

Transport d'une espèce chimique dans un fluide

Enseignant responsable
Bernard GLEYSE

Étudiants :
Marie-Ève CLAVEL
Eliass EL ALAMI
Jospeh LEFEVRE

Xavier LEMAHIEU
Nathan ROUXELIN
Mathilde TAVERNIER

Date de remise du rapport : 15/06/2015

Référence du projet : STPI¹/P6/2015 – 7

Intitulé du projet : *Transport d'une espèce chimique dans un fluide*

Type de projet : *Modélisation*

Objectifs du projet :

L'objectif de ce projet est d'étudier une solution simple pour modéliser le transport d'une espèce chimique dans un fluide.

Voici les différents objectifs de notre projet :

- Établir le modèle physique qui consiste en une équation : $\frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) = f(x, t)$*
- Étudier les solutions théoriques de cette équation*
- Étudier des méthodes numériques pour approximer ces solutions*
- Implémenter ces méthodes en Pascal*
- Déterminer des applications de ce modèle.*

Table des matières

Introduction	4
I Méthodologie, organisation du travail	5
II Travail réalisé et résultats	7
1 Justification physique de l'équation de transport	8
2 Résolution analytique de l'équation de transport	10
2.1 Équation homogène	10
2.2 Cas d'une équation avec un terme source	11
2.3 Application	12
3 Résolution numérique de l'équation de transport	13
3.1 Méthode des différences finies	13
3.2 Schéma explicite	14
3.3 Schéma implicite	14
3.4 Comparaison	15
4 Applications	19
4.1 Que représente physiquement l'équation de transport ?	19
4.2 Application de nos programmes à une situation physique concrète	19
Conclusion et perspectives	21
Bibliographie	22
A Programmes réalisés	23
A.1 Exemple 1 sans terme source	23
A.2 Exemple 1 avec terme source	25
A.3 Script Python pour calculer les écarts	28
A.4 Script Python pour tracer la courbe à un temps donné	28
A.5 Script Python pour l'animation	30

Introduction

Le projet P6 est le troisième projet de notre cursus à l'INSA de Rouen et le premier dans le domaine de la physique. C'est aussi la première fois que nous travaillons dans un grand groupe imposé.

Ce projet étant un projet de modélisation, il implique donc de la physique mais aussi des mathématiques et de l'informatique. Cela permet donc à chaque membre du groupe de travailler sur la partie du projet qui l'intéresse.

L'intitulé exact de notre projet est : **Transport d'une espèce chimique dans un fluide.**

Le but est de déterminer l'évolution de la concentration d'une espèce chimique dans un fluide en mouvement en fonction de la position et du temps. Ce modèle s'applique typiquement à une usine qui rejette un polluant dans l'air.

Nous avons établi le modèle physique en dimension 1. C'est une équation aux dérivées partielles. Nous avons ensuite étudié des méthodes numériques pour la résoudre.

Première partie

Méthodologie, organisation du travail

Pour pouvoir mener notre projet à bien, nous avons commencé par nous familiariser avec le sujet. Pour cela, nous avons constitué une bibliographie à partir de ressources trouvées sur Internet et à la bibliothèque.

Réaliser cette bibliographie nous a permis de bien cerner ce que l'on attendait de nous. Nous avons également divisé notre projet en trois grandes parties : une partie théorique, une partie informatique et une partie applications.

Nous avons ensuite réparti les tâches entre les différents membres du groupe. Nous avons choisi de travailler par petits groupes : chaque binôme est responsable d'une des parties du projet.

Tâche	Membre(s) du groupe	Partie
Réalisation de la bibliographie	Tout le groupe	
Justification de l'équation de transport	Nathan R., Mathilde T.	Théorique
Résolution de l'équation de transport	Nathan R., Mathilde T.	
Méthode explicite	Joseph L.	Informatique
Méthode implicite	Marie-Ève C.	
Applications	Eliass E., Xavier L., Marie-Ève C.	Applications
Rapport	Nathan R., Mathilde T., Joseph L.	
Préparation de la soutenance	Tout le groupe	

Cette organisation ne nous a toutefois pas empêchés de sortir des groupes pour nous entraider lorsqu'il y en avait besoin.

Deuxième partie

Travail réalisé et résultats

Chapitre 1

Justification physique de l'équation de transport

On se donne un fluide en mouvement le long de l'axe des réels. On note $u(x, t)$ son profil densité et $v(x, t)$ son profil vitesse.

On suit un domaine $]a(t); b(t)[$, d'extrémités régulières, dans son mouvement. Si l'on suppose qu'il n'y a pas de création de masse dans l'intervalle, on a

$$M(t) = \int_{a(t)}^{b(t)} u(x, t) dx = \text{constante}$$

Ainsi, quelque soit $t \in \mathbb{R}^+$, on a $M'(t) = 0$. Or, on a aussi :

$$M'(t) = \int_{a(t)}^{b(t)} \frac{\partial u}{\partial t}(x, t) dx + b'(t)u(b(t), t) - a'(t)u(a(t), t)$$

Le domaine est en mouvement avec le fluide, on peut donc relier $a'(t)$, $b'(t)$ et $v(x, t)$ de la manière suivante : $a'(t) = v(a(t), t)$ et $b'(t) = v(b(t), t)$. Donc

$$\begin{aligned} M'(t) &= \int_{a(t)}^{b(t)} \frac{\partial u}{\partial t}(x, t) dx + v(b(t), t)u(b(t), t) - v(a(t), t)u(a(t), t) \\ &= \int_{a(t)}^{b(t)} \frac{\partial u}{\partial t}(x, t) dx + \int_{a(t)}^{b(t)} \frac{\partial}{\partial x}(u(x, t)v(x, t)) dx \\ &= \int_{a(t)}^{b(t)} \left(\frac{\partial u}{\partial t}(x, t) + \frac{\partial}{\partial x}(u(x, t)v(x, t)) \right) dx \\ &= 0. \end{aligned}$$

Comme ce résultat est vrai pour tout intervalle $]a(t); b(t)[\subset \mathbb{R}$, on en déduit que

$$\forall (x, t) \in \mathbb{R}^2, \frac{\partial u}{\partial t}(x, t) + \frac{\partial}{\partial x}(u(x, t)v(x, t)) = 0.$$

Si on se place dans le cas $v(x, t) = c$, on obtient l'équation :

$$\frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) = 0.$$

Dans le cas de l'équation avec un terme source, il suffit de remarquer que :

$$M'(t) = \int_a^b f(x, t) dx$$

En effet, la variation de matière dans l'intervalle considéré est égale à la somme de toutes les variations ponctuelles données par $f(x, t)$.

Le même raisonnement nous permet d'établir l'équation avec un terme source :

$$\frac{\partial u}{\partial t}(x, t) + c \frac{\partial u}{\partial x}(x, t) = f(x, t).$$

Chapitre 2

Résolution analytique de l'équation de transport

On cherche à résoudre l'équation :

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = f(x, t)$$

Avec les conditions :

$$\begin{cases} u(x, 0) = u_0(x) & \text{si } x \geq ct \\ u(0, t) = \varphi(t) & \text{si } x < ct \end{cases}$$

On a aussi $x \in [0, 1]$ et $t \in \mathbb{R}^+$.

On cherche des solutions u de classe \mathcal{C}^1 .

2.1 Équation homogène

Si $x \geq ct$

On pose $C_\xi = \{(x(t); t), t \in \mathbb{R}^+ \mid x'(t) = c \text{ et } x(0) = \xi\}$.

On pose $g(t) = u(x(t); t)$ On a donc

$$\begin{aligned} g'(t) &= \frac{\partial u}{\partial t}(x(t), t) + x'(t) \frac{\partial u}{\partial x}(x(t), t) \\ &= \frac{\partial u}{\partial t}(x(t), t) + c \frac{\partial u}{\partial x}(x(t), t) \\ &= 0. \end{aligned}$$

Donc $u(x(t), t) = u(x(0), 0) = u_0(\xi)$.

On cherche alors à calculer ξ .

On a $x'(t) = c$ d'où $x(t) = \int x'(t) dt = ct + x(0) = ct + \xi$.

Donc $(x(t), t) \in C_\xi \Leftrightarrow x = ct + \xi \Leftrightarrow \xi = x - ct$.

D'où :

$$u(x(t), t) = u_0(\xi) = u_0(x - ct).$$

Si $x < ct$

On peut utiliser un raisonnement similaire. On obtient alors :

$$u(x(t), t) = \varphi\left(\frac{ct - x}{c}\right).$$

Résultat : Solutions de l'équation homogène

Les solutions du problème sont :

$$\begin{cases} u(x, t) = u_0(x - ct) & \text{si } x \geq ct \\ u(x, t) = \varphi\left(\frac{ct-x}{c}\right) & \text{si } x < ct \end{cases}$$

On passe d'une solution à l'autre en $t = \frac{1}{c}$.

2.2 Cas d'une équation avec un terme source

Dans ce cas, la méthode des caractéristiques est toujours valable.

Si $x \geq ct$

On pose $C_\xi = \{(x(t); t), t \in \mathbb{R}^+ \mid x'(t) = c \text{ et } x(0) = \xi\}$.

On pose $g(t) = u(x(t), t)$ et on a $g'(t) = \frac{\partial u}{\partial t}(x(t), t) + c \frac{\partial u}{\partial x}(x(t), t) = f(x, t)$. On peut alors intégrer d'où

$$\begin{aligned} \int_0^t g'(\tau) d\tau &= \int_0^t f(x(\tau), \tau) d\tau \\ g(t) - g(0) &= \int_0^t f(x(\tau), \tau) d\tau \\ u(x(t), t) &= u(x(0), 0) + \int_0^t f(x(\tau), \tau) d\tau. \end{aligned}$$

Calculons $g(0)$:

On a $g(0) = u(x(0), 0) = u_0(\xi)$ et $x(t) = ct + \xi \Rightarrow \xi = x - ct$. Donc $g(0) = u_0(x - ct)$.

Calculons $f(x(\tau), \tau)$:

On a $x(\tau) = c\tau + \xi = x - c(t - \tau)$ d'où $f(x(\tau), \tau) = f(x - c(t - \tau), \tau)$.

La solution est donc :

$$u(x, t) = u_0(x - ct) + \int_0^t f(x - c(t - \tau), \tau) d\tau.$$

Si $x < ct$

Par le même raisonnement, on obtient :

$$u(x, t) = \varphi\left(\frac{ct-x}{c}\right) + \int_0^x f\left(e, t - \frac{1}{c}(x-e)\right) de$$

Résultat : Solutions de l'équation avec second membre

Les solutions du problème sont :

$$\begin{cases} u(x, t) = u_0(x - ct) + \int_0^t f(x - c(t - \tau), \tau) d\tau & \text{si } x \geq ct \\ u(x, t) = \varphi\left(\frac{ct-x}{c}\right) + \int_0^x f\left(e, t - \frac{1}{c}(x-e)\right) de & \text{si } x < ct \end{cases}$$

On passe d'une solution à l'autre en $t = \frac{1}{c}$.

2.3 Application

On se donne les conditions suivantes :

$$\begin{cases} u_0(x) = x & \text{si } x \geq ct \\ \varphi(t) = \sin(10\pi t) & \text{si } x < ct \end{cases}$$

De plus la vitesse est constante et $c = 4$.

On obtient alors le résultat suivant :

Résultat : *Solutions de l'équation homogène*

Les solutions du problème sont :

$$\begin{cases} u(x, t) = x - \frac{t}{4} & \text{si } x \geq 4t \\ u(x, t) = \sin(10\pi \frac{4t-x}{4}) & \text{si } x < 4t \end{cases}$$

On passe d'une solution à l'autre en $t = \frac{1}{4}$.

Chapitre 3

Résolution numérique de l'équation de transport

3.1 Méthode des différences finies

3.1.1 Principe

Les but est d'approximer les dérivées partielles par des quotients finis de la forme $\frac{\partial u}{\partial t} \approx \frac{u(x,t+h)-u(x,t)}{h}$ avec h petit et fixé.

Nous verrons qu'il existe plusieurs schémas possibles pour calculer cette approximation.

Pour réaliser ce travail, nous devons d'abord discrétiser l'espace-temps. Pour cela, on prend :

- N le nombre de points d'espace,
- T la durée,
- h le pas d'espace,
- τ le pas de temps.

3.1.2 Discrétiser l'espace

L'intervalle d'origine est $[0, 1]$, on choisit donc le nombre de points $N \in \mathbb{N}$ et on a $h = \frac{1}{N}$.

On définit alors la suite des points de l'espace :

$$x_n = x_0 + nh = nh.$$

On a donc $x_0 = 0$ et $x_N = 1$.

3.1.3 Discrétiser le temps

L'intervalle d'origine est \mathbb{R}^+ . On ne peut pas faire comme pour l'espace. On fixe donc τ petit. Si possible on choisit un τ dont la représentation machine est exacte. On étudie alors l'équation à T fixé.

On définit de même la suite des points de temps : $t_p = p\tau$.

On peut alors poser les deux suites suivantes :

$$u_n^p = u(x_n, t_p) \text{ et } f_n^p = f(x_n, t_p).$$

Nous allons maintenant nous intéresser à deux schémas possibles pour la résolution numérique de notre équation.

3.2 Schéma explicite

3.2.1 Approximation des dérivées partielles

On utilise les développements de Taylor suivants :

$$u(x_n, t_{p+1}) = u(x_n, t_p) + \tau \frac{\partial u}{\partial t}(x_n, t_p) + o(\tau) \quad (3.1)$$

$$u(x_{n-1}, t_p) = u(x_n, t_p) - h \frac{\partial u}{\partial x}(x_n, t_p) + o(h) \quad (3.2)$$

On obtient donc :

$$\frac{\partial u}{\partial t} \approx \frac{u_n^{p+1} - u_n^p}{\tau} \text{ et } \frac{\partial u}{\partial x} \approx \frac{u_n^p - u_{n-1}^p}{h}.$$

Les conditions initiales nous donnent les valeurs des $u_0^p = u_0(t_p) = u_0(p\tau)$ et les conditions aux limites nous donnent $u_n^0 = \varphi(x_n) = \varphi(nh)$.

3.2.2 Détermination de l'équation

On veut résoudre

$$\frac{u_n^{p+1} - u_n^p}{\tau} + c \frac{u_n^p - u_{n-1}^p}{h} = f_n^p \Leftrightarrow u_n^{p+1} = \tau f_n^p + u_n^p \left(1 - \frac{c\tau}{h}\right) + \frac{c\tau}{h} u_{n-1}^p$$

On pose $\gamma = \frac{c\tau}{h}$ d'où :

$$u_n^{p+1} = \tau f_n^p + u_n^p(1 - \gamma) + \gamma u_{n-1}^p.$$

Remarque : Dans le cas de l'équation sans terme source, on a $\forall (n, p) \in \mathbb{N}^2, f_n^p = 0$, d'où :

$$u_n^{p+1} = u_n^p(1 - \gamma) + \gamma u_{n-1}^p.$$

3.2.3 Résolution

On peut déduire u_n^{p+1} de u_n^p et u_{n-1}^p . En connaissant les conditions initiales et limites, on déduit une donnée inconnue de deux données connues, d'où le nom de schéma explicite.

Attention : Ce schéma est simple à mettre en œuvre, mais il présente un inconvénient majeur : il n'est pas toujours stable, il ne converge donc pas toujours vers la solution.

La convergence est acquise quand $\gamma < 1$.

3.3 Schéma implicite

Nous allons maintenant étudier le schéma implicite. Il présente l'avantage d'être toujours stable et donc de toujours converger vers la solution.

3.3.1 Approximation des dérivées partielles

Ce schéma est très similaire au schéma explicite. La seule différence réside dans l'approximation de la dérivée en espace. On a ici :

$$\frac{\partial u}{\partial x} \approx \frac{u_n^{p+1} - u_{n-1}^{p+1}}{h}.$$

3.3.2 Détermination de l'équation

L'équation du schéma implicite est donc :

$$\frac{u_n^{p+1} - u_n^p}{\tau} + c \frac{u_n^{p+1} - u_{n-1}^{p+1}}{h} = f_n^p$$

c'est-à-dire, avec $\gamma = \frac{c\tau}{h}$:

$$u_n^{p+1}(1 + \gamma) - \gamma u_{n-1}^{p+1} = \tau f_n^p + u_n^p.$$

On pose $v_n = u_n^{p+1}$ et $w_n = \tau f_n^p + u_n^p$.

Remarque : Dans le cas de l'équation sans terme source, on a $w_n = u_n^p$.

Notre équation s'écrit alors :

$$v_n(1 + \gamma) - \gamma v_{n-1} = w_n.$$

Si on connaît u_n^p , alors on peut en déduire les valeurs de u_n^{p+1} grâce au système suivant :

$$\begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -\gamma & 1 + \gamma & \ddots & & \vdots \\ 0 & -\gamma & 1 + \gamma & \ddots & \vdots \\ \vdots & \ddots & -\gamma & 1 + \gamma & 0 \\ 0 & \cdots & 0 & -\gamma & 1 + \gamma \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ \vdots \\ w_N \end{pmatrix}$$

3.3.3 Résolution

On connaît v_0 , on peut donc en déduire v_1 :

$$-\gamma v_0 + (1 + \gamma)v_1 = w_1 \Leftrightarrow v_1 = \frac{w_1 + \gamma v_0}{1 + \gamma}.$$

De même v_i donne v_{i+1} . On obtient ainsi tous les u_n^{p+1} .

Remarque : Il n'est donc pas nécessaire de mettre en place un algorithme pour résoudre ce système linéaire.

3.4 Comparaison

Les modèles explicites et implicites sont testés sur 2 exemples différents.

Premier exemple : On prend comme condition initiale $U_0(x) = x$ et comme condition à la limite $\varphi(t) = \sin(10\pi t)$. On prend également $C = 4$. On a, dans cet exemple, $\gamma = \frac{c\tau}{h} = 0.8 < 1$, donc le schéma explicite est utilisable.

Deuxième exemple : On prend comme condition initiale $U_0 = 0$, et comme condition aux limites $\varphi := C \times fH(t) \times \sin(t) = fH(t) \times \sin(t)$, car dans cet exemple, $C = 1$. On a $fH(t)$ une fonction échelon, c'est à dire que si $t > 0$, $fH(t) = 1$ et sinon $fH(t) = 0$. On a, dans cet exemple, $\gamma = \frac{c\tau}{h} = 0.2 < 1$. Le schéma explicite est donc utilisable.

Pour chaque exemple, il y aura 4 groupes de fichiers.

- Un groupe de fichiers contient les données du modèle implicite sans terme source
- Un groupe de fichiers contient les données du modèle implicite avec terme source
- Un groupe de fichiers contient les données du modèle explicite sans terme source
- et le dernier groupe de fichiers contient les données du modèle explicite avec terme source.

3.4.1 Outils créés

Nous avons fait différents petits programmes (implémentés en Python) pour traiter nos résultats.

- Il y a un programme permettant de tracer les courbes explicites, implicites et analytiques pour un temps donné. Ces courbes représentent donc la concentration en fonction de la position x .
- Il y a un programme qui permet de faire une animation, affichant pour un temps donné les courbes. Chaque image de l'animation correspond à un temps donné, et donc ce programme permet d'appréhender l'évolution de la concentration en fonction de x au cours du temps.
- Il y a un programme qui fait la somme des écarts ponctuels (c'est à dire point par point) entre les courbes implicites et explicites et la courbe théorique. Cela permet donc de comparer les écarts entre les deux modèles et la solution théorique pour la totalité des temps (le programme fait la somme des sommes des écarts ponctuels pour un temps donné pour chacun des temps).

Ces outils prennent la forme de scripts Python, respectivement contenus dans les fichiers `trace_une_courbe_base.py` (ou `trace_une_courbe_source.py` pour les exemples avec terme source), `animation.py` (ou `animation_source.py` pour les exemples avec terme source) et `differenceMoyenne_base.py` (ou `differenceMoyenne_source.py` pour les exemples avec terme source).

3.4.2 Comparaison des deux modèles

Conditions de stabilité

Avant de comparer les résultats obtenus par les modèles implicites et explicites, il est important de rappeler que le modèle explicite nécessite des conditions de stabilité pour pouvoir être appliqué (c'est à dire $\gamma = \frac{c\tau}{h} < 1$), alors que le modèle implicite peut s'appliquer dans tous les cas.

Premier exemple

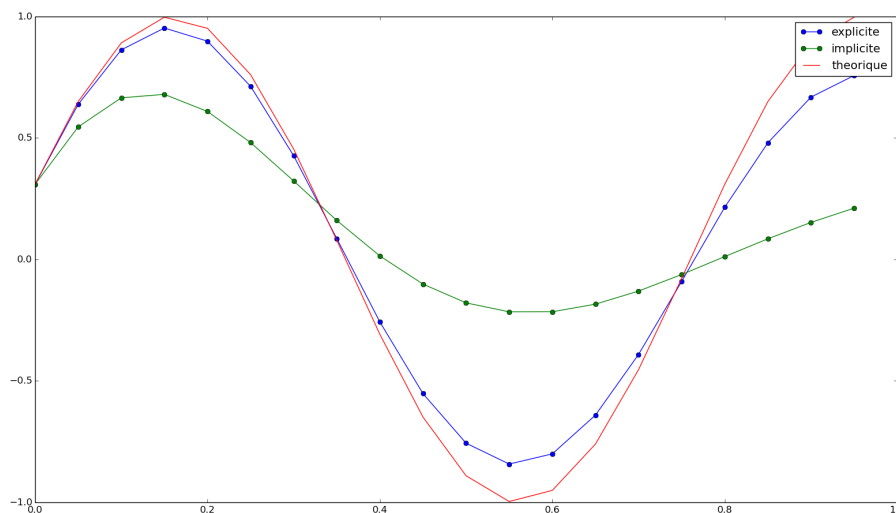


FIGURE 3.1 – Résultats de nos programmes et solution théorique – Premier exemple

Grâce au programme `animation.py`, on observe que les courbes sont plus ou moins sinusoidales. La courbe du schéma explicite est très proche de celle théorique, par contre il y a des écarts relativement importants entre la courbe du schéma implicite et la courbe théorique. Ces deux dernières ont les mêmes variations, mais l'amplitude des sinusoides de la courbe du schéma implicite est beaucoup plus faible que celle des sinusoides de la solution théorique. Graphiquement, on pourrait dire que le modèle implicite 'lisse' les courbes.

L'insertion d'un terme source n'apporte pas beaucoup de changements à la représentation graphique des solutions. Les différences entre les courbes des modèles explicites et implicites et celle de la solution théorique restent du même ordre. L'écart entre les solutions du modèle explicites et les solutions théoriques est juste légèrement supérieur à celui sans terme source. Sans terme source, d'après notre programme `differenceMoyenne_base.py`, on a une somme des écarts entre le modèle explicite et la solution théorique de 3.6, alors qu'avec terme source, d'après notre programme `differenceMoyenne_source.py`, cette somme des écarts est de 5.1 (pour la somme des écarts entre le modèle implicite et la solution théorique, il est de 17.3 sans terme source et 17.5 avec terme source).

Conclusion : Sur ce premier exemple, on remarque que les solutions du modèle explicite sont assez proches de celles théoriques, alors que les solutions du modèle implicite sont légèrement différentes, bien que les variations des courbes représentant les solutions du modèle implicite et les solutions théoriques soient les mêmes.

Deuxième exemple

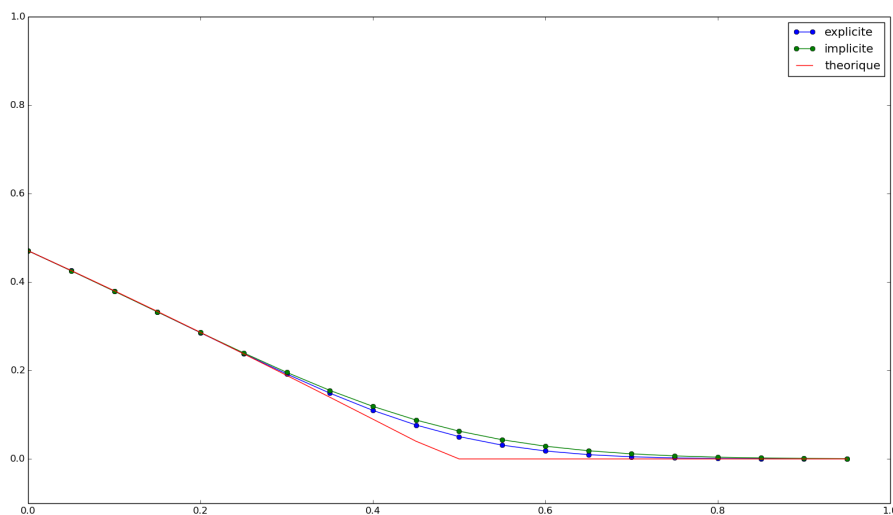


FIGURE 3.2 – Résultats de nos programmes et solution théorique – Deuxième exemple

Dans cet exemple, la représentation graphique de la solution théorique sans terme source est composée de 2 droites :

- Une droite $y = 0$.
- La deuxième droite va de $x = 0$ à $x = t$. Elle correspond à $y = t - x$ (donc en $x = t$, $y = 0$) en général (sauf à un temps qui tend vers 1, la courbe prendra un aspect assez particulier).

Sans terme source, on a le même constat que pour le premier exemple : le modèle explicite approxime bien la solution théorique, alors que le modèle implicite lisse un peu plus la courbe. Par contre, dans cet exemple, le lissage de la courbe n'a pas de grosses conséquences (car il n'y a pas beaucoup de variations). La seule variation se situe aux alentours de $x = a$, c'est à dire $x = t$. Les deux modèles approximent donc bien la solution théorique.

Malheureusement, il y a des problèmes en présence du terme source. Les solutions des modèles explicites et implicites varient beaucoup alors que la solution théorique ne varie presque pas ! Les deux modèles ont donc beaucoup de mal à approximer la solution théorique, pour des raisons que nous ne saurions expliquer.

Conclusion Dans cet exemple, les deux modèles approximent bien la solution théorique. Par contre, nous avons un problème inexplicable avec le terme source...

3.4.3 Conclusion de cette comparaison

On ne peut évidemment pas définir un modèle comme étant meilleur que l'autre. Cependant, voici les avantages et inconvénients de chacun des modèles :

- **Le modèle explicite** approxime très bien les solutions théoriques. En revanche, il nécessite le respect de la condition de stabilité.
- Bien qu'il suive correctement les variations de la solution théorique, **le modèle implicite** a tendance à lisser les courbes. Cependant, le gros avantage de ce modèle est qu'il n'y a pas de condition de stabilité à respecter.

Chapitre 4

Applications

4.1 Que représente physiquement l'équation de transport ?

C'est une équation de convection.

La convection est un phénomène physique clé dans le mouvement de fluide. Il représente l'ensemble des mouvements internes animant un fluide et qui impliquent le transport des propriétés de portions de fluide au cours de son déplacement. Il sert de mode de transfert de chaleur, se produisant entre une surface et un fluide à son contact ou bien lors d'un échange de chaleur.

Dans la convection libre le mouvement du fluide est dû à sa source sous l'effet de gradients de masse volumique et d'un champ de pesanteur, alors que pour la convection forcée le mouvement est dû à un apport d'énergie mécanique externe. La convection mixte représente le couplage des deux phénomènes, celle-ci est souvent assimilée aux mouvements de fluides dans des canalisations.

4.2 Application de nos programmes à une situation physique concrète

On peut prendre un exemple concret de l'équation de transport : le transport de pollution par le vent. Nous vous proposons donc de traiter un cas particulier : On étudie la quantité de CO_2 au-dessus d'une ville pendant 1h. Cette ville s'étant sur 10km, avec à l'entrée ouest de la ville une usine qui utilise des hauts fourneaux 3 fois par heure, ce qui entraîne une surproduction de CO_2 relâché dans l'air. On modélise cette sur pollution par la fonction du temps suivante :

$$\varphi(t) = 5|\sin(10t)|$$

Cependant, l'usine n'est pas la seule source de pollution, la ville et ses voitures dégagent aussi du CO_2 . C'est pourquoi, en plus de φ , il faut ajouter un terme source. Celui-ci est représenté ci-dessous :

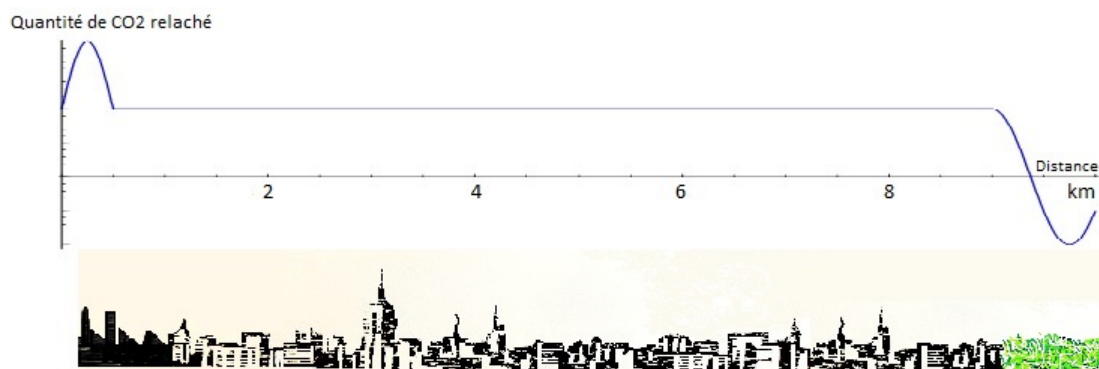


FIGURE 4.1 – Terme source

Le premier pique correspond à une pollution continue au-dessus de l'usine (bien plus faible que lors des piques représentés par φ), ensuite, le plateau représente une pollution continue dégagé par

la ville, enfin, le creux à la fin correspond au CO_2 absorbé par la végétation, ainsi qu'à une dépression qui dévie les particules de la droite d'étude.

Au moment de l'étude, on imagine un vent de 10km/h qui va d'ouest en est. On a utilisé les programmes expliqués dans les chapitres précédents pour calculer la théorie de cette expérience, et on voit que dans le cas présent, le terme source est négligeable devant la pollution industrielle. Finalement, au bout d'une heure, on observe φ , non plus en fonction du temps, mais de l'espace avec, donc, un changement d'échelle.

Conclusion et perspectives

Travail réalisé

Contrairement à d'autres projets de modélisation, nous n'avons pas continué un projet déjà existant.

Nous avons étudié la théorie de l'équation de transport, c'est-à-dire sa justification et sa résolution théorique. Nous nous sommes ensuite intéressés à sa résolution numérique avec la méthode des différences finies. Pour finir, nous avons appliqué notre modèle à une situation physique concrète.

Apport personnel de ce projet

On peut diviser les apports de ce projet en quatre domaines.

Sur le plan mathématiques, il nous a permis de découvrir une méthode de résolution numérique des équations aux dérivées partielles.

Sur le plan informatique, il nous a permis de mettre en pratique les connaissances de nos cours d'informatique et de MAO.

Sur le plan physique, il nous a permis de comprendre comment établir un modèle physique associé à un phénomène simple.

Sur le plan du travail en groupe, ils nous a permis d'apprendre à nous organiser et à nous séparer le travail.

Poursuite de ce projet

Pour poursuivre ce projet, plusieurs pistes sont envisageables. On pourrait étendre le modèle à deux ou trois dimensions spatiales : le modèle deviendrait alors plus intéressant et plus facilement applicable à des problèmes physiques comme la convection. Il est également possible de s'intéresser au cas des vitesses non-constantes. On peut aussi utiliser d'autres schémas de résolution numérique.

D'autres projets sont possibles sur le thème du transport de polluant dans l'air. On peut par exemple étudier la forme des panaches de fumée à la sortie des cheminées des usines.

Bibliographie

- [1] *Analyse numérique des équations aux dérivées partielles*
Laurent DI MENZA
- [2] *Méthodes de Monte-Carlo pour les équations de transport et de diffusion*
Bernard LAPEYRE, Étienne PARDOUX et Rémi SENTIS.
- [3] *Méthodes numériques pour les EDP en mécanique*
Cours de Pierre BUFFAT (Université C.Bernard de Lyon)
http://www.ufrmeca.univ-lyon1.fr/~buffat/COURS/COURSDF_HTML/node25.html
- [4] *Transport d'une solution saline en cellule de Hele-Shaw - Expériences et simulations*
Thèse de Christophe FELDER
<https://tel.archives-ouvertes.fr/tel-00006120/>
- [5] *Cours de Mécanique des Fluides*
Cours de l'ESPCI
<https://cours.espci.fr/site.php?id=2&fileid=300>
- [6] *Méthode des caractéristiques*
Wikipédia
http://fr.wikipedia.org/wiki/M%C3%A9thode_des_caract%C3%A9ristiques
- [7] *Méthode des différences finies*
Wikipédia
http://fr.wikipedia.org/wiki/M%C3%A9thode_des_diff%C3%A9rences_finies
- [8] *Construction du Schéma explicite*
ENSTA-Paristech
http://perso.ensta-paristech.fr/~fliss/Sonia_Fliss_web_page/Enseignement_files/Amphi2_TransportNum_print.pdf
- [9] *Schéma implicite de Crank Nicolson*
Laurent RISSER
http://laurent.risser.free.fr/THESE/coursM11_2006.pdf
- [10] *Déplacements océaniques*
Interstices.info
https://interstices.info/jcms/int_70245/comprendre-la-circulation-oceanique
- [11] *La convection dans le manteau terrestre*
Seismoatschool
http://www.seismoatschool.ethz.ch/index.php?option=com_content&view=article&id=82%3A1a-convection-dans-le-manteau-terrestre&catid=10%3Aterre&lang=fr
- [12] *Résolution numérique des équations d'Euler monodimensionnelles*
<http://www.hach.ulg.ac.be/cms/system/files/Cours%20Grenoble.pdf>
- [13] *Résolution d'équations différentielles*
Vincent GUINOT
<http://vincentguinot.free.fr/MNAFr.pdf>
- [14] *Méthode des différences finies*
Université du Texas
<http://farside.ph.utexas.edu/teaching/329/lectures/node90.html>

Annexe A

Programmes réalisés

A.1 Exemple 1 sans terme source

Programme permettant de calculer les solutions des modèles explicites et implicites et la solution théorique pour l'équation de convection sans terme source de l'exemple 1 :

```

program mod1;
uses sysutils;

// Exemple 1

const P = 100; // nombre de points dans le temps
N = 20; // nombre de points dans l espace
C = 4;
tau = 0.01;
T = P*tau; // longueur de l intervalle de temps
L = 1; // longueur de l intervalle dans l espace
h = L/N;

type Matrice = array[0..N, 0..P] of real;

function U0 (x : real) : real;
begin
    U0:=x;
end;

function phi (t : real) : real;
begin
    phi := sin(10*PI*t)
end;

procedure ecrireAnalytiqueT(Uth : Matrice; t : Integer);
var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_th_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N do
    Begin
        writeln(fichier, i*h, '␣', Uth[i][t]);
    end;
    close(fichier);
end;

procedure ecrireExpliciteT(Uexp : Matrice; t : Integer);

```



```

var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_exp_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N-1 do
    Begin
        writeln(fichier, i*h, '␣', Uexp[i][t]);
    end;
    close(fichier);
end;

procedure ecrireImpliciteT(Uimp : Matrice; t : Integer);
var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_imp_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N-1 do
    Begin
        writeln(fichier, i*h, '␣', Uimp[i][t]);
    end;
    close(fichier);
end;

var xi, ti : Integer;
    x, temps, r : real;
    Uexp, Uimp, Uth : Matrice;

Begin
    r := (C*tau)/h;
    // on entre les vecteurs U0 et phi pour la matrice solution
    explicite
    for xi := 1 to N do
        Uexp[xi,0] := U0(xi*h);
    for ti := 1 to P do
        Uexp[0,ti] := phi(ti*tau);

    // on entre les vecteurs U0 et phi pour la matrice solution
    implicite
    for xi := 1 to N do
        Uimp[xi,0] := U0(xi*h);
    for ti := 1 to P do
        Uimp[0,ti] := phi(ti*tau);

    // on calcule la solution numerique explicite
    for ti := 0 to P - 1 do
        for xi := 1 to N do
            Uexp[xi][ti+1] := (1-r)*Uexp[xi][ti] + r*Uexp[xi-1][
                ti];

    // on calcule la solution numerique implicite
    for ti := 0 to P-1 do
        for xi := 0 to N-1 do
            Uimp[xi+1,ti+1] := (1/(1+r))*(Uimp[xi+1,ti]+r*Uimp[
                xi,ti+1]);

```

```

// on calcule la solution analytique
for xi := 0 to N do
    for ti := 0 to P do
        begin
            x := xi * h;
            temps := ti * tau;
            if (c*temps) < x then
                Uth[xi,ti] := U0(x - C*temps)
            else
                Uth[xi,ti] := phi(temps - x/C);
        end;

writeln('On doit avoir R<1');
writeln('Ici, R=', (C*tau)/h);
if ((C*tau)/h > 1) then
    writeln('Ce n'est pas le cas ici...')
else
    Begin
        writeln('La condition est remplie, donc on peut utiliser le
            modele');
        for ti := 0 to P do
            ecrireExpliciteT(Uexp, ti);
        for ti := 0 to P do
            ecrireImpliciteT(Uimp, ti);
        for ti := 0 to P do
            ecrireAnalytiqueT(Uth, ti);
    end
end.
    
```

A.2 Exemple 1 avec terme source

Programme permettant de calculer les solutions des modèles explicites et implicites et la solution théorique pour l'équation de convection avec terme source de l'exemple 1 :

```

program mod1;
uses sysutils;

// Exemple 1

const P = 100; // nombre de points dans le temps
      N = 20; // nombre de points dans l'espace
      C = 4;
      tau = 0.01;
      T = P*tau; // longueur de l'intervalle de temps
      L = 1; // longueur de l'intervalle dans l'espace
      h = L/N;

type Matrice = array[0..N, 0..P] of real;

function source(x, t : real):real;
Begin
    source := 1;
end;

function U0 (x : real) : real;
begin
    U0:=x;
end;
    
```

```

function phi (t : real) : real;
begin
    phi := sin(10*PI*t)
end;

procedure ecrireAnalytiqueT(Uth : Matrice; t : Integer);
var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_source_th_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N do
    Begin
        writeln(fichier, i*h, '␣', Uth[i][t]);
    end;
    close(fichier);
end;

procedure ecrireExpliciteT(Uexp : Matrice; t : Integer);
var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_source_exp_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N-1 do
    Begin
        writeln(fichier, i*h, '␣', Uexp[i][t]);
    end;
    close(fichier);
end;

procedure ecrireImpliciteT(Uimp : Matrice; t : Integer);
var i : Integer;
    nom_fichier : string;
    fichier : TEXT;
Begin
    nom_fichier := 'resultat_source_imp_t' + FloatToStr(t*tau) + '.txt';
    assign(fichier, nom_fichier);
    rewrite(fichier);
    for i := 0 to N-1 do
    Begin
        writeln(fichier, i*h, '␣', Uimp[i][t]);
    end;
    close(fichier);
end;

var xi, ti : Integer;
    x, temps, r : real;
    Uexp, Uimp, Uth : Matrice;

Begin
    r := (C*tau)/h;
    // on entre les vecteurs U0 et phi
    for xi := 1 to N do
        Uexp[xi,0] := U0(xi*h);
    for ti := 1 to P do

```

```

        Uexp[0,ti] := phi(ti*tau);
// on entre les vecteurs U0 et phi pour la matrice solution
    implicite
    for xi := 0 to N do
        Uimp[xi,0] := U0(xi*h);
    for ti := 0 to P do
        Uimp[0,ti] := phi(ti*tau);
// on calcule la solution numerique
    for ti := 0 to P - 1 do
        for xi := 1 to N do
            Uexp[xi][ti+1] := (1-r)*Uexp[xi][ti] + r*Uexp[xi-1][
                ti] + tau*source(xi,ti);
// on calcule la solution numerique implicite
    for ti := 0 to P-1 do
        for xi := 0 to N-1 do
            Uimp[xi+1,ti+1] := (Uimp[xi+1,ti]+tau*source(xi,ti)+
                r*Uimp[xi,ti+1])/(1+r);
// on calcule la solution analytique
    for xi := 0 to N do
        for ti := 0 to P do
            begin
                x := xi * h;
                temps := ti * tau;
                if (c*temps) < x then
                    Uth[xi,ti] := U0(x - C*temps) + source(x,
                        temps)*temps
                else
                    Uth[xi,ti] := phi(temps - x/C);
            end;
    writeln('On doit avoir R<1');
    writeln('Ici, R=', (C*tau)/h);
    if ((C*tau)/h > 1) then
        writeln('Ce n'est pas le cas ici...')
    else
    Begin
        writeln('La condition est remplie, donc on peut utiliser le
            modele');
        for ti := 0 to P do
            ecrireExpliciteT(Uexp, ti);
        for ti := 0 to P do
            ecrireImpliciteT(Uimp, ti);
        for ti := 0 to P do
            ecrireAnalytiqueT(Uth, ti);
    end
end.

```

```

import numpy as np
import os

diffExp = 0
diffImp = 0
for j in range(1,100):
    t=j/100
    explicite = open("resultat_source_exp_t"+str(t)+".txt", "r")
    implicite = open("resultat_source_imp_t"+str(t)+".txt", "r")
    theorique = open("resultat_source_th_t"+str(t)+".txt", "r")
    texte_exp = explicite.read()
    texte_imp = implicite.read()
    texte_th = theorique.read()

```

```

texte_exp = texte_exp.split()
texte_imp = texte_imp.split()
texte_th = texte_th.split()

i = 0
nbPoints = len(texte_exp)
while i < nbPoints-1:
    diffExp += (abs(float(texte_th[i+1]) - float(texte_exp[i+1])
                )/nbPoints) # calcul l'ecart moyen par point
    diffImp += (abs(float(texte_th[i+1]) - float(texte_imp[i+1])
                )/nbPoints)
    i += 2

print(diffExp)
print(diffImp)

```

A.3 Script Python pour calculer les écarts

```

import numpy as np
import os

diffExp = 0
diffImp = 0
for j in range(1,100):
    t=j/100
    explicite = open("resultat_exp_t"+str(t)+".txt", "r")
    implicite = open("resultat_imp_t"+str(t)+".txt", "r")
    theorique = open("resultat_th_t"+str(t)+".txt", "r")
    texte_exp = explicite.read()
    texte_imp = implicite.read()
    texte_th = theorique.read()

    texte_exp = texte_exp.split()
    texte_imp = texte_imp.split()
    texte_th = texte_th.split()

    i = 0
    nbPoints = len(texte_exp)
    while i < nbPoints-1:
        diffExp += (abs(float(texte_th[i+1]) - float(texte_exp[i+1])
                    )/nbPoints) # calcul l'ecart moyen par point
        diffImp += (abs(float(texte_th[i+1]) - float(texte_imp[i+1])
                    )/nbPoints)
        i += 2

print(diffExp)
print(diffImp)

```

A.4 Script Python pour tracer la courbe à un temps donné

```

from pylab import *
import matplotlib.pyplot as plt
import numpy as np
import os

t=input("entrez le temps ")

```

```

mon_fichier = open("resultat_exp_t"+str(t)+".txt", "r")
texte_fichier = mon_fichier.read()
texte_fichier = texte_fichier.split()
x = list()
y = list()
i = 0
while i < len(texte_fichier)-1:
    x.append(texte_fichier[i])
    y.append(texte_fichier[i+1])
    i += 2
x = np.asarray(x)
y = np.asarray(y)
plt.plot(x, y, "b:o",label="explicite_a_t="+str(t))
mon_fichier.close()

mon_fichier = open("resultat_imp_t"+str(t)+".txt", "r")
texte_fichier = mon_fichier.read()
texte_fichier = texte_fichier.split()
x = list()
y = list()
i = 0
while i < len(texte_fichier)-1:
    x.append(texte_fichier[i])
    y.append(texte_fichier[i+1])
    i += 2
x = np.asarray(x)
y = np.asarray(y)
plt.plot(x, y, "g:o",label="implicite_a_t="+str(t))
mon_fichier.close()

mon_fichier = open("resultat_th_t"+str(t)+".txt", "r")
texte_fichier = mon_fichier.read()
texte_fichier = texte_fichier.split()
x = list()
y = list()
i = 0
while i < len(texte_fichier):
    x.append(texte_fichier[i])
    y.append(texte_fichier[i+1])
    i += 2
x = np.asarray(x)
y = np.asarray(y)
plt.plot(x, y, "r",label="theorie_a_t="+str(t))
legend()

plt.show() # affiche la figure a l'ecran

mon_fichier.close()
    
```

A.4.1 Script pour calculer la somme des écarts

Avec terme source :

```

import numpy as np
import os

diffExp = 0
diffImp = 0
for j in range(1,100):
    t=j/100
    explicite = open("resultat_source_exp_t"+str(t)+".txt", "r")
    
```

```

implicite = open("resultat_source_imp_t"+str(t)+".txt", "r")
theorique = open("resultat_source_th_t"+str(t)+".txt", "r")
texte_exp = explicite.read()
texte_imp = implicite.read()
texte_th = theorique.read()

texte_exp = texte_exp.split()
texte_imp = texte_imp.split()
texte_th = texte_th.split()

i = 0
nbPoints = len(texte_exp)
while i < nbPoints-1:
    diffExp += (abs(float(texte_th[i+1]) - float(texte_exp[i+1])
                )/nbPoints) # calcul l'ecart moyen par point
    diffImp += (abs(float(texte_th[i+1]) - float(texte_imp[i+1])
                )/nbPoints)
    i += 2

print(diffExp)
print(diffImp)
    
```

A.5 Script Python pour l'animation

```

from pylab import *
import matplotlib.pyplot as plt
import numpy as np
import os
fig, ax = plt.subplots()
t = 0
while t < 1:

    fichier_exp = open("resultat_exp_t"+str(t)+".txt", "r")
    fichier_imp = open("resultat_imp_t"+str(t)+".txt", "r")
    fichier_th = open("resultat_th_t"+str(t)+".txt", "r")
    texte_exp = fichier_exp.read()
    texte_exp = texte_exp.split()
    texte_imp = fichier_imp.read()
    texte_imp = texte_imp.split()
    texte_th = fichier_th.read()
    texte_th = texte_th.split()
    xexp = list()
    yexp = list()
    ximp = list()
    yimp = list()
    xth = list()
    yth = list()
    i = 0
    while i < len(texte_exp)-1:
        xexp.append(texte_exp[i])
        yexp.append(texte_exp[i+1])
        ximp.append(texte_imp[i])
        yimp.append(texte_imp[i+1])
        xth.append(texte_th[i])
        yth.append(texte_th[i+1])
        i += 2

    xexp = np.asarray(xexp)
    yexp = np.asarray(yexp)
    fichier_exp.close()
    
```

```

ximp = np.asarray(ximp)
yimp = np.asarray(yimp)
fichier_imp.close()
xth = np.asarray(xth)
yth = np.asarray(yth)
fichier_th.close()

t = t + 0.01
t = round(t, 2)

points, = ax.plot(xexp, yexp, marker='o', label='explicite')
points, = ax.plot(ximp, yimp, marker='o', label='implicite')
points, = ax.plot(xth, yth, label='theorique')
legend()
ax.set_xlim(0, 1)
ax.set_ylim(-0.1, 1)
plt.pause(0.1)
ax.clear()
    
```