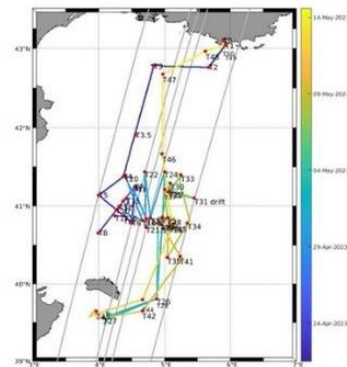
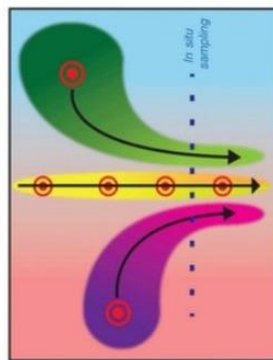


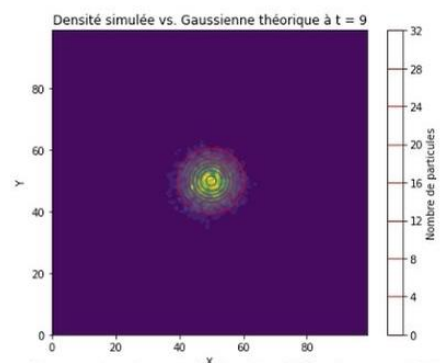
## Rapport de Stage : Modèle de dispersion dans un courant à jet



Campagne océanographique BioSWOT-Med  
(Doglioli A. *et al.*, 2023)



Déplacement des particules  
dans un courant à jet  
(Mangolt *et al.*, 2023)



Exemple de modèle de diffusion en 2D

Lieu : Institut Méditerranéen d'Océanologie - 13009, Marseille  
Dates : 22 Mai 2025 - 04 Juillet 2025

Maître de Stage : Andrea M. DOGLIOLI

Juliette GARNIER  
L3 SVT - parcours Mer

Année 2024 / 2025

## Table des matières :

Remerciements.....	3
I – Introduction .....	3
II - Matériel et Méthodes.....	5
1. Diffusion .....	9
a. Comparaison Solution théorique et Modèle réel .....	9
b. Ajout de conditions de bord .....	11
2. Advection et Diffusion .....	14
a. Action d’un courant uniforme selon Ox .....	14
b. Action d’un jet gaussien.....	18
3. Tests de mélange des masses d’eau .....	20
a. Absence d’advection .....	20
b. Influence d’un jet gaussien .....	21
4. Utilisation de paramètres in situ issus de la campagne BioSWOT-Med.....	22
a. Diffusion au sein du front Nord-Baléares.....	24
b. Diffusion au sein du Golfe du Lion .....	30
c. Population de phytoplanctons soumise à un terme de “Grazing” .....	31
IV – Conclusion .....	34
VI – Annexes.....	35

# Remerciements

Je remercie mon tuteur de stage A. Doglioli pour ses conseils avisés et sa volonté de transmettre qui ont permis de m'aiguiller et ont contribué à la bonne réalisation de ce stage de fin de licence. J'adresse également mes remerciements au centre de recherche du MIO qui m'a permis de travailler dans de très bonnes conditions.

## I – Introduction

L'océanographie est une discipline complexe qui nécessite un couplage des approches mathématiques, physiques, biologiques et chimiques pour appréhender l'ensemble des paramètres qui caractérisent les océans. Dans le cadre de la troisième année de licence en sciences de la mer, il m'a été proposée de réaliser un modèle théorique de dispersion dans un courant à jet qui emploie des paramètres conservatifs de salinité et non conservatifs de chlorophylle a issus de la campagne océanographique BioSWOT-Med qui s'est déroulée du 20 Avril au 14 Mai 2023 au sein du font Nord-Baléares avec une zone d'étude à 100 km au nord-est de l'île de Minorque (Doglioli A. *et al.*, 2025). Cette campagne s'est notamment basée sur des mesures à haute résolution de l'anomalie de la surface de la mer (SSH) par le satellite SWOT (Sea Water and Ocean Topography) qui a permis l'utilisation de stratégies d'échantillonnage adaptées aux fines échelles (1-100 km) capables d'enregistrer des structures éphémères allant d'une durée d'un jour à un an tels que les fronts, les tourbillons et les structures filamenteuses ; ce qui constitue une première en océanographie.

Un front est une structure à méso-échelle caractérisée par quelques centaines de mètres à plusieurs dizaines de kilomètres. Il marque la frontière entre deux masses d'eaux aux propriétés physiques et biogéochimiques distinctes dont le gradient des différents paramètres scalaires (Salinité, Température, Chlorophylle a, etc.) y est bien plus marqué qu'au sein de la circulation à grande échelle (Pedlosky J., 1979). En effet, ce front peut être défini comme une zone de front halin marquée par la convergence horizontale des eaux atlantiques sud (AWs) jeunes et peu salées et méditerranéennes nord (mAWs) vieilles et plus salées dans les 200 premiers mètres au nord-ouest de la mer Méditerranée. L'existence d'un front thermique saisonnier est également à prendre en compte et se présente sur les 50 premiers mètres de la colonne d'eau s'étendant des Pyrénées jusqu'au nord-ouest de la Corse. Enfin, d'autres

auteurs l'associent à une circulation de fermeture entre Minorque et la "Gyre Nord" Corse qui représente une boucle cyclonique de courants (Q.B. Barral, 2022).

Par ailleurs, la mise en évidence de l'influence du forçage physique induit par la frontogenèse au sein de l'organisation des structures biologiques a notamment permis de définir les fronts écologiques comme la délimitation des niches écologiques, d'après l'étude menée par F. d'Oviedo *et al.* (2010). Ainsi, l'apparition des structures physiques de fine échelle peut être corrélée à la mise en place d'épisodes de bloom phytoplanctoniques agissant comme moteur de la diversité planctonique et influençant l'ensemble de la chaîne trophique.

La caractérisation de ce front étant encore peu documentée, il est nécessaire de réaliser des modèles théoriques en amont, notamment sur la base de données satellitaires puis de tester leur validité et les perfectionner en implémentant des mesures réalisées sur le terrain. Grâce aux paramètres échantillonnés *in situ*, il est possible de comparer les jeux de données obtenus avec les équations théoriques de l'hydrodynamique. Mais on peut également s'en servir dans des modèles afin de reconstituer l'évolution spatiotemporelle d'une zone géographique donnée et de réaliser des simulations de l'impact du réchauffement climatique sur les structures identifiées, comme les fronts. C'est ce dernier volet qui a été exploité au cours de ce stage où l'utilisation de petits programmes informatiques réalisés sur le logiciel Python ont eu pour but de répondre aux questions suivantes :

- Dans quelle mesure les masses d'eau se diffusent au sein du front thermohalin ?
- Comment évolue la tâche de diffusion d'une concentration de particules selon sa position au sein d'un jet gaussien ?
- Quelle est la proportion de mélange de deux masses d'eau aux propriétés biogéochimiques différentes au cours du temps ?

De ce fait, ce stage a pour objectif d'identifier le taux de mélange des masses d'eaux qui se rencontrent au sein du front Nord-Baléares et de caractériser l'évolution spatiotemporelle de la concentration en chlorophylle a en son sein.

## II - Matériel et Méthodes

Pour répondre au mieux à la problématique du stage nous nous sommes d'abord appuyés sur les lois physiques régissant la dynamique des océans afin de caractériser la mise en mouvement des masses d'eau et la diffusion des particules au sein d'un fluide. Les lois de conservation régissent en effet la circulation océanique et l'on distingue la loi de la conservation de la masse :  $\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = 0$  dont on peut déduire l'équation de diffusion qui décrit le déplacement des particules diffusées de façon aléatoire selon un mouvement brownien :

$\text{div} \left[ D_\phi \vec{\text{grad}} \phi \right] = \frac{\partial \phi}{\partial t}$  (avec  $\phi$  la grandeur diffusée). Ainsi que la loi de conservation de la quantité de

mouvement :  $\vec{p} = m\vec{v}$  dont découlent les équations d'Euler :  $\rho \frac{d\vec{v}}{dt} = -\vec{\text{grad}} p + \rho \vec{f}$  (valables pour un fluide parfait, en absence de viscosité). Ces dernières équations sont à l'origine des équations de Navier-Stokes qui décrivent la mise en mouvement des masses d'eau avec prise en compte de la viscosité et de la rotation de la Terre sous l'hypothèse que le fluide soit incompressible :

$$\frac{d\vec{v}}{dt} = -\frac{1}{\rho} \vec{\nabla} p + \vec{g} - 2\vec{\Omega} \times \vec{v} + \nu \vec{\nabla}^2 \vec{v}$$

(ici sous forme vectorielle). L'utilisation des équations de

Navier-Stokes permet notamment de caractériser les écoulements turbulents. En effet, la turbulence signifie que l'écoulement d'un fluide présente en tout point des structures tourbillonnaires à l'apparence très désordonnée, qui s'oppose aux fluides laminaires s'écoulant de façon régulière. Même si les équations de Navier-Stokes ne sont pas résolubles directement, il est néanmoins possible de déterminer la vitesse de notre fluide en procédant à la décomposition de Reynolds. Nous l'utilisons ici puisque les mesures de la vitesse des masses d'eau au sein du front Nord-Baléares présentent un nombre de Reynolds largement supérieur à 1 traduisant la présence d'un régime d'écoulement turbulent. Ainsi, cette méthode consiste à associer à chaque composante de la vitesse une partie moyennée en fonction du temps additionnée à un écart à la moyenne, comme indiqué ci-contre, (A. Doglioli, 2025) :

Décomposition de Reynolds en 1D :  $u = \bar{u} + u'$  avec  $\bar{u}' = 0$

Par la suite, nous avons appliqué les équations de l'hydrodynamique à nos modèles informatiques. Afin de réaliser un modèle de dispersion dans un courant à jet, il a été nécessaire d'effectuer plusieurs sous-programmes simples et indépendants qui répondent

chacun à un objectif de la problématique donnée. Ainsi, chaque étape correspond à une partie distincte de l'équation qui régit notre modèle :

Equation générale du modèle d'advection - diffusion 2D :

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x} \left( D_x \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left( D_y \frac{\partial U}{\partial y} \right) - v_x \frac{\partial U}{\partial x} - v_y \frac{\partial U}{\partial y} + F(U) \quad (\text{J.-C. Poggiale, 2021})$$

Où  $U$  représente la variable d'état de la concentration à l'instant  $t$  localisée au point  $(x,y)$ ,  $F(U)$  est une fonction décrivant les processus de variation de  $U$  et  $D_x$  ainsi que  $D_y$  sont les coefficients de diffusion selon les composantes  $x$  et  $y$ . Ici les deux premiers termes de l'égalité représentent la diffusion des particules selon l'axe  $Ox$  et  $Oy$  tandis que les deux derniers termes de l'égalité régissent l'advection avec  $v = (u,v)$  le champ de vitesse en deux dimensions.

Dans un premier temps, nous avons cherché à comparer l'évolution de la diffusion de la concentration de particules au cours du temps entre un jeu de données réaliste et un taux de diffusion théorique. Pour cette étape, nous avons suivi le modèle de l'équation de diffusion

(1D) :  $\frac{\partial U}{\partial t} = \frac{\partial}{\partial x} \left( D_x \frac{\partial U}{\partial x} \right)$ . Pour ce faire, nous avons simulé une trajectoire aléatoire de nos particules suivant un mouvement brownien pour caractériser l'évolution réaliste de leur position au cours du temps. Puis nous avons cumulé leur probabilité d'apparition dans un plan afin d'en extraire leur densité. Nous avons en parallèle défini une diffusion théorique sous forme de gaussienne qui dépend de paramètres constants comme un coefficient de diffusion établi par le modélisateur. Enfin, nous avons établi une boucle temporelle pour simuler l'évolution de la densité de nos particules réelles et théoriques au cours du temps. Afin d'obtenir une vision d'ensemble du transport diffusif, nous avons effectué une première simulation en une dimension en représentant la densité de particules issues d'un jeu de données aléatoire sous forme d'histogramme et la diffusion théorique de celle-ci avec une

gaussienne défini par l'équation :  $G_1(t,x,y) = \frac{1}{\sqrt{\pi Dt}} e^{-\frac{(x-x_0)^2}{4Dt}}$ . Puis nous avons simulé la diffusion en 2D

en imaginant l'observateur au-dessus d'une masse d'eau immobile et regardant une tâche d'encre se répandre dans celle-ci. Pour cette seconde simulation, la densité de particules

théoriques a été représentée par la gaussienne :  $G_2(t,x,y) = \frac{C_0}{\pi Dt} e^{-\frac{(x-x_0)^2}{4Dt}} e^{-\frac{(y-y_0)^2}{4Dt}}$ .

Par la suite, afin de rendre notre simulation réaliste, nous avons également défini des conditions de bord agissant comme limites de notre zone d'étude. Ainsi, en se plaçant en deux dimensions, on fait l'hypothèse que pour une certaine valeur de  $x$  et de  $y$  de notre particule  $P(x,y)$ , on compte sa position dans la densité de particules si celle-ci se trouve au sein de notre grille d'observation et on ne prend pas en compte sa position si celle-ci sort de notre zone d'étude. Cette étape nous permet de replacer notre phénomène de diffusion au sein de la zone d'étude qui se trouvait à une centaine de kilomètres de la côte. Pour ce faire, on a attribué une valeur aberrante aux particules dont on ne souhaitait pas dénombrer la densité. Cela avait également pour but de ne pas dénombrer plusieurs fois la même particule et de simuler un phénomène de perte de matière initialement introduite dans notre système afin d'obtenir des densités de particules se rapprochant le plus possible des observations.

Après l'étude de la diffusion seule, nous avons souhaité comprendre son évolution sous l'influence d'une advection. Dans un premier temps, nous avons soumis notre diffusion à un champ de vitesse homogène en deux dimensions simulant un courant uniforme. Pour ce faire, nous avons introduit des valeurs de vitesse constantes simulant un champ de vitesse homogène et nous avons également inséré notre programme de diffusion en le faisant varier dans l'espace sur un second axe de déplacement. Ce programme s'est basé sur la partie simulant l'advection en 1D du modèle final advection-diffusion :  $\frac{\partial C}{\partial t} = -v \frac{\partial C}{\partial x}$ . Dans un second temps, nous avons soumis notre modèle de diffusion à un courant non uniforme agissant comme un jet gaussien pour simuler l'évolution de la vitesse au sein du front Nord-Baléares. Ici le jet gaussien agit dans le plan  $(O,x,y)$ . La norme du champ de vitesse est plus élevée au centre du jet où le gradient est plus important et il décline vers les bords où les vitesses s'affaiblissent et reprennent la valeur moyenne de l'écoulement de cette région. Pour prendre le modèle dans son ensemble, nous avons également simulé l'influence de différentes positions de la tâche de diffusion au sein du jet gaussien afin de caractériser les proportions de mélange locaux au sein du front Nord-Baléares pour pouvoir le recouper plus tard avec des données *in situ*.

Dans le but d'appréhender l'évolution de la rencontre de deux masses d'eau au sein d'un front vertical comme retrouvé lors de la campagne océanographique PROTEVSMED-SWOT (2018), nous avons également simulé l'évolution de la proportion de mélange de deux

masses d'eau aux propriétés singulières. Pour ce faire, nous avons défini deux jeux de données positionnés selon les  $y$  positifs pour le premier et selon les  $y$  négatifs pour le second plaçant ainsi leur rencontre à l'origine de l'axe  $y$ . Ici, nous avons fait l'hypothèse de l'absence d'un courant advectant les masses d'eau afin de déterminer l'évolution de la proportion de mélange des deux masses d'eau au cours du temps et de connaître l'instant  $t$  où le milieu peut être considéré comme homogène.

Enfin, afin de caractériser au mieux l'évolution spatiotemporelle du front Nord-Baléares, nous avons implémenté des données *in situ* issues de la campagne BioSWOT-Med dans le programme théorique d'advection-diffusion. Pour ce faire, nous avons utilisé un fichier netCDF comportant des mesures réalisées le 5 mai 2023 et suivant la convention standard "Climate and Forecast" de la version 1.7. Puis nous avons extrait les paramètres de vitesse géostrophique zonale et méridienne en mètres par seconde ainsi que les coordonnées de latitude et longitude respectivement en degré nord et degré est. Les fichiers utilisés comportent un volume illimité avec des données en 64-Bit, ce qui a nécessité l'utilisation de commandes spécifiques pour implémenter correctement ces données tout en n'excédant pas l'espace de stockage des ordinateurs.

Nous avons également simulé une densité de phytoplancton suivant un processus mort/naissance influencé par l'évolution de sa propre population ainsi que celle du zooplancton agissant comme prédateur de cette espèce. Concernant la concentration en chlorophylle  $a$ , nous avons considéré ce paramètre comme non conservatif avec l'introduction d'un terme de "grazing" pour simuler un taux de prédation par le zooplancton au cours du temps. Nous avons fait varier ce paramètre au cours des simulations afin d'appréhender les différentes projections dans lesquelles un bloom phytoplanctonique évoluerait.

# III - Résultats et Discussion

## 1. Diffusion

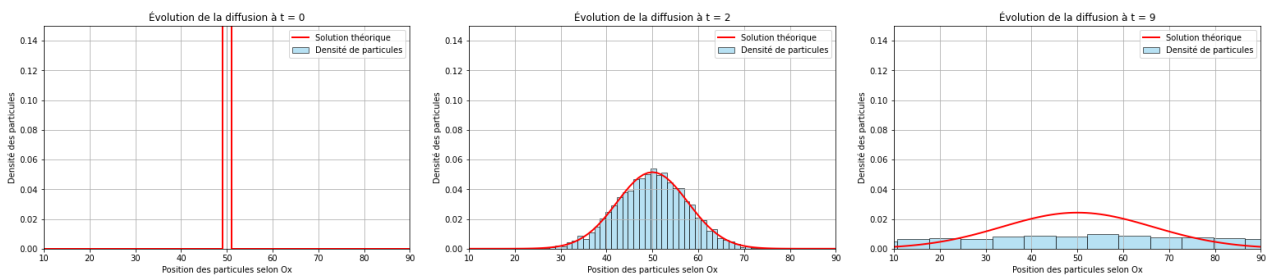
### a. Comparaison Solution théorique et Modèle réel

Le transport diffusif au sein du front Nord-Baléares a été représenté ici par un modèle simple d'abord en 1D puis en 2D à l'aide d'un jeu de données suivant un mouvement aléatoire défini par une loi uniforme. La solution théorique de ce modèle a pour image une gaussienne centrée qui agit comme modèle idéal de la distribution. De ce fait, Il a été nécessaire de définir certains paramètres comme la position initiale de nos particules que l'on place au centre de notre grille, la taille de notre grille de mesure que l'on définit de 100 unités selon x et y ainsi que le coefficient de diffusion de notre gaussienne qui doit être proportionnel à l'évolution de la concentration de nos particules dans le temps, ici de 15.

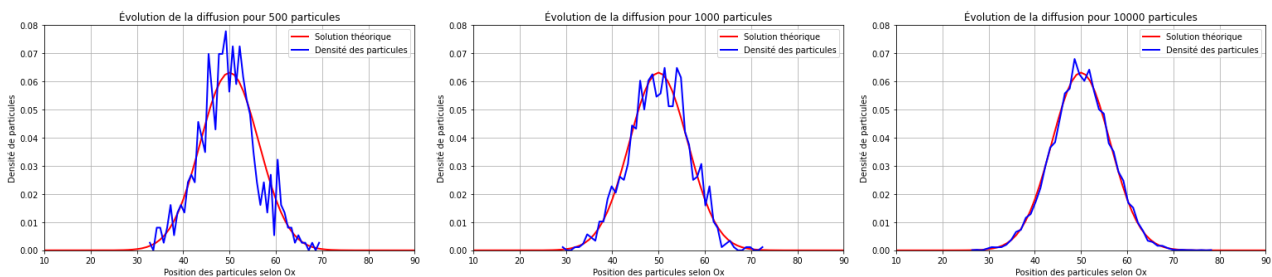
#### Cas d'une solution à 1 Dimension (selon Ox) :

D'après la figure 1, on peut voir que la position des particules au début de la simulation (à  $t = 0$ ) est confondue avec le schéma théorique de diffusion avec une valeur de densité à 1 indiquant que la concentration totale se situe exactement au centre de notre grille d'observation. Puis, au bout de 2 pas de temps, notre concentration initiale diminue au centre de notre grille et s'étend de part et d'autre de son centre atteignant une position minimale de 35 à gauche et maximale de 65 à droite. Le taux de concentration en particules étant toujours plus élevé au centre et faible aux extrémités de l'histogramme. On observe ici que la diffusion réelle reste conforme au schéma de diffusion théorique. Enfin, à la fin de notre simulation (à  $t = 9$ ) la concentration de départ a été divisée par un facteur de 3 et s'est étendue de 90 unités d'espace en théorie et de seulement 60 unités en pratique. Cette première simulation nous indique que le transport diffusif suit bien une distribution gaussienne en une dimension où la concentration de particules y est plus importante au centre et que le pas de temps choisi permet largement d'apprécier le phénomène. Cependant on peut constater que la gaussienne théorique a une distribution bien plus large que l'histogramme des concentrations, ce qui pourrait expliquer la marge d'erreur observée entre les modèles théoriques et les mesures de terrain.

Dans un second modèle, nous avons représenté l’histogramme du jeu de données sous forme de courbe dont chaque point correspond au nombre de particules dénombrées par pas d’espace. Nous pouvons observer que plus le jeu de données comporte de particules et plus la densité observée est lissée en suivant le schéma de diffusion théorique à l’identique, ici les densités concordent pour un jeu de données de 10 000 particules (figure 2). Ainsi, cette simulation à un pas de temps donné nous permet de comprendre l’influence du poids du jeu de données sur la résolution du modèle de diffusion de particules. Il sera donc important de réaliser les simulations du modèle d’advection-diffusion avec des jeux de données suffisamment importants pour limiter le bruit des échantillons afin d’obtenir des résultats significatifs qui puissent être interprétés.



**Figure 1 : Histogramme de l’évolution de la densité groupée de particules en fonction du temps et de la gaussienne théorique associée (cf. Annexes I)**



**Figure 2 : Evolution de la concentration à 20% du temps de simulation en comparant le modèle théorique avec un jeu de donnée de 500, 1 000 et 10 000 particules (cf. Annexes II)**

**Cas d’une solution à 2 Dimensions (Ox, Oy) :**

Pour cette seconde simulation on a fait le choix de représenter le transport diffusif en 2D sous forme de carte de concentration où la couleur violette indique un milieu non concentré en particules et la couleur jaune correspond à un milieu très concentré. À l’inverse de la précédente simulation on s’est focalisé sur la largeur de la gaussienne théorique comme indicateur d’étalement de la diffusion et non sur son sommet pour pouvoir interpréter les

résultats suivants, voir figure 3. Ainsi à l’instant initial, les concentrations théoriques et réelles sont confondues en un point rouge au centre de la grille d’étude. Puis la diffusion des particules dans le milieu a été réalisée suivant le même principe que le modèle en 1D et l’on peut constater qu’à l’inverse de celle-ci, la gaussienne théorique simule un taux de diffusion légèrement inférieur à la diffusion réelle. Ces deux premiers modèles nous indiquent comment évolue un taux de diffusion d’un jeu de données dans un milieu homogène. Cependant, il devra être adapté au jeu de données étudié puisque certains paramètres peuvent influencer la diffusion des particules comme la masse volumique du milieu étudié (ici la mer Méditerranée à une masse volumique moyenne de  $1025 \text{ kg.m}^{-3}$ ) ou encore la profondeur des données *in situ* où la densité des masses ainsi que les régimes de circulation peuvent varier. Ce dernier point sera à prendre en compte pour l’étude avec prise en compte du transport advectif.

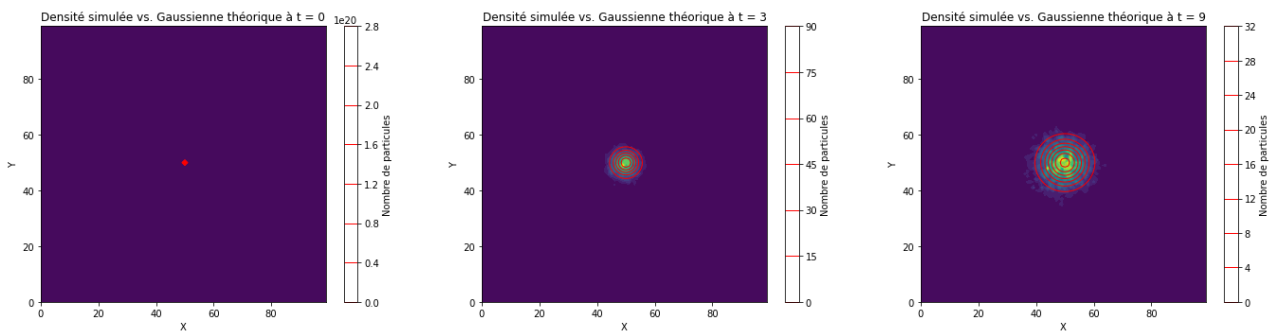


Figure 3 : Évolution en 2D de la densité groupée de particules en fonction du temps et de la gaussienne théorique associée (cf. Annexes III)

## b. Ajout de conditions de bord

Dans le but de simuler un modèle de diffusion réaliste, nous avons établi des conditions au bord de notre grille d’observation agissant comme un “phénomène de perte” des particules. Pour y parvenir nous avons utilisé différentes méthodes afin de sélectionner la plus adaptée à notre modèle d’abord en 1D puis en 2D. Dans un premier temps, nous avons utilisé une méthode assignant à toute particule externe aux bornes de définition du modèle une valeur aberrante ou un Nan (Not a number). Puis nous avons stocké ces valeurs dans une matrice non rappelée lors du dénombrement des particules ce qui a permis de l’isoler de la matrice de concentrations et ainsi comptabiliser et afficher uniquement les particules encore présentes au sein du modèle, (voir figure 4). En effet, on observe qu’au bout de trois pas de temps, une particule est sortie de la simulation et le taux de particule décroît lentement

jusqu'à la fin de la simulation. Ici 398 particules sont sorties de notre grille d'observation au bout de 10 pas de temps avec un coefficient de diffusion fixé à 15. C'est ce coefficient de diffusion qui détermine la vitesse de sortie des particules : plus celui-ci est élevé et plus le taux de particule va chuter rapidement au cours du temps.

Par ailleurs, il est important de prendre en considération les biais de simulation. Ici lorsqu'on définit les conditions d'observation, il est nécessaire d'utiliser la condition "or" et non "and" dans python afin d'indiquer à la console que si une particule sort du modèle selon l'axe des x ou des y elle ne sera alors plus comptabilisée. L'ajout de cette condition permet d'éviter qu'une particule soit dénombrée alors qu'elle n'est plus localisée dans l'une des deux grilles d'observation. De plus, c'est normal si notre nombre de particules ne diminue pas au premier temps de simulation et c'est même bon signe car cela signifie qu'on a choisi un bon domaine d'observation (si le domaine est trop petit : notre nombre de particules diminue trop vite). Les limites de ce modèle sont notamment l'adaptation de celui-ci à un jeu de donnée *in situ* puisque trop de paramètres sont fixés par l'utilisateur. En effet, il est possible que certaines données soient endommagées ou non définies lors de mesures sur le terrain ce qu'il est difficile de définir comme condition limite dans le cas de notre modèle où il faudrait les définir manuellement, rendant notre modèle très peu ergonomique.

Dans un second temps, nous avons simulé des conditions de bord avec un paramètre se basant uniquement sur les dimensions du domaine d'observation définies au préalable par l'utilisateur ou indiquée par des données *in situ* auquel est ajouté un pas d'espace. Lorsque ce paramètre est respecté, les valeurs de position des particules sont alors rejetées de la matrice de concentrations. Cette méthode étant la plus simple à implémenter dans nos programmes, nous avons choisi de l'utiliser dans des modèles 1D, 2D ainsi qu'avec les jeux de données issus de la campagne BioSWOT-Med 2023. En effet, elle fonctionne efficacement pour des modèles en 2D où le dénombrement des particules s'effectue suivant les mêmes conditions de définition du modèle.

Nous avons également affiché les calculs de densité moyenne de particule par cellule de comptage qui diminue lentement au cours du temps suivant le nombre de particules encore présentes dans la zone d'étude. De plus, en calculant l'écart type à chaque pas de temps, on constate que les données sont centrées autour de la moyenne en début de simulation (écart-type = 0), puis au bout de 10 pas de temps, celui-ci augmente et tend vers une valeur de 20

pour la simulation en 1D et 60 en 2D. Cela reflète ainsi l'importante distribution des particules autour de la moyenne à la fin de la simulation. L'écart type est influencé par la taille du jeu de données avec un écart faible à la moyenne associé à un ensemble de données important. Cela nous permet également d'évaluer le bon nombre de particules à associer à notre jeu de données. Pour avoir des résultats significatifs, un jeu de données de 5 000 particules est suffisant en 1D puisqu'au bout de 10 pas de temps, seulement 20% de notre jeu de données est dispersé autour de la moyenne fixée comme le centre de notre grille. Mais avec le même coefficient de diffusion, il y a 60% de particules qui sont dispersées autour de la moyenne pour la simulation en 2D au bout de 10 pas de temps. Ce résultat élevé nous indique qu'il sera important d'utiliser un plus grand nombre de particules et de sélectionner un coefficient de diffusion plus faible (d'un ordre de grandeur 10 à 1000 fois inférieur à celui actuel) afin de tendre vers davantage de réalisme. En effet, il est également important de prendre en considération le coefficient de diffusion qui influe largement sur la dispersion de notre jeu de données et donc sur l'écart-type des données par rapport à la moyenne.

De ce fait d'après la première loi phénoménologique de Fick (A. Fick, 1855), les particules se diffusent des zones de forte concentration vers les zones de faible concentration : il existe donc une relation linéaire entre le flux de matière et le gradient de concentration de celle-ci tel que  $\mathbf{j}_j = -D_{ij} \nabla C_j$  où  $D_{ij}$  représente le coefficient de diffusion ( $m^2.s^{-1}$ ),  $\mathbf{j}_j = \frac{J_j}{M_j}$  : flux molaire ( $mol m^{-2} s^{-1}$ ) et  $C_j = \frac{\rho C_j}{M_j}$  : concentration molaire ( $mol m^{-3}$ ) avec  $M_j$  la masse molaire ( $Kg.mol^{-1}$ ). Cette équation est valable pour un milieu incompressible et tend à atteindre un état stationnaire où le nombre de particules par unité de volume sera identique pour l'ensemble de la grille d'étude ou du fluide considéré au bout d'un nombre de pas de temps important. Ici, elle nous permet de définir un coefficient de diffusion conforme à notre grille d'étude en  $m^2.s^{-1}$ .

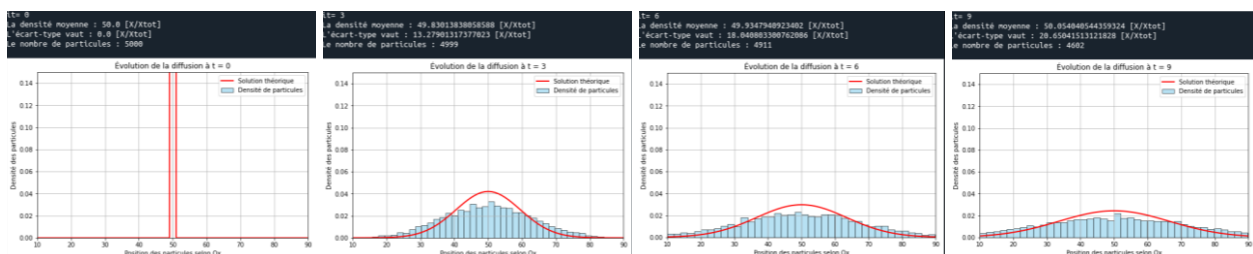


Figure 4 : Modèle de Diffusion 1D avec ajout de conditions de bord, méthode de définition du domaine (cf. Annexes IV)

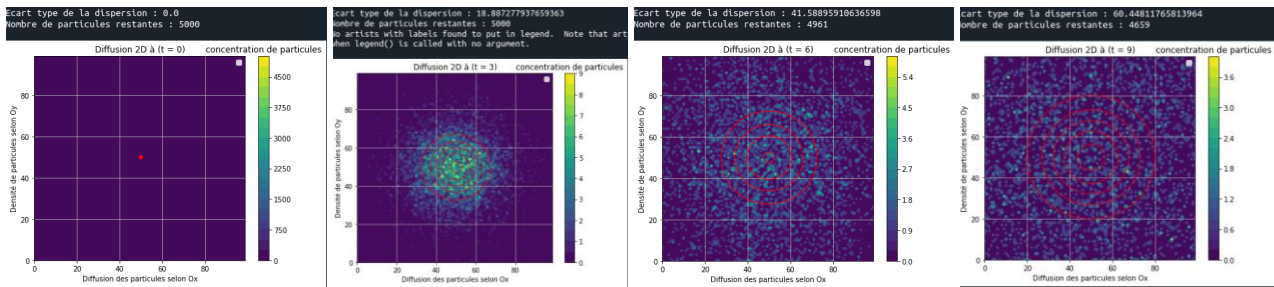


Figure 5 : Modèle de Diffusion 2D avec conditions de bord, même méthode (cf. Annexes IV)

## 2. Advection et Diffusion

Durant la seconde partie de ce stage, nous avons travaillé sur la modélisation de l'équation d'advection du modèle d'advection-diffusion afin de simuler la diffusion de particules au sein d'un champ de vitesse d'abord uniforme, puis de réaliser cette simulation au sein d'un jet gaussien, modèle idéalisé du front Nord-Baléares. Pour réaliser ce modèle il a été nécessaire de vérifier la condition de Courant Friedrichs-Lewy (CFL) qui permet de s'assurer qu'une particule de fluide ne parcourt pas plus d'un pas d'espace par pas de temps. Cette condition s'énonce comme suit : 
$$CFL = \frac{u \cdot \Delta t}{\Delta x} \leq C_{max}$$
 où  $u$  est la composante zonale de la vitesse ( $m \cdot s^{-1}$ ) et  $C_{max}$  le coefficient de définition limite qu'on prendra ici égal à 1. Dans le cadre de nos simulations on a posé le CFL comme toujours inférieur ou égal à 1 afin de faciliter le bon déroulement du stage étant donné que la recherche de la validité de cette condition était peu intéressante dans le cadre du modèle de simulation lagrangienne.

### a. Action d'un courant uniforme selon Ox

Par la suite, nous avons choisi d'appliquer un courant uniforme à notre modèle de diffusion afin d'évaluer l'étendue de la dispersion lorsque nos paramètres sont soumis à un champ de vitesse constant. Ainsi, nous avons d'abord initialisé les composantes de vitesse zonale et méridienne théoriques puis nous les avons associées à notre modèle de diffusion. Le champ de vitesse 2D dépend de conditions initiales définies par une matrice remplie de 1 de la taille de la grille d'étude où l'on dissocie les distances selon  $x$  et  $y$  afin de modéliser un bassin rectangulaire pour gagner en réalisme. Par la suite, on associe à cette matrice la norme de la vitesse souhaitée pour créer un champ de vitesse constant. Puis on définit des indices de temps et d'espace dans cette matrice afin de la faire évoluer le long de la grille

d'étude. Une fois les deux composantes définies, elles sont implémentées dans le modèle d'advection-diffusion selon le principe suivant : à chaque pas de temps, la position de notre particule dépend de sa position au pas de temps précédent ainsi que de la vitesse associée à ce pas de temps additionnée à un déplacement aléatoire suivant une loi normale de paramètre  $\mu$  et  $\sigma$ . (où  $\mu$  est la moyenne de la distribution centrée sur 0 et  $\sigma$  l'écart-type défini à 1). Enfin on injecte l'ensemble des paramètres de diffusion et d'advection dans une boucle spatio-temporelle qui détermine la position de nos particules au pas de temps suivant.

Afin de tester notre modèle d'advection-diffusion nous avons réalisé trois tests au sein d'un front séparant deux masses d'eau chacune advectée dans un sens contraire. De ce fait, nous avons fait l'hypothèse que l'écoulement s'effectuait selon la composante zonale (positive vers l'est sur la partie sud de la grille d'observation et négative sur la seconde partie) et que la vitesse méridienne du fluide était nulle, voir figure 6. Ainsi, on regarde comment évolue la diffusion au sein de chaque masse d'eau et du front horizontal. De ce fait, on s'attend à ce que la tâche de diffusion de la masse d'eau au sud soit advectée selon les  $O_x > 0$ , que celle advectée dans la masse d'eau au nord soit advectée selon les  $O_x < 0$  et qu'au sein du front la tâche de diffusion soit étirée en filaments selon les  $O_x > 0$  et  $O_x < 0$ . D'après les figures 7 et 8, la diffusion des particules s'effectue bien selon les hypothèses énoncées et la diffusion au sein du front tend à s'amincir et former une fine couche séparant les deux masses d'eau. Si l'on regarde maintenant l'évolution de la densité de particules avec la figure 9, on observe que celle-ci diminue de façon exponentielle au cours du temps avant d'atteindre une densité de 67,5 particules/cellule de dénombrement pour  $U_1$  et  $U_2$  et une densité moins élevée de 16,8 particules/cellule de dénombrement pour  $U_3$ . Cet écart de concentration est notamment dû au caractère davantage diffusif des fronts où l'advection plus importante (cas d'un jet gaussien) ou l'inversion des champs de vitesse avec au centre une vitesse nulle (cas d'un front horizontal) montre le rôle amplificateur de l'advection dans les phénomènes de diffusion. Enfin, on constate que l'ajout de conditions de bord simulant la zone de mesures satellites ou l'échantillonnage *in situ* agit bien comme un phénomène de zoom sur une zone géographique donnée en ne comptabilisant plus les données exclues de cette zone. En effet, dans la figure 10, si la zone d'étude est centrée sur le quart ouest de la

zone géographique définie, les particules diffusées et advectées à l'extérieur de la zone ne seront plus visibles et leur nombre diminuera au cours du temps.

Nous avons également modélisé la rencontre de deux masses d'eau au sein d'un front vertical où la composante zonale de la vitesse devient alors nulle et la composante méridienne est positive vers le nord sur la partie ouest de la simulation et négative sur la partie est, voir figure 11. Cette simulation nous confirme le caractère diffusif joué par les fronts séparant distinctement deux masses d'eau.

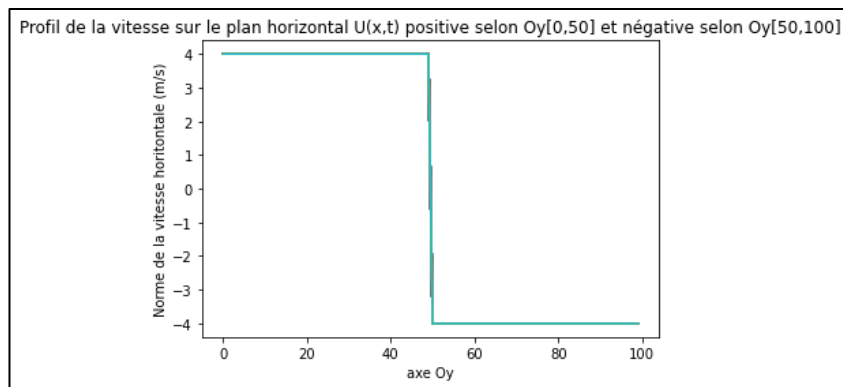


Figure 6 : Profil de la vitesse au niveau du front horizontal de rencontre de deux masses d'eau

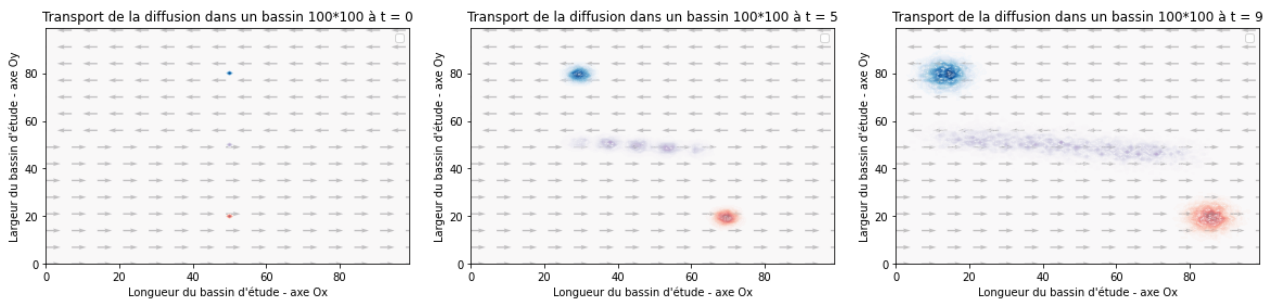


Figure 7 : Evolution de trois concentrations de 5000 particules selon leur position dans le modèle d'Advection-Diffusion au sein d'un front horizontal avec un champ de vitesse constant (cf. Annexe VI)

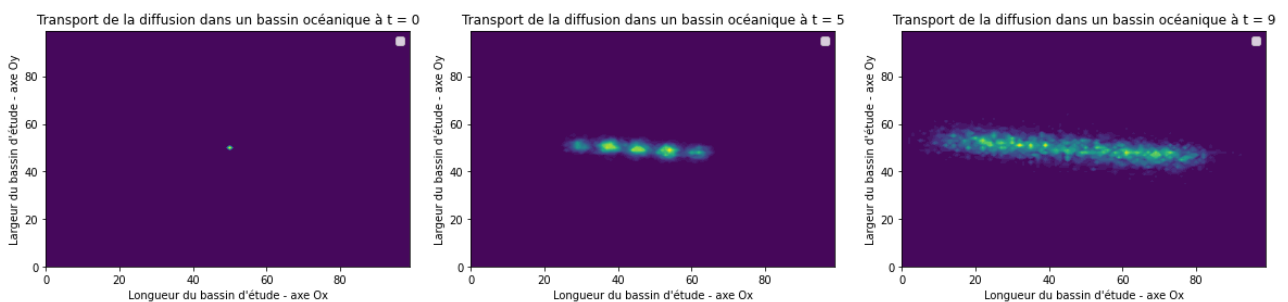


Figure 8 : Evolution de la concentration d'un jeu de donnée de 5000 particules à l'interface de deux courants évoluant avec un champ de vitesse de norme positive sur la première moitié de l'axe  $O_y$  et négative sur la seconde partie

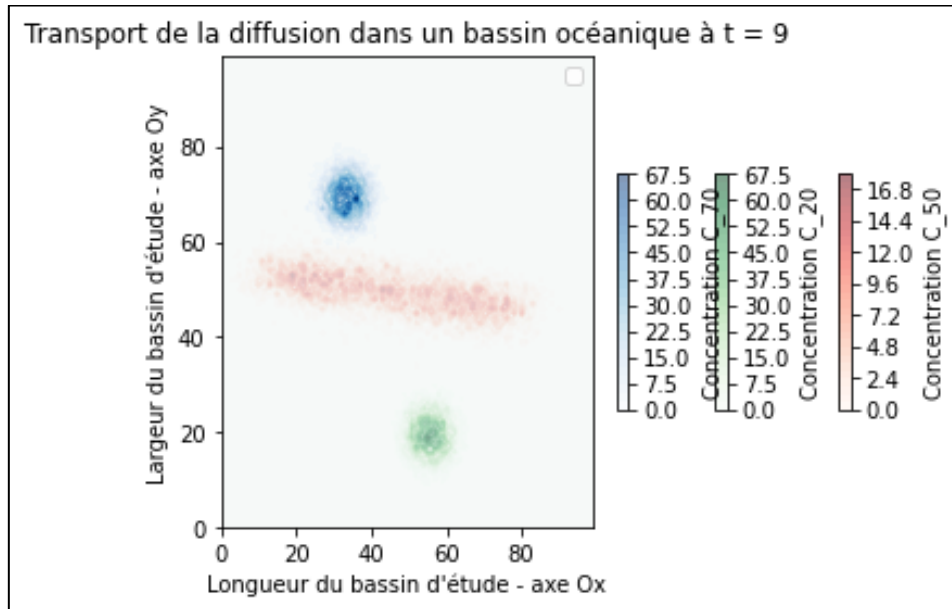


Figure 9 : Évolution de la concentration à 90% de notre temps de simulation d'un jeu de donnée de 5 000 particules selon leur position au sein d'un front horizontal de 100\*100 où  $U_1(x,t)$  varie selon  $O_x > 0$  en vert,  $U_2(x,t)$  varie selon  $O_x < 0$  en bleu et  $U_3(x,t)$  varie selon  $O_x > 0$  et  $O_x < 0$  en rouge

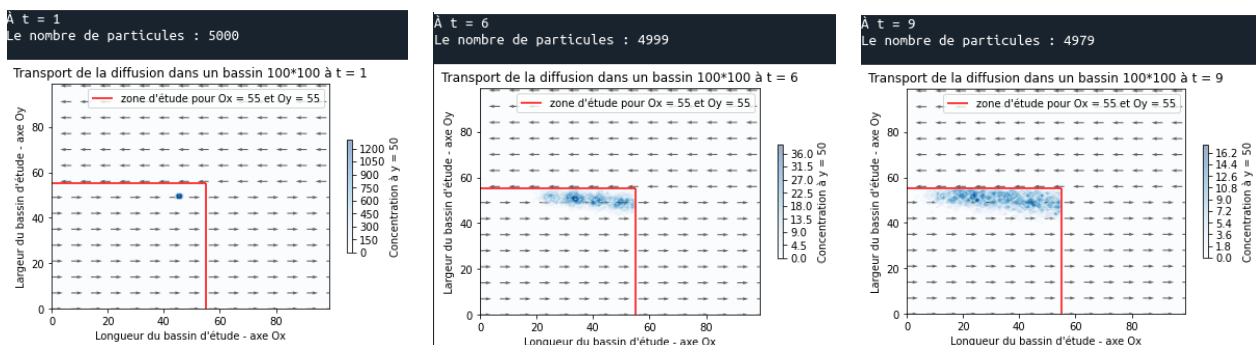


Figure 10 : Evolution de la concentration d'un jeu de donnée de 5000 particules à l'interface de deux courants évoluant en norme de sens opposée avec ajout de conditions de bord (cf. Annexe VI)

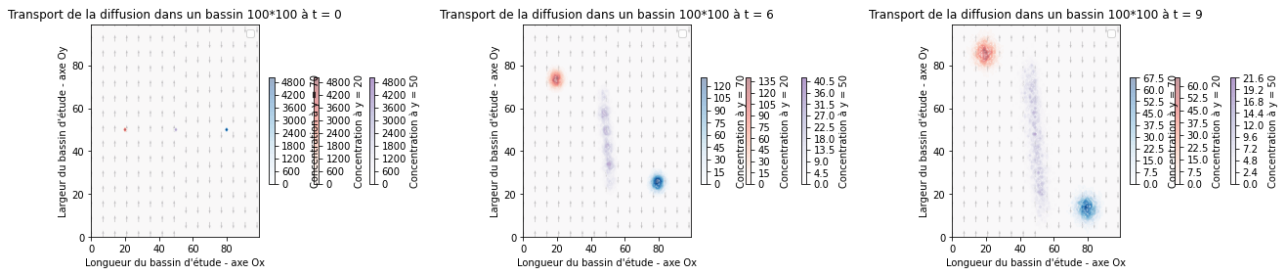


Figure 11 : Evolution de trois concentrations de 5000 particules selon leur position dans le modèle d'Advection-Diffusion au sein d'un front vertical à  $O_x = 50$  avec un champ de vitesses homogène

### b. Action d'un jet gaussien

D'autre part, nous avons modélisé un jet gaussien afin de caractériser de manière théorique l'évolution de la diffusion au sein du front Nord-Baléares. La différence ici entre la simulation d'un courant homogène et celle d'un jet gaussien réside dans l'initialisation des paramètres d'advection qui ne seront plus considérés comme constants mais suivant les paramètres d'une loi uniforme. Pour ce faire, nous avons associé une fonction simulant une gaussienne aux indices d'une matrice de la taille de la zone d'étude considérée. Nous avons ainsi obtenu le profil d'un jet gaussien comportant un champ de vitesse de norme importante au centre de la gaussienne et décroissant aux limites de celle-ci, voir figure 12.

Par la suite, nous avons simulé différentes positions du patch de diffusion afin d'appréhender le comportement d'un patch de phytoplanctons en l'absence de grazing au sein du front Nord-Baléares de manière théorique. D'après la figure 13, on observe que la diffusion des particules s'effectue en tâche circulaire au centre de la gaussienne et tend à prendre une forme allongée au nord et au sud de son pic d'intensité. En effet, la déformation du patch de diffusion peut s'expliquer par la tendance des particules à être davantage advectées au centre ce qui leur donne cette forme ovale. D'autre part, lorsqu'on regarde la barre de couleur sur la gauche du graphe indiquant la concentration par pas d'espace au cours du temps, on constate que sous l'influence du phénomène de diffusion les 3 patchs sont bien plus concentrés en leur centre. Cependant, les deux patchs situés sur la pente nord et sud de la gaussienne ont un patch excentré vers l'avant de la gaussienne laissant derrière lui une trainée de particules assemblées en filament moins concentré que l'on peut observer au bout d'un pas de temps supérieur à 20. De ce fait, les différentes concentrations de particules suivent bien le profil de la gaussienne et leur placement au sein de cette courbe influe grandement sur la forme de leur

distribution. Cela nous laisse penser qu'il existe une importante diversité de profils planctoniques auxquels s'additionne la turbulence des eaux de surface induite par le vent dessinant ainsi des petits amas de plancton concentrés au sein de tourbillons ou advectés davantage par le jet gaussien au sein du front Nord-Baléares.

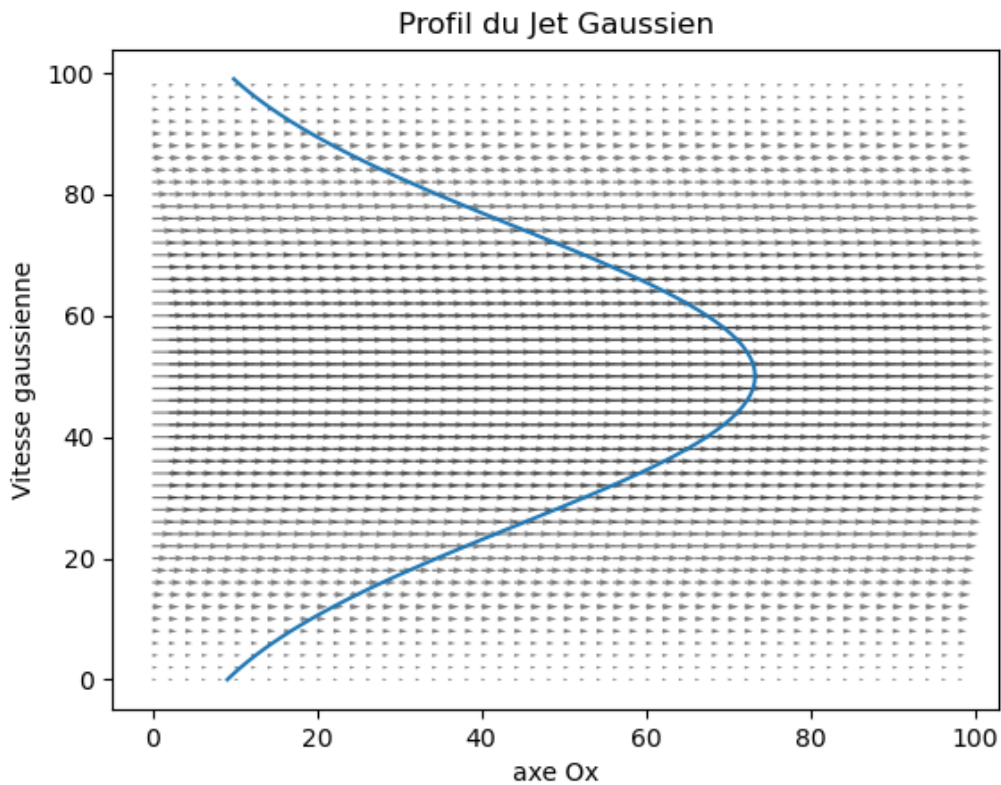


Figure 12 : Profil d'un champ de vitesse sous la forme d'un jet gaussien se propageant vers l'est suivant les Ox positifs (cf. Annexe VII)

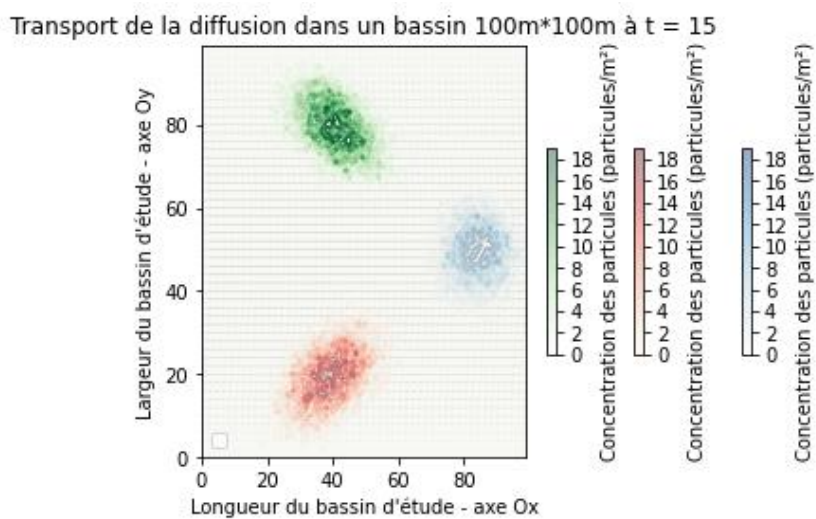


Figure 13 : Evolution de trois concentrations de 5000 particules selon leur position dans le modèle d'Advection-Diffusion au bout de 15 pas de temps au sein d'un jet gaussien avec un pic à  $O_x = 50$  (cf. Annexe VII)

### 3. Tests de mélange des masses d'eau

Afin de tester maintenant la proportion de mélange de deux masses d'eau au sein du front Nord Baléares à l'aide du modèle d'advection-diffusion, nous avons d'abord fait l'hypothèse d'une absence d'advection afin de déterminer l'évolution de la proportion de la couche de mélange au cours du temps. Puis nous avons soumis notre modèle à un courant à jet qui tend à simuler le front Nord-Baléares.

#### a. Absence d'advection

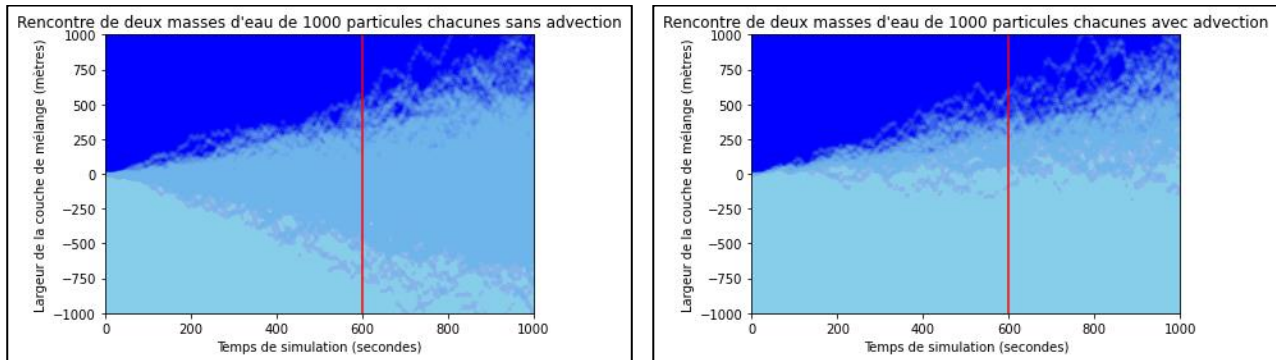
La réalisation de cette première simulation en l'absence de courant nous a permis de caractériser l'évolution spatiotemporelle de l'interface entre deux masses d'eau de caractéristiques différentes. Ainsi, initialement les deux masses d'eau sont placées aux extrémités de la zone d'observation et elles n'échangent aucune molécule entre elles. Puis au cours de la simulation on va observer la formation d'une couche de mélange issue d'une proportion non égale de chaque masse d'eau. Cette inégalité est modélisée par le déplacement aléatoire de chaque particule suivant un mouvement brownien qui tend à simuler une diffusion turbulente associée aux fluides réels. En faisant l'hypothèse que le milieu considéré est homogène et que les forces externes de pression et de températures agissant sur nos particules sont négligeables, notre diffusion obéit à la Loi de Fick comme énoncée précédemment :  $\frac{\partial C_i}{\partial t} + \nabla \cdot \mathbf{j}_i = 0$  (Fick A., 1855). Où le premier terme représente la dérivée spatiale de la concentration molaire et le second terme la divergence du flux de diffusion molaire. Etant donné que les particules se déplacent depuis les fortes vers les faibles concentrations, on observe un modèle de diffusion comme présenté figure 14. Ainsi, on peut observer qu'au bout de 10 minutes ( $t = 600$ ), chaque masse d'eau est constituée d'une moitié de particules d'origine et d'une autre moitié chargée en particules issues des deux masses d'eau dans des proportions similaires. Cette proportion de mélange dépend du coefficient de diffusion établi par le simulateur. Ici le coefficient choisi est très faible, ce qui

nous permet de calculer la vitesse de diffusion des particules des deux masses d'eau de façon arbitraire. En effet, d'après le graphique, la masse d'eau bleu clair se diffuse à une vitesse de  $0,87 \text{ m.s}^{-1}$  moyennée sur 10 simulations tandis que les particules de la seconde masse d'eau se diffusent à une vitesse de  $0,99 \text{ m.s}^{-1}$ . On constate que la vitesse moyenne de diffusion des deux masses d'eau est proche de  $1 \text{ c m.s}^{-1}$  mais les particules des jeux de données ont une vitesse différente. Cet écart de vitesse peut être lié à une sous-estimation de la densité de particule dû à la non prise en compte des valeurs aberrantes mais également au nombre insuffisant de particules par jeu de données ou encore à la vitesse moyennée sur un nombre de simulation trop faible. Enfin, au bout de 16 min 40 s on observe que la couche de mélange occupe l'ensemble des deux masses d'eau et qu'elles se sont plus distinctes à l'œil nu, reflétant ainsi une diffusion totale des particules dans la zone d'étude à la fin de la simulation.

#### b. Influence d'un jet gaussien

Lors de la seconde simulation, nous avons souhaité prendre en compte l'influence de l'advection et donc d'un champ de vitesse suivant la définition d'un jet gaussien dans le but d'appréhender la proportion de mélange de deux masses d'eau au sein du front Nord-Baléares. Pour cette simulation nous avons implémenté les termes de champ de vitesse gaussien dans le modèle en reformatant la matrice contenant les coordonnées de vitesse afin de l'adapter aux limites de définition du modèle. On s'attend ici à obtenir un mélange des deux masses d'eau moins diffusif avec un front davantage étiré. En effet, au bout de 10 minutes (600 pas de temps), la taille de la couche d'eau contenant des particules issues des deux masses d'eau est 1,48 fois moins importante que celle donnée par le modèle sans advection (1180 m sans advection contre 796 m sous l'influence d'un jet gaussien). D'autre part, à la fin de la simulation seulement 58% de la largeur de la zone d'étude est occupée par la couche de mélange ce qui indique que la couche de mélange diminue de 42% sous l'influence des paramètres de vitesse pour les mêmes temps de simulation. Nous pouvons également faire la remarque que la masse d'eau des Oy positifs (représentée en bleu foncé sur la simulation) a une diffusion de l'ordre d'une centaine de mètres qui est donc assez faible proportionnellement à la largeur du bassin océanique simulé. Ainsi, ces deux simulations représentant la proportion de mélange des deux masses d'eau à chaque pas de temps nous permettent de confirmer l'impact d'un jet gaussien plus intense au centre de ce front qui aura

pour effet d'affiner la couche de mélange en inhibant la diffusion à leur rencontre. Un tel phénomène sera donc à prendre en compte lors de nos simulations futures avec l'utilisation de paramètres *in situ*.



a. Absence d'advection

b. Influence d'un Jet Gaussien

**Figure 14 : Evolution de la rencontre de deux masses d'eau situées au Nord et au Sud d'un front théorique horizontal avec un coefficient de diffusion faible de l'ordre de 0.1 (cf. Annexe VIII)**

#### 4. Utilisation de paramètres *in situ* issus de la campagne BioSWOT-Med

Dans cette dernière section, nous avons travaillé avec des données *in situ* réalisées le 5 mai 2023 lors de la campagne océanographique BioSWOT-Med. L'utilisation de relevés de terrain vont notamment nous permettre de tester la validité de nos modèles en vérifiant si les trajectoires réalisées par nos patchs de concentration connues évoluent selon les prédictions d'advection et de diffusion déterminées par nos modèles. Par ailleurs, nous pourrions également déterminer les limites de nos modèles et les solutions à apporter pour les adapter aux mesures de terrain.

Pour utiliser les données *in situ*, nous avons réalisé un travail de vérification des grandeurs utilisées en s'assurant qu'elles correspondaient bien aux paramètres énoncés dans les équations d'advection-diffusion. Puis après avoir associé chaque grandeur à chaque paramètre, nous avons vérifié la correspondance entre les unités des modèles théoriques et celles du fichier netCDF en s'assurant qu'elles obéissaient bien aux unités du système international. Après ce travail au préalable, nous avons extrait les données dont dépendent nos modèles théoriques en utilisant les bibliothèques python "xarray" et "netCDF4". Puis nous avons tracé les premiers profils de vitesse géostrophique associés aux coordonnées

géographiques grâce aux bibliothèques “matplotlib” et “cartopy” tout en s’assurant de la bonne association des données entre elles grâce à la bibliothèque “Pandas”. Etant donné mon manque d’expérience dans l’utilisation de ce type de fichier, je m’attendais à obtenir des profils similaires à ceux publiés dans les travaux réalisés sur la campagne BioSWOT-Med, voir figure 15. Cela a été confirmé par mon Maître de stage qui a sûrement m’aiguillé pertinemment lors de l’utilisation de ce type de données. Ainsi, le profil de vitesses géostrophiques obtenu figure 16 nous permet de visualiser l’ensemble du champ de vitesse de surface entre Toulon (43,08’N/ 5,56’E) et Alger (36°42’N/ 3°05’E) sur deux bandes de 60 km de large séparées d’une bande nadir de 20 km. C’est à partir de ce champ de vitesse que nous avons réalisé toutes les simulations de nos patches de diffusion entrés d’abord au sein du front Nord Baléares, puis sur des structures éphémères dans le bassin du Golfe du Lion.

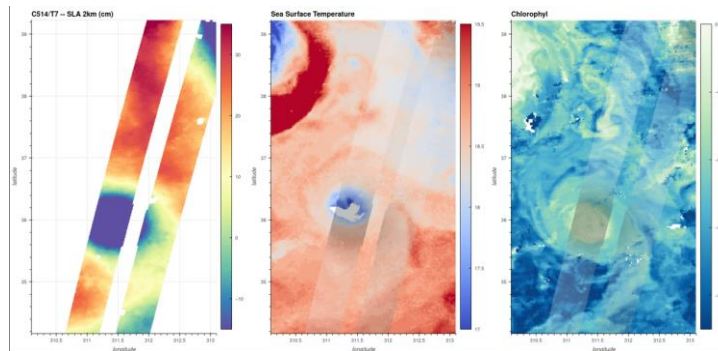


Figure 15 : Exemple de 3 cartographies d’anomalie de la hauteur de la surface de la mer, de la température de surface et de chlorophylle a (de gauche à droite) avec une résolution de 2 km (BioSWOT-Med, 2023)

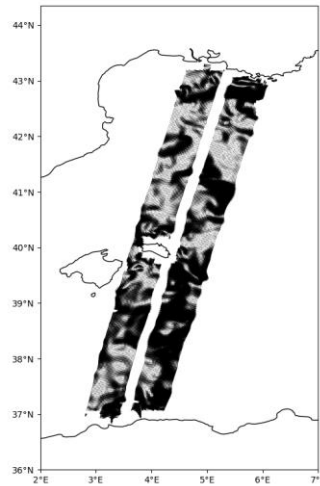


Figure 16 : Profil des composantes zonales et méridiennes de la vitesse géostrophique au sein de la mer Méditerranée le long du transect réalisé par le satellite SWOT (cf. Annexes IX)

#### a. Diffusion au sein du front Nord-Baléares

L'identification de ce front a été permise grâce aux travaux réalisés lors de la campagne BloSWOT-Med (2023). De ce fait, le front Nord-Baléares est localisé entre 40,5°N et 41,5°N de latitude (A. Petrenko), voir figure 17. Pour la suite des simulations, nous avons utilisé une méthode simple d'interpolation consistant à assimiler la partie ouest des données à la latitude la plus proche et nous avons fait de même pour la partie est des données. Cela nous a permis de simuler les différentes trajectoires sans se soucier de l'arrêt du programme dû à l'absence de champ de vitesse entre les deux bandes réalisées par le satellite SWOT. Pour adapter notre modèle de diffusion au champ de vitesse *in situ* nous avons testé différentes valeurs de coefficient de diffusion représentés figure 18. On observe que la diminution du coefficient de diffusion est proportionnelle à l'augmentation de la taille de la tâche de particules. Souhaitant modéliser un patch de diffusion sur plusieurs jours, nous avons sélectionné un faible coefficient de diffusion pour qu'il soit davantage soumis à l'advection et que l'on puisse mesurer la tâche à la fin de la simulation. En effet, on constate qu'au bout de 4h de simulation, la concentration en particules est trop faible et on distingue peu les particules sur le fond de carte "viridis" avec un coefficient de 0.5. Tandis que les différentes échelles de concentration sont davantage visibles avec un coefficient deux fois moins élevé. Cependant, ce sera le coefficient de diffusion d'un millièème qui sera sélectionné pour la suite de nos simulations étant donné sa capacité à être davantage soumis à l'advection.

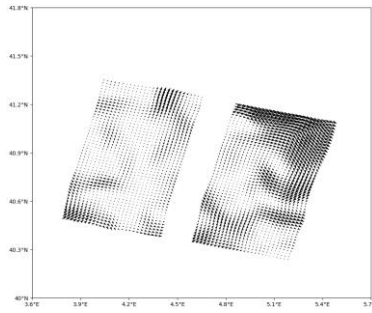


Figure 17 : Profil des vitesses géostrophiques de surface au sein du front Nord-Baléares

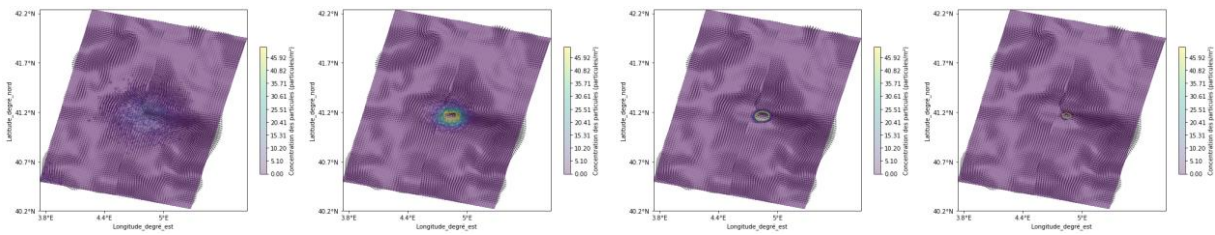


Figure 18 : Evolution d'un patch de 5000 particules à 41,2°N/ 4,8°E au bout de 4h10min et 26s sous l'influence de différents coefficients de diffusion (D) de gauche à droite à D=0.5, D=0.1, D= 0.01 et D=0.001

Dans un second temps, nous avons comparé les travaux sur des données *in situ* avec les données obtenues par le satellite SWOT. Pour ce faire, nous avons réalisé la simulation d'un patch de particules initialisé à 43.83°N/ 4.93°E pendant 4 jours qui correspond à la durée de collecte des données par les bouées ; puis nous avons comparé les profils obtenus. On observe que les particules n'ont pas été advectées au bout de 21 heure de simulation mais que la taille du patch a doublé dû à la diffusion des particules, d'après la figure 19. Puis au bout d'un jour et 9 heures, les particules sont davantage advectées ce qui donne une forme ovale au patch de diffusion. Enfin, au bout de 3 jours et 11 heures de simulation, le patch est davantage soumis à la diffusion avec un patch plus large et peu étiré dû à une faible advection. Au bout de 4 jours, notre patch se retrouve donc à quelques kilomètres de sa position d'origine et on observe que d'après la simulation il a été piégé au sein d'une structure tourbillonnaire peu intense. Concernant les données *in situ*, elles ont été réalisées par deux types de bouées. On distingue les bouées CODE (Coastal Ocean Dynamics Experiment) échantillonnant les vitesses sur 1 mètre depuis la surface et les bouées SVP (Surface Velocity Programme) mesurant les vitesses à 15 mètres de la surface. Après 4 jours, les bouées CODE larguées le 5 mai se situent 20 km au sud de leur position d'origine et sont advectées vers le sud-est les jours

suivant. La simulation réalisée avec les paramètres du satellite SWOT diverge des données *in situ* obtenues avec les bouées CODE. Cependant les données des bouées SVP indiquent que les particules se retrouvent piégées au sein d'une structure tourbillonnaire au bout de 4 jours, correspondant davantage à la simulation réalisée durant le stage. La différence entre les bouées CODE et SVP est la profondeur des vitesses de surface échantillonnées. Ainsi, les données satellitaires suivent davantage le profil donné par les bouées SVP puisque les mesures d'altimétrie réalisées par le satellite SWOT ne prennent pas en considération l'influence de l'ensemble des paramètres météorologiques de surface tels que le forçage du vent. En effet, la tramontane, vent dominant du nord-ouest aux latitudes des îles Baléares a soufflé de façon brève et intense au sein du front Nord-Baléares le 8 mai ; ce qui correspond à la date de changement de direction des bouées CODE vers le sud-est. La trajectoire de ces bouées illustre bien le phénomène de courant d'inertie qui correspond à une advection initiée par un bref coup de vent diminuant au cours du temps sous l'influence des frottements avec les masses d'eau plus en profondeur. Ce phénomène est observable à partir du 11 mai (figure 20) où les bouées décrivent des cercles résultant de l'influence de la pseudo force de Coriolis qui agit davantage sur les particules d'eau à cause de la diminution du gradient de pression dans l'écoulement géostrophique, ce qui les conduit à décrire un mouvement centripète. Cependant, les bouées SVP mesurant les vitesses à 15 mètres de profondeur reçoivent un trop faible pourcentage des vitesses initiées par le coup de vent. En effet, pour que les particules d'eau de subsurface soient influencées par les courants de surface il faut qu'il y ait un vent continu, ce qui permettrait la mise en place de la spirale d'Ekman induite par l'équilibre géostrophique rompu et à la dominance de la pseudo force de Coriolis. Cette simulation nous permet de mettre en lumière une limite de l'utilisation des données satellitaires. En effet, pour améliorer cette simulation il est important de recouper données satellitaires et données *in situ*, ce qui est possible seulement pour un faible pourcentage de données satellitaires. De ce fait, il est impossible de remplir les océans de bouées dérivantes, il faut alors réaliser des approximations et utiliser des modèles numériques pour pallier à ce problème de manque de données de terrain.

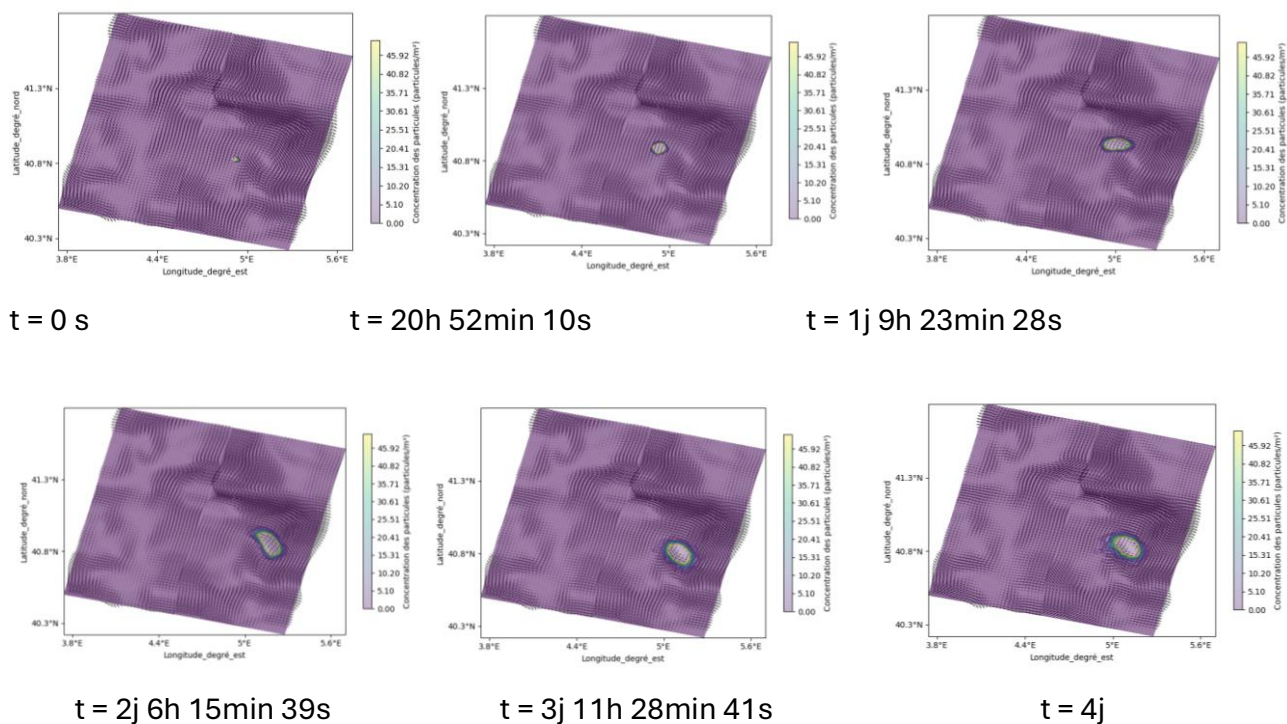


Figure 19 : Simulation de 5000 particules initialisées à (43.83°N/ 4.93°E) dans le modèle de Diffusion-Advection recoupé avec le fichier des données satellitaires de SWOT du 5/05/23 (avant l'épisode intense de Mistral du 8/05/23 à une vitesse > 14 m/s), (cf. Annexe IX)

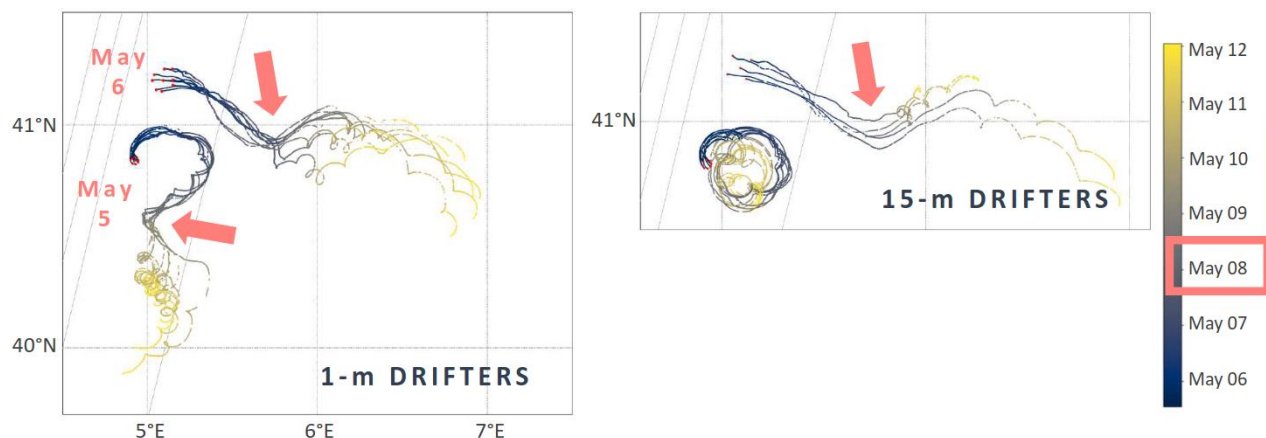


Figure 20 : Trajectoires des bouées dérivantes lancées au sein du front Nord Baléares entre le 5/05/23 et le 12/05/23 mesurant les vitesses de surface à 1 mètre (bouées CODE) et 15 mètres de profondeur (bouées SVP) lors de la campagne BioSWOT-Med (A. Joël, 2024)

Au cours d'une troisième simulation, nous avons cherché à appréhender la proportion de mélange de deux patches de taille similaire au sein du front Nord-Baléares. Pour ce faire, nous avons initialisé deux patches de particules à 41,2°N séparés d'une distance de 2 km chacun, voir figure 21. On observe qu'au bout de 5h 31s, les deux patches se rencontrent et arborent une forme différente dû au champ de vitesse différent qu'ils ont préalablement rencontré. En effet, le patch au sud est étiré et davantage diffus tandis que le patch au nord a une forme plus arrondie et se diffuse plus lentement. A 8h20min52s, les deux patches sont localisés aux mêmes coordonnées géographiques et sont alors complètement confondus. Puis à 10h26min5s ils se retrouvent advectés par les vitesses extérieures d'une structure tourbillonnaire et arborent ainsi une forme davantage étirée. Cette simulation nous permet de simuler la vitesse de mélange de deux populations de plancton. Mais elle peut également décrire la proportion de mélange de deux masses d'eau aux propriétés distinctes où il sera important de prendre en considération l'écart de densité pouvant impacter cette vitesse.

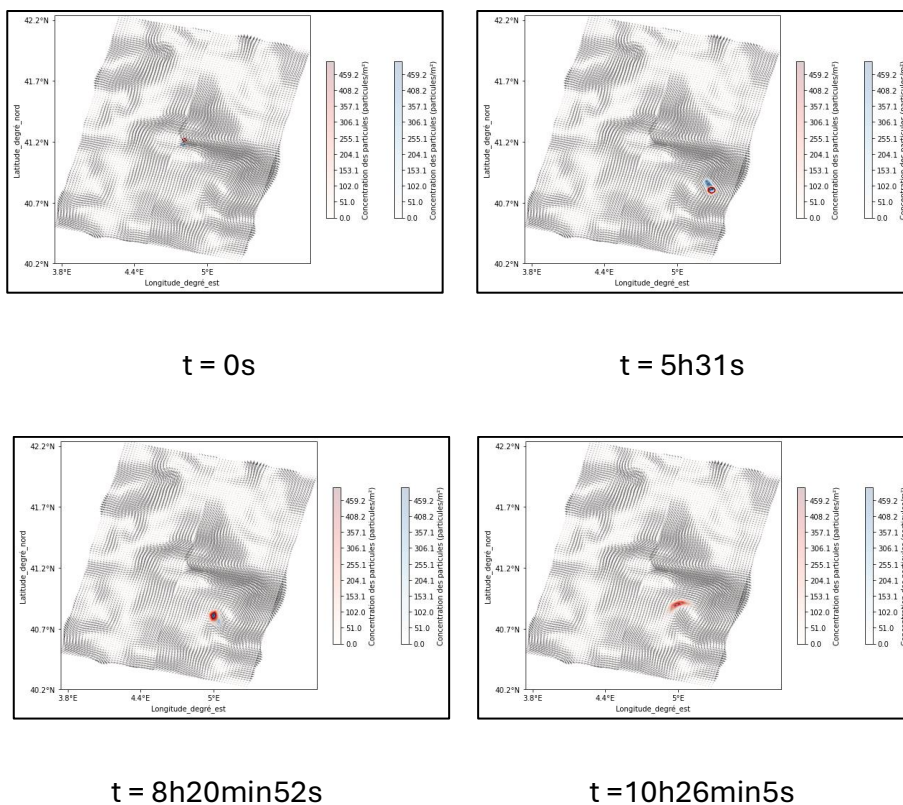


Figure 21 : Evaluation de la proportion de mélange de 2 patches ayant pour position initiale 41.22°N/ 4.81°E (patch de particules rouges) et 41.18°N/ 4.8°E (patch de particules bleues)

D'autre part, nous nous sommes intéressés au phénomène de diffusion au sein des bandes est et ouest du satellite SWOT. Ainsi, pour cette simulation nous avons placé deux patches de particules à la même latitude chacun placé au sein des différentes bandes de données issues du satellite puis nous avons évalué leur dispersion au bout de 10 heures puis de 1 jour. Dans la figure 22, on peut voir qu'après 10 heures de simulation les patches de particules se diffusent selon un mouvement d'advection-diffusion avec des tâches étirées et élargies sous l'effet des deux phénomènes. Cependant le patch dans la bande ouest va davantage s'étirer en filament tandis que celui côté est sera plus diffus. Enfin, à la fin de la simulation nous avons obtenu des figures similaires de part et d'autre de chaque bande avec un centre large et diffus et une structure filamenteuse à l'extérieur de la tâche ; mettant en avant l'influence d'une structure tourbillonnaire. Mais le patch situé sur la bande ouest s'est scindé en deux suite à sa rencontre avec deux structures tourbillonnaires de sens anticyclonique situées dans deux bassins d'attraction différents. Cette simulation nous permet de voir un exemple d'interaction entre un flux de matière et une structure cyclonique mais également de relever les principales caractéristiques d'un tourbillon qui se matérialise par le mouvement en rotation de fluide autour d'un axe rectiligne ou incurvé, ce qui peut exercer une influence sur le devenir d'un patch de particules immiscé dans ce type de structure.

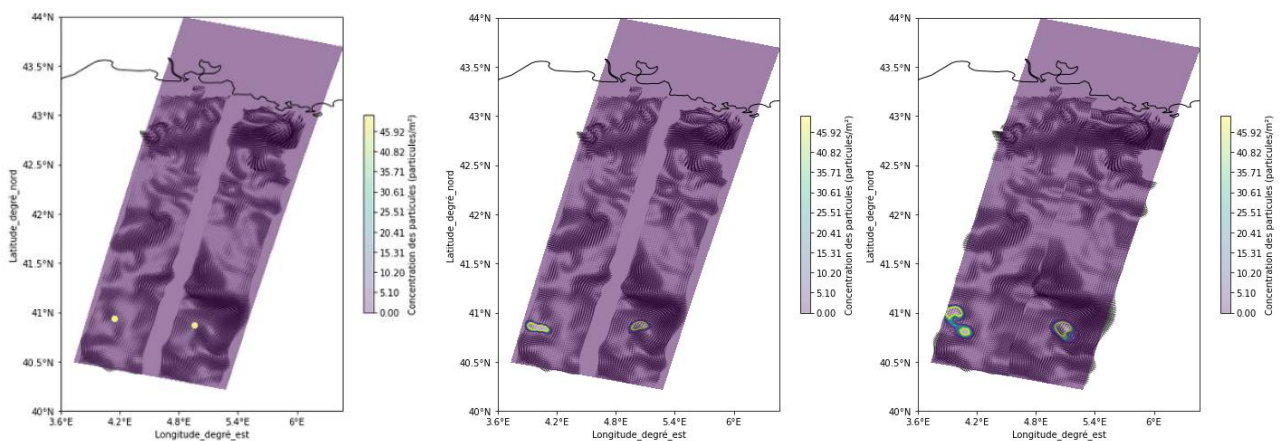
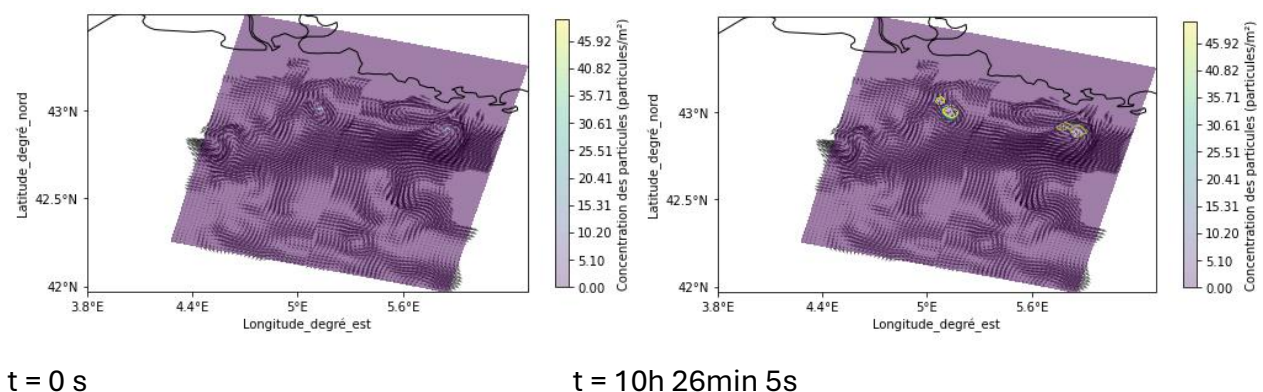
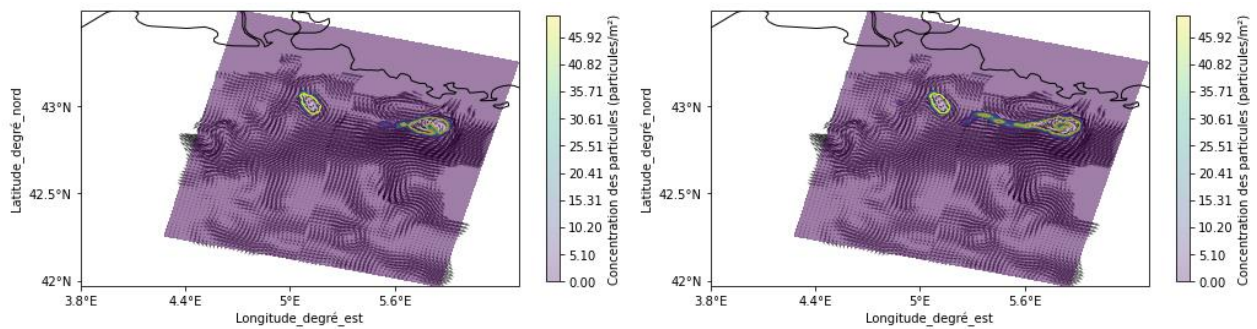


Figure 22 : Simulation de 2 jeux de données initialisés à la même latitude (40.83°N) avec un faible coefficient de diffusion ( $D=0.001$ ) au sein des bandes est et ouest de la trace SWOT du 05/05/23 avec mise en évidence des structures tourbillonnaires

## b. Diffusion au sein du Golfe du Lion

Lors de cette section, nous avons également eu recours à des tests sur des patches de particules au sein de structures tourbillonnaires mais ici au sein de structures tourbillonnaires dans le Golfe du Lion où les deux concentrations de particules ont été initialisées sur chaque bande réalisée par le satellite SWOT à des latitudes et des longitudes différentes. Nous avons souhaité visualiser les différentes structures tourbillonnaires afin de mieux les caractériser ce qui peut être utile lors du suivi d'un contaminant ou du déplacement d'une efflorescence printanière ou automnale. Ainsi, d'après la figure 23 on observe qu'au bout de seulement 10 heures et 26 minutes de simulation nous avons des structures de patches similaires au sein des tourbillons sur chaque bande. En effet, les tâches de concentration ont doublé leur volume et une partie de celle-ci est restée au centre du tourbillon, là où elle a été initialisée. Tandis qu'une portion plus importante d'environ 60% de la tâche totale s'est détachée du patch principal et se situe à quelques kilomètres au sud (bande ouest) et à l'est (bande est) sur le champ de vitesse du satellite SWOT. La dissociation du patch en sous parties peut s'expliquer par la différence de norme et de sens de petits groupes de vitesses au sein du champ de vitesses géostrophiques. Mais également par la présence de zones de front où l'advection y est plus intense et va agir comme frontière entre les différentes concentrations, comme énoncé lors du modèle théorique de l'initialisation du jet gaussien (partie III – 2. b.).





t = 17h 6min 46s

t = 1j 1h 2min 36s

Figure 23 : Evolution de patches de diffusion au sein de 2 structures tourbillonnaires de forme différente identifiées à l'est (42.90°N/ 5.85°E) et à l'ouest (43.01°N/ 5.13°E) des bandes de données d'altimétrie du satellite SWOT (5/05/23)

### c. Population de phytoplanctons soumise à un terme de “Grazing”

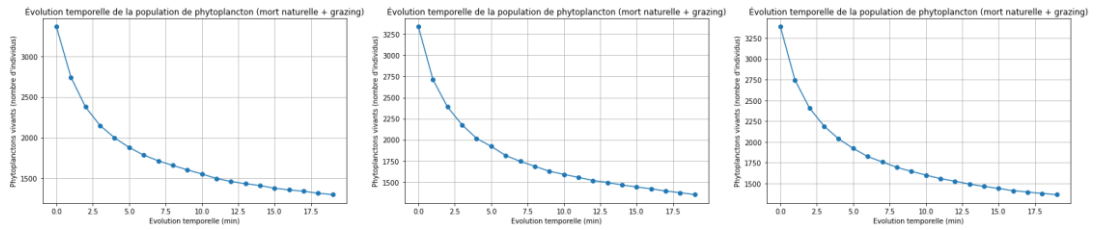
Dans cette dernière partie, nous nous sommes intéressés à l'évolution d'une population de phytoplanctons soumise à des conditions proches des données de chlorophylle a échantillonnées par le satellite SWOT. En effet, la résolution des données de ce satellite sont capables de détecter les caractéristiques océaniques de fine et méso-échelle (15-150 km) telles que les tourbillons auxquels nous nous sommes intéressés précédemment. Mais il est également possible de percevoir l'ensemble de la dynamique de surface telle que le mélange entre eaux de surface et eaux profondes qui permettent la remontée de nutriments, facteurs clés de la formation d'efflorescence planctoniques qui permettent à leur tour l'entrée en énergie dans les écosystèmes et la régulation de la chaîne trophique.

Ainsi, nous avons fait l'hypothèse que nous nous trouvons dans des eaux de surface faibles en nutriments avec une intensité lumineuse importante. De ce fait, nous avons défini notre patch de particules à l'aide d'une population initiale de phytoplancton et son terme de mortalité associé avec un tirage au sort dont la probabilité de mourir augmente avec le temps. Puis nous avons défini un terme de grazing basé sur une population initiale de zooplancton, un taux de grazing et un tirage au sort sur l'ensemble de la population qui estime la proportion de phytoplancton broutée par le zooplancton. Par la suite, nous avons tracé l'évolution du nombre d'individus de la population de phytoplancton en fonction du temps afin de déterminer les limites de cette population vivante.

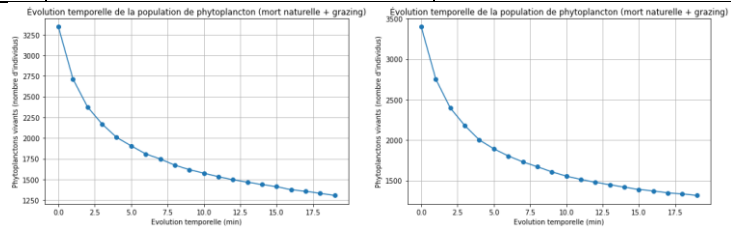
Ainsi, on peut voir d'après la figure 24 que la population de phytoplancton suit une loi exponentielle décroissante et arbore différents taux de décroissance selon la taille de la population de zooplancton utilisée. En effet, on observe graphiquement que plus la proportion de zooplancton par rapport aux phytoplanctons est importante et plus la courbe diminue rapidement. En outre, nous vérifions cela par le calcul en exprimant " $\mu$ " le taux de décroissance net suivant une loi exponentielle, c'est à dire la différence sur le temps  $t$ , entre la croissance cellulaire  $k$  et la mortalité  $g$  (mortalité naturelle + grazing) de la population de phytoplancton d'après la formule :  $\mu = \ln(P_t/P_0)/t = k-g$  où  $P_t$  est la population finale et  $P_0$  la population initiale de densité de phytoplancton, (C. Vasseur, 1996).

Ainsi, pour une échelle d'un millième de zooplancton de la population totale de phytoplancton, on obtient un  $\mu = -0,066$ . Tandis que pour un centième de zooplancton,  $\mu = -0,065$  et pour un dixième  $\mu = -0,067$ . D'autre part, avec une population de zooplancton égale à celle du phytoplancton on trouve  $\mu = -0,068$  et pour une population 10 fois supérieure on trouve un taux de décroissance  $\mu = -0,069$  associés. Le taux de décroissance augmente donc bien lorsque la population de zooplancton est multipliée par 10. Cependant, certaines valeurs peuvent sembler aberrantes comme le taux de décroissance obtenu pour un population de zooplancton occupant une proportion d'un centième. En effet, il devrait être supérieur à  $-0,066$ . Cet écart peut notamment être dû à la densité de phytoplancton initiale qui diminue davantage lorsque la population de zooplancton augmente.

Afin d'améliorer ce modèle il aurait été nécessaire de l'implémenter dans le fichier SWOT pour tester sa validité ainsi que ses limites et nous aurions pu obtenir une figure semblable à celle issue des travaux réalisés lors de la campagne BioSWOT-Med (voir figure 25). En effet, on distingue trois phases sur la courbe présentée caractéristiques d'une zone de mélange au sein du front Nord-Baléares avec un plateau atteint par la production primaire en début et en fin de simulation.



Zooplanktons/ Phytoplanctons	1/1000	1/100	1/10
---------------------------------	--------	-------	------



Zooplanktons/ Phytoplanctons		1/1	10/1
---------------------------------	--	-----	------

Figure 24 : Evolution de la population de phytoplancton théorique sous l'influence du "grazing" avec la prise en compte d'un taux de mortalité naturelle (cf. Annexe X)

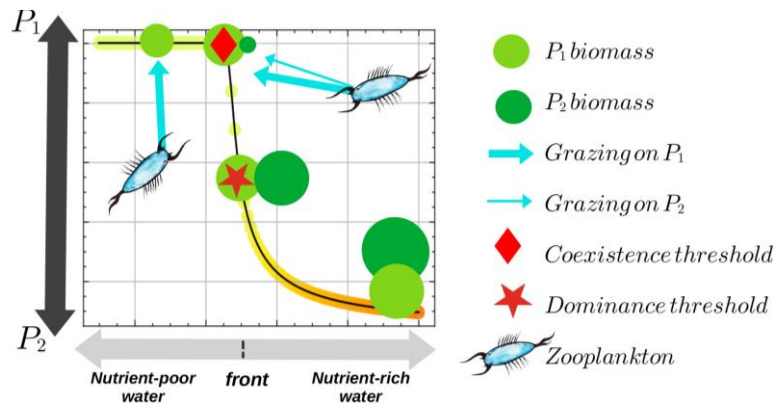


Figure 25 : Diagramme conceptuel de la distribution des communautés phytoplanctoniques présentes le long d'un gradient de nutriments au niveau du front Nord (L. Oms, 2024)

## IV – Conclusion

La réalisation de ce stage m'a permis de mettre en application les notions de cours théoriques de Dynamique des Océans et de mettre en relations différents enseignements de licence Science de la Mer. En effet, ce stage a été marqué par un premier temps d'élaboration de modèles théoriques basés sur les équations du mouvement de Navier-Stokes. Durant cette phase nous avons veillé à la conformité de nos modèles en jouant sur les paramètres de définition et en vérifiant leurs unités. Nous avons également construit le modèle final en définissant d'abord la diffusion puis l'advection. Puis dans un second temps nous avons utilisé des données *in situ* qui ont permis d'ajouter du concret à nos modèles et également de tester leur validité.

## V – Références bibliographiques



Doglioli A.M. and Ballerini T. (eds), 2025. *People, science and instruments of the BioSWOT-Med oceanographic campaign*. Lu::Ce edizioni DOI: <https://doi.org/10.17600/18002392>

Pedlovsky J., 1979. *Geophysical fluid Dynamics*. Springer Verlag (Ed.), p569-570

Quentin-Boris Barral. *Caractérisation du front Nord-Baléares : Variabilité et rôle de la circulation des masses d'eau en Méditerranée Occidentale*. Sciences de la Terre. Université de Toulon, 2022. Français. [〈NNT : 2022TOUL0006〉](#). [〈tel-04399738〉](#)

poly\_DynamiqueOceans2\_2025.pdf.

[https://www.cppm.in2p3.fr/~rnenwein/jkuthagrmp41qfemolr5/poly\\_DynamiqueOceans2\\_2025.pdf](https://www.cppm.in2p3.fr/~rnenwein/jkuthagrmp41qfemolr5/poly_DynamiqueOceans2_2025.pdf).

J.-C. Poggiale, C. Pinazo, A. Doglioli, D. Nerini, M. Gauduchon, K. Lajaunie, 2021. *MODELISATION DE PROCESSUS ENVIRONNEMENTAUX, Transport diffusif uniquement, p-14*, Université d'Aix-Marseille, L3 SVT parcours Mer

Oms, L., M. Messié, J.-C. Poggiale, G. Grégori, and A. Doglioli. 2024. Fine-scale phytoplankton community transitions in the oligotrophic ocean: A Mediterranean Sea case study. *Journal of Marine Systems*, 246:104021, <https://doi.org/10.1016/j.jmarsys.2024.104021>.

Demol, M., M. Berta, L. Gomez Navarro, L. Izard, F. Ardhuin, M. Bellacicco, L. Centurioni, F. D'Ovidio, L. Diaz-Barroso, A. Doglioli, F. Dumas, P. Garreau, A. Joël, I. Lizaran, M. Menna, A. Mironov, B. Mourre, M. Pacciaroni, A. Pascual, A. Ponte, E. Reyes, L. Rousselet, D. R. Tarry, and E. Verger-Miralles. 2023. A Drifter Dataset for the Western Mediterranean Sea collected during the SWOT mission calibration and validation phase. <https://doi.org/10.17882/100828>.

BioSWOT-Med Campaign, Campagne océanographique BioSWOT-Med,  
<https://campagnes.flotteoceanographique.fr/campagnes/18002392/>  
10.17600/18002392.

DOI:

Doglioli, A.M., Moving Vessel Profiler (MVP) schematic illustration provided for the  
BioSWOT-Med campaign (2023–2024),  
[https://youtu.be/BKv\\_2x-hcE4?si=8qpyj6e0QXSc7mlm](https://youtu.be/BKv_2x-hcE4?si=8qpyj6e0QXSc7mlm)

<http://mio.pytheas.univ-amu.fr/~petrenko/julio.htm> mouillage JULIO ADCP + CTD + pressure  
at 100 depth ADCP (300kHz) horizontal currents throughout the water column every 4 m  
frequency 1/2h Raisons scientifiques : Preprint at  
[https://people.mio.osupytheas.fr/~doglioli/BioSWOT/BioSWOT-Med\\_2023/WP-Physics/Petrenko\\_etal\\_EGU2024.pdf](https://people.mio.osupytheas.fr/~doglioli/BioSWOT/BioSWOT-Med_2023/WP-Physics/Petrenko_etal_EGU2024.pdf).

## VI – Annexes

```

#Comparaison Solution théorique et Modèle réel
#Utilisation des posX et non des Concentrations C pour la représentation 1D
#simulation 1D - sans conditions de bord
import numpy as np
import matplotlib.pyplot as plt

#paramètres du modèle
itmax=10
ipmax=5000
grid_size = 100

#positions initiales des particules en 1D
posX0=50
posY0=50

#paramètres gaussienne théorique
x=np.array(range(0,ipmax))
y=np.array(range(0,ipmax))
D=15 #coef de diffusion
deltat=1.0 #pas de temp

#Gaussienne 1D
def gauss1(x,t,D):
    return (1/np.sqrt(4*np.pi*D*time))*np.exp(-((x-posX0)**2./(4*D*time)))

#diffusion des particules à partir d'une position initiale centrée sur [50;50]
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#générer des trajectoires aléatoires au cours du temps (position de chaque particule)
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        posX[it+1,ip]=posX[it,ip]+np.random.normal(0,np.sqrt(4*D*time))
        posY[it+1,ip]=posY[it,ip]+np.random.normal(0,np.sqrt(4*D*time))
#loi normale de paramètre mu et sigma défini avec notre jeu de donnée

#convertir les trajectoires en concentration (densité de particules par espace au cours du temps)
C=np.zeros(grid_size*grid_size*itmax).reshape(grid_size,grid_size,itmax)

for it in range(itmax):
    for ip in range(ipmax):
        i=int(posX[it,ip])
        j=int(posY[it,ip])
        if 0 <= i < grid_size and 0 <= j < grid_size: # éviter les débordements
            C[i,j,it]=C[i,j,it]+1

```

```

# boucle temporelle
for it in range(itmax):
    plt.figure(figsize=(8, 5))

    # Solution théorique (gaussienne)
    print('it=',it)
    time=it*deltat+1e-18 # éviter division par 0
    gauss1=(1/np.sqrt(np.pi*D*time))*np.exp(-((x-posX0)**2./(4*D*time)))
    plt.plot(gauss1 ,color = 'red', linewidth=2, label='Solution théorique')

    # Histogramme des positions des particules selon Ox à l'instant it
    X = posX[it]
    plt.hist(X, bins=50, density=True, alpha=0.6, color='skyblue', edgecolor='black', label='Densité de particules')
    plt.legend()

    # Limites, légendes, titres
    plt.xlim(10, 90)
    plt.ylim(0, 0.15)
    plt.xlabel('Position des particules selon Ox')
    plt.ylabel('Densité des particules')
    plt.title("Evolution de la diffusion à t = {it}")
    plt.legend()
    plt.grid()
    plt.show()

```

## Annexe I : Script python du Modèle de diffusion 1D - Histogramme

```

# boucle temporelle
for it in range(itmax):
    plt.figure(figsize=(8, 5))

    # Solution théorique (gaussienne)
    print('it=', it)
    time=it*deltat+1e-18 # éviter division par 0
    gauss1=(1/np.sqrt(4*np.pi*D*time))*np.exp(-(x-posX0)**2./(4*D*time))
    x_gauss = X[:len(gauss1)] # pour éviter les dépassements
    plt.plot(x_gauss, gauss1, color='red', linewidth=2, label='Solution théorique')

    # Densité des positions des particules selon Ox à l'instant it
    X = posX[it]
    # Calcul de la densité normalisée
    counts, bin_edges = np.histogram(X, bins=50, density = True) #ici density trace les fréquences en %
    bin_centers = (bin_edges[:-1] + bin_edges[1:]) / 2 # milieux des intervalles
    # Tracer la densité comme une courbe
    plt.plot(bin_centers, counts, color='blue', linewidth=2, label='Densité des particules')

    # Limites, légendes, titres
    plt.xlim(10, 90)
    plt.ylim(0, 0.15)
    plt.xlabel('Position des particules selon Ox')
    plt.ylabel('Densité des particules')
    plt.title(f"Évolution de la diffusion à t = {it} pour {ipmax} particules")
    plt.legend()
    plt.grid()
    plt.show()

```

## Annexe II : Script python du Modèle de Diffusion 1D – Courbe de densité

```

# Simulation 2D - avec profil à y=50
import numpy as np
import matplotlib.pyplot as plt
# Paramètres de simulation 2D
itmax = 10
ipmax = 5000
posX0, posY0 = 50, 50
grid_size = 100
X, Y = np.meshgrid(np.arange(grid_size), np.arange(grid_size)) #simuler une grille 2D à partir vecteurs 1D
D = 15 # coef de diffusion
deltat = 1.0

# Initialisation des trajectoires
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#Gaussienne 2D
def gauss2(X,Y,t,D):
    return (ipmax/np.sqrt(4*np.pi*D*t))* np.exp(-((X-posX0)**2+(Y-posY0)**2)/(4*D*t))

for it in range(itmax - 1):
    for ip in range(ipmax):
        time = deltat*it
        posX[it+1,ip]=posX[it,ip]+np.random.normal(0,np.sqrt(2*D*deltat)) #ecart type = données dans dimension
        posY[it+1,ip]=posY[it,ip]+np.random.normal(0,np.sqrt(4*D*deltat)) # et Variance = données au carré
#prendre l'écart type et non la variance car c'est une mesure de dispersion plus facile à interpréter

# Calcul de la concentration spatiale
C = np.zeros((grid_size, grid_size, itmax))

for it in range(itmax): #ici on va jusqu'à itmax pour prendre en compte t = 9
    for ip in range(ipmax):
        i = int(posX[it, ip])
        j = int(posY[it, ip])
        if 0 <= i < grid_size and 0 <= j < grid_size: # éviter les débordements
            if i == 0 or j == 0 or i == grid_size - 1 or j == grid_size - 1:
                C[i, j, it] = -1 # Marque cette cellule comme "perdue"
            else:
                if C[i, j, it] != -1: # S'assurer de ne pas incrémenter une case marquée
                    C[i, j, it] += 1

```

```

# Visualisation + Gaussienne analytique
for it in range(itmax):
    plt.figure(figsize=(6, 5))
    time = it * deltat + 1e-18 # éviter division par 0

    # Calcul de la gaussienne analytique
    G2 = gauss2(X, Y, time, D)
    G2 *= np.sum(C[:, :, it]) # ajuste l'amplitude: adapter la gaussienne aux densités de notre simulation

    # Affichage des contours 2D
    plt.contour(G2, levels=5, colors='r', linewidths=1, label='Gaussienne théorique')
    contour = plt.contourf(C[:, :, it], levels=20, cmap='viridis')
    cbar=plt.colorbar()
    cbar.ax.set_title('concentration de particules', rotation=0)
    plt.xlabel("Diffusion des particules selon Ox")
    plt.ylabel("Densité de particules selon Oy")
    plt.title(f"Diffusion 2D à (t = {it})")
    # Profil 1D à y = 50
    plt.figure(figsize=(6, 3))
    x = np.arange(grid_size)
    plt.plot(C[:, 50, it]/ipmax, label="Simulation (coupure Y=50)", color='blue')
    plt.plot(x, G2[:, 50]/np.sum(G2), label="Gaussienne 2 (Y=50)", color='red') # normalisation pour comparaison

    plt.xlim(0, 100)
    plt.ylim(0, 0.015)
    plt.title(f"Profil de diffusion selon X à Y=50 (t = {it})")
    plt.xlabel("Position des particules selon Ox")
    plt.ylabel("Densité de particules")
    plt.legend()
    plt.grid()
    plt.tight_layout()
    plt.show()

```

## Annexe III : Script python du Modèle de Diffusion 2D et du profil à y = 50

```

#Ajout de conditions de bord - avec paramètres de définition du domaine
# nombre de particules commence à diminuer au bout de t=3
# due à npsqrt(4*D*it) trop petit à t=1 et non def à t=0 car vaut 0
#bon signe : échelle de résolution adaptée
#simulation 1D
import numpy as np
import matplotlib.pyplot as plt

#paramètres du modèle
itmax=10
ipmax=5000
grid_size_X = 100
grid_size_Y = 100

out = grid_size+1 #paramètre de bord

#positions initiales des particules en 1D
posX0 , posY0 =50 , 50

#paramètres gaussienne théorique
x=np.array(range(0,ipmax))
y=np.array(range(0,ipmax))
D=15 #coef de diffusion
deltat=1.0 #pas de temps

#Gaussienne 1D
def gauss1(x,t,D):
    return (1/np.sqrt(4*np.pi*D*time))*np.exp(-((x-posX0)**2./((4*D*time)))

#diffusion des particules à partir d'une position initiale centrée sur [50;50]
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#générer des trajectoires aléatoires au cours du temps (position de chaque particule)
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        dx = np.random.normal(0,np.sqrt(4*D*time)) #bruit des particules
        dy = np.random.normal(0,np.sqrt(4*D*time))
        xnew = posX[it,ip] + dx #position totale = pos moy + pos bruit
        ynew = posY[it,ip] + dy
        # Condition de bord : supprimer particules si sortie du domaine selon x ET y
        if 0 <= xnew < grid_size_X or 0 <= ynew < grid_size_Y: #attention valable pour toutes les directions du mod
            posX[it+1,ip] = xnew
            posY[it+1,ip] = ynew
        else:
            posX[it+1,ip] = out
            posY[it+1,ip] = out

# boucle temporelle
for it in range(itmax):
    plt.figure(figsize=(8, 5))

    # Solution théorique (gaussienne)
    print('it=',it)
    time=it*deltat+1e-18 # éviter division par 0
    G1 = gauss1(x,time,D)
    plt.plot(G1 ,color= 'red', linewidth=2, label='Solution théorique')

    # Histogramme des positions des particules selon Ox à l'instant it
    X = posX[it]
    X = X[X!=out]
    #paramètres de l'histogramme
    Xmoy = np.mean(X)
    ecarttype = np.std(X)
    effectif = len(X)
    print("La densité moyenne :",Xmoy,"[X/Xtot]")
    print("L'écart-type vaut :",ecarttype,"[X/Xtot]")
    print("Le nombre de particules :",effectif)
    plt.hist(X, bins=50, density=True, alpha=0.6, color='skyblue', edgecolor='black', label='Densité de particules')
    plt.legend()

    # Limites, légendes, titres
    plt.xlim(10, 90)
    plt.ylim(0, 0.15)
    plt.xlabel('Position des particules selon Ox')
    plt.ylabel('Densité des particules')
    plt.title(f"Évolution de la diffusion à t = {it}")
    plt.legend()
    plt.grid()
    plt.show()

```

Annexe IV : Script python du Modèle de Diffusion 1D avec conditions de bord

```

# Simulation 2D - avec ajout de conditions de bord
import numpy as np
import matplotlib.pyplot as plt
# Paramètres de simulation 2D
itmax = 10
ipmax = 5000
posX0, posY0 = 50, 50
grid_size = 100
X, Y = np.meshgrid(np.arange(grid_size), np.arange(grid_size)) #simuler une grille 2D à partir vecteurs
D = 15 # coef de diffusion
deltat = 1.0
out = grid_size+1 #paramètre de bord

# Initialisation des trajectoires
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#Gaussienne 2D
def gauss2(X,Y,t,D):
    return (ipmax/np.sqrt(4*np.pi*D*t))* np.exp(-((X-posX0)**2+(Y-posY0)**2)/(4*D*t))

#générer des trajectoires aléatoires au cours du temps (position de chaque particule)
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        dx = np.random.normal(0,np.sqrt(4*D*time)) #bruit des particules
        dy = np.random.normal(0,np.sqrt(4*D*time))
        xnew = posX[it,ip] + dx #position totale = pos moy + pos bruit
        ynew = posY[it,ip] + dy
        # Condition de bord : supprimer particules si sortie du domaine selon x ET y
        if 0 <= xnew < grid_size or 0 <= ynew < grid_size: #attention valable pour toutes les directions
            posX[it+1,ip] = xnew
            posY[it+1,ip] = ynew
        else:
            posX[it+1,ip] = out
            posY[it+1,ip] = out

# #convertir les trajectoires en concentration (densité de particules par espace au cours du temps)
C=np.zeros(grid_size*grid_size*itmax).reshape(grid_size,grid_size,itmax)

for it in range(itmax):
    for ip in range(ipmax):
        if posX[it,ip] != out or posY[it,ip] != out:
            i=Int(posX[it,ip])
            j=Int(posY[it,ip])
            if 0 <= i < grid_size and 0 <= j < grid_size: # éviter les débordements
                C[i,j,it]=C[i,j,it]+1

```

```

# Visualisation + Gaussienne analytique
for it in range(itmax):
    plt.figure(figsize=(6, 5))
    time = it * deltat + 1e-18 # éviter division par 0

    # Calcul de la gaussienne analytique
    G2 = gauss2(X, Y, time, D)
    G2 *= np.sum(C[:, :, it]) # ajuste l'amplitude: adapter la gaussienne aux densités de notre simulation

    # Affichage des contours 2D
    plt.contour(G2, levels=5, colors='r', linewidths=1, label='Gaussienne théorique')
    contour = plt.contourf(C[:, :, it], levels=20, cmap='viridis')
    cbar=plt.colorbar()
    cbar.ax.set_title('concentration de particules', rotation=0)

    # Positions des particules à l'instant it
    Xvaleur = posX[it]
    Yvaleur = posY[it]
    X_valide = (Xvaleur != out) & (Yvaleur != out)
    Xvaleur = Xvaleur[X_valide]
    Yvaleur = Yvaleur[X_valide]
    #calcul de densité, effectif et écart type 2D
    effectif = len(Xvaleur)
    Xmoy = effectif / (grid_size**2)
    distance_carre = (Xvaleur - posX0)**2 + (Yvaleur - posY0)**2
    ecart_type = np.sqrt(np.mean(distance_carre)) # écart-type radial
    print(f"t = {it}")
    print(f"Densité moyenne par cellule de comptage (centre de masse) : {Xmoy}")
    print(f"Ecart type de la dispersion : {ecart_type}")
    print(f"Nombre de particules restantes : {effectif}")

    plt.xlabel("Diffusion des particules selon Ox")
    plt.ylabel("Densité de particules selon Oy")
    plt.title(f"Diffusion 2D à (t = {it})")
    plt.legend()
    plt.grid()
    plt.tight_layout()
    plt.show()

```

## Annexe V : Script python du Modèle de Diffusion 2D avec conditions de bord

```
#paramètres du modèle
itmax=10
ipmax=5000
deltat=1.0 #pas de temp
D=0.1 #coef de diffusion - attention a prendre petit car sinon sortie de la grille

#paramètres de grille
grid_size_X = 100
grid_size_Y = 100
#condition de bord - 2nde grille superposée
grid_size_X_b = 55
grid_size_Y_b = 55
out_X = grid_size_X_b+1 #condition de bord x
out_Y = grid_size_Y_b+1 #condition de bord y

#positions initiales des particules
posX0=50
posY0=50

#Paramètres champ de vitesse
x=np.array(range(1,grid_size_X))
y=np.array(range(1,grid_size_Y))
# Coordonnées de grille
x = np.arange(0, grid_size_X)
y = np.arange(0, grid_size_Y)
X, Y = np.meshgrid(x, y)

#Vitesse moyenne
Umoy = 4
Vmoy = 0

#matrice initiale des vitesses
U=Umoy*np.ones(grid_size_Y*grid_size_X).reshape(grid_size_Y,grid_size_X)
V=Vmoy*np.ones(grid_size_Y*grid_size_X).reshape(grid_size_Y,grid_size_X)
U[50:]= -U[50:] #pour 2 courants rencontre en sens inverse

#tracé profil vitesse horizontale
plt.plot(U)
plt.title('Profil de la vitesse horizontale U(x,t) positive selon Oy[0,50] et négative selon Oy[50,100]')
plt.xlabel('axe Oy')
plt.ylabel('Norme de la vitesse horitontale (m/s)')

#matrice initiale des positions - diffusion des particules à partir d'une position initiale centrée sur [50;50]
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
```

```

#modèle Advection - Diffusion
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        ipart = int(posY[it,ip])
        jpart = int(posX[it,ip])
        xnew=posX[it,ip]+np.random.normal(0,np.sqrt(4*D*time))
        ynew=posY[it,ip]+np.random.normal(0,np.sqrt(4*D*time))
        # Condition de bord : supprimer particules si sortie du domaine selon x ET y
        if 0 <= xnew < grid_size_X_b or 0 <= ynew < grid_size_Y_b: #attention valable pour toutes les directions
            posX[it+1,ip] = xnew+U[ipart,jpart]#position particules précédente + sa vitesse précédente associée +
            posY[it+1,ip] = ynew+V[ipart,jpart]
        else:
            posX[it+1,ip] = out_X
            posY[it+1,ip] = out_Y

#convertir les trajectoires en concentration (densité de particules par espace au cours du temps)
C=np.zeros(grid_size_X*grid_size_Y*itmax).reshape(grid_size_X,grid_size_Y,itmax)

for it in range(itmax):
    for ip in range(ipmax):
        i=int(posY[it,ip])
        j=int(posX[it,ip])
        if 0 <= i < grid_size_X_b and 0 <= j < grid_size_Y_b: # éviter les débordements
            C[i,j,it]=C[i,j,it]+1

# boucle temporelle
for it in range(itmax):
    time = it*deltat+1e-18
    plt.figure()
    print(f"À t = {it}")
    step = 7 # on affiche un vecteur sur 7
    #Dénombrement de particules restantes dans étude - conditions de bord
    I = posX[it]
    I = I[I!=out_X]
    effectif = len(I)

    plt.quiver(X[::step, ::step], Y[::step, ::step], U[::step, ::step], V[::step, ::step], scale=100, color = 'b')
    contour = plt.contourf(C[:, :, it], levels=30, cmap='Blues', alpha = 0.4)
    plt.colorbar(contour,label = 'Concentration à y = 50', shrink=0.5)
    plt.vlines(x=55, ymin=0, ymax=55, color='red', label = "zone d'étude")
    plt.hlines(y=55, xmin=0, xmax=55, color = 'red')
    plt.xlabel("Longueur du bassin d'étude - axe 0x")
    plt.ylabel("Largeur du bassin d'étude - axe 0y")
    plt.title(f"Transport de la diffusion dans un bassin {grid_size_X}*{grid_size_Y} à t = {it}")
    print("Le nombre de particules :",effectif)
    plt.legend()
    plt.show()

```

Annexe

## VI : Script python du Modèle d'Advection-Diffusion 2D - champ de vitesse constant

```

import numpy as np
import matplotlib.pyplot as plt

#Affichage de 1 concentration - front jet gaussien
#2) ADVECTION DIFFUSION 2D

#paramètres du modèle
itmax=20
ipmax=5000
deltat=1.0 #pas de temp (s)

#constantes de campagne BioSwotMed
D = 0.1 #coef de diffusion - attention a prendre petit car sinon sortie de la grille (m²/s)
N = 300 #paramètre de largeur gaussienne (m/s)
c = 50 #coef vitesse - hauteur gaussienne (m/s)
a = 50 #position pic gaussienne = gaussienne centrée (m)

#grille d'étude
grid_size_X = 100
grid_size_Y = 100
out_X = grid_size_X+1 #condition de bord x
out_Y = grid_size_Y+1 #condition de bord y

#champ de vitesse
x=np.array(range(1,grid_size_X))
y=np.array(range(1,grid_size_Y))

#grille champ de vitesse
x = np.arange(0, grid_size_X)
y = np.arange(0, grid_size_Y)
X, Y = np.meshgrid(x, y)

```

```

#Champ de vitesse 2D
#vitesse méridienne (Sud>Nord) - nulle
Vmoy = 0
V=Vmoy*np.ones(grid_size_Y*grid_size_X).reshape(grid_size_Y,grid_size_X)
#vitesse zonale (Ouest>Est) - Gaussienne
def Umoy(y,N):
    return (1/np.sqrt(4*np.pi*N))*np.exp(-((y-a)**2/(4*N)))*c
u_profile = Umoy(y, N) # profil horizontal - renommer fonction
U = np.tile(u_profile.reshape(-1, 1), (1, grid_size_X)) # répéter en colonnes

#positions initiales des particules - centre
posX0_0=0
posY0_0=50

#matrice initiale des positions - diffusion des particules à partir d'une posit
posX_0=posX0_0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY_0=posY0_0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

```

```

#modèle Advection - Diffusion
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        ipart = int(posY_0[it,ip])
        jpart = int(posX_0[it,ip])
        ipart = max(0, min(grid_size_Y - 1, ipart))
        jpart = max(0, min(grid_size_X - 1, jpart))
        posX_0[it+1,ip]=posX_0[it,ip]+U[ipart,jpart]*np.random.normal(0,np.sqrt(4*D*time))
        posY_0[it+1,ip]=posY_0[it,ip]+V[ipart,jpart]*np.random.normal(0,np.sqrt(4*D*time))

#conversion des trajectoires en concentration (densité de particules par espace au cours du temps)
C_0=np.zeros(grid_size_X*grid_size_Y*itmax).reshape(grid_size_X,grid_size_Y,itmax)

for it in range(itmax):
    for ip in range(ipmax):
        i=int(posY_0[it,ip])
        j=int(posX_0[it,ip])
        if 0 <= i < grid_size_X and 0 <= j < grid_size_Y: # éviter les débordements
            C_0[i,j,it]=C_0[i,j,it]+1

```

```

#Affichage profil de vitesse
plt.figure()
step = 2
plt.quiver(X[::step, ::step], Y[::step, ::step], U[::step, ::step], V[::step, ::step], scale=20, color = 'black')
plt.plot(u_profile*90,x)
plt.title('Profil du Jet Gaussien')
plt.xlabel('axe Ox')
plt.ylabel('Vitesse gaussienne')

```

```

# boucle temporelle
for it in range (itmax):
    time = it*deltat+1e-18
    plt.figure()
    print(f'A t = {it}')
    step = 2 # espacement affichage vecteurs
    plt.quiver(X[::step, ::step], Y[::step, ::step], U[::step, ::step], V[::step, ::step], scale=20, color = 'b')
    # plt.xlim(X.min(), X.max()) #calibrer affichage
    # plt.ylim(Y.min(), Y.max())
    # plt.scatter(posX[it, :], posY[it, :], s=0.1, color='red', alpha=0.5, label="Particules")
    levels = np.array(range(0,20))

    contour_0= plt.contourf(C_0[:, :, it], levels, cmap='Blues', alpha = 0.4)
    plt.colorbar(contour_0,label = 'Concentration des particules (particules/m²)', shrink=0.5)
    plt.axvline(x=it, color='red', linestyle='-', label="Déplacement des particules au sein du front")
    plt.xlabel("Longueur du bassin d'étude - axe Ox")
    plt.ylabel("Largeur du bassin d'étude - axe Oy")
    plt.title(f"Transport de la diffusion dans un bassin (grid_size_X)={grid_size_Y} à t = {it}")
    plt.legend()
    plt.show()

```

## Annexe VII : Script python – Modèle d'advection diffusion – jet gaussien

```

#3) Test de mélange des masses d'eau - Modèle de diffusion sans advection - sans conditions de bord
#paramètres du modèle
itmax = 1000
ipmax = 1000
#jeu de donnée des masses d'eau
x1, x2 = 1000,1000
#constantes de campagne BioSwotMed
D = 0.1 #coef de diffusion - attention a prendre petit car sinon sortie de la grille (m²/s)
N = 100 #paramètre de largeur gaussienne (m²/s)
c = 19000 #coef vitesse - hauteur gaussienne (m/s)
a = 50 #pic de vitesse = gaussienne centrée (m)
#paramètres de la grille
grid_size_X = 100
grid_size_Y = 100
out_X = grid_size_X+1 #condition de bord x
out_Y = grid_size_Y+1 #condition de bord y

#positions initiales des particules en 2D
posX0 = np.array(range(0,x1))
posY0 = np.array(range(-x2,0))

#Grille des particules de chaque masse d'eau
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#####
#AVEC ADVECTION
#####
#Champ de vitesse 2D
#vitesse méridienne (Sud>Nord) - nulle
Vmoy = 0
V=Vmoy*np.ones(grid_size_Y*grid_size_X).reshape(grid_size_Y,grid_size_X)
#vitesse zonale (Ouest>Est) - Gaussienne
def Umoy(y,N):
    return (1/np.sqrt(4*np.pi*N))*np.exp(-((y-a)**2./(4*N)))*c
u_profile = Umoy(y, N) # profil horizontal - renommer fonction
U = np.tile(u_profile.reshape(-1, 1), (1, grid_size_X)) # répéter en colonnes et répliquer un tabl

#modèle Advection - Diffusion
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        ipart = int(posY[it,ip])
        jpart = int(posX[it,ip])
        ipart = max(0, min(grid_size_Y - 1, ipart))
        jpart = max(0, min(grid_size_X - 1, jpart))
        posX[it+1,ip]=posX[it,ip]+U[ipart,jpart]+np.random.normal(0,np.sqrt(4*D*time))
        posY[it+1,ip]=posY[it,ip]+V[ipart,jpart]+np.random.normal(0,np.sqrt(4*D*time))
#####

```

```

#diffusion des particules - générer des trajectoires aléatoires au cours du temps
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        posX[it+1,ip]=posX[it,ip]+np.random.normal(0,np.sqrt(4*D*time)) #variance
        posY[it+1,ip]=posY[it,ip]+np.random.normal(0,np.sqrt(4*D*time))

#affichage
plt.xlim(0,1000)
plt.ylim(-1000,1000)
plt.plot(posX, '.', color = 'blue')
plt.plot(posY, '.', color = 'skyblue')
plt.axvline(x=600, color = 'red')
plt.xlabel('Temps de simulation (secondes)')
plt.ylabel('Largeur de la couche de mélange (mètres)')
plt.title(f"Rencontre de deux masses d'eau de {x1*x2} particules sans advection")
plt.show()

```

Annexe VIII : Script python de la rencontre de deux masses d'eau au sein d'un front avec et sans advection

```

import matplotlib.pyplot as plt
from netCDF4 import Dataset as netcdf_dataset
import numpy as np
import cartopy.crs as ccrs
import cartopy.mpl.ticker as cticker

#Advection - fichier netCDF
#ouvrir le fichier netCDF
SWOT_v3 = '/anuhone/p22011306/Python3/Stage_L3_Modèle de dispersion dans un courant à jet/SWOT_L3_LR_SSH_E
dataset = netcdf_dataset(SWOT_v3)
#Zone frontale utilisée
u = dataset.variables['ugos'][:,7200:7275,1:69]
v = dataset.variables['vgos'][:,7200:7275,1:69]
lat = dataset.variables['latitude'][:,7200:7275,1:69]
long = dataset.variables['longitude'][:,7200:7275,1:69]

# Marque colonnes égales à 0
is_zeros_col_u = np.all(u == 0, axis=0)
is_zeros_col_v = np.all(v == 0, axis=0)

# Colonnes index
cols_u = np.arange(u.shape[1])
cols_v = np.arange(v.shape[1])

# Parcours des colonnes vides
for j in cols_u[is_zeros_col_u]:
    valid_cols_u = cols_u[~is_zeros_col_u] # Chercher les colonnes non nulles
    nearest = valid_cols_u[np.argmin(np.abs(valid_cols_u - j))] # Trouver la colonne valide la plus proche
    u[:, j] = u[:, nearest] # Remplacer toute la colonne par la colonne la plus proche

for j in cols_v[is_zeros_col_v]:
    valid_cols_v = cols_v[~is_zeros_col_v]
    nearest = valid_cols_v[np.argmin(np.abs(valid_cols_v - j))]
    v[:, j] = v[:, nearest]

```

```

#Diffusion
#paramètres du modèle
itmax=26
ipmax=5000
deltat=1.0 #pas de temp
D=0.01 #coef de diffusion - attention a prendre petit car sinon sortie de la grille

#paramètres de grille
grid_size_X = 200
grid_size_Y = 68

#positions initiales des particules
posX0=20
posY0=35

#matrice initiale des positions - diffusion des particules à partir d'une position initiale centrée
posX=posX0*np.ones(itmax*ipmax).reshape(itmax,ipmax)
posY=posY0*np.ones(itmax*ipmax).reshape(itmax,ipmax)

#modèle Advection - Diffusion
for it in range(itmax-1):
    for ip in range(ipmax):
        time=it*deltat+1e-18 # éviter division par 0
        ipart = int(posY[it,ip])
        jpart = int(posX[it,ip])
        ipart = max(0, min(grid_size_Y - 1, ipart))
        jpart = max(0, min(grid_size_X - 1, jpart))
        posX[it+1,ip]=posX[it,ip]+(u[jpart,jpart])*10+np.random.normal(0,np.sqrt(4*D*time))
        posY[it+1,ip]=posY[it,ip]+(v[jpart,jpart])*10+np.random.normal(0,np.sqrt(4*D*time))

#convertir les trajectoires en concentration (densité de particules par espace au cours du temps)
C=np.zeros(grid_size_X*grid_size_Y*itmax).reshape(grid_size_X,grid_size_Y,itmax)

for it in range(itmax):
    for ip in range(ipmax):
        i = int(posY[it, ip])
        j = int(posX[it, ip])
        if 0 <= i < grid_size_Y and 0 <= j < grid_size_X:
            C[i, j, it] += 1

# boucle temporelle
for it in range (itmax):
    time = it*deltat+1e-18
    print(f"À t = {it}")
    step = 20 # espacement affichage vecteurs
    fig = plt.figure(figsize=(10,8))

```

```

#Affichage champ de vitesse
ax = plt.axes(projection=ccrs.PlateCarree())
ax.quiver(long, lat, u, v,transform=ccrs.PlateCarree(), scale =10, alpha = 0.6)
ax.coastlines()

levels = np.linspace(0,50)
contour = ax.contourf(long, lat, C[:, :, it], levels, cmap='viridis', alpha = 0.6, transform=ccrs.PlateCarree())
plt.colorbar(contour,label = 'Concentration des particules (particules/m²)', shrink=0.5)

# Define the xticks for longitude
ax.set_xticks(np.arange(3.6,6.5,0.6), crs=ccrs.PlateCarree())
lon_formatter = cticker.LongitudeFormatter()
ax.xaxis.set_major_formatter(lon_formatter)
plt.xlabel('Longitude_degré_est')

# Define the yticks for latitude
ax.set_yticks(np.arange(40,44.5,0.5), crs=ccrs.PlateCarree())
lat_formatter = cticker.LatitudeFormatter()
ax.yaxis.set_major_formatter(lat_formatter)
plt.ylabel('Latitude_degré_nord')

plt.show()

```

## Annexe IX : Script de comparaison entre les données *in situ* dans le modèle d'Advection-Diffusion avec les données du satellite SWOT

```

#Modèle de grazing du phytoplancton
import numpy as np
import matplotlib.pyplot as plt

#paramètres du modèle
pop_phytop = 5000
pop_zoop = 50000
itmax = 10 #nombre de pas de temps de la simulation

#paramètres exponentiels, à adapter selon paramètres in situ
drop_death = 1 #taux de mortalité suit une loi exponentielle : augmente avec le temps
grazing_rate = 1 #taux de grazing par pas de temps

#Probabilité âge phytoplancton
def phytop_life():
    age = np.random.uniform(0,1) #proba initiale de vie
    lifetime = np.random.exponential(1.0)
    alive = True
    return {"age": age, "lifetime": lifetime, "alive": alive} #utilisation de dictionnaires

#Tirage au sort mort phytoplancton
def phytop_death(proba):
    proba["age"] += 1
    death_chance = proba["age"] / (proba["lifetime"] + proba["age"])
    for i in range(drop_death):
        if np.random.rand() > death_chance:
            proba["alive"] = False

#Zooplancton
def grazing(phytop_life, grazing_rate):
    alive_phyto = [proba for proba in phytop_life if proba["alive"]]
    if not alive_phyto:
        victims = np.random.choice(alive_phyto, min(grazing_rate, len(alive_phyto)), replace=False)
        for v in victims:
            v["alive"] = False
    return victims

```

```

#Grazing phytoplankton par le zooplankton
phytos = [phytop_life() for ip in range(pop_phytop)]
def phytop_grazing():
    for it in range(itmax):
        for ip in phytos:
            if ip["alive"]:
                phytop_death(ip)
        for ip in range(pop_zoop):
            grazing(phytos, grazing_rate)
        alive = sum(ip["alive"] for ip in phytos)
        print(f"A t = {it}: {alive} phytoplanctons vivants")

# Boucle de simulation
phyto_alive_counts = []
for it in range(itmax):
    for ip in phytos:
        if ip["alive"]:
            phytop_death(ip)
    for ip in range(pop_zoop):
        grazing(phytos, grazing_rate)
    alive = sum(ip["alive"] for ip in phytos)
    phyto_alive_counts.append(alive) # Enregistrer le nombre vivant
    print(f"A t = {it}: {alive} phytoplanctons vivants")

# Tracer le résultat
plt.figure(figsize=(8, 5))
plt.plot(range(itmax), phyto_alive_counts, marker='o')
plt.xlabel("Évolution temporelle (min)")
plt.ylabel("Phytoplanctons vivants (nombre d'individus)")
plt.title("Évolution temporelle de la population de phytoplancton (mort naturelle + grazing)")
plt.grid()
plt.show()

```

## Annexe X : Modèle de Grazing du phytoplancton par le zooplancton + mortalité naturelle