

# Croco\_pytools

Une version python des croco tools

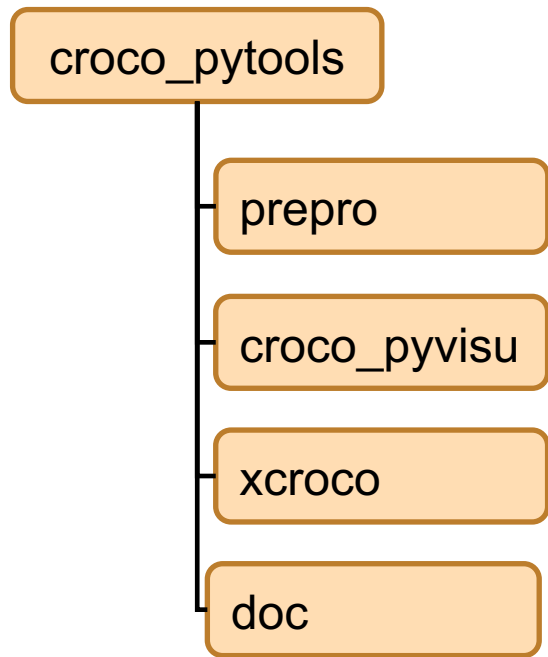
# Vous avez dit croco\_pytools?

- Forte demande d'outils python
- Volonté de regrouper les scripts faits par plusieurs personnes
- Avoir une première brique pour faire évoluer les outils aux différents usages



Création CROCO\_PYTOOLS

## Structure des croco\_pytools



# Philosophie des outils de pre-processing

- Un script par étape de pré-processing (grille, ibc, marée, rivière)
- Utilisation d'un reader, e.g. dictionnaire, pour « décrypter » le format des données d'inputs
  - Facilite la lecture de nouveaux formats
  - Evite l'étape de prétraitement des données des croco\_tools matlab
- S'appuient pour le moment en partie via f2py sur des outils de A. Shepetkin
  - réflexion / bench à mener sur l'utilisation de f2py, xarray/dask, ou numba pour la gestion des grosses grilles



```
if input == 'mercator':
    dico={ 'depth': 'depth', \
          'lonr': 'longitude', 'lonu': 'longitude', 'lonv': 'longitude', \
          'latr': 'latitude', 'latu': 'latitude', 'latv': 'latitude', \
          'ssh': 'zos', \
          'temp': 'thetao', \
          'salt': 'so', \
          'u': 'uo', \
          'v': 'vo', \
          'time': 'time', \
          'time_dim': 'time' \
    }
```

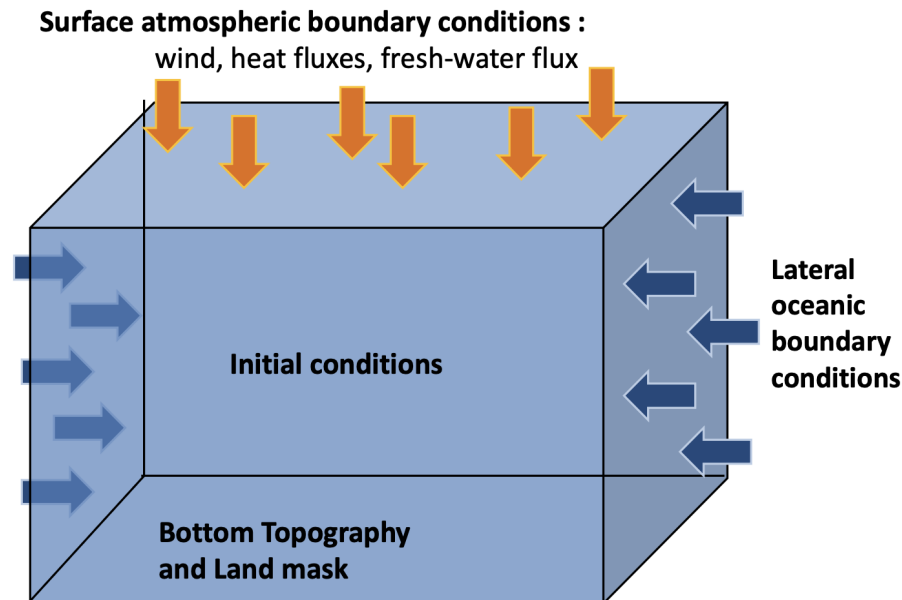
# Contenu actuel des outils (V 1.0)

## Pre-processing

- Bathymetrie
- Initial/boundary conditions (IBC)
- Marée
- Rivières

Datasets globaux d'entrée:

- GEBCO,ETOPO,SRTM
- Mercator, WOA,SODA,ECCO
- TPXO,FES,Previmer
- Données issues de **Climate Data Store**



# Etapes de création de grille

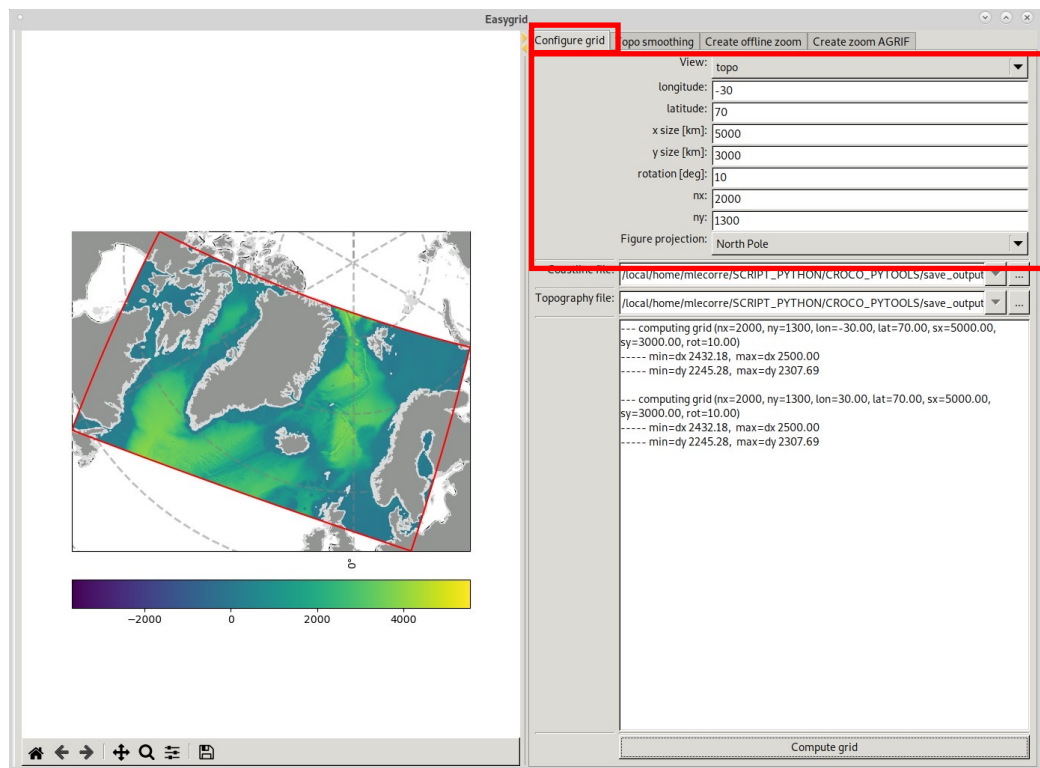
- Pour les grilles parents

## Etapes construction d'une grille:

1. **Placement de grille**
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

### Informations complémentaires:

- Projection de Mercator à l'équateur
- Possibilité de rotation de grille



# Etapes de création de grille

- Pour les grilles parents

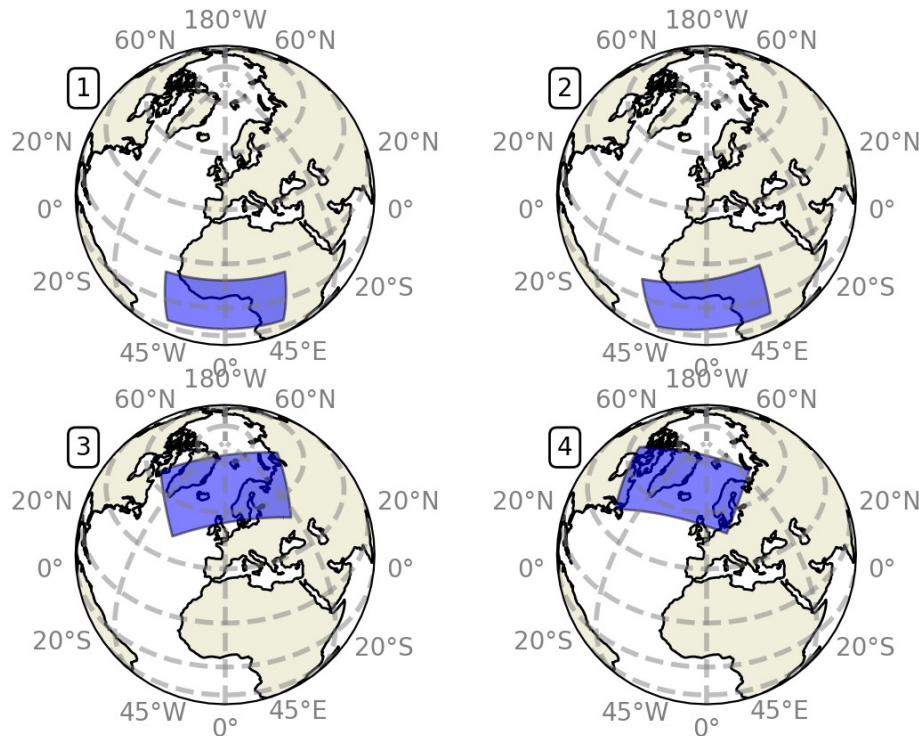
Etapes construction d'une grille:

1. **Placement de grille**
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

Informations complémentaires:

- Projection de Mercator à l'équateur
- Possibilité de rotation de grille

Etape de construction d'une grille CROCO





# Etapas de création de grille

- Pour les grilles parents

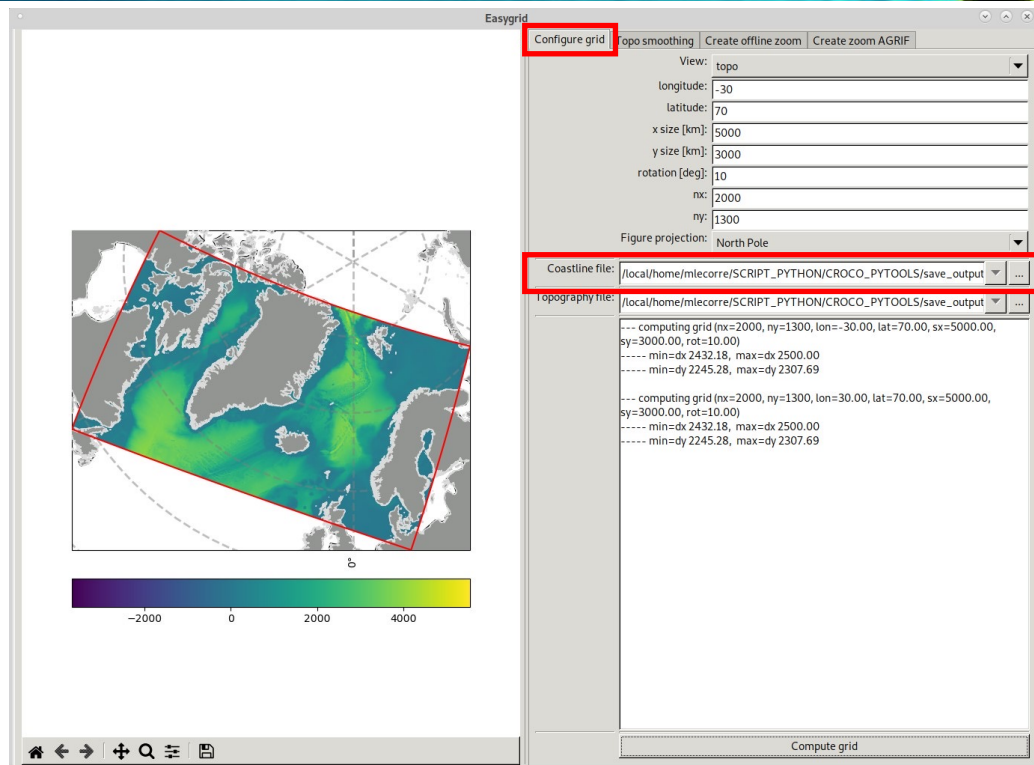
## Etapas construction d'une grille:

1. Placement de grille
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

Méthode utilisée: **GEOPANDAS**

Informations complémentaires:

- Lecture de Shapefile (.shp)
- Gestion WETDRY



# Etapes de création de grille

- Pour les grilles parents

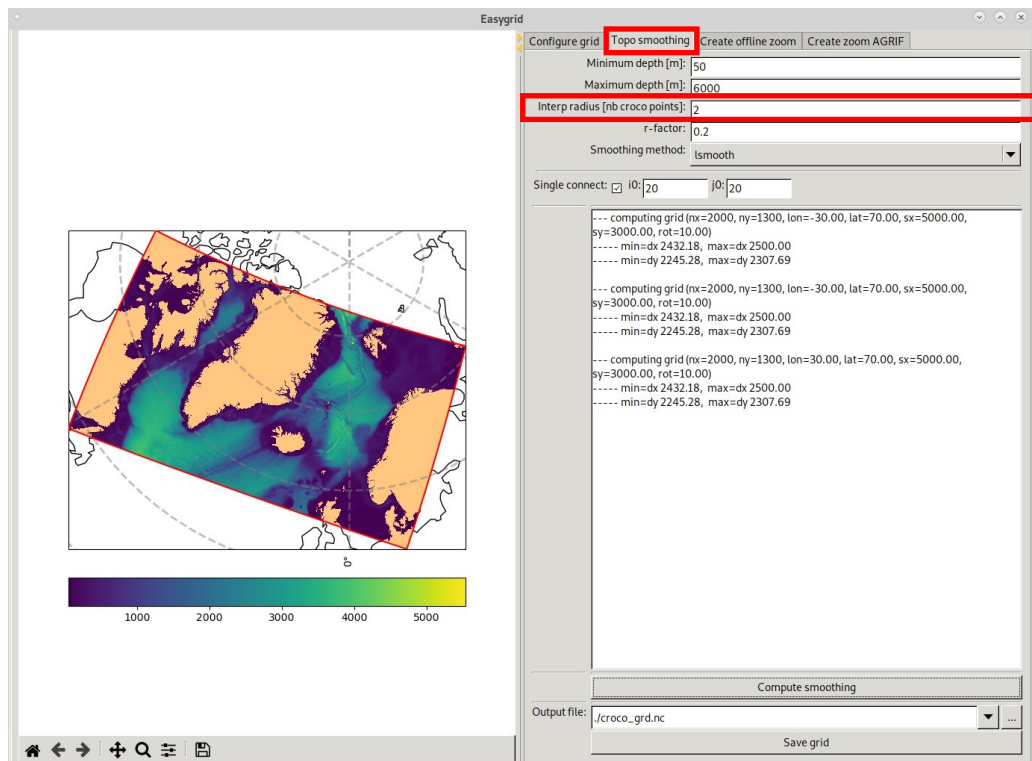
## Etapes construction d'une grille:

1. Placement de grille
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

Méthode utilisée: **Wrapper f2py (outils Sasha)**

Informations complémentaires:

➤ Fenêtre de Hann ou Interpolation bilinéaire





# Etapes de création de grille

- Pour les grilles parents

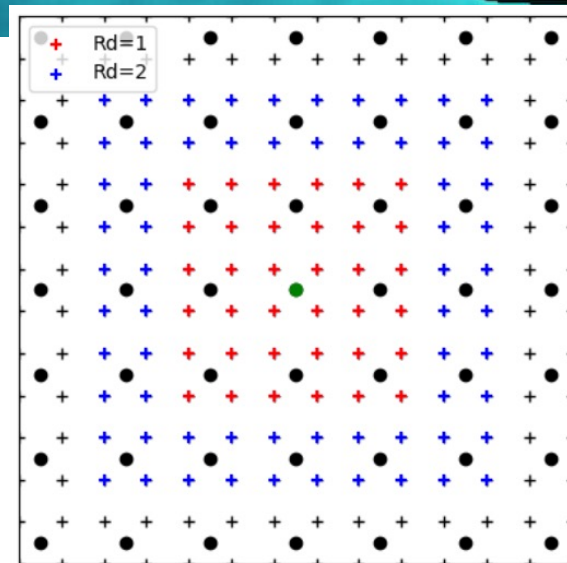
## Etapes construction d'une grille:

1. Placement de grille
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

Méthode utilisée: **Wrapper f2py (outils Sasha)**

Informations complémentaires:

➤ Fenêtre de Hann ou Interpolation bilinéaire



$$W(r) = 0.5 + 0.5 \cos\left(\pi * \frac{r}{width}\right), r < width$$

- $r$  est la distance un point CROCO et les points de la topo mère
- $width$  la largeur de la fenêtre en nombre de points de grille CROCO

# Etapes de création de grille

- Pour les grilles parents

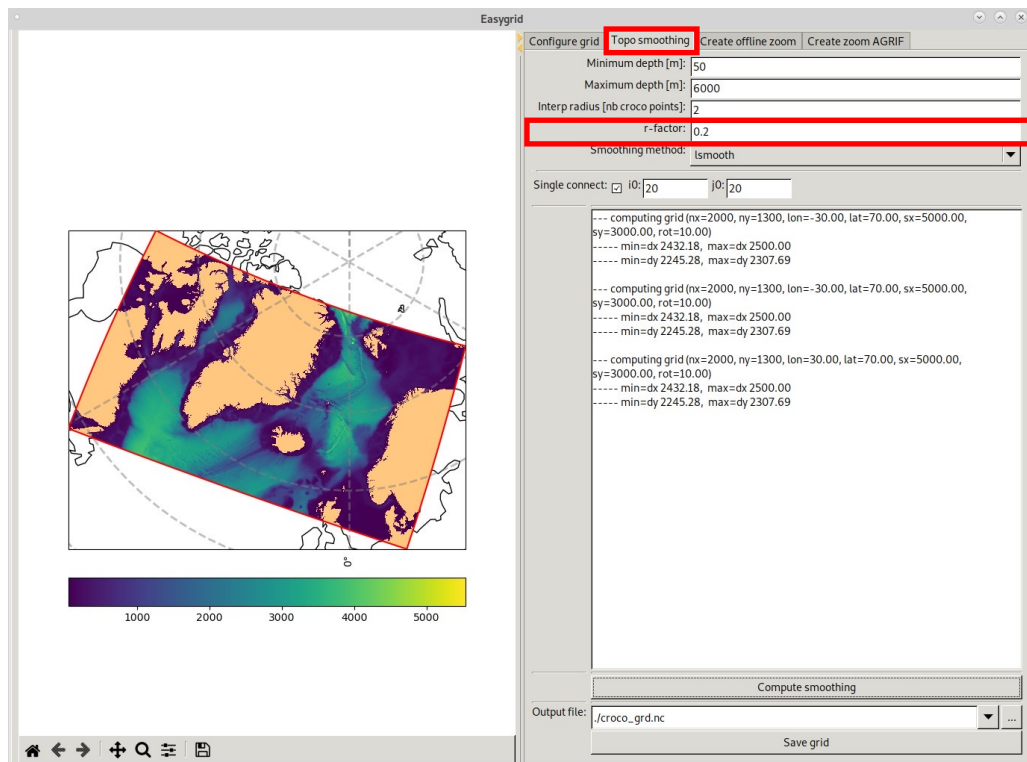
## Etapes construction d'une grille:

1. Placement de grille
2. Réalisation du masque
3. Dégradation/interpolation de la topographie
4. Lissage de la topographie

Méthode utilisée: **Wrapper f2py (outils Sasha)**

Informations complémentaires:

- Application Laplacien (nine point stencil)
- Plusieurs type de lissage (mask dép, log,...)



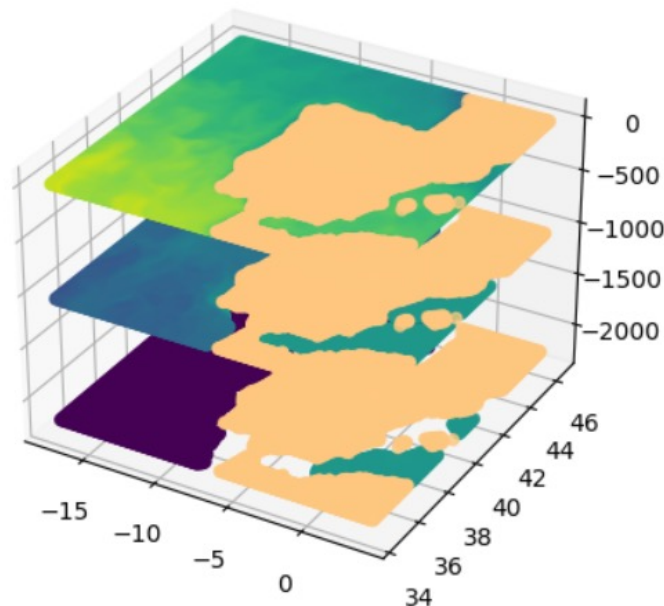
# Conditions limites et initiale

Objectif: Interpoler d'une grille d'entrée Z en grille CROCO en sigma

Valeur de température à différentes profondeurs Z

## Etapes construction IBC:

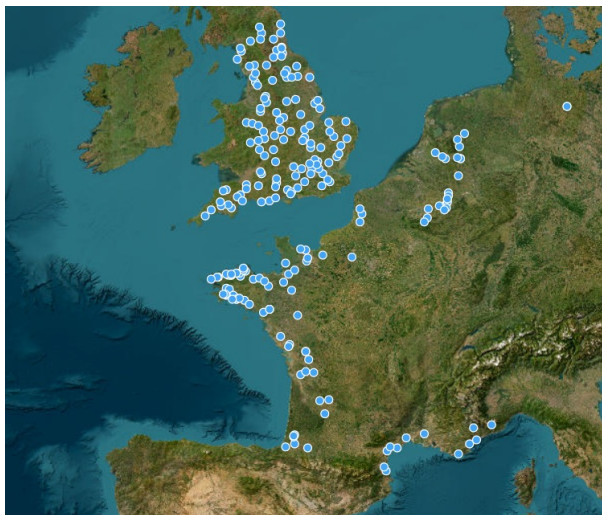
1. Extrapolation sur la terre
  - Plus proche voisin
2. Interpolation horizontale
  - Spline (lib pyinterp)
3. Interpolation verticale
  - Interpolation linéaire



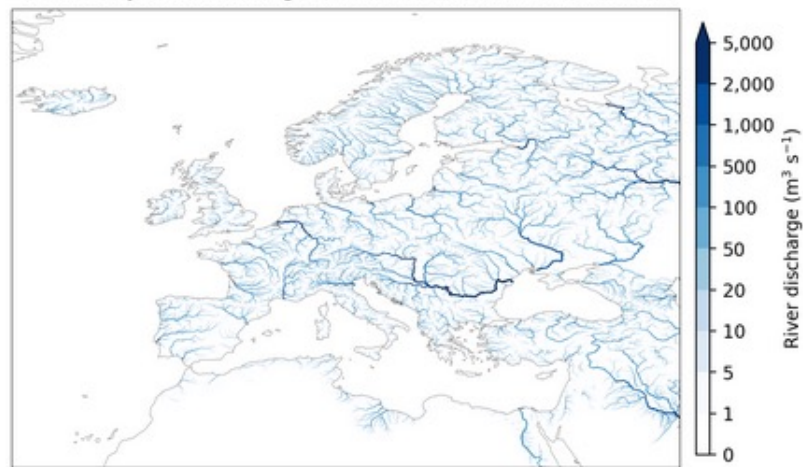
# Cas des rivières

## Problèmes rencontrés:

- Non uniformité des formats d'inputs
- Auto positionnement des rivières
- Gestion des rivières à haute résolution (positionnement plusieurs points?)



Mean daily river discharge from 1992 to 2022 for EFAS v5.0



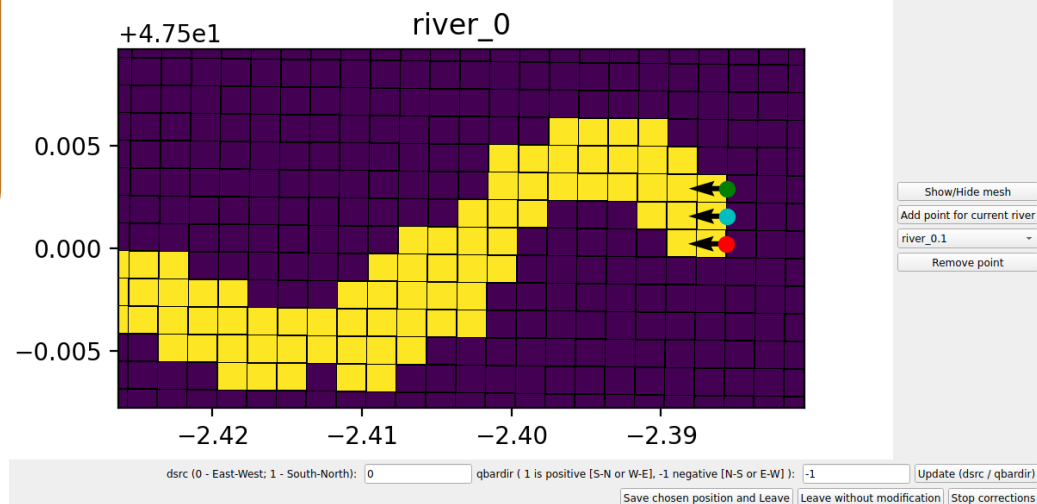
Observations ou réanalyses

# Cas des rivières

- Gestion listes de rivière ou données 2-D
- Possibilité d'utiliser une interface graphique (facilite le positionnement des rivières)
- Possibilité de rajouter plusieurs points pour une seule rivière
- Génération d'un .txt à recopier dans le croco.in

Line to enter in the croco.in file in the psource\_ncfile section :

```
-----
psource_ncfile:  Nsrc  Isrc  Jsrc  Dsrc  qbardir  Lsrc  Tsrc  runoff  file name
                  ./croco_runoff.nc
8
1284  325  0    -1    T T  14.0  5.0
1284  324  0    -1    T T  14.0  5.0
801   508  1    -1    T T  14.0  5.0
713   539  1    -1    T T  14.0  5.0
1501  174  0    -1    T T  14.0  5.0
1501  173  0    -1    T T  14.0  5.0
1501  172  0    -1    T T  14.0  5.0
996   414  0     1    T T  14.0  5.0
```





# Un mot sur les zooms

Ces outils permettent de gérer facilement la création de zooms (AGRIF ou offline)

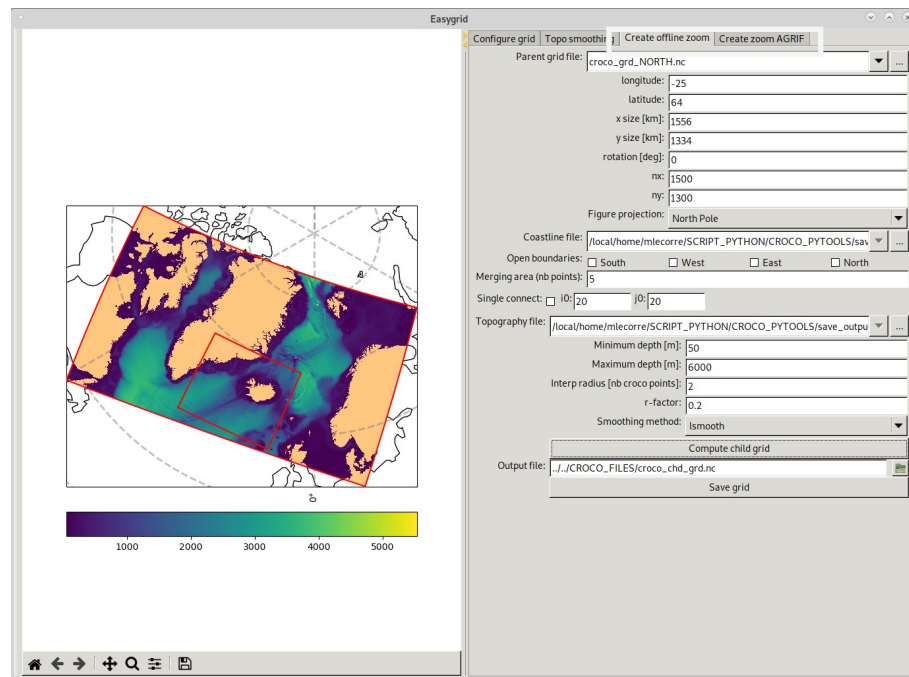
- Pour les grilles (deux onglets dans l'interface graphique):

- Match de la topographie aux frontières
- Match des masques aux frontières

- Pour les IBC (zooms offline):

=> *make\_zoom\_cond.py*

- Utilisation de scripts Fortran (wrapper f2py des r2r de Sasha )
- Peut gérer différents traceurs





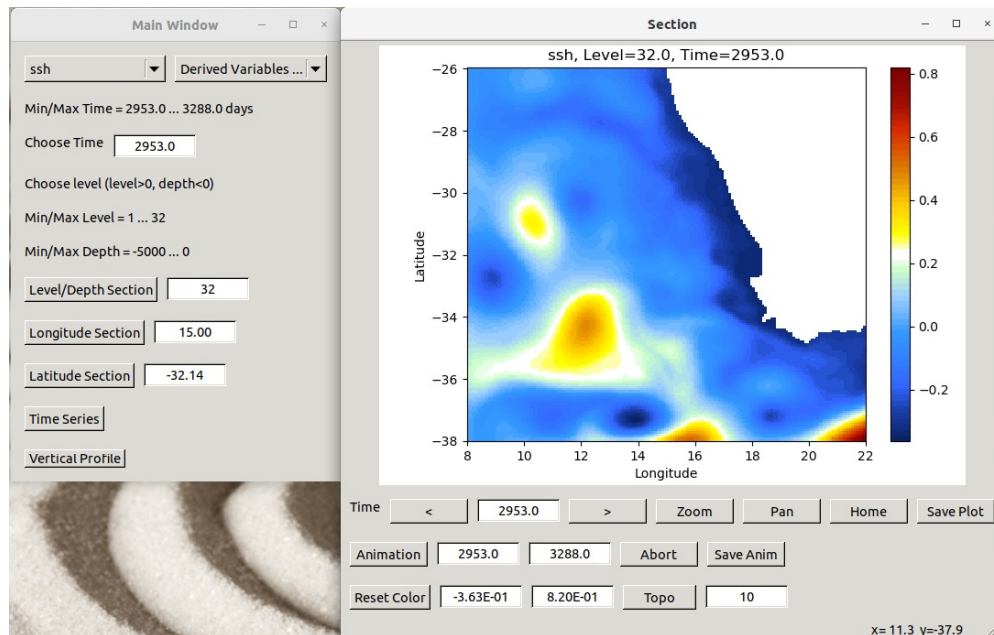
# Et pour le post-processing?

## Croco\_pyvisu:

Pour visualiser les sorties de croco

## Xcroco:

- Lit les sorties de CROCO
- Opérations sur les grilles (u2rho,...)
- Réalise divers diagnostics (PV, buoyancy ,...)
- Réalisation de coupes (transect, isoslice,...)



# Où les trouver?

Les outils et la documentation sont disponibles sur le site web



of the GHU (Globe)

HOME

DOWNLOAD

DOCUMENTATION

TRAININGS

USER MEETINGS

## Download stable release

Recommended download for non-developers : **last stable release**

- [Last CROCO stable release \(v2.0.0\)](#) : 22 April 2024, Major release.
- [Last CROCO\\_TOOLS stable release \(v2.0.0\)](#) : 22 April 2024, Major release.
- [CROCO\\_PYTOOLS stable release \(V1.0.1\)](#) : 03 June 2024, Bug-Fix release.

## CROCO\_PYTOOLS documentation

- [pre-processing tools \(prepro\)](#)
- [post-processing tools \(xcroco\)](#)
- [croco\\_pyvisu](#)

## Installation:

Deux environnements (un prépro, un post-pro) qui contiennent tout ce qui est nécessaire.

- Interface graphique
- Téléchargement données (CMEMS, CDS)
- Compilation outils fortran



Et pour la suite?

# Limites des outils actuels

- Scripts pré-pro/post-pro dissociés
- Redondance de certaines fonctions (coord sigma, changement grille,...)
- Sous-performances de certains scripts pour les grilles massives
- Utilisation d'une interface graphique (wxpython- peu optimisé pour l'utilisation distante et sur des calculateurs)

# Et pour la suite...

Une évolution des outils : → pycroco

- Format d'une librairie python
- Créer un pool de fonctions utiles au traitement de croco (prépro, postpro)
- Appelable en ligne de commande
- Ajout de fonctionnalités manquantes (forçages atmo, projections de grille, édition/merge de bathy...)

## Objectifs:

- Outils faciles à prendre en main
- Permettant la gestion de grilles massives
- S'adapter aux utilisations/besoins de chacun

# Que choisir...?

- Quel type d'objet manipuler dans nos scripts?
  - Une classe CROCO globale?
  - Une classe pour la grille, une pour les variables?
- Utilisation de notebook?
  - Remplacement de l'interface graphique (règle le problème de lenteur)
  - Facilite la création d'un éditeur de mask / bathy
  - Adapté pour des tutoriels, formations etc....



# Que choisir...?


- Comment rendre performante cette librairie?
  - Wrapper f2py
  - Xarray/dask
  - Numba
- Interpolations :
  - Veut-on laisser le choix de la méthode d'interpolation à l'utilisateur?
  - librairies : xgcm, xesmf...
  - online / pré-pro : dans online forçages atmo pour le moment ne gère que les grilles régulières (lon et lat définis en 1D)
  - Voudrait-on réaliser un online ini/bry à la manière des forçages atmo?

**Besoin de discussions sur ces choix à faire**

# Organisation

- Regroupement de dev indépendants
- Des canaux de discussion
  - Mattermost inria → pycroco
  - Réunion pydev mensuelle
- Un répertoire « bac à sable » dans le gitlab



croco-ocean / **pycroco\_sandbox**  Developer

## **Merci pour votre attention**

N'hésitez pas à nous rejoindre pour faire faire évoluer les outils!

RDV sur le mattermost dans le salon

→→→pycroco ←←←

# Title

- Text