



Rapport de stage : Schema adaptatif entre les modèles de KDV-BBM et de BURGERS

JEZEQUEL Yohann

Entreprise : Inria Bordeaux

Tuteur universitaire : BESSEMOULIN Marianne

Tuteur entreprise : KAZOLEA Maria, PARISOT Martin



Avril - Septembre 2023

Table des matières

1	Introduction	2
1.1	Problématique et objectifs	2
1.2	Introduction du modèle hyperbolique avec contraintes	2
2	Problème discret	4
2.1	Discrétisation et décomposition du problème	4
2.2	Résolution du problème hyperbolique	5
2.3	Résolution du problème de projection	6
2.4	Analyse paramétrique de la méthode	7
3	Projection sur un domaine réduit : approche itérative	11
3.1	Estimation de l'erreur et domaine de correction	11
3.2	Approche itérative	12
3.2.1	Etude d'un cas à vitesse de choc nulle	12
3.2.2	Etude d'un cas à vitesse de choc positive	17
3.3	Erreur et temps de calcul de la méthode itérative	18
4	Approche explicite du domaine optimal	19
4.1	Opérateurs nécessaires pour une approche directe du domaine	19
4.2	Présentation de méthodes explicites appuyées sur de la régression linéaire	19
4.2.1	Présentation des opérateurs, et estimation d'erreur	19
4.3	Méthode d'approche explicite par régression linéaire	20
4.3.1	Méthode 1 : $\mathcal{D} = [\bar{x}_\tau + a_1 \bar{\sigma}_\tau; \bar{x}_\tau + a_2 \bar{\sigma}_\tau]$	20
4.3.2	Méthode 2 : $\mathcal{D} = [b_1 \bar{x}_\tau + a_1 \bar{\sigma}_\tau, b_2 \bar{x}_\tau + a_2 \bar{\sigma}_\tau]$	23
4.4	Méthode par majoration : $\mathcal{D} = [\bar{x}_\tau - a_{sup} \bar{\sigma}_\tau; \bar{x}_\tau + a_{sup} \bar{\sigma}_\tau]$	26
4.5	Temps de calcul de la méthode explicite par majoration	28
5	Conclusion	29

1 Introduction

1.1 Problématique et objectifs

Dans le domaine de l'analyse numérique, on doit toujours faire un choix entre la précision d'une approximation et son temps de calcul. Ce choix se fait en fonction des conditions d'utilisations. Un modèle rapide à calculer est généralement une forme simplifiée d'un problème plus général, ou plus complexe.

Afin de calculer plus rapidement un problème complexe avec un temps de calcul long, le but de ce rapport est de développer une méthode permettant de réduire le temps de calcul d'un modèle complexe, en s'appuyant sur la résolution d'un problème plus rapide à résoudre numériquement.

L'utilisation d'une telle méthode peut par exemple servir dans le domaine de la sécurité pour les risques côtiers, où l'on a besoin de résultats rapides, avec des estimations les plus précises possible pour agir avec le plus d'efficacité.

1.2 Introduction du modèle hyperbolique avec contraintes

On se place dans un espace $\Omega \times [0, T]$ avec $\Omega = [0, L] \subseteq \mathbb{R}$ et $T \geq 0$.

Le problème se décompose en trois éléments, présentés de la manière suivante :

$$\begin{cases} \partial_t U + A(U) \partial_x U &= -\phi, \\ \langle U, \phi \rangle &= 0, \\ L(U) &= 0. \end{cases} \quad (1)$$

Dans notre cas, l'ensemble des fonctions étudiées sont dans $L^\infty([0, T], L^2(\Omega))$.

Dans le cadre de notre observation, on pose $U = \begin{pmatrix} u \\ w \end{pmatrix}$, $A(U) = \begin{pmatrix} u & 0 \\ 0 & c_2 \end{pmatrix}$. Le système hyperbolique est composé d'une équation de Burgers sur u , et d'une équation de transport linéaire de vitesse constante c_2 sur w , avec un second membre $\phi = \begin{pmatrix} \phi_1(t, x) \\ \phi_2(t, x) \end{pmatrix}$.

Le produit scalaire employé est $\langle \cdot \rangle_{L^2(\Omega)}$, défini par

$$\langle U, \phi \rangle_{L^2(\Omega)} = \int_{\Omega} (u\phi_1 + w\phi_2) dx \quad (2)$$

et notre fonction de contraintes entre u et w est définie par $L(U)$, tel que :

$$L(U) = w + \alpha \partial_x u = 0 \quad (3)$$

avec α une constante.

On note également pour conditions aux bords :

$$\begin{cases} u(t, 0) &= u_L, \\ u(t, L) &= u_R, \\ u\phi_2 &= 0 \text{ p.p sur } \partial\Omega. \end{cases}$$

La mise en place de ces éléments permet d'affirmer que si U est solution de (1), alors u est solution de l'équation BBM-KdV (aussi appelée équation de Korteweg-de Vries) :

$$\partial_t u + u \partial_x u - \alpha^2 \partial_{xxt} u - c_2 \alpha^2 \partial_{xxx} u = 0 \quad (4)$$

Cette équation est utilisée pour calculer la propagation de vagues dans des milieux à faible profondeur.

Preuve:

Soit $u(x, t)$, $w(x, t)$ ϕ_1 , ϕ_2 des fonctions de $L^\infty([0, T], L^2(\Omega))$

Par définition du produit scalaire, $\forall(u, w)$ vérifiant (1) :

$$\int_{\Omega} (u\phi_1 + w\phi_2) dx = 0.$$

En utilisant (3) :

$$\begin{aligned} \int_{\Omega} (u\phi_1 - \alpha\partial_x u\phi_2) dx &= 0, \\ \int_{\Omega} (u(\phi_1 + \alpha\partial_x \phi_2)) dx - \int_{\partial\Omega} u\phi_2 \partial\sigma &= 0. \end{aligned}$$

On a par condition aux bords :

$$\int_{\partial\Omega} u\phi_2 \partial\sigma = 0.$$

Ce qui permet d'établir l'égalité :

$$\phi_1 = -\alpha\partial_x \phi_2 \text{ sur } \Omega \tag{5}$$

En reprenant la première équation du problème, on obtient :

$$\begin{aligned} \partial_t u + u\partial_x u &= \alpha\partial_x \phi_2, \\ \partial_t u + u\partial_x u &= -\alpha\partial_x (\partial_t w + c_2\partial_x w), \\ \partial_t u + u\partial_x u &= \alpha^2\partial_x (\partial_{tx} u + c_2\partial_{xx} u). \end{aligned}$$

Ce qui donne la formule (4)

■

Réécrire l'équation de BBM-KdV sous la forme (1) plutôt que sous la forme (4) va permettre d'approcher numériquement une solution à un problème de BBM-KdV, avec un temps de calcul plus court, ce qui est le thème de ce rapport.

2 Problème discret

2.1 Discrétisation et décomposition du problème

Considérons une discrétisation de l'espace $\Omega \times [0, T]$.
Soit $(t^n)_{n \in \mathbb{N}}$, tel que $t^0 = 0$ et $t^{n+1} = t^n + \delta_t^n$, les points de discrétisation en temps et $(\omega_i)_{i \in \{1, \dots, p\}}$ une discrétisation uniforme de Ω .

On note $\delta_x = |\omega_{i+1} - \omega_i|$. A partir de la suite $(\omega_i)_{i \in \{1, \dots, p\}}$, on définit les volumes de contrôle $(x_i)_{i \in \{1, \dots, p\}}$ tel que $x_i = [\omega_i - \frac{\delta_x}{2}, \omega_i + \frac{\delta_x}{2}]$

On note p le nombre de cellules issues de la discretisation de Ω

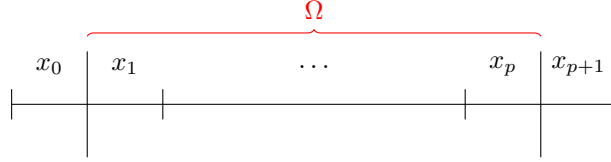


FIGURE 1 – Discrétisation de Ω

La figure 1 illustre le découpage de Ω en p cellules.

Dans cette partie, on note $\partial\Omega$ les cellules frontalières à Ω . On définit $\partial\Omega$ par deux cellules x_0 et x_{p+1} i.e $\partial\Omega = \{x_0, x_{p+1}\}$.

On pose $U_i^n = \begin{pmatrix} u_i^n \\ w_i^n \end{pmatrix}$ avec :

$$u_i^n \approx \frac{1}{\delta_x} \int_{x_i} u(t^n, x) dx$$

$$w_i^n \approx \frac{1}{\delta_x} \int_{x_i} w(t^n, x) dx$$

L'algorithme de résolution de l'ensemble de (1) est un algorithme de résolution implicite qui se décompose en deux étapes, présentes sur chaque pas de temps.

Pour illustrer ces étapes, on se donne une estimation de U au temps t^n , que l'on notera U^n , le but de la démarche est alors de trouver U^{n+1} .

Etape 1 :

En partant de U^n , le but est de trouver $U^* = \begin{pmatrix} u_i^* \\ w_i^* \end{pmatrix}$, solution du problème :

$$\begin{cases} \partial_t u + u \partial_x u &= 0, \\ \partial_t w + c_2 \partial_x w &= 0, \end{cases} \quad (6)$$

ce qui correspond, d'un point de vue numérique, à poser :

$$\begin{cases} u_i^* &= u_i^n - \frac{\delta_x}{\delta_t^n} (\mathcal{F}_{i+\frac{1}{2}}^u - \mathcal{F}_{i-\frac{1}{2}}^u), \\ w_i^* &= w_i^n - \frac{\delta_x}{\delta_t^n} (\mathcal{F}_{i+\frac{1}{2}}^w - \mathcal{F}_{i-\frac{1}{2}}^w), \end{cases}$$

avec \mathcal{F} un flux numérique.

La méthode employée sera explicitée dans la section 2.2

Etape 2 :

Le but de la seconde méthode consiste à trouver U^{n+1} , en partant d'une projection de la solution U^* sur l'ensemble \mathbb{A} défini par :

$$\mathbb{A} := \left\{ U = \begin{pmatrix} u \\ w \end{pmatrix} \in L^\infty([0, T], L^2(\Omega)) ; L(U) = 0 \right\}.$$

D'un point de vue numérique, l'expression se traduit par :

$$\mathcal{P}(U^{*,n}) = \begin{cases} u^{n+1} & = u^* - \delta_t^n \phi_1^{n+1}. \\ w^{n+1} & = w^* - \delta_t^n \phi_2^{n+1}. \\ L(U^{n+1}) & = 0. \\ \langle U^{n+1}, \phi^{n+1} \rangle_{L^2(\Omega)} & = 0. \end{cases} \quad (7)$$

Plus de précisions sur la seconde étape seront apportées dans la section 2.3.

2.2 Résolution du problème hyperbolique

Partons de l'équation (6), et rappelons la forme discrète de ce problème hyperbolique :

$$U_i^{*,n} = U_i^n - \frac{\delta_x}{\delta_t^n} (\mathcal{F}_{i+\frac{1}{2}} - \mathcal{F}_{i-\frac{1}{2}})$$

avec pour conditions initiales

$$\begin{aligned} u_i^0 &= u^0(x_i) \\ w_i^0 &= w^0(x_i) \end{aligned}$$

Avec $\mathcal{F}_{i+\frac{1}{2}}$ le flux numérique entre les cellules x_i et x_{i+1} sur les variables U_i^n et U_{i+1}^n , dans nos observations, nous utiliserons un flux upwind, propre à chaque variable observée.

$$\mathcal{F}_{i+\frac{1}{2}} = \begin{pmatrix} \mathcal{F}_{i+\frac{1}{2}}^u \\ \mathcal{F}_{i+\frac{1}{2}}^w \end{pmatrix} = \mathcal{F}(U_i^n, U_{i+1}^n)$$

avec

$$\mathcal{F}^u(U_L, U_R) = \begin{cases} \frac{u_L^2}{2} & \text{si } \frac{u_L + u_R}{2} \geq 0, \\ \frac{u_R^2}{2} & \text{si } \frac{u_L + u_R}{2} < 0, \end{cases}$$

$$\mathcal{F}^w(U_L, U_R) = c_2^- w_R + c_2 w_L$$

$c_2^- = \min(c_2, 0)$, et $c_2^+ = \max(c_2, 0)$.

Les deux problèmes sont résolus simultanément, par conséquent le pas de temps δ_t^n doit correspondre aux contraintes imposées par la CFL de chaque variable composant le problème. Il est calculé comme suit

$$\delta_t^n \leq \frac{\delta_x}{\max(|c_2|, \max(|u_i^n|))}.$$

2.3 Résolution du problème de projection

La résolution du problème de projection demande l'utilisation de divers opérateurs discrets.

On pose $\nabla_i u$, dérivée discrète d'une fonction u dans le volume de contrôle x_i . On utilisera l'opérateur de gradient centré, défini par :

$$\nabla_i u = \frac{u_{i+1} - u_{i-1}}{2\delta_x}, \quad (8)$$

et $\langle \cdot, \cdot \rangle_h : \mathbb{R}^{2p} \times \mathbb{R}^{2p} \rightarrow \mathbb{R}$, le produit scalaire dans l'espace discretisé, défini par :

$$\langle U, \phi^n \rangle_h = \sum_{i=1}^p (u_i \phi_{1,i}^n + w_i \phi_{2,i}^n) = 0, \quad (9)$$

avec :

$$\begin{aligned} \phi_{1,i}^n &\approx \frac{1}{\delta_x} \int_{x_i} \phi_1(t^n, x) dx, \\ \phi_{2,i}^n &\approx \frac{1}{\delta_x} \int_{x_i} \phi_2(t^n, x) dx. \end{aligned}$$

Pour l'étape 2 de la résolution du problème, calculer $\mathcal{P}_h(U^{*,n})$ revient à poser l'équation discrétisée :

$$(1 - \alpha^2 \nabla_i \nabla_i) u^{n+1} = u_i^* + \alpha \nabla_i w_i^* \quad (10)$$

pour trouver u^{n+1} , puis on pose $w^{n+1} = -\alpha \nabla_i u^{n+1}$

Preuve:

Partons de la formule de $\mathcal{P}(U^{*,n})$:

$$\mathcal{P}(U^{*,n}) = \begin{cases} u^{n+1} & = u^* - \delta_t^n \phi_1^{n+1}. \\ w^{n+1} & = w^* - \delta_t^n \phi_2^{n+1}. \\ L(U^{n+1}) & = 0. \\ \langle U^{n+1}, \phi^{n+1} \rangle_{L^2(\Omega)} & = 0. \end{cases}$$

Par (5), et par la ligne 2 de $\mathcal{P}(U^{*,n})$:

$$u_i^{n+1} = u_i^* + \alpha \nabla_i (w_i^* - w_i^{n+1}),$$

D'après (3), on a $w_i^{n+1} = -\alpha \nabla_i u^{n+1}$, $\forall i$,

ce qui nous donne donc l'équation (10). ■

Pour obtenir une fonction U^{n+1} appartenant à l'ensemble \mathbb{A} , on calcule ensuite w_i^{n+1} à l'aide de (3).

Un problème intervient si Ω devient borné, avec l'utilisation de l'opérateur de gradient centré (8).

Proposition 1 :

Soit Ω un domaine borné, alors on a

$$\begin{aligned} u_0^n \phi_{2,1}^n + u_1^n \phi_{2,0}^n &= 0, \\ u_p^n \phi_{2,p+1}^n + u_{p+1}^n \phi_{2,p}^n &= 0. \end{aligned} \quad (11)$$

Preuve:

Partons de (9) :

$$\langle U^n, \phi^n \rangle_h = \sum_{i=1}^p (u_i^n \phi_{1,i}^n + w_i^n \phi_{2,i}^n) = 0.$$

En utilisant (3)

$$\sum_{i=1}^p (u_i^n \phi_{1,i}^n - \alpha \nabla_i u^n \phi_{2,i}^n) = 0.$$

Par (5)

$$-\alpha \sum_{i=1}^p (u_i^n \nabla_i \phi_2^n + \nabla_i u^n \phi_{2,i}^n) = 0.$$

Par (8),

$$\frac{-\alpha}{2\delta_x} \sum_{i=1}^p \left(u_i^n (\phi_{2,i+1}^n - \phi_{2,i-1}^n) + (u_{i+1}^n - u_{i-1}^n) \phi_{2,i}^n \right) = 0.$$

Cette formule se simplifie par :

$$-u_0^n \phi_{2,1}^n - u_1^n \phi_{2,0}^n + u_p^n \phi_{2,p+1}^n + u_{p+1}^n \phi_{2,p}^n = 0.$$

En isolant les deux bords de Ω , on sépare alors l'égalité précédente en deux égalités pour obtenir (11) ■

Dans le cadre de ce travail, on pose $u_0^n = u_1^n = u_L$ et $u_p^n = u_{p+1}^n = u_R$, ce qui amène aux égalités

$$\begin{cases} \phi_{2,0}^n &= -\phi_{2,1}^n. \\ \phi_{2,p+1}^n &= -\phi_{2,p}^n. \end{cases}$$

2.4 Analyse paramétrique de la méthode

Pour analyser correctement les résultats, il est nécessaire de regarder les paramètres à étudier. Une telle observation se fait à partir d'un calcul adimensionnel porté sur la projection.

On note $[T]$ l'échelle en temps, $[L]$ l'échelle en espace et \hat{u} , la forme adimensionnée de u tel que $u = \hat{u}[u]$

D'après (4) on a

$$\partial_t \hat{u} + \hat{u} \frac{[u][T]}{[L]} \partial_x \hat{u} - \frac{\alpha^2}{[L]^2} \partial_{xt} \hat{u} - \frac{c_2 \alpha^2 [T]}{[L]^3} \partial_{xxx} \hat{u} = 0 \quad (12)$$

Afin d'analyser au mieux le résultat, il est nécessaire de faire varier les coefficients adimensionnés $\frac{[u][T]}{[L]}$, $\frac{\alpha}{[L]}$, et $\frac{c_2 [T]}{[L]}$. Dans le cadre de notre observation, nous travaillerons sur un même domaine en espace et en temps. On doit alors modifier les valeurs de α , de c_2 et de u , qui sera observé par le biais de la vitesse du choc σ (obtenu à partir de Rankine - Hugoniot), pour analyser au mieux le comportement de la solution.

Le cas test pour les différentes observations est le suivant.

Sur l'intervalle $[0,10]$ décomposé en 1000 cellules, on pose pour fonction u_0 :

$$u_0(x) = \begin{cases} 2 & \text{si } x < 1, \\ 1 & \text{si } x > 2, \\ 3 - x & \text{sinon.} \end{cases}$$

La valeur de $w_0(x)$ est donnée par (3).

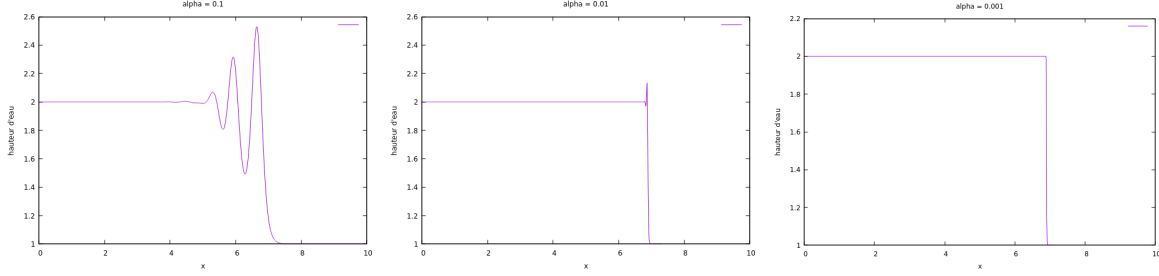


FIGURE 2 – Représentation de u selon diverses valeurs de α

La figure 2 montre le comportement de u la hauteur d'eau pour différentes valeurs de α . Ces graphes ont été réalisés au temps $T = 2.5$, avec pour valeur $\alpha = 0.1$ à gauche, $\alpha = 0.01$ au centre et $\alpha = 0.001$ à droite.

Sur ces différents graphes, il apparait une oscillation en amont de la position du choc dont l'amplitude décroît lorsque le paramètre α décroît. Ce comportement s'explique par (10) qui tend vers l'égalité $u^{n+1} = u^*$ lorsque α tend vers 0.

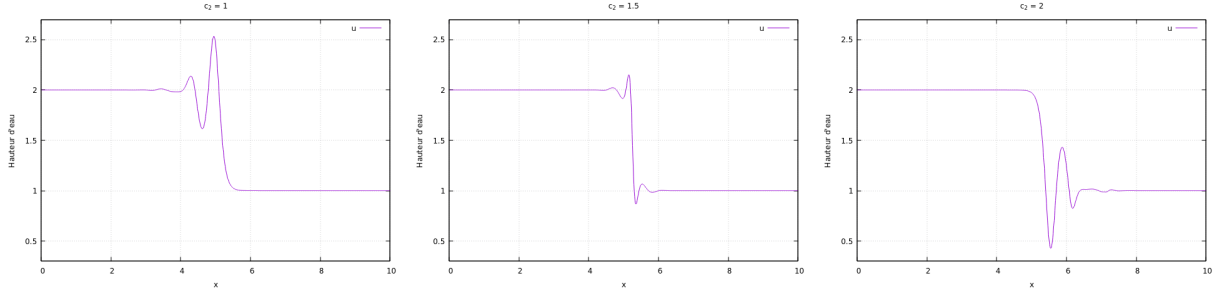


FIGURE 3 – Représentation des oscillations en fonction de la valeur de c_2

La figure 3 montre le positionnement des oscillations en fonction de la valeur du paramètre c_2 . On a $c_2 = 1$ sur le graphe de gauche, $c_2 = 1.5$ sur le graphe central et $c_2 = 2$ sur le graphe de droite. Ces graphes ont été réalisés au temps $T = 2.5$, avec $\alpha = 0.1$.

Sur les différents graphes, il apparait que les oscillations ont une position différente en fonction de la vitesse du choc σ . Si la valeur de $c_2 < \sigma$ les oscillations se forment avant le choc, si $c_2 > \sigma$ les oscillations se forment après le choc, et si $c_2 = \sigma$ les oscillations se forment avant et après le choc. Dans les trois cas, les oscillations ont un comportement commun, plus elles sont proches du choc et plus l'amplitude de l'oscillation est élevée.

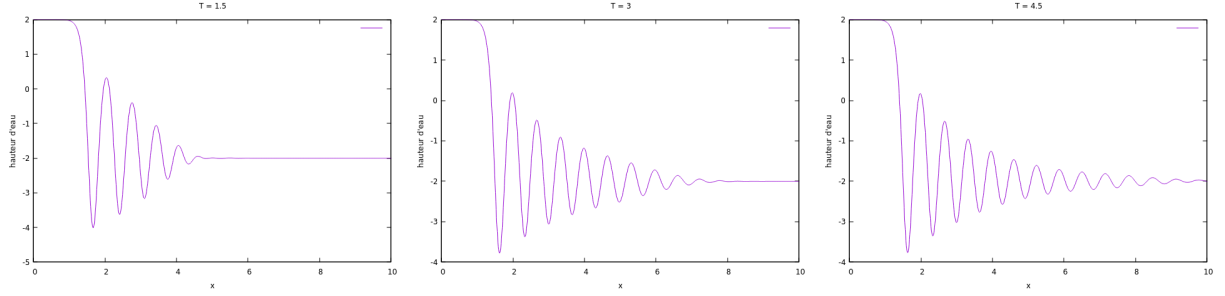


FIGURE 4 – Représentation des oscillations avec une vitesse de choc nulle, en différents temps.

La figure 4 représente les oscillations d'un même cas test sur différents temps. Ces observations ont été faites en modifiant le u_0 usuel en utilisant celui de la section 3.2.1. Les paramètres sont $c_2 = 2$ et $\alpha = 0.1$. Les graphes ont été faits aux temps $T = 1.5$ à gauche, $T = 3$ au centre et $T = 4.5$ à droite.

Ces trois graphes montrent une propagation à vitesse c_2 des oscillations dans le domaine Ω . Notons également que plus une oscillation est loin de la position du choc, et plus son amplitude sera faible.

Temps de calcul

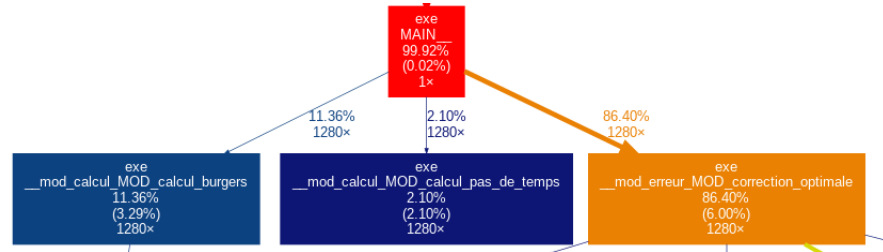


FIGURE 5 – Schéma de répartition du temps de calcul

La figure 5 est un graphe de la répartition du temps de calcul pour le cas test de la section 2.4 pour une durée de $T = 5$. Le bloc "mod_calcul_Burgers" à droite, représente la 1ère étape de résolution, avec le bloc "Calcul_du_pas_de_temps" au centre, qui a pour but de calculer le pas de temps. Le bloc de droite "Correction_optimale" correspond à la partie du code qui résout l'étape 2. Chaque blocs est pointé par une flèche qui montre le nombre de fois où le bloc est appelé lors de la résolution du problème, et le symbole % montre le pourcentage du temps passé dans le bloc, comparé au temps de calcul total.

Le graphe montre que le temps de calcul est inégalement répartie entre les différentes étapes. L'étape 1 avec le calcul du pas de temps prends près de 13.6% du temps de calcul total, tandis que l'étape 2 prends près de 86.4% du temps de calcul.

Cette inégalité du temps de calcul est dû au système matriciel qui demande également aux matrices de se construire (dans notre cas, il a été préférable d'effectuer un stockage pentadiagonal pour les matrices) en chaque pas de temps, et de résoudre le problème matriciel. Il est donc préférable de chercher à optimiser la seconde étape dans l'optique de gagner un maximum de temps de calcul.

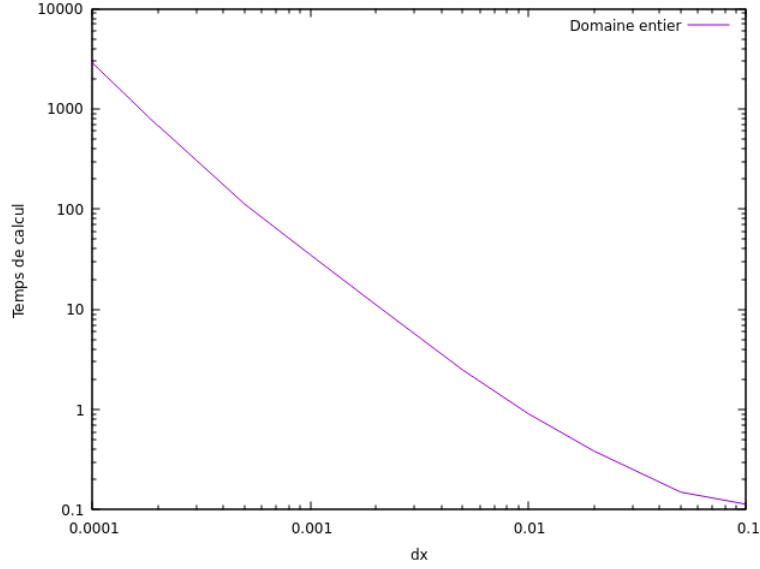


FIGURE 6 – Temps de calcul de la méthode sur le domaine entier

La figure 6 représente le temps de calcul nécessaire pour faire le cas test de la section 2.4 pour une durée de $T = 5$ avec différents raffinements de maillages (allant de $\delta_x = 0.1$ à $\delta_x = 10^{-4}$). Le temps de calcul est exprimé en seconde.

Ces valeurs servent de valeurs de temps de calcul de référence, l'objectif est alors de développer une méthode pouvant réduire les temps relevés. Pour ce faire, la méthode employée consistera à résoudre localement l'étape 2 de la résolution sur un domaine que l'on souhaite optimal (les critères définissant un domaine comme optimal sont donnés dans la section 3.1).

3 Projection sur un domaine réduit : approche itérative

3.1 Estimation de l'erreur et domaine de correction

Comme présenté dans la section 2.1, l'algorithme se décompose en 2 étapes distinctes. Le temps de calcul pour effectuer ces étapes est inégal et une grande partie de ce dernier correspond à la résolution de la seconde étape (présentée dans la section 2.3). Cette inégalité vient principalement de la résolution du problème matriciel intervenant sur la seconde étape comme le montre la figure 5.

Pour satisfaire une problématique de réduction de temps de calcul, la piste privilégiée est de faire l'étape 2 sur un sous-domaine afin de réduire la taille du système à résoudre. Pour connaître le domaine, l'approche est la suivante : "Si je suis trop loin de mon plan de projection \mathbb{A} et que je ne crée pas trop d'erreur aux bords, alors je projette".

L'idée d'un "trop loin" étant vague, il est alors préférable de quantifier une distance à notre zone de projection. Pour cela nous introduisons $\epsilon_i(u, w)$, qui représente une erreur de projection sur une cellule x_i . Cette distance sera calculée entre l'étape 1 et l'étape 2 avec pour but de définir un sous-domaine pour calculer l'étape 2. Elle se définit de la manière suivante :

$$\epsilon_i(u^*, w^*) = |w_i^* + \alpha \nabla_i u^*| \quad (13)$$

D'un point de vue numérique, on se doit également d'établir un seuil de tolérance $\bar{\epsilon}$, pour lequel, si l'erreur ϵ_i est au dessus, alors on projette sur x_i , et sinon, alors on ne projette pas.

La valeur de $\bar{\epsilon}$ sert à déterminer un domaine \mathcal{D} prenant en compte l'ensemble des cellules ayant une erreur non tolérable pour l'utilisateur, et d'effectuer la seconde étape uniquement sur \mathcal{D} . Elle doit correspondre au cas voulu par l'utilisateur en fonction de sa nécessité à être proche de KdV-BBM. En effet, un choix trop élevé de $\bar{\epsilon}$ peut alors donner un domaine de projection nul sur l'ensemble de la solution et entraîner la résolution d'un problème de Burgers en u et un problème de transport linéaire en w sans se soucier du domaine de fonctions admissible \mathbb{A} .

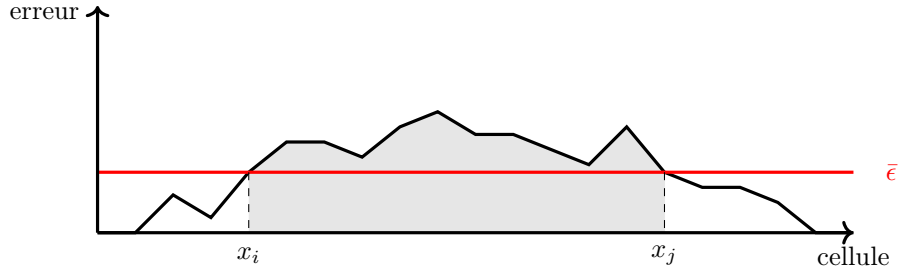


FIGURE 7 – Illustration du domaine $\mathcal{D} = [x_i, x_j]$

L'utilisation d'une telle approche pour \mathcal{D} a pour effet de retirer l'erreur de projection à l'intérieur du domaine mais il crée également de l'erreur au bord. Cette apparition d'erreur au bord fait suite à l'utilisation de l'opérateur gradient centré qui a pour effet de modifier l'erreur dans les cellules voisines de \mathcal{D} pouvant ainsi devenir non négligeable (i.e. $\epsilon_{i-1} > \epsilon_{max}$ ou/et $\epsilon_{j+1} > \epsilon_{max}$). Un tel effet aura pour conséquence de donner une approximation erronée du problème.

D'un point de vue numérique il est donc nécessaire de prendre un domaine plus grand, tel que l'erreur de projection engendrée au bord reste tolérable ($< \bar{\epsilon}$) sur Ω .

L'ensemble du problème de trouver un \mathcal{D} optimal réside dans le fait que le domaine se doit d'être assez grand pour éviter des erreurs aux bords trop élevées, mais également de trouver un domaine de projection \mathcal{D} le plus petit possible, afin de satisfaire au mieux la problématique de réduction du temps de calcul.

3.2 Approche itérative

Soit x_p^n et x_j^n , deux points de discretisation de Ω tel que $x_p^0 = \min_{x \in \Omega}(\epsilon_i \geq \bar{\epsilon})$ et $x_j^0 = \max_{x \in \Omega}(\epsilon_i \geq \bar{\epsilon})$. L'approche naïve d'obtention du domaine optimal est une approche itérative qui se déroule de la manière suivante :

1. Initialise $\mathcal{D} = [x_p^0, x_j^0]$, et $n = 0$.
2. Calcul de la projection et des contraintes sur \mathcal{D} .
3. si $\epsilon_{i-1} > \bar{\epsilon}$ alors $x_p^{n+1} = x_p^n - \delta_x$ et retour à 2 avec $n = n + 1$.
4. si $\epsilon_{j+1} > \bar{\epsilon}$ alors $x_j^{n+1} = x_j^n + \delta_x$ et retour à 2 avec $n = n + 1$.
5. Sortie de $\mathcal{D} = [x_p^n, x_j^n]$

Une telle approche a pour but d'obtenir un domaine \mathcal{D} optimal, dans le sens où \mathcal{D} est le plus petit domaine tel que la projection engendre une erreur plus petite que $\bar{\epsilon}$ sur l'ensemble du domaine. Un tel domaine est appelé \mathcal{D}_{opt} .

Cette approche possède néanmoins un défaut vis à vis du temps de calcul, à cause de l'approche qui est itérative, demandant de réaliser plusieurs fois la seconde étape, sur un domaine de plus en plus grand. Une telle réalisation ne permet pas de satisfaire la problématique d'optimisation du temps de calcul, qui est dans certains cas plus long que de faire une projection sur l'ensemble du domaine.

Pour satisfaire la problématique d'optimisation du temps de calcul, il est donc nécessaire de mettre en place une méthode directe de la détermination du domaine de projection \mathcal{D}_{opt} .

L'idée est à présent de prédire le comportement du domaine. Il est donc nécessaire de comprendre comment il évolue. Pour ce faire nous nous appuierons sur deux cas tests.

3.2.1 Etude d'un cas à vitesse de choc nulle

Le premier cas est posé tel que :

$$u_0(x) = \begin{cases} 2 & si & x < 1, \\ -2 & si & x > 2, \\ 6 - 4x & sinon. \end{cases}$$

Le but de ce cas test est de voir le comportement du domaine dans un cas où la vitesse du choc en u (notée σ) est nulle.

Pour satisfaire une analyse correcte du domaine, on va poser un jeu de données de base. L'observation sera effectuée sur une partition formée de 1000 cellules de l'intervalle $[0,10]$ pendant une durée de $T = 10$, α prendra pour valeur 0.1, $c_2 = 2$ et $\bar{\epsilon} = 10^{-3}$. Ce jeu de variables sera utilisé dans l'ensemble des observations, une précision sera donnée dans la présentation des figures en cas contraire.

On prendra pour condition au bord $u(t, 0) = 2$, et $u(t, 10) = -2$.

Pour l'analyse paramétrique de ce cas test, chaque paramètre sera changé un à un, dans le but d'observer au mieux l'implication de chacun des paramètres dans l'obtention du domaine \mathcal{D}_{opt} .

Variation du paramètre α

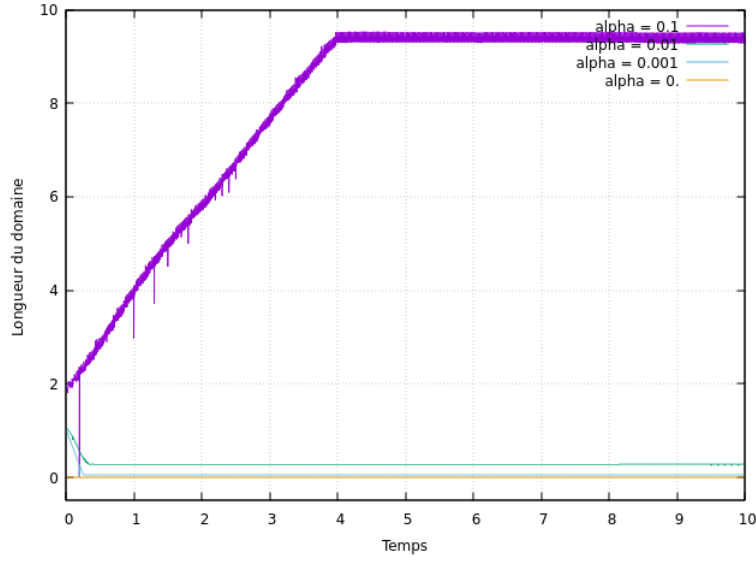


FIGURE 8 – Représentation de la longueur du domaine en fonction du temps, par différentes valeurs de α .

La figure représente la taille du domaine \mathcal{D}_{opt} au cours du temps selon plusieurs valeurs de α lorsque la vitesse du choc est $\sigma = 0$. Sur ce graphe, $\alpha = 0.1$ sur la courbe violette, $\alpha = 0.01$ sur la courbe verte, $\alpha = 0.001$ sur la courbe bleue, et $\alpha = 0$ sur la courbe jaune.

Ce graphe montre que pour $\alpha = 0.1$, le domaine augmente, jusqu'à atteindre une valeur qui borne sa longueur. Pour des valeurs de α plus petites, le domaine optimal se réduit pour retrouver une longueur fixe au cours du temps. Pour le cas particulier de $\alpha = 0$, le domaine optimal \mathcal{D}_{opt} n'est pas calculé en tout pas de temps.

Ces différents comportements du domaine optimal sont liés à l'influence de α sur les oscillations observées sur les figures 2 et 4, par le fait que la position du domaine optimal dépend de la position des oscillations. La courbe violette s'explique par la figure 4 qui montre que les oscillations prennent tout le domaine, ce qui explique l'augmentation de la longueur du domaine, puis son blocage qui est lié au fait que le domaine optimal touche le bord de Ω .

Pour $\alpha = 0.01$ et $\alpha = 0.001$, la baisse et la stagnation de la longueur du domaine optimal s'explique par le fait que les oscillations normalement présentes ont une amplitude trop faible à partir d'une certaine distance du choc, ce qui crée une erreur locale plus faible que le seuil de tolérance. Pour le cas particulier de $\alpha = 0$, il y a l'égalité $u^{n+1} = u^*$ sur Ω d'après (10). Il n'y a donc pas de calcul de \mathcal{D}_{opt} (et pas de calcul de l'étape 2).

Variation du paramètre c_2

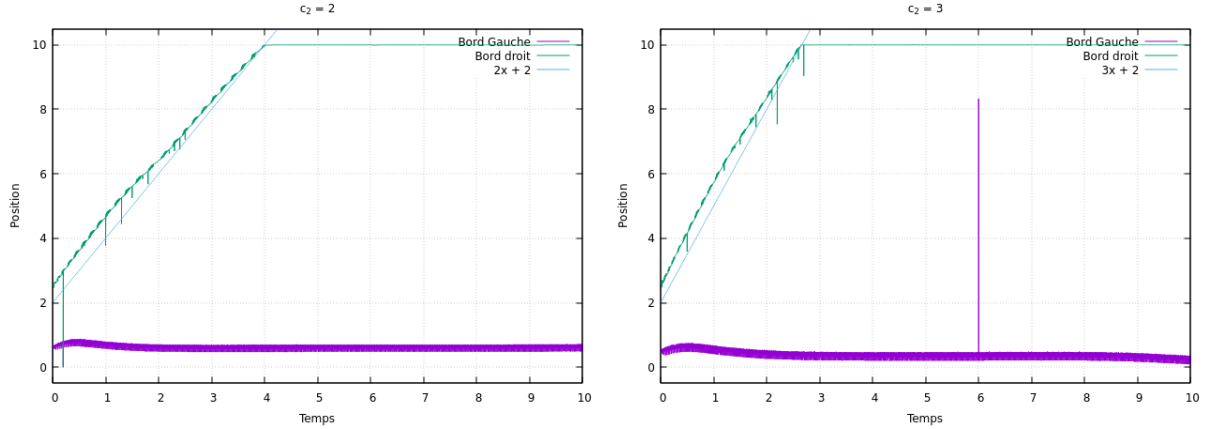


FIGURE 9 – Position des bords du domaine \mathcal{D}_{opt} en fonction du temps, selon différentes valeurs de c_2

La figure 9 montre l'évolution de la position des bords du domaine \mathcal{D}_{opt} , avec $c_2 = 2$ à gauche et $c_2 = 3$ à droite. Pour chacun des graphes, la courbe violette représente le bord gauche du domaine, et la courbe verte représente le bord droit.

Sur chacun des graphes, le bord gauche évolue peu, tandis que le bord droit parcourt le domaine Ω à une vitesse jusqu'à se trouver bloqué à une valeur fixe pour ces deux cas.

Pour chacun des cas tests, le bord droit parcourt le domaine Ω suivant une vitesse approchant les valeurs de c_2 attribués pour chaque cas test. Le blocage en $x = 10$ correspond au bord droit de Ω est atteint par le domaine optimal \mathcal{D}_{opt} . Sur le graphe de gauche, le bord droit est égal au bord gauche en $x = 0$ sur une itération, ce qui correspond numériquement au fait que le domaine optimal n'est pas calculé. Un tel cas intervient si l'ensemble des erreurs locales sont inférieures au seuil de tolérance $\bar{\epsilon}$. Les petites irrégularités dans l'évolution des bords sont liées à la propagation et l'accumulation de l'erreur autour du domaine, s'accumulant au cours du temps.

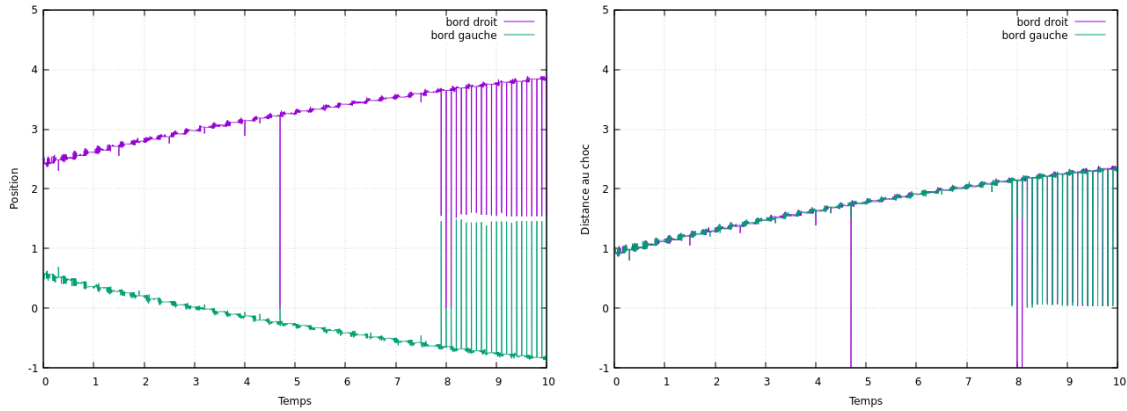


FIGURE 10 – Position du domaine \mathcal{D}_{opt} en fonction du temps avec $c_2 = 0$

Sur la figure 10 le domaine Ω a été modifié et vaut maintenant $[-4, 4]$ et $c_2 = 0$ pour chacun des graphes. Le graphe de gauche représente la position des bords du domaine \mathcal{D}_{opt} dans Ω , tandis que le graphe de droite représente la distance entre les bords de \mathcal{D}_{opt} et la position du choc. Pour chaque graphe, la courbe violette

correspond au bord droit de \mathcal{D}_{opt} et la courbe verte correspond au bord gauche.

Ces deux graphes montrent que les bords du domaine s'éloignent de la position du choc au cours du temps. Cet éloignement est réalisé de manière symétrique par rapport à la position du choc. On constate également des légères instabilités sur les positions des bords.

L'éloignement des bords du domaine \mathcal{D}_{opt} est lié à la propagation des oscillations au cours du temps, tandis que les petites instabilités sont liées à une accumulation d'erreur que crée cette propagation.

On observe également des points en temps régulier où le domaine \mathcal{D}_{opt} est beaucoup moins grand. Cela est dû à des étapes où le pas de temps est très faible, car il y a eu nécessité de regarder la solution en des temps précis, le δ_t était donc beaucoup plus faible qu'à l'usuel pour tomber sur ces temps, ce qui n'a pas donné le temps à l'erreur d'apparaître, donnant ainsi un \mathcal{D}_{opt} plus petit.

Variation de $\bar{\epsilon}$

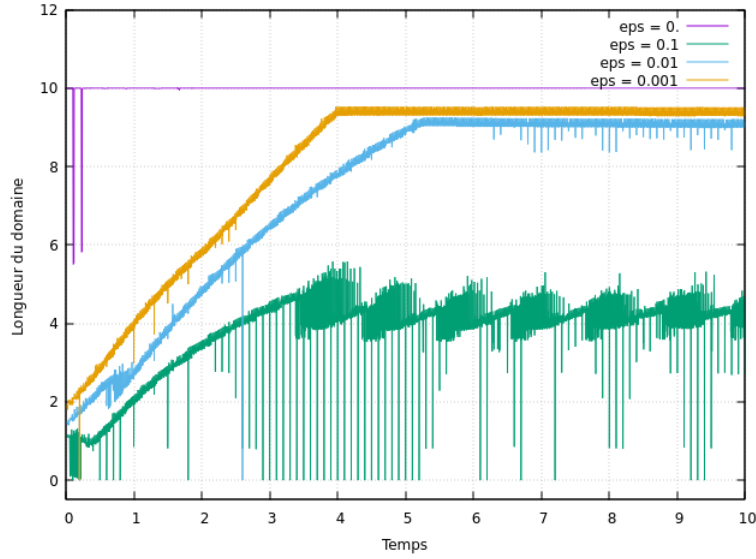


FIGURE 11 – Longueur de \mathcal{D}_{opt} au cours du temps en fonction de la tolérance $\bar{\epsilon}$

La figure 11 montre la longueur de \mathcal{D}_{opt} en fonction de différentes tolérance (c_2 et Ω ont repris leurs valeurs initiales). On a $\bar{\epsilon} = 0.1$ sur la courbe verte, $\bar{\epsilon} = 0.01$ sur la courbe bleue, $\bar{\epsilon} = 0.001$ sur la courbe jaune et $\bar{\epsilon} = 0$ sur la courbe violette.

Ce graphe montre que la taille du domaine de \mathcal{D}_{opt} augmente lorsque la valeur de $\bar{\epsilon}$ diminue, il y a également une réduction des instabilités dans la taille du domaine lorsque $\bar{\epsilon}$ diminue.

Variation du nombre de cellules

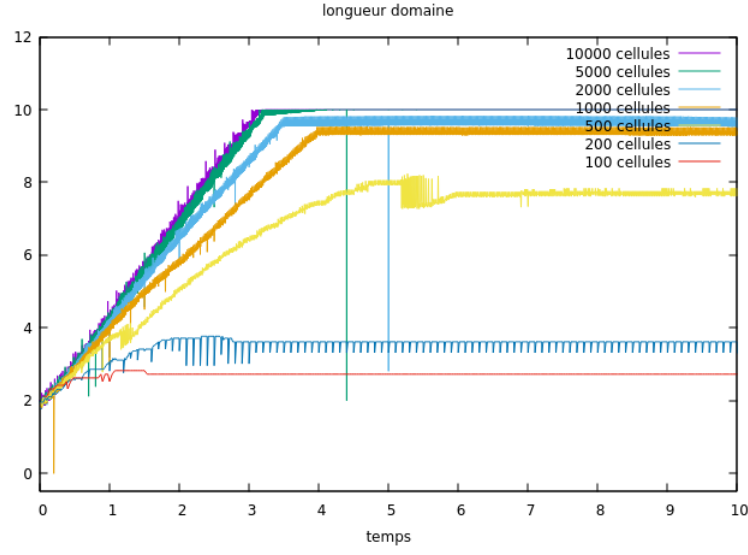


FIGURE 12 – Evolution de la longueur du domaine sur différents maillages, en fonction du temps.

La figure 12 représente la taille du domaine \mathcal{D}_{opt} pour différents maillages. La courbe rouge représente un maillage de 100 cellules, la courbe orange représente un maillage de 1000 cellules, et la courbe violette représente un maillage de 10 000 cellules.

La longueur du domaine \mathcal{D}_{opt} s'accroît en fonction du nombre de cellules, et converge.

Observation du domaine en fonction du nombre de cellules : Evolution des bords du domaine optimal

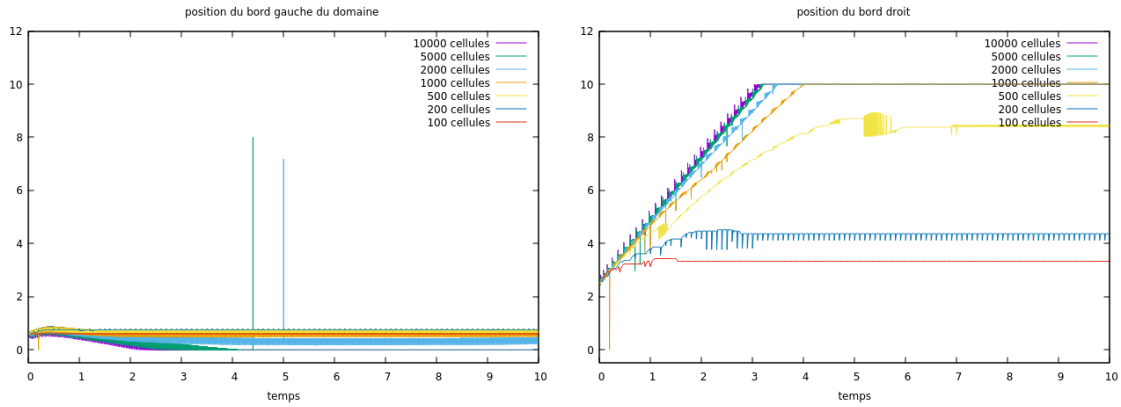


FIGURE 13 – Positions des bords du domaine \mathcal{D}_{opt}

La figure 13 est constituée de deux graphes qui représentent les positions des bords du domaine \mathcal{D}_{opt} pour différents maillages. Le graphe de gauche représente le bord gauche de \mathcal{D}_{opt} , et le graphe de droite représente le bord droit. La courbe rouge représente un maillage de 100 cellules, la courbe orange représente un maillage de 1000 cellules, et la courbe violette représente un maillage de 10 000 cellules.

Sur le graphe de gauche, la position du bord de gauche décroît de plus en plus avec le raffinement du maillage, tandis que sur le graphe de droite, le bord droit s'accroît de plus en plus. Chaque bord converge en fonction du nombre de cellules. Pour les maillages les plus fins, les bords de gauche de \mathcal{D}_{opt} fixés à 0, et les bords de droite fixés à 10, sont liés aux bords du domaine $\Omega = [0, 10]$.

3.2.2 Etude d'un cas à vitesse de choc positive

Pour le second cas on prend $u_0(x)$ tel que

$$u_0(x) = \begin{cases} 2 & \text{si } x < 1. \\ 1 & \text{si } x > 2. \\ 3 - x & \text{sinon.} \end{cases}$$

Le but de ce cas test est de regarder un cas où la vitesse du choc en u notée σ est non nulle (ici $\sigma = 1.5$ d'après Rankine-Hugoniot).

On prendra pour condition au bord $u(t, 0) = 2$, et $u(t, 10) = 1$.

On pose les mêmes jeux de paramètres que sur la section 3.2.1. Ne seront affichés que les graphes apportant une idée nouvelle, n'apparaissant pas sur l'étude du cas précédent.

Variation de c_2

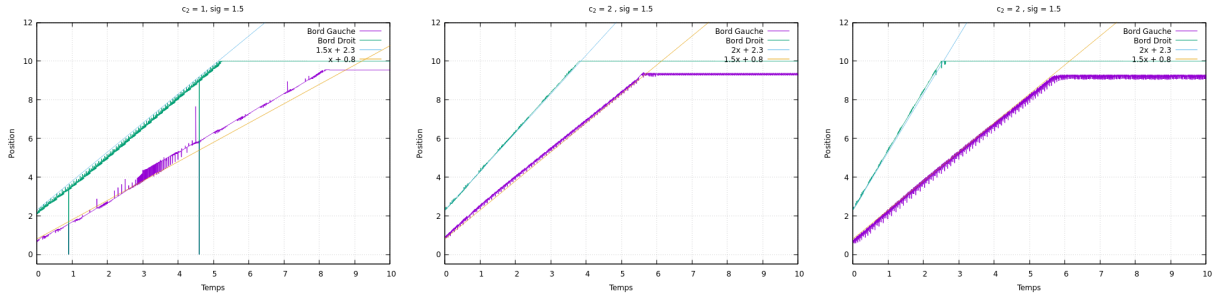


FIGURE 14 – Evolution de la position des bords du domaine pour plusieurs valeurs de c_2 en fonction du temps

La figure 14 est composée de trois graphes représentant l'évolution au cours du temps de la position des bords du domaine \mathcal{D}_{opt} en fonction de diverses valeurs de c_2 . Sur chaque graphe, la courbe violette représente le bord gauche du domaine et la courbe verte représente le bord droit. Sur le graphe de gauche, $c_2 = 1$, sur le graphe central $c_2 = 2$, et sur le graphe de droite $c_2 = 3$. Il y a également deux fonctions affines sur chaque graphe, en bleu une droite affine de pente $\max(\sigma, c_2)$, et en jaune une droite affine de pente $\min(\sigma, c_2)$.

Sur les différents graphes, on voit un déplacement des deux bords de \mathcal{D}_{opt} dans Ω au cours du temps. L'évolution de la courbe verte au cours du temps coïncide avec la droite bleue, et l'évolution de la courbe violette coïncide avec la droite jaune.

Les positions des bords de \mathcal{D}_{opt} suivent une évolution de vitesse $\min(\sigma, c_2)$ et $\max(\sigma, c_2)$ jusqu'à être confrontés aux conditions aux bords (qui sont ici des conditions de Dirichlet non homogène), qui est une source d'erreur dans notre cas, lorsque le domaine optimal touche le bord puisque l'on fixe la valeur de u .

3.3 Erreur et temps de calcul de la méthode itérative

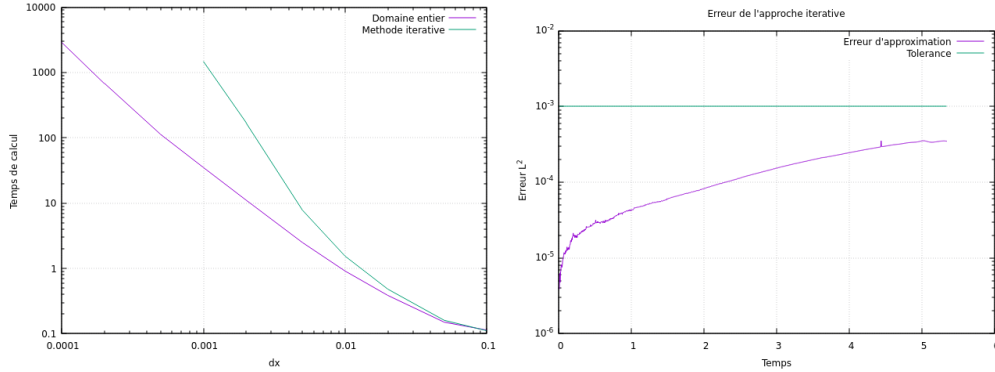


FIGURE 15 – Comparasion entre le temps de calcul de l’approche itérative, et de la projection sur un domaine entier (à gauche). Norme L^2 de l’erreur d’approximation, en fonction du temps (à droite)

La figure 15 est décomposée en deux graphes. Le graphe de gauche représente le temps de calcul (en secondes) nécessaire à la résolution du cas test présenté dans la section 3.2.2 sur une durée de $T = 5$ en fonction de différents maillages. Il est composé de deux courbes. La courbe violette représente le temps de calcul obtenu sur la projection d’un domaine entier (c’est la même courbe que celle présente en figure 6), et la courbe verte représente le temps de calcul obtenu pour résoudre le problème à l’aide de la méthode itérative. Le graphe de droite représente l’évolution en temps de la norme L^2 sur le domaine Ω de la différence entre la méthode de projection sur le domaine entier et la méthode par projection locale sur un domaine obtenue de manière itérative.

Sur le graphe de gauche, on remarque que la courbe verte croît plus rapidement que la courbe violette. Par exemple pour un maillage de 10 000 cellules (ce qui représente $\delta_x = 10^{-3}$), le temps de calcul est de 1454 secondes pour la méthode itérative, tandis que la projection sur tout le domaine résout le problème en 35 secondes.

Sur le graphe de droite, la courbe d’erreur admet des valeurs faibles qui s’accroissent avec le temps. Cet accroissement est lié à la propagation de l’erreur dans le temps, puisque l’on effectue des approximations, à partir de données contenant une erreur.

L’accroissement trop élevé du temps de calcul est dû aux projections successives qui sont nécessaires pour obtenir un domaine optimal (ce qui a déjà été énoncé au début de la section 3.2). Bien qu’elle soit très longue à calculer, cette méthode a néanmoins un point positif qui est que le domaine \mathcal{D}_{opt} est une bonne (si ce n’est la meilleure) estimation du domaine sur lequel on souhaite faire l’étape de projection. Le but est à présent de trouver une méthode explicite pour calculer un domaine \mathcal{D} contenant \mathcal{D}_{opt} avec pour objectif de s’en rapprocher le plus possible. Ce point positif est également souligné par le graphe de droite, qui montre que l’erreur ne dépasse pas 10^{-3} , ce qui témoigne de l’efficacité de la méthode du point de vue de la précision de la résolution.

4 Approche explicite du domaine optimal

4.1 Opérateurs nécessaires pour une approche directe du domaine

Pour satisfaire l'approche directe du domaine, il est nécessaire pour les méthodes de résolutions employées d'introduire deux opérateurs \bar{x}_τ et $\bar{\sigma}_\tau$, correspondant à la moyenne empirique et à l'écart-type empirique de l'erreur de projection. Ils sont calculés de la manière suivante

$$\bar{x}_\tau = \int_{\Omega} x \frac{\epsilon(x) \mathbb{1}_{\epsilon(x) > \tau \bar{\epsilon}}(x)}{\|\epsilon \mathbb{1}_{\epsilon(x) > \tau \bar{\epsilon}}\|_{L^2(\Omega)}} dx, \quad (14)$$

$$\bar{\sigma}_\tau = \sqrt{\int_{\Omega} (x - \bar{x}_\tau)^2 \frac{\epsilon(x) \mathbb{1}_{\epsilon(x) > \tau \bar{\epsilon}}(x)}{\|\epsilon \mathbb{1}_{\epsilon(x) > \tau \bar{\epsilon}}\|_{L^2(\Omega)}^2} dx}. \quad (15)$$

Autrement dit, à partir de la Figure 7, n'est prise en compte que la partie grisée pour $\tau = 1$, et toutes les erreurs relevées pour $\tau = 0$.

La mise en place de ces paramètres a pour but de trouver une fonction les utilisant pour approcher au mieux les bords du domaine optimal \mathcal{D}_{opt} obtenu par la méthode itérative. Les approches devront majorer le domaine, pour éviter une propagation de l'erreur de projection, et rester sous le seuil de tolérance $\bar{\epsilon}$

4.2 Présentation de méthodes explicites appuyées sur de la régression linéaire

4.2.1 Présentation des opérateurs, et estimation d'erreur

La première méthode présentée ici consiste à approcher les bords du domaine D_{opt} , par une combinaison linéaire entre les deux moments empiriques.

Par conséquent, si on note x_l, x_r les deux bords du domaines, on obtient que l'on cherche une matrice $A = (A_l, A_r)$:

$$\langle (\bar{x}_{\tau_1}, \bar{\sigma}_{\tau_1}, \dots, \bar{x}_{\tau_n}, \bar{\sigma}_{\tau_n}), A_l \rangle = x_l$$

$$\langle (\bar{x}_{\tau_1}, \bar{\sigma}_{\tau_1}, \dots, \bar{x}_{\tau_n}, \bar{\sigma}_{\tau_n}), A_r \rangle = x_r$$

Afin de trouver les coefficients, on se servira d'une base de données de m cas tests E_m , et des différents domaines obtenus de manière itérative pour ces cas tests D_m .

Le but sera alors de trouver la matrice A telle que

$$E_m A = D_m.$$

Pour la suite, on pose $E = E_m$ et $D = D_m$.

Proposition 2 La matrice A est donnée par $A = (E^t E)^{-1} E^t D$.

Preuve:

On a $EA = D$ ce qui implique que $E^t EA = E^t D$
 $E^t E$ est une matrice symétrique, donc elle est inversible

Donc $A = (E^t E)^{-1} E^t D$. ■

Pour tester convenablement la qualité de l'approche, il faut intégrer un opérateur d'efficacité $\hat{\epsilon}$. On le définira de la manière suivante :

$$\hat{\epsilon} = 1 - \int_{\Omega} \frac{\epsilon_{\mathcal{D}_{opt i}}(x) dx}{\epsilon(x)}$$

avec $\epsilon_{\mathcal{D}_{opti}}$ l'erreur prélevée sur le domaine Ω lors de l'approche itérative du problème.

Si l'estimateur d'erreur $\hat{\epsilon}$ est positif alors le domaine obtenu de manière explicite est trop petit. Si $\hat{\epsilon}$ est négatif, alors le domaine calculé explicitement est trop grand. L'objectif est alors de faire tendre cette erreur vers 0 en utilisant un minimum de paramètres d'observations dans une optique d'optimisation du temps de calcul. Une méthode sera toutefois supposée utilisable pour certains jeux de paramètres si $\hat{\epsilon} \leq 0$.

Dans les observations, on appellera erreur de projection, une erreur définie par :

$$Err_{proj} = \max_j (\|L(U^j)\|_{L_2(\Omega)})$$

avec $L(U^j)$ défini par (3) au temps $t = t_j$.

4.3 Méthode d'approche explicite par régression linéaire

Tout au long du stage, plusieurs pistes ont été étudiées pour obtenir le plus petit domaine calculé de manière explicite \mathcal{D} , majorant \mathcal{D}_{opt} .

Dans cette partie, certaines méthodes présentées n'ont pu aboutir pour des raisons techniques (base de test trop coûteuse à faire, manque de temps sur le stage, ...).

Dans le but de comparer les différentes méthodes, un même cas test (présenté dans la section 3.2.2) sera utilisé avec $\tau = 0$ (sauf énonciation du cas contraire), sur lequel sera appliqué une analyse paramétrique de l'erreur de projection, en fonction des paramètres numériques (nombre de cellules, tolérance d'erreur, valeur de τ).

4.3.1 Méthode 1 : $\mathcal{D} = [\bar{x}_\tau + a_1 \bar{\sigma}_\tau ; \bar{x}_\tau + a_2 \bar{\sigma}_\tau]$

L'idée derrière cette méthode est de retrouver un domaine majorant \mathcal{D}_{opt} en partant du point \bar{x}_τ .

L'obtention des coefficients a_1 et a_2 s'effectue à l'aide de la méthode expliquée dans la section 4.2.1, avec une matrice $A = (a_1, a_2)$

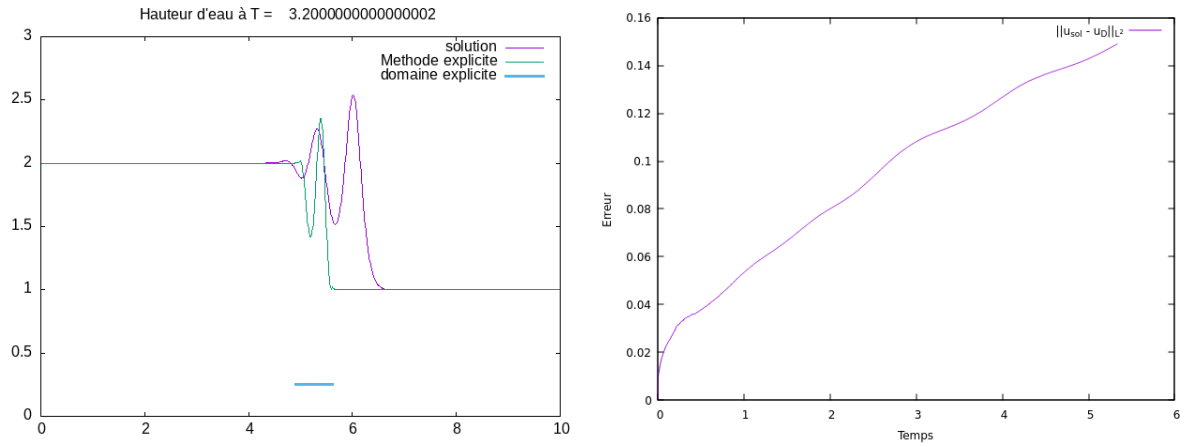


FIGURE 16 – Superposition entre la solution exacte et l'approximation (à gauche). Norme L^2 de l'erreur d'approximation en fonction du temps (à droite).

La figure 16 se décompose en deux graphes. Le graphe de gauche se décompose en trois courbes. La courbe violette représente la solution obtenue à partir de la projection sur tout le domaine, la courbe verte représente l'approximation obtenue à l'aide de la méthode explicite, et la courbe bleue représente le domaine \mathcal{D} calculé à partir de cette même méthode. Le graphe de droite représente l'évolution en temps de la norme L^2 sur le domaine Ω de la différence entre la méthode de projection sur le domaine entier et la méthode par projection locale sur le domaine \mathcal{D} .

Le graphe de gauche montre que les résultats de cette méthode explicite sont éloignés de la solution. Ce résultat est confirmé par le graphe de droite, qui montre que l'erreur dépasse 0.1 pour cette méthode, tandis qu'elle ne dépassait pas 10^{-3} sur la méthode itérative (cf figure 15).

Cette mauvaise approximation est liée à une mauvaise approche du domaine, qui est pour notre cas plus petit que le domaine optimal \mathcal{D}_{opt} . La projection ne peut donc se faire sur un domaine assez grand, ce qui crée une forte erreur.

Variation de $\bar{\epsilon}$:

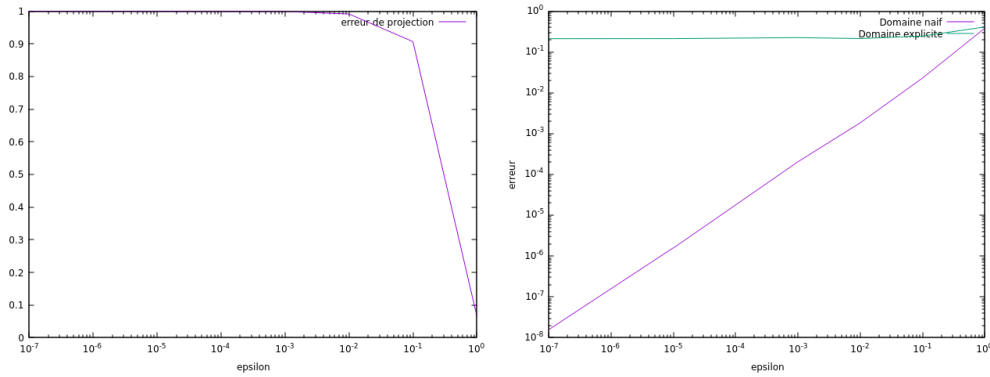


FIGURE 17 – Valeur de $\hat{\epsilon}$ (à gauche) et valeur de Err_{proj} (à droite), en fonction de la tolérance $\bar{\epsilon}$

La Figure 17 se décompose en 2 graphes, celui de gauche représente la valeur de $\hat{\epsilon}$ en fonction de la tolérance $\bar{\epsilon}$, le graphe de droite représente la valeur de Err_{proj} en fonction de la tolérance, la courbe verte représente cette valeur en utilisant le domaine de projection \mathcal{D} , et la courbe violette représente Err_{proj} appliquée sur le même cas test, mais utilisant le domaine de projection \mathcal{D}_{opt} .

Sur le graphe de gauche, on remarque que $\hat{\epsilon}$ augmente lorsque $\bar{\epsilon}$ diminue, ce qui montre que la méthode de calcul explicite du domaine n'est pas adaptée au cas où la tolérance $\bar{\epsilon}$ tend vers 0. L'explication de cela est donnée sur le graphe de droite, qui montre que l'utilisation de la projection sur \mathcal{D}_{opt} fait tendre l'erreur Err_{proj} vers 0 avec une vitesse de convergence de 1 lorsque la tolérance tend vers 0, mais que l'utilisation du domaine \mathcal{D} fait apparaître la présence d'un minimum strictement positif de l'erreur de valeur 0.2.

Variation de τ :

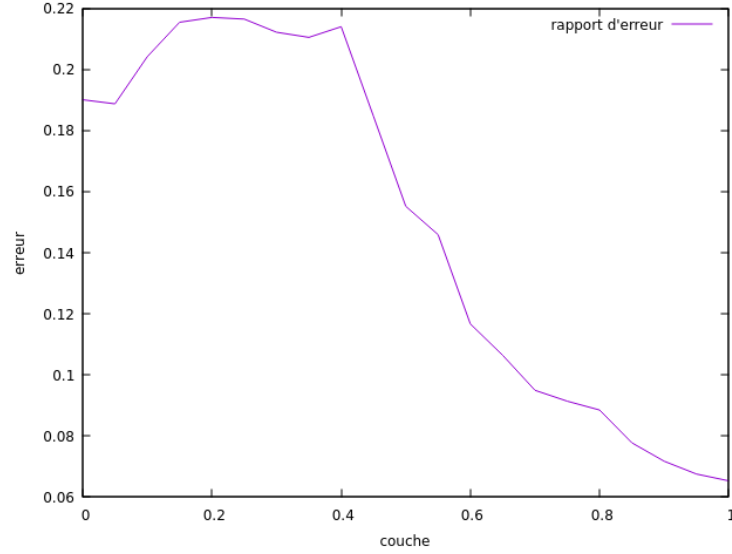


FIGURE 18 – Valeur de $\hat{\epsilon}$ en utilisant le domaine \mathcal{D} , en fonction de la valeur τ .

La Figure 18 représente la valeur de l'erreur de projection $\hat{\epsilon}$, sur un cas où le domaine explicite \mathcal{D} est calculé à partir de différentes valeurs de τ .

La valeur de τ fait varier $\hat{\epsilon}$. Pour avoir un résultat se rapprochant au mieux du résultat obtenu de manière itérative, il est alors préférable d'utiliser un τ avec une valeur élevée, se rapprochant de 1.

Variation du nombre de cellules :

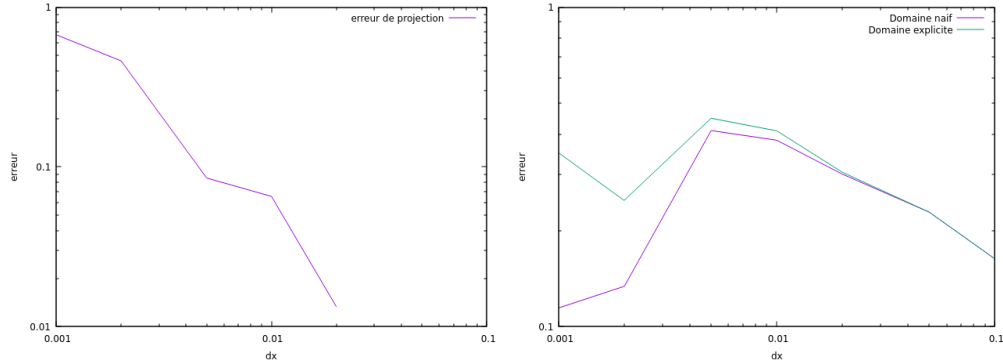


FIGURE 19 – Valeur de $\hat{\epsilon}$ (à gauche) et valeur de Err_{proj} (à droite), en fonction du maillage utilisé.

La figure 19 est composée de deux graphes. Le graphe de gauche représente la valeur de $\hat{\epsilon}$ en fonction du nombre de cellules dans le domaine, et le graphe de droite représente l'erreur de projection Err_{proj} sur différents maillages avec en violet l'erreur obtenue à l'aide d'un domaine itératif \mathcal{D}_{opt} , et en vert celle obtenue à partir d'un domaine explicite \mathcal{D} .

Sur le graphe de gauche, $\hat{\epsilon}$ tend vers 0 pour des maillages plus grossiers (Par l'utilisation de l'échelle log, les derniers points ne sont pas affichés car $\log(0)$ n'est pas défini) ce qui est confirmé par le graphe de droite.

Cette première méthode est donc utilisable pour des maillages peu raffinés. Dans le but d'obtenir une méthode convenable pour un plus grand nombre de mailles, il est nécessaire d'améliorer cette méthode. Un premier axe d'amélioration consiste à augmenter le nombre de variables à observer.

4.3.2 Méthode 2 : $\mathcal{D} = [b_1\bar{x}_\tau + a_1\bar{\sigma}_\tau, b_2\bar{x}_\tau + a_2\bar{\sigma}_\tau]$

Dans ce second cas, on ajoute \bar{x}_τ comme un matériel d'observation nécessaire pour l'estimation de \mathcal{D} .

Les coefficients a_1 , a_2 , b_1 et b_2 sont obtenus à l'aide de la méthode expliquée dans la section 4.2.1, en posant $A = (A_l, A_r)$, avec $A_l = (a_1, b_1)$ et $A_r = (a_2, b_2)$

Nous nous mettons dans le cadre d'observations de la méthode 1.

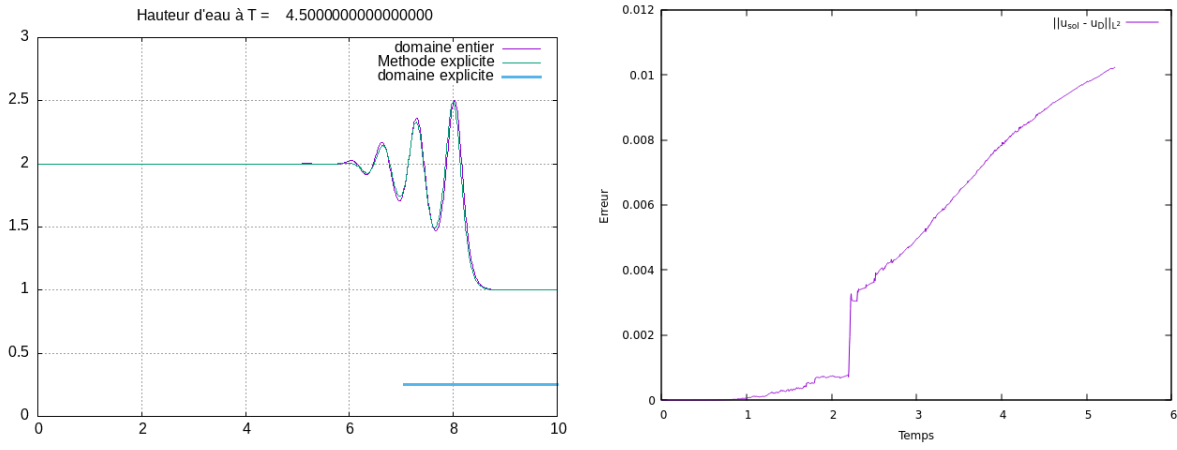


FIGURE 20 – Superposition entre la solution exacte et l'approximation avec $\tau = 0.7$ (à gauche). Norme L^2 de l'erreur de cette même approximation, en fonction du temps (à droite).

La figure 20 se décompose en deux graphes. Le graphe de gauche se décompose en trois courbes, la courbe violette représente la solution obtenue à partir de la projection sur tout le domaine, la courbe verte représente l'approximation obtenue par projection sur \mathcal{D} (avec $\tau = 0.7$), et la courbe bleue représente le domaine \mathcal{D} calculé à partir de la méthode 2. Le graphe de droite représente l'évolution en temps de la norme L^2 sur le domaine Ω de la différence entre la méthode de projection sur le domaine entier et la méthode par projection locale sur \mathcal{D} .

Sur le graphe de gauche, les courbes verte et violette sont très proches. Ce résultat est également montré par le graphe de droite, qui montre une erreur croissante, montant jusqu'à 10^{-2} . Cette erreur est plus élevée que celle observée sur la méthode itérative qui montrait une erreur ne dépassant pas 10^{-3} (figure 15), mais elle reste moins grande que l'erreur prélevée sur la méthode présentée en section 4.3.1, qui montrait une erreur allant au-delà de 10^{-1} (figure 16).

Cette méthode semble plus précise que la méthode explicite précédente, ce qui est lié à une meilleure estimation explicite \mathcal{D} du domaine \mathcal{D}_{opt} . Il reste cependant à voir les limites de cette approche.

Variation de la tolérance d'erreur $\bar{\epsilon}$

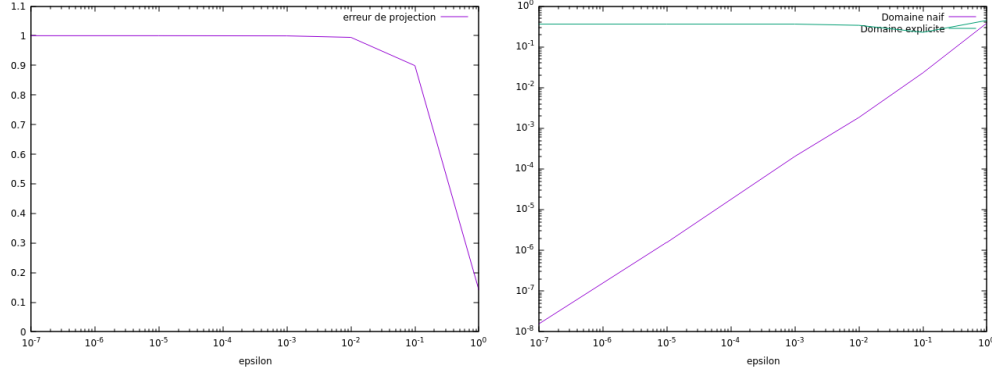


FIGURE 21 – Valeur de $\hat{\epsilon}$ (à gauche) et valeur de Err_{proj} (à droite), en fonction de la tolérance $\bar{\epsilon}$

La Figure 21 se décompose en 2 graphes, le graphe de gauche représente la valeur de $\hat{\epsilon}$ en fonction de la tolérance $\bar{\epsilon}$, et le graphe de droite représente la valeur de Err_{proj} en fonction de la tolérance, la courbe verte la valeur observé en utilisant le domaine de projection \mathcal{D} , et la courbe violette représente Err_{proj} obtenue sur le même cas test en reprenant $\tau = 1$, mais utilisant le domaine de projection \mathcal{D}_{opt} .

Sur le graphe de gauche, on remarque que $\hat{\epsilon}$ augmente lorsque $\bar{\epsilon}$ diminue, ce qui montre que la méthode de calcul explicite du domaine n'est pas adaptée au cas où la tolérance $\bar{\epsilon}$ tend vers 0. L'explication de cela est donnée sur le graphe de droite, qui montre que l'utilisation de la projection sur \mathcal{D}_{opt} fait tendre l'erreur Err_{proj} vers 0 avec une vitesse de convergence de 1 lorsque la tolérance tend vers 0, mais que l'utilisation du domaine \mathcal{D} , fait apparaître la présence d'un minimum strictement positif de l'erreur de valeur 0.4, ce qui est un minimum plus élevé que sur la méthode 1 comme le montre la figure 17 pour un même cas test, ce qui n'est pas en adéquation en comparant les résultats obtenus sur les figure 16 et 20. La seule différence avec la figure 20 est la valeur de τ . Il est donc nécessaire d'en étudier les variations.

Variation de τ

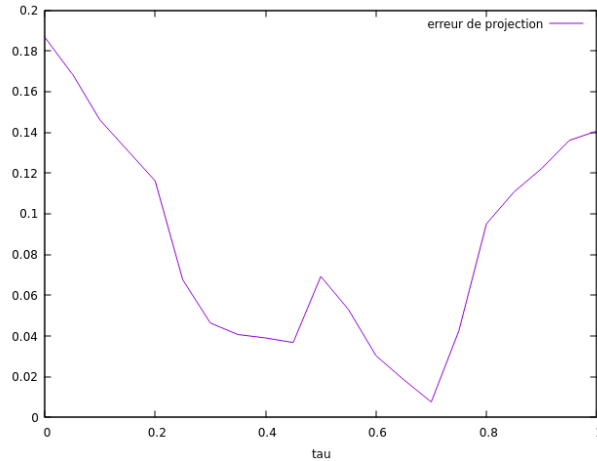


FIGURE 22 – Valeur de $\hat{\epsilon}$ en utilisant le domaine \mathcal{D} , en fonction de la valeur τ .

La Figure 22 représente la valeur de l'erreur de projection $\hat{\epsilon}$, sur un cas où le domaine explicite \mathcal{D} est

calculé à partir de différentes valeurs de τ .

Sur ce graphe, la valeur de τ fait varier la valeur de l'erreur de projection Err_{proj} et pour $\tau = 0.7$ on s'aperçoit que l'erreur est très faible (d'une valeur de l'ordre de 0.01), ce qui est également illustré sur la figure 20. Par comparaison avec la figure 18, l'erreur est plus forte pour la méthode 2 lorsque $\tau = 0.7$ ce résultat est en lien avec l'erreur relevée sur la figure 17 qui était plus élevée que sur la figure 21.

Dans la suite des observations, la valeur $\tau = 0.7$ sera sélectionnée car c'est la valeur qui nous donne une erreur la plus proche de l'erreur engendrée par le domaine optimal \mathcal{D}_{opt} .

Variation du nombre de cellules :

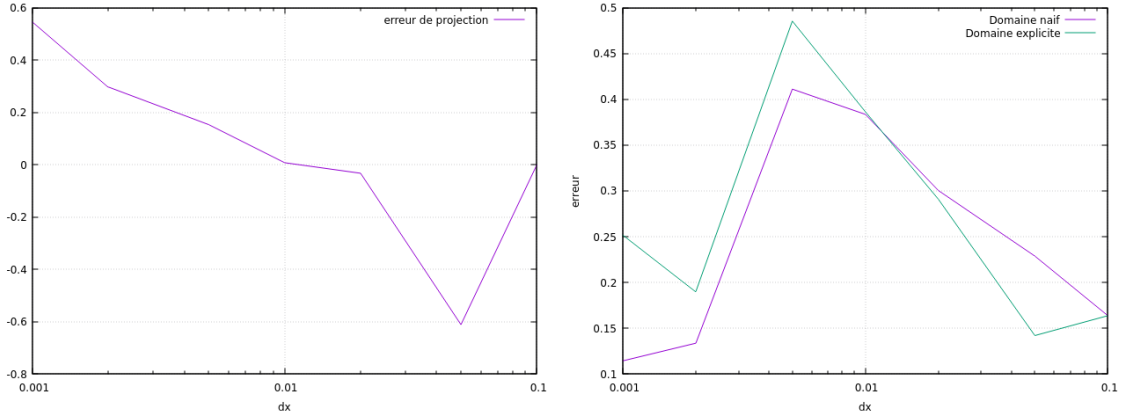


FIGURE 23 – Valeur de $\hat{\epsilon}$ (à gauche) et valeur de Err_{proj} (à droite), en fonction du maillage utilisé, avec $\tau = 0.7$

La figure 23 est composée de deux graphes. Le graphe de gauche représente la valeur de $\hat{\epsilon}$ en fonction du nombre de cellules dans le domaine. Le graphe de droite représente l'erreur de projection Err_{proj} sur différents maillages avec en violet l'erreur obtenue à l'aide d'un domaine itératif \mathcal{D}_{opt} , et en vert celle obtenue à partir d'un domaine explicite \mathcal{D} .

Sur le graphe de gauche, l'erreur $\hat{\epsilon}$ est négative pour des maillages avec des cellules de longueur plus grande que $\delta_x = 10^{-2}$, la projection sur \mathcal{D} donne un résultat globalement plus précis que celle sur \mathcal{D}_{opt} . Ce résultat est également présent sur le graphe de droite.

En comparant ces résultats avec ceux de la méthode 1 présents sur la figure 19, il apparaît que les résultats obtenus à l'aide de \mathcal{D} de la seconde méthode sont plus précis, et que la méthode est utilisable pour des maillages grossiers. Cependant il faudrait une plus grande base de données pour montrer si la méthode est utilisable pour des maillages plus fins. Nous avons donc ici une contrainte technique qui nous empêche de voir si le résultat serait tout autant efficace pour des maillages plus fins..

Cette méthode est donc utilisable mais peu utile pour notre travail puisque les cas qui nous intéressent sont ceux possédant des maillages raffinés, puisqu'ils donnent une approche plus précise du résultat, et que les temps de calculs sont plus longs.

4.4 Méthode par majoration : $\mathcal{D} = [\bar{x}_\tau - a_{sup}\bar{\sigma}_\tau ; \bar{x}_\tau + a_{sup}\bar{\sigma}_\tau]$

Le but de ce troisième cas d'observation consiste à majorer le domaine optimal \mathcal{D}_{opt} à l'aide d'un domaine \mathcal{D} , qui est centré en \bar{x}_τ . Pour ce faire, on note :

- $\bar{\sigma}_{\tau,i}$, l'écart type empirique de l'erreur relevé à l'instant t_i ,
- $\bar{x}_{\tau,i}$, la moyenne empirique de l'erreur à l'instant t_i ,
- $\mathcal{D}_{opt,i} = [x_{l,i}, x_{r,i}]$, le domaine optimal relevé à un instant t_i .

Alors a_{sup} est donné par la formule

$$a_{sup} = \max_i \left(\max \left(\frac{|x_{r,i} - \bar{x}_{\tau,i}|}{\sigma_{\tau,i}}, \frac{|\bar{x}_{\tau,i} - x_{l,i}|}{\sigma_{\tau,i}} \right) \right).$$

Nous nous mettons dans le cadre d'observations de la section 4.3.

Dans le cadre de l'analyse paramétrique de cette méthode, le paramètre τ ne sera pas observé, et fixé à valeur 0.

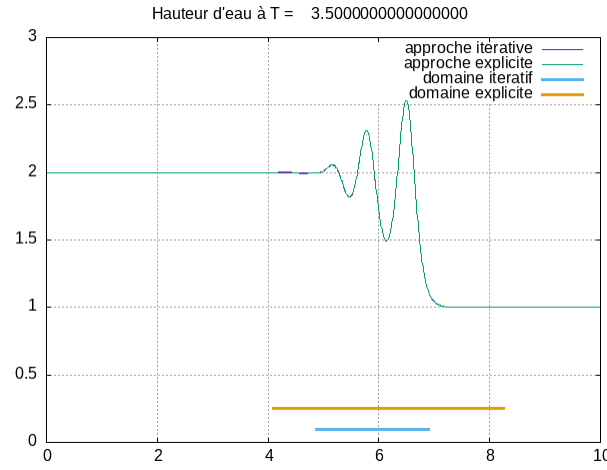


FIGURE 24 – Superposition entre la solution exacte et l'approximation

La figure 24 se décompose en 4 courbes, la courbe violette représente la solution obtenue à partir de la projection sur tout le domaine, la courbe verte représente l'approximation obtenue par projection sur \mathcal{D} la courbe bleue représente le domaine \mathcal{D}_{opt} , et la courbe jaune représente \mathcal{D} , obtenu à l'aide de la méthode par majoration.

On aperçoit que les courbes vertes et violettes sont confondues, et que le domaine itératif (en bleu) est plus petit que le domaine obtenu de manière explicite (en jaune). La superposition nous informe que cette méthode donne une approximation proche de celui utilisant une projection sur tout le domaine (dont l'on quantifiera l'erreur d'approximation dans la section 4.5). Ces deux résultats sont liés à la construction du domaine \mathcal{D} qui contient le domaine \mathcal{D}_{opt} .

Variation de la tolérance d'erreur $\bar{\epsilon}$:

La figure 25 se décompose en deux graphes. Le graphe de gauche est constitué de deux courbes. La courbe violette représente la valeur de Err_{proj} calculée en utilisant le domaine de projection \mathcal{D} sous divers seuils de tolérance $\bar{\epsilon}$, et la courbe verte représente la tolérance $\bar{\epsilon}$, en fonction de la tolérance $\bar{\epsilon}$, c'est la courbe d'une fonction identité. Le graphe de droite représente la valeur du coefficient a_{sup} en fonction de la tolérance $\bar{\epsilon}$.

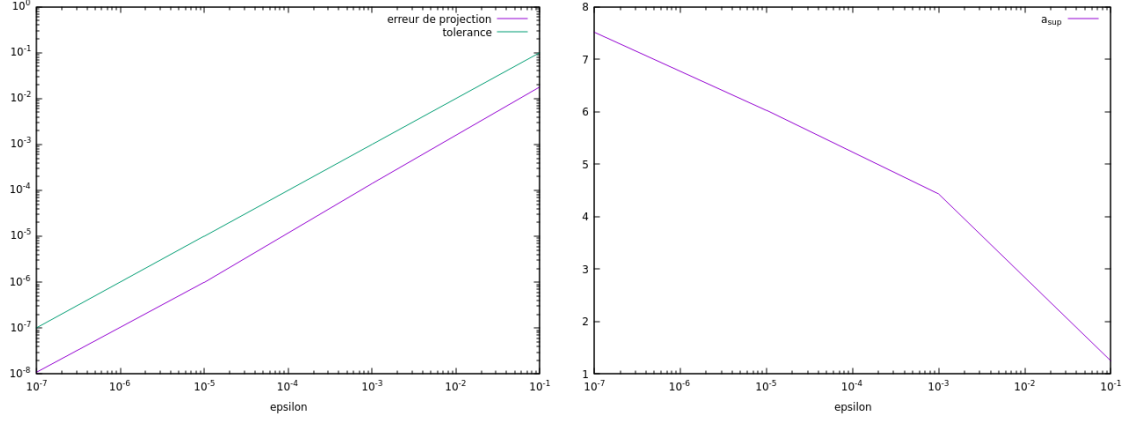


FIGURE 25 – Erreur de projection Err_{proj} en fonction de la tolérance $\bar{\epsilon}$ (à gauche). Valeur du coefficient a_{sup} en fonction de la tolérance $\bar{\epsilon}$ (à droite)

Sur le graphique de gauche, l'erreur Err_{proj} est plus faible que la tolérance $\bar{\epsilon}$. Le schéma converge à l'ordre 1 vers le schéma de résolution de l'équation de BBM-KdV sur l'ensemble du domaine, lorsque la tolérance $\bar{\epsilon}$ tend vers 0. Sur le graphe de droite, la réduction de la tolérance a pour effet d'augmenter la valeur du coefficient a_{sup} , ce qui a pour effet d'agrandir le domaine \mathcal{D} (puisque la valeur de $\tau = 0$, l'écart type σ_τ est le même pour toute tolérance), et ainsi projeter sur un plus grand domaine, ce qui a pour effet de réduire l'erreur de projection Err_{proj} , comme l'atteste le graphe de gauche. Ce résultat est en concordance avec celui présenté dans la figure 11 qui montre que la réduction de la tolérance $\bar{\epsilon}$ a pour effet d'augmenter la longueur de \mathcal{D}_{opt}

Variation du nombre de cellules :

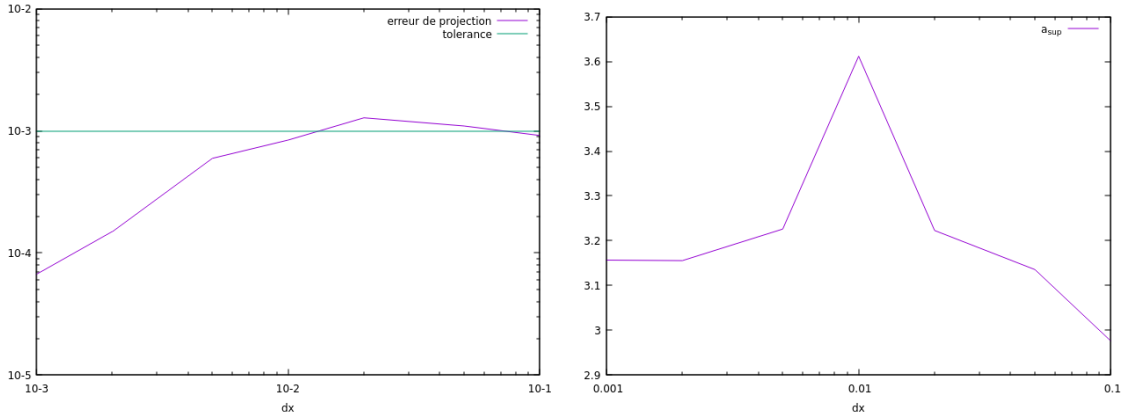


FIGURE 26 – Valeur de Err_{proj} selon différents maillages (à gauche). Valeur de a_{sup} selon différents maillages (à droite).

La figure 26 est décomposée en deux graphes. Sur le graphe de gauche, la courbe violette représente l'erreur de projection Err_{proj} en fonction du maillage, et la courbe verte représente la tolérance d'erreur $\bar{\epsilon}$. Sur le graphe de droite, il est représenté la valeur du coefficient a_{sup} sur différents maillages.

Sur le graphe de gauche, il apparait que l'erreur de projection Err_{proj} diminue lorsque le maillage se raffine. Sur le graphe de droite, il apparait que la valeur de a_{sup} varie peu entre les différents maillages observés

(a_{sup} prends des valeurs entre 2.95 et 3.25), excepté en $\delta_x = 0.01$ où le coefficient augmente pour prendre la valeur 3.7. De plus, pour une valeur de a_{sup} égale sur deux maillages, l'erreur de projection Err_{proj} diminue lorsque le maillage se raffine.

4.5 Temps de calcul de la méthode explicite par majoration

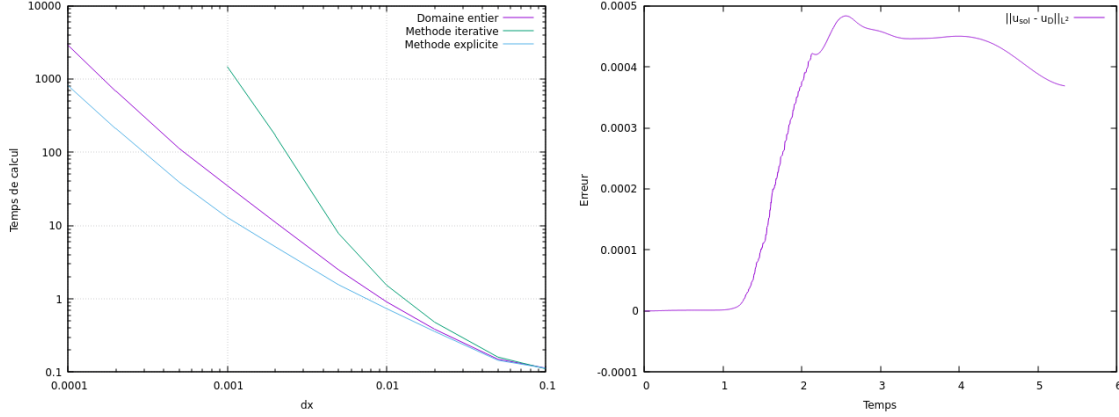


FIGURE 27 – Comparasion entre les temps de calculs de différentes méthodes (à gauche). Norme L^2 de l'erreur d'approximation de la méthode explicite par majoration, en fonction du temps (à droite)

La figure 27 se décompose en deux graphes obtenus à partir de la résolution numérique du cas test de la section 3.2.2 avec une durée de $T = 5$. Le graphe de gauche est composée de trois courbes, la courbe violette représente le temps de calcul obtenu sur la projection d'un domaine entier (c'est la même courbe que celle présente en figure 6), et la courbe verte représente le temps de calcul obtenu pour résoudre le problème à l'aide de la méthode itérative (c'est la même courbe que celle présente en figure 15), et la courbe bleue représente le temps de calcul avec une projection opérée sur un domaine \mathcal{D} obtenu à l'aide de la méthode explicite par majoration étudiée en section 4.4, les observations seront concentrées sur cette dernière. Le graphe de droite représente l'erreur en norme L^2 de la différence entre le résultat en u de la méthode par projection sur tout le domaine Ω et, le résultat en u de obtenu par projection sur ce même domaine \mathcal{D} . Cette différence est étudiée en fonction du temps.

Sur le graphe de gauche, la courbe bleue, croit le moins vite parmi l'ensemble des courbes, ce qui signifie que le temps de calcul pour cette méthode explicite est moins grand que la méthode de projection sur tout le domaine. Par exemple, pour 100 000 cellules, ce qui représente $\delta_x = 10^{-4}$, le temps de calcul est de 2900 secondes pour la méthode de projection sur le domaine entier, tandis qu'il est de 825 secondes pour la méthode explicite.

Sur Le graphe de droite, l'erreur croit en fonction du temps. L'erreur observée est faible et admet des valeurs proche (avec parfois des valeurs moins élevées) de celles observées pour la méthode itérative (figure 15). Ce graphe d'erreur nous montre alors que la méthode explicite de la section 4.4 est une méthode précise pour résoudre numériquement le problème (1).

La méthode de détermination explicite du domaine de projection par un principe de majoration, présentée dans la section 4.4, est donc une approche rapide et précise pour résoudre numériquement BBM-KdV (4).

5 Conclusion

Dans ce rapport de stage nous avons présenté, que l'équation de BBM-KdV peut s'écrire sous une forme hyperbolique avec un terme source réalisant une projection sur un sous espace vectoriel de L^2 . D'un point de vue numérique, cette nouvelle formulation se résouds par deux étapes (la première consiste en la résolution d'une étape hyperbolique, et la seconde étape est une projection sur un sous espace vectoriel) s'effectuant successivement pour chaque pas de temps.

Le temps de calcul de la résolution est principalement occupé par la seconde étape de résolution qui nécessite la résolution d'un système matriciel. Pour réduire le temps de calcul on se concentre sur l'étape de projection sur un sous-domaine de l'espace observé. L'introduction d'un critère d'erreur a été effectué pour le délimiter. La mise en place de ce sous-domaine a été réalisé de plusieurs méthodes dont certaines n'ont put être concluante pour des raisons techniques, cependant deux d'entre elles ont été retenues, pour des raisons différentes.

La première est une approche itérative qui, en dépit de son temps de calcul trop élevé, nous a donné une estimation du domaine optimal et nous a permis d'obtenir des valeurs objectifs pour les bords du domaine. La seconde méthode est une méthode explicite qui repose sur une majoration du domaine optimal qui approxime numériquement la résolution d'un problème de KdV-BBM plus rapidement que l'une projection sur l'ensemble du domaine (ce qui représente la méthode de résolution numérique par défaut), permettant de répondre à nos attendus de stages.

Le modèle de KdV-BBM est un modèle simplifié, ne prenant pas en compte de la physique. Pour des approches plus concrètes, on pourrait regarder le résultat que peut donner cette équation sur différentes EDP, par exemple Green-Naghdi, qui peut s'écrire à l'aide de l'équation de Shallow-Water pour la partie hyperbolique avec une projection sur un sous espace vectoriel de L^2 . On pourrait, dans ce cas également observer comment définir le domaine optimal dans le cas plusieurs vagues (en le décomposant en plusieurs domaines optimaux par exemple), comment le caractériser dans un espace de plusieurs dimensions, ou de traiter un cas avec une bathymétrie.