

ENS DE LYON, UNIVERSITÉ LYON1

RAPPORT DE MASTER 2

---

**Mise en place d'un modèle numérique  
hydrodynamique de l'anse du Cul-de-Loup  
(Normandie, France) Opérateur expédition  
produits finis**

---

*Auteure* : Mme A. BONVALET

*Tuteur* : M. E. POIZOT

Janvier 2025 — Juin 2025

# Table des matières

<b>1</b>	<b>INTRODUCTION</b>	<b>2</b>
<b>2</b>	<b>ZONE D'ÉTUDE</b>	<b>3</b>
<b>3</b>	<b>MATÉRIELS ET MÉTHODES</b>	<b>4</b>
3.1	Le code de calcul CROCO . . . . .	4
3.1.1	Présentation générale . . . . .	4
3.1.2	Les variables modélisées . . . . .	4
3.1.3	Physique du modèle . . . . .	6
3.1.4	La grille CROCO . . . . .	6
3.1.5	L'enchaînement de plusieurs modélisations . . . . .	7
3.2	Les outils de pré-traitement . . . . .	7
3.2.1	La gestion de plusieurs grilles CROCO . . . . .	7
3.2.2	Les conditions initiales . . . . .	8
3.2.3	Les conditions aux bordures . . . . .	8
3.2.4	Les forçages . . . . .	9
3.3	Mesures de terrain - Le projet PROTEC . . . . .	9
3.4	Outils de Post-traitement . . . . .	9
<b>4</b>	<b>RÉSULTATS</b>	<b>10</b>
4.1	Le modèle "Anse du Cul-de-Loup" . . . . .	10
4.1.1	Données brutes utilisées . . . . .	10
4.1.2	Paramètres généraux choisis . . . . .	10
4.1.3	Augmentation de la résolution . . . . .	10
4.2	Résultats (sortie de modèle) . . . . .	12
<b>5</b>	<b>DISCUSSION</b>	<b>13</b>
<b>6</b>	<b>CONCLUSION</b>	<b>14</b>
<b>7</b>	<b>schémas numériques</b>	<b>15</b>
7.1	Shema WENO de cinquième ordre (Weighted Essentially Non-Oscillatory scheme) . . . . .	15
<b>8</b>	<b>MATÉRIELS ET MÉTHODES</b>	<b>15</b>
8.1	Le code de calcul CROCO . . . . .	15
8.1.1	Processus généraux . . . . .	16
8.1.2	Détail des processus du pré-traitement . . . . .	20
8.1.3	Détail des processus du modèle . . . . .	26
8.2	Le modèle "Anse du Cul-de-Loup" . . . . .	26
8.3	Mesures de terrain - Le projet PROTEC . . . . .	26

# 1 INTRODUCTION

A faire presque à la fin, mais les grandes idées seront :

- changement climatique, pression anthropique : besoin de gérer l'espace côtier
- modèles numériques, outils pour réaliser divers scénarios pour une bonne gestion
- anse du cul-de-loup (ADCL) : un bon exemple de nécessité de gestion raisonnée
- utilisation du code de calcul CROCO pour réaliser un modèle numériques de ADCL

## **2 ZONE D'ÉTUDE**

Pour moi, je t'en écrirai la majeure partie

## 3 MATÉRIELS ET MÉTHODES

### 3.1 Le code de calcul CROCO

#### 3.1.1 Présentation générale

Le modèle CROCO (*Coastal and Regional Ocean C*ommunity model) est une évolution du code ROMS\_AGRIF (*Regional Oceanic Modeling System with mesh refinement*). Le code ROMS\_AGRIF a été développé jusqu'en 2014 par l'IRD (*Institut de Recherche pour le Développement*) et l'INRIA (*Institut National de Recherche en Informatique et en Automatique*) qui ont participé au développement du code CROCO avec l'UPS (?), l'IFREMER (*Institut Français de Recherche pour l'Exploitation de la Mer*), le SHOM (*Service hydrographique et océanographique de la Marine*), le CNRS (*Centre National de la Recherche Scientifique*) et les participants de ROMS comme et principalement l'UCLA (*University of California, Los Angeles*) et la DGEO (*Departamento de Geofísica Universidad de Concepción*). Le code et la documentation de ROMS\_AGRIF ont progressivement été modifiés pour devenir le modèle communautaire CROCO.

Ce modèle permet désormais de réaliser des modélisations sur des grilles dont la résolution peut être kilométrique (modèle régional) à métrique (modèle côtier).

CROCO permet d'activer la parallélisation des calculs effectués lors de la modélisation. Ainsi, pour certaines configurations, il est possible de modéliser des temporalités de l'ordre de l'année pour un coût de calcul abordable. (ex .)

Le système d'équations utilisées par le modèle est flexible, il permet de choisir la méthode de résolution physico-chimique voulue et adaptée à la configuration. Les systèmes d'équations sont progressivement vérifiées par la communauté avec la mise en place de cas test dont la physique est simple.

Les forçages physico-chimiques initiaux et aux limites sont la source des mouvements dans la masse d'eau. Ils sont intégralement choisis par l'utilisateur - ice de CROCO. Ces forçages peuvent contenir une large variété de données spatiales et temporelles. On relèvera notamment la possibilité d'indiquer la valeur des températures, des salinités, des hauteurs d'eau relativement au niveau marins de référence, des vitesses de courant et des conditions atmosphériques à proximité de la surface. Les données des forçages sont rassemblées au sein de plusieurs fichiers. Ces dernières sont regroupées dans cinq principaux fichiers comme listé ci-dessous :

- le contexte physique du domaine modélisé (bathymétrie, limite côte-océan, etc.) ;
- les phénomènes physiques externes à la masse d'eau qui forcent son déplacement :
  - . la marée (amplitude et vitesse de déplacement de l'onde de marée),
  - . les conditions atmosphériques (vent, pluie, etc.),
- les conditions physico-chimiques internes à la masse d'eau aux limites durant la période modélisées (température, salinité, etc.),
- les mêmes conditions physico-chimiques dans l'ensemble de la zone modélisée au départ du modèle.

Ces fichiers doivent respecter une structure et une nomenclature particulière afin d'être utilisables par le code CROCO. Un schéma global de la mise en place du modèle est présenté en Figure 1

Enfin, en même temps que le développement du modèle, des outils communautaires de pré- et de post-traitement adaptés au fonctionnement de CROCO ont été développés. Ces outils ont d'abord été écrits en Matlab sous le nom de CROCO\_tools mais, pour mieux correspondre à la dynamique open-source de CROCO, de nouveaux outils, les CROCO\_Pytools, sont en cours de développement en Python.

#### 3.1.2 Les variables modélisées

Le modèle CROCO permet la propagation de variables physico-chimiques qui ont un effet rétroactifs sur leur propre propagation ou-et sur la propagation d'autres variables. Selon leur type, les paramètres physico-chimiques sont définis à différentes localisations relativement au maillage. La table 1 illustre les trois positions, appelées ici U, V et H, que peuvent avoir ces variables.

Une importante part des variables sont localisées en **position centrale H** au maillage. Ces variables sont :

- la hauteur de la colonne d'eau relativement au niveau marin moyen (à vérifier),
- la température de l'eau,
- la salinité de l'eau,
- l'ensemble des paramètres décrivant les ondes de marée (phase et amplitude de la hauteur d'eau ainsi que celles des deux composantes des courants induits),
- l'ensemble des paramètres atmosphériques à l'exception du vent (précipitation, température de l'air, vitesse absolue et direction du vent à 10m au dessus de la surface de l'eau, etc.).

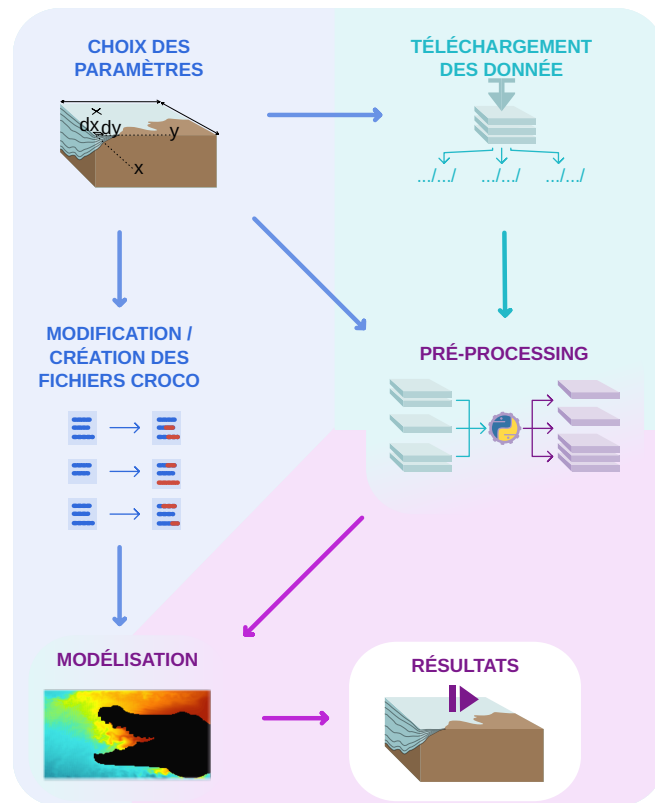


FIGURE 1 – Mise en place du modèle CROCO avec une grille simple.

Les flèches larges correspondent à l'utilisation d'un ou plusieurs fichiers pour la génération d'un ou de plusieurs autres. L'action nécessaire est décrite en bout de flèche. Lorsque les flèches ne se suivent pas, les étapes peuvent être effectuées en parallèle. Toutefois la mise en place de la modélisation est plus intuitive en suivant l'ordre : **choix des paramètres**, **téléchargement des données**, **pré-traitement**, **modification des fichiers CROCO**, **modélisation** (compilation et exécution de CROCO).

Les paramètres localisés à la **position U** du maillage sont :

- la composante Est-Ouest de la vitesse de déplacement de l'eau,
- la contrainte à la surface de l'eau causée par la composante Est-Ouest du vent,
- la composante Est-Ouest de la vitesse de déplacement de l'air à 10m.

Enfin, les paramètres localisés à la **position V** du maillage sont :

- la composante Nord-Sud de la vitesse de déplacement de l'eau,
- la contrainte à la surface de l'eau causée par la composante Nord-Sud du vent,
- la composante Nord-Sud de la vitesse de déplacement de l'air à 10m.

Pour tous les paramètres, ils ne sont pas exprimés et pas pris en compte aux points indiqués comme émergés par la grille. Certains de ces paramètres sont exprimés sur l'intégralité du maillage horizontal et vertical. D'autres ne le sont qu'aux limites de la grille, c'est à dire sur tout le tour de cette dernière. D'autres encore sont exprimés sur toute l'étendue horizontale de la grille mais uniquement en surface ou au fond.

L'ensemble de ces paramètres définissent les forçages physico-chimiques imposés au modèle. Ils sont regroupés dans plusieurs fichiers selon la manière dont ils influent sur le modèle.

La première catégorie contient les forçages qui définissent les conditions initiales. Le fichier correspondant contient donc les données de hauteur d'eau, de courant, de température et de salinité sur l'ensemble de la grille à l'initialisation du modèle.

La seconde catégorie contient les forçages qui définissent les conditions aux bords latéraux de la grille. Le fichier correspondant contient donc les données de hauteur d'eau, de courant, de température et de salinité aux bords de la grille durant le temps de la modélisation.

La troisième catégorie contient les forçages de marée. Le fichier correspondant contient donc, pour  $n$  harmoniques de marée, les amplitudes et les phases des hauteurs d'eau et éventuellement les deux composantes horizontales de la vitesse de déplacement de l'onde de marée.

La quatrième catégorie contient les forçages atmosphériques. Ils concernent la surface de l'eau et l'air qui la surplombe pour chaque maille de la grille CROCO. Le fichier correspondant contient donc notamment l'humidité,

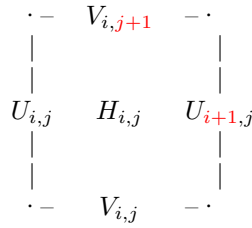


TABLE 1 – Représentation d’une maille  $i, j$  horizontale de la grille CROCO. Les points sont des nœuds et les tirets sont les cotés de la maille. La position à laquelle est calculé chaque type paramètre de la maille est représentée par  $U_{i,j}$ ,  $V_{i,j}$  et  $H_{i,j}$  leur première lettre correspondent au type des paramètres qu’elles représentent. L’existence des paramètres  $V_{i,j+1}$  et  $U_{i+1,j}$  suppose que la maille  $i, j$  ne se situe pas à l’extrémité d’une ligne ou d’une colonne de la grille CROCO.

la température, la pression de l’air au dessus de la surface de l’eau, la vitesse du vent à 10m au dessus de la surface, des paramètres sur les contraintes de surfaces imposés par le vent et les radiations solaires reçues par la masse d’eau pour les courtes et longues longueurs d’ondes.

### 3.1.3 Physique du modèle

Lors de sa compilation, CROCO utilise le fichier *cppdef.h*, l’un des quatre fichiers dont le contenu est modifié par l’utilisateur `·ice`, pour déterminer quelles équations physiques doivent être résolues par le modèle. Ces fichiers sont représentés en bleu en Figure 1.

Pour les configurations, le contenu du fichier a été modifié pour que les équations résolues correspondent à un schéma d’advection de cinquième ordre 7.1.

#### Les schémas d’advection et de diffusion

Les équations utilisées et résolues par CROCO durant la modélisation sont flexibles. Ainsi, l’utilisateur `·ice` choisit quels schémas numériques sont utilisés pour la résolution de l’advection d’une part et de la diffusion d’autre part. Les schémas numériques possibles pour l’advection sont au minimum de second ordre et au maximum de cinquième ordre. Les schémas numérique de diffusion sont laplaciens ou bilaplaciens. (ça se dit probablement autrement !) Le choix des schémas est libre pour l’utilisateur `·ice`, (JULLIEN et al. s. d.(b)) il est toutefois important de noter que les schémas choisis doivent être cohérents entre eux. D’autre part, certains schémas sont mieux adaptés que d’autre à certaines situations modélisées, il est donc important de bien comprendre sa configuration avant de choisir les schémas numériques que le modèle doit utiliser.

Dans notre configuration, le nombre de couches verticales étant faibles ( $\leq 30$ ) et la topographie étant assez découpée par endroits, le schéma d’advection choisi est celui quasi-monotone de cinquième ordre de WENO (Weighted Essentially Non-Oscillatory scheme). Ses particularités sont décrites dans la documentation de croco et en Annexes de ce rapport 7.1.

D’autre part, le choix des schémas numériques de diffusion verticale est principalement dicté par la structure de la grille du modèle. Dans notre configuration, les couches horizontales de la grille sont d’épaisseur variables et suivent la topographie. Le schéma de diffusion verticale choisi est Laplacien (ça se dit probablement autrement !) et est résolu dans la direction locale de chaque couche de la grille avec une viscosité turbulente paramétrisée avec/selon Smagorinsky. Enfin, le schéma de diffusion horizontale est de type GLS (Generic Length Scale) en K-OMEGA provenant de KOLMOGOROV 1941.

Le détail des clefs activées pour correspondre à ce modèle physique se trouve en Annexes. (à faire) Ces clefs ont été principalement déterminées sur la base de la documentation principale (JULLIEN et al. s. d.(a)). Elles ont de plus été adaptées pour correspondre au modèle de plage de MARCHESIELLO et al. 2021 dont le modèle n’est pas à la même échelle que ceux réalisés dans le cadre de ce stage mais il fait aussi face à des dynamiques de découvrement régulier de mailles de la grille dans la zone d’action des vagues. En effet, dans notre situation, le découvrement s’effectue pour les mailles de la zone de balancement des marées et nécessite ainsi la même robustesse que le modèle précédemment cité.

### 3.1.4 La grille CROCO

Le maillage horizontal suivi par la modélisation est déterminé par les données de la grille CROCO. Cette grille est localisée en longitude et latitude à la surface de la planète. Elle définit par ailleurs l’étendue horizontale

et la résolution du maillage. La grille CROCO contient aussi une donnée de bathymétrie / topographie pour chacune de ses mailles. Enfin, elle contient un masque définissant quelles mailles sont fixées comme toujours émergées. Le modèle ne sera ainsi jamais exécuté au niveau de ces mailles.

Les résolutions verticales du maillage complet utilisé par le modèle sont déterminées par le fichier de conditions initiales. Selon les paramètres choisis, la résolution verticale peut ainsi varier verticalement selon deux paramètres  $\sigma_s$  et  $\sigma_b$ . Ces derniers décrivent le taux d'augmentation ou de diminution de la résolution verticale à l'approche de la surface et du fond. Leurs effets peuvent être visualisés en Figure ??

### 3.1.5 L'enchaînement de plusieurs modélisations

Le modèle CROCO permet de mettre en place l'étude d'un lieu en plusieurs étapes de modélisation.

L'usage principal de cette fonctionnalité consiste à emboîter plusieurs grilles de résolution croissante et de taille décroissante autour d'une zone d'intérêt. Toutes les grilles contenues dans la grille principale extraient leur conditions aux bords de la grille qui les précède. Cette méthode d'emboîtement de grilles permet de s'affranchir du besoin de données pour les conditions aux bords de la grille d'intérêt tout en maintenant un coût de modélisation assez léger. Les avantages procurés par cet usage correspondent bien au contexte de l'étude où la zone d'étude est fortement restreinte dans l'espace et où les données doivent y être finement modélisées. Il existe de type de mise en place de système en grilles emboîtées dans CROCO :

- le premier type d'emboîtement des grilles (Offline zoom) est linéaire, c'est à dire qu'une grille fille est toujours générée après l'exécution de la modélisation CROCO sur sa grille mère. les données de sortie du précédent modèle sont utilisées pour générer les conditions initiales et aux limites de la grille fille. Cette méthode est illustrée en Figure XXX,
- le second type d'emboîtement (AGRIF) est simultané et rétrospectif. Ce qui signifie que toutes les grilles sont générées avec les données du pré-traitement avant la modélisation. Ensuite, pendant la modélisation, CROCO utilise alternativement les données de la grille mère pour les conditions aux bords de la grille fille puis les valeurs au bords de la grille fille pour contraindre les valeurs au sein de la grille mère.

Un second usage consiste à mettre en continuité plusieurs grilles de même résolution afin de suivre le trait de côte par exemple. Ainsi, moins de données de conditions aux limites sont nécessaires par rapport à une modélisation de multiples grilles de manière indépendante. Cet usage ne correspond pas aux besoins de cette étude qui a pour but de modéliser une zone aisément assimilable à un rectangle, c'est à dire que cette zone peut être efficacement représentée par une unique grille.

## 3.2 Les outils de pré-traitement

Dans le cadre du stage, ce sont les outils Python qui sont utilisés pour le pré-traitement. Ces outils sont identifiés comme les "CROCO Pytools" et sont les plus récents intégrés dans l'univers CROCO.

Les CROCO Pytools sont encore en cours de développement, les utiliser permet de participer à la vérification de leur fonctionnement pour la communauté. Certaines fonctionnalités sont encore manquantes dans les CROCO Pytools. Il est donc nécessaire de les ajouter soit en modifiant les scripts déjà présents, soit d'écrire des scripts supplémentaires pour réaliser certaines tâches spécifiques.

La création des fichiers nécessaires au démarrage de CROCO, suit l'ordre suivant :

- création des grilles de calcul décrivant la morphologie du domaine modélisé ;
- création des conditions de forçages pour la mise en mouvement de la masse d'eau ;
- création des conditions aux bordures du domaine modélisé ;
- création des conditions initiales à l'intérieur du domaine modélisé.

La Figure 2 présente le fonctionnement des CROCO Pytools et des scripts écrits dans le cadre du stage pour le pré-traitement, c'est à dire pour générer les fichiers nécessaires au modèle CROCO à partir des données brutes. Les chemins de fichiers présentés dans le rapport correspondent à la nomenclature définie en Annexes.

### 3.2.1 La gestion de plusieurs grilles CROCO

D'autre part, les CROCO Pytools permettent la préparation de deux types de grilles emboîtées (Offline Zoom et AGRIF)

La méthode d'emboîtement des grilles AGRIF était préférée. En effet, elle simplifie la modélisation du côté de l'utilisateur : il n'a besoin de mettre en place et de démarrer le modèle qu'une unique fois pour obtenir les résultats de haute résolution des grilles filles. D'autre part, elle a plus de chances de mieux représenter la réalité grâce à son fonctionnement en aller retour entre les grilles.



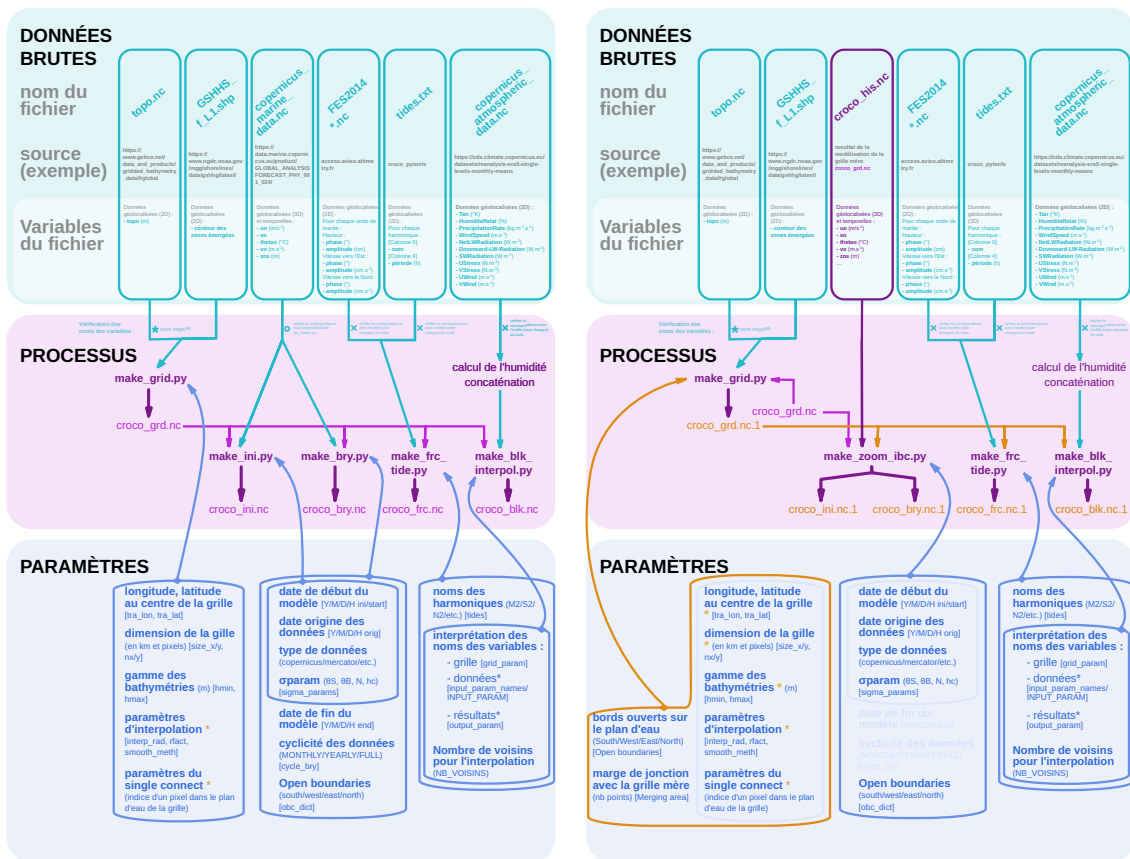


FIGURE 2 – Pré-traitement avec les `croco_pytools` pour une grille simple (mère) à gauche et pour une sous-grille imbriquée (fille) à droite.

Les paramètres marqués d'une astérisque (\*) bleue sont ceux qui doivent être modifiés en fonction du code pour lesquels ils sont utilisés (`make_frc_tide.py` ou `make_blk_interpol.py`). Les paramètres marqués d'une astérisque (\*) orange sont des paramètres qui ont la même signification pour la grille mère et pour la grille fille mais leur valeur peut être différente pour chaque grille. Dans les données brutes de la grille fille, `croco_his.nc` provient de l'exécution du modèle CROCO sur la grille mère `croco_grd.nc`. La signification des paramètres cités dans la figure est détaillée en Annexes.

Toutefois, cette méthode a été abandonnée durant le stage car il est apparu, pendant les phases de test, que (à lister... trop de dépendances et de contraintes sur les données utilisées). Ainsi, c'est la méthode Offline zoom qui est utilisée pour la suite du rapport.

Chaque grille CROCO, pour être utilisée par le modèle, doit être liée à d'autres fichiers de données qui permettent l'initialisation et la mise en mouvement de la masse d'eau. Ces fichiers sont décrits dans les paragraphes suivants.

### 3.2.2 Les conditions initiales

Le fichier "`croco_ini.nc`" contient notamment des données comme la température, la salinité, et la hauteur de la surface de l'eau. Ces données couvrent chaque maille de la grille CROCO et correspondent aux paramètres physiques de la zone au début de la modélisation.

### 3.2.3 Les conditions aux bordures

Le fichier "`croco_bry.nc`" contient notamment les données comme la température, la salinité et éventuellement la hauteur d'eau et les vitesses des courants. Ces données couvrent seulement les bords extérieurs de la grille CROCO sur toute la durée temporelle de la modélisation.

### 3.2.4 Les forçages

L'ensemble de des fichiers de forçage ont été générés à partir de données brutes et couvrent exactement les mailles de la grille CROCO. Les outils de pré-traitement originaux n'ont été utilisés que pour la génération des conditions initiales et au bord. Les autres outils pour les forçages n'étaient pas existants ou pas fonctionnels avec nos données.

Ainsi, deux scripts Python principaux ont été écrits dans le cadre du stage. Le premier script permet de générer un fichier de marée pour CROCO à partir des données des modèles TPXO10 ou FES2014. Dans ce script, les amplitudes et phases des vitesses des ondes de marée sont transformées en ellipse de périodicité selon les formules utilisée dans le `croco_pytool` de référence,

$$SemiMajAxis = |wp| + |wm|$$

$$Eccentricity = \frac{|wp| + |wm|}{|wp| - |wm|}$$

$$Inclinaison = \frac{arg(wm) + arg(wp)}{2\pi} . 180$$

$$PhaseAngle = \frac{arg(wm) - arg(wp)}{2\pi} . 180$$

avec

$$wp = \frac{u * + j.v *}{2} \quad \text{et} \quad \overline{wm} = \frac{u * - j.v *}{2}$$

avec

$$u * = \exp\left(\frac{\pi PhaseU}{180} . (-1j)\right) . AmplU \quad \text{et} \quad v * = \exp\left(\frac{\pi PhaseV}{180} . (-1j)\right) . AmplV$$

avec  $PhaseU$ ,  $PhaseV$ ,  $AmplU$ ,  $AmplV$  les phases et amplitudes des composantes respectivement Sud-Ouest et Nord-Sud de la vitesse d'une onde de marée.

Le second génère un fichier de conditions atmosphériques en surface de l'eau à partir des données issues de Copernicus notamment. Au préalable, les valeurs d'humidité relative  $RH$  sont calculées et stockées dans une nouvelle variable selon la méthode décrite par VAISALA 2013 en considérant une température entre  $-20$  et  $+50^{\circ}C$  :

$$RH = 10^{7,591386 \cdot \left( \frac{T_{dewpt}}{T_{dewpt} + 240,7263} - \frac{T}{T + 240,7263} \right)}$$

avec  $T_{dewpt}$  la température du point de rosée de l'air à  $2m$  au dessus de la surface de l'eau et  $T$  la température de l'air à la même altitude.

## 3.3 Mesures de terrain - Le projet PROTEC

Pour moi : quelques mots sur les données disponibles pour comparer les résultats modèle avec des mesures de terrain (cf Discussion)

## 3.4 Outils de Post-traitement

à voir, détailler :

- l'écriture de scripts pour extraire le point le plus proches,
- les choix pour les interpolations (closest neighbor),
- les REQM utilisés (RMSEDEI).

## 4 RÉSULTATS

### 4.1 Le modèle "Anse du Cul-de-Loup"

La lecture de la documentation, des codes des CROCO Pytools et de fichiers d'exemple de sortie du pré-traitement Matlab ont permis de réaliser des choix pour le pré-traitement. Ce dernier a abouti à la préparation correcte des fichiers pour l'exécution du code CROCO. Les choix qui ont été faits sont détaillés dans les parties suivantes.

#### 4.1.1 Données brutes utilisées

Plusieurs types de fichiers de données brutes sont recherchées. L'ensemble de ces fichiers est représenté en Figure 2. La majorité des données proviennent des bases de données de Copernicus (marine ou climate). Les données de marée du modèle TPXO10 ont été testées, mais ce sont les résultats du modèle FES2014 qui ont finalement été retenus. (indiquer pourquoi ? meilleure résolution dans notre zone ?)

Pour la majorité de la surface couverte par la grille mère, les données de bathymétrie proviennent du GEBCO 2024. Ces données sont continues dans l'espace. Par ailleurs, la campagne PROTEC offre des données de haute résolution mais discontinues sur la zone de l'Anse du Cul-de-Loup. Les données provenant de GEBCO et PROTEC sont donc fusionnées entre elles afin d'obtenir une cartographie bathymétrique de haute résolution dans les zones où les données le permettent.

Enfin, les fichiers vectoriels géo-référencés qui définissent le trait de côte proviennent de la National Oceanic and Atmospheric Administration (NOAA). Ces données vectorielles sont corrigées par les données de la campagne PROTEC qui sont plus précises mais qui couvrent seulement le Nord du Cotentin.

Une fois ces données acquises, il convient de générer la grille du modèle ainsi que ses fichiers d'initialisation et de contraintes aux limites. Les paramètres de cette grille sont utilisés par les codes de pré-traitement pour transformer les données brutes en données utilisables par CROCO.

#### 4.1.2 Paramètres généraux choisis

Paramètres de la grille (étendue, localisation, résolution horizontale et verticale, etc.), choix du LES.

Les paramètres généraux sont majoritairement déterminés par l'utilisateur lors de l'exécution du CROCO Pytool de création de la grille. Cette grille est fondamentale au fonctionnement de CROCO, elle détermine en quels points sont résolues les équations du modèle. Ses principales propriétés sont :

- sur chaque plan horizontal, la grille couvre la même zone rectangulaire sur Terre, à l'exception des zones émergées,
- les espacements en latitude et longitude des mailles sont réguliers.

D'autre part, les propriétés verticales ( $\sigma_{param}$ ) de la modélisation sont d'une grande importance, notamment pour la qualité de la résolution des turbulences et des mélanges avec la surface du plan d'eau. Ces propriétés ne sont pas directement déterminées dans le fichier contenant la grille CROCO. Toutefois, les  $\sigma_{param}$  doivent être fixés par l'utilisateur lors de la génération des conditions initiales et aux limites, comme représenté dans la Figure 2.

- le nombre de points alignés verticalement est constant quelque soit la bathymétrie dans le plan d'eau,
- les espacements verticaux entre les mailles de la grille sont variables. Ils respectent les paramètres  $\theta_s$ ,  $\theta_b$ ,  $hc$  et  $n$  qui imposent des répartitions verticales respectant la bathymétrie en chaque point comme illustré par la Figure 3.

Les paramètres entrés par l'utilisateur fixent la localisation en latitude et longitude de la grille, sa résolution dans les trois dimensions de l'espace ainsi que le choix de la restriction de la modélisation à un seul plan d'eau. De plus, la grille est générée en utilisant les données brutes de bathymétrie et de trait de côte, comme décrites dans la partie précédente.

La localisation et l'étendue spatiale de la grille sont choisies afin que ses faces ouvertes sur le plan d'eau soient le moins possible recoupées par des zones émergées.

#### 4.1.3 Augmentation de la résolution

Nombre et paramètres des grilles imbriquées

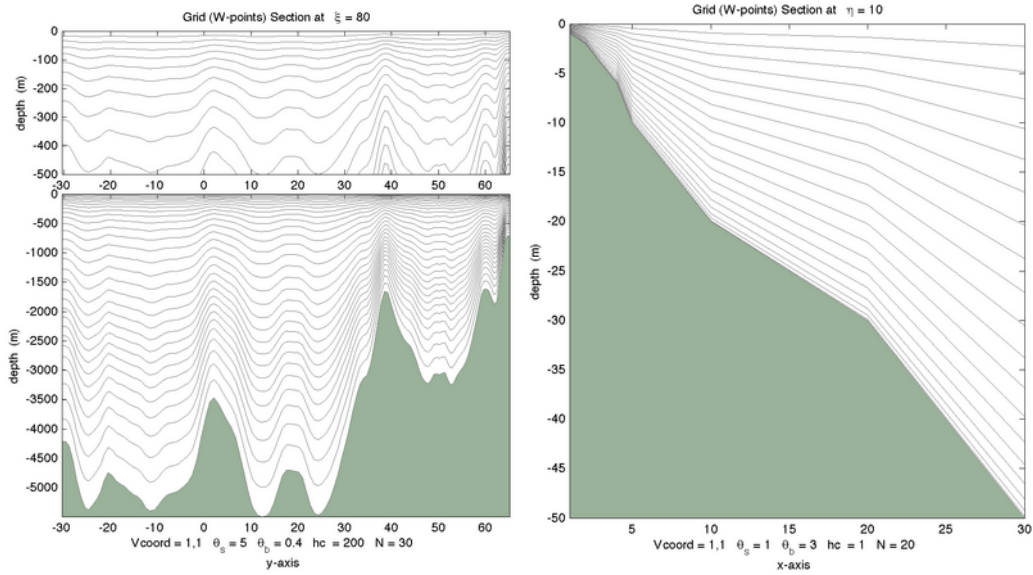


FIGURE 3 – Exemples de résolutions verticales variables déterminées par la topographie et les paramètres  $\sigma_{param}$  entrés par l'utilisateur · ice.

À gauche, les paramètres choisis sont  $\theta_s = 7$ ,  $\theta_b = 0.1$ , à droite ils sont  $\theta_s = 7$ ,  $\theta_b = 3$ .

La zone d'étude a un rayon d'une dizaine de kilomètre et la résolution des données attendues est d'au moins quelques dizaines de mètre. Toutefois, les données brutes (les modèles de marées et les données générales marines et atmosphérique) qui sont utilisées pour fixer les conditions initiales et aux limites pour un modèle dans cette zone ont une résolution largement inférieure à celle attendue en sortie de modèle. Il est donc intéressant de mettre en place un ensemble de modélisations en grilles emboîtées afin de décrire correctement les courants autour du Nord du Cotentin. L'imbrication successive des grilles permet d'aboutir à la modélisation plus réaliste et stable numériquement d'une zone restreinte à l'Anse du Cul-de-Loup. Trois grilles emboîtées ont ainsi été réalisées ici, elles sont illustrées en Figure 4.

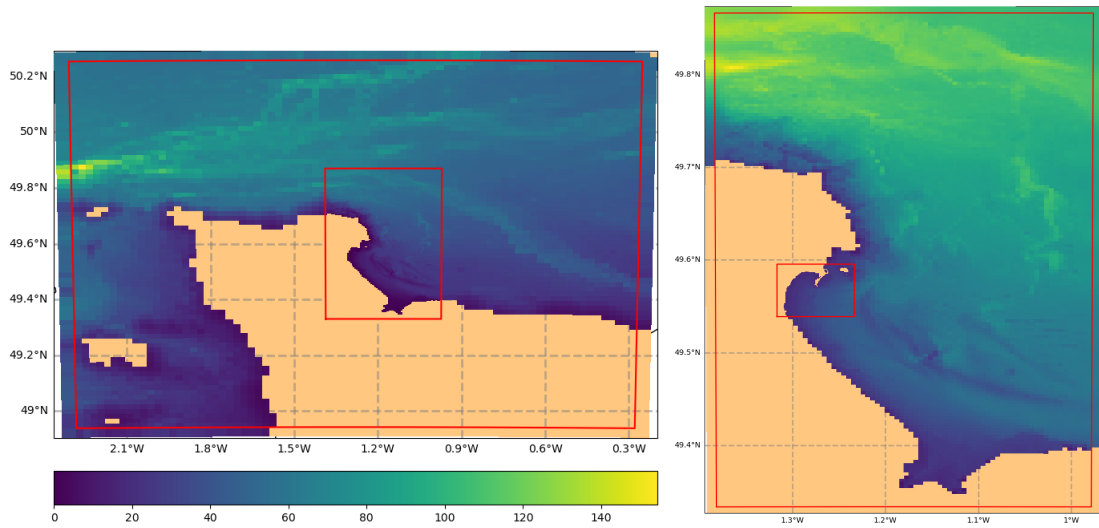


FIGURE 4 – Visualisation des grilles emboîtées choisies pour la modélisation de l'Anse du Cul-de-Loup. À gauche, la grille mère est délimitée par le grand rectangle rouge, sa grille fille est délimitée par le petit rectangle rouge. À droite, la même grille fille correspond au grand rectangle, elle est mère de la grille petite-fille délimitée par le petit rectangle rouge. Les pixels oranges sont les mailles considérées comme obligatoirement émergées. Ainsi, les équations du modèle n'y seront pas résolues. Les autres couleurs (du bleu au jaune) représentent la bathymétrie dont les échelles respectives sont représentées en dessous de chaque graphique.

La position des grilles emboîtées est déterminée selon deux règles :

- les bords de la grille fille doivent être assez loin des bords ouverts sur le plan d'eau de la grille mère, leur centre est donc généralement proche,
- au niveau de l'intersection entre les bords d'une grille et le tracé de la côte, ce dernier doit être le plus

simple possible, c'est à dire de préférence rectiligne en Nord-Sud ou Est-Ouest. De plus, pour les grilles filles, le tracé de la côte au niveau des bords doit être aligné avec le tracé de la côte de la grille mère.

## 4.2 Résultats (sortie de modèle)

Sorties selon les choix de modélisation :

- mode hydro (RANS, LES, etc.) (à voir)
- effet de l'atmosphère (bulk / climato)
- couplage avec les vagues (WaveWatch-III) (pas encore fait)

Visualisation : snapshots représentatifs de la marée, zoom sur des zones dynamiques

Comparaisons : différences, variations de l'écart type

## 5 DISCUSSION

Comparaison des résultats du modèle avec les mesures de terrain : graphiques, indicateurs de qualité (RMSE, etc.)

## 6 CONCLUSION

Ca sera fait à la fin ou presque

## BIBLIOGRAPHIE

### Références

- [Jul+a] S. JULLIEN et al. *Model Documentation, Appendices, cppdefs.h*. URL : [https://croco-ocean.gitlabpages.inria.fr/croco\\_doc/model/model.appendices.cppdefs.h.html](https://croco-ocean.gitlabpages.inria.fr/croco_doc/model/model.appendices.cppdefs.h.html).
- [Jul+b] S. JULLIEN et al. *Model Documentation, Numerics, Advection Schemes*. URL : [https://croco-ocean.gitlabpages.inria.fr/croco\\_doc/model/model.numerics.advec.html](https://croco-ocean.gitlabpages.inria.fr/croco_doc/model/model.numerics.advec.html).
- [Kol41] Andrej Nikolaevich KOLMOGOROV. “Equations of turbulent motion in an incompressible fluid”. In : *Dokl. Akad. Nauk SSSR*. T. 30. 1941, p. 299-303.
- [Mar+21] Patrick MARCHESIELLO et al. “Tridimensional nonhydrostatic transient rip currents in a wave-resolving model”. In : *Ocean Modelling* 163 (2021), p. 101816. ISSN : 1463-5003. DOI : <https://doi.org/10.1016/j.ocemod.2021.101816>. URL : <https://www.sciencedirect.com/science/article/pii/S1463500321000676>.
- [Vai13] VAISALA. “Humidity conservation formulas, Calculation formulas for humidity”. In : (2013). URL : <https://www.hatchability.com/Vaisala.pdf>.

# ANNEXES

## 7 schémas numériques

### 7.1 Shema WENO de cinquième ordre (Weighted Essentially Non-Oscillatory scheme)

Les variables sont disposées en C-grille d'Arakawa. Les valeurs des traceurs sont notées  $q_i$  et les vitesses  $u_i$ . Dans le cadre de ce schémas d'advection, trois valeurs aux interfaces sont évaluées en se utilisant trois stencils différents. Une moyenne non linéaire entre ces trois valeurs est utilisée comme suit pour déterminer la valeur calculée :

$$\tilde{q}_{k-\frac{1}{2}} = a_1 k - \frac{1}{2}^{(1)} + a_2 k - \frac{1}{2}^{(2)} + a_2 k - \frac{1}{2}^{(2)}$$

Les constantes de pondération sont définies selon les règles suivantes :

1.  $\sum_{j=0}^2 a_j = 1$ ,
2. propriété ENO (Essentially Non Oscillatory),
3. cinquième ordre à condition que  $q(x)$  soit peut rugueux.

C'est un schéma à variation bornée (TBV, Total variation Bounded). source : JULLIEN et al. s. d.(b)

## 8 MATÉRIELS ET MÉTHODES

### 8.1 Le code de calcul CROCO

mettre les principes généraux de CROCO (origine, type de modèle, etc...)

+ organisation des fichiers

/ ..... le chemin peut être différent pour PATH-TO-CROCO/ et PATH-TO-CONFIG/





```

|
|_ gshhs/
|   |_ GSHHS_f_L1.shp
|   |_ Ini/
|       |_ copernicus_marine_data.nc
|       |_ Topo/
|           |_ topo.nc
|_ preproOUTPUT/
|_ cppdefs.h
|_ param.h
|_ jobcomp
|_ croco.in

```

### 8.1.1 Processus généraux

#### Installation

Explication de l'installation des pytools, de croco et de l'activation de l'environnement Anaconda/Miniconda *croco\_pyenv*.

##### installation de croco

```

# en se plaçant dans PATH-TO-CROCO/
git clone --branch v2.0.1 https://gitlab.inria.fr/croco-ocean/croco.git croco-v2.0.1

```

##### installation de miniConda

```

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
sudo chmod +x Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -p $HOME/miniconda3
source $HOME/miniconda3/bin/activate
conda init

```

##### installation des croco\_pytools

```

# en se plaçant dans PATH-TO-CROCO/
git clone --branch release https://gitlab.inria.fr/croco-ocean/croco_pytools.git
cd croco_pytools/prepro/
# conda activate croco\_pyenv
python3 install.py
> Do you want to install conda environment? [y,[n]]: y
> Do you want to compile fortran tools? y,[n]: y

```

Une fois croco et les croco\_pytools installés, l'architecture de la configuration doit être générée. Nativement, croco contient un fichier *create\_config.bash* qui effectue ce travail. Toutefois, comme l'architecture choisie pour ce rapport est différente de celle de référence, un nouveau fichier *create\_config.bash* a été écrit, son contenu est défini ci-après.

##### contenu de *create\_config.bash* adapté à l'architecture choisie pour ce rapport

```

# croco source directory
# -----
CROCO_DIR=~ /PATH-TO-CROCO /croco

# Home and Work configuration directories
# -----
MY_CONFIG_HOME=~ /PATH-TO-CONFIGS /CONFIGS

# croco_tools directory for matlab tools
# -----
# TOOLS_DIR=~ /PATH-TO-CROCO /croco_pytools
# For pre-traitement WITH MATLAB:

```

```

#cp ${TOOLS_DIR}/crocotools_param.m ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/
#cp ${TOOLS_DIR}/start.m ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/

# Configuration name
# -----
MY_CONFIG_NAME=CONFIG-NAME # put the name you want

# For configuration initialisation
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets/Bry
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets/Bulk
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets/gshhs
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets/Ini
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/datasets/Topo

# For pre-treatment WITH PYTHON :
mkdir ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/preproOUTPUT

# For CROCO compiling
cp ${CROCO_DIR}/OCEAN/cppdefs.h ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/
cp ${CROCO_DIR}/OCEAN/param.h ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/
cp ${CROCO_DIR}/OCEAN/jobcomp ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/

# For running
cp ${CROCO_DIR}/OCEAN/croco.in ${MY_CONFIG_HOME}/${MY_CONFIG_NAME}/

```

## Plan général des étapes

Le schéma Figure1 résume les étapes de décision et de traitement de données et d'exécution de codes décrit ci-après.

### Plan des processus généraux à suivre pour mettre en place un modèle fonctionnel

#### Paramètres du modèle

Avant de rechercher les données qui seraient trop lourdes à télécharger à l'échelle globale, il convient de déterminer les propriétés principales de la zone d'étude et de son modèle. Ces paramètres sont :

1. les latitudes et longitudes du centre de la zone d'étude totale,
2. l'étendue en kilomètres de la zone d'étude dans le sens de la latitude et de la longitude,
3. la gamme de bathymétrie que doit couvrir le modèle (minimum et maximum),
4. les paramètres d'interpolation qui doivent être utilisés pour les données de bathymétrie,
5. les coordonnées d'un point de la grille se trouvant dans le plan d'eau d'intérêt si le modèle ne doit tourner que sur un plan d'eau,
6. les dates de début et de fin de la modélisation,
7. les valeurs associées à la répartition des mailles verticales de la grille. Leur nombre  $N$  doit aussi être déterminé.

#### Données en entrée

En entrée, les données brutes nécessaires doivent toujours couvrir spatialement l'intégralité de la zone d'étude. Elles doivent être placées dans le dossier PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/datasets/. Elles sont :

1. la bathymétrie [topo.nc] couvrant au moins la zone d'étude,
2. le masque côtier [GSHHS\_f\_L1.shp] comprenant au moins les côtes de la zone d'étude,
3. les données marines [copernicus\_marine\_data.nc] couvrant une zone la plus large possible autour et comprenant la zone d'étude. Ces données doivent aussi couvrir le plus finement possible la gamme temporelle mentionnée dans la section précédente,
4. les données issues d'un modèle global ou local de marée, par exemple TPXO10 ou FES2014 [FES2014\*.nc] incluant des données de hauteur d'eau et de courant pour les différentes fréquences de vagues de marée (M2, S2, N2, K2, K1, O1, P1, Q1, MF, MM,...)

5. le fichier texte présent nativement dans les croco\_pytools [tides.txt] contenant les noms et les périodes des différentes harmoniques des vagues de marée.
6. les données atmosphériques [copernicus\_atmospheric\_data.nc] couvrant une zone large autour et comprenant la zone d'étude. Ces données doivent couvrir le plus finement possible la gamme temporelle mentionnée dans la section précédente.

Les **paramètres** qu'il conviendra de récupérer dans la documentation des données de forçage (initial et aux limites) sont :

1. les dates qui correspondent à l'origine du temps donné dans les fichiers téléchargées (généralement 1970/01/01),
2. la nomenclature des variables devra aussi être vérifiée en fonction du contenu de *croco\_pytools/prepro/Readers/ibc\_reader.py*,
3. l'orientation des limites de la grille ouvertes dans le plan d'eau,
4. la cyclicité potentielle des données entrées.

### Pré-traitement

Le pré-traitement des données brutes n'est détaillé ici que pour les croco\_pytools. Les outils Matlab ne sont pas abordés.

#### - modifications basiques des fichiers netCDF

Certaines modifications peuvent être préférablement réalisées avant d'utiliser les outils de pré-traitement. Ces modifications peuvent être effectuées en utilisant la librairie NCO de traitement des fichiers netCDF

##### Changer le nom d'une variables du fichier

```
# renome la variable t2m en HR
nco cname,t2m,HR
```

Cette commande n'est pas obligatoire car les fichiers "ibc\_reader.py", "make\_frc\_tide.py" et "make\_blk\_interpol.py" peuvent être adaptés à la main. La commande reste utile si on souhaite dupliquer une variable (La marche à suivre est : dupliquer un fichier en in1.nc et in2.nc, puis renommer la variable A en A2 du fichier nc2.nc, puis concaténer les fichiers in1.nc et in2.nc. Le fichier de sortie contient alors les deux variables identiques A et A2).

##### Rognage des données selon une dimension

```
# rogne selon les indices de la dimension time et
# selon les valeurs des dimensions longitude et latitude
ncks -d time,240,359 -d longitude,-2.,3. -d latitude,48.,50. in.nc out.nc
```

Si des entiers (exemple : 40) sont donnés pour la gamme à restreindre d'une dimension, ces valeurs sont interprétées comme les **indices** des dimensions minimum et maximum. Si des décimaux (exemple : 40.) sont donnés, les valeurs sont interprétées comme les **valeurs** minimales et maximales des dimensions.

##### Concaténation de plusieurs fichiers selon les dimensions communes

```
# concatene tous les fichiers data_*.nc en un fichier data_tot_calc.nc
ncks -h -A raw_download/data_*.nc data_tot_calc.nc
```

##### Effectuer une opération arithmétique sur plusieurs variables du fichier

```
# calcule et stocke dans RH les valeurs d'humidite relative selon
# les valeurs des variables d2m (temperature du point de rosee)
# et t2m (temperature)
ncap2 -s 'RH=10^(7.591386*(d2m/(d2m+240.7263)-t2m/(t2m+240.7263)))'\
in.nc out.nc
```

#### Effectuer une opération arithmétique sur plusieurs variables du fichier

```
# change a "new attribute str value" la valeur de l'attribut "attribute"
# de la variable "variable" du fichier "in.nc"
ncatted -a attribute,variable,o,c,"new attribute str value" in.nc
```

#### - Génération de la grille

La grille *croco\_grd.nc* est la base des prochaines actions de pré-traitement ainsi que du modèle. Elle contient toutes les données relatives aux coordonnées de chaque maille de la grille sur laquelle le va travailler, ainsi que les données de bathymétrie, un paramètre de Coriolis et les coordonnées curvilinéaires des mailles.

#### générer la grille en mode graphique

```
# dans PATH-TO-CROCO/croco_pytools/prepro/
python3 make_grid.py
> Do you want to use interactive grid maker ?
> (e.g., for grid rotation or parameter adjustments) : y,[n] y
```

Dans la fenêtre graphique, les champs doivent être modifiés pour correspondre aux paramètres voulus. Ils peuvent être aussi déterminés et choisis par tâtonnement en utilisant les boutons *Compute grid* et *Compute smoothing* successivement.

Les valeurs utilisées pour modéliser le Cotentin (France) ainsi que les visualisation proposées sont présentées en Figure 5.

Une fois la grille souhaitée obtenue, la sauvegarder en utilisant le bouton *Save grid*.

#### - Génération des fichiers de données

Tous les codes de pré-traitement suivants peuvent être exécutés dans n'importe quel ordre en se plaçant dans le répertoire du code voulu et en indiquant l'intitulé de ce code dans la commande :

#### exécuter un code de pré-traitement

```
python3 <name_of_the_pre-pro_code>.py
```

Au début de chaque code, une partie dans l'entête encadrée par `###...## USER CHANGES ###...###` et `###...## end USER CHANGES ###...###` doit être relue pour indiquer les bon paramètres ainsi que les noms et registres des fichiers de données brutes nécessaires.

Les codes devant être exécutés sont dans le répertoire *croco\_pytools/* :

1. *prepro/make\_ini.py* génération des conditions initiales à partir des données marines,
2. *prepro/make\_bry.py* génération des conditions (de pression, température, salinité, etc.) aux bords ouverts sur le plan d'eau de la grille, à partir des données marines,
3. *make\_frc\_tide.py* génération des conditions de forçage par les modèles de marées pour les bords ouverts sur le plan d'eau de la grille, à partir des données issues d'un modèle de marée,
4. *make\_blk\_interpol.py* génération des conditions de forçage atmosphériques à la surface du plan d'eau sur l'ensemble de la grille, à partir des données atmosphériques.

Ces codes vont respectivement générer les fichiers de données :

1. *croco\_ini.py*
2. *croco\_bry.py*
3. *croco\_frc.py*
4. *croco\_blk.py*

Le schéma XXX détaille les dépendances de fichiers et les paramètres à spécifier pour chaque code de pré-traitement.

Attention, à partir de deux grilles filles, il convient de corriger manuellement le fichier *AGRIF\_FixedGrids.in*.

#### Modélisation

#### - Modification des fichiers de compilation

Les fichiers qui doivent être modifiés (ou écrits) sont :

- croco.in (+ croco.in.n avec n allant de 1 au nombre de grilles enfants)
- param.h
- cppdefs.h
- jobcomp

#### - Compilation du modèle

##### commande pour compiler le code CROCO

```
# dans PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/  
./jobcomp
```

#### - Exécution du modèle

##### commande pour exécuter le code CROCO

```
# dans PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/  
./croco
```

#### Fichiers de sortie

La compilation du modèle produit les fichiers :

- de dossier Compile dans son intégralité,
- croco,
- ncjoin,
- partit,
- kRGB61.txt.1,
- kRGB61.txt

### 8.1.2 Détail des processus du pré-traitement

Un schéma détaillé de la marche à suivre pour générer les fichiers nécessaires au modèle à partir des données brutes, c'est à dire le pré-traitement, est présenté en Figure 2. Les chemins de fichiers présentés dans le rapport correspondent à la nomenclature définie en section 8.1.1.

[LA SUITE EN ANNEXES ?]

#### make\_grid.py

L'entête du code *make\_grid.py* doit être adaptée aux paramètres choisis et à l'emplacement des données.

##### modifications utilisateur · ice de make\_grid.py

```
#--- USER CHANGES -----  
  
# Grid center [degree]  
tra_lon = 15 # Longitude of the grid center  
tra_lat = -32 # Latitude of the grid center  
  
# Grid size [km]  
size_x = 1556  
size_y = 1334  
  
# Grid number of points  
# Note: grid resolution is grid size / number of points  
nx = 39  
ny = 40  
  
# Grid rotation [degree]  
rot = 0  
  
# Smoothing parameters
```

Configure grid

View: grid outline

longitude: -1.28  
latitude: 49.6  
x size [km]: 150  
y size [km]: 150  
rotation [deg]: 0  
nx: 80  
ny: 80  
Figure projection: Plate

Coastline file: ./../CONFIGS/CUL\_DE\_LOUP/datasets/gshhs/GSHHS\_shp/ff/GSHHS\_f\_L1  
Topography file: ./../CONFIGS/CUL\_DE\_LOUP/datasets/Topo/GEBCO\_2024\_sub\_ice\_topo

```

--- computing grid (nx=80, ny=80, lon=-1.28, lat=49.60, sx=150.00, sy=150.00,
rot=0.00)
--- min=dx 1874.87, max=dx 1875.00
--- min=dy 1874.87, max=dy 1875.00

--- computing grid (nx=80, ny=80, lon=-1.28, lat=49.60, sx=150.00, sy=150.00,
rot=0.00)
--- min=dx 1874.87, max=dx 1875.00
--- min=dy 1874.87, max=dy 1875.00

```

Compute grid

Topo smoothing

Create offline zoom

Create zoom AGRIF

Minimum depth [m]: 0

Maximum depth [m]: 200

Interp radius [nb croco points]: 3

r-factor: 0.2

Smoothing method: lsmooth

Single connect: ☒ i0: 20 j0: 20

```

--- computing grid (nx=80, ny=80, lon=-1.28, lat=49.60, sx=150.00, sy=150.00,
rot=0.00)
--- min=dx 1874.87, max=dx 1875.00
--- min=dy 1874.87, max=dy 1875.00

--- computing grid (nx=80, ny=80, lon=-1.28, lat=49.60, sx=150.00, sy=150.00,
rot=0.00)
--- min=dx 1874.87, max=dx 1875.00
--- min=dy 1874.87, max=dy 1875.00

```

Compute smoothing

Output file: ./../CONFIGS/CUL\_DE\_LOUP/croco\_grd.nc

Save grid

FIGURE 5 – Caption.

```

# (see online documentation for more details)
hmin      = 50      # Minimum depth [m]
hmax      = 6000    # Maximum depth [m]
interp_rad = 2      # Interpolation radius in number of points
#                      (usually between 2 and 8)
rfact     = 0.2     # Maximum r-fact to reach
#                      (the lower it is, the smoother it will be)
smooth_meth = 'lsmooth' # Smoothing method ('smooth', 'lsmooth',
#                      'lsmooth_legacy', 'lsmooth2', 'lsmooth1',
#                      'cond_rx0_topo')

# Topo/Bathy file
topofile = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/dataset/Topo/etopo2.nc'

# Coastline file (for the mask)
shp_file = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/dataset/gshhs/GSHHS_i_L1.shp'

# Single Connect [Mask water not connected to the main water body]
sgl_connect=[False,20,20] # point indices inside the main water body
#                      (True or False)

# Output grid file
output_file="PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/preproOUTPUT/croco_grd.nc"

#--- END USER CHANGES -----

```

+ modifs en graphic mode si on souhaite générer des sous-grilles (AGRIF) voir Figure 5.

**make\_ini.py**

L'entête du code *make\_ini.py* doit être adaptée aux paramètres choisis et à l'emplacement des données.

21

### modifications utilisateur · ice de *make\_ini.py*

```
#--- USER CHANGES -----

# Dates
# starting date
Yini, Mini, Dini, Hini = '2013', '11', '28', '01' # Month and days need to be
#           2-digits format reference time (default = ini time)
Yorig, Morig, Dorig= '1970', '01', '01' # Month and days need to be
#           2-digits format

# Input data information and formatting
inputdata = 'copernicus' # Input data dictionary as defined in the
#           Readers/ibc_reader.py
input_dir = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/dataset/Ini/'
input_prefix='copernicus_marine_data'
input_file = f'{input_dir}{input_prefix}Y{Yini}M{Mini}.cdf'
input_file = f'{input_dir}{input_prefix}.nc'
multi_files=False # If variables are in different netcdf
if multi_files: # Multiple files
input_file = { 'ssh' : input_dir + input_prefix + 'ETAN.Y2013M01.nc',\
               'temp' : input_dir + input_prefix + 'THETA.Y2013M01.nc',\
               'salt' : input_dir + input_prefix + 'SALT.Y2013M01.nc',\
               'u' : input_dir + input_prefix + 'EVEL.Y2013M01.nc',\
               'v' : input_dir + input_prefix + 'NVEL.Y2013M01.nc'\
             }

# time index to use in the file
tndx = 0

# default value to consider a z-level fine to be used
Nzgoodmin = 4

# tracers
tracers = ['temp','salt']

# CROCO grid informations
croco_dir = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/preproOUTPUT/'
croco_grd = 'croco_grd.nc'
sigma_params = dict(theta_s=7, theta_b=2, N=32, hc=200) # Vertical streching,
#           sig_surf/sig_bot/ nb level/critical depth

# Ini file informations
ini_filename = 'croco_ini.nc' # output will be put in croco_dir by default

# Conserv OGCM transport option
conserv=1 # Correct the horizontal transport i.e. remove the integrated
#           transport and add the OGCM transport

#--- END USER CHANGES -----
```

Pour que les codes d'initialisation et de conditions aux limites fonctionnent correctement, il convient de s'assurer que le code *Readers/ibc\_reader.py* soit correctement rempli. Ici, on ajoute le dictionnaire pour les données copernicus.

### modifications de *Readers/ibc\_reader.py*

```
# ajouter dans la fonction lookvar(input):
# avant la ligne "else:" (vers la ligne 55)

elif input == 'copernicus':
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'\
}

```

## make\_bry.py

L'entête du code *make\_bry.py* doit être adaptée aux paramètres choisis et à l'emplacement des données.

### modifications utilisateur · ice de *make\_bry.py*

```

#--- USER CHANGES -----

# Dates
Yorig, Morig, Dorig = '1970', '01', '01' # origin of time as: days since
#      Yorig-Morig-Dorig-00h00
Ystart, Mstart, Dstart, Hstart = '2013', '11', '28', '01' # Starting month
Yend, Mend, Dend, Hend = '2014', '03', '01', '01' # Ending month

# Input data information and formatting
inputdata = 'copernicus' # Input data dictionary as defined in the
#      Readers/ibc_reader.py
input_dir = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/datasets/Bry/'
input_prefix = 'copernicus_marine_data' # Please use * to include all files
multi_files = False
if multi_files: # Multiple data files. Time is read in ssh file
input_file = {'ssh':sorted(glob.glob(input_dir+input_prefix+'ETAN*.nc')),\
'temp':sorted(glob.glob(input_dir+input_prefix+'THETA*.nc')),\
'salt':sorted(glob.glob(input_dir+input_prefix+'SALT*.nc')),\
'u':sorted(glob.glob(input_dir+input_prefix+'EVEL*.nc')),\
'v':sorted(glob.glob(input_dir+input_prefix+'NVEL*.nc'))\
}
else: # glob all files
input_file = sorted(glob.glob(input_dir + input_prefix))

# default value to consider a z-level fine to be used
Nzgoodmin = 4

# Tracers
tracers = ['temp', 'salt']

# CROCO grid informations
croco_dir = 'PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/preproOUTPUT/'
croco_grd = 'croco_grd.nc'
sigma_params = dict(theta_s=7, theta_b=2, N=32, hc=200) # Vertical streching,
#      sig_surf/sig_bot/ nb level/critical depth

# Bry file informations
bry_filename = 'croco_bry.nc' # output will be put in croco_dir by default
obc_dict = dict(south=1, west=1, east=1, north=1) # open boundaries
#      (1=open , [S W E N])
output_file_format = "FULL" # How outputs are spit (MONTHLY, YEARLY, FULL)
cycle_bry = 0.

# Conserv OGCM transport option
conserv=1 # Correct the horizontal transport i.e. remove the integrated
#      transport and add the OGCM transport

#--- END USER CHANGES -----

```



Quelques modifications doivent aussi être effectuées dans le corps du code comme décrit ci-dessous.

#### modifications supplémentaires de *make\_bry.py*

```
# ligne 111
# ORIGINAL :
start_date = Ystart+Mstart+'01'+'12' # default start day is 1st
# NOUVEAU :
start_date = Ystart+Mstart+Dstart+Hstart # default start day is 1st

# ligne 119
# ORIGINAL :
dtenddt = plt.datetime.datetime(int(Yend),int(Mend),1,12) \
# NOUVEAU :
dtenddt = plt.datetime.datetime(int(Yend),int(Mend),int(Dend),int(Hend)) \
```

#### *make\_blk\_interpol.py*

L'entête du code *make\_blk\_interpol.py* doit être adaptée aux paramètres choisis et à l'emplacement des données.

le code est présenté en annexes, ajouter un schéma du fonctionnement du code.

schéma descriptif du fonctionnement du code [FIGURE À FAIRE]

#### modifications utilisateur · ice de *make\_blk\_interpol.py*

```
##### User changes #####
# don't change de left part of the dictionnary (keys)

INPUT_PARAM = { "lon": "longitude",
                "lat": "latitude",
                "time": "valid_time",
                "Tair": "t2m",
                "HumiditeRelat": "RH",
                "PrecipitationRate": "avg_tprate",
                "WindSpeed": "si10",
                "NetLWRadiation": "avg_snlwrf",
                "DownwardLWRadiation": "avg_sdlwrf",
                "SWRadiation": "avg_snsrwr",
                "UStress": "avg_iewr",
                "VStress": "avg_insr",
                "UWind": "u10",
                "VWind": "v10"
}

INPUT = {
    "dir": "PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/datasets/Bulk/",
    "file": 'cpernicus_atmospheric_data.nc',
    "param": INPUT_PARAM
}

output_param = {"time": "bulk_time",
                "Tair": "tair",
                "HumiditeRelat": "rhum",
                "PrecipitationRate": "prate",
                "WindSpeed": "wspd",
                "NetLWRadiation": "radlw",
                "DownwardLWRadiation": "radlw_in",
                "SWRadiation": "radsw",
                "UStress": "sustr",
                "VStress": "svstr",
                "UWind": "uwnd",
                "VWind": "vwnd"
}
```

```

OUTPUT = {  "dir": "PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/preproOUTPUT/",
            "file": 'croco_blk.nc',
            "param": output_param
}

NB_VOISINS = 8

sigma_params = dict(theta_s=0, theta_b=0, N=1, hc=1)

##### END User changes #####

```

### make\_frc\_tide.py

L'entête du code *make\_frc\_tide.py* doit être adaptée aux paramètres choisis et à l'emplacement des données.

le code complet est présenté en annexes, ajouter un schéma du fonctionnement du code.

schéma descriptif du fonctionnement du code [FIGURE À FAIRE]

#### modifications utilisateur · ice de *make\_frc\_tide.py*

```

# sera modifie pour que ce soit des disctionnaires si on le veut dans le rapport

##### User changes #####

Correction_uv = False
tides = ['M2', 'S2', 'N2', 'K2', 'K1', 'O1', 'P1', 'Q1', 'Mf', 'Mm']

grid_dir = "PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/preproOUTPUT/"
grid_name = "croco_grd.nc"
#grid_param = ["lon_rho", "lat_rho", "lon_u", "lat_u", "lon_v", "lat_v"]
grid_param = ["lon_rho", "lat_rho", "lon_rho", "lat_rho", "lon_rho", "lat_rho"]
print("\n grid param : \n", grid_param)
grid_param_meaning = ["lonH", "latH", "lonU", "latU", "lonV", "latV"]
# don't change this order

input_dir = "PATH-TO-CONFIGS/CONFIGS/CONFIG-NAME/datasets/Bry/AVISO/"
TPX0_or_AVISO = True # 0 or False for TPX0 and 1 or True for AVISO
multi_file = True # 0 or False if monofile and 1 or True if multfile
name_is_tide = True # if the files names ar tide_name.nc
#input_file_name = 'FES2014*.nc'
input_file_name = [ "ocean_tide_extrapolated/",
                    "eastward_velocity/",
                    "northward_velocity/"
]
prefix = ""
sufix = ".nc"
if name_is_tide : # generate automatically the multfile names
path_names = copy.copy(input_file_name)
for i,path in enumerate(path_names):
input_file_name[i] = []
for tide in tides:
input_file_name[i].append(path + prefix + tide.lower() + sufix)
input_file_name = np.array(input_file_name)
print("\ninput directory : \n", input_dir)
print("\ninput files : \n", input_file_name)

input_param_names = [ ["lat", "lon", "phase", "amplitude"],
["lat", "lon", "Ug", "Ua"],
["lat", "lon", "Vg", "Va"],
]
print("\n input param names : \n", input_param_names)
input_param_meaning = [ ["latH", "lonH", "phaseH", "amplitudeH"],
["latU", "lonU", "phaseU", "amplitudeU"],
["latV", "lonV", "phaseV", "amplitudeV"],

```

```

] # keep same names as grid_param_meaning
print("\n input param meaning : \n", input_param_meaning)

output_dir = "../CONFIGS/CUL_DE_LOUP/GRIDS/"
output_file_name = 'croco_frc.nc'
output_param = ["tide_period", "tide_Ephase", "tide_Eamp",
"tide_Cmin", "tide_Cmax", "tide_Cangle", "tide_Cphase"]
# don't change the order

tide_param_path="prepro/Modules/tides.txt"

NB_VOISINS = 8

##### END User changes #####

```

### 8.1.3 Détail des processus du modèle

schéma détaillé de la marche à suivre pour faire fonctionner le modèle à partir des fichiers du pré-traitement.

## 8.2 Le modèle "Anse du Cul-de-Loup"

Grosse partie dans laquelle il va falloir expliquer les différentes étapes de mis en place du modèle ADCL  
détailler les paramètres choisis

## 8.3 Mesures de terrain - Le projet PROTEC

Pour moi : quelques mots sur les données disponibles pour comparer les résultats modèle avec des mesures de terrain (cf Discussion)