

# Rapport de stage : Estimation maille de grille

Breandan Cinato

July 2025

## 1 Introduction

Le but de cet rapport est d'estimer une maille de grille optimal pour une région donnée dans le cadre d'étude en approche lagrangienne. En effet, pour obtenir des résultats représentant le plus fidèlement ce qu'on pourrait observer en conditions in situ, il est primordial de réaliser des simulations numérique avec des conditions spatials en accord avec la diffusion des particules. On notera cette maille tel que :

$$\Delta x = \sqrt{4Dt}$$

avec D le coefficient de diffusion horizontal en  $m^2/s$  et t le temps de simulation en s obtenu en multipliant le temps total de simulation par le pas de temps  $dt$ .

## 2 Méthode 1

### 2.1 Principe 1

Le but de cette première méthode est d'obtenir le coefficient de diffusion  $D_{eff}$  grâce à la pente de la droite associée à la variance en fonction du temps, fonction notée  $\Delta r^2(t)$ .

On donne donc l'expression de la variance tel que :

$$\langle (\Delta r)^2 \rangle (t) = \frac{1}{N} \sum_{i=1}^N \left[ (x_i(t) - \bar{x}(t))^2 + (y_i(t) - \bar{y}(t))^2 \right]$$

Où  $x_i$  et  $y_i$  sont les coordonnées des particules à l'instant t et  $\bar{x}_i$  et  $\bar{y}_i$  sont les coordonnées du barycentre des particules i à l'instant t tel que :

$$\bar{x}(t) = \frac{\sum_{i=1}^N \rho_i x_i(t)}{\sum_{i=1}^N \rho_i} \quad , \quad \bar{y}(t) = \frac{\sum_{i=1}^N \rho_i y_i(t)}{\sum_{i=1}^N \rho_i}$$

Avec  $\rho_i$  la densité spatiale locale.

Une fois la courbe de des variances quadratique moyenne des déplacements en fonction du temps obtenu on fait une régression linéaire pour lisser les données et obtenir une droite afin d'en retirer la pente et calculer  $D_{eff}$ .

A noter que cette technique fonctionne sous l'hypothèse d'un champ de vitesse lagrangienne aléatoire sans mémoire temporelle (décorrélé instantanément), ainsi le déplacement quadratique moyen des particules croît linéairement avec le temps.

En conclusion, cette première méthode d'estimation repose sur le suivi de la croissance de la variance mais étant très simpliste elle n'est pas toujours valable dans les écoulements océaniques de fine-échelle.

## 2.2 Résultats 1

Suivant cette technique on peut ainsi afficher la variance quadratique moyenne des vitesses pour une simulation de diffusion sur 9 jours d'un total de 35 912 particules réalisé sur logiciel Ariane, et obtenir cette courbe :

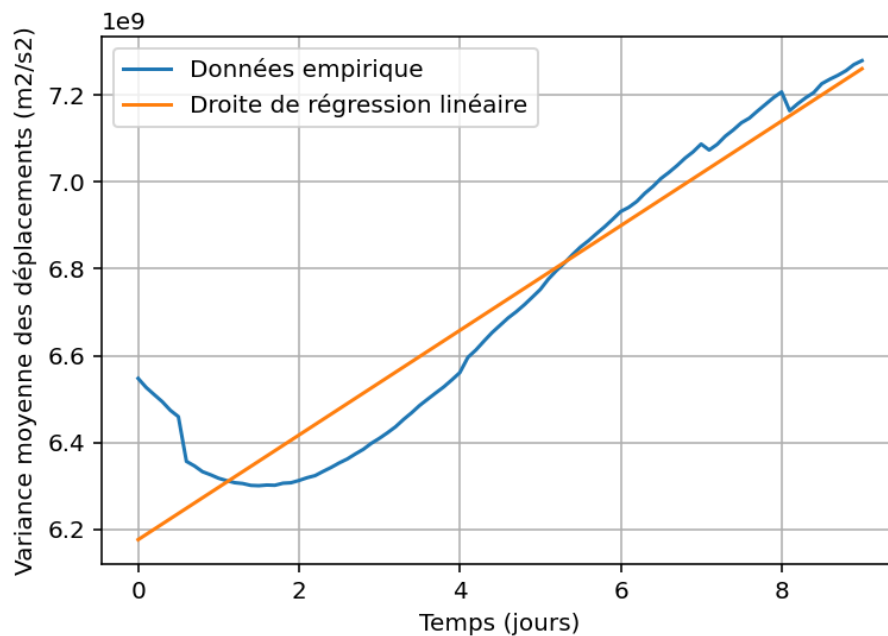


Figure 1: Relation entre la variance des vitesses et le temps dans un champ lagrangien décorrélé.

Puis grâce à la droite obtenu par régression on peut en récupérer la pente ce qui offre une valeur  $D_{eff} = 348.84 \text{ m}^2/s$ . Cette valeur est cohérente par rapport à ce qu'on peut retrouver en Méditerranée, avec des gammes de valeurs variant de  $100 \text{ m}^2/s$  à  $500 \text{ m}^2/s$  en surface et en mésoscale.

On peut maintenant, en se basant sur ce jeu de données, estimé la maille de grille grâce à la formule donnée dans l'introduction, et obtenir ainsi  $\Delta x = 10416.48 \text{ m}$  soit environ 0.1.

### 2.3 Validité 1

Maintenant, on peut essayer d'apporter un garde-fou à cette méthode pour s'assurer qu'elle fonctionne, car obtenir un résultat cohérent peut aussi être dû à une coïncidence ou à la chance. Pour cela on va simuler un mouvement brownien sur 1 000 particules pour 1 000 pas de temps, la variance de la marche aléatoire sera donné tel que :  $\sigma^2 = 2D_{sim} * dt$  avec  $dt = 0.1 \text{ s}$  et  $D_{sim} = 0.01 \text{ m}^2/s$ .

On considérera la méthode fiable si en reproduisant exactement le même protocole sur ce jeu de données simulé par mouvement brownien que sur les positions d'Ariane, on obtient une valeur  $D_{eff} = D_{sim} = 0.01 \text{ m}^2/s$ .

Après avoir relancé cette simulation 10 fois avec des seed différentes pour s'assurer de la constance du résultat final, on remarque que à chaque fois, la méthode renvoie une valeur de coefficient de diffusion similaire ou exact au coefficient initial. On conclut donc que la méthode 1 d'analyse de la variance des vitesses est fiable pour estimer une maille de grille optimal.

## 3 Méthode 2

### 3.1 Principe 2

Cette nouvelle méthode d'analyse et de détermination de pas d'espace optimal pour une étude océanographique en conditions lagrangienne se base sur un concept plus complexe que la méthode 1 qui est le temps d'autocorrélation qu'une masse d'eau qui se déplace va intégrer pour ajuster son mouvement. Ce temps est lié notamment aux structures océanique telle que les fronts ou les tourbillons qui sont courant quel que soit les échelles d'études. Ainsi cette méthode se veut plus réaliste et plus applicable à ce qui pourrait être observé en situation in situ.

Pour ce faire on va dans un premier temps obtenir les vitesses longitudinale et latitudinale U et V respectivement grâce aux positions récupéré dans le même jeu de donnée que celui utilisé dans la méthode 1 grâce à cette formule :

$$U_i, V_i(t) = \frac{X_i(t + \Delta t) - X_i(t)}{\Delta t}$$

Avec  $X_i$  les coordonnées longitude/latitude d'une particule  $i$  et  $\Delta t$  le pas de temps de la simulation, ici en seconde.

Une fois cela fait, on va établir la fonction d'autocorrélation comme décrite ci-dessous en fonction de  $\tau$ . Avec  $\tau$  représentant le lag temporelle par rapport à l'instant actuel, c'est une valeur qui évolue et grandit en fonction du temps, par exemple ici l'advection se déroule sur 9 jours avec un pas de temps  $\Delta t$  de 0.1 jour, il y aura donc 91 valeurs différentes de  $\tau$  allant de 0 à 0.9; et chacune de ces valeurs de  $\tau$  va être comparé à l'entiereté du temps de simulation.

$$R_i(\tau) = \frac{(\mathbf{U}_i(t) - \overline{\mathbf{U}}_i) \cdot (\mathbf{U}_i(t + \tau) - \overline{\mathbf{U}}_i)}{\sigma_i^2}$$

Attention, ici  $\mathbf{U}_i(t)$  représente la moyenne des vitesses  $U$  et  $V$  pour la particule  $i$ . Et  $\overline{\mathbf{U}}_i$  est la moyenne des vitesses combinées sur toute la simulation pour la particule  $i$ .

On peut ensuite moyenner  $R_i$  pour chaque instant  $\tau$  et en obtenir une courbe, pour ensuite affiner les résultats on fait une régression cubique sur ces données pour lisser l'intégration et obtenir le temps d'autocorrélation pour cette simulation grâce à cette formule :

$$T = \int_0^{\infty} R(\tau) d\tau$$

Enfin, on a toutes les valeurs nécessaires pour obtenir le coefficient de diffusion qui prend en compte cette notion d'autocorrélation tel que :

$$K_{def}(\tau) = \sigma^2 T \left[ 1 - \frac{T}{\tau} \left( 1 - e^{-\tau/T} \right) \right]$$

## 3.2 Résultats 2

Grâce à cette deuxième technique on peut dans un premier temps obtenir un affichage graphique de la fonction d'autocorrélation, avec une valeur  $R(\tau = 0) = 1$  signifiant que la particule se souvient entièrement des valeurs initiales qui lui sont attribuées puis qui décroît jusqu'à  $R(\tau = \infty) = 0$  quand  $\tau$  devient très grand.

On note que  $R$  devient ici négatif, ce qui est possible d'après l'article de référence mais pas aussi longtemps car un plateau est supposé se produire à  $R(\tau = \infty) = 0$ . Cependant ce n'est pas une erreur de programme ou informatique mais seulement un soucis de durée de simulation. (Cf partie Validité 2)

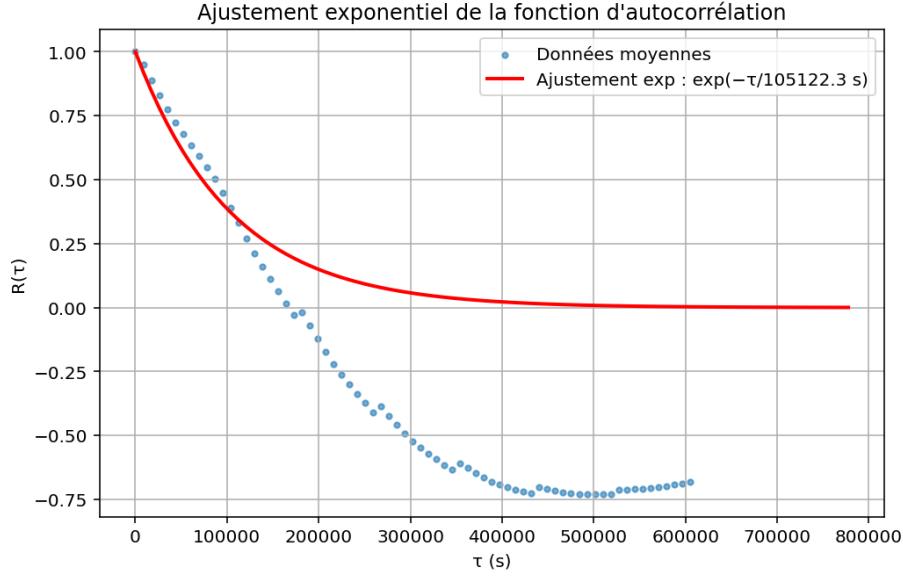


Figure 2: Courbe de tendance de la fonction d'autocorrélation

La partie négative est sûrement dû à des processus et structures physiques (comme des tourbillons ou le passage d'un front) qui vont renvoyer une valeur négative (donc sens contraire au sens initial à  $t=0$  selon le repère choisi). L'utilisation d'une régression exponentielle est d'ailleurs conseillé dans l'article de référence pour pouvoir obtenir un temps d'autocorrélation positif (et donc physiquement juste) tout en évitant d'utiliser des méthodes trop brusques comme la valeur absolu par exemple. On peut aussi faire une régression cubique sur les données et intégrer seulement les morceaux positif ou nuls.

On obtient donc un temps  $T$  d'autocorrélation tel que  $T = 105\,122$  secondes, soit 1.22 jours, ce qui est légèrement inférieur aux ordres de grandeur des articles scientifiques qui en parle (entre 1.5 et 3 jours en moyenne), mais cet écart est TRES probablement dû à un problème que je vais évoquer dans la partie 'Validité 2'.

Enfin, une fois  $T$  obtenu on peut également obtenir  $D_{eff} = 252.29\ m^2/s$  ainsi que par la suite la maille de grille  $\Delta x = 8859\ m$ . Pour cela on pourra utiliser une approximation de la formule de  $D_{eff}$  présenté précédemment. En effet avec un temps d'autocorrélation quasiment 9 fois inférieur au temps total de simulation, il est correct (et recommandé) de simplifier le calcul par  $D_{eff} = \sigma^2 T$ .

### 3.3 Validité 2

On peut enfin vérifier cette méthode avec un processus similaire à la méthode 1. Cependant, une marche aléatoire classique ne prend pas en compte la notion d'autocorrélation et n'aurait donc aucun intérêt pour vérifier le bon fonctionnement de cette méthode. C'est pourquoi on peut réaliser une simulation de déplacement de particule en suivant un processus d'Ornstein-Uhlenbeck, qui lui va prendre en compte l'autocorrélation et le lag temporelle entre chaque pas dans l'espace, ce qui rendra applicable la méthode que l'on cherche à tester.

Pour cela on va simuler les déplacements de 1 000 particules pendant 15 jours avec un pas de temps entre chaque déplacement de 0.05 jours. On aura aussi les paramètres initiaux suivants:  $T_{sim} = 86400$  s le temps d'autocorrélation des particules et  $D_{sim} = 864$   $m^2/s$  le coefficient de diffusion horizontal des particules simulées.

La encore, le but est d'utiliser exactement le même programme que pour les données issu d'Ariane et de retrouver à la fin des valeurs similaires ou exact à  $T_{sim}$  et  $D_{sim}$ .

Cependant, pour 10 simulations (afin d'augmenter la fiabilité de la conclusion), le programme renvoie des valeurs avec des écarts trop grands avec ce qui a été prédit. Cependant, les courbes obtenues laissent à penser que la méthode fonctionne car elles renvoient un plateau en 0 et peut importe la simulation toutes les courbes renvoyées sont très proche les unes des autres, d'autant plus que l'article de Provencale décrit ces calculs pour des temps longs de simulation.

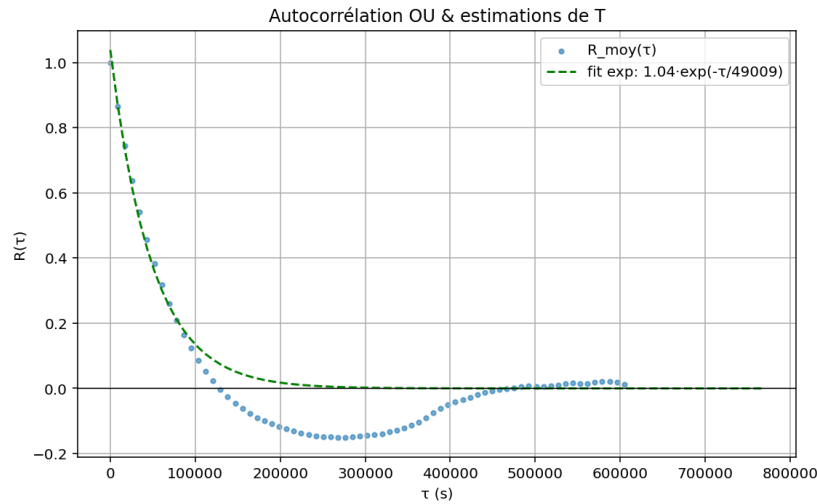


Figure 3: Courbe de tendance de la fonction d'autocorrélation pour une simulation OU.

Donc j'ai augmenté le temps de simulation petit à petit et à chaque test les valeurs de  $T$  et de  $D_{eff}$  se rapprochaient des valeurs attendues. Et ce n'est que pour un temps de simulation de 100 jours (soit 8 640 000 secondes) que pour chaque itérations des calcules et pour des seeds différentes, les écarts entre  $T$  et  $T_{sim}$  ainsi que  $D_{eff}$  et  $D_{sim}$  étaient toujours inférieur à 5%. On en déduit que la méthode 2 fonctionne et est fiable mais seulement pour des temps d'études suffisamment long.

(PS: les valeurs sous estimées par le manque de temps de simu l'étaient de environ 60% donc si on applique ce facteur 1.6 aux données de ariane on obtient ceci :

$T = 168\ 195\ \text{sec}$  (1.9 jours);  $D_{eff} = 403.6\ \text{m}^2/\text{s}$ ;  $\Delta x = 11195.9\ \text{m}$  (11 km.)

## 4 Méthode 3

### 4.1 Principe 3

Enfin, la troisième méthode repose sur le modèle de Smagorinsky qui permet d'identifier le coefficient de diffusion horizontal à partir du tensor de déformation horizontal grâce aux champs de cisaillement  $S$  et d'élongation/compression  $T$  suivant cette formule :

$$K_{smag} = (C_s \Delta)^2 \cdot |\mathbf{S}|$$

Avec  $C_s$  la constante de Smagorinsky et variant selon le domaine d'étude entre 0.05 et 0.2, pour la suite on prendra une valeur arbitraire satisfaisante pour de l'océanographie en Méditerranée tel que  $C_s = 0.17$ ;  $\Delta$  la résolution spatial du domaine d'étude tel que  $\Delta = \sqrt{dx^2 + dy^2}$ . De plus  $S$  et  $T$  décrit précédemment sont définis comme tel :

$$T = \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}, \quad S = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}.$$

Cette méthode d'analyse permet d'obtenir  $K_{smag}$  sous forme d'une matrice spatial de coefficient de diffusion, ainsi elle ne sera pas testé sur le même jeu de donnée que les deux méthodes précédente mais sur les données du produit L4 (MIOST) de SWOT avec une résolution spatial de 0.1°.

### 4.2 Résultats 3

On obtient comme rendu final une carte entre 4° et 6° de longitude et 40° et 42° de latitude illustrant les différentes valeurs de coefficient de diffusion dans l'espace et permettant de mettre en lumière les structures physiques présentes à la surface de la mer tel que les fronts ou les tourbillons.

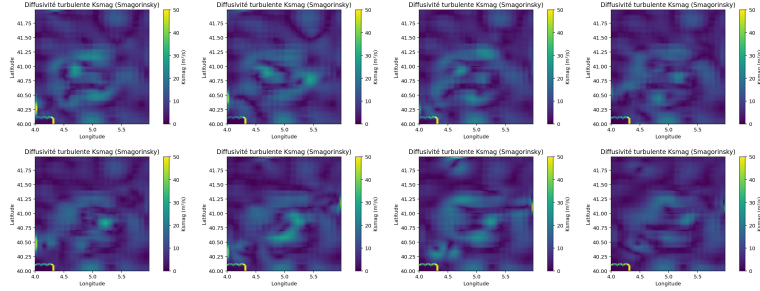


Figure 4: Carte des coefficients de diffusion horizontaux en  $m^2/s$  en mer Méditerranée issu de SWOT de 0 à 8 jours

Cependant on remarque que les valeurs de  $K_{smag}$  sont relativement basses, avec une moyenne d'une dizaine de  $m^2/s$  en moyenne sur l'ensemble de la carte, ce qui reste étrange au vu de l'ordre de grandeur que l'on serait supposé obtenir pour cette région,

L'erreur doit venir en réalité des ordres de grandeur des champs T et S, qui sont de l'ordre de  $10^{-7}$  en moyenne, alors que pour obtenir des valeurs finales qui seraient dans les gammes attendues il faudrait que ces champs soient soumis à un facteur 1 000 pour retourner sur des ordres de grandeur de  $10^{-4}$ . Les autres paramètres sont en effet la constante  $C_s$  mais on peut difficilement augmenter plus cette valeur car elle ne serait alors plus représentative des notions physique de la région d'étude. De plus  $\Delta$  est bien égal à environ 10 km comme l'indique la résolution du paquet de donnée de SWOT, ce qui implique que si erreur il y a, elle doit forcément venir des champs T et S.

On peut aussi tracer un histogramme représentant les valeurs de coefficients de diffusion de la zone pour se représenter leurs fréquence d'occurrence, même si bien sûr les valeurs en soit sont surement largement sous-estimées;

On remarque une fois de plus bien que les petites valeurs dominent la carte avec très peu de cellules avec de grandes valeurs de diffusion.

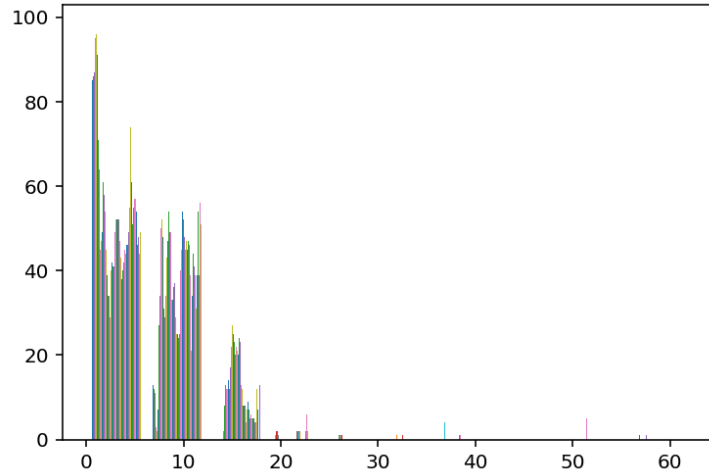


Figure 5: Histogramme des coefficients de diffusion horizontaux en  $m^2/s$  en mer Méditerranée issu de SWOT, abscisse les coefficients et ordonnées le nombre de cellules de l'espace correspondant

### 4.3 Validité 3

Pour ce qui est du test de validité je n'ai pas eu d'idée précise pour tester la fiabilité de la méthode, d'autant plus qu'elle ne "marche pas".

Donc j'ai tenté de faire tourner ce code sur le jeu de donnée simulé par Ariane qui a été testé par les méthodes 1 et 2, pour cela il fallait une grille d'espace préalable (pour déterminer  $\Delta$ , mais comme les méthodes précédentes renvoyaient une valeur fiable de grille je l'ai utilisé ici, donc  $\Delta = 10000 m$ , et je reprends  $C_s = 0.17$  ici aussi.

Je construis des champs de vitesse U et V à partir des positions comme pour la méthode 1 et je filtre dynamiquement les NaN avec une fonction python KD-Tree (ce qui me permet d'ignorer les NaN qui sont très présents dans ces données là tout en gardant le plus d'information possible) et j'obtiens les gradients de vitesses par méthode des moindres carrés, ensuite il me suffit de moyenner  $K_{smag}$  sur chaque instant du temps pour toutes les particules du paquet et obtenir un graphique qui renvoie les valeurs maximal et moyenne de  $K_{smag}$  pour chaque pas de temps, ici l'échelle est en jours.

On voit que la valeur moyenne  $K_{smag} = 17 m^2/s$  ne varie que très peu dans le temps mais que la valeur maximal de  $K_{smag}$  s'emballe à la fin de la série temporelle, cela est dû au manque de donnée et la présence de trop de NaN qui réduisent l'efficacité du calcul.

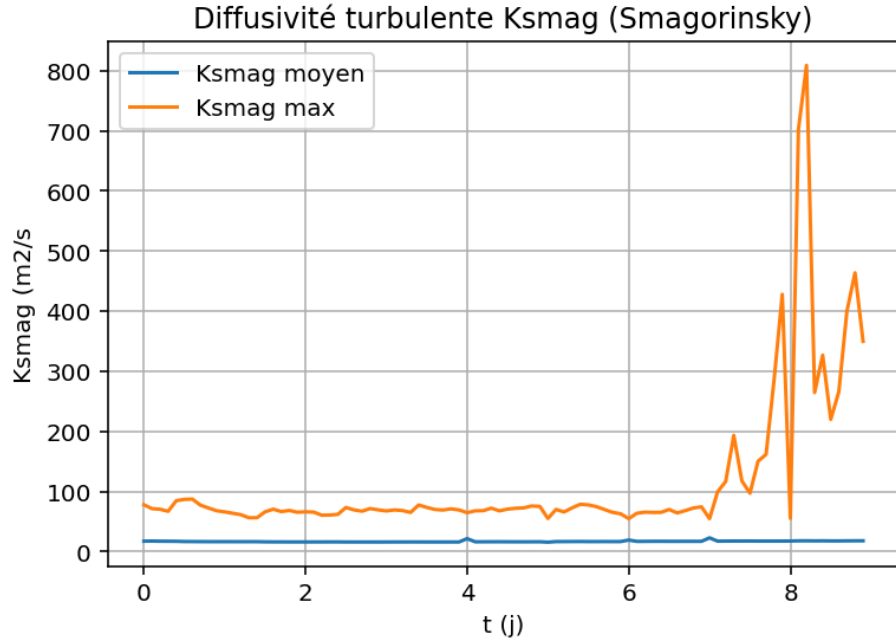


Figure 6: Graphique illustrant la valeur moyenne et maximal de  $K_{smag}$  dans le jeu de donnée d'Ariane à chaque pas de temps

On en déduit que la méthode fonctionne et renvoie les mêmes valeurs de coefficient de diffusion et de maille de grille ( $\Delta x = 2500 \text{ m}$  pour les données SWOT 0.1° et Ariane (ce qui est logique car elles représentent la même région au même moment mais de manière différente), mais renvoie les mauvaises valeurs de diffusion ET de maille de grille, qui devrait être autour de celle trouvé dans la méthode 1. Ce qui implique que la méthode fonctionne et est fiable mais qu'une erreur est présente en son sein. Cependant j'ai bien vérifié que la fonction `np.gradient` renvoie bien les dérivées partielles et j'ai aussi testé la méthode d'euler (implicite et explicite) sur des fonctions simples et elles renvoient la même chose que `np.gradient`, donc le calcul des champs T et S semble finalement être correct.

Après quelques recherches il semblerait que l'erreur vienne de l'échelle de résolution spatial de la zone. D'après les références scientifiques sur le sujet, cette méthode est surtout utilisé sur de très fine échelle et plus souvent pour les sciences atmosphériques. Sans détaillé ici les exigences mathématiques pour que la méthode fonctionne correctement, il semblerait que la résolution spatial, ici  $\Delta = 10000 \text{ m}$ , est littéralement 10 000x trop élevé, ce qui explique les erreurs dans les champs de cisaillement, d'élongation et de compression.

En somme, la méthode 3 est une méthode fiable et fonctionnels pour déterminer un coefficient de diffusion et une maille de grille mais seulement sous certaines conditions, qui ne sont ici pas respectées.

## 5 Conclusion

Très brièvement, on conclut que par rapport aux dispositions relatives aux données disponibles (temps d'étude, résolution spatiale et temporelle, location, etc...) seule la méthode 1 fonctionne vraiment correctement. En effet il faudrait plus de temps d'étude pour utiliser correctement la méthode 2 et une maille de grille plus petite pour utiliser la méthode 3.

On validera quand même au final  $\Delta x = 10 \text{ km}$  environ pour étudier les processus et propriétés de la zone sans altération !

## 6 Références

### 6.1 Données coeff diffu et T

Doglioli and Gregori, 2023. BioSWOT-Med cruise, RV L'Atalante.

Griffa et al. (2004)

LaCasce (2008)

### 6.2 Resolution Smagorinsky

On the model coefficients for the standard and the variational multi-scale Smagorinsky model. - J.Meyers et al. (2006)