

TRAVAIL ENCADRÉ DE RECHERCHE

Modélisation d'une onde dans un milieu liquide-gaz

Thomas Di Giusto
Lecoutère Simon

Encadré par Ana C. Matos

01/06/2017

Remerciements

Nous souhaitons remercier notre encadrante, Pr. Ana Matos, pour l'aide qu'elle nous a apportée, autant par les axes de travail qu'elle nous a indiqué que par sa disponibilité qui nous a permis de nous sentir réellement suivi durant la rédaction de ce rapport. Nous souhaitons également remercier nos professeurs Thomas Rey et André De Laire qui ont eu l'amabilité de répondre à quelques questions sur notre T.E.R.

Table des matières

1	Introduction	1
2	Position du problème	2
3	Établissement de l'équation	4
I.	Température constante	4
II.	Température non constante	11
III.	Juxtaposition de deux milieux d'indices différents	12
IV.	Considérations annexes	13
a.	Ajout d'un terme de dissipation	13
b.	Dans un milieu adiabatique	13
V.	Retour sur l'équation des ondes	14
4	Simulation numérique avec la méthode d'Euler	17
I.	Cas de référence	17
II.	Changement d'homogénéité	19
III.	Impulsion initiale	20
IV.	Température non constante	21
V.	Phénomène de dissipation	22
5	Interprétations	23
I.	Température constante	23
a.	Conditions initiales nulles	23
b.	Conditions initiales non nulles	30
II.	Température non constante	44
III.	Phénomène de dissipation	50
IV.	Temps d'exécution et stabilité numérique	55
a.	Temps d'exécution	55
b.	Stabilité	56
6	Schéma de Runge-Kutta	59
I.	Méthode	59
II.	Stabilité	61

7 Applications	64
I. Sécurité d'un pipeline	64
II. Rideaux de bulles	65
8 Le Code	66
I. C++	67
II. Maple	83
a. Code Source	83
b. Graphiques	85

Chapitre 1

Introduction

Notre étude va porter sur le comportement d'un mélange de deux fluides. Nous choisissons d'étudier un mélange liquide-gaz. Nous désignons par liquide une entité liquide homogène ou une réunion de plusieurs entités liquides différentes, et nous désignons par gaz l'entité compressible choisie qui est présente sous forme de bulles infimes de tailles *a priori* différentes dans le liquide.

Ce compte-rendu est décomposé en plusieurs parties : Nous allons définir les lois physiques du problème ainsi que nos approximations. Nous allons ensuite établir l'équation générale qui régit le mouvement de ce mélange selon plusieurs situations. Ensuite, toujours selon ces mêmes situations, nous allons construire notre modélisation numérique à partir des approximations que nous avons précédemment choisies et de la méthode des différences finies. Après quoi nous allons étudier et interpréter les résultats obtenus lors de notre simulation numérique. Une partie sera également consacrée à des techniques de simulations différentes en vue d'un résultat de meilleure qualité.

Chapitre 2

Position du problème

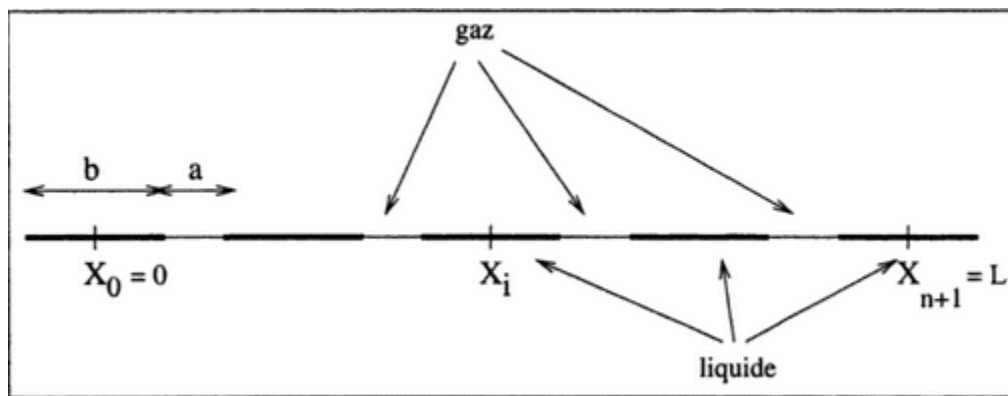


FIGURE 2.1 – Modèle Physique

Dans tout le problème, on se place en dimension 1. C'est donc un modèle linéique que nous décrirons par la suite.

Soit ρ la masse linéique du liquide

$$\rho = \frac{m_{\text{liquide}}}{L_{\text{liquide}}} \Leftrightarrow m_{\text{liquide}} = \rho \cdot b$$

D'après la loi des gaz parfaits :

$$P.V = N.R.T$$

où :

R : la constante des gaz parfaits, $R = 8.314$

P : la pression du gaz

V : le volume du gaz

N : le nombre de molécules du gaz

T : la température du gaz

Comme T est constante, on pose $k = R.T$ qui est constant.

On a donc : $P.V = k.N$

De plus, on suppose les extrémités fixes :

$$\begin{cases} x_0 = 0 \\ x_{n+1} = L \end{cases}$$

On pose :

$N = \beta(a + b)$ où β est le nombre de molécules de gaz par unité de longueur.

$h = a + b$

$X_i = ih$ la position des centres de segment fluide à l'équilibre.

On va chercher à établir le système d'équations qui régit le mouvement des bouchons.

Chapitre 3

Établissement de l'équation

Dans cette partie, on va établir les équations dans le cas d'une température constante et d'une température non constante ainsi que dans le cas d'un milieu adiabatique puis avec un terme de dissipation.

Notations :

$$\begin{aligned}\frac{du}{dt}(x, t) &= \dot{u}(x, t); & \frac{du}{dx}(x, t) &= u'(x, t) \\ \frac{d^2u}{dt^2}(x, t) &= \ddot{u}(x, t); & \frac{d^2u}{dx^2}(x, t) &= u''(x, t)\end{aligned}$$

I. Température constante

On note $x_i = X_i + u_i$ la position du centre du i^{eme} segment de fluide avec $u_i = u(x_i)$ les écarts des positions des centres.

On va établir l'équation du système :

Le système : élément de fluide centré en x_i

Les forces : F_i : résultante des pressions P^d et P^g respectivement exercées à droite et à gauche par le gaz.

Théorème 1.

$$\vec{F}_i = k.\beta.h.\left(\frac{1}{a + u_i - u_{i-1}} - \frac{1}{a + u_{i+1} - u_i}\right).\vec{u}_x$$

Démonstration. En effet : $P.V = k.N$

$$\begin{aligned}\Leftrightarrow P &= \frac{k.N}{V} \\ \Leftrightarrow P &= \frac{k.\beta.(a + b)}{V} \\ \Leftrightarrow P &= \frac{k.\beta.h}{V}\end{aligned}$$

avec $N = \beta.(a + b)$ et $h = (a + b)$

Or, le volume V représente une longueur dans le cadre du modèle 1D que nous allons noter L . De plus, P_i est la force de pression s'exerçant à gauche et à droite du fluide en x_i :

$$\vec{P}_i^g = \frac{k \cdot \beta \cdot h}{a + u_i - u_{i-1}} \vec{u}_x$$

$$\vec{P}_i^d = \frac{-k \cdot \beta \cdot h}{a + u_{i+1} - u_i} \vec{u}_x$$

Or, F_i est la somme des pressions exercées à droite et à gauche par le gaz. On obtient ainsi la relation :

$$F_i = P_i^g + P_i^d = k \cdot \beta \cdot h \cdot \left(\frac{1}{a + u_i - u_{i-1}} - \frac{1}{a + u_{i+1} - u_i} \right)$$

□

On applique ensuite la relation fondamentale de la dynamique :

$$\sum_i (\vec{F}_i) = m_{\text{fluide}} \cdot \vec{a}$$

$$\Leftrightarrow \vec{F}_i = \rho \cdot b \cdot \vec{u}_i$$

On projete sur l'axe (Ox) :

$$F_i = \rho \cdot b \cdot \ddot{u}_i$$

$$\Leftrightarrow F_i = \rho \cdot b \cdot \frac{dv}{dt}$$

En notant $u = (u_1, \dots, u_n)$, on obtient :

$$F(u) = \rho \cdot b \cdot \frac{dv}{dt}$$

On arrive au système différentiel suivant, noté (S) :

$$\begin{cases} \frac{du}{dt} = v \\ \rho \cdot b \cdot \frac{dv}{dt} = F(u) \end{cases}$$

Théorème 2 (Existence et Unicité de la solution).

Le système (S) admet une unique solution.

Démonstration. Penchons nous sur l'existence et l'unicité d'une solution à (S). Fixons la variable en espace.

Ce système peut alors se réécrire :

$$u' = v, \quad v' = F(u(t), t) \quad \text{avec } u = (u_1, \dots, u_N) \text{ et } v = (v_1, \dots, v_N)$$

avec

$$F = \begin{bmatrix} F_1 \\ \cdot \\ \cdot \\ F_N \end{bmatrix}$$

et $\forall i = 1, \dots, N$

$$F_i(u, t) = \frac{k\beta h}{\rho_n b} \left(\frac{1}{a + u_i - u_{i-1}} - \frac{1}{a + u_{i+1} - u_i} \right)$$

Toujours dans nos considérations dans le cadre isotherme, on peut considérer le système (S) comme un système différentiel résultant de la discrétisation spatiale de $u = (u_1, \dots, u_N)$, avec pour conventions $u_0 \equiv U_0(t)$ borné par \mathbf{a} sur $[-T, T]$ et $u_{N+1} \equiv 0$.

Ainsi :

$$\begin{aligned} u'_i &= v_i \\ v'_i &= F_i \end{aligned}$$

Si on pose que $\|U_0\|_\infty = 0.001 < \mathbf{a}$, alors le théorème de Cauchy-Lipschitz est vérifié sur $[-T, T]$ et donc l'existence d'une unique solution y est garantie.

Dans la suite, nous allons considérer un cadre plus précis où le milieu est adiabatique. Cette situation sera définie et étudiée dans les sections suivantes. Continuons à travailler sur $[-T, T]$ au voisinage de 0.

Rappelons que

$$F_n(u, t) = F_n(u) = \frac{k\beta h}{\rho_n b} \left(\frac{1}{a + u_n - u_{n-1}} - \frac{1}{a + u_{n+1} - u_n} \right)$$

et en supposant que l'indice du segment liquide où $u_{n(t)}(t)$ est maximal varie, on peut faire l'approximation suivante :

$$\omega_n = \kappa_0 - \kappa_1 n h$$

où κ_0, κ_1 dépendant des caractéristiques du mélange. On suppose que tous les ρ_n sont égales à ρ , que tous les α_n sont égales à α . Ainsi on peut réécrire l'équation d'évolution en prenant

$$F_n(u) = \gamma \frac{\beta b f h}{\rho b} \left(\frac{\kappa_0 - \kappa_1(n-1)h}{a + u_n - u_{n-1}} - \frac{\kappa_0 - \kappa_1 n h}{a + u_{n+1} - u_n} \right)$$

Nous pouvons dire que si :

$$\sup_{2 \leq j \leq N-1} |u_{j+1} - u_j| \leq \frac{a}{2}$$

et

$$\sup_{2 \leq j \leq N-1} |\tilde{u}_{j+1} - \tilde{u}_j| \leq \frac{a}{2}$$

Alors $\forall i \in 2, \dots, N-1$

$$|F_i(u) - F_i(\tilde{u})| \leq 16 \frac{k\beta h}{a^2} \sup_{1 \leq j \leq N} |u_j - \tilde{u}_j|$$

et comme $\sup(|u_N|, |\tilde{u}|) \leq \frac{a}{2}$:

$$\|F(u, t) - F(\tilde{u}, t)\| \leq 16 \frac{k\beta h}{(\mathbf{a} - \|U_0\|_{[0, T]})^2} \|u - \tilde{u}\|$$

On a donc $\|F(u, t) - F(\tilde{u}, t)\| \leq \kappa \cdot \|u - \tilde{u}\|$ avec $\kappa \leq 1$.

On a prouvé que F est une application κ -lipschitzienne.

En appliquant le théorème de Cauchy-Lipschitz, on peut conclure qu'il existe une unique solution au système (S) dans le cadre de nos considérations initiales. \square

Remarque : La solution de (S) est bornée dans la boule $[0, T]$ lorsque les perturbations initiales du systèmes sont nulles ou très petites.

Modifions dans un premier temps l'expression des F_i :

$$\begin{aligned} F_i &= k.\beta.h.\left(\frac{1}{a + u_i - u_{i-1}} - \frac{1}{a + u_{i+1} - u_i}\right) \\ &= k.\beta.h.\frac{u_{i+1} - 2u_i + u_{i-1}}{(a + u_i - u_{i-1})(a + u_{i+1} - u_i)} \end{aligned}$$

On pose : $h=a+b$ et $\alpha = \frac{a}{h} \Leftrightarrow a = \alpha.h$

Donc :

$$\begin{aligned} F_i &= k.\beta.h.\frac{u_{i+1} - 2u_i + u_{i-1}}{(\alpha.h + u_i - u_{i-1})(\alpha.h + u_{i+1} - u_i)} \\ &= k.\beta.h.\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \frac{1}{\left(\alpha + \frac{u_i - u_{i-1}}{h}\right)\left(\alpha + \frac{u_{i+1} - u_i}{h}\right)} \end{aligned}$$

Nous voulons maintenant approximer la dérivée seconde et la dérivée première. Pour cela, nous allons utiliser le schéma d'Euler qui s'obtient par un développement de Taylor.

On effectue un développement en espace avec un pas $h = a+b$ et en temps avec un pas δt . En une dimension, cela donne :

$$u_{i+1} \approx u(x_i + h) = u(x_i) + h.u'(x_i) + \frac{h^2}{2}u''(x_i) + o(h^2)$$

$$u_{i-1} \approx u(x_i - h) = u(x_i) - h.u'(x_i) + \frac{h^2}{2}u''(x_i) + o(h^2)$$

Ainsi :

$$\begin{aligned} u_{i+1} + u_{i-1} &= 2u(x_i) + h^2.u''(x_i) \approx 2u_i + h^2.u''(x_i) + o(h^2) \\ \Leftrightarrow u''(x_i) &\approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} \end{aligned}$$

et on peut étendre ce procédé aux dimensions d'espace et de temps :

$$\frac{u_i^m - u_{i-1}^m}{h} \approx \frac{u_{i+1}^m - u_i^m}{h} \approx u'(X_i)$$

$$\frac{u_i^m - u_i^{m-1}}{\delta t} \approx \frac{u_i^{m+1} - u_i^m}{\delta t} \approx \dot{u}(X_i)$$

Penchons nous sur la résultante des forces discrétisée :

$$F_i = k.\beta.h.u''(x)\left(\frac{1}{(\alpha + u'(x))^2}\right)$$

En utilisant la deuxième équation de (S), on a :

$$\rho.b.\ddot{u} = k.\beta.h.u''\left(\frac{1}{(\alpha + u')^2}\right)$$

$$\Leftrightarrow \frac{\rho.b.\ddot{u}}{h} = k.\beta.u'' \left(\frac{1}{(\alpha + u')^2} \right)$$

$$\Leftrightarrow \rho.(1 - \alpha).\ddot{u} = k.\beta.u'' \left(\frac{1}{(\alpha + u')^2} \right)$$

car $(1 - \alpha) = \frac{b}{h}$

$$\Leftrightarrow \rho.(1 - \alpha).\ddot{u} - k.\beta.u'' \left(\frac{1}{(\alpha + u')^2} \right) = 0$$

Si on suppose maintenant $u' \ll \alpha$, on obtient :

$$\frac{-k\beta}{(\alpha + u')^2} \approx \frac{-k\beta}{\alpha^2}$$

Et l'équation devient :

$$\rho(1 - \alpha)\ddot{u} - \frac{k\beta}{\alpha^2}u'' = 0$$

$$\ddot{u} - \frac{k\beta}{\rho(1 - \alpha)\alpha^2}u'' = 0$$

$$\ddot{u} - c^2u'' = 0$$

La forme de l'équation obtenue est celle d'une équation d'onde de célérité c . C'est-à-dire que c est la vitesse de propagation de l'élément de fluide dans le milieu considéré.

Approche variationnelle

Posons $\Omega \equiv [0,1] \times [0,T]$ et $v(x,t)$ une fonction test.

$$\int_{\Omega} (d_{tt}u - c^2d_{xx}u)v = 0$$

$$\int_{\Omega} d_t u d_t v - c^2 d_x u d_x v = \int_{d\Omega} d_t u v - c^2 d_x u v$$
(3.1)

qu'on notera :

$$a(u, v) \equiv F(v)$$
(3.2)

où a est une fonction bilinéaire continue coercive et F est une fonction linéaire continue. Alors on peut appliquer le théorème de Lax-Milgram qui assure dans ces conditions l'existence et l'unicité de la solution.

Ceci est valable dans un cadre 2D qui dépasse les limites de notre étude mais nous notons qu'il est possible de résoudre l'équation des ondes avec une méthode d'éléments finis en faisant une semi-discrétisation en espace.

Séparation des variables

Soit u une fonction qui dépend de x et t .

On peut donc poser $u(x,t)=f(x)g(t)$ où f et g sont des fonctions de classe C^2 sur $[-T,T]$.

On a :

$$\frac{\partial u^2}{\partial t^2} = f(x)g''(t)$$

Et :

$$\frac{\partial u^2}{\partial x^2} = f''(x)g(t)$$

Donc l'équation des ondes devient :

$$f(x)g''(t) - c^2 f''(x)g(t) = 0$$

D'où :

$$\frac{g''(t)}{g(t)} = c^2 \frac{f''(x)}{f(x)}$$

Le terme de gauche ne dépend que de t et le terme de droite que de x . Les deux termes sont par conséquent constants. Donc, $\exists \lambda \in \mathbb{R}$ tel que :

$$\frac{1}{c^2} \frac{g''(t)}{g(t)} = \lambda = \frac{f''(x)}{f(x)}$$

On obtient le système différentiel :

$$\begin{cases} g''(t) - c^2 \lambda g(t) = 0 \\ f''(x) - \lambda f(x) = 0 \end{cases}$$

- Si $\lambda=0$, on a une solution de la forme :

$$\begin{cases} g(t) = At^2 + Bt + C \\ f(x) = A'x^2 + B'x + C'\lambda f(x) = 0 \end{cases}$$

La solution diverge en $+\infty$. Donc $\lambda \neq 0$

- Si $\lambda > 0$, on a une solution en x de la forme : $f(x) = Ae^{\sqrt{\lambda}x} + Be^{-\sqrt{\lambda}x}$ La solution diverge également en $+\infty$.

- La seule solution possible est donc pour $\lambda < 0$.

On trouve ainsi :

$$\begin{cases} g(t) = A\cos(\sqrt{\lambda}ct) + B\sin(\sqrt{\lambda}ct) \\ f(x) = A'\cos(\sqrt{\lambda}x) + B'\sin(\sqrt{\lambda}x) \end{cases}$$

Finalement, on trouve $u(x,t)$.

Analyse de Fourier

Soit l'équation :

$$\frac{\partial^2 u}{\partial t^2} - \frac{1}{c^2} \frac{\partial^2 u}{\partial x^2} = 0$$

avec les conditions aux limites :

$$\begin{cases} u(0, x) = f(x), u'(0, x) = 0, & x \in [0, 1] \\ u(t, 0) = u(t, 1) = 0, & t \in [0, T] \end{cases}$$

On considère le prolongement de f et u par imparité sur $[-1, 1]$ et par 2-périodicité. On va développer $u(x, t)$ en série de Fourier :

$$u(x, t) = \sum_{k=1}^{+\infty} b_k(u(t, \cdot)) \sin(k\pi x)$$

On pose $b_k(u(t, \cdot)) = \widehat{u}_k(t)$

Or, on sait que :

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t^2} \left(\sum_{k=1}^{+\infty} \widehat{u}_k(t) \sin(k\pi x) \right) \quad (3.3)$$

$$= \sum_{k=1}^{+\infty} \widehat{u}_k''(t) \sin(k\pi x) \quad (3.4)$$

et

$$\frac{\partial^2 u}{\partial x^2} = - \sum_{k=1}^{+\infty} \widehat{u}_k(t) (\pi^2 k^2) \sin(k\pi x)$$

Donc l'équation des ondes devient :

$$\sum_{k=1}^{+\infty} (\widehat{u}_k''(t) + \widehat{u}_k(t) (\pi^2 k^2)) \sin(k\pi x) = 0$$

Comme la famille $(\sin(k\pi x))_{k \geq 1}$ constitue une base hilbertienne, on peut que l'équation est équivalente à :

$$u_k''(t) + \widehat{u}_k(t) (\pi^2 k^2) = 0$$

On a : $\widehat{u}_k(0) = b_k(f)$ de solution :

$$\begin{cases} \widehat{u}_k(t) = A \cos(k\pi t) + B \sin(k\pi t) \\ \widehat{u}_k'(t) = -A \pi k \sin(k\pi t) + B \pi k \cos(k\pi t) \end{cases}$$

Les conditions initiales nous donnent :

$$\begin{cases} \widehat{u}_k(0) = A = b_k(f) \\ \widehat{u}_k'(0) = B = 0 \end{cases}$$

Ainsi : $\widehat{u}_k(t) = b_k(f) \cos(k\pi t)$

A partir d'une situation donnée, nous avons donc fait des approximations qui nous paraissent réalistes pour établir l'équation ci-dessus qui sera plus simple à résoudre numériquement et qui a la forme d'une équation d'onde. Nous devons encore valider cette approche avec les résultats numériques.

II. Température non constante

On va maintenant étudier la cas d'une température affine. On rappelle que l'équation du mouvement pour l'élément de liquide centré en x_i est :

$$\rho b \ddot{u}_i = -(P_i^d - P_i^g) \quad (3.5)$$

On suppose ici que les tailles des bulles et des masses de liquide définissent une échelle microscopique, alors que la température T varie à l'échelle macroscopique.

Cette hypothèse permet de supposer que la température est constante dans la bulle entre la i -ème et la $i+1$ -ème masse de liquide :

$$T(x) = T(0) + \frac{1}{2}T'(x_i + x_{i+1}); \quad T' = \frac{dT}{dx} = cst$$

On en déduit la pression dans la bulle :

$$P_i^d = k\beta h \frac{T_i + T_{i+1} + T'(u_i + u_{i+1})}{2(a + u_{i+1} - u_i)}$$

où $T_i = T(X_i) = T'h_i$, $\forall i = 0, \dots, N + 1$

En réinjectant P_i^d et P_i^g dans l'équation, on obtient l'équation :

$$\rho b \ddot{u}_i = -k\beta h \left(\frac{T_i + T_{i+1} + T'(u_i + u_{i+1})}{2(a + u_{i+1} - u_i)} - \frac{T_{i-1} + T_i + T'(u_{i-1} + u_i)}{2(a + u_i - u_{i-1})} \right)$$

En utilisant les approximations de la dérivée première et de la dérivée seconde :

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

et

$$u'(x_i) \approx \frac{u_i - u_{i-1}}{h} \approx \frac{u_{i+1} - u_i}{h}$$

on obtient ainsi une équation aux dérivées partielles conduisant au profil de la variable $u(t, x_i) = u_i(t)$ de la forme :

$$\ddot{u} = \frac{k\beta}{2\rho(1-\alpha)} \frac{(2Tu'' - 2\alpha T' - 2T'u' + 2T'uu'' + 2T'\alpha u' - 2T'u'^2)}{(\alpha + u')^2}$$

toujours en posant $1 - \alpha = \frac{b}{h}$

De même, en supposant que u et u' sont petits devant α , on se ramène finalement à l'équation :

$$\ddot{u} = \frac{k\beta}{\rho\alpha^2(1-\alpha)} (Tu'' - 2\alpha T') = \frac{k\beta T(x)}{\rho\alpha^2(1-\alpha)} u'' - \frac{2k\beta\alpha T'(x)}{\rho\alpha^2(1-\alpha)} \quad (3.6)$$

Ainsi, en posant $c^2(x) = \frac{k\beta T(x)}{\rho\alpha^2(1-\alpha)}$, on obtient l'équation :

$$\ddot{u} - c^2(x)u'' = -\frac{2k\beta\alpha T'(x)}{\rho\alpha^2(1-\alpha)}$$

On retrouve une nouvelle fois une équation d'onde.

III. Juxtaposition de deux milieux d'indices différents

On va maintenant étudier la juxtaposition de deux milieux liquides différents.

On suppose que le milieu possède $2n+2$ bouchons qui se trouvent centrés aux points $X_i = ih$, $0 \leq i \leq 2n+1$ à l'équilibre. On pose $X_{2n+1} = L$ et on supposera que les bouchons numérotés de 0 à n sont de densité ρ_1 alors que ceux numérotés de $n+1$ à $2n$ sont de densité ρ_2 .

En revenant à l'équation du principe fondamental de la dynamique, on écrit l'équation du mouvement pour le i -ème bouchon de densité ρ_1 et pour le j -ème bouchon de densité ρ_2 :

$$\rho_1 b \ddot{u}_i = k \cdot \beta \cdot h \cdot \frac{1}{(a + u_i - u_{i-1})(a + u_{i+1} - u_i)}$$

et

$$\rho_2 b \ddot{u}_j = k \cdot \beta \cdot h \cdot \frac{1}{(a + u_j - u_{j-1})(a + u_{j+1} - u_j)}$$

On suppose de plus que $u_k(t) = u(t, x_k)$ où u est de classe C^2 selon x sur $[0, L/2]$ et sur $[L/2, L]$ et que l'on a $|u'| \ll \alpha$.

En reprenant l'équation d'onde, on obtient :

$$\ddot{u} - \frac{k\beta}{\rho_1(1-\alpha)\alpha^2} u'' = 0, \quad 0 < x < L/2$$

et

$$\ddot{u} - \frac{k\beta}{\rho_2(1-\alpha)\alpha^2} u'' = 0, \quad L/2 < x < L$$

avec les deux vitesses de propagation dans les milieux situés respectivement à gauche et à droite de l'interface $x=L/2$:

$$c_1 = \sqrt{\frac{k\beta}{\rho_1\alpha^2(1-\alpha)}}; \quad c_2 = \sqrt{\frac{k\beta}{\rho_2\alpha^2(1-\alpha)}}$$

IV. Considérations annexes

a. Ajout d'un terme de dissipation

Nous allons étudier l'équation avec l'ajout d'un terme de dissipation. En effet, il est judicieux de prendre en compte le fait que le système perd de l'énergie au cours du temps. En mécanique, ce phénomène provient de la considération des forces de frottements, et on peut ici considérer un phénomène similaire quand les bulles de gaz se déplacent dans le liquide.

Nous proposons l'équation suivante :

$$\begin{cases} \frac{du}{dt} = v \\ \frac{dv}{dt} = F(u) - \frac{\kappa v}{\rho b} \end{cases}$$

avec κ un paramètre du terme dissipatif positif

b. Dans un milieu adiabatique

Nous pouvons nous intéresser à l'équation d'évolution dans un milieu homogène non plus isotherme mais adiabatique, c'est-à-dire sans échange de chaleur avec le milieu extérieur. Cette considération est d'autant plus fine que le passage de l'onde à travers le milieu se traduit par une augmentation locale de la température. Nous avons pris en compte une évolution affine de la température en première approximation. Cela a pour influence de faire varier k linéairement en espace :

$$k = k_{init} - \kappa_1 n h$$

où k varie de k_{init} jusque 1.

V. Retour sur l'équation des ondes

Nous allons revenir sur l'établissement de l'équation des ondes en 1 dimension, qu'on peut aussi appeler équation de d'Alembert car c'est lui qui l'établit en premier dans la résolution du problème des cordes vibrantes. On place le contexte dans un problème d'acoustique.

En considérant l'écoulement d'un fluide, on peut raisonnablement émettre l'hypothèse qu'aucune particule n'est perdue (ou un nombre très certainement négligeable). De facto la masse est conservée. Si maintenant on néglige les forces de frottements fluides, la gravitation, on peut considérer que seule les forces de pressions sont exercées sur un volume infinitésimal du fluide et s'intéresser à l'évolution de la masse pendant un intervalle de temps infinitésimal. D'après le principe fondamental de la dynamique appliqué à l'élément de fluide :

$$\frac{\partial \rho(x, t)}{\partial t} + \frac{\partial}{\partial x} [\rho(x, t)v(x, t)] = 0$$

A l'échelle macroscopique, l'équation globale s'écrit

$$div(\mu_T v) + \frac{\partial \mu_T}{\partial t} = 0$$

avec v la vitesse d'une particule de fluide et μ_i la masse volumique du fluide.

En posant $\mu = \mu_T - \mu_0$:

$$div((\mu_0 + \mu)v) + \frac{\partial \mu}{\partial t} = 0$$

$$\mu_0 div(v) + \mu div(v) + v grad(\mu) + \frac{\partial \mu}{\partial t} = 0$$

Par approximation linéaire du premier ordre, on ne garde pas $\mu div(v)$ et $v.grad(\mu)$ et on obtient l'équation suivante :

$$\mu_0 div(v) + \frac{\partial \mu}{\partial t} = 0$$

appelée équation de conservation de la masse.

Comme il n'y a pas de viscosité, l'équation du mouvement du fluide est :

$$\mu_T \left(\frac{\partial v}{\partial t} + (v.grad)v \right) = -grad(P) + f_V$$

avec f_V la force volumique statique égale à $grad(P_0)$ et P la pression du fluide.

Après linéarisation, l'équation du mouvement devient :

$$\mu_0 \frac{\partial v}{\partial t} = -grad(p)$$

aussi appelée équation d'Euler.

Or, connaissant la relation entre la surpression et la variation de la masse volumique :

$$\mu \approx \mu_0 \chi_S p$$

On peut alors réécrire les équations, et on obtient le système :

$$\begin{aligned}\frac{\partial v}{\partial t} &= -\frac{1}{\mu_0} \text{grad}(p) \\ \frac{\partial \rho}{\partial t} &= -\frac{1}{\chi_S} \text{div}(v)\end{aligned}$$

A partir d'ici, on peut choisir de développer selon la surpression p ou selon la vitesse v :

Surpression p

$$\begin{aligned}\frac{\partial v}{\partial t} &= -\frac{1}{\mu_0} \text{grad}(p) \\ \Leftrightarrow \text{div}\left(\frac{\partial v}{\partial t}\right) &= -\frac{1}{\mu_0} \text{div}(\text{grad}(p)) \\ \Leftrightarrow \frac{\partial}{\partial t}\left(-\chi_S \frac{\partial p}{\partial t}\right) &= -\frac{1}{\mu_0} \Delta p \Leftrightarrow \Delta p - \mu_0 \chi_S \frac{\partial^2 p}{\partial t^2} = 0\end{aligned}$$

On obtient alors l'équation de propagation de d'Alembert, une équation des ondes :

$$\Delta p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0$$

avec $c = \frac{1}{\sqrt{\mu_0 \chi_S}}$

Vitesse v

$$\text{div}(v) = -\chi_S \frac{\partial \rho}{\partial t}$$

En prenant le gradient et en dérivant par rapport au temps l'équation suivante :

$$\mu_0 \frac{\partial v}{\partial t} = -\text{grad}(p)$$

On obtient :

$$\begin{aligned}-\text{grad}\left(\frac{\partial p}{\partial t}\right) &= \frac{1}{\chi_S} \text{grad}(\text{div}(v)) \\ &= \frac{1}{\chi_S} (\text{rot}(\text{rot}(v)) + \Delta v) \\ &= \mu_0 \frac{\partial^2 v}{\partial t^2}\end{aligned}$$

Faisons l'hypothèse que l'on est dans un cas où $\text{rot}(v) = 0$ (le rotationnelle de la vitesse est nul car la vitesse est portée par une seule composante). On obtient ainsi l'équation suivante, qui est aussi une équation de d'Alembert :

$$\Delta v - \frac{1}{c^2} \frac{\partial^2 v}{\partial t^2} = 0$$

avec $c = \frac{1}{\sqrt{\mu_0 \chi_S}}$

Chapitre 4

Simulation numérique avec la méthode d'Euler

I. Cas de référence

Dans cette partie, on se propose de résoudre numériquement le système (S) pour modéliser le mouvement des bouchons.

Nous choisissons d'utiliser la méthode des différences finies. Pour cela, nous allons discrétiser le système en temps et en espace, et nous allons utiliser les schémas d'Euler.

L'étude d'un cas de référence sera effectuée au préalable, puis nous tenterons d'appliquer nos méthodes à des situations où les conditions initiales et celles du milieu ont été modifiées.

Nous décidons donc premièrement de construire notre simulation sur le cas où le milieu est homogène, la température constante et avec des conditions initiales nulles.

On introduit une discrétisation en temps de $[0, T]$ en $M+1$ points :

$$t_0 = 0 < t_1 = \delta t < \dots < t_M = M\delta t = T$$

On note u_n^m l'approximation de u_n et v_n^m l'approximation de v_n au temps t_m .

On reprend le schéma (S) :

$$\begin{cases} \frac{du}{dt} = v \\ \rho \cdot b \cdot \frac{dv}{dt} = F(u) \end{cases}$$

A l'aide du schéma d'Euler en temps, on obtient :

$$u' \approx \frac{u_n^m - u_n^{m-1}}{\delta t}$$

$$v' \approx \frac{v_n^{m+1} - v_n^m}{\delta t}$$

Donc la discrétisation en temps et en espace du schéma (S) donne :

$$\begin{cases} \frac{u_n^m - u_n^{m-1}}{\delta t} = v_n^m \\ \rho b \frac{v_n^{m+1} - v_n^m}{\delta t} = k\beta h \left(\frac{1}{a+u_n^m - u_{n-1}^m} - \frac{1}{a+u_{n+1}^m - u_n^m} \right) \end{cases}$$

En réinjectant v_n^m , on a :

$$\rho b \left(\frac{\frac{u_n^{m+1} - u_n^m}{\delta t} - \frac{u_n^m - u_n^{m-1}}{\delta t}}{\delta t} \right) = k\beta h \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right)$$

D'où :

$$u_n^{m+1} - 2u_n^m + u_n^{m-1} = \frac{k\beta h \delta t^2}{\rho b} \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right)$$

Donc :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \frac{k\beta h \delta t^2}{\rho b} \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right)$$

On obtient ainsi :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \delta t^2 \mu \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right) \quad (4.1)$$

avec $\mu = \frac{k\beta h}{\rho b}$

Couplons ce schéma avec les conditions aux limites suivantes :

$$\begin{aligned} u_0^m &= \lambda(t), u_N^m = 0, & \forall m = 0, \dots, M \\ u_n^0 &= U_n^0, u_n^{-1} = U_n^{-1}, & \forall n = 0, \dots, N \end{aligned}$$

Nous pouvons donc simuler le mouvement des petits segments S_i de liquide : on calcule les u_n^m , écart entre la position du liquide à l'équilibre et sa position au temps t_m .

II. Changement d'homogénéité

Durant les simulations nous serons parfois amenés à considérer un changement de milieu. Nous avons vu comment se comporte théoriquement l'onde au travers d'une juxtaposition de milieux différents. On peut implémenter une telle situation dans notre algorithme : nous considérons une masse volumique ρ globale qui est celle de l'eau ($\rho_{eau} = 1.0$). Dans le cas d'une juxtaposition de milieux différents, on peut par exemple prendre le cas du brome liquide et de l'eau ($\rho_{brome} = 3.0$) et alors nous multiplions par 3 la valeur de ρ tant que l'on se trouve dans le brome liquide. Concrètement dans cet exemple, on considère plutôt $\mu = \frac{kh}{\rho b}$ qu'on divisera par 3 seulement durant la moitié des itérations en espace (puisque l'on considère deux milieux qui se séparent au centre). Dans le cas où le milieu est globalement homogène mais comporte un saut en son centre, nous allons diviser μ seulement à la $\frac{N}{2}$ -ème itération en espace (parmi N itérations en espace). De ce fait, on peut facilement simuler un enchaînement d'autant de milieux qu'on veut en les caractérisant chacun par ρ_i leur masse volumique.

III. Impulsion initiale

Il est intéressant d'étudier une situation où l'on donne une vitesse initiale au déplacement des bouchons. En effet, si nous considérons l'équation des ondes approchée par notre équation d'évolution, la vitesse $c(x)$ dépend de plusieurs paramètres physiques comme la masse volumique, et la vitesse est modifiée si l'on modifie ces paramètres. Nous pouvons également agir sur la vitesse de l'onde d'une différente manière : en modifiant la vitesse initiale.

Pour rappel, nous utilisons un schéma d'Euler explicite pour approcher la dérivée seconde dans l'équation d'évolution. Les termes u_n^{m+1} , u_n^m et u_n^{m-1} impliquent que lorsque nous calculons le vecteur solution à une itération de temps donnée, nous avons besoin des anciens vecteurs solutions aux deux dernières itérations de temps. Nous devons donc définir les deux premiers vecteurs solutions d'une autre manière qu'avec le schéma.

Nous définissons donc les vecteurs u^{-1} et u^0 de telle sorte à ce que nous puissions choisir la vitesse initiale de la solution. En effet, si l'on considère la vitesse initiale approchée selon un schéma d'Euler explicite :

$$v_n^0 = \frac{u_n^1 - u_n^0}{\delta t}$$

Les conditions initiales sont définies de la manière suivante :

$$U(0, :) = U_0 = u_n^0 \quad \forall n \in 0, \dots, N$$

$$U(N+1, :) = U_{N+1} = u_n^{N+1} \quad \forall n \in 0, \dots, N$$

et

$$U(:, 0) = V_0 = u_0^m \quad \forall m \in 0, \dots, M$$

$$U(:, -1) = V_{-1} = u_{-1}^m \quad \forall m \in 0, \dots, M$$

On peut aussi donner les conditions initiales sous la forme $(u(0), v(0))$ tel que :

$$u_n(0) = V_0(n)$$

et

$$v_n(0) = \frac{V_0(n) - V_{-1}(n)}{\delta t}$$

Sauf indication contraire, nous considérons la position initiale :

$$U_0 = \frac{(h, \dots, h)}{10}$$

de telle sorte à ce que $\|U_0\|_\infty = 0.001 < a$

IV. Température non constante

Nous allons maintenant nous intéresser au cas où la température n'est plus constante mais affine. On prend une température de la forme : $T(x) = ax+b$ avec les conditions initiales que nous donnerons par la suite.

On reprend l'équation du système pour une température non constante :

$$\ddot{u} - c^2(x)u'' = -\frac{2k\beta\alpha T'(x)}{\rho\alpha^2(1-\alpha)}$$

En discrétisant en temps et en espace, on obtient :

$$\frac{u_n^{m+1} - 2u_n^m + u_n^{m-1}}{\delta t^2} = \frac{k\beta}{\rho\alpha^2(1-\alpha)} \left(T \frac{u_{n+1}^m - 2u_n^m - u_{n-1}^m}{h^2} - 2\alpha T' \right)$$

D'où :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \frac{k\beta\delta t^2}{\rho\alpha^2(1-\alpha)} \left(T \frac{u_{n+1}^m - 2u_n^m - u_{n-1}^m}{h^2} - 2\alpha T' \right)$$

Ainsi :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \frac{k\beta\delta t^2}{\rho\alpha(1-\alpha)} \left(T \frac{u_{n+1}^m - 2u_n^m - u_{n-1}^m}{\alpha h^2} - 2T' \right) \quad (4.2)$$

avec T la discrétisation de la température.

Pour correctement prendre en compte la température, on va utiliser la fonction *temperature*, présente dans le code, qui prend en compte n, h, a, et b et qui permet de renvoyer la courbe de température discrétisée, adaptée au schéma de notre équation :

$$T(x_n) = a.x_n + b = a.nh + b$$

V. Phénomène de dissipation

Dans cette section, nous allons étudier le système (S) en milieu homogène en y ajoutant un terme de dissipation.

Le système différentiel (S) s'écrit alors sous la forme :

$$\begin{cases} \frac{du}{dt} = v \\ \rho \cdot b \cdot \frac{dv}{dt} = F(u) - \frac{\kappa}{\rho b} v \end{cases}$$

où κ est le terme de dissipation.

En discrétisant à nouveau ce système, on obtient :

$$\begin{cases} \frac{u_n^m - u_n^{m-1}}{\delta t} = v_n^m \\ \frac{v_n^{m+1} - v_n^m}{\delta t} = \frac{k\beta h}{\rho b} \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right) - \frac{\kappa}{\rho b} v_n^m \end{cases}$$

En réinjectant v_n^m , on a :

$$\frac{\frac{u_n^{m+1} - u_n^m}{\delta t} - \frac{u_n^m - u_n^{m-1}}{\delta t}}{\delta t} = \frac{k\beta h}{\rho b} \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right) - \frac{\kappa}{\rho b} \frac{u_n^m - u_n^{m-1}}{\delta t}$$

D'où :

$$u_n^{m+1} - 2u_n^m + u_n^{m-1} = \frac{\delta t^2}{\rho b} \left(k\beta h \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right) - \kappa \delta t \left(\frac{u_n^m - u_n^{m-1}}{\delta t} \right) \right)$$

Ainsi :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \delta t^2 \mu \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right) - \frac{\delta t \kappa}{\rho b} (u_n^m - u_n^{m-1})$$

avec $\mu = \frac{k\beta h}{\rho b}$

Chapitre 5

Interprétations

I. Température constante

On rappelle que l'équation discrétisée du système (S) dans le cas où la température est constante est schématisée comme ceci :

$$u_n^{m+1} = 2u_n^m + u_n^{m-1} + \delta t^2 \mu \left(\frac{1}{a + u_n^m - u_{n-1}^m} - \frac{1}{a + u_{n+1}^m - u_n^m} \right)$$

avec $\mu = \frac{k\beta h}{\rho b}$

Dans toute cette partie, nous implémenterons le schéma avec les paramètres suivants :

$$\begin{array}{cccccc} \rho = 1 & N = 99 & L = 1.0 & \alpha = 0.2 & \beta = 1 & k = 1.0 \\ \delta t = 10^{-3} & M = 250 & h = 0.01 & a = 0.002 & b = 0.008 & \end{array}$$

a. Conditions initiales nulles

Milieu homogène

Nous nous intéressons au cas le plus simple où le milieu est homogène, la perturbation initiale est constante et on considère aussi un équilibre statique initiale.

Notons que u^m représente $u(x, m\delta t)$ sur les figures.

Nous représentons les solutions aux temps caractéristiques $T = 0.046$ s, $T = 0.133$ s, $T = 0.168$ s et $T = 0.227$ s, c'est à dire : u^{46} , u^{133} , u^{168} et u^{227} .

On observe sur les différentes figures un comportement globalement ondulatoire. L'onde numérique est générée par la perturbation initiale (figure 3) et traverse le milieu, notre segment de longueur L (figure 4), avant de se heurter à la paroi et d'être réfléchi (figure 5). L'onde résultante a une vitesse réduite et son amplitude est légèrement amoindrie à l'impact (figure 6). Ce comportement conforte la comparaison du mouvement des bouchons liquides du milieu avec le mouvement d'une onde.

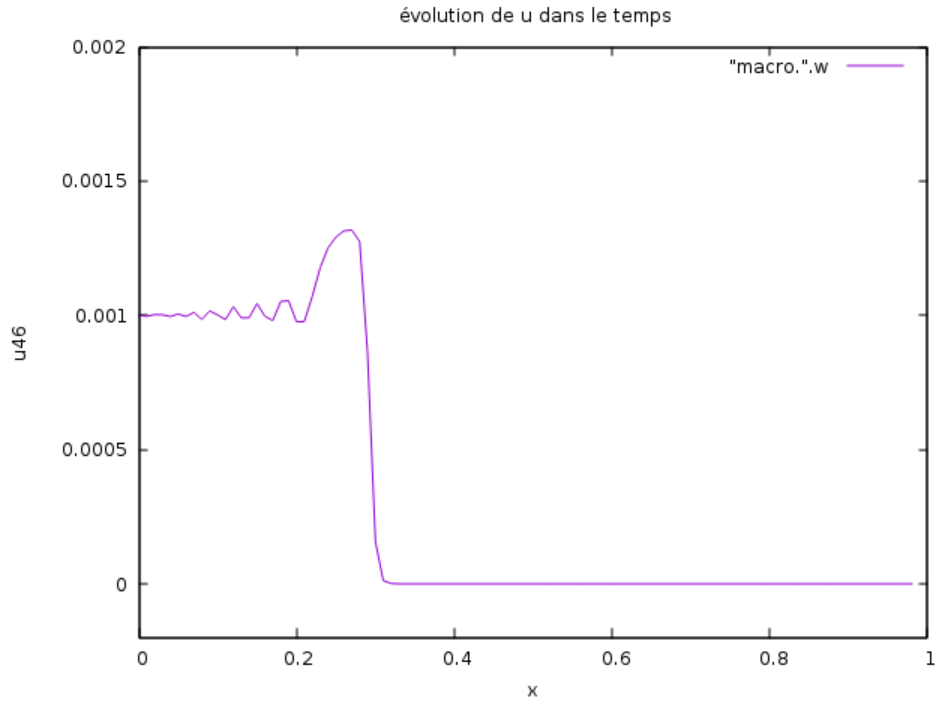


FIGURE 5.3

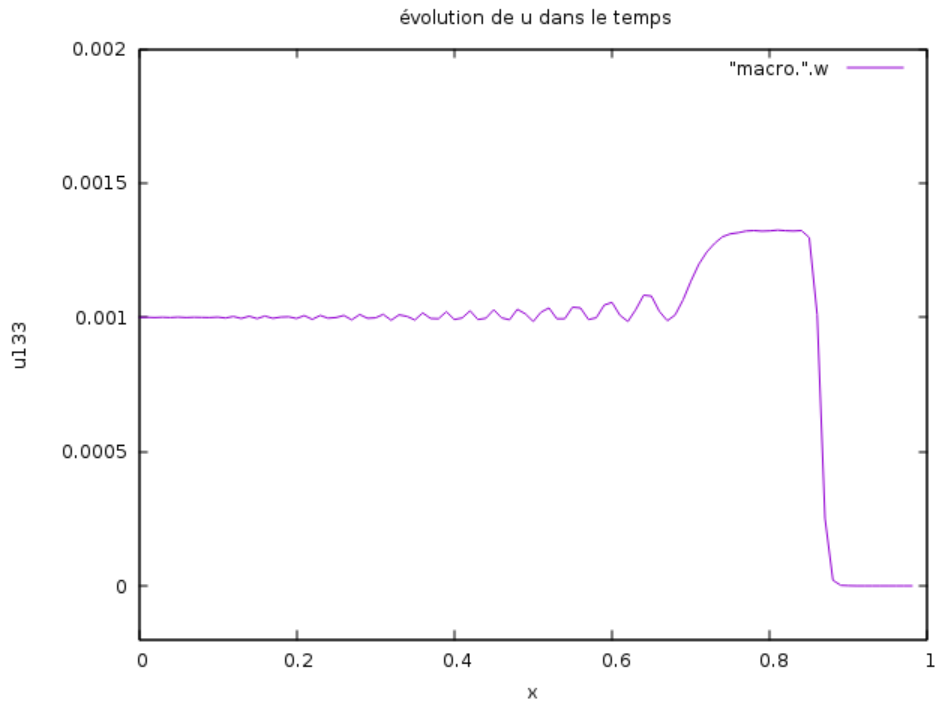


FIGURE 5.4

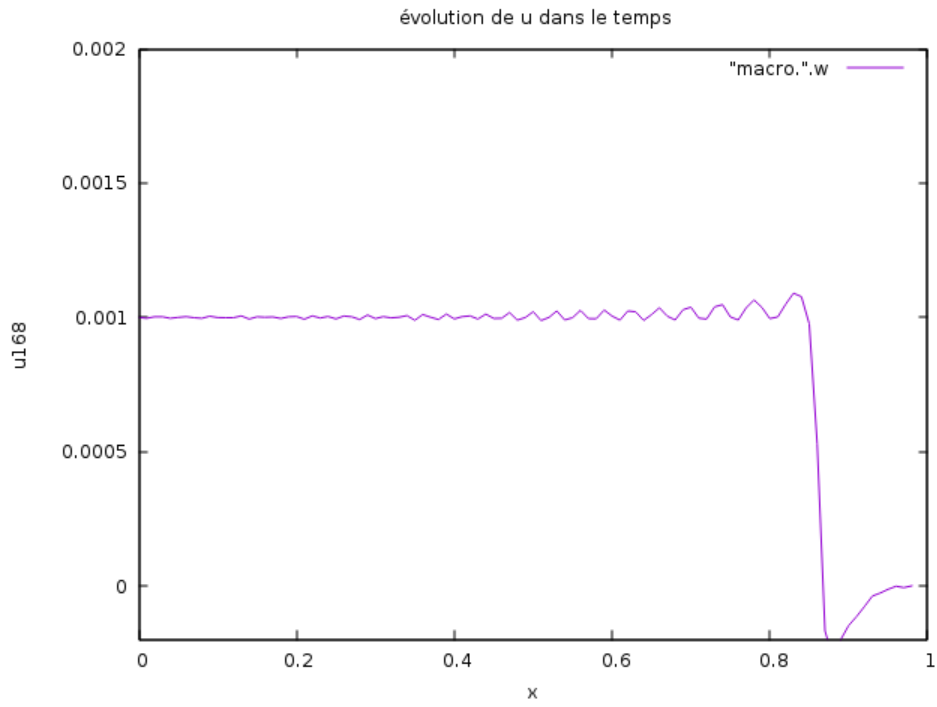


FIGURE 5.5

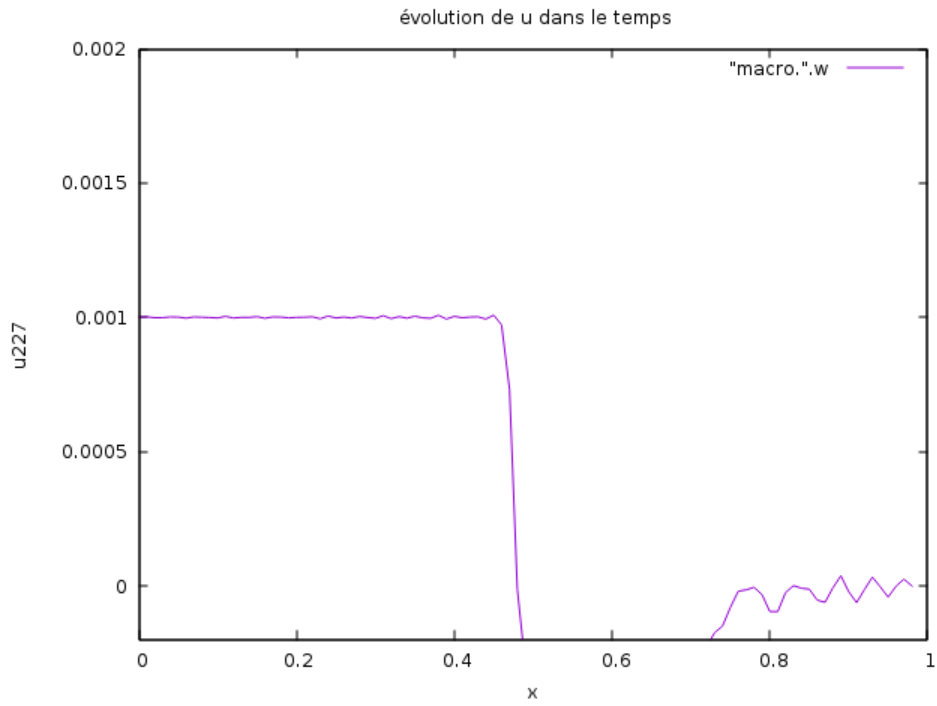


FIGURE 5.6

Remarque : Si on prend $\rho=2$, on observe graphiquement en comparant les figures 5 et 7 qui sont toutes les deux prises au temps $T=0.168s$ que le phénomène ondulatoire met deux fois plus de temps à se propager.

En effet, comme la célérité c vaut $c^2 = \frac{k\beta T}{\rho\alpha^2(1-\alpha)}$, on s'attendait à ce que la célérité diminue lorsqu'on augmente ρ .

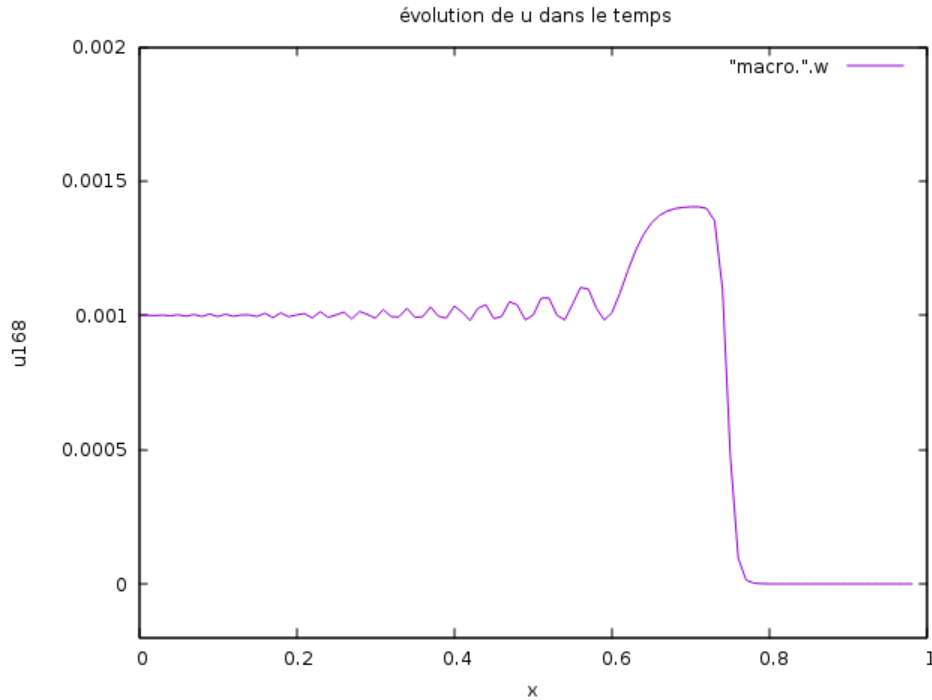


FIGURE 5.7

Milieu non homogène

On va maintenant tester la validité de ce schéma pour un milieu qui a une température constante mais qui n'est plus homogène. On va étudier si le phénomène de propagation vu précédemment permet de mettre en évidence graphiquement si il se produit un changement de milieu.

1er cas :

Dans un premier temps, on modélise une situation où l'un des segments du milieu de masse linéique ρ est remplacé par un segment de masse linéique double. Concrètement, cette modification du milieu consiste à modifier la masse volumique ρ à la moitié du parcours seulement. Comme précédemment, nous représentons u^{82} , u^{147} , u^{169} et u^{190} .

On observe toujours le phénomène ondulatoire mais on remarque qu'à mi-chemin du milieu (figures 8-9), il y a une déformation de la courbe et une partie de l'onde est réfléchiée vers sa source. Comme précédemment, la deuxième partie de l'onde se heurte contre la paroi pour être réfléchiée vers sa source (figures 10-11).

La modification algorithmique du milieu se retrouve graphiquement par une réflexion partielle de l'onde au niveau de la déformation. Ce phénomène est physiquement plausible et même attendu

puisqu'on sait que la célérité d'une onde dépend du milieu de propagation.

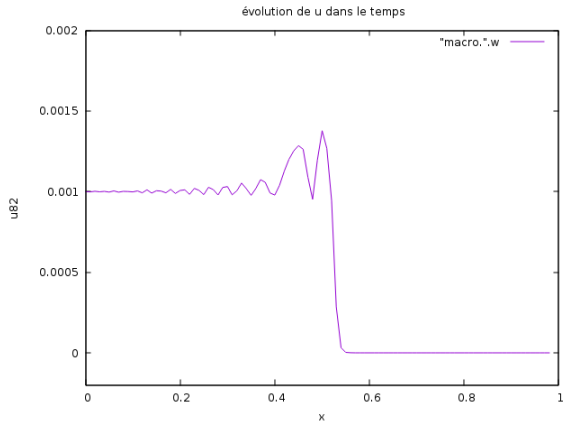


FIGURE 5.8

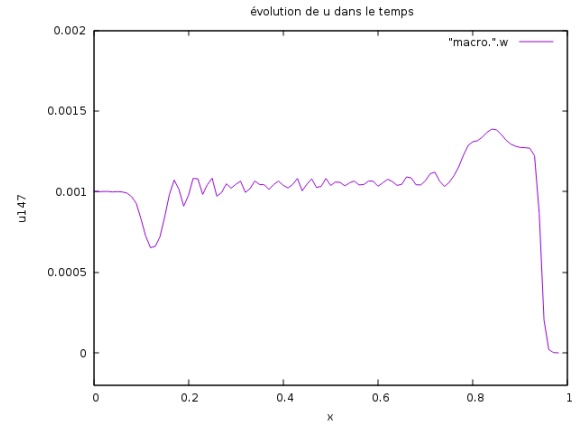


FIGURE 5.9

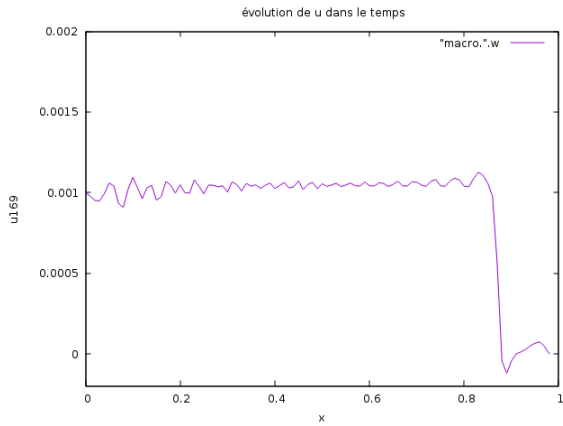


FIGURE 5.10

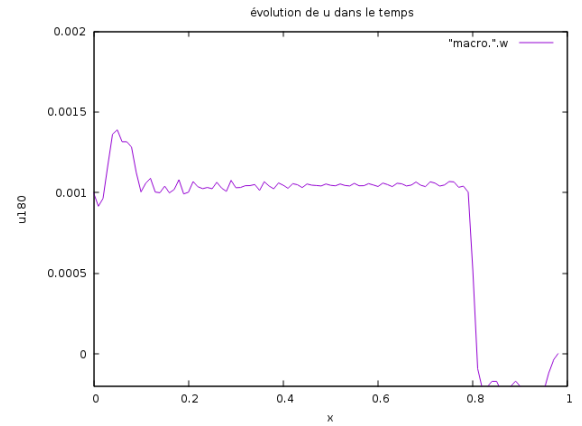
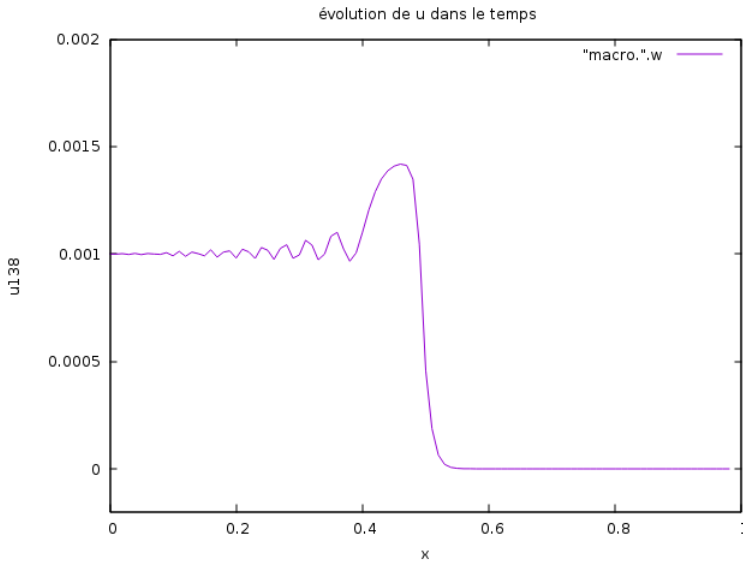


FIGURE 5.11

2ème cas :

Toujours dans l'esprit de tester la validité de ce procédé dans un milieu non homogène, celui-ci est séparé en deux parties : un milieu 1 de masse volumique $\rho_1=3$ et un milieu 2 de masse volumique $\rho_2=1$. Dans un espace discrétisé de 101 éléments, les 51 premiers ont une masse volumique triple des 50 derniers. On a représenté les instants u^{138} , u^{148} , u^{184} et u^{211} .



On s'attend à ce que l'onde soit fortement altérée dès son passage à la frontière entre les deux milieux. La forme de l'impulsion est la même que dans les autres cas précédemment puisque la vitesse initiale est toujours nulle.

Au centre on observe un saut plus important qu'auparavant (figure 12) qui montre la frontière entre les deux milieux et le passage de l'un vers l'autre. A partir de ce moment là, on observe deux phénomènes sur le graphique : une partie de la courbe continue jusqu'à la limite tandis que l'autre partie continue en sens inverse.

Comme précédemment, cela met en évidence un phénomène ondulatoire : une partie de l'onde est renvoyée vers la source tandis que l'autre partie continue de progresser vers la paroi (figure 13-14).

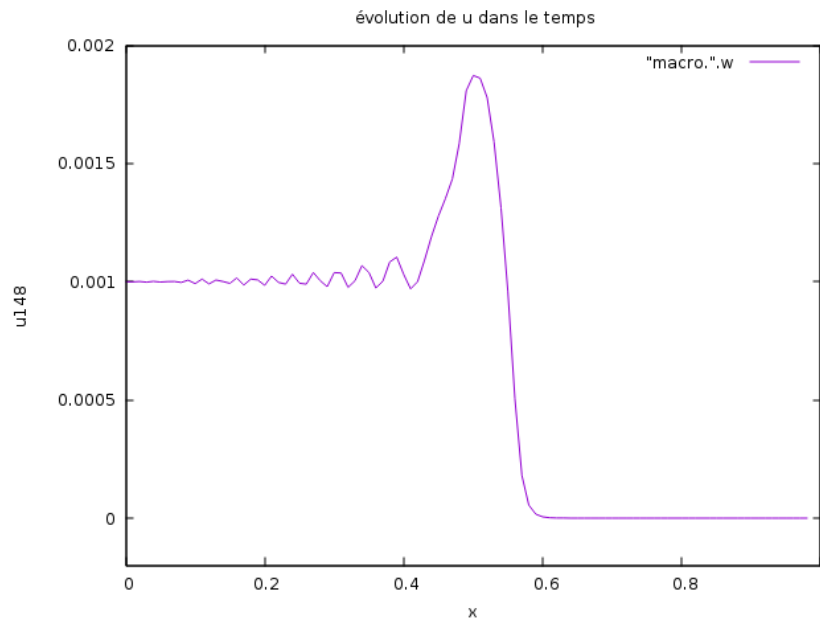


FIGURE 5.12

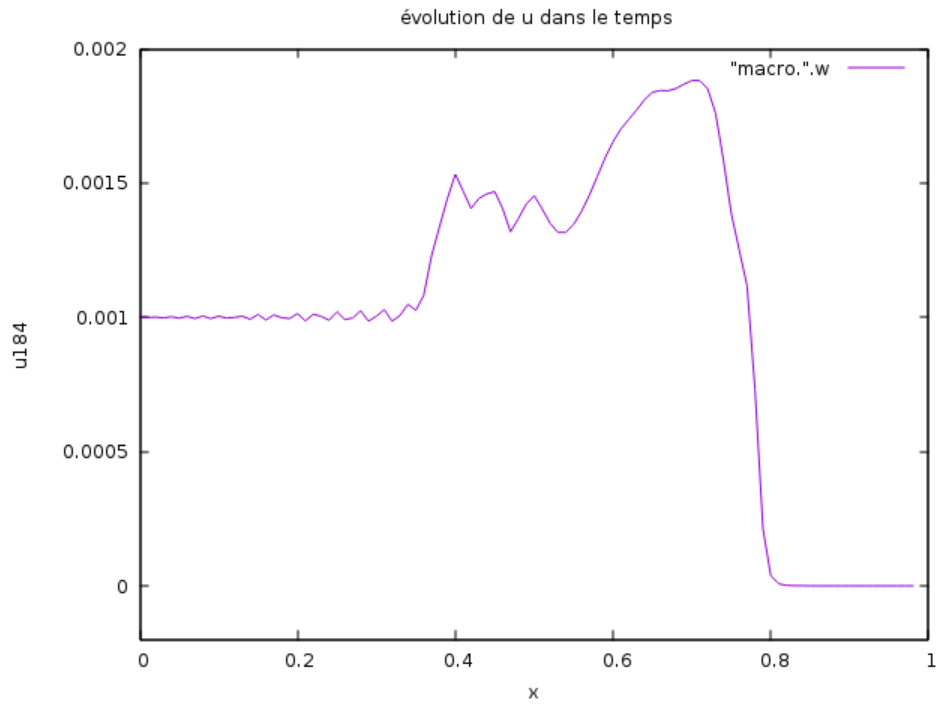


FIGURE 5.13

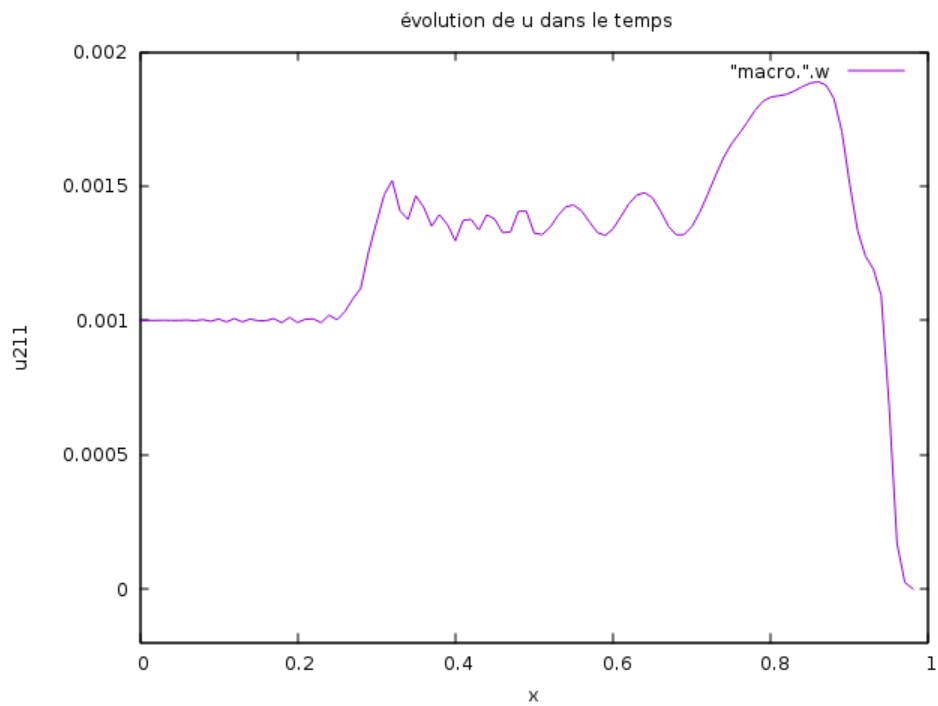


FIGURE 5.14

b. Conditions initiales non nulles

Nous allons maintenant nous pencher sur la modification de la vitesse initiale et de la répercussion d'un tel changement dans le comportement du phénomène ondulatoire. Pour cela, on va reprendre notre algorithme et y ajouter une pente initiale non nulle.

Reprenons notre modèle de référence et changeons la vitesse initiale de déplacement des bouchons. Notons que le schéma de l'équation discrétisée fait intervenir des dérivées secondes en temps et en espace. Il est donc nécessaire de définir les deux premières itérations en temps manuellement avant de faire intervenir le schéma dans l'algorithme. Depuis le début, nous avons considéré une situation où la vitesse initiale était nulle et nous avons donc initialisées à 0 les deux premières itérations en temps de u . Nous allons modifier l'algorithme pour pouvoir considérer une impulsion et une position initiales non nulles. Pour cela nous allons revenir au schéma d'Euler, approchant la première dérivée en temps et en espace par différences finies :

$$\frac{u_i^m - u_i^{m-1}}{\delta t} \approx \frac{u_i^{m+1} - u_i^m}{\delta t} \approx \dot{u}(X_i)$$

On peut donc approcher la vitesse initiale par un schéma explicite :

$$v_i^0 \approx v(X_0) \approx \dot{u}(X_i) \approx \frac{u_i^1 - u_i^0}{\delta t}$$

qu'on écrit plus simplement :

$$v_i^0 = v(X_0) = \dot{u}(X_0) \approx \frac{u_i^1 - u_i^0}{\delta t}$$

Ainsi, on peut écrire u^1 en considérant u^0 et v^0 :

$$u^1 = u^0 + \delta t.v^0$$

Dans la suite, on va principalement traiter un décalage initial de $u_i^0 = 0$ en $\tilde{u}_i^0 = 10^{-4} (1, \dots, 1)^T$ et $v_i^0 = 0$ en $\tilde{v}_i^0 = \pm 10^{-2} (1, \dots, 1)^T$. On a représenté les instants u^{117} , u^{164} , u^{176} et u^{238} .

On observe à nouveau un phénomène de propagation (Figure 15). La pente initiale que l'on vient de soumettre au système est bien visible graphiquement : en effet, la fluctuation u part toujours de 1.10^{-3} en u_0^m , en revanche elle connaît une légère croissance jusqu'à un certain point. Quand l'onde s'abat sur la paroi et repart dans le sens opposé, sa vitesse s'inverse et croît à nouveau (en valeur absolue) jusqu'à ce que u atteigne -1.10^{-3} . Finalement, la solution décroît et réagit de la même façon que précédemment.

On observe donc qu'une vitesse initiale non nulle permet à l'onde d'atteindre une amplitude plus élevée pendant un instant et le fait que sa vitesse diminue par la suite est due à la nature physique du milieu, caractérisé par ρ . On pourra donc s'intéresser au comportement de l'onde dans un milieu non-homogène.

On va maintenant observer le comportement de l'onde que l'on a soumise à une vitesse initiale de pente négative. Pour cela, on a représenté les instants u^{119} , u^{159} , u^{175} et u^{227} .

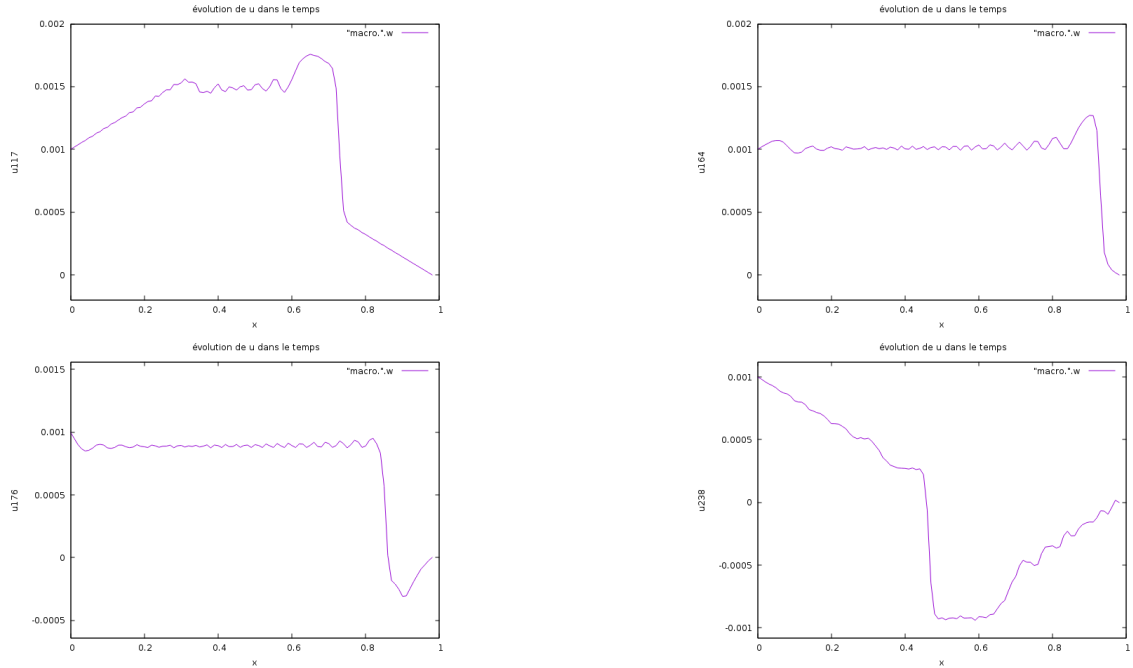


FIGURE 5.15 – snapshots de $u(t, \cdot)$ aux temps $t=0.117$, $t=0.164$, $t=0.176$ et $t=0.238$ avec une pente initiale positive

Nous remarquons graphiquement (figure 16) que le comportement est très similaire à ce que nous avons traité précédemment : en effet, le seul changement notable se situe dans l'ordre d'atteinte des extremas par u .

Les captures prises dans le cas d'une pente positive ou d'une pente négative étant dans les mêmes périodes temporelles, on peut comparer les étapes par lesquelles passent u suivant sa pente initiale : Si nous considérons une période relative de 0.16 secondes (période observée graphiquement et à évoquée seulement dans un but conceptuel) durant laquelle l'onde traverse le milieu dans les deux cas, nous pouvons dire que les deux ondes ont le même comportement à une période près. De plus, si on augmente le nombre d'itérations en temps et qu'on observe le comportement ondulatoire sur une plus grande durée, et en dépit du fait que l'utilisation d'un schéma d'Euler nous force à prendre ce résultat avec précaution, on peut voir que la vitesse de propagation diminue et que l'amplitude semble diminuer légèrement tandis que le pic d'amplitude s'aplatit.

Nous pouvons aussi observer le comportement de u quand le milieu est non-homogène, plus précisément quand le milieu présente une discontinuité en son centre (figure 17). Nous avons observé un saut au niveau de cette discontinuité quand les conditions initiales étaient nulles et ce saut est à nouveau visible quand il y a une vitesse initiale non nulle. Cependant ce phénomène de saut est étouffé dans les oscillations durant la propagation de l'onde. A noter que ce saut se produit bien à chaque fois que l'onde traverse la perturbation graphique, qu'elle vienne de droite ou de gauche.

Nous allons maintenant nous intéresser à la validité de la comparaison entre l'équation d'évolution qui modélise le comportement des écarts de bouchons et l'équation des ondes en 1D. Pour ce faire, nous allons implémenter le schéma de l'équation des ondes, puis nous allons l'intégrer à notre algorithme pour pouvoir résoudre nos deux équations sur la même discrétisation de temps et

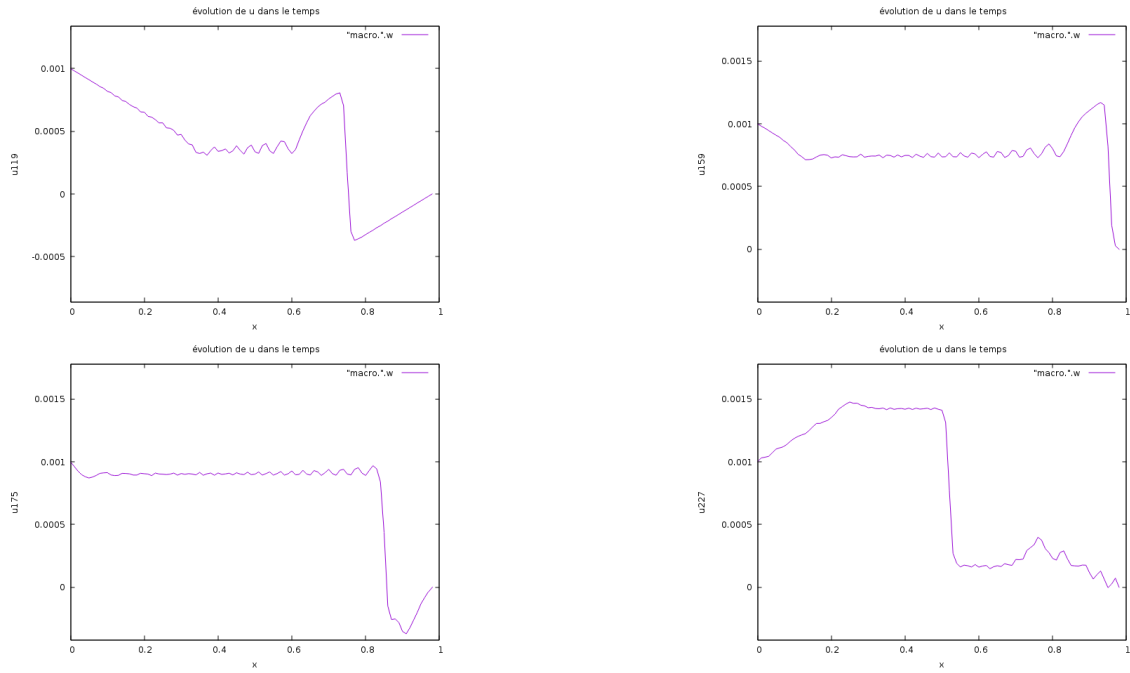
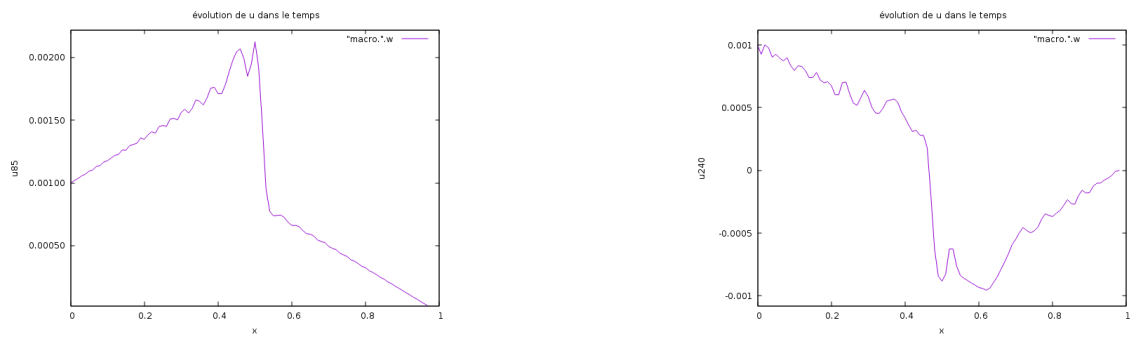


FIGURE 5.16 – snapshots de $u(t, \cdot)$ aux temps $t=0.119$, $t=0.159$, $t=0.175$ et $t=0.227$ avec une pente initiale négative



déplacement vers la droite

déplacement vers la gauche

FIGURE 5.17 – $u(t, \cdot)$ avec une pente initiale traverse une discontinuité au centre

d'espace, et enfin nous allons tracer la solution numérique de ces deux équations. Nous allons aussi nous intéresser à l'évolution de l'erreur qu'on définira en fonction des deux schémas.

Étude de l'erreur

Pour l'étude de l'erreur, nous choisissons de comparer la résolution numérique de l'équation des ondes au cas de référence du mouvements des bouchons. Un tel choix est justifié par plusieurs raisons : l'évoquation de l'équation des ondes s'est faite à partir de ce cas de référence, l'équation de ce cas de référence a la forme d'une équation des ondes. De plus, il est facile d'implémenter des conditions initiales non nulles ou un changement de milieu dans la résolution numérique de l'équation des ondes et faire cela ne changerait que peu les valeurs numériques du calcul de l'erreur. Par notre équation on entendra ici l'équation qui représente le mouvement des bouchons, et on la comparera avec l'équation des ondes, qui est le modèle théorique auquel on se ramène.

On représentera la solution de notre équation en violet tandis que la solution de l'équation des ondes sera représentée en bleu.

Nous commençons par définir l'erreur err par la formule suivante :

$$err = | u - o |$$

où u est le vecteur solution de l'équation d'évolution de notre problème et o est le vecteur solution de l'équation des ondes.

On peut aussi écrire :

$$err_n = | u_n - o_n |$$

De ce fait, on peut évaluer l'évolution de l'erreur dans le temps et repérer les endroits où l'erreur est grande.

Pour pouvoir implémenter cette erreur, on doit résoudre numériquement l'équation des ondes dans notre algorithme en parallèle de notre équation d'évolution.

On rappelle que l'équation des ondes que l'on prétend approcher s'écrit sous la forme :

$$\frac{d^2 u}{dt^2} - c^2 \frac{d^2 u}{dx^2} = 0$$

où $c^2 = \frac{k\beta}{\rho(1-\alpha)\alpha^2}$

et la discrétisation par différence finies de l'équation des ondes est :

$$o_n^{m+1} = 2o_n^m - o_n^{m-1} + c^2 \frac{dt^2}{dx^2} (o_{n+1}^m - 2o_n^m + o_{n-1}^m)$$

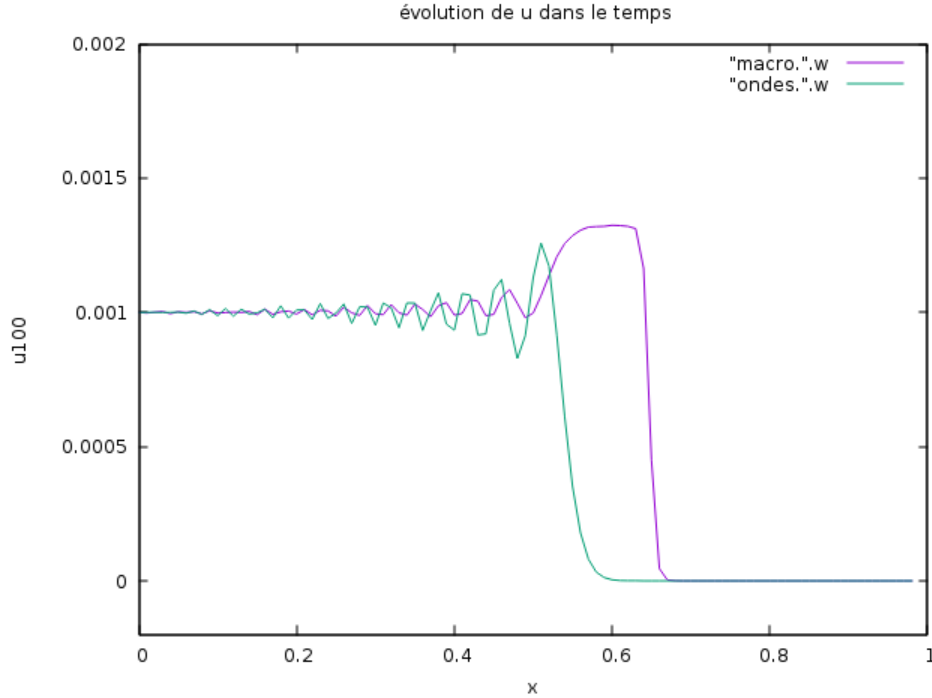


FIGURE 5.18 – Tracé de $u(t,.)$ et de $o(t,.)$ à $t=0.1$ s

On retrouve effectivement le même type de déplacement au cours du temps. On rappelle que ces deux courbes ont été tracées dans les conditions de notre cas de référence, à savoir une vitesse initiale nulle. Et on peut observer que si la comparaison avec une équation des ondes semble correcte, la propagation de notre solution est plus rapide que la propagation de l'onde dans le modèle de l'équation des ondes, et ceci malgré le fait que l'on définisse la vitesse de l'onde avec les paramètres du système physique de notre équation.

Le vecteur *erreur* est tracé sur la figure 19. Il est important de noter que sur le court intervalle de temps que l'on observe (quelques secondes tout au plus), le vecteur *err* est borné. On peut le voir comme une erreur de consistance et en déduire la bonne consistance du schéma. En effet, si on réécrit :

$$err_n^m = |u_n^m - o_n^m|$$

la n -ème composante à l'itération de temps m . On voit que l'erreur est bornée par $1.4e^{-3} = C.\delta t^p$. En fait, le pic qu'on observe sur la courbe et qui se déplace au court du temps est la conséquence du décalage en période entre la solution de notre équation et la solution de l'équation des ondes. On voit bien sur la figure 18 que même si les deux courbes semblent être des ondes, leur période spatiale est manifestement différente (on peut supposer que la différence entre les deux périodes temporelles est négligeable). On peut donc, dans un second temps, considérer l'erreur sans prendre en compte ce décalage de période en modifiant les options de traçage.

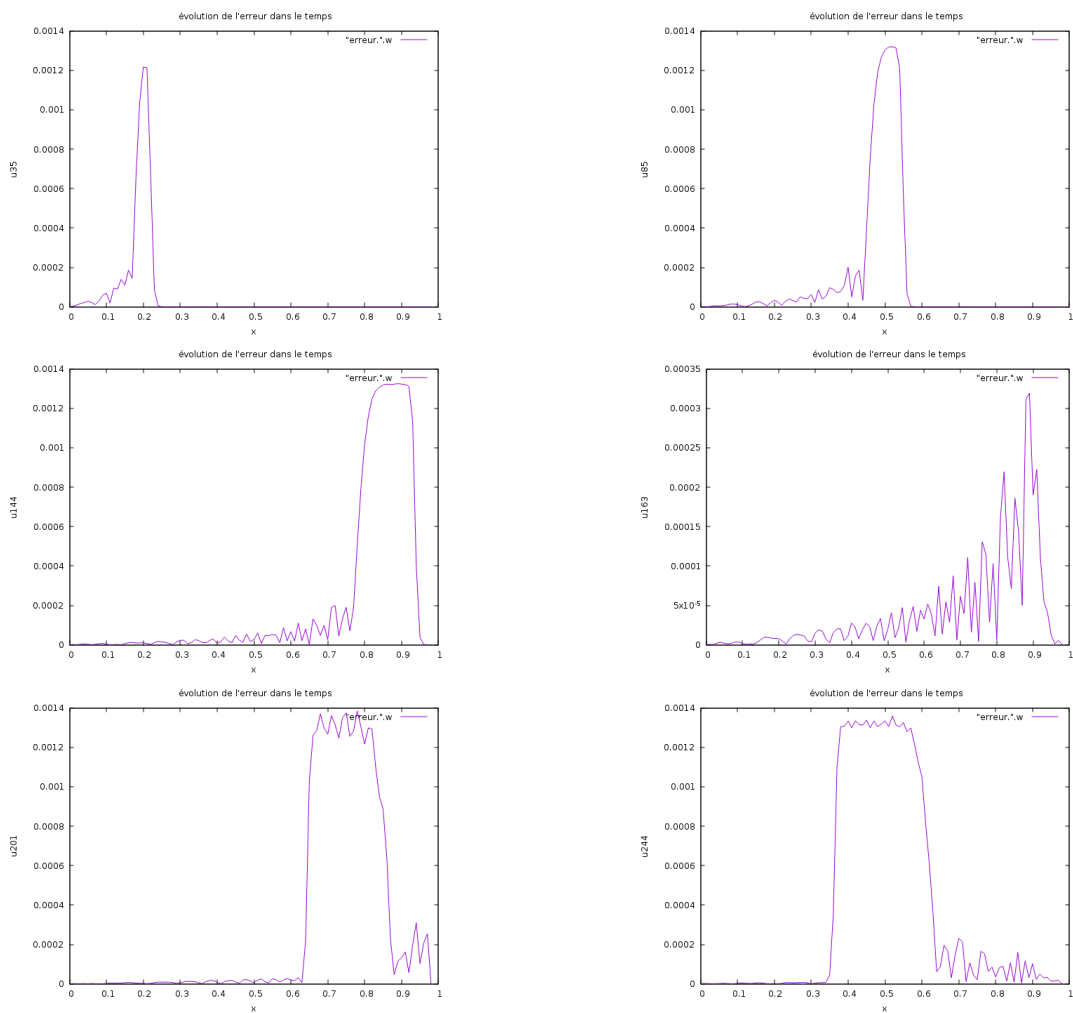


FIGURE 5.19 – tracé du vecteur *err* aux temps $t=0.035$, $t=0.085$, $t=0.144$, $t=0.163$, $t=0.201$ et $t=0.244$

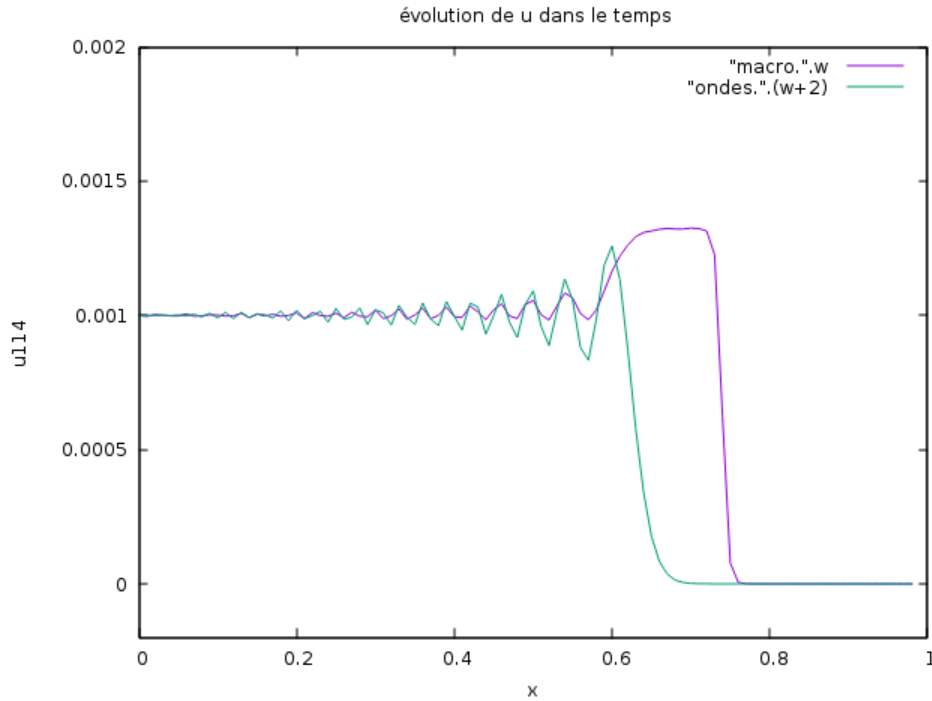


FIGURE 5.20 – Correction de l’erreur

On corrige l’algorithme en décalant les indices pour synchroniser (figure 20) les solutions des deux équations.

On va également modifier le calcul du vecteur d’erreur en multipliant chaque composante par une indicatrice :

$$err_n = |u_n - o_n| \cdot \mathbb{1}_{|u_n - o_n| < 3 \cdot 10^{-4}}$$

On observe alors la courbe de l’erreur corrigée sur la figure 21 :

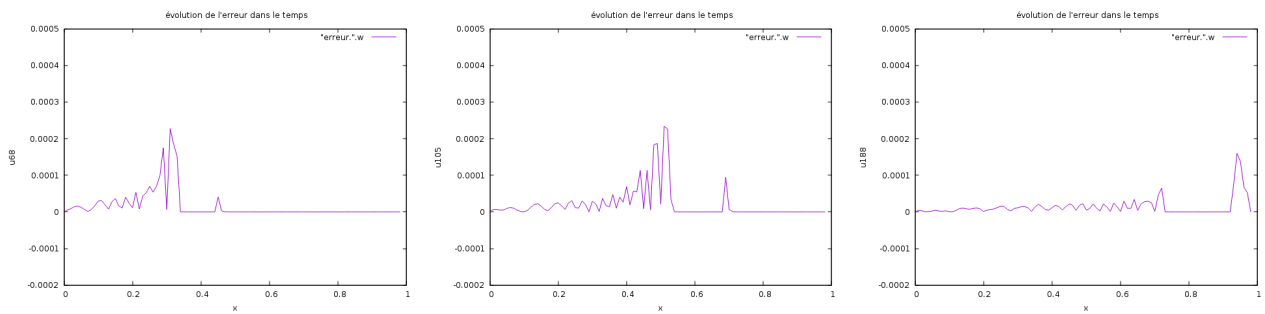


FIGURE 5.21 – Erreur corrigée aux temps $t=0.068$, $t=0.106$ et $t=0.188$

On rappelle qu’on a choisit un pas temporelle $\delta t=0.001$ donc on peut au moins considérer une convergence linéaire, ce qui vient du fait qu’on a approché notre équation avec des schémas d’Euler explicites.

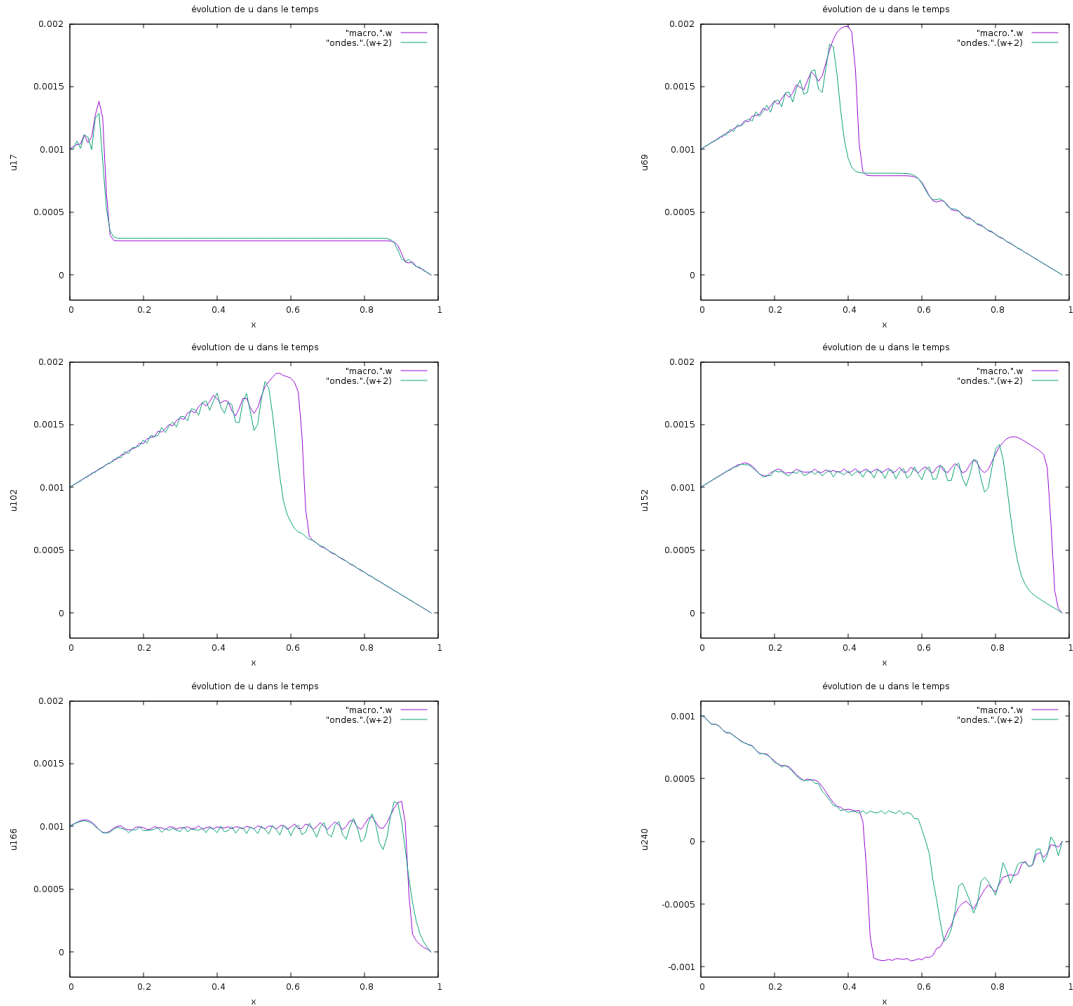


FIGURE 5.22 – traçé du vecteur u et o avec une vitesse initiale non nulle aux temps $t=0.017$, $t=0.069$, $t=0.102$, $t=0.152$, $t=0.166$ et $t=0.240$

Nous pouvons maintenant regarder l'erreur dans le cas d'une vitesse initiale non nulle. On reprend les cas de vitesse initiale non nulle étudiés dans cette section. Les outils utilisés, tels que les schémas d'Euler ou la définition de l'erreur corrigée, ne changent pas.

Notons que comme d'habitude le phénomène ondulatoire se propage de gauche à droite, et qu'une fois qu'il se heurte à la paroi il est renvoyé vers la gauche. Ici, l'onde repart vers la gauche à partir de la 5ème image.

Déjà, nous pouvons observer que les courbes de la solution de notre équation et de la solution à l'équation des ondes coïncident bien dès les premières itérations comme dans le cas d'une vitesse initiale nulle. L'écart entre les deux solutions, qui se situe au niveau du dernier pic ondulatoire, se creuse à travers le temps. Ceci est d'autant plus visible quand l'onde repart après s'être heurté à la paroi de droite où l'écart est encore plus visible (6ème image). Nous pouvons alors penser que la différence entre les deux ondes est une amplitude différente causée par une impulsion initiale différente, ou une vitesse différente. Nous penchons vers la deuxième hypothèse même si le fait que

les deux ondes se confondent à nouveau au niveau de la paroi droite la remet en question. Nous pouvons donc globalement nous satisfaire de la comparaison entre notre solution et la solution d'une équation d'onde.

Remarque : Nous avons inclus dans l'algorithme une option pour choisir la masse volumique ρ de base, c'est-à-dire celle qui caractérise le milieu principal où le deuxième milieu dans le cas d'une juxtaposition de deux milieux. Etant donné que l'on a défini la vitesse de l'onde selon les mêmes paramètres physiques dans les deux équations, des comportements similaires se retrouvent si nous changeons des paramètres particuliers. Par exemple si nous changeons la vitesse ou l'un des paramètres qui la constitue, comme ρ , nous observerons les deux ondes se déplacer avec une nouvelle vitesse mais les deux courbes seront toujours synchronisées de la même façon. C'est pourquoi nous avons choisi de ne pas montrer la comparaison avec une masse volumique différente même si cela peut être instructif d'un point de vue pratique (cf "Applications").

Il est aussi intéressant d'étudier l'erreur quand le milieu est altérée. Nous allons reprendre les cas étudiés précédemment, à savoir une discontinuité au centre du milieu ou bien une juxtaposition de milieu. Nous modifions l'algorithme pour modifier ρ en même temps dans les calculs des deux schémas (via la constante $\mu = \mu(\rho)$ dans l'algorithme).

Même si on voit mieux le phénomène en visionnant le film du déplacement (disponible en lançant le programme), nous pouvons toutefois remarquer le saut en $x=0.5$ et plus précisément l'onde réfléchie qui en incombe dans le cas de la solution de notre équation (figure 24 - première image).

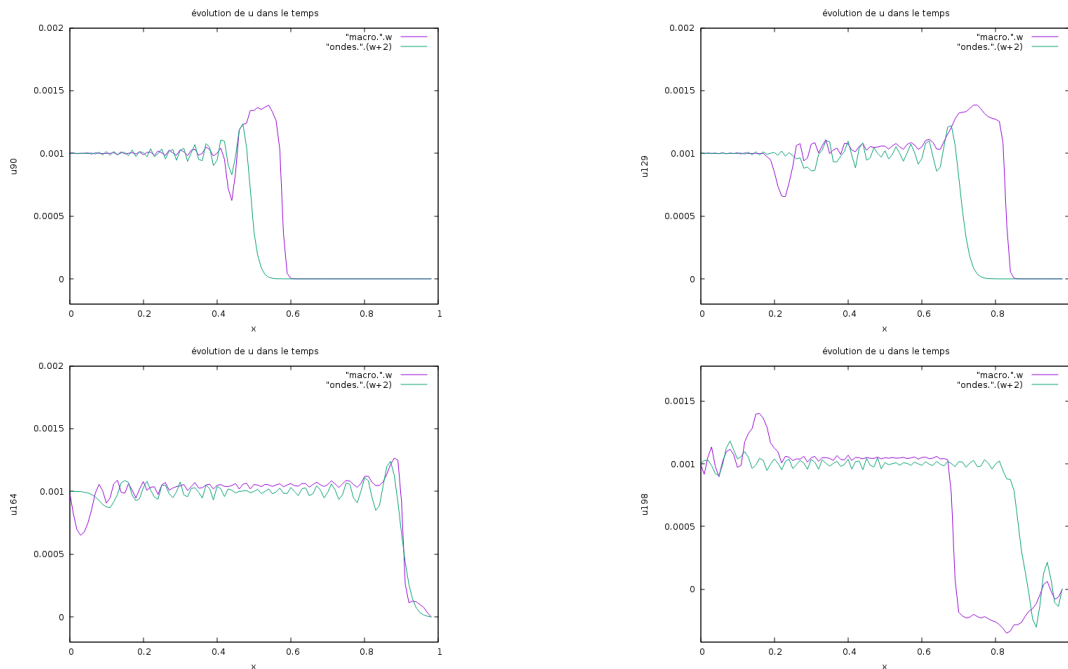


FIGURE 5.23 – tracé des deux solutions aux temps $t=0.090$, $t=0.129$, $t=0.164$ et $t=0.198$ avec une discontinuité au centre

Une onde réfléchi est également visible sur la courbe de la solution à l'équation des ondes (deuxième image) ; c'est logique puisque c'est comme cela qu'une véritable onde se comporte dans ces conditions.

La concordance entre les graphiques des deux solutions renforce notre idée de concordance à un déphasage près. On observe (troisième image) que l'amplitude de chaque oscillation est plus élevée dans le cas de la solution à notre équation. On pourrait donc penser que on peut obtenir une approximation de la solution à notre équation à partir de la solution de l'équation des ondes à laquelle on associe une vitesse particulière et un constante corrective d'amplitude :

$$(S) \Leftrightarrow \frac{d^2u}{dt^2}(x, t) - \tilde{c}^2 \frac{d^2u}{dx^2} = 0$$

avec

$$o = C_{amp}u$$

Nous n'avons cependant pas trouvé de moyen numérique pour déterminer une bonne estimation de la constante C_{amp} si elle existe. Nous avons tenté d'atténuer la solution en fin de chaque itération temporelle ($o = .95o$ par exemple) mais nous avons assisté à la place à l'atténuation du mouvement ondulatoire globale :

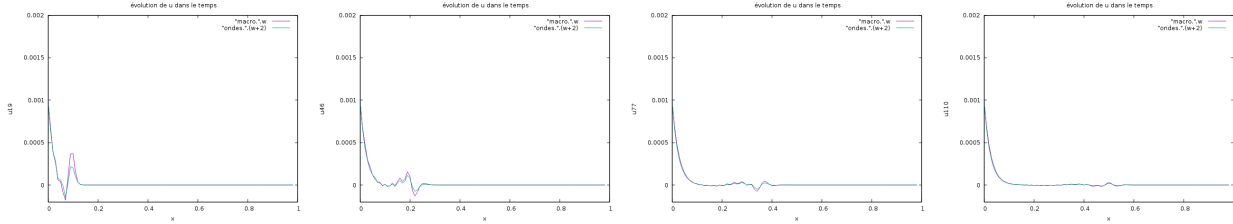
On voit ainsi sur la figure 24 que si l'on réduit de seulement 5%, on a déjà une atténuation du mouvement ondulatoire et dans le cas du mouvements des bouchons, ces derniers se stabilisent en leur position d'équilibre. Plus on réduit le mouvement de l'onde à chaque itération et plus l'équilibrage se déroule rapidement. Après réflexion, on aurait pu s'attendre à ce résultat puisque d'après notre procédé on a réduit l'écart de mouvement avec la position d'équilibre (qui est égale à 0) sur chaque composante et à chaque itération. Cela se traduit par une baisse d'amplitude et par le fait que la solution tende progressivement vers 0 à chaque itération de temps et d'espace.

D'autre part on compare l'effet de l'atténuation selon plusieurs paramètres :

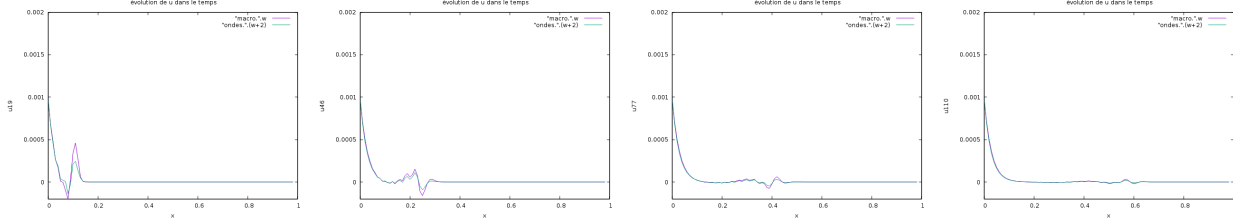
Naturellement, une onde qui se déplace dans un milieu avec une masse volumique plus faible, et donc qui a une plus forte vitesse, s'atténue sur le mme intervalle temporelle mais parcourt une plus longue distance. Si l'on ajoute une impulsion initiale, on peut observer qu'elle est de toute façon vite atténuée indépendamment du pourcentage d'atténuation. Finalement on vérifie graphiquement qu'un plus grand pourcentage d'atténuation lisse la courbe (beaucoup) plus rapidement.

On a choisi de se pencher sur ce phénomène d'atténuation pour illustrer le principe de viscosité, ou encore la résistance à l'écoulement, d'une onde dans un milieu. En réalité, le coefficient d'atténuation dépend du milieu. On peut rencontrer ce phénomène en acoustique, en électromagnétisme ou en mécanique et d'ailleurs l'équation des ondes que l'on a résolu s'y prête bien.

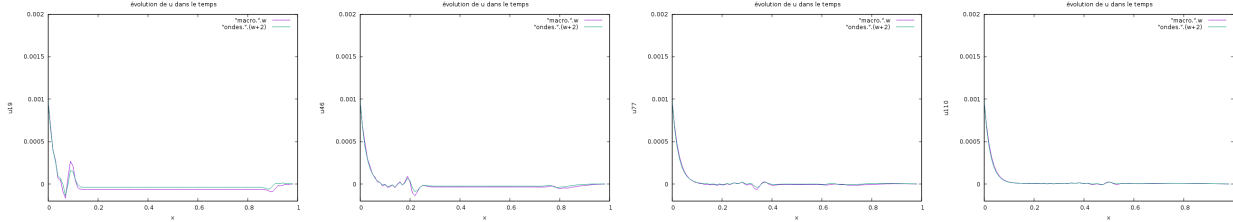
Remarque : Nous pouvons très bien développer en détail l'atténuation en ajoutant une condition dans l'algorithme. Par exemple, une onde atténué quand elle touche le sol se traduit facilement par une condition sur l'amplitude de chaque composante. Nous n'allons pas plus loin par manque de temps et parce que cela sort du cadre de cette étude.



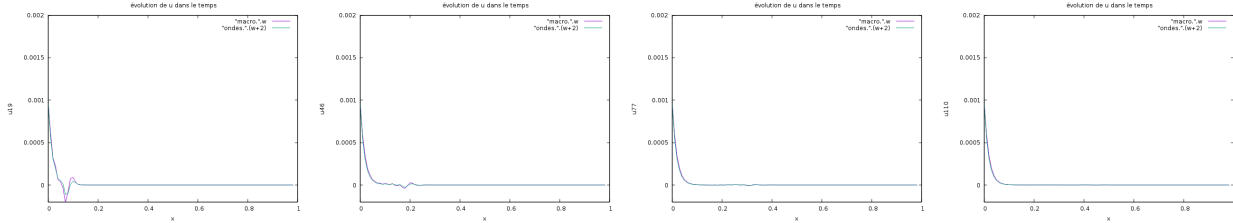
Atténuation de 5% dans l'eau ($\rho = 1$)



Atténuation de 5% dans le pétrole ($\rho = 0.8$)



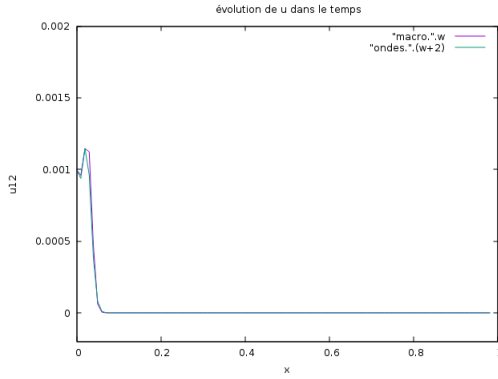
Atténuation de 10% dans l'eau avec pente initiale positive



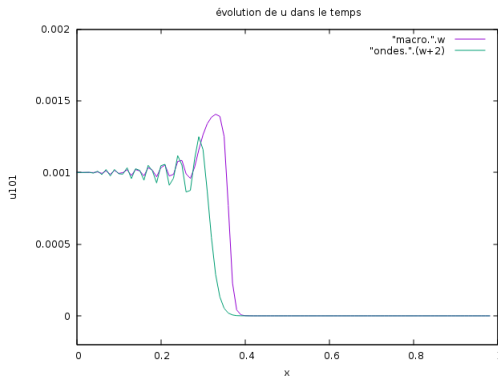
Atténuation de 10% dans l'eau

FIGURE 5.24 – u et o atténués aux temps $t=0.019$, $t=0.046$, $t=0.077$ et $t=0.110$

Une autre manière d'altérer le milieu est, comme nous l'avons vu, de considérer une juxtaposition de milieux de masse volumique différente (ici un milieu avec une masse volumique $\rho_1 = 3$ comme le brome et un milieu avec une masse volumique $\rho_2 = 1$ comme l'eau). Nous allons dans un premier temps revenir repartir du cas de référence et changer le milieu homogène en une juxtaposition d'un milieu d'eau et d'un milieu de brome à 0 degré Celsius (liquide). Nous ferons ensuite varier les paramètres classiques tels que la vitesse initiale et l'atténuation.



Nous considérons les deux ondes avec la correction de déphasage, sans vitesse initiale. Au commencement les ondes ont naturellement la même allure : les deux perturbations se confondent presque. De plus, l'onde et la pseudo-onde se déplacent visiblement à la même vitesse, et la seule différence vraiment notable est encore une fois le pic d'amplitude qui est plus élevé chez la pseudo-onde.



Remarque : Nous n'en avons pas parlé mais nous aurions aussi pu faire partir l'onde de la droite en inversant les indices d'espace et dans ce cas la situation aurait été complètement analogue à ceci près que l'onde débiterait sa course dans le milieu avec la plus petite masse volumique.

Nous pouvons observer sur la figure 25 le passage d'un milieu à un autre de l'onde et de la pseudo-onde. Sur la première image, nous voyons que c'est la pseudo-onde qui, malgré sa forte ressemblance avec l'onde, a un pic d'amplitude plus élevé et plus long au devant et qui de ce fait arrive à la frontière entre les deux milieux en première. Nous observons alors une montée en amplitude puis, sur la deuxième image, la naissance d'une onde réfléchie qui va revenir en arrière. Nous observons également au même moment la traversée de la frontière de l'onde normale (c'est-à-dire de l'équation des ondes) qui monte donc aussi en amplitude. Enfin sur la dernière image nous voyons l'onde et la pseudo-onde, toujours synchronisées en période, qui s'apprêtent à frapper la paroi de droite et dont la partie réfléchie continue son parcours vers la gauche. Nos observations sont faites aux temps avancés $t=0.146$, $t=0.170$ et $t=0.231$.

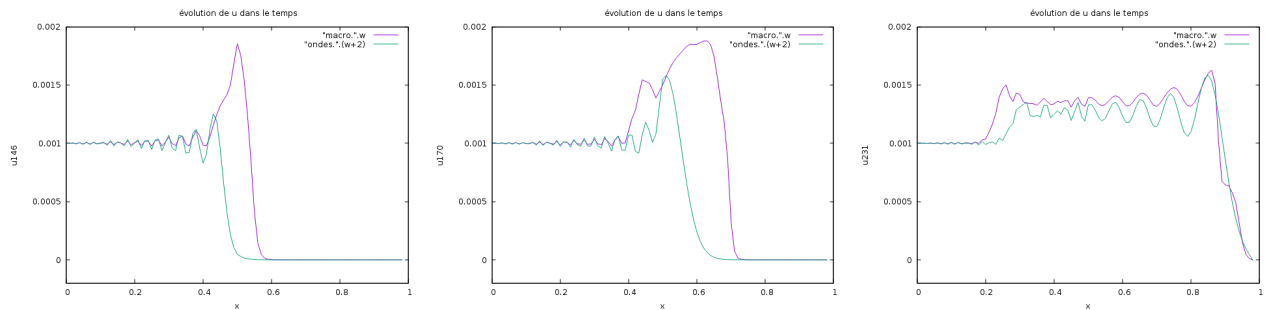


FIGURE 5.25 – L'onde et la pseudo-onde passent d'un milieu à l'autre

Nous nous intéressons ensuite au cas où la vitesse initiale n'est plus nulle et la pente initiale est positive. Comme dans les autres cas, nous pouvons nous attendre à ce que le comportement de l'onde soit similaire dans un premier temps, et que la différence se fera surtout au niveau du passage entre les deux milieux où il y aura certainement un phénomène de réflexion. Nous observons ainsi la figure 26 les graphiques de l'onde et de la pseudo-onde quand elles partent avec une pente positive et qu'elle traverse simultanément deux milieux consécutifs.

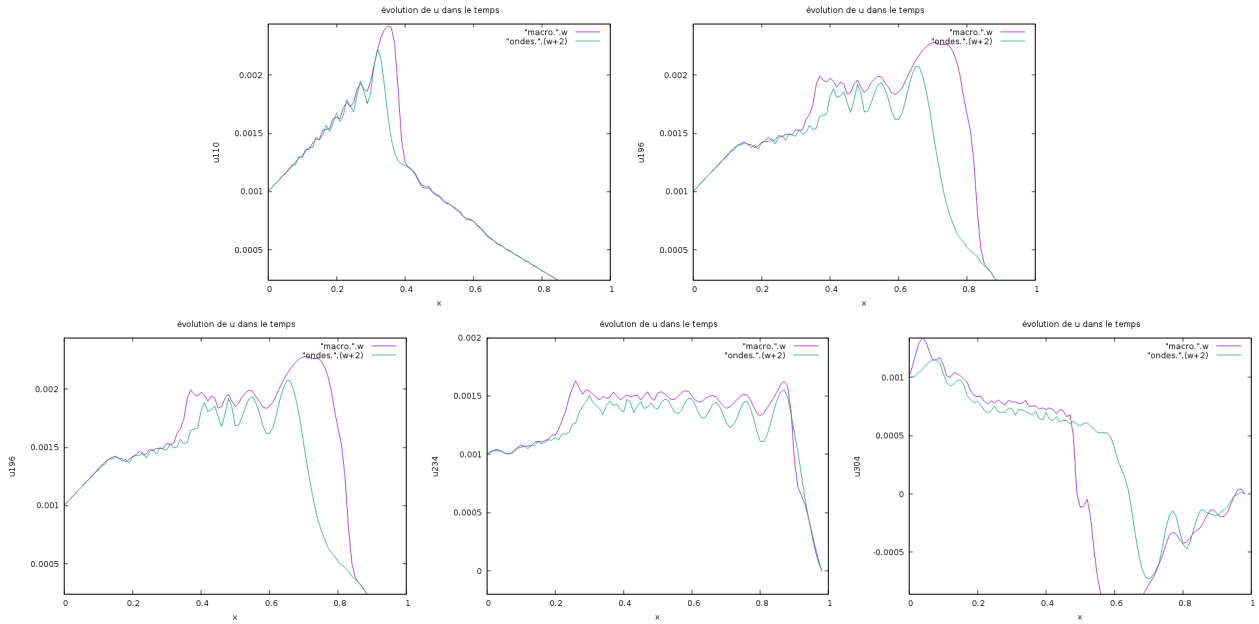


FIGURE 5.26 – L'onde u et la pseudo-onde o traversant deux milieux juxtaposés avec une pente initiale positive

En mesurant à nouveau l'atténuation, on constate que le comportement est le même que dans le cas où il y a seulement une séparation centrale entre deux milieux identiques. D'ailleurs, la réponse sera toujours la même quand il y a atténuation peu importe les conditions dans lesquelles on se place : on obtient toujours une courbe de la forme de celle représentée sur la figure 27.

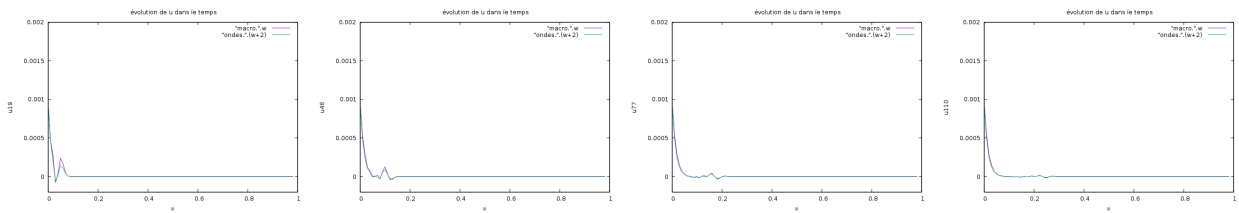


FIGURE 5.27 – L'onde et la pseudo-onde traversant une juxtaposition de milieu et étant atténuées

Conclusion

Au travers de ces différentes observations, il transparaît que dans un milieu non homogène, le comportement de notre solution, tout comme le comportement d'une onde, est modifié dès qu'elle passe d'un milieu à un autre : une partie de l'onde est réfléchiée et l'autre partie est transmise. Ce phénomène a été illustré graphiquement par notre simulation. En effet, notre résolution de l'équation du comportement des bouchons fournit une fonction qui a un comportement ondulatoire. Cette solution est soumise à différentes vitesses initiales ainsi qu'à plusieurs milieux non homogènes et le tracé de son évolution au cours du temps est similaire à la progression d'une onde sinusoïdale dans l'espace. On a observé les réponses aux différentes stimulations que l'on a évoqué. Ce qu'il faut retenir de cet axe d'étude est que dans des conditions simplistes et simplifiées, le mouvement des bouchons dans notre problème a un comportement ondulatoire car la courbe que l'on obtient a l'allure de la solution à l'équation des ondes et que sous certaines conditions initiales modifiées (milieu, vitesse,...) les répercussions sur la courbe sont les mêmes que pour une onde.

De ce fait, nous pourrions nous permettre d'appeler notre solution " l'onde " tout comme nous appelons l'équation du mouvement des bouchons "notre équation". Nous rappelons que l'équation du mouvement des bouchons est l'équation qui encadre le mouvement des segments liquide et gazeux (voir chapitre 1).

II. Température non constante

Milieu homogène avec conditions initiales nulles

Dans cette section, nous allons nous intéresser au cas où la température évolue au sein du milieu. On rappelle que la température est constante dans une bulle, entre le n-ème et n+1-ème bouchon, et vaut :

$$T(0) + \frac{1}{2}T'(x_i + x_{i+1})$$

Nous supposons que la température évolue dans le milieu linéairement, et on va donc traduire T comme une fonction affine de l'espace.

Dans le cas d'une température affine :

$$\ddot{u} - c^2(x)u'' = -\frac{2k\beta\alpha T'(x)}{\rho\alpha^2(1-\alpha)}$$

avec $c^2(x) = \frac{k\beta T(x)}{\rho\alpha^2(1-\alpha)}$

et son schéma discrétisé correspondant est l'équation :

$$u_n^{m+1} = 2u_n^m - u_n^{m-1} + \frac{k\beta\delta t^2}{\rho\alpha(1-\alpha)} \left(T \frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{\alpha h^2} - 2T' \right)$$

On va réaliser nos observations avec un gradient de température de 4% et une température initiale de 1, c'est à dire : $T(x) = .04x + 1$.

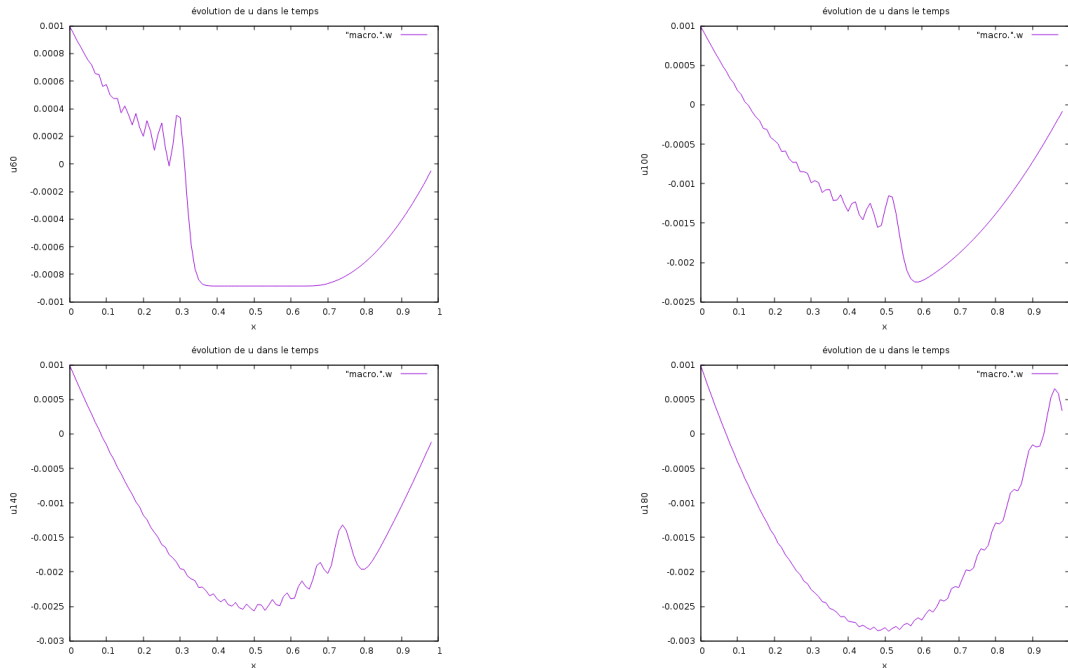


FIGURE 5.1 – Propagation de $u(t, \cdot)$ avec un gradient de température 4 % aux temps $t=0.060$, $t=0.1$, $t=0.140$ et $t=0.180$ - aller

Nous observons que la solution garde une trace de son comportement ondulatoire dans le sens où on voit une sinusoïde suivie d'un pic et d'un retour à l'origine, comme dans le cas de référence. Cependant la courbe s'affaisse (figure 1 - première image) comme dans le cas d'une vitesse à pente négative pour une température constante. Elle continue de s'écraser jusqu'à environ (deuxième et troisième image) en conservant son comportement ondulatoire. Le mouvement atteint finalement la paroi (quatrième image) et est réfléchi.

Nous pouvons ainsi observer son mouvement durant le retour sur la figure 2. Comme il est avisé de ne se fier seulement qu'aux observations dans un court laps de temps, on va considérer que le mouvement de notre solution se termine comme celui d'une onde (figure 2 - troisième image).

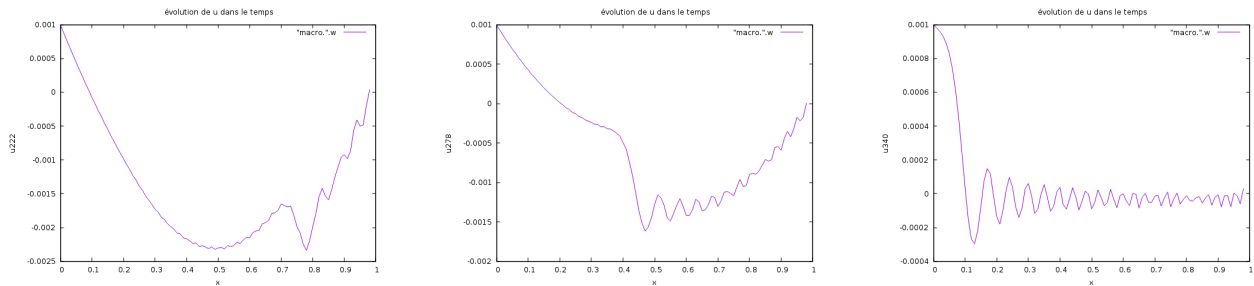


FIGURE 5.2 – Propagation de $u(t,.)$ avec un gradient de température 4 % aux temps $t=0.222$, $t=0.278$ et $t=0.340$ - retour

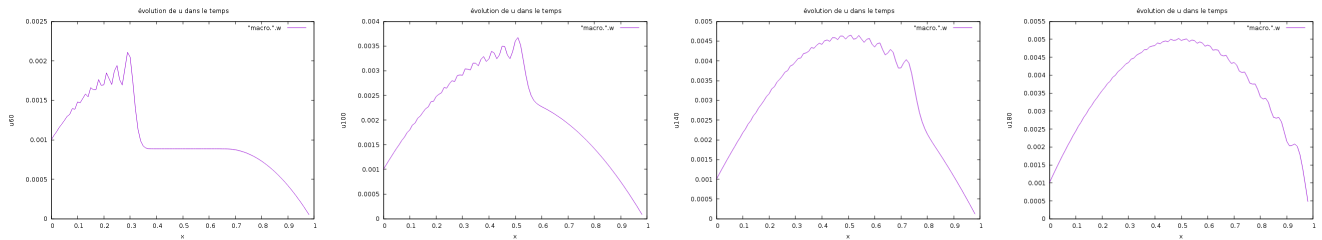


FIGURE 5.3 – Propagation de $u(t,.)$ avec un gradient de température -4 % aux temps $t=0.060$, $t=0.1$, $t=0.140$ et $t=0.180$

Comparons ces observations avec ce que l'on obtient si l'on prend un gradient de température -4% (figure 3).

Nous pouvons remarquer que le comportement est strictement le même que dans le cas d'un gradient positif à ceci près que la pente initiale est inversée.

Ainsi, si nous faisons varier le gradient de température, nous agissons sur l'allure de l'onde initiale qui est gardée le long de la courbe. En effet, on peut observer sur la figure 4 la propagation de $u(t,.)$ avec un gradient de température 8% et on voit que les oscillations caractéristiques d'une onde que nous observions au début sont ici très amorties.

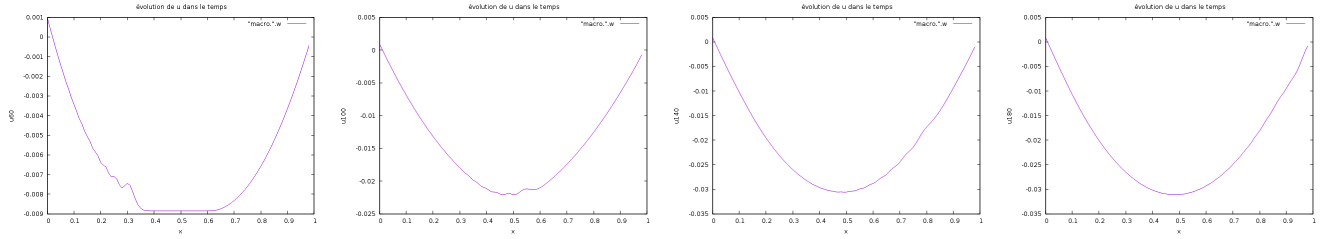


FIGURE 5.4 – Propagation de $u(t,.)$ avec un gradient de température 8 % aux temps $t=0.060$, $t=0.1$, $t=0.140$ et $t=0.180$

Nous pouvons déjà dire que nous avons vérifié numériquement plusieurs phénomènes :

- Plus la température augmente et plus l'onde sera altérée ; elle perdra son caractère ondulatoire et s'amortira.
- Même en présence d'un gradient de température, l'onde déformée retrouve les propriétés qu'elle avait perdues lors de son voyage dans le milieu. On le voit sur la figure 2 dans le cas d'un gradient 4% mais on a assisté au même phénomène avec des gradients différents.

A cela on peut ajouter qu'on a montré que notre simulation numérique était correcte puisque si nous choisissons une température initiale de 1 et un gradient de 0%, c'est-à-dire une température affine de la forme $T(x) = 0.x + 1$, on retombe sur le cas de référence (figure 5).

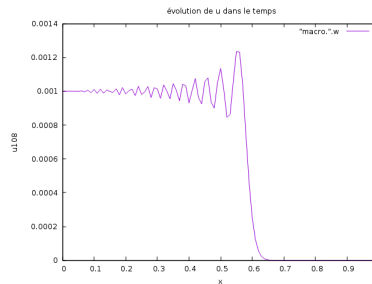


FIGURE 5.5 – Gradient 0% et température initiale 1 - cas de référence

Milieu inhomogène avec conditions initiales non nulles

Nous allons maintenant observer la réaction du système face à un gradient de température dans plusieurs cas : nous allons déjà faire varier le type de milieu, puis nous allons donner une impulsion initiale et enfin nous allons regarder ce qu'il se passe quand on atténue.

Dans un premier temps nous allons reprendre l'expérience initiale, à savoir un gradient de température 4% sans aucune autre modification, et nous allons considérer un milieu avec une discontinuité au centre. Comme dans les autres études que nous avons réalisé, on s'attend à observer une onde réfléchiée et une onde réfractée.

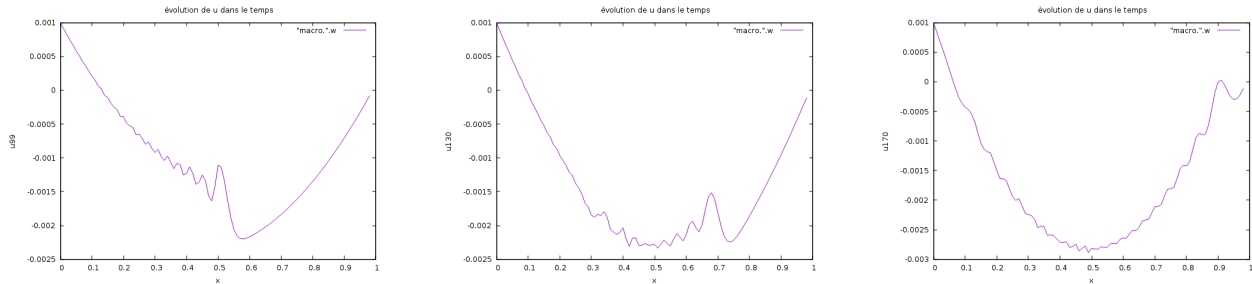


FIGURE 5.6 – $u(t,.)$ - gradient de température 4% - discontinuité au centre

Nous observons bien une onde réfléchiée (figure 6) qui remonte le long de la courbe et une onde diffractée avec une amplitude légèrement amoindrie. On retrouve donc bien le phénomène de réfraction.

Penchons nous maintenant sur un milieu non homogène, composé comme dans la section précédente de deux milieux de masses volumiques différentes séparés au centre.

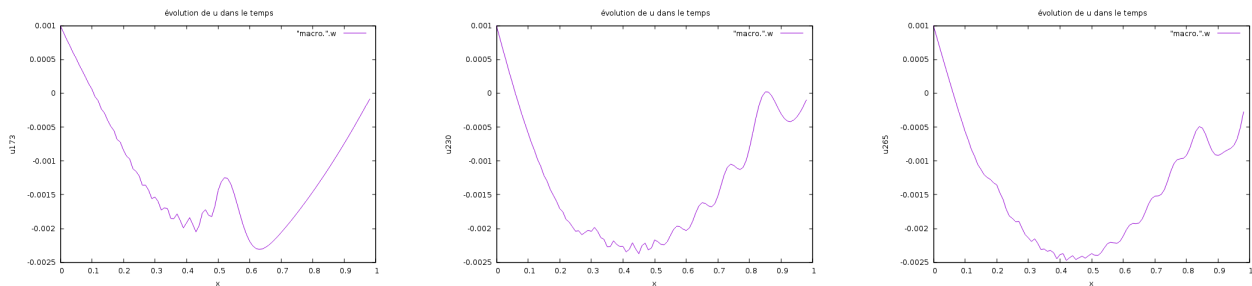


FIGURE 5.7 – $u(t,.)$ - gradient de température 4% - deux milieux juxtaposés

Nous observons encore une fois un phénomène de réflexion d'onde (figure 7) et nous notons que l'onde réfractée semble se déplacer encore plus vite que dans les cas précédents. Ceci est dû au fait que l'onde réfléchiée reste dans le milieu de gauche (celui avec une forte masse volumique) où la température est plus faible.

Nous pouvons donc conclure ici que la température permet manifestement à l'onde d'augmenter sa vitesse et que ce phénomène n'entre pas en contradiction avec les différents paramètres physiques que nous avons étudié.

Regardons à présent les résultats quand on donne une impulsion initiale, toujours avec un gradient de température 4%. Cette observation est intéressante puisque on a déjà remarqué que le gradient de température agit sur la courbe comme une vitesse initiale non nulle.

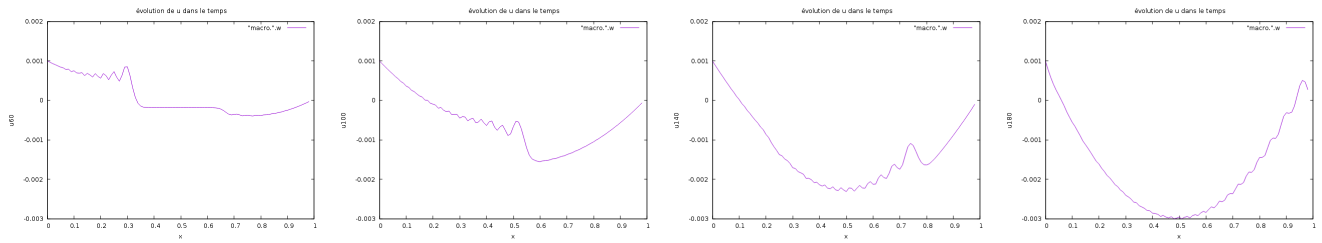


FIGURE 5.8 – $u(t,.)$ - gradient de température 4% - pente initiale positive

En observant les graphiques de la figure 8, nous ne remarquons pas de différences notables avec la même expérience sans vitesse initiale.

Regardons à présente la courbe de déplacement quand la pente initiale est négative (figure 9) et quand on donne une impulsion plus forte (figure 10).

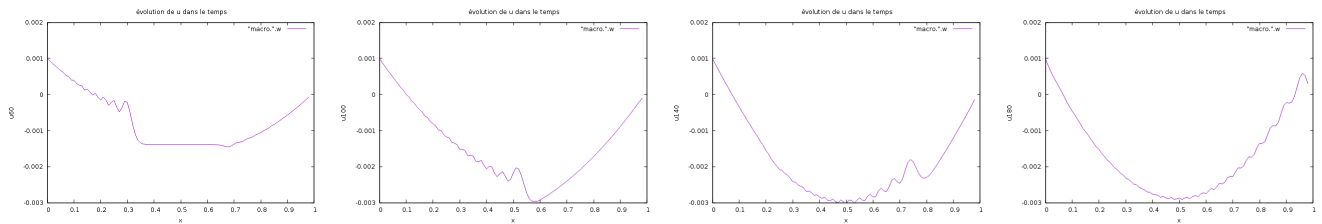


FIGURE 5.9 – $u(t,.)$ - gradient de température 4% - pente initiale negative

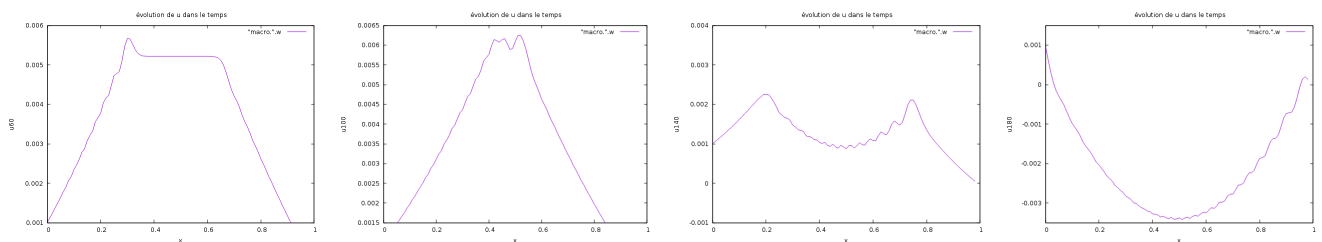


FIGURE 5.10 – $u(t,.)$ - gradient de température 4% - pente initiale positive avec forte impulsion

On remarque sur la figure 9 que si la pente initiale est négative, la courbe reste majoritairement de la même forme. En fait quand on compare les deux jeux de graphiques, on remarque que la pente initiale négative ralentit le mouvement induit par le gradient de température. En effet la courbe aurait du aller vers le haut mais elle est ramenée vers le bas par l'action de la température.

En revanche, on peut voir sur la figure 10 que la courbe n'a pas le même comportement quand on donne une impulsion plus forte. On donne ainsi une vitesse initiale de 0.1 contre 0.01 jusqu'ici. On retrouve bien ici le comportement de l'onde dans le cas d'une pente initiale positive et elle monte davantage puisque l'impulsion initiale est plus élevée. On observe (deuxième image) qu'à partir d'un certain moment la courbe commence à se creuser au niveau du pic d'amplitude : c'est l'action de la

température (gradient positif). Au fil du temps, la courbe continue de se creuser (troisième image) et on retrouve finalement la courbe d'un gradient de température sans vitesse initiale.

Nous concluons ici que l'action de la température non constante sur le déplacement des bouchons prédomine sur l'action d'une quelconque vitesse. En fait, l'effet d'une éventuelle vitesse initiale accordée au mouvement des bouchons est rapidement négligeable devant l'effet du gradient de température, qui consiste en une courbe en cloche le long de laquelle se déplace la perturbation sinusoïdale.

Finalement, nous allons étudier comment se comporte la courbe sous atténuation.

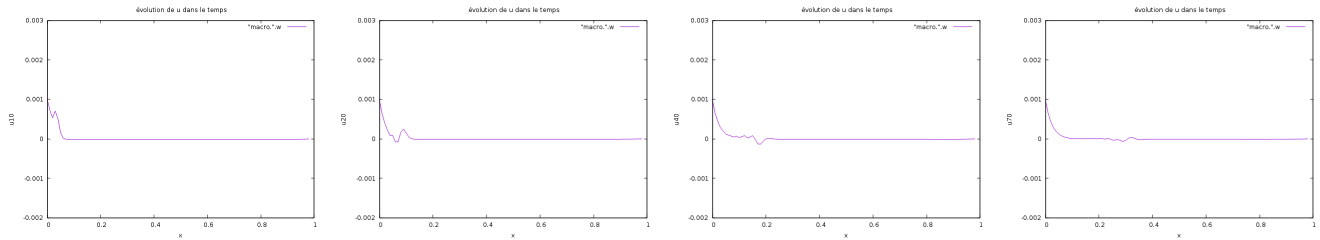


FIGURE 5.11 – $u(t,.)$ - gradient de température 4% - atténuation de 5%

Dans ce cas aussi la courbe associée à un coefficient d'atténuation se comporte de la même façon.

Conclusion

Nous pouvons donc conclure ici que l'ajout d'un gradient de température agit principalement sur la vitesse de la pseudo-onde. Cela permet d'étudier un milieu où la température évolue et le choix d'un modèle affine s'associe bien à un problème en une dimension. Nous avons observé que la courbe de la solution s'affaisse aux environs du centre du milieu et que la sinusoïde, autrefois caractéristique d'une onde et qui se déplaçait sur un axe horizontal, se déplacé ici le long de cette courbe en cloche. Toutefois, nous avons remarqué que quand l'onde se heurte à la paroi et revient, aucune information ne semble avoir été perdue.

Rappelons que physiquement, nous étudions le mouvement des bouchons dans une coupe d'une volume en une dimension et que ce milieu présente une température évolutive, affine. Il ressort de nos observations que l'écart entre la position actuelle du bouchon et sa position d'origine est toujours borné, ou du moins qu'il dépend de l'impulsion initiale. L'ajout d'un gradient de température amène l'uniformisation d'un mouvement, car on se ramenait toujours à une courbe de même allure (les images numéro 4 dans les figures). Enfin, une variation de la température dans le milieu n'affecte pas les propriétés intrinsèques de l'onde, comme son amplitude ou sa période.

III. Phénomène de dissipation

Nous allons modéliser l'équation du mouvement avec le terme de dissipation à laquelle nous allons confronter l'équation des ondes sans dissipation pour mieux observer le phénomène de dissipation. On prendra comme valeur pour le terme de dissipation κv où $\kappa = 0.285$.

On a représenté les instants u^{20} , u^{40} , u^{60} et u^{80} .

On observe graphiquement sur la figure 12 que le mouvement des bouchons (en violet) a la même allure que l'onde au temps $t=0.020$ en étant légèrement atténué. Mais aux instants suivants, on remarque bien la dissipation du mouvement des bouchons alors que l'onde garde la même allure le long du segment.

Cela met bien en évidence le phénomène de dissipation lorsque l'on introduit le terme κv .

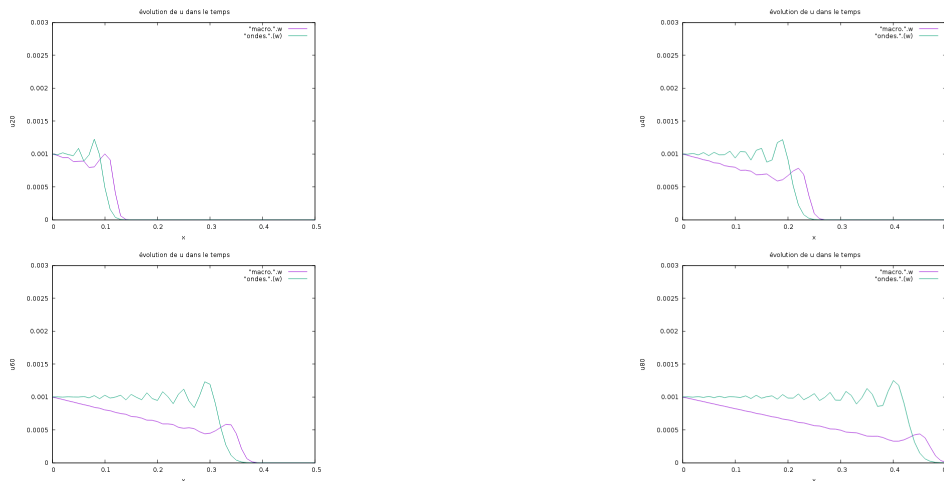


FIGURE 5.12 – Propagation de $u(t,.)$ avec un terme de dissipation aux temps $t=0.020$, $t=0.040$, $t=0.060$ et $t=0.080$ - aller

On observe qu'en fait (figure 13) la courbe du mouvement des bouchons tend vers une droite linéaire. Ceci est dû au fait que nous avons donné une position initiale égale à 0.001 et nous nous limitons à ce cas dans le cadre de ce chapitre.

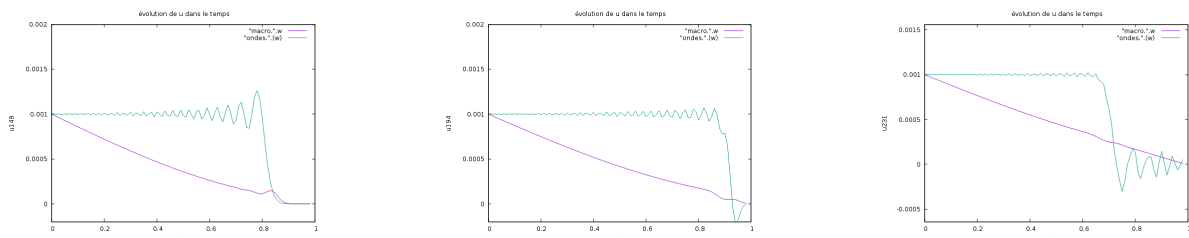


FIGURE 5.13 – Ajout du terme de dissipation - $\kappa = 0.285$

Nous pouvons aussi observer comment faire varier nos paramètres. On vérifie qu'intuitivement, si l'on baisse la valeur absolue du terme de dissipation la courbe sera dissipée moins rapidement (ou moins fort) et inversement. Nous constatons la véracité de ces hypothèses sur la figure 14.

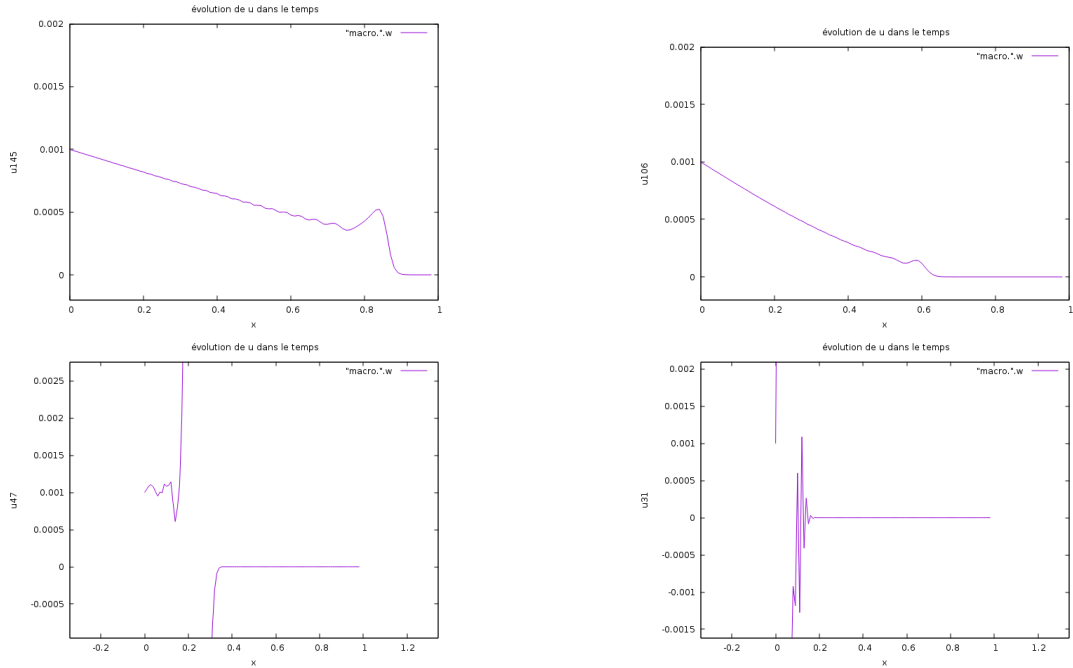


FIGURE 5.14 – Dissipation de u selon plusieurs paramètres κ - Dans le sens de lecture : $\kappa = 0.140$, 0.400 , -0.285 et 14

Il est clairement visible qu'un paramètre κ proche de 0 induit une dissipation plus faible et qu'à l'inverse un κ supérieur induit une dissipation plus prononcée ou plus rapide. On remarque également que le paramètre κ doit être positif pour que l'approche numérique soit valable, ce qui correspond bien au modèle physique. De plus on observe que si en général $\kappa > 0$, le modèle numérique n'est plus valable au-delà d'une certaine valeur (en pratique on a constaté que cette valeur était entre 10 et 14).

Reprenons $\kappa = 0.285$ et observons comment se comporte la solution quand on modifie les conditions initiales.

Dans un premier temps, on observe comment notre solution se comporte quand le milieu n'est plus homogène, mais qu'il a une discontinuité en son milieu (figure 15). On observe encore une onde séparée en deux parties. L'onde réfractée et l'onde réfléchie sont toutefois toutes les deux dissipées dans le temps.



FIGURE 5.15 – Discontinuité au centre du milieu

Dans un second temps, observons la situation où il y a une juxtaposition de milieu (avec les mêmes considérations que précédemment). Ici, nous allons changer le paramètre κ puisque comme le premier milieu par lequel passe l'onde est 3 fois plus dense que le suivant, on observe une augmentation de la vitesse de dissipation et fait que l'onde est totalement dissipée avant d'atteindre le second milieu. Prenons $\kappa = 0.05$. Nous observons sur la figure 16 que l'onde est légèrement dissipée comme attendue, puis qu'elle se sépare en deux et que chaque onde résultante est dissipée à un rythme proportionnel à la masse volumique de son milieu.

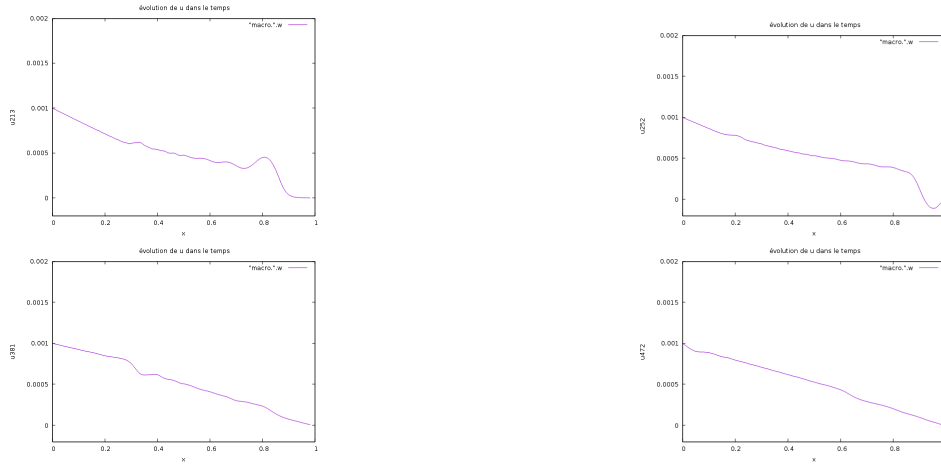


FIGURE 5.16 – Dissipation de u dans le cadre d'une juxtaposition de milieu.

Nous pouvons maintenant nous intéresser à la situation où la vitesse initiale a été modifiée. Nous gardons la même façon de procéder qu'avant :

D'abord, regardons ce qu'il se passe quand on donne à notre solution une vitesse initiale à pente positive. On observe sur la figure 17 que l'onde se comporte globalement comme quand il n'y avait pas de dissipation (il n'y a pas de changement majeur dans l'évolution de la courbe) à ceci près qu'elle est lissée dans le temps à cause de κ (on a repris $\kappa = 0.285$).

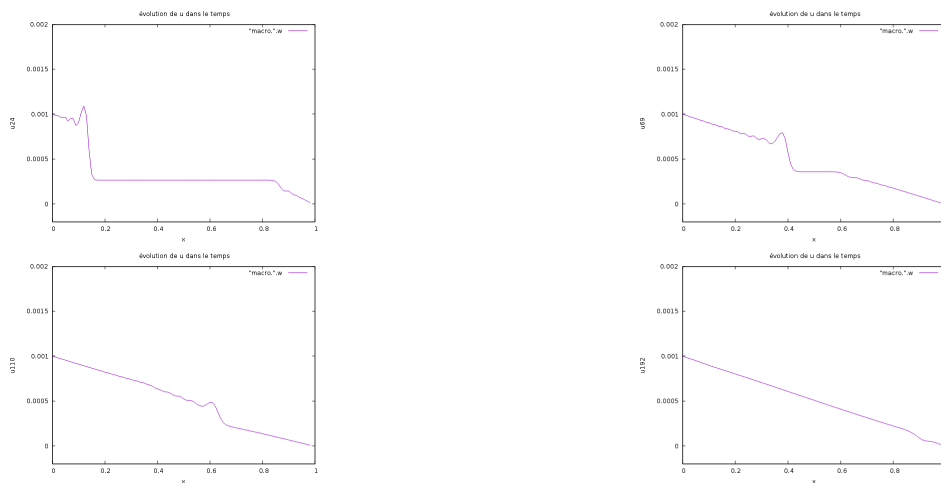


FIGURE 5.17 – Dissipation de u dans le cadre d'une pente initiale positive.

Ensuite, donnons une pente initiale négative. On observe (figure 18) ce à quoi on s'attendait, à savoir que la courbe se comporte comme quand elle subit une pente initiale négative, à ceci près qu'elle est lissée au cours du temps de la même manière que dans nos cas précédents.

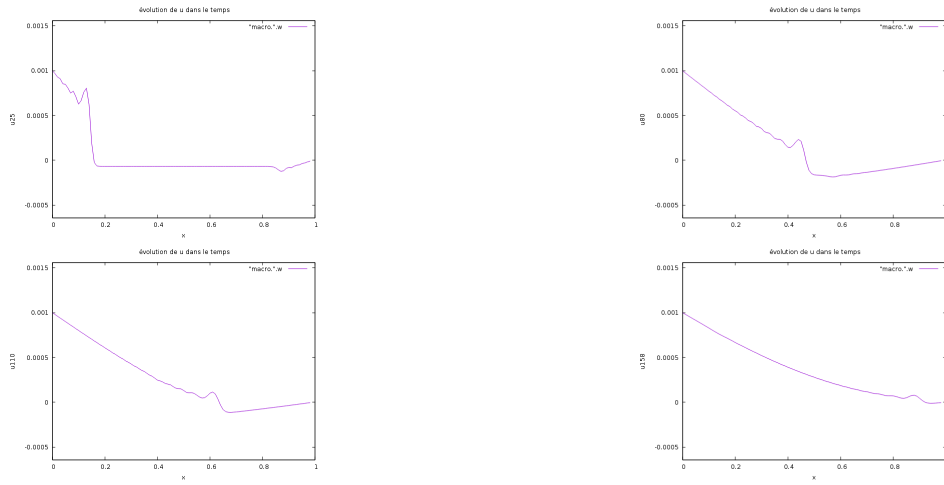


FIGURE 5.18 – Dissipation de u dans le cadre d'une pente initiale négative.

Conclusion

La dissipation est liée à la masse volumique du milieu. Cependant elle ne semble pas être liée aux paramètres intrinsèques de l'onde en question. Elle ne semble pas non plus être liée aux forces extérieures qui agissent sur le système, telle qu'une impulsion initiale. Il est également intéressant de noter que l'on a pu vérifier numériquement le domaine des valeurs possibles de κ . De plus, on peut comparer le phénomène de dissipation à l'atténuation que nous avons précédemment introduit. En fait, on observe graphiquement que l'atténuation telle que nous l'avons codé prédomine sur le phénomène de dissipation (voir figure 19).

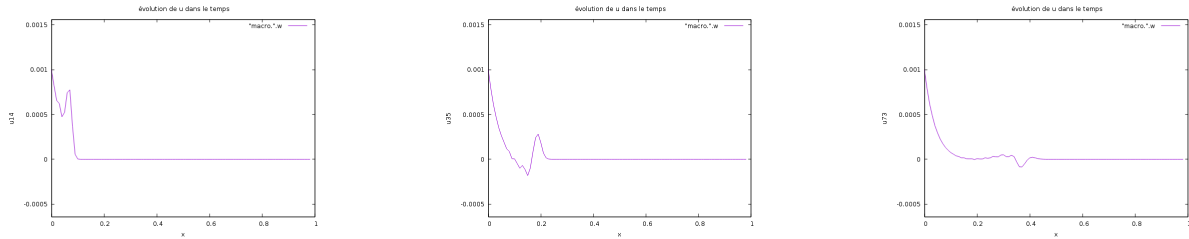


FIGURE 5.19 – Dissipation avec $\kappa = 0.285$ et atténuation de 2%

IV. Temps d'exécution et stabilité numérique

a. Temps d'exécution

Nous allons nous intéresser dans cette section au temps d'exécution de notre programme en fonction de la situation. L'utilité réside dans le fait qu'on pourra comparer le temps d'exécution à l'erreur et ainsi décider si notre simulation est utile ou applicable.

Pour cela, nous allons effectuer plusieurs simulations dans différentes situations, faire la moyenne du temps d'exécution de ces simulations et attribuer le résultat à la situation considérée.

Nous utilisons la fonction `high_resolution_clock::now` de la bibliothèque standard `chrono` en C++ pour avoir un résultat précis du temps d'exécution. Nous démarrons le compteur quand la dernière demande à l'utilisateur a été remplie et nous l'arrêtons à la toute fin de l'algorithme, quand la boucle en temps est terminée et les fichiers écrits. On précise que ces mesures sont faites sur un ordinateur portable équipé d'un processeur Intel Pentium tournant sous Debian où on a réduit au maximum le nombre de processus. On précise le type de machine utilisé car le temps d'exécution varie d'un ordinateur à l'autre.

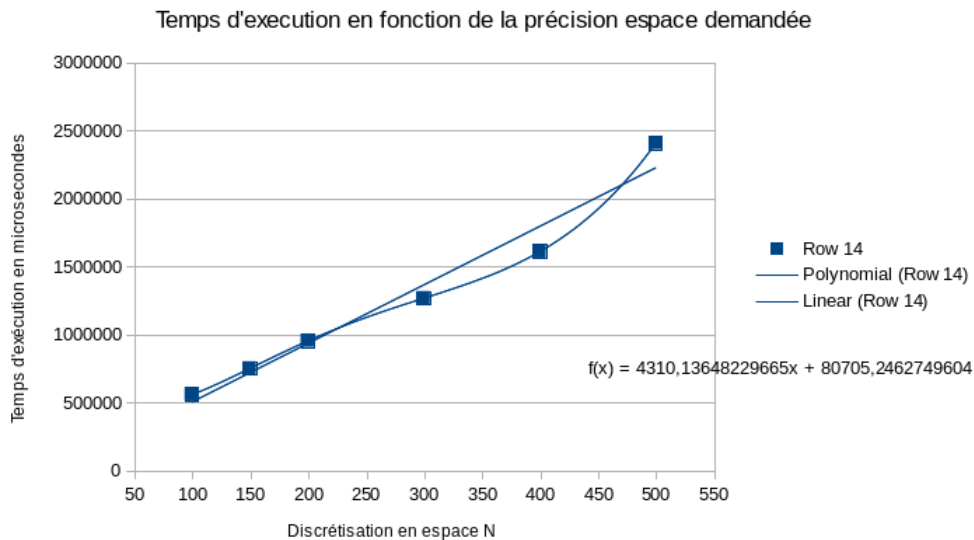


FIGURE 5.1 – Les résultats sur le temps d'exécution

Nous avons réalisé des observations dans 5 cas différents : on a discrétisé l'espace en 100, 150, 200, 300, 400 et 500 points. Pour chacun de ces cas, nous avons réalisés une vingtaine de mesures en temps et nous en avons fait la moyenne. Le graphique ci-dessus représente le temps de calcul en microsecondes selon le nombre de points en espace. Nous avons tracé la courbe de tendance ainsi qu'une interpolation polynomiale de Lagrange de degré 5. Nous pouvons dire que si le temps de calcul semble être linéaire à première vue, mais que pour un nombre plus élevé de points il ne soit plus linéaire. Nous nous attendions de toute façon à autre chose qu'un temps linéaire après avoir construit l'algorithme puisque nous avons plusieurs boucles imbriquées les unes dans les autres.

b. Stabilité

On ne peut pas étudier la stabilité de ce schéma car il n'est pas linéaire. Cependant pour étudier la stabilité, nous allons changer les données dans l'algorithme et montrer que la représentation graphique ne varie pas beaucoup dans certains cas. Ainsi nous allons pouvoir avoir une idée plus précise de la validité de notre modèle et de l'approximation.

Température constante

Nous avons déjà vu que si l'on modifie ρ , la vitesse de propagation de la pseudo-onde est modifiée mais le mouvement reste le même. En fait, on peut poser la quantité $\mu = \frac{k\beta h}{\rho b}$ et faire varier cette quantité dans l'algorithme sans pour autant dénaturer le mouvement. Ainsi toutes les grandeurs qui composent μ ne déstabilisent pas l'algorithme.

Nous avons également vu qu'une vitesse initiale avec une pente égale à 0.01 en valeur absolue est correctement simulé. Si nous choisissons une position initiale nulle et une vitesse initiale nulle, nous obtenons la même courbe que lorsque les conditions initiales sont non nulles.

Si nous prenons une position initiale nulle et une pente initiale positive plus élevée (figure 2), on garde un comportement globale ondulatoire mais la pseudo-onde n'a plus du tout la même forme qu'avant.

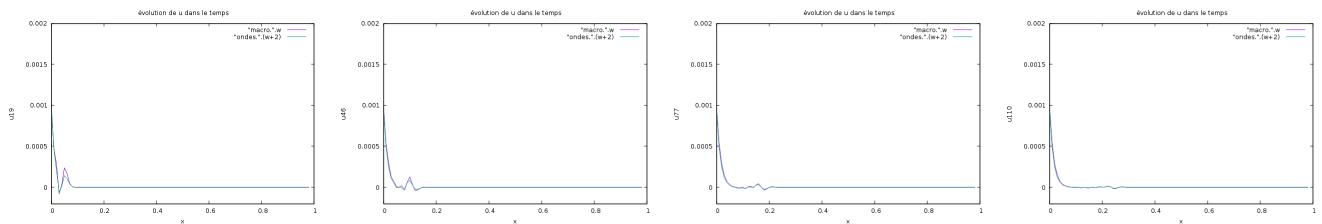


FIGURE 5.2 – $u(t,.)$ - plus forte impulsion initiale

Nous avons également appliqué plusieurs termes d'atténuation et le programme y a toujours répondu de la même manière, à savoir qu'une plus forte atténuation transforme la courbe plus rapidement vers sa position d'équilibre.

Température affine

Nous considérons ici une température affine de la forme

$$T(x) = \alpha x + \beta$$

Nous vérifions aisément que pour $\alpha = 0$ et $\beta = 1$ le cas d'une température constante, notre cas de référence, est retrouvé (figure 3).

Nous remarquons également que l'affaissement de la courbe est directement proportionnelle au gradient de température que l'on choisit (figure 4).

En revanche, le paramètre β n'est pas aussi permissive. En effet si l'on considère à nouveau que la température est constante, il faut prendre $\beta = 1$. Or on a développé un schéma pour étudier comment se comportait la solution dans le cas où la température varie linéairement dans le milieu car c'est une hypothèse physiquement fondée. De ce fait, on s'intéresse à une faible variation de la température étant donné que l'on travail sur des petits segments (h est de l'ordre de 10^{-2}).

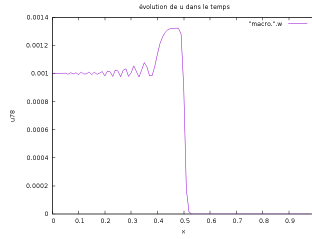


FIGURE 5.3 – $\alpha = 0$ et $\beta = 1$



FIGURE 5.4 – Gradient de température : 10% à gauche et 25% à gauche.

La fonction linéaire de la température doit être suffisamment proche de 1 autour de 0. Ainsi, cela montre que pour correctement apprécier le modèle on doit prendre un gradient relativement petit, mais surtout un paramètre β proche de 1. Dans les faits on prendra toujours 1 et on peut observer (figure 5) que l’algorithme diverge grossièrement quand on change trop fortement ce paramètre.

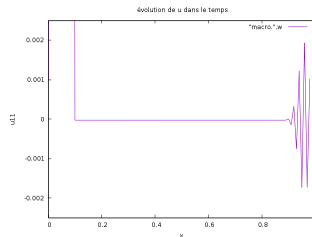


FIGURE 5.5 – $\kappa = 0$

Phénomène de dissipation

Nous avons déjà vu dans la partie consacré à la dissipation que nous ne pouvons pas choisir κ comme bon nous semble. Si l’on vérifie aisément que $\kappa = 0$ revient à choisir le cas sans dissipation (continuité de l’algorithme), on constate qu’un kappa trop élevé ou négatif invalide le schéma numérique, mais puisque ceci est en accord avec la réalité physique nous ne considérons pas cela comme un problème de stabilité. En conclusion, aucun nouveau problème de stabilité vis-à-vis du paramètre κ .

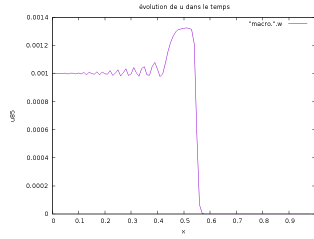


FIGURE 5.6 – $\kappa = 0$

Conclusion

A travers ces quelques observations, on peut conclure que plusieurs paramètres n'entravent pas l'algorithme (pente initiale, gradient de température,..) tandis que plusieurs paramètres comme le paramètre β ou le terme κ réduisent le cadre d'application de notre algorithme, soit à cause d'une réalité physique, ou bien à cause d'un problème numérique tel que la gestion des zéros dans un ordinateur ou une hypothèse de simplification.

Chapitre 6

Schéma de Runge-Kutta

I. Méthode

On rappelle que la méthode Runge-kutta s'applique à une équation du type

$$y' = f(t, y)$$

$$y(t_0) = y_0 \in \mathbb{R}$$

Pour adapter cette méthode à notre problème on va considérer le vecteur

$$U = \begin{pmatrix} \vdots \\ u \\ \vdots \\ \frac{du}{dt} \\ \vdots \end{pmatrix}$$

et la fonction

$$F : \begin{pmatrix} \vdots \\ u \\ \vdots \\ \frac{du}{dt} \\ \vdots \end{pmatrix} \mapsto \begin{pmatrix} \vdots \\ \frac{du}{dt} \\ \vdots \\ \frac{F(u)}{\rho b} \\ \vdots \end{pmatrix}$$

On a construit $F : \mathbb{R}^{2(N+1)} \rightarrow \mathbb{R}^{2(N+1)}$ de telle sorte à ce qu'elle joue le rôle :

$$\dot{U} = F(U)$$

Comme nous considérons le cas de référence, le temps n'est pas un paramètre de F

et

$$\theta = \frac{a^2}{(a - U_O)^2}$$

Nous pouvons d'ailleurs étendre cette matrice au cas où il y a un terme de dissipation de paramètre κ : dans ce cas B_N devient

$$B_N = -\frac{\kappa}{\rho b} I_N$$

II. Stabilité

Nous savons grâce à la fonction *eig* de MATLAB que les valeurs propres de Ψ sont imaginaire pures dans le cas où il n'y a pas de dissipation. Cependant l'ajout d'un terme de dissipation de paramètre $\kappa > 0$ implique que toutes les valeurs propres de Ψ sont maintenant de partie réelle < 0 .

On peut donc ramener l'étude de la stabilité de la position d'équilibre (statique) du système d'équation à l'étude de la stabilité de la position d'équilibre du système linéarisé.

Le système que l'on étudie est hautement non linéaire mais on peut quand même étudier sa stabilité au sens de Liapounov.

Définition - Stabilité de Liapounov. Considérons l'équation suivante

$$\begin{aligned}\dot{x} &= v(x) \\ v(0) &= 0 \\ x(t_0) = 0 &\Leftrightarrow x \equiv 0\end{aligned}$$

avec $v(0) = 0$ la position d'équilibre et où v est un champ de vecteurs au moins 2 fois différentiable sur un domaine de \mathbb{R}^n . De plus on utilise la norme 2 : $\|x\|^2 = x_1^2 + x_2^2 + \dots + x_n^2$.

La position d'équilibre $x = 0$ de l'équation ci-dessus est dite *stable* au sens de Liapounov si $\forall \epsilon > 0, \exists \delta(\epsilon) > 0$ tel que $\forall x_0$ vérifiant $\|x_0\|^2 < \delta$, la solution x avec la condition initiale $x(0) = x_0$ se prolonge sur le demi-axe $t > 0$ et satisfasse $\|x(t)\| < \epsilon \forall t > 0$.

Définition - Stabilité asymptotique de Liapounov. La position d'équilibre $x = 0$ de l'équation

$$\begin{aligned}\dot{x} &= v(x) \\ v(0) &= 0 \\ x(t_0) = 0 &\Leftrightarrow x \equiv 0\end{aligned}$$

est *asymptotiquement stable* si elle est stable au au sens de Liapounov et si

$$\lim_{t \rightarrow \infty} x(t) = 0$$

pour toute solution x vérifiant la condition initiale $x(0)$ située dans un voisinage assez petit de 0.

Théorème 3. *On considère*

$$\dot{x} = Ax$$

l'équation linéarisée de

$$\begin{aligned}\dot{x} &= v(x) \\ v(0) &= 0 \\ x(t_0) = 0 &\Leftrightarrow x \equiv 0\end{aligned}$$

Si toutes les valeurs propres λ de l'opérateur A sont à partie réelle négative, alors la position d'équilibre $x = 0$ de l'équation ci-dessus est asymptotiquement stable.

Démonstration. Nous posons r^2 une forme quadratique définie positive sur \mathbb{R}^n telle que sa dérivée suivant la direction du champ de vecteur Ax soit positive :

$$L_{Ax}r^2 > 0 \quad \forall x \neq 0$$

pour A un opérateur linéaire de \mathbb{R}^n dans \mathbb{R}^n dont chaque valeur propre a une partie réelle positive. En supposant que toutes les valeurs propres λ_k de A sont positives, on peut écrire :

$$L_{R_{Az}}r^2 > 0 \quad \forall z \neq 0$$

On appellera r^2 *fonction de Liapounov*.

On peut prendre pour r^2 la somme des carrés des modules des coordonnées dans une base complexe :

$$r^2 = (z, \bar{z}) = \sum_{k=1}^n z_k \bar{z}_k$$

et on remarque que la dérivée est une forme quadratique :

$$L_{R_{Az}}(z, \bar{z}) = 2\text{Re}(Az, \bar{z})$$

Nous admettons que dans un voisinage suffisamment petit du point $x = 0$, la dérivée de la fonction de Liapounov suivant la direction du champ non linéaire v vérifie :

$$L_v r^2 \leq -\gamma r^2$$

Reprenons l'équation

$$\begin{aligned}\dot{x} &= v(x) \\ v(0) &= 0 \\ x(t_0) = 0 &\Leftrightarrow x \equiv 0\end{aligned}$$

et définissons Φ une solution non triviale et vérifiant une condition initiale située dans un voisinage assez petit de 0. De plus, définissons la fonction ρ :

$$\rho(t) = \ln(r^2(\Phi(t))) \quad \forall t \geq 0$$

On se place dans le cas non trivial où $r^2(\Phi(t)) \neq 0$. Alors

$$\dot{\rho}(t) = \frac{A}{r^2 \circ \Phi} \frac{d}{dt} r^2 \circ \Phi = \frac{L_v r^2}{r^2} \leq -\gamma$$

Donc $r^2 \circ \Phi$ décroît $\forall t \geq 0$ et tend vers 0 lorsque t tend vers l'infini :

$$\rho(t) \leq \rho(0) - \gamma t$$

$$r^2 \circ \Phi(t) \leq r^2 \circ \Phi(0) e^{-\gamma t} \rightarrow 0$$

□

D'après ce théorème, la stabilité asymptotique de la position d'équilibre pour le système linéarisé est vérifiée et donc la stabilité asymptotique pour le système d'équation

$$\begin{cases} u' = v \\ v' = F(u) - \frac{\kappa}{\rho b} v \end{cases}$$

l'est aussi.

Chapitre 7

Applications

Dans ce chapitre, nous allons présenter des applications concrètes relatives à notre étude que nous avons imaginés.

I. Sécurité d'un pipeline



Il est très important de contrôler la structure d'un pipeline pour ne pas risquer un déversement du fluide véhiculé (pétrole, gaz) dans la nature. Cela passe par la vérification des matériaux, la filtration des fluides et par la surveillance de l'écoulement à l'intérieur du tube. On peut regrouper la plupart de ces prérogatives dans l'analyse de comportement du fluide.

Sur une distance L de tube, nous allons isoler quelques coupes et les considérer comme des cylindres d'épaisseur nulle et de même rayon que le pipeline. On place un dispositif qui permet d'envoyer et de recevoir une onde sur le même axe que la diamètre du tube et on suppose que l'onde traverse le fluide durant la totalité de son parcours. Si nous traçons la courbe de l'onde (en 1D donc), on observe un graphique d'allure similaire à ce qu'on a pu observer dans cette étude. On peut alors rechercher d'éventuels saut, changement de milieu, pente non constante, et autre phénomène qui se voient graphiquement et qui sont repérable d'après nos observations. On peut alors détecter théoriquement un changement un changement de milieu, ce qui pourrait se traduire par un polluant dans le fluide, ou encore une brèche dans le matériau du tube, en observant un changement de vitesse ou de pente.

II. Rideaux de bulles

inspiré par "Propagation d'une onde de choc dans un liquide aéré : modélisation et application aux rideaux de bulles" H. Grandjean



A l'inverse, on peut s'intéresser à l'action d'une onde sur son environnement. Les ondes de choc sous-marines, souvent causées par des explosions, constituent un problème pour la faune aquatique (communication des baleines) ou pour des structures matérielles (navires, bâtiments portuaires). Ces ondes de choc, qui sont des ondes de pressions sous-marines, peuvent désorienter les êtres vivants à proximité et même déchirer une coque de bateau. C'est dans le souci de réduire ces ondes qu'ont été mis en place des rideaux de bulles, un dispositif constitué d'un tube aux multiples trous où l'on fait passer de l'air comprimé. L'onde passant par le rideau est alors atténué (l'amplitude maximale est réduite) On peut alors faire le parallèle avec notre étude sur une onde passant à travers un milieu non homogène. Si l'on peut considérer le système eau-bulle comme une juxtaposition de milieu, on peut surtout considérer le rideau de bulles comme un milieu où la masse volumique est différente, plus élevée d'ailleurs, et où l'onde est atténuée. On observe alors une onde réfléchie et une onde diffractée. Ce dispositif ayant déjà été testé et son utilité approuvée, on imagine qu'on peut observer, lors de l'atténuation de l'onde par le rideau, à une onde réfléchie dans les fonds marins.

Chapitre 8

Le Code

I. C++

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <array>
#include <sstream>
#include <cstring>
#include <cmath>
#include <chrono>
#include "include/matrice.h"           //pour la partie RK

using namespace std;

double lambda(double t)
{
    //lambda est la perturbation initiale
    //dans un premier temps on la prend constante = h/10
    return 0.001;//0.001;
}

void write(double *A, int N, int m, double dt) // criture du vecteur A dans le fichier macro.i
{
    // dt le pas de temps
    string mname("macro.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    cout << "la taille du vecteur est de : " << sizeof(A) << endl ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = 0; i < N; i++)
    {
        output << i*dt << " " << A[i] << endl;
    }
    return;
}

void write2(double *A, int N, int m, double dt) // criture du vecteur A dans le fichier ondes.i
{
    // dt le pas de temps
    string mname("ondes.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    cout << "la taille du vecteur est de : " << sizeof(A) << endl ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = 0; i < N; i++)
    {
        output << i*dt << " " << A[i] << endl;
    }
    return;
}

void write3(double *A, int N, int m, double dt) // criture du vecteur A dans le fichier erreur.i
{
    // dt le pas de temps
    string mname("erreur.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    cout << "la taille du vecteur est de : " << sizeof(A) << endl ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = 0; i < N; i++)
```



```

        {
            output << i*dt << " " << A[i] << endl;
        }
    return;
}

void copy( double *A, double *B, int N)
{
    for (int i = 0; i<N; i++)
    {
        B[i] = A[i];
    }
    return ;
}

void affichage( double*A , int N)
{
    //affichage du r sultat :
    for (int i = 0; i<N ; i++)
    {
        cout << A[i] << " ";
    }
    cout << endl;
    return ;
}

//TEMPERATURE NON CONSTANTE
double temperature(double n, double h, double a, double b, int choix)
{
    //n et h pour valuer la fonction
    // a et b les param tres de la droite affine
    //"choix" le choix de la fonction ou sa d riv e
    if (choix==0)
    {
        return (a*n*h+b);
    }
    else if (choix==1)
    {
        return a;
    }
    else
    {
        //choix par default
        return 0;
    }
}

int exp()
{
    int M= 550;//250; // M = T/dt          M le nombre d'it rations en temps
    int N= 99;//99; // N*dx = L et L = 1    N le nombre de points dans chaque ui
    double R= 8.314 ; //pas utile
    double T= 0.12 ;
    double k= 1.0 ;
    double beta = 1.0 ;
    double h= 0.01 ; // = a+b
    double rho = 1.0;//d pend des cas
    double b= 0.008 ; // =h-a
    double dt= 0.001;
    double a= 0.002 ; // =alpha*h
    double alpha= 0.2; // = a/h
    double s[N] ;
    double t[N] ;
    double u[N] ;
}

```

```

// s = u(m-1)
// t = u(m)
// u = u(m+1)
double U0, V0;

int pause;

//partie onde
double c;
double o1[N];
double o2[N];
double o3[N];

//partie erreur
double err[N];

//partie dissipation
double kappa = 0.285;
double kappa0 = 2.0;
double kappa1 = 1/(N*h);

//Choix de la temperature constante ou non EQUATION: y = px+q
double p, q; //donner la valeur par default
p = 0.04; q = 1.0;
double Konst ; // constante pour simplifier quand temp affine
double K;
int choice;
int choix_temperature;
cout << "*****" << endl;
cout << "Temp rature constante ou affine:" << endl;
cout << "1. Temp rature constante" << endl ;
cout << "2. Temp rature affine" << endl;
cout << "3. Dissipation" << endl;
cout << "*****" << endl;
cin >> choice;
switch(choice)
{
    case 1:
        choix_temperature = 1;
        cout << "Temp rature constante." << endl;
        break;

    case 2:
        choix_temperature = 2;
        cout << "Temp rature affine." << endl;
        cout << "Choisir les coefficients de la droite y=ax+b:" << endl;
        cout << "o a=T'(x)=0.04 et b=T(0)=1 par d faut." << endl;
        cout << "a=" ; cin >> p;
        cout << "b=" ; cin >> q; cout << endl;
        break;

    case 3:
        choix_temperature = 3;
        cout << "Ajout d'un terme de dissipation." << endl;
        cout << "Choisir kappa[0.285]:" << endl;
        cin >> kappa;
        //cout << "Choisir kappa0 et kappa1 : " << endl;
        //cout << "kappa0[2.0] = " ; cin >> kappa0;
        //cout << "kappa1[0.285] = " ; cin >> kappa1;
        break;

    default:
        cerr << "Choix invalide. Temp rature constante par d faut." << endl;
        choix_temperature = 1;
        break;
}

```

```

}

//controle:
cout << "a=" << p << "et b=" << q << endl;
//OK
//cin >> pause;

//choix de la situation:
int choix;
cout << "*****" << endl;
cout << "Choisir le cas d'etude:" << endl;
cout << "1. rho=1 (eau)" << endl;
cout << "2. rho=3 (brome 0 C)" << endl;
cout << "3. rho=0.15 (helium -269 C)" << endl;
cout << "4. rho=0.8 (proule)" << endl;

cout << "*****" << endl;
cin >> choix;
switch(choix)
{
    case 1:
        rho = 1.0;
        break;

    case 2:
        rho = 3.0;
        break;

    case 3:
        rho = 0.15;
        break;

    case 4:
        rho = 0.8;
        break;

    default:
        cerr << "Choix invalide. Lancement de 1 par defaut" << endl;
        break;
}
cout << "choix de rho=" << rho << endl;

//choix de la modification ( homogeneit )
int choix_modif = 1;
cout << "*****" << endl;
cout << "Choisir le cas d'homogeneit:" << endl;
cout << "1. milieu homogene" << endl << "2. La 50eme masse lineique est double" << endl << "3. Les 50 premieres masses lineiques sont triples." << endl;
cout << "*****" << endl;
cin >> choix_modif;
//le reste se passe dans le sch ma

double mu= k*beta*h/(rho*b);

//conditions initiales en temps
int choix_condinit = 1;
cout << "*****" << endl;
cout << "Choisir les conditions initiales:" << endl;
cout << "1. conditions initiales nulles" << endl << "2. pente initiale non nulle" << endl;
cout << "*****" << endl;
cin >> choix_condinit;

switch(choix_condinit)
{
    case 1:

```

```

for (int n=0; n<N; n++)
{
    if(choix_temperature==3)
    {
        //          t[n] = 0.005 ; //u(0,n) //m=1
        //          s[n] = t[n] - 0.1*dt ; //u(-1,n) //m=0
        s[n] = 0.0;
        t[n] = 0.0;//0.001;
    }
    else
    {
        s[n] = 0.0 ; //u(-1,n) //m=0
        t[n] = 0.0 ; //u(0,n) //m=1
    }

    //partie onde
    o1[n] = 0.0 ;
    o2[n] = 0.0 ;

    //partie erreur;
    err[n] = 0.0;

}
break;

case 2:
//choix de U0 et V0 initial:
cout << "Choisir U0 et V0 initial: [V0 positif pour une pente positif et vice-versa]"
<< endl;
cout << "U0 [0.0001] <<<" ; cin >> U0;
cout << "V0 [0.01] <<<" ; cin >> V0; cout << endl;

for (int n=0; n<N; n++)
{
    if(choix_temperature==4)
    {
        t[n] = 0.005 ; //u(0,n) //m=1
        s[n] = t[n] - 0.1*dt ; //u(-1,n) //m=0

    }
    else
    {
        s[n] = U0;
        t[n] = s[n] +dt*V0;
    }

    //partie onde
    o1[n] = U0;
    o2[n] = o1[n] +dt*V0;

    //partie erreur
    err[n] = 0.0;

}
//s[0] = 0.001;
//t[0] = s[0]+ dt*0.01;
break;

default: //case 1
cout << "[Choix invalide. cas 1 par d faut.]" << endl;
for (int n=0; n<N; n++)
{
    if(choix_temperature==3)
    {
        t[n] = 0.005 ; //u(0,n) //m=1

```

```

        s[n] = 0.005 - 0.0001 ; //u(-1,n) //m=0
    }
    else
    {
        s[n] = 0.0 ; //u(-1,n) //m=0
        t[n] = 0.0 ; //u(0,n) //m=1
    }
    //partie onde
    o1[n] = 0.0 ;
    o2[n] = 0.0 ;

    //partie erreur
    err[n] = 0.0;

    }
    break;
}

//choix de l'attenuation (voir fin de boucle temporelle)
double attenuation = 1.0;
cout << "Choisir le pourcentage d'attenuation de la solution" << endl;
cout << "Si pas d'attenuation, taper 1" << endl << ">>";
cin >> attenuation;

//mesure du temps - debut
auto start = std::chrono::high_resolution_clock::now();

//creation de macro.0 et macro.1
write(s, N, 0, h);
write(t, N, 1, h);

//partie onde
write2(o1, N, 0, h);
write2(o2, N, 1, h);

//partie erreur
write3(err, N, 0, h);
write3(err, N, 1, h);

c = k*beta/(rho*(1-alpha)*alpha*alpha) ;
cout << "c=" << c << endl;
// cin >> pause;

for (int m=2; m<M; m++) //boucle sur le temps
{
    // A chaque iteration de temps m, on remplit le vecteur u
    // qui contiendra la fonction u(x) au temps m

    //conditions aux limites en espace
    u[0] = lambda(m*dt) ; //0.001
    u[N] = 0;

    //partie onde
    o3[0] = lambda(m*dt) ;
    o3[N] = 0;

    //partie erreur
    err[0] = 0.0;
    err[N] = 0.0;

    Konst = dt*dt*k*beta/(rho*alpha*(1-alpha));

    for (int n=1; n<N; n++) // boucle sur l'espace
    {
        //u[m+1][n] = 2*u[m][n] - u[m-1][n] + dt*dt*mu

```

```

        *( 1/(a+u[m][n]-u[m][n-1]) - 1/(a+u[m][n+1]-u[m][n]) );
if (choix_modif==2 && n==50)
{
    mu = mu/2;
    if (choix_temperature==1)
    {
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    else if (choix_temperature==2)
    {
        cout << "YOUPI" << endl;
        //K = k*beta*h*h/(rho*alpha*alpha*(1-alpha));
        //u[n] = 2*t[n] - s[n] + K
            *( temperature(n,h,p,q,0)*(t[n+1]-2*t[n]+t[n-1])/(dt*dt)
                - 2*temperature(n,h,p,q,1) );
        //Konst = dt*dt*k*beta/(rho*alpha*(1-alpha));
        Konst = Konst/2;
        u[n] = 2*t[n] - s[n] + (t[n+1]-2*t[n]+t[n-1])*Konst
            *temperature(n,h,p,q,0)/(alpha*h*h)
            - 2*temperature(n,h,p,q,1)*Konst ;
        Konst = 2*Konst;

        //partie onde
    }
    else if (choix_temperature==3)
    {
        rho=rho/2;
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) )
            - (dt*kappa/(rho*b))*(t[n]-s[n]);
        //u[n] = 2*t[n]*(1-kappa*dt/(2*rho*b))
            + s[n]*(kappa*dt/(rho*b) -1)
            + (dt*dt*beta*h/(rho*b))*( 1/(a+t[n]-t[n-1])
                - 1/(a+t[n+1]-t[n]) );

        rho=2*rho;
    }
    else
    {
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    mu = 2*mu;
}
else if (choix_modif==3 && n<51)
{
    mu = mu/3;
    if (choix_temperature==1)
    {
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    else if (choix_temperature==2)
    {
        cout << "O1lyuO1lyuOxenFree" << endl;
        //K = k*beta*h*h/(rho*alpha*alpha*(1-alpha));
        //u[n] = 2*t[n] - s[n] + K*( temperature(n,h,p,q,0)
            *(t[n+1]-2*t[n]+t[n-1])/(dt*dt)
                - 2*temperature(n,h,p,q,1) );
        //Konst = dt*dt*k*beta/(rho*alpha*(1-alpha));
        Konst = Konst/3;
        u[n] = 2*t[n] - s[n] + (t[n+1]-2*t[n]+t[n-1])*Konst
            *temperature(n,h,p,q,0)/(alpha*h*h)
            - 2*temperature(n,h,p,q,1)*Konst ;
        Konst = 3*Konst;
    }
}

```

```

        //partie onde
    }
    else if (choix_temperature==3)
    {
        rho=rho/3;
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) ) - (dt*kappa/(rho*b))*(t[n]-s[n]);
        //u[n] = 2*t[n]*(1-kappa*dt/(2*rho*b))
            + s[n]*(kappa*dt/(rho*b) -1)
            + (dt*dt*beta*h/(rho*b))*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
        rho=3*rho;
    }
    else
    {
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    mu = mu*3;
}
else
{
    if (choix_temperature==1)
    {
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    else if (choix_temperature==2)
    {
        //K = k*beta*h*h/(rho*alpha*alpha*(1-alpha));
        //u[n] = 2*t[n] - s[n] + K
            *( temperature(n,h,p,q,0)*(t[n+1]-2*t[n]+t[n-1])/(dt*dt)
            - 2*temperature(n,h,p,q,1) );
        //Konst = dt*dt*k*beta/(rho*alpha*(1-alpha));
        cout << "YOUPI" << endl;
        u[n] = 2*t[n] - s[n] + (t[n+1]-2*t[n]+t[n-1])*Konst
            *temperature(n,h,p,q,0)/(alpha*h*h)
            - 2*temperature(n,h,p,q,1)*Konst ;
        //partie onde
    }
    else if (choix_temperature==3)
    {
        //cout << " OH YEAH " << endl;
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) ) - (dt*kappa/(rho*b))*(t[n]-s[n]);
        //u[n] = 2*t[n]*(1-kappa*dt/(2*rho*b))
            + s[n]*(kappa*dt/(rho*b) -1) + (dt*dt*beta*h/(rho*b))
            *( 1/(a+t[n]-t[n-1])
            - 1/(a+t[n+1]-t[n]) );
    }
    else
    {
        //double mu= k*beta*h/(rho*b);
        u[n] = 2*t[n] - s[n] + dt*dt*mu*( 1/(a+t[n]-t[n-1]) -
            1/(a+t[n+1]-t[n]) );
    }
}

//partie onde
if(choix_modif==2 && n==50)
{
    c=c/2;
    if (choix_temperature==1)
    {

```

```

        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
    else if(choix_temperature==2)
    {
        c = k*beta*temperature(n,h,p,q,0)/(rho*alpha*alpha*(1-alpha));
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
        c = k*beta/(rho*(1-alpha)*alpha*alpha) ;
    }
    else
    {
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
    c=2*c;
}
else if(choix_modif==3 && n<51)
{
    c=c/3;
    if (choix_temperature==1)
    {
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
    else if(choix_temperature==2)
    {
        c = k*beta*temperature(n,h,p,q,0)/(rho*alpha*alpha*(1-alpha));
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
        c = k*beta/(rho*(1-alpha)*alpha*alpha) ;
    }
    else
    {
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
    c=c*3;
}
else
{
    if (choix_temperature==1)
    {
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
    else if(choix_temperature==2)
    {
        c = k*beta*temperature(n,h,p,q,0)/(rho*alpha*alpha*(1-alpha));
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
        c = k*beta/(rho*(1-alpha)*alpha*alpha) ;
    }
    else
    {
        o3[n] = 2*o2[n] - o1[n] + (o2[n+1] -2*o2[n]
            +o2[n-1])*c*dt*dt/(h*h);
    }
}

//partie erreur
err[n] = abs( u[n] - o3[n+2] ) * ( abs( u[n] - o3[n+2] ) < 0.0003 ) ;

//partie att nuation de o3 et/ou u
o3[n] = attenuation*o3[n];

```



```

        u[n] = attenuation*u[n];

        //pour le test
        //cout << "x = " << n*h << " T = " << temperature(n,h,p,q,0)
        << "et T'=" << temperature(n,h,p,q,1) << endl;
    }

    //pivotage :
    copy(t,s, N); //s <- t;
    copy(u,t, N); //t <- u;

    //sauvegarde des u l'iteration m+1 dans macro.m
    write(u, N, m, h);

    //affichage pour test
    cout << endl << "tps=" << m << endl;
    cout << "u=";
    affichage(u, N);

    cout << "m*dt=" << m*dt << endl;

    //partiesondes
    copy(o2,o1,N);
    copy(o3,o2,N);
    write2(o3, N, m, h);
    cout << "o3=";
    affichage(o3,N);

    //partie erreur
    write3(err, N, m, h);
}

//mesure du temps - fin
auto elapsed = std::chrono::high_resolution_clock::now() - start;

//affichage du temps
long long microseconds = std::chrono::duration_cast<std::chrono::microseconds>(elapsed).count();
cout << endl << "temps d'execution=" << microseconds << endl;

//pour le test
//double phi = k*beta*h*h/(rho*alpha*alpha*(1-alpha));
//cout << " K = " << phi;

cout << "LES CHOIX PRIS" << endl;
cout << "choix_modif=" << choix_modif << endl;
cout << "choix_temperature=" << choix_temperature <<endl;

return 0;
}

*/ //RUNGE KUTTA 4
void F_RG(double* uBig, double* k1, int N, double k,
          double beta, double h, double rho, double b, double a)
{
    //Calcule F(uBig) et range le resultat dans k1
    //tout le calcul se fait dans un temporaire

    double* temp;
    temp = new double[2*N];

    //Calcul
    double Konst = k*beta*h/(rho*b);
    //iter 0

```

```

temp[0] = 0.001; //uBig[N];
temp[N] = 1; //Konst*( 1/(a+uBig[0]) - 1/(a+uBig[1]-uBig[0]) );
//iter N-1
temp[N-1] = 0; //uBig[2*N-1];
temp[2*N-1] = -temp[2*N-2]; //Konst*( 1/(a+uBig[N-1]-uBig[N-2]) - 1/(a-uBig[N-1]) );
//int rieur de k1
for (int n=1; n<N-1; n++)
{
    temp[n] = uBig[n+N];
    temp[n+N] = Konst*( 1/(a+uBig[n]-uBig[n-1]) - 1/(a+uBig[n+1]-uBig[n]) );
}
//On range le r sultat dans k1 et on d truit le temp
memcpy( k1, temp, 2*N*sizeof(double) );
delete [] temp;
}

void mult(double* ki, double* temp, double h, int N, double facteur)
{
    for(int m=0;m<2*N;m++)
    {
        temp[m] = ki[m]*h*facteur;
    }
}

void add(double* uBig, double* temp, int N)
{
    for(int m=0;m<2*N;m++)
    {
        temp[m] = uBig[m] + temp[m];
    }
}

void RKiter(double* res, double* uBig, double* k1, double* k2,
double* k3, double* k4, double N, double h)
{
    for(int m=0;m<2*N;m++)
    {
        res[m] = uBig[m] + (h/6)*(k1[m]+2*k2[m]+2*k3[m]+k4[m]);
    }
}

void writeBig(double *uBig, int N, int m, double dt)
{
    // dt le pas de temps
    string mname("RKplot.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    //cout << "la taille du vecteur est de : " << sizeof(uBig) << endl ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = 0; i < N; i++)
    {
        output << i*dt << "␣" << uBig[i] << endl;
    }
    return;
}

int expRG()
{
    //param tres
    int M = 250; //nb iter en temps
    int N = 99; //nb iter en espace
    double dt= 0.001; // pas en temps
    double h = 0.01; //pas en espace
    double k = 1.0 ;
    double beta = 1.0;
}

```

```

double rho = 1.0;
double b = 0.008;
double a = 0.002;
double U0 = 0.0001; //param init
double V0 = 0.01; //param init
int iter = 0; // control of iteration

//definition des vecteurs solutions
double* uBig;

//definition des objets RK4
double* k1;
double* k2;
double* k3;
double* k4;
double* k1bis;
double* k2bis;
double* k3bis;
double* k4bis;
double* temp;
double* res;
//et allocation dynamique
k1 = new double[2*N];
k2 = new double[2*N];
k3 = new double[2*N];
k4 = new double[2*N];
k1bis = new double[2*N];
k2bis = new double[2*N];
k3bis = new double[2*N];
k4bis = new double[2*N];
temp = new double[2*N];
res = new double[2*N];
uBig = new double[2*N];

//initialisation de uBig
for (int n=0; n<N; n++)
{
    uBig[n] = U0;
    uBig[n+N] = V0;
}

// criture de l'it ration 0
writeBig(uBig,N,0,h);

//pour le test
int pause;
cout << "pause_1" << endl;
cin >> pause;

//it ration 1 M
for(int m=0; m<M; m++)
{
    cout << "pause_2" << endl;
    //cin >> pause;

    //construction de k1 et sauvegarde dans k1bis
    F_RG(uBig, k1, N, k, beta, h, rho, b, a);
    memcpy( k1bis, k1, 2*N*sizeof(double) );
    //k1bis = k1;

    cout << "pause_3_affichage_k1" << endl;
    affichage(k1,2*N);
    //cin >> pause;

    //construction de k2 et sauvegarde dans k2bis

```

```

    mult(k1, temp, h, N, 0.5); //temp contient k1*h/2
    add(uBig, temp, N); //temp contient k1*h/2 + uBig
    F_RG(temp, k2, N, k, beta, h, rho, b, a); //construction de k2 via temp
    memcpy( k2bis, k2, 2*N*sizeof(double) );
    //k2bis = k2; //sauvegarde de k2

    cout << "pause_4" << endl;
    //cin >> pause;

    //construction de k3 et sauvegarde dans k3bis
    mult(k2, temp, h, N, 0.5); //temp contient k2*h/2
    add(uBig, temp, N); //temp contient k2*h/2 + uBig
    F_RG(temp, k3, N, k, beta, h, rho, b, a); //construction de k3 via temp
    memcpy( k3bis, k3, 2*N*sizeof(double) );
    //k3bis = k3; //sauvegarde de k3

    cout << "pause_5" << endl;
    //cin >> pause;

    //construction de k4 et sauvegarde dans k4bis
    mult(k3, temp, h, N, 1.0); //temp contient k3*h
    add(uBig, temp, N); //temp contient k3*h + uBig
    F_RG(temp, k4, N, k, beta, h, rho, b, a); //construction de k4 via temp
    memcpy( k4bis, k4, 2*N*sizeof(double) );
    //k4bis = k4; //sauvegarde de k4

    cout << "pause_6" << endl;
    //cin >> pause;

    //construction du nouveau uBig depuis res
    RKiter(res, uBig, k1bis, k2bis, k3bis, k4bis, N, h);
    memcpy( uBig, res, 2*N*sizeof(double));

    cout << "pause_7" << endl;
    //cin >> pause;

    //affichage du nouveau uBig, qui contient u et v
    affichage(uBig,2*N);

    cout << "pause_8" << endl;
    //cin >> pause;

    // criture de l'iteration m+1
    writeBig(uBig,N,m+1,h);

    cout << "ITERATION:_:" << iter++ << endl;
};

//d sallocation
delete [] k1;
delete [] k2;
delete [] k3;
delete [] k4;
delete [] k1bis;
delete [] k2bis;
delete [] k3bis;
delete [] k4bis;
delete [] temp;
delete [] res;
delete [] uBig;

system("gnuplot_RK.gnuplot;_rm_RKplot.*;");

return 0;
}

```

```

vecteur F_RK(vecteur uBig, matrice PSI, int N)
{
    vecteur res(2*N);
    res = uBig;
    return PSI*res;
}

void writeRK(vecteur A, int N, int m, double dt)
{
    string mname("RKplot.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = 0; i < N; i++)
    {
        output << i*dt << " " << A.p[i] << endl;
    }
    //output.close();
    return;
}

void writeRKvit(vecteur A, int N, int m, double dt)
{
    string mname("RKplotvit.");
    ostringstream ost ; ost << m ;
    mname += ost.str() ;
    ofstream output(mname.c_str(), ios::out);
    for (int i = N; i < 2*N; i++)
    {
        output << (i-N)*dt << " " << A.p[i] << endl;
    }
    //output.close();
    return;
}

int expRK()
{
    //param tres
    int M = 250; //nb iter en temps
    int N = 9; //nb iter en espace
    double dt= 0.001; // pas en temps
    double h = 0.01; //pas en espace
    double k = 1.0 ;
    double beta = 1.0;
    double rho = 1.0;
    double b = 0.008;
    double a = 0.002;
    double U0 = 0.0005;//0.0001; //param init
    double V0 = 0.1;//0.01; //param init
    int iter = 0; // control of iteration
    int pause;

    //special RK
    double X = k*beta*h/(rho*b*a*a);
    double theta = a*a/( (a-U0)*(a-U0) );
    double kappa = 0.0;

    //definition des objets RK4
    vecteur uBig(2*N);
    vecteur k1(2*N);
    vecteur k2(2*N);
    vecteur k3(2*N);

```

```

vecteur k4(2*N);
vecteur k1bis(2*N);
vecteur k2bis(2*N);
vecteur k3bis(2*N);
vecteur k4bis(2*N);
vecteur temp(2*N);
vecteur res(2*N);

//creation de la matrice phi
matrice PSI(2*N,2*N); //on consid ere A(i,j) et on remplit ligne par ligne
for(int i=0;i<N;i++) //Partie sup rieure
{
    PSI(i,N+i) = 1;
}
PSI(N,0) = X*(-1-theta); //N- me ligne
PSI(N,1) = X;
PSI(N,N) = -kappa/(rho*b);
for(int i=0;i<N-2;i++) //Int rieur de la Partie inf rieur e
{
    PSI(N+1+i,i) = X;
    PSI(N+1+i,i+1) = -2*X;
    PSI(N+1+i,i+2) = X;
    PSI(N+1+i,N+1+i) = -kappa/(rho*b);
}
PSI(2*N-1,N-2) = X; //2N-1- me ligne
PSI(2*N-1,N-1) = -2*X;
PSI(2*N-1,2*N-1) = -kappa/(rho*b);
cout << PSI;

cin >>pause;

//initialisation de uBig initial
// uBig(0) = U0;
// uBig(N) = V0;
for(int i=0; i<N;i++)
{
    uBig(i) = U0;
    uBig(i+N) = V0;
}

// criture de u0 et v0
writeRK(uBig,N,0,h);
writeRKvit(uBig,N,0,h);

// cin >> pause;

//it ration 1 M
for(int m=1; m<M; m++)
{
    cout << "┐ITERATION┐:┐" << m << endl;

    //creation et sauvegarde de k1
    k1 = F_RK(uBig, PSI, N);
    k1bis = k1;

    //creation et sauvegarde de k2
    k2 = F_RK(uBig + (h/2)*k1, PSI, N);
    k2bis = k2;

    //creation et sauvegarde de k3
    k3 = F_RK(uBig + (h/2)*k2, PSI, N);
    k3bis = k3;

    //creation et sauvegarde de k4
    k4 = F_RK(uBig + h*k3, PSI, N);

```

```

        k4bis = k4;

        //creation et criture du nouveau uBig
        res = uBig + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
        uBig = res ;
        cout << "vecteur uBig:" << uBig << endl;
        writeRK(uBig,N,m,h);
        writeRKvit(uBig,N,m,h);
};

//system("gnuplot RK.gnuplot ; rm RKplot.*;");

//pour le test
cout << "Programme lanc avec succ s! Youpi!" << endl;

system("gnuplot RK.gnuplot; rm RKplot.*; rm RKplotvit.*;");

return 0;
}

int main()
{
    cout << "Choisir quel thode utiliser" << endl;
    cout << "1. Diff rences Finies" << endl;
    cout << "2. Runge-Kutta4-m thode 1" << endl;
    cout << "3. Runge-Kutta4-m thode 2" << endl;
    cout << ">>";
    int choix = 1;
    cin >> choix;
    switch(choix)
    {
        case 1:
            exp();
            break;

        case 2:
            expRG();
            break;

        case 3:
            expRK();
            break;

        default:
            exp();
            break;
    }
    return 0;
}

```

II. Maple

a. Code Source

```

> rho := 1 ; N := 99 ; L := 1.0 ; h := 10-2 ; alpha := 0.2 ; beta := 0.2 ; k := 1.0 ; c(x)
  := sqrt( ( k·beta·T(x) ) / ( rho·α2·(1 - alpha) ) );

=
> eq1 := v(x, t) = diff(u(x, t), t) :
  eq2 := diff(v(x, t), t)·rho·(1 - alpha) = ( k·beta·diff(u(x, t), x, x) ) / ( alpha + diff(u(x, t), x) )2 :

=
> eq := subs(eq1, eq2) :
  pdsolve(eq, u);

=
> eq3 := diff(v(x, t), t)·rho·(1 - alpha) = ( k·beta·diff(u(x, t), x, x) ) / α2 :

=
> ondes := subs(eq1, eq3) :
  pdsolve(ondes, u);
> pdsolve(diff(u(x, t), t) = c2·diff(u(x, t), x, x), u);
>
>
>
>
> restart :
  nb_points := 100;
  u := array(1..nb_points) :
> mu := ( k·beta·h ) / ( rho·b );
  F := (u, i) → ( mu / ( a + u[i] - u[i - 1] ) - mu / ( a + u[i + 1] - u[i] ) );
  F( u, ( nb_points ) / 2 );

=
> L := 1. ;
  h := L / nb_points :
  alpha := 0.2 :
  beta := 0.2 :
  k := 1 :
  deltat := 0.01 :
  assign( solve( { h = a + b, alpha = a / h } ) );
  rho := ( nb_points ) / L ;
  mu := ( k·beta·h ) / ( rho·b );
  u := [ h / 10, seq(0, i = 2..nb_points) ] :

=
> nb_t := -1 :
  ut[nb_t] := u :
  nb_t := nb_t + 1 :
  ut[nb_t] := u :

```



```

> nb_pas := 600 ;
  for pas to nb_pas do
    ut[nb_t + 1] := 2·ut[nb_t] - ut[nb_t - 1] + [0, seq(F(ut[nb_t], i)·delta2, i = 2 ..nb_points
      - 1), 0] ;
    nb_t := nb_t + 1 ;
  od ;
  nb_t :

> plot( [seq( [seq( [i, ut[40·t][i]], i = 1 ..nb_points) ], t = 6 ..10) ] ) ;

> plot( [seq( [seq( [i, ut[100·t][i]], i = 1 ..nb_points) ], t = 3 ..6) ] ) ;

> mu :=  $\frac{k \cdot \text{beta} \cdot h}{\text{rho} \cdot b}$  ;
  F := (u, i) →  $\frac{\text{mu}}{a + u[i] - u[i - 1]} - \frac{\text{mu}}{a + u[i + 1] - u[i]}$  ;
  F( $u, \frac{\text{nb\_points}}{2}$ ) :

> Fm := (u, i) → if i = 50 then  $\frac{F(u, i)}{2}$  else F(u, i) fi:

> u := [ $\frac{h}{10}, \text{seq}(0, i = 2 ..\text{nb\_points})$ ] ;
  nb_t := -1 ;
  ut[nb_t] := u ;
  nb_t := nb_t + 1 ;
  ut[nb_t] := u ;

> eval(seq(Fm(u, j), j = 2 ..nb_points - 1)) :

> nb_pas := 300 ;
  for pas to nb_pas do
    ut[nb_t + 1] := 2·ut[nb_t] - ut[nb_t - 1] + [0, seq(Fm(ut[nb_t], i)·delta2, i = 2 ..nb_points
      - 1), 0] ;
    nb_t := nb_t + 1 ;
  od ;
  nb_t :

> plot( [seq( [i, ut[300][i]], i = 1 ..nb_points) ] ) ;

> plot( [seq( [seq( [i, ut[100·t][i]], i = 1 ..nb_points) ], t = 1 ..3) ] ) ;

```

b. Graphiques

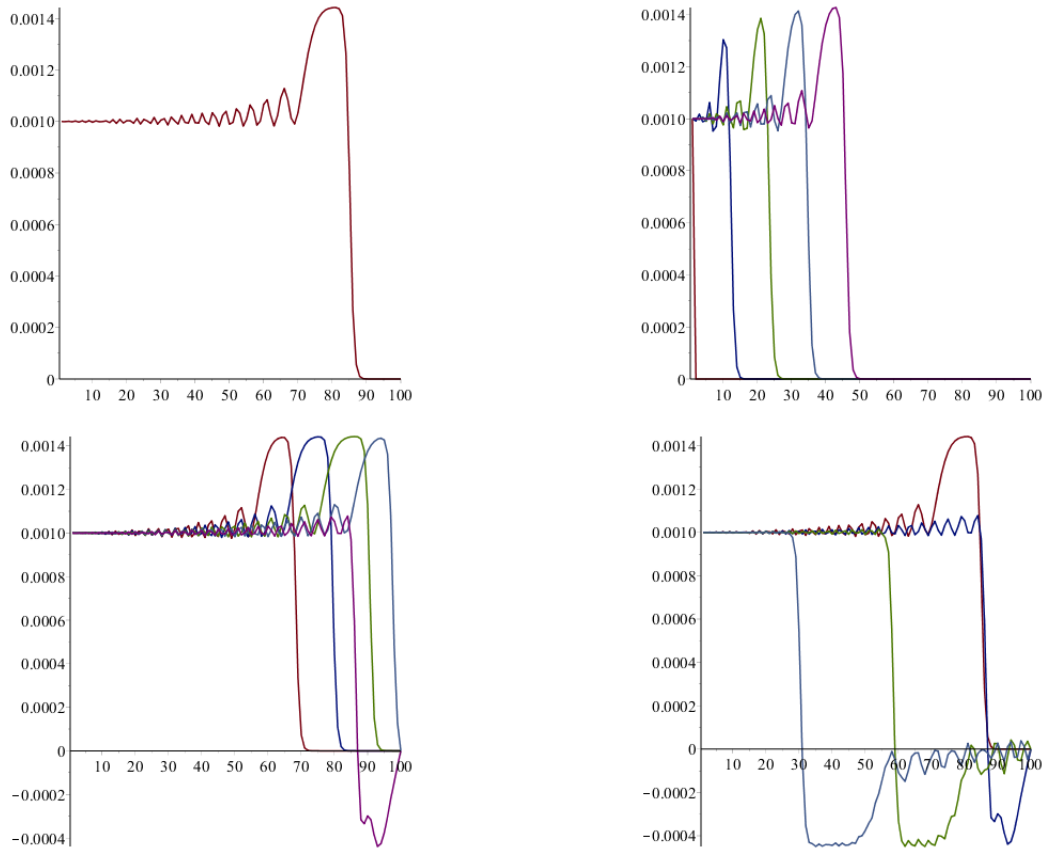


FIGURE 8.1 – Cas de référence

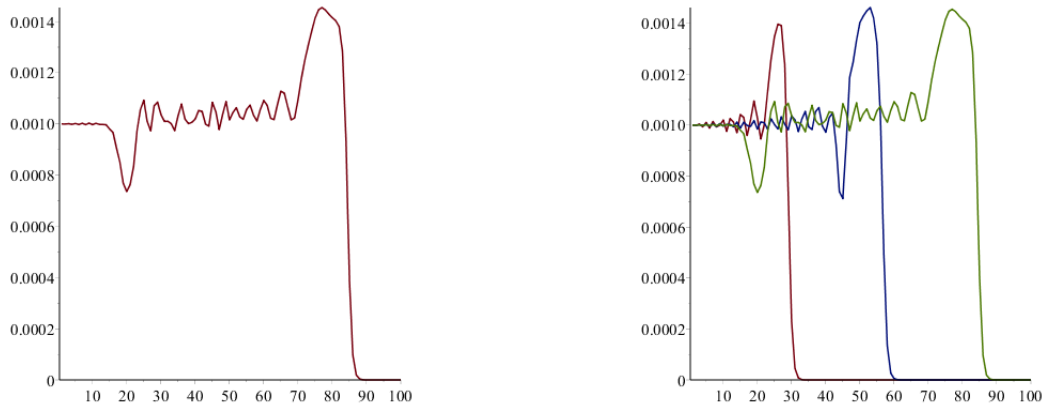


FIGURE 8.2 – Segment du centre deux fois plus lourd

Bibliographie

- [1] *Traitements Thermiques des Dechets*. Ed. Techniques Ingénieur.
- [2] Bertrand Maury. *Modélisation mathématique : Un autre regard*. Ed. Alain Lichnewsky, 2001.
- [3] V. Arnold. *Equations Différentielles Ordinaires*. MIR, 1974.
- [4] Michelle Schatzman. *Numerical Analysis : A Mathematical Introduction*. Oxford University Press, 2002.
- [5] Phénomènes physiques. <https://sites.google.com/site/sfagabe/themes/acoustique-de-l-environnement/propagation-acoustique-en-milieu-exterieur/phenomenes-physiques>.
- [6] Simulation en mécanique des fluides numériques. <http://users.aims.ac.za/~doriano/research/masters/fluidsimul.pdf>.
- [7] Olivier Granier. Étude des ondes sonores dans les fluides. http://olivier.granier.free.fr/ci/fluides/co/CI_Ondes-sonores_web.html.
- [8] Hervé Grandjean. Propagation d'une onde de choc dans un liquide aéré : modélisation et application aux rideaux de bulles. <https://tel.archives-ouvertes.fr/tel-00776609v2/document>.