HAL
archives-ouvertes.fr

Une Methode de partition de domaine pour l'equation d'advection-diffusion

Pierre Aubert, Marie-Claude Ciccoli, Jean-Antoine Desideri

▶ **To cite this version:**

Pierre Aubert, Marie-Claude Ciccoli, Jean-Antoine Desideri. Une Methode de partition de domaine pour l'equation d'advection-diffusion. [Rapport de recherche] RR-1740, INRIA. 1992. <inria-00076979>

HAL Id: inria-00076979
https://hal.inria.fr/inria-00076979

Submitted on 29 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNE METHODE DE PARTITION DE DOMAINE APPLIQUEE A UNE EQUATION D'ADVECTION-DIFFUSION

Rapport pour le
DEA DE MATHEMATIQUES
Option : ANALYSE NUMERIQUE
Encadrement : M. Désidéri † et M.C. Ciccoli †
† Projet Sinus INRIA Sophia-Antipolis

Auteur : Pierre Aubert

25 Juin 1992

Abstract

Les méthodes de décomposition de domaine pour résoudre les équations aux dérivées partielles sont particulièrement adaptées au calcul parallèle.Elles sont basées sur le découpage du domaine de calcul en sous-domaines plus petits, sur lesquels on résout des sous-problèmes indépendants de taille réduite.

Ce rapport présente l'étude d'un modèle linéaire d'advection-diffusion en une puis deux dimensions d'espace. Il comporte une étude théorique du problème modèle et des résultats numériques.

Table des matières

Table des figures

Chapitre 1

Introduction

1.1 Motivation

L'avènement des ordinateurs parallèles a motivé le développement de nouveaux algorithmes de résolution des équations aux dérivées partielles. D'un coté, les algorithmes classiques ont été adaptés aux nouvelles architectures, de l'autre on développe de nouvelles méthodes, intrinsèquement parrallélisables, dans l'espoir de résoudre en les décomposant en une suite de problèmes plus simples pour lesquels des méthodes très efficaces existent.

Les méthodes de décomposition de domaine sont un des axes de cette recherche. Elles consistent en un découpage du domaine de calcul en sous-domaines sur lesquels on résoudra des sous-problèmes indépendants de taille réduite.

Les avantages espérés sont nombreux :

- En découpant de gros problèmes, on peut espérer résoudre des problèmes complexes hors d'atteinte pour les ordinateurs scalaires actuels.

- Le fait de résoudre des problèmes plus petits améliore le conditionnement et donc accèlere les vitesses de convergence des algorithmes de résolution des équations aux dérivées partielles.

- On peut résoudre des équations différentes sur chacun des sous-domaines en fonction des caractéristiques physiques du problème (partition par modèle physique).

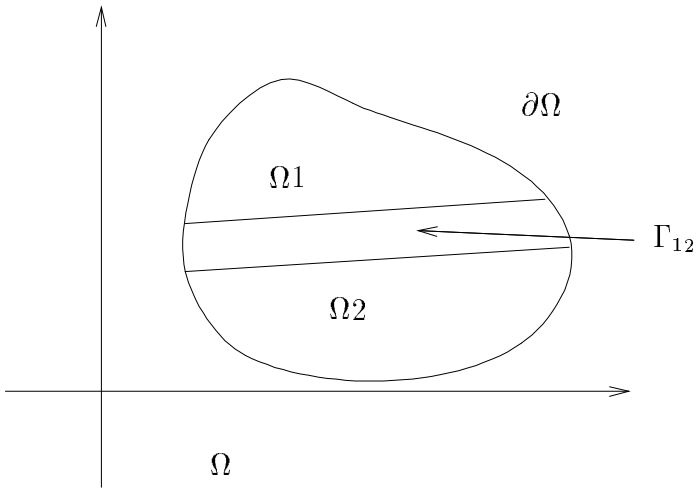


Figure 1.1 : Recouvrement

- On espère pouvoir exploiter les différents processeurs des machines parallèlles (par exemple Meiko) de manière efficace.

1.2 Méthodes générales de décomposition de domaine

Soit Ω un domaine ouvert de $I\!R^2$ de frontière $\partial\Omega$. On considère le problème suivant :

$$Lu = f \text{ dans } \Omega$$

où L est un opérateur différentiel, f un terme source donné et u l'inconnue. On doit rajouter au problème précédent des conditions aux limites, ansi que des conditions initiales.

Partitionons alors Ω en deux sous-domaines pour simplifier:

Il existe deux méthodes principales:

- Le recouvrement (Cf figure 1.1) où les deux domaines Ω_1 et Ω_2 empiètent l'un sur l'autre et où alors Γ_{12} est une surface.

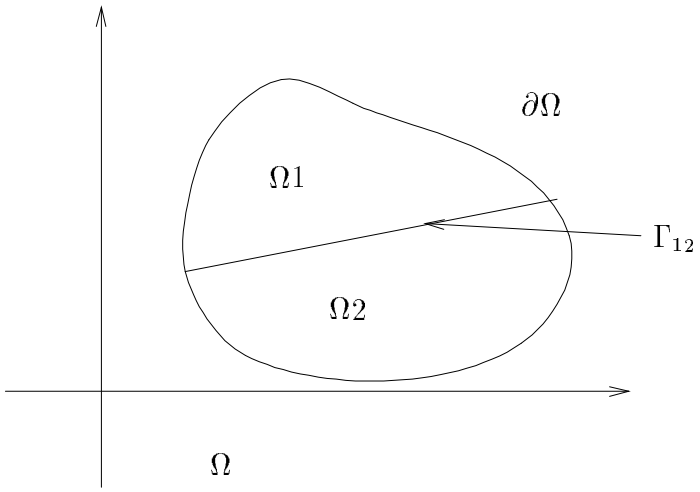


Figure 1.2 : Partition

- La partition (Cf figure 1.2) où le domaine Ω est découpé en deux parties distinctes Ω_1 et Ω_2 de frontière commune Γ_{12}.

Cette étude se limite à une technique de partition.

1.3 Présentation de l'étude

Les méthodes de décomposition de domaine ont été appliquées à de nombreux problèmes, le recouvrememt, par exemple, est étudié à l'INRIA par [CDP92], la partition a été traitée pour des problèmes elliptiques par [GPD83] et [Qua91].

Par contre, pour les équations qui ne sont pas de type elliptique, la littérature est beaucoup plus pauvre.

Nous avons donc commencé par étudier un problème modèle : on a choisi l'équation d'advection-diffusion d'abord avec une dimension d'espace puis deux.

Les objectifs sont les suivants :

1. Peut-on résoudre une équation non-elliptique par un algorithme de partition ?

2. Quel type de convergence obtient-on ?

3. Peut-on tirer profit d'une architecture parallèle ?

Chapitre 2

Problème monodimensionnel

1

2.1 Etude théorique

2.1.1 Position du problème

On veut résoudre le problème modèle suivant : c'est une équation d'advection-diffusion avec des conditions au bord de type Dirichlet et-ou de type Neumann et un terme source s.On désire une solution globale sur l'intervalle $[0,1]$. Le système s'écrit alors :

$$\begin{cases} U_t + cU_x - \nu U_{xx} = s \\ 0 \leq x \leq 1 \\ + \text{ des conditions aux bords } x = 0 \text{ et } x = 1 \\ + \text{ une condition initiale } u(x,0) \text{ donné} \end{cases} \tag{2.1}$$

On pose alors

$$U_t = \frac{U^{n+1} - U^n}{\Delta t}$$

et on ramène la résolution du problème complet à une succession de problèmes à chaque pas de temps. On ne s'intéresse ici qu'à la résolution de ce problème. Le système qui donne $U^{n+1} = U$ est alors le suivant :

$$\begin{cases} \alpha U + cU_x - \nu U_{xx} = f, \forall 0 \leq x \leq 1 \\ + \text{ des conditions aux bords en } x = 0 \text{ et en } x = 1 \end{cases} \tag{2.2}$$

où $\alpha = \frac{1}{\Delta t}$ et $f = s + \alpha U^n$.

Dans ce qui suit, on choisit d'étudier le problème (2.3).

$$\begin{cases} \alpha U + cU_x - \nu U_{xx} = f \\ U(0) = 1 \\ U(1) = 0 \\ 0 \leq x \leq 1 \end{cases} \tag{2.3}$$

2.1.2 Ecriture du système en partition

On veut résoudre séparément sur l'intervalle $[0, \frac{1}{2}]$ et sur $[\frac{1}{2}, 1]$. On introduit la valeur v à l'interface, $x = \frac{1}{2}$.On résout alors les deux sous-problèmes suivants :

$$\begin{cases} \alpha U^{(1)} + cU_x^{(1)} - \nu U_{xx}^{(1)} = f \\ U^{(1)}(0) = U_0 \\ U^{(1)}(\frac{1}{2}) = v \\ 0 \leq x \leq \frac{1}{2} \end{cases} \tag{2.4}$$

et

$$\begin{cases} \alpha U^{(2)} + cU_x^{(2)} - \nu U_{xx}^{(2)} = f \\ U^{(2)}(\frac{1}{2}) = v \\ U^{(2)}(1) = 0 \\ \frac{1}{2} \leq x \leq 1 \end{cases} \tag{2.5}$$

La fonction U dont $U^{(1)}$ et $U^{(2)}$ sont les restrictions aux intervalles $[0, \frac{1}{2}]$ et $[\frac{1}{2}, 1]$ respectivement est continue et satisfait l'équation différentielle par morceaux. Ce n'est cependant pas une solution "lisse" du problème initial, à moins que la dérivée soit elle-même continue. On est donc amené à chercher la valeur de v qui minimise le critère :

$$J(v) = \frac{1}{2} \left(U_x^{(1)}(\frac{1}{2}) - U_x^{(2)}(\frac{1}{2}) \right)^2 \tag{2.6}$$

Pour cela, on itère sur v et afin de construire une méthode de descente, il convient d'évaluer, le gradient $\nabla J(v)$ ce qui est fait dans la section suivante.

2.1.3 Evaluation du gradient

Pour évaluer le gradient de $J(v)$, on perturbe le contrôle v de δv et on évalue $\delta U^{(1)}$, $\delta U^{(2)}$ et finalement δJ.
$\delta U^{(1)}$ et $\delta U^{(2)}$ sont les solutions des deux systèmes linéarisés suivant :

$$\begin{cases} \alpha\delta U^{(1)} + c\delta U_x^{(1)} - \nu\delta U_{xx}^{(1)} = 0 \\ \delta U^{(1)}(0) = 0 \\ \delta U^{(1)}(\frac{1}{2}) = \delta v \\ 0 \leq x \leq \frac{1}{2} \end{cases} \tag{2.7}$$

et

$$\begin{cases} \alpha\delta U^{(2)} + c\delta U_x^{(2)} - \nu\delta U_{xx}^{(2)} = 0 \\ \delta U^{(2)}(\frac{1}{2}) = \delta v \\ \delta U^{(2)}(1) = 0 \\ \frac{1}{2} \leq x \leq 1 \end{cases} \tag{2.8}$$

Or, on a :

$$\delta J(v) = \left(U_x^{(1)}(\frac{1}{2}) - U_x^{(2)}(\frac{1}{2})\right)\left(\delta U_x^{(1)}(\frac{1}{2}) - \delta U_x^{(2)}(\frac{1}{2})\right)$$

Il convient donc d'exprimer $\delta U_x^{(1)}(\frac{1}{2})$ et $\delta U_x^{(1)}(\frac{1}{2})$ en fonction de δv. Pour cela on utilise la technique classique d'élimination par l'équation adjointe. On introduit un état adjoint $\lambda(x)$ à $\delta U^{(1)}$ et un état adjoint $\mu(x)$ à $\delta U^{(2)}$.On effectue le produit scalaire de (2.7) par $\lambda(x)$ et de (2.7) par $\mu(x)$ et on intègre par partie, ce qui donne :

Calcul pour le système 2.7 ,
On écrit le résultat de l'intégration de chaque terme :

$$\begin{aligned}
\int_0^{\frac{1}{2}} \lambda\delta U^{(1)} &= \text{idem} \\
\int_0^{\frac{1}{2}} \lambda\delta U_x^{(1)} &= \left[\lambda\delta U^{(1)}\right]_0^{\frac{1}{2}} - \int_0^{\frac{1}{2}} \lambda_x\delta U^{(1)} \\
&= \lambda(\tfrac{1}{2})\delta v - \int_0^{\frac{1}{2}} \lambda_x\delta U^{(1)} \\
\int_0^{\frac{1}{2}} \lambda\delta U_{xx}^{(1)} &= \left[\lambda\delta U_x^{(1)}\right]_0^{\frac{1}{2}} - \int_0^{\frac{1}{2}} \lambda_x\delta U_x^{(1)} \\
&= \lambda(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) - \lambda(0)\delta U_x^{(1)}(0) \\
&\quad - \left[\lambda_x\delta U^{(1)}\right]_0^{\frac{1}{2}} + \int_0^{\frac{1}{2}} \lambda_{xx}\delta U^{(1)} \\
&= \lambda(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) - \lambda_x(\tfrac{1}{2})\delta v + \int_0^{\frac{1}{2}} \lambda_{xx}\delta U^{(1)}
\end{aligned}$$

On regroupe ces expressions :

$$\begin{array}{lccc}
\int_0^{\frac{1}{2}} & (\alpha\lambda - c\lambda_x - nu\lambda_{xx}) & \delta U^{(1)} & \\
+ & (\nu\lambda(0)) & \delta U_x^{(1)} & \\
+ & c\lambda\tfrac{1}{2}\delta v + \nu\lambda_x(\tfrac{1}{2})\delta v - \nu\lambda(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) & & = 0
\end{array}$$

On choisit alors λ tel que :

$$\begin{cases}
\alpha\lambda - c\lambda_x - \nu\lambda_{xx} = 0 \\
\lambda(0) = 0 \\
\lambda(\frac{1}{2}) = 1 \\
0 \leq x \leq \frac{1}{2}
\end{cases} \tag{2.9}$$

On a alors :

$$\delta U_x^{(1)}(\frac{1}{2}) = \frac{c}{\nu}\delta v + \lambda_x(\frac{1}{2})\delta v \tag{2.10}$$

Calcul pour le système 2.8 ,
On écrit le résultat de l'intégration de chaque terme :

$$\begin{aligned}
\int_{\frac{1}{2}}^1 \mu\delta U^{(1)} &= \text{idem} \\
\int_{\frac{1}{2}}^1 \mu\delta U_x^{(1)} &= \left[\mu\delta U^{(1)}\right]_{\frac{1}{2}}^1 - \int_{\frac{1}{2}}^1 \lambda_x\delta U^{(1)} \\
&= -\mu(\tfrac{1}{2})\delta v - \int_{\frac{1}{2}}^1 \lambda_x\delta U^{(1)} \\
\int_{\frac{1}{2}}^1 \mu\delta U_{xx}^{(1)} &= \left[\mu\delta U_x^{(1)}\right]_{\frac{1}{2}}^1 - \int_{\frac{1}{2}}^1 \lambda_x\delta U_x^{(1)} \\
&= -\mu(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) - \mu(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) \\
&\quad - \left[\lambda_x\delta U^{(1)}\right]_{\frac{1}{2}}^1 + \int_{\frac{1}{2}}^1 \lambda_{xx}\delta U^{(1)} \\
&= -\mu(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) - \mu_x(1)\delta v + \int_{\frac{1}{2}}^1 \mu_{xx}\delta U^{(1)}
\end{aligned}$$

On regroupe ces expressions :

$$\begin{array}{lccl}
\int_{\frac{1}{2}}^1 & (\alpha\mu - c\lambda_x - nu\lambda_{xx}) & \delta U^{(1)} & \\
+ & (\nu\mu(1)) & \delta U_x^{(1)} & \\
+ & c\mu\tfrac{1}{2}\delta v + \nu\lambda_x(\tfrac{1}{2})\delta v - \nu\mu(\tfrac{1}{2})\delta U_x^{(1)}(\tfrac{1}{2}) & & = 0
\end{array}$$

On choisit alors $\mu(x)$ tel que :

$$\begin{cases}
\alpha\mu - c\mu_x - \nu\mu_{xx} = 0 \\
\mu(1) = 0 \\
\mu(\frac{1}{2}) = 1 \\
\frac{1}{2} \leq x \leq 1
\end{cases} \tag{2.11}$$

On a alors :

$$\boxed{\delta U_x^{(2)}(\frac{1}{2}) = \frac{c}{\nu}\delta v + \mu_x(\frac{1}{2})\delta v} \tag{2.12}$$

Remarque Si on avait imposer une condition de Newmann en $x = 1$, on aurait obtenu

$$c\mu(1) + \nu\mu_x(1) = 0$$

comme condition au bord pour le système adjoint en μ.

En reportant les résultats de (2.10) et (2.12) dans l' expression de δJ, on obtient :

$$\begin{aligned} \delta J(v) &= K \times \delta v \\ K &= \left(U_x^{(1)}(\tfrac{1}{2}) - U_x^{(2)}(\tfrac{1}{2})\right)\left(\delta U_x^{(1)}(\tfrac{1}{2}) - \delta U_x^{(2)}(\tfrac{1}{2})\right) \end{aligned} \tag{2.13}$$

Ceci permet d'identifier le gradient K égal à $J^{'}(v)$.

2.1.4 Méthode de descente

La connaissance de la dérivée de $\nabla J(v)$ conduit à construire le nouvel itéré au moyen de la formule :

$$v^{n+1} = v^n - \frac{J(v^n)}{\nabla J(v^n)} \tag{2.14}$$

(Méthode de Newton appliquée à $J(v) = 0$).

2.2 Implémentation et résultats numériques

2.2.1 Algorithme

La programmation a été éffectué en Fortran sur un SUN et représente environ 300 lignes de code. L'algorithme implémenté est décrit sur la figure (2.2.1). La résolution de chaque sous-système se fait en différences finies centrées à l'ordre deux. Le calcul numérique de $\lambda_x(\frac{1}{2}) - \mu_x(\frac{1}{2})$ qui a une importance primordiale est explicité section suivante.

2.2.2 Analyse de la convergence

Importance de la fonctionelle J

La convergence de l'algorithme de Newton qui généralement est quadratique se dégrade lorsque le zéro de $J(v)$ cherché est aussi un zéro de ∇J. C'est fatalement ce qui se produit pour la fonctionelle (2.6) en vertu de (2.13).Sur les figures (2.3),(2.4), (2.5) et (2.7) on constate une convergence linéaire. La vitesse de convergence est d'environ 13 décades pour 10 itérations.

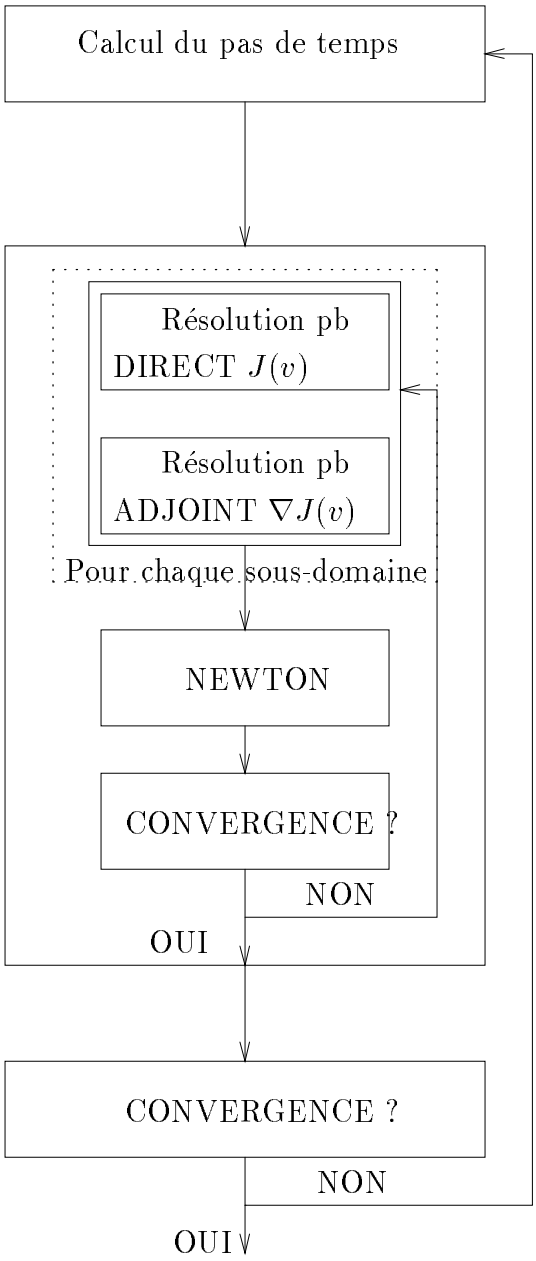
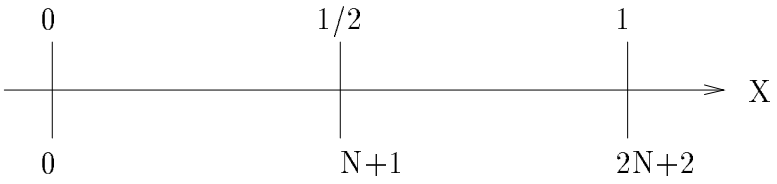


Figure 2.1 : Algorithme de résolution



Nombre de points de discrétisation.

Figure 2.2 : Schéma détaillant la discrétisation

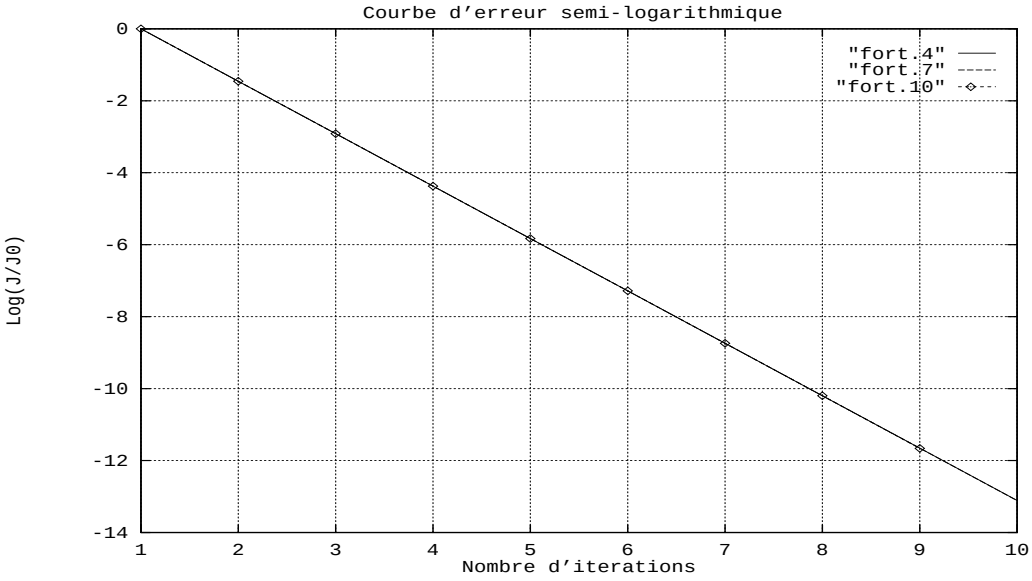


Figure 2.3 : $J = J_1$ Convergence linéaire

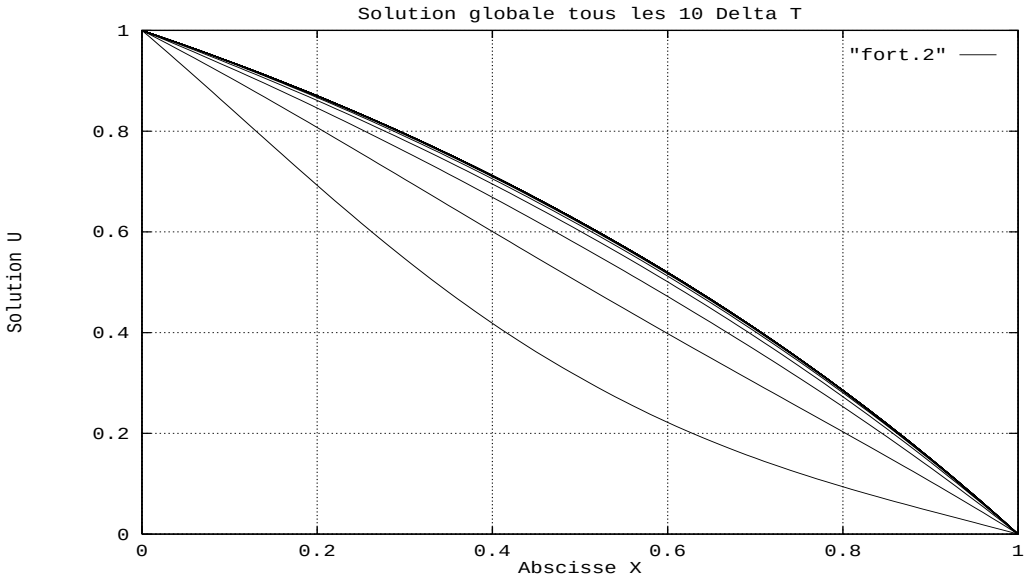


Figure 2.4 : $J = J_1$ Solution tous les dix pas de temps

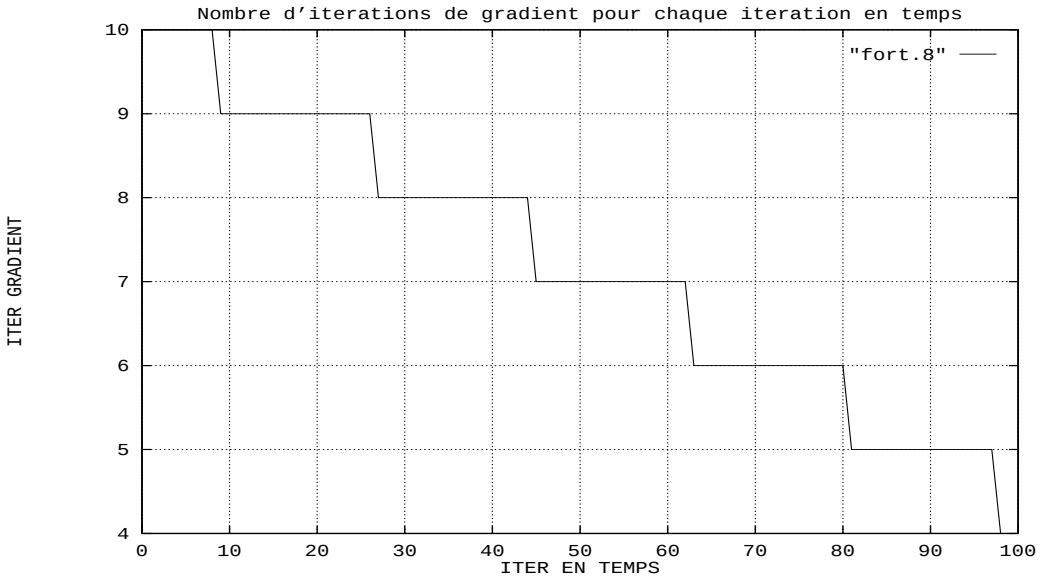


Figure 2.5 : $J = J_1$ Nombre d'itérations par itération de l'algorithme de descente

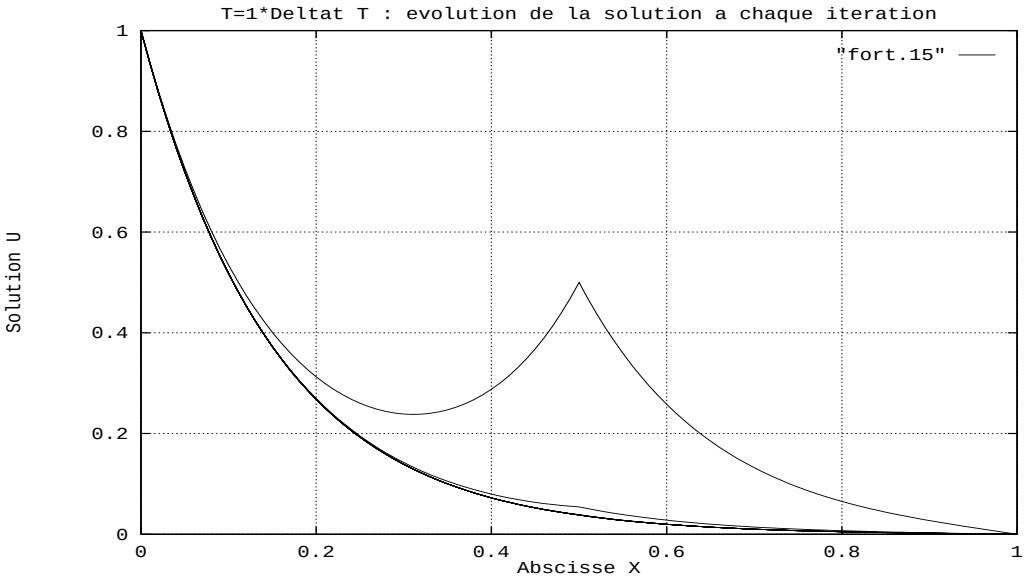


Figure 2.6 : $J = J_1$ Solution à $t = \Delta t$
La valeur intiale de l'interface a été choisie égale à la demi somme des valeurs des conditions aux limites de Dirichlet.

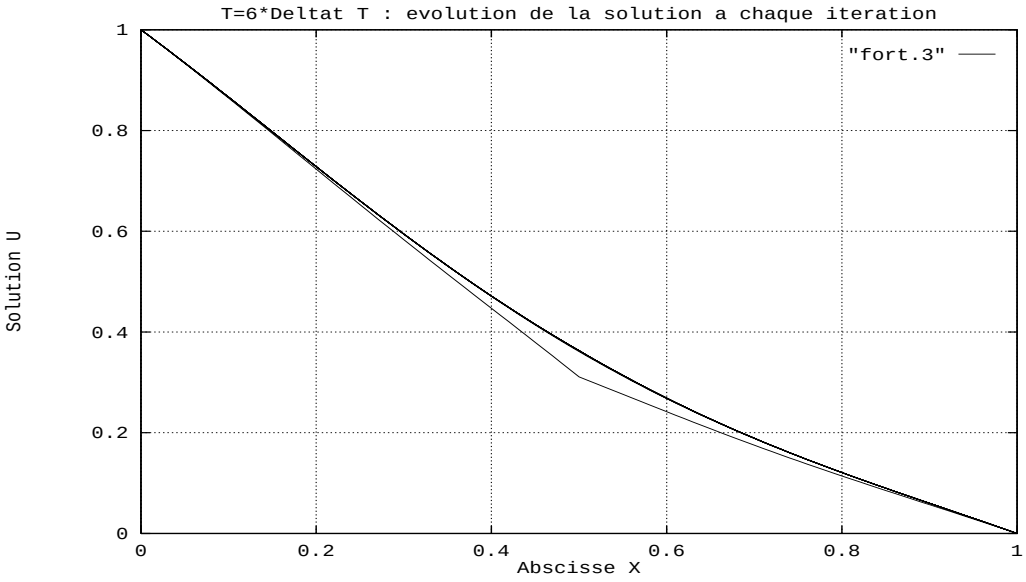


Figure 2.7 : $J = J_1$ Solution à $t = 6\Delta t$

Dans ce cas monodimensionel où on cherche à résoudre une seule équation scalaire $J(v) = 0$, on peut facilement contourner cette difficulté en donnent au critère la nouvelle définition suivante :

$$J = U_x^{(1)}(\tfrac{1}{2}) - U_x^{(2)}(\tfrac{1}{2}) \tag{2.15}$$

Alors, avec la même définition des systèmes adjoints, on a :

$$\delta J = \delta U_x^{(1)}(\tfrac{1}{2}) - \delta U_x^{(2)}(\tfrac{1}{2}) \neq 0 \tag{2.16}$$

et la convergence est quadratique.Comme J est en fait une fonction afffine de v la convergence est atteinte en exactement 2 itérations.

Importance de la discrétisation

Nous avons essayé trois approximations de la dérivée pour calculer $\lambda_x(\frac{1}{2}) - \mu_x(\frac{1}{2})$.

- Une différence centrée qui assure l'ordre 1 pour la dérivée ce qui est consistant avec le schéma (à l'ordre 2).

- Une différence centrée d'ordre 2

- La valeur explicite que l'on calcule analytiquement à partir de (α, ν, c).En effet les deux systèmes adjoints sont homogènes et linéaires.

Nous avons constaté une convergence quadratique seulement pour le premier cas, en effet les deux autres solutions dégénèrent vers une convergence linéaire.

Influence du nombre de Reynolds

Le calcul formel fournit :

$$\begin{aligned} \delta J(v) = & \quad \delta U_x^{(1)}(\tfrac{1}{2}) - \delta U_x^{(2)}(\tfrac{1}{2}) \\ = & \quad \frac{k}{\nu} \frac{sh(\frac{k}{2\nu})}{2sh^2(\frac{k}{4\nu})} \end{aligned} \tag{2.17}$$

avec $k = \sqrt{c^2 + 4\alpha\nu}$ On pose $R_e = \frac{ch}{\nu}$ et $h = 1$ On remarque que δJ ne dépend que de (α, c, ν). Quand $R_e \to \infty$, alors $\delta J \to 0$. donc **la convergence dégénère lorsqu'on approche le cas hyperbolique**.

Résultats quadratiques

On résout le même problème que précédement, mais ici avec

$$J(v) = U_x^{(1)}(\frac{1}{2}) - U_x^{(2)}(\frac{1}{2})$$

La convergence est alors quadratique : Cf figures (2.8),(2.9),(2.11) et (2.12).

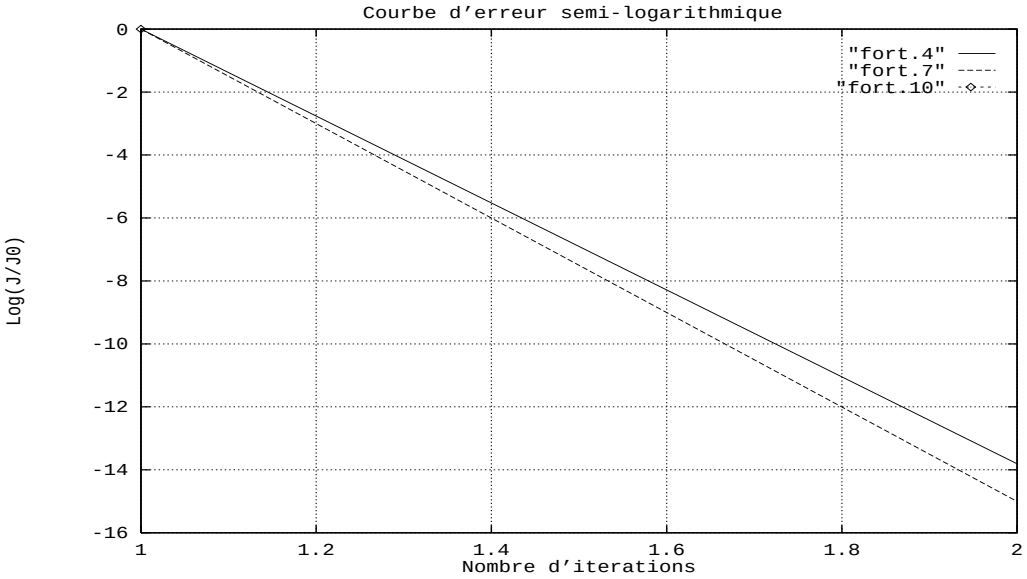


Figure 2.8 : $J = J_2$Erreur

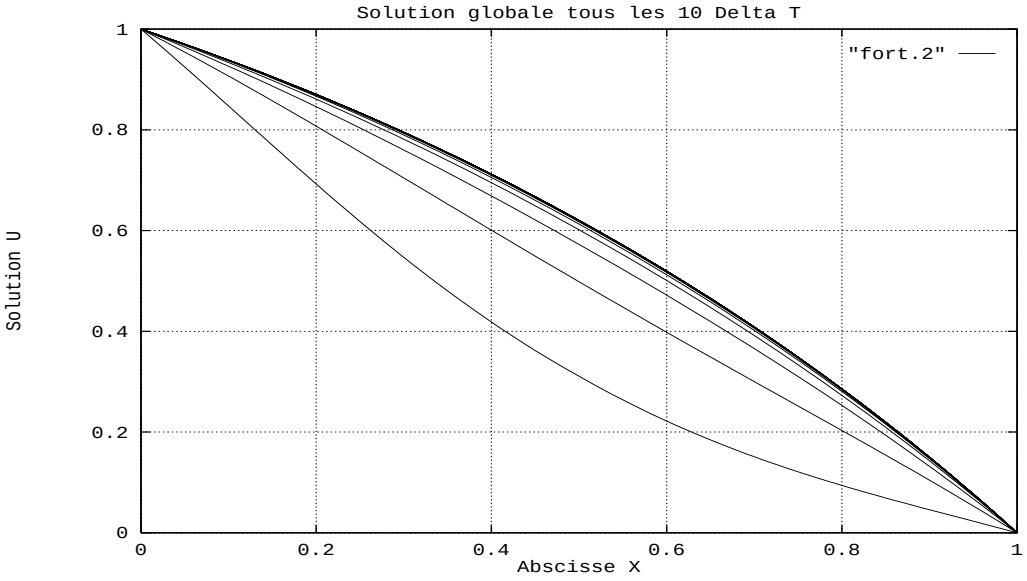


Figure 2.9 : $J = J_2$Solution tous les dix pas de temps

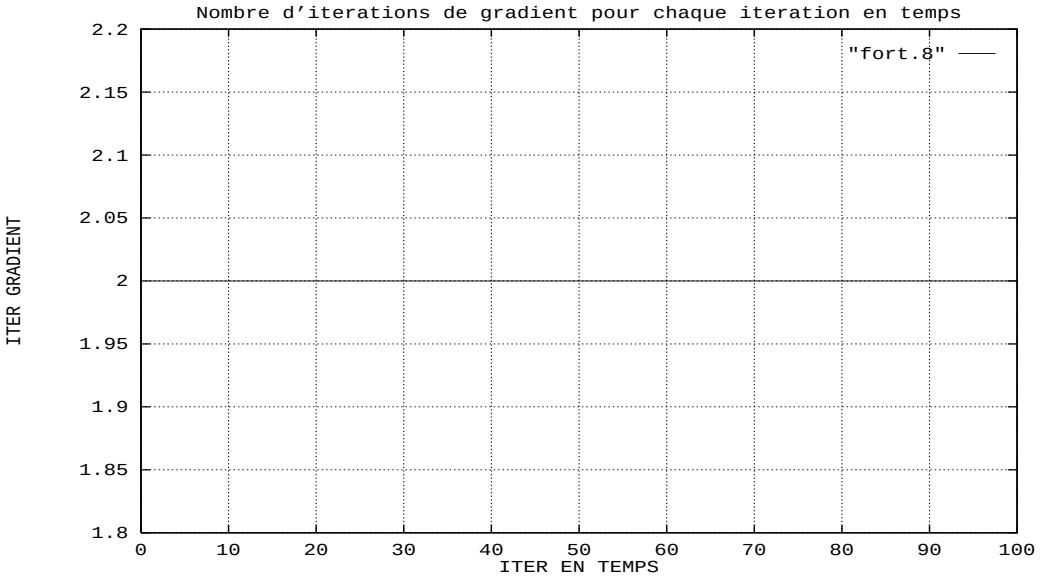


Figure 2.10 : $J = J_2$Nombre d'itérations

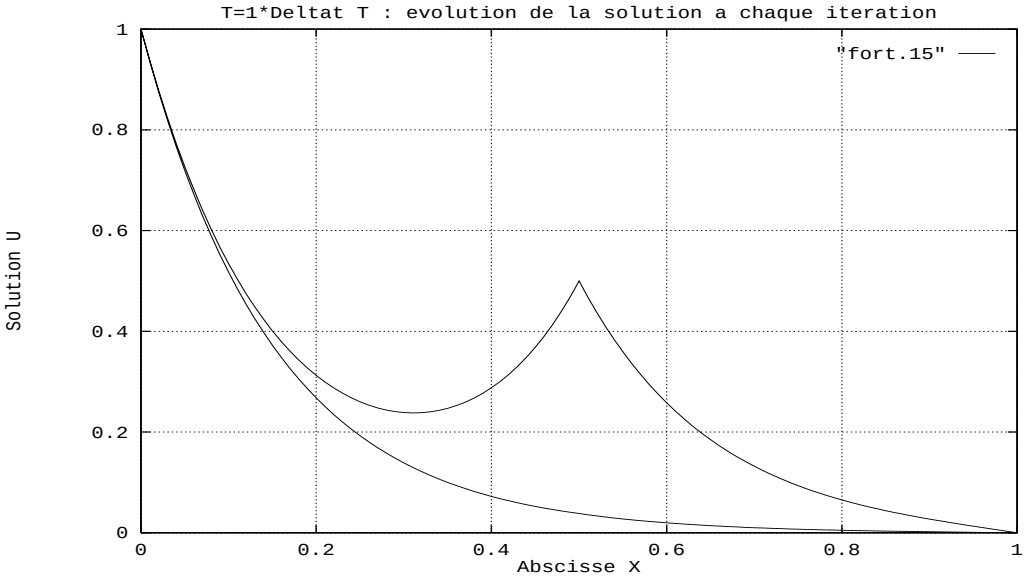


Figure 2.11 : $J = J_2$ Solution à $t = \delta t$

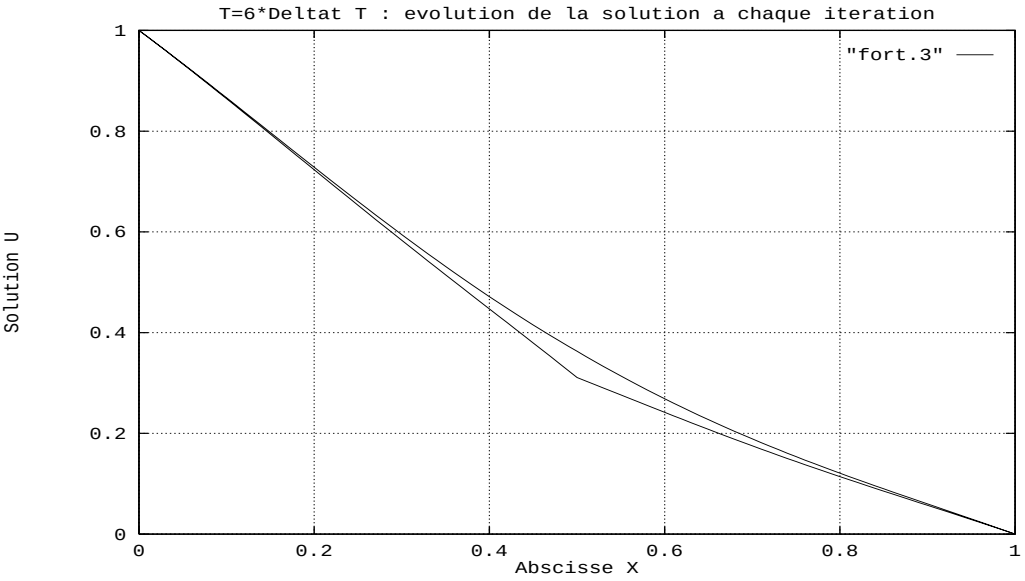


Figure 2.12 : $J = J_2$ Solution à $t = 6\delta t$

Chapitre 3

Problème bidimensionnel

3.1 Etude théorique

3.1.1 Modélisation du problème

On résout sur Ω :

$$U_t + aU_x + bU_y - \nu(U_{xx} + U_{yy}) = 0 \tag{3.1}$$

on pose $U_t = \frac{U^{n+1}-U^n}{\Delta t}$, alors $U^{n+1} = U$ est solution de (3.2),

$$\begin{cases} \alpha U + aU_x + bU_y - \nu(U_{xx} + U_{yy}) = f, & \forall -1 \leq x \leq 1, \forall -1 \leq y \leq 1 \\ U(-1,y) = g(y) & \forall -1 \leq y \leq 1 \\ U_y(x,-1) = U_y(x,1) = 0 & \forall -1 \leq x \leq 1 \\ U_x(1,y) = 0 & \forall -1 \leq y \leq 1 \end{cases} \tag{3.2}$$

où $\alpha = \frac{1}{\Delta t}$ et $f = \alpha U^n(x,y)$ et par exemple :

$$\begin{cases} g_1(y) = & U_{01} & \text{si } y \leq 0 \\ = & U_{02} & \text{si } 0 \leq y \\ g_2(y) = & \exp(-\frac{1}{(y-1)^2}) & \forall -1 \leq y \leq 1 \end{cases} \tag{3.3}$$

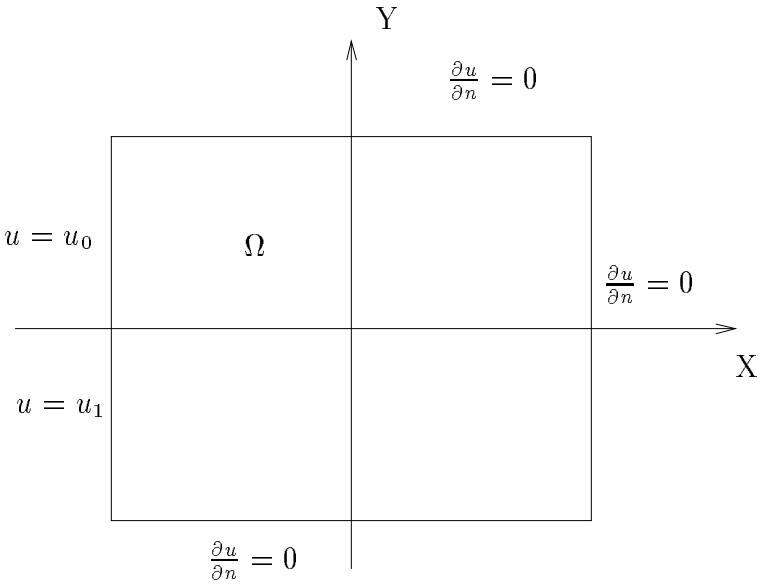


Figure 3.1 : Domaine modèle et conditions aux limites

3.1.2 Ecriture du système en partition

A titre d'exemple, on découpe le domaine Ω en deux sous-domaines symétrique par rapport à l'axe $x = 0$.Sur le sous-domaine 1, *i.e.* $(x, y) \in [-1, 0] \times [-1, 1]$, on résout :

$$\begin{cases} \alpha u^{(1)} + a u_x^{(1)} + b u_y^{(1)} - \nu(u_{xx}^{(1)} + u_{yy}^{(1)}) = f, & \forall (x,y) \in [-1,0] \times [-1,1] \\ u^{(1)}(-1, y) = g(y), & \forall y \in [-1,1] \\ u_y^{(1)}(x, -1) = u_y^{(1)}(x, 1) = 0, & \forall x \in [-1,0] \\ u^{(1)}(0, y) = v(y), & \forall y \in [-1,1] \end{cases} \tag{3.4}$$

et sur le sous-domaine 2,*i.e.* $(x,y) \in [0,1] \times [-1,1]$, on résout :

$$\begin{cases} \alpha u^{(2)} + a u_x^{(2)} + b u_y^{(2)} - \nu(u_{xx}^{(2)} + u_{yy}^{(2)}) = f, & \forall (x,y) \in [0,1] \times [-1,1] \\ u^{(2)}(0,y) = v(y), & \forall y \in [-1,1] \\ u_y^{(2)}(x,-1) = u_y^{(2)}(x,1) = 0, & \forall x \in [0,1] \\ u_x^{(2)}(1,y) = 0, & \forall y \in [-1,1] \end{cases} \tag{3.5}$$

On introduit à nouveau une fonctionnelle J telle que :

$$J = \frac{1}{2} \int_{-1}^{1} (u_x^{(1)} - u_x^{(2)})^2(0,y) w(y) dy \tag{3.6}$$

Alors le gradient de J vaut :

$$\delta J = \int_{-1}^{1} (u_x^{(1)} - u_x^{(2)})(\delta u_x^{(1)} - \delta u_x^{(2)})(0,y) w(y) dy \tag{3.7}$$

La fonction u dont $u^{(1)}$ et $u^{(2)}$ sont les restrictions à Ω_1 et Ω_2 respectivement est continue; c'est une solution lisse du problème de départ si et seulement si la dérivée normale à l'interface est continue. Il s'agit à nouveau de trouver v de manière à réaliser cette continuité.Par contre ici, on itère non plus sur une nombre rééel mais sur une fonction $v(y)$.

3.1.3 Résolution

$\delta U^{(1)}$ et $\delta U^{(2)}$ sont les solutions des deux systèmes linéarisés suivant :

$$\begin{cases} \alpha \delta u^{(1)} + a \delta u_x^{(1)} + b \delta u_y^{(1)} - \nu(\delta u_{xx}^{(1)} + \delta u_{yy}^{(1)}) = 0, & \forall (x,y) \in [-1,0] \times [-1,1] \\ \delta u^{(1)}(-1,y) = 0, & \forall y \in [-1,1] \\ \delta u_y^{(1)}(x,-1) = \delta u_y^{(1)}(x,1) = 0, & \forall y \in [-1,1] \\ \delta u^{(1)}(0,y) = \delta v(y), & \forall y \in [-1,1] \end{cases} \tag{3.8}$$

et

$$\left\{\begin{array}{rl} \alpha\delta u^{(2)} + a\delta u_x^{(2)} + b\delta u_y^{(2)} - \nu(\delta u_{xx}^{(2)} + \delta u_{yy}^{(2)}) = 0, & \forall (x,y) \in [0,1]\times[-1,1] \\ \delta u^{(2)}(0,y) = \delta v(y), & \forall y \in [-1,1] \\ \delta u_y^{(2)}(x,-1) = \delta u_y^{(2)}(x,1) = 0, \forall y \in [-1,1] & \\ \delta u_x^{(2)}(1,y) = 0, & \forall y \in [-1,1] \end{array}\right. \tag{3.9}$$

- Résolution de (3.8)

  1. On introduit la fonction suivante :
     $\lambda^{(1)}$ variable adjointe de $\delta u^{(1)}$ fonction de $((x,y) \in [-1,0]\times[-1,1])$
  2. On multiplie alors (3.8) par λ et on intègre sur le sous-domaine 1.

$$\begin{array}{l} \int_{-1}^{0}\int_{-1}^{1} \lambda^{(1)}\{\alpha\delta u^{(1)} + a\delta u_x^{(1)} + b\delta u_y^{(1)} \\ -\nu(\delta u_{xx}^{(1)} + \delta u_{yy}^{(1)})\}\ dx\ dy \ = \ 0 \end{array} \tag{3.10}$$

  3. On intègre alors par partie et on simplifie en tenant compte des conditions aux limites homogènes :

$$\begin{array}{l} \int_{-1}^{0}\int_{-1}^{1} \left\{\alpha\lambda^{(1)} - a\lambda_x^{(1)} - b\lambda_y^{(1)} - \nu(\lambda_{xx}^{(1)} + \lambda_{yy}^{(1)})\right\}\ \delta u^{(1)}\ dx\ dy \\ + \int_{-1}^{1} \nu\lambda^{(1)}(-1,y)\ \delta u_x^{(1)}(-1,y)\ dy \\ + \int_{-1}^{0} \left\{[b\lambda^{(1)} + \nu\lambda_y^{(1)}](x,1)\ \delta u^{(1)}(x,1)\right. \\ \left. -[b\lambda^{(1)} + \nu\lambda_y^{(1)}](x,-1)\ \delta u^{(1)}(x,-1)\right\}\ dx \\ + \int_{-1}^{1} \left\{-\nu\lambda^{(1)}(0,y)\ \delta u_x^{(1)}(0,y) + [a\lambda^{(1)} + \nu\lambda_x^{(1)}](0,y)\ \delta v(y)\right\} dy = 0 \end{array} \tag{3.11}$$

  4. Finalement, on pose $\lambda^{(1)}$ **solution du système adjoint :**

$$\left\{\begin{array}{rl} \alpha\lambda^{(1)} - a\lambda_x^{(1)} - b\lambda_y^{(1)} - \nu(\lambda_{xx}^{(1)} + \lambda_{yy}^{(1)}) = 0, & \forall (x,y) \in [-1,0]\times[-1,1] \\ \lambda^{(1)}(-1,y) = 0, & \forall y \in [-1,1] \\ b\lambda^{(1)} + \nu\lambda_y^{(1)}(x,-1) = 0, & \forall x \in [-1,0] \\ b\lambda^{(1)} + \nu\lambda_y^{(1)}(x,1) = 0, & \forall x \in [-1,0] \\ \lambda^{(1)}(0,y) = \left(u_x^{(1)} - u_x^{(2)}\right)(0,y), & \forall y \in [-1,1] \end{array}\right. \tag{3.12}$$

et on obtient le résultat suivant :

$$\nu \int_{-1}^{1} \left(u_x^{(1)} - u_x^{(2)}\right)(0,y)\delta U_x^{(1)}(0,y)dy = \int_{-1}^{1} \left(a\lambda^{(1)} + \nu\lambda_x^{(1)}\right)(0,y)dy \tag{3.13}$$

- Résolution de (3.9)

1. On introduit la fonction suivante :
   $\lambda^{(2)}$ variable adjointe de $\delta u^{(2)}$ de $((x,y) \in [0,1][-1,1])$
2. On multiplie alors (3.9) par $\lambda^{(2)}$ et on intègre sur le sous-domaine 2.

$$\begin{array}{l} \int_0^1 \int_{-1}^1 \lambda^{(2)}\{\alpha\delta u^{(2)} + a\delta u_x^{(2)} + b\delta u_y^{(2)} \\ -\nu(\delta u_{xx}^{(2)} + \delta u_{yy}^{(2)})\}\ dx\ dy \ = \ 0 \end{array} \tag{3.14}$$

3. On intègre alors par partie et on simplifie en tenant compte des conditions aux limites homogènes :

$$\begin{array}{l} \int_0^1 \int_{-1}^1 \left\{\alpha\lambda^{(2)} - a\lambda_x^{(2)} - b\lambda_y^{(2)} - \nu(\lambda_{xx}^{(2)} + \lambda_{yy}^{(2)})\right\}\ \delta u^{(2)}\ dx\ dy \\ + \int_{-1}^1 [a\lambda^{(2)} + \nu\lambda_x^{(2)}](1,y)\delta u^{(2)}(1,y)dy \\ + \int_0^1 \left\{[b\lambda^{(2)} + \nu\lambda_y^{(2)}](x,1)\ \delta u^{(2)}(x,1)\right. \\ \left. -[b\lambda^{(2)} + \nu\lambda_y^{(2)}](x,-1)\ \delta u^{(2)}(x,-1)\right\} dx \\ + \int_{-1}^1 \left\{\nu\lambda^{(2)}(0,y)\ \delta u_x^{(2)}(0,y) - [a\lambda^{(2)} + \nu\lambda_x^{(2)}](0,y)\ \delta v(y)\right\} dy = 0 \end{array} \tag{3.15}$$

4. Finalement, on pose $\lambda^{(2)}$ **solution du système adjoint :**

$$\left\{ \begin{array}{l} \forall (x,y) \in [0,1] \times [-1,1] \\ \alpha\lambda^{(2)} - a\lambda_x^{(2)} - b\lambda_y^{(2)} - \nu(\lambda_{xx}^{(2)} + \lambda_{yy}^{(2)}) = 0, \\ (a\lambda^{(2)} + \nu\lambda_x^{(2)})(1,y) = 0, \forall y \in [-1,1] \\ b\lambda^{(2)} + \nu\lambda_y^{(2)}(x,-1) = 0 b\lambda^{(2)} + \nu\lambda_y^{(2)}(x,1) = 0 \\ \lambda^{(2)}(0,y) = \left(u_x^{(1)} - u_x^{(2)}\right)(0,y) \\ \forall y \in [-1,1] \end{array} \right. \tag{3.16}$$

et on obtient le résultat suivant :

$$\begin{array}{l} \nu \int_{-1}^{1} \left(u_x^{(1)} - u_x^{(2)}\right)(0,y)\delta U_x^{(2)}(0,y)dy = \\ \int_{-1}^{1} \left(a\lambda^{(2)} + \nu\lambda_x^{(2)}\right)(0,y)dy \end{array} \tag{3.17}$$

- A partir de (3.13) et (3.17), on calcule la valeur de δJ :

$$\begin{array}{rl} \delta J(v) = & \int_{-1}^{1} (u_x^{(1)} - u_x^{(2)})(\delta u_x^{(1)} - \delta u_x^{(2)})(0,y)w(y)dy \\ = & \frac{1}{\nu}\left(\int_{-1}^{1} \left(a\lambda^{(1)} + \nu\lambda_x^{(1)}\right)(0,y)dy \right. \\ & \left. -\int_{-1}^{1} \left(a\lambda^{(2)} + \nu\lambda_x^{(2)}\right)(0,y)dy\right) \\ = & \int_{-1}^{1} \left(\lambda_x^{(1)} - \lambda_x^{(2)}\right)(0,y)\delta v(y)\ dy \end{array} \tag{3.18}$$

$$\begin{array}{rl} \delta J(v) = & \int_{-1}^{1} K(y)\ \delta v(y)\ dy \\ K(y) = & \left(\lambda_x^{(1)} - \lambda_x^{(2)}\right)(0,y) \end{array} \tag{3.19}$$

K est ici le saut de dérivée normale, *i.e.* la différence entre les valeurs sur Ω_1 et Ω_2.

- ainsi que l'expression de l'algorithme de descente :

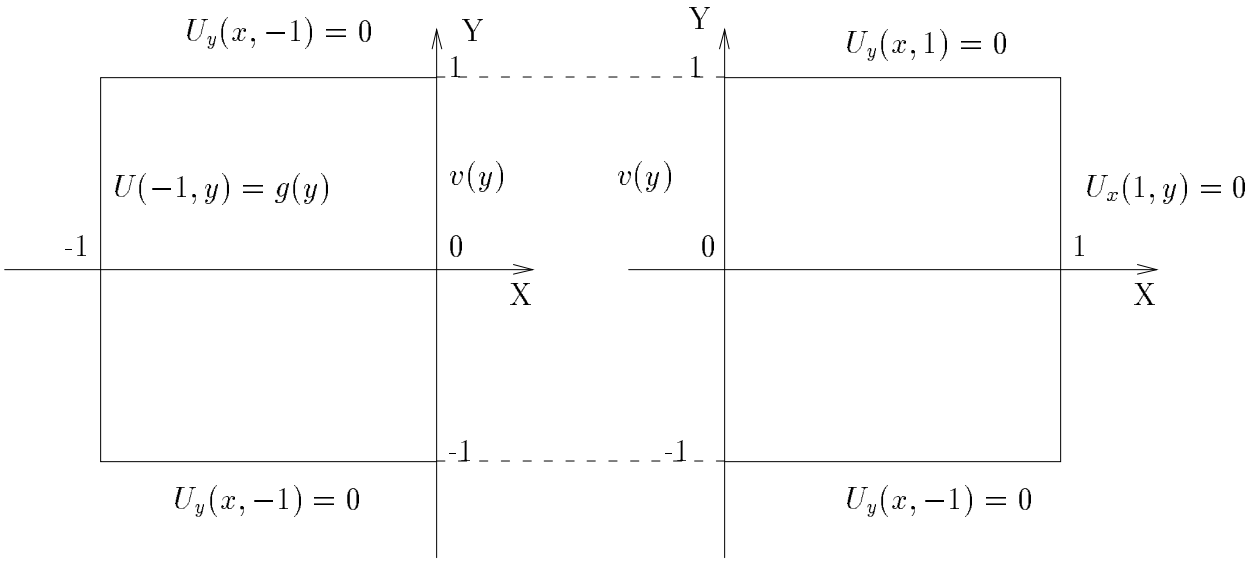
$$\delta v(y) = \frac{-JK(y)}{\int_{-1}^{1} K^2(y)\ dy}$$

on obtient alors

$$\delta J = -J \tag{3.20}$$

Problème direct : deux sous-domaines

Equation générique : $\alpha U + aU_x + bU_y - \nu(U_{xx} + U_{yy}) = f$



Problème adjoint : deux sous-domaines.

Equation générique : $\alpha\lambda - a\lambda_x - b\lambda_y - \nu(\lambda_{xx} + \lambda_{yy}) = 0$

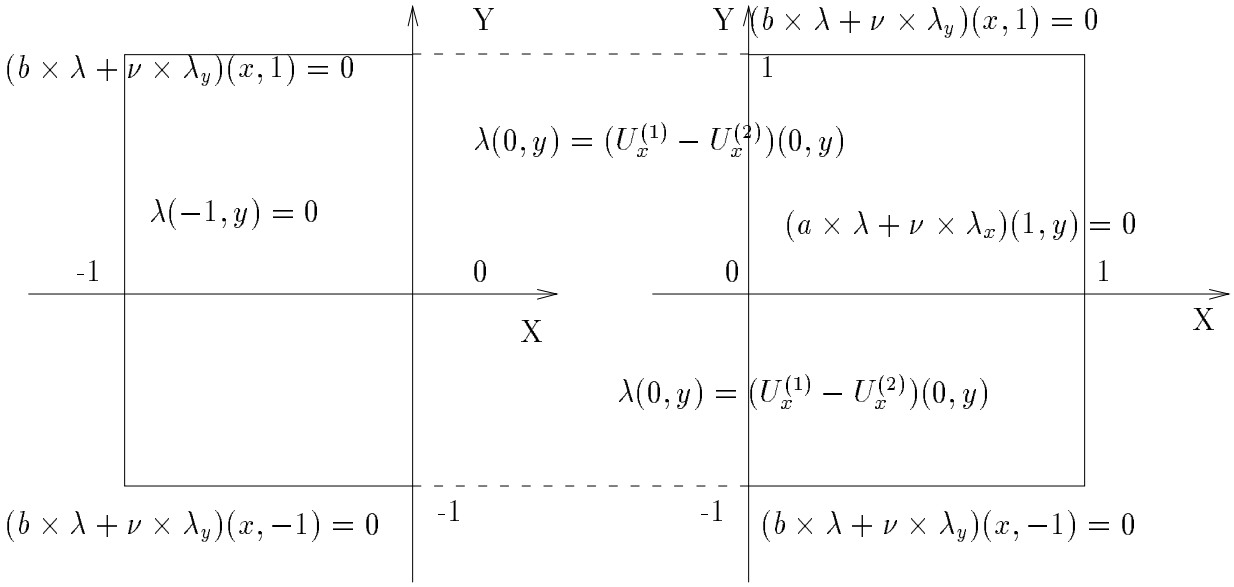


Figure 3.2 : Conditions aux limites en 2D

3.1.4 Remarques

1. Soit Ω une fonction à valeur positive, la définition :

$$\delta v(y) = \frac{-J\Omega(y)K(y)}{\int_{-1}^{1} K^2(y)\Omega(y)dy} \tag{3.21}$$

   conduit également à $\delta J = -J$.

2. La méthode de descente proposée ici n'est pas une méthode de Newton car l'itération est une application d'un espace fonctionnel dans $I\!R$ (et non un point fixe d'un espace dans lui-même). Donc la convergence de l'algorithme n'est pas quadratique en général, indépendament de la difficulté liée au gradient nul à convergence.

3.2 Implémentation et résultats numériques

On résout le problème direct et le problème adjoint par une méthode de volumes finis.Les systèmes linéaires sont résolus par la méthode de Gauss-Seidel.

Cas-test 1 : On impose une condition d'entrée discontinue

1. La condition au bord est discontinue. Cf (3.3)

2. Le nombre d'itérations de l'algorithme est encore beaucoup trop important de l'ordre de 100 pour $\epsilon = 10^{-4}$. De plus il n'est pas décroissant au cours du temps. Cf (3.6)

3. Aux premières itérations, l'algorithme de descente oscille, puis il a un comportement favorable.La convergence est rapide. Mais dès que le gradient tend vers zéro, la convergence se dégrade. Cf (3.7)

4. A $t = 2\delta t$ la convergence est rapide.Cf (3.8)

5. A $t = 3\delta t$ la convergence est rapide, mais elle oscille.Cf (3.9)

Cas-test 2 : On impose une condition d'entrée lisse

1. La condition au bord est de type gaussienne. Cf (3.10)
2. Le nombre d'itérations de l'algorithme se dégrade. De plus il n'est pas décroissant au cours au cours du temps. Cf (3.13)
3. La décroissance du critére J est très lente mais régulière.

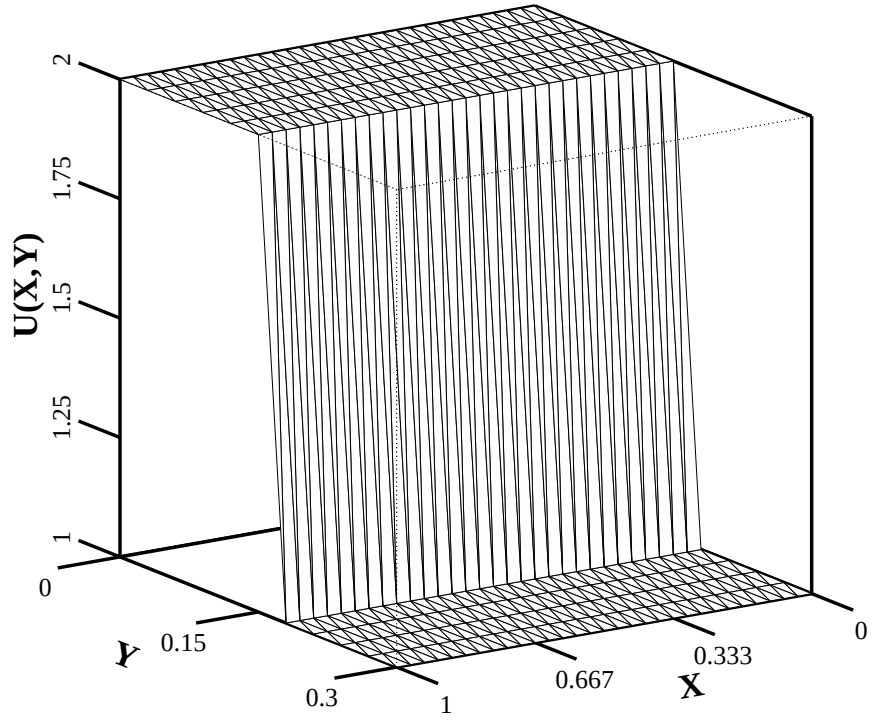


Figure 3.3 : Cas-test Choc : Condition initiale

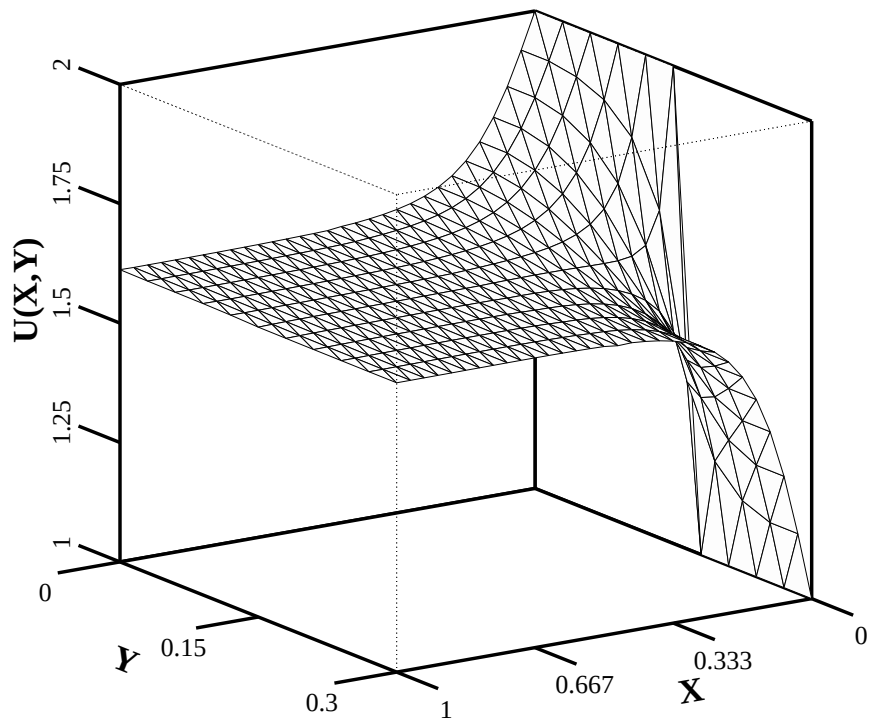


Figure 3.4 : Cas-test Choc : Solution finale à $t = 4\delta t$

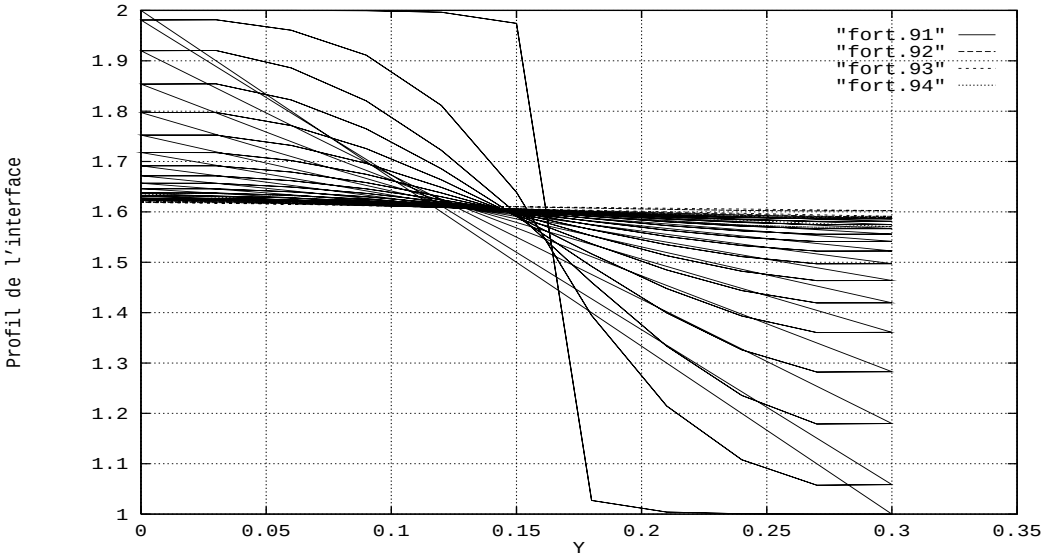


Figure 3.5 : Cas-test Choc : Evolution des valeurs à l'interface

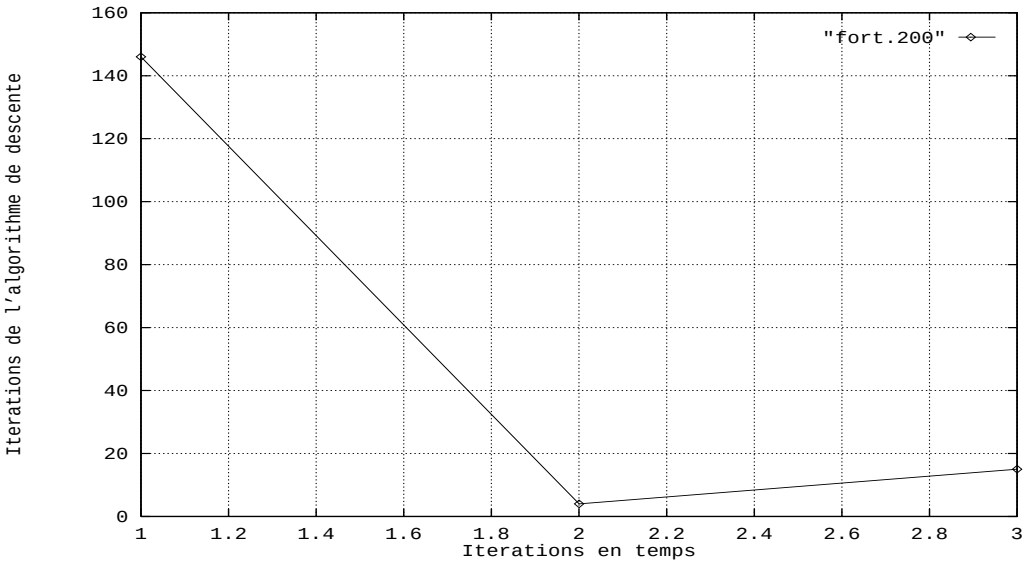


Figure 3.6 : Cas-test Choc : Nombre d'itérations pour l'algorithme de descente

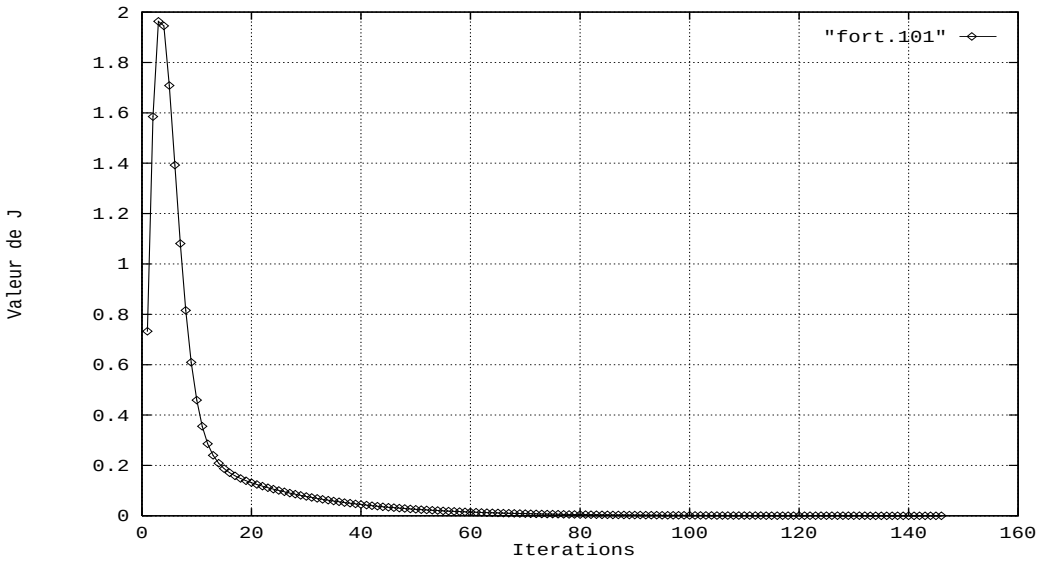


Figure 3.7 : Cas-test Choc : Valeur de J, critère de convergence à $t = \delta t$

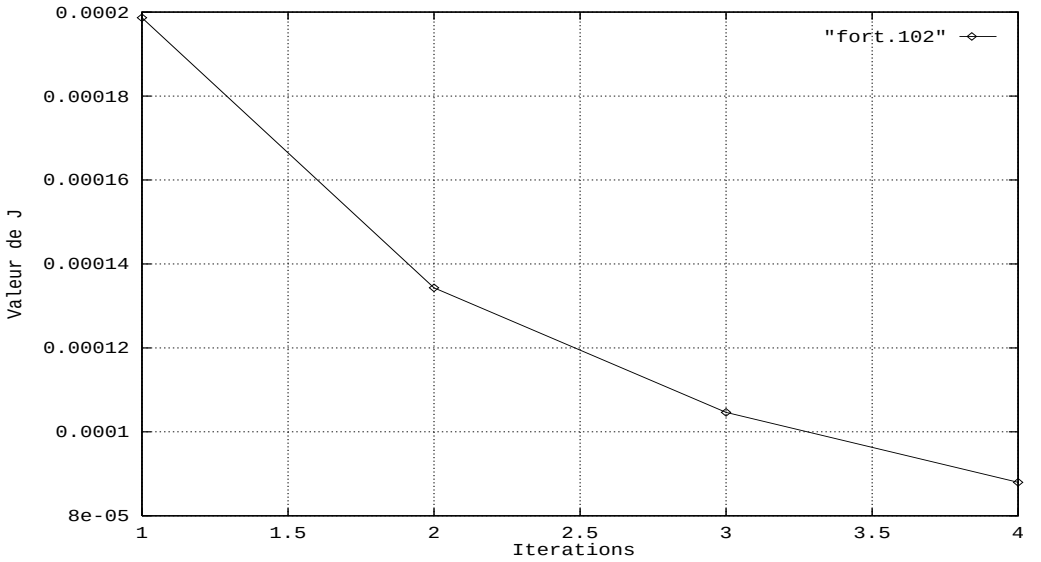


Figure 3.8 : Cas-test Choc : Valeur de J, critère de convergence à $t = 2\delta t$

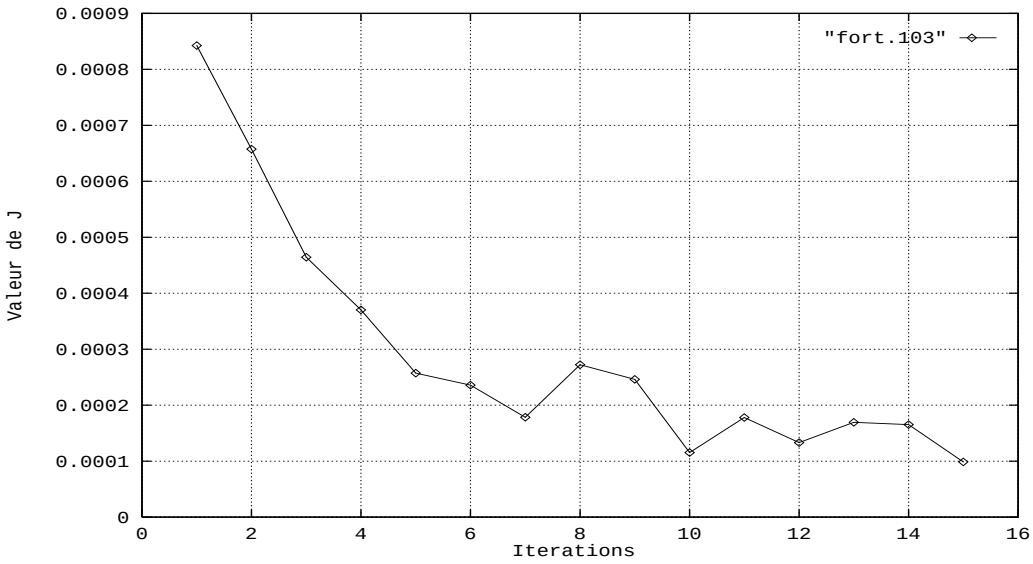


Figure 3.9 : Cas-test Choc : Valeur de J, critère de convergence à $t = 3\delta t$

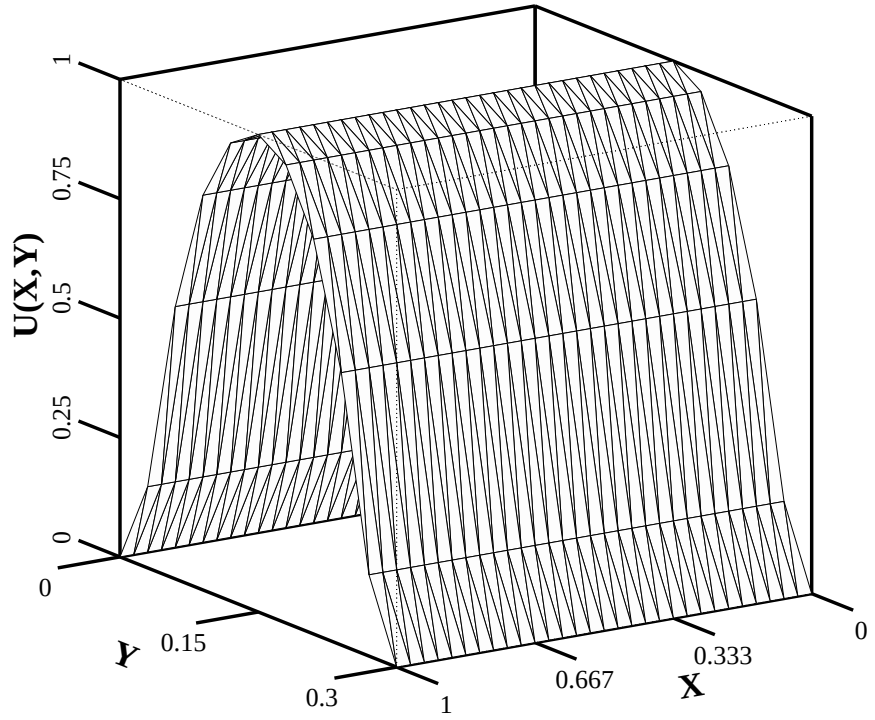


Figure 3.10 : Cas-test Gauss : Condition initiale

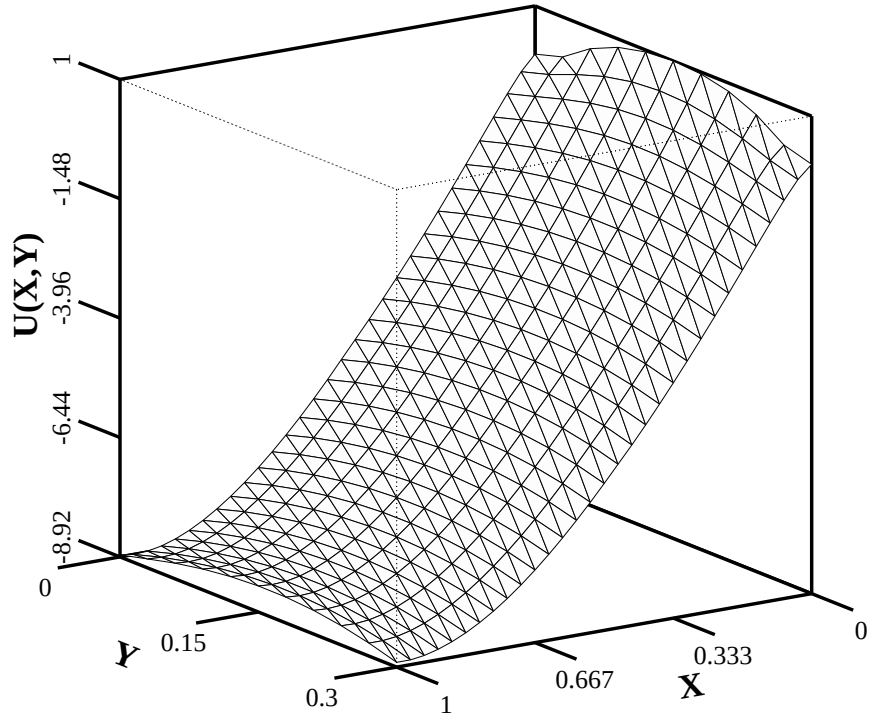


Figure 3.11 : Cas-test Gauss : Solution finale

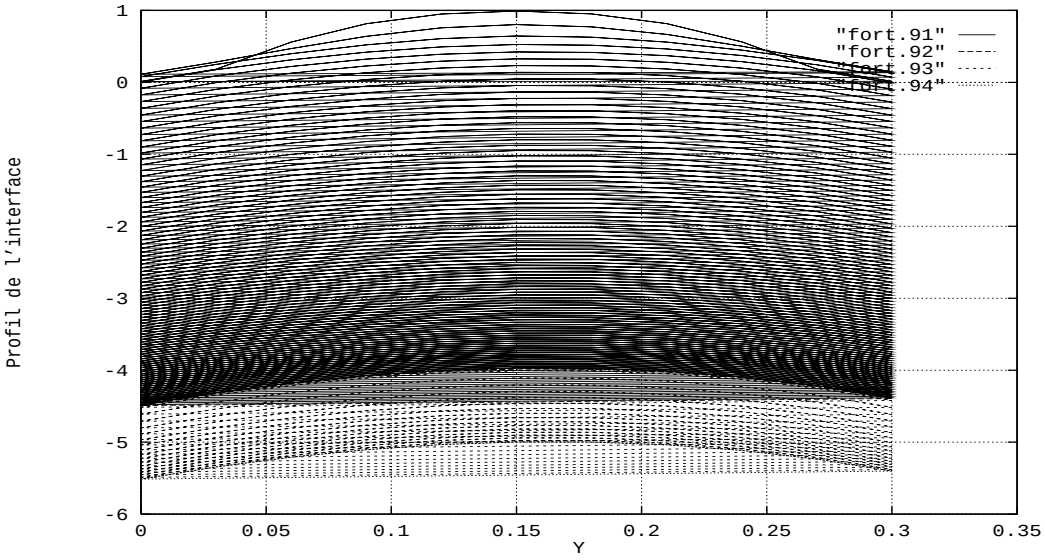


Figure 3.12 : Cas-test Gauss : Evolution des valeurs à l'interface

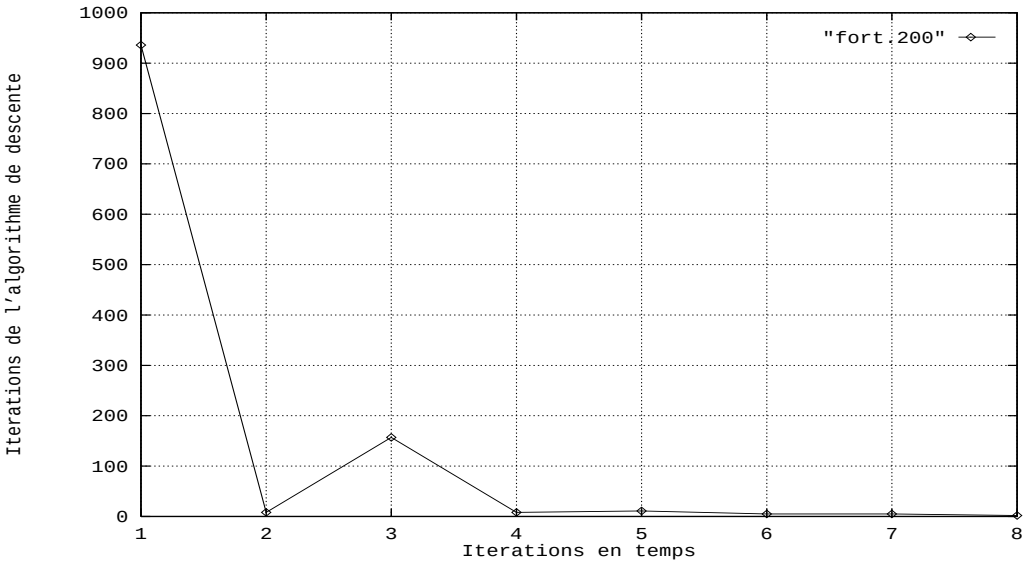


Figure 3.13 : Cas-test Gauss : Nombre d'itérations pour l'algorithme de descente

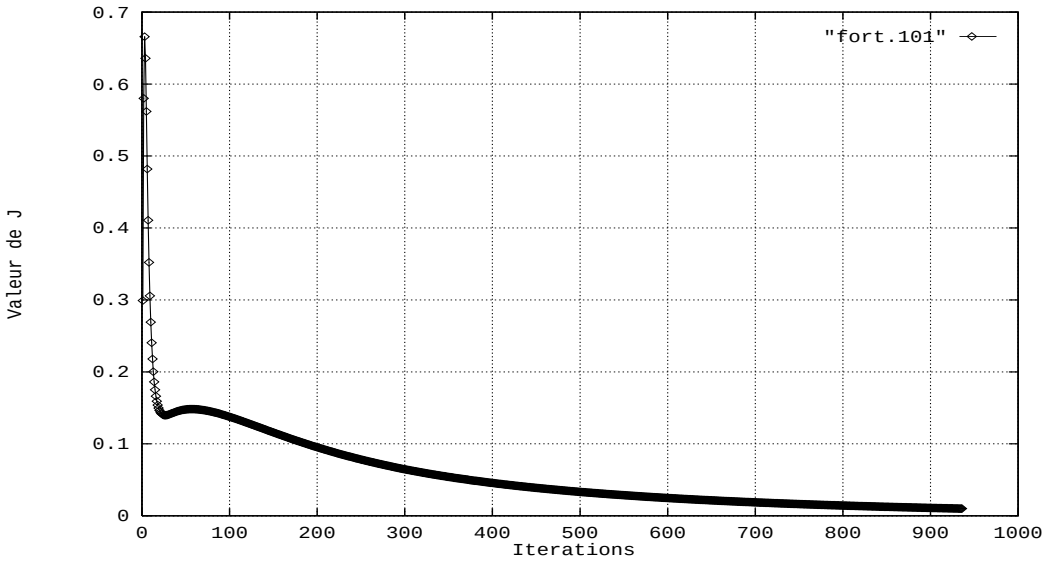


Figure 3.14 : Cas-test Gauss : Valeur de J, critère de convergence à $t = \delta t$

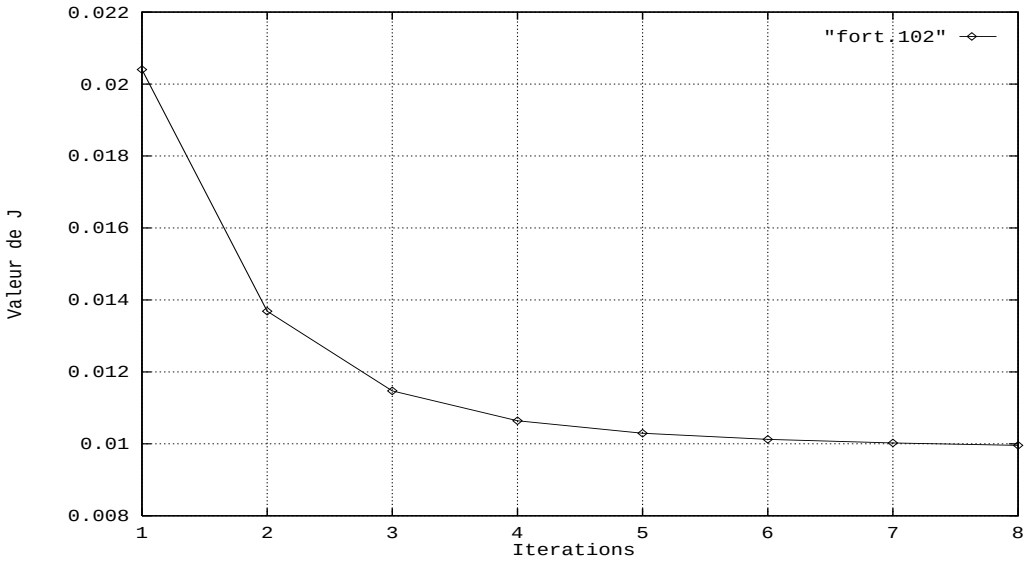


Figure 3.15 : Cas-test Gauss : Valeur de J, critère de convergence à $t = 2\delta t$

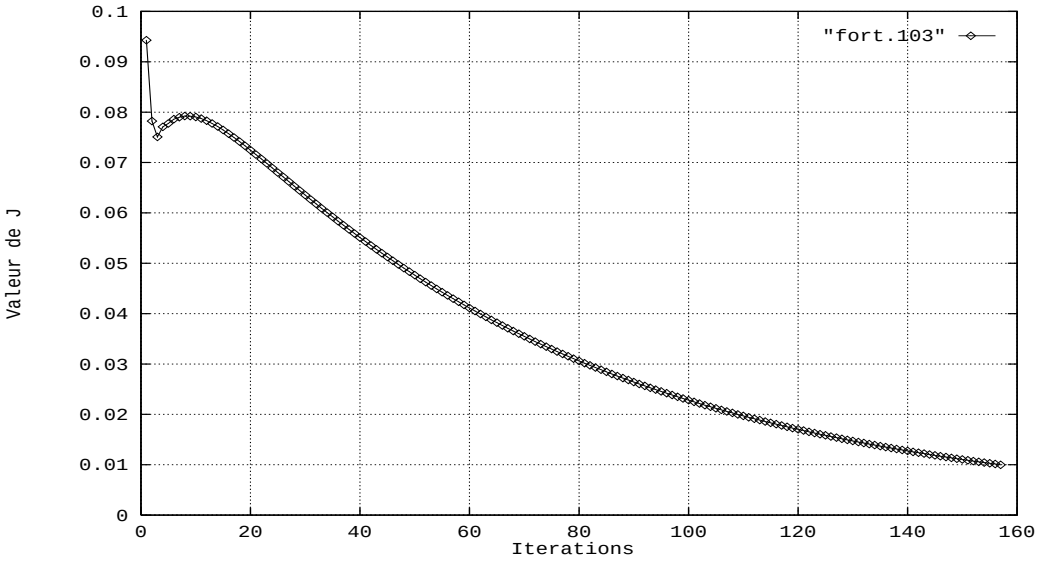


Figure 3.16 : Cas-test Gauss : Valeur de J, critère de convergence à $t = 3\delta t$

Chapitre 4

Conclusion

Bilan La résolution par une méthode de partition de domaine de l'équation d'advection-diffusion est donc réalisable.Mais parce que l'on n'a pas cherché à optimiser la vitesse de convergence de la méthode, celle-ci est beaucoup trop faible. Néanmoins elle pourrait certainement être améliorée en remplaçant l' algorithme de descente par un algorithme de gradient conjugué.

De part sa conception, cet algorithme est aisément parallélisable.Le temps de calcul des états directs et adjoints peut-être divisé par le nombre de processeurs d'une machine "mimd". On peut donc espérer un fort gain de temps sur une machine parallèle.

De plus, la résolution de sous-problèmes améliore le conditionement des systèmes linéaires.Si l'algorithme de descente converge raisonablement vite, on gagnera alors du temps même sur une machine scalaire.

Perspectives

1. Le problème d'optimisation n'a pas été étudié. Il est donc certainement possible d'améliorer la convergence, avec des préconditionneurs où d'autres algorithmes.

2. Néanmoins le problème principal est la vitesse de convergence.En effet, le fait que $\nabla J = 0$ au minimum dégrade irrémédiablement la convergence avec la fonctionelle J quadratique choisie. On propose donc de calculer, analogue du cas monodimensionnel, un critère J local : Pour chaque noeud frontière :

$$J_i = (U_x^{(1)} - U_x^{(2)})_i$$

alors

$$\delta J_i = L_i^t \delta v(y)$$

où L_i^t est une forme linéaire solution d'un problème adjoint obtenu avec une condition à l'interface égale à un dirac placé au point i. On a alors :

$$\delta J = L^t \delta v$$

avec

$$L = (L_1 L_2 \dots L_n)$$

et

$$\delta J = \begin{pmatrix} \delta J_1 \\ \delta J_2 \\ \vdots \\ \delta J_n \end{pmatrix}$$

Ce qui amène à résoudre le système :

$$L^t \delta v(y) = -J$$

où on a posé :

$$J = \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_n \end{pmatrix}$$

C'est un système linéaire plein non symétrique, que l'on peut résoudre par une méthode de type GMRES.

3. Cette méthode est particulièrement bien adaptée au calcul parallèle où on peut programmer la résolution simultanée des systèmes directs et adjoints.

Références

[CDP92] MC. Ciccoli, JA. Désidéri, and J. Périaux. Introduction of domain decomposition techniques in hyperbolic parabolic flow problems. *Sixth International Conference on Domain Decomposition in Science and Engineering. Como, Italia*, 1992.

[GPD83] R. Glowinski, J. Périaux, and Q.V. Dinh. Domain decomposition methode for nonlinear problems in fluid dynamics. *Computer methods in applied mechanics and engineering*, 40:207–109, 1983.

[Qua91] Alfio Quarteroni. Domain decomposition and parallel processing for the numerical solution of partial differential equation. *Surveys on Mathematics for Industry*, 1991.