

Modélisation numérique : création d'un modèle océanique en 2D avec frontière ouverte et une île suivi d'une étude de cas dans le Golfe de Corée Orientale

Josselin Fatah-Roux

Université de Toulon
Master 2 Physique Science de l'Ingénieur
Spécialité Physique Surveillance de l'Environnement
vufic@outlook.com

21 décembre 2017

Résumé

De nos jours l'importance des modèles océaniques n'est plus à prouver. À la pointe de la technologie ils sont une source d'information incroyable sur la dynamique des océans. Cependant il est essentiel de ne pas perdre de vue toute la complexité qui peut se trouver derrière de tels outils.

Ce travail consiste à créer un modèle océanique 2D avec frontière ouverte et une île. Dans un premier temps on le réalisera en Lagrangien puis dans un second temps en Eulérien. On l'appliquera enfin au Golfe de Corée Orientale comme étude de cas. Les résultats entrepris ici montrent principalement la présence d'une zone tourbillonnaire .

Mots-clés

Océanographie; Modèle numérique 2D; Équations de Kowalik; Processus océanique; Méthode Eulérienne; Méthode Lagrangienne; Golfe de Corée Orientale

1 Introduction

La modélisation est la représentation d'un système complexe par un autre beaucoup plus simple à comprendre. Elle est très utile dans le domaine de l'océanographie pour prédire la circulation océanique

mais également pour obtenir les principales variables physiques de cette circulation (température, salinité, vitesse du courant ...) permettant alors de comprendre les mécanismes qui régissent l'océan (physique et biologique).

Un modèle océanique est dit modèle numérique car basé sur un ensemble de fonctions mathématiques (pas forcément linéaire) dont la résolution se fait sur ordinateur, par conséquent sur un nombre fini de valeurs discrètes, grâce à une méthode numérique. Il existe différentes méthodes numériques dont la plus utilisée est celle des différences finies de part son coût, son efficacité, sa simplicité et sa facilité. Ainsi les modèles d'océan qui l'utilise se distinguent d'abord par leur système d'équation, puis leur discrétisation. Toutefois tous comportent les équations primitives qui sont à la base de la mécanique des fluides : conservation de la masse, advection et Navier-Stokes avec approximation hydrostatique et de Boussinesq.

L'intérêt de tels modèles est qu'ils répondent à de nombreuses questions liées à l'environnement et la sécurité. En effet, pouvoir prédire permet à l'homme de mieux se préparer à des situations futures. De ce fait connaître par avance l'état de mer reviendrait à prévoir les tsunamis, les tempêtes, les inondations, les marées noires et bien d'autres phénomènes. Dès lors il est possible de mieux gérer la surveillance et la sécurité maritimes.

Table des matières

1	Introduction	1
2	Matériels et Méthodes	3
2.1	Rappels	3
2.2	Modélisation du courant (\bar{u} et \bar{v})	6
2.2.1	Équation de Kowalik	6
2.2.2	Stress de surface dû au vent	6
2.2.3	Équation de Kowalik en différences finies	7
2.2.4	Dimensionnement du futur modèle	7
2.2.5	Détermination de la vitesse du vent	7
2.2.6	Résolution de l'équation de Kowalik en différences finies	9
2.2.7	Les équations du transport en différences finies	9
2.2.8	Création d'îles ou de côtes	9
2.2.9	Frontières fermées et frontière ouverte	10
2.2.10	Obtention de NaN (couleur blanche) sur les île/côtes	12
2.3	Description Lagrangienne	12
2.4	Description Eulérienne	13
2.5	Application dans le Golfe de Corée Orientale	14
3	Résultats	15
3.1	Courantologie de surface	15
3.2	Bouées flottantes positionnées à différents endroits	15
3.3	Masse d'eau appliquée à tout le domaine	16
3.4	Masse d'eau entrante	17
4	Interprétation	17
5	Discussion	18
5.1	Équation de Kowalik trop simplifié	18
5.2	Modèle non vérifiable	18
5.3	Modèle en 2D	18
5.4	Les côtes	18
5.5	Le vent	18
6	Conclusion	18
7	Annexes	19
7.1	Réalisation du masque	19
7.2	Calcul de l'équation de Kowalik en différence finie	19
7.3	Calcul des équations du transport en différence finie	20
7.4	Graphes	20
7.5	Description Lagrangienne	21
7.6	Description Eulérienne	22
7.6.1	Masse d'eau appliquée à tout le domaine	22
7.6.2	Masse d'eau entrante	24
7.6.3	plot_controle_masse.m	26
7.7	suplabel.m	28

2 Matériels et Méthodes

2.1 Rappels

Équations primitives :

- ★ Navier-Stokes : $\frac{\partial \vec{V}}{\partial t} + (\vec{V} \cdot \nabla) \vec{V} = -\frac{1}{\rho} \nabla P + 2\Omega \times \vec{V} + \vec{g} + K \Delta \vec{V}$
- ★ Conservation de la température : $\frac{\partial T}{\partial t} + \nabla(\vec{V}T) = \nabla(K_T \nabla T) + \underbrace{(Sources - Pertes)}_{\text{Flux de chaleur}}$
- ★ Conservation de la salinité : $\frac{\partial S}{\partial t} + \nabla(\vec{V}S) = \nabla(K_T \nabla S) + \underbrace{(Sources - Pertes)}_{\text{Précipitation/évaporation+fleuves}}$
- ★ État de l'eau de mer : $\rho = f(T, S, z)$

Méthode des différences finies :

Soit Ψ une fonction qui dépend de x et de y :

- ★ Approximation de la dérivée première de Ψ par rapport à x :

- Décentrage amont :

$$\begin{aligned} \Psi(x + \Delta x, y) &= \Psi(x, y) + \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} + \dots \\ \Leftrightarrow \Psi(x + \Delta x, y) - \Psi(x, y) - \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} &= \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial x} &= \frac{\Psi(x + \Delta x, y) - \Psi(x, y)}{\Delta x} - \underbrace{\frac{(\Delta x)^1}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2}}_{\text{Erreur de troncature}} + o(\Delta x^2) \end{aligned}$$

Erreur de troncature = $o(\Delta x)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial x} = \frac{\Psi(x + \Delta x, y) - \Psi(x, y)}{\Delta x} + o(\Delta x)$$

$$\text{Pour } \Delta x \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial x} \approx \frac{\Psi(x + \Delta x, y) - \Psi(x, y)}{\Delta x}}$$

- Décentrage aval :

$$\begin{aligned} \Psi(x - \Delta x, y) &= \Psi(x, y) - \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} - \dots \\ \Leftrightarrow \Psi(x, y) - \Psi(x - \Delta x, y) + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} &= \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial x} &= \frac{\Psi(x, y) - \Psi(x - \Delta x, y)}{\Delta x} + \underbrace{\frac{(\Delta x)^1}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2}}_{\text{Erreur de troncature}} + o(\Delta x^2) \end{aligned}$$

Erreur de troncature = $o(\Delta x)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial x} = \frac{\Psi(x, y) - \Psi(x - \Delta x, y)}{\Delta x} + o(\Delta x)$$

$$\text{Pour } \Delta x \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial x} \approx \frac{\Psi(x, y) - \Psi(x - \Delta x, y)}{\Delta x}}$$

- Centré :

$$\begin{aligned}\Psi(x + \Delta x, y) - \Psi(x - \Delta x, y) &= 2(\Delta x)^1 \frac{\partial \Psi(x, y)}{\partial x} + \frac{2(\Delta x)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + \dots \\ \Leftrightarrow \Psi(x + \Delta x, y) - \Psi(x - \Delta x, y) - \frac{2(\Delta x)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} &= 2(\Delta x)^1 \frac{\partial \Psi(x, y)}{\partial x} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial x} &= \frac{\Psi(x + \Delta x, y) - \Psi(x - \Delta x, y)}{2\Delta x} - \underbrace{\frac{2(\Delta x)^2}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + o(\Delta x^3)}_{\text{Erreur de troncature}}\end{aligned}$$

Erreur de troncature = $o(\Delta x^2)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial x} = \frac{\Psi(x + \Delta x, y) - \Psi(x - \Delta x, y)}{2\Delta x} + o(\Delta x^2)$$

$$\text{Pour } \Delta x \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial x} \approx \frac{\Psi(x + \Delta x, y) - \Psi(x - \Delta x, y)}{2\Delta x}}$$

- ★ Approximation de la dérivée seconde de Ψ par rapport à x :

- $\Psi(x + \Delta x, y) = \Psi(x, y) + \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + \frac{(\Delta x)^4}{4!} \frac{\partial^4 \Psi(x, y)}{\partial x^4} + \dots$
- $\Psi(x - \Delta x, y) = \Psi(x, y) - \frac{(\Delta x)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} - \frac{(\Delta x)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + \frac{(\Delta x)^4}{4!} \frac{\partial^4 \Psi(x, y)}{\partial x^4} - \dots$
- $\Psi(x + \Delta x, y) + \Psi(x - \Delta x, y) = 2\Psi(x, y) + \frac{(\Delta x)^2}{1!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{(\Delta x)^4}{12} \frac{\partial^4 \Psi(x, y)}{\partial x^4} + \dots$
- $\Leftrightarrow \frac{\partial^2 \Psi(x, y)}{\partial x^2} = \frac{\Psi(x + \Delta x, y) + \Psi(x - \Delta x, y) - 2\Psi(x, y)}{\Delta x^2} - \underbrace{\frac{(\Delta x)^2}{12} \frac{\partial^4 \Psi(x, y)}{\partial x^4} + o(\Delta x^4)}_{\text{Erreur de troncature}}$

Erreur de troncature = $o(\Delta x^2)$

$$\text{D'où } \frac{\partial^2 \Psi(x, y)}{\partial x^2} = \frac{\Psi(x + \Delta x, y) + \Psi(x - \Delta x, y) - 2\Psi(x, y)}{\Delta x^2} + o(\Delta x^2)$$

$$\text{Pour } \Delta x \rightarrow 0 : \boxed{\frac{\partial^2 \Psi(x, y)}{\partial x^2} \approx \frac{\Psi(x + \Delta x, y) + \Psi(x - \Delta x, y) - 2\Psi(x, y)}{\Delta x^2}}$$

- ★ Approximation de la dérivée première de Ψ par rapport à y :

- Décentrage amont :

$$\begin{aligned}\Psi(x, y + \Delta y) &= \Psi(x, y) + \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial y} + \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} + \dots \\ \Leftrightarrow \Psi(x, y + \Delta y) - \Psi(x, y) - \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} &= \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial y} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial y} &= \frac{\Psi(x, y + \Delta y) - \Psi(x, y)}{\Delta y} - \underbrace{\frac{(\Delta y)^1}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} + o(\Delta y^2)}_{\text{Erreur de troncature}}\end{aligned}$$

Erreur de troncature = $o(\Delta y)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial y} = \frac{\Psi(x, y + \Delta y) - \Psi(x, y)}{\Delta y} + o(\Delta y)$$

$$\text{Pour } \Delta y \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial y} \approx \frac{\Psi(x, y + \Delta y) - \Psi(x, y)}{\Delta y}}$$

- Décentrage aval :

$$\begin{aligned} \Psi(x, y - \Delta y) &= \Psi(x, y) - \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial y} + \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} - \dots \\ \Leftrightarrow \Psi(x, y) - \Psi(x, y - \Delta y) + \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} &= \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial y} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial y} &= \frac{\Psi(x, y) - \Psi(x, y - \Delta y)}{\Delta y} + \underbrace{\frac{(\Delta y)^1}{2!} \frac{\partial^2 \Psi(x, y)}{\partial y^2} + o(\Delta y^2)}_{\text{Erreur de troncature}} \end{aligned}$$

Erreur de troncature = $o(\Delta y)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial y} = \frac{\Psi(x, y) - \Psi(x, y - \Delta y)}{\Delta y} + o(\Delta y)$$

$$\text{Pour } \Delta y \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial y} \approx \frac{\Psi(x, y) - \Psi(x, y - \Delta y)}{\Delta y}}$$

- Centré :

$$\begin{aligned} \Psi(x, y + \Delta y) - \Psi(x, y - \Delta y) &= 2(\Delta y)^1 \frac{\partial \Psi(x, y)}{\partial y} + \frac{2(\Delta y)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial y^3} + \dots \\ \Leftrightarrow \Psi(x, y + \Delta y) - \Psi(x, y - \Delta y) - \frac{2(\Delta y)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial y^3} &= 2(\Delta y)^1 \frac{\partial \Psi(x, y)}{\partial y} + \dots \\ \Leftrightarrow \frac{\partial \Psi(x, y)}{\partial y} &= \frac{\Psi(x, y + \Delta y) - \Psi(x, y - \Delta y)}{2\Delta y} - \underbrace{\frac{2(\Delta y)^2}{3!} \frac{\partial^3 \Psi(x, y)}{\partial y^3} + o(\Delta y^3)}_{\text{Erreur de troncature}} \end{aligned}$$

Erreur de troncature = $o(\Delta y^2)$

$$\text{D'où } \frac{\partial \Psi(x, y)}{\partial y} = \frac{\Psi(x, y + \Delta y) - \Psi(x, y - \Delta y)}{2\Delta y} + o(\Delta y^2)$$

$$\text{Pour } \Delta y \rightarrow 0 : \boxed{\frac{\partial \Psi(x, y)}{\partial y} \approx \frac{\Psi(x, y + \Delta y) - \Psi(x, y - \Delta y)}{2\Delta y}}$$

★ Approximation de la dérivée seconde de Ψ par rapport à x :

- $\Psi(x, y + \Delta y) = \Psi(x, y) + \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{(\Delta y)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + \frac{(\Delta y)^4}{4!} \frac{\partial^4 \Psi(x, y)}{\partial x^4} + \dots$

$$\begin{aligned} \Psi(x, y - \Delta y) &= \Psi(x, y) - \frac{(\Delta y)^1}{1!} \frac{\partial \Psi(x, y)}{\partial x} + \frac{(\Delta y)^2}{2!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} - \frac{(\Delta y)^3}{3!} \frac{\partial^3 \Psi(x, y)}{\partial x^3} + \frac{(\Delta y)^4}{4!} \frac{\partial^4 \Psi(x, y)}{\partial x^4} - \dots \end{aligned}$$

$$\Psi(x, y + \Delta y) + \Psi(x, y - \Delta y) = 2\Psi(x, y) + \frac{(\Delta y)^2}{1!} \frac{\partial^2 \Psi(x, y)}{\partial x^2} + \frac{(\Delta y)^4}{12} \frac{\partial^4 \Psi(x, y)}{\partial x^4} + \dots$$

$$\Leftrightarrow \frac{\partial^2 \Psi(x, y)}{\partial y^2} = \frac{\Psi(x, y + \Delta y) + \Psi(x, y - \Delta y) - 2\Psi(x, y)}{\Delta y^2} - \underbrace{\frac{(\Delta y)^2}{12} \frac{\partial^4 \Psi(x, y)}{\partial y^4} + o(\Delta y^4)}_{\text{Erreur de troncature}}$$

Erreur de troncature = $o(\Delta y^2)$

$$\text{D'où } \frac{\partial^2 \Psi(x, y)}{\partial y^2} = \frac{\Psi(x, y + \Delta y) + \Psi(x, y - \Delta y) - 2\Psi(x, y)}{\Delta y^2} + o(\Delta y^2)$$

$$\text{Pour } \Delta y \rightarrow 0 : \boxed{\frac{\partial^2 \Psi(x, y)}{\partial y^2} \approx \frac{\Psi(x, y + \Delta y) + \Psi(x, y - \Delta y) - 2\Psi(x, y)}{\Delta y^2}}$$

Conditions aux limites :

On note : $C(x) = C_i$, $C(x + \Delta x) = C_{i+1}$ et $C(x - \Delta x) = C_{i-1}$

★ Dirichlet :

$C_{i-1} = C_0$ et $C_{i+1} = C_N$: on fixe une valeur aux frontières

★ Neumann :

$$\left. \frac{\partial C}{\partial x} \Big|_{i=1} = 0 \text{ et } \frac{\partial C}{\partial x} \Big|_{i=N} = 0 \right\} : \text{ on fixe le gradient (0=fermeture des frontières)}$$

2.2 Modélisation du courant (\bar{u} et \bar{v})

2.2.1 Équation de Kowalik

L'équation de Kowalik en utilisant l'approximation de Boussinesq et hydrostatique s'écrit :

$$R1\Delta\Psi + \frac{f}{H}J(H, \Psi) - \frac{R1}{H}(\nabla H \nabla \Psi) + \beta \frac{\partial \Psi}{\partial x} = \nabla \times \vec{\tau} + \frac{1}{H} \left(\frac{\partial H}{\partial x} \tau_y - \frac{\partial H}{\partial y} \tau_x \right)$$

-R1 est un scalaire (coefficient de frottements).

-J est l'opérateur Jacobien.

-\beta est la variation latitudinale du paramètre de Coriolis f.

-\nabla \times \vec{\tau} est le stress de surface dû au vent.

-H est la profondeur.

H étant décidée constante l'équation est réduite à :

$$R1\Delta\Psi + \beta \frac{\partial \Psi}{\partial x} = \nabla \times \vec{\tau}$$

De plus à l'échelle considérée il n'y a pas de grosse variation de Coriolis ce qui amène à prendre un $\beta=0$. D'où l'équation finale :

$$\boxed{R1\Delta\Psi = \nabla \times \vec{\tau}}$$

2.2.2 Stress de surface dû au vent

La contrainte tangentielle à la surface de la mer est due au vent. Le choix se porte uniquement sur un vent selon l'axe des x. Mathématiquement on l'exprime ainsi :

$$\boxed{\tau_x = \rho_a c_d w_x^2}$$

où ρ_a est la densité de l'air ($1.23 \text{ kg} \cdot \text{m}^{-3}$), c_d le coefficient de résistance aérodynamique (10^{-3}) et w_x la vitesse du vent sur x qui sera déterminé un peu plus loin.

2.2.3 Équation de Kowalik en différences finies

En appliquant ce qui a été dit sur le stress de surface :

$$\begin{aligned}
 R1\Delta\Psi &= \nabla \times \vec{\tau} \\
 \Leftrightarrow R1\left(\frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2}\right) &= \frac{\partial\tau_x(y)}{\partial x} - \underbrace{\frac{\partial\tau_y(x)}{\partial y}}_{\text{Disparaît car le vent uniquement sur x}} \\
 \Leftrightarrow R1\left(\frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2}\right) &= \frac{\partial\tau_x}{\partial x}
 \end{aligned}$$

En différences finies cela donne :

$$\begin{aligned}
 R1\left(\frac{\Psi_{i-1,j} - 2\Psi_{i,j} + \Psi_{i+1,j}}{\Delta y^2} + \frac{\Psi_{i,j-1} - 2\Psi_{i,j} + \Psi_{i,j+1}}{\Delta x^2}\right) &= \underbrace{\frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x}}_{\text{Centré pour des raisons de stabilité}} \\
 \Leftrightarrow \left(-\frac{2R1}{\Delta x^2} - \frac{2R1}{\Delta y^2}\right)\Psi_{i,j} &= -\frac{R1\Psi_{i-1,j}}{\Delta y^2} - \frac{R1\Psi_{i+1,j}}{\Delta y^2} - \frac{R1\Psi_{i,j-1}}{\Delta x^2} - \frac{R1\Psi_{i,j+1}}{\Delta x^2} + \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x} \\
 \Leftrightarrow \Psi_{i,j} &= \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1\Psi_{i-1,j}}{\Delta y^2} + \frac{R1\Psi_{i+1,j}}{\Delta y^2} + \frac{R1\Psi_{i,j-1}}{\Delta x^2} + \frac{R1\Psi_{i,j+1}}{\Delta x^2} - \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x}\right) \\
 \Leftrightarrow \Psi_{i,j} &= \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2}(\Psi_{i+1,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta x^2}(\Psi_{i,j+1} + \Psi_{i,j-1}) - \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x}\right)
 \end{aligned}$$

2.2.4 Dimensionnement du futur modèle

Les dimensions du modèles sont : $dimx = 200\Delta x$ et $dimy = 100\Delta y$ avec $\Delta x = \Delta y = 1km$.

Schématiquement on obtient :

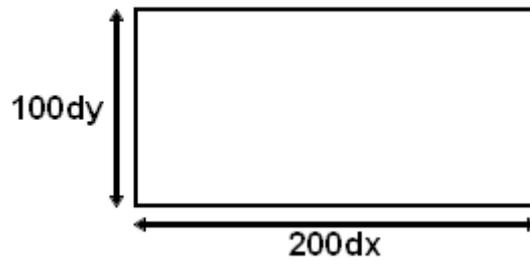


FIGURE 1 – Dimensions

2.2.5 Détermination de la vitesse du vent

Ce que l'on veut avant tout, c'est des iso-lignes de courant tourbillonnaire dont la valeur augmente en s'approchant du milieu.

En image :

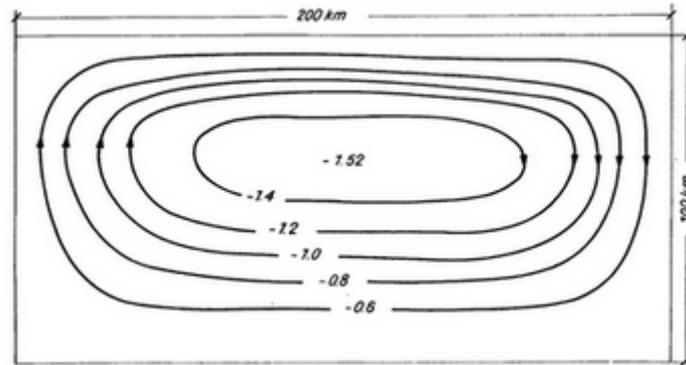


FIGURE 2 – Iso-lignes de courant tourbillonnaire. Les nombres sont en million de tonne par seconde (Source : Numerical Modelling of Marine Hydrodynamics)

Pour avoir ce résultat il faut que la vitesse du vent w_x soit une fonction affine partant de 2m/s allant jusqu'à 15m/s comme représenté ci-dessous :

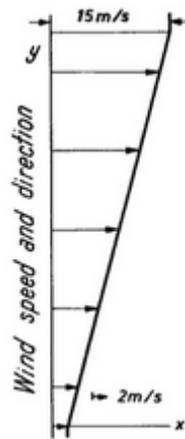


FIGURE 3 – Fonction affine w_x (Source : Numerical Modelling of Marine Hydrodynamics)

Par conséquent w_x s'écrit :

$$w_x(y) = ay + b.$$

Avec les dimensions du modèle donné en 2.5 on en déduit a et b :

$$\text{À } y=0, w_x(0) = 2m/s : 2 = a \times 0 + b \Leftrightarrow b = 2$$

$$\text{À } y=99, w_x(99) = 15m/s : 15 = 99a + 2 \Leftrightarrow a = \frac{13}{99}$$

D'où $w_x(y) = \frac{13}{99}y + 2$ (y de 0→99 par pas de 1 donne 100 valeurs)

2.2.6 Résolution de l'équation de Kowalik en différences finies

L'équation trouvée étant la suivante :

$$\Psi_{i,j} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2} (\Psi_{i+1,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta x^2} (\Psi_{i,j+1} + \Psi_{i,j-1}) - \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x} \right)$$

La solution de celle-ci dépend d'elle même. La résolution se fait donc par itération :

$$\Psi_{i,j} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2} (\Psi_{it_{i+1,j}} + \Psi_{i-1,j}) + \frac{R1}{\Delta x^2} (\Psi_{it_{i,j+1}} + \Psi_{i,j-1}) - \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x} \right)$$

Avec $\Psi_{i,j} = 0$ et $\Psi_{it_{i,j}}^0 = 0$ (initialisées toutes les deux à 0) $\xrightarrow{\text{on obtient}}$ $\Psi_{i,j}^0$

Or $\Psi_{it_{i,j}}^1 = \frac{\Psi_{it_{i,j}}^0 + \Psi_{i,j}^0}{2}$ d'où avec $\Psi_{i,j}^0$ et $\Psi_{it_{i,j}}^1 \xrightarrow{\text{on obtient}}$ $\Psi_{i,j}^1$

Ainsi $\Psi_{it_{i,j}}^2 = \frac{\Psi_{it_{i,j}}^1 + \Psi_{i,j}^1}{2}$ d'où avec $\Psi_{i,j}^1$ et $\Psi_{it_{i,j}}^2 \xrightarrow{\text{on obtient}}$ $\Psi_{i,j}^2$

Le dernier résultat donnant la solution recherchée :

$$\boxed{\Psi_{i,j} sol = \Psi_{i,j}^N}$$

Bien évidemment plus N sera grand, meilleure sera la convergence.

2.2.7 Les équations du transport en différences finies

Les équations du transport sont :

$$M_x = \frac{\partial \Psi}{\partial y} = \int_H^z u \rho dz$$

$$M_y = \frac{\partial \Psi}{\partial x} = \int_H^z v \rho dz$$

En différences finies avec un schéma centré (toujours pour des raisons de stabilité) :

$$M_x = \frac{\Psi_{i+1,j} - \Psi_{i-1,j}}{2\Delta y}$$

$$M_y = \frac{\Psi_{i,j-1} - \Psi_{i,j+1}}{2\Delta x}$$

Dans le cas d'un transport moyen :

$$\bar{u} = \frac{M_x}{\rho H}$$

$$\bar{v} = \frac{M_y}{\rho H}$$

2.2.8 Création d'îles ou de côtes

La réalisation d'îles ou de côtes nécessite une matrice de même dimension que $\Psi_{i,j} sol$ remplie de 0 et de 1 (appelée masque). Les 0 représentant la terre et les 1 l'eau. La multiplication terme à terme entre celle-ci et $\Psi_{i,j} sol$ permet l'intégration d'un ou plusieurs obstacles dans le modèle. Exemple ci-dessous :

$$\underbrace{\begin{pmatrix} 0.8 & 0.7 & 0.1 & 0.3 & 0.1 & 0.6 & 1 \\ 0.3 & 0.5 & 0.2 & 1 & 0.4 & 0.8 & 0.3 \\ 0.7 & 0.1 & 0.3 & 0.8 & 0.1 & 0.9 & 0.5 \\ 0.7 & 0.4 & 0.2 & 0.1 & 0.1 & 0.9 & 0.4 \\ 0.5 & 0.5 & 0.2 & 0.7 & 0.4 & 0.8 & 0.3 \end{pmatrix}}_{\Psi_{i,j}^{sol}} \times \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}}_{\text{île carré (bloc de 0)}} = \underbrace{\begin{pmatrix} 0.8 & 0.7 & 0.1 & 0.3 & 0.1 & 0.6 & 1 \\ 0.3 & 0.5 & 0 & 0 & 0 & 0.8 & 0.3 \\ 0.7 & 0.1 & 0 & 0 & 0 & 0.9 & 0.5 \\ 0.7 & 0.4 & 0 & 0 & 0 & 0.9 & 0.4 \\ 0.5 & 0.5 & 0.2 & 0.7 & 0.4 & 0.8 & 0.3 \end{pmatrix}}_{\text{île dans le modèle}}$$

2.2.9 Frontières fermées et frontière ouverte

Frontières fermées :

Fermer les frontières revient à ajouter un obstacle comme précédemment mais cette fois-ci uniquement sur le pourtour du modèle. Pour cela $\Psi_{i,j}$ est initialisé à 0 (effectué dans la partie 2.2.6) et l'itération ne doit alors se faire qu'en dehors de la première ligne et colonne sur x et y mais également de la dernière ligne et colonne sur x et y. Illustration :

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}=0} \text{ et } \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^0=0} \rightarrow \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^0 : 4 \text{ frontières fermées}}$$

$$\Psi_{i,j} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta x^2} (\Psi_{i+1,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta y^2} (\Psi_{i,j+1} + \Psi_{i,j-1}) - \frac{\tau_x(i,j+1) - \tau_x(i,j-1)}{2\Delta x} \right)$$

$$\frac{1}{2} \left(\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^0} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^0} \right) = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.1 & 0.05 & 0.3 & 0.4 & 0 \\ 0 & 0.05 & 0.15 & 0.4 & 0.35 & 0.45 & 0 \\ 0 & 0.2 & 0.1 & 0.1 & 0.2 & 0.45 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^1}$$

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^0} \text{ et } \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.1 & 0.05 & 0.3 & 0.4 & 0 \\ 0 & 0.05 & 0.15 & 0.4 & 0.35 & 0.45 & 0 \\ 0 & 0.2 & 0.1 & 0.1 & 0.2 & 0.45 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^1} \rightarrow \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{2,2} & \gamma_{2,3} & \gamma_{2,4} & \gamma_{2,5} & \gamma_{2,6} & 0 \\ 0 & \gamma_{3,2} & \gamma_{3,3} & \gamma_{3,4} & \gamma_{3,5} & \gamma_{3,6} & 0 \\ 0 & \gamma_{4,2} & \gamma_{4,3} & \gamma_{4,4} & \gamma_{4,5} & \gamma_{4,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^1}$$

$$\frac{1}{2} \left(\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_{2,2} & \gamma_{2,3} & \gamma_{2,4} & \gamma_{2,5} & \gamma_{2,6} & 0 \\ 0 & \gamma_{3,2} & \gamma_{3,3} & \gamma_{3,4} & \gamma_{3,5} & \gamma_{3,6} & 0 \\ 0 & \gamma_{4,2} & \gamma_{4,3} & \gamma_{4,4} & \gamma_{4,5} & \gamma_{4,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^1} + \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0.1 & 0.05 & 0.3 & 0.4 & 0 \\ 0 & 0.05 & 0.15 & 0.4 & 0.35 & 0.45 & 0 \\ 0 & 0.2 & 0.1 & 0.1 & 0.2 & 0.45 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^1} \right) = \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} & \alpha_{2,5} & \alpha_{2,6} & 0 \\ 0 & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} & \alpha_{3,5} & \alpha_{3,6} & 0 \\ 0 & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} & \alpha_{4,5} & \alpha_{4,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^2}$$

$$\dots \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \beta_{2,2} & \beta_{2,3} & \beta_{2,4} & \beta_{2,5} & \beta_{2,6} & 0 \\ 0 & \beta_{3,2} & \beta_{3,3} & \beta_{3,4} & \beta_{3,5} & \beta_{3,6} & 0 \\ 0 & \beta_{4,2} & \beta_{4,3} & \beta_{4,4} & \beta_{4,5} & \beta_{4,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_{\Psi_{i,j}^{sol} = \Psi_{i,j}^N \text{ avec frontières fermées}}$$

Frontière ouverte :

Il est évident que si l'on veut une frontière ouverte il est alors question d'enlever la ligne ou colonne de 0 adéquate en laissant l'itération se faire sur celle-ci. Toutefois le calcul ne peut se faire sans la valeur $\dim(x \text{ ou } y)+1$ ou $\dim(x \text{ ou } y)-1$ (tout dépend où se trouve la frontière ouverte) car justement la matrice $\Psi_{i,j}$ est limitée par ces dimensions $\dim x$ et $\dim y$. Ainsi il est important de rajouter une condition aux limites. Exemples :

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0.1 & 0.5 & 0.7 & 0.2 & 0.4 & 0 \end{pmatrix}$$

Condition :

$$\Psi_{\dim y, j} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2} (\Psi_{i,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta x^2} (\Psi_{i,j+1} + \Psi_{i,j-1}) - \frac{\tau_x(i, j+1) - \tau_x(i, j-1)}{2\Delta x} \right)$$

$$\begin{pmatrix} 0 & 0.1 & 0.5 & 0.7 & 0.2 & 0.4 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Condition :

$$\Psi_{1, j} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2} (\Psi_{i+1,j} + \Psi_{i,j}) + \frac{R1}{\Delta x^2} (\Psi_{i,j+1} + \Psi_{i,j-1}) - \frac{\tau_x(i, j+1) - \tau_x(i, j-1)}{2\Delta x} \right)$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0 \\ 0.6 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0 \\ 0.7 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Condition :

$$\Psi_{i, 1} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta y^2} (\Psi_{i+1,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta x^2} (\Psi_{i,j+1} + \Psi_{i,j}) - \frac{\tau_x(i, j+1) - \tau_x(i, j)}{2\Delta x} \right)$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.2 & 0.1 & 0.6 & 0.8 & 0.1 \\ 0 & 0.1 & 0.3 & 0.8 & 0.7 & 0.9 & 0.6 \\ 0 & 0.4 & 0.2 & 0.2 & 0.4 & 0.9 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Condition :

$$\Psi_{i, \dim x} = \frac{1}{\frac{2R1}{\Delta x^2} + \frac{2R1}{\Delta y^2}} \left(\frac{R1}{\Delta x^2} (\Psi_{i+1,j} + \Psi_{i-1,j}) + \frac{R1}{\Delta y^2} (\Psi_{i,j} + \Psi_{i,j-1}) - \frac{\tau_x(i, j) - \tau_x(i, j-1)}{2\Delta x} \right)$$

Ajout d'une frontière fermée sur une frontière fermée :

L'ajout d'une frontière fermée sur une frontière fermée se situant à l'opposé du vent permet d'éviter l'écrasement du courant. Illustration :

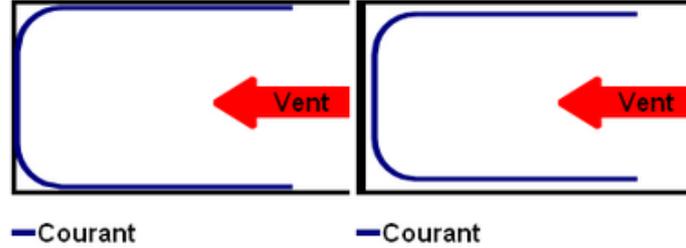


FIGURE 4 – À gauche sans double frontière et à droite avec double frontière

2.2.10 Obtention de NaN (couleur blanche) sur les île/côtes

Lorsque la matrice $\Psi_{i,j}sol$ a été multipliée par le masque pour intégrer les îles/côtes il est alors nécessaire de la re-multiplier par une matrice toto qui est la division terme à terme du masque par ce masque. Le résultat donne des NaN (couleur blanche) sur toutes les zones de terre.

2.3 Description Lagrangienne

L'approche lagrangienne est intéressante dans le sens où elle permet de suivre la dérive d'une ou plusieurs particules. Mathématiquement elle s'énonce comme suit :

$$\begin{cases} u = \frac{dx}{dt} \\ v = \frac{dy}{dt} \end{cases} \Leftrightarrow \begin{cases} xL(p, l) = xL(p, l-1) + dt * uL(xL(p, l-1), yL(p, l-1)) \\ yL(p, l) = yL(p, l-1) + dt * vL(xL(p, l-1), yL(p, l-1)) \end{cases} \text{ avec } \begin{cases} l : \text{le temps} \\ p : \text{la particule} \end{cases}$$

Si xL et yL sont en mètre il faut dès lors les diviser par dx (1000m) pour qu'ils entrent dans l'ordre de grandeur de la grille i et j initialement produite (celle-ci étant en km) :

$$\begin{cases} iL(p, l) = \frac{xL(p, l)}{dx} \\ jL(p, l) = \frac{yL(p, l)}{dy} \end{cases}$$

Au départ on a $xL(p, 1)$ et $yL(p, 1)$ qui sont des nombres entiers choisies et dont leur valeur correspond à la position de la particule p dans une grille i et j en mètre. En appliquant iL et jL on obtient les points i et j de la grille en km. Comme ils sont forcément eux aussi entiers il est possible d'avoir le \bar{u} et \bar{v} du $uL(p, 1)$ et $vL(p, 1)$ en cette position. Le $xL(p, 2)$ et $yL(p, 2)$ sont ensuite respectivement :

$$\begin{cases} xL(p, 2) = xL(p, 1) + dt * uL(p, 1) \\ yL(p, 2) = yL(p, 1) + dt * vL(p, 1) \end{cases}$$

Or $uL(p, 1)$ et $vL(p, 1)$ ne sont pas forcément entier et dans le cas où ils ne le sont pas cela nous amène à $xL(p, 2)$ et $yL(p, 2)$ non entier et donc à un $iL(p, 2)$ et $jL(p, 2)$ non entier. Ainsi il est impossible d'avoir le \bar{u} et \bar{v} du $uL(p, 2)$ et $vL(p, 2)$ car en dehors de tout point de la grille. Pour pallier à ce problème on passe par l'interpolation qui avec les données alentours reconstruit une valeur fiable :

$$\begin{cases} uL(p, l) = \text{interp2}(\bar{u}, jL(p, l), iL(p, l)) \\ vL(p, l) = \text{interp2}(\bar{v}, jL(p, l), iL(p, l)) \end{cases}$$

D'où :

$$\begin{cases} xL(p, l) = xL(p, l - 1) + dt * \text{interp2}(\bar{u}, jL(p, l - 1), iL(p, l - 1)) \\ yL(p, l) = yL(p, l - 1) + dt * \text{interp2}(\bar{v}, jL(p, l - 1), iL(p, l - 1)) \end{cases} \Leftrightarrow$$

$$\boxed{\begin{cases} xL(p, l) = xL(p, l - 1) + dt * \text{interp2}(\bar{u}, \frac{yL(p, l - 1)}{dy}, \frac{xL(p, l - 1)}{dx}) \\ yL(p, l) = yL(p, l - 1) + dt * \text{interp2}(\bar{v}, \frac{yL(p, l - 1)}{dy}, \frac{xL(p, l - 1)}{dx}) \end{cases}}$$

NB : interp2 est la fonction sur MATrix LABoratory permettant l'interpolation de données bidimensionnelle.

2.4 Description Eulérienne

Alors que la description lagrangienne est plus adaptée pour des objets d'étude dont la taille est suffisamment conséquente (déchets, méduses, bouées ...), la description eulérienne se focalise sur un ensemble de petites particules (concentration d'un fluide). Dans le premier cas on suit la particule (repère mobile), dans le second on observe la réaction du fluide (repère fixe). Par conséquent l'approche eulérienne s'exprime autrement que l'approche lagrangienne. Mathématiquement, elle s'énonce ainsi :

$$\frac{\partial C(x, y, t)}{\partial t} + \frac{\partial(u(x, y).C(x, y, t))}{\partial x} + \frac{\partial(v(x, y).C(x, y, t))}{\partial y} = S(x, y, t) \text{ avec } S(x, y, t) = 0$$

En différence finie :

$$\frac{C(i, j, l + 1) - C(i, j, l)}{\Delta t} + \underbrace{F_x + F_y}_{\text{Flux en x et y}} = 0$$

$$\Leftrightarrow C(i, j, l + 1) = C(i, j, l) - \Delta t(F_x + F_y)$$

$$\Leftrightarrow C(i, j, l + 1) = C(i, j, l) + \left[\frac{\Delta t}{\Delta x} (F_{x\text{aval}} + F_{x\text{amont}}) + \frac{\Delta t}{\Delta y} (F_{y\text{aval}} + F_{y\text{amont}}) \right]$$

Pour contrôler le flux aux frontières il est nécessaire de passer par une grille de type Arakawa C. Cela signifie que la grille des scalaires (u,v et c) est l'association d'une grille unew et vnew décalés d'un demi pas d'espace. La figure ci-dessous résume le fonctionnement :

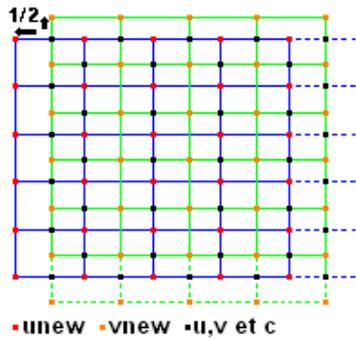


FIGURE 5 – Grille Arakawa C

NB : il est nécessaire d'ajouter une ligne et colonne à u_{new} et v_{new} afin que la grille scalaire est la même dimension que celles-ci initialement.

En suivant cette grille Arakawa C les flux x et y en amont et en aval valent pour :

$$\begin{cases} u_{new} > 0 \rightarrow F_{x,amont} = u_{new}(i, j) * C(i - 1, j, l - 1) \\ u_{new} < 0 \rightarrow F_{x,amont} = u_{new}(i, j) * C(i, j, l - 1) \end{cases}$$

$$\begin{cases} u_{new} > 0 \rightarrow F_{x,aval} = -u_{new}(i + 1, j) * C(i, j, l - 1) \\ u_{new} < 0 \rightarrow F_{x,aval} = -u_{new}(i + 1, j) * C(i + 1, j, l - 1) \end{cases}$$

$$\begin{cases} u_{new} > 0 \rightarrow F_{y,amont} = v_{new}(i, j) * C(i, j - 1, l - 1) \\ u_{new} < 0 \rightarrow F_{y,amont} = v_{new}(i, j) * C(i, j, l - 1) \end{cases}$$

$$\begin{cases} u_{new} > 0 \rightarrow F_{y,aval} = -v_{new}(i, j + 1) * C(i, j, l - 1) \\ u_{new} < 0 \rightarrow F_{y,aval} = -v_{new}(i, j + 1) * C(i, j + 1, l - 1) \end{cases}$$

Bilan de masse :

$$\text{Concentration totale : } C_{TOT} = \int C(x, y, z) dx dy dz$$

$$\text{Flux théorique : } Flux_{théo} = \frac{dC_{TOT}}{dt}$$

$$\text{Flux expérimentale : } Flux_{exp} = \int (u_{new} \text{ ou } v_{new}) \cdot C(x, y, z) (dx \text{ ou } dy) dz$$

Condition de stabilité :

$$\Delta t = \frac{\Delta x}{\max(|u_{new}|)}$$

2.5 Application dans le Golfe de Corée Orientale

Les programmes commentés sont disponibles dans la partie "Annexes". Ils ne sont pas explicites car construits selon la logique décrite dans "Matériels et Méthodes".

L'image pour effectuer le masque est la suivante :



FIGURE 6 – Golfe de Corée Orientale (en noir l'eau et en blanc les côtes)

Elle a été réalisée grâce au logiciel PhotoFiltre 6. Lors de son insertion sur MATrix LABo-ratory celle-ci est inversée ce qui amène à un stress de surface non plus de la droite vers la gauche mais de la gauche vers la droite.

NB : la direction et le sens de ce vent peut toujours être changé directement dans le code (Cf. 2.2.3 et 2.2.5).

3 Résultats

Les résultats sont seulement avec un vent de gauche à droite.

3.1 Courantologie de surface

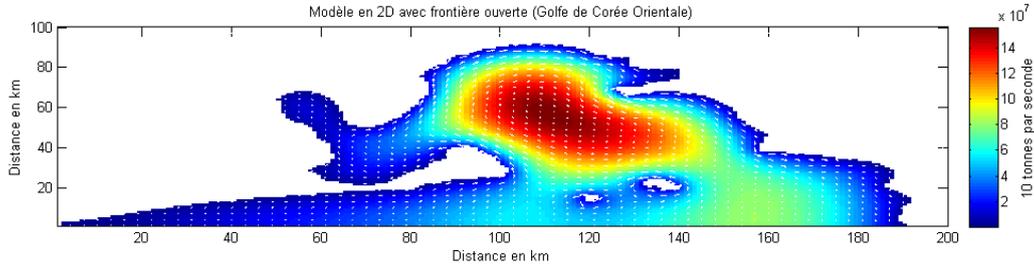


FIGURE 7 – Courant pour un vent de gauche à droite

On observe un très fort courant dans la zone rectangulaire de 100 à 140km sur l'axe des x et de 30 à 80km sur l'axe des y . Le reste du domaine étant beaucoup moins impacté.

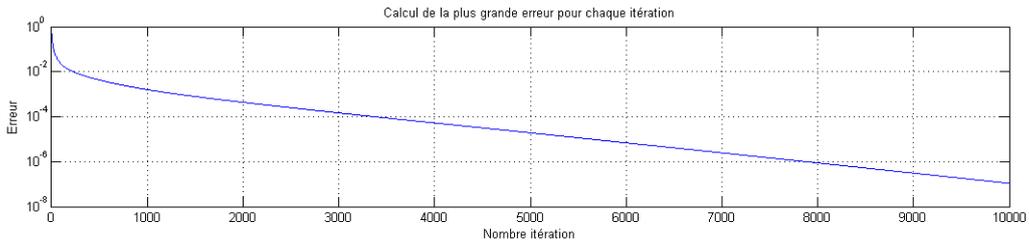


FIGURE 8 – Calcul de la plus grande erreur pour chaque itération

On constate qu'à 5000 itérations l'ordre de grandeur de l'erreur est d'environ 10^{-4} (relativement faible) ce qui indique une bonne convergence de la fonction de courant $\Psi_{i,j}$.

3.2 Bouées flottantes positionnées à différents endroits

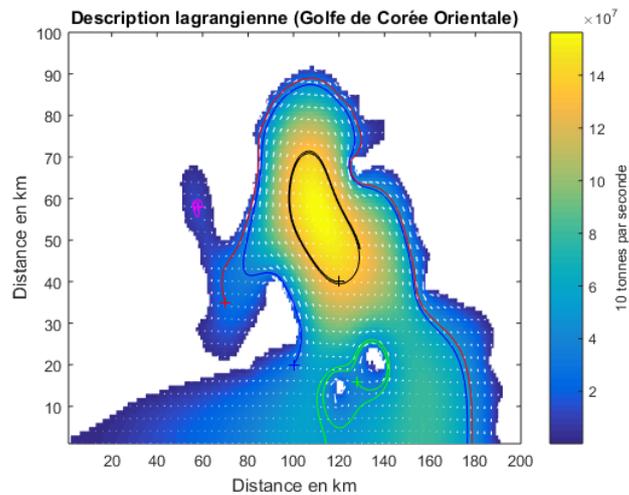


FIGURE 9 – Bouées placées à différents positions

La bouée rouge, bleu et noire confirme la présence d'un fort courant et montre l'existence d'un tourbillon.

La verte révèle ce qui semble être une convergence des courants provoquant une trajectoire elliptique de celle-ci autour des deux îles.

Quant à la magenta elle signale une zone où il n'y a que très peu d'échange - bouée ne bougeant quasiment pas.

3.3 Masse d'eau appliquée à tout le domaine

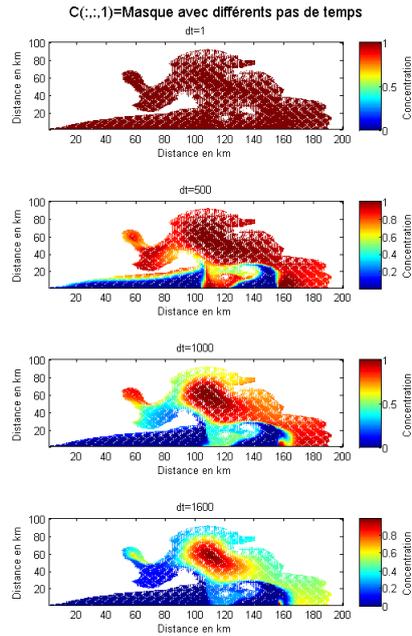


FIGURE 10 – Masse d'eau sur tout le domaine

En appliquant une masse d'eau à tout le domaine la partie dans le tourbillon reste piégée alors que l'autre est rapidement évacuée.

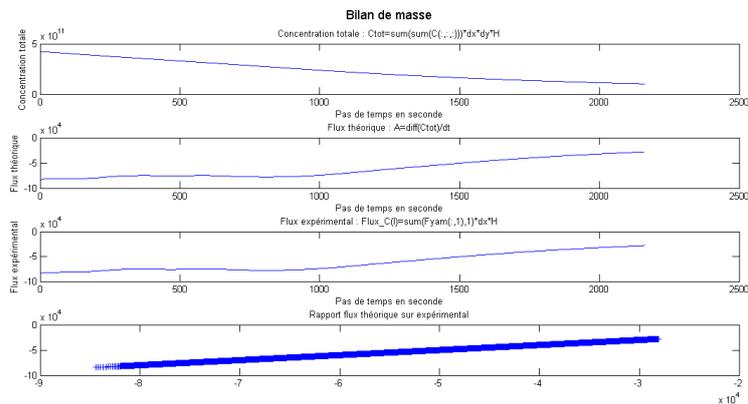


FIGURE 11 – Bilan de masse

Le graphique au-dessus confirme une baisse de la concentration totale dû à l'évacuation. De plus il indique une augmentation du flux théorique et expérimentale.

3.4 Masse d'eau entrante

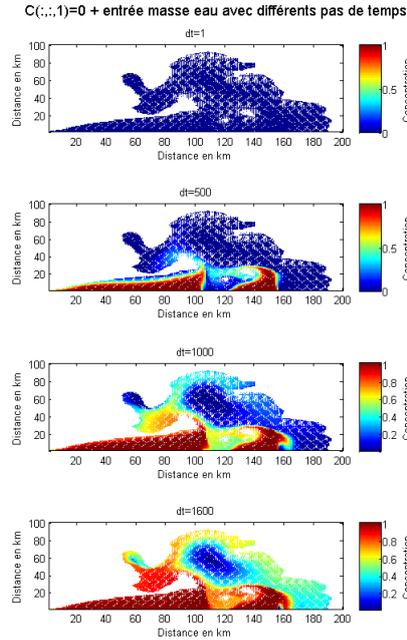


FIGURE 12 – Masse d'eau entrante

La masse d'eau entrante se diffuse sur tout le territoire hormis dans le tourbillon.

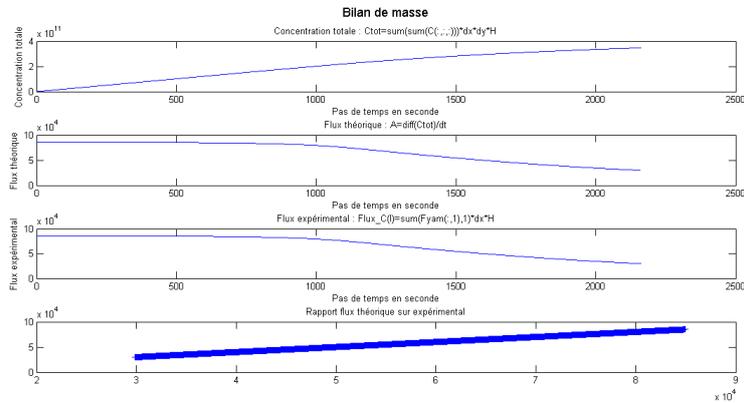


FIGURE 13 – Bilan de masse

Le graphique au-dessus confirme une augmentation de la concentration totale dû à l'entrée de la masse d'eau. Il indique également une baisse du flux théorique et expérimentale.

4 Interprétation

D'après les résultats obtenus précédemment on peut supposer que dans le cas d'une pollution immédiate sur tout le domaine la zone la plus impactée sera celle où se trouve le tourbillon. Cependant dans le cas d'une pollution lente entrante c'est l'inverse. Le tourbillon joue donc un rôle très important sur l'impact environnementale côtier de la région.

5 Discussion

5.1 Équation de Kowalik trop simplifié

L'équation de Kowalik a été énormément simplifiée particulièrement grâce à la profondeur choisie constante. Or la bathymétrie peut considérablement changer les résultats. De plus il avait été décidé de prendre la variation latitudinale du paramètre de Coriolis f nulle. Bien qu'à l'échelle considérée il n'y est pas de grosse variation il y en a tout de même des petites qui peuvent avoir un certain impact.

5.2 Modèle non vérifiable

Un modèle doit être jugé sur sa fiabilité, par comparaison aux données terrain, pour être le plus réaliste possible. Ainsi les simulations sont contrôlées par des observations. Or d'une part aucune mesures in situ n'a été fait à ce jours par manque de temps et de moyens. Et d'autre part malgré une recherche documentaire sur les courants du Golfe de Corée Orientale rien de probant n'a pu être trouvé. Le modèle ne peut donc pas être vérifié.

5.3 Modèle en 2D

Avec un modèle uniquement 2D la source d'information qu'il peut apporter est réduite contrairement à un modèle en 3D. D'où une complexité plus grande pour la compréhension de certains phénomènes.

5.4 Les côtes

Le tracé des côtes est plus que sommaire. En effet plusieurs zones difficiles - trop carrées et laissant le flux entrer dans les terre ou le bloquant - ont du être lissées. Des îles ont été également supprimées car très petites et donc trop pixelisées provoquant le même problème. Par conséquent tout cela à un impact négatif sur les résultats.

5.5 Le vent

Le vent appliqué n'est pas un vent typique de la région étudiée. De surcroît celui créé n'est qu'une fonction affine allant dans un seul sens uniquement. Il aurait fallu utiliser les données d'un modèle atmosphérique comme ARPEGE.

6 Conclusion

Dans ce projet il a été possible de constater la difficulté à produire un modèle numérique de circulation océanique crédible. La simplification des équations qui dans tout modèle est primordial - pour des raisons de limite des processeurs/de compréhension humaine - revient soit à négliger des variables qui semblent à première vue peu importante mais qui dans la réalité peuvent avoir de grands impacts (seule ou ensemble), soit à exprimer de manière approximative/empirique certains phénomènes qui sont mal compris (la turbulence par exemple) amenant ainsi à une modification non voulu des résultats. En conséquence il est essentiel de retenir que les modèles océaniques actuels sont le fruit d'un travail relativement complexe entre scientifique de différents bords.

Références

- [1] ZYGMUNT, KOWALIK. 1980. *Numerical Modelling of Marine Hydrodynamics*.
- [2] SIMON, MATHIAS. 2008. *Introduction au modèle ROMS : Application au golfe du Mexique et étude du «Loop Current»*.
- [3] STEELE, JOHN. 2001. *Encyclopedia of Ocean Sciences*.

7 Annexes

7.1 Réalisation du masque

```
%Creation des cotes :
I=imread('ile.jpg');
II=I(:,:,:)<30;
Masque=+II(:,:,1);
pcolor(Masque)
Masque(1,191:200)=0;
```

7.2 Calcul de l'équation de Kowalik en différence finie

```
%Les parametres :
R1=5e-6;
Rhoa=1.23;
Cd=1e-3;

dx=1000;
dy=1000;

dimX=200;
dimY=100;

y=(0:1:99);
Wy=(13/99)*y+2;
Taux=Rhoa*Cd*(Wy.^2);

Psiit=zeros(dimX,dimY);
Psi=zeros(dimX,dimY);

Alpha=1/((2*R1/(dx^2))+(2*R1/(dy^2)));
Betha=R1/(dx^2);
Gamma=R1/(dy^2);

Mx=zeros(dimX,dimY);
My=zeros(dimX,dimY);
Rhoeau=1025;
H=50;

%Iteration :
for k=1:10000
k

%Calcul de la matrice psi :
for i=2:dimX-1
for j=1:dimY-1

%Frontiere ouverte :
if j==1
Psi(i,j)=Alpha*(Betha*(Psi(i-1,j)+Psiit(i+1,j))...
+Gamma*(Psiit(i,j)+Psiit(i,j+1))...
-((Taux(j+1)-Taux(j))/(dy)));
else
Psi(i,j)=Alpha*(Betha*(Psi(i-1,j)+Psiit(i+1,j))...
+Gamma*(Psi(i,j-1)+Psiit(i,j+1))...
-((Taux(j+1)-Taux(j))/(dy)));
end
end
end
```

```

-((Taux(j+1)-Taux(j-1))/(2*dy)));
end

end
end

Psi=Psi.*Masque';

%Calcul de la plus grande erreur pour chaque iteration :
Dpsi(k)=max(max(abs((Psi-Psiit)./Psi)));

%Calcul de la matrice psiit :
Psiit=((Psiit+Psi)/2).*Masque';

end

```

7.3 Calcul des équations du transport en différence finie

```

%Transport :
for i=2:dimX-1

j=1;
Mx(i,j)=(Psi(i,j+1)-Psi(i,j))/(dy);
My(i,j)=-(Psi(i+1,j)-Psi(i-1,j))/(2*dx);

for j=2:dimY-1
Mx(i,j)=(Psi(i,j+1)-Psi(i,j-1))/(2*dy);
My(i,j)=-(Psi(i+1,j)-Psi(i-1,j))/(2*dx);
end
end

ub=Mx/(Rhoeau*H);
vb=My/(Rhoeau*H);

```

7.4 Graphes

```

xgraph=(1:dimX);
ygraph=(1:dimY);

toto=Masque./Masque;
Na=1;
figure(1)
clf
subplot(2,1,1)
pcolor(xgraph,ygraph,(abs(Psi).*toto)');c=colorbar;shading
↳ flat
title('Modele en 2D avec frontiere ouverte (Golfe de Coree
↳ Orientale)')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(c,'10 tonnes par seconde')
hold on
quiver(xgraph(1:Na:end),ygraph(1:Na:end),ub(1:Na:end,1:Na:end)
↳ ',vb(1:Na:end,1:Na:end) ','w');
subplot(2,1,2)
semilogy(Dpsi);grid on

```

```

title('Calcul de la plus grande erreur pour chaque iteration')
xlabel('Nombre iteration')
ylabel('Erreur')
Titre='Modele en 2D avec frontiere ouverte et calcul de la plus
↳ grande erreur pour chaque iteration';
[ax,h3]=suplabel(Titre,'t');
set(h3,'FontSize',13)

```

7.5 Description Lagrangienne

```

%Les parametres :
load('ilefo.mat')
dimt=1000;
dimp=5;
dt=2500;
xL=zeros(dimp,dimt);
yL=zeros(dimp,dimt);
iL=zeros(dimp,dimt);
jL=zeros(dimp,dimt);
xL(:,1)=[128000 100000 120000 70000 58000];
yL(:,1)=[16000 20000 40000 35000 58000];
iL(:,1)=xL(:,1)/dx;
jL(:,1)=yL(:,1)/dy;
uL=zeros(dimp,dimt);
vL=zeros(dimp,dimt);

for p=1:dimp
for l=2:dimt
uL(p,l)=interp2(ub,jL(p,l-1),iL(p,l-1));
vL(p,l)=interp2(vb,jL(p,l-1),iL(p,l-1));
xL(p,l)=xL(p,l-1)+dt*uL(p,l);
yL(p,l)=yL(p,l-1)+dt*vL(p,l);
iL(p,l)=xL(p,l)/dx;
jL(p,l)=yL(p,l)/dy;
end
end

figure(1)
clf
pcolor(xgraph,ygraph,(abs(Psi).*toto)');c=colorbar;shading
↳ flat
title('Description lagrangienne (Golfe de Coree Orientale)')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(c,'10 tonnes par seconde')
hold on
quiver(xgraph(1:3:end),ygraph(1:3:end),ub(1:3:end,1:3:end)',vb
↳ (1:3:end,1:3:end)', 'w');

%Premiere particule :
plot(xL(1,1)/dy,yL(1,1)/dx,'+g')
plot(xL(1,:)/dy,yL(1,:)/dx,'g')
plot(xL(1,dimt)/dy,yL(1,dimt)/dx,'og')

%Deuxieme particule :
plot(xL(2,1)/dy,yL(2,1)/dx,'+b')

```

```

plot(xL(2,:)/dy,yL(2,:)/dx,'b')
plot(xL(2,dimt)/dy,yL(2,dimt)/dx,'ob')

%Troisieme particule :
plot(xL(3,1)/dy,yL(3,1)/dx,'+k')
plot(xL(3,:)/dy,yL(3,:)/dx,'k')
plot(xL(dimp,dimt)/dy,yL(dimp,dimt)/dx,'ok')

%Quatrieme particule :
plot(xL(4,1)/dy,yL(4,1)/dx,'+r')
plot(xL(4,:)/dy,yL(4,:)/dx,'r')
plot(xL(dimp,dimt)/dy,yL(dimp,dimt)/dx,'or')

%Cinquieme particule :
plot(xL(5,1)/dy,yL(5,1)/dx,'+m')
plot(xL(5,:)/dy,yL(5,:)/dx,'m')
plot(xL(dimp,dimt)/dy,yL(dimp,dimt)/dx,'om')
hold off

```

7.6 Description Eulérienne

7.6.1 Masse d'eau appliquée à tout le domaine

```

%Les parametres :
load('ilefo.mat')
unew=zeros(dimX+1,dimY);
vnew=zeros(dimX,dimY+1);

%Pour un certains jours de simulation en seconde :
Ttot=60*86400;

%Calcul de unew et vnew :
for j=1:dimY
for i=2:dimX-1
unew(i,j)=(ub(i,j)+ub(i-1,j))/2;
end
end

for i=1:dimX
vnew(i,1)=vb(i,1);
end

for i=1:dimX
for j=2:dimY
vnew(i,j)=(vb(i,j)+vb(i,j-1))/2;
end
end

%Condition de stabilite :
unewabs=abs(unew);
vnewabs=abs(vnew);
maxunew=max(max(unewabs));
maxvnew=max(max(vnewabs));
dtmax=dx/maxunew;

% Afin d'avoir un dt en nombre rond par rapport a dtmax :

```

```

dt=floor(dtmax/100)*100;

%dimt se calcul en fonction du nombre de secondes de simulation
↳ et de dt
dimt=Ttot/dt;

C=zeros(dimX,dimY,dimt);
Fyam=zeros(dimX,dimY);
Fyam=zeros(dimX,dimY);
Fyam=zeros(dimX,dimY);
Fyam=zeros(dimX,dimY);

%En entree : pas de masse d'eau :
C_bound=zeros(1,dimY);

%Tout le modele est rempli d'eau :
C(:, :, 1)=Masque';

for l=2:dimt

%Calcul de Fxam et Fxav :
for j=1:dimY
for i=2:dimX-1

if unew(i,j)>0
Fyam(i,j)=unew(i,j)*C(i-1,j,l-1);
else
Fyam(i,j)=unew(i,j)*C(i,j,l-1);
end

if unew(i+1,j)>0
Fyam(i,j)=-unew(i+1,j)*C(i,j,l-1);
else
Fyam(i,j)=-unew(i+1,j)*C(i+1,j,l-1);
end

end
end

%Calcul de Fyam et Fyav :
for i=1:dimX

j=1;
if vnew(i,j)>0
Fyam(i,j)=vnew(i,j)*C_bound(1,j);
else
Fyam(i,j)=vnew(i,j)*C(i,j,l-1);
end

if vnew(i,j+1)>0
Fyam(i,j)=-vnew(i,j+1)*C(i,j,l-1);
else
Fyam(i,j)=-vnew(i,j+1)*C(i,j+1,l-1);
end

```

```

for j=2:dimY-1

if vnew(i,j)>0
Fyam(i,j)=vnew(i,j)*C(i,j-1,l-1);
else
Fyam(i,j)=vnew(i,j)*C(i,j,l-1);
end

if vnew(i,j+1)>0
Fyav(i,j)=-vnew(i,j+1)*C(i,j,l-1);
else
Fyav(i,j)=-vnew(i,j+1)*C(i,j+1,l-1);
end

end
end

%Flux :
Flux_C(1)=sum(Fyam(:,1),1)*dx*H;
C(:, :, l)=C(:, :, l-1)+dt/dx*(Fyam(:, :)+Fyav(:, :))+dt/dy*(Fyam
↪(:, :)+Fyav(:, :));

end

%Bilan de masse :
Ctot=sum(sum(C(:, :, :)))*dx*dy*H;

A=diff(Ctot)/dt;

run plot_controle_masse.m

```

7.6.2 Masse d'eau entrante

```

%Les parametres :
load('ilefo.mat')
unew=zeros(dimX+1,dimY);
vnew=zeros(dimX,dimY+1);

%Pour un certains jours de simulation en seconde :
Ttot=60*86400;

%Calcul de unew et vnew :
for j=1:dimY
for i=2:dimX-1
unew(i,j)=(ub(i,j)+ub(i-1,j))/2;
end
end

for i=1:dimX
vnew(i,1)=vb(i,1);
end

for i=1:dimX
for j=2:dimY
vnew(i,j)=(vb(i,j)+vb(i,j-1))/2;

```

```

end
end

%Condition de stabilite :
unewabs=abs(unew);
vnewabs=abs(vnew);
maxunew=max(max(unewabs));
maxvnew=max(max(vnewabs));
dtmax=dx/maxunew;

% Afin d'avoir un dt en nombre rond par rapport a dtmax :
dt = floor(dtmax/100)*100;

% dimt se calcul en fonction du nombre de secondes de simulation
↔ et de dt :
dimt=Ttot/dt;

C=zeros(dimX,dimY,dimt);
Fxm=zeros(dimX,dimY);
Fxmav=zeros(dimX,dimY);
Fym=zeros(dimX,dimY);
Fymav=zeros(dimX,dimY);

% En entree : une masse d'eau
C_bound=ones(1,dimY);

% Tout le modele est vide :
C(:, :, 1)=0;

for l=2:dimt

% Calcul de Fxm et Fxmav :
for j=1:dimY
for i=2:dimX-1

if unew(i,j)>0
Fxm(i,j)=unew(i,j)*C(i-1,j,l-1);
else
Fxm(i,j)=unew(i,j)*C(i,j,l-1);
end

if unew(i+1,j)>0
Fxmav(i,j)=-unew(i+1,j)*C(i,j,l-1);
else
Fxmav(i,j)=-unew(i+1,j)*C(i+1,j,l-1);
end

end
end

% Calcul de Fym et Fymav :
for i=1:dimX

j=1;

```

```

if vnew(i,j)>0
Fyam(i,j)=vnew(i,j)*C_bound(1,j);
else
Fyam(i,j)=vnew(i,j)*C(i,j,l-1);
end

if vnew(i,j+1)>0
Fyav(i,j)=-vnew(i,j+1)*C(i,j,l-1);
else
Fyav(i,j)=-vnew(i,j+1)*C(i,j+1,l-1);
end

for j=2:dimY-1

if vnew(i,j)>0
Fyam(i,j)=vnew(i,j)*C(i,j-1,l-1);
else
Fyam(i,j)=vnew(i,j)*C(i,j,l-1);
end

if vnew(i,j+1)>0
Fyav(i,j)=-vnew(i,j+1)*C(i,j,l-1);
else
Fyav(i,j)=-vnew(i,j+1)*C(i,j+1,l-1);
end

end
end

%Flux :
Flux_C(1)=sum(Fyam(:,1),1)*dx*H;
C(:, :, 1)=C(:, :, l-1)+dt/dx*(Fyam(:, :)+Fyav(:, :))+dt/dy*(Fyam
↪(:, :)+Fyav(:, :));

end

%Bilan de masse :
Ctot=sum(sum(C(:, :, :)))*dx*dy*H;

A=diff(Ctot)/dt;

run plot_controle_masse.m

```

7.6.3 plot_controle_masse.m

```

Na=1;

figure(1)
clf
pcolor(xgraph,ygraph,(Psi.*toto)');c=colorbar;shading flat
title('Modele en 2D avec frontiere ouverte')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(c,'Concentration')
hold on
quiver(xgraph(1:Na:end)+0.5,ygraph(1:Na:end)+0.5,ub(1:Na:end),1:

```

```

    ↪ Na:end)', vb(1:Na:end, 1:Na:end)', 'w');
hold off
title('dt=1')
xlabel('Distance en km')
ylabel('Distance en km')

figure(2)
subplot(4,1,1)
pcolor((C(:, :, 1).*toto'))); shading flat; a=colorbar;
hold on
quiver(xgraph(1:Na:end), ygraph(1:Na:end), ub(1:Na:end, 1:Na:end)
    ↪ ', vb(1:Na:end, 1:Na:end)', 'w');
hold off
caxis([0 1])
title('dt=1')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(a, 'Concentration')

subplot(4,1,2)
pcolor((C(:, :, 500).*toto'))); shading flat; b=colorbar;
hold on
quiver(xgraph(1:Na:end), ygraph(1:Na:end), ub(1:Na:end, 1:Na:end)
    ↪ ', vb(1:Na:end, 1:Na:end)', 'w');
hold off
title('dt=500')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(b, 'Concentration')

subplot(4,1,3)
pcolor((C(:, :, 1000).*toto'))); shading flat; d=colorbar;
hold on
quiver(xgraph(1:Na:end), ygraph(1:Na:end), ub(1:Na:end, 1:Na:end)
    ↪ ', vb(1:Na:end, 1:Na:end)', 'w');
hold off
title('dt=1000')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(d, 'Concentration')

subplot(4,1,4)
pcolor((C(:, :, 1600).*toto'))); shading flat; e=colorbar;
hold on
quiver(xgraph(1:Na:end), ygraph(1:Na:end), ub(1:Na:end, 1:Na:end)
    ↪ ', vb(1:Na:end, 1:Na:end)', 'w');
hold off
title('dt=1600')
xlabel('Distance en km')
ylabel('Distance en km')
ylabel(e, 'Concentration')

Titre='C(:, :, 1)=Masque avec differents pas de temps';
%Titre='C(:, :, 1)=0 + entree masse eau avec differents pas de
    ↪ temps';

```

```

[ax,h3]=suplabel(Titre,'t');
set(h3,'FontSize',13)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3)
subplot(4,1,1)
plot(squeeze(Ctot))
title('Concentration totale : Ctot=sum(sum(C(:,:,:)))*dx*dy*H')
xlabel('Pas de temps en seconde')
ylabel('Concentration totale')

subplot(4,1,2)
plot(squeeze(A))
title('Flux theorique : A=diff(Ctot)/dt')
xlabel('Pas de temps en seconde')
ylabel('Flux theorique')

subplot(4,1,3)
plot(Flux_C)
title('Flux experimental : Flux\_C(1)=sum(Fyam(:,1),1)*dx*H')
xlabel('Pas de temps en seconde')
ylabel('Flux experimental')

subplot(4,1,4)
plot(squeeze(A),Flux_C(2:end),'+')
title('Rapport flux theorique sur experimental')

Titre='Bilan de masse';
[ax,h3]=suplabel(Titre,'t');
set(h3,'FontSize',13)

```

7.7 suplabel.m

```

function [ax,h]=suplabel(text,whichLabel,supAxes)
% Places text as a title, xlabel, or ylabel on a group of
% ↪ subplots.
% Returns a handle to the label and a handle to the axis.
% [ax,h]=suplabel(text,whichLabel,supAxes)
% returns handles to both the axis and the label.
% ax=suplabel(text,whichLabel,supAxes)
% returns a handle to the axis only.
% suplabel(text) with one input argument assumes whichLabel='x'
% ↪ '
%
% whichLabel is any of 'x', 'y', 'yy', or 't', specifying
% ↪ whether the
% text is to be the xlabel, ylabel, right side y-label,
% or title respectively.
%
% supAxes is an optional argument specifying the Position of
% ↪ the
% "super" axes surrounding the subplots.
% supAxes defaults to [.08 .08 .84 .84]
% specify supAxes if labels get chopped or overlay subplots
%

```

```

% EXAMPLE:
% subplot(2,2,1);ylabel('ylabel1');title('title1')
% subplot(2,2,2);ylabel('ylabel2');title('title2')
% subplot(2,2,3);ylabel('ylabel3');xlabel('xlabel3')
% subplot(2,2,4);ylabel('ylabel4');xlabel('xlabel4')
% [ax1,h1]=suplabel('super X label');
% [ax2,h2]=suplabel('super Y label','y');
% [ax3,h2]=suplabel('super Y label (right)','yy');
% [ax4,h3]=suplabel('super Title' , 't');
% set(h3,'FontSize',30)
%
% SEE ALSO: text, title, xlabel, ylabel, zlabel, subplot,
%           suplabel (Matlab Central)

% Author: Ben Barrowes <barrowes@alum.mit.edu>

%modified 3/16/2010 by IJW to make axis behavior re "zoom" on
↪ exit same as
%at beginning. Requires adding tag to the invisible axes

currax=findobj(gcf,'type','axes','-not','tag','suplabel');

if nargin < 3
supAxes=[.08 .08 .84 .84];
ah=findall(gcf,'type','axes');
if ~isempty(ah)
supAxes=[inf,inf,0,0];
leftMin=inf; bottomMin=inf; leftMax=0; bottomMax=0;
axBuf=.04;
set(ah,'units','normalized')
ah=findall(gcf,'type','axes');
for ii=1:length(ah)
if strcmp(get(ah(ii),'Visible'),'on')
thisPos=get(ah(ii),'Position');
leftMin=min(leftMin,thisPos(1));
bottomMin=min(bottomMin,thisPos(2));
leftMax=max(leftMax,thisPos(1)+thisPos(3));
bottomMax=max(bottomMax,thisPos(2)+thisPos(4));
end
end
supAxes=[leftMin-axBuf,bottomMin-axBuf,leftMax-leftMin+axBuf*2,
↪ bottomMax-bottomMin+axBuf*2];
end
end
if nargin < 2, whichLabel = 'x'; end
if nargin < 1, help(mfilename); return; end

if ~isstr(text) | ~isstr(whichLabel)
error('text and whichLabel must be strings')
end
whichLabel=lower(whichLabel);

ax=axes('Units','Normal','Position',supAxes,'Visible','off',
↪ tag,'suplabel');

```

```

if strcmp('t',whichLabel)
set(get(ax,'Title'),'Visible','on')
title(text);
elseif strcmp('x',whichLabel)
set(get(ax,'XLabel'),'Visible','on')
xlabel(text);
elseif strcmp('y',whichLabel)
set(get(ax,'YLabel'),'Visible','on')
ylabel(text);
elseif strcmp('yy',whichLabel)
set(get(ax,'YLabel'),'Visible','on')
ylabel(text);
set(ax,'YAxisLocation','right')
end

for k=1:length(currax), axes(currax(k));end % restore all other
↪ axes

if (nargout < 2)
return
end
if strcmp('t',whichLabel)
h=get(ax,'Title');
set(h,'VerticalAlignment','middle')
elseif strcmp('x',whichLabel)
h=get(ax,'XLabel');
elseif strcmp('y',whichLabel) | strcmp('yy',whichLabel)
h=get(ax,'YLabel');
end

%%ah=findall(gcf,'type','axes');
%%'sssssssss',kb

```