generó un dominio atravesado por el antimeridiano sobre el Pacífico Norte (Figura 13). Dichos dominios replican a los dominios utilizados por el caso de estudio denominado “*mww3\_case\_02*”, el cual se puede descargar desde [https://ftp.emc.ncep.noaa.gov/static\\_files/public/WW3/ww3\\_from\\_ftp.7.12.6.tar.gz](https://ftp.emc.ncep.noaa.gov/static_files/public/WW3/ww3_from_ftp.7.12.6.tar.gz).



**Fig. 13:** Dominio sobre el Pacífico Norte generado a partir de los módulos `checklon_anti` y `cuttinglongitude`.

Por último, en el caso que no se cuente con una batimetría regular, se tiene un módulo que descarga la batimetría GEBCO<sup>7</sup> (del inglés, General Bathymetric Chart of the Oceans) desde su repositorio de origen.

- `get_gebco(outdir, chunk_size=128)`

## 2.7 Módulos de escritura

- `write_ww3file(fname, narray, N1=None)`
- `write_ww3obstr(fname, Sx, Sy, N2)`
- `write_ww3meta(fname, longitude, latitude, N1-1, N2-1)`

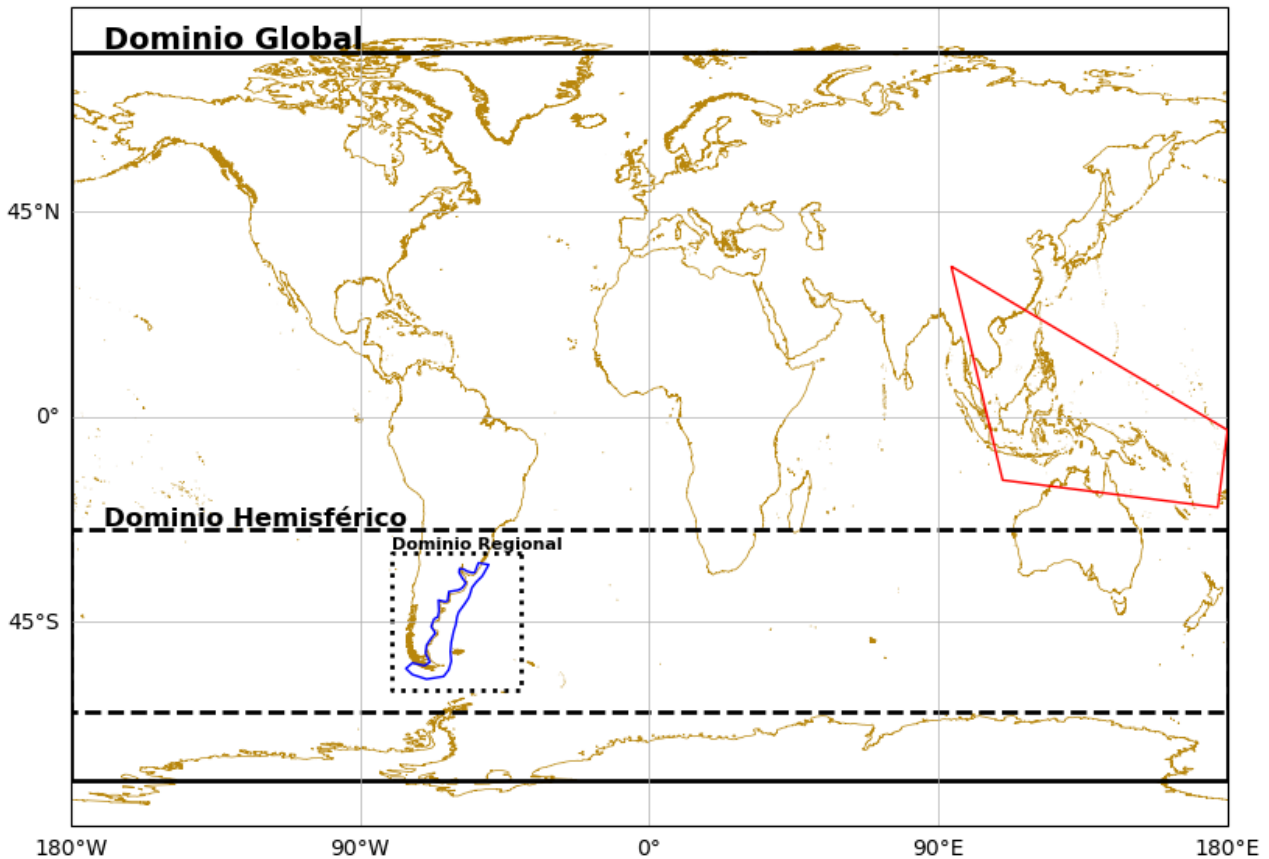
Las batimetrías y las máscaras se escriben en formato de texto mediante el módulo `write_ww3file`, que acepta como parámetros de entrada el nombre `fname` con el que se guardará el archivo, la matriz `ndarray` en formato Numpy 2D array y un factor multiplicativo para evitar los números decimales, salvo para la máscara dada su naturaleza de valores enteros. Las obstrucciones de subgrilla, en cambio, se escriben en un sólo archivo contemplando ambas direcciones `Sx` y `Sy` mediante el módulo `write_ww3obstr`, dicho módulo acepta un factor multiplicativo. En general se recomienda el factor `N1=1000` y `N2=100` para la batimetría y la obstrucción de la subgrilla respectivamente, mientras que para la máscara no se le ingresa dicho valor (`N1=None`). Los metadatos para generar parcialmente el archivo “.inp”, el cual se requiere la ejecución de los módulos del modelo WWIII, se generan mediante la función `write_ww3meta`. La particularidad de esta función es que no requiere las matrices de los dominios, sino los siguientes parámetros de entrada:

1. `fname`: Nombre del archivo de las obstrucciones a guardar
2. `latitude`: Numpy 1D Array de discretización regular
3. `longitude`: Numpy 1D Array de discretización regular
4. `N1-1`: Inversa del factor multiplicativo con el que se ha escrito la batimetría (módulo `write_ww3file`)
5. `N2-1`: Idem `N1-1` pero para las obstrucciones (módulo `write_ww3obstr`).

## 3. EFICIENCIA TEMPORAL

Para evaluar los tiempos de ejecución en la generación de grillas se diseñó un esquema variando la resolución espacial de tres dominios anidados: un dominio global, uno hemisférico y otro regional (Figura 14). Los detalles de la resolución espacial se pueden consultar en la Tabla III. Los casos a2 y d2 difieren de sus originales a1 y d1, ya que se le han agregado polígonos de enmascaramiento y de región activa, respectivamente (ver Figura 14). Para el cálculo de los tiempos de cómputo se utilizó 1 (un) nodo del HPC “Huayra Muyu” del SMN y los dominios se fueron ejecutando de forma secuencial, es decir primero el dominio global, luego el dominio hemisférico, y finalmente, el dominio regional.

<sup>7</sup> [https://www.gebco.net/data\\_and\\_products/gridded\\_bathymetry\\_data/](https://www.gebco.net/data_and_products/gridded_bathymetry_data/)



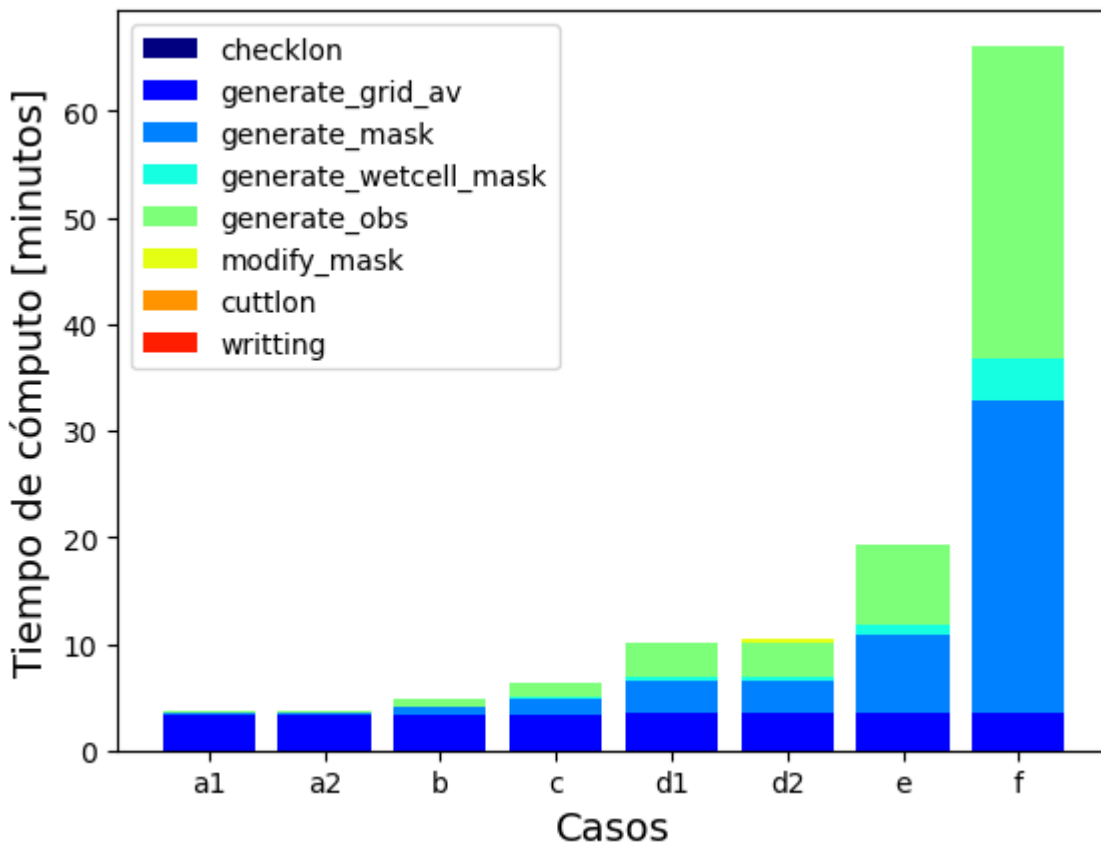
**Fig. 14:** Dominios global (rectángulo sólido), hemisférico (rectángulo rayado) y regional (rectángulo punteado). Polígono de enmascaramiento sobre la región del sudeste asiático (línea roja) y polígono de región activa a lo largo de la costa del dominio regional (línea azul) utilizados en el caso a2 y d2 (ver Tabla III), respectivamente.

**Tabla III:** La diferencia entre los casos a1 y a2 radica en que a este último se le agregó un polígono de enmascaramiento sobre la región del sudeste asiático (ver línea roja en Figura 10) con vértices (94,33), (180,-3), (177,-20), (110,-14), (94,33). Por otra parte, a diferencia de d1, d2 presenta un polígono de región activa a lo largo de la costa del dominio regional (ver línea azul en Figura 14). Las coordenadas deltas se representan como “dx, dy”, mientras que el dominio se representa como “latitud norte, latitud sur, longitud este, longitud oeste”.

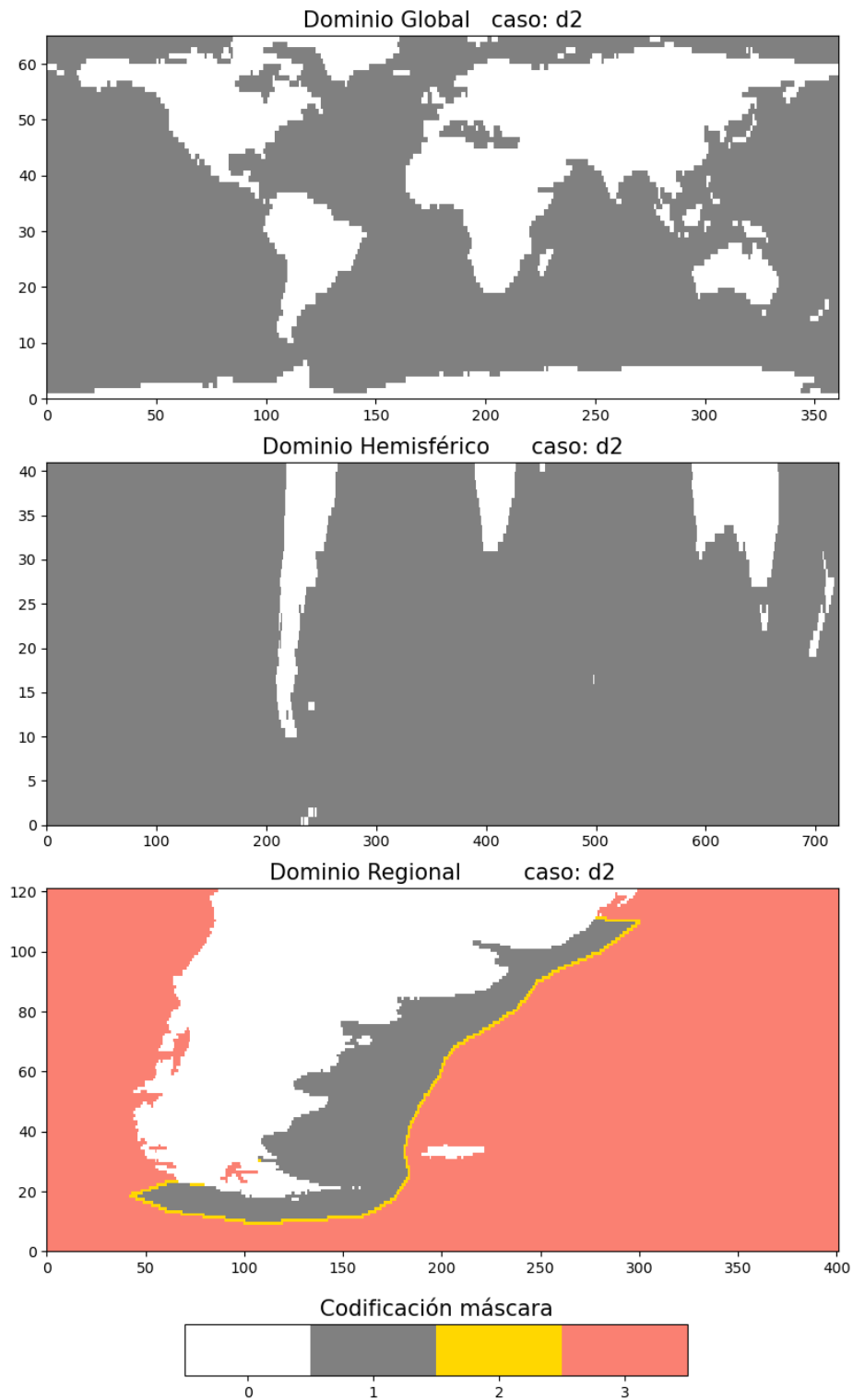
casos	dominio global		dominio regional		dominio local	
	deltas	dominio	deltas	dominio	deltas	dominio
a1	5°, 5°	80, -80, 180, -180	2.5°, 2.5°	-25, -65, 180, -180	1°, 1°	-30, -60, -40, -80
a2	5°, 5°	80, -80, 180, -180	2.5°, 2.5°	-25, -65, 180, -180	1°, 1°	-30, -60, -40, -80
b	2.5°, 5°	80, -80, 180, -180	1°, 2.5°	-25, -65, 180, -180	0.25°, 0.5°	-30, -60, -40, -80
c	2.5°, 2.5°	80, -80, 180, -180	1°, 1°	-25, -65, 180, -180	0.25°, 0.25°	-30, -60, -40, -80
d1	1°, 2.5°	80, -80, 180, -180	0.5°, 1°	-25, -65, 180, -180	0.1°, 0.25°	-30, -60, -40, -80
d2	1°, 2.5°	80, -80, 180, -180	0.5°, 1°	-25, -65, 180, -180	0.1°, 0.25°	-30, -60, -40, -80

e	1°, 1°	80, -80, 180, -180	0.5°, 0.5°	-25, -65, 180, -180	0.1°, 0.1°	-30, -60, -40, -80
f	0.5°, 0.5°	80, -80, 180, -180	0.25°, 0.25°	-25, -65, 180, -180	0.05°, 0.05°	-30, -60, -40, -80

Como es esperable, el tiempo de cómputo (Figura 15) aumenta a medida que la resolución de los dominios aumenta, se puede demorar desde 5 hasta 60 minutos en generar de forma secuencial los tres dominios en baja y alta resolución, respectivamente. Esta relación no es aplicable cuando se distingue el tiempo de cómputo por módulo individual. Por ejemplo, se observa que el tiempo de cómputo por el módulo *generate\_grid\_av* se mantiene constante al aumentar la resolución de los dominios. Por el contrario, los módulos más afectados por el cambio de resolución están relacionados al cómputo de la máscara por polígonos costeros y a las obstrucciones de subgrilla, los módulos *generate\_obs* y *generate\_mask*, respectivamente. Una mayor cantidad de nodos disponibles podría paralelizar la generación de los dominios, no así la secuencialidad de los módulos.



**Fig. 15:** Suma del tiempo de cómputo de los tres dominios (global, hemisférico y regional) en función de los casos y de los módulos (ver etiquetas).



**Fig. 16:** Dominio en puntos de retícula (adimensional) para el caso "d2" (ver Tabla III).

## 6. CONCLUSIONES

Mediante la utilización de código abierto Python, se desarrollaron una serie de módulos capaces de generar dominios regulares necesarios para la ejecución del modelo de olas WAVEWATCH III®, brindando así la posibilidad de estar en un estado del arte permanente. Se estudió la eficiencia de los módulos en la generación de distintas combinaciones de 3 dominios: global, hemisférico y regional, los resultados utilizando sólo 1 (un) nodo del HPC “Huayra Muyu” del Servicio Meteorológico Nacional muestra que los dominios de alta resolución se generan en 1 hora. Se espera que la paralelización mejore aún más la eficiencia temporal de la generación de los dominios. La generación rápida y automática de dominios permitirá explorar el desempeño de los mismos y de las obstrucciones de subgrilla en el bloqueo del mar de fondo potencialmente propagante hacia el Mar Argentino. Estos estudios se enmarcan dentro de la actualización del sistema de pronósticos marinos que se está llevando a cabo dentro del Servicio Meteorológico Nacional en el marco del proyecto PronoMAR. Finalmente, dado que es un módulo dinámico, el mismo está sujeto a actualizaciones constantes, se recomienda ejecutar los ejemplos mencionados en el gitlab y si existen inconvenientes realizar las sugerencias correspondientes.

## REFERENCIAS

Chawla, A. and H. L. Tolman, 2007: Automated grid generation for WAVEWATCH III. Technical note 254, NCEP/NOAA/NWS, National Center for Environmental Prediction, Washington DC.

Tolman, H. L., 2002b: Testing of WAVEWATCH III version 2.22 in NCEP’s NWW3 ocean wave model suite. Technical note 214, NCEP/NOAA/NWS, National Center for Environmental Prediction, Washington DC.

Tolman, H. L., 2003: Treatment of unresolved islands and ice in wind wave models. Ocean Modelling, 5, 219 – 231.

## Instrucciones para publicar Notas Técnicas

En el SMN existieron y existen una importante cantidad de publicaciones periódicas dedicadas a informar a usuarios distintos aspectos de las actividades del servicio, en general asociados con observaciones o pronósticos meteorológicos.

Existe no obstante abundante material escrito de carácter técnico que no tiene un vehículo de comunicación adecuado ya que no se acomoda a las publicaciones arriba mencionadas ni es apropiado para revistas científicas. Este material, sin embargo, es fundamental para plasmar las actividades y desarrollos de la institución y que esta dé cuenta de su producción técnica. Es importante que las actividades de la institución puedan ser comprendidas con solo acercarse a sus diferentes publicaciones y la longitud de los documentos no debe ser un limitante.

Los interesados en transformar sus trabajos en Notas Técnicas pueden comunicarse con Ramón de Elía ([rdelia@smn.gov.ar](mailto:rdelia@smn.gov.ar)), Luciano Vidal ([lvidal@smn.gov.ar](mailto:lvidal@smn.gov.ar)) o Martín Rugna ([mrugna@smn.gov.ar](mailto:mrugna@smn.gov.ar)) de la Dirección Nacional de Ciencia e Innovación en Productos y Servicios, para obtener la plantilla WORD que sirve de modelo para la escritura de la Nota Técnica. Una vez armado el documento deben enviarlo en formato PDF a los correos antes mencionados. Antes del envío final los autores deben informarse del número de serie que le corresponde a su trabajo e incluirlo en la portada.

La versión digital de la Nota Técnica quedará publicada en el Repositorio Digital del Servicio Meteorológico Nacional. Cualquier consulta o duda al respecto, comunicarse con Melisa Acevedo ([macevedo@smn.gov.ar](mailto:macevedo@smn.gov.ar)).