

IPSA

MA323 - DIFFÉRENCES FINIES

AÉRO 3

Analyse de la dispersion d'un polluant et application à la modélisation fluviale

Étudiants :

Guillaume FAIVRE

Emma FILIPPONE

Titouan MILLET

Camille PIEUSSERGUES

Professeurs :

H. EL-OTMANY

C. PESCHARD

L. BLETZACKER

3 juin 2023



Cette page est laissée intentionnellement vierge.

— Date de remise du rapport : 04/06/2023

— Référence du projet : Ma323/S6/ 2022 – 2023

— Intitulé du projet : Analyse de la dispersion d'un polluant et application à la modélisation fluviale

— Type de projet : expérimental, simulation

— Objectifs du projet :

L'objectif de cette étude est de modéliser l'évolution de la concentration d'un polluant déversé dans un cours d'eau, en prenant en compte les phénomènes de convection, de diffusion et de dispersion. À travers des schémas aux différences finies et des simulations numériques, nous cherchons à observer le déplacement de la concentration dans le temps et l'espace, afin de mieux comprendre le comportement du polluant et son impact potentiel sur l'environnement

— Mots-clefs du projet :

1. Pollution de l'eau
2. Modélisation
3. Schéma des méthodes de différences finis
4. Programmation Python

Table des matières

1	Méthodologie et organisation du travail	1
1.1	Logiciels utilisés	1
2	Motivation et modélisation mathématique	2
3	Variation de la concentration de polluant dans le fleuve	3
3.1	Phénomène de convection en 1D	3
3.2	Phénomène de convection-diffusion en 1D	10

Notations, acronymes

N	nombre des points intérieurs de l'espace
h	pas de discrétisation du temps
t_n, x_i	les coordonnées des points de la discrétisation
c	variable de convection
d	variable de diffusion

1 Méthodologie et organisation du travail

Nous avons adopté une méthodologie de travail en groupe pour aborder les différentes questions, car nous avons remarqué que les questions étaient interdépendantes. Dans un premier temps, nous avons simplifié le problème en le réduisant à une dimension, ce qui nous a permis de nous familiariser avec les concepts mathématiques et de programmation.

Pour maintenir un travail collaboratif, nous avons utilisé l’IDE VS Code avec l’extension Live Share, ce qui nous a permis de travailler simultanément sur le même projet. De plus, nous avons utilisé un dépôt GitHub pour maintenir le code à jour et assurer un suivi efficace des modifications. Pour les aspects mathématiques, nous avons travaillé à partir de documents partagés sur nos iPad.

Cette méthodologie de travail a été bénéfique pour l’ensemble du groupe, car elle nous a permis de rester tous à jour et d’avoir accès aux mêmes ressources. Elle a favorisé une collaboration étroite et efficace, ce qui a été essentiel pour avancer dans notre projet de manière cohérente et productive.

1.1 Logiciels utilisés

- **VS Code** est un environnement de développement intégré (IDE) populaire et largement utilisé. Il offre une interface conviviale et de nombreuses fonctionnalités avancées pour faciliter le développement de logiciels.
VS Code prend en charge une grande variété de langages de programmation et offre des fonctionnalités telles que la coloration syntaxique, l’autocomplétion, la mise en forme de code, le débogage, le contrôle de version et bien d’autres encore. Il est également extensible grâce à un système d’extensions, ce qui permet aux développeurs d’ajouter des fonctionnalités supplémentaires en installant des extensions à partir de la marketplace de VS Code.
- **Live Share** est une fonctionnalité très utile qui permet aux développeurs de collaborer en temps réel sur un même projet, même s’ils se trouvent sur des machines différentes. Avec Live Share, les utilisateurs peuvent partager leur environnement de développement, ce qui permet à d’autres personnes d’accéder et de modifier le code en temps réel. Cela facilite grandement la collaboration à distance, car les membres de l’équipe peuvent travailler ensemble sur le même projet sans avoir besoin de partager physiquement le même espace de travail.

2 Motivation et modélisation mathématique

Depuis longtemps, la pollution de l'eau dans les écoulements à surface libre est un sujet d'intérêt majeur pour les chercheurs scientifiques et l'humanité en général. Les recherches sur la qualité de l'eau et ses ressources se multiplient pour répondre à la demande exprimée par les spécialistes de la protection de l'environnement et de la santé publique. Les chercheurs s'intéressent tant à la quantité qu'à la qualité de l'eau, car ils comprennent que la qualité de l'eau a un impact sur son utilisation par les êtres humains, et inversement. Lorsque nous utilisons l'eau, nous altérons sa qualité plutôt que sa quantité. Ce cercle vicieux met en évidence le fait que la pratique consistant à rejeter directement les eaux usées non traitées et les déchets chimiques dans les cours d'eau et les océans n'est plus acceptable si l'on souhaite éviter la dégradation de l'environnement. Les processus naturels de décomposition dans les masses d'eau ne suffisent plus à éliminer ces apports de polluants. La technologie peut aider à réduire ou éliminer les substances nocives pour l'environnement, mais que se passe-t-il lorsque les contaminants ne sont pas éliminés, même avec les méthodes de traitement de l'eau les plus modernes ? Même à des concentrations minimales, ils peuvent s'accumuler et devenir extrêmement nuisibles en raison de leur persistance. Dans ce cas, la seule façon de protéger les générations futures et l'écosystème est d'empêcher la présence de produits chimiques dans les réseaux hydrographiques.

Dans cette perspective, les modèles numériques de qualité des eaux reconstituent différents mécanismes présents dans le milieu et fournissent toutes les informations nécessaires. La vitesse de l'écoulement joue un rôle important dans le transport de la pollution, ce qui rend essentielle sa représentation précise à l'aide d'un modèle mathématique prenant en compte des phénomènes physiques tels que la diffusion, l'advection, la dispersion, l'adsorption, la désorption, la précipitation, la sédimentation, l'évaporation, etc. Ces phénomènes dépendent à la fois de la nature des substances rejetées et des caractéristiques hydrodynamiques propres au cours d'eau. En ce qui concerne les substances polluantes, leur dispersion est principalement gouvernée par la turbulence, ce qui nécessite une prise en compte précise de ce phénomène. Malgré les nombreuses recherches déjà effectuées sur la modélisation numérique des dispersions de polluants dans les cours d'eau, certains phénomènes restent encore simplifiés. Dans ce projet, l'objectif est d'étudier l'évolution d'un polluant déversé d'abord dans un fleuve, puis dans l'océan lorsque le fleuve se jette dedans. Le fleuve et l'océan sont deux milieux soumis à des courants, et l'objectif est d'observer les effets de ces courants sur la dispersion du polluant. Ainsi, nous proposons de modéliser l'évolution de la concentration du polluant, notée $u(t, x)$, dans le temps t et à la position $x \in \mathbb{R}^d$,

$$\begin{cases} \frac{\partial u}{\partial t} + V \nabla u - \nu \Delta u &= f \\ u(0, x) &= u_0(x) \end{cases} \quad (1)$$

3 Variation de la concentration de polluant dans le fleuve

Pour commencer, nous allons étudier l'équation de convection, pour laquelle il est possible de calculer des solutions explicites. L'équation de convection est généralement utilisée pour modéliser le transport d'une substance dans un milieu où un courant ou un écoulement est présent. Elle peut être représentée mathématiquement par l'équation suivante :

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

où $u(t, x)$ est la concentration du polluant à l'instant t et à la position x , et c est la vitesse de convection.

3.1 Phénomène de convection en 1D

Pour mieux comprendre ce phénomène, on simplifie le problème en négligeant les termes de diffusion et de source ($\nu = 0$ et $f = 0$), ainsi qu'en considérant la vitesse constante et positive, $V > 0$. Le problème modèle de convection s'écrit ainsi

$$\begin{cases} \frac{\partial u}{\partial t} + V \frac{\partial u}{\partial x} = 0, \forall x \in \mathbb{R}, \forall t > 0 \\ u(t = 0, x) = u_0(x), \forall x \in \mathbb{R} \end{cases} \quad (2)$$

1. Nous posons la fonction $\phi : \mathbb{R} \rightarrow \mathbb{R}$ comme suit : $\forall t \in \mathbb{R}, \phi(t) = u(t, X(t))$.

Calculons la dérivée de ϕ à l'aide de la règle de la chaîne pour les fonctions composées :

$$\forall t \in \mathbb{R}, \phi'(t) = \frac{d\phi}{dt} = \frac{\partial u}{\partial t} + \frac{dX}{dt} \frac{\partial u}{\partial x} = 0$$

Or, nous savons que $\frac{\partial u}{\partial t} + V \frac{\partial u}{\partial x} = 0$, d'où :

$$\begin{aligned} \forall t \in \mathbb{R}, \frac{\partial u}{\partial t} + \frac{dX}{dt} \frac{\partial u}{\partial x} &= \frac{\partial u}{\partial t} + V \frac{\partial u}{\partial x} \\ &\Rightarrow \frac{dX}{dt} = V \\ &\Rightarrow X(t) = Vt + C \end{aligned}$$

D'après les conditions initiales, nous déterminons la constante C :

$$\begin{aligned} X(t=0) &= V \times 0 + C \\ \Rightarrow C &= X(t=0) \end{aligned}$$

D'où finalement :

$$\forall t \in \mathbb{R}, X(t) = Vx + X(t=0)$$

Ainsi, X répond à l'équation d'une droite donc les courbes caractéristiques $t \mapsto (t, X(t))$, le long desquelles u est constante, sont des droites de pente V.

2. Nous allons écrire le schéma explicite centré pour résoudre le problème modèle de convection :

$$\frac{u_j^{n+1} - u_j^n}{\tau} + V \frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0$$

$$\Leftrightarrow (u_j^{n+1} - u_j^n) + \frac{V\tau}{2h}(u_{j+1}^n - u_{j-1}^n) = 0$$

$$\Leftrightarrow (u_j^{n+1} - u_j^n) + c(u_{j+1}^n - u_{j-1}^n) = 0$$

$$\Leftrightarrow u_j^{n+1} = \frac{c}{2}u_{j-1}^n + u_j^n - \frac{c}{2}u_{j+1}^n$$

$$\Leftrightarrow \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -\frac{c}{2} & 0 & 0 & 0 & \cdots & \frac{c}{2} \\ \frac{c}{2} & 1 & -\frac{c}{2} & 0 & 0 & \cdots & 0 \\ 0 & \frac{c}{2} & 1 & -\frac{c}{2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -\frac{c}{2} & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_N^n \\ u_{N+1}^n \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{pmatrix} = \left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} - \begin{pmatrix} 0 & \frac{c}{2} & 0 & 0 & 0 & \cdots & -\frac{c}{2} \\ -\frac{c}{2} & 0 & \frac{c}{2} & 0 & 0 & \cdots & 0 \\ 0 & -\frac{c}{2} & 0 & \frac{c}{2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \frac{c}{2} & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \right] \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_N^n \\ u_{N+1}^n \end{pmatrix}$$

$$\Leftrightarrow U^{n+1} = (I_N - M)U^n$$

Nous avons trouvé une relation de récurrence qui décrit le problème modèle de convection par

$$\text{le schéma explicite centré, en posant } c = \frac{V\tau}{h} \text{ et } M = \begin{pmatrix} 0 & \frac{c}{2} & 0 & 0 & 0 & \cdots & -\frac{c}{2} \\ -\frac{c}{2} & 0 & \frac{c}{2} & 0 & 0 & \cdots & 0 \\ 0 & -\frac{c}{2} & 0 & \frac{c}{2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \frac{c}{2} & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

Nous allons maintenant pouvoir résoudre le problème à l'aide de Python :

```

1  def f(N):
2      x      = np.linspace(0,a,N+1)
3      n_max  = int(T/tau)
4      t      = np.linspace(0,n_max*tau,n_max+1)
5      U      = np.zeros((n_max+1,N+1))
6      U[0,:] = 1/(sigma*np.sqrt(2*np.pi))*np.exp(-((x-a/2)**2)/(2*sigma**2))
7      un = U[0,: N]
8      c   = V * tau / h
9      return x, n_max, t, U, un, c
10
11 def mat_id(N):
12     return np.eye(N)
13
14 def mat_M(N,c):
15     M = np.zeros((N,N))
16     d_c = -c/2 * np.ones((N-1,1))
17     M += np.diagflat(d_c,-1) + np.diagflat(-d_c,1)
18     M[0,N-1] = -c/2
19     M[N-1,0] = c/2
20     return M
21
22 def SolutionEC(N):
23     x, n_max, t, U, un, c = f(N)
24     Me = mat_M(N,c)
25     In = mat_id(N)
26
27     for n in range(n_max):
28         un = (In-Me) @ un
29         U[n+1,:N] = un
30         U[n+1,N] = U[n+1,0]
31     return x,t,U
32
33 l= [0,0.5,1,1.5,2]
34 for i in l:
35     plt.plot(xEC,uEC[int(i/tau),:],label="t="+str(tEC))
36     plt.pause(1)
37 plt.title("Explicite centré")
38 plt.show()

```

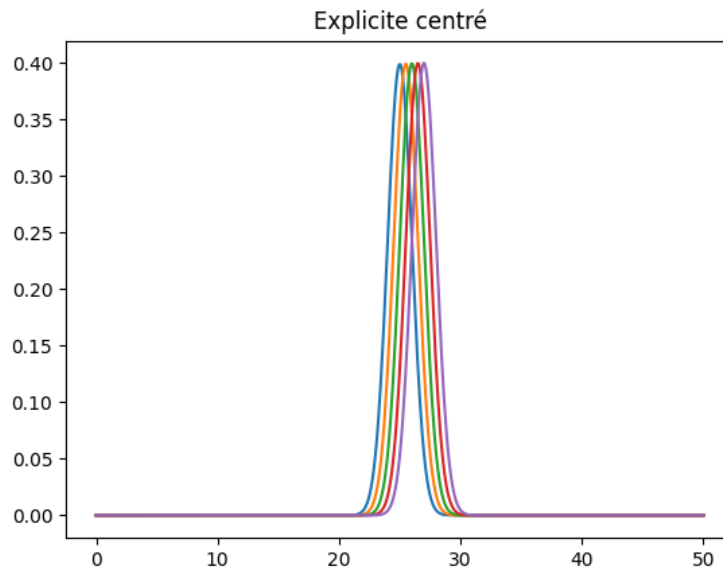


FIGURE 1 – Évolution de la concentration à l'aide d'un schéma explicite centré

Avec l'animation, nous remarquons le déplacement de la concentration au fil du temps, ce qui est clairement visible sur la courbe. Ce déplacement de la concentration peut être attribué à la convection, c'est-à-dire au transport du polluant par le courant ou l'écoulement du cours d'eau. La vitesse de convection, représentée par le paramètre c dans l'équation de convection, influence la rapidité et la direction du déplacement de la concentration. Dans l'animation, nous pouvons voir comment la concentration se propage progressivement

Ecrivons maintenant le schéma de Crank-Nicolson :

$$\begin{aligned}
& \ll \frac{\text{Explicite centré} + \text{Implicite centré}}{2} \gg \\
& \left[\begin{cases} \frac{u_j^{n+1} - u_j^n}{\tau} + V \frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0 \\ \frac{u_j^{n+1} - u_j^n}{\tau} + V \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} = 0 \end{cases} \right] \\
& \frac{u_j^{n+1} - u_j^n}{\tau} + V \frac{u_{j+1}^n - u_{j-1}^n}{2h} + \frac{u_j^{n+1} - u_j^n}{\tau} + V \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} = 0 \\
& \Rightarrow 2 \left(\frac{u_j^{n+1} - u_j^n}{\tau} \right) + \frac{V}{2h} (u_{j+1}^n - u_{j-1}^n + u_{j+1}^{n+1} - u_{j-1}^{n+1}) = 0 \\
& \Rightarrow \frac{2}{\tau} u_j^{n+1} - \frac{2}{\tau} u_j^n + \frac{V}{2h} u_{j+1}^n - \frac{V}{2h} u_{j-1}^n + \frac{V}{2h} u_{j+1}^{n+1} - \frac{V}{2h} u_{j-1}^{n+1} = 0 \\
& \Rightarrow \frac{2}{\tau} u_j^{n+1} + \frac{V}{2h} u_{j+1}^{n+1} - \frac{V}{2h} u_{j-1}^{n+1} = \frac{2}{\tau} u_j^n - \frac{V}{2h} u_{j+1}^n + \frac{V}{2h} u_{j-1}^n \\
& \Rightarrow -\frac{V}{2h} u_{j-1}^{n+1} + \frac{2}{\tau} u_j^{n+1} + \frac{V}{2h} u_{j+1}^{n+1} = \frac{V}{2h} u_{j-1}^n + \frac{2}{\tau} u_j^n - \frac{V}{2h} u_{j+1}^n \\
& \Rightarrow -\frac{V\tau}{4h} u_{j-1}^{n+1} + u_j^{n+1} + \frac{V\tau}{4h} u_{j+1}^{n+1} = \frac{V\tau}{4h} u_{j-1}^n + u_j^n - \frac{V\tau}{4h} u_{j+1}^n \\
& \Rightarrow \begin{pmatrix} 1 & \frac{V\tau}{4h} & 0 & 0 & 0 & \cdots & -\frac{V\tau}{4h} \\ -\frac{V\tau}{4h} & 1 & \frac{V\tau}{4h} & 0 & 0 & \cdots & 0 \\ 0 & -\frac{V\tau}{4h} & 1 & \frac{V\tau}{4h} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \frac{V\tau}{4h} & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{pmatrix} \\
& = \begin{pmatrix} 1 & -\frac{V\tau}{4h} & 0 & 0 & 0 & \cdots & \frac{V\tau}{4h} \\ \frac{V\tau}{4h} & 1 & -\frac{V\tau}{4h} & 0 & 0 & \cdots & 0 \\ 0 & \frac{V\tau}{4h} & 1 & -\frac{V\tau}{4h} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -\frac{V\tau}{4h} & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_N^n \\ u_{N+1}^n \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
& \Rightarrow \left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} + \frac{\tau}{2} \begin{pmatrix} 0 & \frac{V\tau}{4h} & 0 & 0 & 0 & \cdots & -\frac{V\tau}{4h} \\ -\frac{V\tau}{4h} & 0 & \frac{V\tau}{4h} & 0 & 0 & \cdots & 0 \\ 0 & -\frac{V\tau}{4h} & 0 & \frac{V\tau}{4h} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \frac{V\tau}{4h} & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \right] \begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{pmatrix} \\
& = \left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} - \frac{\tau}{2} \begin{pmatrix} 0 & \frac{V\tau}{4h} & 0 & 0 & 0 & \cdots & -\frac{V\tau}{4h} \\ -\frac{V\tau}{4h} & 0 & \frac{V\tau}{4h} & 0 & 0 & \cdots & 0 \\ 0 & -\frac{V\tau}{4h} & 0 & \frac{V\tau}{4h} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \frac{V\tau}{4h} & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \right] \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_N^n \\ u_{N+1}^n \end{pmatrix}
\end{aligned}$$

$$\Rightarrow (I_N + M)U^{n+1} = (I_N - M)U^n \quad (3)$$

Avec le schéma déterminé à l'équation précédente (3), nous construisons un script python nous permettant de visualiser l'évolution de concentration de polluant :

```

1      def mat_M_CN(N,c):
2          M = np.zeros((N,N))
3          d_c = -c/2 * np.ones((N-1,1))
4          M += np.diagflat(d_c,-1) + np.diagflat(-d_c,1)
5          return M
6
7      def SolutionCN(N):
8          x, n_max, t, U, un, c = f(N)
9          c = (V * tau) / (2*h)
10
11         Me = mat_M_CN(N,c)
12         In = mat_id(N)
13
14         for n in range(n_max):
15             un = np.linalg.solve(In + Me, (In - Me) @ un)
16             U[n+1,:N] = un
17             U[n+1,N] = U[n+1,0]
18
19         return x,t,U
20

```

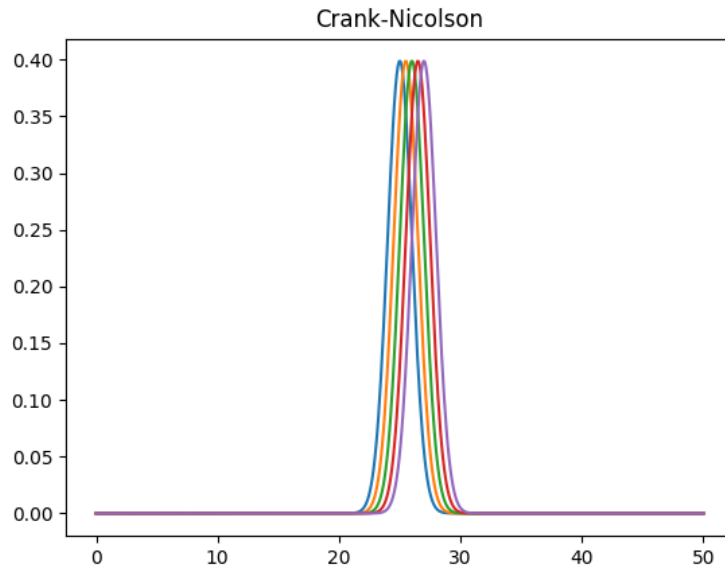


FIGURE 2 – Évolution de la concentration à l'aide d'un schéma explicite centré

Avec l'animation, nous remarquons le déplacement de la concentration au fil du temps, ce qui est clairement visible sur la courbe. Ce déplacement de la concentration peut être attribué à la convection, c'est-à-dire au transport du polluant par le courant ou l'écoulement du cours d'eau. La vitesse de convection, représentée par le paramètre c dans l'équation de convection, influence la rapidité et la direction du déplacement de la concentration. Dans l'animation, nous pouvons voir comment la concentration se propage progressivement.

3.2 Phénomène de convection-diffusion en 1D

1. À partir des équations précédentes de Crank-Nicolson nous pouvons écrire le nouveau schéma :

$$\left(I_N + \frac{\tau}{2}M - \frac{\nu\tau}{2}A\right)U^{n+1} = \left(I_N - \frac{\tau}{2}M + \frac{\nu\tau}{2}A\right)U^n$$

Nous cherchons donc à identifier les matrices A et M :

$$\begin{aligned} & \frac{u_j^{n+1} - u_j^n}{\tau} + \frac{V}{2} \left(\frac{u_{j+1}^n - u_{j-1}^n}{2h} + \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} \right) \\ & - \frac{\nu}{2} \left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} + \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} \right) = 0 \\ \Leftrightarrow & \frac{-u_j^n}{\tau} + \frac{V}{4h} (u_{j+1}^n - u_{j-1}^n) + \frac{-\nu}{2h^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \\ & \frac{u_j^{n+1}}{\tau} + \frac{V}{4h} \left(\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} \right) + \frac{-\nu}{2h^2} (u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) = 0 \\ \Leftrightarrow & -u_j^n + \frac{V\tau}{4h} (u_{j+1}^n - u_{j-1}^n) + \frac{-\nu\tau}{2h^2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \\ & u_j^{n+1} + \frac{V\tau}{4h} \left(\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} \right) + \frac{-\nu\tau}{2h^2} (u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) = 0 \end{aligned}$$

Nous posons :

$$c = \frac{V\tau}{h} \quad d = \frac{\nu\tau}{h^2} \quad (4)$$

$$\begin{aligned} \Leftrightarrow & -u_j^n + \frac{c}{4} (u_{j+1}^n - u_{j-1}^n) + \frac{-d}{2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) + \\ & u_j^{n+1} + \frac{c}{4} \left(\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} \right) + \frac{-d}{2} (u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) = 0 \end{aligned}$$

$$\begin{aligned} & u_j^{n+1} + \frac{c}{4} (u_{j+1}^{n+1} - u_{j-1}^{n+1}) + \frac{-d}{2} (u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) \\ & = \\ & -u_j^n + \frac{c}{4} (u_{j+1}^n - u_{j-1}^n) + \frac{-d}{2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) \end{aligned} \quad (5)$$

D'après l'équation (5), nous pouvons identifier les matrices M et A de la façons suivante :

$$\begin{pmatrix} u_1^{n+1} - \cancel{u_0^{n+1}} \\ u_2^{n+1} - u_1^{n+1} \\ \vdots \\ \cancel{u_{N+1}^{n+1}} - u_{N-1}^{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & -1 & 0 & 1 \\ 0 & \dots & \dots & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_N^{n+1} \end{pmatrix}$$

$$\begin{pmatrix} u_2^n - 2u_1^n + \cancel{u_0^n} \\ u_3^n - 2u_2^n + u_1^n \\ \vdots \\ \cancel{u_{N+1}^n} - 2u_N^n + u_{N-1}^n \end{pmatrix} = \begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 1 & -2 & 1 \\ 0 & \dots & \dots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_2^{n+1} \\ \vdots \\ u_N^{n+1} \end{pmatrix}$$

En nous appuyant sur les variables que nous avons introduite à l'équation (4), nous trouvons les matrices suivante :

$$M = \frac{1}{2h} \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & -1 & 0 & 1 \\ 0 & \dots & \dots & 0 & -1 & 0 \end{pmatrix} \quad A = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 1 & -2 & 1 \\ 0 & \dots & \dots & 0 & 1 & -2 \end{pmatrix}$$

Enfin nous pouvons écrire le schéma de Crank-Nicolson comme suit :

$$\left(I_N + \frac{V\tau}{2}M - \frac{\nu\tau}{2}A\right)U^{n+1} = \left(I_N - \frac{V\tau}{2}M + \frac{\nu\tau}{2}A\right)U^n \quad (6)$$

2. Avec le schéma déterminé à l'équation précédente (6), nous construisons un script python nous permettant de visualiser l'évolution de concentration de polluant :

```

1  def mat_M(N):
2      M = np.zeros((N,N))
3      M += np.diagflat(-1*np.ones((N-1,1)),-1) + np.diagflat(1*np.ones((N-1,1)),1)
4      M = M*(1/(2*h))
5      return M
6  def mat_A(N):
7      A = -2 * np.eye(N)
8      A += np.diagflat(-1*np.ones((N-1,1)),-1) + np.diagflat(1*np.ones((N-1,1)),1)
9      A = A * (1/h**2)
10     return A
11
12     ...
13
14  for n in range(n.max):
15      un = np.linalg.solve((In + c * M - d * A), (In - c * M + d * A) @ un)
16      U[n+1,:N] = un
17      U[n+1,N] = U[n+1,0]
18

```

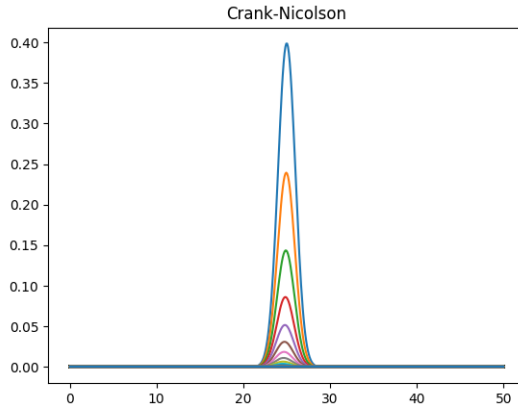


FIGURE 3 – Évolution de la concentration à l'aide d'un schéma explicite centré

Dans l'analyse de la figure illustrant le phénomène de convection-diffusion en 1D, nous observons plusieurs observations significatives. Tout d'abord, nous constatons que la concentration du polluant a tendance à diminuer au fil du temps. Cela peut être attribué à la diffusion, où le polluant se mélange progressivement avec le milieu environnant, entraînant une diminution de sa concentration initiale. Cependant, une autre observation importante est que nous ne remarquons plus de déplacement significatif de la concentration. Cela peut être dû à l'absence de mouvement convectif dans le système étudié. En l'absence de convection, la diffusion devient le principal mécanisme de dispersion du polluant.

3. Afin d'améliorer notre modèle, nous appliquons des conditions de type Neumann homogène qui reviennent à imposer :

$$\frac{u_{N+1} - u_N}{h} = 0$$

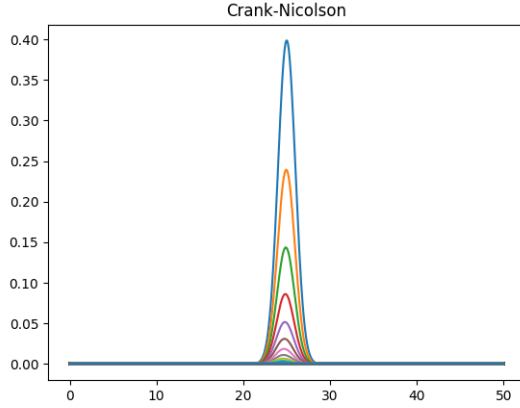
$$u_{N+1} = u_N \times h$$

Nous modifions les matrices A et M de la façon suivante :

$$M = \frac{1}{2h} \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & h \\ 0 & \dots & \dots & -1 & 0 & 0 \\ 0 & \dots & \dots & 0 & -1 & -h \end{pmatrix} \quad A = \frac{1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & h \\ 0 & \dots & \dots & 1 & -2 & -2h \\ 0 & \dots & \dots & 0 & 1 & h \end{pmatrix}$$

```

1 def mat_M(N):
2     M = np.zeros((N,N))
3     M += np.diagflat(-1*np.ones((N-1,1)),-1) + np.diagflat(1*np.ones((N-1,1))
4     M[:, -1] = M[:, -2]*h
5     M = M*(1/(2*h))
6     return M
7
8 def mat_A(N):
9     A = -2 * np.eye(N)
10    A += np.diagflat(-1*np.ones((N-1,1)),-1) + np.diagflat(1*np.ones((N-1,1))
11    A[:, -1] = A[:, -2]*h
12    A = A * (1/h**2)
13    return A
14
```



Après une modification des matrices dans le code et une nouvelle simulation numérique, nous ne remarquons pas de changement sur nos courbes.

FIGURE 4 – Évolution de la concentration à l'aide d'un schéma explicite centré

4. Afin de simuler la l'action de l'usine déversant un polluant, nous introduisons le terme de source $f(t, x)$:

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) + V \frac{\partial u}{\partial x}(t, x) - \nu \frac{\partial^2 u}{\partial x^2}(t, x) = f(t, x), & \forall x \in \mathbb{R}, \forall t > 0 \\ u(t = 0, x) = u_0(x), & \forall x \in \mathbb{R} \end{cases}$$

À partir de l'équation (6), nous récrivons le schéma de Crank-Nickolson en y ajoutant le terme source :

$$\begin{aligned} u_j^{n+1} + \frac{c}{4} (u_{j+1}^{n+1} - u_{j-1}^{n+1}) &+ \frac{-d}{2} (u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) \\ &= \\ -u_j^n + \frac{c}{4} (u_{j+1}^n - u_{j-1}^n) &+ \frac{-d}{2} (u_{j+1}^n - 2u_j^n + u_{j-1}^n) + f(t, x) \end{aligned} \quad (7)$$

Sous forme matricielle nous aurons la matrice (F) aura la forme suivante :

$$F = \begin{pmatrix} f_0^0 & f_1^0 & f_2^0 & \cdots & f_N^0 \\ f_0^1 & f_1^1 & f_2^1 & \cdots & f_N^1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_0^{n_{\max}} & f_1^{n_{\max}} & f_2^{n_{\max}} & \cdots & f_N^{n_{\max}} \end{pmatrix}$$

où chaque élément f_j^n est la valeur de la fonction source $f(t_n, x_j)$ au temps t_n et à la position x_j .

Le schéma de Crank-Nicolson au format matricielle, se réécrit à partir de l'équation (5) :

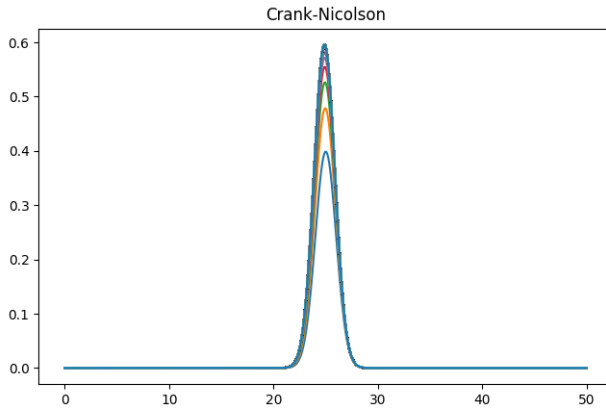
$$\left(I_N + \frac{V\tau}{2}M - \frac{\nu\tau}{2}A\right)U^{n+1} = \left(I_N - \frac{V\tau}{2}M + \frac{\nu\tau}{2}A\right)U^n + F \quad (8)$$

Nous modifions le code de la façon suivante :

```

1 def mat_F(N):
2     F = np.zeros((n_max + 1, N)) # Initialisation de la matrice F
3     x = np.linspace(0, a, N + 1) # Grille des positions
4     # Remplissage des coefficients de la matrice F
5     for n in range(n_max):
6         for j in range(N):
7             f_nj = 1 / (sigma * np.sqrt(2 * np.pi)) * np.exp(-((x[j] - a/2)
8             **2) / (2 * sigma**2))
9             F[n, j] = f_nj
10
11     return F
12
13
14 for n in range(n_max):
15     un = np.linalg.solve((In + c * M - d * A), (In - c * M + d * A) @ (np.add(un,
16     Fnj[0, :]))))
17     U[n+1, :N] = un
18     U[n+1, N] = U[n+1, 0]

```



L'introduction d'une source de polluant peut se traduire par une augmentation nette de la concentration dans le système étudié. Cette source continue d'alimenter le système en polluant, ce qui entraîne une augmentation progressive de la concentration au fil du temps. Cette observation souligne l'importance de prendre en compte les sources de polluants dans la modélisation et la gestion des phénomènes de convection-diffusion en 1D.

FIGURE 5 – Évolution de la concentration à l'aide d'un schéma explicite centré