



Stage de fin d'études à l'ONERA

Méthode de parallélisation pour des problèmes en espace-temps : Couplage Pararéel-Schwarz relaxation d'onde

Nesrine YAHIAOUI

MACS 3

Sous la direction de Juliet RYAN

Jeudi 19 Septembre 2013

SOMMAIRE

1 Remerciements	4
2 Introduction	5
3 Environnement du stage	7
3.1 Présentation de l'entreprise	7
3.2 Présentation du département	7
4 Parallélisation en temps : Pararéel	8
4.1 Présentation	8
4.2 Application à la résolution d'EDP linéaires en parallèle	11
5 Parallélisation en espace : Schwarz relaxation d'onde	14
5.1 Introduction	14
5.2 Présentation	15
5.3 Méthodes de Schwarz relaxation d'onde optimisées	19
6 Parallélisation en espace et en temps : Couplage Pararéel/SWR	21
7 Résultats numériques	24
7.1 Équation de la chaleur	24
7.2 Équation d'advection diffusion	27
7.3 Cas multi-domaines	30
8 Conclusion	35
9 Annexes	36
9.1 Bibliographie	36
9.2 Codes	37

1 Remerciements

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de Sup Galilée et les intervenants professionnels responsables de la formation la Macs, pour avoir assuré la partie théorique de celle-ci.

Je remercie également Madame Ryan pour l'aide et les conseils concernant les missions évoquées dans ce rapport, qu'elle m'a apportée lors des différents suivis.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m'ont fait vivre durant ces six mois de stage au sein de l'entreprise Onera : Encore une fois Madame Ryan pour son accueil et la confiance qu'elle m'a accordée dès mon arrivée dans l'entreprise.

Mes collègues Oana (ancienne étudiante de la Macs) et Thuy, pour m'avoir intégrée rapidement au sein de l'entreprise et m'avoir accordée toute leur confiance ; pour le temps qu'elle m'ont consacré tout au long de cette période, sachant répondre à toutes mes interrogations. Monsieur Juvigny ainsi que l'ensemble du personnel de DTIM pour leur accueil sympathique et leur coopération professionnelle tout au long de ces six mois.

2 Introduction

Des phénomènes physiques de plus en plus complexes sont modélisés par des systèmes d'équations aux dérivées partielles. Ces équations sont elles-mêmes discrétisées pour être résolues par ordinateur. Si les progrès technologiques ont constamment amélioré les performances des processeurs, ces performances atteignent aujourd'hui un plateau en terme de mémoire et de puissance de calcul. Les ordinateurs monoprocesseurs ont laissé la place aux ordinateurs parallèles à mémoire distribuée. Pour utiliser au mieux ces calculateurs parallèles formés d'un grand nombre de processeurs, il fallait inventer des nouveaux algorithmes parallèles. Les méthodes de décomposition de domaines sont naturellement adaptées à cette problématique. Elles permettent de structurer et gérer le domaine géométrique en sous-domaines. Mais les méthodes de décomposition des domaines sont encore bien plus qu'un outil de calcul parallèle. Elles permettent de modéliser le couplage entre différentes physiques.

Mais avec l'avènement des calculateurs parallèles à grande échelle composés de centaines de milliers de processeurs, le besoin de parallélisation est devenu tel que souvent la parallélisation en espace ne suffit plus pour utiliser tous les noeuds du calculateur. Il apparaît donc intéressant de pouvoir utiliser le parallélisme aussi dans la direction du temps. On imagine donc bien qu'il faut un mécanisme global pour paralléliser en temps de façon convaincante. Un de ces mécanismes est connu sous le nom de la méthode du tir, destinée à résoudre une équation différentielle ordinaire du deuxième ordre en dimension 1 avec des données aux deux extrémités. Une variante de cette méthode, dont l'application aux équations aux dérivées partielles a reçu récemment beaucoup d'attention, est l'algorithme pararéel.

D'autre part, avec l'utilisation des ordinateurs massivement parallèles, il peut arriver que le nombre de processeurs soit trop grand pour les nécessités de la parallélisation en espace. Un exemple extrême est la résolution d'une équation différentielle ordinaire scalaire. Le calcul d'une composante ne peut évidemment pas être parallélisé. Par contre, pour des calculs en temps très long, il est nécessaire de paralléliser les calculs dans la direction du temps. C'est l'objet des méthodes de Schwarz relaxation d'onde.

Les premiers travaux sur ces méthodes de parallélisations ont commencé en 1990 avec Womble. Puis Martin Gander et Vandevaille ont mis l'algorithme pararéel sous une écriture multigrille [1]. D'autres articles ont été également publiés par Jun Liu et Yao-Lin Jiang sur l'algorithme pararéel/relaxation d'onde [2]. Ce sujet d'étude est en cours de recherche, où on voit aujourd'hui des articles récents sur le couplage des méthodes de parallélisation espace/temps (Gander, 2013) dans [3].

Tous les problèmes d'ingénierie aujourd'hui sont résolus en parallèle sur des ordinateurs composés de centaines d'ordinateurs. Ce document se propose d'exposer les méthodes de parallélisation susceptibles de s'appliquer à ces nouveaux outils. Émile Picard nous enseigne dans [5] que pour comprendre une théorie, il est bon d'avoir en tête un problème modèle. Nous choisirons donc dans tout ce document un fil conducteur, l'équation de la chaleur

$$\partial_t u - \Delta u = f$$

représentant les variations en temps et en espace de la température d'un corps emplissant le domaine Ω , soumis à une source de chaleur f (qui sera appelée second membre), avec une température initiale donnée dans tout le domaine, et des conditions aux limites sur le bord du domaine $\partial\Omega$, par exemple de Dirichlet (la température est fixée), soit $u = g$. $\partial_t u$ est la dérivée en temps de u , Δ est

l'opérateur de Laplace. Pour calculer sur un ordinateur une solution approchée de cette équation, on peut commencer par une semi-discrétisation en temps. La discrétisation en espace de cette équation par une méthode de type éléments finis ou volumes finis, mène à un système linéaire.

Au cours de ce projet nous présenterons les méthodes de pararéel et Schwarz relaxation d'onde ainsi que le couplage de ces deux méthodes sur le problème modèle : l'équation de la chaleur. On étudiera également une autre variante du problème qui est l'équation d'advection diffusion.

3 Environnement du stage

3.1 Présentation de l'entreprise

L'Onera est aujourd'hui le premier acteur français de la R&T aéronautique, spatiale et de la défense. L'Office national d'études et de recherches aérospatiales compte 2 000 salariés, dont 1 500 chercheurs, ingénieurs et techniciens, répartis sur huit sites en France. Les ressources de l'Onera proviennent de deux sources :

- pour 60 %, de prestations de recherches contractuelles pour les agences de programme et l'industrie
- pour 40 %, d'une subvention annuelle de l'Etat français.

La subvention finance essentiellement les recherches à long terme, qui préparent l'avenir. Les contrats de recherche financent les travaux à moyen et court termes, plus proches de l'application.

Tout l'enjeu de la politique scientifique de l'Onera est d'organiser ce flux de connaissances, qui va de l'acquisition de savoirs nouveaux à leur transfert vers l'industrie.

Avec cinq fois plus d'activité sous contrat par chercheur que la moyenne, l'Onera est quotidiennement dans une logique d'évaluation des retombées de chaque projet, en termes de compétitivité pour l'industrie, de progrès pour la défense, d'avancées pour l'environnement et d'accès à l'emploi pour ses doctorants.

3.2 Présentation du département

L'Onera est organisé en 17 départements dont un laboratoire mixte Onera/CNRS regroupés en quatre branches scientifiques : les matériaux et structures, la mécanique des fluides et énergétique, la physique, les traitements de l'information et systèmes.

Le département de Traitement de l'information et modélisation (DTIM) fait partie de la branche traitements de l'information et systèmes.

La simulation numérique de phénomènes physiques est un enjeu majeur dans toutes les phases de conception et d'évaluation de systèmes aéronautiques ou spatiaux ; elle est donc utilisée au sein de plusieurs départements de l'Onera. En offrant une expertise dans la modélisation, le développement de méthodes et leur mise en œuvre, expertise transverse aux différentes branches physiques, le département DTIM contribue à ce que l'Onera soit à la hauteur de cet enjeu.

Deux axes thématiques sont observés :

- la modélisation mathématiques et simulation numérique
- le calcul haute performance (CHP)

Les objectifs de CHP sont :

- Augmenter le champ d'application et la qualité de la simulation numérique par le développement de modèles multi-domaines, multi-échelles et par le couplage.
- Mettre en place des méthodologies parallèles efficaces pour résoudre ces modèles sur des calculateurs massivement parallèles à mémoire distribuée

4 Parallélisation en temps : Pararéel

4.1 Présentation

Au cours des dernières décennies, les chercheurs ont essayé de paralléliser des algorithmes dans la direction de temps. Un exemple est l'algorithme de Womble [22], où les systèmes résultant d'une discrétisation implicite en temps sont résolus en utilisant une méthode itérative et l'itération du temps suivant est commencée, avant que l'itération du temps actuel ait convergée. Il est alors possible de réitérer plusieurs niveaux de temps simultanément, mais le gain possible utilisant un ordinateur parallèle est petit, voir par exemple [3].

Une approche différente pour obtenir un parallélisme de petite échelle en temps est d'utiliser des méthodes de prédicteur-correcteur, où l'étape de prédiction et l'étape de correction peuvent être exécutées par deux (ou plusieurs) processeurs en parallèle, si c'est organisé correctement. Une classe entière de telles méthodes a été proposée dans [19], et un bon parallélisme de petite échelle peut être réalisé.

Une troisième approche, très différente sont les algorithmes de relaxation d'onde, inventés dans [15], qui sont basés sur une décomposition du système pour être résolu dans des sous-systèmes. Une itération est alors utilisée, qui résout des problèmes dépendant de temps dans chaque sous-système et communique des informations aux interfaces aux sous-systèmes voisins pour converger vers la solution globale en espace-temps [13, 12]. Un progrès substantiel a été fait sur de telles méthodes pour des EDP d'évolution. Si une décomposition multi-grille est utilisée, au lieu d'une décomposition de domaine, on obtient les méthodes dites multi-grille paraboliques, qui sont aussi appelées des méthodes multi-grille de relaxation d'onde.

Finalement, la dernière classe de méthodes, qui se concentre entièrement sur la parallélisation dans la direction de temps, est basée sur des méthodes de tir en temps. Un premier pas historique dans cette direction est [20] et pour une première analyse voir [2]. Le plus récent algorithme dans cette classe est l'algorithme pararéel, inventé dans [16]. Pour une vue d'ensemble historique de telles méthodes, de nouvelles références et une évaluation précise de la convergence de l'algorithme pararéel voir [9, 4].

On propose dans cette partie un schéma élaboré par LIONS, MADAY et TURINICI, permettant de profiter d'une architecture parallèle pour la discrétisation en temps d'une équation d'évolution aux dérivées partielles. Cette méthode, basée sur un schéma d'Euler, combine des résolutions grossières et des résolutions fines et indépendantes en temps en s'inspirant de ce qui est classique en espace. La parallélisation qui en résulte se fait dans la direction temporelle ce qui en revanche non classique. Elle a pour principale motivation les problèmes en temps réel, d'où la terminologie de « pararéel ».

Ils considèrent une équation d'évolution aux dérivées

$$\frac{\partial u}{\partial t} + Au = f \quad \text{dans l'intervalle de temps } [0, T]$$

avec des conditions initiales $u(t = 0) = u_0$ et des conditions aux limites qu'il est inutile de préciser ici. L'inconnu u peut être scalaire ou vectorielle, l'EDP linéaire ou non linéaire...

Ils proposent un schéma en temps compatible avec une résolution en parallèle et avec une précision souhaitée. Pour cela, on choisit un entier N représentant le nombre de pas de temps et on note

$\Delta T = T/N$ le pas de temps et $T^n = n\Delta T$ pour $n = 0, \dots, N$. On introduit ensuite à ces instants des fonctions λ_n pour $n = 0, \dots, N-1$ (avec $\lambda_0 = u_0$) et on résout, dans l'intervalle $[T^n, T^{n+1}]$, l'équation

$$\frac{\partial u_n}{\partial t} + Au_n = f_n, \quad f_n = f|_{[T^n, T^{n+1}]}$$

avec pour condition initiale $u_n(t = T^n) = \lambda_n$ et les mêmes conditions aux limites que pour u .

On peut ainsi résoudre en parallèle la collection de u_n , pour $n = 0, \dots, N-1$. Cette collection coïncide avec $\{u|_{[T^n, T^{n+1}]}\}_n$ quand $u_n(T^{n+1,-}) = \lambda_{n+1}$ pour $n = 0, \dots, N-1$ où $u_n(T^{n+1,-}) = \lim_{\delta \rightarrow 0} u_n(T^{n+1} - \delta)$. Nous montrerons comment trouver ces valeurs de façon itérative.

Pour commencer, on expose l'idée sur un exemple simple d'une équation différentielle linéaire

$$\begin{cases} \frac{dy}{dt}(t) = -ay(t) \text{ sur } [0, T] \\ y(t=0) = y_0 \end{cases}$$

On considère le schéma d'Euler implicite

$$\begin{cases} \frac{Y^{n+1} - Y^n}{\Delta T} + aY^{n+1} = 0 \\ Y^0 = y_0 \end{cases}$$

puis on utilise les valeurs précédemment calculées pour résoudre de façon exacte, sur chaque intervalle de temps $[T^n, T^{n+1}]$

$$\begin{cases} \frac{dy^n}{dt}(t) = -ay^n(t) \text{ sur } [T^n, T^{n+1}] \\ y^n(t = T^n) = Y^n \end{cases}$$

On propose maintenant une procédure itérative pour améliorer la précision de ce schéma.

On pose donc $Y_1^n = Y^n$ et, défini sur $[T^n, T^{n+1}]$, $y_1^n(t) = y^n(t)$. Puis, supposant connus Y_k^n et $y_k^n(t)$ sur $]T^n, T^{n+1}]$,

i. on introduit les sauts $S_k^n = y_k^{n-1}(T^n) - Y_k^n$

ii. puis on propage les sauts

$$\begin{cases} \frac{\delta_k^{n+1} - \delta_k^n}{\Delta T} + a\delta_k^{n+1} = \frac{S_k^n}{\Delta T} \quad (*) \\ \delta_k^0 = 0 \end{cases}$$

iii. ensuite on pose $Y_{k+1}^n = y_k^n(T^n) + \delta_k^n$ et on résout de façon exacte, et en parallèle

$$\begin{cases} \frac{dy_{k+1}^n}{dt}(t) = -ay_{k+1}^n(t) \text{ sur } [T^n, T^{n+1}] \\ y_{k+1}^n(t = T^n) = Y_{k+1}^n \end{cases}$$

On obtient la proposition suivante :

Proposition 1 *Le schéma précédent est d'ordre k au sens où il existe une constante c_k telle que*

$$\forall n, 0 \leq n \leq N-1, |Y_k^n - y(T^n)| + \max_{t \in [T^n, T^{n+1}]} |y_k^n(t) - y(t)| \leq c_k \Delta T^k$$

Démonstration : la proposition est évidente au rang $k = 1$, on la démontre par récurrence. On rappelle tout d'abord que $y_k^{n-1}(T^n) = e^{-a\Delta T} Y_k^{n-1}$ et que donc $S_k^n = e^{-a\Delta T} Y_k^{n-1} - Y_k^n$.

Par ailleurs la solution de (*) est

$$\delta_k^n = \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} S_k^p = \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} Y_k^{p-1} - Y_k^p),$$

et ainsi

$$Y_{k+1}^n = e^{-a\Delta T} Y_k^{n-1} + \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} Y_k^{p-1} - Y_k^p)$$

en rappelant que la solution exacte $y(T^n) = e^{-na\Delta T} y_0$ on en déduit que

$$\begin{aligned} Y_{k+1}^n - e^{-na\Delta T} y_0 &= e^{-a\Delta T} [Y_k^{n-1} - e^{-(n-1)a\Delta T} y_0] \\ &+ \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} [Y_k^{p-1} - e^{-(p-1)a\Delta T} y_0] - [Y_k^p - e^{-pa\Delta T} y_0]), \end{aligned}$$

ou encore, en introduisant $\varepsilon_k^n = Y_k^n - y(T^n)$,

$$\begin{aligned} \varepsilon_{k+1}^n &= e^{-a\Delta T} \varepsilon_k^{n-1} + \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (e^{-a\Delta T} \varepsilon_k^{p-1} - \varepsilon_k^p) \\ &= e^{-a\Delta T} \varepsilon_k^{n-1} + \sum_{p=0}^{n-2} (1 + a\Delta T)^{1+p-n} (e^{-a\Delta T} \varepsilon_k^p) - \sum_{p=1}^{n-1} (1 + a\Delta T)^{p-n} (\varepsilon_k^p) \\ &= \varepsilon_k^{n-1} (e^{-a\Delta T} - (1 + a\Delta T)^{-1}) + \sum_{p=1}^{n-2} (1 + a\Delta T)^{p-n} ((1 + a\Delta T) e^{-a\Delta T} - 1) \varepsilon_k^p \end{aligned}$$

de sorte que si, à l'ordre k il existe c_k indépendante de ΔT telle que, pour tout n , $|\varepsilon_k^n| \leq c_k \Delta T^k$, alors

$$\varepsilon_{k+1}^n \leq c_k \Delta T^k (e^{-a\Delta T} - (1 + a\Delta T)^{-1}) + c_k \left[\sum_{p=1}^{n-2} ((1 + a\Delta T) e^{-a\Delta T} - 1) \right] \Delta T^k \leq \alpha_N c_k \left(\frac{Ta^2}{2} \right) \Delta T^{k+1},$$

avec une constante α_N indépendante de k (et de l'ordre de 2) ce qui montre la proposition avec $c_{k+1} = \alpha_N c_k (Ta^2/2)$.

- Remarque 1** i. Pour ce qui vient d'être donné soit effectivement un schéma, il faut bien évidemment remplacer la résolution exacte entre T^n et T^{n+1} par une résolution précise, avec n'importe quel schéma classique utilisant un pas de temps δt petit par rapport à ΔT . L'intérêt de ce schéma réside dans le fait que les résolutions sur les intervalles $[T^n, T^{n+1}]$ peuvent se faire en parallèle.
- ii. La grille grossière (pas de temps ΔT) permet de propager rapidement l'information sur tout l'intervalle $[0, T]$, la grille fine (pas de temps δt) permet d'obtenir la précision du schéma. Cette combinaison de calculs sur deux grilles généralise un concept bien connu dans les méthodes de décomposition des domaines pour aborder le parallélisme.

4.2 Application à la résolution d'EDP linéaires en parallèle

Pour vérifier le bon comportement de ce schéma sur une EDP elliptique, on considère de nouveau l'équation de chaleur en dimension 1 d'espace sur l'intervalle $\Omega = [0, 1]$, et sur l'intervalle de temps $[0, T]$. Le problème aux limites

$$\begin{aligned}\partial_t u - \partial_{xx} u &= f \text{ dans } \Omega \times [0, T] \\ u(., 0) &= u_0 \text{ dans } \Omega \\ u(0, .) &= g_g \\ u(1, .) &= g_d\end{aligned}\tag{1}$$

a une solution unique $u(x, t)$.

Pour calculer une solution approchée on utilise le schéma d'Euler implicite. Partageons l'intervalle de temps $[0, T]$ en sous-intervalles $[t_n, t_{n+1}]$ de longueur Δt . Notons $u_n(x)$ l'approximation de u à l'instant t_n au point x , calculée par la formule de récurrence

$$\frac{u^{n+1} - u^n}{\Delta t} - \Delta u^{n+1} = f^{n+1},$$

où f_{n+1} représente $f(x, t_{n+1})$. Pour passer du temps t_n au temps t_{n+1} , il faut donc résoudre l'équation elliptique

$$(\eta - \Delta)u = f$$

dans le domaine Ω , où f est maintenant une fonction indépendante du temps.

Heat.m est l'implémentation d'un solveur pour ce problème, où on a choisi $f = 0$ pour simplifier. Pour obtenir une solution de référence, nous partons d'une donnée initiale gaussienne. Au temps initial nous éteignons le chauffage au centre du bureau pour observer le refroidissement de l'objet, puis nous chauffons l'extrémité gauche à partir de l'instant $t = 0.1s$. Le temps de l'observation est de $0.2s$. BarTime.m effectue cette simulation et l'évolution de la solution est représentée ci-dessous.

Partageons l'intervalle $[0, T]$ en $I + 1$ sous intervalles $[T_0 = 0, T_1], \dots, [T_I, T_{I+1} = T]$, et donnons nous $I + 1$ valeurs U_0, \dots, U_I .

Comme nous l'avons précisé précédemment l'algorithme pararéel, proposé par Lions, Maday et Turinici en 2001 [3] est très similaire à une méthode de tir. L'algorithme pararéel donné précédemment peut se réécrire de la façon suivante ℓ est défini à l'aide des deux opérateurs :

1. $G(T_{i+1}, T_i, U)$ qui donne une approximation grossière (et bon marché) de la solution $u(T_{i+1})$ avec condition initiale $u(T_i) = U$,

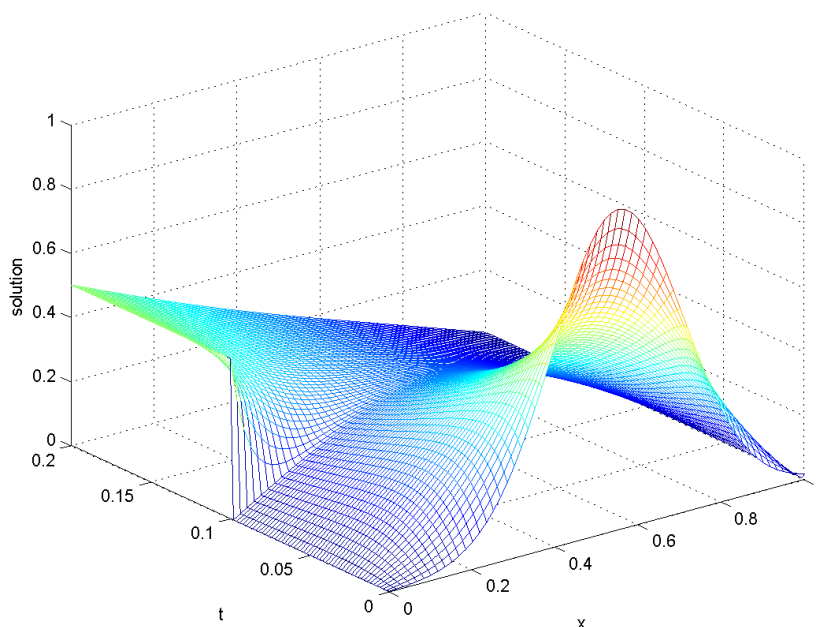


FIGURE 1 – Exemple d’une solution instationnaire de l’équation de la chaleur

2. $F(T_{i+1}, T_i, U)$ qui donne une approximation précise (et onéreuse) de la solution $u(T_{i+1})$ avec condition initiale $u(T_i) = U$.

L’algorithme démarre avec une approximation initiale $U^0 = \{U_i^0\}$ aux instants T_0, T_1, \dots, T_I , calculée par exemple avec G , et fait ensuite pour $n = 0, 1, \dots$ l’itération de correction $U_{i+1}^{n+1} = G(T_{i+1}, T_i, U_i^{n+1}) + F(T_{i+1}, T_i, U_i^n) - G(T_{i+1}, T_i, U_i^n)$. Dans cette itération, la partie onéreuse du calcul, représentée par F , peut être effectuée en parallèle, et seule la partie bon marché G est séquentielle. La méthode est très simple à utiliser : il suffit d’un seul solveur, qu’on appelle soit avec une haute résolution, soit avec une faible résolution, sur des sous-intervalles en temps, et on combine le résultat comme indiqué dans la formule ci-dessus. Cette simplicité d’utilisation explique en partie l’intérêt suscité par cette méthode.

On peut démontrer que cette méthode converge superlinéairement vers la solution, et sous certaines hypothèses, la solution peut être calculée plus rapidement en parallèle qu’avec un seul processeur.

Parareel.m est une implémentation simple de l’algorithme, en utilisant le programme Heat.m précédent, pour résoudre le même problème modèle, mais maintenant en parallélisant seulement en temps.

Sur cette figure nous montrons finalement que l’algorithme converge linéairement dans ce cas.

Pour avoir une idée du nombre d’itérations effectivement nécessaires dans cet exemple, on peut calculer l’erreur d’approximation du schéma : avec le programme BarTime.m on calcule une solution de référence u_r sur un maillage beaucoup plus fin, et qui est considérée comme la solution exacte. L’erreur du schéma est mesurée par la norme (maximum en temps et en espace par exemple) de la différence entre u et u_r sur la grille de u .

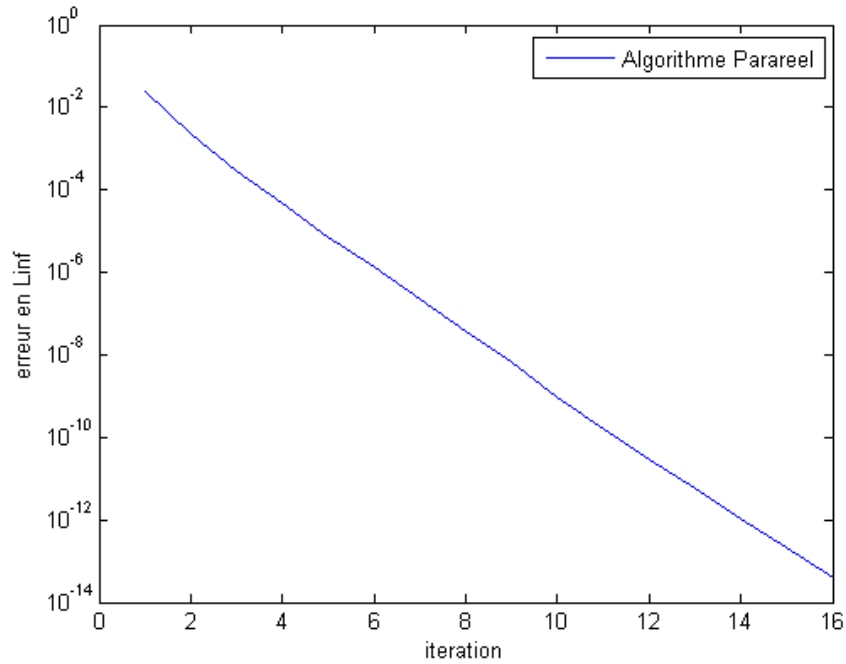


FIGURE 2 – Erreur L^∞ en fonction des itérations pour l'équation de la chaleur traitée par l'algorithme pararéel

On voit alors sur la figure précédente que l'erreur 0.125 est atteinte par l'algorithme pararéel en 2 itérations. Si l'on utilisait 20 processeurs, on pourrait alors calculer cette solution 10 plus vite en parallélisant en temps qu'avec un seul processeur, si l'on néglige le coût de la solution grossière et de la communication.

Ce schéma n'est pas la première tentative de parallélisation en temps et les potentialités de ce schémas peuvent encore être améliorées en utilisant des discrétisations en espace différentes pour les résolutions avec les pas de temps ΔT et δt . Dans le même sens que [6], on peut en effet proposer d'utiliser un maillage H grossier pour propager les sauts entre $u_n(T^{n,-})$ et $u_{n+1}(T^{n,+})$ et par ailleurs un maillage h fin pour les résolutions en parallèle sur chaque $[T^n, T^{n+1}]$. Le schéma de base étant implicite, donc inconditionnellement stable, il n'a pas fallu se préoccuper ici d'une éventuelle condition de type CFL entre les paramètres de discrétisation en espace et en temps. Les gains obtenus semblent être un pas supplémentaire vers le temps réel pour des problèmes dépendants du temps.

5 Parallélisation en espace : Schwarz relaxation d'onde

5.1 Introduction

Dans les problèmes d'évolution en temps, de type chaleur, pour calculer de façon approchée une solution en espace et en temps, on choisira de préférence un schéma implicite en temps. On est alors amené à résoudre à chaque pas de temps une équation elliptique qui pourra être résolue par une méthode de décomposition de domaines. Dans la perspective du couplage de modèles, ou de raffinement local en temps, il peut cependant être intéressant de pouvoir itérer entre les sous-domaines sur un intervalle de temps composé de plusieurs pas de temps. Cela évite également d'avoir à communiquer entre les processeurs à chaque pas de temps, augmentant ainsi la ratio entre temps de calcul et temps de communication. Tout cela fait l'objet des algorithmes de Schwarz relaxation d'onde introduites dans [7] et les méthodes optimisées développées par la suite.

Schwarz relaxation d'onde est une classe d'algorithmes de décomposition de domaine pour résoudre des problèmes d'évolution en parallèle. La façon classique d'appliquer des méthodes de décomposition de domaine aux problèmes d'évolution est la discrétisation de la dimension de temps d'abord uniformément sur le domaine entier, et ensuite appliquer la décomposition de domaine à chaque pas de temps séparément pour résoudre la séquence de problèmes d'état stables obtenus de la discrétisation implicite en temps. Des expériences numériques pour cette approche de la méthode Schwarz avec recouvrement pour l'équation de chaleur bidimensionnelle peuvent être trouvées dans [Meu91].

Pour des problèmes hyperboliques, le cas de l'équation d'onde avec un module de compressibilité discontinu et des champs de densité par sous-domaine a été analysé dans [BGT97] et l'avantage de grilles différentes en espace en raison de la décomposition de domaine a été souligné ("la possibilité de l'affectation à chaque sous-domaine son propre pas spatial rend les simulations numériques beaucoup moins coûteuses"), mais en raison de la discrétisation uniforme en temps, on ne peut pas avoir une discrétisation espace-temps optimale proche de la condition CFL dans chaque sous-domaine dans l'approche classique.

Comme mentionné ci-dessus, le principal inconvénient de cette approche classique est que l'on doit utiliser une discrétisation uniforme en temps sur le domaine entier et on perd ainsi une des principales caractéristiques des méthodes de décomposition de domaine, à savoir, traiter le problème sur chaque sous-domaine différemment numériquement, avec discrétisation approprié aussi bien en temps qu'en espace et adaptée aux sous-problèmes.

La méthode Schwarz relaxation d'onde n'a pas cet inconvénient ; l'algorithme utilise aussi une décomposition de domaine avec recouvrement en espace, comme l'algorithme de Schwarz classique pour des problèmes d'états stables (voir [Sch70]), mais alors l'algorithme résout des problèmes d'évolution sur les sous-domaines et utilise une itération pour converger vers la solution du problème original posé sur le domaine entier, comme dans des méthodes de relaxation d'onde, ce qui explique le nom de l'algorithme. Puisque les problèmes de sous-domaine sont résolus tant dans l'espace que le temps sur les sous-domaines, et des discrétisations appropriées en espace et en temps peuvent être appliquées par sous-domaine. De plus, la communication n'est pas exigée à chaque pas ; le calcul peut être exécuté sur plusieurs pas de temps dans une fenêtre en temps avant que les informations ne soient échangées avec les sous-domaines voisins. Ceci peut être avantageux dans un environnement parallèle où le coût de démarrage d'une connexion avec un autre processeur est significatif.

Une première utilisation de ce type d'algorithme peut être trouvée dans le rapport de recherche [MS87] , où il a été appliqué à l'équation d'onde unidimensionnelle et on montre que l'algorithme

converge dans un nombre fini d'itérations dans ce cas. L'algorithme a été analysé pour des problèmes hyperboliques plus généraux. L'algorithme appliqué aux problèmes paraboliques a été d'abord analysé pour l'équation de chaleur dans [GZ97, GZ02] et indépendamment pour des problèmes de diffusion-convection dans [GK02]. Appliqué aux problèmes paraboliques, l'algorithme a la propriété intéressante d'avoir deux régimes de convergence différents, selon la longueur de l'évolution de temps sur les sous-domaines avant que les informations ne soient échangées avec les sous-domaines voisins. La performance de l'algorithme peut être améliorée en utilisant des meilleures conditions de transmission.

5.2 Présentation

Considérons de nouveau le problème modèle défini dans (1) et considérons la décomposition de Ω en $\Omega_1 = [0, \beta]$ et $\Omega_2 = [\alpha, 1]$, avec $\Gamma_1 = \{\beta\}$ et $\Gamma_2 = \{\alpha\}$, $\delta = \beta - \alpha$. L'algorithme de Schwarz relaxation d'onde parallèle s'écrit pour $n = 1, 2, \dots$

$$\begin{array}{ll} \partial_t u_1^n - \partial_{xx} u_1^n = f \text{ dans } \Omega_1 \times [0, T] & \partial_t u_2^n - \partial_{xx} u_2^n = f \text{ dans } \Omega_2 \times [0, T] \\ u_1^n(., 0) = u_0 \text{ dans } \Omega_1 & u_2^n(., 0) = u_0 \text{ dans } \Omega_2 \\ u_1^n(0, .) = g_g & u_2^n(0, .) = g_d \\ u_1^n(\beta, .) = u_2^{n-1}(\beta, .) \text{ sur } [0, T] & u_2^n(1, .) = u_1^{n-1}(\alpha, .) \text{ sur } [0, T] \end{array}$$

(ce système sera noté (A))

Les données initiales de l'algorithme (à ne pas confondre avec la donnée initiale du problème d'évolution u_0) sont des fonctions du temps $g_i(t)$ qui servent de condition à la limite sur $\Gamma_i \times [0, T]$. L'algorithme de Schwarz relaxation d'onde alterné s'obtient de la même façon en remplaçant la condition de transmission en $x = \alpha$ par $u_2^n = u_1^n$.

Théorème 1 *Pour tout α et β tels que $\delta = \beta - \alpha > 0$, les algorithmes de Schwarz relaxation d'onde alterné et parallèle convergent. Pour $i = 1$ ou 2 , on a pour l'algorithme parallèle l'estimation*

$$\sup_{\substack{t \in [0, T] \\ x \in \Omega_i}} |u_i^{2n}(x, t) - u(x, t)| \leq \left(\frac{\alpha(1 - \beta)}{\beta(1 - \alpha)} \right)^n \sup_i \sup_{\substack{t \in [0, T] \\ x \in \Gamma_i}} |u_i^0(x, t) - u(x, t)|$$

Démonstration : Nous traitons de l'algorithme parallèle, et cette fois nous utilisons une preuve basée sur le principe de maximum. Notons $e_i^n = u_i^n - u$ l'erreur dans le sous-domaines i à l'étape n , qui est maintenant une fonction de x et t . Par linéarité, les erreurs sont solutions des problèmes aux limites homogènes (i.e. avec $f \equiv 0$ et $u_0 \equiv 0$), et les mêmes conditions de transmission.

Définissons un algorithme stationnaire, pour des fonctions \tilde{e} de l'espace uniquement

$$\begin{array}{ll} \partial_{xx} \tilde{e}_1^n = 0 \text{ dans } \Omega_1, & \partial_{xx} \tilde{e}_2^n = 0 \text{ dans } \Omega_2, \\ \tilde{e}_1^n(0) = 0, & \tilde{e}_2^n(1) = 0, \\ \tilde{e}_1^n(\beta) = \sup_{t \in [0, T]} |e_1^n(\beta, t)|, & \tilde{e}_2^n(\alpha) = \sup_{t \in [0, T]} |e_2^n(\alpha, t)|, \end{array}$$

Fixons les données initiales de l'algorithme stationnaire par $\tilde{g}_i = \sup_{t \in [0, T]} |g_i(t) - u_{|\Gamma_i(t)}|$. Puisque les \tilde{e}_i^n sont harmoniques, elles vérifient le principe du maximum : Nous allons montrer que pour tout (x, t) dans $\Omega_i \times [0, T]$, $|e_i^n(x, t)| \leq \tilde{e}_i^n(x)$. Pour cela définissons les fonctions de x et t , $d_i^{n\pm} = \tilde{e}_i^n \pm e_i^n$. Par définition des \tilde{e}_i^n , les $d_i^{n\pm}$ sont positives ou nulles sur Γ_i , et vérifient

$$\begin{aligned} \partial_t d_1^{n\pm} - \partial_{xx} d_1^{n\pm} &= 0 \text{ dans } \Omega_1 \times [0, T], & \partial_t d_2^{n\pm} - \partial_{xx} d_2^{n\pm} &= 0 \text{ dans } \Omega_2 \times [0, T], \\ d_1^{n\pm}(\cdot, 0) &\geq 0 \text{ dans } \Omega_1, & d_2^{n\pm}(\cdot, 0) &\geq 0 \text{ dans } \Omega_1, \\ d_1^{n\pm}(0, \cdot) &= 0, & d_2^{n\pm}(1, \cdot) &= 0, \\ d_1^{n\pm}(\beta, \cdot) &\geq 0 \text{ sur } [0, T], & d_2^{n\pm}(\alpha, \cdot) &\geq 0 \text{ sur } [0, T], \end{aligned}$$

Par le principe du maximum pour l'équation de la chaleur, nous en déduisons que $d_i^{n\pm}$ est positif ou nul sur $\Omega_i \times [0, T]$, et donc que $|e_i^n| \leq \tilde{e}_i^n$ sur $\Omega_i \times [0, T]$. Nous pouvons alors écrire la suite des égalités et inégalités

$$\tilde{e}_1^n(\beta) = \sup_{t \in [0, T]} |e_1^n(\beta, t)| = \sup_{t \in [0, T]} |e_2^{n-1}(\beta, t)| \leq \tilde{e}_2^{n-1}(\beta) = \frac{1-\beta}{1-\alpha} \tilde{e}_2^{n-1}(\alpha),$$

$$\tilde{e}_2^{n-1}(\alpha) = \sup_{t \in [0, T]} |e_2^{n-1}(\alpha, t)| = \sup_{t \in [0, T]} |e_1^{n-2}(\alpha, t)| \leq \tilde{e}_1^{n-2}(\alpha) = \frac{\alpha}{\beta} \tilde{e}_1^{n-2}(\beta).$$

Nous en déduisons que la suite $\tilde{e}_1^{2n}(\beta)$ est une suite géométrique,

$$\tilde{e}_1^{2n}(\beta) \leq \left(\frac{\alpha(1-\beta)}{\beta(1-\alpha)} \right) \tilde{e}_1^{2n-2}(\beta).$$

la raison est strictement inférieure à 1, cette suite converge vers 0, de même que $\tilde{e}_2^{2n}(\alpha)$. Par le principe du maximum les suites \tilde{e}_i^{2n} tendent vers 0 uniformément dans leurs domaines respectifs, il en est de même des suites e_i^{2n} .

La raison de la suite géométrique e_i^{2n} ne dépend pas du temps, la convergence des itérées est donc pour tout temps au moins linéaire. En fait, on peut estimer plus précisément la convergence sur un intervalle de temps borné :

Théorème 2 Dans le cas de deux demi-droites infinies, $\Omega_1 =]-\infty, \beta]$ et $\Omega_2 = [\alpha, +\infty[$ avec $\delta = \beta - \alpha > 0$, les algorithmes de Schwarz relaxation d'onde alterné et parallèle convergent superlinéairement sur un intervalle de temps borné. Par exemple par l'algorithme parallèle, on a dans le sous-domaine de gauche :

$$\sup_{t \in [0, T]} |(u_1^{2n} - u)(\alpha, \cdot)| \leq \operatorname{erf}\left(\frac{n\delta}{\sqrt{T}}\right) \sup_{t \in [0, T]} |u_1^0 - u)(\alpha, \cdot)|$$

où la fonction d'erreur complémentaire notée **erfc** est définie par $\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-s^2} ds$.

Démonstration : La transformée de Laplace d'une fonction intégrable en temps est définie pour $\Re(s) > 0$ par

$$\mathcal{L}f : s \mapsto \hat{f}(s) = \int_0^{+\infty} f(t) e^{-st} dt.$$

En tant que fonction de t , les erreurs e_i^n sont définies sur $[0, T]$, nulles en 0. On peut les prolonger en des fonctions continues sur \mathbb{R}_+ , et prolonger les équations sur tout \mathbb{R}_+ . Leurs transformées de Laplace

en temps $\hat{e}_i^n(x, s)$ sont alors solutions des problèmes

$$\begin{aligned} s\hat{e}_1^n - \partial_{xx}\hat{e}_1^n &= 0 \text{ dans } \Omega_1, & s\hat{e}_2^n - \partial_{xx}\hat{e}_2^n &= 0 \text{ dans } \Omega_2, \\ \hat{e}_1^n(\beta, s) &= \hat{e}_2^{n-1}(\beta, s), & \hat{e}_2^n(\alpha, s) &= \hat{e}_1^{n-1}(\alpha, s). \end{aligned}$$

La solution générale de l'équation différentielle est une combinaison linéaire de $e^{\sqrt{s}x}$ et $e^{-\sqrt{s}x}$. Mais puisque nous cherchons des solutions bornées en la variable x , il ne reste que

$$\hat{e}_1^n(x, s) = \hat{e}_2^{n-1}(\beta, s)e^{\sqrt{s}(x-\beta)}, \quad \hat{e}_2^n(x, s) = \hat{e}_1^{n-1}(\alpha, s)e^{-\sqrt{s}(x-\alpha)}$$

En évaluant aux interfaces sur deux itérations, nous obtenons par récurrence

$$\hat{e}_1^{2n}(\alpha, s) = (\rho^D(s))^n \hat{e}_1^0(\alpha, s), \quad \hat{e}_2^{2n}(\beta, s) = (\rho^D(s))^n \hat{e}_2^0(\beta, s) \quad (*)$$

où le facteur de la convergence ρ^D est défini par

$$\rho^D(s) = e^{-2\delta\sqrt{s}}.$$

Les relations (*) s'interprètent en terme de produit de convolution par transformée de Laplace inverse :

$$\hat{e}_1^{2n}(\alpha, \cdot) = G * \hat{e}_1^0(\alpha, s), \quad \hat{e}_2^{2n}(\beta, \cdot) = G * \hat{e}_2^0(\beta, s),$$

où G est la transformée de Laplace inverse de $(\rho^D(s))^n$, donnée par

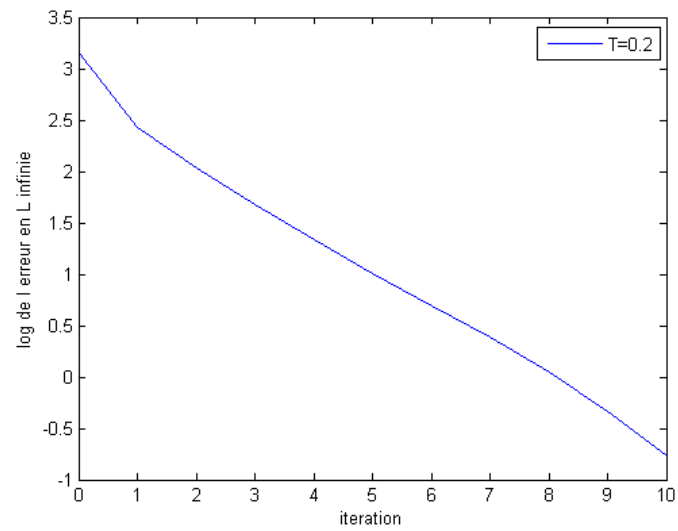
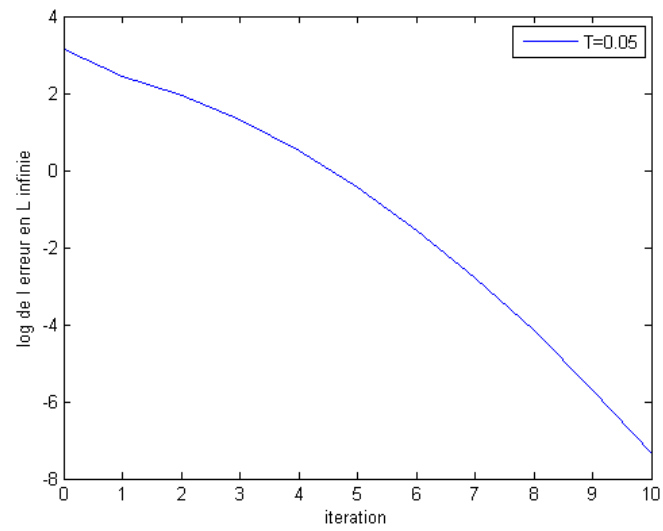
$$G(t) = \frac{n\delta}{\sqrt{\pi}t^3} e^{-\frac{n^2\delta^2}{t}}.$$

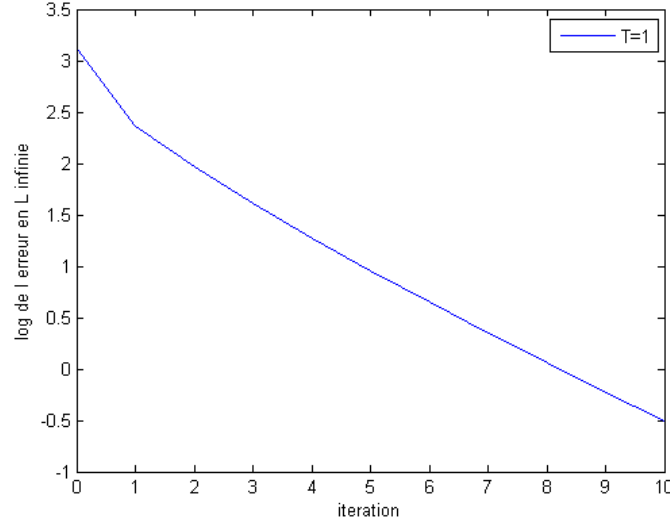
Nos pouvons alors écrire, pour tout $t \in [0, T]$,

$$|\hat{e}_1^{2n}(\alpha, t)| = \left| \int_0^t \hat{e}_1^0(\alpha, t-\tau) G(\tau) d\tau \right| \leq \sup_{t \in [0, T]} |\hat{e}_1^0(\alpha, t)| \int_0^T G(t) dt = \sup_{t \in [0, T]} |\hat{e}_1^0(\alpha, t)| \operatorname{erfc}\left(\frac{n\delta}{\sqrt{T}}\right).$$

La fonction erfc tend vers 0 à l'infini, la suite $\hat{e}_1^{2n}(\alpha, \cdot)$ tend donc uniformément vers 0.

SWR.m est un script qui permet de tester cet algorithme.





Cette figure représente les courbes de convergence de l'algorithme pour le problème de référence, avec les temps finaux $T = 0.05$, $T = 0.2$ et $T = 1$. On voit bien la convergence linéaire pour $T = 1$, « grand », la convergence super-linéaire pour $T = 0.05$, « petit », et un régime mixte, linéaire suivi de superlinéaire pour le temps intermédiaire $T = 0.2$.

5.3 Méthodes de Schwarz relaxation d'onde optimisées

Comme dans le cas des équation elliptiques, la convergence peut être beaucoup améliorée par l'introduction de condition de transmission de type Robin. Dans le cas parallèle, les conditions de transmissions dans le système (A) sont alors remplacées par

$$\partial_x u_1^n + p u_1^n = \partial_x u_2^{n-1} + p u_2^{n-1} \text{ sur } \Gamma_1 \times [0, T],$$

$$\partial_x u_2^n - p u_2^n = \partial_x u_1^{n-1} - p u_1^{n-1} \text{ sur } \Gamma_2 \times [0, T].$$

L'algorithme est initialisé par les conditions aux limites

$$\partial_x u_2^0 + p u_2^0 = g_2 \text{ sur } \Gamma_1 \times [0, T],$$

$$\partial_x u_1^0 - p u_1^0 = g_1 \text{ sur } \Gamma_2 \times [0, T].$$

Théorème 3 *Pour tout $p > 0$, les algorithmes de Schwarz relaxation d'onde alterné ou parallèle avec des conditions de Robin convergent avec ou sans recouvrement :*

$$\lim_{n \rightarrow +\infty} \sup_{t \in [0, T]} \int (u_i^n(x, t) - u(x, t))^2 dx = 0$$

Heat.m est un solveur Matlab avec des conditions de Robin. Et avec le programme de test Matlab SWRR.m, nous avons établi les courbes de convergence suivantes. Nous utilisons une condition de Robin avec un paramètre très grand pour émuler une condition de Dirichlet sur le bord extérieur.

Il apparaît clairement qu'avec la condition de Robin optimisé, on peut nettement accélérer la convergence, même sur un intervalle de temps de grande taille, $T = 1$. On obtient ainsi une vitesse

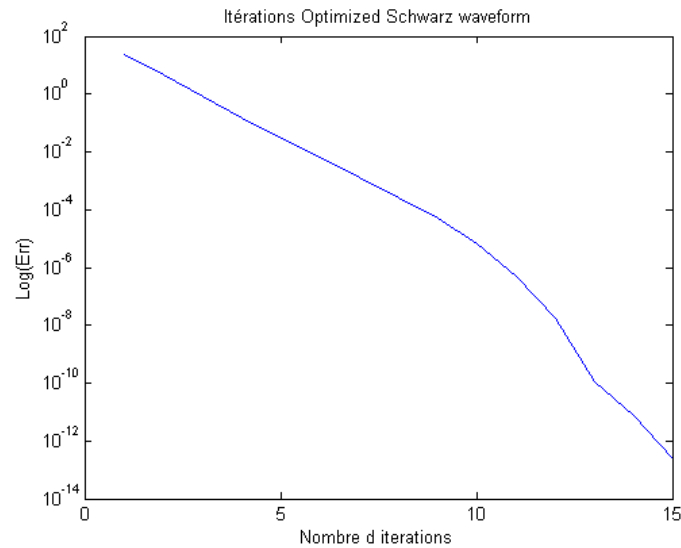


FIGURE 3 – Comparaison des méthodes de Schwarz relaxation d'onde classique et optimisée

de convergence qui était seulement atteignable sur des intervalles de temps très court, les figures précédentes.

Un dernier commentaire important s'impose ici pour les méthodes de Schwarz relaxation d'onde : on n'a pas besoin de grille grossière, même avec beaucoup de sous-domaines, si l'on calcule sur une fenêtre en temps courte, car dans ce cas, la condition initiale est plus importante que les conditions aux interfaces, et la condition initiale est juste. Par contre, si l'on calcule sur un intervalle de temps long, il faut ajouter une composante grille grossière, comme pour les méthodes de décomposition de domaine pour des problèmes stationnaires, pour obtenir une méthode scalable. L'étude de cette question est en cours.

6 Parallélisation en espace et en temps : Couplage Pararéel/SWR

Nous proposons dans cette partie de réaliser un parallélisme espace-temps pour la résolution de l'équation de chaleur, en couplant l'approche de Schwarz relaxation d'onde et l'approche pararéelle. Une façon naturelle de coupler les deux approches est de remplacer dans l'algorithme pararéel le solveur fin par un solveur de relaxation d'onde.

Une stratégie différente est présentée dans [8]. Le domaine $\Omega \times [0, T]$ est décomposé es sous domaines espace-temps $\Omega_{i,n} = \Omega_i \times [T_n, T_{n+1}]$, représentés ci-dessous

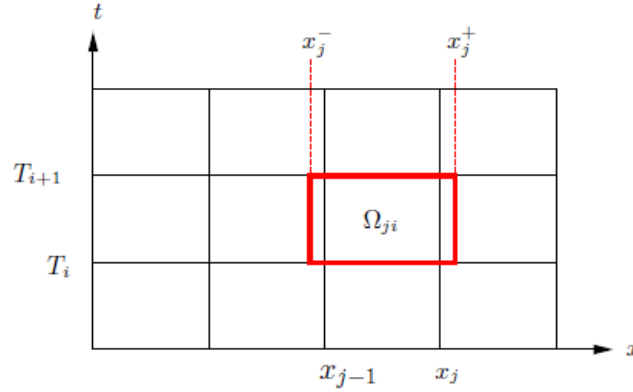


FIGURE 4 – Décomposition espace-temps pour l'algrithme pararéel-Schwarz relaxation d'onde

Ici B^\pm représente un opérateur sur la frontière, comme l'identité pour une condition de Dirichlet, une dérivée en x pour une condition de Neumann, ou encore une combinaison des deux pour une condition de Robin.

Il faut maintenant calculer d'une manière itérative des conditions initiales (donnée par le principe de Pararéel) et des conditions de transmission (donnée par le Schwarz relaxation d'onde) de plus en plus précise pour chaque sous-domaine en espace-temps. Pour cela, l'algorithme pararéel-Schwarz relaxation d'onde approche les conditions initiales par une méthode pararéelle, et les conditions de transmission par une méthode de Shwarz relaxation d'onde.

L'algorithme pararréel, pour le le problème modèle

$$\begin{aligned} \partial_t u - \partial_{xx} u &= 0 \text{ dans } \Omega = (0, 1) \times (0, T), \\ B^- u(0, t) &= g_0(t) \quad t \in (0, T), \\ B^+ u(1, t) &= g_1(t) \quad t \in (0, T), \\ u(x, 0) &= u_0(x) \quad x \in \Omega. \end{aligned} \quad (**)$$

est basé en une décomposition de l'intervalle en temps $[0, T]$ en sous intervalles, donnés par

$$0 = T_0 < T_1 < T_2 < \dots < T_N = T,$$

l'algorithme est alors défini en utilisant deux opérateurs de propagation :

- un opérateur grossier $G(t_2, t_1, u_1, g_0, g_1)$ qui fournit une approximation grossière de la solution $u(x, t_2)$ de $(**)$ avec une condition initiale donnée $u(x, t_1) = u_1(x)$ et de conditions limites données g_0 et g_1 ,
- un opérateur fin $F(t_2, t_1, u_1, g_0, g_1)$, qui donne une approximation plus précise de la même solution avec la condition initiale $u(x, t_1) = u_1(x)$ et les conditions limites g_0 et g_1 .

Commençant par une première approximation U_n^0 aux instants $T_0, T_1, T_2, \dots, T_{N-1}$, l'algorithme Pararéel réalise pour $k = 0, 1, 2, \dots$ l'itération de correction

$$U_{n+1}^{k+1} = F(t_{n+1}, t_n, U_n^k, g_0, g_1) + G(t_{n+1}, t_n, U_n^{k+1}, g_0, g_1) - G(t_{n+1}, t_n, U_n^k, g_0, g_1),$$

qui est rien d'autre que la méthode de tir multiple avec un Jacobien approché.

Contrairement à l'algorithme Pararéel, la méthode de Schwarz relaxation d'ondes pour le problème modèle $(**)$ est basée sur une décomposition spatiale seulement, dans la cas le plus générale en des sous-domaines recouvrant

$$\Omega = \bigcup_{i=1}^I (x_i^-, x_i^+),$$

comme on le montre dans la figure précédente. Ici les frontières x_t^\pm des sous-domaines recouvrant sont construites à partir d'une décomposition non-recouvrante donnée par la décomposition

$$0 := x_0 < x_1 < \dots < x_I := 1,$$

en ajoutant et soustrayant la moitié du recouvrement

$$x_i^- := x_{i-1} - \frac{L}{2}, x_i^+ := x_i + \frac{L}{2}$$

à part pour le premier et le dernier point,

$$x_1^- = x_0 \text{ et } x_N^+ = x_N$$

Donnant une condition initiale aux interfaces, soit $B_i^\pm u_i^0$, l'algorithme de Schwarz relaxation d'onde résoud itérativement pour $k = 1, 2, \dots$ les problèmes des sous-domaines

$$\begin{aligned} \partial_t u_i^k - \partial_{xx} u_i^k &= 0 \text{ dans } \Omega_i \times (0, T) \\ u_i^k(x, 0) &= u_0 \text{ dans } \Omega_i \\ B_i^- u_i^k(x_i^-, t) &= B_i^- u_{i-1}^{k-1}(x_i^-, t) \quad t \in (0, T) \\ B_i^+ u_i^k(x_i^+, t) &= B_i^+ u_{i+1}^{k-1}(x_i^+, t) \quad t \in (0, T) \end{aligned} \quad (***)$$

Là encore, les opérateurs B_i^\pm sont des opérateurs de transmission : dans le cas de l'identité, nous avons l'algorithme de Schwarz relaxation d'onde classique ; pour des conditions de transmission du type Robin ou d'un ordre plus élevé, on obtiendrait un algorithme de Schwarz relaxation d'onde optimisé, si les paramètres des conditions de transmission sont choisis pour optimiser la convergence de l'algorithme.

L'algorithme Pararéel Schwarz relaxation d'onde combine les deux techniques pour une décomposition espace-temps plus générale et propose la stratégie suivante : des conditions initiales $u_{0,i,n}^k(x)$ au temps T_i et des conditions aux limites $B_i^- u_{i-1,n}^k(t)$ et $B_i^+ u_{i+1,n}^k(t)$ sont données pour chaque $\Omega_{i,n}$. Nous calculons maintenant pour tout $i = 1, 2, \dots, I$ et $n = 1, 2, \dots, N$

1. des approximations précises

$$u_{i,n}^{k+1}(x, t) := F_{i,n}(u_{0,i,n}^k, B_i^- u_{i-1,n}^k, B_i^+ u_{i+1,n}^k)$$

dans tous les sous-domaines espace-temps $\Omega_{i,n}$, en parallèle, utilisant l'opérateur d'évolution le plus précis.

2. Pour $n = 0, 1, \dots$, de nouvelles conditions initiales, en utilisant une correction pararéelle en temps,

$$u_{0,i,n+1}^{k+1} = u_{i,n}^{k+1}(\cdot, T_{n+1}) + G_{i,n}(u_{0,i,n}^{k+1}, B_i^- u_{i-1,n}^{k+1}, B_i^+ u_{i+1,n}^{k+1}) \\ - G_{i,n}(u_{0,i,n}^k, B_i^- u_{i-1,n}^k, B_i^+ u_{i+1,n}^{k+1})$$

Nous présentons maintenant un premier résultat de convergence pour l'algorithme paréel-Shwarz relaxation d'onde :

Théorème 4 (*Convergence superlinéaire*) Soient $F_{j,i}$ une solution exacte, $G_{j,i}$ une approximation en temps avec Euler implicite, et la solution exacte en espace, et on prend une décomposition en espace du domaine en deux sous-domaines avec recouvrement. Si l'algorithme utilise les conditions de transmission du type Dirichlet, ie B_j^\pm , l'identité, alors il y a convergence super-linéaire vers la solution.

Nous illustrons dans la partie suivante le comportement de plusieurs variante de l'algorithme avec différents types de conditions de transmission.

7 Résultats numériques

Nous allons présenter dans cette partie les résultats de l'application de l'algorithme Pararéel Schwarz relaxation d'ondes. Nous avons choisi de présenter, dans ce rapport, trois cas qui sont premièrement le cas bi-domaine où nous allons partager le domaine en deux sous-domaines en espace où on présentera les résultats pour l'équation de la chaleur et l'équation d'advection diffusion. Dans le second cas on traitera le cas multi-domaines en espace et dans ce cas on présentera les résultats pour l'équation de la chaleur.

7.1 Équation de la chaleur

Dans toutes les expériences qui suivent, à part dans les cas mentionnés, nous utiliserons le domaine $\Omega = (0, 1)$ et l'intervalle en temps $(0, T)$ avec $T = 0.2$. Nous avons également discrétisé l'équation de la chaleur avec une discrétisation différences finis centrées en espace avec $\Delta x = \frac{1}{60}$, et une discrétisation Euler implicite en temps, avec $\Delta t = \frac{0.2}{100}$, et on utilisera une décomposition en 2 sous-domaines égaux en espace avec un recouvrement de taille $4\Delta x$.

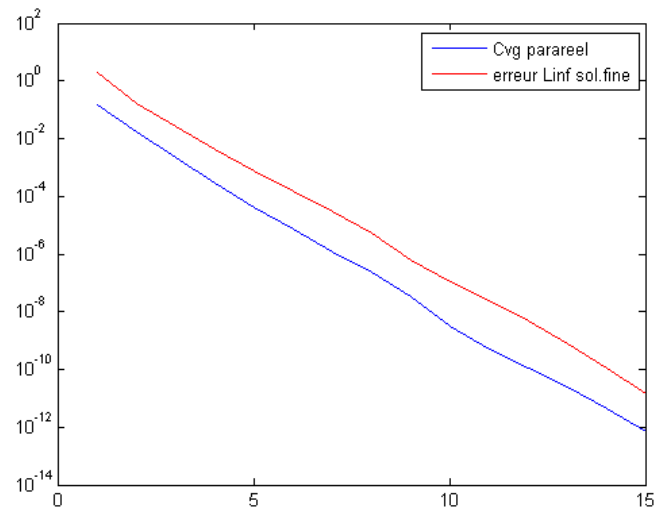


FIGURE 5 – Courbe de convergence de Pararéel Schwarz relaxation d'onde (Dirichlet)

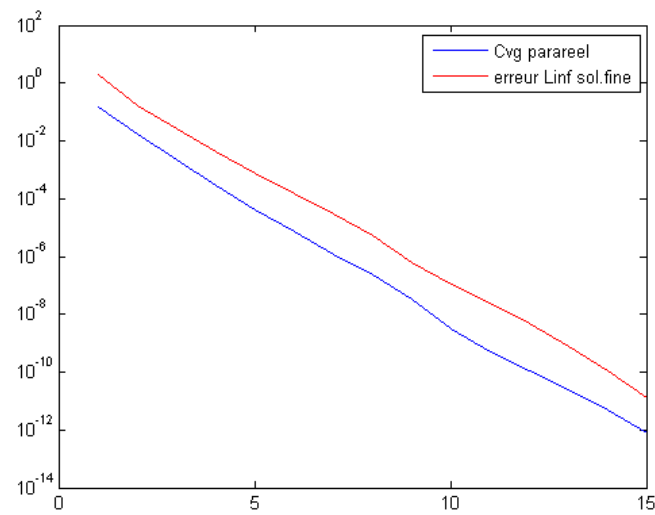
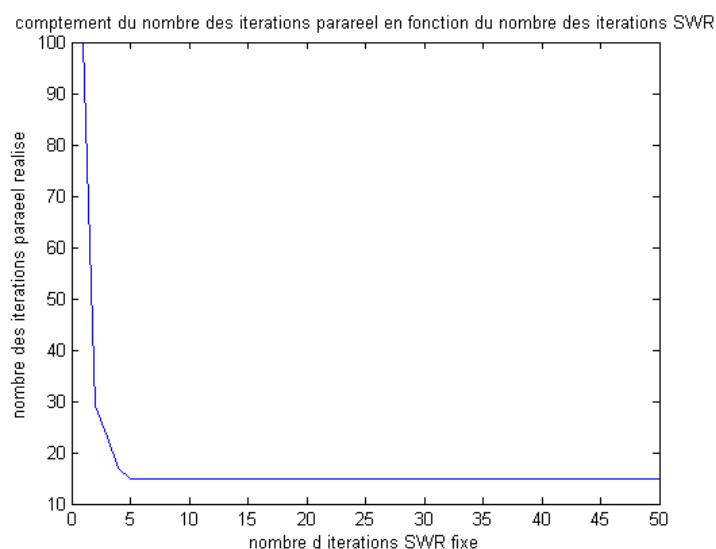
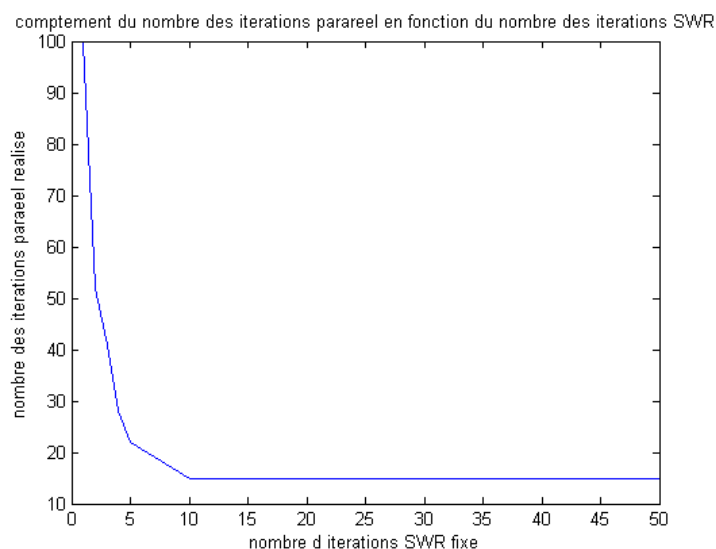


FIGURE 6 – Courbe de convergence de Pararéel Schwarz relaxation d'onde

Coût [a;b]=[0;1] d=6	Nbr iteration max it para = 100 it swr = 100	Nbr iteration max it para = 500 it swr = 10	Nbr iteration max it para = 1000 it swr = 2
Condition de transmission type Dirichlet	it parareel: 16 It swr réalisée: 1600 cout en temps: 3.8 e-1	it parareel: 16 It swr réalisée: 160 cout en temps: 4.8e-2	it parareel: 52 It swr réalisée: 104 cout en temps: 4.2e-2
Condition de transmission type Robin	It parareel: 15 It swr réalisée: 1500 cout en temps: 5.1-1	it parareel: 15 It swr réalisée: 150 cout en temps: 5.7e-2	it parareel: 29 It swr réalisée: 58 cout en temps: 2.4e-2



Nous observons d'après les figures précédentes que dans le cas de l'équation de la chaleur avec une décomposition en espace en deux sous-domaines et avec des conditions de transmission de type Dirichlet puis du type Robin nous obtenons les résultats suivant. On observe d'abord que l'algorithme pararéel Schwarz relaxation converge et que les solutions des sous-problèmes convergent bien vers la solution mono-domaine dans deux cas. Le premier cas lorsque nous laissons converger l'algorithme Schwarz relaxation d'onde, c'est à dire l'opérateur fini et le deuxième cas lorsque nous fixons le nombre d'itérations de cet algorithme, bien sûr en laissant toujours le Pararéel converger i.e. en laissant son nombre d'itérations libre.

On observe alors deux choses intéressantes qui permettent d'améliorer l'algorithme du couplage et qui sont tout d'abord le gain en coût qu'on observe lorsqu'on ne fait pas converger le Schwarz relaxation d'onde et la baisse du nombre d'itérations à effectuer aussi bien pour le Schwarz que le Pararéel et ceci est encore plus vrai avec les conditions du type Robin. Ensuite nous observons que le nombre des itérations de l'algorithme pararéel décroît exponentiellement en fonction du nombre des

itérations de l'algorithme Schwarz relaxation d'onde fixé. Donc ceci montre assez bien qu'il y a un nombre suffisant d'itérations de Schwarz relaxation d'onde pour obtenir la convergence.

7.2 Équation d'advection diffusion

Dans cette partie nous allons étudier l'équation d'advection diffusion donnée par l'expression suivante :

$$\frac{\partial u}{\partial t} - \nu \Delta x + cu = f$$

Et nous observons dans ce qui suit des résultats obtenus avec les mêmes conditions le seul changement qui interviendra est l'utilisation de différents schémas de discrétisations en espace du terme advectif et cela avec la méthode des différences finis.

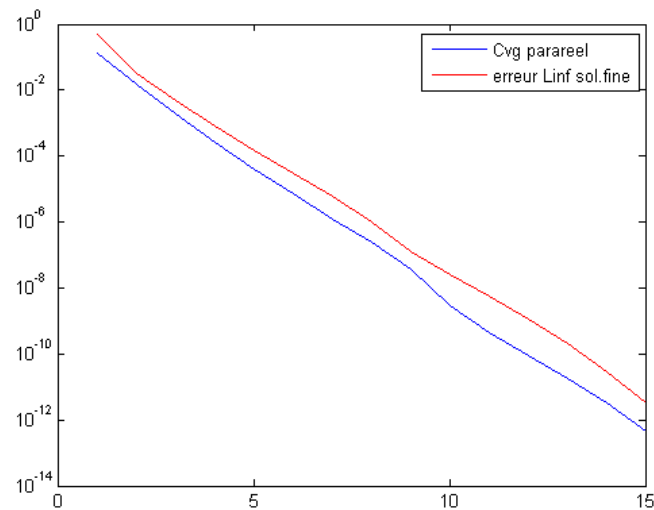


FIGURE 7 – Courbe de convergence de Pararéel Schwarz relaxation d'onde

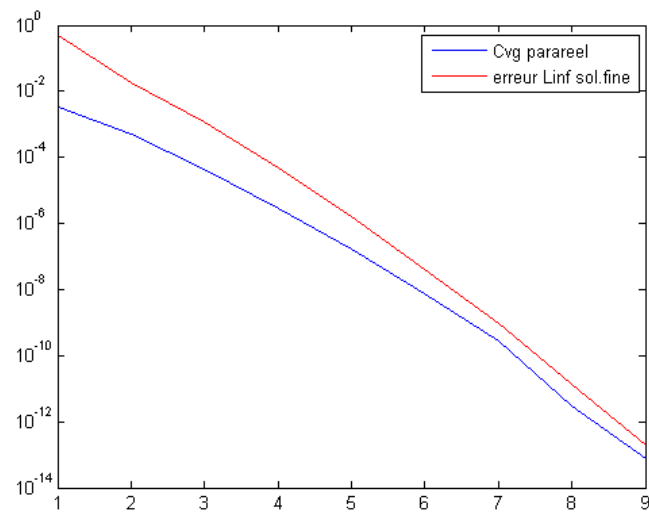
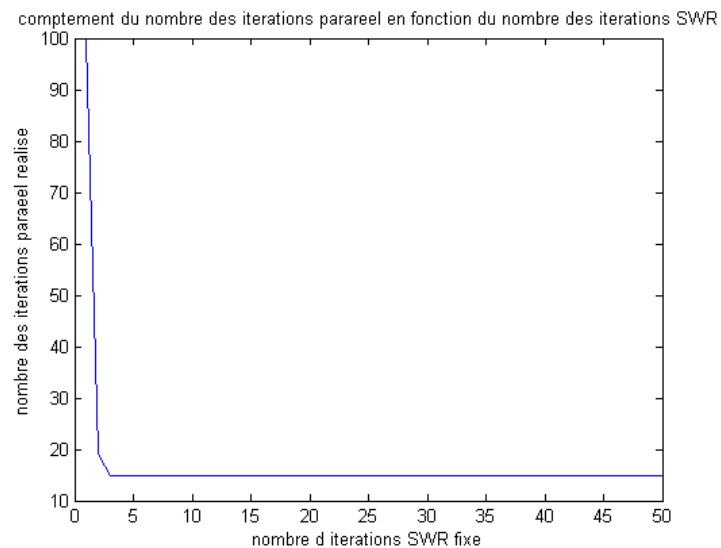


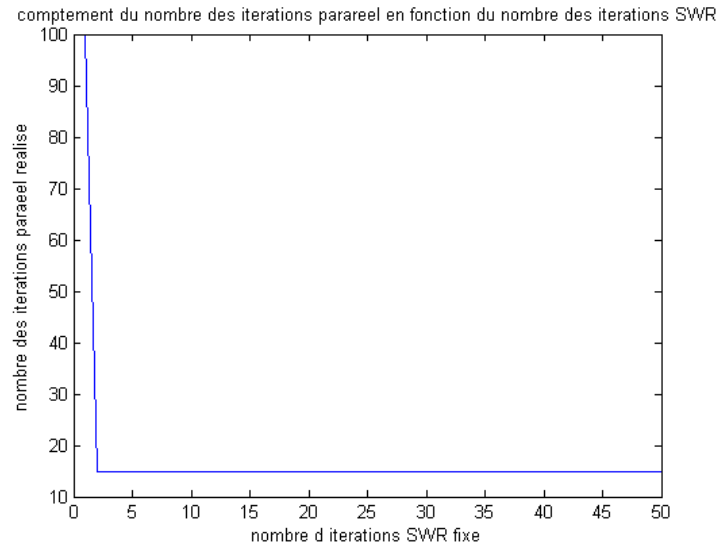
FIGURE 8 – Courbe de convergence de Pararéel Schwarz relaxation d'onde

Coût (nu=1, c=1) (Diffusion centré ordre 2)	Nbr iteration max it para = 100 it swr = 100	Nbr iteration max it para = 500 it swr = 10	Nbr iteration max it para = 1000 it swr = 2
Advection décentré amont ordre 1	it parareel: 15 Itswr réalisée: 104 cout en temps: 6.1e-2	it parareel: 15 Itswr réalisée: 101 cout en temps: 7.2e-2	it parareel: 19 Itswr réalisée: 38 cout en temps: 2.2e-2
Advection décentré aval ordre 1	it parareel: 15 Itswr réalisée: 111 cout en temps: 5.4e-2	it parareel: 15 Itswr réalisée: 105 cout en temps: 5.3e2	it parareel: 20 Itswr réalisée: 40 cout en temps: 2.3e-2
Advection centré ordre 2	it parareel: 15 Itswr réalisée: 113 cout en temps: 5.2e-2	it parareel: 15 Itswr réalisée: 107 Cout en temps: 4.5e-2	it parareel: 20 Itswr réalisée: 40 cout en temps: 2.3e-2

Coût ($\nu=0, c=1$)	Nbr iteration max it para = 100 it swr = 100	Nbr iteration max it para = 500 it swr = 10	Nbr iteration max it para = 1000 it swr = 2
advection décentré amont ordre 1	it parareel: 9 itswr réalisée: 18 cout en temps: 1.04e-2	it parareel: 9 itswr réalisée: 18 cout en temps: 1.03e-2	it parareel: 9 itswr réalisée: 18 cout en temps: 1.02e-2
advection décentré aval ordre 1	it parareel: 8 itswr réalisée: 22 cout en temps: 1.3e-2	it parareel: 8 itswr réalisée: 16 cout en temps: 1.1e-2	it parareel: 8 itswr réalisée: 16 cout en temps: 9.1e-3
advection centré ordre 2	it parareel: 10 itswr réalisée: 38 cout en temps: 1.9e-2	it parareel: 10 itswr réalisée: 38 cout en temps: 1.7e-2	it parareel: 10 itswr réalisée: 20 cout en temps: 1.2e-2

Coût	Nbr iteration max it para = 100 it swr = 100	Nbr iteration max it para = 500 it swr = 10	Nbr iteration max it para = 1000 it swr = 2
($\nu=1, c=1$)	it parareel: 15 itswr réalisée: 63 cout en temps: 5.8e-2	it parareel: 15 itswr réalisée: 63 cout en temps: 5.2e-2	it parareel: 15 itswr réalisée: 30 cout en temps: 2.1e-2
($\nu=0, c=1$)	it parareel: 9 itswr réalisée: 18 cout en temps: 1.6e-2	it parareel: 9 itswr réalisée: 18 cout en temps: 1.7e-2	it parareel: 9 itswr réalisée: 18 cout en temps: 1.5e-2





Donc dans ce qui précède comme courbes et tableaux on étudie l'équation d'advection diffusion, dans différents cas afin de pouvoir valider la convergence de l'algorithme Pararéel Schwarz relaxation d'onde. Pour ce type d'équation on a essayé de jouer sur les paramètres ν et c pour pouvoir ressortir les résultats sur le gain en coût. Donc comme on la vu précédemment avec l'équation de la chaleur que le fait de jouer sur le nombre des itérations de Schwarz relaxation d'onde ne nous empêchera pas d'obtenir la convergence globale de l'algorithme Schwarz relaxation d'onde et qu'au contraire ça nous a permis d'avoir un gain sur le coût.

7.3 Cas multi-domaines

Et enfin dans une dernière partie nous illustrerons les résultats que nous obtenons en faisant augmenter la dépendance de l'algorithme en nombre de sous-domaines en espace sur le cas de l'équation de la chaleur en se situant dans les mêmes conditions précédentes et utilisant cette fois-ci 5 sous-domaines en espace.

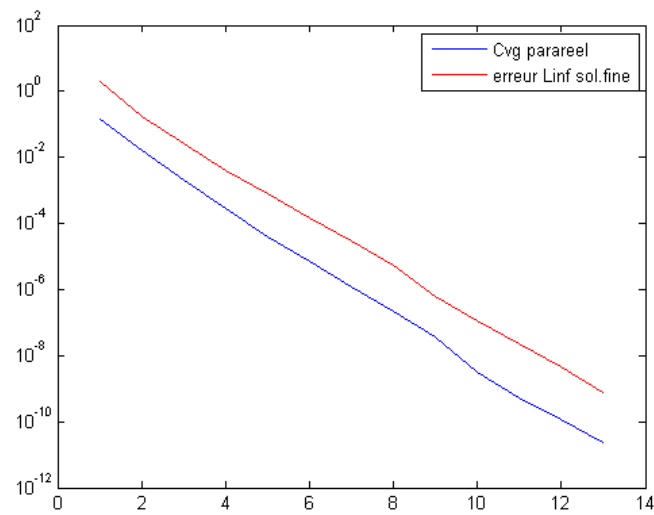


FIGURE 9 – Courbe de convergence de Pararéel Schwarz relaxation d'onde

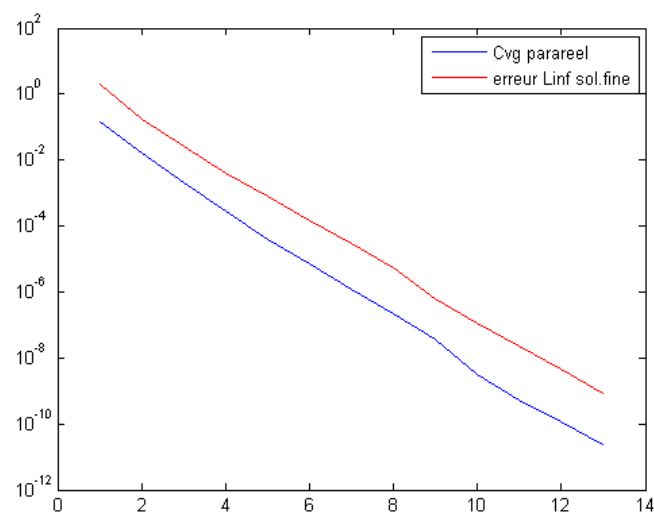
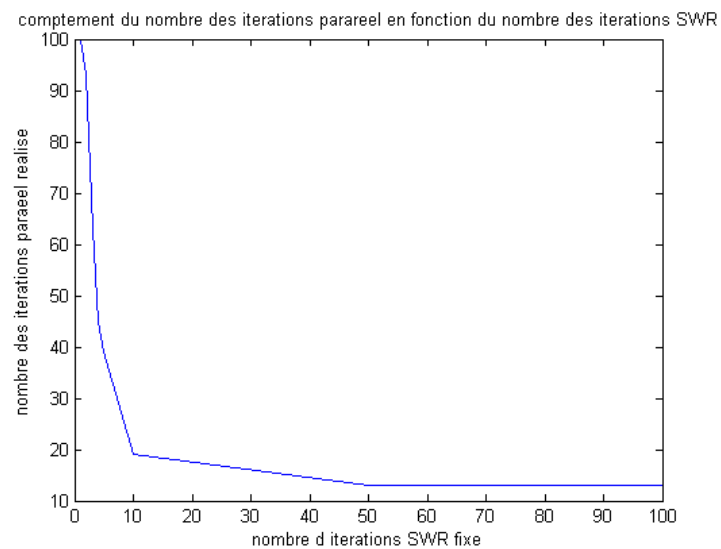
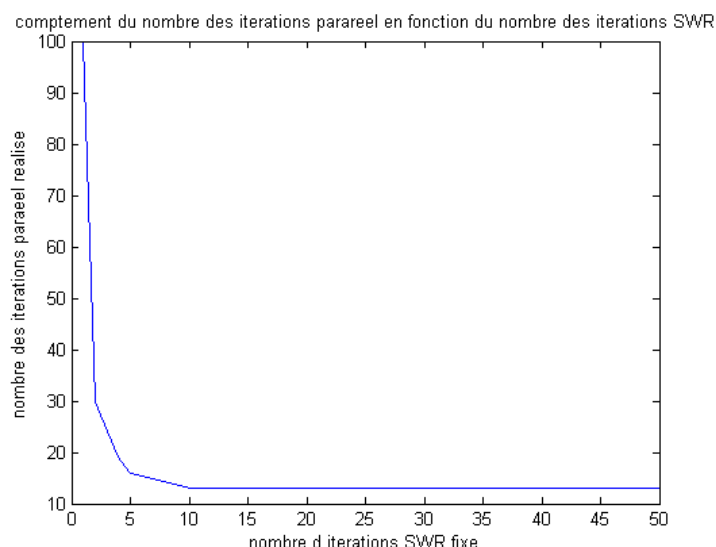


FIGURE 10 – Courbe de convergence de Pararéel Schwarz relaxation d'onde

Coût [a;b]=[0;1] 5 domaines	Nbr iteration max it para = 100 it swr = 100	Nbr iteration max it para = 500 it swr = 10	Nbr iteration max it para = 1000 it swr = 2
Condition de transmission type Dirichlet	it parareel: 13 It swr réalisée: 497 cout en temps: 1.9e-1	it parareel: 19 It swr réalisée: 190 cout en temps: 7.6e-2	it parareel: 93 It swr réalisée: 186 cout en temps: 9.6e-2
Condition de transmission type Robin	it parareel: 13 It swr réalisée: 343 cout en temps: 7.9e-2	it parareel: 13 It swr réalisée: 130 cout en temps: 3.4e-2	it parareel: 30 It swr réalisée: 60 cout en temps: 1.9e-2





On retrouve donc bien les mêmes résultats de convergence avec la première partie aussi bien pour les conditions de transmission du type Dirichlet ou Robin. On retrouve également le même gain en modifiant le nombre des itérations fixé pour l'algorithme Schwarz relaxation d'onde. Cette expérience est intéressante dans le sens où elle nous a permis de valider nos précédentes expériences et elle nous a permis également de voir que l'augmentation du nombre des sous-domaines en espace n'a pas eu une influence considérable vis à vis du gain en coût.

Au cours de ces expériences nous avons vu la convergence de l'algorithme lorsque 1 (Schwarz relaxation d'onde classique) ou plusieurs sous-intervalles en temps sont utilisés. On montre que l'algorithme est tout à fait insensible au nombre des sous-intervalles en temps utilisé. On observe également le comportement de convergence super-linéaire typique aux algorithmes de relaxation d'onde.

Ensuite nous avons observé comment la convergence dépend elle de la longueur totale de l'intervalle du temps T . Pour cela, nous avons laissé tout les autres paramètres inchangés et nous avons choisi $T = 0.1, 0.2, \dots, 1$ et $\Delta t = \frac{T}{100}$ avec 20 sous intervalles en temps pour toutes les simulations. Nous avons observé que la convergence est beaucoup plus rapide pour des intervalles en temps courts comparé aux longs intervalles en temps.

Dans le but de tester la dépendance en nombre de sous-domaines en espace, nous avons utilisé les mêmes paramètres que précédemment, et on décompose maintenant le domaine en 2, 3, 4, 5 sous-domaines en espace et on observe que l'utilisation de plusieurs sous-domaines en espace fait que l'algorithme converge plus lentement. Mais ceci peut être remédié en utilisant un intervalle de temps global plus petit.

Enfin, nous avons testé la dépendance en la taille du recouvrement en utilisant $2\Delta x, 4\Delta x, 6\Delta x$ pour le recouvrement des sous-domaines en espace. Et nous avons observé que l'augmentation de la taille de recouvrement améliore la vitesse de convergence de l'algorithme même si cela augmente le coût de la méthode.

Une meilleure approche était donc d'utiliser les conditions de transmission optimisées en utilisant la même configuration que dans les expériences précédentes. Et on obtient qu'avec les conditions de transmission de Robin la performance de l'algorithme est meilleure que lorsque l'on augmente la taille

de recouvrement, sans aucun coût supplémentaire. En plus la convergence devient plus linéaire et l'algorithme ne dépend plus d'une manière significative du mécanisme de la convergence super-linéaire essentiel avec les conditions de transmission de Dirichlet.

L'algorithme est plus efficace dans la direction du temps que dans la direction spatiale. Cette performance moindre est due aux conditions de transmission de Dirichlet qui ne sont pas très efficaces. Elle peut être améliorée en utilisant les algorithmes de Schwarz relaxation d'onde optimisés avec des conditions de transmission du type Robin.

8 Conclusion

A la fin de ce rapport et au terme de ces six mois de stage, il est alors important de rappeler les objectifs que nous avons atteint et qui sont en particulier les résultats que nous avons illustré à la fin du rapport. Nous avons présenté un algorithme pararéel Schwarz relaxation d'onde général, qui est basé sur une décomposition en espace et en temps d'un problème d'évolution donné, dans le but d'augmenter le parallélisme. Nous avons exposé un résultat théorique de convergence, et on a illustré ensuite la dépendance de l'algorithme en la configuration de la décomposition spatio-temporelle, qui a révélé que pour une convergence rapide, soit un intervalle de temps courts, soit un grand chevauchement, ou des conditions de transmission optimisées doivent être utilisées. Nous avons également illustré les résultats sur le gain en coût en jouant sur les itérations des deux algorithmes couplés.

Nous avons également entamé des travaux sur est une interface Python vers un code de structure 3D Zebulon (Elasticité Dynamique) pour pouvoir implémenter le l'algorithme du couplage de Pararéel et de Schwarz relaxation d'onde dans le cas 2D. ZéBuLoN est un code de calcul éléments finis adapté aux problèmes de mécanique non linéaire, développé conjointement par l'ONERA, Northwest Numerics (Seattle, USA), et le Centre des Matériaux. Programmé en C++, ZéBuLoN présente une structure modulaire orientée objet. Mais ces travaux de recherche reste en cours.

9 Annexes

9.1 Bibliographie

[1] MARTIN J. GANDER AND STEFAN VANDEVALLE, Parallélisme en temps : Can parts of the solution later in time be computed before the solution earlier in time is known?, Université de Genève.

[2] JUN LIU, YAO-LIN JIANG, A parareal waveform relaxation algorithm for semi-linear parabolic partial differential equations, Journal of Computational and Applied Mathematics.

[3] MARTIN J. GANDER AND CHRISTIAN ROHDE, Overlapping Schwarz waveform relaxation for convection-dominated nonlinear conservation laws, SIAM J. COMPUT.

[4] MARTIN J. GANDER, YAO-LIN JIANG, RONG-JIAN LI, Parareal Schwarz Waveform Relaxation Methods, University of Geneva.

[5] JACQUES-LOUIS, YVON MADAY, GABRIEL TURINICI, Résolution d'EDP par un schéma en temps « pararéel », C.R. Acad.Sci. Paris, t. 332,Série I,p.661-668,2001

[6] M. J. GANDER, Optimized Schwarz methods, SIAM J. Numer. Anal., 44 (2006),pp. 699-731.

9.2 Codes