

---

# Méthode de décomposition de domaines pour l'équation de Schrödinger

## THÈSE

présenté et soutenue publiquement le 28 Novembre 2014

pour l'obtention du

**Doctorat de l'université Lille 1**

**Spécialité: Mathématiques Appliquées**

par

**Feng XING**

### Composition du jury

Christophe Besse	Directeur de thèse	Université Toulouse 3
Emmanuel Creusé	Directeur de thèse	Université Lille 1
Claire Chainais-Hillairet	Examineur	Université Lille 1
Stéphane Descombes	Rapporteur	Université Nice Sophia Antipolis
Laurent Di Menza	Rapporteur	Université de Reims
Michel Kern	Examineur	Maison de la Simulation et INRIA

---

Maison de la Simulation



Laboratoire Paul Painlevé



# Résumé

Ce travail de thèse porte sur le développement et la mise en œuvre de méthodes de décomposition de domaines pour les équations de Schrödinger linéaires ou non-linéaires en une ou deux dimensions d'espace.

Dans la première partie, nous nous intéressons à la méthode de relaxation d'ondes de Schwarz (SWR) pour l'équation de Schrödinger en une dimension. Dans le cas où le potentiel est linéaire et indépendant du temps, nous proposons un nouvel algorithme qui est scalable et permet une forte réduction du temps de calcul comparativement à l'algorithme classique. Pour un potentiel linéaire dépendant du temps ou un potentiel non-linéaire, nous utilisons un opérateur linéaire préalablement défini pour le potentiel nul comme un préconditionneur. Cela permet d'assurer une forte scalabilité. Nous généralisons également les travaux de Halpern et Szeftel sur la condition de transmission en utilisant des conditions absorbantes construites récemment par Antoine, Besse et Klein. Par ailleurs, nous portons les codes développés sur Cpu sur des accélérateur Gpu (carte graphique) pour l'équation de Schrödinger en une dimension.

La deuxième partie concerne la méthode SWR et la méthode de décomposition d'espace pour l'équation de Schrödinger en deux dimensions. Nous généralisons le nouvel algorithme et l'algorithme avec préconditionneur proposés dans la première partie au cas de la dimension deux. Par ailleurs, dans le chapitre 6, nous généralisons les travaux de Loisel sur la méthode de Schwarz optimisée avec points de croisement pour l'équation de Laplace, qui conduit à la méthode SWR avec points de croisement.

Dans la dernière partie, nous appliquons les méthodes de décomposition de domaines que nous avons étudiées à la simulation de condensat de Bose-Einstein qui permettent de diminuer le temps de calcul total, mais aussi de réaliser des simulations impossibles sur un seul nœud de calcul.

**Mots-clés:** Équation de Schrödinger, Méthode de décomposition de domaines, Méthode de Relaxation d'onde de Schwarz optimisée (SWR), Méthode de décomposition en espace, Méthode SWR avec points de croisement, Simulation de condensat de Bose-Einstein, accélération GPU.

# Abstract

This thesis focuses on the development and the implementation of domain decomposition methods for the linear and non-linear, one dimensional and two dimensional Schrödinger equations.

In the first part, we focus on the Schwarz waveform relaxation method (SWR) for the one dimensional Schrödinger equation. In the case the potential is linear and time-independent, we propose a new algorithm that is scalable and allows a significant reduction of computation time compared with the classical algorithm. For a time-dependent linear potential or a non-linear potential, we use a previously defined linear operator as preconditioner. This ensures high scalability. We also generalize the work of Halpern and Szeftel on transmission condition. We use the absorbing boundary conditions recently constructed by Antoine, Besse and Klein as the transmission condition. We also carry the codes developed on Cpu on Gpus accelerator (graphics card) for the one dimensional Schrödinger equation.

The second part concerns the SWR method and the domain decomposition in space method for the Schrödinger equation in two dimensions. We generalize the new algorithm and the preconditioned algorithm proposed in the first part to the case of two dimensions. Furthermore, in Chapter 6, we generalize the work of Loisel on the optimized Schwarz method with cross points for the Laplace equation, which leads to the SWR method with cross points.

In the last part, we apply the domain decomposition methods presented in the previous chapters to the simulation of Bose-Einstein condensate, that could not only reduce the total computation time, but also realise the simulations which are not possible on a single calculation node.

**Key words:** Schrödinger equation, Domain decomposition method, Optimized Schwarz waveform relaxation method (SWR), Domain decomposition in space method, SWR method with cross points, Simulation of Bose-Einstein condensate, Gpu acceleration.

# Remerciements

Je voudrais d'abord sincèrement remercier mon directeur de thèse Christophe pour son aide pendant les trois ans passées. Je voudrais remercier Emmanuel Creusé d'avoir bien voulu jouer le rôle de directeur de thèse officiel durant ma dernière année de thèse. Je lui suis extrêmement reconnaissant de son aide. Je remercie également Pierre pour son soutien sur le calcul parallèle, en particulier pour ses conseils sur la programmation des Gpus. Je remercie Edouard qui m'a donné l'opportunité de travailler dans le laboratoire Maison de la Simulation pendant ma thèse et m'a aidé à surmonter quelques difficultés administratives.

Merci à Michel pour ses conseils scientifiques. Merci à Julien pour sa patience quant à la résolution des problèmes sur le calculateur. Merci aussi à Valérie pour m'avoir aidé sur les procédures administratives. Je voudrais également remercier tous les membres du laboratoire Maison de la Simulation et de l'équipe ANEDP du laboratoire Paul Painlevé.

Stéphane Descombes et Laurent Di Menza m'ont fait l'honneur d'accepter de rapporter mes travaux. Je voudrais les remercier grandement. Merci aussi à Claire Chainais-Hillairet et Michel Kern d'avoir accepté de faire une partie du jury.

Merci à Jeaniffer, Sophie et Zifan, avec qui nous avons partagé les trois ans agréables et inoubliables ensembles. Merci aussi à tous mes amis chinois: Chuan Xu, Cihui Pan, Da Huo, Hao Liu, Hongshu Gao, Jianwei Pu, Langshi Chen, Lu Ding, Qiao Lu, Xiaodong Wang, Weihua He, Yiting Cui etc..

Je voudrais surtout remercier mes parents qui m'ont beaucoup aidé et soutenu aussi bien dans mes études que pour ma vie à l'étranger.

# Table des Matières

<b>Table des Matières</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Méthodes de décomposition de domaines . . . . .	2
1.1.1 Méthode de Schwarz classique . . . . .	2
1.1.2 Méthode de Schwarz optimisée . . . . .	3
1.2 Méthodes de décomposition de domaines pour des équations dépendantes du temps . . . . .	5
1.2.1 Méthode SWR . . . . .	5
1.2.2 Méthode de décomposition en espace . . . . .	7
1.2.3 Méthode de décomposition en temps . . . . .	8
1.3 Travaux existants pour l'équation de Schrödinger . . . . .	8
1.4 Conditions absorbantes pour l'équation de Schrödinger . . . . .	10
1.5 Structure de la thèse . . . . .	13
<b>I Méthode de relaxation d'onde de Schwarz pour l'équation de Schrödinger en une dimension</b>	<b>15</b>
<b>2 Méthode SWR pour l'équation de Schrödinger en une dimension avec un potentiel indépendant du temps</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.1.1 Méthode SWR . . . . .	17
2.1.2 Condition de transmission . . . . .	19
2.2 Discrétisation . . . . .	20
2.3 Problème d'interface . . . . .	26
2.4 Nouvel algorithme SWR . . . . .	32
2.4.1 Construction de la matrice $\mathcal{L}$ et du vecteur $d$ . . . . .	33
2.4.2 Stockage de la matrice $\mathcal{L}$ et du vecteur $d$ . . . . .	35
2.5 Convergence . . . . .	36
2.6 Résultats Numériques . . . . .	38
2.6.1 Comparaison des algorithmes classiques et nouveaux . . . . .	41
2.6.1.1 Cas du potentiel $V = x^2$ . . . . .	41
2.6.1.2 Cas du potentiel $V = -x^2$ . . . . .	43
2.6.2 Comparaison des conditions de transmission . . . . .	44
2.6.2.1 $N = 2, V = x^2$ . . . . .	44

2.6.2.2	$N = 2, V = -x^2$	45
2.6.2.3	$N = 500, V = x^2$	46
2.6.2.4	$N = 500, V = -x^2$	47
2.6.3	Effet du maillage	48
2.6.4	Accélération Gpu	50
2.7	Conclusion	51
<b>3</b>	<b>Méthode SWR pour l'équation de Schrödinger en une dimension avec un potentiel général</b>	<b>53</b>
3.1	Introduction	53
3.2	Discrétisation	54
3.3	Algorithme avec préconditionneur	58
3.4	Résultats numériques	60
3.4.1	Comparaison des algorithmes sans et avec préconditionneur	61
3.4.1.1	Cas du potentiel $\mathcal{V} = 5tx$	62
3.4.1.2	Cas du potentiel $\mathcal{V} =  u ^2$	64
3.4.2	Conditions de transmission	65
3.4.2.1	Cas du potentiel $\mathcal{V} = 5tx$	65
3.4.2.2	Cas du potentiel $\mathcal{V} =  u ^2$	66
3.4.2.3	Conditions de transmission générales	67
3.4.3	Effet du maillage	68
3.4.4	Accélération de Krylov pour l'équation de Schrödinger linéaire	69
3.4.5	Accélération Gpu	70
3.5	Conclusion	72
<b>II</b>	<b>Méthode de décomposition de domaines pour l'équation de Schrödinger en deux dimensions</b>	<b>73</b>
<b>4</b>	<b>Méthode de décomposition de domaines pour l'équation de Schrödinger sans potentiel en deux dimensions</b>	<b>75</b>
4.1	Introduction	75
4.2	Discrétisation	78
4.3	Problème d'interface	84
4.4	Nouveaux algorithmes	89
4.4.1	Nouvel algorithme SWR	89
4.4.2	Nouvel algorithme de décomposition en espace	91
4.5	Résultats numériques	92
4.5.1	La méthode SWR vs. la méthode décomposition en espace	93
4.5.2	Comparaisons des algorithmes classiques et nouveaux	94
4.5.3	Influences de paramètres	96
4.5.4	Un autre cas test	99
4.6	Conclusion	100
<b>5</b>	<b>Méthode de décomposition de domaines pour l'équation de Schrödinger avec potentiel général en deux dimensions</b>	<b>101</b>

5.1	Introduction . . . . .	101
5.2	Discrétisation . . . . .	103
5.3	Algorithme avec préconditionneur . . . . .	104
5.4	Résultats numériques . . . . .	106
5.4.1	Comparaison des algorithmes sans et avec préconditionneur . . . . .	106
5.4.1.1	Cas du potentiel $\mathcal{V} = x^2 + y^2$ . . . . .	107
5.4.1.2	Cas du potentiel $\mathcal{V} =  u ^2$ . . . . .	109
5.4.2	Comparaison des conditions de transmission . . . . .	110
5.4.2.1	Cas du potentiel $\mathcal{V} = x^2 + y^2$ . . . . .	111
5.4.2.2	Cas du potentiel $\mathcal{V} =  u ^2$ . . . . .	112
5.4.3	Influence des paramètres . . . . .	113
5.4.3.1	Cas du potentiel $\mathcal{V} = x^2 + y^2$ . . . . .	114
5.4.3.2	Cas du potentiel $\mathcal{V} =  u ^2$ . . . . .	115
5.4.4	Accélération de Krylov pour l'équation de Schrödinger linéaire . . . . .	117
5.5	Conclusion . . . . .	118
<b>6</b>	<b>Méthode SWR pour l'équation de Schrödinger avec point de croisement</b>	<b>121</b>
6.1	Introduction . . . . .	121
6.2	Méthode SWR avec points de croisement . . . . .	122
6.3	Lien avec la méthode SWR sans points de croisement . . . . .	127
6.4	Résultats numériques . . . . .	129
6.4.1	Comparaison des conditions de transmission et des méthode Krylov . . . . .	130
6.4.2	Influence des paramètres . . . . .	132
6.4.3	Plus d'un point de croisement . . . . .	133
6.5	Conclusion . . . . .	134
<b>III</b>	<b>Simulation d'un condensat de Bose-Einstein</b>	<b>135</b>
<b>7</b>	<b>Simulation de condensat de Bose-Einstein</b>	<b>137</b>
7.1	Introduction . . . . .	137
7.2	GPE sous coordonnées transformées de Lagrange . . . . .	138
7.3	Méthode de décomposition de domaines . . . . .	139
7.4	Résultats numériques . . . . .	140
7.4.1	Comparaison des méthodes . . . . .	140
7.4.2	Dynamique des réseaux de vortex quantifiés . . . . .	142
7.5	Conclusion . . . . .	145
	<b>Conclusion et perspective</b>	<b>146</b>
<b>A</b>	<b>Forme discrète du problème, une dimension</b>	<b>151</b>
A.1	Condition de transmission Robin . . . . .	151
A.2	Condition de transmission $S_0^2$ . . . . .	153
A.3	Condition de transmission $S_0^3$ . . . . .	154
A.4	Condition de transmission $S_0^4$ . . . . .	155

---

A.5	Condition de transmission $S_1^2$	156
A.6	Condition de transmission $S_1^4$	157
A.7	Condition de transmission $S_2^2$	158
A.8	Condition de transmission $S_2^4$	160
<b>B</b>	<b>Introduction à la librairie PETSc</b>	<b>163</b>
<b>C</b>	<b>Introduction de GPGPU</b>	<b>165</b>
	<b>Bibliographie</b>	<b>169</b>



# Chapitre 1

## Introduction

Nous nous intéressons dans cette thèse à la méthode de décomposition de domaines pour l'équation de Schrödinger en une ou deux dimensions

$$\begin{cases} i\partial_t u + \Delta u + \mathcal{V}u = 0, & (t, \mathbf{x}) \in ]0, T[ \times \Omega, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \end{cases} \quad (1.1)$$

où  $\Omega$  est un sous ensemble ouvert de  $\mathbb{R}$  ou  $\mathbb{R}^2$ , la donnée initiale  $u_0 \in L^2(\Omega)$ , et  $\mathcal{V}$  est un potentiel réel linéaire ou non-linéaire

- cadre linéaire :  $\mathcal{V} = V(t, \mathbf{x})$ ,
- cadre non-linéaire :  $\mathcal{V} = V(t, \mathbf{x}) + f(u)$ , ex:  $f(u) = |u|^2$ .

Les conditions aux limites de (1.1) seront précisées dans chaque chapitre. L'équation de Schrödinger est une équation fondamentale en mécanique quantique. Elle décrit l'évolution dans le temps d'une particule massive non relativiste, et remplit ainsi le même rôle que la relation fondamentale de la dynamique en mécanique classique. La résolution numériques de l'équation de Schrödinger est un enjeu et que les mathématiciens utilisent notamment pour cela des méthodes de décomposition de domaines. En mathématiques, la méthode de décomposition de domaines est une méthode théorique et numérique pour résoudre de manière itérative par exemple des équation aux dérivées partielles. Elle est basée sur la décomposition de domaines physiques.

Un des raisons d'utiliser cette méthode pour le calcul de la solution numérique de l'équation de Schrödinger est que la résolution sur le domaine entier  $\Omega$  peut être rendue difficile si celui-ci est grand. Ainsi, sa taille pourrait faire que

- la capacité de la mémoire de l'ordinateur ne soit pas suffisante,
- le temps de calcul soit extrêmement long.

En outre, la méthode de décomposition de domaines permet de créer des algorithmes parallèles qui s'adaptent à l'architecture des grands calculateurs modernes.

## 1.1 Méthodes de décomposition de domaines

Nous présentons ici deux méthodes de décomposition de domaines pour des équation stationnaires à savoir la méthode de Schwarz classique et la méthode de Schwarz optimisée. On se réfère aux articles suivants [30, 31, 33].

### 1.1.1 Méthode de Schwarz classique

La première méthode de décomposition de domaines a été proposée par H.A. Schwarz en 1870 pour étudier l'existence théorique de l'équation de Poisson  $\Delta u = 0$  sur un domaine composé d'un disque et d'un rectangle (voir figure 1.1a). Une condition de Dirichlet est utilisée sur l'interface  $\Gamma_1$  et  $\Gamma_2$  comme condition de transmission, où  $\Gamma_1 = \partial\Omega_2 \cap \Omega_1$ ,  $\Gamma_2 = \partial\Omega_1 \cap \Omega_2$ . Cette méthode est nommée par "Alternating Schwarz method" et donnée par

$$\begin{cases} -\Delta u_1^{k+1} = 0, (x, y) \in \Omega_1, \\ u_1^{k+1} = u_2^k, (x, y) \in \Gamma_1, \end{cases} \quad \begin{cases} -\Delta u_2^{k+2} = 0, (x, y) \in \Omega_2, \\ u_2^{k+2} = u_1^{k+1}, (x, y) \in \Gamma_2. \end{cases}$$

L'exposant  $k$  désigne le numéro de l'itération dans le processus itératif de Schwarz tandis que l'indice 1 ou 2 fait ici référence au sous-domaine considéré. On remarquera ici que cette méthode est séquentielle. En effet, elle nécessite de résoudre d'abord l'équation sur le domaine  $\Omega_1$  pour avoir la connaissance de  $u_1^{k+1}$  sur  $\Gamma_2$ , ce qui donne la possibilité de résoudre l'équation de Poisson sur  $\Omega_2$ .

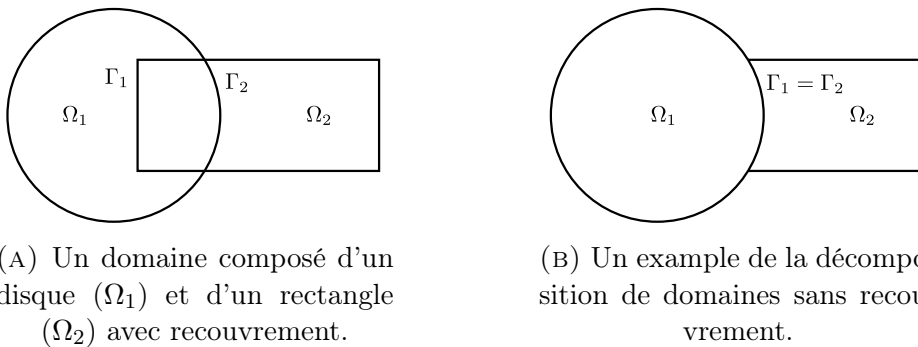


FIGURE 1.1: Décomposition de domaine.

En raison du développement des calculateurs parallèles, la méthode de décomposition de domaines a retrouvé un grand intérêt lors des dernières décennies. P.L. Lions [52, 53] a par exemple introduit la méthode "Parallel Schwarz method", qui est donnée par

$$\begin{cases} -\Delta u_1^{k+1} = 0, (x, y) \in \Omega_1, \\ u_1^{k+1} = u_2^k, (x, y) \in \Gamma_1, \end{cases} \quad \begin{cases} -\Delta u_2^{k+1} = 0, (x, y) \in \Omega_2, \\ u_2^{k+1} = u_1^k, (x, y) \in \Gamma_2. \end{cases}$$

La différence notable par rapport à la méthode originelle de Schwarz se situe sur la condition de transmission pour le calcul de  $u_2$  où il n'est plus nécessaire de connaître  $u_1$  à l'itération  $k + 1$  pour résoudre le problème. Cette méthode donne donc une possibilité de faire des algorithmes parallèles.

### 1.1.2 Méthode de Schwarz optimisée

La méthode de Schwarz optimisée concerne une équation du type

$$\mathcal{L}_x u(x, y) = f, \quad (x, y) \in \Omega,$$

où  $\mathcal{L}_x$  est l'opérateur différentiel, ex:  $\mathcal{L}_x = -\Delta$  pour l'équation de Poisson.

Elle est donnée par

$$\begin{cases} \mathcal{L}_x u_1^{k+1} = f, & (x, y) \in \Omega_1, \\ B_1 u_1^{k+1} = B_1 u_2^k, & (x, y) \in \Gamma_1, \end{cases} \quad \begin{cases} \mathcal{L}_x u_2^{k+1} = f, & (x, y) \in \Omega_2, \\ B_2 u_2^{k+1} = B_2 u_1^k, & (x, y) \in \Gamma_2. \end{cases}$$

La différence par rapport à la méthode de Schwarz classique provient des conditions de transmission entre sous domaines. La condition de transmission  $B_j, j = 1, 2$  est une condition de type Dirichlet-Neumann

$$B_j = \partial_{\mathbf{n}_j} + S_j, \quad j = 1, 2,$$

où  $\partial_{\mathbf{n}_j}, j = 1, 2$  est la dérivée normale à l'interface entre les sous-domaines et  $S_j, j = 1, 2$  est l'opérateur de transmission.

P.L. Lions propose dans [53] d'utiliser une condition de Robin pour l'équation de Poisson avec

$$S_j = p, \quad p \in \mathbb{R}^+.$$

Avec cette condition de transmission, la méthode converge pour une décomposition de domaines avec et sans recouvrement (voir figures 1.1a et 1.1b).

La méthode de Schwarz optimisée et les conditions de transmission ont beaucoup été étudiées. Nous donnons ci-dessous quelques exemples. Nous ne précisons cependant pas ici géométries utilisées des articles cités, car elle change régulièrement. La figure 1.2 montre un exemple de décomposition de domaines avec recouvrement régulièrement.

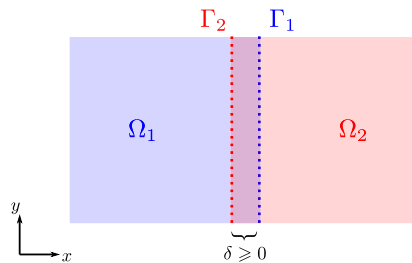


FIGURE 1.2: La décomposition de domaine, 2D.

Le problème de convection-diffusion [50, 56, 60] :

$$\mathcal{L}_x = c + a(x, y)\partial_x + b(x, y)\partial_y - \nu\Delta,$$

où  $c$  est une constante,  $(a, b)$  est le champ de vitesse et  $\nu$  est la viscosité. Dans le cas où  $a$  et  $b$  sont constants,

$$\mathcal{L}_x = -\nu(\partial_x - Op(\sigma^+))(\partial_x - Op(\sigma^-)),$$

où

$$\sigma^\pm = \frac{a \pm \sqrt{a^2 + 4\nu c - 4i\nu b\xi + 4\xi^2\nu^2}}{2\nu}.$$

Ici,  $Op(\sigma)$  est l'opérateur pseudo-différentiel dont le symbole  $\sigma$ , défini par

$$Op(\sigma)u = \int_{\mathbb{R}} \sigma(x, y, \xi) \widehat{u}(x, \xi) e^{iy\xi} d\xi,$$

où  $\widehat{u}(x, \xi)$  est la transformée de Fourier de  $u(x, y)$  en  $y$

$$\widehat{u}(x, \xi) = \frac{1}{2\pi} \int_{\mathbb{R}} u(x, y) e^{-iy\xi} dy.$$

La condition optimale est basée sur la factorisation d'opérateur

$$S_1 = -Op(\sigma^-), \quad S_2 = Op(\sigma^+).$$

Les opérateurs ci-dessus n'étant pas locaux, les auteurs [50, 56, 60] proposent d'utiliser des approximations locales de la condition optimale. Le développement de Taylor du symbole est utilisé dans [56]. Dans [50],

$$S_j = -c_1^j + c_2^j \frac{\partial}{\partial y} - c_3^j \frac{\partial^2}{\partial y^2}, \quad i = 1, 2,$$

où les coefficients  $c_1^j$ ,  $c_2^j$  et  $c_3^j$  sont calculés de manière à optimiser le taux de convergence de la méthode Schwarz optimisée.

Dans le cas où l'opérateur  $\mathcal{L}_x = \eta - \Delta$ ,  $\eta > 0$  Gander [30] analyse la méthode de Schwarz optimisée. La condition optimale est

$$S_1 = Op(\sigma_1), \quad S_2 = Op(\sigma_2),$$

et

$$\sigma_1 = \sigma_2 = \sqrt{\xi^2 + \eta}.$$

La condition optimale permet à la méthode de converger en deux itérations pour deux sous domaines. Deux types de condition optimisée sont proposés. Une est basée sur approximation polynomiale du symbole

$$\text{Ordre 0 (Robin): } S_1 = Op(p_1) = p_1, \quad S_2 = Op(p_2) = p_2,$$

$$\text{Ordre 2: } S_1 = Op(p_1 + q_1\xi^2) = p_1 - q_1 \frac{\partial^2}{\partial y^2},$$

$$S_2 = Op(p_2 + q_2\xi^2) = p_2 - q_2 \frac{\partial^2}{\partial y^2}$$

où  $p_1$ ,  $p_2$ ,  $q_1$  et  $q_2$  sont des paramètres à optimiser pour assure un bon taux de convergence. Une autre est basée sur le développement de Taylor du symbole

$$\begin{aligned} \text{Ordre 0: } S_1 = S_2 &= Op(\sqrt{\eta}) = \sqrt{\eta}, \\ \text{Ordre 2: } S_1 = S_2 &= Op\left(\sqrt{\eta} + \frac{\xi^2}{2\sqrt{\eta}}\right) = \sqrt{\eta} - \frac{1}{2\sqrt{\eta}} \cdot \frac{\partial^2}{\partial y^2}. \end{aligned}$$

Des travaux ont également été entrepris pour l'équation de Helmholtz [22, 26, 39] où

$$\mathcal{L}_x = \Delta + k^2.$$

La méthode Schwarz optimisée pour l'équation de Helmholtz est introduite dans [26] en utilisant la condition de type Robin. Dans [22], les auteurs propose une condition quasi-optimale. C'est une approximation de Padé où l'on approcher la racine carrée par  $m$  termes:

$$\sqrt{z} \approx \sum_{s=0}^m a_s^m - \sum_{s=1}^m \frac{a_s^m d_s^m}{z + d_s^m}$$

avec  $a_0^m = 0$ ,  $\theta$  est une constante, et pour  $1 \leq s \leq m$ :

$$a_s^m = \frac{e^{i\theta/2}}{m \cos^2\left(\frac{(2s-1)\pi}{4m}\right)}, \quad d_s^m = e^{i\theta} \tan^2\left(\frac{(2s-1)\pi}{4m}\right).$$

L'équation de Maxwell a été étudiée dans [27, 61].

## 1.2 Méthodes de décomposition de domaines pour des équations dépendantes du temps

Pour des équations dépendantes du temps, il existe trois familles de méthodes de décomposition de domaines. Une bonne base bibliographique est la thèse de Hoang [46].

### 1.2.1 Méthode SWR

La méthode de relaxation d'onde de Schwarz (Schwarz waveform relaxation method, SWR) est une classe de méthodes de décomposition de domaines pour des équations aux dérivées partielles dépendantes du temps, basée sur une décomposition en temps-espace. Le domaine de temps-espace  $]0, T[ \times \Omega$  est décomposé en divers sous domaines avec ou sans recouvrement (par exemple pour deux sous domaines, voir figure 1.3 et 1.4).  $\Gamma_1$  et  $\Gamma_2$  représente les interfaces entre  $\Omega_1$  et  $\Omega_2$ . Pour la décomposition sans recouvrement, on a  $\Gamma_1 = \Gamma_2$ .

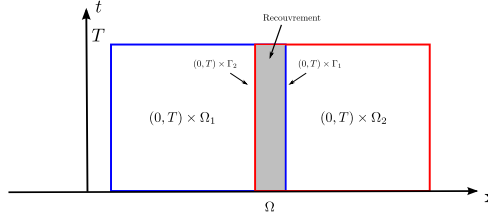


FIGURE 1.3: Décomposition de domaines temps-espace avec recouvrement.

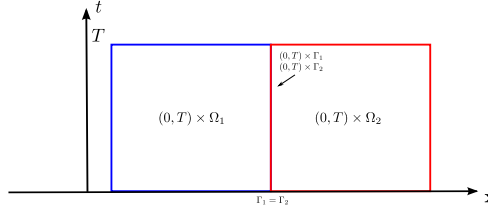


FIGURE 1.4: Décomposition de domaines temps-espace sans recouvrement.

L'algorithme pour deux sous domaines est donnée par

$$\begin{cases} \mathcal{L}u_1^{k+1} = h, (t, \mathbf{x}) \in ]0, T[ \times \Omega_1, \\ u_1^{k+1}(0, \mathbf{x}) = u_0(\mathbf{x}), \mathbf{x} \in \Omega_1, \\ B_1 u_1^{k+1} = B_1 u_2^k, \mathbf{x} \in \Gamma_1, \end{cases} \quad (1.2)$$

$$\begin{cases} \mathcal{L}u_2^{k+1} = h, (t, \mathbf{x}) \in ]0, T[ \times \Omega_2, \\ u_2^{k+1}(0, \mathbf{x}) = u_0(\mathbf{x}), \mathbf{x} \in \Omega_2, \\ B_2 u_2^{k+1} = B_2 u_1^k, \mathbf{x} \in \Gamma_2, \end{cases}$$

où  $\mathcal{L}$  est l'opérateur aux dérivées partielles et  $h$  est une fonction. Par exemple, nous pouvons considérer

- l'équation de Schrödinger:  $\mathcal{L}u = (i\partial_t + \Delta + \mathcal{V})u$ ,
- l'équation de la chaleur:  $\mathcal{L}u = (\partial_t - \Delta)u$ ,
- l'équation de convection-diffusion:  $\mathcal{L}u = \partial_t u + a(x, y)\partial_x u + b(x, y)\partial_y u - \nu\Delta u$ ,
- l'équation de réaction-diffusion semi-linéaire:  $\mathcal{L}u = \partial_t u - \Delta u + f(u)$ ,

Le terme central ici est la condition de transmission  $B_j, j = 1, 2$ . Elle joue un rôle très important pour la méthode SWR.

La méthode SWR est formulée dans [29] et [41] indépendamment pour des équations paraboliques et hyperboliques. La condition de Dirichlet

$$B_1 = B_2 = I, \text{ où } I \text{ est opérateur d'identité}$$

est utilisée sur l'interface entre les sous domaines avec recouvrement. Avec la condition Dirichlet, la méthode SWR ne fonctionne que pour la décomposition avec recouvrement.

La méthode de relaxation d'onde optimisée (optimized Schwarz waveform relaxation, noté également par SWR dans cette thèse) est introduite dans [35] pour des équations paraboliques et hyperboliques en utilisant la condition de transmission du type Dirichlet-Neumann

$$B_1 = \partial_{\mathbf{n}_1} + S_1, \quad B_2 = \partial_{\mathbf{n}_2} + S_2.$$

où  $\partial_{\mathbf{n}_j}, j = 1, 2$  est la dérivée normale,  $S_j, j = 1, 2$  est l'opérateur de transmission. La méthode SWR converge pour la décomposition avec et sans recouvrement. Pour la condition optimale,  $S_j, j = 1, 2$  sont des opérateurs qui viennent de la factorisation de  $\mathcal{L}$ . Cependant, la condition optimale n'est pas toujours disponible. Elle est en outre souvent coûteuse à implémenter (non locale en général). Ainsi, des approximations sont considérées.

Pour le problème de convection-diffusion, la méthode SWR est appliquée et analysée dans [55]. Sur l'interface, les conditions Robin et Ventcell sont utilisées

$$\begin{aligned} \text{Robin: } \quad S_1 &= -\frac{a-p}{2\nu}, \quad S_2 = \frac{a+p}{2\nu}, \\ \text{Ventcell: } \quad S_1 &= -\frac{a-p}{2\nu} - q\frac{\partial}{\partial t} - b\frac{\partial}{\partial y}, \quad S_2 = \frac{a+p}{2\nu} + q\frac{\partial}{\partial t} + b\frac{\partial}{\partial y}, \end{aligned}$$

où  $p$  et  $q$  sont deux paramètres à optimiser pour avoir un bon taux de convergence. Les paramètres de la condition de Robin et Ventcell sont optimisés dans [19, 32, 55]. La méthode SWR est étendue au problème avec coefficients discontinus [21, 34], au problème hétérogène [42] et au problème de la formulation mixte [46, 47]. Dans [23], SWR est utilisée pour le problème semi-linéaire avec les conditions Robin, Ventcell et une condition de transmission non-linéaire.

La méthode SWR est introduite pour l'équation des ondes en une dimension, linéaire et semi-linéaire dans [36, 44].

## 1.2.2 Méthode de décomposition en espace

Les équations sont d'abord discrétisées en temps sur le domaine entier avec des méthodes implicites. Ensuite, on applique les méthodes de décomposition de domaines à ce problème stationnaire. Cette méthode est utilisée pour des équations hyperboliques dans [12, 66] et pour des équations paraboliques dans [24, 25]. On généralise dans cette thèse la méthode de décomposition en espace à l'équation de Schrödinger linéaire en deux dimensions  $\mathcal{V} = V(t, x, y)$  (voir également le chapitre 4). L'intervalle du temps  $]0, T[$  est discrétisé uniformément en  $N_T$  intervalles de longueur  $\Delta t = T/N_T$ . La semi-discrétisation de Crank-Nicolson donne que pour  $1 \leq n \leq N_T$ ,

$$i\frac{u_n - u_{n-1}}{\Delta t} + \Delta\frac{u_n + u_{n-1}}{2} + V(t_n, x, y)\frac{u_n + u_{n-1}}{2} = 0, \quad (1.3)$$

où  $u_n(x, y) = u(t_n, x, y)$  et  $t_n = n\Delta t$ . En introduisant la nouvelle variable  $v_n = \frac{u_n + u_{n-1}}{2}$  avec  $v_0 = u_0$ , la formule (1.3) s'écrit comme

$$\left(\frac{2i}{\Delta t} + V(t_n, x, y)\right)v_n + \Delta v_n = 2i\frac{u_{n-1}}{\Delta t}.$$

C'est une équation stationnaire de type

$$\mathcal{L}_x u = f, \quad (1.4)$$

où  $u$  est l'inconnue et  $f$  est une fonction donnée. Pour (1.3), on a

$$\mathcal{L}_x = \frac{2i}{\Delta t} + V(t_n, x, y) + \Delta.$$

Les méthodes de décomposition de domaines sont appliqués au problème stationnaire (1.4). Dans cette thèse, on utilise la méthode SWR qui est présentée dans le chapitre 1.1.

**Remarque:** La méthode de décomposition en espace a deux inconvénients par rapport à la méthode SWR. La première est qu'elle impose une discrétisation en temps uniforme. La seconde est qu'elle nécessite de communiquer des données de taille petite un grand nombre de fois. La communication génère un temps de latence qui s'ajoute au temps de calcul. Cependant, elle possède un avantage important. Comme elle traite un problème qui vient d'une seule étape du temps, à l'opposé de la méthode SWR qui elle considère un problème sur tous les étapes du temps, le nombre d'itérations pour la méthode de décomposition en espace peut être inférieur à celui de la méthode SWR.

### 1.2.3 Méthode de décomposition en temps

L'équation est d'abord discrétisée en espace. Le problème se ramène alors à la méthode de décomposition de domaines pour l'équation ordinaire. Dans cette thèse, nous ne considérons pas la méthode de décomposition en temps.

## 1.3 Travaux existants pour l'équation de Schrödinger

La méthode SWR est introduite et analysée pour l'équation de Schrödinger linéaire en une dimension par Halpern et Szeftel dans [43, 45]. Ils proposent d'utiliser quatre conditions de transmission.

- Condition de Dirichlet.  $B_1 = B_2 = I$ , où  $I$  est opérateur d'identité. Cette condition ne fonctionne que pour la décomposition avec recouvrement.
- Condition optimale. Si le potentiel  $\mathcal{V}$  est constant, la condition optimale est disponible. Elle est donnée par les conditions de transmission

$$B_1 = \partial_{\mathbf{n}_1} + S_1, \quad B_2 = \partial_{\mathbf{n}_2} + S_2,$$

où  $S$  est un opérateur pseudo-différentiel. Le symbole de  $S_j$  est  $(\tau - V)^{1/2}$  où

$$(\tau - V)^{1/2} = \begin{cases} \sqrt{\tau - V}, & \text{si } \tau \geq V, \\ -i\sqrt{-\tau + V}, & \text{si } \tau < V. \end{cases}$$



Ici, les opérateurs pseudo-différentiel en temps  $P(t, \partial_t, x)$  avec le symbole total  $p(t, \tau, x)$  sont définis par

$$P(t, \partial_t, x)u(t, x) = Op(\sigma)u = \int_{\mathbb{R}} \sigma(t, \tau, x) \widehat{u}(\tau, x) e^{it\tau} d\tau,$$

où  $\sigma(t, \tau, x)$  est le symbole de l'opérateur et  $\widehat{u}(\tau, x)$  est la transformée de Fourier de  $u(t, x)$  en temps

$$\widehat{u}(\tau, x) = \frac{1}{2\pi} \int_{\mathbb{R}} u(t, x) e^{-it\tau} dt.$$

Pour le cas de deux sous domaines avec ou sans recouvrement, la condition optimale permet à la méthode SWR de converger en deux itérations. En revanche, si le potentiel  $V$  est variable, cette condition n'est pas toujours disponible.

- Condition quasi-optimale. Si le potentiel  $\mathcal{V} = V(x)$ , alors les auteurs considèrent  $B_1 = \partial_{\mathbf{n}_1} + S_1$ ,  $B_2 = \partial_{\mathbf{n}_2} + S_2$  où

$$S_1 = Op((\tau - V|_{x \in \Gamma_1})^{1/2}), \quad S_2 = Op((\tau - V|_{x \in \Gamma_2})^{1/2}).$$

On remarque ici que  $V|_{x \in \Gamma_1}, V|_{x \in \Gamma_2}$  sont deux constantes réelles.

- Condition de Robin. Ici, on a  $B_1 = \partial_{\mathbf{n}_1} + S_1$ ,  $B_2 = \partial_{\mathbf{n}_2} + S_2$  où

$$S_j = -ip, \quad p \in \mathbb{R}^+.$$

Le paramètre  $p$  est choisi pour minimiser le facteur de convergence.

Récemment, l'article de Antoine, Lorin et Bandrauk [11] considère l'équation de type Schrödinger

$$i\partial_t u(t, \mathbf{x}) = H_{v,l} u(t, \mathbf{x}),$$

avec

$$\begin{aligned} H_v(t) &= -\Delta + V_c(\mathbf{x}) - i\mathbf{A}(t) \cdot \nabla + \|\mathbf{A}(t)\|^2, \\ H_l(t) &= -\Delta + V_c(\mathbf{x}) + \mathbf{x} \cdot \mathbf{E}(t), \end{aligned}$$

où  $(t, \mathbf{x}) \in [0, T] \times \mathbb{R}^d$ ,  $V_c(\mathbf{x})$  est le potentiel d'interaction,  $\mathbf{A}(t)$  est le potentiel électrique et le champ laser  $\mathbf{E}(t)$  est supposé être seulement dépendant du temps. Le domaine complet  $\Omega$  est décomposé en des sous-domaines. Les configurations géométriques sont montrées dans la figure 1.5.

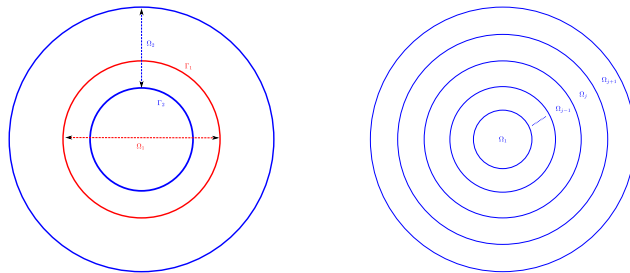


FIGURE 1.5: Configuration géométrique.

Ils appliquent la méthode SWR à ce problème. Sur l'interface, il propose d'utiliser des conditions absorbantes comme condition de transmission, qui est également une idée que nous suivons dans cette thèse. Les conditions de transmission qui sont utilisés par ces auteurs sont un type des conditions construites dans [7] notée par  $S_{2T}$  dans cette thèse (voir la remarque 4.2). Les auteurs montrent également des résultats numériques pour l'équation en une dimension d'espace et que la solution globale peut être retrouvée par cette méthode.

## 1.4 Conditions absorbantes pour l'équation de Schrödinger

Au cours des dernières années, certaines conditions absorbantes pour l'équation de Schrödinger linéaire ou non linéaire avec potentiel dépendant du temps et de l'espace ont été construits en utilisant des techniques du calcul pseudo-différentiel par Antoine, Besse, Klein et Szeftel [1, 3–9]. Dans cette thèse, nous allons utiliser les conditions absorbantes comme condition de transmission dans (1.2).

Une base bibliographique détaillée sur les conditions absorbantes est fournie par la thèse de P. Klein [51]. En  $d$  dimensions, la solution de

$$\begin{cases} i\partial_t u + \Delta u + \mathcal{V}u = 0, & (t, \mathbf{x}) \in \mathbb{R}^+ \times \mathbb{R}^d, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^d, \end{cases} \quad (1.5)$$

est définie sur le domaine non borné  $\{(t, x) \in \mathbb{R}^+ \times \mathbb{R}^d\}$ , où  $u_0$  est supposée à support compact. Pour résoudre le problème numériquement, il est nécessaire de se restreindre à un domaine borné en espace (voir figure 1.6).

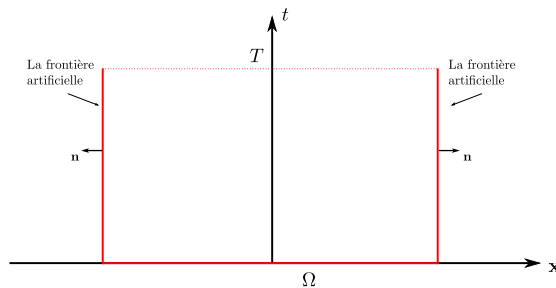


FIGURE 1.6: Le domaine de calcul  $\Omega$ .

Une frontière artificielle est donc créée en  $\partial\Omega$ . Pour traiter ce problème, une condition aux limites artificielle est ajoutée sur cette frontière. Pour  $d = 1$  et  $\mathcal{V} = 0$ , l'équation admet une condition aux limites exacte donnée par

$$\partial_n u + e^{-i\pi/4} \partial_t^{1/2} u = 0, \quad (t, \mathbf{x}) \in ]0, T[ \times \partial\Omega,$$

où l'opérateur de dérivée fractionnaire  $\partial_t^{1/2}$ , avec le symbole  $\sqrt{-i\tau}$ , est défini par

$$\partial_t^{1/2} f(t) = \frac{1}{\sqrt{\pi}} \partial_t \int_0^t \frac{f(s)}{\sqrt{t-s}} ds.$$

La condition aux limites exacte est appelée condition aux limites transparente. Cependant, pour un potentiel général, la condition aux limites transparente n'est pas toujours disponible. Dans les articles [1, 3–9], Antoine, Besse, Klein et Szeftel construisent les conditions absorbantes sur la frontière fictive pour l'équation de Schrödinger linéaire et non-linéaire en une et deux dimensions.

On rappelle ici brièvement la construction des conditions absorbantes pour l'équation de Schrödinger en une dimension. On se réfère à la thèse de P. Klein [51].

**Définition 1.1.** Soit  $\alpha$  un nombre réel et  $\Xi$  un ouvert de  $\mathbb{R}$ , la classe de symboles  $S_S^\alpha(\Xi \times \Xi)$  désigne l'espace vectoriel des fonctions  $C^\infty a(t, \tau, x)$  dans  $\Xi \times \Xi \times \mathbb{R}$  telles que pour tout  $K \in \Xi \times \Xi$  et pour tous les indices  $\beta, \delta, \gamma$ , il existe une constante  $C_{\beta, \delta, \gamma}(K)$  telle que

$$|\partial_\tau^\beta \partial_t^\delta \partial_x^\gamma a(t, \tau, x)| \leq C_{\beta, \delta, \gamma}(K)(1 + |\tau|^{\alpha - \beta}),$$

pour tout  $(t, x) \in K$  et  $\tau \in \mathbb{R}$ .

**Définition 1.2.** Une fonction  $f(t, \tau, x)$  est dite homogène d'ordre  $m$  si et seulement si, pour tout  $\mu > 0$ , on a

$$f(t, \mu\tau, x) = \mu^m f(t, \tau, x).$$

**Définition 1.3.** Un opérateur pseudo-différentiel  $P$  est classique, inhomogènes d'ordre  $M$ ,  $M \in \mathbb{Z}/2$ , si son symbole total, noté  $p$ , admet un développement asymptotique en symboles homogène  $\{p_{M-j/2}\}_{j=0}^{+\infty}$  qui s'écrit

$$p(t, \tau, x) \sim \sum_{j=0}^{+\infty} p_{M-j/2}(t, \tau, x),$$

où chaque fonction  $p_{M-j/2}$  est homogène d'ordre  $2M - j$ , pour  $j \in \mathbb{N}$ . La signification du symbole  $\sim$  est ici la suivante:

$$\forall \tilde{m} \in \mathbb{N}, p - \sum_{j=0}^{\tilde{m}} p_{M-j/2} \in S_S^{M - (\tilde{m} + 1)/2}.$$

L'opérateur est noté  $P \in OPS_S^M$ . La classe  $OPS_S^{-\infty}$  correspond à l'intersection de toute les classes  $OPS_S^M$  pour  $M \in \mathbb{Z}$ .

**Théorème 1.4.** *Il existe deux opérateurs pseudo-différentiels classiques inhomogène  $\Lambda^\pm = \Lambda^\pm(t, \partial_t, x) \in OPS_S^{1/2}$ , réguliers par rapport à la variable d'espace  $x$ , et tels que*

$$i\partial_t + \partial_{xx} + V(t, x) = (\partial_x + i\Lambda^-)(\partial_x + i\Lambda^+) + R, \quad (1.6)$$

où  $R$  est un opérateur régularisant de  $OPS_S^{-\infty}$ . De plus, le symbole total  $\lambda^\pm$ ,  $\Lambda^\pm = Op(\lambda^\pm)$  admet un développement asymptotique en symboles homogènes

$$\lambda^\pm \sim \sum_{j=0}^{+\infty} \lambda_{1/2-j/2}^\pm.$$

D'après le théorème 1.4, en utilisant la propriété  $\partial_x \Lambda^+ = Op(\partial_x \lambda^+) + \Lambda^+ \partial_x$ , on a

$$i\partial_t + \partial_{xx} + V(t, x) = \partial_{xx} + i(\Lambda^- + \Lambda^+) \partial_x + iOp(\partial_x \lambda^+) - \Lambda^- \Lambda^+ + R.$$

Puis omettant le terme régulier  $R$ , on a

$$\begin{cases} i(\Lambda^- + \Lambda^+) = 0, \\ iOp(\partial_x \lambda^+) - \Lambda^- \Lambda^+ = i\partial_t + V. \end{cases}$$

En passant aux symboles, on obtient le système

$$\begin{cases} i(\lambda^- + \lambda^+) = 0, \\ i\partial_x \lambda^+ - \sum_{s=0}^{\infty} \frac{(-i)^s}{s!} \partial_\tau^s \lambda^- \partial_t^s \lambda^+ \sim -\tau + V, \end{cases} \quad (1.7)$$

où la propriété suivante est utilisée

$$\Lambda^- \Lambda^+ = Op\left(\sum_{s=0}^{\infty} \frac{(-i)^s}{s!} \partial_\tau^s \lambda^- \partial_t^s \lambda^+\right).$$

La condition absorbante est alors

$$\partial_x u + iOp\left(\sum_{j=0}^n \lambda_{1/2-j/2}^+\right)u = 0,$$

où  $n = 0, 1, 2, \dots, +\infty$  est l'ordre d'approximation.

Pour la méthode SWR, on va prendre dans (1.2):

$$S_1 = S_2 = iOp\left(\sum_{j=0}^n \lambda_{1/2-j/2}^+\right) = i \sum_{j=0}^n Op\left(\lambda_{1/2-j/2}^+\right).$$

Au total, trois familles d'opérateurs sont proposées.

**Stratégie potentiel.** Le choix du symbole  $\lambda_{1/2}^+$  n'est pas unique. Pour cette stratégie,  $\lambda_{1/2}^+(\tau) = -\sqrt{\tau}$ , et les symboles  $\lambda_{1/2-j/2}^+, j = 1, 2, \dots, n$  sont calculés par la formule (1.7).

**Stratégie d'approximation de Padé.** Le symbole principal est  $\lambda_{1/2}^+(\tau) = -\sqrt{\tau + V}$ . L'approximation de Padé de la racine carrée (voir précédemment) est utilisée.

**Stratégie changement de jauge.** Le changement de jauge est défini par

$$v = e^{-i \int_0^t V(s,x) ds} u.$$

Pour cette stratégie, l'équation de Schrödinger après le changement de jauge est considérée et le symbole principal est  $\lambda_{1/2}^+(\tau) = -\sqrt{\tau}$ .

Nous donnerons en détail les conditions absorbantes dans le chapitre suivant. Elles sont utilisées comme la condition de transmission.

## 1.5 Structure de la thèse

Nous considérons tout d'abord la méthode SWR sans recouvrement pour l'équation de Schrödinger linéaire en une dimension avec un potentiel indépendant du temps,  $\mathcal{V} = V(x)$ , dans le chapitre 2. Il y a deux points clés pour la méthode SWR, le premier est la condition de transmission, et le second est la scalabilité si le domaine entier est décomposé en beaucoup de sous domaines. Dans ce chapitre, on utilise des conditions absorbantes qui sont construites par Antoine, Besse, Klein et Szeftel [5, 9] comme condition de transmission sur l'interface entre les sous domaines. L'algorithme présenté dans (1.2) s'écrit comme

$$\begin{pmatrix} B_1 u_2^0 \\ B_2 u_1^0 \end{pmatrix} \xrightarrow{\text{Solve}} \begin{pmatrix} u_1^1 \\ u_2^1 \end{pmatrix} \rightarrow \begin{pmatrix} B_2 u_1^1 \\ B_1 u_2^1 \end{pmatrix} \xrightarrow{\text{Comm.}} \begin{pmatrix} B_1 u_2^1 \\ B_2 u_1^1 \end{pmatrix} \rightarrow \cdots \rightarrow \begin{pmatrix} B_1 u_2^k \\ B_2 u_1^k \end{pmatrix} \rightarrow \cdots .$$

Il est initialisé par un vecteur initial. À l'étape "Solve", l'équation est résolue sur les sous domaines pour obtenir  $u_1^1$  et  $u_2^1$ . À l'étape "Comm.", les informations sur l'interface sont échangées. On montrera que l'on peut écrire

$$\begin{pmatrix} B_1 u_2^{k+1} \\ B_2 u_1^{k+1} \end{pmatrix} = \mathcal{L} \begin{pmatrix} B_1 u_2^k \\ B_2 u_1^k \end{pmatrix} + d, \quad (1.8)$$

où  $\mathcal{L}$  est une matrice et  $d$  est un vecteur. On appelle cet algorithme l'algorithme classique. Nous expliquerons que les constructions explicites de la matrice  $\mathcal{L}$  et du vecteur  $d$  ne sont pas coûteuses. Grâce à cette remarque, nous proposerons un nouvel algorithme. Alors, (1.8) se transforme en une multiplication matrice-vecteur et addition de vecteurs. Les résultats numériques permettent de montrer que ce nouvel algorithme permet un gain de temps substantiel par rapport à l'algorithme classique et qu'il est scalable jusqu'à 500 sous domaines.

Ensuite, dans le chapitre 3, nous considérons la méthode SWR pour l'équation de Schrödinger en une dimension avec un potentiel linéaire général  $\mathcal{V} = V(t, x)$  ou un potentiel non-linéaire  $\mathcal{V} = f(u)$ . Comme nous l'avons fait dans le chapitre 2, l'algorithme classique s'écrit comme

$$\begin{pmatrix} B_1 u_2^{k+1} \\ B_2 u_1^{k+1} \end{pmatrix} = \mathcal{R} \begin{pmatrix} B_1 u_2^k \\ B_2 u_1^k \end{pmatrix}. \quad (1.9)$$

Pour  $\mathcal{V} = f(u)$ ,  $\mathcal{R}$  est un opérateur non-linéaire. Pour  $\mathcal{V} = V(t, x)$ ,  $\mathcal{R}$  est un opérateur linéaire, mais il est très coûteux à construire. On utilise alors l'idée du chapitre 2 et nous proposons d'ajouter un préconditionneur dans (1.9). L'algorithme avec préconditionneur est

$$\begin{pmatrix} B_1 u_2^{k+1} \\ B_2 u_1^{k+1} \end{pmatrix} = \begin{pmatrix} B_1 u_2^k \\ B_2 u_1^k \end{pmatrix} + P^{-1}(\mathcal{R} - I) \begin{pmatrix} B_1 u_2^k \\ B_2 u_1^k \end{pmatrix}, \quad (1.10)$$

où  $P$  est une matrice explicitement donnée. Nous proposons alors de prendre

$$P = I - \mathcal{L}_0,$$

où la matrice  $\mathcal{L}_0$  est la matrice  $\mathcal{L}$  dans (1.8) pour  $\mathcal{V} = 0$ . L'algorithme avec préconditionneur se montre être scalable jusqu'à 500 sous domaines d'après les tests numériques.

Puis, le chapitre 4 étudie les méthodes de décomposition de domaines pour l'équation de Schrödinger linéaire en deux dimensions sans recouvrement et  $\mathcal{V} = 0$ . On considère la

méthode SWR et la méthode de décomposition en espace en utilisant la méthode de Schwarz optimisée sur le problème stationnaire. La géométrie utilisée est celle de la figure 1.7. Sur l'interface, on utilise la condition de Robin et un type de condition absorbante comme condition de transmission. L'algorithme est de la même forme que (1.8). Avec quelques hypothèses sur le maillage, la structure de la matrice  $\mathcal{L}$  est un peu spéciale et donne une possibilité de la construire explicitement. Nous propose un nouvel algorithme qui suit l'idée du chapitre 2.

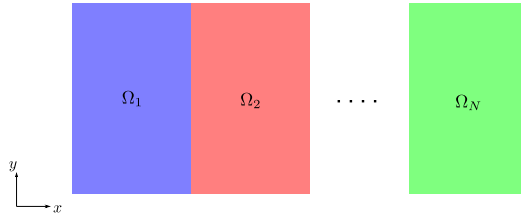


FIGURE 1.7: La décomposition sans recouvrement en deux dimensions.

Le chapitre 5 traite de l'équation de Schrödinger avec un potentiel linéaire non nul ou un potentiel non-linéaire  $\mathcal{V} = f(u)$ . Nous réutilisons l'idée du chapitre 3. Nous proposons donc un algorithme avec préconditionneur comme (1.10), où le préconditionneur est la matrice  $\mathcal{L}$  qui est construit dans le chapitre 4 pour  $\mathcal{V} = 0$ .

Nous généralisons une méthode de Schwarz optimisée avec points de croisement proposée par Loisel pour l'équation de Laplace à l'équation de Schrödinger qui conduit à une méthode SWR avec points de croisement dans le chapitre 6.

Dans le dernier chapitre, on considère la simulation de condensats de Bose-Einstein (BEC)[2, 14, 17] avec rotation en deux dimensions. Un des modèles pour les BEC est l'équation Gross-Pitaevskii (GPE). On applique les méthodes présentées dans les chapitres précédents à l'équation GPE.

Enfin, dans les sous chapitres 2.6.4 et 3.4.5, on présente les résultats numériques en utilisant l'accélération de Gpu pour l'équation de Schrödinger linéaire et non-linéaire en une dimension.

# Partie I

## Méthode de relaxation d'onde de Schwarz pour l'équation de Schrödinger en une dimension





# Chapitre 2

## Méthode SWR pour l'équation de Schrödinger en une dimension avec un potentiel indépendant du temps

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>17</b>
2.1.1	Méthode SWR	17
2.1.2	Condition de transmission	19
<b>2.2</b>	<b>Discrétisation</b>	<b>20</b>
<b>2.3</b>	<b>Problème d'interface</b>	<b>26</b>
<b>2.4</b>	<b>Nouvel algorithme SWR</b>	<b>32</b>
2.4.1	Construction de la matrice $\mathcal{L}$ et du vecteur $d$	33
2.4.2	Stockage de la matrice $\mathcal{L}$ et du vecteur $d$	35
<b>2.5</b>	<b>Convergence</b>	<b>36</b>
<b>2.6</b>	<b>Résultats Numériques</b>	<b>38</b>
2.6.1	Comparaison des algorithmes classiques et nouveaux	41
2.6.2	Comparaison des conditions de transmission	44
2.6.3	Effet du maillage	48
2.6.4	Accélération Gpu	50
<b>2.7</b>	<b>Conclusion</b>	<b>51</b>

---

## 2.1 Introduction

### 2.1.1 Méthode SWR

Nous nous intéressons dans ce chapitre au développement de la méthode de relaxation d'onde de Schwarz (SWR) pour l'équation de Schrödinger linéaire définie en une dimension d'espace

sur un domaine spatial borné  $]a_0, b_0[$  et pour  $t \in ]0, T[$ , à laquelle on adjoint un potentiel spatial indépendant du temps, et des conditions aux limites de Neumann homogènes.

$$\begin{cases} \mathcal{L}u := (i\partial_t + \partial_{xx} + V(x))u = 0, & (t, x) \in ]0, T[ \times ]a_0, b_0[, \\ u(0, x) = u_0(x), & x \in ]a_0, b_0[, \\ \partial_{\mathbf{n}}u(t, x) = 0, & x = a_0, b_0, t \in ]0, T[, \end{cases} \quad (2.1)$$

où  $V(\cdot) \in \mathcal{C}^\infty(\mathbb{R})$  est un potentiel réel,  $a_0, b_0 \in \mathbb{R}$ ,  $\partial_{\mathbf{n}}$  désigne l'opérateur de dérivée normale au bord,  $u_0 \in L^2(\mathbb{R})$  la donnée initiale et  $\mathcal{L}$  l'opérateur de Schrödinger linéaire.

Le domaine de temps-espace  $]0, T[ \times ]a_0, b_0[$  est décomposé en  $N$  sous-domaines  $]0, T[ \times ]a_j, b_j[, j = 0, 1, 2, \dots, N$  sans recouvrement (voir figure 2.1 pour  $N = 3$ ). Les points  $a_j$  et  $b_j$  désignent les extrémités des sous-domaines  $\Omega_j = ]a_j, b_j[$ . On définit également les normales unitaires sortantes aux sous-domaines  $\Omega_j$  par  $\mathbf{n}_j$ . Comme nous sommes en une dimension d'espace, elles ne peuvent prendre que les deux valeurs  $(-1, 0)^T$  et  $(1, 0)^T$  suivant la direction considérée.

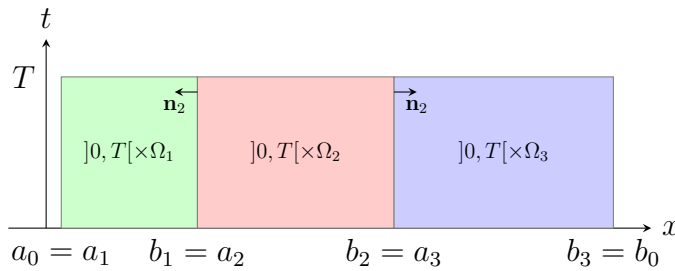


FIGURE 2.1: Décomposition de domaine sans recouvrement,  $N = 3$ .

Comme nous l'avons rappelé en introduction, la méthode SWR consiste à appliquer la suite d'itérations suivante à chaque sous-domaine

$$\begin{cases} \mathcal{L}u_j^{k+1} = 0, & (t, x) \in ]0, T[ \times \Omega_j, j = 2, \dots, N - 1, \\ u_j^{k+1}(0, x) = u_0(x), & x \in \Omega_j, \\ B_j u_j^{k+1} = B_j u_{j-1}^k, & x = a_j, \\ B_j u_j^{k+1} = B_j u_{j+1}^k, & x = b_j. \end{cases} \quad (2.2)$$

La notation  $u_j^k$  désigne la solution de (2.2) sur le domaine  $\Theta_j = ]0, T[ \times \Omega_j$  à l'itération  $k = 0, 1, \dots$  de la méthode SWR. Ainsi, on transmet au bord du domaine les informations provenant des solutions à l'itération précédente sur les domaines adjacents  $\Theta_{j-1}$  et  $\Theta_{j+1}$ . Ces informations transitent via les opérateurs de transmission  $B_j, j = 1, 2, \dots, N$ , définis par

$$B_j = \partial_{\mathbf{n}_j} + S_j, \quad (2.3)$$

où les opérateurs  $S_j$  peuvent prendre différentes formes (multiplication par un scalaire, opérateur de dérivation fractionnaire, etc ... voir Introduction).

Les deux domaines extrémaux nécessitent un traitement particulier car ils doivent prendre en compte les conditions aux limites de Neumann imposées dans (2.1) aux points  $a_0$  et  $b_0$ .

Ainsi, on a

$$\left\{ \begin{array}{l} \mathcal{L}u_1^{k+1} = 0, (t, x) \in \Theta_1, \\ u_1^{k+1}(0, x) = u_0(x), x \in \Omega_1, \\ \partial_{\mathbf{n}_1} u_1^{k+1} = 0, x = a_1, \\ B_1 u_1^{k+1} = B_1 u_2^k, x = b_1, \end{array} \right. \quad \left\{ \begin{array}{l} \mathcal{L}u_N^{k+1} = 0, (t, x) \in \Theta_N, \\ u_N^{k+1}(0, x) = u_0(x), x \in \Omega_N, \\ B_N u_N^{k+1} = B_N u_{N-1}^k, x = a_N, \\ \partial_{\mathbf{n}_N} u_N^{k+1} = 0, x = b_N. \end{array} \right.$$

Le but de ce chapitre est de comprendre l'influence des opérateurs  $S_j$  sur la dynamique de la méthode itérative SWR. Comme nous l'avons rappelé en introduction, Halpern et Szeftel [45] ont montré que la méthode SWR est optimale pour  $S_j = Op(\sqrt{\tau - V})$  si le potentiel considéré est constant. Cependant, cet opérateur est très complexe à calculer. Ainsi, des approximations sont considérées. Ils ont également montré comment optimiser les conditions aux limites de type Robin dans ce cadre. Nous nous intéressons ainsi à des conditions de transmission provenant des conditions aux limites artificielles pour (2.1) et les comparerons à celles de type Robin. L'autre intérêt de ce chapitre est de montrer comment écrire le problème des  $N$  sous-domaines sur l'intervalle de temps  $]0, T[$  sous la forme d'un opérateur global à partir des flux qui permet de réduire drastiquement le temps de résolution numérique de la méthode SWR.

## 2.1.2 Condition de transmission

Hormis les conditions de type Robin

$$B_j = \partial_{\mathbf{n}_j} - ip, \quad p \in \mathbb{R}^+, \quad (2.4)$$

nous proposons d'utiliser les opérateurs  $S_j$  provenant des conditions aux limites artificielles pour (2.1) déterminées dans [9] et [51] pour un potentiel général  $\mathcal{V} = V(t, x)$ . Ces auteurs proposent ainsi trois familles de conditions aux limites s'écrivant

$$\partial_{\mathbf{n}} u + S_l^M u = 0,$$

sur le bord du domaine de calcul considéré  $\Omega$ ,  $M$  désignant l'ordre d'approximation de la condition aux limites. On indice par  $l$  ces familles de conditions aux limites et reprenons ici les dénominations de [51] à savoir stratégie potentielle pour  $l = 0$ , stratégie de changement de jauge pour  $l = 1$  et stratégie d'approximation de Padé pour  $l = 2$ . Une des questions que nous aborderons dans ce chapitre est de comprendre si la convergence de la méthode SWR s'améliore avec  $M$ .

Nous faisons donc prendre aux différents opérateurs  $S_j$  les formes rappelées ci-dessous suivant les stratégies considérées (bien que nous travaillons dans ce chapitre avec un potentiel indépendant du temps, nous donnons ici la forme générale des opérateurs).

**Stratégie potentiel  $l = 0$  ([9])**

$$\begin{aligned} \text{Ordre } 2: \quad Su &= S_0^2 u = e^{-i\frac{\pi}{4}} \partial_t^{1/2} u, \\ \text{Ordre } 3: \quad Su &= S_0^3 u = S_0^2 u - e^{i\frac{\pi}{4}} \frac{V(t, x)}{2} I_t^{1/2} u, \\ \text{Ordre } 4: \quad Su &= S_0^4 u = S_0^3 u - i \frac{\partial_{\mathbf{n}} V(t, x)}{4} I_t u, \end{aligned}$$

où nous rappelons que l'opérateur de dérivée fractionnaire  $\partial_t^{1/2}$  est défini par

$$\partial_t^{1/2} f(t) = \frac{1}{\sqrt{\pi}} \partial_t \int_0^t \frac{f(s)}{\sqrt{t-s}} ds,$$

**Stratégie de changement de jauge  $l = 1$  ([5, 6])**

$$\begin{aligned} \text{Ordre } 2: \quad Su &= S_1^2 u = e^{-i\frac{\pi}{4}} e^{i\mathcal{V}(t, x)} \partial_t^{1/2} (e^{-i\mathcal{V}(t, x)} u), \\ \text{Ordre } 4: \quad Su &= S_1^4 u = S_1^2 u - i \operatorname{sgn}(\partial_{\mathbf{n}} V) \frac{\sqrt{|\partial_{\mathbf{n}} V(t, x)|}}{2} e^{i\mathcal{V}(t, x)} I_t \left( \frac{\sqrt{|\partial_{\mathbf{n}} V(t, x)|}}{2} e^{-i\mathcal{V}(t, x)} u \right), \end{aligned}$$

où  $\operatorname{sgn}(\cdot)$  est la fonction signe et

$$\mathcal{V}(t, x) = \int_0^t V(s, x) ds.$$

**Stratégie d'approximation de Padé  $l = 2$  ([5, 6])**

$$\begin{aligned} \text{Ordre } 2: \quad Su &= S_2^2 u = -i \sqrt{i \partial_t + V(t, x)} u, \\ \text{Ordre } 4: \quad Su &= S_2^4 u = S_2^2 u + \operatorname{sgn}(\partial_{\mathbf{n}} V(t, x)) \frac{\sqrt{|\partial_{\mathbf{n}} V(t, x)|}}{2} (i \partial_t + V(t, x))^{-1} \left( \frac{\sqrt{|\partial_{\mathbf{n}} V(t, x)|}}{2} u \right). \end{aligned}$$

## 2.2 Discrétisation

La discrétisation de l'équation de Schrödinger et des conditions de transmission joue un rôle fondamental dans notre analyse. On présente ici la discrétisation pour l'équation de Schrödinger linéaire avec un potentiel général  $\mathcal{V} = V(t, x)$ . L'intervalle de temps  $]0, T[$  est discretisé uniformément en  $N_T$  intervalles de longueur  $\Delta t = T/N_T$ . Le terme  $u_{j,n}^k$  représente l'approximation de  $u_j^k(n\Delta t, x)$  et  $V_n = V(n\Delta t, x)$ . Nous appliquons une semi-discrétisation de type Crank-Nicolson pour l'équation de Schrödinger sur chaque sous-domaine  $\Theta_j$ ,  $j = 1, 2, \dots, N$ . Celle-ci est donnée par

$$i \frac{u_{j,n}^k - u_{j,n-1}^k}{\Delta t} + \partial_{xx} \frac{u_{j,n}^k + u_{j,n-1}^k}{2} + \frac{V_n + V_{n-1}}{2} \frac{u_{j,n}^k + u_{j,n-1}^k}{2} = 0, \quad 1 \leq n \leq N_T.$$

La mise en œuvre est facilitée par l'introduction de nouvelles variables  $v_{j,n}^k = \frac{u_{j,n}^k + u_{j,n-1}^k}{2}$  et  $W_n = \frac{V_n + V_{n-1}}{2}$  où  $v_{j,0}^k = u_{j,0}^k$ . Le schéma s'écrit alors

$$2i \frac{v_{j,n}^k}{\Delta t} + \partial_{xx} v_{j,n}^k + W_n v_{j,n}^k = 2i \frac{u_{j,n-1}^k}{\Delta t}. \quad (2.5)$$

C'est ce schéma qui sera mis en œuvre et que nous utiliserons pour la description complète de la méthode SWR discrète.

Concernant l'approximation spatiale de (2.5), on utilise une méthode d'éléments finis  $P_1$ . La formulation variationnelle pour (2.5) consiste à chercher la fonction inconnue  $v_{j,n}^k$  dans l'espace de Sobolev  $H^1(\Omega_j)$  avec des fonctions test  $\phi \in H^1(\Omega_j)$ . On a pour  $1 \leq n \leq N_T$ ,

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{a_j}^{b_j} v_{j,n}^k \phi dx - \int_{a_j}^{b_j} \partial_x v_{j,n}^k \partial_x \phi + \int_{a_j}^{b_j} W_n v_{j,n}^k \phi dx \\ & + \partial_x v_{j,n}^k(b_j) \phi(b_j) - \partial_x v_{j,n}^k(a_j) \phi(a_j) = \frac{2i}{\Delta t} \int_{a_j}^{b_j} u_{j,n-1}^k \phi dx. \end{aligned} \quad (2.6)$$

L'utilisation des conditions de transmission donne pour chaque sous-domaine les conditions aux limites suivantes:

- $j = 1$ ,

$$\begin{cases} \partial_{\mathbf{n}_1} v_{1,n}^k = 0, & x = a_1, \\ \partial_{\mathbf{n}_1} v_{1,n}^k + \bar{S} v_{1,n}^k = \partial_{\mathbf{n}_1} v_{2,n}^{k-1} + \bar{S} v_{2,n}^{k-1}, & x = b_1, \end{cases}$$

- $j = 2, 3, \dots, N-1$ ,

$$\begin{cases} \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S} v_{j,n}^k = \partial_{\mathbf{n}_j} v_{j-1,n}^{k-1} + \bar{S} v_{j-1,n}^{k-1}, & x = a_j, \\ \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S} v_{j,n}^k = \partial_{\mathbf{n}_j} v_{j+1,n}^{k-1} + \bar{S} v_{j+1,n}^{k-1}, & x = b_j, \end{cases} \quad (2.7)$$

- $j = N$ ,

$$\begin{cases} \partial_{\mathbf{n}_N} v_{N,n}^k + \bar{S} v_{N,n}^k = \partial_{\mathbf{n}_N} v_{N-1,n}^{k-1} + \bar{S} v_{N-1,n}^{k-1}, & x = a_N, \\ \partial_{\mathbf{n}_N} v_{N,n}^k = 0, & x = b_N, \end{cases}$$

où  $\bar{S}$  désigne une discrétisation des opérateurs  $S$ . Pour chacune des stratégies, la semi-discrétisation est donnée par

**Stratégie potentiel  $l = 0$  ([9])**

$$\text{Ordre } 2 : \quad \bar{S} v_{j,n}^k = \bar{S}_0^2 v_{j,n}^k = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} v_{j,s}^k,$$

$$\text{Ordre } 3 : \quad \bar{S} v_{j,n}^k = \bar{S}_0^3 v_{j,n}^k = \bar{S}_0^2 v_{j,n}^k - e^{i\pi/4} \sqrt{\frac{\Delta t}{2}} \frac{W_n}{2} \sum_{s=0}^n \alpha_{n-s} v_{j,s}^k,$$

$$\text{Ordre } 4 : \quad \bar{S} v_{j,n}^k = \bar{S}_0^4 v_{j,n}^k = \bar{S}_0^3 v_{j,n}^k - i \frac{\partial_{\mathbf{n}_j} W_n \Delta t}{4} \frac{\Delta t}{2} \sum_{s=0}^n \gamma_{n-s} v_{j,s}^k,$$

où

$$\begin{aligned} (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \dots) &= (1, 1, \frac{1}{2}, \frac{1}{2}, \frac{3}{8}, \frac{3}{8}, \frac{3 \cdot 5}{2 \cdot 4 \cdot 6}, \dots), \\ \beta_s &= (-1)^s \alpha_s, \forall s \geq 0, \\ (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \dots) &= (1, 2, 2, 2, \dots). \end{aligned}$$

**Stratégie de changement de jauge  $l = 1$  ([5, 6])**

$$\begin{aligned} \text{Ordre } 2: \quad \bar{S}v_{j,n}^k &= \bar{S}_1^2 v_{j,n}^k = e^{-i\pi/4} e^{iW_n} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} e^{-iW_s} v_{j,s}^k, \\ \text{Ordre } 4: \quad \bar{S}v_{j,n}^k &= \bar{S}_1^4 v_{j,n}^k = \bar{S}_1^2 v_{j,n}^k \\ &\quad - i \operatorname{sgn}(\partial_{\mathbf{n}_j} W_n) \frac{\sqrt{|\partial_{\mathbf{n}_j} W_n|}}{2} e^{iW_n} \frac{\Delta t}{2} \sum_{s=0}^n \gamma_{n-s} \frac{\sqrt{|\partial_{\mathbf{n}_j} W_s|}}{2} e^{-iW_s} v_{j,s}^k, \end{aligned}$$

où  $W_n = \frac{\nu_n + \nu_{n-1}}{2}$ .

**Stratégie d'approximation de Padé  $l = 2$  ([5, 6])**

$$\begin{aligned} \bar{S}v_{j,n}^k &= \bar{S}_2^2 v_{j,n}^k = -i \left( \sum_{s=0}^m a_s^m \right) v_{j,n}^k + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n + d_s^m} v_{j,n}^k \\ &\quad + i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n + d_s^m} \varphi_{j,n-1}^s, \\ \bar{S}v_{j,n}^k &= \bar{S}_2^4 v_{j,n}^k = \bar{S}_2^2 v_{j,n}^k + \frac{\partial_{\mathbf{n}_j} W_n}{4} \frac{1}{\frac{2i}{\Delta t} + W_n} v_{j,n}^k \\ &\quad + \operatorname{sgn}(\partial_{\mathbf{n}_j} W_n) \frac{\sqrt{|\partial_{\mathbf{n}_j} W_n|}}{2} \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n} \psi_{j,n-1}, \end{aligned}$$

où  $\varphi_{j,n}^s, \phi_{j,n}, s = 1, 2, \dots, m$  sont introduits comme fonctions auxiliaires

$$\begin{cases} \varphi_{j,n-\frac{1}{2}}^s = \frac{1}{\frac{2i}{\Delta t} + W_n + d_s^m} v_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n + d_s^m} \varphi_{j,n-1}^s, & s = 1, 2, \dots, m \\ \varphi_{j,n}^s = 2\varphi_{j,n-\frac{1}{2}}^s - \varphi_{j,n-1}^s, \\ \varphi_{j,0}^s = 0, \end{cases}$$

et

$$\begin{cases} \psi_{j,n-\frac{1}{2}} = \frac{\sqrt{|\partial_{\mathbf{n}_j} W_n|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_n} v_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n} \psi_{j,n-1}, \\ \psi_{j,n} = 2\psi_{j,n-\frac{1}{2}} - \psi_{j,n-1}, \\ \psi_{j,0} = 0. \end{cases}$$

On rappelle également ici la condition de Robin

$$S = S_p = -ip, \quad p \in \mathbb{R}^+.$$

et son approximation

$$\overline{S}v_{j,n}^k = \overline{S}_p v_{j,n}^k = -ip \cdot v_{j,n}^k.$$

Afin de donner la formulation complètement discrète des  $N$  problèmes aux limites, nous introduisons des flux aux interfaces. On définit pour  $1 \leq n \leq N_T$  et  $k = 0, 1, 2, \dots$  les flux à gauche  $l_{j,n}^k$  et les flux à droite  $r_{j,n}^k$  pour les domaines  $\Omega_j$  par (voir figure 2.2)

- $j = 1,$

$$l_{1,n}^k = 0, \quad r_{1,n}^k = \partial_{\mathbf{n}_1} v_{1,n}^k(b_1) + \overline{S}v_{1,n}^k(b_1),$$

- $j = 2, 3, \dots, N - 1,$

$$l_{j,n}^k = \partial_{\mathbf{n}_j} v_{j,n}^k(a_j) + \overline{S}v_{j,n}^k(a_j), \quad r_{j,n}^k = \partial_{\mathbf{n}_j} v_{j,n}^k(b_j) + \overline{S}v_{j,n}^k(b_j), \quad (2.8)$$

- $j = N,$

$$l_{N,n}^k = \partial_{\mathbf{n}_N} v_{N,n}^k(a_N) + \overline{S}v_{N,n}^k(a_N), \quad r_{N,n}^k = 0.$$

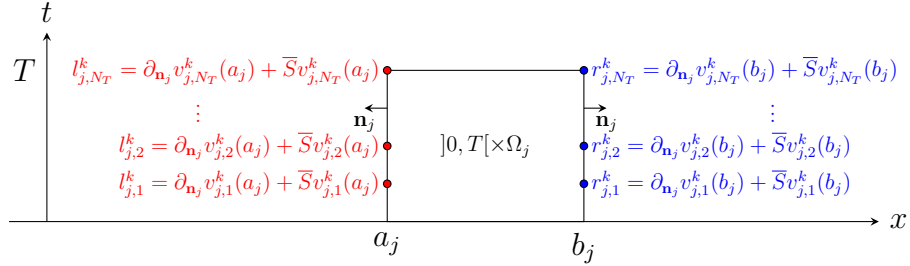


FIGURE 2.2: Illustration de la définition de  $l_{j,n}^k$  et  $r_{j,n}^k$ .

Pour les conditions de transmission  $S_0^M$ ,  $M = 2, 3$ ,  $S_1^2$  et  $S_2^2$ , en utilisant (2.7), on a

$$\begin{aligned} r_{1,n}^k &= \partial_{\mathbf{n}_1} v_{1,n}^k(b_1) + \overline{S}v_{j,n}^k(b_1) = \partial_{\mathbf{n}_1} v_{2,n}^{k-1}(a_2) + \overline{S}v_{2,n}^{k-1}(a_2) \\ &\quad - \left( \partial_{\mathbf{n}_2} v_{2,n}^{k-1}(a_2) + \overline{S}v_{2,n}^{k-1}(a_2) \right) + 2\overline{S}v_{2,n}^{k-1}(a_2) = -l_{2,n}^{k-1} + 2\overline{S}v_{2,n}^{k-1}(a_2). \end{aligned}$$

Les flux permettent alors de réécrire les conditions de transmission comme

$$\begin{cases} l_{1,n}^k = 0, \\ r_{1,n}^k = -l_{2,n}^{k-1} + 2\overline{S}v_{2,n}^{k-1}(a_2), \end{cases} \quad \begin{cases} l_{N,n}^k = -r_{N-1,n}^{k-1} + 2\overline{S}v_{N-1,n}^{k-1}(b_{N-1}), \\ r_{N,n}^k = 0. \end{cases} \quad (2.9)$$

$$\begin{cases} l_{j,n}^k = -r_{j-1,n}^{k-1} + 2\overline{S}v_{j-1,n}^{k-1}(b_{j-1}), \\ r_{j,n}^k = -l_{j+1,n}^{k-1} + 2\overline{S}v_{j+1,n}^{k-1}(a_{j+1}), \end{cases} \quad j = 2, 3, \dots, N - 1$$

Les conditions de transmission  $S_0^4$ ,  $S_1^4$  et  $S_2^4$  contiennent la dérivée normale de  $W_n$  ( $\partial_{\mathbf{n}_j} W_n$ ). Nous pouvons également obtenir la formule (2.9) mais avec  $\overline{S} = \overline{S}_0^3$  pour la condition de transmission  $S_0^4$ ,  $\overline{S} = \overline{S}_1^2$  pour la condition de transmission  $S_1^4$  et  $\overline{S} = \overline{S}_2^2$  pour la condition de transmission  $S_2^4$ .

Comme nous réalisons une approximation de type éléments finis  $P_1$ , notons pour  $j = 0, 1, 2, \dots, N$ ,  $k = 0, 1, 2, \dots$  et  $0 \leq n \leq N_T - 1$

- $\mathbf{v}_{j,n}^k$ : le vecteur nodal de l'interpolation  $P_1$  de  $v_{j,n}^k$  avec  $N_j$  nœuds,
- $\mathbf{u}_{j,n}^k$ : le vecteur nodal de l'interpolation  $P_1$  de  $u_{j,n}^k$  avec  $N_j$  nœuds,
- $\mathbb{M}_j$ : la matrice de masse pour les éléments finis  $P_1$ ,
- $\mathbb{S}_j$ : la matrice de raideur pour les éléments finis  $P_1$ ,
- $\mathbb{M}_{j,W_n}$ : la matrice de masse généralisée associée à  $\int_{a_j}^{b_j} W_n v \phi dx$ ,

et  $Q_j$  la matrice

$$Q_j = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{C}^{2 \times N_j}.$$

Matriciellement, la formulation variationnelle (2.6) conduit aux  $N$  systèmes linéaires

$$(\mathbb{A}_{j,n} - \mathbb{B}_{j,n}) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T \begin{pmatrix} l_n^k \\ r_n^k \end{pmatrix}, \quad (2.10)$$

où  $\mathbb{A}_{j,n} = \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j + \mathbb{M}_{j,W_n}$ . La matrice  $\mathbb{B}_{j,n} \in \mathbb{C}^{N_j \times N_j}$  est la matrice des contributions aux bords issues de la condition de transmission à l'étape de temps  $n$ , tandis que  $\mathbf{b}_{j,n}^k$  représente le vecteur des contributions aux bords.

$$\mathbb{B}_{j,n} = \begin{pmatrix} B_{j,n}^l & 0 & \cdots & \cdots & 0 \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & 0 \\ 0 & 0 & \cdots & 0 & B_{j,n}^r \end{pmatrix} \in \mathbb{C}^{N_j \times N_j}, \quad \mathbf{b}_{j,n}^k = \begin{pmatrix} b_{j,n}^{k,l} \\ 0 \\ \vdots \\ 0 \\ b_{j,n}^{k,r} \end{pmatrix} \in \mathbb{C}^{N_j}, \quad (2.11)$$

Si nous notons

$$Q_{j,l} = (1 \ 0 \ 0 \ \cdots \ 0 \ 0) \in \mathbb{C}^{N_j}, \quad Q_{j,r} = (0 \ 0 \ 0 \ \cdots \ 0 \ 1) \in \mathbb{C}^{N_j},$$

la forme discrète de la condition des transmission est donnée par

$$\begin{aligned} l_{j+1,n}^{k+1} &= -r_{j,n}^k + 2\tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} &= -l_{j,n}^k + 2\tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N. \end{aligned} \quad (2.12)$$

La définition de l'opérateur  $\tilde{S}$  et des quantités scalaires  $B_{j,n}^l$ ,  $B_{j,n}^r$ ,  $b_{j,n}^l$ ,  $b_{j,n}^r$  dépendent de la condition de transmission. Nous les donnons ci-dessous pour la condition  $S_0^2$ . Pour les autres conditions de transmission présentées dans la sous-section 2.1.2, on se réfère à l'annexe A. Nous avons ainsi à résoudre pour chaque sous-domaine et chaque pas de temps les  $N$



systèmes linéaires suivants

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_0^2(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_0^2(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_0^2(Q_{j,l}\mathbf{v}_{j,n}^k) = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} (Q_{j,l}\mathbf{v}_{j,s}^k), \\ \tilde{S}_0^2(Q_{j,r}\mathbf{v}_{j,n}^k) = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} (Q_{j,r}\mathbf{v}_{j,s}^k), \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T (l_{j,n}^k, r_{j,n}^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 2, 3, \dots, N, \\ B_{j,n}^r = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 1, 2, \dots, N-1, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s} Q_{j,l} \mathbf{v}_{j,s}^k, \quad j = 2, 3, \dots, N, \\ b_{j,n}^{k,r} = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s} Q_{j,r} \mathbf{v}_{j,s}^k, \quad j = 1, 2, \dots, N-1. \end{array} \right. \quad (2.13)$$

Nous remarquons finalement que la mise en œuvre des conditions aux limites de type Dirichlet-Neumann conduit à

$$\begin{aligned} \mathbb{B}_{1,n} \mathbf{v}_{1,n}^k + \mathbf{b}_{1,n}^k &= Q_{1,r}^T \cdot \tilde{S}(Q_{1,r} \mathbf{v}_{1,n}^k), \\ \mathbb{B}_{N,n} \mathbf{v}_{N,n}^k + \mathbf{b}_{N,n}^k &= Q_{N,l}^T \cdot \tilde{S}(Q_{N,l} \mathbf{v}_{N,n}^k), \end{aligned} \quad (2.14)$$

$$\mathbb{B}_{j,n} \mathbf{v}_{j,n}^k + \mathbf{b}_{j,n}^k = \begin{pmatrix} \tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k) \\ 0 \\ \vdots \\ 0 \\ \tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k) \end{pmatrix} = Q_{j,l}^T \cdot \tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k) + Q_{j,r}^T \cdot \tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N-1.$$

Sur le domaine complet  $]a_0, b_0[$ , l'équation de Schrödinger couplée aux conditions aux limites de Neumann homogènes est également discrétisée par la même méthode. Ainsi, on a

$$\mathbb{A}_{0,n} \mathbf{v}_{0,n} = \frac{2i}{\Delta t} \mathbb{M}_0 \mathbf{u}_{0,n-1}. \quad (2.15)$$

Le but de la méthode SWR est que, pour  $n = 1, 2, \dots, N$ ,

$$\lim_{k \rightarrow +\infty} \mathbf{v}_{j,n}^k = R_j \mathbf{v}_{0,n}.$$

où  $R_j \in \mathbb{R}^{N_j \times N_0}$  est la matrice de restriction des nœuds dans  $]a_0, b_0[$  à  $]a_j, b_j[$ .



- pour  $j = 2, \dots, N - 1$ ,

$$\begin{pmatrix} r_{j-1,1}^{k+1} \\ r_{j-1,2}^{k+1} \\ \vdots \\ r_{j-1,N_T}^{k+1} \end{pmatrix} = X^{j,1} \begin{pmatrix} l_{j,1}^k \\ l_{j,2}^k \\ \vdots \\ l_{j,N_T}^k \end{pmatrix} + X^{j,2} \begin{pmatrix} r_{j,1}^k \\ r_{j,2}^k \\ \vdots \\ r_{j,N_T}^k \end{pmatrix} + d_{j-1,r},$$

$$\begin{pmatrix} l_{j+1,1}^{k+1} \\ l_{j+1,2}^{k+1} \\ \vdots \\ l_{j+1,N_T}^{k+1} \end{pmatrix} = X^{j,3} \begin{pmatrix} l_{j,1}^k \\ l_{j,2}^k \\ \vdots \\ l_{j,N_T}^k \end{pmatrix} + X^{j,4} \begin{pmatrix} r_{j,1}^k \\ r_{j,2}^k \\ \vdots \\ r_{j,N_T}^k \end{pmatrix} + d_{j+1,l},$$
(2.20)

- pour  $j = N$ ,

$$\begin{pmatrix} r_{N-1,1}^{k+1} \\ r_{N-1,2}^{k+1} \\ \vdots \\ r_{N-1,N_T}^{k+1} \end{pmatrix} = X^{N,1} \begin{pmatrix} l_{N,1}^k \\ l_{N,2}^k \\ \vdots \\ l_{N,N_T}^k \end{pmatrix} + d_{N-1,r}.$$

Pour démontrer (2.20), il suffit de vérifier que la formule est juste pour toutes les conditions de transmission présentées dans la section 2.1.2. Pour  $1 \leq n \leq N_T$ , à l'itération  $k$ , sur  $]a_j, b_j[$ , on utilise  $l_{j,s}^k$  et  $r_{j,s}^k, s = 1, 2, \dots, N_T$  pour calculer  $\mathbf{v}_{j,n}^k$  par (2.10). On construit ensuite  $r_{j-1,n}^{k+1}$  ( $j \geq 2$ ) et  $l_{j+1,n}^{k+1}$  ( $j \leq N - 1$ ) par (2.9). De plus, nous considérons l'équation de Schrödinger linéaire. Ainsi, il n'est pas difficile de retrouver la relation (2.20) intuitivement. Nous démontrerons cette relation dans la proposition 2.1 pour la condition de transmission  $S_0^2$  et dans la proposition 2.2 pour un opérateur  $S$  général.

**Proposition 2.1.** *Si l'on considère la condition de transmission mettant en jeu l'opérateur  $S_0^2$ , et si on suppose que les matrices  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$ ,  $n = 1, 2, \dots, N_T$  ne sont pas singulières, alors les  $N$  équations (2.13) s'écrivent sous la forme globale du problème d'interface (2.17)*

$$g^{k+1} = \mathcal{L}g^k + d.$$

*Preuve.* D'abord, d'après (2.13), on a

$$\begin{aligned} (\mathbb{A}_{j,1} - \mathbb{B}_{j,1})\mathbf{v}_{j,1}^k &= \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,0} + e^{-\frac{i\pi}{4}} Q_j^T \sqrt{\frac{2}{\Delta t}} \beta_1 Q_j \mathbf{v}_{j,0}^k - Q_j^T (l_{j,1}^k, r_{j,1}^k)^T, \\ (\mathbb{A}_{j,2} - \mathbb{B}_{j,2})\mathbf{v}_{j,2}^k &= \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} Q_j^T \sum_{q=0}^1 \beta_{2-q} Q_j \mathbf{v}_{j,q}^k - Q_j^T (l_{j,2}^k, r_{j,2}^k)^T \\ &= \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,1}^k - \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,0} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} Q_j^T \sum_{q=0}^1 \beta_{2-q} Q_j \mathbf{v}_{j,q}^k - Q_j^T (l_{j,2}^k, r_{j,2}^k)^T, \\ (\mathbb{A}_{j,3} - \mathbb{B}_{j,3})\mathbf{v}_{j,3}^k &= \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,2}^k - \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} Q_j^T \sum_{q=0}^2 \beta_{3-q} Q_j \mathbf{v}_{j,q}^k - Q_j^T (l_{j,3}^k, r_{j,3}^k)^T, \\ &= \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,2}^k - \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,1}^k + \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,0} \end{aligned}$$

$$+ e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} Q_j^T \sum_{q=0}^2 \beta_{3-q} Q_j \mathbf{v}_{j,q}^k - Q_j^T (l_{j,3}^k, r_{j,3}^k)^T,$$

et de même pour  $n > 4$ .

Nous nous rappelons que  $\mathbf{v}_{j,0}^k = \mathbf{u}_{j,0}$ . Ainsi, nous pouvons voir que

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T (l_{j,n}^k, r_{j,n}^k)^T \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j + e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_{n-q} Q_j^T Q_j \right) \mathbf{v}_{j,q}^k \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( (-1)^{n-1} \frac{2i}{\Delta t} \mathbb{M}_j + e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_n Q_j^T Q_j \right) \mathbf{u}_{j,0}. \end{aligned} \quad (2.21)$$

Par récurrence sur  $n$ , nous pouvons voir facilement que  $\mathbf{v}_{j,n}^k$  est fonction linéaire de  $l_{j,s}^k$  et  $r_{j,s}^k$ ,  $s = 1, 2, \dots, n$ .

Ensuite, en considérant les deux premières formules de (2.13), il suffit de vérifier que  $\tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k)$  et  $\tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k)$  sont fonctions linéaires de  $\mathbf{v}_{j,s}^k$ ,  $s = 1, 2, \dots, n$ . La conclusion est alors directe.  $\square$

**Proposition 2.2.** *Quelque soit la condition de transmission issue de la sous-section 2.1.2, et sous l'hypothèse que les matrices  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$ ,  $n = 1, 2, \dots, N_T$  ne sont pas singulières, alors le problème prend la forme globale du problème d'interface (2.17).*

*Preuve.* La démonstration est similaire à celle réalisée précédemment. Pour chaque condition de transmission, il suffit de refaire le calcul pour obtenir l'expression de  $\mathbf{v}_{j,n}^k$ . On se réfère à l'annexe A pour les détails.  $\square$

Intéressons nous maintenant à la structure des sous blocs

$$\begin{aligned} X^{1,4} &= \{x_{n,s}^{1,4}\}_{1 \leq n, s \leq N_T}, \\ X^{j,1} &= \{x_{n,s}^{j,1}\}_{1 \leq n, s \leq N_T}, X^{j,2} = \{x_{n,s}^{j,2}\}_{1 \leq n, s \leq N_T}, \\ X^{j,3} &= \{x_{n,s}^{j,3}\}_{1 \leq n, s \leq N_T}, X^{j,4} = \{x_{n,s}^{j,4}\}_{1 \leq n, s \leq N_T}, j = 2, 3, \dots, N-1, \\ X^{N-1,1} &= \{x_{n,s}^{N-1,1}\}_{1 \leq n, s \leq N_T}. \end{aligned}$$

Pour 5 pas de temps, cette structure est décrite ci-dessous

$$\begin{pmatrix} * & & & & \\ * & * & & & \\ * & * & * & & \\ * & * & * & * & \\ * & * & * & * & * \end{pmatrix}, \quad N_T = 5.$$

Ainsi, chaque sous-diagonale a chaque élément identique.

**Proposition 2.3.** *Pour la condition de transmission mettant en jeu l'opérateur  $S_0^2$ , si  $\mathcal{V} = V(x)$  et on suppose que  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$ ,  $n = 1, 2, \dots, N_T$  ne sont pas singulière, alors les*

matrices  $X^{1,4}, X^{j,1}, X^{j,2}, X^{j,3}, X^{j,4}, j = 2, 3, \dots, N-1$  et  $X^{N,1}$  sont des matrices triangulaires inférieures qui satisfont

$$\begin{aligned} x_{n,s}^{1,4} &= x_{n-1,s-1}^{1,4}, \\ x_{n,s}^{j,1} &= x_{n-1,s-1}^{j,1}, \quad x_{n,s}^{j,2} = x_{n-1,s-1}^{j,2}, \\ x_{n,s}^{j,3} &= x_{n-1,s-1}^{j,3}, \quad x_{n,s}^{j,4} = x_{n-1,s-1}^{j,4}, \quad j = 2, 3, \dots, N-1, \\ x_{n,s}^{N,1} &= x_{n-1,s-1}^{N,1}, \end{aligned}$$

pour  $2 \leq s \leq n \leq N_T$ .

*Preuve.* Sans perte de généralité, nous considérons ici  $j = 2, 3, \dots, N-1$ . D'abord, nous désignons

$$\mathbb{Y}_{n,q}^j = \begin{cases} -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}, & q = n, \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j + e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_{n-q} Q_j^T Q_j \right), & q = 1, 2, \dots, n-1. \end{cases}$$

Si le potentiel  $\mathcal{V} = V(x)$  est indépendant du temps, alors il est facile de voir

$$\mathbb{A}_{j,1} = \mathbb{A}_{j,2} = \dots = \mathbb{A}_{j,N_T}, \quad \mathbb{B}_{j,1} = \mathbb{B}_{j,2} = \dots = \mathbb{B}_{j,N_T}.$$

Ainsi pour  $2 \leq s \leq n \leq N_T$ ,

$$\mathbb{Y}_{n,s}^j = \mathbb{Y}_{n-1,s-1}^j. \quad (2.22)$$

Ensuite, d'après (2.21), on a

$$\mathbf{v}_{j,n}^k = \mathbb{Y}_{n,n}^j Q_j^T (l_{j,n}^k, r_{j,n}^k)^T + \sum_{q=1}^{n-1} \mathbb{Y}_{n,q}^j \mathbf{v}_{j,q}^k + \mathbb{U}_{j,n} \mathbf{u}_{j,0}. \quad (2.23)$$

où  $\mathbb{U}_{j,n} = (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j + e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \beta_n Q_j^T Q_j \right)$ . Par récurrence, nous pouvons obtenir une expression de  $\mathbf{v}_{j,n}^k$ :

$$\mathbf{v}_{j,n}^k = \sum_{q=1}^n \mathbb{L}_{n,q}^j Q_j^T (l_{j,q}^k, r_{j,q}^k)^T + U_{j,n} \mathbf{u}_{j,0}, \quad (2.24)$$

où  $\mathbb{L}_{n,q}^j, q = 1, 2, \dots, n$  et  $U_{j,n}$  sont des matrices. Par exemple,  $\mathbb{L}_{n,n}^j = \mathbb{Y}_{n,n}^j$ . Nous allons montrer que pour  $2 \leq s \leq n \leq N_T$ ,

$$\mathbb{L}_{n,s}^j = \mathbb{L}_{n-1,s-1}^j. \quad (2.25)$$

En remplaçant  $\mathbf{v}_{j,q}^k$  dans (2.23) par (2.24), on a

$$\begin{aligned} \mathbf{v}_{j,n}^k &= \mathbb{Y}_{n,n}^j Q_j^T (l_{j,n}^k, r_{j,n}^k)^T + \sum_{q=1}^{n-1} \mathbb{Y}_{n,q}^j \left( \sum_{p=1}^q \mathbb{L}_{q,p}^j Q_j^T (l_{j,p}^k, r_{j,p}^k)^T + U_{j,q} \mathbf{u}_{j,0} \right) + \mathbb{U}_{j,n} \mathbf{u}_{j,0} \\ &= \mathbb{Y}_{n,n}^j Q_j^T (l_{j,n}^k, r_{j,n}^k)^T + \sum_{p=1}^{n-1} \left( \sum_{q=p}^{n-1} \mathbb{Y}_{n,q}^j \mathbb{L}_{q,p}^j \right) Q_j^T (l_{j,p}^k, r_{j,p}^k)^T + \left( \sum_{q=1}^{n-1} \mathbb{Y}_{n,q}^j U_{j,q} + \mathbb{U}_{j,n} \right) \mathbf{u}_{j,0}. \end{aligned}$$

Comparons la formule ci-dessus avec (2.24), on obtient

$$\mathbb{L}_{n,s} = \begin{cases} \mathbb{Y}_{n,n}^j, \\ \sum_{q=s}^{n-1} \mathbb{Y}_{n,q}^j \mathbb{L}_{q,s}^j, 1 \leq s < n, \end{cases} \Rightarrow \mathbb{L}_{n-1,s-1} = \begin{cases} \mathbb{Y}_{n-1,n-1}^j, \\ \sum_{q=s-1}^{n-2} \mathbb{Y}_{n-1,q}^j \mathbb{L}_{q,s-1}^j, 2 \leq s < n. \end{cases} \quad (2.26)$$

En utilisant (2.22) et par récurrence sur  $n$ , on a

$$\begin{aligned} \mathbb{L}_{n,n} &= \mathbb{Y}_{n,n} = \mathbb{Y}_{n-1,n-1} = \mathbb{L}_{n-1,n-1}, \\ \mathbb{L}_{n,s} &= \sum_{q=s}^{n-1} \mathbb{Y}_{n,q}^j \mathbb{L}_{q,s}^j = \sum_{q=s}^{n-1} \mathbb{Y}_{n-1,q-1}^j \mathbb{L}_{q-1,s-1}^j = \sum_{q=s-1}^{n-2} \mathbb{Y}_{n-1,q}^j \mathbb{L}_{q,s-1}^j = \mathbb{L}_{n-1,s-1}, 2 \leq s < n. \end{aligned}$$

La formule (2.25) est ainsi démontrée.

Puis, nous remplaçons  $\mathbf{v}_{j,n}^k$  dans les deux premières formules de (2.13) par (2.24). Alors

$$\begin{aligned} l_{j+1,n}^{k+1} &= -r_{j,n}^k + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{p=1}^n \beta_{n-p} \sum_{q=1}^p \mathbb{L}_{p,q}^j Q_j^T(l_{j,q}^k, r_{j,q}^k)^T + \dots \\ &= -r_{j,n}^k + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=1}^n Q_{j,r} \left( \sum_{p=q}^n \beta_{n-p} \mathbb{L}_{p,q}^j \right) Q_j^T(l_{j,q}^k, r_{j,q}^k)^T + \dots, \quad (2.27) \\ r_{j-1,n}^{k+1} &= -l_{j,n}^k + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=1}^n Q_{j,l} \left( \sum_{p=q}^n \beta_{n-p} \mathbb{L}_{p,q}^j \right) Q_j^T(l_{j,q}^k, r_{j,q}^k)^T + \dots, \end{aligned}$$

où nous désignons les termes qui sont indépendants de  $l_{j,s}^k$  et  $r_{j,s}^k$ ,  $s = 1, 2, \dots, N_T$  par "... "

De plus, d'après (2.20), on a

$$\begin{aligned} l_{2,n}^{k+1} &= \sum_{s=1}^{N_T} x_{n,s}^{1,4} r_{1,s}^k + \dots, \quad r_{N-1,n}^{k+1} = \sum_{s=1}^{N_T} x_{n,s}^{N,1} l_{N,s}^k + \dots, \\ r_{j-1,n}^{k+1} &= \sum_{s=1}^{N_T} x_{n,s}^{j,1} l_{j,s}^k + \sum_{s=1}^{N_T} x_{n,s}^{j,2} r_{j,s}^k + \dots, \quad l_{j+1,n}^{k+1} = \sum_{s=1}^{N_T} x_{n,s}^{j,3} l_{j,s}^k + \sum_{s=1}^{N_T} x_{n,s}^{j,4} r_{j,s}^k + \dots. \end{aligned}$$

En comparant la formule ci-dessus avec (2.27), on obtient pour  $1 \leq n < s \leq N_T$ ,

$$x_{n,s}^{j,1} = x_{n,s}^{j,2} = x_{n,s}^{j,3} = x_{n,s}^{j,4} = 0,$$

et pour  $1 \leq s \leq n \leq N_T$ ,

$$\begin{aligned} x_{n,s}^{j,1} &= -1 + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,l} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p,s}^j \right) Q_{j,l}^T, \\ x_{n,s}^{j,2} &= 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,l} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p,s}^j \right) Q_{j,r}^T, \end{aligned}$$

$$x_{n,s}^{j,3} = 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,r} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p,s}^j \right) Q_{j,l}^T,$$

$$x_{n,s}^{j,4} = -1 + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,r} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p,s}^j \right) Q_{j,r}^T,$$

où nous utilisons  $Q_j^T (l_{j,q}^k, r_{j,q}^k)^T = Q_{j,l}^T l_{j,q}^k + Q_{j,r}^T r_{j,q}^k$ .

À la fin, en utilisant (2.25), on a pour  $2 \leq s \leq n \leq N_T$ ,

$$\begin{aligned} x_{n,s}^{j,1} &= -1 + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,l} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p,s}^j \right) Q_{j,l}^T \\ &= -1 + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,l} \left( \sum_{p=s}^n \beta_{n-p} \mathbb{L}_{p-1,s-1}^j \right) Q_{j,l}^T \\ &= -1 + 2e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} Q_{j,l} \left( \sum_{p=s-1}^{n-1} \beta_{n-1-p} \mathbb{L}_{p,s}^j \right) Q_{j,l}^T = x_{n-1,s-1}^{j,1}. \end{aligned}$$

De la même manière, nous pouvons démontrer  $x_{n,s}^{j,2} = x_{n-1,s-1}^{j,2}$ ,  $x_{n,s}^{j,3} = x_{n-1,s-1}^{j,3}$  et  $x_{n,s}^{j,4} = x_{n-1,s-1}^{j,4}$ .  $\square$

**Proposition 2.4.** *Avec une condition de transmission quelconque de la sous-section 2.1.2, si  $\mathcal{V} = V(x)$  et sous l'hypothèse que  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$ ,  $n = 1, 2, \dots, N_T$  ne sont pas singulières, alors les matrices  $X^{1,4}$ ,  $X^{j,1}$ ,  $X^{j,2}$ ,  $X^{j,3}$ ,  $X^{j,4}$ ,  $j = 2, 3, \dots, N-1$  et  $X^{N,1}$  sont des matrices triangulaires inférieures qui satisfont*

$$\begin{aligned} x_{n,s}^{1,4} &= x_{n-1,s-1}^{1,4}, \\ x_{n,s}^{j,1} &= x_{n-1,s-1}^{j,1}, \quad x_{n,s}^{j,2} = x_{n-1,s-1}^{j,2}, \\ x_{n,s}^{j,3} &= x_{n-1,s-1}^{j,3}, \quad x_{n,s}^{j,4} = x_{n-1,s-1}^{j,4}, \quad j = 2, 3, \dots, N-1, \\ x_{n,s}^{N,1} &= x_{n-1,s-1}^{N,1}, \end{aligned}$$

pour  $2 \leq s \leq n \leq N_T$ .

*Preuve.* La preuve est très similaire à celle de la proposition 2.3. Il y a cependant deux petites différences. La première est la définition de  $\mathbb{Y}_{n,q}^j$ . Dans la preuve précédente,  $\mathbb{Y}_{n,q}^j$  est défini selon l'expression de  $\mathbf{v}_{j,n}^k$  (voir la formule (2.21)). Il suffit de redéfinir  $\mathbb{Y}_{n,q}^j$  ici directement d'après les expressions de  $\mathbf{v}_{j,n}^k$  présentées dans l'annexe A. Par ailleurs, nous avons obtenu (2.27) avec (2.13) pour la condition de transmission  $S_0^2$ . Dans cette preuve, nous pouvons obtenir les formules similaires que (2.27) en utilisant la forme discrète des conditions de transmission qui sont aussi montrées dans l'annexe A.  $\square$

## 2.4 Nouvel algorithme SWR

La mise en œuvre standard de la méthode SWR pour les équations dépendants du temps conduit à l'algorithme classique suivant

---

**Algorithme 1** : Algorithme classique

---

- 1: Initialiser l'itération par  $g^0$ .
  - 2: Résoudre l'équation de Schrödinger sur chaque sous-domaine avec  $g^k$ .
  - 3: Échanger les valeurs aux interfaces et calculer  $g^{k+1}$ .
  - 4: Répéter les étapes 2 et 3 jusqu'à  $\|g^{k+1} - g^k\| < \varepsilon$ ,  $\varepsilon \ll 1$ .
- 

Nous remarquons ici que la condition d'arrêt  $\|g^{k+1} - g^k\| < \varepsilon$  porte sur les interfaces. Comme nous le voyons, l'algorithme classique nécessite de résoudre  $K$  fois l'équation de Schrödinger sur chaque sous-domaine, où  $K$  correspondrait au nombre d'itérations de Schwarz nécessaire pour arriver à convergence. Nous présentons dans cette sous-section un nouvel algorithme pour  $V = V(x)$  plus efficace puisqu'il ne nécessite qu'une seule fois la résolution de l'équation de Schrödinger. Ce nouvel algorithme est équivalent à l'algorithme classique mais réduit donc de manière significative les calculs.

Avant de donner ce nouvel algorithme, nous voyons que l'algorithme classique se base sur l'écriture de

$$g^{k+1} = \mathcal{R}g^k,$$

où l'opérateur  $\mathcal{R}$  contient l'étape 2 et 3. Nous avons montré dans la proposition 2.2 que

$$g^{k+1} = \mathcal{R}g^k = \mathcal{L}g^k + d. \quad (2.28)$$

Il est facile de voir que (2.28) n'est juste que la méthode de point fixe pour

$$(I - \mathcal{L})g = d. \quad (2.29)$$

Un gros avantage d'interpréter (2.28) comme une méthode de point fixe pour (2.29) est que nous pouvons utiliser n'importe quelle autre méthode pour résoudre cette équation. Ainsi, nous pouvons utiliser des méthodes Krylov (Gmres, Bicgstab, etc.) [62], qui permettent d'envisager des accélérations de convergence.

Pour utiliser des méthodes Krylov ou la méthode point fixe, il suffit de définir l'application de  $I - \mathcal{L}$  au vecteur  $g$  par

$$(I - \mathcal{L})g = I - \mathcal{R}g + d.$$



L'algorithme classique peut alors être réécrit avec

---

**Algorithme 2** : Algorithme classique, version 2
 

---

- 1: Construire  $d = \mathcal{R} \cdot \mathbf{0}$  dans (2.29) explicitement, où  $\mathbf{0}$  est le vecteur nul.
  - 2: Définir l'application de  $I - \mathcal{L}$  au vecteur dans (2.29).
  - 3: Résoudre le système linéaire (2.29) par une méthode itérative (méthode point fixe ou méthodes Krylov).
  - 4: Résoudre l'équation de Schrödinger sur chaque sous-domaine pour chaque pas de temps en utilisant les conditions aux limites obtenues à l'étape 3.
- 

Si la méthode point fixe est utilisé à l'étape 3, nous retrouvons la première version de l'algorithme classique. Cette version 2 de l'algorithme classique permet d'utiliser les méthode Krylov pour accélérer la convergence. Cependant, l'application de  $(I - \mathcal{L})$  au vecteur  $g$  est une opération très coûteuse. Grâce aux propositions 2.3 et 2.4, on propose un nouvel algorithme ci-dessous

---

**Algorithme 3** : Nouvel algorithme
 

---

- 1: Construire  $\mathcal{L}$  et  $d$  dans (2.29) explicitement.
  - 2: Résoudre le système linéaire (2.29) par une méthode itérative.
  - 3: Résoudre l'équation de Schrödinger sur chaque sous-domaine pour chaque pas de temps en utilisant les conditions aux limites issues de l'étape 2.
- 

Nous présentons ci-dessous la construction de la matrice  $\mathcal{L}$  et du vecteur  $d$ . Nous verrons que cette construction n'est pas coûteuse. Concernant la mise en œuvre, nous montrerons ensuite comment  $\mathcal{L}$  et  $d$  sont stockés pour une utilisation du parallélisme. Nous utilisons ici la librairie standard PETSc. Sous cette forme, la mémoire nécessaire pour chaque processus MPI est indépendante du nombre de sous-domaines. On se réfère à l'annexe B sur les notations de MPI et la librairie PETSc.

### 2.4.1 Construction de la matrice $\mathcal{L}$ et du vecteur $d$

On se sert des formules (2.28) et (2.12) pour faire les constructions. Numériquement, on considère  $l_{j,n}^k$  et  $r_{j,n}^k$  comme l'entrée,  $l_{j-1,n}^{k+1}$  et  $r_{j+1,n}^{k+1}$  comme la sortie:

$$\text{entrées: } l_{j,n}^k, r_{j,n}^k \longrightarrow (2.12) \longrightarrow \text{sorties: } l_{j-1,n}^{k+1}, r_{j+1,n}^{k+1}.$$

Il est facile de voir que

$$d = \begin{pmatrix} d_{1,r} \\ d_{2,l} \\ d_{2,r} \\ \vdots \\ d_{N,l} \end{pmatrix} = \mathcal{R} \cdot \mathbf{0}.$$

Les composantes de  $d$  sont obtenues par

$$d_{j-1,r} = \begin{pmatrix} r_{j-1,1}^{k+1} \\ r_{j-1,2}^{k+1} \\ \vdots \\ r_{j-1,N_T}^{k+1} \end{pmatrix}, \quad d_{j+1,l} = \begin{pmatrix} l_{j+1,1}^{k+1} \\ l_{j+1,2}^{k+1} \\ \vdots \\ l_{j+1,N_T}^{k+1} \end{pmatrix},$$

où des quantités scalaires  $r_{j-1,s}^{k+1}, l_{j+1,s}^{k+1}, s = 1, 2, \dots, N_T$  sont données par la formule (2.12) avec

$$l_{j,s}^k = r_{j,s}^k = 0, s = 1, 2, \dots, N_T.$$

L'équation est résolue numériquement sur chaque sous-domaine une fois. Remarquons que cette construction fonctionne pour  $\mathcal{V} = V(t, x)$ .

D'après les propositions 2.4 et 2.3, si  $V = V(x)$ , il suffit pour construire la matrice  $\mathcal{L}$  de calculer la première colonne des blocs  $X^{1,4}, X^{j,1}, X^{j,2}, X^{j,3}, X^{j,4}, j = 2, 3, \dots, N-1$  et  $X^{N,1}$ .

La première colonne de  $X^{j,1}$  est

$$X^{j,1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left( X^{j,1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + X^{j,2} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + d_{j-1,r} \right) - d_{j-1,r} = \begin{pmatrix} r_{j-1,1}^{k+1} \\ r_{j-1,2}^{k+1} \\ \vdots \\ r_{j-1,N_T}^{k+1} \end{pmatrix} - d_{j-1,r}.$$

La première colonne de  $X^{j,3}$  est

$$X^{j,3} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left( X^{j,3} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + X^{j,4} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + d_{j+1,l} \right) - d_{j+1,l} = \begin{pmatrix} l_{j+1,1}^{k+1} \\ l_{j+1,2}^{k+1} \\ \vdots \\ l_{j+1,N_T}^{k+1} \end{pmatrix} - d_{j+1,l}.$$

Des quantités scalaires  $r_{j-1,s}^{k+1}, l_{j+1,s}^{k+1}, s = 1, 2, \dots, N_T$  sont obtenues par la formule (2.12) avec

$$l_{j,s}^k = r_{j,s}^k = 0, s = 1, 2, \dots, N_T \text{ à l'exception de } l_{j,1}^k = 1.$$

L'équation est résolue numériquement sur le sous-domaine  $\Omega_j$  une fois.

De la même façon, la première colonne de  $X^{j,2}$  et de  $X^{j,4}$  est

$$X^{j,2} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left( X^{j,2} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + X^{j,4} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + d_{j-1,r} \right) - d_{j-1,r} = \begin{pmatrix} r_{j-1,1}^{k+1} \\ r_{j-1,2}^{k+1} \\ \vdots \\ r_{j-1,N_T}^{k+1} \end{pmatrix} - d_{j-1,r},$$

et

$$X^{j,4} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left( X^{j,2} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + X^{j,4} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + d_{j+1,l} \right) - d_{j+1,l} = \begin{pmatrix} l_{j+1,1}^{k+1} \\ l_{j+1,2}^{k+1} \\ \vdots \\ l_{j+1,N_T}^{k+1} \end{pmatrix} - d_{j+1,l},$$

où des quantités scalaires  $r_{j-1,s}^{k+1}, l_{j+1,s}^{k+1}, s = 1, 2, \dots, N_T$  sont obtenues par la formule (2.12), mais avec

$$l_{j,s}^k = r_{j,s}^k = 0, s = 1, 2, \dots, N_T \text{ à l'exception de } r_{j,1}^k = 1.$$

L'équation est résolue numériquement sur le sous-domaine  $\Omega_j$  une fois.

En conclusion, il suffit de résoudre l'équation sur chaque sous-domaine trois fois pour construire explicitement le problème d'interface. La construction n'est pas coûteuse. Au total, l'équation est résolue sur chaque sous-domaine quatre fois dans le nouvel algorithme. On va comparer les deux algorithmes dans la section 2.6.1.

### 2.4.2 Stockage de la matrice $\mathcal{L}$ et du vecteur $d$

On stocke la matrice transposée de  $\mathcal{L}$  de manière distribuée en utilisant la librairie PETSc. Comme nous le voyons ci-dessous, le première bloc colonne de  $\mathcal{L}$  est affecté au processus MPI 0. La deuxième et la troisième colonnes bloc sont liées au processus MPI 1, et ainsi de suite. La mémoire consommée pour chaque processus est au maximum la somme de 4 blocs. La taille de chaque bloc est  $N_T \times N_T$ . Chaque bloc contient  $(N_T + 1) \times N_T/2$  éléments non nuls d'après les propositions 2.2 et 2.1.

$$\mathcal{L} = \begin{pmatrix} \overbrace{\phantom{X^{1,4}}}^{\text{MPI 0}} & \overbrace{\phantom{X^{2,1} \quad X^{2,2}}}^{\text{MPI 1}} & \overbrace{\phantom{X^{3,1} \quad X^{3,2}}}^{\text{MPI 2}} & & \overbrace{\phantom{X^{N-1,1} \quad X^{N-1,2}}}^{\text{MPI } N-2} & \overbrace{\phantom{X^{N,1}}}^{\text{MPI } N-1} \\ X^{1,4} & X^{2,1} & X^{2,2} & & & \\ & X^{2,3} & X^{2,4} & X^{3,1} & X^{3,2} & \\ & & & \dots & & \\ & & X^{3,3} & X^{3,4} & & \\ & & & & X^{N-1,1} & X^{N-1,2} \\ & & & & \dots & \\ & & & & X^{N-1,3} & X^{N-1,4} & X^{N,1} \end{pmatrix}. \quad (2.30)$$

Le vecteur  $d$  est aussi stocké sous la forme PETSc. Le premier bloc est dans le processus 0, le deuxième et le troisième sont dans le processus 1, etc. Le dernier bloc est dans le processus  $N$ . Chaque processus contient au maximum  $2 \times N_T$  éléments.

$$d = 2 \begin{pmatrix} d_{1,r} \\ \vdots \\ d_{j,l} \\ d_{j,r} \\ \vdots \\ d_{N,l} \end{pmatrix} \begin{array}{l} \} \text{MPI 0} \\ \\ \} \text{MPI } j-1 \\ \\ \} \text{MPI } N-1 \end{array}$$

## 2.5 Convergence

Nous rappelons d'abord la définition standard de l'espace complexe muni de la norme  $l^2$

$$l_n^2(\mathbb{C}) := \left\{ (x_1, x_2, \dots, x_n) \in \mathbb{C}^n \mid \|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{1/2} \right\}.$$

On va montrer que si  $g_j^k$ ,  $j = 1, 2, \dots, N$ , converge vers la limite (notée  $g_j$ ), alors pour tout  $n = 0, 1, \dots, N_T - 1$ ,

- $\mathbf{v}_{j,n}^k$  converge vers la limite notée  $\mathbf{v}_{j,n}$ .
- $\mathbf{v}_{j,n}$  est la restriction de  $\mathbf{v}_{0,n}$  sur  $\Omega_j$ .

**Proposition 2.5.** *Avec la condition de transmission mettant en jeu l'opérateur  $S_0^2$ , et sous l'hypothèse que pour  $j = 1, 2, \dots, N$ ,  $\mathbb{A}_{j,s} - \mathbb{B}_{j,s}$ ,  $s = 1, 2, \dots, N_T$  ne sont pas singulière, si  $g^k$  converge dans  $l_{(2N-1)N_T}^2(\mathbb{C})$ , alors  $\mathbf{v}_j^k$  converge dans  $l_{N_j}^2(\mathbb{C})$ .*

*Preuve.* Considérons (2.23), pour  $n = 1, \dots, N_T$ , on a

$$\begin{aligned} \|\mathbf{v}_{j,n}^{k+1} - \mathbf{v}_{j,n}^k\|_{l_{N_j}^2} &\leq \|\mathbb{Y}_{n,n}\|_2 \cdot \left\| Q_j^T (l_{j,n}^{k+1} - l_{j,n}^k, r_{j,n}^{k+1} - r_{j,n}^k)^T \right\|_{l_{N_j}^2} \\ &\quad + \sum_{q=1}^{n-1} \|\mathbb{Y}_{n,q}\|_2 \|\mathbf{v}_{j,q}^{k+1} - \mathbf{v}_{j,q}^k\|_{l_{N_j}^2}. \end{aligned}$$

Ainsi, la preuve est réalisée par récurrence sur  $n$ .  $\square$

**Proposition 2.6.** *Sous une condition de transmission avec  $S$  donnée dans la sous-section 2.1.2, sous l'hypothèse que pour  $j = 1, 2, \dots, N$ ,  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$  n'est pas singulière, si  $g^k$  converge dans  $l_{(2N-1)N_T}^2(\mathbb{C})$ , alors  $\mathbf{v}_j^k$  converge dans  $l_{N_j}^2(\mathbb{C})$ .*

*Preuve.* La preuve est quasiment la même que celle de la proposition 2.5 à l'exception que les définitions des matrices  $\mathbb{Y}_{j,n}^j$  sont différents.  $\square$

Notons par  $\mathbf{v}_{j,n+1}$  (resp.  $\mathbf{u}_{j,n+1}$ ) la limite de  $\mathbf{v}_{j,n+1}^k$  (resp.  $\mathbf{u}_{j,n+1}^k$ ).

**Proposition 2.7.** *Avec la condition de transmission mettant en jeu l'opérateur  $S_0^2$ , sous l'hypothèse que pour  $j = 1, 2, \dots, N$ ,  $\mathbb{A}_{j,s} - \mathbb{B}_{j,s}$ ,  $s = 1, 2, \dots, N_T$  n'est pas singulière et  $g^k$  converge vers  $g$ , alors  $\mathbf{v}_{j,n}$  satisfait*

$$\mathbf{v}_{j,n} = R_j \mathbf{v}_{0,n},$$

pour  $n = 1, \dots, N_T - 1$ .  $R_j \in \mathbb{R}^{N_0 \times N_j}$  est la matrice de restriction de  $]a_0, b_0[$  à  $]a_j, b_j[$ .

*Preuve.* Sous les hypothèses, en utilisant la proposition 2.6, et à partir de (2.12), on a

$$\begin{cases} l_{j+1,n} = -r_{j,n} + 2\tilde{S}(Q_{j,r}\mathbf{v}_{j,n}), & j = 1, 2, \dots, N-1, \end{cases} \quad (2.31a)$$

$$\begin{cases} r_{j-1,n} = -l_{j,n} + 2\tilde{S}(Q_{j,l}\mathbf{v}_{j,n}), & j = 2, 3, \dots, N, \end{cases} \quad (2.31b)$$

$$\begin{cases} (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n} = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1} + \mathbf{b}_{j,n} - Q_j^T(l_{j,n}, r_{j,n})^T. \end{cases} \quad (2.31c)$$

Pour  $n = 1, 2, \dots, N_T$ ,

$$l_{j+1,n} + r_{j,n} = 2\tilde{S}(Q_{j,r}\mathbf{v}_{j,n}) = 2\tilde{S}(Q_{j+1,l}\mathbf{v}_{j+1,n}).$$

Alors

$$Q_{j,r}\mathbf{v}_{j,n} = Q_{j+1,l}\mathbf{v}_{j+1,n},$$

$$Q_{j,r}\mathbf{u}_{j,n} = Q_{j+1,l}\mathbf{u}_{j+1,n}.$$

Autrement dit,  $\mathbf{v}_{j,n}$  et  $\mathbf{v}_{j+1,n}$  (resp.  $\mathbf{u}_{j,n}$  et  $\mathbf{u}_{j+1,n}$ ) prennent la même valeur sur l'interface  $x = b_j$  à l'étape du temps  $n, n = 1, \dots, N_T$ . Il existe un et un seul vecteur  $\mathbf{w}_n$  qui satisfait

$$\mathbf{v}_{j,n} = R_j\mathbf{w}_n.$$

En multipliant (2.31a) par  $R_j^T Q_{j,r}^T$  et en sommant de 1 à  $N-1$ , on obtient

$$\sum_{j=1}^{N-1} R_j^T Q_{j,r}^T l_{j+1,n} + \sum_{j=1}^{N-1} R_j^T Q_{j,r}^T r_{j,n} = 2R_j^T Q_{j,r}^T \sum_{j=1}^{N-1} \tilde{S}(Q_{j,r}\mathbf{v}_{j,n}). \quad (2.32)$$

En multipliant (2.31b) par  $R_j^T Q_{j,l}^T$  et sommant de 1 à  $N$ , on a

$$\sum_{j=2}^N R_j^T Q_{j,l}^T l_{j,n} + \sum_{j=2}^N R_j^T Q_{j,l}^T r_{j-1,n} = 2R_j^T Q_{j,l}^T \sum_{j=2}^N \tilde{S}(Q_{j,l}\mathbf{v}_{j,n}). \quad (2.33)$$

L'addition de (2.32) et (2.33) donne

$$\sum_{j=2}^N R_j^T Q_{j,l}^T (\tilde{S}Q_{j,l}\mathbf{v}_{j,n}) + \sum_{j=1}^{N-1} R_j^T Q_{j,r}^T (\tilde{S}Q_{j,r}\mathbf{v}_{j,n}) = \sum_{j=2}^N R_j^T Q_{j,l}^T l_{j,n} + \sum_{j=1}^{N-1} R_j^T Q_{j,r}^T r_{j,n}, \quad (2.34)$$

où on utilise

$$R_j^T Q_{j,r}^T = R_{j+1}^T Q_{j+1,l}^T, \quad j = 1, 2, \dots, N-1.$$

En multipliant (2.31c) par  $R_j^T$  et sommant de 0 à  $N$ , on obtient

$$\begin{aligned} \sum_{j=1}^N (R_j^T \mathbb{A}_{j,n} R_j) \mathbf{w}_n &= \frac{2i}{\Delta t} \sum_{j=1}^N (R_j^T \mathbb{M}_j) \mathbf{u}_{j,n-1} + \sum_{j=2}^N R_j^T Q_{j,l}^T (\tilde{S}Q_{j,l}\mathbf{v}_{j,n}) \\ &\quad + \sum_{j=1}^{N-1} R_j^T Q_{j,r}^T (\tilde{S}Q_{j,r}\mathbf{v}_{j,n}) - \sum_{j=2}^N R_j^T Q_{j,l}^T l_{j,n} - \sum_{j=1}^{N-1} R_j^T Q_{j,r}^T r_{j,n}, \end{aligned}$$

alors d'après (2.34) et la définition de  $\mathbf{w}_n$ , on a

$$\sum_{j=1}^N (R_j^T \mathbb{A}_{j,n} R_j) \mathbf{w}_n = \frac{2i}{\Delta t} \sum_{j=1}^N (R_j^T \mathbb{M}_j) \mathbf{u}_{j,n}. \quad (2.35)$$

De plus, sur le domaine entier  $]a_0, b_0[$ ,

$$\sum_{j=1}^N (R_j^T \mathbb{A}_{j,n} R_j) \mathbf{v}_{0,n} = \mathbb{A}_0 \mathbf{v}_{0,n} = \frac{2i}{\Delta t} \mathbb{M}_0 \mathbf{u}_{0,n} = \frac{2i}{\Delta t} \sum_{j=1}^N (R_j^T \mathbb{M}_j R_j) \mathbf{u}_{0,n-1}. \quad (2.36)$$

Par récurrence, on va montrer

$$R_j \mathbf{u}_{0,n} = \mathbf{u}_{j,n}, \quad R_j \mathbf{v}_{0,n} = \mathbf{v}_{j,n}, \quad n = 0, 1, 2, \dots, N_T. \quad (2.37)$$

Si  $n = 0$ , alors la formule (2.37) est juste. Si on suppose que la formule (2.37) est juste pour  $n = 0, 1, \dots, s$ , alors la comparaison des formules (2.35) et (2.36) donne

$$\mathbf{w}_{s+1} = \mathbf{v}_{0,s+1}.$$

Puis

$$R_j \mathbf{v}_{0,s+1} = \mathbf{v}_{j,s+1}, \quad R_j \mathbf{u}_{0,s+1} = \mathbf{u}_{j,s+1}.$$

□

## 2.6 Résultats Numériques

Le domaine  $]a_0, b_0[ = ]-21, 21[$  est décomposé en  $N$  sous-domaines égaux. On fixe pour tous les résultats le temps final à  $T = 0.5$  et le pas de temps à  $\Delta t = 0.001$ . La donnée initiale est une gaussienne

$$u_0(x) = e^{-(x+10)^2 + 20i(x+10)},$$

qui donne lieu à une solution qui se déplace sur la droite et subit une dispersion. On considère ici deux potentiels. Le premier est  $V = x^2$  ce qui constitue un potentiel répulsif (sous-entendu à l'extérieur de  $]a_0, b_0[$ ). L'autre est  $V = -x^2$ , qui lui est attractif. Le potentiel attractif a pour effet de localiser la solution à l'intérieur du domaine. La condition d'arrêt pour les méthodes SWR est  $\|g^{k+1} - g^k\| < 10^{-10}$ .

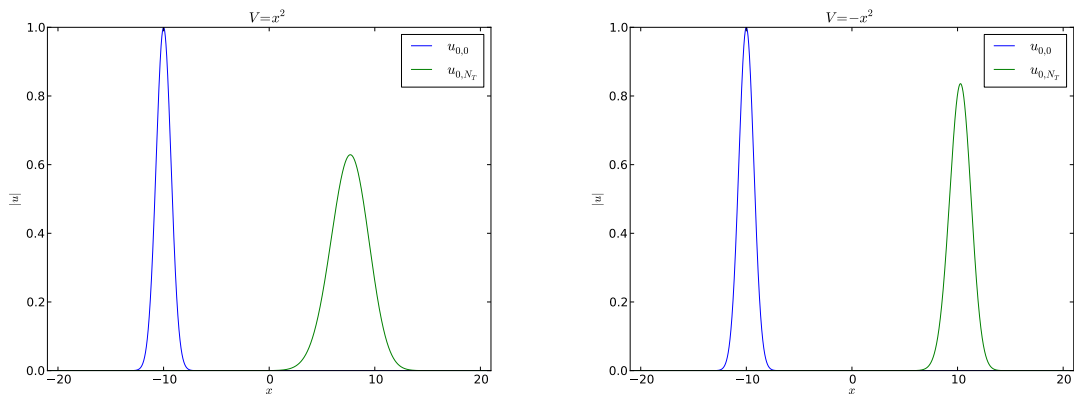


FIGURE 2.3:  $|u_{0,0}|$  et  $|u_{0,N_T}|$  sur  $]a_0, b_0[$ ,  $V = x^2$  (gauche) et  $V = -x^2$  (droite),  $\Delta x = 10^{-5}$ .

Comme les matrices  $\mathbb{M}_j$ ,  $\mathbb{S}_j$  et  $\mathbb{M}_{j,V}$  sont tri-diagonales symétriques en dimension une, la mémoire consommée est faible. Il est ainsi possible de résoudre numériquement l'équation de Schrödinger sur le domaine complet  $]0, T[ \times ]a_0, b_0[$  avec une machine standard (ici, nous utilisons un nœud de calcul doté de 32 Go de mémoire). Les modules des solutions au temps final  $t = T$  sont présentées dans la figure 2.3.

Pour la mise en œuvre des algorithmes de la méthode SWR, nous disposons d'un cluster composé de 92 nœuds dotés chacun de 16 cœurs et de 32 Go de mémoire. On utilise 1 processus MPI par sous-domaine et 16 processus MPI par nœud. Les communications MPI sont gérées par PETSc et Intel MPI. Les systèmes linéaires (2.13) liés aux équations de Schrödinger sont résolus par méthode directe LU. Nous utilisons la bibliothèque MKL Pardiso. Nous présentons sur les figures 2.4 et 2.5 l'organisation d'un calcul distribué pour les deux algorithmes de la méthode SWR pour 3 sous-domaines. Nous avons également réalisé des tests sur un cluster de carte graphiques GPU (voir C).

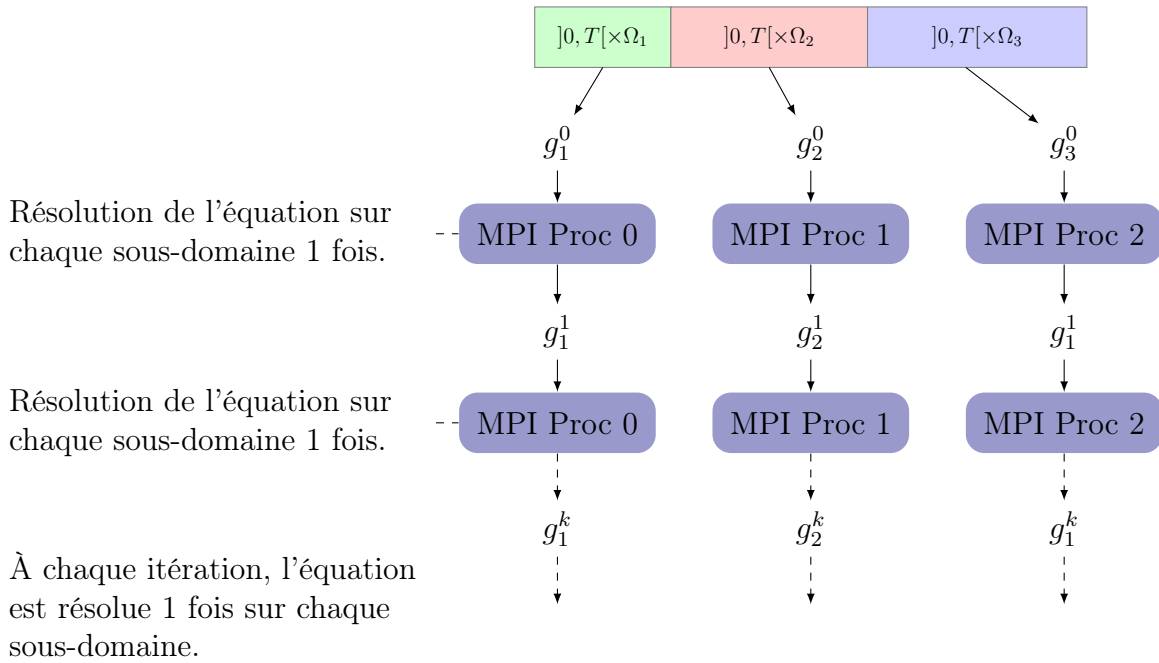


FIGURE 2.4: Mise en œuvre par calcul distribué de l'algorithme classique SWR, pour  $N = 3$  sous-domaines.

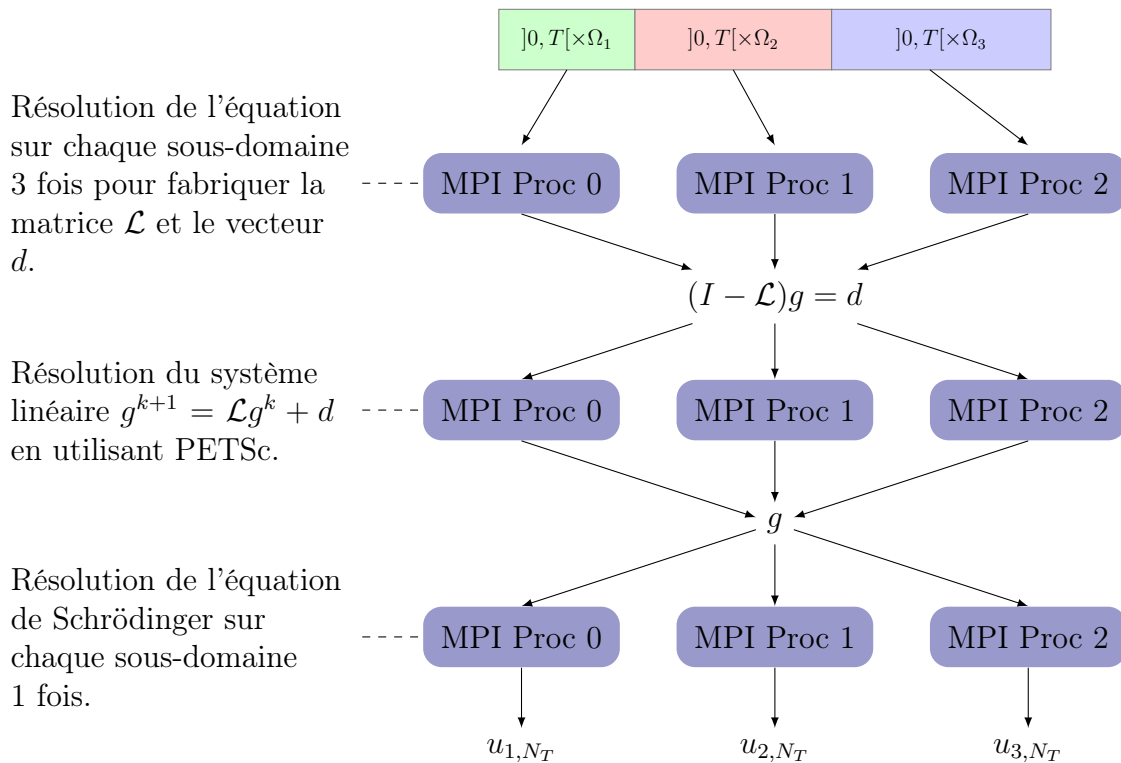


FIGURE 2.5: Mise en œuvre par calcul distribué du nouvel algorithme SWR, pour  $N = 3$  sous-domaines.



### 2.6.1 Comparaison des algorithmes classiques et nouveaux

Nous nous intéressons dans cette partie à observer la robustesse des algorithmes, à savoir s'ils convergent ou non pour les deux potentiels envisagés. De même, nous observerons les temps de calcul et la scalabilité forte du calcul distribué des deux algorithmes.

Nous définissons ici

- $T^{\text{ref}}$ : le temps de calcul sur un processeur de la résolution numérique de l'équation de Schrödinger sur le domaine complet  $]0, T[\times]a_0, b_0[$ ,
- $T^{\text{old}}$ : le temps de calcul obtenu avec l'algorithme SWR classique sur  $N$  sous-domaines,
- $T^{\text{new}}$ : le temps de calcul obtenu avec le nouvel algorithme SWR pour  $N$  sous-domaines.

#### 2.6.1.1 Cas du potentiel $V = x^2$

Nous avons testé les algorithmes pour  $N = 2$ ,  $N = 10$  et  $N = 100$  sous-domaines et la condition de transmission mettant en jeu  $S_0^2$ . Nous expliquerons la raison d'utiliser  $S_0^2$  à la fin de la sous-section 2.6.2. Nous présentons dans la figure 2.6 et le tableau 2.1 l'historique de la convergence et les temps de calcul pour  $V = x^2$  pour la méthode SWR utilisant la méthode du point fixe sur le problème d'interface.

Nous constatons que si le nombre de sous-domaines  $N \geq 100$ , alors le nouvel algorithme et l'algorithme classique ne convergent pas tous (avant 2000 itérations).

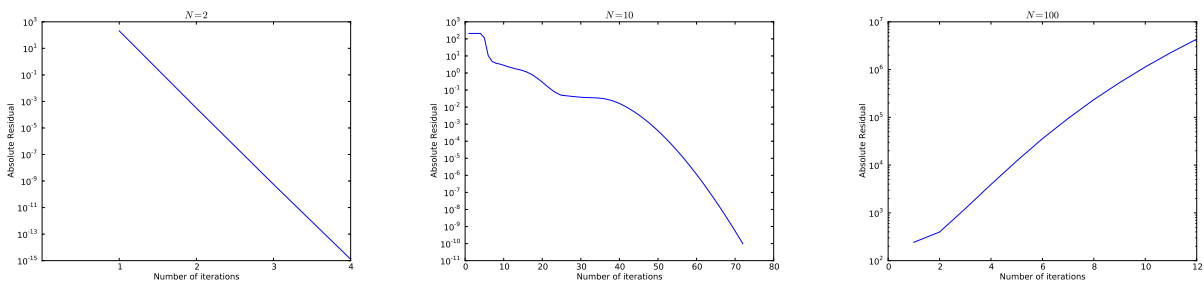


FIGURE 2.6: Historique de la convergence,  $N = 2, 10, 100$ ,  $S_0^2$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ .

$N$	2	10	100
$T^{\text{ref}}$	400.83		
$T^{\text{old}}$	968.64	3242.65	-
$T^{\text{new}}$	775.98	177.88	-

-: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 2.1: Temps de calcul en secondes,  $S_0^2$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ .

Nous utilisons ensuite la version des algorithmes mettant en œuvre les méthodes de Krylov (Gmres ou Bicgstab) sur le problème d'interface au lieu de la méthode de point fixe. Les résultats sont présentés dans les tableaux 2.2 et 2.3.

$N$	2	10	100	500	1000
$T^{\text{ref}}$	400.83				
$T^{\text{old}}$	966.59	2754.16	2217.59	-	-
$T^{\text{new}}$	778.40	176.57	19.21	9.21	-

-: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 2.2: Temps de calcul en secondes,  $S_0^2$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ , Gmres.

$N$	2	10	100	500	1000
$T^{\text{ref}}$	400.83				
$T^{\text{old}}$	1157.78	3027.63	876.43	1087.63	1227.38
$T^{\text{new}}$	773.79	176.91	18.19	7.71	8.71

TABLEAU 2.3: Temps de calcul en secondes,  $S_0^2$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ , Bicgstab.

Comme on peut le constater, l'utilisation des méthodes de Krylov permet d'obtenir maintenant une convergence des méthodes SWR. Les algorithmes sont donc plus robustes. Ils convergent pour 1000 sous-domaines en utilisant la méthode Bicgstab, pour 500 sous-domaines avec la méthode Gmres. L'algorithme classique et le nouvel algorithme sont équivalents en théorie, mais à cause de la précision numérique, nous constatons dans le tableau 2.2, que pour  $N = 500$ , le nouvel algorithme conduit à une méthode convergente, ce qui n'est pas le cas pour le classique.

Par rapport à la scalabilité et au temps de calcul, le nouvel algorithme est scalable jusqu'à 500 sous-domaines en utilisant la méthode Gmres ou Bicgstab, et surtout prend beaucoup moins de temps que l'algorithme classique. Nous avons grosso modo

$$\begin{aligned} T^{\text{old}} &= T_{\text{sub}} \times N_{\text{iter}} + \dots, \\ T^{\text{new}} &= T_{\text{sub}} \times 4 + T_{Ld} + \dots, \end{aligned}$$

où  $T_{\text{sub}}$  est le temps pour résoudre l'équation sur un sous-domaine,  $T_{Ld}$  est le temps pour résoudre le problème d'interface, "... représente la partie négligeable du temps de calcul comme la construction de matrices pour la méthode d'éléments finis, etc.

- Si le nombre de sous-domaines  $N$  n'est pas trop grand,  $T_{\text{sub}} \gg T_{Ld}$  et le minimum de  $N_{\text{iter}}$  est 3 dans tous les tests.
- Si le nombre de sous-domaines  $N$  est grand,  $T_{Ld} \sim T_{\text{sub}}$  et  $N_{\text{iter}} \gg 4$ . C'est la raison pour laquelle le nouvel algorithme est plus performant et prend moins de temps de calcul. Cependant,  $T_{Ld}$  grandit en augmentant le nombre de sous-domaines. À partir d'un très grand nombre de sous-domaines, le nouvel algorithme perd en scalabilité. Le nombre de communications MPI augmente également ce qui impacte la performance.

L'algorithme classique n'est pas scalable du fait que le nombre d'itérations est plus important pour plus de sous-domaines.

### 2.6.1.2 Cas du potentiel $V = -x^2$

Dans cette partie, on compare les deux algorithmes pour le potentiel  $V = -x^2$ . L'historique de la convergence et les temps de calcul sont montrés dans la figure 2.7 et le tableau 2.4 où l'on utilise la méthode de point fixe sur le problème d'interface. L'algorithme converge pour 100 sous-domaines, mais pas pour  $N = 500, 1000$ .

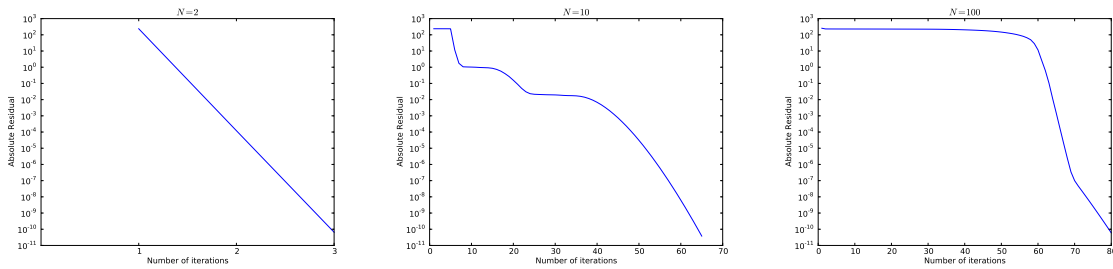


FIGURE 2.7: Historique de la convergence,  $N = 2, 10, 100$ ,  $S_0^2$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ .

$N$	2	10	100
$T^{\text{ref}}$	403.56		
$T^{\text{old}}$	773.07	2937.77	359.30
$T^{\text{new}}$	773.72	178.30	18.19

TABLEAU 2.4: Temps de calcul en secondes,  $S_0^2$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ .

On montre dans les tableaux 2.5 et 2.6 les mêmes résultats avec les méthodes Gmres et Bicgstab. On peut voir que

- Les algorithmes sont maintenant robustes pour 1000 sous-domaines.
- Ils sont scalables jusqu'à 500 sous-domaines.
- Le nouvel algorithme permet un très gros gain en temps de calcul comparé à l'algorithme classique.

$N$	2	10	100	500	1000
$T^{\text{ref}}$	403.56				
$T^{\text{old}}$	771.82	2577.51	2249.54	907.06	739.65
$T^{\text{new}}$	777.42	177.20	18.95	6.86	8.17

TABLEAU 2.5: Temps de calcul en secondes,  $S_0^2$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ , Gmres.

$N$	2	10	100	500	1000
$T^{\text{ref}}$	403.56				
$T^{\text{old}}$	774.19	2760.11	679.72	799.09	845.65
$T^{\text{new}}$	774.44	177.02	18.18	6.83	7.12

TABLEAU 2.6: Temps de calcul en secondes,  $S_0^2$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ , Bicgstab.

## 2.6.2 Comparaison des conditions de transmission

Nous comparons ici les conditions de transmission qui sont présentées dans la section 2.1.2 ainsi que la condition Robin dans le cadre du nouvel algorithme. Le choix théorique du paramètre  $p$  dans la condition de transmission Robin n'étant pas connu pour nous, nous cherchons le meilleur paramètre numériquement. Nous montrons d'abord les résultats pour  $N = 2$  avec les méthodes de point fixe, Gmres et Bicgstab. Nous faisons ensuite des comparaisons pour  $N = 500$ .

Dans (2.17), on utilise un vecteur aléatoire comme vecteur initial pour assurer que tous les fréquences sont présentes [31, 45]. D'après notre tests, l'algorithme initialisé avec un vecteur de zéros converge plus vite qu'avec un vecteur aléatoire. Ainsi il prend moins de temps de calcul. Cependant dans cette sous-section, notre but est de comparer des conditions de transmission. Comme cela est expliqué dans [31], une initialisation avec un vecteur nul pour calculer une solution lisse fait que l'erreur ne contient que des basses fréquences, ceci à cause de la linéarité. Ainsi, nous pourrions tirer de fausses conclusions. Un exemple est montré dans [31]. Dans cette thèse, nous utilisons donc un vecteur aléatoire comme vecteur initial si nous nous intéressons à la comparaison des conditions de transmission et à l'effet du maillage. Le vecteur nul est utilisé si nous considérons plutôt le temps de calcul.

### 2.6.2.1 $N = 2$ , $V = x^2$

Nous considérons d'abord le potentiel répulsif ( $V = x^2$ ). Nous comparons le nombre d'itérations, le temps total pour réaliser une simulation complète à convergence et le temps requis ( $T_{Ld}$ ) pour résoudre le problème d'interface dans le tableau 2.7 pour les différentes implémentations point fixe, Gmres et Bicgstab. Comme on peut le remarquer, les temps totaux sont tous quasiment identiques. Le temps de résolution du problème d'interface  $T_{Ld}$  est presque nul relativement au temps total de simulation. On s'intéresse donc plutôt au nombre d'itérations. On peut faire les constats suivants

- le nombre d'itérations requises concernant la condition de transmission de Robin, dont nous rappelons que nous avons cherché un optimal numérique pour chaque méthode itérative (ici  $p = 44$  pour le point fixe et  $p = 6$  pour Gmres et Bicgstab), est plus important que pour les trois autres stratégies,
- dans chaque stratégie, le nombre d'itérations n'est quasiment pas sensible à l'ordre,

- pour la stratégie d'approximation de Padé, le nombre d'itérations diminue en augmentant le paramètre de Padé ( $m$ ), soit le nombre de fonctions de Padé pour l'approximation de la racine carrée.

Stratégie		Point fixe			Gmres			Bicgstab		
		$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$
$S_0^M$	$S_0^2$	6	0.006	774.36	5	0.005	774.23	3	0.002	774.08
	$S_0^3$	6	0.002	773.17	5	0.002	771.36	3	0.002	773.81
	$S_0^4$	6	0.002	777.93	5	0.002	773.49	3	0.002	773.78
$S_1^M$	$S_1^2$	6	0.006	775.31	5	0.005	773.48	3	0.002	773.83
	$S_1^4$	6	0.002	773.99	5	0.002	773.93	3	0.002	773.53
$S_2^M$	$S_2^{2,20}$	191	0.062	773.60	28	0.010	773.95	15	0.010	775.20
	$S_2^{2,50}$	77	0.025	777.57	27	0.010	773.15	15	0.010	773.28
	$S_2^{2,100}$	39	0.013	773.92	23	0.008	774.15	13	0.009	773.71
	$S_2^{4,20}$	191	0.062	773.41	28	0.010	776.95	15	0.010	773.53
	$S_2^{4,50}$	76	0.025	773.49	27	0.010	775.39	15	0.010	773.85
	$S_2^{4,100}$	39	0.013	773.97	23	0.008	773.17	13	0.008	773.96
Robin*		1109	0.340	773.16	44	0.021	775.17	26	0.023	775.05

\* les paramètres de Robin sont  $p = 44$  (point fixe),  $p = 6$  (Gmres) et  $p = 6$  (Bicgstab).

TABLEAU 2.7: Comparaison des conditions de transmission pour  $N = 2$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ .

### 2.6.2.2 $N = 2$ , $V = -x^2$

Nous reproduisons les mêmes tests que dans la section précédente. Les résultats sont présentés dans le tableau 2.8. Les conclusions sont tout à fait similaires.

Stratégie		Point fixe			Gmres			Bicgstab		
		$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$
$S_0^M$	$S_0^2$	6	0.005	775.71	5	0.002	774.17	3	0.002	773.78
	$S_0^3$	6	0.002	774.21	5	0.002	779.58	3	0.002	773.32
	$S_0^4$	6	0.002	769.00	5	0.002	774.17	3	0.002	773.64
$S_1^M$	$S_1^2$	6	0.002	773.38	5	0.002	773.24	3	0.002	773.81
	$S_1^4$	6	0.002	773.85	5	0.002	773.59	3	0.002	774.49
$S_2^M$	$S_2^{2,20}$	191	0.062	773.30	28	0.010	774.53	16	0.011	773.10
	$S_2^{2,50}$	76	0.025	773.64	27	0.010	773.30	15	0.010	773.57
	$S_2^{2,100}$	39	0.013	776.31	23	0.008	775.21	13	0.009	773.60
	$S_2^{4,20}$	181	0.059	769.91	28	0.010	774.55	15	0.010	773.57
	$S_2^{4,50}$	77	0.025	775.99	27	0.010	773.45	15	0.010	773.30
	$S_2^{4,100}$	39	0.013	775.42	23	0.008	773.77	13	0.009	774.78
Robin*		1112	0.360	774.71	47	0.017	776.37	27	0.018	777.39

\* les paramètres de Robin sont  $p = 44$  (point fixe),  $p = 5$  (Gmres) et  $p = 5$  (Bicgstab).

TABLEAU 2.8: Comparaison des conditions de transmission pour  $N = 2$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ .

### 2.6.2.3 $N = 500$ , $V = x^2$

Nous faisons des comparaisons pour  $N = 500$ ,  $V = x^2$  dans cette sous-section. Le tableau 2.9 présente les résultats. Nous pouvons constater que

- l'algorithme n'est pas robuste pour la stratégie potentiel,
- le nombre d'itérations n'est pas sensible à l'ordre dans chaque stratégie,
- le nombre d'itérations diminue en augmentant le paramètre de Padé ( $m$ ) pour la stratégie d'approximation de Padé,
- pour l'implémentation point fixe, seule la stratégie d'approximation Padé en ordre élevé et celle de Robin permettent de faire converger l'algorithme,
- le nombre d'itérations requises concernant Bicgstab est moindre que celui pour le point fixe et Gmres.

		Point fixe			Gmres			Bicgstab		
Stratégie		$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$
$S_0^M$	$S_0^2$	-			-			-		
	$S_0^3$	-			-			-		
	$S_0^4$	-			-			-		
$S_1^M$	$S_1^2$	-			1179	3.197	7.08	381	1.974	5.85
	$S_1^4$	-			1243	3.376	7.31	378	1.664	5.48
$S_2^M$	$S_2^{2,20}$	-			1101	3.965	7.98	521	2.828	6.80
	$S_2^{2,50}$	-			922	2.855	7.05	368	2.248	6.30
	$S_2^{2,100}$	377	0.82	4.87	895	2.257	6.10	340	1.637	5.49
	$S_2^{4,20}$	-			1093	3.685	7.62	511	2.445	6.32
	$S_2^{4,50}$	-			908	2.442	6.53	346	2.008	6.01
	$S_2^{4,100}$	374	0.816	4.87	871	2.420	6.35	342	1.783	5.83
Robin*		1421	3.037	7.00	990	2.673	6.42	359	1.864	5.89

\*: les paramètres de Robin sont  $p = 40$  (point fixe),  $p = 24$  (Gmres) et  $p = 40$  (Bicgstab).  
 -: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 2.9: Comparaison des conditions de transmission pour  $N = 500$ ,  $V = x^2$ ,  $\Delta x = 10^{-5}$ .

Remarquons que si le vecteur initial est le vecteur nul, la stratégie potentiel est robuste (voir tableau 2.10).

		$N_{iter}$	$T_{Ld}$	$T_{total}$
Gmres	$S_0^2$	1790	5.408	9.21
	$S_0^3$	930	2.890	6.90
	$S_0^4$	920	2.444	6.91
Bicgstab	$S_0^2$	607	3.884	7.71
	$S_0^3$	467	2.430	6.44
	$S_0^4$	445	2.528	6.54

TABLEAU 2.10: Nombre d'itérations et temps de calcul en second pour le stratégie potentiel avec le vecteur initial zéro,  $N = 500$ ,  $V = x^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

#### 2.6.2.4 $N = 500$ , $V = -x^2$

Nous faisons les mêmes tests que dans la section précédente pour  $V = -x^2$ . On peut voir que

- pour la stratégie d'approximation Padé en ordre bas, l'algorithme n'est pas robuste,
- le nombre d'itérations requises n'est pas sensible à l'ordre dans chaque stratégie,

- les méthodes Krylov (Gmres et Bicgstab) ne peuvent pas diminuer toujours le nombre d'itérations.

Stratégie		Point fixe			Gmres			Bicgstab		
		$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$	$N_{iter}$	$T_{Ld}$	$T_{total}$
$S_0^M$	$S_0^2$	357	0.775	4.68	1023	2.883	6.91	368	1.646	5.51
	$S_0^3$	337	0.734	4.62	977	2.620	6.55	345	1.831	5.77
	$S_0^4$	337	0.733	4.65	978	2.681	6.54	350	1.739	5.73
$S_1^M$	$S_1^2$	341	0.745	4.62	1010	2.364	6.20	353	2.102	6.00
	$S_1^4$	340	0.743	4.63	1023	3.454	7.19	351	2.225	6.06
$S_2^M$	$S_2^{2,20}$	-	-	-	1240	3.368	7.34	440	2.626	6.64
	$S_2^{2,50}$	-	-	-	997	2.320	6.30	352	2.240	6.16
	$S_2^{2,100}$	336	0.735	4.62	998	3.055	7.03	333	1.603	5.62
	$S_2^{4,20}$	-	-	-	1216	3.349	7.31	464	2.044	6.05
	$S_2^{4,50}$	-	-	-	1043	3.907	7.85	336	1.756	5.63
	$S_2^{4,100}$	336	0.733	4.60	1024	2.424	6.35	334	1.989	5.95
Robin*		1690	3.628	7.52	1060	3.000	6.80	318	1.41	5.32

\*: les paramètres de Robin sont  $p = 45$  (point fixe),  $p = 19$  (Gmres) et  $p = 6$  (Bicgstab).  
 -: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 2.11: Comparaison des conditions de transmission pour  $N = 500$ ,  $V = -x^2$ ,  $\Delta x = 10^{-5}$ .

Nous terminons cette sous-section concernant la comparaison des conditions de transmission par une petite conclusion. Si le nombre de sous-domaines  $N$  n'est pas très grand, le stratégie potentiel en ordre 2 avec la méthode Bicgstab est un bon choix. Si  $N$  est grand, la méthode Bicgstab permet aussi à la plupart des algorithmes de converger, mais il est difficile d'avoir une conclusion générale pour les conditions de transmission.

*Remarque 2.8.* Comme nous l'avons indiqué précédemment, nous revenons sur notre choix comme condition de transmission standard de la stratégie d'ordre 2 ( $S_0^2$ ). En effet, il nous semble raisonnable la considérer car

- si le vecteur initial est zéro, l'algorithme est robuste et le temps de calcul pour  $S_0^2$  est similaire aux autres méthodes,
- si  $N$  n'est pas très grand, c'est un des meilleurs choix,
- la mise en œuvre de  $S_0^2$  est grandement faciliter par rapport aux autres méthodes

### 2.6.3 Effet du maillage

Nous nous intéressons maintenant à l'effet du maillage dans cette sous-section. Dans le cadre du nouvel algorithme, nous faisons des tests pour des pas de temps  $\Delta t$  et d'espace  $\Delta x$



différents. Le maillage le plus grossier que nous utilisons ( $\Delta x = 10^{-2}$ ) contient 4201 nœuds sur le domaine complet  $]a_0, b_0[$ . La condition de transmission choisie est  $S_0^2$  et le vecteur initial est un vecteur aléatoire. Les résultats sont montrés dans les tableaux 2.12 et 2.13 pour  $V = x^2$  et  $V = -x^2$  respectivement. On peut voir que la convergence ne semble pas sensible à  $\Delta x$ . En revanche, plus  $\Delta t$  est petit, plus le nombre d'itérations augmente.

$\Delta t$	$\Delta x$	Point fixe		Gmres		Bicgstab	
		$N = 10$	$N = 100$	$N = 10$	$N = 100$	$N = 10$	$N = 100$
$10^{-2}$	$10^{-2}$	96	-	87	808	39	95
$10^{-2}$	$10^{-3}$	96	-	83	742	39	100
$10^{-2}$	$10^{-4}$	96	-	83	660	39	99
$10^{-2}$	$10^{-5}$	96	-	85	809	39	104
$10^{-3}$	$10^{-2}$	105	-	110	870	51	130
$10^{-3}$	$10^{-3}$	105	-	117	906	50	129
$10^{-3}$	$10^{-4}$	105	-	114	901	50	136
$10^{-3}$	$10^{-5}$	105	144	115	436	49	82
$10^{-4}$	$10^{-2}$	904	911	408	684	186	273
$10^{-4}$	$10^{-3}$	106	201	102	610	51	202
$10^{-4}$	$10^{-4}$	106	201	104	617	50	193
$10^{-4}$	$10^{-5}$	106	201	102	620	51	200

-: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 2.12: Nombre d'itérations pour  $\Delta t$  et  $\Delta x$  différents,  $V = x^2$ .

$\Delta t$	$\Delta x$	Point fixe		Gmres		Bicgstab	
		$N = 10$	$N = 100$	$N = 10$	$N = 100$	$N = 10$	$N = 100$
$10^{-2}$	$10^{-2}$	96	123	84	91	37	44
$10^{-2}$	$10^{-3}$	95	121	89	97	37	44
$10^{-2}$	$10^{-4}$	96	123	85	98	39	45
$10^{-2}$	$10^{-5}$	96	121	85	93	38	44
$10^{-3}$	$10^{-2}$	105	144	115	425	51	86
$10^{-3}$	$10^{-3}$	105	144	115	425	51	85
$10^{-3}$	$10^{-4}$	105	144	114	437	50	86
$10^{-3}$	$10^{-5}$	105	144	112	444	51	86
$10^{-4}$	$10^{-2}$	904	911	385	694	182	272
$10^{-4}$	$10^{-3}$	106	202	104	637	52	197
$10^{-4}$	$10^{-4}$	106	201	106	629	50	196
$10^{-4}$	$10^{-5}$	106	201	104	628	51	208

TABLEAU 2.13: Nombre d'itérations pour  $\Delta t$  et  $\Delta x$  différents,  $V = -x^2$ .

### 2.6.4 Accélération Gpu

Nous avons montré dans la sous-section 2.6.2 que la résolution de l'équation de Schrödinger sur sous-domaines est la partie la plus coûteuse (dont le temps de est  $\approx T_{total} - T_{Ld}$ ) si le nombre de sous-domaines  $N$  n'est pas très grand. Dans cette sous-section, nous considérons des cartes graphiques (Gpu) pour remplacer les nœuds de calcul classiques (Cpu) pour accélérer les temps de simulations. La figure 2.8 montre cette nouvelle l'implémentation. On résout l'équation sur chaque sous-domaine par un Gpu au lieu d'un Cpu. On utilise 1 Processus MPI / domaine et 1 Gpu / Processus MPI. Du fait que nous ne disposons que de 8 Gpu dans notre calculateur, nous comparons les temps de calcul pour  $N = 2, 4, 8$ . On se réfère à l'annexe C pour les détails sur l'accélération par Gpu et les bibliothèques associées.

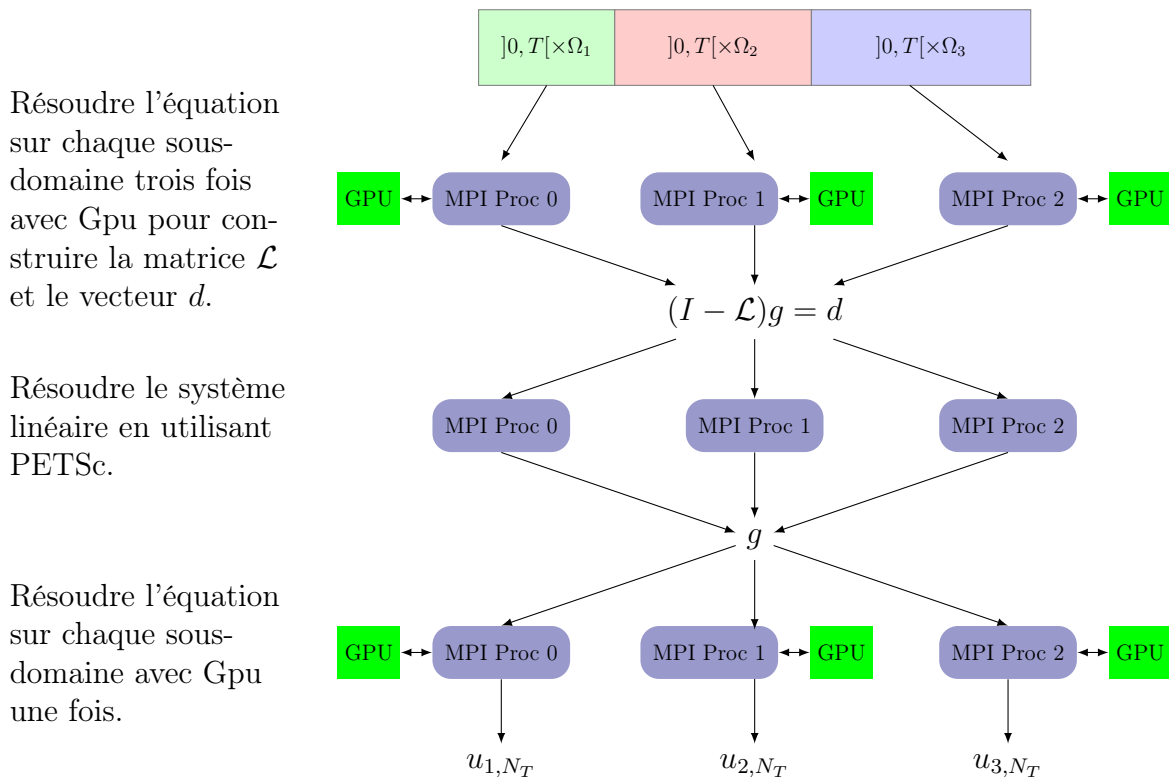


FIGURE 2.8: Implémentation de l'algorithme avec Gpu accélération.

Nous présentons dans les tableaux 2.14 et 2.15 les temps de calcul avec et sans accélération Gpu pour  $V = x^2$  et  $V = -x^2$  respectivement où l'on choisit  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ . La condition de transmission est  $S_0^2$ . Nous utilisons la méthode Bicgstab et le vecteur initial zéro. Nous pouvons voir que les Gpus permettent de diminuer notablement le temps de calcul. Cependant, avec ces données, l'algorithme avec accélération Gpu n'est pas scalable de 4 sous-domaines à 8 sous-domaines. Il y a pour cela deux raisons :

- le nombre d'itérations requises pour  $N = 8$  est légèrement supérieur que celui pour  $N = 4$ ,
- pour  $N = 8$ , la taille du problème sur un sous-domaine n'est ici pas assez grand pour mettre en œuvre toute la puissance des Gpus.

$N$	2	4	8
Temps CPU	773.79	391.43	203.21
Temps GPU	28.02	16.21	13.22
Speed Up	28	24	15

TABLEAU 2.14: Comparaison des temps de calcul sans et avec accélération Gpu,  $V = x^2$ .

$N$	2	4	8
Temps CPU	774.44	392.98	203.18
Temps GPU	27.90	16.13	12.54
Speed Up	18	24	16

TABLEAU 2.15: Comparaison des temps de calcul sans et avec accélération Gpu,  $V = -x^2$ .

Les tableaux 2.16 et 2.17 présentent les résultats de tests similaires pour un maillage plus fin où  $\Delta t = 10^{-3}$ ,  $\Delta x = 5 \times 10^{-6}$ . Comme les temps de calcul sans accélération Gpu pourraient être très longs, nous ne considérons dans ces tests que des temps de calcul avec accélération Gpu. On peut cette fois ci constater que l'algorithme est bien scalable.

$N$	2	4	8
Temps GPU	51.95	28.15	16.35

TABLEAU 2.16: Temps de calcul avec accélération Gpu, Bicgstab,  $S_0^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 5 \times 10^{-6}$ ,  $V = x^2$ .

$N$	2	4	8
Temps GPU	51.95	28.21	16.30

TABLEAU 2.17: Temps de calcul avec accélération Gpu, Bicgstab,  $S_0^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 5 \times 10^{-6}$ ,  $V = -x^2$ .

## 2.7 Conclusion

Notre but dans ce chapitre est de chercher un algorithme parallèle qui est robuste et scalable. Nous avons présenté deux algorithmes : l'algorithme classique (préalablement existant) et le nouvel algorithme (que nous proposons). Après comparaison numérique des deux algorithmes dans le sous-section 2.6.1, il est clair que le nouvel algorithme est meilleur que l'algorithme classique. Il prend en effet beaucoup moins de temps de calcul. En outre, le nouvel algorithme est scalable jusqu'à 500 sous-domaines dans nos tests. Ce nouvel algorithme est basé sur les propriétés du problème d'interface global.

Concernant l'interface entre des sous-domaines, nous avons considéré la condition de transmission de type Robin (issue des travaux de [45]) et trois nouvelles stratégies : la stratégie potentiel, la stratégie de changement de jauge et la stratégie d'approximation de Padé. Elles sont comparées numériquement dans la sous-section 2.6.2 dans le cadre du nouvel algorithme

avec trois méthodes de résolution : point fixe, Gmres et Bicgstab. Les temps de calcul totaux sont très similaires lorsque l'algorithme converge. Du point de vue de la robustesse et du nombre d'itérations requis, Bicgstab est un bon choix. Il n'est pas facile de faire une conclusion générale sur les conditions de transmission. Cependant, nous avons préféré utiliser la condition  $S_0^2$  dans ce chapitre.

Enfin, nos test finaux nous permettent de conclure que des calculs réalisés sur des GPU permettent une forte accélération des calculs.

# Chapitre 3

## Méthode SWR pour l'équation de Schrödinger en une dimension avec un potentiel général

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>53</b>
<b>3.2</b>	<b>Discrétisation</b>	<b>54</b>
<b>3.3</b>	<b>Algorithme avec préconditionneur</b>	<b>58</b>
<b>3.4</b>	<b>Résultats numériques</b>	<b>60</b>
3.4.1	Comparaison des algorithmes sans et avec préconditionneur	61
3.4.2	Conditions de transmission	65
3.4.3	Effet du maillage	68
3.4.4	Accélération de Krylov pour l'équation de Schrödinger linéaire	69
3.4.5	Accélération Gpu	70
<b>3.5</b>	<b>Conclusion</b>	<b>72</b>

---

### 3.1 Introduction

Nous considérons dans ce chapitre la méthode de Schwarz (SWR) pour l'équation de Schrödinger en une dimension d'espace sur un domaine spatial borné  $]a_0, b_0[$  et pour  $t \in ]0, T[$ , mais cette fois avec un potentiel général. L'équation considérée est donc

$$\begin{cases} \mathcal{L}u := (i\partial_t + \partial_{xx} + \mathcal{V})u = 0, & (t, x) \in ]0, T[ \times ]a_0, b_0[, \\ u(0, x) = u_0(x), & x \in ]a_0, b_0[, \\ \partial_n u = 0, & x = a_0, b_0, \end{cases} \quad (3.1)$$

où  $\mathcal{V}$  est un potentiel réel linéaire dépendant du temps et de l'espace ou un potentiel réel non-linéaire

- $\mathcal{V} = V(t, x) \in \mathcal{C}^\infty(]0, T[ \times \mathbb{R})$ ,
- $\mathcal{V} = f(u)$ , ex:  $f(u) = |u|^2$ .

Nous réutilisons les notations et symboles définis dans le chapitre 2. Comme précédemment, nous décomposons le domaine temps-espace  $]0, T[ \times ]a_0, b_0[$  en  $N$  sous-domaines  $]0, T[ \times ]a_j, b_j[$ ,  $j = 1, 2, \dots, N$  sans recouvrement (voir figure 2.1). La méthode SWR donnée par (2.2) dans la section 2.1.1 reste ici valide, mais en considérant maintenant l'opérateur de Schrödinger général  $\mathcal{L} = i\partial_t + \partial_{xx} + \mathcal{V}$ . Nous utilisons dans ce chapitre principalement deux types de conditions de transmission

$$B_j = \partial_{\mathbf{n}_j} + S_j,$$

1. Robin :  $S_j = S_p = -ip$ ,  $p \in \mathbb{R}^+$ ,
2.  $S_0^2$  :  $S_j = S_0^2 = e^{-i\frac{\pi}{4}} \partial_t^{1/2}$ .

Cependant, nous rappelons que nous avons présenté trois familles des conditions de transmission pour  $\mathcal{V} = V(t, x)$  dans la sous-section 2.1.2. Comme ces conditions proviennent de conditions aux limites artificielles, nous pouvons utiliser les travaux [9, 51] pour un potentiel non-linéaire  $\mathcal{V} = f(u)$ . Les auteurs y proposent trois familles des conditions absorbantes : stratégie potentiel ( $S_0^M$ ), stratégie de changement de gauge ( $S_1^M$ ) et stratégie d'approximation de Padé ( $S_2^M$ ) où  $M$  est l'ordre. Formellement, elles sont données par les formules de la sous-section 2.1.2 en remplaçant  $V(t, x)$  par  $f(u)$ . Ainsi, nous proposons des les utiliser ici comme condition de transmission. Nous discuterons leurs qualités et défauts respectifs dans le paragraphe 3.4.2.3. à la fin de la sous-section 3.4.2. Nous pouvons alors nous servir des résultats numériques à bien expliquer.

## 3.2 Discrétisation

La discrétisation de l'équation de Schrödinger linéaire et des conditions de transmission expliquée dans le chapitre 2.2 reste ici valide. On ne présente ici que prise en compte de la nonlinéarité ( $\mathcal{V} = |u|^2$ ). Nous rappelons que l'intervalle de temps  $]0, T[$  est discretisé uniformément en  $N_T$  intervalles de longueur  $\Delta t = T/N_T$ . Dans le cas non-linéaire, un schéma usuel, utilisé pour le calcul de solutions de type soliton, est développé par Durán et Sanz-Serna [28]

$$i \frac{u_{j,n}^k - u_{j,n-1}^k}{\Delta t} + \partial_{xx} \frac{u_{j,n}^k + u_{j,n-1}^k}{2} + f\left(\frac{u_{j,n}^k + u_{j,n-1}^k}{2}\right) \frac{u_{j,n}^k + u_{j,n-1}^k}{2} = 0, \quad 1 \leq n \leq N_T,$$

Comme on peut le remarquer, il s'agit d'une généraliation immédiate du schéma de Crank-Nicolson pour l'équation linéaire. Pour faciliter l'implémentation, on introduit les nouvelles variables  $v_{j,n}^k = \frac{u_{j,n}^k + u_{j,n-1}^k}{2}$  avec  $v_{j,0}^k = u_{j,0}^k$ . Le schéma s'écrit alors

$$2i \frac{v_{j,n}^k}{\Delta t} + \partial_{xx} v_{j,n}^k + f(v_{j,n}^k) v_{j,n}^k = 2i \frac{u_{j,n-1}^k}{\Delta t}. \quad (3.2)$$

La formulation variationnelle pour toutes fonctions test  $\phi \in H^1((a_j, b_j))$  est

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{a_j}^{b_j} v_{j,n}^k \phi dx - \int_{a_j}^{b_j} \partial_x v_{j,n}^k \partial_x \phi + \int_{a_j}^{b_j} f(v_{j,n}^k) v_{j,n}^k \phi dx \\ & + \partial_x v_{j,n}^k(b_j) \phi(b_j) - \partial_x v_{j,n}^k(a_j) \phi(a_j) = \frac{2i}{\Delta t} \int_{a_j}^{b_j} u_{j,n-1}^k \phi dx, \end{aligned} \quad (3.3)$$

L'équation est donc non-linéaire, l'approximation en est faite par une procédure de point fixe. Au temps  $t_n$ , on pose  $\zeta_j^0 = v_{j,n-1}^k$ , et on calcule la solution  $v_{j,n}^k$  comme la limite du schéma itératif en  $s$ :

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{a_j}^{b_j} \zeta_j^{s+1} \phi dx - \int_{a_j}^{b_j} \partial_x \zeta_j^{s+1} \partial_x \phi + \partial_x \zeta_j^{s+1}(b_j) \phi(b_j) - \partial_x \zeta_j^{s+1}(a_j) \phi(a_j) \\ & = \frac{2i}{\Delta t} \int_{a_j}^{b_j} u_{j,n-1}^k \phi dx - \int_{a_j}^{b_j} f(\zeta_j^s) \zeta_j^s \phi dx. \end{aligned} \quad (3.4)$$

Les conditions de transmission sont

$$\begin{aligned} \text{Robin:} & \begin{cases} \partial_{\mathbf{n}_j} \zeta_j^{s+1} - ip \zeta_j^{s+1} = l_{j,n}^k, & x = a_j, \\ \partial_{\mathbf{n}_j} \zeta_j^{s+1} - ip \zeta_j^{s+1} = r_{j,n}^k, & x = b_j, \end{cases} \\ S_0^2: & \begin{cases} \partial_{\mathbf{n}_j} \zeta_j^{s+1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0 \zeta_j^{s+1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^{n-1} \beta_{n-q} v_{j,q}^k = l_{j,n}^k, & x = a_j, \\ \partial_{\mathbf{n}_j} \zeta_j^{s+1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0 \zeta_j^{s+1} + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^{n-1} \beta_{n-q} v_{j,q}^k = r_{j,n}^k, & x = b_j, \end{cases} \end{aligned}$$

où les flux à gauche  $l_{j,n}^k$  et les flux à droite  $r_{j,n}^k$  sont définis dans la section 2.2 (voir formules (2.8)). Les conditions aux limites de Neumann homogènes sont imposées aux points  $a_0$  et  $b_0$ , un traitement particulier pour les deux domaines extrémaux est donc nécessaire :

$$\begin{cases} \partial_{\mathbf{n}_1} \zeta_1^{s+1} = 0, & x = a_0, \\ \partial_{\mathbf{n}_N} \zeta_N^{s+1} = 0, & x = b_0. \end{cases}$$

L'approximation spatiale est réalisée par la méthode des éléments finis  $P_1$ . Notons ainsi

- $\zeta_j^s$ : le vecteur nodal de l'interpolation  $P_1$  de  $\zeta_j^s$ ,
- $\mathbf{b}_{j,f(v)}$ : le vecteur associée à  $\int_{a_j}^{b_j} f(v) v \phi dx$ .

La formulation faible (3.4) s'écrit matriciellement

$$\left( \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j - \mathbb{B}_{j,n} \right) \zeta_j^{s+1} = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k - \mathbf{b}_{j,f(\zeta_j^s)} + \mathbf{b}_{j,n}^k - Q_j^T (l_n^k, r_n^k)^T, \quad (3.5)$$

où

$$\mathbb{B}_{j,n} = \begin{pmatrix} B_{j,n}^l & 0 & \cdots & \cdots & 0 \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & 0 \\ 0 & 0 & \cdots & 0 & B_{j,n}^r \end{pmatrix} \in \mathbb{C}^{N_j \times N_j}, \quad \mathbf{b}_{j,n}^k = \begin{pmatrix} b_{j,n}^{k,l} \\ 0 \\ \vdots \\ 0 \\ b_{j,n}^{k,r} \end{pmatrix} \in \mathbb{C}^{N_j}. \quad (3.6)$$

Les quantités scalaires  $B_{j,n}^l$ ,  $B_{j,n}^r$ ,  $b_{j,n}^{k,l}$  et  $b_{j,n}^{k,r}$  dépendent de la condition de transmission:

$$\text{Robin : } \begin{cases} B_{1,n}^l = 0, \quad B_{1,n}^r = -ip, \quad b_{1,n}^{k,l} = b_{1,n}^{k,r} = 0, \\ B_{j,n}^l = B_{j,n}^r = -ip, \quad b_{j,n}^{k,l} = b_{j,n}^{k,r} = 0, \quad j = 2, 3, \dots, N-1, \\ B_{N,n}^l = -ip, \quad B_{N,n}^r = 0, \quad b_{N,n}^{k,l} = b_{N,n}^{k,r} = 0, \end{cases}$$

$$S_0^2 : \begin{cases} B_{1,n}^l = 0, \quad B_{1,n}^r = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}}, \\ B_{j,n}^l = B_{j,n}^r = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}}, \quad j = 2, 3, \dots, N-1, \\ B_{N,n}^l = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}}, \quad B_{N,n}^r = 0, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^{n-1} \beta_{n-q} Q_{j,l} \mathbf{v}_{j,q}^k, \quad j = 2, 3, \dots, N, \\ b_{j,n}^{k,r} = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^{n-1} \beta_{n-q} Q_{j,r} \mathbf{v}_{j,q}^k, \quad j = 1, 2, \dots, N-1. \end{cases}$$

Numériquement la condition d'arrêt de la procédure de point fixe en  $s$  est

$$\| \zeta_j^{s+1} - \zeta_j^s \|_{l^2} \leq \varepsilon,$$

où  $\varepsilon = 10^{-12}$  dans nos codes.

En conclusion, les conditions de transmission discrètes sont

$$\begin{aligned} l_{j+1,n}^{k+1} &= -r_{j,n}^k + 2\tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} &= -l_{j,n}^k + 2\tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \end{aligned} \quad (3.7)$$

où  $\tilde{S}$  est une forme discrète de  $S$ :

$$\text{Robin : } \begin{cases} \tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k) = \tilde{S}_p(Q_{j,l} \mathbf{v}_{j,n}^k) = -ip \cdot (Q_{j,l} \mathbf{v}_{j,n}^k), \\ \tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k) = \tilde{S}_p(Q_{j,r} \mathbf{v}_{j,n}^k) = -ip \cdot (Q_{j,r} \mathbf{v}_{j,n}^k), \end{cases}$$

$$S_0^2 : \begin{cases} \tilde{S}(Q_{j,l} \mathbf{v}_{j,n}^k) = \tilde{S}_0^2(Q_{j,l} \mathbf{v}_{j,n}^k) = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^n \beta_{n-q} (Q_{j,l} \mathbf{v}_{j,q}^k), \\ \tilde{S}(Q_{j,r} \mathbf{v}_{j,n}^k) = \tilde{S}_0^2(Q_{j,r} \mathbf{v}_{j,n}^k) = e^{-i\pi/4} \sqrt{\frac{2}{\Delta t}} \sum_{q=0}^n \beta_{n-q} (Q_{j,r} \mathbf{v}_{j,q}^k). \end{cases}$$



Sur le domaine complet  $]a_0, b_0[$ , la forme discrète de l'équation est

$$\begin{aligned} \left(\frac{2i}{\Delta t}\mathbb{M}_0 - \mathbb{S}_0\right)\zeta_0^s &= \frac{2i}{\Delta t}\mathbb{M}_0\mathbf{u}_{0,n-1}^k - \mathbf{b}_{0,f(\zeta_0^{s-1})}, \\ \mathbf{v}_{0,n}^k &= \lim_{s \rightarrow +\infty} \zeta_0^s. \end{aligned}$$

où la condition aux limites Neumann homogène est utilisée. Nous rappelons que le but de la méthode SWR est que

$$\mathbf{v}_{j,n} = R_j \mathbf{v}_{0,n}, \quad n = 1, \dots, N_T,$$

où  $R_j \in \mathbb{R}^{N_0 \times N_j}$  est la matrice de restriction des nœuds dans  $]a_0, b_0[$  à  $]a_j, b_j[$ .

*Remarque 3.1.* Nous pourrions utiliser un autre schéma (schéma relaxation) pour discrétiser l'équation de Schrödinger non-linéaire [20]. L'équation de Schrödinger sur chaque sous-domaine

$$\begin{cases} i\partial_t u_j^k + \partial_{xx} u_j^k + |u_j^k|^2 u_j^k = 0, & (t, x) \in \Theta_j, \\ u_j^k(0, x) = u_0(x), \\ CL., \end{cases}$$

est réécrite comme

$$\begin{cases} i\partial_t u_j^k + \partial_{xx} u_j^k + \phi_j^k u_j^k = 0, & (t, x) \in \Theta_j \\ \phi_j^k = |u_j^k|^2, \\ u_j^k(0, x) = u_0(x), \\ CL., \end{cases}$$

où  $CL.$  désigne la condition aux limites. La semi-discrétisation en temps est

$$\begin{cases} i\frac{u_{j,n}^k - u_{j,n-1}^k}{\Delta t} + \partial_{xx} \frac{u_{j,n}^k + u_{j,n-1}^k}{2} + \phi_{j,n-1/2}^k \frac{u_{j,n}^k + u_{j,n-1}^k}{2} = 0, & 1 \leq n \leq N_T, \\ \frac{\phi_{n-1/2}^k + \phi_{n-3/2}^k}{2} = |u_{j,n-1}^k|^2, \\ u_{j,0}^k(x) = u_0(x), \quad \phi_{j,-1/2}^k = |u_{j,0}^k|^2, \\ CL. \end{cases}$$

Le schéma s'écrit alors

$$\begin{cases} 2i\frac{v_{j,n}^k}{\Delta t} + \partial_{xx} v_{j,n}^k + \phi_{j,n-1/2}^k v_{j,n}^k = 2i\frac{u_{j,n-1}^k}{\Delta t}, \\ \phi_{n-1/2}^k = 2|u_{j,n-1}^k|^2 - \phi_{n-3/2}^k, \\ u_{j,0}^k(x) = u_0(x), \quad \phi_{j,-1/2}^k = |u_{j,0}^k|^2, \\ CL. \end{cases}$$

En utilisant la méthode des éléments finis  $P_1$ , matriciellement, on a

$$\left(\frac{2i}{\Delta t}\mathbb{M}_j - \mathbb{S}_j + \mathbb{M}_{\phi_{j,n-1/2}^k} - \mathbb{B}_{j,n}\right)\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_n^k, r_n^k)^T,$$

où  $\phi_{j,n-1/2}^k$  est le vecteur nodal de l'interpolation  $P_1$  de  $\phi_{j,n-1/2}^k$ .

Par rapport au schéma de Durán et Sanz-Serna, qui est également basé sur une méthode de centrage en temps de type Crank-Nicolson, ce système permet d'éviter un traitement numérique coûteux de la non-linéarité et de préserver la flexibilité de choix de discrétisation spatiale. Comme nous sommes en une dimension d'espace, il est facile de faire appel à la méthode directe LU pour résoudre les systèmes linéaires ci-dessus. Cependant, et contrairement à la méthode de Durán et Sanz-Serna, la matrice  $\mathbb{M}_{\phi_{j,n-1/2}^k}$  dépend du temps. Nous devons

donc faire la factorisation LU beaucoup de fois. Nous voyons dans la formule (3.5) que pour le schéma de Durán et Sanz-Serna, les matrices sont identiques pour  $n = 1, 2, \dots, N_T$ , nous faisons alors une seule fois la factorisation LU. D'après nos tests numériques, le schéma de Durán et Sanz-Serna prends ainsi moins de temps de calcul. Ainsi dans ce chapitre, nous préférons avoir recours au schéma de Durán et Sanz-Serna.

### 3.3 Algorithme avec préconditionneur

Nous avons écrit les  $N$  problèmes (2.13) sur chaque sous-domaine de manière globale pour l'équation de Schrödinger avec un potentiel linéaire  $\mathcal{V} = V(t, x)$  dans la sous-section 2.3. Nous nous rappelons que le problème d'interface est

$$g^{k+1} = \mathcal{R}g^k = \mathcal{L}g^k + d, \quad (3.8)$$

où le vecteur aux interfaces global  $g^k$  est défini par (2.16). Comme nous l'avons expliqué au chapitre précédent, l'équation (3.8) est juste la méthode de point fixe pour

$$(I - \mathcal{L})g = d.$$

Pour l'équation de Schrödinger avec un potentiel non-linéaire  $\mathcal{V} = f(u)$ , de la même manière, nous pouvons écrire les  $N$  problèmes (3.5)-(3.7) comme

$$g^{k+1} = \mathcal{R}_{nl}g^k, \quad (3.9)$$

où  $\mathcal{R}_{nl}$  est cette fois un opérateur non-linéaire. L'application de l'opérateur  $\mathcal{R}_{nl}$  au vecteur  $g^k$  consiste en deux parties :

1. la résolution de l'équation non-linéaire sur chaque sous-domaine avec  $g^k$  par (3.5),
2. le calcul de  $g^{k+1}$  par (3.7).

Si le potentiel  $\mathcal{V} = V(t, x)$  est dépendant du temps, la matrice  $\mathcal{L}$  dans (3.8) n'a plus les propriétés qui sont montrées dans les propositions 2.3 et 2.4. Ainsi, il n'est plus possible de la construire sans être obligé de réaliser beaucoup de calculs. Si le potentiel est non-linéaire, comme nous l'avons dit ci-dessus  $\mathcal{R}_{nl}$  est un opérateur non-linéaire. À chaque itération, il faut recalculer  $\mathcal{R}g^k$  ou  $\mathcal{R}_{nl}g^k$  ce qui est très coûteux. Nous proposons ici d'ajouter un préconditionneur  $P^{-1}$  ( $P$  est une matrice non singulière) dans (3.8) et (3.9) pour diminuer le nombre d'itérations requises. Les algorithmes avec préconditionneur s'écrivent alors

- $\mathcal{V} = V(t, x)$ ,

$$g^{k+1} = I - P^{-1}(I - \mathcal{R})g^k, \quad (3.10)$$

$$P^{-1}(I - \mathcal{L}) = P^{-1}d. \quad (3.11)$$

- $\mathcal{V} = f(u)$ ,

$$g^{k+1} = I - P^{-1}(I - \mathcal{R}_{nl})g^k, \quad (3.12)$$

où  $I$  est l'opérateur identité.

Nous expliquons maintenant quel préconditionneur nous allons utiliser. Considérons l'équation de Schrödinger linéaire sans potentiel ( $\mathcal{V} = 0$ )

$$\begin{cases} i\partial_t u + \partial_{xx} u = 0, & (t, x) \in ]0, T[ \times ]a_0, b_0[, \\ u(0, x) = u_0(x), & x \in ]a_0, b_0[, \\ \partial_{\mathbf{n}} u = 0, & x = a_0, b_0. \end{cases} \quad (3.13)$$

Son problème d'interface est

$$g^{k+1} = \mathcal{L}_0 g^k + d,$$

où le symbole  $\mathcal{L}_0$  est utilisé pour mettre en exergue qu'ici le potentiel est nul. La condition de transmission est la même que celle pour (3.1). Nous proposons alors de construire le préconditionneur comme

$$P = I - \mathcal{L}_0. \quad (3.14)$$

Nous avons deux raisons de penser que cela constitue un bon choix:

- $\mathcal{L}_0$  peut être construit facilement car le potentiel nul est indépendant du temps. Pour cela, il suffit de résoudre l'équation (3.13) deux fois sur chaque sous-domaine, cette construction étant aussi scalable,
- intuitivement, l'opérateur de Schrödinger sans potentiel est grosso modo une approximation de l'opérateur de Schrödinger avec potentiel:

$$i\partial_t + \partial_{xx} \approx i\partial_t + \partial_{xx} u + \mathcal{V},$$

ainsi

$$\begin{aligned} P &= I - \mathcal{L}_0 \approx I - \mathcal{L}, \\ P &= I - \mathcal{L}_0 \approx I - (\mathcal{R}_{nl} - \mathcal{R}_{nl} \cdot \mathbf{0}). \end{aligned}$$

Les algorithmes avec préconditionneur pour  $\mathcal{V} = V(t, x)$  s'écrivent alors

---

**Algorithme 4 :** Algorithme avec préconditionneur,  $\mathcal{V} = V(t, x)$

---

- 1: Initialiser l'itération par  $g^0$ ,
  - 2: À chaque itération:
    - 2.1:  $\mathcal{R}g^k$ ,
    - 2.2:  $g^k - \mathcal{R}g^k$ ,
    - 2.3:  $P^{-1}(g^k - \mathcal{R}g^k)$ ,
    - 2.4:  $g^{k+1} = g^k - P^{-1}(g^k - \mathcal{R}g^k)$ ,
  - 3: Répéter l'étape 2 jusqu'à  $\|g^{k+1} - g^k\| < \varepsilon$ ,  $\varepsilon \ll 1$ .
- 

Nous retrouvons la première version de l'algorithme avec préconditionneur pour  $\mathcal{V} = V(t, x)$  si la méthode de point fixe est utilisée à l'étape 4.

---

**Algorithme 5 :** Algorithme avec préconditionneur,  $\mathcal{V} = V(t, x)$ , version 2

---

- 1: Construire  $d = \mathcal{R} \cdot \mathbf{0}$  dans (3.11) explicitement,
- 2: Définir l'application de  $I - \mathcal{L}$  au vecteur quelconque  $v$  dans (3.11):

$$(I - \mathcal{L})v = I - \mathcal{R}v + d,$$

- 3: Définir l'application de  $P^{-1}$  au vecteur dans (3.11),
  - 4: Résoudre (3.11) par une méthode itérative (point fixe ou Krylov),
  - 5: Résoudre l'équation de Schrödinger sur chaque sous-domaine pour chaque pas de temps en utilisant les conditions aux limites obtenues à l'étape 4.
- 

L'algorithme avec préconditionneur pour  $\mathcal{V} = |u|^2$  est

---

**Algorithme 6 :** Algorithme avec préconditionneur,  $\mathcal{V} = |u|^2$

---

- 1: Initialiser l'itération par  $g^0$ ,
  - 2: À chaque itération:
    - 2.1:  $\mathcal{R}_{nl}g^k$ ,
    - 2.2:  $g^k - \mathcal{R}_{nl}g^k$ ,
    - 2.3:  $P^{-1}(g^k - \mathcal{R}_{nl}g^k)$ ,
    - 2.4:  $g^{k+1} = g^k - P^{-1}(g^k - \mathcal{R}_{nl}g^k)$ ,
  - 3: Répéter l'étape 2 jusqu'à  $\|g^{k+1} - g^k\| < \varepsilon$ ,  $\varepsilon \ll 1$ .
- 

Nous présentons maintenant l'application du préconditionneur. La matrice transposée de  $P$  est stockée sous la forme PETSc (voir la sous-section 2.4.2). Pour un vecteur quelconque  $y$ , le vecteur  $x := P^{-1}y$  est calculé par la résolution du système linéaire

$$Px = (I - \mathcal{L}_0)x = y. \quad (3.15)$$

Nous ne construisons pas explicitement la matrice  $P^{-1}$  car l'inverse d'une matrice distribuée est beaucoup trop coûteux numériquement. Le système linéaire (3.15) est donc résolu par les méthodes Krylov (Gmres ou Bicgstab) initialisé par le vecteur nul et utilisant la librairie PETSc. Nous avons vu dans le chapitre 2 que ce système linéaire est robuste numériquement. D'après les résultats numériques de la sous-section 2.6.1, nous savons que la résolution de ce système prend très peu de temps ( $T_{Ld} \approx 0$ ) si le nombre de sous-domaines  $N$  n'est pas très grand. Lorsque  $N$  est grand, elle ne consomme non plus pas beaucoup de temps en comparaison de la résolution des équations sur chaque sous-domaine.

### 3.4 Résultats numériques

Dans cette sous-section, nous présentons les résultats numériques et en particulier l'avantage de l'utilisation du préconditionneur. Nous comparons donc d'abord les algorithmes sans et avec préconditionneur. Nous allons ensuite montrer que l'algorithme avec préconditionneur est presque indépendant du maillage en espace et en temps. Dans la sous-section 3.4.2, les différentes conditions de transmission sont comparées dans le cadre de l'algorithme avec préconditionneur. Pour l'équation de Schrödinger linéaire ( $\mathcal{V} = V(t, x)$ ), on utilise les

méthodes de Krylov (Gmres ou Bicgstab) au lieu de la méthode point fixe pour (3.11). Enfin, nous étudions à nouveau l'utilisation d'accélérateurs graphiques.

Le domaine  $]a_0, b_0[ = ]-21, 21[$  est décomposé en  $N$  sous-domaines égaux. On fixe le temps final à  $T = 0.5$  et le pas de temps à  $\Delta t = 0.001$ . Pour l'équation de Schrödinger linéaire, le potentiel et la donnée initiale sont

$$\mathcal{V} = 5tx, \quad u_0(x) = e^{-(x+10)^2 + 20i(x+10)}.$$

Pour l'équation de Schrödinger non-linéaire, on prend

$$\mathcal{V} = |u|^2, \quad u_0(x) = 2\operatorname{sech}(\sqrt{2}(x+10))e^{20i(x+10)},$$

ceci conduisant à une donnée de type soliton.

La condition d'arrêt pour les méthodes SWR est  $\|g^{k+1} - g^k\|_{l^2} < 10^{-10}$ . Les modules des solutions au temps final  $t = T$  résolus sur le domaine complet  $]0, T[\times]a_0, b_0[$  sont présentées dans la figure 3.1.

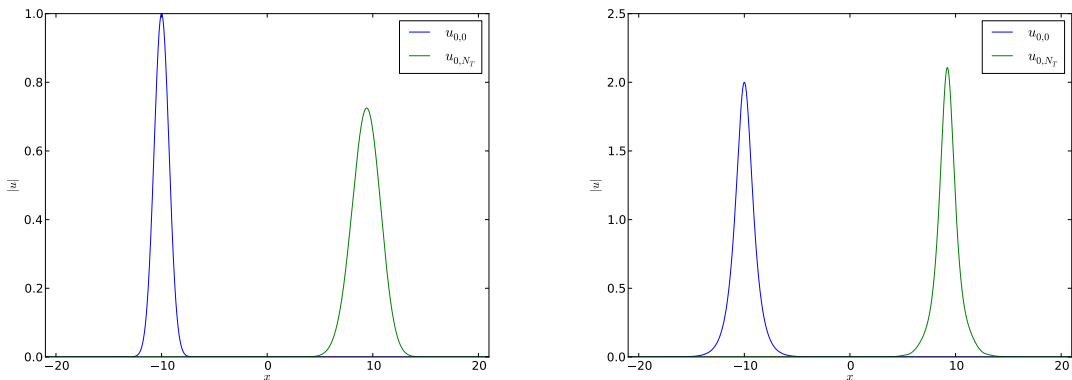


FIGURE 3.1:  $|u_{0,0}|$  et  $|u_{0,N_T}|$  sur  $]a_0, b_0[$ ,  $\mathcal{V} = 5tx$  (gauche) et  $\mathcal{V} = |u|^2$  (droite),  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

### 3.4.1 Comparaison des algorithmes sans et avec préconditionneur

Dans cette partie, nous nous intéressons à observer la robustesse, les temps de calcul et la scalabilité forte du calcul distribué des algorithmes sans et avec préconditionneur pour les deux potentiels.

Nous définissons d'abord

- $N_{\text{nopc}}$ : le nombre d'itérations requises avec l'algorithme sans préconditionneur sur  $N$  sous-domaines,
- $T_{\text{nopc}}$ : le temps de calcul obtenu avec l'algorithme sans préconditionneur sur  $N$  sous-domaines,

- $N_{\text{pc}}$ : le nombre d'itérations requises avec l'algorithme avec préconditionneur sur  $N$  sous-domaines,
- $T_{\text{pc}}$ : le temps de calcul obtenu avec l'algorithme avec préconditionneur pour  $N$  sous-domaines,
- $T^{\text{ref}}$ : le temps de calcul sur un processeur de la résolution numérique de l'équation de Schrödinger sur le domaine complet  $]0, T[ \times ]a_0, b_0[$ .

Comme nous l'avons expliqué dans la sous-section 2.6.2, le choix du vecteur d'initialisation à l'interface est important. Nous utilisons le vecteur nul comme vecteur initial dans cette sous-section.

### 3.4.1.1 Cas du potentiel $\mathcal{V} = 5tx$

Nous faisons des tests des algorithmes sans et avec préconditionneur pour  $N = 10$ ,  $N = 100$  et  $N = 1000$  sous-domaines avec la condition de transmission mettant en jeu  $S_0^2$ . La figure 3.2 présente l'historique de la convergence pour  $\mathcal{V} = 5tx$ , où  $\Delta x = 10^{-5}$ . Nous pouvons constater que si  $N$  n'est pas grand, il n'y a pas de grandes différences entre les algorithmes sans et avec préconditionneur. En revanche, si  $N$  est grand, comme dans l'algorithme sans préconditionneur, chaque domaine ne communique à chaque itération qu'avec ses domaines voisins, on peut constater que l'algorithme sans préconditionneur converge très lentement au début des itérations. Il y a un plateau dans les historiques de la convergence. Dans l'algorithme avec préconditionneur, le plateau disparaît. Le préconditionneur permet en fait une communication avec les domaines éloignés et améliore grandement la convergence.

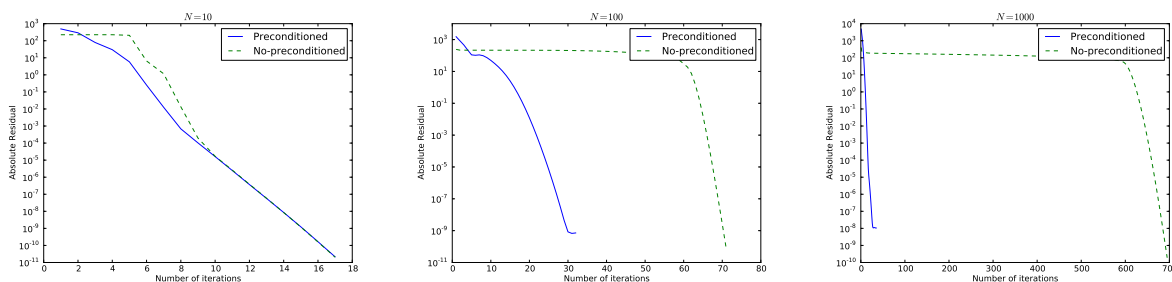


FIGURE 3.2: Historique de la convergence,  $N = 10, 100, 1000$ ,  $S_0^2$ ,  $V = 5tx$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

Nous présentons les nombres d'itérations requis et les temps de calculs dans le tableau 3.1 pour  $N = 10$ ,  $N = 100$ ,  $N = 500$ ,  $N = 1000$ . Le temps de calcul pour  $N = 2$  est trop long pour être présenté ici, nous faisons ainsi des tests à partir de  $N = 10$ . Nous pouvons voir que le préconditionneur permet de faire diminuer beaucoup le nombre d'itérations et le temps de calcul. La scalabilité forte de l'algorithme sans préconditionneur est très faible. En effet, le nombre d'itérations requises augmente avec le nombre de sous-domaines considérées. L'algorithme avec préconditionneur est lui beaucoup plus scalable (jusqu'à 500 sous-domaines). En revanche, on perd en scalabilité de  $N = 500$  à  $N = 1000$  sous domaines. En effet, le nombre d'itérations pour  $N = 1000$  est bien supérieur à celui pour  $N = 500$ .

Nous pouvons l'expliquer par l'implémentation du préconditionneur. Le temps  $T_{\text{pc}}$  se compose principalement de trois parties importantes : l'application de  $\mathcal{R}$  aux vecteurs (l'étape 1, noté par  $T_1$ ), la construction du préconditionneur (noté par  $T_{3c}$ ) et l'application du préconditionnement (l'étape 3, noté par  $T_3$ ). On a ainsi

$$T_{\text{pc}} \approx T_1 + T_{3c} + T_3. \quad (3.16)$$

Si  $N$  n'est pas très grand,  $T_1 \sim T_{3c} \gg T_3$ . En augmentant le nombre de sous-domaines,  $T_1$  et  $T_{3c}$  diminuent et  $T_3$  augmente. Ainsi, si  $N$  est grand,  $T_3$  n'est plus négligeable par rapport à  $T_1$  et  $T_{3c}$ . Cependant, il n'est pas très convenable d'estimer les temps  $T_1$  et  $T_3$  dans notre code car nous faisons appel aux "free-matrix" solveurs de la librairie PETSc (voir Annexe B). Pour confirmer notre explication, nous faisons un autre test en utilisant un maillage moins fin en espace ( $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ). La taille du problème d'interface (3.8) est la même, ainsi  $T_3$  est similaire à celui du test précédent ( $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ ). Mais cette fois la taille du problème sur un sous-domaine est 10 fois plus petite. Ainsi, les temps  $T_1$  et  $T_{3c}$  sont moindres. L'algorithme avec préconditionneur doit donc devenir moins scalable. Les résultats sont présentés dans le tableau 3.2. Nous pouvons voir que le temps de calcul ( $T_{\text{pc}}$ ) pour  $N = 1000$  est plus important que celui pour  $N = 500$ . Nous pouvons donc conclure que pour ce cas test, l'algorithme avec préconditionneur n'est pas très scalable de  $N = 100$  à  $N = 500$ . Malgré cette remarque, nous voyons d'après nos tests que l'utilisation du préconditionneur permet de diminuer beaucoup le nombre d'itérations et le temps de calcul en comparaison à la méthode sans préconditionneur.

$N$	10	100	500	1000
$N_{\text{nopc}}$	17	71	349	695
$N_{\text{pc}}$	17	32	31	35
$T^{\text{ref}}$	6496.3			
$T_{\text{nopc}}$	10123.1	3217.0	2466.5	2238.0
$T_{\text{pc}}$	10128.9	1432.7	250.0	170.7

TABLEAU 3.1: Nombre d'itérations et temps de calcul en secondes des algorithmes sans et avec préconditionneur,  $S_0^2$ ,  $\mathcal{V} = 5tx$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

$N$	10	100	500	1000
$N_{\text{nopc}}$	17	71	349	695
$N_{\text{pc}}$	17	32	26	25
$T^{\text{ref}}$	507.5			
$T_{\text{nopc}}$	681.9	223.8	210.2	191.2
$T_{\text{pc}}$	694.3	107.6	38.4	54.5

TABLEAU 3.2: Nombre d'itérations et temps de calcul en secondes des algorithmes sans et avec préconditionneur,  $S_0^2$ ,  $\mathcal{V} = 5tx$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ .

### 3.4.1.2 Cas du potentiel $\mathcal{V} = |u|^2$

Nous reproduisons les mêmes tests que dans la sous-section précédente pour  $\mathcal{V} = |u|^2$ . Les historiques de la convergence pour  $N = 10$ ,  $N = 100$  et  $N = 1000$  sont montrés dans la figure 3.3. Les tableaux 3.3 et 3.4 présentent les nombres d'itérations et les temps de calcul pour  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$  et  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$  respectivement. Les conclusions sont tout à fait similaires.

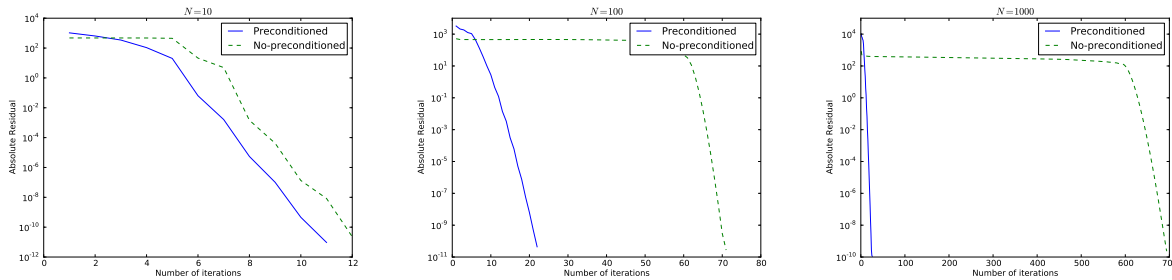


FIGURE 3.3: Historique de la convergence,  $N = 10, 100, 1000$ ,  $S_0^2$ ,  $\mathcal{V} = |u|^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

$N$	10	100	500	1000
$N_{\text{nopc}}$	12	71	349	694
$N_{\text{pc}}$	11	22	25	26
$T^{\text{ref}}$	3200.8			
$T_{\text{nopc}}$	2582.3	1332.2	1248.0	1129.7
$T_{\text{pc}}$	2446.7	408.2	117.6	83.8

TABLEAU 3.3: Nombre d'itérations et temps de calcul en secondes des algorithmes sans et avec préconditionneur,  $S_0^2$ ,  $\mathcal{V} = |u|^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-5}$ .

$N$	10	100	500	1000
$N_{\text{nopc}}$	12	71	349	694
$N_{\text{pc}}$	11	22	25	16
$T^{\text{ref}}$	299.1			
$T_{\text{nopc}}$	233.8	121.6	116.5	117.2
$T_{\text{pc}}$	222.6	43.4	35.0	61.3

TABLEAU 3.4: Nombre d'itérations et temps de calcul en secondes des algorithmes sans et avec préconditionneur,  $S_0^2$ ,  $\mathcal{V} = |u|^2$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ .

*Remarque 3.2.* D'après les tableaux 3.1 et 3.3, nous pouvons voir que le temps de calcul pour l'équation de Schrödinger non-linéaire est inférieur à celui pour l'équation de Schrödinger linéaire avec un potentiel dépendant du temps. La méthode directe LU est utilisée pour résoudre les systèmes linéaire (3.5) pour l'équation de Schrödinger non-linéaire. Les matrices  $\frac{2i}{\Delta t} \mathbf{M}_j - \mathbf{S}_j - \mathbf{B}_{j,n}$  sont identiques pour  $n = 1, 2, \dots, N_T$ . Ainsi, il suffit de faire une fois la



factorisation LU qui est la partie la plus coûteuse dans ce type de méthode de résolution. En revanche, pour l'équation de Schrödinger linéaire avec un potentiel dépendant du temps, les matrices  $\mathbb{A}_{j,n} - \mathbb{B}_{j,n}$  (voir (2.10)) sont différentes pour  $n = 1, 2, \dots, N_T$ , ainsi le nombre total de fois où la factorisation LU est nécessaire est  $N_T \times$  Nombre d'itérations.

### 3.4.2 Conditions de transmission

Nous nous intéressons ici à la comparaison des conditions de transmission de Robin et  $S_0^2$  dans le cadre de l'algorithme avec préconditionneur. Comme dans le chapitre 1, le choix théorique du paramètre de Robin ne nous est pas connu. Nous étudions ce paramètre par des tests numériques. Le vecteur aléatoire est utilisé comme vecteur initial aux interfaces dans cette sous-section.

#### 3.4.2.1 Cas du potentiel $\mathcal{V} = 5tx$

D'abord nous considérons le potentiel linéaire dépendant du temps  $\mathcal{V} = 5tx$ . Les nombre d'itérations et les temps de calcul totaux pour la condition  $S_0^2$  et la condition Robin avec des paramètres  $p$  différents sont présentés dans le tableau 3.5. Nous pouvons constater que

- la condition de transmission  $S_0^2$  est plus robuste que celle de Robin,
- les nombre d'itérations pour les deux conditions de transmission sont similaires,
- pour la condition de transmission de Robin, le nombre d'itérations n'est pas sensible au paramètre  $p$ ,
- si  $N$  est grand, le temps de calcul pour la condition de transmission  $S_0^2$  est inférieur à celui de Robin.

		$N = 2$		$N = 10$		$N = 100$	
		$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$
	$p$						
Robin	5	7	1659.3	23	946.1	37	141.9
	10	7	1660.5	24	984.5	35	124.5
	15	6	1421.9	26	1067.3	33	115.1
	20	6	1431.3	27	1102.3	35	119.2
	25	6	1431.4	28	1146.0	-	
	30	6	1424.6	28	1140.6	-	
	35	6	1430.2	29	1184.5	-	
	40	6	1429.3	29	1191.7	-	
	45	6	1430.4	29	1173.0	-	
	50	6	1430.3	29	1193.5	-	
$S_0^2$		7	1657.1	23	931.9	33	111.8

-: l'algorithme ne converge pas avant 2000 itérations.

TABLEAU 3.5: Comparaison des conditions de transmission pour  $N = 2, 10, 100$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ,  $\mathcal{V} = 5tx$ .

### 3.4.2.2 Cas du potentiel $\mathcal{V} = |u|^2$

Nous faisons les mêmes tests que dans la sous-section précédente. Le tableau 3.6 présentent les résultats. Nous pouvons voir que les algorithmes sont robustes. Les autres conclusions sont similaires.

		$N = 2$		$N = 10$		$N = 100$	
		$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$
	$p$						
Robin	5	9	1042.6	12	257.9	21	55.7
	10	8	920.3	11	230.9	22	50.6
	15	8	920.3	11	228.7	22	46.7
	20	8	914.5	11	226.1	22	43.7
	25	8	913.0	11	226.4	22	43.6
	30	8	919.2	11	227.6	22	43.9
	35	8	922.1	11	231.8	22	44.4
	40	8	922.8	12	250.2	22	45.0
	45	8	921.7	12	252.5	22	46.0
	50	8	928.3	12	253.3	22	46.7
$S_0^2$		8	909.5	11	229.1	22	40.6

TABLEAU 3.6: Comparaison des conditions de transmission pour  $N = 2, 10, 100$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ,  $\mathcal{V} = |u|^2$ .

### 3.4.2.3 Conditions de transmission générales

Dans cette sous-section, nous comparons numériquement les trois familles des conditions de transmission : stratégie potentiel ( $S_0^M$ ), stratégie de changement de jauge ( $S_1^M$ ) et stratégie d'approximation de Padé ( $S_2^M$ ) dans le cadre de l'algorithme avec préconditionneur. Nous présentons tout d'abord les nombres d'itérations requises et les temps de calcul pour  $\mathcal{V} = V(t, x)$  dans le tableau 3.7. Nous pouvons constater que

- dans chaque stratégie, le nombre d'itérations n'est pas très sensible à l'ordre à l'exception de  $N = 10$ , où il est possible de diminuer un peu le nombre d'itérations en augmentant l'ordre,
- pour la stratégie d'approximation de Padé, le nombre d'itérations n'est pas sensible au paramètre de Padé ( $m$ ).

D'après les tests réalisés, il est difficile de conclure qu'une condition de transmission est meilleure qu'une autre en terme de temps de calcul ou bien de nombre d'itérations à part si nous considérons un nombre moyen de sous domaines ( $N = 10$ ). Si nous considérons en revanche un grand nombre de sous-domaines, il est clair que nous ne pouvons pas identifier la meilleure méthode. Ceci justifie notre choix d'utiliser dans nos tests de ce chapitre en priorité  $S_0^2$ . Cette condition de transmission est ici la plus simple à implémenter et ce comporte en moyenne comme les différentes autres conditions.

		$N = 2$		$N = 10$		$N = 100$	
		$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$
$S_0^M$	$S_0^2$	7	1657.1	23	931.9	33	111.8
	$S_0^3$	7	1651.1	12	488.0	31	104.9
	$S_0^4$	5	1192.7	12	487.1	31	104.0
$S_1^M$	$S_1^2$	7	1660.0	21	853.9	31	105.0
	$S_1^4$	5	1194.4	20	814.4	31	104.2
$S_2^M$	$S_2^{2,20}$	7	1663.9	11	449.9	32	109.3
	$S_2^{2,50}$	7	1654.5	10	410.3	31	103.9
	$S_2^{2,100}$	7	1658.7	9	368.2	31	106.6
	$S_2^{4,20}$	5	1197.8	11	451.5	32	108.9
	$S_2^{4,50}$	5	1194.3	10	409.3	31	104.4
	$S_2^{4,100}$	5	1191.8	9	369.3	31	105.0

TABLEAU 3.7: Comparaison des trois stratégies pour  $N = 2, 10, 100$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ,  $\mathcal{V} = 5tx$ .

Ensuite nous présentons dans le tableau 3.8 les nombre d'itérations requises et les temps de calcul pour  $\mathcal{V} = f(u)$ . Les conditions de transmission  $S_0^4$ ,  $S_1^4$  et  $S_2^{4,m}$  contiennent la dérivée normale de  $f(u)$  ( $\partial_{\mathbf{n}_j} f(u)$ ). Nous n'avons pas trouvé une bonne discrétisation de ce terme tel que la continuité de  $\mathbf{u}_{j,n}$  assure la continuité de  $\partial_{\mathbf{n}_j} f(\mathbf{u}_j)$ . Ainsi nous ne pouvons pas

obtenir la solution  $\mathbf{u}_{j,n}$  qui satisfait  $\mathbf{u}_{j,n} = R_j \mathbf{u}_{0,n}$  pour  $n = 1, 2, \dots, N_T$ . Nous pouvons voir que le nombre d'itérations n'est pas sensible à la condition de transmission et son ordre, particulièrement si  $N$  est grand. Mais le temps de calcul pour la stratégie d'approximation de Padé est beaucoup plus grand que pour les deux autres stratégies. La raison est suivante. Sur chaque sous-domaine, la non-linéarité est approchée par une procédure de point fixe en  $s$  (voir formule (3.4)). Cette procédure de point fixe converge beaucoup plus lentement en utilisant la stratégie d'approximation de Padé que les deux autres stratégies. Cette observation est aussi constatée dans [51]. En conclusion dans le cas non-linéaire, nous préférons aussi la stratégie potentiel d'ordre 2.

		$N = 2$		$N = 10$		$N = 100$	
		$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$
$S_0^M$	$S_0^2$	8	909.5	11	229.1	22	40.6
	$S_0^3$	7	802.1	10	205.8	22	41.6
$S_1^M$	$S_1^2$	7	802.3	10	205.6	22	41.4
$S_2^M$	$S_2^{2,20}$	7	1732.5	10	572.0	22	128.6
	$S_2^{2,50}$	7	4042.9	10	1342.3	23	310.3
	$S_2^{2,100}$	7	7900.5	10	2640.0	22	576.0

TABLEAU 3.8: Comparaison des trois stratégies pour  $N = 2, 10, 100$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ,  $\mathcal{V} = |u|^2$ .

### 3.4.3 Effet du maillage

Nous considérons ici l'effet du maillage dans le cadre de l'algorithme avec préconditionneur. Le vecteur initial est un vecteur aléatoire. La condition de transmission est  $S_0^2$ . Nous présentons les nombres d'itérations dans le tableau 3.9 pour  $\mathcal{V} = 5tx$  et dans le tableau 3.10 pour  $\mathcal{V} = |u|^2$ . Nous pouvons constater que l'algorithme avec préconditionneur n'est pas sensible à la variation sur  $\Delta x$  pour les potentiels linéaire et non-linéaire. En revanche, le nombre d'itérations augmente quand  $\Delta t$  diminue (plus étapes du temps).

$\Delta t$	$\Delta x$	$N = 2$	$N = 10$	$N = 100$
$10^{-2}$	$10^{-2}$	6	22	31
$10^{-2}$	$10^{-3}$	6	22	31
$10^{-2}$	$10^{-4}$	6	22	31
$10^{-3}$	$10^{-2}$	7	23	34
$10^{-3}$	$10^{-3}$	7	23	33
$10^{-3}$	$10^{-4}$	7	23	33
$10^{-4}$	$10^{-2}$	7	24	37
$10^{-4}$	$10^{-3}$	7	24	38
$10^{-4}$	$10^{-4}$	7	24	36

TABLEAU 3.9: Nombre d'itérations pour  $\Delta t$  et  $\Delta x$  différents,  $V = 5tx$ .

$\Delta t$	$\Delta x$	$N = 2$	$N = 10$	$N = 100$
$10^{-2}$	$10^{-2}$	4	8	11
$10^{-2}$	$10^{-3}$	4	8	11
$10^{-2}$	$10^{-4}$	4	8	11
$10^{-3}$	$10^{-2}$	8	11	22
$10^{-3}$	$10^{-3}$	8	11	22
$10^{-3}$	$10^{-4}$	8	11	22
$10^{-4}$	$10^{-2}$	8	11	22
$10^{-4}$	$10^{-3}$	8	11	22
$10^{-4}$	$10^{-4}$	8	11	22

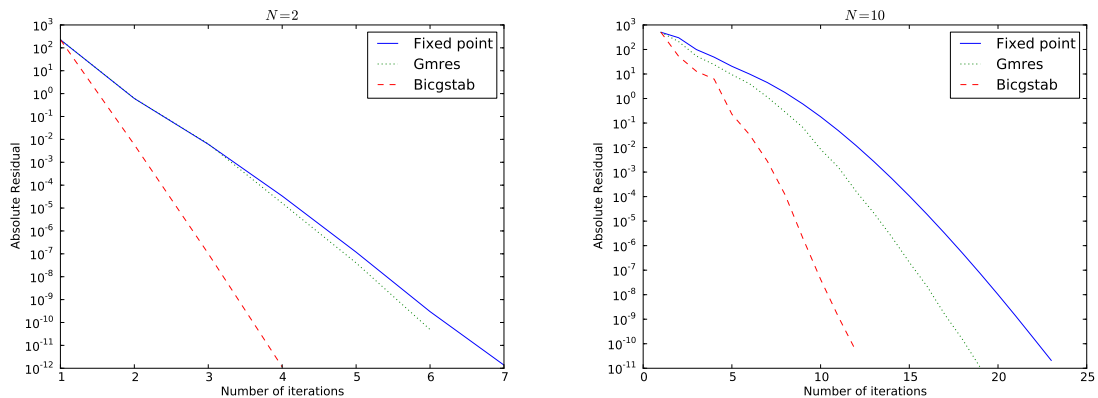
TABLEAU 3.10: Nombre d'itérations pour  $\Delta t$  et  $\Delta x$  différents,  $V = |u|^2$ .

### 3.4.4 Accélération de Krylov pour l'équation de Schrödinger linéaire

Nous nous intéressons à accélération de Krylov pour l'équation de Schrödinger linéaire avec le potentiel  $\mathcal{V} = V(t, x)$ . Nous considérons le système linéaire (3.11):

$$P^{-1}(I - \mathcal{L})g = P^{-1}d,$$

qui nous permet d'utiliser les méthodes Krylov (Gmres et Bicgstab) au lieu de la méthode de point fixe. L'historique de la convergence est montré dans la figure 3.4 pour  $N = 2$ ,  $N = 10$  et  $N = 100$ . Le tableau 3.11 présente les nombres d'itérations et les temps de calcul. Nous utilisons ici un vecteur aléatoire comme vecteur initial.



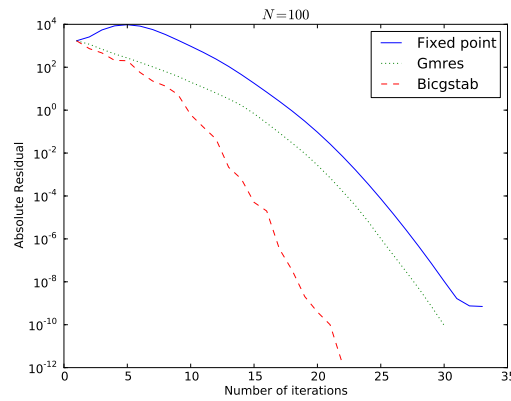


FIGURE 3.4: Historique de la convergence,  $N = 10, 100, 1000$ ,  $S_0^2$ ,  $V = 5tx$ ,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ .

$N$	Point fixe		Gmres		Bicgstab	
	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$	$N_{iter}$	$T_{total}$
2	7	1657.1	6	1890.4	4	2122.9
10	23	931.9	19	856.4	12	1011.5
100	33	111.8	30	123.3	22	151.3

TABLEAU 3.11: Temps de calcul en secondes,  $\Delta t = 0.001$ ,  $\Delta x = 10^{-4}$ ,  $V = 5tx$ .

D'après des tests ci-dessus, nous pouvons conclure que

Nombre d'itérations : Point fixe > Gmres > Bicgstab,  
 Temps de calcul : Bicgstab > Gmres > Point fixe.

Les méthodes Krylov diminuent le nombre d'itérations, mais augmentent le temps de calcul. La raison est en fait qu'à chaque itération, un plus grand nombre d'opérations sont nécessaires pour les méthodes Krylov comparativement à la méthode point fixe. En effet, deux produits matrice-vecteur doivent être effectués pour la méthode Bicgstab alors que pour la méthode Gmres, il faut faire un produit matrice-vecteur et résoudre un problème de minimisation.

### 3.4.5 Accélération Gpu

Dans l'algorithme avec préconditionneur, l'équation de Schrödinger est résolue sur chaque sous-domaines deux fois pour construire le préconditionneur et l'équation avec potentiel est résolue  $N_{iter}$  fois au total. Notre motivation d'utiliser l'accélération Gpu est de diminuer le temps de calcul consommé par la résolution d'équation. Nous utilisons 1 Processus MPI/domaine et 1 Gpu/Processus MPI. Nous faisons les tests pour  $N = 2$ ,  $N = 10$  et  $N = 100$ . Les tableaux 3.12 et 3.13 présentent les temps de calcul pour les potentiels  $V = 5tx$  et  $V = |u|^2$  respectivement pour  $\Delta t = 0.01$ ,  $\Delta x = 10^{-5}$ . Le vecteur initial est un vecteur nul et la condition de transmission est toujours  $S_0^2$ .

$N$	2	4	8
Temps Cpu	630.0	2565.6	1283.9
Temps Gpu	9.0	6.2	8.8

TABLEAU 3.12: Temps de calcul sans et avec accélération Gpu,  $\Delta t = 0.01$ ,  $\Delta x = 10^{-5}$ ,  
 $V = 5tx$ .

$N$	2	4	8
Temps Cpu	373.6	526.7	316.0
Temps Gpu	73.9	40.1	34.0

TABLEAU 3.13: Temps de calcul sans et avec accélération Gpu,  $\Delta t = 0.01$ ,  $\Delta x = 10^{-5}$ ,  
 $V = |u|^2$ .

Nous pouvons constater que les Gpus permettent d'accélérer beaucoup le calcul. Dans le cas linéaire, l'accélération est très importantes. Les raisons sont que :

- nous utilisons 1 cœur de Cpu/Proc MPI (Cpu) et 1 Gpu/Proc MPI (Gpu) mais 1 cœur de Cpu est plus faible que 1 Gpu matériellement;
- pour les Gpu, la librairie (cuSPARSE) utilise une autre méthode que la méthode direct LU car la méthode direct LU n'est pas adaptée à l'architecture Gpu.

Comme dans la sous-section 2.6.4, nous constatons que l'algorithme avec préconditionneur n'est pas scalable de  $N = 4$  à  $N = 8$ .

Finalement, nous présentons les résultats de tests pour un maillage plus fin en espace ( $\Delta x = 5 \times 10^{-6}$ ) dans les tableaux 3.14 et 3.15. Pour le potentiel non-linéaire, l'algorithme est scalable, mais pas pour le potentiel linéaire dépendant du temps. Nous répétons ici la composition de temps total pour l'algorithme avec préconditionneur (voir formule (3.16)):

$$T_{pc} \approx T_1 + T_{3c} + T_3.$$

$T_1$  et  $T_{3c}$  sont accélérés par le Gpu, mais pas le temps du préconditionnement ( $T_3$ ).  $T_3$  est grand par rapport à  $T_1$  et  $T_{3c}$  avec l'accélération Gpu. De plus, nous constatons que le temps de calcul pour le potentiel  $\mathcal{V} = |u|^2$  est plus important que pour le potentiel  $\mathcal{V} = V(t, x)$  ce qui contredit la remarque 3.2. Ceci s'explique car la méthode utilisé sur les Gpu ne fait pas de pré-traitement (comme la factorisation LU) et la non-linéarité nécessite plus d'opérations.

$N$	2	4	8
Gpu time	15.8	9.9	9.1

TABLEAU 3.14: Temps de calcul avec Gpu accélération,  $\Delta t = 0.01$ ,  $\Delta x = 5 \times 10^{-6}$ ,  
 $V = 5tx$ .

$N$	2	4	8
Gpu time	134.3	73.7	46.0

TABLEAU 3.15: Temps de calcul avec Gpu accélération,  $\Delta t = 0.01$ ,  $\Delta x = x = 5 \times 10^{-6}$ ,  
 $V = |u|^2$ .

### 3.5 Conclusion

Ce chapitre est consacré à la généralisation de la méthode SWR à l'équation de Schrödinger (une dimension en espace) avec un potentiel linéaire dépendant du temps ou un potentiel non-linéaire. L'idée est d'utiliser l'opérateur linéaire préalablement défini comme un préconditionneur. D'après les résultats numériques présentés dans la sous-section 3.4.1, cela permet de diminuer beaucoup le temps de calcul total et d'avoir une forte scalabilité, même dans les cas fortement non-linéaire.

Nous utilisons principalement deux conditions de transmission dans ce chapitre, la condition de transmission de Robin et la stratégie potentiel d'ordre 2 ( $S_0^2$ ). Les conditions de transmission sont comparées dans le cadre de l'algorithme avec préconditionneur. Nous n'avons observé que des petites différences pour le temps de calcul, mais  $S_0^2$  rend l'algorithme plus robuste. De plus, la condition de transmission évite des beaucoup de tests pour chercher le paramètre optimal.

Pour l'équation de Schrödinger linéaire, nous réalisons également des tests avec accélération Krylov dans le cadre de l'algorithme avec préconditionneur. Les méthodes Krylov (Gmres et Bicgstab) diminuent le nombre d'itérations, mais augmentent le temps de calcul.

À la fin, on montre que les Gpus peuvent accélérer beaucoup le calcul numérique.



## Partie II

# Méthode de décomposition de domaines pour l'équation de Schrödinger en deux dimensions



# Chapitre 4

## Méthode de décomposition de domaines pour l'équation de Schrödinger sans potentiel en deux dimensions

### Sommaire

---

<b>4.1</b>	<b>Introduction</b>	<b>75</b>
<b>4.2</b>	<b>Discrétisation</b>	<b>78</b>
<b>4.3</b>	<b>Problème d'interface</b>	<b>84</b>
<b>4.4</b>	<b>Nouveaux algorithmes</b>	<b>89</b>
4.4.1	Nouvel algorithme SWR	89
4.4.2	Nouvel algorithme de décomposition en espace	91
<b>4.5</b>	<b>Résultats numériques</b>	<b>92</b>
4.5.1	La méthode SWR vs. la méthode décomposition en espace	93
4.5.2	Comparaisons des algorithmes classiques et nouveaux	94
4.5.3	Influences de paramètres	96
4.5.4	Un autre cas test	99
<b>4.6</b>	<b>Conclusion</b>	<b>100</b>

---

### 4.1 Introduction

Nous nous intéressons dans ce chapitre à la méthode de décomposition de domaines pour l'équation de Schrödinger sans potentiel en deux dimensions d'espace sur un domaine spatial borné  $\Omega_0 = ]x_l, x_r[ \times ]y_d, y_u[$ ,  $x_l, x_r, y_d, y_u \in \mathbb{R}$

$$\begin{cases} (i\partial_t + \Delta + V)u = 0, & (t, x, y) \in ]0, T[ \times \Omega_0, \\ u(0, x, y) = u_0(x, y), & (x, y) \in \Omega_0, \end{cases} \quad (4.1)$$

où le potentiel  $V(t, x, y) = 0$  pour ce chapitre et la donnée initiale  $u_0 \in L^2(\Omega_0)$ . Pour faciliter l'implémentation, nous utilisons les conditions aux limites suivantes:

$$\begin{aligned} \partial_{\mathbf{n}} u &= 0, \quad y = y_d, y_u, \\ B_0 u &= 0, \quad x = x_l, x_r, \end{aligned}$$

où l'opérateur  $B_0$  est donné par la formule (4.3) et  $\partial_{\mathbf{n}}$  désigne l'opérateur de dérivée normal au bord. Comme nous prenons un domaine assez grand et que les valeurs sur le bord de  $\Omega_0$  sont très petite (en dehors de la précision de l'ordinateur), les conditions aux limites ci-dessus sont adaptés.

L'intervalle  $]x_l, x_r[$  est décomposé en  $N$  sous intervalles  $]a_j, b_j[$  sans recouvrement. Les points  $a_j$  et  $b_j$  désignent les extrémités des sous intervalles  $]a_j, b_j[$ . Ainsi le domaine complet  $\Omega_0$  est décomposé en  $N$  sous-domaines  $\Omega_j = ]a_j, b_j[ \times ]y_d, y_u[, j = 1, 2, \dots, N$  sans recouvrement (voir figure 4.1 pour  $N = 3$ ). Nous définissons également les normales unitaires sortantes aux sous-domaines  $\Omega_j$  par  $\partial_{\mathbf{n}_j}$ .

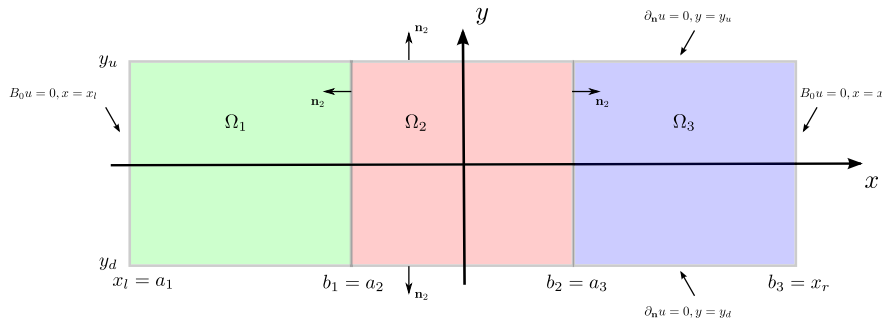


FIGURE 4.1: Configuration géométrique.

Il existe évidemment d'autres façons de décomposer le domaine complet  $\Omega_0$ . Une possibilité est décrite dans la figure 4.2b pour  $N = 4$ . Les intervalles  $]x_l, x_r[$  et  $]y_d, y_u[$  sont ainsi décomposés simultanément en sous intervalles dans les deux directions de l'espace. Il existe cependant un point de croisement. Il est bien connu que la méthode de décomposition de domaines avec point de croisement est un problème difficile en mathématique car le problème devient singulier en ce point. Nous reviendrons sur ce problème et proposerons une méthode pour traiter ce point dans le chapitre 6. Une autre possibilité est illustrée dans la figure 4.2b pour  $N = 3$ . Le domaine complet est décomposé en un cercle et des anneaux. Cette façon a beaucoup de désavantages pour le calcul parallèle. Les tailles des interfaces sont très différentes (particulièrement si  $N$  est grand). Il est aussi difficile de contrôler que les tailles des sous-domaines soient similaires. Nous nous restreignons donc dans ce chapitre à la première description.

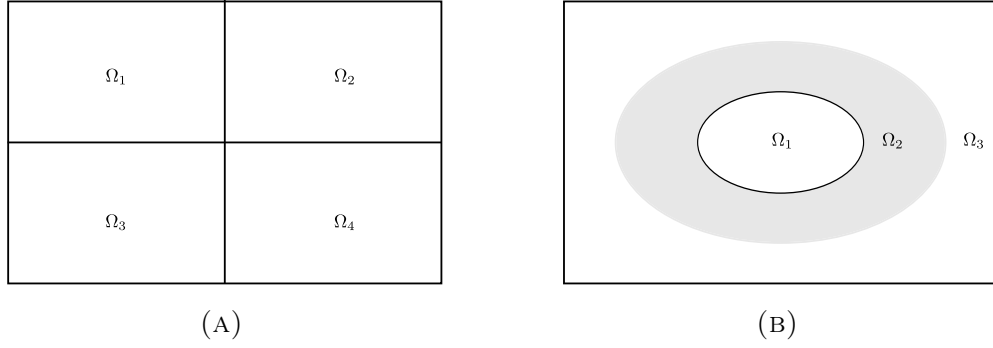


FIGURE 4.2: Deux autres façons de décomposition d'espace.

Nous considérons deux méthodes de décomposition de domaines : la méthode SWR et la méthode de décomposition en espace. Nous avons beaucoup étudié la méthode SWR dans la première partie pour l'équation de Schrödinger en une dimension. Elle est donnée ici en deux dimensions par

$$\left\{ \begin{array}{ll} \mathcal{L}u_j^{k+1} = 0, & (t, x, y) \in ]0, T[ \times \Omega_j, j = 2, \dots, N-1, \\ u_j^{k+1}(0, x, y) = u_0(x, y), & (x, y) \in \Omega_j, \\ B_j u_j^{k+1} = B_j u_{j-1}^k, & x = a_j, \\ B_j u_j^{k+1} = B_j u_{j+1}^k, & x = b_j, \\ \partial_{\mathbf{n}_j} u_j^{k+1} = 0, & y = y_l, y_d, \end{array} \right. \quad (4.2)$$

avec un traitement particulier des deux sous-domaines extrémaux

$$\left\{ \begin{array}{l} \mathcal{L}u_1^{k+1} = 0, (t, x, y) \in ]0, T[ \times \Omega_1, \\ u_1^{k+1}(0, x, y) = u_0(x, y), (x, y) \in \Omega_1, \\ B_1 u_1^{k+1} = 0, x = a_1, \\ B_1 u_1^{k+1} = B_1 u_2^k, x = b_1, \\ \partial_{\mathbf{n}_1} u_1^{k+1} = 0, y = y_l, y_d, \end{array} \right. \quad \left\{ \begin{array}{l} \mathcal{L}u_N^{k+1} = 0, (t, x, y) \in ]0, T[ \times \Omega_N, \\ u_N^{k+1}(0, x, y) = u_0(x, y), (x, y) \in \Omega_N, \\ B_N u_N^{k+1} = B_N u_{N-1}^k, x = a_N, \\ B_N u_N^{k+1} = 0, x = b_N, \\ \partial_{\mathbf{n}_N} u_N^{k+1} = 0, y = y_l, y_d, \end{array} \right.$$

où  $\mathcal{L} := i\partial_t + \Delta + V$  est l'opérateur de Schrödinger linéaire et  $B_j$  est la condition de transmission:

$$B_j = \partial_{\mathbf{n}_j} + S_j, \quad j = 0, 1, \dots, N. \quad (4.3)$$

Nous utilisons majoritairement deux types de conditions de transmission ici

$$\begin{aligned} \text{Robin: } & S_j = -ip, \quad p \in \mathbb{R}^+, \\ S_{2p}^m : & S_j = -i\sqrt{i\partial_t + \Delta_{\Gamma_j} + V}. \end{aligned}$$

Comme nos sous-domaines sont des bandes verticales, l'opérateur de Laplace-Beltrami  $\Delta_{\Gamma_j}$  est  $\partial_y^2$ . Nous pouvons également considérer d'autres conditions de transmission basées sur des opérateurs provenant de conditions aux limites absorbantes construites récemment [51]. Nous verrons ces opérateurs dans la sous-section 4.2.

Comme nous l'avons rappelé dans l'introduction, la méthode de décomposition en espace consiste à appliquer la méthode de Schwarz optimisée à l'équation de Schrödinger semi-discrète:

$$\begin{aligned} \frac{2i}{\Delta t}v_{0,n} + \Delta v_{0,n} + W_n v_{0,n} &= \frac{2i}{\Delta t}u_{0,n-1}, \\ u_{0,n} &= 2v_{0,n} - u_{0,n-1}, \end{aligned} \quad (4.4)$$

où  $W_n(x, y) = (V(t_n, x, y) + V(t_{n-1}, x, y))/2$ . Comme l'intervalle  $]t_{n-1}, t_n[$  ne contient qu'une étape du temps, numériquement, la méthode de décomposition en espace peut être considérée comme

---

**Algorithme 7 :** Méthode de décomposition en espace

---

La donnée initiale est  $u_{0,0}$ ,

**for**  $n = 1, 2, \dots, N_T$  **do**

    La méthode SWR appliquée au problème

$$\begin{cases} \mathcal{L}u = 0, (t, x, y) \in ]t_{n-1}, t_n[ \times \Omega_0, \\ u(0, x, y) = u_{0,n-1}(x, y), (x, y) \in \Omega_0, \end{cases}$$

    la solution en temps  $t_n$  est noté par  $u_{0,n}$  où  $t_n = n\Delta t$ .

---

Ainsi, la procédure de la discrétisation en espace sera similaire à celle pour la méthode SWR.

Le but de ce chapitre est d'étendre le nouvel algorithme que nous avons proposé pour l'équation de Schrödinger en une dimension à deux dimensions. L'autre intérêt est de comprendre l'influence des conditions de transmission sur la convergence. De plus, nous allons comparer la méthode SWR et la méthode de décomposition en espace.

## 4.2 Discrétisation

Nous présentons brièvement la discrétisation de l'équation de Schrödinger linéaire avec un potentiel linéaire général  $V(t, x, y)$ . La discrétisation est similaire à celle réalisée en une dimension. L'intervalle de temps  $]0, T[$  est discrétisé uniformément en  $N_T$  intervalles de longueur  $\Delta t = T/N_T$ . Nous désignons  $t_n = n\Delta t$ ,  $V_n(x, y) = V(t_n, x, y)$  et  $u_{j,n}^k(x, y)$  une approximation de  $u_j^k(t_n, x, y)$ . Conformément à la première partie, on discrétise (4.2) sur chaque sous-domaine par la méthode Crank-Nicolson en temps

$$i \frac{u_{j,n}^k - u_{j,n-1}^k}{\Delta t} + \Delta \frac{u_{j,n}^k + u_{j,n-1}^k}{2} + \frac{V_n + V_{n-1}}{2} \frac{u_{j,n}^k + u_{j,n-1}^k}{2} = 0, \quad 1 \leq n \leq N_T.$$

En introduisant les nouvelles variables  $v_{j,n}^k = \frac{u_{j,n}^k + u_{j,n-1}^k}{2}$  avec  $v_{j,0}^k = u_{j,0}^k$  et  $W_n = \frac{V_n + V_{n-1}}{2}$ , le schéma s'écrit alors

$$2i \frac{v_{j,n}^k}{\Delta t} + \Delta v_{j,n}^k + W_n v_{j,n}^k = 2i \frac{u_{j,n-1}^k}{\Delta t}. \quad (4.5)$$

Nous remarquons ici que l'équation semi-discrète en temps sur le domaine complet est juste celle donnée par (4.4).

Nous utilisons la méthode des éléments finis  $Q_1$  pour l'approximation spatiale. La formulation variationnelle pour une fonction test  $\phi \in H^1(\Omega_j)$  est

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{\Omega_j} v_{j,n}^k \phi d\Omega - \int_{\Omega_j} \nabla v_{j,n}^k \nabla \phi d\Omega + \int_{\Omega_j} W_n v_{j,n}^k \phi d\Omega \\ & + \int_{\Gamma_j} \partial_{\mathbf{n}_j} v_{j,n}^k \phi d\Gamma = \frac{2i}{\Delta t} \int_{\Omega_j} u_{j,n-1}^k \phi d\Omega, \end{aligned} \quad (4.6)$$

où  $\Gamma_j = \{x = a_j\} \times ]y_d, y_u[ \cup \{x = b_j\} \times ]y_d, y_u[$ . L'utilisation des conditions de transmission donne les conditions aux limites pour chaque sous-domaine suivantes:

$$\begin{cases} \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S}_j v_{j,n}^k = l_{j,n}^k, & x = a_j, \\ \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S}_j v_{j,n}^k = r_{j,n}^k, & x = b_j. \end{cases}$$

Le flux à gauche  $l_{j,n}^k(y), y \in ]y_d, y_u[$  et le flux à droite  $r_{j,n}^k(y), y \in ]y_d, y_u[$  sont définis pour chaque sous-domaine par

$$\begin{aligned} l_{j,n}^k(y) &= \partial_{\mathbf{n}_j} v_{j,n}^k(a_j, y) + \bar{S}_j v_{j,n}^k(a_j, y), \quad j = 2, 3, \dots, N \\ r_{j,n}^k(y) &= \partial_{\mathbf{n}_j} v_{j,n}^k(b_j, y) + \bar{S}_j v_{j,n}^k(b_j, y), \quad j = 1, 2, \dots, N-1, \\ l_{1,n}^k(y) &= 0, \quad r_{N,n}^k(y) = 0, \end{aligned} \quad (4.7)$$

où  $\bar{S}_j$  désigne une semi-discrétisation de  $S_j$ . Pour les deux conditions de transmission considérées, on a

- Robin:  $\bar{S}_j v_{j,n}^k = -ip \cdot v_{j,n}^k$ ,
- $S_{2p}^m$ :  $\bar{S}_j v_{j,n}^k = -i \sum_{s=0}^m a_s^m v_{j,n}^k + \sum_{s=1}^m a_s^m d_s^m \varphi_{j,s}^{n-1/2}$  sur  $\Gamma_j$ , les fonctions auxiliaires  $\varphi_{j,s}^{n-1/2}$ ,  $s = 1, 2, \dots, m$  satisfont

$$\begin{cases} \left( \frac{2i}{\Delta t} + \Delta_{\Gamma_j} + W_n + d_s^m \right) \varphi_{j,s}^{n-1/2} - v_{j,n}^k = \frac{2i}{\Delta t} \varphi_{j,s}^{n-1}, \\ \varphi_{j,s}^n = 2\varphi_{j,s}^{n-1/2} - \varphi_{j,s}^{n-1}, \\ \varphi_{j,s}^0 = 0, \end{cases}$$

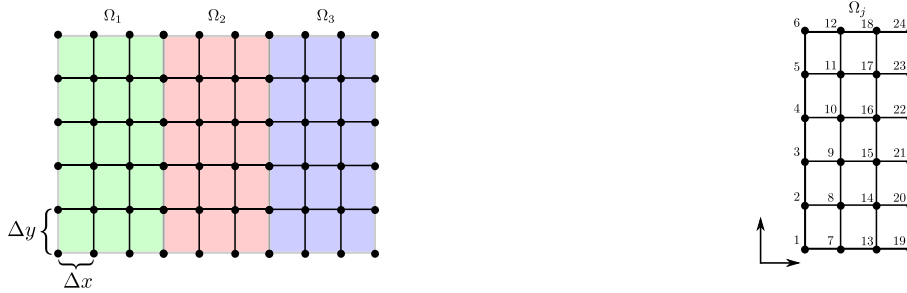
$$\text{où } a_s^m = \frac{e^{i\theta/2}}{m \cos^2(\frac{(2s-1)\pi}{4m})}, \quad d_s^m = e^{i\theta} \tan^2(\frac{(2s-1)\pi}{4m}), \quad \theta = \frac{\pi}{4}.$$

Nous pouvons réécrire les conditions de transmission semi-discrète comme

$$\begin{cases} l_{j,n}^{k+1}(y) = -r_{j-1,n}^k(y) + 2\bar{S}_j v_{j-1,n}^k(b_{j-1}, y), \quad j = 2, 3, \dots, N \\ r_{j,n}^{k+1}(y) = -l_{j+1,n}^k(y) + 2\bar{S}_j v_{j+1,n}^k(a_{j+1}, y), \quad j = 1, 2, \dots, N-1. \end{cases} \quad (4.8)$$

La méthode des éléments finis  $Q_1$  est utilisée pour approcher l'équation semi-discrète (4.5). Un exemple du maillage rectangulaire est montré dans la figure 4.3a (pour  $N = 3$ ). La numérotation des éléments commence d'abord par la coordonnée  $y$ , puis par la coordonnée  $x$  ensuite (voir figure 4.3b). La taille d'un élément discret est  $(\Delta x, \Delta y)$ . Nous désignons par

$N_x$  le nombre de nœuds dans la direction  $x$  sur chaque sous-domaine et par  $N_y$  celui dans la direction  $y$ .



(A) Éléments rectangulaire pour  $N = 3$ ,  $N_x = 4$ ,  $N_y = 6$ . (B) Numérotation des éléments sur le domaine  $\Omega_j$ .

FIGURE 4.3: Illustration des éléments rectangulaires et de la numérotation.

Sur l'élément rectangulaire, les intégrations numériques sont effectuées par la quadrature classique suivante

$$\begin{aligned} \int_a^b \int_c^d f(x, y) dx dy &\approx \frac{d-c}{6} \int_a^b (f(x, c) + 4f(x, \frac{c+d}{2}) + f(x, d)) dx \\ &= \frac{d-c}{6} \frac{b-a}{6} \left( f(a, c) + 4f(\frac{a+b}{2}, c) + f(b, c) \right. \\ &\quad \left. + 4(f(a, \frac{c+d}{2}) + 4f(\frac{a+b}{2}, \frac{c+d}{2}) + f(b, \frac{c+d}{2})) \right. \\ &\quad \left. + f(a, d) + 4f(\frac{a+b}{2}, d) + f(b, d) \right), \end{aligned}$$

et sur le bord, par la formule de Simpson

$$\int_a^b f(x) dx \approx \frac{b-a}{6} (f(a) + 4f(\frac{a+b}{2}) + f(b)).$$

Notons pour  $j = 1, 2, \dots, N$

- $\mathbb{M}_j$ : la matrice de masse,
- $\mathbb{S}_j$ : la matrice de raideur,
- $\mathbb{M}_{j, W_n}$ : la matrice de masse généralisée associée à  $\int_{\Omega_j} W_n v_{j,n}^k \phi d\Omega$ ,
- $\mathbb{M}^{\Gamma_j}$ , la matrice de masse des éléments du bord,
- $\mathbb{M}_f^{\Gamma_j}$ , la matrice de masse généralisée associée à la fonction  $f$  sur le bord,
- $\mathbb{S}^{\Gamma_j}$ : la matrice de raideur sur le bord,
- $\mathbb{S}_f^{\Gamma_j}$ : la matrice de raideur associée à la fonction  $f$  sur le bord,
- $\mathbf{v}_{j,n}^k$ : le vecteur nodal de l'interpolation de  $v_{j,n}^k$ ,



- $\mathbf{u}_{j,n}^k$ : le vecteur nodal de l'interpolation de  $u_{j,n}^k$ ,
- $\mathbf{l}_{j,n}^k$  ( $\mathbf{r}_{j,n}^k$  recp.): le vecteur nodal de l'interpolation de  $l_{j,n}^k$  ( $r_{j,n}^k$  recp) sur l'interface,
- $\varphi_s^n$  ( $\varphi_s^{n-1/2}$  recp.): le vecteur nodal de l'interpolation de  $\varphi_s^n$  ( $\varphi_s^{n-1/2}$  recp.) sur l'interface,
- $Q_{j,l}$  ( $Q_{j,r}$  recp.): la matrice de restriction des nœuds dans  $\Omega_j$  aux ceux sur l'interface  $\{x = a_j\}$  ( $\{x = b_j\}$  recp.) (voir figure 4.4),
- $Q_j$ : la matrice de restriction des nœuds dans  $\Omega_j$  aux ceux sur l'interfaces  $\{x = a_j\} \cup \{x = b_j\}$  (voir figure 4.4).

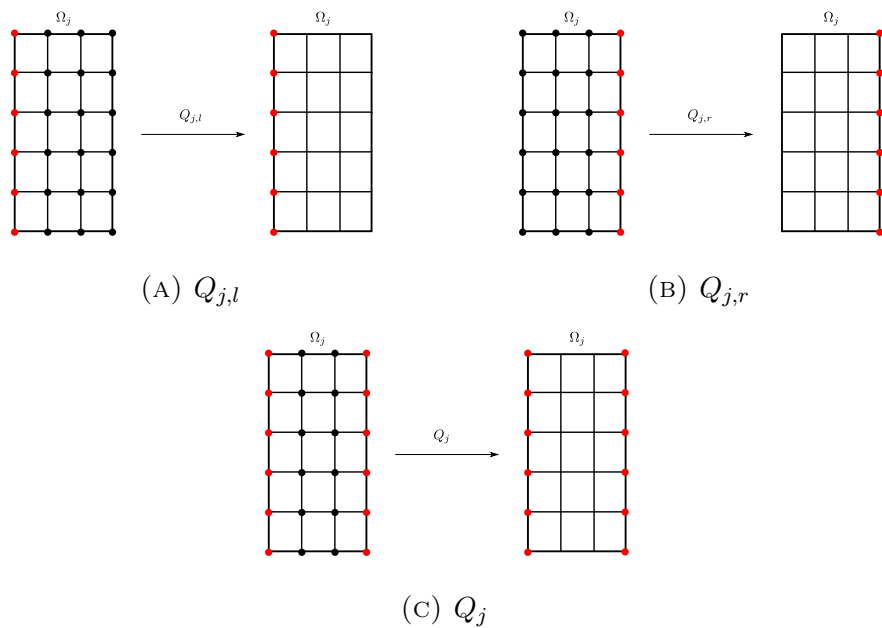


FIGURE 4.4: Les matrices des restriction.

Matriciellement, la formulation variationnelle (4.6) conduit aux équations suivantes pour les deux conditions de transmission.

- Robin

$$\left( \mathbb{A}_{j,n} + ip \cdot \mathbb{M}^{\Gamma_j} \right) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k - \mathbb{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \quad (4.9)$$

où  $\mathbb{A}_{j,n} = \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j + \mathbb{M}_{j,W_n}$ . La taille de ce système linéaire est  $N_x \times N_y$ .

- $S_{2p}^m$

$$\left\{ \begin{array}{l} \left( \mathbb{A}_{j,n} + i \left( \sum_{s=0}^m a_s^m \right) \cdot \mathbb{M}^{\Gamma_j} \right) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k, \\ + i \sum_{s=1}^m a_s^m d_s^m \mathbb{M}^{\Gamma_j} Q_j^T \boldsymbol{\varphi}_{j,s}^{n-1/2} - \mathbb{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{1}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \\ - Q_j \mathbb{M}^{\Gamma_j} \mathbf{v}_{j,n}^k + Q_j \left( \frac{2i}{\Delta t} \mathbb{M}^{\Gamma_j} - \mathbb{S}^{\Gamma_j} + \mathbb{M}_{W_n}^{\Gamma_j} + d_s^m \mathbb{M}^{\Gamma_j} \right) Q_j^T \boldsymbol{\varphi}_{j,s}^{n-1/2} \\ = \frac{2i}{\Delta t} Q_j \mathbb{M}^{\Gamma_j} Q_j^T \boldsymbol{\varphi}_{j,s}^{n-1}, \text{ pour } 1 \leq s \leq m, \\ \boldsymbol{\varphi}_s^0 = \mathbf{0} \text{ pour } 1 \leq s \leq m. \end{array} \right. \quad (4.10)$$

Cela constitue un système linéaire avec  $m+1$  inconnues  $(\mathbf{v}_{j,n}^k, \boldsymbol{\varphi}_{j,1}^{n-1/2}, \boldsymbol{\varphi}_{j,2}^{n-1/2}, \dots, \boldsymbol{\varphi}_{j,m}^{n-1/2})$ .  
Le système (4.10) s'écrit alors

$$\mathcal{A}_{j,n} \boldsymbol{\psi}_j^{n-1/2} = \frac{2i}{\Delta t} \mathcal{M}_j \boldsymbol{\psi}_j^{n-1} - \begin{pmatrix} \mathbb{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{1}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (4.11)$$

où

$$\mathcal{A}_{j,n} = \begin{pmatrix} \mathbb{A}_{j,n} + i \left( \sum_{s=0}^m a_s^m \right) \cdot \mathbb{M}^{\Gamma_j} & \mathbb{B}_{j,1} & \mathbb{B}_{j,2} & \cdots & \mathbb{B}_{j,m} \\ \mathbb{C}_j & \mathbb{D}_{j,1}^n & & & \\ \mathbb{C}_j & & \mathbb{D}_{j,2}^n & & \\ \vdots & & & \ddots & \\ \mathbb{C}_j & & & & \mathbb{D}_{j,m}^n \end{pmatrix}, \quad \boldsymbol{\psi}_j^{n-1/2} = \begin{pmatrix} \mathbf{v}_{j,n}^k \\ \boldsymbol{\varphi}_{j,1}^{n-1/2} \\ \boldsymbol{\varphi}_{j,2}^{n-1/2} \\ \vdots \\ \boldsymbol{\varphi}_{j,m}^{n-1/2} \end{pmatrix},$$

$$\mathcal{M}_j = \begin{pmatrix} \mathbb{M}_j & & & & \\ & Q_j \mathbb{M}^{\Gamma_j} Q_j^T & & & \\ & & Q_j \mathbb{M}^{\Gamma_j} Q_j^T & & \\ & & & \ddots & \\ & & & & Q_j \mathbb{M}^{\Gamma_j} Q_j^T \end{pmatrix}, \quad \boldsymbol{\psi}_j^{n-1} = \begin{pmatrix} \mathbf{u}_{j,n-1}^k \\ \boldsymbol{\varphi}_{j,1}^{n-1} \\ \boldsymbol{\varphi}_{j,2}^{n-1} \\ \vdots \\ \boldsymbol{\varphi}_{j,m}^{n-1} \end{pmatrix},$$

avec

$$\begin{aligned} \mathbb{B}_{j,s} &= -i a_s^m d_s^m \mathbb{M}^{\Gamma_j} Q_j^T, \quad 1 \leq s \leq m, \\ \mathbb{C}_j &= -Q_j \mathbb{M}^{\Gamma_j}, \\ \mathbb{D}_{j,s}^n &= Q_j \left( \frac{2i}{\Delta t} \mathbb{M}^{\Gamma_j} - \mathbb{S}^{\Gamma_j} + \mathbb{M}_{W_n}^{\Gamma_j} + d_s^m \mathbb{M}^{\Gamma_j} \right) Q_j^T, \quad 1 \leq s \leq m. \end{aligned}$$

La taille du système linéaire (4.11) est  $N_x \times N_y + m \cdot 2N_y$  pour  $j = 2, 3, \dots, N-1$  et  $N_x \times N_y + m \cdot N_y$  pour  $j = 1, N$ . Numériquement, on résout (4.11) pour obtenir  $\mathbf{v}_{j,n}^k$  et  $\boldsymbol{\varphi}_{j,s}^{n-1/2}$ ,  $s = 1, 2, \dots, m$ . Il est clair que la condition de transmission  $S_{2p}^m$  consomme plus de mémoire que la condition de transmission de Robin. L'implémentation de  $S_{2p}^m$  est aussi plus coûteuse quand l'ordre  $m$  augmente.

Finalement, nous devons ainsi à résoudre pour chaque sous-domaine et chaque pas de temps les  $N$  systèmes linéaires suivants

- Robin

$$\begin{cases} \mathbf{r}_{j-1,n}^{k+1} = -\mathbf{l}_{j,n}^k + 2Q_{j,l} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 2, 3, \dots, N, \\ \mathbf{l}_{j+1,n}^{k+1} = -\mathbf{r}_{j,n}^k + 2Q_{j,r} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 1, 2, \dots, N-1, \\ \tilde{S}_j \mathbf{v}_{j,n}^k = -ip \cdot \mathbf{v}_{j,n}^k, \\ \text{Équation (4.9),} \end{cases} \quad (4.12)$$

- $S_{2p}^m$

$$\begin{cases} \mathbf{r}_{j-1,n}^{k+1} = -\mathbf{l}_{j,n}^k + 2Q_{j,l} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 2, 3, \dots, N, \\ \mathbf{l}_{j+1,n}^{k+1} = -\mathbf{r}_{j,n}^k + 2Q_{j,r} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 1, 2, \dots, N-1, \\ \tilde{S}_j \mathbf{v}_{j,n}^k = -i \left( \sum_{s=0}^m a_s^m \right) \mathbf{v}_{j,n}^k + i \sum_{s=1}^m a_s^m d_s^m \boldsymbol{\varphi}_{j,s}^{n-1/2}, \\ \text{Équation (4.11).} \end{cases} \quad (4.13)$$

*Remarque 4.1.* Pour la méthode de décomposition en espace, le domaine  $]t_{n-1}, t_n[ \times \Omega_0, t_n = n\Delta t$  ne contient qu'une étape de temps. Au temps  $t_n$ , les  $N$  problèmes sont

$$\text{Robin : } \left( \mathbb{A}_{j,n} + ip \cdot \mathbb{M}^{\Gamma_j} \right) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1} - \mathbb{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \quad (4.14)$$

$$S_{2p}^m : \begin{cases} \left( \mathbb{A}_{j,n} + i \left( \sum_{s=0}^m a_s^m \right) \cdot \mathbb{M}^{\Gamma_j} \right) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1} \\ + i \sum_{s=1}^m a_s^m d_s^m \mathbb{M}^{\Gamma_j} Q_j^T \boldsymbol{\varphi}_{j,s}^{1/2} - \mathbb{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \\ -Q_j \mathbb{M}^{\Gamma_j} \mathbf{v}_{j,n}^k + Q_j \left( \frac{2i}{\Delta t} \mathbb{M}^{\Gamma_j} - \mathbb{S}^{\Gamma_j} + \mathbb{M}_{W_n}^{\Gamma_j} + d_s^m \mathbb{M}^{\Gamma_j} \right) Q_j^T \boldsymbol{\varphi}_{j,s}^{1/2} \\ = 0, \text{ pour } 1 \leq s \leq m, \end{cases} \quad (4.15)$$

où  $\mathbf{u}_{j,n-1}$  est la solution numérique au temps  $t_{n-1}$ .

*Remarque 4.2.* Nous donnons dans cette remarque les autres conditions de transmission que nous pouvons déduire des conditions aux limites qui sont disponible dans [51]. Ces conditions mettent en jeu la courbure des frontières et le potentiel (linéaire ou nonlinéaire). Dans le cadre de la décomposition géométrique des domaines que nous considérons ici, celle ci ne fait intervenir que des frontières verticales. En outre, le potentiel considéré ici est  $V = 0$ . La forme des conditions de transmission est donc simplifiée et nous avons donc deux autres conditions de transmission:

$$\begin{aligned} S_{2T}, \text{ Ordre 1: } S_j u &= e^{-i\pi/4} \partial_t^{1/2} u, \\ S_{2T}, \text{ Ordre 3: } S_j u &= e^{-i\pi/4} \partial_t^{1/2} u - e^{i\pi/4} \frac{\Delta_{\Gamma_j}}{2} I_t^{1/2} u. \end{aligned}$$





Ensuite, en remplaçant  $\mathbf{v}_j^k$  par  $g_j^k$  et  $\mathbf{F}_j$  dans (4.12) et en utilisant (4.19), nous pouvons vérifier (4.18) facilement et avoir les expressions explicites des blocs

$$\begin{aligned} X^{j,1} &= -I - 2ip \cdot \mathbf{Q}_{j,l}(\mathbf{A}_j - \mathbf{B}_j)^{-1} \mathbf{M}^{\Gamma_j} \mathbf{Q}_{j,l}^T, \\ X^{j,2} &= -2ip \cdot \mathbf{Q}_{j,l}(\mathbf{A}_j - \mathbf{B}_j)^{-1} \mathbf{M}^{\Gamma_j} \mathbf{Q}_{j,r}^T, \\ X^{j,3} &= -2ip \cdot \mathbf{Q}_{j,r}(\mathbf{A}_j - \mathbf{B}_j)^{-1} \mathbf{M}^{\Gamma_j} \mathbf{Q}_{j,l}^T, \\ X^{j,4} &= -I - 2ip \cdot \mathbf{Q}_{j,r}(\mathbf{A}_j - \mathbf{B}_j)^{-1} \mathbf{M}^{\Gamma_j} \mathbf{Q}_{j,r}^T. \end{aligned} \quad (4.20)$$

□

**Proposition 4.4.** *Si l'on considère la condition de transmission  $S_{2p}^m$ , et sous l'hypothèse que les systèmes linéaires (4.11) ne sont pas singuliers, alors le problème prend la forme globale du problème d'interface (4.16).*

*Preuve.* La démonstration est très similaire à celle de la proposition précédente. D'abord le système linéaire (4.11) est réécrit d'une manière globale:

$$\begin{aligned} (\mathbf{A}_1 - \mathbf{B}_1) \mathbf{v}_1^k &= \mathbf{F}_1 - \mathbf{M}^{\Gamma_1} \mathbf{Q}_{1,r}^T g_1^k, \\ (\mathbf{A}_j - \mathbf{B}_j) \mathbf{v}_j^k &= \mathbf{F}_j - \mathbf{M}^{\Gamma_j} \mathbf{Q}_j^T g_j^k, \quad j = 2, 3, \dots, N, \\ (\mathbf{A}_N - \mathbf{B}_N) \mathbf{v}_N^k &= \mathbf{F}_N - \mathbf{M}^{\Gamma_N} \mathbf{Q}_{N,l}^T g_N^k, \end{aligned} \quad (4.21)$$

où  $\mathbf{B}_j$ ,  $j = 1, 2, \dots, N$  est donnée par (4.24). Nous montrons maintenant l'expression de la matrice  $\mathbf{B}_j$ . D'après la formule (4.10), pour  $s = 1, 2, \dots, m$ ,  $n = 1, 2, \dots, N_T$ , on a

$$\begin{aligned} \mathbb{C}_j \mathbf{v}_{j,n}^k + \mathbb{D}_{j,s}^n \varphi_{j,s}^{n-1/2} &= \frac{2i}{\Delta t} Q_j \mathbf{M}^{\Gamma_j} Q_j^T \varphi_{j,s}^{n-1}, \quad n \geq 1, \\ \Rightarrow \varphi_{j,s}^{n-1/2} &= -(\mathbb{D}_{j,s}^n)^{-1} \mathbb{C}_j \mathbf{v}_{j,n}^k + (\mathbb{D}_{j,s}^n)^{-1} \frac{2i}{\Delta t} Q_j \mathbf{M}^{\Gamma_j} Q_j^T \varphi_{j,s}^{n-1}, \\ \varphi_{j,s}^{n-1} &= 2\varphi_{j,s}^{n-3/2} - \varphi_{j,s}^{n-2}, \quad \varphi_{j,s}^0 = 0, \quad n \geq 2. \end{aligned}$$

Par récurrence, nous pouvons avoir une expression de  $\varphi_{j,s}^{n-1/2}$ :

$$\varphi_{j,s}^{n-1/2} = \sum_{p=1}^n \mathbb{L}_{j,s}^{n,p} \mathbf{v}_{j,p}^k, \quad (4.22)$$

où  $\mathbb{L}_{j,s}^{n,p}$  sont matrices. En remplaçant  $\varphi_{j,s}^{n-1/2}$  par  $\mathbf{v}_{j,p}^k$  dans la première formule de (4.10), nous avons alors

$$\left( \mathbb{A}_{j,n} + i \left( \sum_{s=0}^m a_s^m \right) \cdot \mathbf{M}^{\Gamma_j} \right) \mathbf{v}_{j,n}^k + \sum_{p=1}^n \mathbb{Y}_j^{n,p} \mathbf{v}_{j,p}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k - \mathbf{M}^{\Gamma_j} Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \quad (4.23)$$

où  $\mathbb{Y}_j^{n,p} := -i \sum_{s=1}^m a_s^m d_s^m \mathbf{M}^{\Gamma_j} Q_j^T \mathbb{L}_{j,s}^{n,p}$  car

$$-i \sum_{s=1}^m a_s^m d_s^m \mathbf{M}^{\Gamma_j} Q_j^T \cdot \sum_{p=1}^n \mathbb{L}_{j,s}^{n,p} \mathbf{v}_{j,p}^k = -i \sum_{p=1}^n \left( \sum_{s=1}^m a_s^m d_s^m \mathbf{M}^{\Gamma_j} Q_j^T \mathbb{L}_{j,s}^{n,p} \right) \mathbf{v}_{j,p}^k.$$



où  $x_{n,s}^{j,1}, x_{n,s}^{j,2}, x_{n,s}^{j,3}, x_{n,s}^{j,4} \in \mathbb{C}^{N_y \times N_y}$ . Cette structure est décrite ci-dessous pour 3 pas de temps et six nœuds sur l'interface entre deux sous-domaines. Chaque sous-diagonale a un bloc identique. Nous présentons cette propriété mathématiquement dans les propositions 4.6 et 4.7 avec la condition de transmission de Robin et  $S_{2p}^m$  respectivement. Les démonstrations sont similaires à celles de l'équation de Schrödinger en une dimension démontrées dans le chapitre 2. La différence formelle est que les flux  $l_{j,n}^k$  et  $r_{j,n}^k$  sont des scalaires en une dimension, mais ici  $\mathbf{l}_{j,n}^k$  et  $\mathbf{r}_{j,n}^k$  sont vecteurs.

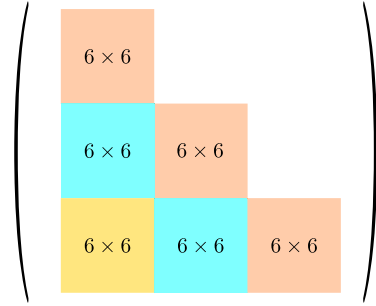


FIGURE 4.5: La structure des blocs,  $N_T = 3$ ,  $N_y = 6$ .

**Proposition 4.6.** *Si l'on considère la condition de transmission de Robin, si  $V = V(x, y)$  et si on suppose que les matrices  $\mathbb{A}_{j,s} + ip \cdot \mathbb{M}^{\Gamma_j}$ ,  $s = 1, 2, \dots, N_T$  ne sont pas singulières, alors les matrices  $X^{1,4}$ ,  $X^{j,1}$ ,  $X^{j,2}$ ,  $X^{j,3}$ ,  $X^{j,4}$ ,  $j = 2, 3, \dots, N-1$  et  $X^{N,1}$  sont des matrices triangulaires inférieures par bloc qui satisfont*

$$\begin{aligned}
 x_{n,s}^{1,4} &= x_{n-1,s-1}^{1,4}, \\
 x_{n,s}^{j,1} &= x_{n-1,s-1}^{j,1}, \quad x_{n,s}^{j,2} = x_{n-1,s-1}^{j,2}, \\
 x_{n,s}^{j,3} &= x_{n-1,s-1}^{j,3}, \quad x_{n,s}^{j,4} = x_{n-1,s-1}^{j,4}, \quad j = 2, 3, \dots, N-1, \\
 x_{n,s}^{N,1} &= x_{n-1,s-1}^{N,1},
 \end{aligned} \tag{4.26}$$

pour  $2 \leq s \leq n \leq N_T$ .

**Proposition 4.7.** *Pour la condition de transmission  $S_{2p}^m$ , si  $V = V(x, y)$  et sous l'hypothèse que les systèmes linéaires (4.11) ne sont pas singuliers, alors les matrices  $X^{1,4}$ ,  $X^{j,1}$ ,  $X^{j,2}$ ,  $X^{j,3}$ ,  $X^{j,4}$ ,  $j = 2, 3, \dots, N-1$  et  $X^{N,1}$  sont des matrices triangulaires inférieures par bloc qui satisfont (4.26).*

Nous considérons maintenant la structure des sous blocs de  $\mathcal{L}$  pour  $V = 0$ .

**Proposition 4.8.** *Supposons que les matrices  $\mathbb{A}_{j,s} + ip \cdot \mathbb{M}^{\Gamma_j}$ ,  $s = 1, 2, \dots, N_T$  ne sont pas singulières avec la condition de transmission de Robin, si le potentiel  $V = 0$  et les tailles des sous-domaines  $\Omega_j$  sont égaux ( $b_1 - a_1 = b_2 - a_2 = \dots = b_N - a_N$ ), alors les sous blocs dans la matrice  $\mathcal{L}$  satisfont*

$$\begin{aligned}
 X^{2,1} &= X^{3,1} = \dots = X^{N,1}, \quad X^{2,2} = X^{3,2} = \dots = X^{N-1,2}, \\
 X^{2,3} &= X^{3,3} = \dots = X^{N-1,3}, \quad X^{1,4} = X^{2,4} = \dots = X^{N-1,4}.
 \end{aligned} \tag{4.27}$$



*Preuve.* Sous les hypothèses, il est facile de voir que

$$\begin{aligned} \mathbb{A}_1 = \mathbb{A}_2 = \dots = \mathbb{A}_N, \quad \mathbb{M}_{j,W_n} = 0, \quad \mathbb{M}^{\Gamma_1} = \mathbb{M}^{\Gamma_2} = \dots = \mathbb{M}^{\Gamma_N}, \\ \mathbb{Q}_{1,l} = \mathbb{Q}_{2,l} = \dots = \mathbb{Q}_{N,l}, \quad \mathbb{Q}_{1,r} = \mathbb{Q}_{2,r} = \dots = \mathbb{Q}_{N,r}, \end{aligned}$$

ainsi

$$\mathbf{A}_1 = \mathbf{A}_2 = \dots = \mathbf{A}_N, \quad \mathbf{M}^{\Gamma_1} = \mathbf{M}^{\Gamma_2} = \dots = \mathbf{M}^{\Gamma_N}, \quad \mathbf{Q}_1 = \mathbf{Q}_2 = \dots = \mathbf{Q}_N.$$

La conclusion est directe de (4.20).  $\square$

**Proposition 4.9.** *Sous l'hypothèse que les systèmes linéaires (4.11) ne sont pas singuliers avec la condition de transmission  $S_{2p}^m$ , si le potentiel  $V = 0$  et les tailles des sous-domaines  $\Omega_j$  sont égales ( $b_1 - a_1 = b_2 - a_2 = \dots = b_N - a_N$ ), alors les sous blocs dans la matrice  $\mathcal{L}$  satisfont (4.27).*

*Preuve.* La démonstration est presque identique à celle de la proposition 4.8. Nous pouvons voir la conclusion directement en calculant les expressions explicites des blocs dans la proposition 4.4.  $\square$

## 4.4 Nouveaux algorithmes

Les propositions dans la sous-section précédente nous donne une possibilités de construire le problème d'interface en deux dimensions qui ne nécessite pas trop de calcul si le potentiel  $V = 0$ . Nous présentons dans cette partie les algorithmes nouveaux pour la méthode SWR et la méthode de décomposition en espace pour l'équation de Schrödinger sans potentiel.

### 4.4.1 Nouvel algorithme SWR

La nouvel algorithme SWR que nous proposons ici est identique à celui de l'algorithme 3 pour l'équation de Schrödinger en une dimension

---

#### Algorithme 8 : Nouvel algorithme SWR

---

- 1: Construire  $\mathcal{L}$  et  $d$  globales en temps dans  $(I - \mathcal{L}) = d$  explicitement.
  - 2: Résoudre le système linéaire  $(I - \mathcal{L}) = d$  par méthode itérative.
  - 3: Résoudre l'équation de Schrödinger sur chaque sous-domaine en  $]0, T[$  en utilisant les conditions aux limites issues de l'étape 2.
- 

Nous présentons maintenant la construction de la matrice  $\mathcal{L}$  et le vecteur  $d$ . Nous utilisons la formule (4.12) avec la condition de transmission de Robin et la formule (4.13) avec la condition de transmission  $S_{2p}^m$

$$\text{entrées : } \mathbf{l}_{j,n}^k, \mathbf{r}_{j,n}^k \longrightarrow \begin{array}{l} \text{Robin : (4.12)} \\ S_{2p}^m : (4.13) \end{array} \longrightarrow \text{sorties : } \mathbf{l}_{j+1,n}^{k+1}, \mathbf{r}_{j-1,n}^{k+1}.$$

Il est facile de voir d'après (4.16) que

$$d = \begin{pmatrix} d_{1,r} \\ d_{2,l} \\ d_{2,r} \\ \vdots \\ d_{N,l} \end{pmatrix} = \mathcal{R} \cdot \mathbf{0}, \text{ où } d_{j-1,r} = \begin{pmatrix} \mathbf{r}_{j-1,1}^{k+1} \\ \mathbf{r}_{j-1,2}^{k+1} \\ \vdots \\ \mathbf{r}_{j-1,N_T}^{k+1} \end{pmatrix}, \quad d_{j+1,l} = \begin{pmatrix} \mathbf{l}_{j+1,1}^{k+1} \\ \mathbf{l}_{j+1,2}^{k+1} \\ \vdots \\ \mathbf{l}_{j+1,N_T}^{k+1} \end{pmatrix},$$

les vecteurs  $\mathbf{r}_{j-1,n}^{k+1}$  et  $\mathbf{l}_{j+1,n}^{k+1}$ ,  $n = 1, 2, \dots, N_T$  sont obtenus par (4.12) ou (4.13) avec

$$\mathbf{l}_{j,n}^k = \mathbf{r}_{j,n}^k = \mathbf{0}, \quad n = 1, 2, \dots, N_T, \quad j = 1, 2, \dots, N.$$

L'équation de Schrödinger est résolue sur chaque sous-domaine pour toutes les étapes du temps une fois. Le vecteur  $d$  est stocké de manière distribuée en utilisant la librairie PETSc comme nous le faisons dans la sous-section 2.4.2.

$$d = 2 \begin{pmatrix} d_{1,r} \\ \vdots \\ d_{j,l} \\ d_{j,r} \\ \vdots \\ d_{N,l} \end{pmatrix} \begin{array}{l} \} \text{MPI } 0, N_T \times N_y \text{ éléments} \\ \\ \} \text{MPI } j-1, 2N_T \times N_y \text{ éléments} \\ \\ \} \text{MPI } N-1, N_T \times N_y \text{ éléments} \end{array}$$

Nous voyons dans les propositions 4.8 et 4.9 que si les géométries des sous-domaines sont identiques et le potentiel est nul, alors il suffit de calculer quatre sous blocs pour connaître explicitement la matrice  $\mathcal{L}$ . Sans perte de généralité, les blocs  $X^{2,1}$ ,  $X^{2,2}$ ,  $X^{2,3}$  et  $X^{2,4}$  sont construits.

Nous définissons les vecteurs élémentaires  $\mathbf{e}_s = (0, 0, \dots, 1, \dots, 0)$ , tels que tous les éléments de  $\mathbf{e}_s$  sont nuls sauf le  $j$ -ième qui vaut un. Les colonnes  $j, j = 1, 2, \dots, N_y$  de  $X^{2,1}$  et  $X^{2,3}$  sont

$$X^{2,1} \begin{pmatrix} \mathbf{e}_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{1,1}^{k+1} \\ \mathbf{r}_{1,2}^{k+1} \\ \vdots \\ \mathbf{r}_{1,N_T}^{k+1} \end{pmatrix} - d_{1,r} \text{ et } X^{2,3} \begin{pmatrix} \mathbf{e}_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{l}_{3,1}^{k+1} \\ \mathbf{l}_{3,2}^{k+1} \\ \vdots \\ \mathbf{l}_{3,N_T}^{k+1} \end{pmatrix} - d_{3,l},$$

où les vecteurs  $\mathbf{r}_{1,n}^{k+1}$  et  $\mathbf{l}_{3,n}^{k+1}$ ,  $n = 1, 2, \dots, N_T$  sont obtenus par (4.12) ou (4.13) avec

$$\begin{aligned} \mathbf{l}_{2,1}^k &= \mathbf{e}_j, \quad \mathbf{l}_{2,n}^k = 0, \quad n = 2, 3, \dots, N_T, \\ \mathbf{r}_{2,n}^k &= \mathbf{0}, \quad n = 1, 2, \dots, N_T. \end{aligned}$$

De la même façon, les colonnes  $j, j = 1, 2, \dots, N_T$  des  $X^{2,2}$  et  $X^{2,4}$  sont

$$X^{2,2} \begin{pmatrix} \mathbf{e}_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{1,1}^{k+1} \\ \mathbf{r}_{1,2}^{k+1} \\ \vdots \\ \mathbf{r}_{1,N_T}^{k+1} \end{pmatrix} - d_{1,r} \text{ et } X^{2,4} \begin{pmatrix} \mathbf{e}_j \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{l}_{3,1}^{k+1} \\ \mathbf{l}_{3,2}^{k+1} \\ \vdots \\ \mathbf{l}_{3,N_T}^{k+1} \end{pmatrix} - d_{3,l},$$

où les vecteurs  $\mathbf{r}_{1,n}^{k+1}$  et  $\mathbf{l}_{3,n}^{k+1}$ ,  $n = 1, 2, \dots, N_T$  sont obtenus par (4.12) ou (4.13) avec

$$\begin{aligned} \mathbf{l}_{2,n}^k &= \mathbf{0}, \quad n = 1, 2, \dots, N_T, \\ \mathbf{r}_{2,1}^k &= \mathbf{e}_j, \quad \mathbf{r}_{2,n}^k = \mathbf{0}, \quad n = 2, 3, \dots, N_T. \end{aligned}$$

Concrètement, l'équation de Schrödinger est résolue sur un seul sous-domaine (ex.  $\Omega_2$ ) pour toutes les étapes du temps  $2N_y$  fois pour construire la matrice  $\mathcal{L}$ . Les résolutions sont tous indépendantes. Nous fixons 1 processus MPI/domaine. Pour construire la matrice  $\mathcal{L}$ , nous utilisons les  $N$  processus MPI pour résoudre l'équation sur un seul sous-domaine (ex.  $\Omega_2$ )  $2N_y$  fois. Chaque processus MPI résout l'équation de Schrödinger sur un seul sous-domaine au maximum

$$N_{\text{mpi}} := \left\lceil \frac{2N_y}{N} \right\rceil + 1 \text{ fois,}$$

où  $[x]$  est la partie entière de  $x$ . Cette construction est super-scalable en théorie. Si  $N$  est doublé, alors la taille de sous-domaine est divisée par deux et  $N_{\text{mpi}}$  est aussi environ divisé par deux.

La matrice transposée de  $\mathcal{L}$  est stockée de manière distribuée sous la forme de la librairie PETSc (voir 2.30). La taille de chaque bloc est  $(N_T \times N_y) \times (N_T \times N_y)$ . Chaque bloc contient  $(N_T + 1) \times N_T/2 \times N_y^2$  éléments non nuls.

#### 4.4.2 Nouvel algorithme de décomposition en espace

Nous présentons ici le nouvel algorithme de la méthode de décomposition en espace. Les problèmes d'interface des problèmes

$$\begin{cases} \mathcal{L}u = 0, & (t, x, y) \in ]t_{n-1}, t_n[ \times \Omega_0, \\ u(0, x, y) = u_{0,n-1}(x, y), & (x, y) \in \Omega_0, \end{cases}$$

sont notés par  $(I - \mathcal{L}_{t_n})g_{t_n} = d_{t_n}$ . Considérant (4.20), nous pouvons voir que

$$\mathcal{L}_{t_1} = \mathcal{L}_{t_2} = \dots = \mathcal{L}_{t_{N_T}}.$$

---

#### Algorithme 9 : Nouvel algorithme de décomposition en espace

---

1: Construire  $\mathcal{L}_{t_1}$  en  $(0, \Delta t) \times \Omega_0$  explicitement.

2: La donnée initiale est  $u_{0,0}$ .

**for**  $n = 1, 2, \dots, N_T$  **do**

    2.1: Construire  $d_{t_n}$  en temps  $t_n$ ,

    2.2: Résoudre le système linéaire  $(I - \mathcal{L}_{t_1})g_{t_n} = d_{t_n}$  par méthode itérative, le vecteur initial est choisi comme  $g_{t_{n-1}}$ .

    2.3: Résoudre l'équation de Schrödinger sur chaque sous-domaine en  $]t_{n-1}, t_n[$  en utilisant les conditions aux limites issues de l'étape 2.2 pour obtenir  $u_{0,n}$ .

---

Concernant la construction du problème d'interface, pour la méthode SWR, on résout l'équation de Schrödinger sur un sous-domaine  $2N_y$  fois en temps  $]0, T[$ . Mais pour la

méthode de décomposition en espace, il suffit de résoudre l'équation de Schrödinger sur un sous-domaine  $2N_y$  fois en temps  $(0, \Delta t)$ .

## 4.5 Résultats numériques

Le domaine  $\Omega_0 = ]-16, 16[ \times ]-8, 8 - \Delta y[$  est décomposé en  $N$  sous domaines égaux (ex. pour  $N = 3$  voir figure 4.3a). La taille d'un élément est  $\Delta x \times \Delta y$ . Nous utilisons deux maillages différents

$$\begin{aligned} \Delta x &= 1/128, \quad \Delta y = 1/8, \\ \Delta x &= 1/2048, \quad \Delta y = 1/128. \end{aligned}$$

Avec le premier maillage, il est possible de résoudre l'équation de Schrödinger sur le domaine complet  $\Omega_0$  avec une machine standard (32 Go de mémoire/nœud). Ainsi, nous pouvons voir si les algorithmes parallèles permettent de diminuer le temps de calcul total. Nous allons aussi tester la scalabilité forte jusqu'à 1024 sous-domaines. Pour éviter que chaque sous-domaine ne contienne pas assez de nœuds si  $N$  est grand, le deuxième maillage est utilisé. La donnée initiale pour les chapitres 4.5.1, 4.5.2 et 4.5.3 est une gaussienne (voir figure 4.6)

$$u_0(x, y) = e^{-x^2 - y^2 - 0.5ix}. \quad (4.28)$$

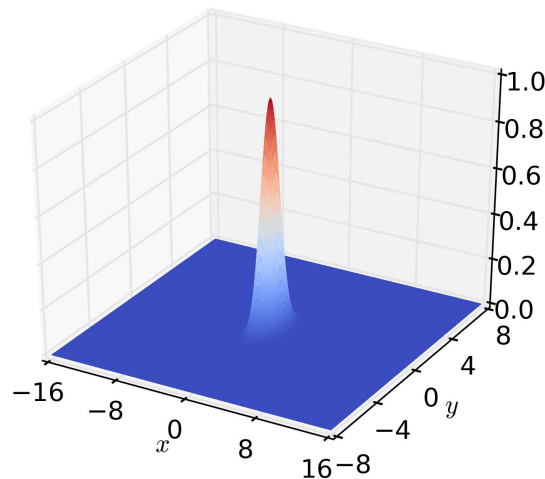


FIGURE 4.6: La donnée initiale  $|u_0|$ ,  $u_0 = e^{-x^2 - y^2 - 0.5ix}$ .

Nous comparons d'abord la méthode SWR et la méthode de décomposition en espace. Ensuite, les deux conditions de transmissions et les algorithmes sont comparés numériquement dans la sous-section suivante. Puis, nous nous intéressons à l'influence du paramètre dans les conditions de transmission. À la fin, on montre des tests avec une donnée initiale différente qui donne lieu à une solution qui se déplace sur la droite et subit une dispersion.

### 4.5.1 La méthode SWR vs. la méthode décomposition en espace

Nous comparons la méthode SWR et la méthode décomposition en espace dans le cadre de l'algorithme classique. Le pas de temps est fixé à  $\Delta t = 0.01$ . Le maillage est  $\Delta x = 1/128$ ,  $\Delta y = 1/8$  et la donnée initiale est donnée par (4.28). Nous considérons différents temps finaux  $T$ . Les deux méthodes sont appliquées à  $]0, T[ \times \Omega_0$ . Nous désignons par

- $N_{iter}^S$ : le nombre d'itérations requises avec la méthode SWR,
- $N_{iter}^D$ : le nombre d'itérations requises pour la première étape du temps avec la méthode de décomposition en espace (les nombres d'itérations sont similaire pour chaque étape du temps),
- $T^S$ : le temps de calcul total avec la méthode SWR,
- $T^D$ : le temps de calcul total avec la méthode de décomposition en espace.

Le tableau 4.1 présente les nombres d'itérations et les temps de calculs avec les deux méthodes pour  $N = 2$  et  $N = 32$ , où la condition de transmission est  $S_{2p}^m$ ,  $m = 5$ , le vecteur initial est le vecteur zéro et la méthode Gmres est utilisée sur le problème d'interface. Nous pouvons constater que le nombre d'itérations est plus important avec la méthode SWR en augmentant le temps final  $T$ . L'équation de Schrödinger est résolue à chaque étape de temps  $N_{iter}^S$  fois. En revanche, avec la méthode de décomposition en espace, l'équation de Schrödinger est résolue sur tous les étapes du temps environ  $N_{iter}^D$  fois. Il est clair que  $N_{iter}^S > N_{iter}^D$ . Ainsi, le temps de calcul pour la méthode SWR est plus important que pour la méthode de décomposition en espace.

$T$	$N = 2$				$N = 32$			
	$N_{iter}^S$	$N_{iter}^D$	$T^S$	$T^D$	$N_{iter}^S$	$N_{iter}^D$	$T^S$	$T^D$
0.05	17	9	17.6	12.3	17	10	1.5	1.9
0.1	25	9	44.5	21.5	25	10	3.5	1.8
0.15	30	9	78.4	30.9	31	10	6.4	2.6
0.2	44	9	147.9	40.0	45	10	11.8	3.4
0.25	51	9	215.0	49.3	52	10	16.9	4.1
0.3	55	9	271.0	58.6	55	10	21.2	4.8
0.35	58	9	332.1	67.9	59	10	26.3	5.6
0.4	61	9	402.9	77.2	62	10	32.0	6.3
0.45	64	9	474.5	86.4	65	10	37.6	7.1
0.5	68	9	557.6	96.1	73	10	46.3	7.8

TABLEAU 4.1: Les nombres d'itérations et les temps de calcul avec la méthode SWR et la méthode décomposition en espace,  $N = 2, 32$ .

Dans les sous-sections suivantes, nous ne considérons plus que la méthode de décomposition en espace car elle nécessite moins de temps de calcul.

## 4.5.2 Comparaisons des algorithmes classiques et nouveaux

Nous nous intéressons dans cette partie aux performances (nombre d'itérations et temps de calcul) des algorithmes classiques et nouveaux avec les deux conditions de transmission. De même, nous observons la scalabilité forte du calcul distribué des deux algorithmes. Les deux algorithmes et les deux conditions de transmission sont comparées dans le cadre de la méthode de décomposition en espace.

Le temps final est fixé à  $T = 0.5$  et le pas de temps est  $\Delta t = 0.01$ . La méthode Gmres est utilisée sur le problème d'interface dans cette partie. Le vecteur initial est le vecteur zéro. Comme n'y a pas de résultat sur le choix optimal du paramètre  $p$  dans la condition de transmission de Robin, nous faisons des tests avec des  $p$  différents pour trouver le  $p$  optimal numérique. Nous verrons dans la sous-section suivante que le nombre d'itérations et le temps de calcul ne sont pas sensibles à  $p$  en utilisant la méthode Gmres sur le problème d'interface. Ainsi, il est difficile de faire le choix d'un  $p$  optimal. Nous prenons donc ici  $p = 15$  pour le maillage  $\Delta x = 1/128$ ,  $\Delta y = 1/8$  et  $p = 10$  pour le maillage  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ . Pour la condition de transmission  $S_{2p}^m$ , nous prenons  $m = 5$  d'après les tests de la sous-section suivante.

Nous considérons d'abord le maillage grossier ( $\Delta x = 1/128$ ,  $\Delta y = 1/8$ ). La figure 4.7 présente l'historique de la convergence de la première étape du temps pour  $N = 2$  et  $N = 32$ . Nous montrons aussi le nombre d'itérations requises pour la première étape du temps et les temps de calcul totaux dans le tableau 4.2 pour  $N = 2, 4, 8, 16, 32$ . Nous pouvons constater que

- le nouvel algorithme permet un gain en temps de calcul comparé à l'algorithme classique,
- le nombre d'itérations est indépendant du nombre de sous-domaine et l'algorithme est scalable,
- l'algorithme converge plus vite avec la condition de transmission  $S_{2p}^m$ , mais il prend plus de temps de calcul car l'application de la condition de transmission  $S_{2p}^m$  est plus coûteuse que la condition de transmission de Robin.

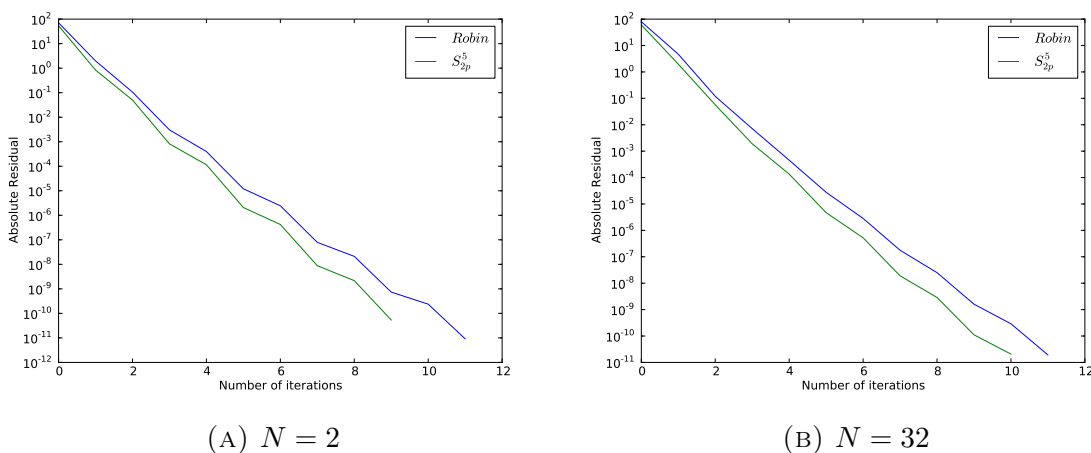


FIGURE 4.7: Historique de la convergence du premier étape du temps pour  $N = 2, 32$ .

$N$		2	4	8	16	32
Nombre d'itération	Robin	11	11	11	11	11
	$S_{2p}^5$	9	9	9	9	10
Temps de calcul	Robin cls.	63.1	32.6	17.5	11.0	5.4
	Robin new	30.0	9.7	4.4	2.5	1.3
	$S_{2p}^5$ cls.	96.1	49.8	26.7	15.0	7.8
	$S_{2p}^5$ new	38.2	14.7	6.6	3.4	1.8

TABLEAU 4.2: Nombre d'itérations de la première étape du temps et temps de calcul total en secondes,  $T = 0.5$ ,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

Nous faisons ensuite les tests avec un maillage plus fin  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ . Le domaine complet est décomposé en  $N = 128, 256, 512, 1024$  sous-domaines. Nous présentons dans la figure 4.8 l'historique de la convergence pour  $N = 256, 1024$ , dans le tableau 4.3 les nombres d'itérations et les temps de calcul totaux. Nous pouvons voir que les algorithmes classiques et nouveaux ne sont pas très scalable car le nombre d'itérations augmente avec le nombre de sous-domaines. Le nouvel algorithme prend cependant moins de temps de calcul que l'algorithme classique. De plus, le nombre d'itérations requises pour la condition de transmission  $S_{2p}^m$  est inférieur à celui de la condition de transmission de Robin, mais les temps de calcul sont similaires.

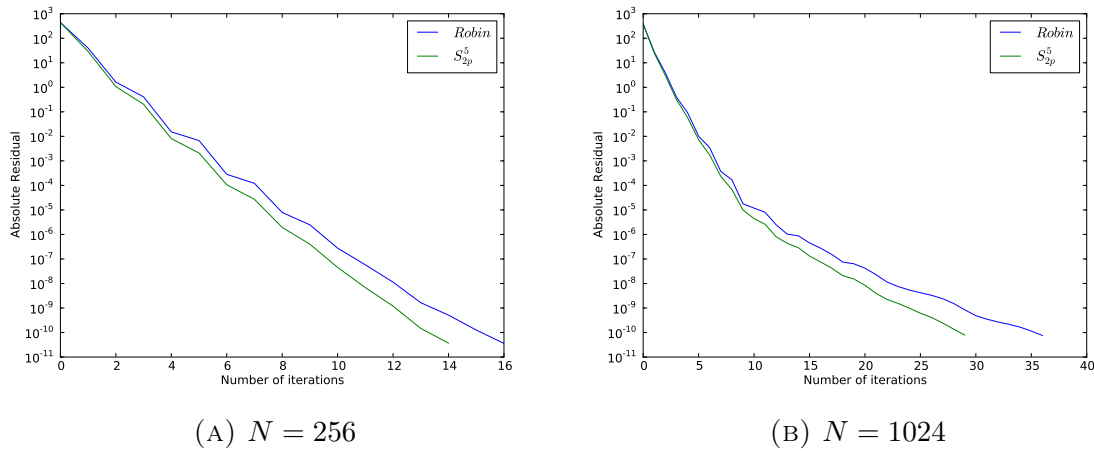


FIGURE 4.8: Historique de la convergence du premier étape du temps pour  $N = 256, 1024$ .

$N$		128	256	512	1024
Nombre d'itération	Robin	14	16	22	36
	$S_{2p}^5$	12	14	19	29
Temps de calcul	Robin cls.	250.2	143.8	92.5	101.4
	Robin new	59.1	38.1	36.2	52.3
	$S_{2p}^5$ cls.	*	187.5	162.6	127.5
	$S_{2p}^5$ new	*	42.7	36.2	45.0

\*: le mémoire est insuffisante.

TABLEAU 4.3: Nombre d'itérations de la première étape du temps et temps de calcul total en secondes,  $T = 0.5$ ,  $\Delta t = 0.01$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ .

D'après nos tests, le nouvel algorithme diminue le temps de calcul. Le nombre d'itérations pour la condition de transmission  $S_{2p}^5$  est moindres, mais pas le temps total. L'algorithme n'est pas très scalable pour  $N$  très grand.

### 4.5.3 Influences de paramètres

Nous étudions dans cette partie l'influence des paramètres dans les conditions de transmission:

- $m$  dans la condition de transmission  $S_{2p}^m$ ,
- $p$  dans la condition de transmission Robin.

Le pas de temps est fixé à  $\Delta t = 0.01$ . Nous faisons des tests pour observer les nombres d'itérations de la première étape du temps. Le maillage est toujours  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ . Deux vecteurs initiaux sont utilisés : le vecteur zéro et le vecteur aléatoire. Sur le problème d'interface, nous considérons trois implémentations différents : point fixe, Gmres et Bicgstab.

#### Influence de l'ordre $m$ dans la condition de transmission $S_{2p}^m$

Nous présentons les nombres d'itérations pour  $N = 2$  dans le tableau 4.4 et pour  $N = 32$  dans le tableau 4.5. Nous pouvons constater que

- le nombre d'itérations n'est pas sensible au paramètre  $m$  si les méthode Gmres ou Bicgstab sont utilisés sur le problème d'interface,
- si nous appliquons la méthode de point fixe sur l'interface, le nombre d'itérations diminue au début en augmentant l'ordre d'approximation de Padé, mais augmente ensuite,
- les méthode Krylov peuvent accélérer la convergence,
- l'influence des vecteurs initiaux est claire.

Nous avons expliqué dans la sous-section 4.2 que l'application de la condition de transmission  $S_{2p}^m$  est plus coûteuse en augmentant l'ordre  $m$ . Grâce aux constats ci-dessus, nous pouvons éviter d'utiliser un grand  $m$  en remplaçant la méthode de point fixe sur le problème d'interface par la méthode Gmres ou Bicgstab. D'après nos expériences numériques, l'algorithme avec Gmres prend moins de temps de calcul que la méthode Bicgstab. C'est la raison pour laquelle nous avons utilisé Gmres dans les deux sous-sections précédentes.



$m$	Point fixe		Gmres		Bicgstab	
	Zéro	Aléatoire	Zéro	Aléatoire	Zéro	Aléatoire
3	34	117	11	15	6	10
4	25	90	10	15	5	9
5	21	74	9	15	5	8
6	18	64	9	15	5	8
7	17	57	9	15	5	9
8	18	52	9	15	5	9
9	19	49	9	16	5	9
10	21	47	10	16	5	9
15	31	43	11	18	6	11
20	43	46	11	20	6	12

TABLEAU 4.4: Le nombre d'itérations pour des  $m$  différents  $N = 2$ .

$m$	Point fixe		Gmres		Bicgstab	
	Zéro	Aléatoire	Zéro	Aléatoire	Zéro	Aléatoire
3	34	124	11	30	6	17
4	26	96	10	28	6	16
5	21	79	10	27	5	15
6	18	68	9	26	5	15
7	17	61	9	25	5	14
8	18	56	9	25	5	14
9	19	52	10	25	5	14
10	21	50	10	25	5	14
15	32	46	11	26	6	15
20	43	50	12	28	6	16

TABLEAU 4.5: Le nombre d'itérations pour des  $m$  différents  $N = 32$ .

### Influence de $p$ dans la condition de transmission Robin

Les tableaux 4.6 et 4.7 présentent les nombres d'itérations avec des  $p$  différents pour  $N = 2$  et  $N = 32$  respectivement. Nous pouvons voir que

- l'algorithme n'est pas sensible à  $p$  si la méthode Gmres ou Bicgstab est utilisée sur le problème d'interface,
- si la méthode de point fixe est appliquée sur le problème d'interface, il existe un  $p$  optimal, souligné par la couleur rouge,
- les méthodes Krylov permettent d'accélérer la convergence,
- l'influence des vecteurs initiaux est claire.

Ainsi, nous pouvons utiliser les méthode Krylov pour diminuer le nombre d'itérations et éviter de faire beaucoup de tests pour trouver le paramètre  $p$  optimal.

$p$	Point fixe		Gmres		Bicgstab	
	Zéro	Aléatoire	Zéro	Aléatoire	Zéro	Aléatoire
5	56	544	11	35	6	10
10	34	292	11	32	6	9
15	32	226	11	31	6	11
20	35	198	11	31	6	11
25	40	189	11	32	6	12
30	46	185	11	32	6	13
35	52	190	12	33	6	13
40	58	196	12	33	6	14
45	65	205	12	34	6	13
15	32					
28		183				

TABLEAU 4.6: Le nombre d'itérations pour des  $p$  différents  $N = 2$ .

$p$	Point fixe		Gmres		Bicgstab	
	Zéro	Aléatoire	Zéro	Aléatoire	Zéro	Aléatoire
5	57	580	11	35	6	21
10	34	315	11	32	6	19
15	32	239	11	31	6	18
20	35	209	11	31	6	19
25	40	200	11	32	6	19
30	46	199	11	32	6	19
35	52	204	12	33	6	19
40	59	209	12	33	6	20
45	65	222	12	34	6	20
15	32					
26		198				

TABLEAU 4.7: Le nombre d'itérations pour des  $p$  différents  $N = 32$ .

En conclusion, nous pouvons voir clairement la différence entre les deux conditions de transmission si la méthode de point fixe est appliqué sur le problème d'interface et le vecteur initial est un vecteur aléatoire (voir les tableaux 4.5 et 4.7 ensemble). Le nombre d'itérations pour la condition de transmission  $S_{2p}^m$  est très inférieur à celui pour la condition de transmission de Robin. Mais la différence est moindre en utilisant les méthodes Krylov et le vecteur zéro comme le vecteur initial. Du point de vue du temps de calcul, le vecteur zéro et les méthodes Gmres sont un bon choix. D'après des tests dans la sous-section précédente, les temps de calcul pour les deux conditions de transmission sont similaires.

#### 4.5.4 Un autre cas test

Dans cette sous-section, nous faisons un autre cas test avec une donnée initiale différente pour confirmer nos résultats (voir figure 4.9)

$$u_0(x, y) = e^{-(x+8)^2 - y^2 + 16i(x+8)}.$$

La solution numérique au temps final  $T = 0.5$  sur le domaine complet est montrée dans la figure 4.10. L'étape du temps est fixé à  $\Delta t = 0.001$ . L'onde bouge du point  $(-8, 0)$  au point  $(8, 0)$  environ. Avec cette donnée initiale, il n'y a pas de sous-domaine dans lequel la solution est quasiment nulle (ce qui peut être le cas avec l'exemple précédent).

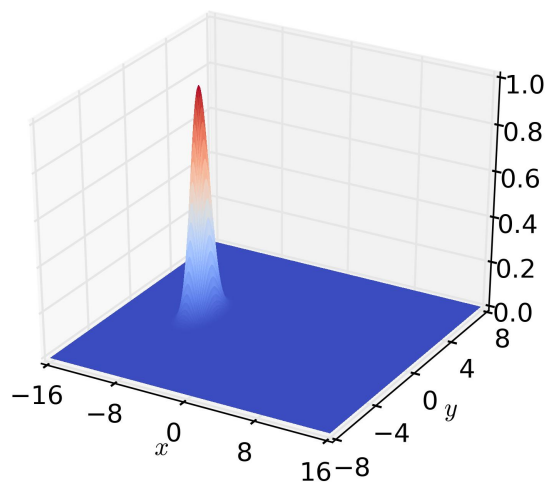


FIGURE 4.9: La donnée initiale  $|u_0|$ ,  $u_0 = e^{-(x+8)^2 - y^2 + 16i(x+8)}$ .

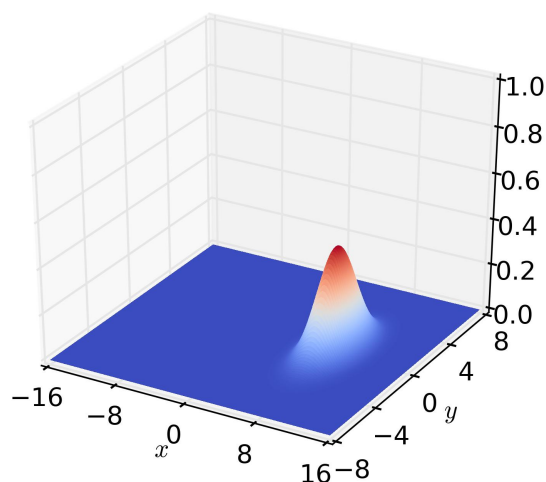


FIGURE 4.10: Le module de la solution numérique au temps final  $T = 0.5$ .

Nous prenons  $p = 10$  dans la condition de transmission de Robin et  $m = 5$  dans la condition de transmission  $S_{2p}^m$ . La méthode Gmres est appliquée au problème d'interface. Les tableaux

4.8 et 4.9 présentent les temps de calcul pour les deux maillages différents  $\Delta x = 1/128$ ,  $\Delta y = 1/8$  et  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ . Comme précédemment, nous pouvons voir que le nouvel algorithme prend moins de temps de calcul que l'algorithme classique. Les algorithmes avec la condition de transmission  $S_{2p}^5$  sont plus coûteux que ceux avec la condition de transmission de Robin.

$N$	2	4	8	16	32
Robin cls.	394.5	256.0	149.9	90.1	43.6
Robin new	116.7	57.8	30.6	18.7	9.6
$S_{2p}^5$ cls.	615.9	403.0	235.8	133.8	68.7
$S_{2p}^5$ new	178.4	88.3	46.6	26.0	13.9

TABLEAU 4.8: Temps de calcul en secondes,  $T = 0.5$ ,  $\Delta t = 0.001$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

$N$	128	256	512	1024
Robin cls.	1866.9	1162.4	716.8	404.3
Robin new	427.2	268.6	277.3	259.3
$S_{2p}^5$ cls.	*	1628.9	937.9	665.3
$S_{2p}^5$ new	*	384.0	304.1	291.5

\*: mémoire insuffisante.

TABLEAU 4.9: Temps de calcul en secondes,  $T = 0.5$ ,  $\Delta t = 0.001$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$

## 4.6 Conclusion

Dans ce chapitre, on a appliqué la méthode SWR et la méthode de décomposition en espace à l'équation de Schrödinger linéaire en deux dimensions. Numériquement, la méthode de décomposition en espace prend moins de temps de calcul d'après nos tests.

Sur l'interface, les conditions de transmission de Robin et  $S_{2p}^m$  sont utilisées. Les deux conditions de transmission sont comparées numériquement. L'algorithme avec  $S_{2p}^m$  converge plus vite par rapport au nombre d'itérations dans nos tests.

Pour  $V = 0$ , nous proposons un nouvel algorithme. Il est basé sur la construction explicite du problème d'interface. Mathématiquement, le nouvel algorithme et l'algorithme classique sont équivalents, mais numériquement, le nouvel algorithme prend moins de temps de calcul d'après nos expériences.

# Chapitre 5

## Méthode de décomposition de domaines pour l'équation de Schrödinger avec potentiel général en deux dimensions

### Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>101</b>
<b>5.2</b>	<b>Discrétisation</b>	<b>103</b>
<b>5.3</b>	<b>Algorithme avec préconditionneur</b>	<b>104</b>
<b>5.4</b>	<b>Résultats numériques</b>	<b>106</b>
5.4.1	Comparaison des algorithmes sans et avec préconditionneur	106
5.4.2	Comparaison des conditions de transmission	110
5.4.3	Influence des paramètres	113
5.4.4	Accélération de Krylov pour l'équation de Schrödinger linéaire	117
<b>5.5</b>	<b>Conclusion</b>	<b>118</b>

---

### 5.1 Introduction

Dans ce chapitre, nous souhaitons étudier la méthode de décomposition en espace pour l'équation de Schrödinger en deux dimensions avec un potentiel général

$$\begin{cases} \mathcal{L}u := (i\partial_t + \Delta + \mathcal{V})u = 0, & (t, x, y) \in ]0, T[ \times \Omega_0, \\ u(0, x, y) = u_0(x, y), & (x, y) \in \Omega_0, \end{cases} \quad (5.1)$$

où le potentiel réel  $\mathcal{V}$  est ici soit

- $\mathcal{V} = V(t, x, y) \in \mathcal{C}^\infty(\mathbb{R} \times \mathbb{R}^2)$ ,
- $\mathcal{V} = f(u)$ , ex:  $f(u) = |u|^2$ ,

- $\mathcal{V} = V(t, x, y) + f(u)$ .

La configuration géométrique est similaire à celle du chapitre 4. Nous étendons les résultats du chapitre 3 au cas de l'équation en deux dimensions. Comme nous l'avons remarqué dans le chapitre précédent, la méthode de décomposition en espace a plus de potentiel que la méthode SWR. Nous ne considérons donc plus la méthode SWR dans ce chapitre.

L'équation de Schrödinger avec un potentiel  $\mathcal{V} = V(t, x, y) + f(u)$  est discrétisée d'abord par le schéma de Durán-Sanz-Serna sur le domaine complet  $\Omega_0$ . Nous ne répétons pas ici les symboles et notations qui sont définis dans le chapitre précédent (voir la sous-section 4.2). Pour  $1 \leq n \leq N_T$ , on a

$$i \frac{u_{0,n} - u_{0,n-1}}{\Delta t} + \Delta \frac{u_{0,n} + u_{0,n-1}}{2} + \frac{V_n + V_{n-1}}{2} \frac{u_{0,n} + u_{0,n-1}}{2} + f\left(\frac{u_{0,n} + u_{0,n-1}}{2}\right) \frac{u_{0,n} + u_{0,n-1}}{2} = 0.$$

En introduisant les nouvelles variables  $v_{0,n} = \frac{u_{0,n} + u_{0,n-1}}{2}$  avec  $v_{0,0} = u_{0,0}$  et  $W_n = \frac{V_n + V_{n-1}}{2}$ , le schéma s'écrit alors

$$\mathcal{L}_x v_{0,n} = 2i \frac{v_{0,n-1}}{\Delta t}, \quad (5.2)$$

où

$$\mathcal{L}_x v_{0,n} := 2i \frac{v_{0,n}}{\Delta t} + \Delta v_{0,n} + W_n v_{0,n} + f(v_{0,n}) v_{0,n}.$$

La méthode de Schwarz optimisée est appliquée à (5.2) pour calculer  $v_{0,n}$  en supposant que  $u_{0,n-1}$  est connu:

$$\begin{cases} \mathcal{L}_x v_{j,n}^k = \frac{2i}{\Delta t} u_{j,n-1}, \\ \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S}_j v_{j,n}^k = l_{j,n}^k, x = a_j, \\ \partial_{\mathbf{n}_j} v_{j,n}^k + \bar{S}_j v_{j,n}^k = r_{j,n}^k, x = b_j, \end{cases} \quad \begin{cases} l_{j,n}^{k+1} = -r_{j-1,n}^k + 2\bar{S}_j v_{j-1,n}^k(b_{j-1}, y), j \neq 1, \\ r_{j,n}^{k+1} = -l_{j+1,n}^k + 2\bar{S}_j v_{j+1,n}^k(a_{j+1}, y), j \neq N, \\ l_{1,n}^{k+1}(y) = 0, r_{N,n}^{k+1}(y) = 0, \end{cases} \quad (5.3)$$

où  $j = 1, 2, \dots, N$ .  $u_{j,n-1}$  désigne la restriction de  $u_{0,n-1}$  dans la sous-domaine  $\Omega_j$  et les flux  $l_{j,n}^k(y)$  et  $r_{j,n}^k(y)$  sont définis par (4.7) dans la sous-section 4.2.  $\bar{S}$  est l'opérateur de transmission. Comme précédemment, nous utilisons dans ce chapitre les deux conditions de transmission

- Robin:  $\bar{S}_j v_{j,n}^k = -ip \cdot v_{j,n}^k$ ,
- $S_{2p}^m$ :  $\bar{S}_j v_{j,n}^k = -i \sum_{s=0}^m a_s^m v_{j,n}^k + \sum_{s=1}^m a_s^m d_s^m \varphi_{j,s}$ ,  $x = a_j, b_j$ , les fonctions auxiliaires  $\varphi_{j,s}$ ,  $s = 1, 2, \dots, m$  sont définies respectivement par

$$\left( \frac{2i}{\Delta t} + \Delta_{\Gamma_j} + W_n + f(v_{j,n}^k) + d_s^m \right) \varphi_{j,s} = v_{j,n}^k.$$

Le but de ce chapitre est d'appliquer la méthode de décomposition en espace à l'équation de Schrödinger en deux dimension avec un potentiel général. Nous étendons l'algorithme avec préconditionneur proposé pour l'équation de Schrödinger en une dimension (voir chapitre 3) au cas de la dimension 2. Nous nous intéressons aussi à l'influence des conditions de transmission sur la convergence.

## 5.2 Discrétisation

Nous présentons brièvement la méthode des éléments finis  $Q_1$  pour l'approximation spatiale de (5.3). La formulation variationnelle pour une fonction test  $\phi$  est

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{\Omega_j} v_{j,n}^k \phi d\Omega - \int_{\Omega_j} \nabla v_{j,n}^k \nabla \phi d\Omega + \int_{\Omega_j} W_n v_{j,n}^k \phi d\Omega \\ & + \int_{\Omega_j} f(v_{j,n}^k) v_{j,n}^k \phi d\Omega + \int_{\Gamma_j} \partial_{\mathbf{n}_j} v_{j,n}^k \phi d\Gamma = \frac{2i}{\Delta t} \int_{\Omega_j} u_{j,n-1} \phi d\Omega. \end{aligned} \quad (5.4)$$

Ce problème étant non-linéaire, l'approximation est faite par une procédure de point fixe. Nous posons  $\zeta_j^0 = v_{j,n-1}$ , et nous calculons  $v_{j,n}$  comme la limite du schéma itératif indiqué par  $r$

$$\begin{aligned} & \frac{2i}{\Delta t} \int_{\Omega_j} \zeta_j^r \phi d\Omega - \int_{\Omega_j} \nabla \zeta_j^r \nabla \phi d\Omega + \int_{\Omega_j} W_n \zeta_j^r \phi d\Omega \\ & + \int_{\Gamma_j} \partial_{\mathbf{n}_j} \zeta_j^r \phi d\Gamma = \frac{2i}{\Delta t} \int_{\Omega_j} u_{j,n-1} \phi d\Omega - \int_{\Omega_j} f(\zeta_j^{r-1}) \zeta_j^{r-1} \phi d\Omega, \end{aligned} \quad (5.5)$$

avec les conditions aux limites suivantes

$$\begin{cases} \partial_{\mathbf{n}_j} \zeta_j^r + \bar{S}_j \zeta_j^r = l_{j,n}^k, & x = a_j, \\ \partial_{\mathbf{n}_j} \zeta_j^r + \bar{S}_j \zeta_j^r = r_{j,n}^k, & x = b_j, \end{cases}$$

où

- Robin:  $\bar{S}_j \zeta_j^r = -ip \cdot \zeta_j^r$ ,
- $S_{2p}^m$ :  $\bar{S}_j \zeta_j^r = -i \sum_{s=0}^m a_s^m \zeta_j^r + \sum_{s=1}^m a_s^m d_s^m \varphi_{j,s}^r$ ,  $x = a_j, b_j$ , les fonctions auxiliaires  $\varphi_{j,s}^r$ ,  $s = 1, 2, \dots, m$  satisfont

$$\varphi_{j,s}^0 = 0, \quad \left( \frac{2i}{\Delta t} + W_n + \Delta_{\Gamma_j} + d_s^m \right) \varphi_{j,s}^r = \zeta_j^r - f(\zeta_j^{r-1}) \varphi_{j,s}^{r-1}.$$

$\varphi_{j,s}$  est la limite de  $\varphi_{j,s}^r$  par rapport au paramètre  $r$ .

Nous désignons

- $\mathbb{M}_{f(\zeta_j^{r-1})}$ : la matrice de masse généralisée associée à  $\int_{\Omega_j} f(\zeta_j^{r-1}) \zeta_j^{r-1} \phi d\Omega$ ,
- $\mathbb{M}_{f(\zeta_j^{r-1})}^{\Gamma_j}$ : la matrice de masse généralisée associée à  $\int_{\Gamma_j} f(\zeta_j^{r-1}) \zeta_j^{r-1} \phi d\Gamma$  sur le bord.

Matriciellement, la formulation variationnelle (5.5) conduit aux  $N$  systèmes linéaires suivants

- Robin

$$\left( \mathbb{A}_{j,n} + ip \cdot \mathbb{M}^{\Gamma_j} \right) \zeta_j^r = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1} - \mathbb{M}_{f(\zeta_j^{r-1})} \zeta_j^{r-1} - \mathbb{M}^{\Gamma_j} \cdot Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix}, \quad (5.6)$$

où  $\mathbb{A}_{j,n} = \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j + \mathbb{M}_{j,W_n}$ .

- $S_{2p}^m$

$$\begin{aligned}
 & \begin{pmatrix} \mathbb{A}_{j,n} + i(\sum_{s=1}^m a_s^m) \cdot \mathbb{M}^{\Gamma_j} & \mathbb{B}_{j,1} & \mathbb{B}_{j,2} & \cdots & \mathbb{B}_{j,m} \\ \mathbb{C}_j & \mathbb{D}_{j,1} & & & \\ \mathbb{C}_j & & \mathbb{D}_{j,2} & & \\ \vdots & & & \ddots & \\ \mathbb{C}_j & & & & \mathbb{D}_{j,m} \end{pmatrix} \begin{pmatrix} \zeta_j^r \\ \varphi_{j,1}^r \\ \varphi_{j,2}^r \\ \vdots \\ \varphi_{j,m}^r \end{pmatrix} \\
 &= \frac{2i}{\Delta t} \begin{pmatrix} \mathbb{M}_j \mathbf{u}_{j,n-1} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} \mathbb{M}_{f(\zeta_j^{r-1})} \zeta_j^{r-1} \\ \mathbb{M}_{f(\zeta_j^{r-1})} \varphi_{j,1}^{r-1} \\ \vdots \\ \mathbb{M}_{f(\zeta_j^{r-1})} \varphi_{j,m}^{r-1} \end{pmatrix} - \begin{pmatrix} \mathbb{M}^{\Gamma_j} \cdot Q_j^T \begin{pmatrix} \mathbf{l}_{j,n}^k \\ \mathbf{r}_{j,n}^k \end{pmatrix} \\ 0 \\ \vdots \\ 0 \end{pmatrix}.
 \end{aligned} \tag{5.7}$$

Les conditions de transmission discrètes sont

$$\begin{cases} \mathbf{l}_{j+1,n}^{k+1} = -\mathbf{r}_{j,n}^k + 2Q_{j,r} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 2, 3, \dots, N, \\ \mathbf{r}_{j-1,n}^{k+1} = -\mathbf{l}_{j,n}^k + 2Q_{j,l} \cdot \tilde{S}_j \mathbf{v}_{j,n}^k, & j = 1, 2, \dots, N-1, \\ \mathbf{l}_{1,n}^{k+1} = 0, \mathbf{r}_{N,n}^{k+1} = 0, \end{cases} \tag{5.8}$$

où

- Robin:  $\tilde{S}_j \mathbf{v}_{j,n}^k = -ip \cdot \mathbf{v}_{j,n}^k$ .
- $S_{2p}^m$ :  $\tilde{S}_j \mathbf{v}_{j,n}^k = -i(\sum_{s=0}^m a_s^m) \mathbf{v}_{j,n}^k + i \sum_{s=1}^m a_s^m d_s^m \varphi_{j,s}$ ,  $\varphi_{j,s}$  est la limite de  $\varphi_{j,s}^r$  en  $r$ .

### 5.3 Algorithme avec préconditionneur

De la même manière que le chapitre précédent, nous pouvons écrire les  $N$  systèmes linéaires (5.8) (5.6) (5.7) à l'étape de temps  $n$  pour le potentiel  $\mathcal{V} = V(t, x, y) + f(u)$  comme

$$g_{t_n}^{k+1} = \mathcal{R}_{nl} g_{t_n}^k = I - (I - \mathcal{R}_{nl}) g_{t_n}^k, \tag{5.9}$$

où le vecteur aux interfaces  $g_{t_n}^k$  à l'étape de temps  $n$  est défini par

$$g_{t_n}^k = (\mathbf{r}_{1,n}^{k,T}, \mathbf{l}_{2,n}^{k,T}, \mathbf{r}_{2,n}^{k,T}, \dots, \mathbf{l}_{N,n}^{k,T})^T \in \mathbb{C}^{(2N-2) \times N_y}.$$

Si le potentiel est linéaire  $\mathcal{V} = V(t, x, y)$ , alors l'opérateur  $\mathcal{R}_{nl}$  est linéaire:

$$\mathcal{R}_{nl} g_{t_n}^k = \mathcal{L} g_{t_n}^k + d_{t_n}.$$

Ainsi (5.9) est la méthode du point fixe pour  $(I - \mathcal{L}) g_{t_n} = d_{t_n}$ .

Nous avons étudié le cas  $\mathcal{V} = 0$  dans le chapitre 4. Il est possible de construire la matrice  $\mathcal{L}$  en parallèle et cette construction est super-scalable en théorie. Nous étendons donc l'algorithme



avec préconditionneur du chapitre 3 à deux dimensions

$$g_{t_n}^{k+1} = I - P^{-1}(I - \mathcal{R}_{nl})g_{t_n}^k, \quad (5.10)$$

$$P = I - \mathcal{L}_0, \quad (5.11)$$

où la matrice  $\mathcal{L}_0$  désigne la matrice  $\mathcal{L}$  dans le cas sans potentiel. Pour un vecteur quelconque  $y$ , le vecteur  $x := P^{-1}y$  est calculé par la résolution du système linéaire

$$Px = y.$$

Celui-ci est résolu par les méthodes de Krylov en utilisant la librairie PETSc. La taille de la matrice  $P$  est  $(2N - 2)N_y \times (2N - 2)N_y$ .  $P$  est stockée sous la forme de PETSc. Nous avons expliqué précédemment que la mémoire consommée par chaque processus MPI pour le stockage de  $P$  est  $4(N_y)^2$  pour  $j = 2, 3, \dots, N - 1$  et  $2(N_y)^2$  pour  $j = 1, N$ . La taille des matrices pour chaque sous-domaine est  $N_x \times N_y$ . Ainsi, la mémoire consommée par la méthode directe LU est environ  $(N_x \times N_y)^2$ . Ceci est usuellement largement plus important que  $4(N_y)^2$ . Ainsi, le stockage en mémoire de  $P$  est négligeable.

Cependant, la taille totale augmente évidemment avec le nombre de sous-domaines  $N$ . De plus le nombre d'opérations pour la multiplication de  $\mathcal{L}_0$  et d'un vecteur (ce qui constitue l'opération de base des méthodes de Krylov) n'est plus négligeable par rapport à la résolution des équations sur chaque sous-domaine. Ainsi, l'application de préconditionneur prend beaucoup plus de temps en augmentant  $N$  que dans le cas monodimensionnel. Nous verrons en effet dans la sous-section suivante (résultats numériques) que l'algorithme avec préconditionneur n'est pas toujours meilleur que l'algorithme sans préconditionneur.

En conclusion, les algorithmes s'écrivent sous la forme

---

**Algorithme 10 :** Algorithme sans préconditionneur

---

- 1: Initialiser l'itération par  $g_{t_1}^0$ ,
  - 2: **for**  $n = 1, 2, \dots, N_T$  **do**
    - 2.1: **while**  $\|g_{t_n}^{k+1} - g_{t_n}^k\| > \varepsilon, \varepsilon \ll 1$  **do**
      - └  $g_{t_n}^{k+1} = \mathcal{R}_{nl}g_{t_n}^k$ ,
    - 2.2: Résoudre l'équation de Schrödinger sur chaque sous-domaine à l'étape du temps  $n$  ( $t = n\Delta t$ ) avec  $g_{t_n}^{k+1}$  pour obtenir  $v_{j,n}$ , et puis  $u_{j,n} = 2v_{j,n} - u_{j,n-1}$ ,
    - 2.3:  $g_{t_{n+1}}^0 = g_{t_n}^{k+1}$ .
- 

Nous remarquons que si le potentiel est linéaire  $\mathcal{V} = V(t, x, y)$ , nous pouvons, à l'étape du 2.1 des algorithmes ci-dessus, également utiliser les méthodes de Krylov pour calculer la limite de  $g_{t_n}^k$  à la place de la méthode de point fixe, comme nous le faisons pour l'équation en une dimension.

**Algorithme 11** : Algorithme avec préconditionneur

---

```

1: Initialiser l'itération par  $g_{t_1}^0$ ,
2: for  $n = 1, 2, \dots, N_T$  do
  2.1: while  $\|g_{t_n}^{k+1} - g_{t_n}^k\| > \varepsilon, \varepsilon \ll 1$  do
    2.1.1:  $\mathcal{R}_{nl}g_{t_n}^k$ ,
    2.1.2:  $g_{t_n}^k - \mathcal{R}_{nl}g_{t_n}^k$ ,
    2.1.3:  $P^{-1}(g_{t_n}^k - \mathcal{R}_{nl}g_{t_n}^k)$ ,
    2.1.4:  $g_{t_n}^{k+1} = g_{t_n}^k - P^{-1}(g_{t_n}^k - \mathcal{R}_{nl}g_{t_n}^k)$ ,
  2.2: Résoudre l'équation de Schrödinger sur chaque sous-domaine à l'étape du temps  $n$ 
  ( $t = n\Delta t$ ) avec  $g_{t_n}^{k+1}$  pour obtenir  $v_{j,n}$ , et puis  $u_{j,n} = 2v_{j,n} - u_{j,n-1}$ .
  2.3:  $g_{t_{n+1}}^0 = g_{t_n}^{k+1}$ .

```

---

## 5.4 Résultats numériques

Nous présentons les résultats numériques dans cette partie. Tout d'abord, les algorithmes sans et avec préconditionneur sont comparés pour deux potentiels :

$$\mathcal{V} = x^2 + y^2, \quad \mathcal{V} = |u|^2.$$

Ensuite, nous comparons les deux conditions de transmission dans le cadre des algorithmes avec et sans préconditionneur. Nous faisons des tests pour étudier l'influence des paramètres dans les conditions de transmission dans la sous-section 5.4.3. À la fin de ce chapitre, nous considérons la accélération de Krylov pour l'équation de Schrödinger linéaire.

Le domaine complet  $\Omega_0 = ]-16, 16[ \times ]-8, 8 - \Delta y[$  est décomposé en  $N$  sous-domaines égaux (voir figure 4.1 pour  $N = 3$ ). Le temps final est fixé à  $T = 0.5$ . Comme dans le chapitre précédent, nous utilisons deux maillages différents:

$$\begin{aligned} \Delta x &= 1/128, \quad \Delta y = 1/8, \\ \Delta x &= 1/2048, \quad \Delta y = 1/64. \end{aligned}$$

où la taille d'un élément est  $\Delta x \times \Delta y$ . La donnée initiale est

$$u_0(x, y) = e^{-x^2 - y^2 - 0.5ix}.$$

### 5.4.1 Comparaison des algorithmes sans et avec préconditionneur

Nous nous intéressons à observer la robustesse, les temps de calcul et la scalabilité forte des algorithmes sans et avec préconditionneur pour les deux potentiels en mettant en jeu la condition de transmission  $S_{2p}^m$ . Le vecteur zéro est utilisé comme vecteur initial (voir la sous-section 2.6.2). Nous répétons tout d'abord quelques définitions.  $N_{\text{nopc}}$  (resp.  $N_{\text{pc}}$ ) désigne le nombre d'itérations requises avec l'algorithme sans préconditionneur (resp. avec préconditionneur).  $T_{\text{nopc}}$  (resp.  $T_{\text{pc}}$ ) est le temps de calcul obtenu avec l'algorithme sans préconditionneur (resp. avec préconditionneur).  $T^{\text{ref}}$  est le temps de calcul sur un processeur de la résolution numérique de l'équation de Schrödinger sur le domaine complet  $]0, T[ \times \Omega_0$

lorsque cela est possible (cela dépend de la quantité de mémoire utilisée pour le calcul sur un nœud).

#### 5.4.1.1 Cas du potentiel $\mathcal{V} = x^2 + y^2$

Nous faisons des tests des algorithmes sans et avec préconditionneur pour  $N = 2, 4, 8, 16, 32$  sous-domaines. Le maillage est  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ . Les historiques de la convergence de la première étape du temps sont présentés dans la figure 5.1 pour  $N = 2$  et  $N = 32$ . Nous montrons le nombre d'itérations requises et le temps de calcul pour réaliser une simulation complète à convergence dans le tableau 5.1. Nous verrons l'influence d'ordre d'approximation de Padé  $m$  dans la sous-section 5.4.3. Les paramètres  $m$  que nous utilisons ici sont des optimaux numériques. Nous pouvons ainsi faire des constats suivants :

- tous les algorithmes sont robustes,
- le nombre d'itérations requises est indépendant du nombre de sous-domaines,
- les algorithmes sans et avec préconditionneur sont scalables,
- le préconditionneur diminue le nombre d'itérations et le temps de calcul total,
- les temps  $T_{\text{nopc}}$ ,  $T_{\text{pc}}$  sont inférieurs à  $T^{\text{ref}}$  pour un nombre de sous-domaine assez grand (ici  $N = 32$ ).

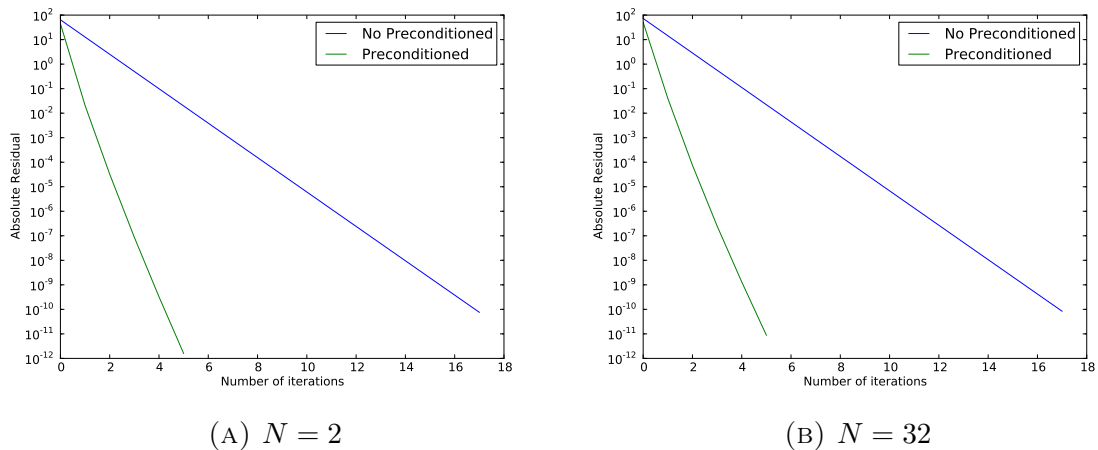


FIGURE 5.1: Historique de la convergence du premier étape du temps des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

$N$	2	4	8	16	32
$N_{\text{nopc}}, m = 7$	17	17	17	17	17
$N_{\text{pc}}, m = 5$	5	5	5	5	5
$T^{\text{ref}}$	16.1				
$T_{\text{nopc}}$	142.7	75.3	40.1	23.9	12.1
$T_{\text{pc}}$	91.3	43.3	22.8	13.1	7.3

TABLEAU 5.1: Nombre d'itérations du premier étape du temps et le temps de calcul total en secondes des algorithmes sans et avec préconditionneur,  $T = 0.5$ ,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

Nous utilisons ensuite le maillage plus fin ( $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ ) qui contient  $2^{25}$  ( $\approx 3.36 \times 10^7$ ) nœuds sur le domaine complet  $\Omega_0$ .

Nous ne pouvons pas réaliser la simulation sur le domaine complet car la mémoire nécessaire n'est pas disponible sur un de nos nœud de calcul. Hormis les temps de calcul, nous voyons ici un des autres intérêts des méthodes de décomposition de domaines.

Nous reproduisons les mêmes tests pour  $N = 256$ ,  $N = 512$  et  $N = 1024$ . La figure 5.2 présente les historiques de la convergence pour  $N = 256$  et 512. Les temps de calcul totaux sont montrés dans le tableau 5.2. Les paramètres  $m$  que nous utilisons sont aussi présentés dans le tableau 5.2. Nous pouvons voir que l'algorithme avec préconditionneur est plus robuste avec un nombre d'itérations requises moindre. Les deux algorithmes ne sont cependant pas scalables car :

- pour l'algorithme sans préconditionneur, le nombre d'itérations augmente avec le nombre de sous-domaines,
- pour l'algorithme avec préconditionneur, la taille du préconditionneur est  $(2N - 2) \times N_T \times N_y$ . Celle-ci augmente avec le nombre de sous-domaines  $N$ . Ainsi, l'application du préconditionnement prend plus de temps de calcul.

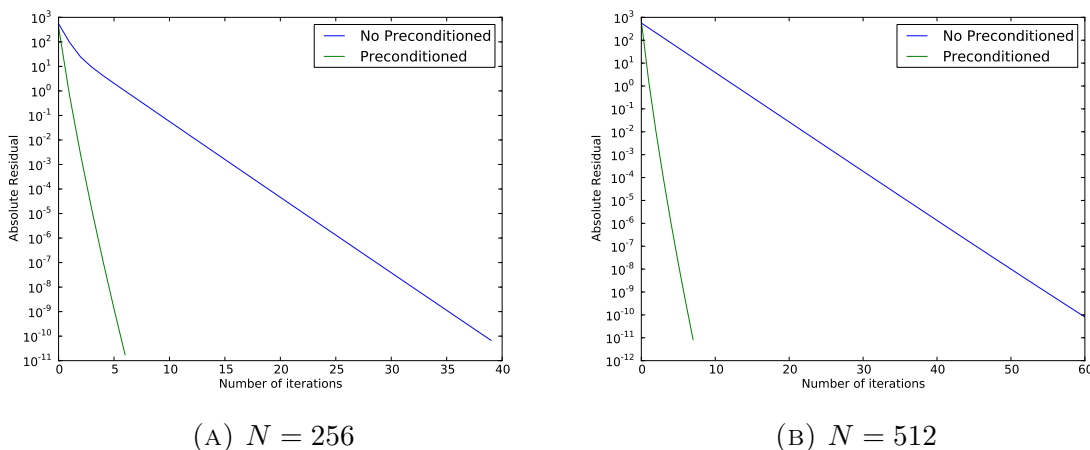


FIGURE 5.2: Historique de la convergence du premier étape du temps pour les algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/128$ .

$N$		256	512	1024
Sans préconditionneur	$m$	8	12	*
	$T_{\text{nopc}}$	417.9	344.9	
Avec préconditionneur	$m$	5	6	*
	$T_{\text{pc}}$	259.9	268.7	

\*: l'algorithme n'est pas robuste.

TABLEAU 5.2: Temps de calcul total en secondes des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ .

Nous remarquons que les algorithmes ne sont pas robustes pour  $N = 1024$ . Nous n'obtenons plus la convergence. Cela rappelle les résultats que nous avons obtenus en 1D pour le potentiel  $V(x) = x^2$ .

#### 5.4.1.2 Cas du potentiel $\mathcal{V} = |u|^2$

Nous reproduisons les mêmes tests que dans la sous-section précédente pour  $\mathcal{V} = |u|^2$ . Nous considérons d'abord le maillage  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ . Les historiques de la convergence sont montrés dans la figure 5.3. Le tableau 5.3 présente les nombres d'itérations et les temps de calculs. Les conclusions qui peuvent être tirées sont similaires au cas du potentiel  $\mathcal{V} = x^2 + y^2$ .

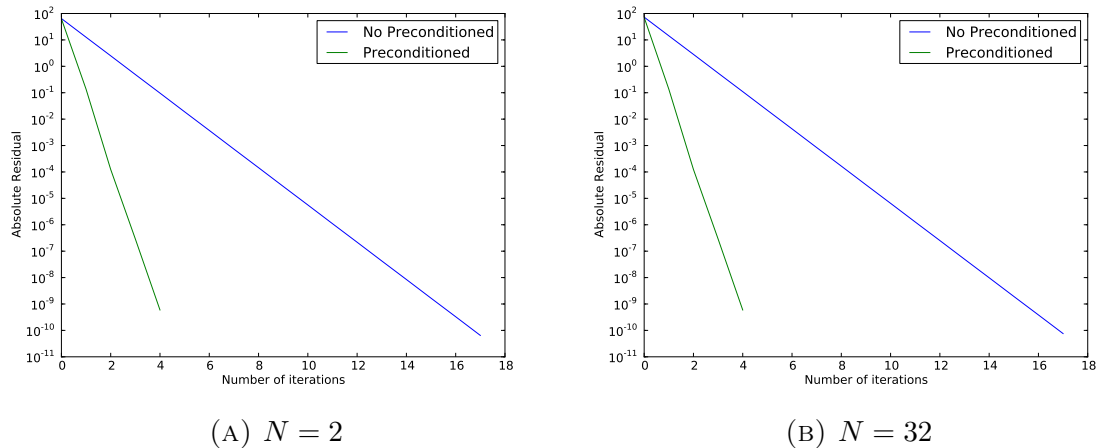


FIGURE 5.3: Historique de la convergence du premier étape du temps des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

$N$	2	4	8	16	32
$m$	7				
$N_{\text{nopc}}$	18	18	18	18	18
$N_{\text{pc}}$	6	6	6	6	6
$T^{\text{ref}}$	7.4				
$T_{\text{nopc}}$	25.70	13.23	6.86	3.71	1.99
$T_{\text{pc}}$	32.87	11.82	4.85	2.35	1.21

TABLEAU 5.3: Nombre d'itérations et temps de calcul en secondes des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

Nous montrons les résultats pour le maillage fin  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$  dans la figure 5.4 et le tableau 5.4. Les algorithmes sont plus robustes que dans le cas du potentiel linéaire puisque nous arrivons maintenant à réaliser des simulations pour  $N = 1024$  sous-domaines. Cependant, les temps de calculs augmentent car le nombre de communications entre les processus MPI est plus important.

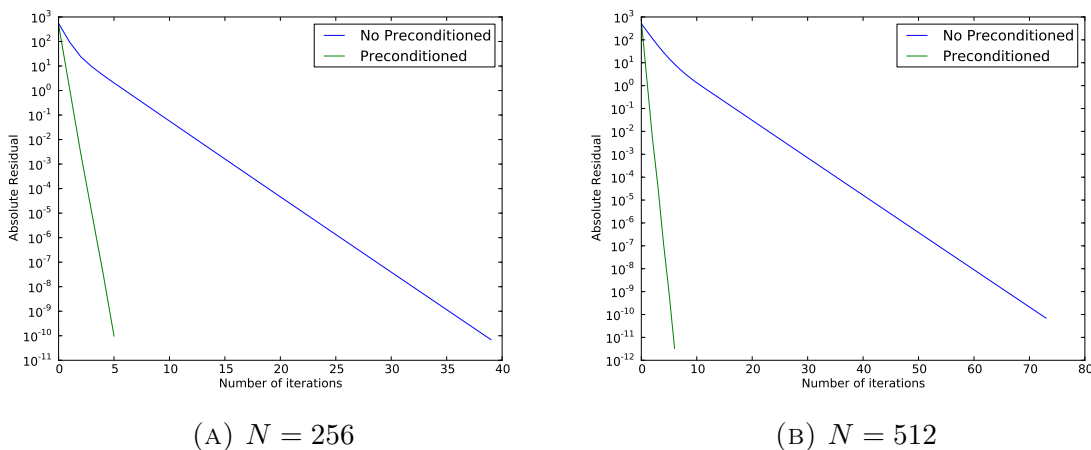


FIGURE 5.4: Historique de la convergence du premier étage du temps pour les algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ .

$N$		256	512	1024
Sans préconditionneur	$m$	8	12	11
	$T_{\text{nopc}}$	3582.4	2681.5	2516.6
Avec préconditionneur	$m$	6	4	5
	$T_{\text{pc}}$	596.9	376.1	441.9

TABLEAU 5.4: Temps de calcul total en secondes des algorithmes sans et avec préconditionneur,  $\Delta t = 0.5$ ,  $\Delta x = 1/2048$ ,  $\Delta y = 1/64$ .

## 5.4.2 Comparaison des conditions de transmission

Nous comparons ici la condition de transmission de Robin et  $S_{2p}^m$  dans le cadre des algorithmes sans et avec préconditionneur. Le vecteur initial est un vecteur aléatoire. Le pas

de temps est  $\Delta t = 0.01$ . Nous faisons des tests pour observer la convergence de la première étape du temps. Le maillage utilisé est  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

#### 5.4.2.1 Cas du potentiel $\mathcal{V} = x^2 + y^2$

Nous présentons d'abord les historiques de la convergence de la première étape du temps avec les algorithmes sans et avec préconditionneur pour  $N = 2$  et  $N = 32$  sous-domaines dans la figure 5.5. Nous pouvons constater que dans le cadre de l'algorithme sans préconditionneur, l'algorithme converge plus vite avec la conditions de transmission  $S_{2p}^m$  que avec la condition de transmission de Robin. Cependant, nous ne voyons presque pas de différences entre  $S_{2p}^m$  et Robin en utilisant le préconditionneur. Le tableau 5.5 montre les paramètres que nous utilisons ici et les nombres d'itérations ( $N_{iter}$ ) pour  $N = 2, 4, 8, 16, 32$ . L'algorithme avec préconditionneur ne semble pas très robuste d'après les résultats présentés ici. Ce constat est opposé à celui nous faisons dans la sous-section précédente. Une des raisons est que la donnée initiale ici est un vecteur aléatoire au lieu du vecteur zéro.

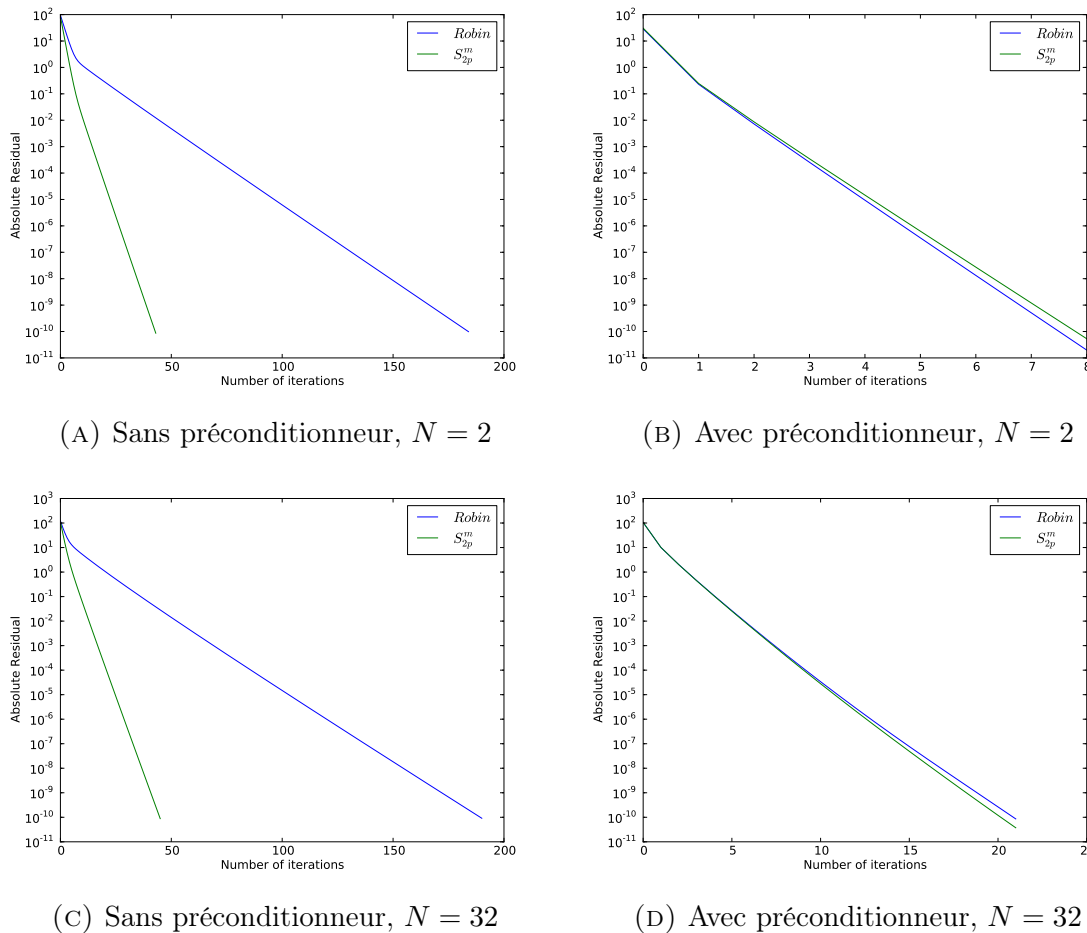


FIGURE 5.5: Historique de la convergence de la première étape du temps,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

$N$			2	4	8	16	32
Sans Pd.	Robin	$p$	30	30	30	30	30
		$N_{iter}$	179	183	183	188	190
	$S_{2p}^m$	$m$	13	13	13	13	13
		$N_{iter}$	43	43	44	45	45
Avec Pd.	Robin	$p$	5	14	14	14	14
		$N_{iter}$	8	13	17	20	21
	$S_{2p}^m$	$m$	3	5	7	7	7
		$S_{2p}^m$	8	13	17	19	20

TABLEAU 5.5: Nombre d'itérations de la première étape du temps des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

#### 5.4.2.2 Cas du potentiel $\mathcal{V} = |u|^2$

Nous reproduisons les mêmes tests que dans la sous-section précédente pour  $\mathcal{V} = |u|^2$ . La figure 5.6 présente les historiques de la convergence. Nous montrons dans le tableau 5.6 les paramètres que nous utilisons et les nombres d'itérations. Les conclusions sont similaires à celle que nous avons faites dans le cas du potentiel linéaire, mis à part que l'algorithme avec préconditionneur est maintenant scalable. Dans la formule (5.6), si le potentiel est  $\mathcal{V} = |u|^2$ , alors

$$\mathbb{A}_{j,n} = \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j.$$

C'est ainsi la même matrice que  $\mathbb{A}_{j,n}$  pour  $\mathcal{V} = 0$ . À l'inverse, si le potentiel est  $\mathcal{V} = x^2 + y^2$ , alors

$$\mathbb{A}_{j,n} = \frac{2i}{\Delta t} \mathbb{M}_j - \mathbb{S}_j + \mathbb{M}_{j,W_n}.$$

Grosso modo, le préconditionneur  $P^{-1}$  est plus proche de l'inverse du problème d'interface pour le potentiel non-linéaire que pour le potentiel linéaire. Nous pensons que c'est une des raisons qui fait que l'algorithme avec préconditionneur est ici plus scalable.



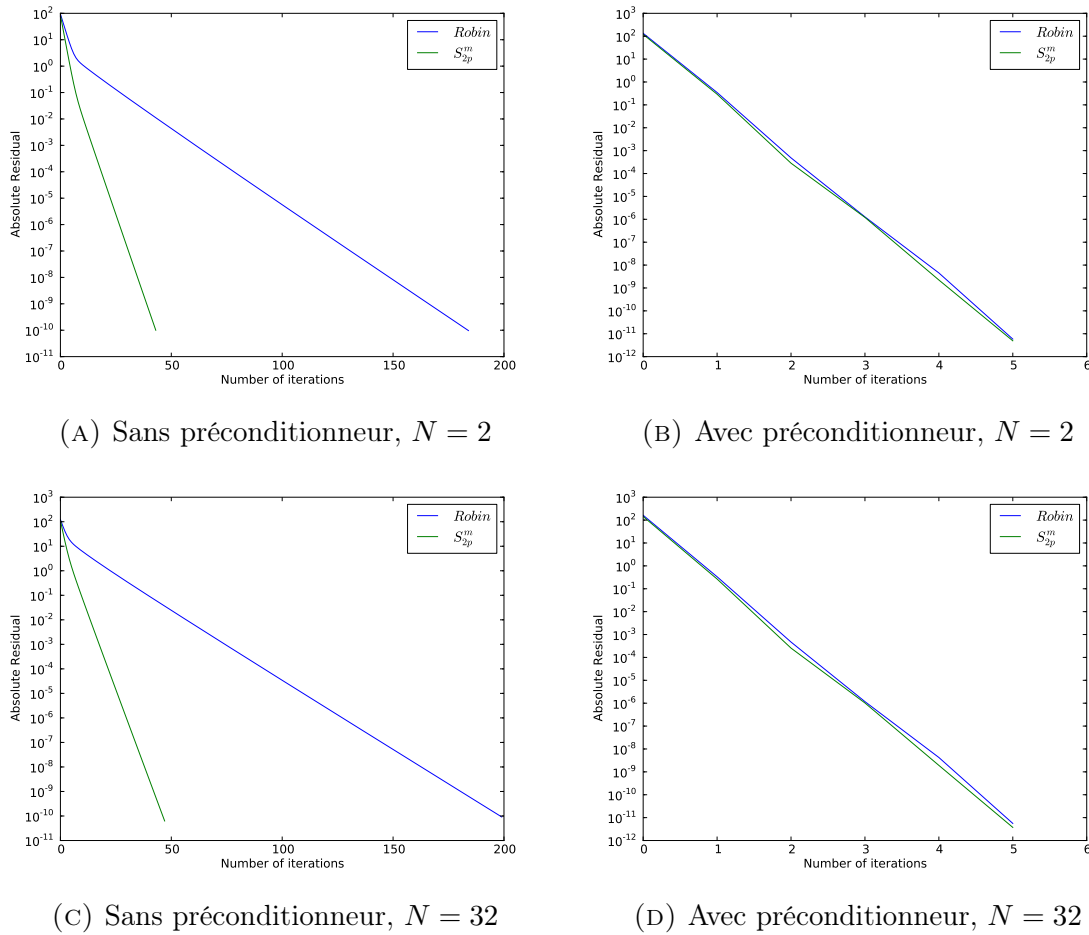


FIGURE 5.6: Historique de la convergence de la première étape du temps,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

$N$			2	4	8	16	32
Sans Pd.	Robin	$p$	29	27	29	29	28
		$N_{iter}$	185	190	193	197	199
	$S_{2p}^m$	$m$	13	14	13	13	12
		$N_{iter}$	44	45	46	47	48
Avec Pd.	Robin	$p$	29	27	29	29	28
		$N_{iter}$	6	6	6	6	6
	$S_{2p}^m$	$m$	13	14	13	13	12
		$S_{2p}^m$	6	6	6	6	6

TABLEAU 5.6: Nombre d'itérations de la première étape du temps des algorithmes sans et avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

### 5.4.3 Influence des paramètres

Nous nous intéressons à l'influence des paramètres dans les conditions de transmission, soit à

- l'ordre d'approximation de Padé ( $m$ ) dans la condition de transmission  $S_{2p}^m$ ,
- $p$  dans la condition de transmission de Robin.

Nous faisons des tests numériques avec des paramètres numériques différents pour voir l'influence du paramètre dans les algorithmes sans et avec préconditionneur. Le pas de temps est fixé à  $\Delta t = 0.01$  et le maillage est  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

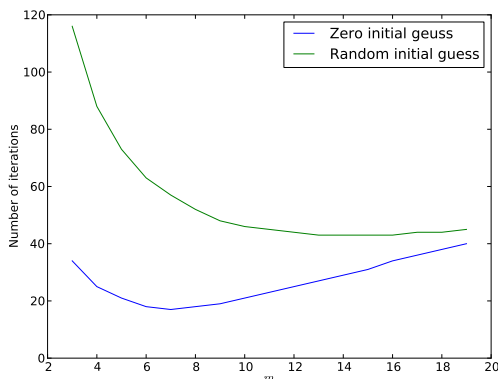
#### 5.4.3.1 Cas du potentiel $\mathcal{V} = x^2 + y^2$

L'équation de Schrödinger linéaire est considérée tout d'abord.

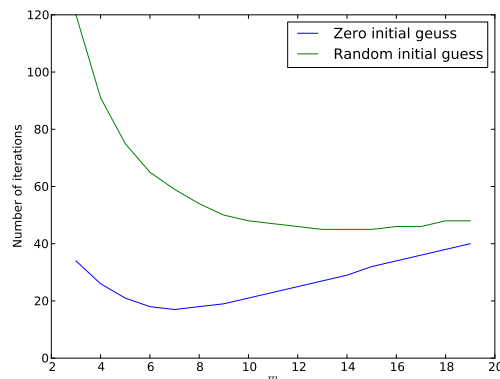
##### Influence de l'ordre $m$ de la condition de transmission $S_{2p}^m$

La figure 5.7 présente les nombres d'itérations requis par rapport à l'ordre d'approximation de Padé  $m$ . Dans les quatre sous-figures, nous considérons l'algorithme sans préconditionneur pour 2 sous-domaines, l'algorithme sans préconditionneur pour 32 sous-domaines, l'algorithme avec préconditionneur pour 2 sous-domaines et l'algorithme avec préconditionneur pour 32 sous-domaines respectivement. Dans chaque figure, la ligne verte (resp. bleu) montre les résultats en utilisant le vecteur zéro (resp. aléatoire) comme vecteur initial. Nous pouvons constater que

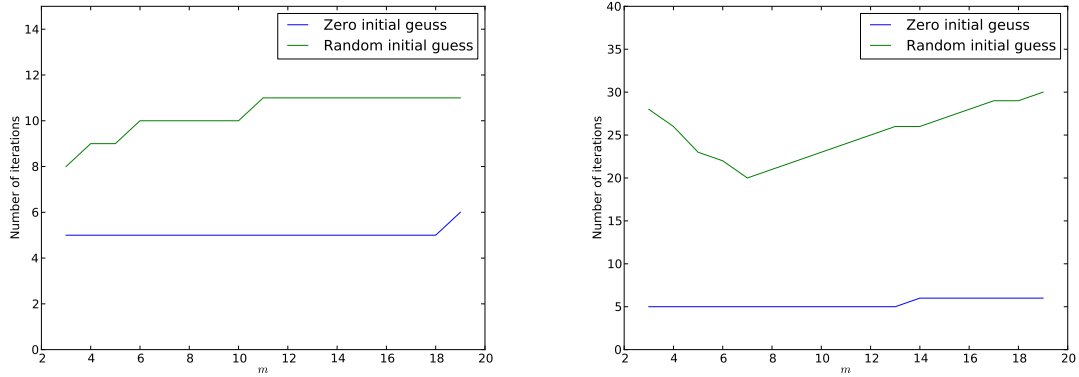
- Pour l'algorithme sans préconditionneur, si le vecteur initial est un vecteur nul, il existe un paramètre  $m$  optimal. Ce constat n'est pas conforme à nos prévisions car l'ordre le plus élevé devrait faire converger l'algorithme plus rapidement. Nous pensons que c'est le vecteur initial zéro qui nous donne des informations peu exactes.
- Pour l'algorithme sans préconditionneur, si le vecteur initial est un vecteur aléatoire, le nombre d'itérations requises commence par diminuer en augmentant l'ordre  $m$ , la concavité changeant pour  $m$  est grand. La condition de transmission  $S_{2p}^m$  étant basée sur une approximation de Padé formelle de l'opérateur racine carrée, cette approximation pourrait se dégrader pour  $m$  grand.
- L'influence de  $m$  n'est pas très importante pour l'algorithme avec préconditionneur car le préconditionneur cache les informations sur l'ordre.



(A) Sans préconditionneur,  $N = 2$



(B) Sans préconditionneur,  $N = 32$

(A) Avec préconditionneur,  $N = 2$ (B) Avec préconditionneur,  $N = 32$ FIGURE 5.8: Nombre d'itérations pour des  $m$  différents,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

### Influence de $p$ de la condition de transmission de Robin

Les nombres d'itérations requis sont montrés dans le tableaux 5.7 avec des  $p$  différents pour  $N = 2$  et  $N = 32$  (Nous montrons ici les résultats pour  $p = 5, 10, \dots, 50$ ). Nous considérons aussi les algorithmes sans et avec préconditionneur (nPd./Pd.) en utilisant le vecteur zéro (Zéro) ou le vecteur aléatoire (Aléatoire) comme vecteur initial. Nous pouvons voir que

- Pour l'algorithme avec préconditionneur, les nombres d'itérations requis sont quasiment similaires dans chaque cas (Zéro/Aléatoire,  $N = 2, 32$ ).
- Il existe un paramètre  $p$  optimal pour chaque cas (Zéro/Aléatoire,  $N = 2, 32$ ).

$p$	$N = 2$				$N = 32$			
	Zéro		Aléatoire		Zéro		Aléatoire	
	nPd.	Pd.	nPd.	Pd.	nPd.	Pd.	nPd.	Pd.
5	56	5	519	8	57	5	554	32
10	34	5	290	9	34	5	300	24
15	32	5	223	10	32	5	225	21
20	35	5	195	11	35	5	199	23
25	40	5	184	11	40	6	190	25
30	46	5	179	11	46	6	190	27
35	52	5	185	12	52	6	194	29
40	58	6	193	12	59	6	203	31
45	65	6	199	12	65	6	212	32
50	72	6	210	12	72	6	222	33

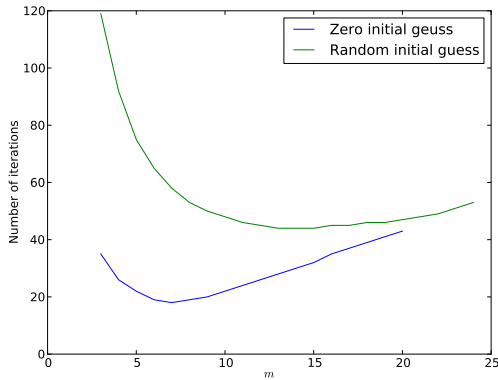
TABLEAU 5.7: Le nombre d'itérations pour des  $p$  différents.

#### 5.4.3.2 Cas du potentiel $\mathcal{V} = |u|^2$

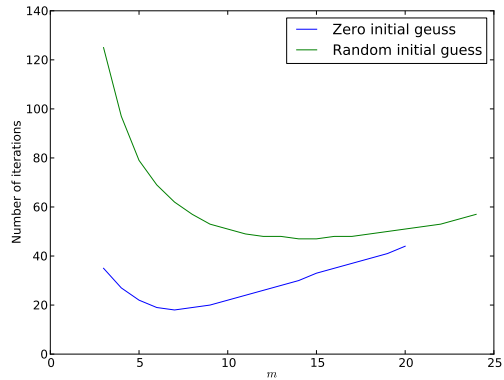
Nous reproduisons ici les tests de la sous-section précédente pour  $\mathcal{V} = |u|^2$ .

**Influence de l'ordre  $m$  de la condition de transmission  $S_{2p}^m$**

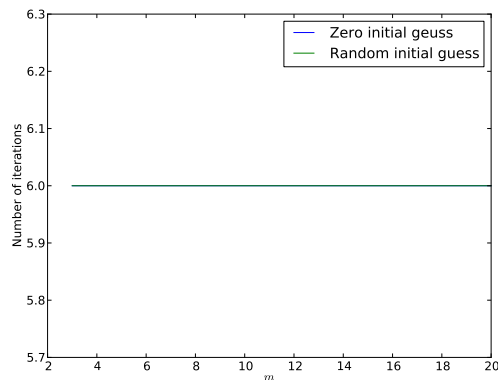
Les résultats concernant l'ordre d'approximation  $m$  sont montrés dans le tableau 5.8. Les conclusions sont les mêmes que pour le potentiel linéaire.



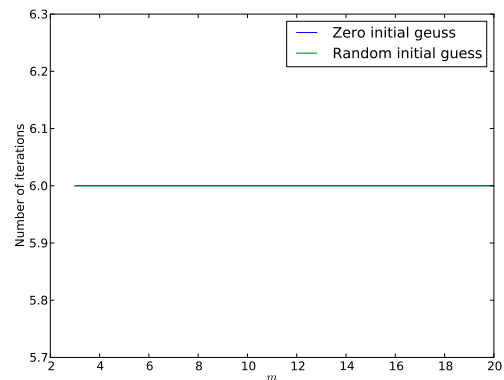
(A) Sans préconditionneur,  $N = 2$



(B) Sans préconditionneur,  $N = 32$



(A) Avec préconditionneur,  $N = 2$



(B) Avec préconditionneur,  $N = 32$

FIGURE 5.10: Nombre d'itérations avec préconditionneur,  $\Delta t = 0.01$ ,  $\Delta x = 1/128$ ,  $\Delta y = 1/8$ .

**Influence de  $p$  dans la condition de transmission de Robin**

Nous présentons les résultats dans le tableau 5.8. Les conclusions sont similaires à celles faites pour le potentiel  $\mathcal{V} = x^2 + y^2$ .

$p$	$N = 2$				$N = 32$			
	Zéro		Aléatoire		Zéro		Aléatoire	
	nPd.	Pd.	nPd.	Pd.	nPd.	Pd.	nPd.	Pd.
5	57	6	548	6	57	6	582	6
10	35	6	297	6	35	6	316	6
15	32	6	227	6	33	6	241	6
20	36	6	200	6	36	6	212	6
25	41	6	189	6	41	6	200	6
30	46	6	186	6	47	6	200	6
35	53	6	188	6	53	6	204	6
40	59	6	194	6	60	6	211	6
45	66	6	208	6	66	6	223	6
50	73	6	216	6	73	6	234	6

TABLEAU 5.8: Le nombre d'itérations pour des  $p$  différents.

#### 5.4.4 Accélération de Krylov pour l'équation de Schrödinger linéaire

Pour l'équation de Schrödinger linéaire, il est possible d'utiliser les méthodes de Krylov au lieu de la méthode de point fixe pour la résolution du système linéaire sans et avec préconditionneur

$$(I - \mathcal{L})g_{t_n} = d_{t_n},$$

$$P^{-1}(I - \mathcal{L})g_{t_n} = P^{-1}d_{t_n}.$$

Dans cette partie, nous faisons des tests numériques pour observer si les méthodes de Krylov permettent d'accélérer la convergence (le nombre d'itérations et le temps de calcul). Nous prenons  $T = 0.5$  et  $\Delta t = 0.01$ . Le vecteur initial est le vecteur nul.

Nous nous intéressons au nombre d'itérations de la première étape du temps et aux temps totaux pour  $N = 2$  et  $N = 512$  dont les maillages utilisés sont

$$N = 2 : \quad \Delta x = 1/128, \quad \Delta y = 1/8,$$

$$N = 512 : \quad \Delta x = 1/2048, \quad \Delta y = 1/64.$$

Le tableau 5.9 montre les nombres d'itération ( $N_{iter}$ ) et les temps de calcul totaux (Temps) pour les algorithmes sans (nPd.) et avec (Pd.) préconditionneur avec la condition de transmission de Robin. Nous montrons dans le tableau 5.10 les mêmes tests avec la condition de transmission  $S_{2p}^m$ . Nous pouvons faire les constats suivantes

- les méthodes de Krylov permettent de diminuer le nombre d'itérations requises, particulièrement pour l'algorithme sans préconditionneur,
- pour l'algorithme sans préconditionneur, les temps totaux des différentes méthodes sont

Point fixe > Bicgstab > Gmres,

- pour l'algorithme avec préconditionneur, si  $N$  n'est pas très grand, il est difficile de dire quelle méthode est la meilleure. En revanche, si  $N$  est grand, alors on a clairement

$$\text{Point fixe} \sim \text{Bicgstab} < \text{Gmres}.$$

La méthode Gmres prend beaucoup plus de temps que les deux autres méthodes. Dans ce cas, l'application du préconditionneur prend plus de temps. Nous n'avons cependant pas d'explications actuellement.

		$N = 2$		$N = 512$	
		$N_{iter}$	Temps	$N_{iter}$	Temps
nPd.	Point fixe	32	152.0	398	1257.2
	Gmres	11	66.0	23	112.4
	Bicgstab	6	75.2	13	122.8
Pd.	Point fixe	6	61.4	6	270.6
	Gmres	4	56.3	6	583.2
	Bicgstab	2	59.7	3	316.8

TABLEAU 5.9: Nombre d'itérations requises et temps de calcul total, Robin

		$N = 2$		$N = 512$	
		$N_{iter}$	Temps	$N_{iter}$	Temps
nPd.	Point fixe	17	142.7	60	344.9
	Gmres	9	95.2	19	157.4
	Bicgstab	5	103.6	11	172.0
Pd.	Point fixe	5	91.3	7	268.7
	Gmres	5	88.8	6	413.4
	Bicgstab	3	89.3	3	250.1

TABLEAU 5.10: Nombre d'itérations requises et temps de calcul total,  $S_{2p}^m$

## 5.5 Conclusion

Le problème que nous considérons dans ce chapitre est l'équation de Schrödinger en deux dimensions avec un potentiel général non nul, linéaire ou non-linéaire. On considère la méthode de décomposition en espace. Sur l'interface, les conditions de Robin et  $S_{2p}^m$  sont utilisées et comparées numériquement. On adapte l'idée du chapitre 3. Un opérateur préalablement défini est utilisé comme un préconditionneur. Le préconditionneur permet toujours de faire diminuer beaucoup le nombre d'itérations, mais pas nécessairement le temps de calcul total.

Pour l'équation non-linéaire, l'algorithme avec préconditionneur est plus robuste. Il nécessite moins de nombre d'itérations. Dans nos tests, il prend aussi moins de temps de calcul total.

Pour l'équation linéaire, une autre façon d'accélérer la convergence est d'utiliser les méthodes de Krylov sur l'interface au lieu de la méthode de point fixe. Nous avons montré que les

méthodes de Krylov et le préconditionneur permettent tous les deux de beaucoup diminuer le nombre d'itérations. En outre, l'algorithme sans préconditionneur accéléré par les méthodes de Krylov prend moins de temps de calcul total.





# Chapitre 6

## Méthode SWR pour l'équation de Schrödinger avec point de croisement

### Sommaire

---

<b>6.1</b>	<b>Introduction</b>	<b>121</b>
<b>6.2</b>	<b>Méthode SWR avec points de croisement</b>	<b>122</b>
<b>6.3</b>	<b>Lien avec la méthode SWR sans points de croisement</b>	<b>127</b>
<b>6.4</b>	<b>Résultats numériques</b>	<b>129</b>
6.4.1	Comparaison des conditions de transmission et des méthode Krylov	130
6.4.2	Influence des paramètres	132
6.4.3	Plus d'un point de croisement	133
<b>6.5</b>	<b>Conclusion</b>	<b>134</b>

---

### 6.1 Introduction

Dans ce chapitre, nous considérons la méthode SWR pour l'équation de Schrödinger linéaire en deux dimensions

$$\begin{cases} \mathcal{L}u := (i\partial_t + \Delta + V)u = 0, & (t, x, y) \in ]0, T[ \times \Omega_0, \\ u(0, x, y) = u_0(x, y), & (x, y) \in \Omega_0, \\ \partial_{\mathbf{n}}u = 0, & (t, x, y) \in ]0, T[ \times \partial\Omega_0, \end{cases} \quad (6.1)$$

où  $\Omega_0$  est un ensemble ouvert borné de  $\mathbb{R}^2$ ,  $\partial\Omega_0$  est le bord de  $\Omega_0$ ,  $V(t, x, y)$  est un potentiel réel et la donnée initiale  $u_0 \in L^2(\Omega_0)$ . Le domaine complet  $\Omega_0$  est décomposé en  $N$  sous-domaines  $\Omega_j, j = 1, \dots, N$  sans recouvrement avec points de croisement comme cela est montré dans la figure 6.1 (ex:  $N = 4, 9$ ). Supposons que  $\Omega_0 = ]x_l, x_r[ \times ]y_d, y_u[, x_l, x_r, y_d, y_u \in \mathbb{R}$ , alors pour  $N = 4$ , une possibilité de découpage du domaine  $\Omega_0$  est

$$\Omega_1 = ]x_l, \frac{x_l + x_r}{2}[ \times ]\frac{y_d + y_u}{2}, y_u[, \quad \Omega_2 = ]\frac{x_l + x_r}{2}, x_r[ \times ]\frac{y_d + y_u}{2}, y_u[,$$

$$\Omega_3 = ]x_l, \frac{x_l + x_r}{2}[\times]y_d, \frac{y_d + y_u}{2}[, \quad \Omega_4 = ]\frac{x_l + x_r}{2}, x_r[\times]y_d, \frac{y_d + y_u}{2}[.$$

La méthode de décomposition de domaines avec points de croisement est un problème difficile, car il faut s'assurer que la discrétisation des équations et des conditions de transmission maintienne la continuité de la solution [38]. Dans [54], l'auteur formule un système de la méthode de Schwarz optimisée sans recouvrement avec points de croisement pour l'équation de Laplace. La convergence et le choix du paramètre dans la condition de transmission de Robin sont étudiés dans [37, 54] pour l'équation de Laplace et l'équation de Helmholtz définie positive.

Le but de ce chapitre est de généraliser la méthode proposée par [54] à l'équation de Schrödinger (que l'on peut voir ici comme une équation de Laplace dépendant du temps) qui conduit à la méthode de relaxation d'ondes de Schwarz optimisée (SWR) avec points de croisement. Nous montrons dans les résultats numériques que nous pouvons obtenir la solution de (6.1) par cette méthode.

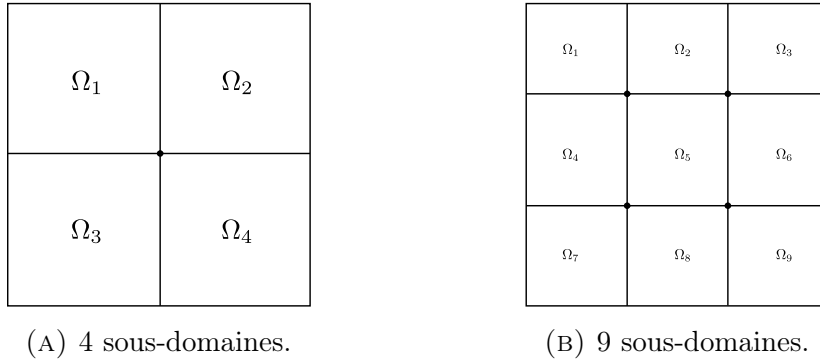


FIGURE 6.1: Décomposition de domaine avec points de croisement.

## 6.2 Méthode SWR avec points de croisement

Sans perdre de généralité, nous présentons dans cette partie la méthode pour 4 sous-domaines  $\Omega_j, j = 1, 2, 3, 4$  (voir figure 6.1a). Le problème local sur  $]0, T[\times\Omega_j$  est

$$\begin{cases} \mathcal{L}u_j = (i\partial_t + \Delta + V)u_j = 0, & (t, x, y) \in ]0, T[\times\Omega_j \\ u_j(0, x, y) = u_0(x, y), & (x, y) \in \Omega_j \\ \partial_{\mathbf{n}_j}u_j = 0, & (t, x, y) \in ]0, T[\times(\partial\Omega_0 \cap \partial\Omega_j), \\ \partial_{\mathbf{n}_j}u_j + S_ju_j = \lambda_j, & t \in ]0, T[, (x, y) \in \Gamma_j, \end{cases} \quad (6.2)$$

où  $j = 1, 2, 3, 4$ ,  $\Gamma_j = \bigcup_{k=1}^4 \partial\Omega_j \cap \partial\Omega_k$ .  $S_j$  est l'opérateur de transmission qui est choisi parmi les deux choix suivants

$$\begin{aligned} \text{Robin: } S_j &= -ip, \quad p \in \mathbb{R}, \\ S_{2p}^m : S_j &= -i\sqrt{i\partial_t + \Delta_{\Gamma_j} + V}, \end{aligned} \quad (6.3)$$

où  $\Delta_{\Gamma_j}$  est l'opérateur Laplace-Beltrami. Comme  $\Gamma_j$  n'est pas une singulière, nous ne considérons que la forme discrète de  $S_j$ . La quantité  $\lambda_j$  est ici introduite comme un flux. Nous allons établir un système linéaire d'inconnue  $\lambda_j$  qui est très similaire au problème d'interface dans les chapitres précédents. La solution globale peut être récupérée avec  $\lambda_j$ .

La discrétisation de (6.2) a été présentée en détail dans la sous-section 4.2. Par ailleurs, nous définissons

$$\lambda_{j,n} = \frac{\lambda_j(t_n, x, y) + \lambda_j(t_{n-1}, x, y)}{2} \text{ sur } \Gamma_j,$$

$\boldsymbol{\lambda}_{j,n}$ : le vecteur nodal de l'interpolation de  $\lambda_{j,n}$ .

Nous présentons ci-dessous cette méthode en utilisant la condition de transmission de Robin. Nous avons expliqué dans la remarque 4.5 que la différence entre la condition de transmission de Robin et la condition de transmission  $S_{2p}^m$  est juste liée à la définition d'une matrice ( $\mathbf{B}_j$ ). Ainsi, nous pouvons directement obtenir cette méthode pour la condition de transmission  $S_{2p}^m$ . Tout d'abord, comme nous l'avons fait dans la proposition 4.3 (voir la formule (4.19)), nous pouvons avoir sur les sous-domaines les systèmes linéaires

$$(\mathbf{A}_j - \mathbf{B}_j)\mathbf{v}_j = \mathbf{F}_j - \mathbf{M}^{\Gamma_j} \mathbf{Q}_j^T \boldsymbol{\lambda}_j, \quad j = 1, 2, 3, 4, \quad (6.4)$$

et sur le domaine complet

$$\mathbf{A}_0 \mathbf{v}_0 = \mathbf{F}_0. \quad (6.5)$$

où

$$\mathbf{A}_j = \begin{pmatrix} \mathbb{A}_{j,1} & & & & \\ -\frac{4i}{\Delta t} \mathbb{M}_j & \mathbb{A}_{j,2} & & & \\ \vdots & & \ddots & & \\ (-1)^{N_T-1} \frac{4i}{\Delta t} \mathbb{M}_j & \cdots & -\frac{4i}{\Delta t} \mathbb{M}_j & \mathbb{A}_{j,N_T} & \end{pmatrix}, \quad \mathbf{v}_j = \begin{pmatrix} \mathbf{v}_{j,1} \\ \mathbf{v}_{j,2} \\ \vdots \\ \mathbf{v}_{j,N_T} \end{pmatrix},$$

$$\mathbf{F}_j = \frac{2i}{\Delta t} \begin{pmatrix} \mathbb{M}_j \mathbf{u}_{j,0} \\ -\mathbb{M}_j \mathbf{u}_{j,0} \\ \vdots \\ (-1)^{N_T-1} \mathbb{M}_j \mathbf{u}_{j,0} \end{pmatrix}, \quad \boldsymbol{\lambda}_j = \begin{pmatrix} \boldsymbol{\lambda}_{j,1} \\ \boldsymbol{\lambda}_{j,2} \\ \vdots \\ \boldsymbol{\lambda}_{j,N_T} \end{pmatrix},$$

$\mathbf{B}_j = -ip \cdot \mathbf{M}^{\Gamma_j}$ ,  $\mathbf{Q}_j = \text{diag}_{N_T} \{Q_j\}$  et  $Q_j$  est la matrice de restriction des nœuds dans  $\Omega_j$  à ceux sur l'interface  $\Gamma_j$ . Supposons que  $\mathbf{A}_j - \mathbf{B}_j$  n'est pas singulière, alors

$$\mathbf{v}_j = (\mathbf{A}_j - \mathbf{B}_j)^{-1} (\mathbf{F}_j - \mathbf{M}^{\Gamma_j} \mathbf{Q}_j^T \boldsymbol{\lambda}_j). \quad (6.6)$$

Nous définissons les variables suivantes

$$\begin{aligned} \mathbf{v} &= (\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T, \mathbf{v}_4^T)^T, \quad \boldsymbol{\lambda} = (\boldsymbol{\lambda}_1^T, \boldsymbol{\lambda}_2^T, \boldsymbol{\lambda}_3^T, \boldsymbol{\lambda}_4^T)^T, \\ \mathbf{A} &= \text{diag}\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4\}, \quad \mathbf{B} = \text{diag}\{\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4\}, \\ \mathbf{F} &= (\mathbf{F}_1^T, \mathbf{F}_2^T, \mathbf{F}_3^T, \mathbf{F}_4^T)^T, \quad \mathbf{M}^\Gamma = \text{diag}\{\mathbf{M}^{\Gamma_1}, \mathbf{M}^{\Gamma_2}, \mathbf{M}^{\Gamma_3}, \mathbf{M}^{\Gamma_4}\}, \\ \mathbf{Q} &= \text{diag}\{\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \mathbf{Q}_4\}. \end{aligned}$$

Ainsi

$$\mathbf{v} = (\mathbf{A} - \mathbf{B})^{-1} (\mathbf{F} - \mathbf{M}^\Gamma \mathbf{Q}^T \boldsymbol{\lambda}). \quad (6.7)$$

Nous pouvons maintenant nous attarder à établir un système linéaire pour  $\boldsymbol{\lambda}$  de la forme

$$\mathbf{A}_{2LM}\boldsymbol{\lambda} = \mathbf{y}. \quad (6.8)$$

En suivant les mêmes procédés que [54], nous pouvons montrer que  $\mathbf{v}$  calculé par (6.7) complété par  $\boldsymbol{\lambda}$  solution de (6.8) satisfait  $\mathbf{v}_j = \mathbf{R}_j \mathbf{v}_0$  (voir la remarque 6.2). Nous avons besoin de trois étapes pour construire le système linéaire (6.8).

**Étape 1:** Nous désignons par  $R_j$  la matrice de restriction des nœuds dans  $\Omega_0$  à ceux dans  $\Omega_j$  et  $\mathbf{R}_j = \text{diag}_{N_T}\{R_j\}$ . En multipliant (6.4) par  $\mathbf{R}_k^T$  et sommant de 1 à 4, nous avons alors

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{A}_k \mathbf{R}_k \mathbf{v}_0 - \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{R}_k \mathbf{v}_0 = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{F}_k - \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}^{\Gamma_k} \mathbf{Q}_k^T \boldsymbol{\lambda}_k.$$

En utilisant la méthode des éléments finis en espace, nous avons les propriétés suivantes

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{A}_{k,n} \mathbf{R}_k = \mathbf{A}_{0,n}, \quad \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}_k \mathbf{R}_k = \mathbf{M}_0, \quad n = 1, 2, \dots, N_T.$$

Ainsi

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{A}_k \mathbf{R}_k = \mathbf{A}_0, \quad \mathbf{F}_0 = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{F}_k.$$

En utilisant (6.5), on obtient

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{v}_k = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{R}_k \mathbf{v}_0 = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}^{\Gamma_k} \mathbf{Q}_k^T \boldsymbol{\lambda}_k,$$

alors pour  $j = 1, 2, 3, 4$ ,

$$\mathbf{Q}_j \mathbf{R}_j \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{v}_k = \mathbf{Q}_j \mathbf{R}_j \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}^{\Gamma_k} \mathbf{Q}_k^T \boldsymbol{\lambda}_k. \quad (6.9)$$

Nous définissons

$$\mathbf{P} = \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \cdots & \mathbf{R}_1 \mathbf{R}_4^T \\ \vdots & \ddots & \vdots \\ \mathbf{R}_4 \mathbf{R}_1^T & \cdots & \mathbf{R}_4 \mathbf{R}_4^T \end{pmatrix}.$$

Après remplacement de  $\mathbf{v}_k$  dans (6.9) par la formule (6.6) qui fait intervenir  $\boldsymbol{\lambda}$ , nous obtenons

$$\mathbf{QPB}(\mathbf{A} - \mathbf{B})^{-1} \mathbf{M}^{\Gamma} \mathbf{Q}^T \boldsymbol{\lambda} + \mathbf{QPM}^{\Gamma} \mathbf{Q}^T \boldsymbol{\lambda} = \mathbf{QPB}(\mathbf{A} - \mathbf{B})^{-1} \mathbf{F}. \quad (6.10)$$

**Étape 2:** Le vecteur  $\mathbf{v}$  contient quatre valeurs du point de croisement partagée par  $\Omega_j$ ,  $j = 1, 2, 3, 4$  pour tous les pas du temps. En revanche, deux valeurs des points sont partagées sur les points d'interface de deux domaines.  $\mathbf{v}$  est dite continue si toutes les valeurs au même point sont identiques. Une autre manière pour vérifier la continuité de  $\mathbf{v}$  est de vérifier la légalité

$$\mathbf{G}\mathbf{v} = 0, \quad (6.11)$$

où  $\mathbf{G}$  est une matrice. En remplaçant  $\mathbf{v}$  par  $\boldsymbol{\lambda}$ , on a

$$\mathbf{G}(\mathbf{A} - \mathbf{B})^{-1}\mathbf{M}^\Gamma\mathbf{Q}^T\boldsymbol{\lambda} = \mathbf{G}(\mathbf{A} - \mathbf{B})^{-1}\mathbf{F}. \quad (6.12)$$

Nous montrons d'abord l'idée de la construction de la matrice  $\mathbf{G}$  et donnons cette matrice ensuite.

Les quatre éléments d'un vecteur  $(a_1, a_2, a_3, a_4)$ ,  $a_1, a_2, a_3, a_4 \in \mathbb{C}$  prennent les mêmes valeurs si et seulement si

$$\begin{pmatrix} 1 & -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = 0.$$

De même, les deux éléments d'un vecteur  $(a_1, a_2)$ ,  $a_1, a_2 \in \mathbb{C}$  prennent les mêmes valeurs si et seulement si

$$\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = 0.$$

Il est cependant possible d'utiliser d'autres matrices en lieu et place des deux définies ci-dessus. Ainsi, le choix de  $\mathbf{G}$  n'est pas unique.

Nous désignons par

- $Q_{12}$  (resp.  $Q_{14}$ ) : la matrice de restriction des nœuds dans  $\Omega_1$  à ceux sur  $\Gamma_1 \cap \Gamma_2 \setminus C_p$  (resp.  $\Gamma_1 \cap \Gamma_4 \setminus C_p$ ) où  $C_p$  est l'ensemble des points de croisement (voir figure 6.2),
- $Q_{1c}$  : la matrice de restriction des nœuds dans  $\Omega_1$  à  $C_p$ ,
- $\mathbf{Q}_{12} = \text{diag}_{N_T}\{Q_{12}\}$ ,  $\mathbf{Q}_{14} = \text{diag}_{N_T}\{Q_{14}\}$  et  $\mathbf{Q}_{1c} = \text{diag}_{N_T}\{Q_{1c}\}$ .

De la même manière, nous pouvons également définir  $Q_{21}$ ,  $Q_{23}$ ,  $Q_{2c}$ ,  $Q_{32}$ ,  $Q_{34}$ ,  $Q_{3c}$ ,  $Q_{43}$ ,  $Q_{41}$ ,  $Q_{4c}$ .

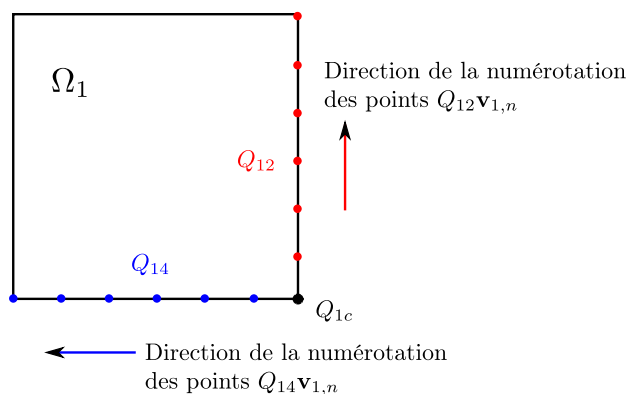


FIGURE 6.2: Illustration des définitions des matrices  $Q_{12}$ ,  $Q_{14}$  et  $Q_{1c}$ .

Si nous définissons,

$$\mathbf{G}_B = \begin{pmatrix} \mathbf{Q}_{12}^T \mathbf{Q}_{12} + \mathbf{Q}_{14}^T \mathbf{Q}_{14} & -\mathbf{Q}_{12}^T \mathbf{Q}_{21} & & -\mathbf{Q}_{14}^T \mathbf{Q}_{41} \\ -\mathbf{Q}_{21}^T \mathbf{Q}_{12} & \mathbf{Q}_{21}^T \mathbf{Q}_{21} + \mathbf{Q}_{23}^T \mathbf{Q}_{23} & -\mathbf{Q}_{23}^T \mathbf{Q}_{32} & \\ & -\mathbf{Q}_{32}^T \mathbf{Q}_{23} & \mathbf{Q}_{32}^T \mathbf{Q}_{32} + \mathbf{Q}_{34}^T \mathbf{Q}_{34} & -\mathbf{Q}_{34}^T \mathbf{Q}_{43} \\ -\mathbf{Q}_{41}^T \mathbf{Q}_{14} & & -\mathbf{Q}_{43}^T \mathbf{Q}_{34} & \mathbf{Q}_{43}^T \mathbf{Q}_{43} + \mathbf{Q}_{41}^T \mathbf{Q}_{41} \end{pmatrix},$$

$$\mathbf{G}_C = \begin{pmatrix} \mathbf{Q}_{1c}^T & & & \\ & \mathbf{Q}_{2c}^T & & \\ & & \mathbf{Q}_{3c}^T & \\ & & & \mathbf{Q}_{4c}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Q}_{1c} & -\frac{1}{3}\mathbf{Q}_{2c} & -\frac{1}{3}\mathbf{Q}_{3c} & -\frac{1}{3}\mathbf{Q}_{4c} \\ -\frac{1}{3}\mathbf{Q}_{1c} & \mathbf{Q}_{2c} & -\frac{1}{3}\mathbf{Q}_{3c} & -\frac{1}{3}\mathbf{Q}_{4c} \\ -\frac{1}{3}\mathbf{Q}_{1c} & -\frac{1}{3}\mathbf{Q}_{2c} & \mathbf{Q}_{3c} & -\frac{1}{3}\mathbf{Q}_{4c} \\ -\frac{1}{3}\mathbf{Q}_{1c} & -\frac{1}{3}\mathbf{Q}_{2c} & -\frac{1}{3}\mathbf{Q}_{3c} & \mathbf{Q}_{4c} \end{pmatrix},$$

alors

$$\mathbf{G} = \mathbf{G}_B + \mathbf{G}_C.$$

**Étape 3:** En multipliant l'équation (6.12) par  $-\mathbf{QB}$  et en ajoutant (6.10), on obtient la formule (6.8)

$$\mathbf{A}_{2LM} \boldsymbol{\lambda} = \mathbf{y},$$

où

$$\begin{aligned} \mathbf{A}_{2LM} &= (\mathbf{QPB} - \mathbf{QBG})(\mathbf{A} - \mathbf{B})^{-1} \mathbf{M}^T \mathbf{Q}^T + \mathbf{QPM}^T \mathbf{Q}^T, \\ \mathbf{y} &= (\mathbf{QPB} - \mathbf{QBG})(\mathbf{A} - \mathbf{B})^{-1} \mathbf{F}. \end{aligned}$$

Nos tests montrent que numériquement, la méthode point fixe explose pour le système linéaire (6.8). Nous résolvons donc (6.8) par les méthode de Krylov Gmres et Bicgstab. Les résultats numérique sont montrés dans la sous-section 6.4.

*Remarque 6.1.* Nous remarquons que nous ne pouvons pas appliquer cette méthode directement au cas de l'équation de Schrödinger non-linéaire avec points de croisement. Si le potentiel est non-linéaire, on peut obtenir un système similaire à (6.8), mais  $\mathbf{A}_{2LM}$  est remplacé par un opérateur non-linéaire. Puisque la méthode de point fixe explose pour le système linéaire (6.8), elle pourrait donc ne pas fonctionner non plus pour le cas non-linéaire.

*Remarque 6.2.* Nous souhaitons montrer ici que si  $\boldsymbol{\lambda}$  satisfait (6.10) et (6.12), alors le vecteur  $\mathbf{v}_j$  calculé par (6.6) satisfait  $\mathbf{v}_j = \mathbf{R}_j \mathbf{v}_0$ , c'est à dire

$$\begin{aligned} (6.10) \\ (6.12) \end{aligned} \Leftrightarrow \mathbf{v}_j = \mathbf{R}_j \mathbf{v}_0 \quad (\mathbf{v}_{j,n} = R_j \mathbf{v}_{0,n}, n = 1, 2, \dots, N_T). \quad (6.13)$$

Tout d'abord, si  $\boldsymbol{\lambda}$  satisfait (6.12), on a

$$\mathbf{G} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \end{pmatrix} = 0.$$

Ainsi, les  $\mathbf{v}_j$  prennent les valeurs identiques sur le point de croisement et les points d'interface. Il existe donc un vecteur  $\mathbf{w}$  tel que  $\mathbf{v}_j = \mathbf{R}_j \mathbf{w}$ . Nous devons ensuite montrer que  $\mathbf{w} = \mathbf{v}_0$ .

Si  $\boldsymbol{\lambda}$  satisfait (6.10), alors on a

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{v}_k = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}^{\Gamma_k} \mathbf{Q}_k^T \boldsymbol{\lambda}_k.$$

Multipliant (6.4) par  $\mathbf{R}_k^T$  et sommant de 1 à 4, nous obtenons

$$\sum_{k=1}^4 \mathbf{R}_k^T \mathbf{A}_k \mathbf{R}_k \mathbf{w} - \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{B}_k \mathbf{R}_k \mathbf{v}_k = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{F}_k - \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{M}^{\Gamma_k} \mathbf{Q}_k^T \boldsymbol{\lambda}_k.$$

Ainsi

$$\mathbf{A}_0 \mathbf{w} = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{A}_k \mathbf{R}_k \mathbf{w} = \sum_{k=1}^4 \mathbf{R}_k^T \mathbf{F}_k = \mathbf{F}_0.$$

En comparant la formule ci-dessus avec (6.5), nous pouvons conclure que  $\mathbf{w} = \mathbf{v}_0$ .

### 6.3 Lien avec la méthode SWR sans points de croisement

Nous montrons dans cette sous-section que la méthode SWR avec points de croisement est une généralisation de la méthode SWR sans points de croisement. Afin d'éviter d'introduire beaucoup de notations et symboles qui pourraient compliquer l'exposé, nous considérons la condition de transmission de Robin dans cette partie. La configuration géométrique dans cette sous-section est différente de celle du reste du chapitre. Le domaine complet  $\Omega_0$  est décomposé en des sous-domaines sans points de croisement comme nous le faisons dans les chapitre 4 et 5. Sans perdre de généralité, on considère ici deux sous-domaines  $\Omega_1$  et  $\Omega_2$ . Pour simplifier la présentation, la condition Neumann homogène est utilisée sur le bord de  $\Omega_0$ . Nous pouvons également construire un système linéaire en suivant les trois étapes de la sous-section précédente. Le premier et le troisième étapes sont tout à fait mêmes. Il suffit de redéfinir la matrice  $\mathbf{G}$ :

$$\mathbf{G} = \mathbf{G}_B,$$

où  $\mathbf{Q}_1 = \text{diag}_{N_T} \{Q_1\}$ ,  $\mathbf{Q}_2 = \text{diag}_{N_T} \{Q_2\}$  et

$$\mathbf{G}_B = \begin{pmatrix} \mathbf{Q}_1^T \mathbf{Q}_1 & -\mathbf{Q}_1^T \mathbf{Q}_2 \\ -\mathbf{Q}_2^T \mathbf{Q}_1 & \mathbf{Q}_2^T \mathbf{Q}_2 \end{pmatrix}.$$

La définition de  $Q_1$  (resp.  $Q_2$ ) est illustrée dans la figure 6.3. On enlève la matrice  $\mathbf{G}_C$  dans  $\mathbf{G}$  puisque il n'y a pas de points de croisement. Ce système linéaire est donné par

$$\mathbf{A}_{ncp} \begin{pmatrix} \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{pmatrix} = \mathbf{y}_{ncp}, \quad (6.14)$$

où

$$\mathbf{A}_{ncp} = \left( \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \mathbf{R}_1 \mathbf{R}_2^T \\ \mathbf{R}_2 \mathbf{R}_1^T & \mathbf{R}_2 \mathbf{R}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \\ & \mathbf{B}_2 \end{pmatrix} - \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \\ & \mathbf{B}_2 \end{pmatrix} \mathbf{G} \right)$$

$$\begin{aligned} & \cdot \begin{pmatrix} \mathbf{A}_1 - \mathbf{B}_1 & \\ & \mathbf{A}_2 - \mathbf{B}_2 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{M}^{\Gamma_1} & \\ & \mathbf{M}^{\Gamma_2} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix}^T \\ & + \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \mathbf{R}_1 \mathbf{R}_2^T \\ \mathbf{R}_2 \mathbf{R}_1^T & \mathbf{R}_2 \mathbf{R}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{M}^{\Gamma_1} & \\ & \mathbf{M}^{\Gamma_2} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix}^T, \end{aligned}$$

et

$$\begin{aligned} y_{ncp} = & \left( \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \mathbf{R}_1 \mathbf{R}_2^T \\ \mathbf{R}_2 \mathbf{R}_1^T & \mathbf{R}_2 \mathbf{R}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \\ & \mathbf{B}_2 \end{pmatrix} - \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \\ & \mathbf{B}_2 \end{pmatrix} \mathbf{G} \right) \\ & \cdot \begin{pmatrix} \mathbf{A}_1 - \mathbf{B}_1 & \\ & \mathbf{A}_2 - \mathbf{B}_2 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{pmatrix}. \end{aligned}$$

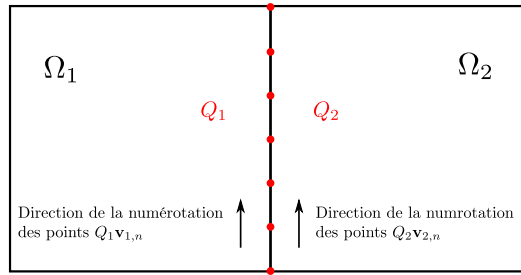


FIGURE 6.3: Illustration des définitions des matrices  $Q_1$  et  $Q_2$ ,  $N = 2$  sans points de croisement.

Nous allons montrer que si la méthode point fixe est appliquée à (6.14), nous pouvons retrouver l'algorithme SWR classique présenté dans le chapitre 4. Tout à bord, nous pouvons vérifier les propriétés suivantes en utilisant les définitions des matrices

- $\mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_1^T \mathbf{Q}_1^T = (\mathbf{Q}_1 \mathbf{R}_1) \cdot (\mathbf{Q}_1 \mathbf{R}_1)^T = I$ ,  $\mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_2^T \mathbf{Q}_2^T = (\mathbf{Q}_1 \mathbf{R}_1) \cdot (\mathbf{Q}_2 \mathbf{R}_2)^T = I$ ,  
 $\mathbf{Q}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{Q}_1^T = (\mathbf{Q}_2 \mathbf{R}_2) \cdot (\mathbf{Q}_1 \mathbf{R}_1)^T = I$ ,  $\mathbf{Q}_2 \mathbf{R}_2 \mathbf{R}_2^T \mathbf{Q}_2^T = (\mathbf{Q}_2 \mathbf{R}_2) \cdot (\mathbf{Q}_2 \mathbf{R}_2)^T = I$ ,
- $\mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T = -ip \cdot \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} \mathbf{Q}_1^T = -ip \cdot \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \mathbf{Q}_2^T = \mathbf{Q}_2 \mathbf{B}_2 \mathbf{Q}_2^T$ .

Ainsi on a

$$\begin{aligned} & \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \mathbf{R}_1 \mathbf{R}_2^T \\ \mathbf{R}_2 \mathbf{R}_1^T & \mathbf{R}_2 \mathbf{R}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{M}^{\Gamma_1} & \\ & \mathbf{M}^{\Gamma_2} \end{pmatrix} \\ & = \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{R}_1 \mathbf{R}_1^T & \mathbf{R}_1 \mathbf{R}_2^T \\ \mathbf{R}_2 \mathbf{R}_1^T & \mathbf{R}_2 \mathbf{R}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{M}^{\Gamma_1} & \\ & \mathbf{M}^{\Gamma_2} \end{pmatrix} \\ & = \begin{pmatrix} \mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_1^T \mathbf{Q}_1^T & \mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_2^T \mathbf{Q}_2^T \\ \mathbf{Q}_2 \mathbf{R}_2 \mathbf{R}_1^T \mathbf{Q}_1^T & \mathbf{Q}_2 \mathbf{R}_2 \mathbf{R}_2^T \mathbf{Q}_2^T \end{pmatrix} \begin{pmatrix} \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} & \\ & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \\ \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \end{pmatrix}, \end{aligned}$$

et

$$\begin{pmatrix} \mathbf{Q}_1 & \\ & \mathbf{Q}_2 \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 & \\ & \mathbf{B}_2 \end{pmatrix} \mathbf{G} = \begin{pmatrix} \mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T \mathbf{Q}_1 & -\mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T \mathbf{Q}_2 \\ -\mathbf{Q}_2 \mathbf{B}_2 \mathbf{Q}_2^T \mathbf{Q}_1 & \mathbf{Q}_2 \mathbf{B}_2 \mathbf{Q}_2^T \mathbf{Q}_2 \end{pmatrix}$$



$$= \begin{pmatrix} \mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T \mathbf{Q}_1 & -\mathbf{Q}_2 \mathbf{B}_2 \mathbf{Q}_2^T \mathbf{Q}_2 \\ -\mathbf{Q}_1 \mathbf{B}_1 \mathbf{Q}_1^T \mathbf{Q}_1 & \mathbf{Q}_2 \mathbf{B}_2 \mathbf{Q}_2^T \mathbf{Q}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_1 \mathbf{B}_1 & -\mathbf{Q}_2 \mathbf{B}_2 \\ -\mathbf{Q}_1 \mathbf{B}_1 & \mathbf{Q}_2 \mathbf{B}_2 \end{pmatrix}.$$

Ensuite

$$\begin{aligned} \mathbf{y}_{ncp} - \mathbf{A}_{ncp} \begin{pmatrix} \lambda_1^k \\ \lambda_2^k \end{pmatrix} &= \begin{pmatrix} \mathbf{Q}_1 \mathbf{B}_1 & \mathbf{Q}_2 \mathbf{B}_2 \\ \mathbf{Q}_1 \mathbf{B}_1 & \mathbf{Q}_2 \mathbf{B}_2 \end{pmatrix} \begin{pmatrix} (\mathbf{A}_1 - \mathbf{B}_1)^{-1} (\mathbf{F}_1 - \mathbf{B}_1 \mathbf{Q}_1^T \lambda_1^k) \\ (\mathbf{A}_2 - \mathbf{B}_2)^{-1} (\mathbf{F}_2 - \mathbf{B}_2 \mathbf{Q}_2^T \lambda_2^k) \end{pmatrix} \\ &\quad - \begin{pmatrix} \mathbf{Q}_1 \mathbf{B}_1 & -\mathbf{Q}_2 \mathbf{B}_2 \\ -\mathbf{Q}_1 \mathbf{B}_1 & \mathbf{Q}_2 \mathbf{B}_2 \end{pmatrix} \begin{pmatrix} (\mathbf{A}_1 - \mathbf{B}_1)^{-1} (\mathbf{F}_1 - \mathbf{B}_1 \mathbf{Q}_1^T \lambda_1^k) \\ (\mathbf{A}_2 - \mathbf{B}_2)^{-1} (\mathbf{F}_2 - \mathbf{B}_2 \mathbf{Q}_2^T \lambda_2^k) \end{pmatrix} \\ &\quad - \begin{pmatrix} \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} \mathbf{Q}_1^T & \\ & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \mathbf{Q}_2^T \end{pmatrix} \begin{pmatrix} \lambda_1^k + \lambda_2^k \\ \lambda_1^k + \lambda_2^k \end{pmatrix} \\ &= 2 \begin{pmatrix} \mathbf{Q}_2 \mathbf{B}_2 \mathbf{v}_2^k \\ \mathbf{Q}_1 \mathbf{B}_1 \mathbf{v}_1^k \end{pmatrix} - \begin{pmatrix} \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} \mathbf{Q}_1^T & \\ & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \mathbf{Q}_2^T \end{pmatrix} \begin{pmatrix} \lambda_1^k + \lambda_2^k \\ \lambda_1^k + \lambda_2^k \end{pmatrix}. \end{aligned}$$

Finalement, on peut avoir

$$\begin{aligned} \begin{pmatrix} \lambda_1^{k+1} \\ \lambda_2^{k+1} \end{pmatrix} &= \begin{pmatrix} \lambda_1^k \\ \lambda_2^k \end{pmatrix} + \begin{pmatrix} \mathbf{Q}_1 \mathbf{M}^{\Gamma_1} \mathbf{Q}_1^T & \\ & \mathbf{Q}_2 \mathbf{M}^{\Gamma_2} \mathbf{Q}_2^T \end{pmatrix}^{-1} \left( \mathbf{y}_{ncp} - \mathbf{A}_{ncp} \begin{pmatrix} \lambda_1^k \\ \lambda_2^k \end{pmatrix} \right) \\ &= - \begin{pmatrix} \lambda_2^k \\ \lambda_1^k \end{pmatrix} + 2 \begin{pmatrix} (\mathbf{Q}_2 \mathbf{M}^{\Gamma_1} \mathbf{Q}_2^T)^{-1} \mathbf{Q}_2 \mathbf{B}_2 \mathbf{v}_2^k \\ (\mathbf{Q}_1 \mathbf{M}^{\Gamma_2} \mathbf{Q}_1^T)^{-1} \mathbf{Q}_1 \mathbf{B}_1 \mathbf{v}_1^k \end{pmatrix} = - \begin{pmatrix} \lambda_2^k \\ \lambda_1^k \end{pmatrix} - 2ip \cdot \begin{pmatrix} \mathbf{Q}_2 \mathbf{v}_2^k \\ \mathbf{Q}_1 \mathbf{v}_1^k \end{pmatrix}. \quad (6.15) \end{aligned}$$

Ceci n'est rien d'autre que la méthode SWR classique.

Nous pouvons voir par les calculs ci-dessus que la méthode SWR avec points de croisement est une généralisation de la méthode SWR classique sans points de croisement. Une condition de continuité ( $\mathbf{G}_c \mathbf{v} = 0$ ) est ajoutée qui assure que sur le point de croisement, les  $\mathbf{v}_j$ ,  $j = 1, 2, 3, 4$ , prennent les valeurs identiques. Ainsi nous pouvons établir une équivalence entre la solution globale ( $\mathbf{v}_0$ ) et les solutions locales ( $\mathbf{v}_j, j = 1, 2, 3, 4$ ) (voir formule (6.13)). À partir de (6.13), un système linéaire de flux  $\lambda$  est établi et résolu.

## 6.4 Résultats numériques

Nous présentons quelques résultats numériques dans cette partie. Nous fixons le temps final à  $T = 0.5$  et le pas temps à  $\Delta t = 0.01$ . Le vecteur nul est utilisé comme vecteur initial. Le potentiel et la donnée initiale sont respectivement

$$V = x^2 + y^2, \quad u_0 = e^{-x^2 - y^2 - 0.5xi}.$$

Dans les sous-sections 6.4.1 et 6.4.2, le domaine complet  $\Omega_0 = ] - 21, 21[ \times ] - 21, 21[$  est décomposé en quatre sous-domaines:

$$\begin{aligned} \Omega_1 &= ] - 21, 0[ \times ] 0, 21[, \quad \Omega_2 = ] 0, 21[ \times ] 0, 21[, \\ \Omega_3 &= ] - 21, 0[ \times ] - 21, 0[, \quad \Omega_4 = ] 0, 21[ \times ] - 21, 0[. \end{aligned}$$

Nous utilisons trois maillages différents qui ont respectivement 9465, 58150 et 233368 nœuds sur le domaine complet  $\Omega_0$ . Nous utilisons Gmsh [40] pour fabriquer les maillages triangulaires (voir figure 6.4a).

Dans la sous-section 6.4.3, nous faisons les tests pour plus qu'un point de croisement. Le domaine complet est décomposé en 9 sous-domaines:

$$\begin{aligned} \Omega_1 &= ] - 21, -7[ \times ] 7, 21[, \quad \Omega_2 = ] - 7, 7[ \times ] 7, 21[, \quad \dots \\ \Omega_4 &= ] - 21, -7[ \times ] - 7, 7[, \quad \dots, \quad \Omega_9 = ] 7, 21[ \times ] - 21, -7[. \end{aligned}$$

Il est plus difficile de fabriquer des maillages contenant les mêmes nombres de nœuds que dans le cas  $N = 4$ . Ainsi nous n'utilisons que deux maillages qui ont 9472 et 233537 nœuds sur le domaine complet.

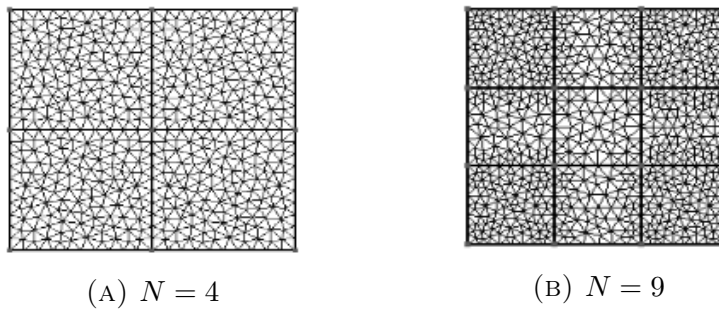


FIGURE 6.4: Exemples des maillages fabriqués par Gmsh.

### 6.4.1 Comparaison des conditions de transmission et des méthode Krylov

D'abord, nous présentons dans la figure 6.5 l'historique de la convergence avec la condition de transmission de Robin pour les trois maillages. " $N_p$ " désigne le nombre de nœuds sur  $\Omega_0$ . Le système linéaire (6.8) est résolu par Gmres ou Bicgstab. Le paramètre  $p$  est l'optimal numérique. Nous pouvons faire les constats suivants

- La convergence n'est pas très sensible au maillage. En revanche, le maillage  $N_p = 9465$  est trop grossier pour la résolution de l'équation de Schrödinger. Ainsi la convergence avec ce maillage n'est pas très convainquante.
- Le nombre d'itérations requises avec la méthode Bicgstab est inférieur à celui obtenu avec la méthode Gmres.

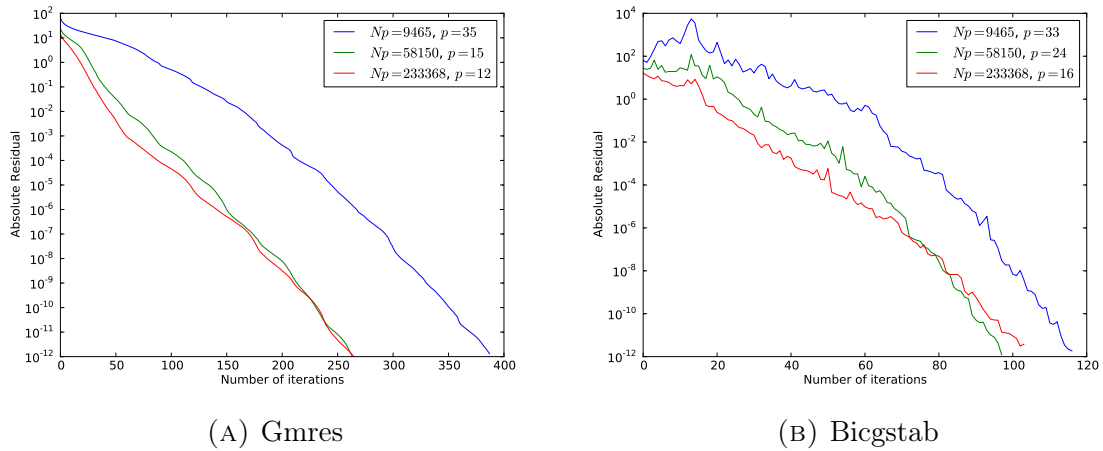
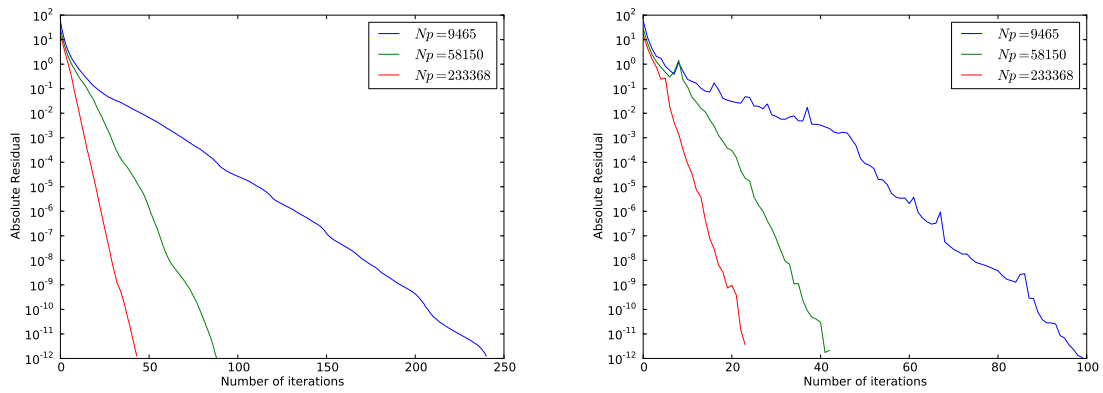


FIGURE 6.5: Historique de la convergence, Robin.

Nous reproduisons ensuite les tests avec la condition de transmission  $S_{2p}^m$ ,  $m = 50$ . L'historique de la convergence est montré dans la figure 6.6. Nous pouvons voir que l'algorithme converge plus vite avec un maillage plus fin. Nous constatons également que le nombre d'itérations requises avec la méthode Bicgstab est moindre qu'avec la méthode Gmres.

FIGURE 6.6: Historique de la convergence,  $S_{2p}^m$ ,  $m = 50$ .

Nous montrons enfin dans le tableau 6.1 les temps de calcul des tests qui réalisés pour la figure 6.5 et la figure 6.6. Les temps plus petits pour chaque maillage sont soulignés. Pour un maillage fin, la condition de transmission  $S_{2p}^m$  est bien meilleure que celle de Robin. Il est cependant difficile de conclure quel algorithme est meilleur entre Gmres et Bicgstab.

		9465	58150	233368
$S_{2p}^m$	Gmres, m=20	64.2	80.8	<u>146.9</u>
	Gmres, m=50	87.3	109.5	163.6
	Gmres, m=100	152.0	178.2	245.5
	Bicgstab, m=20	40.8	<u>71.8</u>	159.6
	Bicgstab, m=50	70.2	104.0	174.2
	Bicgstab, m=100	102.8	166.6	249.6
Robin	Gmres	25.0	117.9	517.1
	Bicgstab	<u>15.1</u>	86.2	397.7

TABLEAU 6.1: Temps de calcul total en secondes.

## 6.4.2 Influence des paramètres

Nous nous intéressons d'abord à l'influence de l'ordre d'approximation Padé  $m$ . Nous présentons dans la figure 6.7 l'historique de la convergence avec des  $m$  différents pour un maillage fin. Nous pouvons faire les constats similaires à ceux des chapitres 4 et 5 : à partir d'un certain niveau, si  $m$  est assez grand, on ne peut plus faire converger l'algorithme plus rapidement en augmentant l'ordre  $m$ . Cependant, le seuil d'ordre obtenu ici (entre 20 et 50) est plus grand que celui dans les deux chapitres précédents. Une des raisons possible est que nous utilisons la méthode SWR au lieu de la méthode de décomposition en espace. Pour vérifier notre explication, nous faisons des tests en fixant le temps final  $T = \Delta t$ , ainsi il y a qu'une étape du temps. La figure 6.8 montre les résultats des tests. Nous pouvons voir que le seuil est entre 10 et 20.

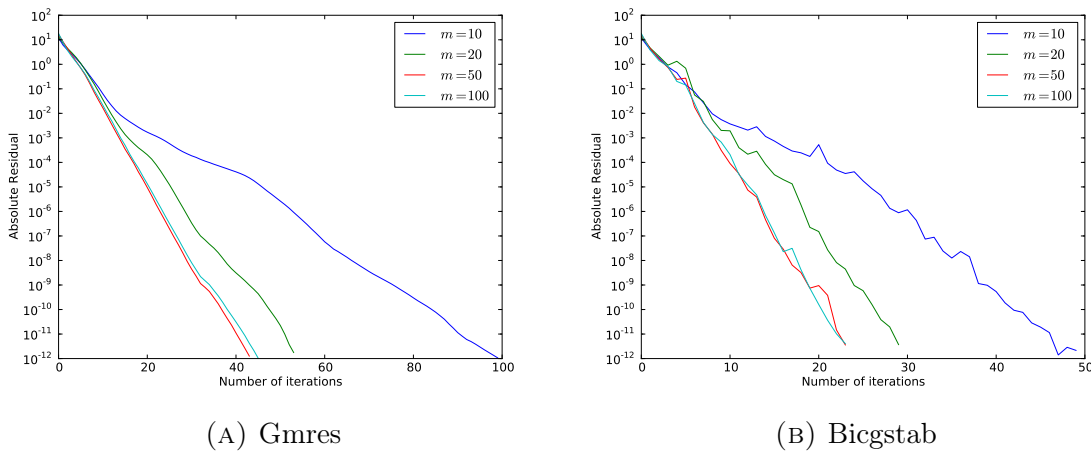


FIGURE 6.7: Historique de la convergence pour l'ordre de Padé  $m$  différents,  $Np = 233368$ ,  $T = 0.5$ ,  $\Delta t = 0.01$ .

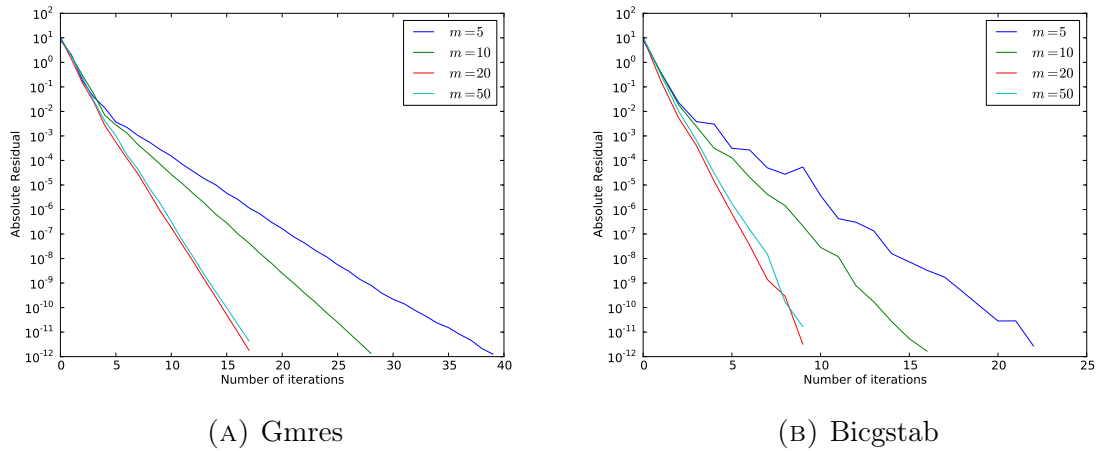


FIGURE 6.8: Historique de la convergence pour l'ordre de Padé  $m$  différents,  $Np = 233368$ ,  $T = \Delta t = 0.01$ .

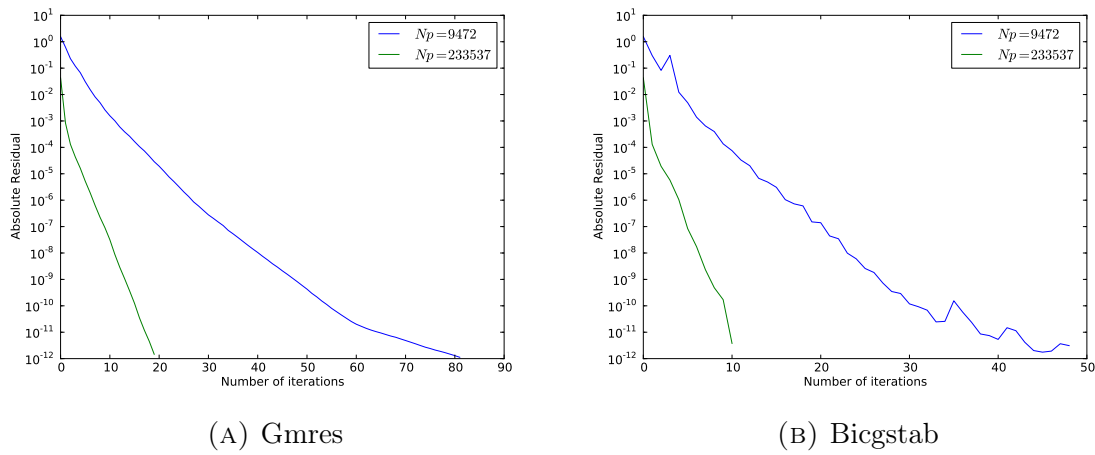
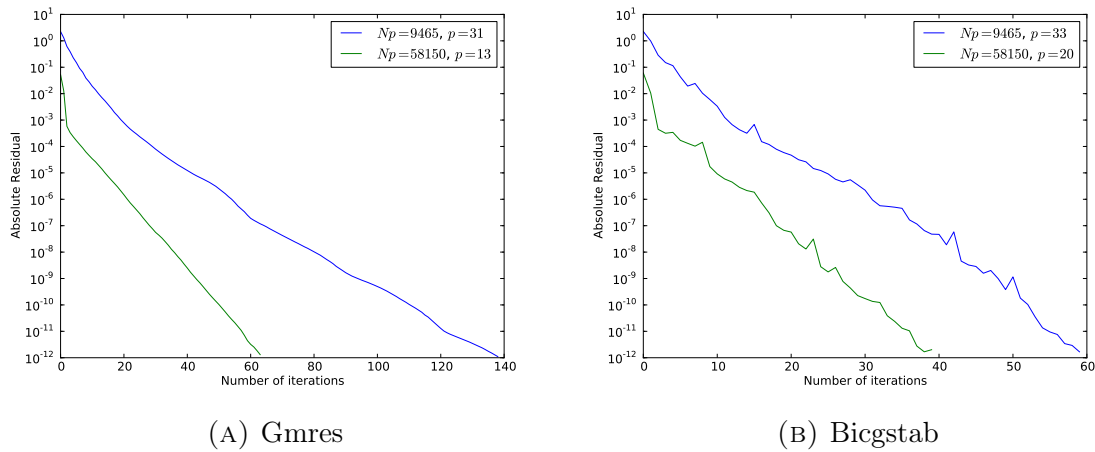
Nous considérons maintenant l'influence du paramètre  $p$  dans la condition de transmission de Robin. Nous présentons dans la figure 6.2 les nombres requis pour des  $p$  différents en utilisant les trois maillages et les deux méthodes de Krylov. Les meilleurs sont mis en avant par la couleur rouge. Nous pouvons voir qu'il existe un  $p$  optimal.

$Np$	Gmres			Bicgstab		
	9465	58150	233368	9465	58150	233368
$p = 10$	870	374	289	291	136	113
$p = 15$	634	264	257	193	112	107
$p = 20$	501	283	272	154	105	106
$p = 25$	435	277	271	134	102	111
$p = 30$	389	302	283	122	102	116
$p = 35$	388	311	289	122	105	119
$p = 40$	417	294	323	126	110	127
$p = 45$	449	309	338	133	117	129
$p = 50$	478	339	329	139	122	139
$p = 12$			265			
$p = 33$				117		
$p = 24$					98	
$p = 16$						104

TABLEAU 6.2: Le nombre d'itérations pour des  $p$  différents.

### 6.4.3 Plus d'un point de croisement

Nous montrons des tests pour  $3 \times 3$  sous-domaines (voir figure 6.4b). La figure 6.9 (resp. la figure 6.10) présente l'historique de la convergence avec la condition de transmission  $S_{2p}^m$  (resp. Robin). Nous pouvons voir que cette méthode fonctionne pour plus d'un point de croisement. Nous pouvons aussi constater que l'influence du maillage sur la convergence est similaire au cas  $N = 4$ .


 FIGURE 6.9: Historique de la convergence,  $N = 9$ ,  $S_{2p}^m$ ,  $m = 50$ .

 FIGURE 6.10: Historique de la convergence,  $N = 9$ , Robin.

## 6.5 Conclusion

Dans ce chapitre, nous généralisons une méthode proposée dans [37, 54] pour l'équation de Laplace et l'équation de Helmholtz définie positive avec points de croisement à l'équation de Schrödinger qui conduit à la méthode de SWR avec points de croisement. Nous utilisons deux conditions de transmission sur l'interface à savoir Robin et  $S_{2p}^m$  (version discrète). Dans la sous-section 6.4 (résultats numériques), nous montrons que cette méthode fonctionne pour l'équation de Schrödinger avec un ou plusieurs points de croisement. Par ailleurs, nous comparons les conditions de transmission et analysons les influences des paramètres numériquement.

## Partie III

# Simulation d'un condensat de Bose-Einstein





# Chapitre 7

## Simulation de condensat de Bose-Einstein

### Sommaire

---

<b>7.1</b>	<b>Introduction</b>	<b>137</b>
<b>7.2</b>	<b>GPE sous coordonnées transformées de Lagrange</b>	<b>138</b>
<b>7.3</b>	<b>Méthode de décomposition de domaines</b>	<b>139</b>
<b>7.4</b>	<b>Résultats numériques</b>	<b>140</b>
7.4.1	Comparaison des méthodes	140
7.4.2	Dynamique des réseaux de vortex quantifiés	142
<b>7.5</b>	<b>Conclusion</b>	<b>145</b>

---

### 7.1 Introduction

Un condensat de Bose-Einstein (BEC) est un état de la matière d'un gaz dilué de bosons refroidis à des températures très proches du zéro absolu. Dans ces conditions, une grande partie des bosons occupent l'état quantique le plus bas, les effets quantiques devenant alors apparents à l'échelle macroscopique. Ces effets sont appelés phénomènes quantiques macroscopiques. Albert Einstein et Satyendranath Bose ont prédit ce phénomène en 1925. Le premier condensat gazeux a été produit en 1995 par Eric Cornell et Carl Wieman.

Nous nous intéressons à la simulation de BEC avec rotation en deux dimensions dans ce chapitre. Un des modèles pour BEC est l'équation Gross-Pitaevskii (GPE) [2, 13, 16, 18]:

$$\begin{cases} i\partial_t u + \frac{1}{2}\Delta u - V(x, y)u - \beta|u|^2u + \omega \cdot L_z u = 0, \\ u(0, x, y) = u_0(x, y), \end{cases} \quad (7.1)$$

où  $(t, x, y) \in ]0, T[ \times \mathbb{R}^2$ . La constante  $\beta$  décrit la force des interactions à courte portée à deux corps (positive pour l'interaction répulsive et négative pour l'interaction attractive) dans un

condensat. La constante  $\omega \in \mathbb{R}$  représente la vitesse angulaire, et le moment angulaire  $L_z$  est

$$L_z = -i(x\partial_y - y\partial_x).$$

Le potentiel que nous considérons ici est

$$V(x, y) = \frac{1}{2}(\gamma_x^2 x^2 + \gamma_y^2 y^2), \quad \gamma_x, \gamma_y \in \mathbb{R}.$$

L'équation GPE est donc un type de l'équation de Schrödinger non-linéaire.

Notre but dans ce chapitre est d'appliquer les méthodes de décompositions de domaine que nous avons présentés dans le chapitre 5 à la simulation de BEC. Comme nous l'avons vu, les méthodes nous permettent de produire des algorithmes parallèles qui permettent de diminuer le temps de calcul total, mais aussi de réaliser des simulations non possibles sur un seul nœud de calcul.

## 7.2 GPE sous coordonnées transformées de Lagrange

Une des difficultés dans les simulations des condensats de Bose-Einstein provient du terme de rotation. Récemment, les auteurs de [17] ont introduit une transformation de coordonnées qui permettent d'écrire l'équation GPE dans ce nouveau jeu de coordonnées transformées (7.3) pour lesquelles le terme de rotation ne pose plus de difficultés. Le prix à payer est de transformer une équation GPE avec terme de rotation en une nouvelle équation de Schrödinger non linéaire avec un potentiel dépendant du temps. Les résultats que nous avons obtenus dans les chapitres précédents sont toutefois applicables dans ce cas.

Pour  $\forall t \geq 0$ , nous définissons la matrice de rotation orthogonale  $A(t)$  par

$$A(t) = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix}.$$

Nous introduisons les coordonnées transformées de Lagrange  $(\tilde{x}, \tilde{y})$  par

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = A^{-1}(t) \begin{pmatrix} x \\ y \end{pmatrix} = A^T(t) \begin{pmatrix} x \\ y \end{pmatrix}, \quad (x, y)^T \in \mathbb{R}^2. \quad (7.2)$$

Dans ces nouvelles coordonnées, l'équation GPE (7.1) se transforme en

$$\begin{cases} i\partial_t \tilde{u} + \frac{1}{2}\Delta \tilde{u} - V_t(t, \tilde{x}, \tilde{y})\tilde{u} - \beta|\tilde{u}|^2\tilde{u} = 0, \\ \tilde{u}(0, \tilde{x}, \tilde{y}) = \tilde{u}_0(\tilde{x}, \tilde{y}), \end{cases} \quad (7.3)$$

où pour  $\forall t \geq 0$ ,  $(\tilde{x}, \tilde{y}) \in \mathbb{R}^2$ ,

$$\tilde{u}(t, \tilde{x}, \tilde{y}) := u(t, x, y), \quad V_t(t, \tilde{x}, \tilde{y}) := V(x, y), \quad \text{où } (x, y)^T = A(t)(\tilde{x}, \tilde{y})^T, \quad (7.4)$$

et la donnée initiale  $\tilde{u}_0(\tilde{x}, \tilde{y})$  est

$$\tilde{u}_0(\tilde{x}, \tilde{y}) = u_0(\tilde{x}, \tilde{y}).$$

Dans [17], les auteurs utilisent une méthode TSSP (time-splitting spectral method) pour résoudre (7.3) et ainsi calculer  $\tilde{u}$ .

Une fois la solution  $\tilde{u}$  obtenue numériquement, il est possible de reconstruire la solution  $u$ . D'après les formules (7.2) (7.4), on a

$$u(t, x, y) = \tilde{u}(t, \tilde{x}, \tilde{y}),$$

et

$$\begin{pmatrix} x \\ y \end{pmatrix} = A(t) \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \cos(\omega t) & \sin(\omega t) \\ -\sin(\omega t) & \cos(\omega t) \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix},$$

où la matrice  $A(t)$  définit une rotation pour le temps  $t$ . Au temps  $t$ , le domaine de définition de  $\tilde{u}(t, \tilde{x}, \tilde{y})$  est  $(\tilde{x}, \tilde{y}) \in \Omega$  et le domaine de définition de  $u(t, x, y)$  est  $(x, y) \in A(t)\Omega$  (voir figure 7.1). Les domaines  $A(t)\Omega$  pour  $t \geq 0$  sont différents, mais ils partagent un disque commun. La valeur de  $u(t, x, y)$  à l'intérieur du carré maximal (la zone valide) qui se trouve entièrement dans le disque interne peut être calculé facilement par interpolation. La zone valide est

$$\left] \frac{a_0}{\sqrt{2}}, \frac{b_0}{\sqrt{2}} \right[ \times \left] \frac{y_d}{\sqrt{2}}, \frac{y_u}{\sqrt{2}} \right[.$$

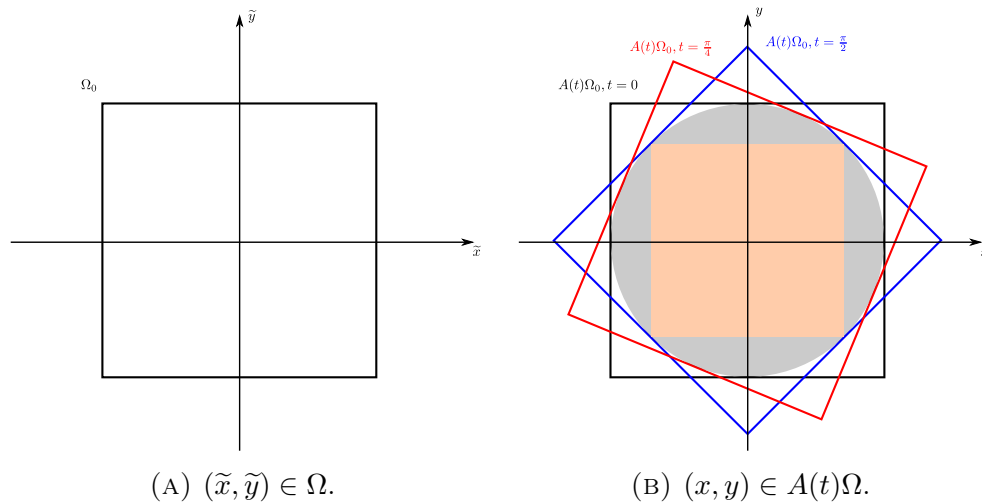


FIGURE 7.1: (a) Le domaine de calcul  $(\tilde{x}, \tilde{y}) \in \Omega$ . (b) Le domaine  $(x, y) \in A(t)\Omega$  où  $\omega = 0.5$  en quelques temps différents:  $t = 0$ ,  $t = \pi/4$  et  $t = \pi/2$ .

## 7.3 Méthode de décomposition de domaines

Nous avons présentés les méthodes de décomposition en espace pour l'équation de Schrödinger non-linéaire dans le chapitre 5. Formellement, la seule différence entre l'équation (7.3) et l'équation de Schrödinger (5.1) que nous avons étudiée est la constante devant l'opérateur  $\Delta$ . Dans ce chapitre, nous appliquons ainsi directement les techniques à (7.3). Le domaine de calcul complet est  $]0, T[ \times \Omega$ . Il est décomposé en  $N$  sous-domaines verticaux comme ceux montrés dans la figure 4.1, et donc sans points de croisement. La discrétisation et les algorithmes sans et avec préconditionneur sont présentés en détail dans la sous-section 5.1.

Nous utilisons deux conditions de transmission:

- Robin:  $S_j = -ip$ ,  $p \in \mathbb{R}^+$ ,
- $S_{2p}^m$ :  $S_j = -i\sqrt{i\partial_t + \frac{1}{2}\Delta_{\Gamma_j} - V_t - \beta} \cdot |^2$ ,

où le paramètre  $p$  optimal est déterminé via les tests avec des  $p$  différents.

## 7.4 Résultats numériques

Dans cette sous-section, nous montrons d'abord que la méthode de décomposition en espace donne une solution en haute précision. L'erreur entre les solutions obtenus par les méthodes sans et avec décomposition de domaine est estimé par

$$Err(t) := \left( \sum_{j=1}^N \|\tilde{u}_j(t, x, y) - \tilde{u}|_{\Omega_j}(t, x, y)\|_{l^2}^2 \right)^{1/2},$$

où  $\tilde{u}_j$ ,  $j = 0, 1, \dots, N$  est la solution calculée sur  $\Omega_j$  et  $\tilde{u}|_{\Omega_j}$  est la restriction de  $\tilde{u}$  à  $\Omega_j$ . Ensuite nous comparons les méthodes sans et avec préconditionneur. À la fin, nous montrons les simulations dynamique des réseaux de vortex quantifiés.

### 7.4.1 Comparaison des méthodes

Le domaine complet est  $\Omega = ]-16, 16[ \times ]-16, 16[ - \Delta y$ . Nous utilisons une gaussienne comme donnée initiale (voir figure 7.2a):

$$u_0(x, y) = \frac{1}{\pi^{1/4}} e^{-\frac{(x^2+2y^2)}{2}}, \quad (x, y) \in \mathbb{R}^2.$$

La vitesse angulaire est  $\omega = 0.4$ . Le coefficient  $\beta$  est 10.15.

Nous fixons l'étape du temps à  $\Delta t = 0.0001$ . Le premier maillage que nous utilisons est  $\Delta x = \Delta y = 1/32$ . Sur  $\Omega$ , cela génère des inconnues de taille  $1024 \times 1024$ . On utilise toujours la méthode directe pour traiter le système linéaire qui vient de la méthode des éléments finis  $Q_1$ . Pour ce cas test, il est possible de résoudre l'équation (7.3) numériquement sur le domaine complet sous la limitation de 32Go de mémoire. Cependant, le temps de calcul est très long pour les comparaisons. Ainsi nous utilisons un petit temps final  $t = 0.1$ . La solution en temps  $t = 0.1$  est montrée dans la figure 7.2b.

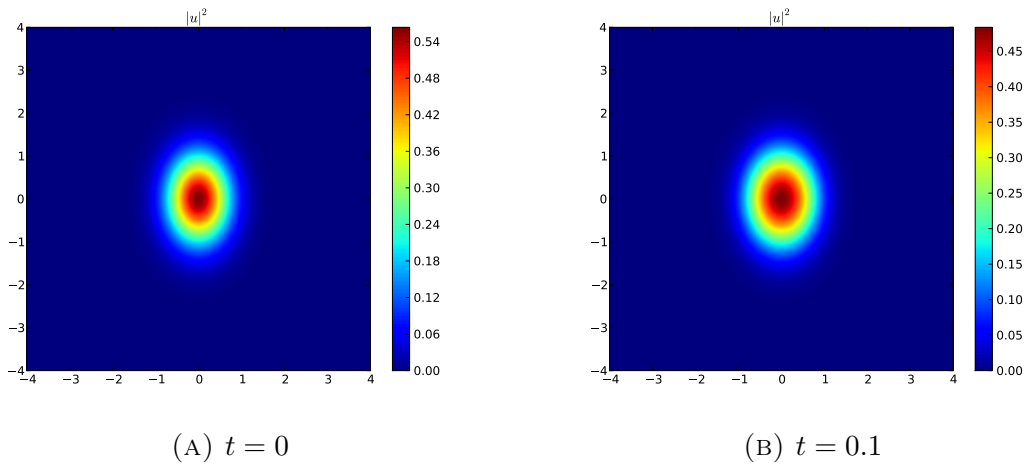


FIGURE 7.2: La solution initiale et la solution en  $t = 0.1$  affichées dans le domaine  $] -4, 4[ \times ] -4, 4[$ .

Nous présentons dans le tableau 7.1 l'erreur  $Err(0.1)$ . La précision pour la condition de transmission Robin est très bonnes. Même si les résultats semblent meilleurs pour l'algorithme avec préconditionneur, nous ne pouvons cependant pas conclure ici que qu'il donne une solution plus précise. En effet, les conditions d'arrêt des algorithmes sans et avec préconditionneur sont un peu différentes

$$\| g^{k+1} - g^k \| = \begin{cases} \| (I - \mathcal{R}_{nl})g^k \|, \\ \| P^{-1}(I - \mathcal{R}_{nl})g^k \| . \end{cases}$$

$N$		2	4	8	16	32
Robin	nPd.	$5.7 \times 10^{-11}$	$5.6 \times 10^{-11}$	$5.7 \times 10^{-11}$	$5.7 \times 10^{-11}$	$6.0 \times 10^{-11}$
	Pd.	$7.8 \times 10^{-12}$	$7.9 \times 10^{-12}$	$7.9 \times 10^{-12}$	$8.0 \times 10^{-12}$	$8.6 \times 10^{-12}$
$S_{2p}^m$	nPd.	—	$9.9 \times 10^{-7}$	$9.9 \times 10^{-7}$	$9.9 \times 10^{-7}$	$2.1 \times 10^{-8}$
	Pd.	$9.9 \times 10^{-7}$	$9.9 \times 10^{-7}$	$9.9 \times 10^{-7}$	$9.9 \times 10^{-7}$	$2.1 \times 10^{-8}$

—: excède la limitation du temps de calcul maximal (20 heures).

TABLEAU 7.1:  $Err(t)$ ,  $t = 0.1$ ,  $\Delta t = 0.0001$ ,  $\Delta x = \Delta y = 1/32$ .

Les temps de calcul sont montrés dans le tableau 7.2.  $T^{\text{ref}}$  désigne le temps de calcul sur un processeur de la résolution numérique de l'équation sur le domaine complet  $(0, 0.1) \times \Omega$  complet. Comme la condition aux limites sur  $\Omega$  est associé à la condition de transmission, le temps de référence  $T^{\text{ref}}$  pour Robin est différent de celui pour  $S_{2p}^m$ . On peut voir que l'implémentation de la condition de transmission  $S_{2p}^m$  est plus coûteuse que celle de la condition de transmission Robin. Le potentiel  $V_t$  dans ce chapitre est dépendant du temps, ainsi à chaque étape du temps, il faut construire les matrices associés à la méthode des éléments finis. Cette construction pour  $S_{2p}^m$  est beaucoup plus coûteuse que pour Robin. De plus, nous utilisons la méthode directe LU, la factorisation LU pour  $S_{2p}^m$  est également plus coûteuse que Robin. C'est également pour cette raison que le préconditionneur permet de diminuer le temps de calcul dans le cas où le nombre de points en  $y$  est presque égal à celui en  $x$ .

Nous pouvons voir que les algorithmes sans ( $T_{\text{nopc}}$ ) et avec ( $T_{\text{nopc}}$ ) préconditionneur sont scalables. Dans ce cas test, le préconditionneur permet de diminuer le temps de calcul.

$N$		2	4	8	16	32
Robin, $p = 180$	$T^{\text{ref}}$	5.68				
	$T_{\text{nopc}}$	5.68	2.66	1.28	0.68	0.33
	$T_{\text{pc}}$	3.49	1.60	0.77	0.44	0.24
$S_{2p}^m$ , $m = 76$	$T^{\text{ref}}$	8.41				
	$T_{\text{nopc}}$	> 20	10.70	7.40	5.07	4.23
	$T_{\text{pc}}$	6.30	3.52	2.30	1.68	1.37

TABLEAU 7.2: Temps de calcul en heure,  $T = 0.1$ ,  $\Delta t = 0.0001$ ,  $\Delta x = \Delta y = 1/32$ .

Nous faisons également des tests avec un maillage plus fin  $\Delta x = 1/1024$ ,  $\Delta y = 1/64$ . Le domaine complet  $\Omega$  est décomposé en  $N = 128, 256, 512, 1024$  sous-domaines. Les temps de calcul sont présentés dans le tableau 7.3. Nous pouvons constater que les algorithmes sans et avec préconditionneur sont tous scalables. Le préconditionneur permet de diminuer le temps de calcul total pour réaliser une simulation complète. En revanche, comme l'application du préconditionneur consomme de la mémoire, celle-ci n'est pas suffisante dans le cas  $N = 128$ .

$N$	128	256	512	1024
$T_{\text{nopc}}$ , $p = 95$	19.3	8.8	5.0	2.1
$T_{\text{pc}}$ , $p = 95$	*	2.3	1.6	0.8

\*: le mémoire n'est pas suffisamment.

TABLEAU 7.3: Temps de calcul en heure,  $T = 0.1$ ,  $\Delta t = 0.0001$ ,  $\Delta x = 1/1024$ ,  $\Delta y = 1/64$ .

## 7.4.2 Dynamique des réseaux de vortex quantifiés

Nous appliquons notre méthode numérique parallèle pour étudier la dynamique des réseaux de vortex quantifiés pour des BEC avec rotation. Dans cette simulation, le potentiel et les paramètres de non linéarités et de rotation sont les suivants

$$V(x, y) = \frac{1}{2}(x^2 + y^2), \quad \beta = 1000, \quad \omega = 0.9.$$

La donnée initiale  $u_0$  est un réseau de vortex stationnaire [10, 13]. La solution stationnaire de (7.1)  $\phi$  est définie par

$$u(t, x, y) = \phi(x, y)e^{-i\mu t}, \quad (7.5)$$

où  $\mu$  est le potentiel chimique de condensation. En substituant (7.5) dans (7.1), on a

$$\mu\phi = -\frac{1}{2}\phi + V\phi + \beta|\phi|^2\phi - \omega L_z\phi,$$

avec la contrainte de normalisation

$$\|\phi\|_2^2 = \int_{\mathbb{R}^2} |\phi(x, y)|^2 dx dy = 1.$$

Ceci est donc un problème de valeurs propres à une contrainte non-linéaire . Les valeurs propres  $\mu$  peuvent être calculées à partir de sa fonction propre correspondant  $\phi$  par

$$\mu_{\beta, \omega}(\phi) = E_{\beta, \omega}(\phi) + \frac{\beta}{4} \int_{\mathbb{R}^2} |\phi(x, y)|^4 dx dy,$$

où

$$E_{\beta, \omega}(\phi) = \frac{1}{2} \int_{\mathbb{R}^2} (|\nabla \phi|^2 + V|\phi|^2 + \beta|\phi|^4 - \omega \bar{\phi} L_z \phi) dx dy. \quad (7.6)$$

L'état fondamental d'un BEC est défini comme la solution du problème de minimisation, noté par  $\phi_g$ ,

$$E_{\beta, \omega}(\phi_g) = \min_{\phi \in S} E_{\beta, \omega}(\phi),$$

où  $S = \{\phi \mid \|\phi\|_2 = 1, E_{\beta, \omega} < \infty\}$ .

Pour notre simulation, nous prenons donc comme donnée initiale

$$u_0(x, y) = \phi_g(x, y). \quad (7.7)$$

Celle-ci est calculé par la méthode BESP (Backward Euler sine pseudospectra) [15] en utilisant la librairie GPELab [10], boîte à outils Matlab qui est développé pour le calcul des états fondamentaux ou la dynamique des systèmes quantiques modélisées par des équations de Gross-Pitaevskii.

Le domaine complet  $\Omega = ]-16, 16[ \times ]-16, 16 - \Delta y[$  est décomposé en  $N = 32$  sous-domaines. Nous fixons le pas de temps à  $\Delta t = 0.0001$ . L'utilisation de GPELab pour générer la donnée initiale pose cependant un problème. En effet, comme GPELab est basé sur un code Matlab séquentiel, le maillage ne peut être trop fin et nous utilisons ici  $\Delta x = \Delta y = 1/32$ . Ainsi la taille du problème n'est pas suffisamment grande pour tester un très grand nombre de sous-domaines (ex.  $N = 128$ ). Le paramètre  $p$  ici est  $p = 180$ . La figure 7.3 montre les tracés des contour de la solution  $|u(t, x, y)|^2$  à quelques temps différents, illustrés dans la zone valide  $] -16/\sqrt{2}, 16/\sqrt{2}[ \times ] -16/\sqrt{2}, (16 - \Delta y)/\sqrt{2}[$ . Le temps total pour réaliser cette simulation complète est d'environ 16 heures.

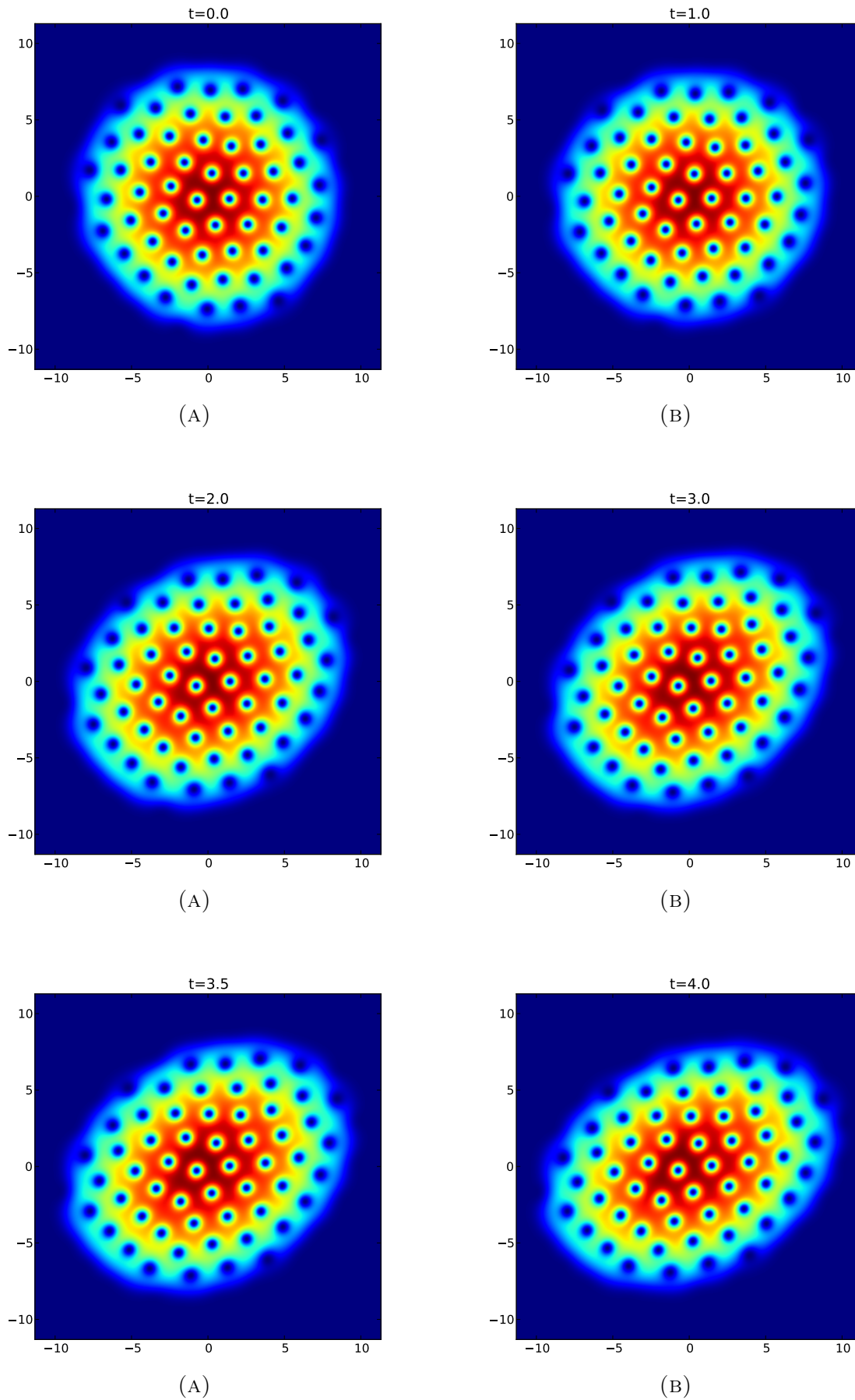


FIGURE 7.5: Tracé de contours de la solutions  $|u(t, x, y)|^2$  en quelques temps différents.



## 7.5 Conclusion

Nous appliquons les méthodes de décomposition en espace sans et avec préconditionneur (voir chapitre 5) à la simulation de BEC avec rotation en deux dimensions dans ce chapitre. Nous montrons d'abord par les tests numériques que les solutions obtenues par les algorithmes parallèles ont une haute précision. Nos tests montrent que nos algorithmes permettent de diminuer le temps de calcul comparé au calcul sur un processeur sans utilisant les méthodes de décomposition de domaine. Les algorithmes sont aussi scalables. Finalement, nous présentons des simulations de la dynamique des réseaux de vortex quantifiés pour BEC avec rotation.



# Conclusion et perspective

## Conclusion

Dans cette thèse, nous avons étudié principalement les méthodes de décomposition de domaines pour l'équation de Schrödinger linéaire et non-linéaire en une dimension et en deux dimensions. Celles-ci nous permettent de réaliser des algorithmes parallèles qui s'adaptent à l'architecture des grands calculateurs modernes. Avec les méthodes de décomposition de domaines, nous pouvons résoudre l'équation de Schrödinger numériquement dans un domaine d'espace grand et diminuer le temps de calcul total. Il y a deux points-clés pour ces méthodes : la condition de transmission et la scalabilité.

Dans le premier chapitre, nous avons considéré la méthode de relaxation d'onde de Schwarz (SWR) pour l'équation de Schrödinger linéaire en une dimension avec un potentiel indépendant du temps. Nous avons proposé un nouvel algorithme qui permet une forte réduction du temps de calcul par rapport à l'algorithme classique. Ce nouvel algorithme est scalable (scalabilité forte) jusqu'à 500 sous-domaines (500 processus MPI) dans nos tests. Il permet également de diminuer beaucoup le temps de calcul de la résolution numérique de l'équation de Schrödinger sur le domaine complet. Nous avons écrit la méthode d'une manière globale en temps ce qui permet de simplifier l'écriture et d'utiliser des méthodes Krylov au lieu de la méthode de point fixe. Nous avons aussi généralisé les travaux de Halpern et Szeftel sur la condition de transmission. Nous utilisons trois types de conditions absorbantes construites récemment par Antoine, Besse et Klein comme la condition de transmission. Les trois types de conditions de transmission sont la stratégie potentielle (ordre 2, 3, 4), la stratégie de changement de gauge (ordre 2, 4) et la stratégie d'approximation de Padé (ordre 2, 4 et le paramètre de l'ordre d'approximation de Padé  $m$ ). Les nouvelles conditions de transmission et la condition de transmission de Robin sont comparées numériquement dans le cadre du nouvel algorithme.

Le deuxième chapitre est consacré à la généralisation à l'équation de Schrödinger avec un potentiel général (linéaire dépendant du temps ou non-linéaire). Nous avons utilisé l'opérateur linéaire préalablement défini dans le premier chapitre pour l'équation de Schrödinger sans potentiel comme un préconditionneur. Il permet de diminuer beaucoup le nombre d'itérations requis et le temps de calcul total. Dans nos tests, l'algorithme avec préconditionneur est scalable jusqu'à 500 sous-domaines. Par ailleurs, nous avons comparé la condition de transmission de Robin et les nouvelles conditions de transmission numériquement.

Nous nous sommes ensuite intéressés à l'équation de Schrödinger en deux dimensions dans les chapitres 4, 5 et 6.

La méthode SWR et la méthode de décomposition en espace ont été utilisées dans le chapitre 4. D'après nos tests, la méthode de décomposition en espace utilise moins de temps de calcul. Nous avons aussi généralisé le nouvel algorithme en deux dimensions pour l'équation de Schrödinger sans potentiel. Le nouvel algorithme diminue également le temps de calcul sous la condition de quelques restrictions sur maillage. Nous avons également étudié les conditions de transmission numériquement (Robin et la stratégie d'approximation de Padé). La stratégie d'approximation de Padé est bien meilleure que celle de Robin par rapport au nombre d'itérations requises. Cette différence se gomme si les méthodes Krylov sont utilisées sur le problème d'interface en lieu et place de la méthode de point fixe.

Nous avons considéré la méthode de décomposition en espace et l'algorithme avec préconditionneur dans le chapitre 5 pour l'équation de Schrödinger avec un potentiel non nul. Le préconditionneur diminue beaucoup le nombre d'itérations requises. Le temps de calcul est aussi moindre si le maillage satisfait quelques restrictions.

Dans le chapitre 6, nous avons généralisé une méthode de Schwarz optimisée avec points de croisement proposée par Loisel pour l'équation de Laplace à l'équation de Schrödinger qui conduit à une méthode SWR avec points de croisement. Nous avons montré par des résultats numériques que cette méthode fonctionne de manière convenable. Nous avons également comparé les conditions de transmission utilisées (Robin et la stratégie d'approximation de Padé).

Dans le dernier chapitre, nous avons appliqué les méthodes étudiées dans les chapitres précédents à la simulation du condensat de Bose-Einstein (Équation Gross-Pitaevskii avec rotation). Nous avons réalisé des simulations avec les algorithmes parallèles qui permettent de diminuer le temps de calcul total.

Concernant la partie purement informatique, nous avons porté les codes en Cpu à l'accélérateur Gpu (carte graphique) pour l'équation de Schrödinger en une dimension. Nous avons constaté que les temps de calcul avec accélération Gpu sont fortement réduits.

## Perspective

Nous n'avons pas considéré ici la méthode de décomposition en espace pour l'équation de Schrödinger en une dimension. Selon les résultats en 2D, cette méthode pourrait diminuer le nombre d'itérations requis, et donc ainsi le temps de calcul total. Un autre travail futur est de bien comparer les performances sans et avec accélération Gpu. Par exemple, en Cpu, une façon d'accélération standard est d'utiliser OpenMP (Open Multi-Processing) [65]. Nous pourrions donc comparer les approches Gpu et OpenMP.

Pour l'équation de Schrödinger en deux dimensions, une perspective est d'optimiser les codes du nouvel algorithme et de l'algorithme avec préconditionneur. Nous voyons dans le chapitre 4 que la matrice  $\mathcal{L}$  pour  $V = 0$  est composée de blocs identiques sous quelques hypothèses. Dans les codes actuels, chaque processus MPI a une copie des blocs. Nous pensons qu'il pourrait être possible d'éviter ce problème en changeant la façon du stockage de la matrice. Cette optimisation pourrait diminuer le temps de calcul et le mémoire consommée. Il est également possible de porter les codes sur des accélérateurs Gpu. Cependant, comme il

n'existe pas de bibliothèques de résolution par méthode directe pour une matrice creuse générale sur Gpu, le changement de solveur local est nécessaire.

Dans le chapitre 6, nous ne proposons qu'une méthode qui fonctionne pour les points de croisement. La méthode de décomposition de domaines avec points de croisement est un problème difficile et intéressant en mathématiques. Nous pensons qu'il pourrait être possible de généraliser le nouvel algorithme à ce problème.

Nous souhaitons également pousser nos investigations concernant la simulation des condensats de Bose-Einstein par exemple en utilisant l'algorithme avec points de croisement.



# Annexe A

## Forme discrète du problème, une dimension

Nous présentons dans cette annexe quelques détails liés au chapitre 2.

1. La forme discrète du problème pour les conditions de transmission présentées dans la sous-section 2.1.2. Elle est donnée par (A.1) où la définition de l'opérateur  $\tilde{S}$  et des quantités scalaires  $B_{j,n}^l$ ,  $B_{j,n}^r$ ,  $b_{j,n}^l$ ,  $b_{j,n}^r$  dépendent de la condition de transmission.
2. L'expression de  $\mathbf{v}_{j,n}^k$  pour les conditions de transmission (celle pour  $S_0^2$  est donnée par (2.21)).

$$\begin{cases} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}(Q_{j,r}\mathbf{v}_{j,n}^k), & j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}(Q_{j,l}\mathbf{v}_{j,n}^k), & j = 2, 3, \dots, N, \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_{j,n}^k, r_{j,n}^k)^T, \end{cases} \quad (\text{A.1})$$

où

$$\mathbb{B}_{j,n} = \begin{pmatrix} B_{j,n}^l & 0 & \dots & \dots & 0 \\ 0 & 0 & & & 0 \\ \vdots & & \ddots & & \vdots \\ \vdots & & & 0 & 0 \\ 0 & 0 & \dots & 0 & B_{j,n}^r \end{pmatrix}, \quad \mathbf{b}_{j,n}^k = \begin{pmatrix} b_{j,n}^{k,l} \\ 0 \\ \vdots \\ 0 \\ b_{j,n}^{k,r} \end{pmatrix}.$$

### A.1 Condition de transmission Robin

- **Forme discrète du problème**

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_p(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_p(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_p(Q_{j,l}\mathbf{v}_{j,n}^k) = -ip \cdot (Q_{j,l}\mathbf{v}_{j,n}^k), \\ \tilde{S}_p(Q_{j,r}\mathbf{v}_{j,n}^k) = -ip \cdot (Q_{j,r}\mathbf{v}_{j,n}^k), \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_{j,n}^k, r_{j,n}^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = -ip, \quad j = 2, 3, \dots, N, \\ B_{j,n}^r = -ip, \quad j = 1, 2, \dots, N-1, \\ b_{j,n}^{k,l} = 0, \quad j = 1, 2, \dots, N, \\ b_{j,n}^{k,r} = 0, \quad j = 1, 2, \dots, N. \end{array} \right.$$

• Une expression de  $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}Q_j^T(l_{j,n}^k, r_{j,n}^k)^T + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j \right) \mathbf{v}_{j,q}^k \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j \right) \mathbf{u}_{j,0}. \end{aligned}$$



## A.2 Condition de transmission $S_0^2$

### • Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_0^2(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_0^2(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_0^2(Q_{j,l}\mathbf{v}_{j,n}^k) = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k), \\ \tilde{S}_0^2(Q_{j,r}\mathbf{v}_{j,n}^k) = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k), \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_{j,n}^k, r_{j,n}^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 2, 3, \dots, N, \\ B_{j,n}^r = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 1, 2, \dots, N-1, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k), \quad j = 2, 3, \dots, N, \\ b_{j,n}^{k,r} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k), \quad j = 1, 2, \dots, N-1. \end{array} \right.$$

### • Une expression de $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T(l_{j,n}^k, r_{j,n}^k)^T \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_{n-q} Q_j^T Q_j \right) \mathbf{v}_{j,q}^k \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_n Q_j^T Q_j \right) \mathbf{u}_{j,0}. \end{aligned}$$

### A.3 Condition de transmission $S_0^3$

- Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_0^3(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_0^3(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_0^3(Q_{j,l}\mathbf{v}_{j,n}^k) = \tilde{S}_0^2(Q_{j,l}\mathbf{v}_{j,n}^k) - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(a_j)}{2} \sum_{s=0}^n \alpha_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k), \\ \tilde{S}_0^3(Q_{j,r}\mathbf{v}_{j,n}^k) = \tilde{S}_0^2(Q_{j,r}\mathbf{v}_{j,n}^k) - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(b_j)}{2} \sum_{s=0}^n \alpha_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k), \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_n^k, r_n^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0 - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(a_j)}{2} \alpha_0, \quad j = 2, 3, \dots, N, \\ B_{j,n}^r = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0 - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(b_j)}{2} \alpha_0, \quad j = 1, 2, \dots, N-1, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k) \\ \quad - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(a_j)}{2} \sum_{s=0}^{n-1} \alpha_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k), \quad j = 2, 3, \dots, N, \\ b_{j,n}^{k,r} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^{n-1} \beta_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k) \\ \quad - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(b_j)}{2} \sum_{s=0}^{n-1} \alpha_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k), \quad j = 1, 2, \dots, N-1. \end{array} \right.$$

- Une expression de  $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T(l_{j,n}^k, r_{j,n}^k)^T \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_{n-q} Q_j^T Q_j \right. \\ &\quad \left. - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(a_j)}{2} \alpha_{n-q} Q_{j,l}^T Q_{j,l} - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(b_j)}{2} \alpha_{n-q} Q_{j,r}^T Q_{j,r} \right) \mathbf{v}_{j,q}^k \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_n Q_j^T Q_j \right. \\ &\quad \left. - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(a_j)}{2} \alpha_n Q_{j,l}^T Q_{j,l} - e^{\frac{ipi}{4}} \sqrt{\frac{\Delta t}{2}} \frac{W_n(b_j)}{2} \alpha_n Q_{j,r}^T Q_{j,r} \right) \mathbf{u}_{j,0}. \end{aligned}$$

## A.4 Condition de transmission $S_0^4$

### • Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_0^3(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_0^3(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_n^k, r_n^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_0 - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(a_j)}{2}\alpha_0 + i\frac{\Delta t}{2}\frac{\partial_x W_n(a_j)}{4}\gamma_0, \quad j = 2, 3, \dots, N, \\ B_{j,n}^r = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_0 - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(b_j)}{2}\alpha_0 - i\frac{\Delta t}{2}\frac{\partial_x W_n(b_j)}{4}\gamma_0, \quad j = 1, 2, \dots, N-1, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\sum_{s=0}^{n-1}\beta_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k) - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(a_j)}{2}\sum_{s=0}^{n-1}\alpha_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k) \\ \quad + i\frac{\Delta t}{2}\frac{\partial_x W_n(a_j)}{4}\sum_{s=0}^{n-1}\gamma_{n-s}(Q_{j,l}\mathbf{v}_{j,s}^k), \quad j = 2, 3, \dots, N, \\ b_{j,n}^{k,r} = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\sum_{s=0}^{n-1}\beta_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k) - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(b_j)}{2}\sum_{s=0}^{n-1}\alpha_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k) \\ \quad - i\frac{\Delta t}{2}\frac{\partial_x W_n(b_j)}{4}\sum_{s=0}^{n-1}\gamma_{n-s}(Q_{j,r}\mathbf{v}_{j,s}^k), \quad j = 1, 2, \dots, N-1. \end{array} \right.$$

### • Une expression de $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}Q_j^T(l_{j,n}^k, r_{j,n}^k)^T \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}\sum_{q=1}^{n-1}\left((-1)^{n-1-q}\frac{4i}{\Delta t}\mathbb{M}_j + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_{n-q}Q_j^T Q_j\right. \\ &\quad - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(a_j)}{2}\alpha_{n-q}Q_{j,l}^T Q_{j,l} - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(b_j)}{2}\alpha_{n-q}Q_{j,r}^T Q_{j,r} \\ &\quad \left. + i\frac{\Delta t}{2}\frac{\partial_x W_n(a_j)}{4}\gamma_{n-q}Q_{j,l}^T Q_{j,l} - i\frac{\Delta t}{2}\frac{\partial_x W_n(b_j)}{4}\gamma_{n-q}Q_{j,r}^T Q_{j,r}\right)\mathbf{v}_{j,q}^k \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}\left(\frac{2i}{\Delta t}(-1)^{n-1}\mathbb{M}_j + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_n Q_j^T Q_j\right. \\ &\quad - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(a_j)}{2}\alpha_n Q_{j,l}^T Q_{j,l} - e^{\frac{ipi}{4}}\sqrt{\frac{\Delta t}{2}}\frac{W_n(b_j)}{2}\alpha_n Q_{j,r}^T Q_{j,r} \\ &\quad \left. + i\frac{\Delta t}{2}\frac{\partial_x W_n(a_j)}{4}\gamma_n Q_{j,l}^T Q_{j,l} - i\frac{\Delta t}{2}\frac{\partial_x W_n(b_j)}{4}\gamma_n Q_{j,r}^T Q_{j,r}\right)\mathbf{u}_{j,0}. \end{aligned}$$

## A.5 Condition de transmission $S_1^2$

### • Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_1^2(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_1^2(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_1^2(Q_{j,l}\mathbf{v}_{j,n}^k) = e^{-\frac{i\pi}{4}} e^{i\mathcal{W}_n(a_j)} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} e^{-i\mathcal{W}_s(a_j)} (Q_{j,l}\mathbf{v}_{j,s}^k), \\ \tilde{S}_1^2(Q_{j,r}\mathbf{v}_{j,n}^k) = e^{-\frac{i\pi}{4}} e^{i\mathcal{W}_n(b_j)} \sqrt{\frac{2}{\Delta t}} \sum_{s=0}^n \beta_{n-s} e^{-i\mathcal{W}_s(b_j)} (Q_{j,r}\mathbf{v}_{j,s}^k), \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T (l_n^k, r_n^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 1, 2, \dots, N-1, \\ B_{j,n}^r = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} \beta_0, \quad j = 2, 3, \dots, N, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(a_j)} \sum_{s=0}^{n-1} \beta_{n-s} e^{-i\mathcal{W}_s(a_j)} (Q_{j,l}\mathbf{v}_{j,s}^k), \quad j = 1, 2, \dots, N-1, \\ b_{j,n}^{k,r} = e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(b_j)} \sum_{s=0}^{n-1} \beta_{n-s} e^{-i\mathcal{W}_s(b_j)} (Q_{j,r}\mathbf{v}_{j,s}^k), \quad j = 2, 3, \dots, N. \end{array} \right.$$

### • Une expression de $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T (l_{j,n}^k, r_{j,n}^k)^T \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(a_j) - i\mathcal{W}_q(a_j)} \beta_{n-q} Q_{j,l}^T Q_{j,l} \right. \\ &\quad \left. + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(b_j) - i\mathcal{W}_q(b_j)} \beta_{n-q} Q_{j,r}^T Q_{j,r} \right) \mathbf{v}_{j,q}^k \\ &+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(a_j)} \beta_n Q_{j,l}^T Q_{j,l} \right. \\ &\quad \left. + e^{-\frac{i\pi}{4}} \sqrt{\frac{2}{\Delta t}} e^{i\mathcal{W}_n(b_j)} \beta_n Q_{j,r}^T Q_{j,r} \right) \mathbf{u}_{j,0}. \end{aligned}$$

## A.6 Condition de transmission $S_1^4$

### • Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_1^2(Q_{j,r}\mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_1^2(Q_{j,l}\mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})\mathbf{v}_{j,n}^k = \frac{2i}{\Delta t}\mathbb{M}_j\mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T(l_n^k, r_n^k)^T, \\ B_{1,n}^l = B_{N,n}^r = 0, \\ B_{j,n}^l = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_0 + i\operatorname{sgn}(\partial_x W_n(a_j))\frac{|\partial_x W_n(a_j)|}{4}\frac{\Delta t}{2}\gamma_0, \quad j = 1, 2, \dots, N-1, \\ B_{j,n}^r = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}\beta_0 - i\operatorname{sgn}(\partial_x W_n(b_j))\frac{|\partial_x W_n(b_j)|}{4}\frac{\Delta t}{2}\gamma_0, \quad j = 2, 3, \dots, N, \\ b_{1,n}^{k,l} = b_{N,n}^{k,r} = 0, \\ b_{j,n}^{k,l} = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(a_j)}\sum_{s=0}^{n-1}\beta_{n-s}e^{-i\mathcal{W}_s(a_j)}(Q_{j,l}\mathbf{v}_{j,s}^k) \\ \quad + i\operatorname{sgn}(\partial_x W_n(a_j))\frac{\sqrt{|\partial_x W_n(a_j)|}}{2}e^{i\mathcal{W}_n(a_j)}\frac{\Delta t}{2}\sum_{s=0}^{n-1}\gamma_{n-s}\frac{\sqrt{|\partial_x W_s(a_j)|}}{2}e^{-i\mathcal{W}_s(a_j)}(Q_{j,l}\mathbf{v}_{j,s}^k), \\ b_{j,n}^{k,r} = e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(b_j)}\sum_{s=0}^{n-1}\beta_{n-s}e^{-i\mathcal{W}_s(b_j)}(Q_{j,r}\mathbf{v}_{j,s}^k) \\ \quad - i\operatorname{sgn}(\partial_x W_n(b_j))\frac{\sqrt{|\partial_x W_n(b_j)|}}{2}e^{i\mathcal{W}_n(b_j)}\frac{\Delta t}{2}\sum_{s=0}^{n-1}\gamma_{n-s}\frac{\sqrt{|\partial_x W_s(b_j)|}}{2}e^{-i\mathcal{W}_s(b_j)}(Q_{j,r}\mathbf{v}_{j,s}^k). \end{array} \right.$$

### • Une expression de $\mathbf{v}_{j,n}^k$

$$\begin{aligned} \mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}Q_j^T(l_{j,n}^k, r_{j,n}^k)^T \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}\sum_{q=1}^{n-1}\left((-1)^{n-1-q}\frac{4i}{\Delta t}\mathbb{M}_j\right. \\ &\quad + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(a_j)-i\mathcal{W}_q(a_j)}\beta_{n-q}Q_{j,l}^T Q_{j,l} + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(b_j)-i\mathcal{W}_q(b_j)}\beta_{n-q}Q_{j,r}^T Q_{j,r} \\ &\quad + i\operatorname{sgn}(\partial_x W_n(a_j))\frac{\sqrt{|\partial_x W_n(a_j)|}}{2}e^{i\mathcal{W}_n(a_j)-i\mathcal{W}_q(a_j)}\frac{\Delta t}{2}\gamma_{n-q}\frac{\sqrt{|\partial_x W_q(a_j)|}}{2}Q_{j,l}^T Q_{j,l} \\ &\quad \left. - i\operatorname{sgn}(\partial_x W_n(b_j))\frac{\sqrt{|\partial_x W_n(b_j)|}}{2}e^{i\mathcal{W}_n(b_j)-i\mathcal{W}_q(b_j)}\frac{\Delta t}{2}\gamma_{n-q}\frac{\sqrt{|\partial_x W_q(b_j)|}}{2}Q_{j,r}^T Q_{j,r}\right)\mathbf{v}_{j,q}^k \\ &\quad + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1}\left(\frac{2i}{\Delta t}(-1)^{n-1}\mathbb{M}_j\right. \\ &\quad + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(a_j)}\beta_n Q_{j,l}^T Q_{j,l} + e^{-\frac{i\pi}{4}}\sqrt{\frac{2}{\Delta t}}e^{i\mathcal{W}_n(b_j)}\beta_n Q_{j,r}^T Q_{j,r} \\ &\quad \left. + i\operatorname{sgn}(\partial_x W_n(a_j))\frac{\sqrt{|\partial_x W_n(a_j)|}}{2}e^{i\mathcal{W}_n(a_j)}\frac{\Delta t}{2}\gamma_n\frac{\sqrt{|\partial_x W_0(a_j)|}}{2}Q_{j,l}^T Q_{j,l}\right) \end{aligned}$$

$$-i \operatorname{sgn}(\partial_x W_n(b_j)) \frac{\sqrt{|\partial_x W_n(b_j)|}}{2} e^{iW_n(b_j)} \frac{\Delta t}{2} \gamma_n \frac{\sqrt{|\partial_x W_0(b_j)|}}{2} Q_{j,r}^T Q_{j,r} \mathbf{u}_{j,0}.$$

## A.7 Condition de transmission $S_2^2$

### • Forme discrète du problème

$$\left\{ \begin{array}{l} l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_2^2 \cdot (Q_{j,r} \mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\ r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_2^2 \cdot (Q_{j,l} \mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\ \tilde{S}_2^2 \cdot (Q_{j,l} \mathbf{v}_{j,n}^k) = -i \left( \sum_{s=0}^m a_s^m \right) \cdot (Q_{j,l} \mathbf{v}_{j,n}^k) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot (Q_{j,l} \mathbf{v}_{j,n}^k) \\ \quad + i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \varphi_{j,n-1}^s, \\ \tilde{S}_2^2 \cdot (Q_{j,r} \mathbf{v}_{j,n}^k) = -i \left( \sum_{s=0}^m a_s^m \right) \cdot (Q_{j,r} \mathbf{v}_{j,n}^k) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot (Q_{j,r} \mathbf{v}_{j,n}^k) \\ \quad + i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \varphi_{j,n-1}^s, \\ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n}) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T (l_n^k, r_n^k)^T, \\ B_{j,n}^l = -i \left( \sum_{s=0}^m a_s^m \right) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m}, \\ B_{j,n}^r = -i \left( \sum_{s=0}^m a_s^m \right) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m}, \\ b_{j,n}^{k,l} = i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \varphi_{j,n-1}^s, \\ b_{j,n}^{k,r} = i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \varphi_{j,n-1}^s, \\ \text{où pour } s = 1, 2, \dots, m, \\ Q_{j,l} \varphi_{j,n-1/2}^s = \frac{1}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \mathbf{v}_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \varphi_{j,n-1}^s, \\ Q_{j,r} \varphi_{j,n-1/2}^s = \frac{1}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \mathbf{v}_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \varphi_{j,n-1}^s, \\ Q_{j,l} \varphi_{j,n}^s = 2Q_{j,l} \varphi_{j,n-1/2}^s - Q_{j,l} \varphi_{j,n-1}^s, \\ Q_{j,r} \varphi_{j,n}^s = 2Q_{j,r} \varphi_{j,n-1/2}^s - Q_{j,r} \varphi_{j,n-1}^s, \\ Q_{j,l} \varphi_{j,0}^s = Q_{j,r} \varphi_{j,0}^s = 0. \end{array} \right.$$

• Une expression de  $\mathbf{v}_{j,n}^k$

$$\begin{aligned}
\mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T (l_{j,n}^k, r_{j,n}^k)^T + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,q}^k \\
&+ i(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \frac{2}{\frac{2i}{\Delta t} + W_q(a_j) + d_s^m} \right. \\
&\quad \left. \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(a_j) + d_s^m} - 1 \right) Q_{j,l}^T Q_{j,l} \right) \mathbf{v}_{j,q}^k \\
&+ i(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \frac{2}{\frac{2i}{\Delta t} + W_q(b_j) + d_s^m} \right. \\
&\quad \left. \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(b_j) + d_s^m} - 1 \right) Q_{j,r}^T Q_{j,r} \right) \mathbf{v}_{j,q}^k \\
&+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j \right) \mathbf{u}_{j,0}
\end{aligned}$$

## A.8 Condition de transmission $S_2^4$

### • Forme discrète du problème

$$\left\{ \begin{array}{l}
 l_{j+1,n}^{k+1} = -r_{j,n}^k + 2\tilde{S}_2^2 \cdot (Q_{j,r} \mathbf{v}_{j,n}^k), \quad j = 1, 2, \dots, N-1, \\
 r_{j-1,n}^{k+1} = -l_{j,n}^k + 2\tilde{S}_2^2 \cdot (Q_{j,l} \mathbf{v}_{j,n}^k), \quad j = 2, 3, \dots, N, \\
 (\mathbb{A}_{j,n} - \mathbb{B}_{j,n}) \mathbf{v}_{j,n}^k = \frac{2i}{\Delta t} \mathbb{M}_j \mathbf{u}_{j,n-1}^k + \mathbf{b}_{j,n}^k - Q_j^T (l_n^k, r_n^k)^T, \\
 B_{j,n}^l = -i \left( \sum_{s=0}^m a_s^m \right) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} - \frac{\partial_x W_n(a_j)}{4} \frac{1}{\frac{2i}{\Delta t} + W_n(a_j)}, \\
 B_{j,n}^r = -i \left( \sum_{s=0}^m a_s^m \right) + i \sum_{s=1}^m a_s^m d_s^m \frac{1}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} + \frac{\partial_x W_n(b_j)}{4} \frac{1}{\frac{2i}{\Delta t} + W_n(b_j)}, \\
 b_{j,n}^{k,l} = i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \varphi_{j,n-1}^s \\
 \quad - \operatorname{sgn}(\partial_x W_n(a_j)) \frac{\sqrt{|\partial_x W_n(a_j)|}}{2} \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j)} \cdot Q_{j,l} \psi_{j,n-1}, \\
 b_{j,n}^{k,r} = i \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \varphi_{j,n-1}^s \\
 \quad + \operatorname{sgn}(\partial_x W_n(b_j)) \frac{\sqrt{|\partial_x W_n(b_j)|}}{2} \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j)} \cdot Q_{j,r} \psi_{j,n-1}, \\
 \text{où pour } s = 1, 2, \dots, m, \\
 Q_{j,l} \varphi_{j,n-1/2}^s = \frac{1}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \mathbf{v}_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \cdot Q_{j,l} \varphi_{j,n-1}^s, \\
 Q_{j,r} \varphi_{j,n-1/2}^s = \frac{1}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \mathbf{v}_{j,n}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \cdot Q_{j,r} \varphi_{j,n-1}^s, \\
 Q_{j,l} \varphi_{j,n}^s = 2Q_{j,l} \varphi_{j,n-1/2}^s - Q_{j,l} \varphi_{j,n-1}^s, \\
 Q_{j,r} \varphi_{j,n}^s = 2Q_{j,r} \varphi_{j,n-1/2}^s - Q_{j,r} \varphi_{j,n-1}^s, \\
 Q_{j,l} \varphi_{j,0}^s = Q_{j,r} \varphi_{j,0}^s = 0, \\
 \text{et} \\
 Q_{j,l} \psi_{j,n-1/2} = \frac{\sqrt{|\partial_x W_n(a_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_n(a_j)} \cdot Q_{j,l} \mathbf{v}_{j,n-1}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j)} \cdot Q_{j,l} \psi_{j,n-1}, \\
 Q_{j,r} \psi_{j,n-1/2} = \frac{\sqrt{|\partial_x W_n(b_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_n(b_j)} \cdot Q_{j,r} \mathbf{v}_{j,n-1}^k + \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j)} \cdot Q_{j,r} \psi_{j,n-1}, \\
 Q_{j,l} \psi_{j,n} = 2Q_{j,l} \psi_{j,n-1/2} - Q_{j,l} \psi_{j,n-1}, \\
 Q_{j,r} \psi_{j,n} = 2Q_{j,r} \psi_{j,n-1/2} - Q_{j,r} \psi_{j,n-1}, \\
 Q_{j,l} \psi_{j,0} = Q_{j,r} \psi_{j,0} = 0.
 \end{array} \right.$$



• Une expression de  $\mathbf{v}_{j,n}^k$

$$\begin{aligned}
\mathbf{v}_{j,n}^k &= -(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} Q_j^T (l_{j,n}^k, r_{j,n}^k)^T + (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} (-1)^{n-1-q} \frac{4i}{\Delta t} \mathbb{M}_j \mathbf{v}_{j,q}^k \\
&+ i(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(a_j) + d_s^m} \frac{2}{\frac{2i}{\Delta t} + W_q(a_j) + d_s^m} \right. \\
&\quad \left. \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(a_j) + d_s^m} - 1 \right) Q_{j,l}^T Q_{j,l} \right) \mathbf{v}_{j,q}^k \\
&- i \operatorname{sgn}(\partial_x W_n(a_j)) \frac{\sqrt{|\partial_x W_n(a_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_n(a_j)} (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \\
&\quad \cdot \sum_{q=1}^{n-1} \left( \frac{\sqrt{|\partial_x W_q(a_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_q(a_j)} \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(a_j)} - 1 \right) Q_{j,l}^T Q_{j,l} \right) \mathbf{v}_{j,q}^k \\
&+ i(\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \sum_{q=1}^{n-1} \left( \sum_{s=1}^m a_s^m d_s^m \frac{\frac{2i}{\Delta t}}{\frac{2i}{\Delta t} + W_n(b_j) + d_s^m} \frac{2}{\frac{2i}{\Delta t} + W_q(b_j) + d_s^m} \right. \\
&\quad \left. \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(b_j) + d_s^m} - 1 \right) Q_{j,r}^T Q_{j,r} \right) \mathbf{v}_{j,q}^k \\
&+ i \operatorname{sgn}(\partial_x W_n(b_j)) \frac{\sqrt{|\partial_x W_n(b_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_n(b_j)} (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \\
&\quad \cdot \sum_{q=1}^{n-1} \left( \frac{\sqrt{|\partial_x W_q(b_j)|}}{2} \frac{1}{\frac{2i}{\Delta t} + W_q(b_j)} \cdot \prod_{p=q}^{n-1} \left( \frac{\frac{4i}{\Delta t}}{\frac{2i}{\Delta t} + W_p(b_j)} - 1 \right) Q_{j,r}^T Q_{j,r} \right) \mathbf{v}_{j,q}^k \\
&+ (\mathbb{A}_{j,n} - \mathbb{B}_{j,n})^{-1} \left( \frac{2i}{\Delta t} (-1)^{n-1} \mathbb{M}_j \right) \mathbf{u}_{j,0}
\end{aligned}$$



# Annexe B

## Introduction à la librairie PETSc

PETSc (Portable, Extensible Toolkit for Scientific Computation librairie) est composée d'une variété de bibliothèques (semblable à des classes en C++). Chaque bibliothèque manipule une famille particulière d'objets (par exemple, des vecteurs) et les opérations à appliquer à ces objets. Certains des modules PETSc traitent

- un ensemble d'index (SI), y compris les permutations, pour l'indexation dans des vecteurs, renumérotation, etc;
- vecteurs (Vec);
- matrices (Mat) (généralement creuses);
- la gestion des interactions entre les mailles des structures et des vecteurs et des matrices de données (DM);
- plus de quinze méthodes de sous-espaces de Krylov (KSP);
- des dizaines de préconditionneurs, y compris multigrille, résolution par blocs, et solveurs directs à faible densité (PC);
- solveurs non linéaires (SNES);
- timesteppers pour résoudre des EDPs non linéaires dépendant du temps (TS).

La figure B.1 montre un schéma de certains de ces modules.

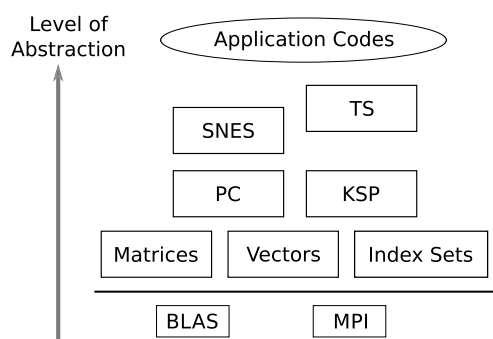


FIGURE B.1: Organisation de la bibliothèque PETSc

Dans cette thèse, on utilise les solveurs Krylov (KSP) avec ou sans préconditionneur (PC) pour résoudre le système linéaire

$$Ax = b, \quad \mathcal{P}Ax = \mathcal{P}b,$$

où  $A$  est une matrice,  $b$  est un vecteur et  $\mathcal{P}$  est un préconditionneur.

KSP supporte des matrices creuses distribuées. La matrice est divisée en diverses parties et chaque partie est stockée dans un processus MPI. L'exemple ci-dessous montre l'application à une  $8 \times 8$  matrice avec 34 éléments non nuls qui est assemblée sur 3 processus.

$$\left( \begin{array}{ccc|ccc|cc} 1 & 2 & 0 & 0 & 3 & 0 & 0 & 4 \\ 0 & 5 & 6 & 7 & 0 & 0 & 8 & 0 \\ 9 & 0 & 10 & 11 & 0 & 0 & 12 & 0 \\ \hline 13 & 0 & 14 & 15 & 16 & 17 & 0 & 0 \\ 0 & 18 & 0 & 19 & 20 & 21 & 0 & 0 \\ 0 & 0 & 0 & 22 & 23 & 0 & 24 & 0 \\ \hline 25 & 26 & 27 & 0 & 0 & 28 & 29 & 0 \\ 30 & 0 & 0 & 31 & 32 & 33 & 0 & 34 \end{array} \right) \left. \begin{array}{l} \} \text{MPI Proc. 0} \\ \\ \} \text{MPI Proc. 1} \\ \\ \} \text{MPI Proc. 2} \end{array} \right\}$$

KSP supporte aussi les matrices de type "matrix-free". La matrice n'est ainsi pas stockée explicitement, mais le produit matrice-vecteur est défini par une fonction. C'est l'opération de base dont nous avons besoin pour les méthodes de Krylov.

Le vecteur parallèle est aussi distribué sur des processus (un exemple est montré ci-dessous).

$$\left( \begin{array}{c} 1 \\ 2 \\ 3 \\ \hline 4 \\ 5 \\ 6 \\ \hline 7 \\ 8 \end{array} \right) \left. \begin{array}{l} \} \text{MPI Proc. 0} \\ \\ \} \text{MPI Proc. 1} \\ \\ \} \text{MPI Proc. 2} \end{array} \right\}$$

De la même manière que pour les matrices de type "matrix-free", le préconditionneur  $\mathcal{P}$  peut-être défini par une fonction.

# Annexe C

## Introduction de GPGPU

On se réfère à [58, 63] pour cette partie. Le GPGPU est l'abréviation de "general-purpose processing on graphics processing units", c'est-à-dire un calcul générique sur un processeur graphique afin de bénéficier de leur capacité de traitement massivement parallèle (ref. Wikipédia).

CUDA (Compute Unified Device Architecture) est une technologie de GPGPU permettant à un processeur graphique (Gpu) d'être utilisé pour exécuter des calculs généraux habituellement exécutés par le processeur central (Cpu). Lancé en 2006 par NVIDIA, elle s'est imposés comme un standard de l'industrie (ref. Wikipédia).

Architecturalement, les Cpus incluent un nombre restreint de cœurs optimisés pour le traitement en série, alors que qu'un Gpu intègre des milliers de cœurs conçus pour traiter efficacement de nombreuses tâches simultanées (voir figure C.1).

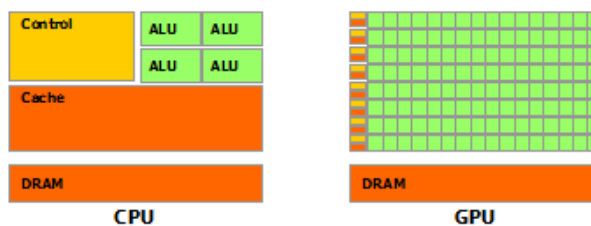


FIGURE C.1: Illustration de l'architectures de Cpu et de Gpu.

L'histoire de l'architecture des Gpus NVIDIA est la suivante

- Architecture Tesla
  - G80 Architecture, 2006.
  - GT200 Architecture, 2008.
- Architecture Fermi, 2010.
- Architecture Kepler, 2012.

- Architecture Maxwell, prévu pour 2014.

Il existe trois séries de Gpus sous l'architecture Kepler

- GeForce: calcul graphique.
- Quadro: visualisation professionnelle.
- Tesla: calcul scientifique.

Le ordinateur que nous avons utilisé pour cette thèse contient 8 NVIDIA Tesla K20 (Architecture Kepler). Les caractéristiques clés des K20 et K40 (le plus récent) sont montrées dans le tableau C.1.

	K20	K40
Performances maximales de double précision en virgule flottante	1.17Tflops	1.43Tflops
Performances maximales de simple précision en virgule flottante	3.52Tflops	4.29Tflops
Bande passante mémoire	208Go/s	288Go/s
Capacité mémoire (GDDR5)	5 Go	12 Go
Nombre de cœurs CUDA	2496	2880

TABLEAU C.1: Caractéristiques clés de NVIDIA Tesla K20 et K40.

Concernant la performance, les Gpu ont évolué vers un processeur massivement parallèle, multitâche avec une puissance de calcul énorme et un très haut débit de la mémoire, comme illustré par les figures C.2a et C.2b.

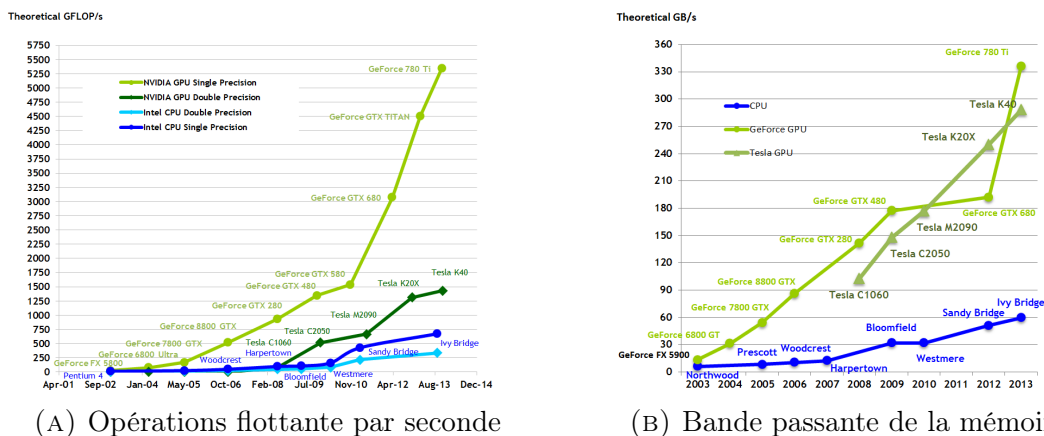


FIGURE C.2: Performance de Cpu et Gpu.

On utilise les Gpus pour accélérer les portions de code les plus lourdes en ressources de calcul, le reste de l'application restant affecté au Cpu (Fig. C.3).

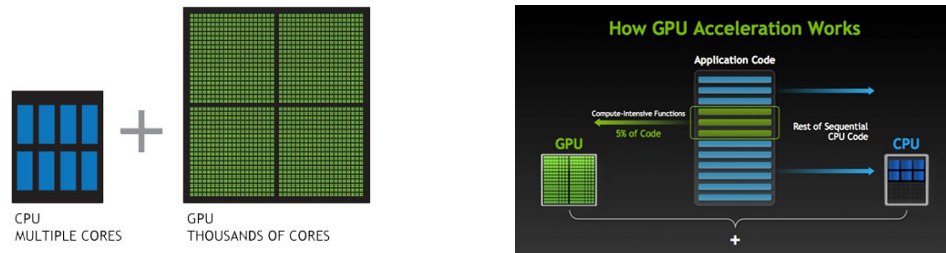


FIGURE C.3: Accélération par Gpu.

Il y a trois façons d'accélérer les applications par Gpu

1. Bibliothèques.
2. directives OpenACC [64].
3. Langage de programmation (CUDA).

Dans nos programmes, on utilise les bibliothèques et le langage de programmation.

Nous avons besoin de résoudre beaucoup de systèmes linéaires car la méthode implicite est utilisée pour discrétiser l'équation. C'est la partie la plus couteuse dans nos programmes. En outre, de nombreuses opérations BLAS (multiplication matrice vecteur,...) sont nécessaires. Comme il existe des bibliothèques qui accélèrent les calculs pour Gpu comme cela est montré dans tableau C.2, on va les utiliser pour diminuer le temps de calcul.

	Cpu	Gpu
Solveur direct	MKL Pardiso	cuSPARSE
Solveur itératif	PETSc, CULA	
Blas	MKL Blas	cuBLAS

TABLEAU C.2: Les bibliothèques de Cpu et Gpu.

- Intel MKL (Intel Math Kernel Library)[49] (bibliothèques de Cpu) est une bibliothèque de routines mathématiques optimisées pour la science, l'ingénierie et les applications financières. Les fonctions mathématiques de base comprennent BLAS, LAPACK, ScaLAPACK, des solveurs pour matrice creuse, transformées de Fourier rapides, et le vecteur mathématiques.
- La bibliothèque cuSPARSE (NVIDIA CUDA Sparse Matrix)[59] fournit une collection de sous-routines d'algèbre linéaire de base utilisées pour les matrices creuses.
- La bibliothèque cuBLAS (NVIDIA CUDA Basic Linear Algebra)[57] est une version accélérée par le GPU de la bibliothèque BLAS qui offre 6x à 17x une performance plus rapide que la dernière MKL BLAS.

- La bibliothèque CULA[48] est un ensemble de bibliothèques d'algèbre linéaire accélérées par le GPU utilisant l'architecture parallèle de calcul NVIDIA CUDA pour améliorer considérablement la vitesse de calcul des mathématiques sophistiquées. Il inclut les solveurs itératives de Krylov (Gmres, Bicgstab, etc.) avec préconditionneurs comme Jacobi, ilu0, etc..



# Bibliographie

- [1] Xavier Antoine, Anton Arnold, Christophe Besse, Matthias Ehrhardt, and Achim Schädlle. A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Commun. Comput. Phys.*, 4(4):729–796, 2009.
- [2] Xavier Antoine, Weizhu Bao, and Christophe Besse. Computational methods for the dynamics of the nonlinear Schrödinger/Gross–Pitaevskii equations. *Computer Physics Communications*, 00:1–23, 2013.
- [3] Xavier Antoine and Christophe Besse. Unconditionally stable discretization schemes of non-reflecting boundary conditions for the one-dimensional Schrödinger equation. *Journal of Computational Physics*, 188(1):157–175, 2003.
- [4] Xavier Antoine, Christophe Besse, and Stephane Descombes. Artificial boundary conditions for one-dimensional cubic nonlinear Schrödinger equations. *SIAM Journal on Numerical Analysis*, 43(6):2272–2293, 2006.
- [5] Xavier Antoine, Christophe Besse, and Pauline Klein. Absorbing boundary conditions for the one-dimensional Schrödinger equation with an exterior repulsive potential. *Journal of Computational Physics*, 228(2):312–335, 2009.
- [6] Xavier Antoine, Christophe Besse, and Pauline Klein. Absorbing Boundary Conditions for General Nonlinear Schrödinger Equations. *SIAM Journal on Scientific Computing*, 33(2):1008–1033, 2011.
- [7] Xavier Antoine, Christophe Besse, and Pauline Klein. Absorbing boundary conditions for the two-dimensional Schrödinger equation with an exterior potential. Part I : Construction and a priori estimates. *Mathematical Models and Methods in Applied Sciences*, 22(10), 2012.
- [8] Xavier Antoine, Christophe Besse, and Pauline Klein. Absorbing boundary conditions for the two-dimensional Schrödinger equation with an exterior potential. Part II: Discretization and numerical results. *Numerische Mathematik*, 125(2):191–223, 2013.
- [9] Xavier Antoine, Christophe Besse, and Jérémie Szeftel. Towards accurate artificial boundary conditions for nonlinear PDEs through examples. *Cubo, A Mathematical Journal*, 11(4):29–48, 2009.
- [10] Xavier Antoine and Romain Duboscq. Computer Physics cations GPELab , a Matlab Toolbox to solve Gross-Pitaevskii Equations I : computation of stationary solutions. *Computer Physics Communications*, 00:1–38, 2014.

- [11] Xavier Antoine, Emmanuel Lorin, and André D. Bandrauk. Domain Decomposition Method and High-Order Absorbing Boundary Conditions for the Numerical Simulation of the Time Dependent Schrödinger Equation with Ionization and Recombination by Intense Electric Field. *Journal of Scientific Computing*, pages 1–27, 2014.
- [12] Alain Bamberger, Roland Glowinski, and Quang Huy Tran. A domain decomposition method for the acoustic wave equation with discontinuous coefficients and grid change. *SIAM Journal on Numerical Analysis*, 34(2):603–639, 1997.
- [13] Weizhu Bao and Yongyong Cai. Mathematical theory and numerical methods for Bose-Einstein condensation. *Kinetic and Related Models*, 6(1):1–135, 2012.
- [14] Weizhu Bao, Yongyong Cai, and Hanquan Wang. Efficient numerical methods for computing ground states and dynamics of dipolar Bose–Einstein condensates. *Journal of Computational Physics*, 229(20):7874–7892, 2010.
- [15] Weizhu Bao, I-Liang Chern, and Fong Yin Lim. Efficient and spectrally accurate numerical methods for computing ground and first excited states in Bose–Einstein condensates. *Journal of Computational Physics*, 219(2):836–854, 2006.
- [16] Weizhu Bao and Qiang Du. Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow. *SIAM Journal on Scientific Computing*, 25(5):1674–1697, 2004.
- [17] Weizhu Bao, Daniel Marahrens, Qinglin Tang, and Yanzhi Zhang. A Simple and Efficient Numerical Method for Computing the Dynamics of Rotating Bose–Einstein Condensates via Rotating Lagrangian Coordinates. *SIAM Journal on Scientific Computing*, 35(6), 2013.
- [18] Weizhu Bao, Peter A. Markowich, and Hanquan Wang. Ground, Symmetric and Central Vortex States in Rotating Bose-Einstein Condensates. *Communications in Mathematical Sciences*, 3(1):57–88, 2005.
- [19] Daniel Bennequin, Martin J. Gander, and Laurence Halpern. A homographic best approximation problem with application to optimized Schwarz waveform relaxation. *Mathematics of Computation*, 78(265):185–223, 2009.
- [20] Christophe Besse. A relaxation scheme for the nonlinear Schrödinger equation. *SIAM Journal on Numerical Analysis*, 42(3):934–952, 2004.
- [21] Eric Blayo, Laurence Halpern, and Caroline Japhet. Optimized Schwarz waveform relaxation algorithms with nonconforming time discretization for coupling convection-diffusion problems with discontinuous coefficients. *Domain decomposition methods in science and engineering XVI*, pages 267–274, 2007.
- [22] Yassine Boubendir, Xavier Antoine, and Christophe Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 231(2):262–280, 2012.
- [23] Filipa Caetano, Martin J. Gander, Laurence Halpern, and Jérémie Szeftel. Schwarz waveform relaxation algorithms for semilinear reaction-diffusion equations. *Networks and Heterogeneous Media*, 5(3):487–505, 2010.

- [24] Xiaochuan Cai. Additive Schwarz algorithms for parabolic convection-diffusion equations. *Numerische Mathematik*, pages 1–23, 1991.
- [25] Xiaochuan Cai. Multiplicative Schwarz methods for parabolic problems. *SIAM Journal on Scientific Computing*, pages 1–18, 1994.
- [26] Bruno Després. Décomposition de domaine et problème de Helmholtz. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 311(6):313–316, 1990.
- [27] Victorita Dolean, Martin J. Gander, and Luca Gerardo-Giorda. Optimized Schwarz Methods for Maxwell's Equations. *SIAM Journal on Scientific Computing*, 31(3):2193–2213, 2009.
- [28] A. Durán and J.M. Sanz-Serna. The numerical integration of relative equilibrium solutions. The nonlinear Schrodinger equation. *IMA Journal of Numerical Analysis*, 20(2):235–261, 2000.
- [29] Martin J. Gander. Overlapping Schwarz waveform relaxation for parabolic problems. *Contemporary Mathematics*, 218:425–431, 1998.
- [30] Martin J. Gander. Optimized Schwarz Methods. *SIAM Journal on Numerical Analysis*, 44(2):699–731, 2006.
- [31] Martin J. Gander. Schwarz methods over the course of time. *Electronic Transactions on Numerical Analysis*, 31:228–255, 2008.
- [32] Martin J. Gander and Laurence Halpern. Optimized Schwarz Waveform Relaxation Methods for Advection Reaction Diffusion Problems. *SIAM Journal on Numerical Analysis*, 45(2):666–697, 2007.
- [33] Martin J. Gander and Laurence Halpern. *Méthodes de décomposition de domaine*. Encyclopédie électronique pour les ingénieurs, 2012.
- [34] Martin J. Gander, Laurence Halpern, and Michel Kern. A schwarz waveform relaxation method for advection-diffusion-reaction problems with discontinuous coefficients and non-matching grids. In *Domain decomposition methods in science and engineering XVI*, pages 283–290. Springer, 2007.
- [35] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. In *Proceedings of the 9th International Conference on Domain Decomposition Methods*, pages 27–36. 1999.
- [36] Martin J. Gander, Laurence Halpern, and Frédéric Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM Journal on Numerical Analysis*, 41(5):1643–1681, 2003.
- [37] Martin J. Gander and Felix Kwok. Best Robin Parameters for Optimized Schwarz Methods at Cross Points. *SIAM Journal on Scientific Computing*, 34(4):A1849–A1879, 2012.
- [38] Martin J. Gander and Felix Kwok. On the applicability of lions' energy estimates in the analysis of discrete optimized schwarz methods with cross points. In *Domain Decomposition Methods in Science and Engineering XX*, pages 475–483. Springer, 2013.

- [39] Martin J. Gander, Frédéric Magoules, and Frédéric Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.
- [40] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [41] Eldar Giladi and Herbert B. Keller. Space-time domain decomposition for parabolic problems. *Numerische Mathematik*, pages 279–313, 2002.
- [42] Laurence Halpern, Caroline Japhet, and Jérémie Szeftel. Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems. *SIAM Journal on Numerical Analysis*, 50(5):2588–2611, 2012.
- [43] Laurence Halpern and Jérémie Szeftel. Optimized and quasi-optimal Schwarz waveform relaxation for the one-dimensional Schrödinger equation. Technical report, CNRS, 2006.
- [44] Laurence Halpern and Jérémie Szeftel. Nonlinear nonoverlapping Schwarz waveform relaxation for semilinear wave propagation. *Mathematics of Computation*, 78(266):865–889, 2009.
- [45] Laurence Halpern and Jérémie Szeftel. Optimized and quasi-optimal Schwarz waveform relaxation for the one dimensional Schrödinger equation. *Mathematical Models and Methods in Applied Sciences*, 20(12):2167–2199, 2010.
- [46] Thi-Thao-Phuong Hoang. *Méthodes de décomposition de domaine espace-temps pour la formulation mixte de problèmes d’écoulement et de transport en milieu poreux*. PhD thesis, Université Pierre et Marie Curie, 2013.
- [47] Thi-Thao-Phuong Hoang, Jérôme Jaffré, Caroline Japhet, Michel Kern, and Jean E. Roberts. Space-Time Domain Decomposition Methods for Diffusion Problems in Mixed Formulations. *SIAM Journal on Numerical Analysis*, 51(6):3532–3559, 2013.
- [48] John R. Humphrey, Daniel K. Price, Kyle E. Spagnoli, Aaron L. Paolini, and Eric J. Kelmelis. Cula: hybrid gpu accelerated linear algebra routines. In *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2010.
- [49] INTEL. *Intel Math Kernel Library Reference manual, MKL11.0 Update 1*.
- [50] Caroline Japhet. *Méthode de décomposition de domaine et conditions aux limites artificielles en mécanique des fluides: méthode Optimisée d’Orde 2*. PhD thesis, Université Paris 13, 1998.
- [51] Pauline Klein. *Construction et analyse de conditions aux limites artificielles pour des équations de Schrödinger avec potentiels et non linéarités*. PhD thesis, Université Henri Poincaré, Nancy 1, 2010.
- [52] Pierre-Louis Lions. On the Schwarz alternating method. I. In *First international symposium on domain decomposition methods for partial differential equations*, pages 1–42, 1988.

- [53] Pierre-Louis Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In *Third international symposium on domain decomposition methods for partial differential equations*, pages 202–223, 1990.
- [54] Sébastien Loisel. Condition Number Estimates for the Nonoverlapping Optimized Schwarz Method and the 2-Lagrange Multiplier Method for General Domains and Cross Points. *SIAM Journal on Numerical Analysis*, 51(6):3062–3083, 2013.
- [55] Véronique Martin. An optimized Schwarz waveform relaxation method for the unsteady convection diffusion equation in two dimensions. *Applied Numerical Mathematics*, 52(4):401–428, 2005.
- [56] Frédéric Nataf and Francois Rogier. Factorization of the convection-diffusion operator and the Schwarz algorithm. *Mathematical Models and Methods in Applied Sciences*, 05(01):67–93, 1995.
- [57] NVIDIA. *Cublas library v5.5*. Number July. 2013.
- [58] NVIDIA. *Cuda c programming guide v5.5*. Number July. 2013.
- [59] NVIDIA. *Cuspars library v5.5*. Number July. 2013.
- [60] Charton Philippe, Frédéric Nataf, and Francois Rogier. Méthode de décomposition de domaine pour l'équation d'advection-diffusion. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 313(9):623–626, 1991.
- [61] Alonso Rodríguez and Luca Gerardo-Giorda. New nonoverlapping domain decomposition methods for the harmonic Maxwell system. *SIAM Journal on Scientific Computing*, 28(1):102–122, 2006.
- [62] Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, 2nd edition, 2003.
- [63] NVIDIA CUDA ZONE Website. <https://developer.nvidia.com/cuda-zone>.
- [64] OpenACC Website. <http://www.openacc-standard.org/>.
- [65] OpenMP Website. <http://www.openmp.org/>.
- [66] Yunhai Wu, Xiaochuan Cai, and David E. Keyes. Additive Schwarz methods for hyperbolic equations. *Contemporary Mathematics*, 218:468–476, 1998.