

Manual restricciones

Rodrigo Cifuentes Lobos

Abril 2022

1 Trabajo previo

Previo a la fase de restricciones de modelos, son cuatro los pasos principales que deben tomarse.

1. Creación de modelos aleatorios de deslizamiento
 - Elección de parámetros de árbol lógico
2. Cálculo de campos de deformación vertical a partir de los modelos aleatorios de deslizamiento
3. Recopilación de datos de deformación, para lo que hay dos opciones
 - Datos cuantitativos (valor de deformación en metros, positivo para alzamiento, negativo para subsidencia)
 - Datos cualitativos (Descripción de si coordenada sufrió alzamiento o subsidencia)
 - El algoritmo no es sensible al ordenamiento de los datos. Sin embargo, para temas de análisis, gráfico de mapas, entre otros procesos posteriores a la creación y restricción de modelos, se sugiere que estos estén ordenados de norte a sur
4. Recopilación de datos de tsunami
 - Pueden ser de altura de ola, inundación o *run-up*
 - La comparación y restricción con los modelos se realiza a través de series de tiempo de mareógrafos virtuales, que están ubicados en el mar. Si los datos están tomados en tierra (inundación o *run-up*), debe ser utilizada alguna relación que los transforme a altura de ola, para ser posicionados en el océano en la misma coordenada del mareógrafo virtual. Se sugiere utilizar la relación dada por Smart *et al.*, 2016.
 - El orden de estos datos **sí** debe ser de norte a sur. Al contrario del caso de los datos de deformación, el proceso de restricción con datos de tsunami **sí** es sensible al orden.

1.1 Funciones

Los primeros dos pasos son realizados de forma secuencial con el mismo programa, `creador_fallas_dom_lay.py`, ubicado en `suite/creacion_modelos`. Una vez decididos los rangos de valores para ser utilizados en las ramas del árbol lógico para la creación de los modelos aleatorios, estos deben ser ingresados en las entradas correspondientes de `creador_fallas_dom_lay.py`, según indica la figura 1.

```
40  """
41  DEFINICION DE INTERVALOS DEL ARBOL LOGICO
42  """
43  N_arbol      = np.array([14,16,18,20])      # complejidad
44  Mw_arbol     = np.linspace(8.6,9.0,5)       # magnitud
45  LN_arbol     = np.linspace(-39.0,-38.0,5)    # Limite norte
46  LS_arbol     = np.linspace(-44.5,-43.0,5)    # Limite sur
47  sflats_arbol = np.array([80.])              # cantidad de subfallas en latitud (along strike) para setear razon de aspecto
48  dist_fosa_arbol = np.array([0., 20., 40., 60.]) # distancias (en km) del limite oeste de la ruptura con respecto de la fosa
49  niterslips   = 100                          # cantidad de modelos de slip creados para cada combinacion de parametros
```

Fig. 1: Variables de entrada para intervalos de ramas del árbol lógico

1.2 Archivo de datos de restricción de deformación

Para los pasos 3 y 4, el formato de los archivos de datos es flexible, siempre que el usuario se cerciore de que las funciones están adaptadas correctamente para la lectura de sus archivos. Por defecto, las funciones leen archivos con los siguientes formatos.

1.2.1 Cuantitativo

Por defecto es un archivo de tres columnas separadas por espacios con extensión **.xyz**. La primera contiene la longitud entre -180° a 180° , la segunda la latitud (-90° a 90°) y la tercera la deformación en metros.

lon 1 (-180° a 180°)	lat 1 (-90° a 90°)	deformación 1
lon 2 (-180° a 180°)	lat 2 (-90° a 90°)	deformación 2
\vdots	\vdots	\vdots
lon n (-180° a 180°)	lat n (-90° a 90°)	deformación n

Table 1: Formato soportado por defecto por funciones de restricción, visualización y análisis de datos de deformación cuantitativos.

1.2.2 cuantitativo

Por defecto es un archivo de tres columnas separadas por espacios con extensión **.xyz**. Al igual que en el caso anterior, las primeras dos columnas son: longitud entre -180° a 180° y la latitud (-90° a 90°). Mientras la tercera es un entero positivo en caso de alzamiento o negativo en caso de subsidencia.

lon 1 (-180° a 180°)	lat 1 (-90° a 90°)	-1000
lon 2 (-180° a 180°)	lat 2 (-90° a 90°)	-1000
\vdots	\vdots	\vdots
lon n (-180° a 180°)	lat n (-90° a 90°)	1000

Table 2: Formato soportado por defecto por funciones de restricción, visualización y análisis de datos de deformación cuantitativos. En este ejemplo, las primeras dos filas corresponden a subsidencia y la n-ésima a alzamiento.

1.3 Archivo de datos de restricción de tsunami

Por defecto es un archivo de cuatro columnas separadas por espacios con extensión **.xyz**. La primera contiene el nombre de la localidad (si el nombre contiene espacios, deben ser reemplazados con guiones, e.g.: Desembocadura del Bio Bio \rightarrow Desembocadura-del-Bio-Bio) la segunda la longitud entre -180° a 180° , la tercera la latitud (-90° a 90°) y la cuarta el dato de tsunami en metros.

Localidad 1	lon 1 (-180° a 180°)	lat 1 (-90° a 90°)	tsu 1
Localidad 2	lon 2 (-180° a 180°)	lat 2 (-90° a 90°)	tsu 2
\vdots	\vdots	\vdots	
Localidad n	lon n (-180° a 180°)	lat n (-90° a 90°)	tsu n

Table 3: Formato soportado por defecto por funciones de restricción, visualización y análisis de datos de deformación cuantitativos. En este ejemplo, las primeras dos filas corresponden a subsidencia y la n-ésima a alzamiento.

2 Restricción por datos de deformación

La restricción por datos de deformación consta de 3 pasos que **deben** ser realizados en el orden indicado. El algoritmo de estimación es **muy** sensible al orden de restricciones, por lo que un cambio en este puede entregar

resultados con grandes sesgos.

Estos pasos son:

1. Restricción por magnitud de datos de deformación
2. Restricción por coincidencias con datos de deformación
3. Restricción por pesos de datos, relacionados a la densidad de la localización geográfica de estos

2.1 Restricción por magnitud de datos de deformación

Dado un terremoto de subducción, la magnitud de las deformaciones verticales en superficie provocados por él son proporcionales a la cantidad de deslizamiento de este terremoto. No se espera que las más grandes deformaciones se encuentren "frente" a valores bajos de deslizamiento, si no que los valores mayores de deformación medidos en tierra deberían estar relacionados con los parches principales de deslizamiento del terremoto. Así, por ejemplo, para el caso chileno, las mayores deformaciones deberían tener latitudes similares a las mayores cantidades de deslizamiento.

Esta restricción elimina aquellos modelos aleatorios cuyos principales parches de *slip* no se encuentren, bajo cierta tolerancia definida por el usuario y dependiente de la magnitud del terremoto, en frente de los mayores valores, en magnitud, de deformación.

Esta restricción es realizada por el programa `restringe_magnitud_datos_deformacion.py`, ubicado en `suite/restriccion`.

Este programa funciona particionando la zona de ruptura asociada al terremoto a estimar en n franjas, siendo n un valor arbitrario.¹ Luego se ubica la franja donde está la mayor deformación y, para cada modelo aleatorio, se compara esta posición con la posición de su mayor parche de deslizamiento, con un rango de tolerancia en grados. Si ambas ubicaciones coinciden, el modelo aleatorio no es descartado. Si estas **no** coinciden, se considera cuan lejos están estos máximos, en términos de cuántas franjas los separan y de acuerdo a esta distancia se le otorga un valor de penalización (mayor mientras más lejos se encuentren) definido por el usuario al inicio. Finalmente este valor de penalización es comparado con un valor aleatorio con distribución uniforme, si la penalización es mayor que el número aleatorio, es descartado el modelo. Así, mientras más lejos se encuentren, mayor es la probabilidad de que el modelo aleatorio sea descartado.

Las entradas definidas por el usuario son:

- Los límites de la zona de ruptura
- Número de franjas
- Rango de tolerancia para el parche principal (*buffer*)
- Penalización

La ubicación de estos es detallada por la figura 2

2.2 Restricción por coincidencias con datos de deformación

Una vez que se ha restringido los modelos descartando aquellos cuyas ubicaciones de sus respectivos parches principales no se condicen con los máximos de deformación, se utilizan los datos de deformación para descartar aquellos modelos aleatorios de deslizamiento que no generen campos de deformación que satisfagan las observaciones.

Para esto, se analiza la cantidad de coincidencias, con una determinada tolerancia, entre los datos de deformación y las correspondientes coordenadas en los campos de deformaciones obtenidos a partir de los modelos aleatorios de deslizamiento. El programa interpola los valores del campo de deformación a las coordenadas de los datos de deformación y compara los valores.

- En el caso cualitativo, compara si los valores interpolados del campo tienen el mismo signo que los datos, significando que en el campo aleatorio se tiene la misma dirección de deformación.
- En el caso de datos cuantitativos, se compara si los valores interpolados se encuentran dentro de un rango $[d_i - \epsilon_i, d_i + \epsilon_i]$, donde d_i es el i -ésimo dato de deformación en metros y ϵ es el valor de tolerancia en metros para los datos de deformación.

¹Por defecto, este valor es $n=5$. Se recomienda aumentar este valor si se tiene una gran cantidad de datos a lo largo de la zona de ruptura y/o datos con gran densidad geográfica.

```

14  """
15  PARAMETROS DE FRANJAS LATITUDINALES
16  """
17  lat_lim_norte = -36.0 # latitud limite norte de zona de ruptura
18  lat_lim_sur   = -46.0 # latitud limite sur de zona de ruptura
19  n_franjas     = 5     # cantidad de franjas que subdividirán la zona de ruptura
20  # limites de franjas
21  limites_franjas = np.linspace(lat_lim_norte, lat_lim_sur, n_franjas+1)
22
23  """
24  PARAMETROS DE RESTRICCION
25  """
26  # tolerancia hacia el norte y sur en latitud del maximo dato de deformacion para encontrar el parche de slip
27  buffer_latitud = 0.2
28  # porcentaje de modelos restringidos por cada franja de distancia con las franjas del rango
29  # ejemplo:
30  # rango:[1,2]
31  # decaimiento de franja 0 sera 20 %, decaimiento de franja 4 sera 20x2=40 %
32  decaimiento    = 40

```

Fig. 2: Variables de entrada para restricción por coincidencia de máximos.

Este proceso es realizado por el programa `restringe_modelo_deformacion_interpolacion.py`. El usuario debe ingresar:

- Tipo de restricción. `flag_tipo`
 - 0: para datos cuantitativos y tolerancias fijas en metros. $[d_i - \epsilon_i, d_i + \epsilon_i]$ con ϵ fijo en metros definido por el usuario
 - 1: para datos cualitativos
 - 2: para datos cuantitativos y tolerancias en función del valor del dato de deformación. $[d_i - \epsilon_i, d_i + \epsilon_i]$ con $\epsilon_i = p \times d_i$, donde p es un porcentaje ingresado por el usuario ($r=\text{tol_porcen}$).
- Tolerancia de coincidencias. `tol`.
 - Cantidad de datos que se permite que **no** sean coincidentes con el valor interpolado del campo de deformación. $\{\text{tol} \in \mathbb{Z}\}$. $0 \leq \text{tol} \leq n_{\text{total}}$. Con n_{total} el total de datos de deformación. Si `tol=0`, **todos** los valores deben satisfacer las observaciones. Si `tol=2`, puede haber dos valores que **no** coincidan y **aun así** el modelo **no** será descartado.
- Rango de tolerancia. `tol_metros`
 - Si se elige `flag_tipo=0`, este valor indica ϵ fijo en metros.
- Rango de tolerancia porcentual. `tol_porcen`
 - Si se elige `flag_tipo=2`, este valor indica p en $\epsilon_i = p \times d_i$, para i -ésimo dato.
- Directorio de destino para modelos que pasen la restricción. `dir_destino`
- Ruta relativa (a directorio con funciones) a directorio con datos de deformación. `ruta_paleodatos`
- Nombre de archivo con datos. `arch_paleodatos`
- Ruta relativa a directorio donde se encuentra los modelos aleatorios que serán sometidos a la restricción. `path_fallas`

Estos *inputs* se muestran en la figura 3

2.3 Restricción por pesos de datos, relacionados a la densidad de la localización geográfica de estos

En la medida que no se tenga una distribución homogénea y densa de datos a lo largo de la zona de ruptura, habrá algunas zonas que tendrán más "poder restrictivo" al tener más datos y requerir que en ese sector el campo de deformación sea más específico en su distribución, para satisfacer a las observaciones. Por otro lado, zonas con

```

20 ***
21 PARAMETROS DE RESTRICCION
22 ***
23 flag_tipo = 0          # switch para controlar si se quiere restringir con coincidencia de signos o con valores de deformacion
24 # 0: valores de deformacion
25 # 1: coincidencia signo
26 # 2: diferencia porcentual
27 tol = 5                # tolerancia de datos
28 tol_metros = 1.5       # tolerancia en metros de rango de restriccion
29 tol_porcen = 75.0       # porcentaje de tolerancia para los datos
30
31
32
33 ***
34 CARGA DATOS
35 ***
36 # ruta del archivo de la fosa
37 ruta_fosa = "../Slab/"
38 # archivo fosa ( primera columna: longitudes, segunda columna: latitudes)
39 arch_fosa = ruta_fosa + "SAM2.txt"
40 # carga de fosa usando funcion del modulo modfallas
41 lonfosa, latfosa = mf.carga_fosa(arch_fosa)
42 # ruta datos de deformacion
43 ruta_paleodatos = "../input/" # ruta relativa a directorio con fallas
44 arch_paleodatos = "def2_obs.xyz"
45 paleodatos = ruta_paleodatos + arch_paleodatos
46 agno = 1960
47
48 ***
49 INFO DE DIRECTORIOS
50 ***
51 dir_destino = "modelos_restringidos"
52 dir_actual = os.getcwd()
53 path_fallas = "../fallas2/modelos_restringidos magnitud datos"

```

Fig. 3: Variables de entrada para restricción por coincidencia con datos de deformación.

menos datos tendrán restricciones más laxas, dejando que puedan pasar (no ser descartados) modelos que en esas zonas tengan distribuciones de deslizamientos que no necesariamente se ajusten a la realidad del terremoto a ser estimado, pero que, mientras en el resto de la zona de ruptura **sí** satisfaga las observaciones, junten el total de coincidencias para **no** ser descartados. Por ejemplo, un terremoto en la zona de ruptura de Valdivia tendría muy difícilmente una gran y/o densa red de observaciones al sur de Aysén (que por problemas logísticos y lo remoto de la zona hace que sea muy difícil medir), sin embargo, más al norte, podría tener muchos datos. Esto podría resultar en que modelos que tengan grandes parches de deslizamiento en la zona de las Guaitecas y Taitao puedan pasar, ya que en esa zona **no** hay datos para constreñir esos parches.

La solución que se propone para este problema es darle más peso a las zonas con mayor cantidad de datos, y penalizando aquellas que tengan menos. Análogamente a la restricción 1), se divide la zona de ruptura en un número arbitrario de franjas (5 por defecto, pero el número puede ser variado por el usuario) y se calcula el porcentaje de datos en cada una de ellas. Se elige como referencia aquella que tenga mayor porcentaje de datos y en función a las proporciones entre los porcentajes de las otras franjas y la franja de referencia, se define un valor de penalización. El valor de penalización de la franja con mayor porcentaje es **siempre** cero. Si más de una franja tiene el mismo porcentaje mayor, ambas son utilizadas como referencia y se les otorga una penalización de cero.

Estas penalizaciones p son utilizadas para descartar un número $n(p)$ de modelos, según los siguientes criterios.

1. Ninguno de los modelos cuyos parches principales de deslizamiento se encuentren en la(s) franja(s) de referencia es descartado.
2. De los modelos cuyos parches de deslizamiento se encuentren fuera de la(s) franja(s), se descarta un porcentaje en función a la penalización dada por una función de las razones de los porcentajes de referencia y la franja n -ésima.
 - Por defecto, este porcentaje p_n de modelos descartados se calcula con regla de tres simple. Para la franja n -ésima:
 - $$p_n = 100 \times \left(1 - \frac{\% \text{franja}_n}{\% \text{referencia}}\right)$$
 - Si una franja tiene 0 % de datos, se descarta el 90 % de ellos.
 - Independiente de la forma por defecto de calcular el porcentaje de penalización p , al ser este un valor ingresado por el usuario, este cálculo puede ser reemplazado por otro según se estime conveniente
3. Por cada franja distinta a la de referencia, se crea una lista con los modelos cuyos parches principales se encuentren en esta y luego, de acuerdo a p_n , se muestrea aleatoriamente el porcentaje correspondiente y se descarta, dejando pasar $(100 - p_n)$ % de los modelos de la n -ésima franja

4. Este proceso debe ser repetido para cada una de las franjas que no sean las de referencia.

El programa `restringe_concentracion_datos_deformacion.py` de `suite/restriccion` se encarga de este proceso, franja por franja. El usuario debe ingresar:

- Límites norte y sur de la zona de ruptura. `lat_lim_norte` y `lat_lim_sur`, respectivamente
- El número de franjas en el que se desea dividir la zona de ruptura, siendo $n=5$ por defecto. `n_franjas`
- La franja a la que se quiere realizar la restricción. `franja_excluir`
- Porcentaje de modelos a excluir (p_n). `porcentaje_exclusion`
- Directorio con modelos aleatorios. `ruta_slip`
- Directorio de destino para los modelos **no** descartados. `dir_destino`

Estos valores pueden encontrarse en la figura 4

```
14 ***
15 PARAMETROS DE FRANJAS LATITUDINALES
16 ***
17 lat_lim_norte = -36.0 # latitud limite norte de zona de ruptura
18 lat_lim_sur   = -46.0 # latitud limite sur de zona de ruptura
19 n_franjas     = 5      # cantidad de franjas que subdividirán la zona de ruptura
20 # franja que se quiere excluir
21 franja_excluir = 5 # NO es un índice, es el número de la franja des 1 a n, índice sería franja_excluir - 1
22 # porcentaje de elementos a excluir
23 porcentaje_exclusion = 90
```

Fig. 4: Variables de entrada para restricción por peso de datos.

Una vez son realizadas estas restricciones, en este preciso orden, se puede pasar al siguiente paso de modelación y restricción por datos de tsunami.

3 Restricción por datos de tsunami

La restricción por datos de tsunami es también un proceso iterativo, en el que el orden de operaciones es **muy** importante, y el no seguirlo puede desembocar en sesgos en la estimación de fuente sísmica. En su forma más básica, este proceso consta de dos partes,

1. Modelación de tsunami
2. Restricción por datos de tsunami

Estas partes tienen, a su vez, distintos pasos a seguir en un proceso iterativo, hasta llegar a los modelos finales.

3.1 Modelación de tsunami

La modelación de tsunami se realiza a través del programa `COMCOTv1.7`, con el que se calculan series de tiempo de mareógrafos virtuales ubicados en las posiciones donde se tiene los datos. Como fue mencionado en la sección 1 **Trabajo previo**, numeral 4, puntos 2 y 3, los archivos de datos de tsunami **deben** estar localizados en el mar para la comparación con mareógrafos virtuales (de ahora en adelante TS, según la nomenclatura de `COMCOT`) y **ordenados estrictamente de norte a sur**.

Las limitaciones dadas por los altos costos computacionales que significarían modelar un tsunami en un área tan grande como una zona de ruptura, con grillas de alta resolución en los lugares donde se tiene datos, además del máximo de 12 grillas que soporta `COMCOT` por modelación, obligan al usuario a dividir el proceso de modelación por áreas. Estas áreas dependerán de la disponibilidad de datos y de grillas de alta resolución. Para la modelación, son necesarios **tres** pasos.

1. Transformar los modelos de deformación de formato `.tt3`, nativo de `Clawpack`, a `.csv`. Esto es realizado con el programa `transforma_deformación_grilla2xyz.py` ubicado en `suite/aux`

2. Definir las áreas para modelación y las grillas que corresponderán a cada una de ellas.

- Se aconseja tener en cuenta la posición de los datos en las distintas áreas, definiendo que para la primera le corresponden los datos de la fila 1 a la i -ésima, a la segunda área de la $i + 1$ -ésima a la j -ésima, y así.

3. Definir el orden de modelación de las áreas. El algoritmo es sumamente sensible a este paso.

Para el caso del paso 3 mencionado anteriormente, es necesario tomar en cuenta dos factores, las magnitudes de los datos (de deformación y tsunami) y la cantidad y densidad de datos de tsunami por área de modelación. Se espera que la magnitud de los datos de tsunami tenga alta correspondencia con las magnitudes de los datos de deformación. Sin embargo, dado que no siempre es conocido si el dato de tsunami corresponde a la primera o n -ésima ola, si corresponde a la ola más "grande" registrada en esa ubicación, si se toma o no en cuenta posibles efectos locales que pueden interferir constructiva o destructivamente con la ola, entre otros, esta correspondencia no siempre es clara. Para la definición del orden, se sugiere tomar en cuenta los siguientes criterios,

1. Comparar perfiles de magnitud de datos de deformación y tsunami. Si coinciden, comenzar la modelación con el área que contenga los maximos datos de tsunami.
2. Si no coinciden, comenzar con el área que tenga más porcentaje de datos. Si hay más de un área con el mismo porcentaje, comenzar con la que tenga datos de mayores amplitudes.
3. Para definir las siguientes áreas, continuar con aquellas que tengan menos datos en orden decreciente de porcentajes.

Una vez elegido el orden de modelación, se debe definir un archivo de entrada `ts_location.dat` con las ubicaciones de los TS dentro del área, correspondientes a las ubicaciones de los datos dentro del área y definir qué grillas serán utilizadas. Con la información de grillas, se debe rellenar la función `formato_comcotctl.py` según instrucciones del manual de modelación. Finalmente, el proceso de creación de directorios específicos para cada modelo, así como la creación de archivos de control para COMCOT, son gestionados por `modela_tsunami.py`. Al correr este programa, se debe seguir las instrucciones dadas por la terminal. Más detalles de este proceso están disponibles en el manual mencionado anteriormente.

3.2 Restricción

Una vez realizadas las modelaciones, se debe realizar la restricción con datos de tsunami. Esta restricción, análogamente a la restricción por coincidencias en deformación, compara los datos de observaciones d_t con los máximos valores η_{max} obtenidos por los mareógrafos virtuales correspondientes, considerando un rango de tolerancia. De manera que si $d_t \in [\eta_{max} - \epsilon, \eta_{max} + \epsilon]$, donde ϵ es el valor de tolerancia en metros, el punto pasa la restricción. Si un porcentaje determinado de puntos, determinado por una tolerancia definida por el usuario, pasa la restricción, el modelo **no** es descartado. En el caso contrario, **sí** se descarta.

Dependiendo de la disponibilidad de datos de marea o no, se puede utilizar uno de los dos programas para restricción de tsunami, respectivamente.

- `restringe_tsunami_conmarea.py`
- `restringe_tsunami_sinmarea.py`

En el primer caso, el programa lee un archivos con datos pleamar y bajamar para los puntos de los TS y añade estos valores a la comparación, incluyéndolos en el rango $d_t \in [\eta_{max} - \epsilon - \text{bajamar}, \eta_{max} + \epsilon + \text{pleamar}]$.

Los *inputs* que deben ser definidos por el usuario son ambos valores de tolerancias y las rutas a directorios relevantes para la restricción, como muestra la siguiente lista y la figura 5.

- Tolerancia de coincidencias. `tol`
- Tolerancia en metros. `tol_metros`
- Directorio con modelos aleatorios. Desde este directorio serán copiados aquellos que pasen la restricción. `dir_fallas`
- Directorio donde se guardan los directorios individuales de modelaciones. `dir_modelaciones`

- Ruta relativa a archivo con datos de tsunami. `dir_input_altura`
- Nombre del archivo con datos de tsunami. `archivo_altura`
- Nombre del archivo con datos de marea. Debe estar en el mismo directorio que `archivo_altura`. `archivo_marea`
- Directorio de destino donde se copiará los modelos que pasen la restricción. **Debe** ser un subdirectorio de `dir_modelaciones`. `dir_mods_rests`

Los dos primeros parámetros son análogos al caso 2.2. En el caso de no contar con un archivo de mareas, el proceso es análogo al anterior, pero no es necesario definir `archivo_marea`.

```

16 """
17 PARAMETROS DE COMPARACION
18 """
19 # tolerancia de puntos
20 tol = 0
21 # tolerancia en metros
22 tol_metros = 0
23
24 """
25 INFO DE DIRECTORIOS
26 """
27 dir_fallas = "../../fallas2/modelos_restringidos/restriccion_fuerte"
28 dir_modelaciones = "../ctl_tsunami/norte_restriccion_fuerte" # ruta al directorio donde se guardan los modelos
29 dir_input_altura = "../input" # directorio donde se encuentra el archivo con altura
30 archivo_altura = "AlturaMaxenMetros.txt" # archivo de datos de altura
31 archivo_marea = "nivel_mareas_tsunami.csv"
32 dir_mods_rests = "modelos_restringidos" # directorio donde se copiara los modelos que pasen las restricciones
33 dir_actual = os.getcwd() # ruta al directorio principal
34

```

Fig. 5: Variables de entrada para restricción de tsunami.

En el caso de no ser la primera área la que se esté modelando, se debe tomar en cuenta la ubicación de los datos en el archivo de localizaciones completo de datos, `ts_location.dat`, esto se debe cambiar en la variable `idx_orden` según indica la figura 6. Es necesario tener en cuenta que dado que este es un índice en Python, este parte de cero.

```

54 """
55 LOOP DE RESTRICCION
56 """
57 # lista donde se guardara los modelos que pasen las restricciones
58 modelos_ok = []
59 idx_orden = 15
60 # se corta el archivo de datos de acuerdo a cantidad de ts
61 lon = lon[idx_orden:n_ts+idx_orden]
62 lat = lat[idx_orden:n_ts+idx_orden]
63 altura = altura[idx_orden:n_ts+idx_orden]

```

Fig. 6: Ubicación de la variable `idx_orden`.

Para continuar con las siguientes áreas de modelaciones, basta con cambiar la ruta de directorio de origen en `modela_tsunami.py` a la ruta de destino de la restricción anterior y volver a correr `modela_tsunami.py` con las grillas correspondientes, para generar los directorios de modelación y sus respectivos archivos de control.

4 Esquema

Diagrama de flujo del proceso de restricción de modelos. Es importante destacar que durante este proceso **ningún** archivo es eliminado, sólo son copiados a directorios distintos aquellos que **no** son descartados.

