Master 2 MACS

Rapport de stage

Estimation du tenseur de diffusion anisotrope dans des écoulements granulaires polydisperses à partir de simulations numériques discrètes

Par

VENEU Romain

Nantes Université

UFR sciences et techniques

2023-2024

Table des matières

Chapitre 1

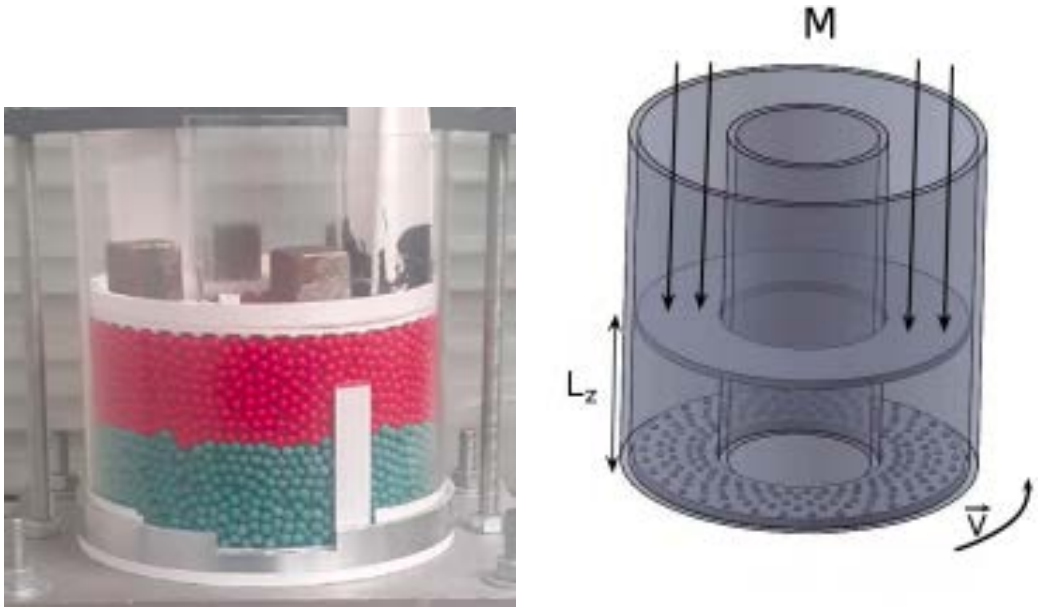
Introduction

1.1 Contexte

La diffusion induite par le cisaillement est un mécanisme clé pour le transfert de masse dans les écoulements granulaires. Dans de tels écoulements, les collisions entre particules induisent des fluctuations de vitesse des grains qui entraînent le mouvement des particules d'une manière analogue à la diffusion thermique dans les gaz denses, ou encore à la dispersion induite par les tourbillons dans les écoulements turbulents. Une telle diffusion induite par cisaillement est importante pour les applications dans les écoulements granulaires industriels et naturels impliquant le mélange et la ségrégation. Une étude a été menée sur l'influence de la pression de confinement sur la diffusion et la ségrégation des particules dans les flux granulaires sous cisaillement. A l'aide de simulations par DEM, les chercheurs ont constaté que la diffusion est indépendante de la pression de surcharge tandis que la ségrégation dépend fortement de cette dernière. Ils ont développé un modèle continu qui intègre ces observations et ont montré que ce modèle prédit avec précision les comportements observés dans les simulations DEM.

Notre problématique se concentre sur la compréhension des mécanismes de mélange des grains dans un milieu granulaire monodisperse, où tous les grains ont la même taille (en contraste avec les milieux polydisperses où les grains ont des tailles va-

riées). Nous focalisons particulièrement notre attention sur le processus de diffusion. Avant d'examiner en détail l'équation de diffusion étudiée, nous avons réalisé une manipulation expérimentale pour approfondir notre compréhension du phénomène. Pour cela, nous avons placé deux couches de 50 millimètres de billes de polipropylène de la même taille (6 millimètres) de couleurs différentes (rouge et vert) dans une cellule de cisaillement, créant ainsi un milieu granulaire monodisperse. Les limites verticales de la cellule sont constituées de deux cylindres coaxiaux ayant des diamètres extérieurs respectifs de 90 mm et 200 mm. La paroi supérieure peut se déplacer librement verticalement mais ne peut pas tourner. On pose sur cette paroi un certain nombre de masses d'acier, pour appliquer une charge variable sur le matériau granulaire. La paroi inférieure se déplace à une vitesse constante pour créer un cisaillement. Les parois latérales sont lisses et les parois inférieure et supérieure sont rugueuses. [1]

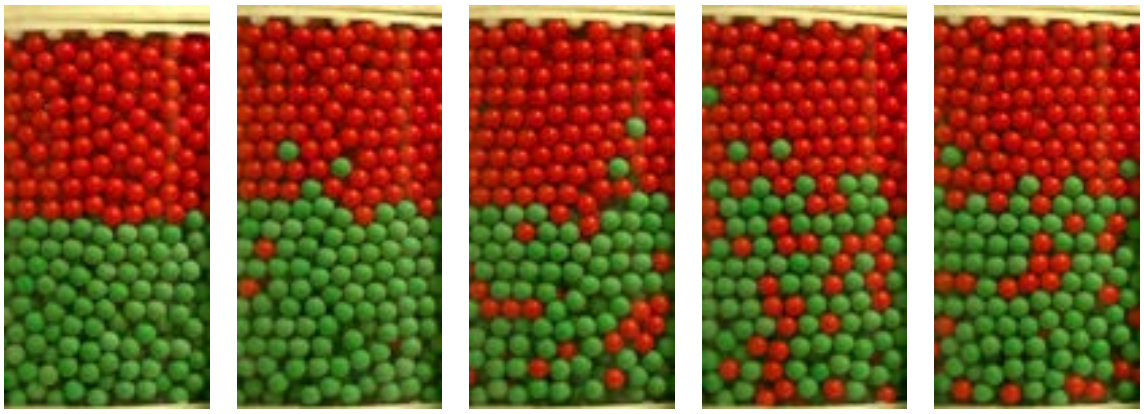


(a) (b)

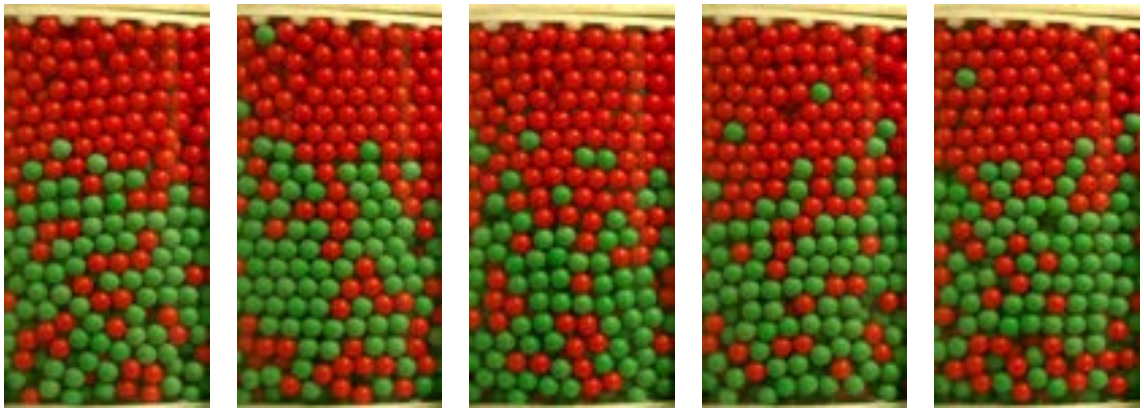
FIGURE 1.1 – Détails sur la cellule de cisaillement
(a) Photo de la manipulation
(b) Schéma de la cellule. La plaque supérieure libre se déplace verticalement, mais est soumise à une force $M\vec{g}$ et la plaque de fond tourne à une vitesse angulaire $\vec{V}$ [1]

.

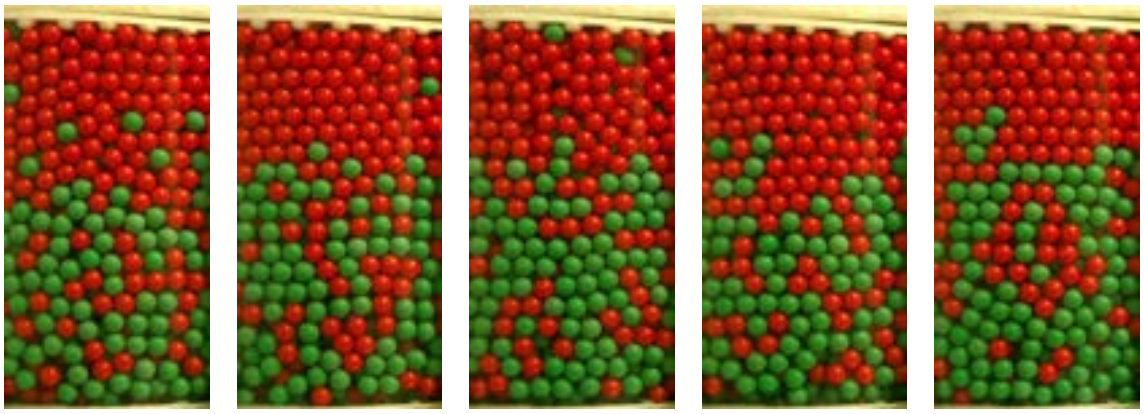
Voici des images illustrant l'évolution des particules à différents moments :



(a) 0 min (b) 5 min (c) 10 min (d) 15 min (e) 20 min



(f) 25 min (g) 30 min (h) 35 min (i) 40 min (j) 45 min



(k) 50 min (l) 55 min (m) 1 h (n) 1 h 5 min (o) 1 h 10 min

FIGURE 1.2 – Évolution du processus au cours du temps

Ces images montrent que le mélange des particules n'est pas uniforme sur toute la hauteur. Pour analyser ce phénomène de diffusion dans un cadre plus général, nous nous intéressons à l'équation de diffusion non stationnaire suivante :

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial z}\left(D(z)\frac{\partial c}{\partial z}\right)$$

avec :

— c la concentration de grains

— D le coefficient de diffusion, non uniforme

Des conditions aux limites de Neumann sont imposées. Physiquement, les conditions de Neumann représentent une condition de "flux imposé". Cela est particulièrement adapté à notre cas, où nous ne connaissons pas la proportion de billes d'un certain type en contact avec la paroi, mais savons que les grains ne peuvent pas sortir du système. Ainsi, le flux aux frontières est nul.
Remarquons que si D est une constante, nous retrouvons l'équation de la chaleur non stationnaire classique.

L'objectif de ce stage est d'estimer le coefficient de diffusion non uniforme D par inversion du problème, à partir de données obtenues par simulation numérique discrète. Dans un premier temps, il s'agira d'estimer la concentration de grains c en supposant que D est connu. Ensuite, dans un cadre de problème inverse, les valeurs de concentration obtenues seront utilisées pour tester une méthode d'estimation de D. Si la méthode fonctionne, il sera alors possible d'estimer D pour n'importe quelles données de c déjà obtenues.

1.2 Vocabulaire

Milieu granulaire : Un milieu granulaire est un ensemble de particules solides, appelées grains, qui interagissent entre elles par des forces de contact. Ces particules peuvent avoir différentes tailles et formes, allant de quelques micromètres à plusieurs centimètres. Les exemples typiques de milieux granulaires incluent le sable, le gravier, les grains de café, ou même des poudres comme la farine.

Les milieux granulaires présentent des propriétés à la fois solides et fluides, selon les conditions. Par exemple, un tas de sable se comporte comme un solide lorsque vous marchez dessus, mais il peut s'écouler comme un liquide si vous le versez d'un récipient. Ces systèmes sont complexes et ne peuvent pas être décrits simplement par les lois de la mécanique des fluides ou des solides traditionnels.
Les milieux granulaires sont d'intérêt dans de nombreux domaines, tels que l'ingénierie civile, la géophysique, l'agriculture, et l'industrie pharmaceutique, en raison de leurs propriétés uniques et des défis qu'ils posent en termes de manipulation et de modélisation.

Diffusion : Le processus par lequel les particules se déplacent d'une région de concentration plus élevée à une région de concentration plus faible, résultant en une répartition plus uniforme des particules. En général, la diffusion est le résultat d'un mouvement aléatoire de particules.

Cisaillement : Un type de déformation dans lequel les différentes couches d'un matériau se déplacent les unes par rapport aux autres dans des directions parallèlement opposées à une surface ou à une interface. En contexte granulaire, le cisaillement se produit lorsque des particules adjacentes glissent ou roulent les unes sur les autres sous l'effet d'une force appliquée.

Ségrégation : La séparation est la répartition inégale de différents composants d'un matériau. En contexte granulaire, cela se produit lorsque des particules de tailles ou de densités différentes se regroupent ou se séparent les unes des autres sous l'effet de forces internes ou externes.

Flux granulaire : La quantité de matériau granulaire qui traverse une surface donnée par unité de temps. Il peut être exprimé en masse, volume ou nombre de particules.

Pression de confinement : La pression exercée sur un matériau dans une direction donnée en raison de la présence d'autres matériaux environnants ou de contraintes externes. En contexte granulaire, il s'agit de la pression appliquée sur les particules d'un matériau granulaire résultant de la superposition des couches de particules et de toute pression externe appliquée sur le système.

Pression de surcharge : Il s'agit d'une pression supplémentaire exercée sur les particules d'un matériau granulaire due à une charge externe appliquée.

Chapitre 2

Etude du problème

Soit Ω un domaine borné de $\mathbb{R}$. Soit $T > 0$ un temps final. On considère le problème suivant :

$$\begin{cases} \frac{\partial c}{\partial t} - \frac{\partial}{\partial z}\left(D(z)\frac{\partial c}{\partial z}\right) = 0 \quad \forall z \in \Omega, \quad \forall t \in]0, T] \\ c(z,0) = c_0(z) \quad \forall z \in \Omega \\ \frac{\partial c}{\partial n} = 0 \end{cases}$$

On suppose que $D \in \mathcal{C}^0(\Omega)$ est bornée telle que

$$0 < D_0 \leq D(z) \leq D_\infty < +\infty$$

et que $c_0 \in L^2(\Omega)$.

2.1 Existence et unicité de la solution du problème

Ici nous montrons que le problème possède une unique solution sur un espace bien défini. Pour cela, posons V un espace de fonctions tests à déterminer.
$\forall v \in V$ on effectue le produit scalaire dans $L^2(\Omega)$ afin d'obtenir :

$$\int_\Omega \frac{\partial c}{\partial t} v dz - \int_\Omega \frac{\partial}{\partial z}\left(D(z)\frac{\partial c}{\partial z}\right) v dz = 0$$

En appliquant la formule de Green sur la deuxième intégrale nous obtenons :

$$\int_\Omega \frac{\partial c}{\partial t} v dz + \int_\Omega D(z) \frac{\partial c}{\partial z} \frac{\partial v}{\partial z} dz - \int_{\partial\Omega} D(z) \frac{\partial c}{\partial z} \cdot n v d\sigma = 0$$

Comme $\frac{\partial c}{\partial n} = 0$ et que Ω et $v(z)$ ne varient pas avec t alors :

$$\frac{d}{dt} \int_\Omega c(z,t) v(z) dz + \int_\Omega D(z) \frac{\partial c}{\partial z} \frac{\partial v}{\partial z} dz = 0 \tag{2.1}$$

Exploitant le fait que z et t jouent des rôles très différents, nous séparons ces variables en considérant désormais la solution $c(z,t)$ comme une fonction de t à valeurs dans un espace de fonctions définies sur Ω.

Plus précisément, si l'on se donne un temps final $T > 0$ (éventuellement $+\infty$) on considère que c est définie par

$$c :]0, T[\to V, \quad t \mapsto c(t, \cdot)$$

et nous continuerons à noter $c(z,t)$ la valeur $c(t)(z)$.

En choisissant $V = H^{1,*}(\Omega) = \{v \in H^1(\Omega), \int_\Omega v dz = 0\}$, on peut alors mettre (2.1) sous la forme d'une sorte d'équation différentielle ordinaire en t. On obtient ainsi la formulation variationnelle suivante :

(PV) : Trouver c fonction de $]0, T[$ à valeurs dans $H^{1,*}(\Omega)$ telle que

$$\begin{cases} \frac{d}{dt}(c(t), v)_{L^2(\Omega)} + a(c(t), v) = 0 \quad \forall v \in H^{1,*}(\Omega), \quad 0 < t < T \\ c(t = 0) = c_0 \end{cases}$$

Remarque 2.1.1. Le choix de $H^{1,*}$ est puissant pour sa norme $\|v\|_{H^{1,*}(\Omega)} = \|\nabla v\|_{L^2(\Omega)}$ qui est la même que celle de H^1_0. Cette norme est nécessaire pour démontrer la coercivité.

Définition 2.1.1. Soit X un espace de Hilbert, ou plus généralement, un espace de Banach défini sur Ω (typiquement, $X = L^2(\Omega), H^1_0(\Omega), H^1(\Omega)$, ou $\mathcal{C}(\bar{\Omega})$). Soit un

temps final $0 < T \leq +\infty$. Pour un entier $k \geq 0$, on note $\mathcal{C}^k([0,T];X)$ l'espace des fonctions k fois continûment différentiables de $[0,T]$ dans X. Si on note $\|v\|_X$ la norme dans X, il est classique que $\mathcal{C}^k([0,T];X)$ est un espace de Banach pour la norme

$$\|v\|_{\mathcal{C}^k([0,T];X)} = \sum_{m=0}^{k} \sup_{0\leq t\leq T} \|\frac{d^m v}{dt^m}(t)\|_X$$

On note $L^2(]0,T[;X)$ l'espace des fonctions de $]0,T[$ dans X telles que la fonction $t \mapsto \|v(t)\|_X$ soit mesurable et de carré intégrable, i.e. que

$$\|v\|_{L^2(]0,T[;X)} = \left(\int_0^T \|v(t)\|_X^2 dt\right)^{\frac{1}{2}} < +\infty$$

Muni de cette norme, $L^2(]0,T[;X)$ est aussi un espace de Banach. De plus, si X est un espace de Hilbert, alors $L^2(]0,T[;X)$ est un espace de Hilbert pour le produit scalaire

$$(u,v)_{L^2(]0,T[;X)} = \int_0^T (u(t),v(t))_X dt$$

Théorème 2.1.1. *Soient V et H deux espaces de Hilbert tels que $V \subset H$ avec injection compacte et V dense dans H. Soit a une forme bilinéaire symétrique continue et coercive dans $V \times V$. Soit un temps final $T > 0$, une donnée initiale $u_0 \in H$ et un terme source $f \in L^2(]0,T[;H)$. Alors le problème*

$$\begin{cases} \frac{d}{dt}(u(t),v)_H + a(u(t),v) = (f(t),v)_H \quad \forall v \in V, \quad 0 < t < T \\ u(t=0) = u_0 \end{cases}$$

(où l'équation a lieu au sens faible dans $]0,T[$) possède une unique solution $u \in L^2(]0,T[;V) \cap \mathcal{C}([0,T];H)$. De plus, il existe une constante $C(\Omega) > 0$ telle que

$$\|u\|_{L^2(]0,T[;V)} + \|u\|_{\mathcal{C}([0,T];H)} \leq C(\Omega)\left(\|u_0\|_H + \|f\|_{L^2(]0,T[;H)}\right)$$

Ici, $H = L^2(\Omega)$, $V = H^{1,*}(\Omega)$ et du fait que $H^{1,*}(\Omega)$ est un sous-espace fermé de $H^1(\Omega)$ on en déduit que $H^{1,*}(\Omega)$ est dense dans $L^2(\Omega)$ et par le théorème de Rellich

que l'injection de $H^{1,*}(\Omega)$ dans $L^2(\Omega)$ est compacte. On a

$$a(c(t), v) = \int_\Omega D(z) \frac{\partial c}{\partial z} \frac{\partial v}{\partial z} dz$$

a est clairement symétrique, et bilinéaire par linéarité de l'intégrale et de $\partial_z \cdot$.

Montrons qu'elle est continue et coercive dans $H^{1,*}(\Omega)$. On a

$$\begin{aligned}
|a(c(t), v)| &\leq \int_\Omega |D(z)| |\frac{\partial c}{\partial z}| |\frac{\partial v}{\partial z}| \, dz && \text{par inégalité triangulaire} \\
&\leq D_\infty \| \frac{\partial c}{\partial z} \|_{L^2(\Omega)} \| \frac{\partial v}{\partial z} \|_{L^2(\Omega)} && \text{par Cauchy-Schwarz et D est continue} \\
&= D_\infty \|c\|_{H^{1,*}(\Omega)} \|v\|_{H^{1,*}(\Omega)} && \text{par définition de la norme } H^{1,*}
\end{aligned}$$

Donc a est continue. De plus on a

$$a(v, v) = \int_\Omega D(z) |\frac{\partial v}{\partial z}|^2 dz \geq D_0 \| \frac{\partial v}{\partial z} \|^2_{L^2(\Omega)} = D_0 \|v\|^2_{H^{1,*}(\Omega)}$$

Donc a est coercive.

Ainsi par le théorème 1.1.1, on en déduit qu'il existe une unique solution $c \in L^2(]0, T[; H^{1,*}(\Omega)) \cap \mathcal{C}([0, T]; L^2(\Omega))$ au problème (PV).

2.2 Schémas aux différences finies pour le problème

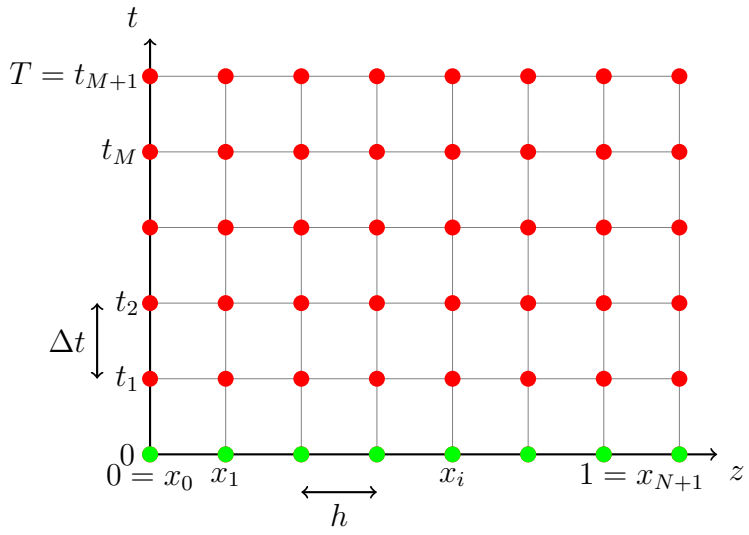
Soit $T > 0$ le temps final. Nous considérons le domaine $\Omega = [0, 1] \times [0, T]$.

Transformons l'expression de l'équation de départ comme suit :

$$\frac{\partial c}{\partial t} - \frac{\partial}{\partial z} \left(D(z) \frac{\partial c}{\partial z} \right) = 0 \Leftrightarrow \frac{\partial c}{\partial t} - D'(z) \frac{\partial c}{\partial z} - D(z) \frac{\partial^2 z}{\partial z^2} = 0 \tag{2.2}$$

Pour approcher la solution de notre problème, nous utilisons la méthode des différences finies. Nous effectuons un maillage de l'espace et du temps dans le domaine Ω, en définissant les pas de discrétisation en espace et en temps respectivement comme suit :

$$\begin{cases} h = \Delta z = \frac{1}{N+1}, \quad z_i = ih, \quad i = 0, ..., N+1 \\ \Delta t = \frac{T}{M+1}, \quad t_j = j\Delta t, \quad j = 0, ...M+1 \end{cases}$$



(z_i, t_j) sont les noeuds du maillage où l'on cherche une approximation de la solution en ces noeuds, i.e. $c_i^j \approx c(z_i, t_j)$. On notera aussi $D_i \approx D(z_i)$.

Soit

$$C^{(j)} = \begin{pmatrix} c_0^j \\ c_1^j \\ \vdots \\ c_{N+1}^j \end{pmatrix}$$

le vecteur des inconnues à l'instant t_j et

$$C^{(0)} = \begin{pmatrix} c_0^0 \\ c_1^0 \\ \vdots \\ c_{N+1}^0 \end{pmatrix}$$

le vecteur de la donnée initiale (à $t = 0$).

On considère l'approximation des dérivées aux noeuds (z_i, t_j) :

— $\frac{\partial^2 c}{\partial z^2}(z_i, t_j) \approx \frac{c_{i-1}^j - 2c_i^j + c_{i+1}^j}{h^2}$

— $\frac{\partial c}{\partial z}(z_i, t_j) \approx \frac{c_{i+1}^j - c_{i-1}^j}{2h}$

— $\frac{\partial c}{\partial t}(z_i, t_j) \approx \begin{cases} \frac{c_i^j - c_i^{j-1}}{\Delta t} & \text{(pour le schéma implicite)} \\ \frac{c_i^{j+1} - c_i^j}{\Delta t} & \text{(pour le schéma explicite)} \end{cases}$

2.2.1 Le schéma explicite

Dans ce paragraphe, nous allons présenter le schéma explicite et examiner sa consistance, sa stabilité et sa convergence.

On considère l'approximation $\frac{\partial c}{\partial t}(z_i, t_j) \approx \frac{c_i^{j+1} - c_i^j}{\Delta t}$.

Ainsi nous pouvons réécrire l'équation (2.2) avec les approximations successives aux noeuds du maillage :

$$\frac{c_i^{j+1} - c_i^j}{\Delta t} - D_i' \frac{c_{i+1}^j - c_{i-1}^j}{2h} - D_i \frac{c_{i-1}^j - 2c_i^j + c_{i+1}^j}{h^2} = 0 \tag{2.3}$$

d'où le schéma itératif : Pour $i = 0, ..., N+1$ et $j = 1, ..., M+1$

$$c_i^{j+1} = c_i^j - \frac{\Delta t}{2h} D_i'(c_{i-1}^j - c_{i+1}^j) - \frac{\Delta t}{h^2} D_i(-c_{i-1}^j + 2c_i^j - c_{i+1}^j) \tag{2.4}$$

ou sous forme matricielle :

$$C^{(j+1)} = C^{(j)} - \Delta t B_{h,i} C^{(j)} - \Delta t A_{h,i} C^{(j)} \tag{2.5}$$

avec

$$A_{h,i} = \frac{D_i}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{pmatrix} \quad \text{et} B_{h,i} = \frac{D_i'}{2h} \begin{pmatrix} 0 & -1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & -1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

C'est un schéma explicite car on calcule $C^{(j+1)}$ directement à partir de $C^{(j)}$.

Proposition 2.2.1. *Sous la condition* $\Delta t \leq \min\left(\frac{h^2}{2D_\infty}, \frac{h}{D'_\infty}\right)$, *le schéma est stable en* $\|\cdot\|_\infty$ *:* $\forall j$ *tel que* $j\Delta t \leq T$, $\|C^{(j)}\|_\infty \leq \|C^{(0)}\|_\infty$

Démonstration. Reprenons le schéma (1.3).

$$c_i^{j+1} = c_i^j - \frac{\Delta t}{2h} D'_i (c_{i-1}^j - c_{i+1}^j) - \frac{\Delta t}{h^2} D_i (-c_{i-1}^j + 2c_i^j - c_{i+1}^j)$$

$$\Leftrightarrow c_i^{j+1} = \left(\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D'_i\right) c_{i-1}^j + \left(1 - 2\frac{\Delta t}{h^2} D_i\right) c_i^j + \left(\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D'_i\right) c_{i+1}^j$$

Par passage à $|\cdot|$:

$$|c_i^{j+1}| \leq |\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D'_i||c_{i-1}^j| + |1 - 2\frac{\Delta t}{h^2} D_i||c_i^j| + |\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D'_i||c_{i+1}^j|$$

Sous la condition $\Delta t \leq \min\left(\frac{h^2}{2D_\infty}, \frac{h}{D'_\infty}\right)$:

$$\begin{aligned} |c_i^{j+1}| &\leq \left(\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D'_i\right) \|C^{(j)}\|_\infty + \left(1 - 2\frac{\Delta t}{h^2} D_i\right) \|C^{(j)}\|_\infty + \left(\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D'_i\right) \|C^{(j)}\|_\infty \\ &\leq \|C^{(j)}\|_\infty \end{aligned}$$

Donc

$$\|C^{(j+1)}\|_\infty \leq \|C^{(j)}\|_\infty$$

. Par récurrence sur $j = 1, ..., M+1$ on en déduit que

$$\|C^{(j)}\|_\infty \leq \|C^{(0)}\|_\infty$$

d'où la stabilité du schéma en $\|\cdot\|_\infty$.

□

Définition 2.2.1. On définit $\Pi_h c(t)$ le vecteur de la solution exacte aux points z_i à l'instant t :

$$\Pi_h c(t) = \begin{pmatrix} c(z_0,t) \\ c(z_1,t) \\ \vdots \\ c(z_{N+1},t) \end{pmatrix}$$

et $\epsilon_h(c)^{(j)}$ l'erreur de consistance à l'itération j :

$$\epsilon_h(c)^{(j)} = \frac{\Pi_h c(t_{j+1}) - \Pi_h c(t_j)}{\Delta t} + B_{h,i}\Pi_h c(t_j) + A_{h,i}\Pi_h c(t_j)$$

Proposition 2.2.2. *Si c solution du problème est de classe $\mathcal{C}^4$ relativement à z et de classe $\mathcal{C}^2$ relativement à t, alors le schéma est consistant en $\|\cdot\|_\infty$, d'ordre 3 en espace et d'ordre 1 en temps.*

Démonstration. Pour $i = 0, ..., N+1$ on a

$$\begin{aligned}\epsilon_h(c)_i^{(j)} &= \frac{1}{\Delta t}\left(c(z_i,t_{j+1}) - c(z_i,t_j)\right) + B_{h,i}\Pi_h c_i^{(j)} + A_{h,i}\Pi_h c_i^{(j)} \\ &= \frac{1}{\Delta t}\left(c(z_i,t_j+\Delta t) - c(z_i,t_j)\right) + \frac{D'(z_i)}{2h}\left(c(z_{i-1},t_j) - c(z_{i+1},tj)\right) + \\ &\frac{D(z_i)}{h^2}\left(-c(z_{i-1},t_j) + 2c(z_i,t_j) - c(z_{i+1},t_j)\right)\end{aligned}$$

c étant assez régulière, on peut effectuer des développements limités de c au voisinage de 0. Ainsi :

$$c(z_i,t_j+\Delta t) = c(z_i,t_j) + \Delta t\frac{\partial c}{\partial t}(z_i,t_j) + \frac{(\Delta t)^2}{2}\frac{\partial^2 c}{\partial t^2}(z_i,\gamma_j), \gamma_j \in [t_j, t_j+\Delta t]$$

$$\implies \frac{1}{\Delta t}\left(c(z_i,t_j+\Delta t) - c(z_i,t_j)\right) = \frac{\partial c}{\partial t}(z_i,t_j) + \frac{\Delta t}{2}\frac{\partial^2 c}{\partial t^2}(z_i,\gamma_j), \gamma_j \in [t_j, t_j+\Delta t]$$

De même on a

$$c(z_{i-1},t_j) = c(z_i - h,t_j) = c(z_i,t_j) - h\frac{\partial c}{\partial z}(z_i,t_j) + \frac{h^2}{2}\frac{\partial^2 c}{\partial z^2}(z_i,t_j) - \frac{h^3}{6}\frac{\partial^3 c}{\partial z^3}(z_i,t_j) + \frac{h^4}{24}\frac{\partial^4 c}{\partial z^4}(\alpha_i,t_j)$$

où $\alpha_i \in [z_i - h, z_i]$

Et

$$c(z_{i+1},t_j) = c(z_i{+}h,t_j) = c(z_i,t_j){+}h\frac{\partial c}{\partial z}(z_i,t_j){+}\frac{h^2}{2}\frac{\partial^2 c}{\partial z^2}(z_i,t_j){+}\frac{h^3}{6}\frac{\partial^3 c}{\partial z^3}(z_i,t_j){+}\frac{h^4}{24}\frac{\partial^4 c}{\partial z^4}(\beta_i,t_j)$$

où $\beta_i \in [z_i, z_i + h]$

Donc

$$\frac{D'(z_i)}{2h}\left(c(z_{i-1},t_j) - c(z_{i+1},t_j)\right) = \frac{D'(z_i)}{2h}\left(-2h\frac{\partial c}{\partial z}(z_i,t_j) - \frac{h^3}{3}\frac{\partial^3 c}{\partial z^3}(z_i,t_j) + \frac{h^4}{24}\frac{\partial^4 c}{\partial z^4}(\alpha_i,t_j) - \frac{h^4}{24}\frac{\partial^4 c}{\partial z^4}(\beta_i,t_j)\right)$$

Et

$$\frac{D(z_i)}{h^2}\left(-c(z_{i-1},t_j) + 2c(z_i,t_j) - c(z_{i+1},t_j)\right) = D(z_i)\left(-\frac{\partial^2 c}{\partial z^2}(z_i,t_j) - \frac{h^2}{12}\frac{\partial^4 c}{\partial z^4}(\phi_i,t_j)\right)$$

où $\phi_i \in [z_i - h, z_i + h]$

En remplaçant les expressions précédentes dans $\epsilon_h(c)_i^{(j)}$ on obtient :

$$\begin{aligned}\epsilon_h(c)_i^{(j)} = {} & \frac{\partial c}{\partial t}(z_i,t_j) + \frac{\Delta t}{2}\frac{\partial^2 c}{\partial t^2}(z_i,\gamma_j) - D'(z_i)\frac{\partial c}{\partial z}(z_i,t_j) \\ & - D'(z_i)\frac{h^2}{6}\frac{\partial^3 c}{\partial z^3}(z_i,t_j) + D'(z_i)\frac{h^3}{48}\frac{\partial^4 c}{\partial z^4}(\alpha_i,t_j) - D'(z_i)\frac{h^3}{48}\frac{\partial^4 c}{\partial z^4}(\beta_i,t_j) \\ & - D(z_i)\frac{\partial^2 c}{\partial z^2}(z_i,t_j) - D(z_i)\frac{h^2}{12}\frac{\partial^4 c}{\partial z^4}(\phi_i,t_j)\end{aligned}$$

En utilisant le fait que $c(z_i,t_j)$ est solution de $\frac{\partial c}{\partial t} - D'(z)\frac{\partial c}{\partial z} - D(z)\frac{\partial^2 z}{\partial z^2} = 0$ et en regroupant les termes en $D'(z_i)$ on obtient :

$$\begin{aligned}\epsilon_h(c)_i^{(j)} = {} & \frac{\Delta t}{2}\frac{\partial^2 c}{\partial t^2}(z_i,\gamma_j) - D'(z_i)\left(\frac{h^2}{6}\frac{\partial^3 c}{\partial z^3}(z_i,t_j) - \frac{h^3}{48}\frac{\partial^4 c}{\partial z^4}(\alpha_i,t_j) + \frac{h^3}{48}\frac{\partial^4 c}{\partial z^4}(\beta_i,t_j)\right) \\ & - D(z_i)\frac{h^2}{12}\frac{\partial^4 c}{\partial z^4}(\phi_i,t_j)\end{aligned}$$

Donc

$$
\begin{aligned}
|\epsilon_h(c)_i^{(j)}| &\leq \frac{\Delta t}{2}\left|\frac{\partial^2 c}{\partial t^2}(z_i,\gamma_j)\right| + D'_\infty\left(\frac{h^2}{6}\left|\frac{\partial^3 c}{\partial z^3}(z_i,t_j)\right| + \frac{h^3}{48}\left|\frac{\partial^4 c}{\partial z^4}(\alpha_i,t_j)\right| + \frac{h^3}{48}\left|\frac{\partial^4 c}{\partial z^4}(\beta_i,t_j)\right|\right) \\
&\quad + D_\infty\frac{h^2}{12}\left|\frac{\partial^4 c}{\partial z^4}(\phi_i,t_j)\right| \\
&\leq \frac{\Delta t}{2}\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^2 c}{\partial t^2}(z,t)\right| + D'_\infty\left(\frac{h^2}{6}\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^3 c}{\partial z^3}(z,t)\right| + \frac{h^3}{48}\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}(z,t)\right| + \right. \\
&\quad \left.\frac{h^3}{48}\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}(z,t)\right|\right) + D_\infty\frac{h^2}{12}\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}(z,t)\right| \\
&\leq \left(\frac{\Delta t}{2} + D'_\infty\frac{h^2}{6} + D'_\infty\frac{h^3}{24} + D_\infty\frac{h^2}{12}\right)\max\left(\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^2 c}{\partial t^2}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^3 c}{\partial z^3}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}\right|\right)
\end{aligned}
$$

Donc

$$
|\epsilon_h(c)_i^{(j)}| \leq \frac{1}{2}\left(\Delta t + D'_\infty\frac{h^2}{3} + D'_\infty\frac{h^3}{12} + D_\infty\frac{h^2}{6}\right)\max\left(\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^2 c}{\partial t^2}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^3 c}{\partial z^3}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}\right|\right)
$$

Par passage au max sur $j = 0, ..., M+1$ et sur $i = 0, ..., N+1$:

$$
\max_{0\leq j\leq M+1}\|\epsilon_h(c)^{(j)}\|_\infty \leq \frac{1}{2}\left(\Delta t + D'_\infty\frac{h^2}{3} + D'_\infty\frac{h^3}{12} + D_\infty\frac{h^2}{6}\right)\max\left(\sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^2 c}{\partial t^2}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^3 c}{\partial z^3}\right|, \sup_{\substack{x\in[0,1]\\ t\in[0,T]}}\left|\frac{\partial^4 c}{\partial z^4}\right|\right)
$$

d'où la consistance du schéma d'ordre 3 en espace et 1 en temps.

□

Proposition 2.2.3. *Si la solution c du problème est de classe $\mathcal{C}^4$ en espace et $\mathcal{C}^2$ en temps alors sous la condition $\Delta t \leq \min\left(\frac{h^2}{2D_\infty}, \frac{h}{D'_\infty}\right)$ le schéma est convergent en $\|\cdot\|_\infty$, d'ordre 3 en espace et d'ordre 1 en temps.*

Démonstration. Pour étudier la convergence du schéma, on considère l'erreur $e^{(j)}$ à

l'instant t_j :

$$e^{(j)} = C^{(j)} - \Pi_h c(t_j)$$

On a

$$\begin{cases} C^{(j+1)} = C^{(j)} - \Delta t B_{h,i} C^{(j)} - \Delta t A_{h,i} C^{(j)} \\ \Pi_h c(t_{j+1}) = \Pi_h c(t_j) - \Delta t B_{h_i} \Pi_h c(t_j) - \Delta t A_{h_i} \Pi_h c(t_j) + \Delta t \epsilon_h(c)^{(j)} \end{cases}$$

Donc par différence :

$$e^{(j+1)} = e^{(j)} - \Delta t B_{h,i} e^{(j)} - \Delta t A_{h,i} e^{(j)} - \Delta t \epsilon_h(c)^{(j)}$$

Pour $i = 0, ..., N+1$,

$$\begin{aligned} e_i^{(j+1)} &= e_i^{(j)} - \Delta t B_{h,i} e_i^{(j)} - \Delta t A_{h,i} e_i^{(j)} - \Delta t \epsilon_h(c)_i^{(j)} \\ &= e_i^{(j)} - \frac{\Delta t}{2h} D_i'(e_{i-1}^{(j)} - e_{i+1}^{(j)}) - \frac{\Delta t}{h^2} D_i(-e_{i-1}^{(j)} + 2e_i^{(j)} - e_{i+1}^{(j)}) - \Delta t \epsilon_h(c)_i^{(j)} \\ &= \left(\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D_i'\right) e_{i-1}^{(j)} + \left(1 - 2\frac{\Delta t}{h^2} D_i\right) e_i^{(j)} + \left(\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D_i'\right) e_{i+1}^{(j)} - \Delta t \epsilon_h(c)_i^{(j)} \end{aligned}$$

Par passage à $|\cdot|$ et sous la condition de stabilité on en déduit que :

$$|e_i^{(j+1)}| \leq \left(\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D_i'\right) |e_{i-1}^{(j)}| + \left(1 - 2\frac{\Delta t}{h^2} D_i\right) |e_i^{(j)}| + \left(\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D_i'\right) |e_{i+1}^{(j)}| + \Delta t |\epsilon_h(c)_i^{(j)}|$$

Donc

$$\begin{aligned} \|e^{(j+1)}\|_\infty &\leq \left(\frac{\Delta t}{h^2} D_i - \frac{\Delta t}{2h} D_i'\right) \|e^{(j)}\|_\infty + \left(1 - 2\frac{\Delta t}{h^2} D_i\right) \|e^{(j)}\|_\infty + \left(\frac{\Delta t}{h^2} D_i + \frac{\Delta t}{2h} D_i'\right) \|e^{(j)}\|_\infty \\ &\quad + \Delta t \|\epsilon_h(c)^{(j)}\|_\infty \\ &\leq \|e^{(j)}\|_\infty + \Delta t \|\epsilon_h(c)^{(j)}\|_\infty \end{aligned}$$

Pour $j = 0$: $\|e^{(1)}\|_\infty \leq \|e^{(0)}\|_\infty + \Delta t \|\epsilon_h(c)^{(0)}\|_\infty \leq \Delta t \|\epsilon_h(c)^{(0)}\|_\infty$

Pour $j = 1$:

$$\begin{aligned}\|e^{(2)}\|_\infty &\leq \|e^{(1)}\|_\infty + \Delta t \|\epsilon_h(c)^{(1)}\|_\infty \\ &\leq \Delta t \left(\|\epsilon_h(c)^{(1)}\|_\infty + \|\epsilon_h(c)^{(0)}\|_\infty \right)\end{aligned}$$

Par récurrence sur $j = 0, ..., M+1$ on en déduit que $\forall j$ tel que $j\Delta t \leq T$:

$$\begin{aligned}\|e^{(j)}\|_\infty &\leq \Delta t \left(\|\epsilon_h(c)^{(j-1)}\|_\infty + ... + \|\epsilon_h(c)^{(1)}\|_\infty + \|\epsilon_h(c)^{(0)}\|_\infty \right) \\ &\leq \Delta t j \max_{0 \leq j \leq M+1} \|\epsilon_h(c)^{(j)}\|_\infty \\ &\leq T \max_{0 \leq j \leq M+1} \|\epsilon_h(c)^{(j)}\|_\infty \\ &\leq \frac{T}{2} \left(\Delta t + D'_\infty \frac{h^2}{3} + D'_\infty \frac{h^3}{12} + D_\infty \frac{h^2}{6} \right) \max \left(\sup_{\substack{x \in [0,1] \\ t \in [0,T]}} \left| \frac{\partial^2 c}{\partial t^2} \right|, \sup_{\substack{x \in [0,1] \\ t \in [0,T]}} \left| \frac{\partial^3 c}{\partial z^3} \right|, \sup_{\substack{x \in [0,1] \\ t \in [0,T]}} \left| \frac{\partial^4 c}{\partial z^4} \right| \right) \\ &\leq Const(z,T)(h^3 + \Delta t)\end{aligned}$$

d'où la convergence du schéma en $\| \cdot \|_\infty$, d'ordre 3 en espace et 1 en temps.

□

2.2.2 Le schéma implicite

Cette fois-ci, nous approchons $\frac{\partial c}{\partial t}(z_i, t_j)$ par $\frac{c_i^j - c_i^{j-1}}{\Delta t}$ afin d'obtenir le schéma suivant :

$$C^{(j)} = C^{(j-1)} - \Delta t B_{h,i} C^{(j)} - \Delta t A_{h,i} C^{(j)} \tag{2.6}$$

$$\Leftrightarrow \left(I + \Delta t B_{h,i} + \Delta t A_{h,i} \right) C^{(j)} = C^{(j-1)} \tag{2.7}$$

Ce schéma est implicite car on doit résoudre un système pour calculer $C^{(j)}$.

Posons $Y_{h,i} = I + \Delta t B_{h,i} + \Delta t A_{h,i}$ et montrons qu'elle est inversible.

$$\forall v \neq 0, \quad \langle Y_{h,i} v, v \rangle = \langle v, v \rangle + \Delta t \langle B_{h,i} v, v \rangle + \Delta t \langle A_{h,i} v, v \rangle > 0$$

car $v \neq 0$, et $A_{h,i}$ et $B_{h,i}$ sont définies positives.

Donc $Y_{h,i}$ est définie positive, donc inversible, donc le schéma est bien défini.

Proposition 2.2.4. *Soit $\epsilon_h(c)^{(j)}$ l'erreur de consistance :*

$$\epsilon_h(c)^{(j)} = \frac{\Pi_h c(t_j) - \Pi_h c(t_{j-1})}{\Delta t} + B_{h,i}\Pi_h c(t_j) + A_{h,i}\Pi_h c(t_j)$$

Si c est de classe $\mathcal{C}^4$ en espace et $\mathcal{C}^2$ en temps, alors le schéma est consistant en $\|\cdot\|_\infty$, d'ordre 3 en espace et 1 en temps.

Démonstration. C'est comme pour le schéma explicite.

□

Proposition 2.2.5. *Le schéma est stable et convergent en $\|\cdot\|_\infty$ avec même ordre de convergence que la consistance, sans condition sur h et Δt.*

2.2.3 Résultats numériques obtenus

Nous commençons par tester nos schémas en utilisant un nombre de points intérieurs $N = 98$ et un temps final $T = 0.01$. Pour la méthode explicite, le pas de temps est fixé à $\Delta t = \frac{h^2}{2D_\infty}$ afin de satisfaire la condition de stabilité. Pour la méthode implicite, nous fixons $\Delta t = 0.001$.

La condition initiale choisie est

$$c_0(z) = \begin{cases} 1 & \text{si} \quad z > 0.5, \\ 0 & \text{sinon.} \end{cases}$$

Nous appellerons cette condition la condition initiale de référence.

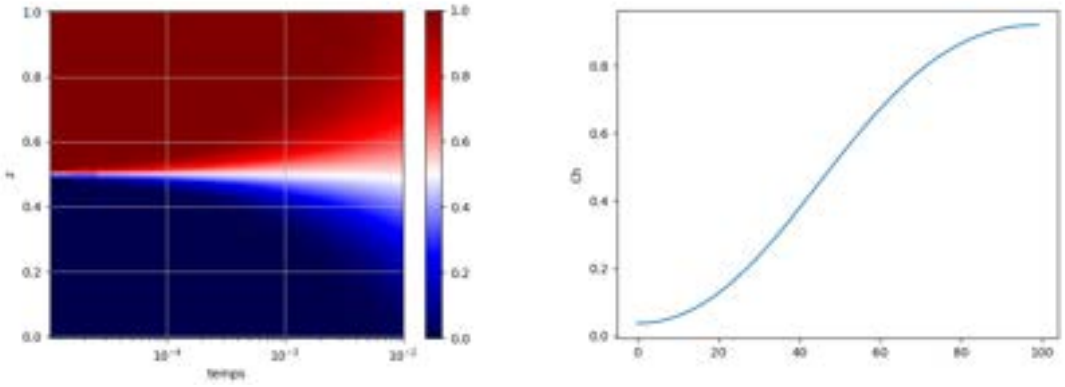
Nous choisissons une diffusion linéaire

$$D(z) = D_0 + (D_\infty - D_0)z,$$

avec $D_0 = 2$ et $D_\infty = 5$.

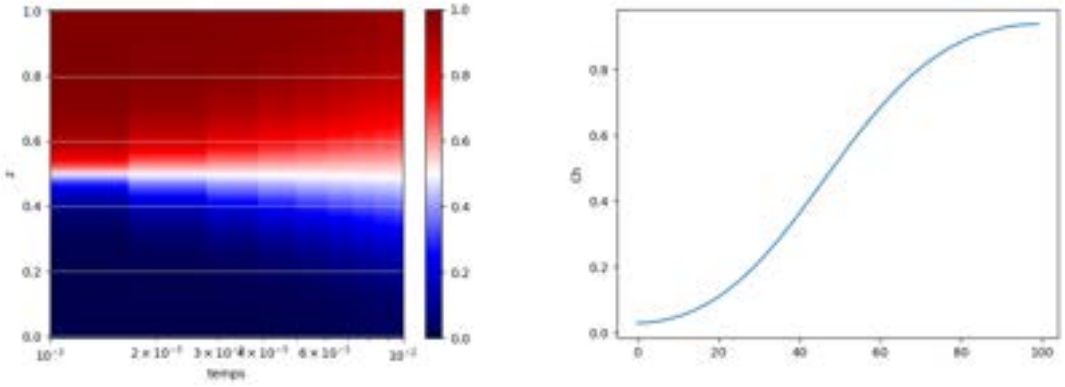
Nous nous attendons à ce que la solution converge vers un état stationnaire une

fois que le mélange des grains soit terminé. Un contour est tracé pour visualiser le mélange, ce qui permet d'observer les valeurs de concentration en fonction du temps et de la position dans l'espace.



(a) Contour (b) Estimation de c au temps final

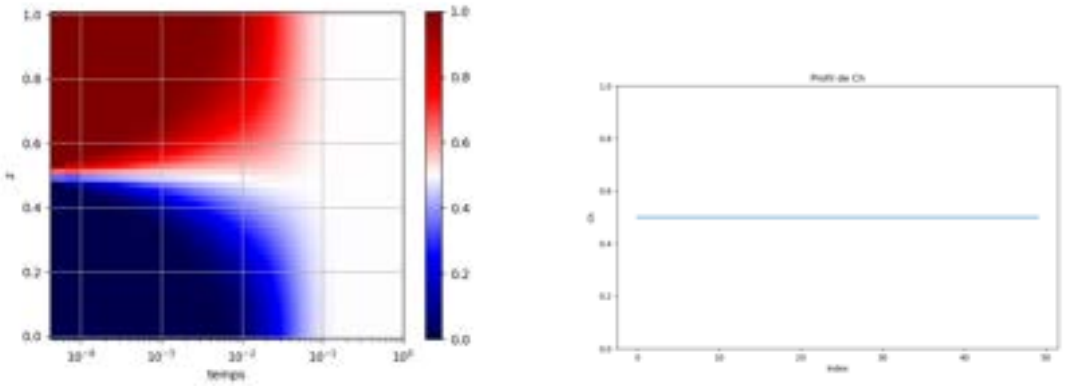
FIGURE 2.1 – Résultats pour le schéma explicite



(a) Contour (b) Estimation de c au temps final

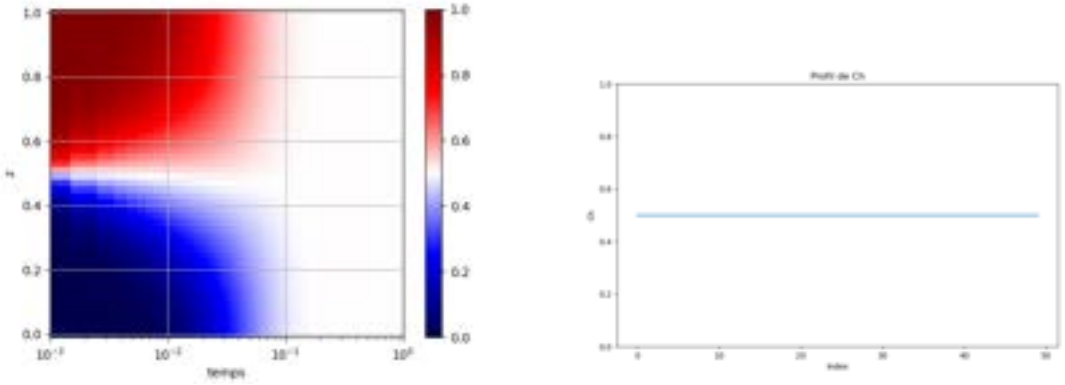
FIGURE 2.2 – Résultats pour le schéma implicite

Le résultat obtenu n'est pas celui attendu, car la solution au temps final n'est pas encore stationnaire. Cela est dû au fait que le temps final T initialement choisi est trop court. Pour remédier à ce problème, nous avons désormais fixé $T = 1$ et $N = 48$, ce qui devrait être suffisant pour obtenir une solution stationnaire. Nous obtenons alors :



(a) Contour (b) Estimation de c au temps final

FIGURE 2.3 – Résultats pour le schéma explicite

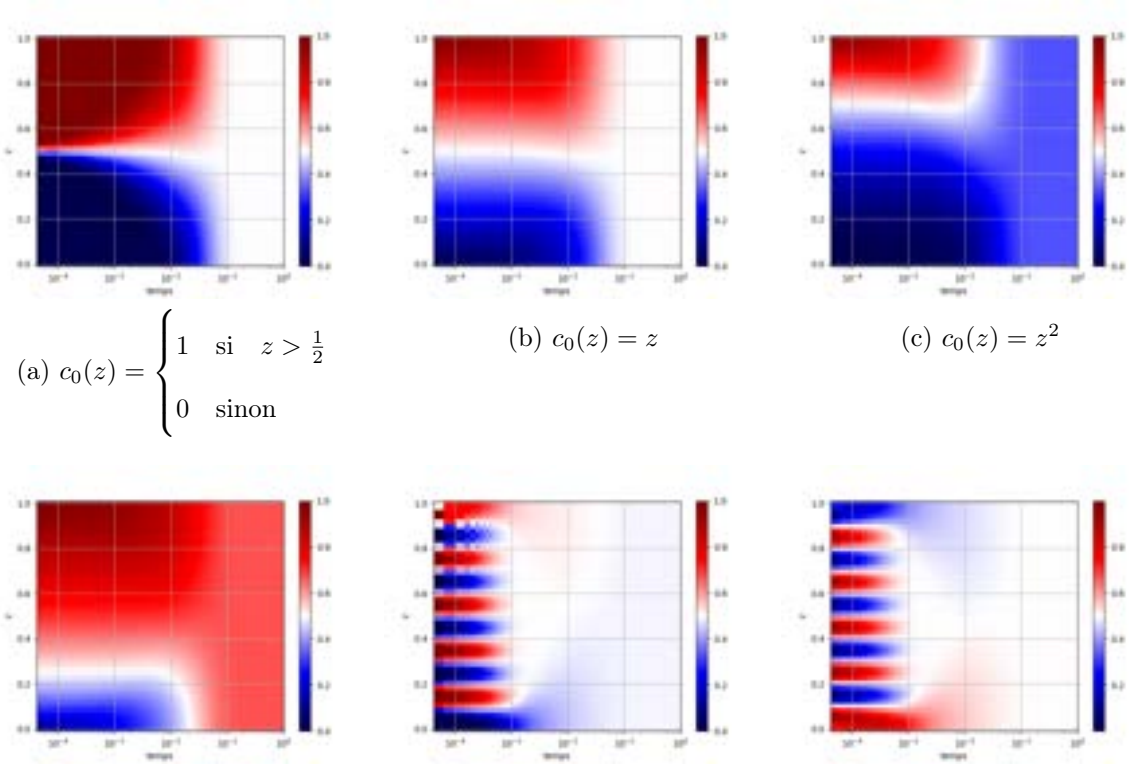


(a) Contour (b) Estimation de c au temps final

FIGURE 2.4 – Résultats pour le schéma implicite

Cette fois-ci, nous obtenons le résultat attendu. Nous observons que, dans le cas implicite, le mélange commence à se produire autour du temps $T = 10^{-2}$ pour $z = 0.8$, tandis qu'il est immédiat pour $z = 0.45$ et $z = 0.55$. Les mêmes observations s'appliquent au cas explicite. Dans les deux méthodes, la solution se stabilise au même temps T, juste en dessous du seuil 10^{-1}.

Pour la suite, nous nous limitons au schéma explicite. Nous donnons d'autres résultats en prenant d'autres profils de conditions initiales c_0 :



(a) $c_0(z) = \begin{cases} 1 & \text{si} \quad z > \frac{1}{2} \\ 0 & \text{sinon} \end{cases}$ (b) $c_0(z) = z$ (c) $c_0(z) = z^2$

(d) $c_0(z) = \sqrt{z}$ (e) dizaine de couches de 0 et 1 (f) $c_0(z) = \frac{1}{2} + \frac{1}{2}\sin(10\pi z)$

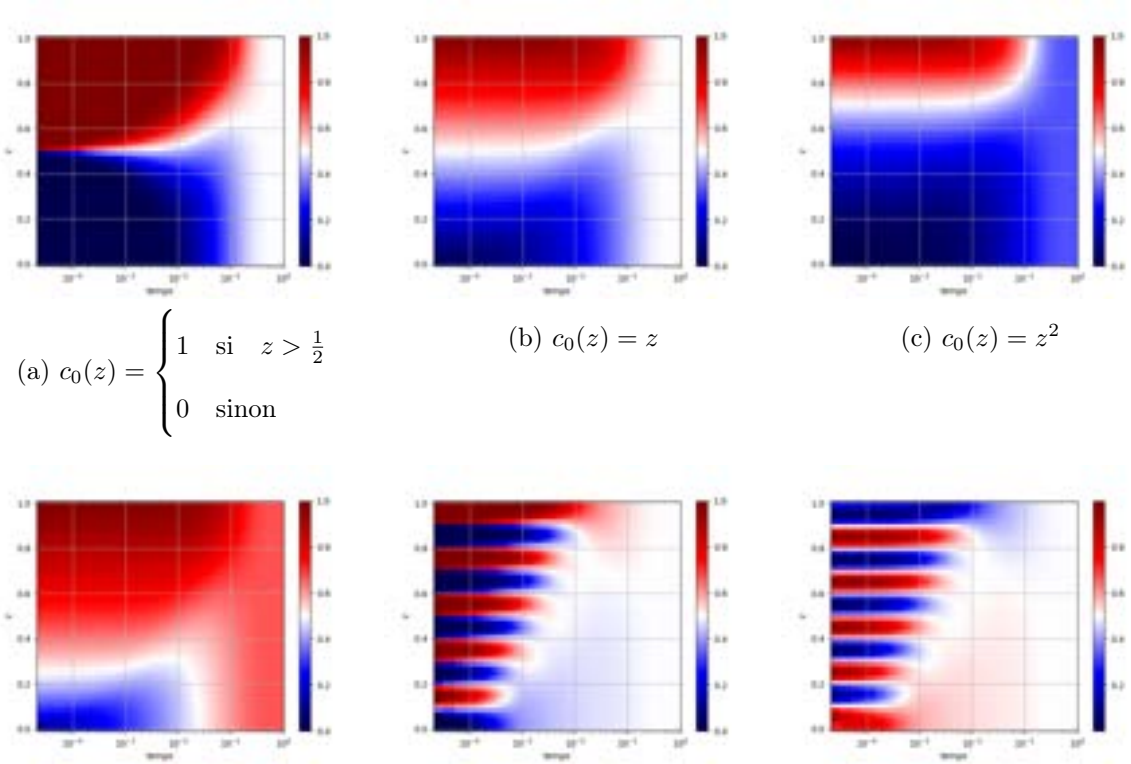
FIGURE 2.5 – Profils de concentration

Ainsi, la solution finit par se stabiliser quel que soit le choix de c_0. Ce choix n'affecte donc pas le résultat final. Cependant, il influence le processus de mélange, en particulier les deux derniers graphes montrent clairement que le mélange est plus prononcé en tout point de l'espace.

Enfin, nous considérons un profil exponentiel pour $D(z)$ défini par

$$D(z) = D_\infty \exp\left(-\frac{z}{z_0}\right)$$

avec $D_\infty = 10$ et $z_0 = 0,25$. Ce choix permet d'obtenir un profil de diffusion qui est beaucoup plus hétérogène. Nous examinons ensuite les résultats obtenus pour différents profils initiaux $c_0(z)$ illustrés dans les figures ci-dessous :



(a) $c_0(z) = \begin{cases} 1 & \text{si} \quad z > \frac{1}{2} \\ 0 & \text{sinon} \end{cases}$ (b) $c_0(z) = z$ (c) $c_0(z) = z^2$

(d) $c_0(z) = \sqrt{z}$ (e) Profil en plusieurs couches de 0 et 1 (f) $c_0(z) = \frac{1}{2} + \frac{1}{2}\sin(10\pi z)$

FIGURE 2.6 – Profils de concentration

Nous observons que la solution se stabilise plus tardivement que dans le cas linéaire, en raison de la nature exponentielle du profil de diffusion $D(z)$. Ce retard dans la stabilisation peut être attribué à une diffusion plus lente dans certaines régions de l'espace, notamment celles où z est élevé, puisque la diffusion est exponentiellement atténuée en fonction de z.

En examinant les différents profils, nous constatons que le mélange des grains est fortement influencé par la forme initiale de c_0. Par exemple, pour les profils où $c_0(z)$ présente une transition abrupte (comme dans le cas du profil en plusieurs couches ou du profil sinusoïdal), le processus de diffusion est moins uniforme.

À l'inverse, pour les profils où $c_0(z)$ est lisse (comme les cas où $c_0(z) = z$ ou $c_0(z) = \sqrt{z}$), le mélange est plus progressif. Cependant, même dans ces cas, la diffusion exponentielle entraîne une stabilisation plus lente par rapport aux situations où la diffusion suit un profil linéaire.

Ces résultats soulignent l'importance de la nature de la diffusion $D(z)$ et de la condition initiale $c_0(z)$ dans la dynamique de mélange des grains, en influençant à la fois la rapidité du mélange et la distribution finale des concentrations.

Chapitre 3

Un problème d'optimisation pour estimer D

Après avoir étudié différents schémas pour l'approximation de la concentration, nous nous tournons maintenant sur l'estimation de la diffusion D à partir de données de concentration obtenues. L'estimation précise de D est essentielle pour mieux comprendre et prédire les phénomènes de diffusion dans les milieux granulaires. Pour ce faire, nous adoptons une approche d'optimisation basée sur le principe du maximum de vraisemblance.

3.1 Le principe du maximum de vraisemblance

3.1.1 Contexte

Soit un modèle M. La vraisemblance du modèle est définit comme la probabilité que ce modèle ait donné lieu à des données observées.

Exemple 3.1.1. Une pièce de monnaie que l'on jette n fois. On cherche à déterminer la probabilité que la pièce tombe sur pile, se basant sur le nombre de fois où elle est tombée sur pile ou face.

Le maximum de vraisemblance est une méthode statistique largement utilisée pour estimer les paramètres d'un modèle (probabiliste). L'idée est de choisir les valeurs

des paramètres qui maximisent la probabilité des données observées étant donné le modèle. Plus formellement, si θ représente le vecteur des paramètres du modèle et x l'ensemble des données observées, alors le maximum de vraisemblance cherche à trouver $\hat{\theta}$ qui maximise la fonction de vraisemblance $L(x|\theta)$:

$$\hat{\theta} = \arg\max_{\theta} L(x|\theta)$$

Définition 3.1.1. La fonction de vraisemblance $L(x|\theta)$ est une fonction de densité de probabilité conditionnelle qui mesure la probabilité de l'échantillon observé x étant donné les paramètres du modèle θ. Pour un ensemble de données indépendantes et identiquement distribuées, la fonction de vraisemblance pour n observations est le produit des densités de probabilité :

$$L(x|\theta) = \prod_{i=1}^{n} f(x_i|\theta) = \prod_{i=1}^{n} f_\theta(x_i)$$

A $x = (x_1, ..., x_n)$ fixé, on cherche à trouver le maximum de cette vraisemblance pour que les probabilités des réalisations observées soient aussi maximum que possible. Ceci est un problème d'optimisation.

3.1.2 Calcul d'un estimateur du maximum de vraisemblance

On utilise généralement le fait que L est dérivable (ce qui n'est pas toujours le cas). Si L admet un maximum global en une valeur $\theta = \hat{\theta}$, alors la dérivée première s'annule en $\theta = \hat{\theta}$ et la dérivée seconde est négative. Réciproquement, si la dérivée première s'annule en $\theta = \hat{\theta}$ et que la dérivée seconde est strictement négative en $\theta = \hat{\theta}$ alors $\theta = \hat{\theta}$ est un maximum local de $L(x|\theta)$. Il est nécessaire de vérifier qu'il s'agit bien d'un maximum global. La vraisemblance étant positive et le log népérien une fonction croissante, il est équivalent et souvent plus simple de maximiser le log népérien de la vraisemblance (le produit se transforme en somme, plus simple à dériver).

• Paramètres du modèle :

$\theta = (D_0, D_\infty, z_0, D_{values})$

• Données observées :

x : les mesures de la concentration de grains à différents points dans le temps et l'espace.

• Estimation des paramètres optimaux :

$\hat{\theta} = (\hat{D_0}, \hat{D_\infty}, \hat{z_0}, \hat{D_{values}})$

3.1.3 La vraisemblance pour notre problème

Étant donné que nous travaillons avec des données continues, la densité de probabilité $f_\theta(x_i)$ peut-être représentée par une densité de probabilité continue, par exemple une loi normale.

Ainsi

$$L(x|\theta) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

3.1.4 Pourquoi supposer une distribution normale ?

(i) [Théorème Central Limite] Pour de nombreux processus aléatoires, la somme (ou la moyenne) de nombreuses petites perturbations indépendantes tend vers une loi normale centrée réduite $\mathcal{N}(0, 1)$ en vertu du TCL. Cela justifie souvent l'hypothèse de normalité pour les erreurs de mesure ou les résidus dans les modèles scientifiques.
(ii) [Simplicité mathématique] La loi normale a des propriétés mathématiques qui facilitent l'estimation et l'inférence statistique. En particulier, la forme de la fonction de vraisemblance est bien connue et tractable.

3.1.5 Lien avec le code Python (voir annexe [4] partie optimisation

Pour le code, nous avons fait les hypothèses suivantes :

(i) Les résidus (différence entre les valeurs simulées à temps consécutifs) suivent une

distribution normale. Ces résidus sont supposés être normalement distribués avec une moyenne de 0 et variance σ^2. Ici σ^2 est arbitrairement fixé à 1. Cela simplifie les calculs et permet de maximiser la log-vraisemblance sans estimer σ^2 séparément.
(ii) La vraisemblance est donnée par :

$$\begin{aligned} L(x|\theta) &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{2\sigma^2}\right) \end{aligned}$$

Donc la log-vraisemblance est donnée par :

$$\log L(x|\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_i(\theta))^2 - \frac{n}{2}\log(2\pi\sigma^2)$$

où y_i sont les données observées et $\mu_i(\theta)$ les valeurs prédites par le modèle de diffusion pour les paramètres θ (dans le code, y_i = Ctime sim et $\mu_i(\theta)$ = Ctime exp).
En fixant σ^2 à 1,

$$\log L(x|\theta) = -\frac{1}{2}\sum(residuals^2) - \frac{n}{2}\log(2\pi)$$

En maximisant cette log-vraisemblance, nous estimons les paramètres θ qui rendent les observations les plus probables sous le modèle donné.
(iii) Dans le code, la fonction de vraisemblance est basée sur la différence entre les concentrations simulées à des temps consécutifs. Si les concentrations simulées sont proches des valeurs observées, cela signifie que les paramètres utilisés pour générer ces concentrations sont probablement proches des valeurs réelles des paramètres de diffusion.

3.1.6 Algorithme : Estimation de la Fonction de Diffusion

Algorithm 1 Estimation de la Fonction de Diffusion

Initialisation

Demander à l'utilisateur de choisir une fonction de diffusion.

Initialiser les paramètres en fonction du choix de l'utilisateur.

Définition des Paramètres

Définir le pas en espace h et le pas de temps dt.

Définir la condition initiale de concentration $c_0(z)$.

Construire les matrices A_h et B_h.

Définir la Fonction de Diffusion

Définir la fonction de diffusion $D(z)$ et sa dérivée en fonction des paramètres.

Implémenter une fonction de vraisemblance pour optimiser les paramètres de $D(z)$.

Optimisation

Définir les paramètres initiaux et les bornes pour l'optimisation.

Mesurer le temps de début et de fin de l'optimisation.

Afficher le temps d'exécution et sauvegarder les valeurs optimales de $D(z)$.

Afficher les résultats de la fonction de diffusion estimée.

Recalcul et affichage

Recalculer le vecteur inconnu C^j à l'instant t_j avec les paramètres optimaux.

Afficher les résultats recalculés avec les nouveaux paramètres.

Calcul de la somme des résidus.

Tracer une carte des résidus.

Calcul de l'erreur entre D expérimentale et D optimale.

3.2 Fonction `minimize` de `scipy.optimize`

Pour effectuer l'optimisation, nous avons utilisé a fonction `minimize` de la bibliothèque `scipy.optimize`. Elle cherche à minimiser une fonction objectif donnée

en ajustant les valeurs des paramètres. Dans notre cas, cette fonction est utilisée pour trouver les paramètres optimaux de la diffusion en ajustant un modèle à des données expérimentales en minimisant une fonction de log-vraisemblance.

3.2.1 Fonction objectif : `likelihood_function`

La fonction `minimize` tente de minimiser une fonction objectif, ici définie comme `likelihood_function`. Cette fonction calcule l'erreur entre les données simulées (calculées à partir des paramètres de diffusion) et les données expérimentales (`Ctime_exp`). La fonction de log-vraisemblance est utilisée pour mesurer cette erreur, en supposant que les résidus suivent une distribution normale. Plus cette erreur est faible, plus les paramètres du modèle ajustent bien les données.

3.2.2 Paramètres initiaux : `x0`

Les paramètres à estimer (comme D_0, D_∞, ou les valeurs segmentées dans le cas de la diffusion par morceaux) sont passés à la fonction `minimize` via l'argument `x0`. Ces paramètres représentent une estimation initiale du modèle. Le choix des paramètres initiaux dépend du type de diffusion sélectionné par l'utilisateur (`choix_D_opt`).

3.2.3 Méthode d'optimisation : `L-BFGS-B`

La méthode `L-BFGS-B` est une méthode de descente de gradient, adaptée pour les problèmes de grande taille, et permet d'imposer des bornes sur les paramètres. Dans notre cas, nous imposons des bornes sur les valeurs des paramètres avec l'argument `bounds`, par exemple :

- Pour une diffusion linéaire : `bounds = [(0.05, 10), (0.05, 10)]` contraint les valeurs des paramètres entre 0.05 et 10.
- Pour une diffusion exponentielle ou par morceaux, des bornes spécifiques sont également appliquées.

3.2.4 Boucle de simulation dans `likelihood_function`

Pour chaque jeu de paramètres, une simulation est effectuée pour calculer la diffusion à chaque pas de temps. Cette simulation calcule l'évolution de la concentration au fil du temps en fonction de la diffusion choisie, en résolvant numériquement les équations différentielles discrétisées. La simulation produit un tableau de concentrations simulées que l'on compare aux données expérimentales.

3.2.5 Résidus et fonction de log-vraisemblance

Une fois la simulation terminée pour un jeu de paramètres donné, la différence entre les concentrations simulées (`Ctime_sim`) et expérimentales (`Ctime_exp`) est calculée. Ces différences, appelées résidus, sont normalisées pour produire une fonction de log-vraisemblance :

$$\texttt{residuals} = \frac{\texttt{Ctime\_sim} - \texttt{Ctime\_exp}}{\texttt{Ctime\_exp}} + \epsilon$$

$$\texttt{log\_likelihood} = \sum \left(\texttt{residuals}^2\right)$$

La division des résidus par $\texttt{Ctime\_exp} + \epsilon$ (où ϵ est une petite valeur, ici 10^{-6}) est utilisée pour normaliser les résidus. Voici les raisons principales de cette approche :
1. **Normalisation des résidus** : En divisant par `Ctime_exp`, on exprime les résidus en termes relatifs plutôt qu'absolus. Cela permet de mieux gérer les situations où les valeurs de `Ctime_exp` peuvent varier considérablement. Les erreurs relatives donnent un meilleur aperçu de la qualité de l'ajustement lorsque les valeurs de `Ctime_exp` sont petites ou grandes. Sans cette normalisation, les grandes valeurs de `Ctime_exp` pourraient dominer la fonction de vraisemblance, tandis que les petites valeurs seraient sous-estimées.
2. **Prévention des divisions par zéro** : Le terme $\epsilon = 10^{-6}$ est ajouté pour éviter la division par zéro lorsque `Ctime_exp` est proche ou égale à zéro. Cela garantit que la fonction de vraisemblance reste stable, même si certaines valeurs de `Ctime_exp` sont très petites ou nulles.

La fonction `minimize` ajuste alors les paramètres de manière à minimiser cette somme des carrés des résidus, ce qui revient à maximiser la vraisemblance de l'ajustement du modèle aux données.

3.2.6 Résultats de l'optimisation

Une fois l'optimisation terminée, `minimize` renvoie un objet contenant les paramètres optimaux dans `result_simulation.x`. Ces paramètres sont ceux qui minimisent la fonction de log-vraisemblance et ajustent donc au mieux la diffusion aux données expérimentales. Le code affiche ces paramètres et les sauvegarde dans un fichier texte.

3.2.7 Résumé

- **Objectif** : Minimiser une fonction de log-vraisemblance qui mesure l'ajustement entre des données simulées et des données expérimentales.
- **Méthode** : `minimize` ajuste les paramètres en utilisant une méthode de descente de gradient tout en respectant des contraintes de bornes pour trouver les paramètres qui minimisent l'erreur entre simulation et données réelles.
- **Paramètres initiaux et contraintes** : Ils sont fournis selon la fonction de diffusion choisie.
- **Résultat** : Un jeu de paramètres optimaux est trouvé, enregistré, et utilisé pour prédire la diffusion.

3.3 Résultats numériques

3.3.1 La méthode fonctionne-t-elle ?

Dans cette partie, nous allons tester la méthode sur des solutions de concentration générées avec une diffusion D prédéfini. Par exemple, nous pouvons générer

une solution en choisissant une forme fonctionnelle pour D (linéaire, exponentielle, etc.), puis, à partir des données de concentration obtenues, estimer les paramètres de D pour voir dans quelle mesure nous nous rapprochons des valeurs initiales. Bien que nous ne nous attendions pas à retrouver exactement ces valeurs, une bonne approximation indiquerait que la méthode est prometteuse. Pour les profils linéaire et exponentiel, l'estimation nécessite à chaque fois la détermination de deux paramètres : D_0 et D_∞ pour le profil linéaire, et z_0 et D_∞ pour le profil exponentiel.

Nous commençons par générer une solution en utilisant un profil linéaire de D avec les paramètres $D_0 = 2$ et $D_\infty = 5$, tout en appliquant la condition initiale de référence. Cette approche permettra d'évaluer la précision de l'estimation des paramètres en comparant les résultats obtenus avec les valeurs initiales utilisées. Voici les résultats obtenus :

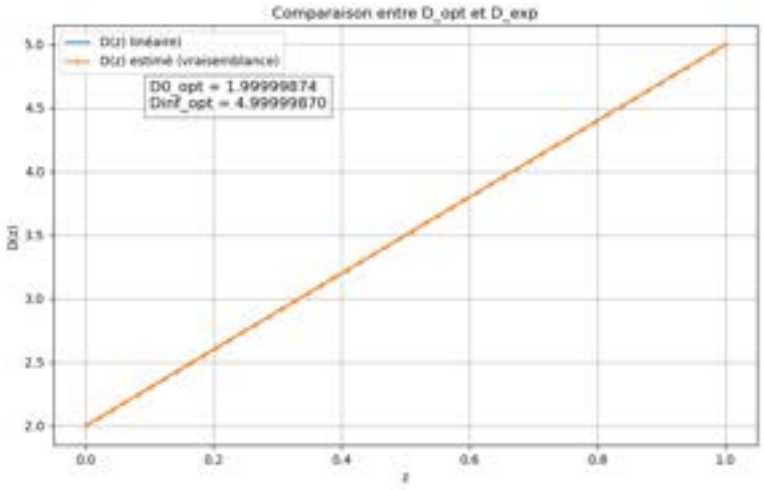
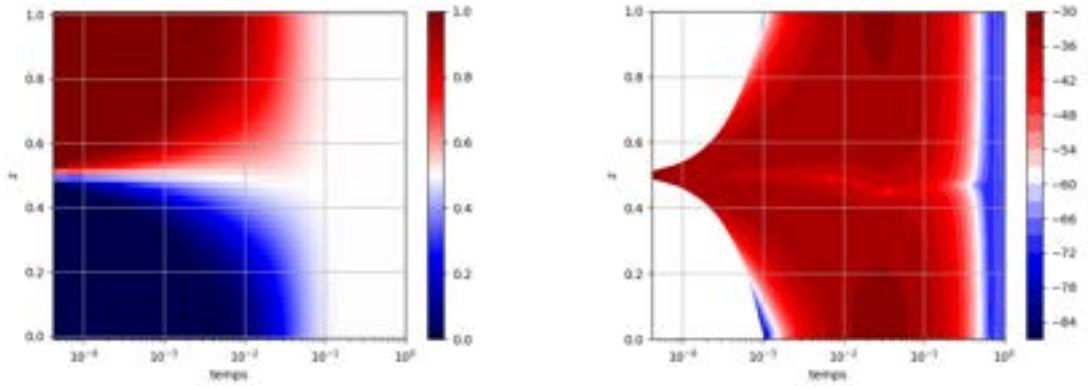


FIGURE 3.1 – Comparaison entre D expérimentale et D optimale



(a) (b)

FIGURE 3.2 – Résultat de l'optimisation : (a) Profil de concentration recalculée ; (b) Carte des résidus

Nous observons que la diffusion estimée se rapproche fortement de la diffusion expérimentale, avec des paramètres optimaux qui valent respectivement 1.99999874 et 4.9999987. L'erreur entre les deux fonctions est de l'ordre de 10^{-6}, ce qui est très satisfaisant. En utilisant ces nouveaux paramètres estimés pour D, nous avons tracé un contour de la concentration, et celui-ci se révèle semblable à celui présenté dans la Section 2.2.3, Figure 2.3 avec les paramètres initiaux. La carte des résidus révèle que les différences entre les valeurs de concentration expérimentales et celles estimées sont très faibles. Ainsi la méthode semble être efficace.

3.3.2 Résultats avec d'autres profils de c_0

Nous considérons des solutions de concentration obtenues à partir des conditions initiales décrites dans la partie précédente, illustrées dans la légende de la figure 2.5, et en utilisant le profil de diffusion exponentielle

$$D(z) = D_\infty \exp\left(-\frac{z}{z_0}\right)$$

avec $D_\infty = 10$ et $z_0 = 0.25$. Nous cherchons à savoir si la condition initiale affecte l'estimation des paramètres de la diffusion, afin de pouvoir considérer à l'avenir des données de concentrations plus pertinentes. La diffusion sera estimée par une diffusion optimale régulière par morceaux. Pour établir un lien avec les paramètres

estimés mentionnés dans la partie 3.1, nous faisons référence aux paramètres $\hat{D_{values}}$ (ce sont donc les D_i qu'on estime). Nous générons des valeurs aléatoires réparties sur un nombre de segments prédéfinis : par exemple, si nous générons deux valeurs, la diffusion aura la forme d'un segment. Il s'agit donc de faire varier le nombre de paramètres D_i pour voir aussi son impact sur l'estimation.

Les résultats obtenus avec la condition initiale définie par un sinus, ainsi qu'avec celle valant 0 ou 1 en certains points, suggèrent que l'augmentation du nombre de paramètres D_i améliore la précision de la méthode, comme nous pouvons le voir ci-dessous :

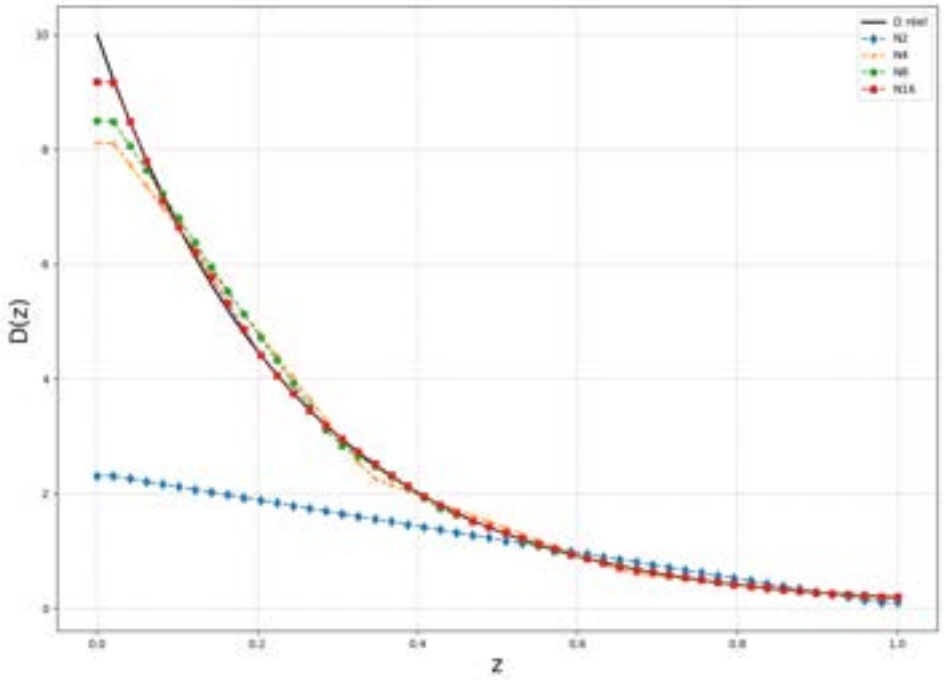
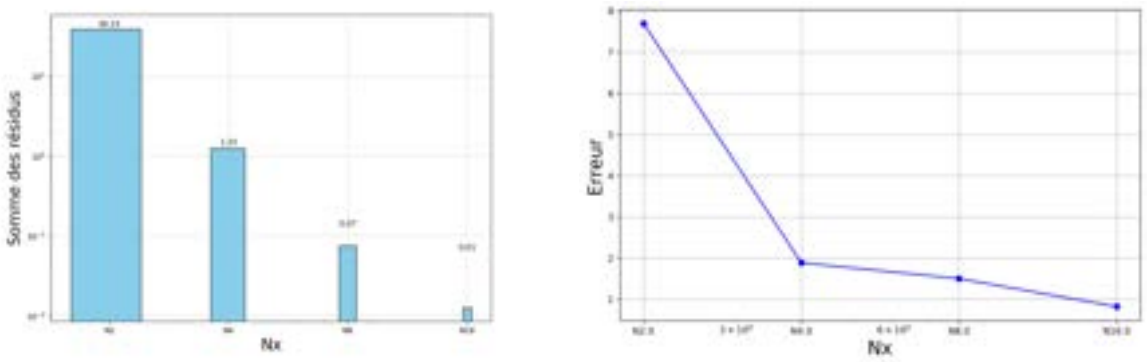


FIGURE 3.3 – Comparaison de D expérimentale avec différentes estimations en utilisant $c_0(z)$ avec une dizaine de couches de 0 et 1 (en légende, Nx signifie x points)



(a) somme des résidus (b) erreur entre Dsim et Dexp

FIGURE 3.4 – Suite des résultats de l'optimisation

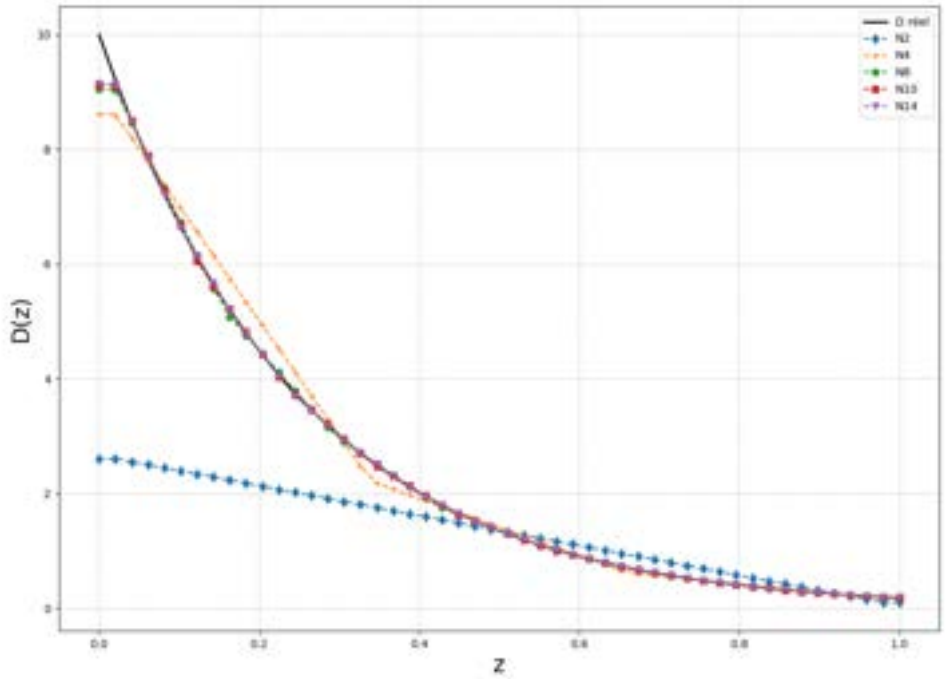
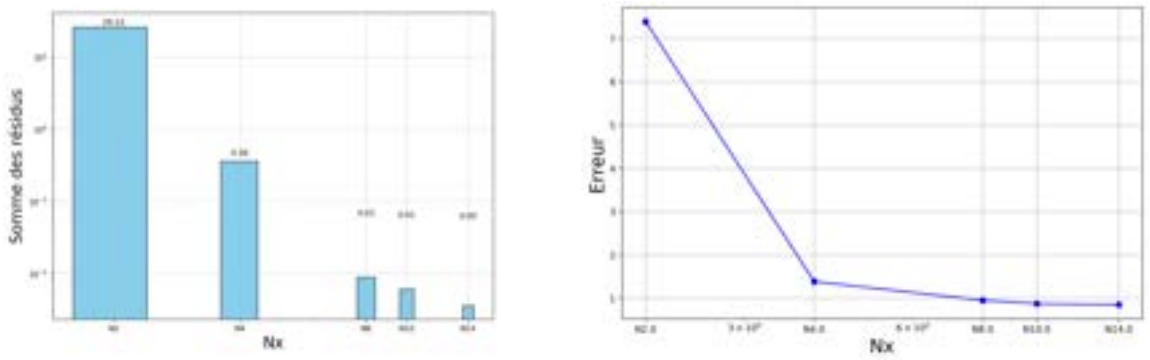


FIGURE 3.5 – Comparaison de D expérimentale avec différentes estimations en utilisant $c_0(z) = \frac{1}{2} + \frac{1}{2}\sin(10\pi z)$ (en légende, Nx signifie x points)



(a) somme des résidus (b) erreur entre Dsim et Dexp

FIGURE 3.6 – Suite des résultats de l'optimisation

Ces résultats montrent que l'approximation de D s'améliore avec l'augmentation du nombre de paramètres estimés. Cependant, ce n'est pas toujours le cas. En effet, si nous prenons en compte la condition initiale de référence, nous observons une augmentation significative de l'erreur lorsque nous passons à l'estimation de 32 paramètres, bien que la somme des résidus reste faible :

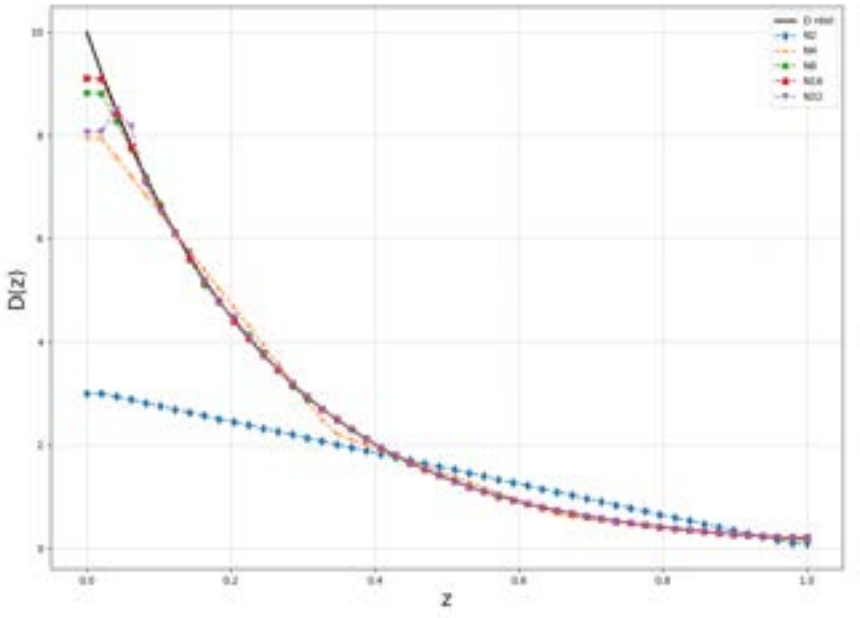
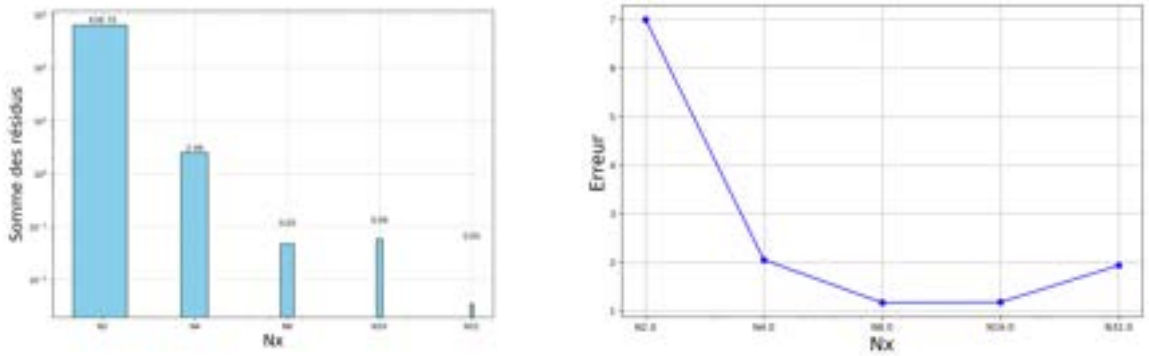


FIGURE 3.7 – Comparaison de D expérimentale avec différentes estimations en utilisant $c_0(z) = \begin{cases} 1 & \text{si} \quad z > \frac{1}{2} \\ 0 & \text{sinon} \end{cases}$ (en légende, Nx signifie x points)

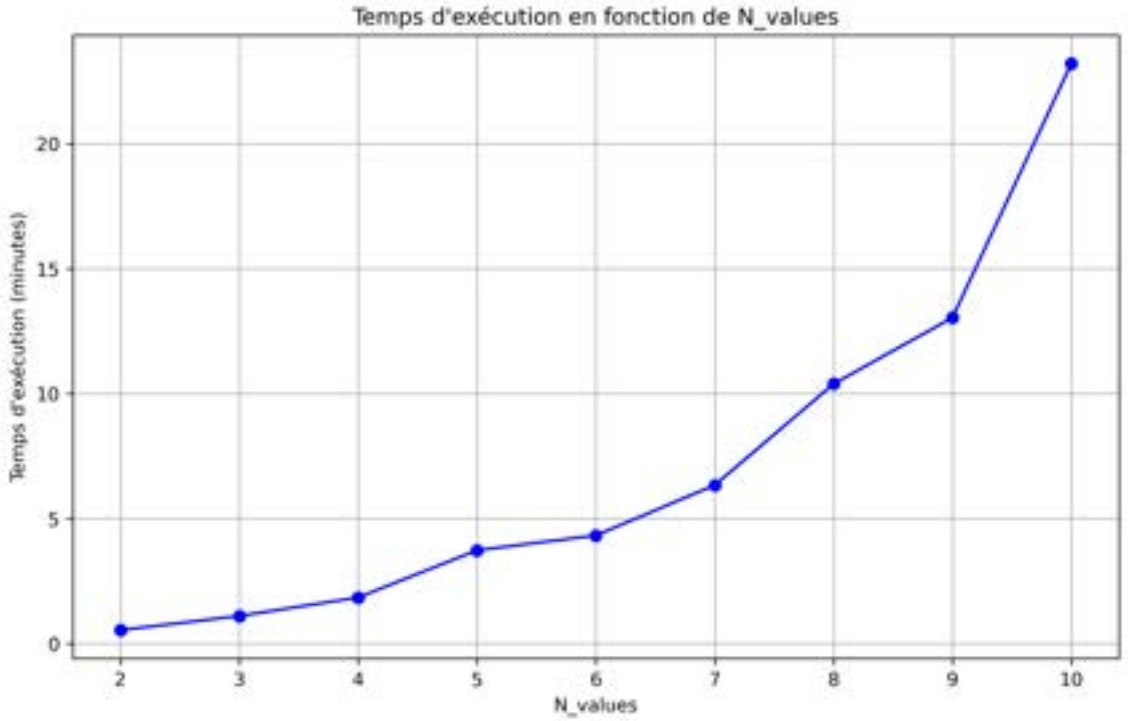


(a) somme des résidus (b) erreur entre Dsim et Dexp

FIGURE 3.8 – Suite des résultats de l'optimisation

Ces résultats montrent que le nombre de points influence clairement l'estimation des paramètres (mais pas toujours comme on pourrait s'y attendre), tout comme le choix de la condition initiale.

Nous traçons un graphe des temps d'exécution pour illustrer que l'augmentation du nombre de points entraîne une augmentation correspondante du temps d'exécution.



Étant donné que nous disposons de 50 points z_i en espace, estimer 50 valeurs D_i pour maximiser la précision et vérifier la diminution de l'erreur serait trop contraignant.

En effet, ce processus prend déjà environ 3 heures pour 32 valeurs.

3.3.3 Résultats avec des données de concentration plus pertinentes

La dernière tâche de ce stage consistait à estimer un profil de diffusion à partir de données de concentration obtenues via des simulations numériques. Cette analyse se base sur la configuration illustrée sur la figure suivante [2] :

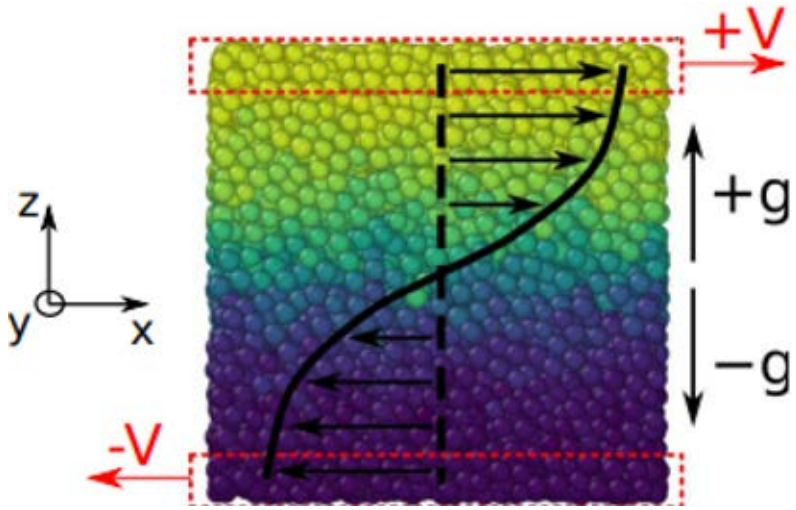
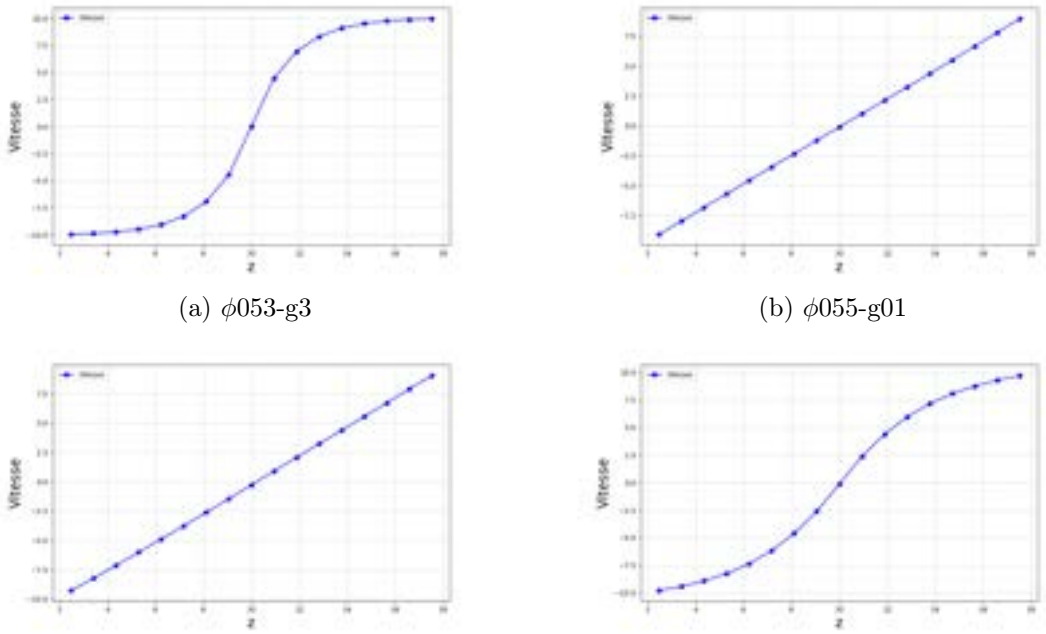


FIGURE 3.9 – Configuration d'écoulement concave étudiée avec son profil de vitesse. La concavité du profil dépend de la fraction volumique des solides (nommée ϕ pour la suite) et de l'accélération g. Le jaune plus clair correspond à la plus grande vitesse à droite, tandis que le bleu plus foncé indique une plus grande vitesse à gauche

.

Les données de concentration c_i pour $i = 1, ..., 20$ sont enregistrées dans un fichier texte. Cependant, nous ne disposons ni d'information sur la condition initiale, ni sur le comportement expérimental de la diffusion. Seules les données c_i sont connues. L'objectif est de trouver la meilleure estimation des paramètres de diffusion pour quatre expériences nommées respectivement ϕ053-g3, ϕ055-g01, ϕ057-g1 et ϕ057-g10, en approchant le profil de diffusion par une fonction continue par morceaux, et d'examiner si cette estimation dépend du profil de vitesse dans chaque cas. ϕ représente la fraction du volume de l'échantillon occupé par les billes en moyenne. En pratique il est lié au nombre des grains du système. Dans les titres des expériences nous avons les valeurs de ϕ et g. Pour le même g, en baissant ϕ nous pouvons passer d'un profil de vitesse linéaire à un non-linéaire. Voici les différents profils de vitesse des expériences :



(a) ϕ053-g3 (b) ϕ055-g01

(c) ϕ057-g1 (d) ϕ057-g10

FIGURE 3.10 – Profils de vitesse pour chacune des expériences

3.3.3.1 Première expérience ϕ053-g3

Pour cette première expérience, la meilleure approximation de D a été obtenue avec 18 valeurs D_i générées, avec une somme des résidus relevée à 0,07. Le graphique ci-dessous illustre que l'augmentation du nombre de valeurs D_i générées n'entraîne pas nécessairement une diminution de la somme des résidus. En effet, on observe une augmentation notable de cette somme lorsque l'on passe à 19 valeurs. Cette variation est également visible sur le profil estimé de D, qui s'éloigne sensiblement du profil obtenu avec 18 valeurs. Ce constat est en adéquation avec notre analyse de la partie 3.3.2. Le profil estimé de D suggère par ailleurs une forme gaussienne.

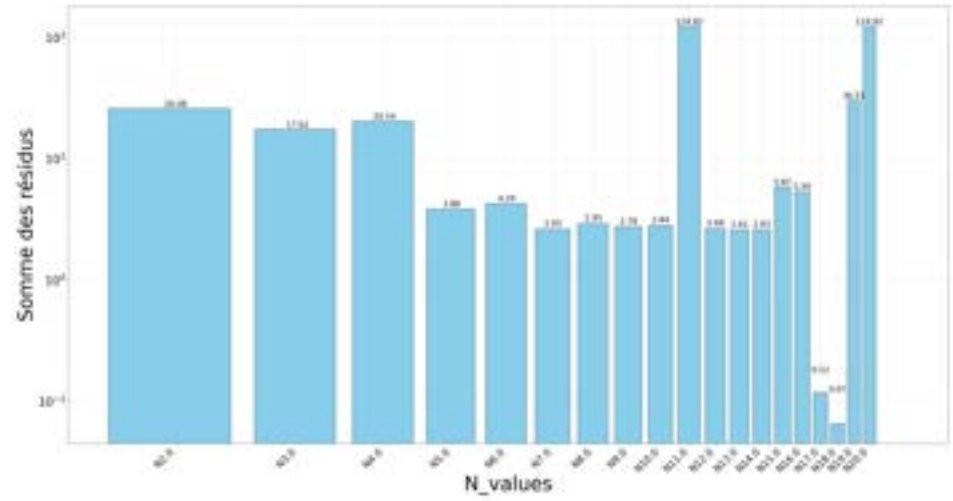


FIGURE 3.11 – Somme des résidus en fonction du nombre de D_i généré

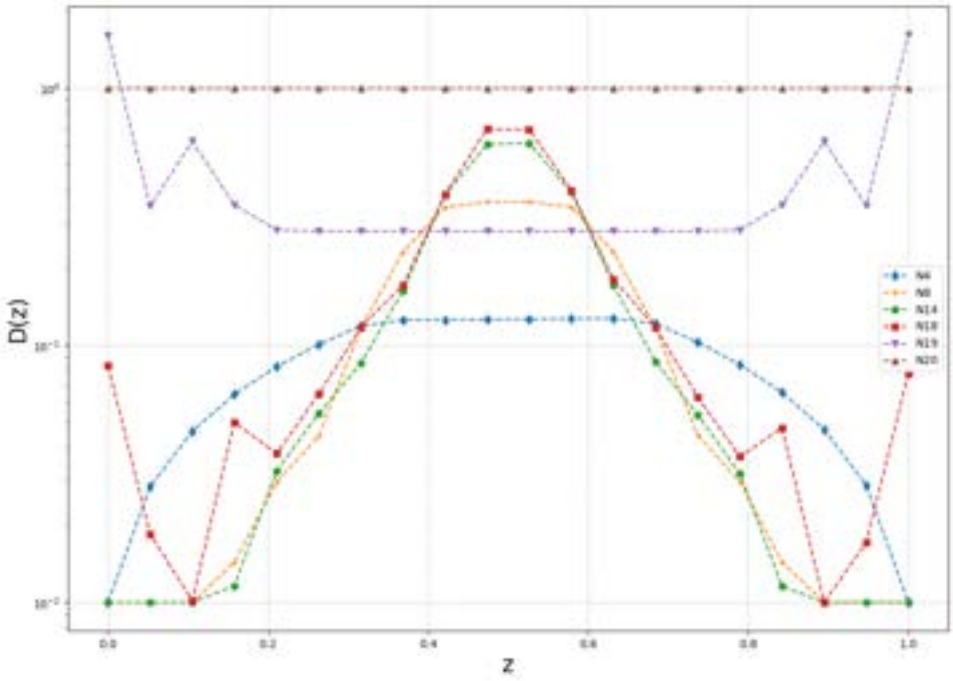
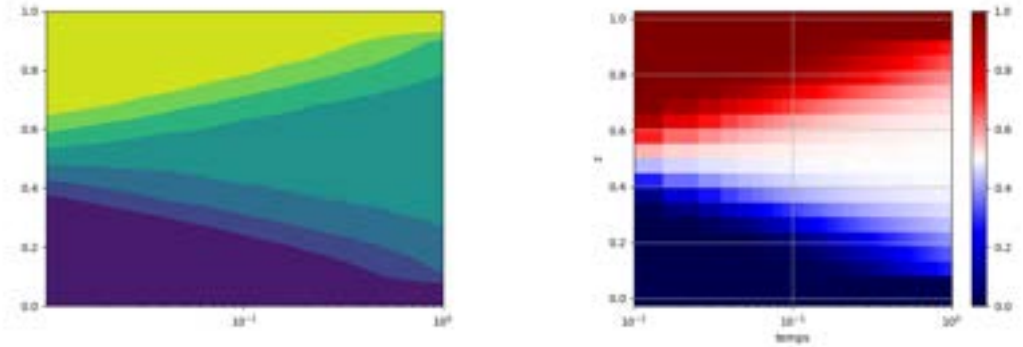


FIGURE 3.12 – Profils d'estimation de D

Nous avons également illustré différents profils de concentration, incluant celui correspondant à la concentration expérimentale (les c_i issus du fichier texte), ainsi que celui obtenu pour la concentration estimée avec la meilleure estimation de D, afin de permettre une comparaison.



(a) (b)

FIGURE 3.13 – Profils de concentration
(a) concentration expérimentale
(b) concentration estimée avec N18

Les deux profils sont similaires, ce qui confirme que le profil de D estimé avec 18 valeurs générées est celui qui se rapproche le plus du profil de diffusion expérimental.

3.3.3.2 Deuxième expérience ϕ055-g01

Ici, la meilleure estimation a été obtenue avec 17 valeurs D_i, pour une somme des résidus valant 0,07. Avec 19 valeurs, nous obtenons également une bonne approximation avec une somme des résidus valant 0,08. Contrairement à la première expérience, l'estimation avec 18 valeurs est moins concluante puisque la somme des résidus est relevée à 0,72.

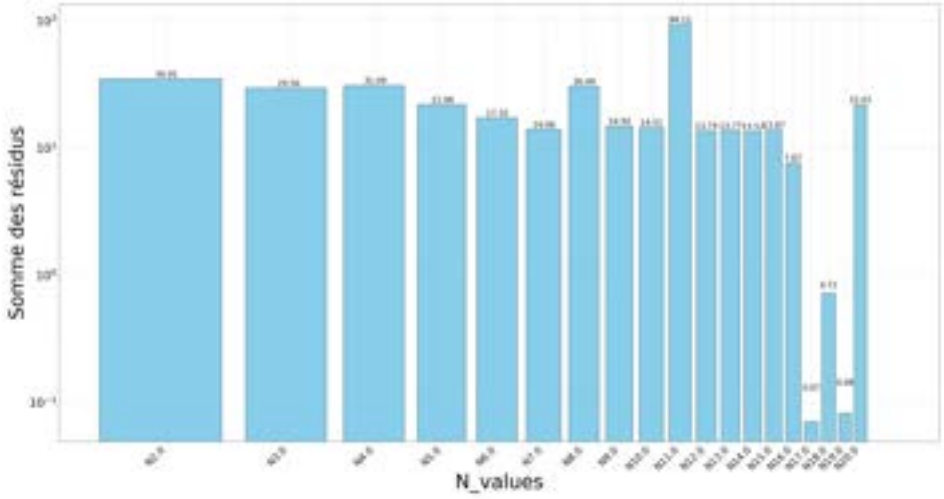


FIGURE 3.14 – Somme des résidus en fonction du nombre de D_i généré

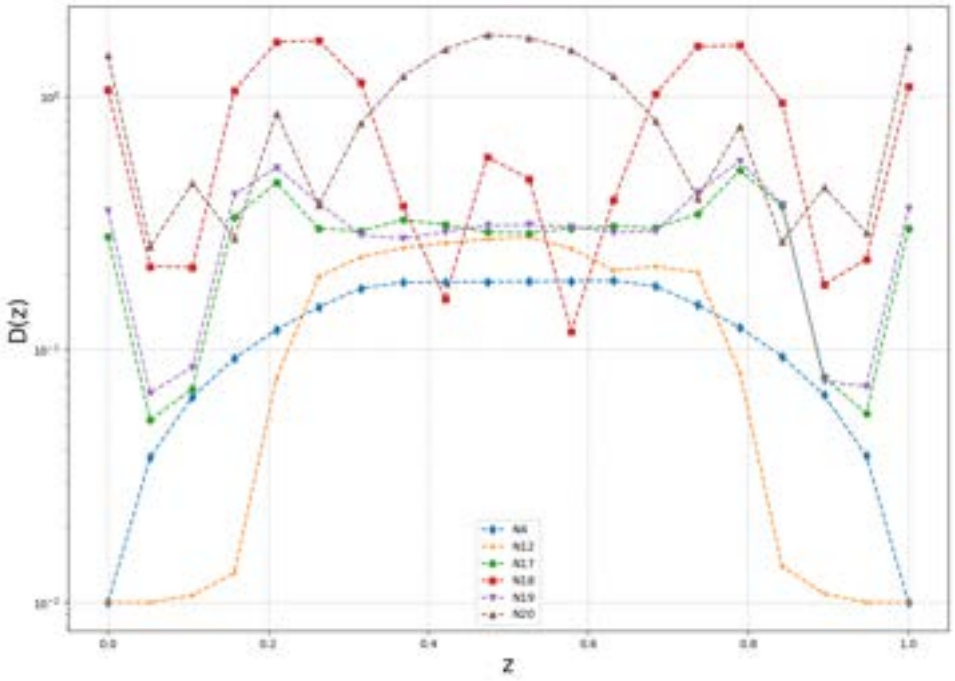
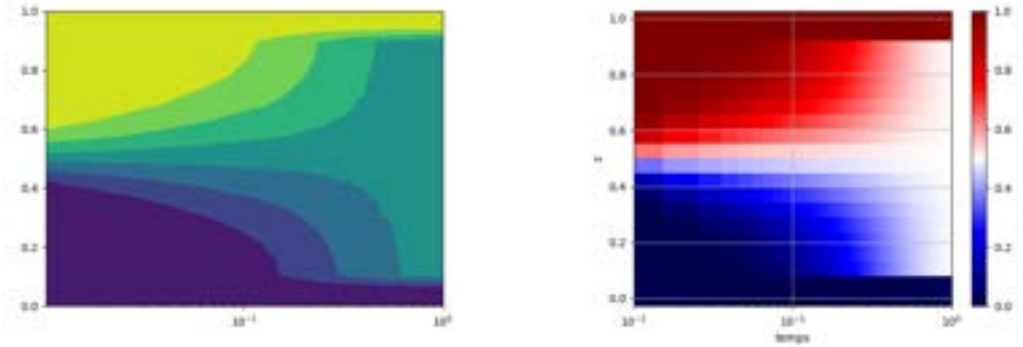


FIGURE 3.15 – Profils d'estimation de D

Nous observons que la partie centrale du profil N17 présente un comportement uniforme, ce qui est directement lié au profil de vitesse linéaire illustré dans la figure 3.10. Cela suggère que le profil de vitesse influence fortement la forme du profil de D.

Comme pour la première expérience, nous comparons les profils de concentration expérimentale et estimée afin de montrer leur similarité.



(a) (b)

FIGURE 3.16 – Profils de concentration
(a) concentration expérimentale
(b) concentration estimée avec N17

3.3.3.3 Troisième expérience ϕ057-g1

La meilleure estimation a été obtenue pour 16 valeurs de D_i, avec une somme des résidus égale à 0,09. Encore une fois, nous constatons qu'une augmentation du nombre de D_i générés n'améliore pas nécessairement les résultats, car la somme des résidus augmente au-delà de cette valeur.

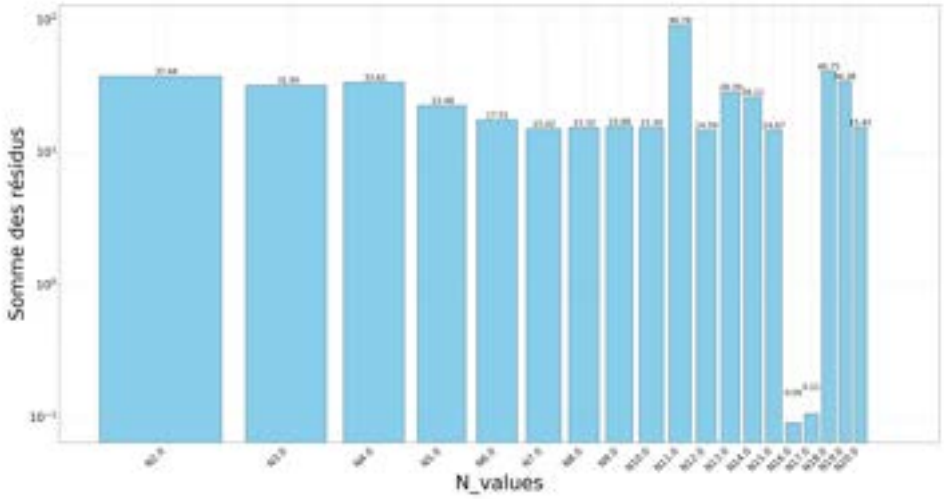


FIGURE 3.17 – Somme des résidus en fonction du nombre de D_i généré

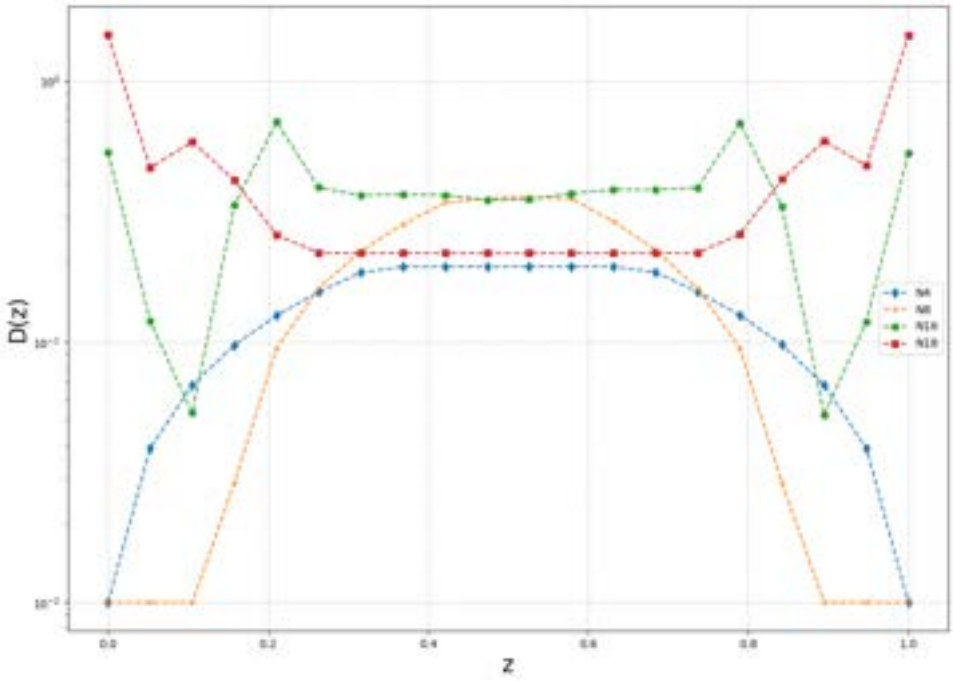
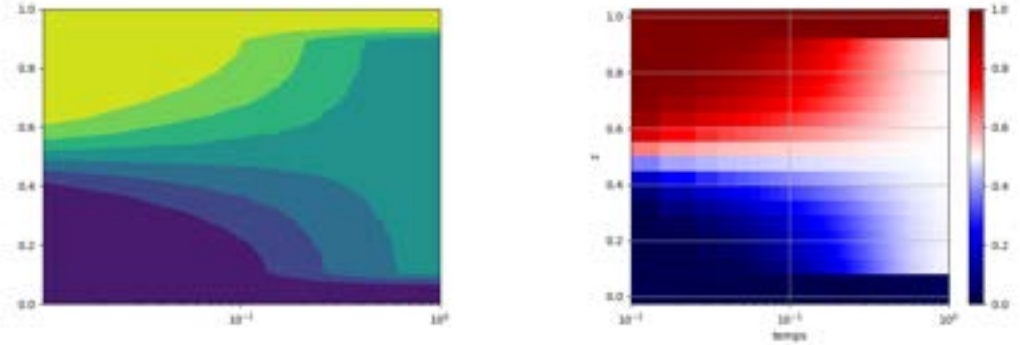


FIGURE 3.18 – Profils d'estimation de D

Comme pour la deuxième expérience, nous retrouvons ce comportement stationnaire au centre du profil N16, lié au profil de vitesse.

Les comparaisons des profils de concentration sont encore très semblables :



(a) (b)

FIGURE 3.19 – Profils de concentration
(a) concentration expérimentale
(b) concentration estimée avec N16

3.3.3.4 Quatrième expérience ϕ057-g10

Pour cette dernière expérience, la meilleure estimation a été obtenue avec 17 valeurs D_i, pour une somme des résidus de 0,07. Il s'agit donc d'un cas similaire à la deuxième expérience.

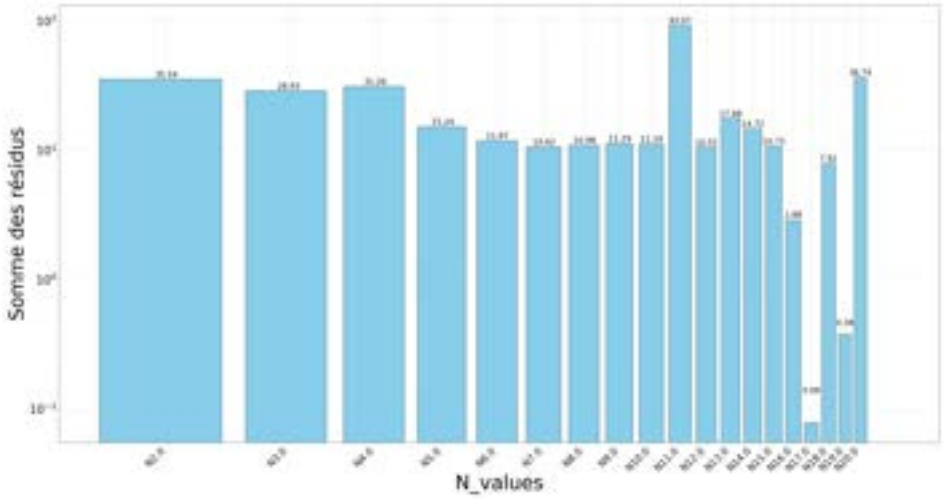


FIGURE 3.20 – Somme des résidus en fonction du nombre de D_i généré

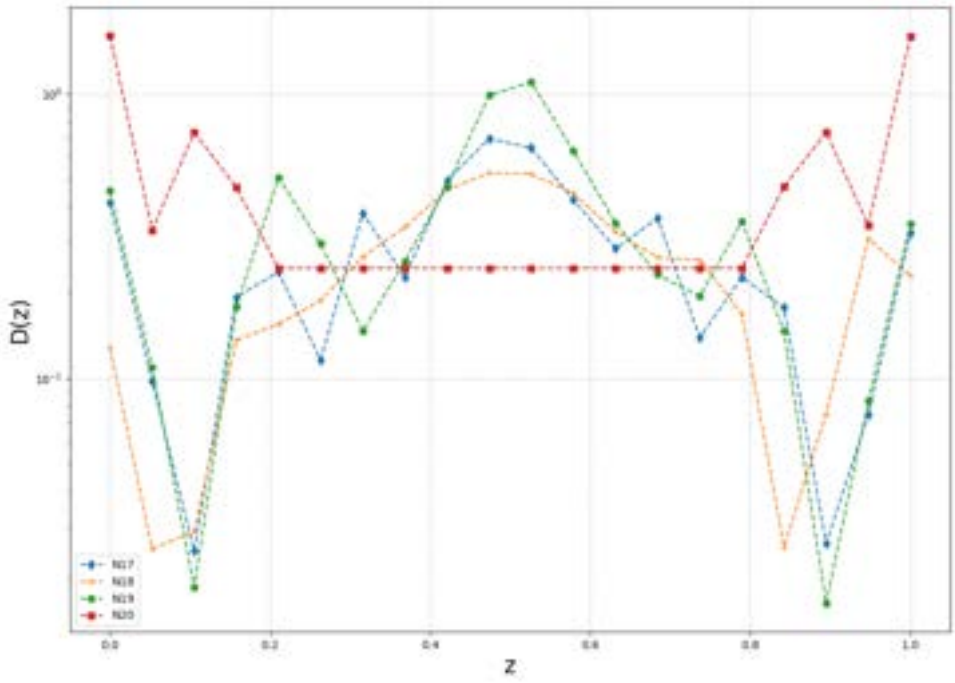
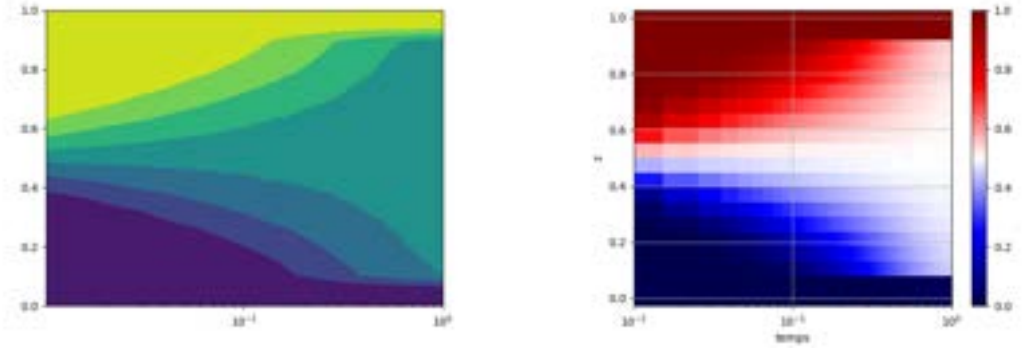


FIGURE 3.21 – Profils d'estimation de D

La courbure constatée sur le profil N17 est liée au profil de vitesse concave, comme observé lors de la première expérience.



(a) (b)

FIGURE 3.22 – Profils de concentration
(a) concentration expérimentale
(b) concentration estimée avec N17

En résumé, l'analyse des résultats montre que l'augmentation du nombre de valeurs D_i générées n'améliore pas systématiquement la précision de l'estimation. En effet, pour certaines expériences, nous avons observé que la somme des résidus ne diminue pas nécessairement avec l'augmentation du nombre de D_i. Par exemple, dans la première expérience, la meilleure estimation a été obtenue avec 18 valeurs D_i, et une augmentation à 19 valeurs a entraîné une augmentation significative de la somme des résidus. Et cela pour chacune des expériences. Ces observations suggèrent que, au-delà d'un certain nombre de valeurs générées, le processus d'estimation peut devenir moins efficace. Il est donc crucial de trouver un équilibre optimal entre le nombre de valeurs générées et la qualité de l'estimation. En pratique, un nombre trop élevé de valeurs peut compliquer le modèle sans nécessairement améliorer la précision, tandis qu'un nombre insuffisant peut conduire à une estimation approximative. La sélection du nombre optimal de D_i doit donc être soigneusement considérée en fonction des caractéristiques spécifiques de chaque problème.

En ce qui concerne le profil de vitesse, il a un impact significatif sur le profil de diffusion estimé. Lorsqu'un profil de vitesse linéaire est utilisé, comme observé dans la deuxième et troisième expérience, une grande partie du profil de diffusion

estimé montre un comportement uniforme. Cela indique que le profil de vitesse linéaire influence directement le profil de diffusion. De même, pour les profils de vitesse concaves, comme dans la première et quatrième expérience, une courbure est observée dans le profil de diffusion estimé liée à la concavité du profil de vitesse. Cette observation confirme que le profil de vitesse concave affecte directement le profil de D, entraînant une forme courbée dans le profil de diffusion.

En conclusion, les profils de vitesse jouent un rôle déterminant dans la forme des profils de diffusion estimés. Les variations dans le profil de vitesse se reflètent directement dans les estimations de D, soulignant l'importance d'intégrer le profil de vitesse dans les modèles de diffusion pour obtenir des estimations précises. La prise en compte du profil de vitesse est donc essentielle pour une estimation fidèle du profil de diffusion.

Chapitre 4

Conclusion

Ce stage a permis de se plonger dans l'étude de la diffusion dans des écoulements granulaires polydisperses, un domaine complexe et en pleine expansion. Grâce à des simulations numériques discrètes, nous avons pu approfondir notre compréhension des mécanismes de diffusion dans les milieux granulaires soumis à des contraintes de cisaillement. Les schémas numériques développés, qu'ils soient explicites ou implicites, ont montré des résultats prometteurs en termes de convergence et de stabilité, avec des temps de calcul raisonnables pour une précision acceptable.

De plus, l'approche par optimisation, basée sur le principe du maximum de vraisemblance, a permis d'estimer les paramètres du tenseur de diffusion avec une précision remarquable. Les tests réalisés sur différents profils de conditions initiales et de fonctions de diffusion ont mis en lumière l'impact significatif des choix initiaux sur le comportement de la solution finale. En particulier, nous avons observé que la nature de la diffusion (linéaire ou exponentielle) et le profil initial influencent fortement la dynamique de mélange des particules.

Les résultats obtenus confirment la validité des méthodes employées, tout en ouvrant la voie à des investigations plus poussées, notamment dans des contextes plus complexes avec des données expérimentales plus variées. Il serait pertinent d'élargir cette étude à des cas tridimensionnels ou de considérer d'autres types de schémas numériques pour améliorer encore la précision et la rapidité des calculs.

En conclusion, ce travail a permis non seulement de valider certaines méthodes

numériques pour l'estimation de la diffusion dans des écoulements granulaires, mais aussi de poser des bases solides pour des études futures dans ce domaine en constante évolution.

Annexes

Code Python : Différences finies pour l'équation de diffusion non stationnaire

```
##### differences finies pour l'equation de diffusion non
    stationnaire avec diffusion non constant sur [0,1]x[0,T]
# D continue et 0 < D_0 <= D(z) <= D_inf < +inf
# conditions de Neumann imposees aux bords
# conditions initiales c_0 connue

import numpy as np
import math
from matplotlib.pylab import *
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy.random as rnd
from numpy.linalg import *
%matplotlib inline
from scipy.optimize import minimize

# Definir les parametres pour la compilation
choix = input("Veuillez choisir une methode (1 pour explicite, 2
    pour implicite) : ")

# Verifier le choix de l'utilisateur
while choix not in ['1', '2']:
```

```
21      print("Choix non valide. Veuillez entrer 1 pour explicite ou 2
            pour implicite.")
22      choix = input("Veuillez choisir une methode (1 pour explicite,
            2 pour implicite) : ")
23
24  # Convertir le choix en methode
25  methode = "explicite" if choix == '1' else "implicite"
26
27  # Afficher la methode choisie par l'utilisateur
28  print("La methode choisie est :", methode)
29
30  # Demande a l'utilisateur d'entrer le nombre de points en espace du
        maillage
31  N = int(input("Veuillez entrer le nombre de points en espace du
        maillage : "))
32
33  # Construire le maillage
34  h = 1/(N+1)
35  X = linspace(0, 1, N+2)
36
37  # Demander a l'utilisateur de choisir une condition initiale
38  choix_c0 = input("Veuillez choisir une condition initiale (1 pour
        c0 constante, 2 pour lineaire, 3 pour z^2, 4 pour sqrt(z), 5
        avec une dizaine de couches, 6 pour un sinus) : ")
39
40  # Definition de la condition initiale
41  def c0(z) :
42      if choix_c0 == '1' :
43          return where(z > 0.5, 1, 0)
44      elif choix_c0 == '2' :
45          return z
46      elif choix_c0 == '3' :
47          return z**2
48      elif choix_c0 == '4' :
49          return np.sqrt(z)
```

```
50      elif choix_c0 == '5' :
51          num_couches = 10
52          long = 1 / num_couches
53          return np.array([1 if int(pos / long) % 2 == 1 else 0 for
                pos in z])
54      elif choix_c0 == '6':
55          return 0.5 + 0.5*sin(10*pi*z)
56
57  # Demander a l'utilisateur de choisir la fonction de diffusion
58  choix_D = input("Veuillez choisir une fonction de diffusion (1 pour
        D(z) lineaire, 2 pour D(z) exponentielle, 3 pour D(z) reguliere
        par morceaux) : ")
59
60  # Definir D selon le choix
61  if choix_D == '1':
62      D_0 = float(input("Veuillez entrer la valeur de D_0 : "))
63      D_inf = float(input("Veuillez entrer la valeur de D_inf : "))
64      def D(z):
65          return D_0 + (D_inf - D_0) * z
66      def derivate_D(z):
67          return (D_inf - D_0) * np.ones(len(z))
68
69  elif choix_D == '2':
70      D_inf = float(input("Veuillez entrer la valeur de D_inf : "))
71      z_0 = float(input("Veuillez entrer la valeur de z_0 : "))
72      def D(z):
73          return D_inf * np.exp(-z / z_0)
74      def derivate_D(z):
75          return -(D_inf / z_0) * np.exp(-z / z_0)
76
77  # Definir le pas de temps selon la methode
78  if methode == "explicite":
79      dt = h**2 / (2 * D_inf)
80  else :
81      dt = float(input("Rentrez le pas de temps : "))
```

```
82
83  T = float(input("Rentrez le temps final T : "))
84  M = int((T/dt) - 1)
85  print('M = ', M)
86
87  Ti = np.linspace(0,T,M+1)
88
89  # Construction de la matrice A_h
90  u = ones(N+2)
91  x = 2*u
92  v = ones(N+1)
93  y = -1*v
94  Ah = (1/h**2)*(diag(x) + diag(y,-1) + diag(y,1))
95  Ah[0,0] /=2
96  Ah[-1,-1]/=2
97
98  # Construction de la matrice B_h
99  a = ones(N+1)
100 b = -1*a
101 Bh = (1/(2*h))*(diag(a,1) + diag(b,-1))
102 Bh[0,1]=0.
103 Bh[-1,-2]=0.
104
105 # Initialisation des C^j selon le schema choisi
106 C = c0(X)
107 Ch = C[0 : N+2]
108 Cjh = zeros(N+2)
109 Ctime=zeros((M+1,N+2))
110 Ctime[0,:]=c0(X)
111
112 # Boucle pour chaque pas de temps
113 for j in range(1, M+1):
114     if methode == "explicite" :
115         Cjh = Ch + dt*(derivate_D(X)*dot(Bh,Ch)) - dt*(D(X)*dot(Ah,
              Ch))
```

```
116     else :
117         I = eye(N+2)
118         P = I - dt*(derivate_D(X)*Bh.T).T + dt*(D(X)*Ah.T).T
119         Cjh = solve(P, Ch)
120     Cjh[0]=Cjh[1]
121     Cjh[-1]=Cjh[-2]
122     Ch = Cjh
123     Ctime[j,:]=Ch
124 
125 # Sauvegarde des donnees dans un fichier texte
126 np.savetxt('Ctime_values_c06.txt',Ctime)
127 print("Les valeurs de Ctime sont sauvegardees dans le fichier '
        Ctime_values.txt'.")
128 
129 # Affichage des resultats
130 plt.contourf(Ti,X,Ctime.T,np.linspace(0,1,20),cmap ='seismic')
131 plt.xscale('log'),plt.xlim((dt,T))
132 plt.xlabel('temps')
133 plt.ylabel('z')
134 plt.grid(True)
135 plt.colorbar()
136 plt.show()
137 
138 Tg, Xg = np.meshgrid(Ti,X)
139 plt.pcolormesh(Tg,Xg,Ctime.T,cmap = 'seismic')
140 plt.xscale('log'),plt.xlim((dt,T))
141 plt.xlabel('temps')
142 plt.ylabel('z')
143 plt.grid(True)
144 plt.colorbar()
145 plt.savefig('concentration_c03_Dexp.png')
146 plt.show()
147 
148 # Profil de Ch
149 plt.figure(figsize=(10, 6))
```

```
plt.plot(Ch)
plt.xlabel('Index')
plt.ylabel('Ch')
plt.title('Profil de Ch')
plt.ticklabel_format(style='plain', axis='y')
plt.ylim(0.0, 1.0)
plt.savefig('concentration_dos.png')
plt.show()
```

Code Python : optimisation

```
##### methode d'optimisation pour estimer D

import numpy as np
import math
from matplotlib.pylab import *
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import numpy.random as rnd
from numpy.linalg import *
%matplotlib inline
from scipy.optimize import minimize
from scipy.optimize import least_squares
from scipy.optimize import differential_evolution
import time

# Lire le fichier texte
nom_fichier = 'Ctime_values_c06.txt'
data = np.loadtxt(nom_fichier)

# Dimensions des donnees
M, N_plus_2 = data.shape
print("Dimensions des donnees (MxN):", M, N_plus_2)

```

```
24 # Extraire les donnees de concentration
25 Ctime_exp = data
26 print(data)
27
28 # Definir les valeurs de z (espace) et de t (temps)
29 X = np.linspace(0, 1, N_plus_2)
30 t = np.linspace(0, 1, M)
31 Tg, Xg = np.meshgrid(t,X)
32 plt.contourf(Tg,Xg,Ctime_exp.T)
33 plt.xscale('log'),plt.xlim((t[1],1))
34
35 # Demander e l'utilisateur de choisir la fonction de diffusion
36 choix_D_opt = input("Veuillez choisir une fonction de diffusion (1
      pour D(z) lineaire, 2 pour D(z) exponentielle, 3 pour D(z)
      reguliere par morceaux) : ")
37
38 while choix_D_opt not in ['1', '2', '3']:
39     print("Choix non valide. Veuillez entrer 1 pour D(z) lineaire,
          2 pour D(z) exponentielle ou 3 pour reguliere par morceaux."
          )
40     choix_D_opt  = input("Veuillez choisir une fonction de
          diffusion (1 pour D(z) lineaire, 2 pour D(z) exponentielle,
          3 pour D(z) reguliere par morceaux) : ")
41
42 # Pas en espace
43 h = 1/(N_plus_2 - 1)
44 dt = 1/M
45
46 # Definition de la condition initiale
47 def c0(z) :
48     return Ctime_exp[0,:]
49
50 # Construction de la matrice A_h
51 u = ones(N_plus_2)
52 x = 2*u
```

```
v = ones(N_plus_2 - 1)
y = -1*v

Ah = (1/h**2)*(diag(x) + diag(y,-1) + diag(y,1))
Ah[0,0] /=2
Ah[-1,-1]/=2

# Construction de la matrice B_h
a = ones(N_plus_2 - 1)
b = -1*a

Bh = (1/(2*h))*(diag(a,1) + diag(b,-1))
Bh[0,1]=0.
Bh[-1,-2]=0.

# Definir la fonction de diffusion et sa derivee
def D(params, z):
    if choix_D_opt == '1':
        D_0, D_inf = params
        return D_0 + (D_inf - D_0) * z
    elif choix_D_opt == '2':
        z_0, D_inf = params
        return D_inf * np.exp(-z / z_0)
    elif choix_D_opt == '3':
        num_segments = len(params)
        dx = 1. / 50.
        xp = np.linspace(0 + dx, 1 - dx, num_segments)
        return np.interp(z, xp, params)

def derivate_D(params, z):
    if choix_D_opt == '1':
        D_0, D_inf = params
        return (D_inf - D_0) * np.ones(len(z))
    elif choix_D_opt == '2':
        z_0, D_inf = params
```

```
88          return -(D_inf / z_0) * np.exp(-z / z_0)
89      elif choix_D_opt == '3':
90          D_piecewise = D(params, z)
91          return np.gradient(D_piecewise) / np.gradient(z)
92
93  # Fonction de vraisemblance
94  def likelihood_function(params):
95      D_func = D(params, X)
96      derivate_D_func = derivate_D(params, X)
97
98      D_inf = max(D_func)
99
100     # Initialisation de la concentration
101     C = c0(X)
102     Ch = C[0: N_plus_2]
103     Ctime_sim = np.zeros((M, N_plus_2))
104     Ctime_sim[0, :] = c0(X)
105
106     dt_stab = h**2 / (2 * D_inf)
107     Q = int(ceil(dt /dt_stab))
108     dt_loc = dt/Q
109
110     for j in range(1, M):
111         for j2 in range(Q):
112             Cjh = Ch + dt_loc * (derivate_D_func * np.dot(Bh, Ch))
                    - dt_loc * (D_func * np.dot(Ah, Ch))
113             Cjh[0] = Cjh[1]
114             Cjh[-1] = Cjh[-2]
115             Ch = Cjh
116         Ctime_sim[j, :] = Ch
117
118     residuals = (Ctime_sim - Ctime_exp)/(Ctime_exp+1e-6)
119     log_likelihood = np.sum((residuals**2))
120     return log_likelihood
121
```

```
122 # Initialisation des parametres selon le choix
123 if choix_D_opt == '1':
124     initial_params = [.05, .05]
125     bounds = [(0.05, 10), (0.05, 10)]
126 elif choix_D_opt == '2':
127     initial_params = [1, 1]
128     bounds = [(0.1, 10), (0.01, 10)]
129 elif choix_D_opt == '3':
130     N_values = int(input("Saisissez un nombre : "))
131     initial_params = [1 for i in range(N_values)]
132     bounds = [(0.1,20) for i in range(N_values)]
133
134 # Mesurer le temps de debut
135 start_time = time.time()
136
137 # Effectuer l'optimisation
138 result_simulation = minimize(likelihood_function, x0=initial_params
        , bounds=bounds, method='L-BFGS-B')
139
140 # Mesurer le temps de fin
141 end_time = time.time()
142
143 # Calculer la duree d'execution
144 execution_time = end_time - start_time
145
146 # Afficher le temps d'execution
147 print(f"Temps d'execution : {execution_time:.2f} secondes")
148
149 # Afficher les parametres optimaux
150 print("Parametres optimaux :")
151 print(result_simulation.x)
152
153 D_opt_sim = D(result_simulation.x,X)
154 derivate_D_opt_sim = derivate_D(result_simulation.x,X)
155
```

```
156 # Sauvegarder les donnees dans un fichier texte
157 np.savetxt('D_sim_N16_c06.txt',D_opt_sim)
158 print("Les valeurs de D_opt_sim ont ete sauvegardees dans le
        fichier.")
159
160 # Calcul de l'erreur
161 if choix_D_opt == '1':
162     err = max(abs(D_opt_sim - (2.0+(5.0-2.0)*X)))
163     print('erreur = :', err)
164 elif choix_D_opt == '2' or choix_D_opt == '3':
165     err = max(abs(D_opt_sim - 10 * np.exp(-X / 0.25)))
166     print('erreur = :', err)
167
168 # Recalcul des resultats avec les parametres optimaux
169 C = c0(X)
170 Ch = C[0: N_plus_2]
171 Ctime_sim = np.zeros((M, N_plus_2))
172 Ctime_sim[0, :] = c0(X)
173
174 dt_stab = h**2 / (2 * max(D_opt_sim))
175 Q = int(ceil(dt / dt_stab))
176 dt_loc = dt / Q
177
178 for j in range(1, M):
179     for j2 in range(Q):
180         Cjh = Ch + dt_loc * (derivate_D_opt_sim * np.dot(Bh, Ch)) -
                dt_loc * (D_opt_sim * np.dot(Ah, Ch))
181         Cjh[0] = Cjh[1]
182         Cjh[-1] = Cjh[-2]
183         Ch = Cjh
184     Ctime_sim[j, :] = Ch
185
186 # Afficher le vecteur inconnu C^j
187 print("Vecteur inconnu C^j a l'instant t_j :", Ch)
188
```

```
189 # Calcul de la somme des residus
190 somme=np.sum((Ctime_sim.T-Ctime_exp.T)**2.)
191 print('somme_residus = :', somme)
192
193 # Tracer les resultats recalcules
194 plt.contourf(Ti, X, Ctime_sim.T, np.linspace(0, 1, 20), cmap='
        seismic')
195 plt.xscale('log')
196 plt.xlim((dt, T))
197 plt.xlabel('temps')
198 plt.ylabel('z')
199 plt.grid(True)
200 plt.colorbar()
201 plt.show()
202
203 # Carte des residus
204 plt.contourf(Ti, X, np.log((Ctime_sim.T-Ctime_exp.T)**2.), 20, cmap
        ='seismic')
205 plt.xscale('log')
206 plt.xlim((dt, T))
207 plt.xlabel('temps')
208 plt.ylabel('z')
209 plt.grid(True)
210 plt.colorbar()
211 plt.savefig('carte_residus.png')
212 plt.show()
```

Bibliographie

[1] ARTONI R., SOLIGO A., PAUL J. M., RICHARD P. (2018), Shear localization and wall friction in confined dense granular flows, *Journal of Fluid Mechanics*, vol. 849, p.395-418.

[2] ARTONI R., RICHARD P., LARCHER M., T.JENKINS J (2022), Self-diffusion in inhomogeneous granular shearing flow, *Physical Review E 106, L032901*

[3] ALEXANDER M. FRY, PAUL B. UMBANHOWAR, JULIO M. OTINO, RICHARD M. LUEPTOW (2019), Diffusion, Mixing, and Segregation in Confined Granular Flows, *AIChE Journal*, vol. 65 no.33, p.875-881