RAPPORT DE STAGE

Compression de maillage et problèmes d'évolution

Intégration temporelle et multirésolution adaptative pour les EDP en temps.

Étudiant : Alexandre EDELINE
École : ENSTA Paris - Institut Polytechnique de Paris
Période : du 14/04/2025 au 15/09/2025

Laboratoire : CMAP - École Polytechnique
Maîtres de stages : Marc MASSOT et Christian TENAUD
Tuteur académique : Patrick CIARLET

8 septembre 2025

Remerciements

Je tiens à remercier...

Abstracts

Résumé en Français

Mots-clés : Schémas Numériques, Simulation des EDP d'Évolution, Multirésolution Adaptative, Méthodes ImEx, Advection-Diffusion-Réaction, Analyse d'erreur numérique, Analyse de stabilité.

Ce papier documente mon projet de fin d'études qui a pris place au laboratoire du Centre de Mathématiques Appliquées de l'École Polytechnique (CMAP). Cette expérience en recherche académique a été une opportunité exceptionnelle car elle m'a permis de mieux comprendre les rouages de la recherche, de mettre en application et en relation les concepts et savoirs-faire acquis au cours de mes études, d'améliorer la communication et le partage de mon travail, d'échanger avec des chercheurs d'horizons divers et découvrir des thèmes et des problématiques scientifiques qu'y m'étaient inconnues, en somme de parfaire mon parcours académique et assurer une heureuse transition avec le monde professionnel. Mon travail de recherche porte sur des méthodes modernes pour la simulation des équations d'advection-diffusion-réaction (ADR), des EDP assez capricieuses, aux applications multiples, régissants entre autres les phénomènes de combustions. Ce rapport contient une introduction aux défis que portent ces équations et une introduction aux stratégies imaginées pour relever ses défis, le tout accompagnés de quelques rappels mathématiques bienvenus. Il inclut bien sûr une présentation de mes contributions : deux études, une portant sur **la multi-résolution adaptative**, j'y présente une étude théorique de l'erreur qu'elle apporte sur un cas particulier ; et l'autre sur **les méthodes Runge et Kutta Implicites-Explicites (ImEx)**, je présente une analyse sur les équations d'ADR de ces méthodes et les compare a un autre méthode plus standard.

English abstract

Keywords : Numerical Schemes, Evolution PDE Simulation, Adaptive Multiresolution, ImEx Methods, Advection-Diffusion-Reaction, Numerical Error Analysis, Stability Analysis.

This paper documents my final year project conducted at the Applied Mathematics laboratory of École Polytechnique (CMAP). This academic research experience has been an exceptional opportunity as it allowed me to better understand the workings of research, to apply and connect the concepts and skills acquired during my studies, to improve communication and sharing of my work, to interact with researchers from diverse backgrounds, and to discover scientific themes and problems that were previously unknown to me, in short, to complete my academic journey and ensure a smooth transition to the professional world. My research work focuses on modern methods for simulating advection-diffusion-reaction (ADR) equations, rather challenging PDEs with multiple applications, governing among others combustion phenomena. This report contains an introduction to the challenges posed by these equations and an introduction to the strategies devised to address these challenges, all accompanied by some welcome mathematical reminders. It naturally includes a presentation of my contributions : two studies, one focusing on **adaptive multiresolution**, where I present a theoretical study of the error it introduces in a particular case, and the other on **Implicit-Explicit Runge-Kutta methods (ImEx)**, where I present an analysis of these methods on ADR equations and compare them to another more standard method.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Présentation du laboratoire

1.1.1 Historique et activités

Le Centre de Mathématiques Appliquées de l'École Polytechnique [1] (CMAP) a été créé en 1974 lors du déménagement de l'École Polytechnique vers Palaiseau. Cette création répond au besoin émergent de mathématiques appliquées face au développement des méthodes de conception et de simulation par calcul numérique dans de nombreuses applications industrielles de l'époque(nucléaire, aéronautique, recherche pétrolière, spatial, automobile). Le laboratoire fut fondé grâce à l'impulsion de trois professeurs : Laurent SCHWARTZ, Jacques-Louis LIONS et Jacques NEVEU. Jean-Claude NÉDÉLEC en fut le premier directeur, et la première équipe de chercheurs associés comprenait P.A. RAVIART, P. CIARLET, R. GLOWINSKI, R. TEMAM, J.M. THOMAS et J.L. LIONS. Les premières recherches se concentraient principalement sur l'analyse numérique des équations aux dérivées partielles. Le CMAP s'est diversifié au fil des décennies, intégrant notamment les probabilités dès 1976, puis le traitement d'images dans les années 1990 et les mathématiques financières à partir de 1997. Le laboratoire a formé plus de 230 docteurs depuis sa création et a donné naissance à plusieurs startups spécialisées dans les applications industrielles des mathématiques appliquées.

1.1.2 La recherche au CMAP

Le CMAP comprend trois pôles de recherche : le pôle analyse, le pôle probabilités et le pôle décision et données. Chaque pôle accueil en son sein plusieurs équipes :

1. **Analyse**
   - ◇ EDP pour la physique.
   - ◇ Mécanique, Matériaux, Optimisation de Formes.
   - ◇ HPC@Maths (calcul haute performance).
   - ◇ PLATON (quantification des incertitudes en calcul scientifique), avec l'INRIA.
2. **Probabilités**
   - ◇ Mathématiques financières.
   - ◇ Population, système particules en interaction.

1. https ://cmap.ip-paris.fr

- ◇ ASCII (interactions stochastiques coopératives), avec l'INRIA.
- ◇ MERGE (évolution, reproduction, croissance et émergence), avec l'INRIA.

3. **Décision et données**
   - ◇ Statistiques, apprentissage, simulation, image.
   - ◇ RandOpt (optimisation aléatoire).
   - ◇ Tropical (algèbre (max, +)), avec lINRIA.

J'ai intégré l'équipe **HPC@Maths pole analyse**. De nombreuses équipe sont partagées entre le CMAP et l'INRIA ce qui démontre l'aspect appliqué du laboratoire.

1.1.3 L'équipe HPC@Math et lenvironnement de travail

.i L'équipe HPC@Math L'équipe HPC@Math [2] travaille à l'interface des mathématiques de la physique (mécanique des fluides, thermodynamique) et de l'informatique pour développer des méthodes numériques complètes (schéma, analyse d'erreur, implémentation) pour la simulation des EDP. Léquipe se centre sur les problèmes multi-échelles ; les EDPs cibles qui typiquement étudiées sont les équations d'advection-réaction-diffusion qui représente de manière générale le couplage entre la mécanique des fluides, la thermodynamique et la chimie (typiquement un problème de combustion). Tout cela se fait dans le contexte HPC (high performance computing). Le HPC désigne l'usage optimal des ressources informatiques disponibles cela peut être développer une simulation efficace sur une petite machine comme des schéma hautement parallélisable dans des paradigmes de calculs hybrides ou dans des contextes hexascale (échelle hexaflopique) [3]. Ainsi l'application des méthodes développées est au cur des réflexions de l'équipe.

.ii Envrionnement de travail

2. https ://initiative-hpc-maths.gitlab.labos.polytechnique.fr/site/index.html
3. Plateformes de calculs ayant une capacité de calcul théorique de 10^{16} opérations par seconde (hexaflops).

Chapitre 2

Description du travail objectifs et état de l'art

Cette partie présente les objectifs du stage et les méthodes employées. Elle introduit également le lecteur au sujet, à ses problématiques et comprend un préambule mathématique présentant un bref état de l'art et les notions élémentaires des différents domaines convoqués.

2.1 Présentation du sujet et problématique générale

Ce travail participe à l'élaboration de méthodes numériques pour l'approximation des équations aux dérivées partielles d'évolution. En particulier les équations d'advection-diffusion-réaction (présentation en 2.2.2). Elles décrivent par exemple les systèmes physiques couplant mécanique des fluides, thermodynamique et réactions chimiques [1]. Ces équations sont difficiles à simuler du fait de leur caractère multi-échelle [2]. Pour gérer les différentes échelles spatiales, des méthodes de compression de maillage sont souvent mises en oeuvre. La méthode de compression utilisée et étudiée ici est la multirésolution adaptative [11]. Les différentes échelles temporelles [3] sont usuellement gérées par force brute ou par séparation d'opérateurs. Pour pallier le problème de la large gamme d'échelles temporelles rencontrées, une approche hybride est ici étudiée : les méthodes implicites-explicites (ImEx) [3]. Ce travail vise donc principalement à comprendre comment la multirésolution adaptative interagit avec les différentes méthodes d'intégration temporelle. Il s'intéresse aux questions suivantes :

- Comment les effets de la compression de maillage par multirésolution adaptative (MRA) sur les solutions numériques dépendent-ils du problème étudié et de la méthode numérique sur lesquelles elle se greffent ?
- Comment évoluent les propriétés des méthodes ImEx selon les caractéristiques des opérateurs des équations de diffusion-réaction [4] ?

1. Typiquement des problèmes de combustion.
2. Une réaction chimique a des temps et distances typiques généralement plusieurs ordres de grandeur plus faibles que les temps et distances typiques de la mécanique des fluides.
3. En termes techniques, les différents termes des équations étudiées ont des raideurs très différentes.
4. Même si l'objectif est bien les équations d'advection-diffusion-réaction, l'étude s'est concentrée par simplicité sur l'interaction entre phénomènes de diffusion et de réactions.

2.2 Quelques notions techniques

2.2.1 Intégrations des EDOs

Les techniques d'approximation d'EDPs d'évolution comportent souvent une étape nécessitant la résolution d'une équation différentielle ordinaire (EDO [5]), c'est à dire une équation différentielle ne faisant intervenir qu'une seule variable différenciée (ici le temps). Nous commençons donc cette section par rappeler quelques notions d'analyse et de simulation des EDOs [6].

Définition 2.2.1 (Équation différentielle ordinaire)**.** Une équation différentielle ordinaire est une équation de la forme :

$$\begin{aligned} u' &= A(u,t) \quad u : t \in \mathbb{R}^+ \mapsto u(t) \in \mathbb{R}^d \\ u(0) &= u_0 \in \mathbb{R}^d. \end{aligned} \tag{2.1}$$

A Schémas explicites et implcites.

L'approximation des EDO se fait grâce à des schémas numériques ; formellement un schéma numérique est un élément de $(\mathbb{R}^d)^{\mathbb{N}}$. Ceux-ci se divisent en deux catégories, les schémas explicites et les schémas implicites Seuls les schémas à un pas sont ici présentés et non pas les schémas multi-pas. Ce choix est fait en raison de la barrière de Dahlquist [7]. Donné un pas de discrétisation temporel Δt, on note u^n l'approximation de la solution d'une EDO au pas de temps n, c'est à dire au temps $t^n = n\Delta t$. L'objectif est d'avoir $u^n \approx u(t = n\Delta t)$.

Définition 2.2.2 (Schéma explicite)**.** Un schéma numérique est dit explicite si le pas de temps $n+1$ est obtenu seulement grâce au pas de temps n, usuellement formulé sous la forme :

$$u^{n+1} = u^n + f(u^n, \Delta t). \tag{2.2}$$

Définition 2.2.3 (Schéma implicite)**.** Un schéma numérique est dit implicite si le pas de temps $n+1$ est obtenu au moins en partie grâce au pas de temps $n+1$, souvent écrit comme :

$$u^{n+1} = u^n + f(u^{n+1}, \Delta t). \tag{2.3}$$

Ainsi, une itération d'un schéma implicite nécessite l'inversion d'un système linéaire ou non linéaire.

De fait une itération implicite est souvent plus coûteuse qu'une itération d'un schéma explicite [8]. Cependant pour des raisons de stabilités (voir C) les méthodes explicites peuvent nécessiter des pas de temps bien plus fin, et donc bien plus d'itérations. Le choix entre méthode explicite et implicite dépend de bien des facteurs (du problème, du niveau de précision voulu, de la difficulté d'implémentation etc...) c'est un enjeu central de la simulation numérique.

5. On utilisera aussi le terme *système dynamique*, même si en toute rigueur ce concept est un peu plus large.
6. Pour nos besoins nous nous restreignons aux EDOs du premier ordre.
7. https://fr.wikipedia.org/wiki/Methode_lineaire_a_pas_multiples#Premiere_et_deuxieme_limites_de_Dahlquist
8. En particulier si la dimension de la solution d est grande.

B Ordre des schémas numériques

C Stabilité des schémas numériques

Un schéma numérique d'ordre p converge vers la solution exacte de l'EDO avec une erreur qui décroît asymptotiquement en Δt^p lorsque le pas de temps diminue. Cependant, cette convergence n'est garantie que si le schéma reste stable. L'instabilité d'un schéma désigne la divergence de la solution numérique : au-delà d'un pas de temps critique Δt_0, la norme de la solution discrète $\|u^n\|$ tend vers l'infini [9]. Cette instabilité peut s'interpréter de deux manières complémentaires : d'un point de vue mathématique, le schéma se comporte comme une suite géométrique de raison $|r| > 1$; d'un point de vue physique, le schéma introduit artificiellement de l'énergie dans le système à chaque itération. La contrainte de stabilité impose donc la contrainte $\Delta t < \Delta t_0$. Lorsque ce seuil Δt_0 est très restrictif, c'est à dire très faible, la résolution de l'EDO nécessite un nombre d'itérations $T_{\text{final}}/\Delta t$ important, ce qui augmente considérablement le coût calculatoire. Les schémas explicites sont généralement plus prompts aux instabilités que les méthodes implicites.

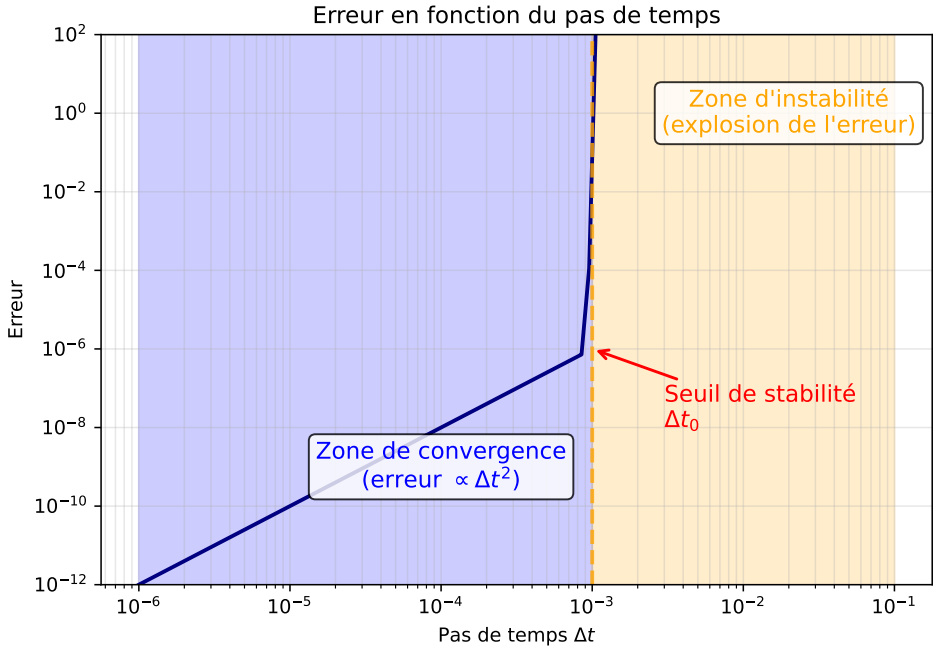


FIGURE 2.1 – Illustration du comportement attendu de l'erreur d'un schéma d'ordre deux dont le seuil d'instabilité serait $\Delta t > 10^{-1}$.

Définition 2.2.4 (Stabilité d'un schéma numérique)**.** Un schéma numérique $\left(u^n\right)_{n\in\mathbb{N}} \in \left(\mathbb{R}^d\right)^{\mathbb{N}}$ est stable si et seulement si :

$$\|u^{n+1}\| \leq \|u^n\|. \tag{2.4}$$

Cette condition est souvent vérifiée à la condition que le pas de discrétisation Δt n'excède pas un seuil de stabilité Δ_0 fonction de l'ODE et le schéma d'intégration.

9. Phénomène communément appelé "explosion" de la solution numérique.

La stabilité d'une méthode d'intégration d'EDO dépend entre autres de l'opérateur intervenant dans l'équation (le A dans l'équation 2.1). Un opérateur tendant à poser des problèmes de stabilité est dit raide.

Définition 2.2.5 (Problème raide)**.** Un système dynamique, est dit raide si les méthodes explicites ne sont pas adaptées à sa résolution. En termes plus mathématiques le système :

$$\frac{\mathrm{d}u}{\mathrm{d}t} = A(u,t), \quad u(t) \in \mathbb{R}^d, \forall t \geq 0. \tag{2.5}$$

est dit raide si la jacobienne de A, J_A possède des valeurs propres négatives de grande amplitudes devant les autres valeurs propres. Si tel est le cas, plusieurs relaxations sont mises en jeu mais chacune avec des temps caractéristiques d'ordres de grandeur différents. Pour les méthodes explicites, si le pas de temps n'est pas assez petit, la relaxation rapide est mal résolue et impose un gradient fort trop longtemps (le gradient devrait s'atténuer, mais à une échelle trop rapide pour être captée pour le pas de temps du schéma) ce qui déstabilise la méthode.

En simplifiant, si un opérateur est raide, il impose une condition de stabilité très restrictive aux méthodes explicites et force à choisir des méthodes implicites [10].

Exemple 2.2.6 (Équation de Dahlquist)**.** Pour saisir de manière plus intuitive le concept de raideur, prenons le cas de l'équation de Dahlquist [11] :

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= -\lambda u, \quad \lambda > 0 \\ u(t=0) &= u_0 \end{aligned} \tag{2.6}$$

La solution analytique est : $u(t) = u_0 e^{-\lambda t}$. Ainsi passé quelque $1/\lambda$ la dynamique du système est au point mort. En pratique la dynamique digne d'intérêt du système se concentre donc entre $t = 0$ et $t = \frac{10}{\lambda}$. Au delà, $u(t > \frac{10}{\lambda}) = o(u_0)$, la dynamique est terminée. Ainsi, si l'on souhaite simuler le comportement d'un tel système, il faut choisir des pas de temps petits devant $|\lambda|^{-1}$. Lorsque λ est de grande amplitude cela devient très contraignant... Si l'on souhaite utiliser des méthodes explicites, c'est encore pire car la raideur du système n'est plus une simple contrainte de précision mais de stabilité. En effet si l'on cherche à approximer le système par un schéma d'Euler explicite, alors : $U^{n+1} = U^n(1-\lambda\Delta t)$ alors la contrainte de stabilité est $\Delta t\,\lambda < 1/2$ ce qui est contraignant si λ est grand. Si $\lambda = 10^5$ alors il faut avoir $\Delta t \approx 10^{-5}$ donc pour simuler le système entre $t = 0$ et $t = 1$ il faut cent mille points! A l'inverse si l'on choisit un schéma d'Euler implicite : $u^{n+1} = u^n - \lambda\Delta t u^{n+1}$, alors la condition de stabilité devient : $\|(1+\lambda\Delta t)^{-1}\| \leq 1$ ce qui est toujours vrai, quelque soit $-\lambda \in \mathbb{R}^-$, la raideur du système n'est pas un problème pour la méthode implicite. On comprend mieux la définition précédente *Un système dynamique, est dit raide si les méthodes explicites ne sont pas adaptées à sa résolution.*

Il existe différents types de stabilité comme la A-stabilité (méthode stable indépendamment de la raideur du problème), la L-stabilité (schéma amortissant les hautes fréquences), par souci de concision cette partie s'achève ici mais ces notions sont développées par exemple dans [10].

10. La réalité est plus nuancée, nous le verrons.
11. C'est le cas le plus simple d'une valeur propre négative

2.2.2 Les équations d'advection-diffusion-réaction

Le contexte physique naturel des équations d'advection, diffusion, réaction est le suivant : des particules sont placées dans un milieu fluide où elles **diffusent**. Ce milieu fluide est en mouvement, il déplace les particules, les **advecte**. Enfin les particules **réagissent** entre elles et ces réactions modifient les grandeurs thermodynamiques (température, pression) et *in fine* les propriétés du milieu fluide. Les équations d'advection, diffusion, réaction modélisent donc ces trois phénomènes et leurs couplages respectifs.

A Les trois opérateurs

A.i Advection L'advection désigne le transport d'une quantité par un flot. L'opérateur d'advection le plus simple est l'opérateur de transport $c\frac{\partial}{\partial x}$:

$$\frac{\partial u}{\partial t} = c\frac{\partial u}{\partial x} \tag{2.7}$$

De manière générale l'opérateur dadvection d'une quantité u par un flot $\underline{a}$ s'écrit $\underline{a}\cdot\underline{\nabla}u$. Par exemple dans les équations de Navier-Stokes, le terme $\underline{v}\cdot\underline{\underline{\nabla}}\,\underline{v}$ modélise la vitesse $\underline{v}$ transportée par elle même. Une version simplifiée de ce phénomène est l'équation bien connue de Bürgers.

Les opérateurs d'advections sont généralement à valeurs propres imaginaires [12]. Ainsi ils sont peu raides mais résonnants. Les méthodes explicites sont souvent les plus adaptées pour les traiter.

A.ii Diffusion La diffusion désigne l'*éparpillement* de particules au sein d'un milieu fluide [13]. Ce phénomène est la limite macroscopique du déplacement microscopique des particules dû à l'agitation thermique. L'opérateur de diffusion le plus classique est celui de l'équation de la chaleur :

$$\frac{\partial u}{\partial t} = D\Delta u. \tag{2.8}$$

Le spectre de cet opérateur est $\mathbb{R}^-$, il est donc infiniment raide. Lorsqu'il est discrétisé seul une partie de sa raideur est captée, en pratique la raideur de l'opérateur augmente de manière quadratique avec la finesse de la discrétisation spatiale.

Cet opérateur est donc moyennement raide (comparé aux opérateurs de réaction). Ainsi on pourrait penser qu'une méthode implicite est adéquate. Cependant ce n'est pas toujours le cas. En effet le coefficient de diffusion est généralement fonction de la température, et donc lopérateur $D(T)\Delta(\cdot)$ varie souvent en temps et en espace. Ainsi il faut inverser à chaque itération l'opérateur implicite. Cette lourde tâche est rendue plus ardue par le fait que l'opérateur de diffusion couple tout l'espace, c'est-à-dire qu'il est non local. [14], il faut inverser une matrice de taille $d >> 1$ dont la structure peut être très hétérogène (car le coefficient de diffusion varie dans l'espace). Aujourd'hui il semble plus pertinent d'utiliser des méthodes explicites stabilisées qui parviennent à gérer la raideur moyenne

12. Par abus, s'il s'agit d'un opérateur non-linéaire on lui associera les valeurs propres de sa Jacobienne.
13. En théorie de l'information cela décrit la tendance de l'entropie augmenter et l'information à se moyenner, se flouter.
14. Si l'opérateur de diffusion eût été local il aurait pu être inversé en résolvant plusieurs petits systèmes et ce, potentiellement en parallèle. Cela serait bien moins coûteux qu'inverser un grand système. Par exemple inverser un matrice pleine de taille 10^6 demande environ 10^{18} opérations, alors qu'inverser 100 systèmes de taille 10^4 n'en demande $100\times10^{12} = 10^{14}$ soit dix mille fois moins. Si ces résolutions sont parallélisés l'accélération est d'un million. Malheureusement comme l'opérateur de diffusion couple tout l'espace ce n'est pas possible en dimension deux ou trois.

comme les méthodes ROK2 et ROK4[1]. Sans entrer dans les détails, la clé de ces méthodes consiste à prendre une méthode numérique explicite standard et lui ajouter des étages de sorte à ce que la fonction d'amplification résultante soit la fonction d'amplification de la méthode standard multipliée par un polynôme minimal sur $\mathbb{R}^-$ comme un polynôme de Tchebichev.

A.iii Réaction Les phénomènes de réaction sont en général bien adaptés aux méthodes implicites car ils sont locaux et extrêmement raides. En effet, les temps typiques d'une réaction chimique [15] sont de l'ordre de la nano-seconde. De fait, les réactions chimiques sont très difficiles à simuler par des méthodes explicites. En revanche les méthodes implicites sont très efficaces dans ce contexte. En effet l'inversion de l'opérateur implicite peut être décomposé en plusieurs résolutions de petits systèmes ce qui est moins coûteux et parallélisable. Cela est permis grâce à la localité des réactions chimiques (à chaque pas de temps les particules au sein d'une cellule ne réagissent qu'avec les autres particules de la même cellule). En pratique cela signifie qu'il est possible de mettre en oeuvre une petite méthode implicite par cellule plutôt qu'une gargantuesque méthode globale; ce qui revient à inverser un opérateur de petite dimension en chaque cellule et non inverser un énorme système.

B Difficultés mathématiques intrinsèques

La simulations des équations d'advection-réaction-diffusion se heurte à deux difficultés majeures, **le couplage des trois opérateurs** mentionnés précédemment et **le caractère multi-échelles des solutions**.

B.i Première difficulté : le couplage des opérateurs Le développement précédent montre que résoudre chaque phénomène individuellement, n'est pas insurmontable. Toutefois, les résoudre tous en même temps, c'est-à-dire les coupler, est en pratique très difficile. En effet, lorsque ces trois opérateurs sont couplés, il en résulte un unique opérateur qui doit être traité par une méthode numérique. C'est là que surgissent les difficultés : si la méthode est explicite (éventuellement stabilisée), la raideur de la réaction impose des pas de temps extrêmement restrictifs, à l'inverse si la méthode est implicite, la non-localité de la diffusion demande l'inversion d'un système de taille déraisonnable. Cette approche naïve, monolithique, n'est donc pas adaptée. Il faut par conséquent, trouver d'autres stratégies.

B.ii Seconde difficulté : le caractère multi-échelles des solutions Les solutions étudiées sont souvent multi-échelles, en temps et en espace. Cela signifie que certaines zones spatio-temporelles requièrent une finesse d'approximation élevée pour pouvoir reproduire fidèlement le comportement physique, tandis qu'en d'autres zones une approximation grossière est suffisante. Prenons l'exemple d'un incendie. Au début le foyer est très restreint et seule cette zone doit être maillée finement, car partout ailleurs *il ne se passe rien*. Petit à petit l'incendie se propage et la zone à mailler finement augmente. Un autre exemple de phénomène multi-échelle est la détonation, il faut mailler finement, au foyer de l'explosion et le front de l'onde de choc. Mais la zone non atteinte par l'explosion, qui n'a pas encore reçu le choc, pourrait être maillée très grossièrement. Dans ces conditions, il est imaginable

15. En réalité une réaction chimique aussi simple en apparence qu'une combustion H_2/O_2 fait intervenir une dizaine de composés et réactions intermédiaires, dont les temps typiques sont très faibles.)

qu'un maillage naïf mène à ce qu'en certains instants, 90% du domaine soit maillé avec un pas d'espace 100 fois plus fin que nécessaire, or en trois dimension mailler 100 fois trop finement multiplie par un million le nombre de cellules. Il y a alors une grande inefficacité computationnelle.

C Les stratégies de simulation

Pour surmonter ces difficultés, il est d'usage d'utiliser de concert des stratégies d'adaptation de maillage et d'intégration en temps spécifiques.

C.i L'adaptation de maillage Pour tirer parti du caractère multi-échelle des solutions, il est courant de recourir à des méthodes d'adaptation de maillage. C'est-à-dire d'employer une résolution variable pour la grille de calcul selon les différentes zones du domaine. L'adaptation doit se mettre à jour au fil des itérations pour suivre la dynamique de la solution car la répartition de la complexité physique évolue tout au long de la simulation. La méthode d'adaptation de maillage sur laquelle le stage se concentre est la multirésolution-adaptative par transformée d'ondelettes introduite par Ami Harten dans les années 1990 [11]. Cette méthode est très étudiée par l'équipe du CMAP et a donné lieu au développement du logiciel Samurai [16].

FIGURE 2.2 – Exemple de maillage adapté par multirésolution adaptative grâce au logiciel Samurai.

La multi-résolution adaptative La multirésolution adaptative se base sur une compression de la solution par transformée d'ondelettes [17]. Les détails mathématiques sont donnés plus tard et s'appuient notamment sur [16] mais pour résumer, la compression se fait de la manière suivante : une transformée multi-échelle sur une base d'ondelettes représente la solution sur différentes échelles d'espace [18]. Cela quantifie l'information portée par chaque niveau de résolution. Enfin la compression consiste à ignorer les échelles comprenant moins d'information qu'un seuil de compression ε

16. https://github.com/hpc-maths/samurai
17. C'est le même procédé qui est à l'oeuvre dans la compression d'image `jpg`.
18. Par exemple les échelles $\Delta x, \Delta x/2, \Delta x/4, \ldots, \Delta x/2^n$.

fixé par l'utilisateur.

Autres méthodes Il existe d'autres stratégies pour raffiner la grille de calcul que la multirésolution adaptative. La plus classique est l'adaptation basée sur la magnitude des gradients de la solution. Une grande magnitude des gradients révèle une complexité locale de la solution, appelant une fine résolution du maillage. Cette approche est simple à mettre en oeuvre et part de la même heuristique que la multirésolution adaptative (détecter où se trouve la complexité de la solution), cependant elle est moins systématique (pas de quantification intrinsèque de l'information perdue) et plus difficile à analyser. Elle est cependant utilisée dans des logiciels industriels comme Ansys [19].

C.ii Les techniques d'intégration Comme expliqué précédemment, la simulation de chaque phénomène intervenant dans les problèmes d'advection-diffusion-réaction est aisée individuellement mais très difficile conjointement. Des stratégies pour intégrer les trois opérateurs en même temps sont donc nécessaires. Elles reposent en réalité sur le fait d'intégrer chaque opérateur... séparément (ou presque).

La séparation d'opérateurs La séparation d'opérateurs (en Anglais : *operator splitting*) consiste à intégrer successivement chaque opérateur. L'intégration d'un opérateur A, dans une équation comme :

$$\frac{\partial u}{\partial t} = Au. \tag{2.9}$$

Peut s'écrire formellement avec la notion d'exponentielle de matrice [20] :

$$u(\Delta t) = e^{\Delta t A} u_0. \tag{2.10}$$

Si deux opérateurs A et B interviennent dans l'EDP cela donne :

$$\frac{\partial u}{\partial t} = (A+B)u. \tag{2.11}$$

$$u(\Delta t) = e^{\Delta t (A+B)} u_0 \tag{2.12}$$

Il est alors tentant d'écrire :

$$u(\Delta t) \approx e^{\Delta t (B)} e^{\Delta t (A)} u_0. \tag{2.13}$$

Ce n'est malheureusement qu'une approximation car $e^{\Delta t (A+B)} = e^{\Delta t (B)} e^{\Delta t (A)}$ n'est vrai que si les opérateurs A et B commutent. Cependant c'est vrai à l'ordre $O(\Delta t)$. Il est donc possible de simuler le

19. https ://www.ansys.com/fr-fr/blog/how-to-accelerate-ansys-fluent-simulations-with-adaptive-meshing
20. Ou de manière plus rigoureuse et plus générale avec la notion de semi-groupe.

problème en séparant les opérateurs de la manière suivante :

$$\begin{aligned} &\text{Étape 1 : simuler } v = e^{\Delta t A} u^n, \\ &\text{Étape 2 : simuler } u^{n+1} = e^{\Delta t B} v, \end{aligned} \tag{2.14}$$

Cet algorithme correspond au schéma de splitting de Lie, introduisant une erreur de l'ordre Δt. Il existe un autre schéma : le splitting de Strang qui est précis à l'ordre Δt^2 grâce au développement $e^{\Delta t(A+B)} = e^{\Delta t B} e^{\Delta t A} e^{\Delta t B}$. Ces méthodes de séparation d'opérateurs, très simples à mettre en oeuvre, permettent de traiter les opérateurs indépendamment les uns à la suite des autres ; chacun avec une méthode numérique adaptée. Malheureusement elles sont aveugles à certains couplages entre les différents phénomènes modélisés et il est difficile de monter au-delà de l'ordre deux en temps. La promesse des approches ImEx présentées par la suite est de combler ces lacunes.

Une étude extensive de l'usage du splitting, pour les équations d'advection-diffusion-réaction couplées à la multirésolution adaptative a été réalisée dans la thèse de Max Duarte, préparée à Centrale Paris sous la direction de Marc Massot [7]. Cela a montré que les techniques d'opérateurs sont très efficaces mais qu'en pratique, une perte de l'ordre formel peut avoir lieu.

Les méthodes ImEx Ces méthodes sont détaillées en 3.1.2. Ces méthodes ImEx[21] [15] [13] sont très proches de la séparation d'opérateurs lors de l'implémentation. Toutefois, elles apportent plus de cohérence mathématique et facilitent la montée en ordre. Une méthode Runge et Kutta est une technique d'intégration en temps dont chaque itération se décompose en plusieurs étapes (aussi appelées étages). Chaque étage engendre une nouvelle approximation, puis en fin d'itération une combinaison linéaire des ces approximations intermédiaires donne la solution au pas de temps suivant. Cette approche permet de monter en ordre efficacement. Dans les méthodes Runge et Kutta ImEx à chaque étage, l'approximation associée est obtenue en ajoutant d'une part des contributions explicites des opérateurs pour lesquels les méthodes explicites sont adaptées, et d'autre part des contributions implicites des opérateurs pour lesquels les méthodes implicites sont adaptées. Ainsi, un traitement différent est appliqué à chaque opérateur tout en conservant une approche globalement cohérente[22].

D Conclusion

Cette introduction a mis en évidence la complexité intrinsèque des équations d'advection-diffusion-réaction, qui réside dans le couplage de trois phénomènes physiques aux propriétés mathématiques antagonistes. L'advection, peu raide, la diffusion, moyennement raide et non-locale, et la réaction, extrêmement raide mais locale, ne peuvent être traitées efficacement par une approche monolithique classique. Les deux défis principaux identifiés : le couplage des opérateurs et le caractère multi-échelles des solutions, nécessitent des stratégies numériques spécifiques. D'une part, l'adaptation de maillage, notamment par multi-résolution adaptative, permet de concentrer les ressources de calcul

21. Les méthodes présentées ici sont les méthodes ImEx Runge et Kutta (RK-ImEx). Il existe également des approches ImEx couplées espace-temps[17].

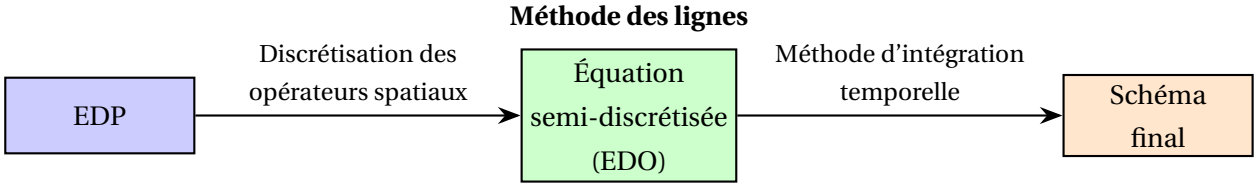
22. Cependant les méthodes de splitting offrent le luxe de choisir chaque technique de résolution numérique indépendamment des autres, ici ce n'est plus le cas des relations d'ordre doivent être respectées ; plus de cohérence mais plus de contrainte.

là où l'information physique est la plus riche. D'autre part, des techniques d'intégration découplées, séparation d'opérateurs ou des méthodes ImEx, exploitant les propriétés spécifiques de chaque phénomène et éviter l'écueil du solver monolithique.

2.2.3 Simulation des EDPs d'évolution

A Les méthodes des lignes.

Définition 2.2.7 (Méthode des lignes)**.** Une méthode des lignes est une famille de méthodes numériques pour approximer les EDP d'évolutions Elle consiste à discrétiser les opérateurs spatiaux de l'équation afin d'obtenir une équation semi-discrétisée en espace, puis à utiliser une technique d'intégration en temps, pour obtenir la discrétisation complète de l'équation.



Il existe également des approches plus sophistiquées, comme les méthodes espace-temps couplées, mais ce stage se concentre sur les méthodes des lignes classiques. Un exemple d'une telle méthode est donnée en [6].

B Les volumes finis

Le paradigme numérique utilisé dans le stage est celui des volumes finis, particulièrement adaptés au lois de conservations. Les volumes finis discrétisent la valeur moyenne sur les mailles, là où les différences finies discrétisent la valeur au nuds du maillage et les éléments finis discrétisent l'espace fonctionnel lui même. Les explications suivantes sont présentés avec plus de détails dans [14].

Définition 2.2.8 (Volumes finis)**.** Donné un maillage de pas de discrétisation Δx d'un domaine Ω en cellules $(C_j)_{j\in J}$ de volume de l'ordre Δx^d (ou d est la dimension), la discrétisation par volume fini approxime les quantités :

$$U_j = \frac{1}{|C_j|}\int_{C_j} u(x)\,d\Omega. \tag{2.15}$$

Les volumes finis brillent lors de la simulation les lois de conservations, c'est à dire les EDPs de la forme :

$$\partial_t u = div(f(u)). \tag{2.16}$$

En effet lorsque cette relation est moyennée sur une cellule C_j du maillage, cela donne :

$$\int_{C_j} \partial_t u = \int_{C_j} div(f(u)), \tag{2.17}$$

$$\partial_t U_j = \int_{\partial C_j} f(u). \tag{2.18}$$

Définition 2.2.9 (Flux physique)**.** Dans l'équation 2.18, le terme $\int_{\partial C_j} f(u)$ est appelé Φ_j le terme de *flux* de la cellule j ; physiquement il quantifie "l'entrée de $f(u)$" au sein de la cellule C_j. En une dimension $\Phi_j = f(u_j^+) - f(u_j^-)$; le flux dépend simplement des valeurs à l'interface de la cellule.

La définition précédente est une intégrale de bord faisant intervenir les valeurs exactes de u le long de l'interface. Malheureusement, paradigme des volumes n'offre qu'un accès au valeurs moyennes de u sur chaque cellule. Ainsi, il faut approximer Φ_j à partir des valeurs moyennes.

Définition 2.2.10 (Flux numérique)**.** Un *flux numérique* Ψ est fonction permettant dapproximer Φ_j à partir des valeurs moyennes sur la cellule j et les cellules voisines : $\Psi(U_{j-s},\dots,U_j,\dots,U_{j+s})$. Le nombre de cellules intervenant dans le calcul s est appelé le *stencil*.

Définition 2.2.11 (Flux numérique d'ordre p)**.** Pour être exploitable, un flux numérique doit être *consistant* à un ordre $p \geq 1$ avec la loi de conservation étudiée. Donné une solution u de régularité C^{p+1} un flux numérique est dit consistant à l'ordre p si, $\forall\, j$:

$$|\Psi\big(\tilde{U}_{j-s}(t),\dots,\tilde{U}_j(t),\dots,\tilde{U}_{j+s}(t)\big) - \Phi_j| = O(\Delta x^p). \tag{2.19}$$

Où $\tilde{U}_j$ désigne la moyenne exacte (et non une approximation) de u sur la cellule C_j.

D'un point de vue méthode des lignes, le paradigme des volumes finis donne l'équation semi-discrétisée en espace suivante :

$$\partial_t U_j(t) = \Psi\big(U_{j-s}(t),\dots,U_j(t),\dots,U_{j+s}(t)\big), \tag{2.20}$$

il ne reste qu'à l'intégrer grâce à une méthode d'intégration des EDOs.

2.2.4 Analyse de schéma numériques

Pour qualifier une schéma numérique, sur différents sujets comme la faisabilité de sa mise en oeuvre, la qualité de la solution qu'il fourni ou encore la dynamique de l'erreur qu'il introduit, divers concepts ont étés développés. Ce qui suit introduit les notions pertinentes pour comprendre les travaux du stage.

A Stabilité d'un schéma numérique

Les schémas simulant les équations au dérivées partielles ont les mêmes nécessités de stabilité, que ceux simulant les EDOs. D'ailleurs les méthodes des lignes transforment précisément une EDP en EDO. Pour éviter les redondances, le lecteur se référera à la partie 2.2.1.

B Analyse de l'erreure

L'analyse de l'ordre de convergence d'un schéma permet de quantifier l'erreur asymptotique (c'est à dire lorsque les pas de temps et d'espace sont "assez petits") que commet le schéma.

B.i Ordre de convergence d'un schéma Les schémas numériques sont désignés par $\left(U_j^n\right)_{n,j}$ où n représente le pas de temps et j la cellule. Ainsi, si la grille contient c cellules, pour tout n, U^n est un élément de l'espace vectoriel $\mathbb{R}^c$. Le vecteur $\left(u(x_j,t^n)\right)_{j\in\{1,\dots,c\}}$ est noté $u(\cdot,t^n)$.

Définition 2.2.12 (Erreur globale)**.** L'erreur globale E_G d'un schéma $\left(U_j^n\right)_{n,j}$ sur un temps $T_f = N_f \Delta t$ peut être définie comme l'erreur au temps final :

$$E_G = \Delta x^d \| U^{N_f} - u(\cdot, T_f) \| \tag{2.21}$$

où $\| \cdot \|$ désigne une norme sur $\mathbb{R}^c$. Ou bien comme l'intégrale de l'erreur sur tous les pas de temps :

$$E_G = \sum_{k=0}^{N_f} \Delta x^d \| U^k - u(\cdot, t^k) \| \tag{2.22}$$

Définition 2.2.13 (Ordre de convergence d'un schéma)**.** Un schéma $\left(U_j^n\right)_{n,j}$ est dit convergent à l'ordre p en temps et q en espace si son erreur de globale s'écrit : $E_G = O(\Delta t^p + \Delta x^q)$.

B.ii Equation équivalente

Définition 2.2.14 (Équation équivalente)**.** L'équation équivalente d'un schéma est l'EDP dont la solution satisfait le schéma. Elle est calculée par des développements de Taylor en temps et en espace. Comparer l'équation équivalente et l'équation cible fait alors naturellement apparaître les termes d'erreur et leur dynamique.

Ce qui suit décrit une méthode générique pour obtenir l'équation équivalente d'un schéma, la notion d'équation équivalente est clé dans la contribution en 3.2.

Première étape : développement de Taylor L'existence d'une fonction assez régulière vérifiant le schéma est supposée. Dans le schéma numérique les termes $u_{k+\delta_x}^{n+\delta_t}$ sont remplacés par leur pendant continu : $u(x+\delta_x\Delta x, t+\delta_t\Delta t)$. Le développement de Taylor suivant est alors réalisé :

$$\begin{aligned} u(x+\delta_x\Delta x, t+\delta_t\Delta t) = u(x,t) &+ \sum_{i=1}^{\infty} \frac{(\delta_x\Delta x)^i}{i!} \frac{\partial^i u}{\partial x^i}(x,t) \\ &+ \sum_{j=1}^{\infty} \frac{(\delta_t\Delta t)^j}{j!} \frac{\partial^j u}{\partial t^j}(x,t) \\ &+ \sum_{i,j\geq 1} \frac{(\delta_x\Delta x)^i (\delta_t\Delta t)^j}{i!\cdot j!} \frac{\partial^{i+j} u}{\partial x^i \partial t^j}(x,t) \end{aligned} \tag{2.23}$$

L'équation aux dérivées partielles qui apparaît est l'équation équivalente. Elle peut être tronquée à l'ordre voulu.

Deuxième étape : procédure de Cauchy-Kovaleskaya (optionnelle) Afin d'enrichir l'analyse, et permettre le développement de schémas couplés espace-temps, l'étape précédente peut être suivie d'une procédure de Cauchy-Kovaleskaya. La procédure de Cauchy-Kovaleskaya consiste à utiliser la relation entre les dérivées en espace et en temps données par l'équation cible et remplacer les dérivées en temps par des dérivées en espace dans l'équation équivalente. Par exemple, pour un schéma

ayant pour équation cible la diffusion $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$, cela consiste à écrire de manière itérée :

$$\begin{aligned}
\frac{\partial^2 u}{\partial t^2} &= D^2 \frac{\partial^4 u}{\partial x^4} \\
\frac{\partial^3 u}{\partial t^3} &= D^3 \frac{\partial^6 u}{\partial x^6} \\
&\vdots \\
\frac{\partial^n u}{\partial t^n} &= D^n \frac{\partial^{2n} u}{\partial x^{2n}}
\end{aligned} \tag{2.24}$$

Dernière étape : relation entre le pas de temps et d'espace (optionnelle) Lorsque le schéma est utilisé en pratique, il est courant d'imposer une relation entre les pas d'espace et de temps, par exemple une condition de stabilité du type $\Delta t \propto Deltax$ ou $\Delta t \propto Deltax^2$. Il est donc utile d'injecter cette relation dans l'équation équivalente pour comprendre le comportement du schéma en conditions réelles.

Ces outils d'analyse numérique constituent les fondements nécessaires pour aborder la simulation numérique des EDPs. La section suivante présente la multi-résolution adaptative qui permet de réduire le coût en calcul et en mémoire de ces méthodes.

2.2.5 La Multirésolution Adaptative

La multi-résolution adaptative (MRA) est une méthode très efficace pour les problèmes multi-échelles. L'objectif est de concentrer les efforts computationnels là où ils sont nécessaires. Concrètement cela consiste à augmenter la résolution de la grille de calcul où la solution est complexe et la diminuer où la solution est simple à décrire. La MRA est donc une méthode de HPC (*high performance computing*) puisqu'elle vise à optimiser l'allocation des ressources de calcul.

Cette partie introduit le lecteur à cette méthode en présentant d'abord le concept mathématique de transformée multi-échelle (ou transformée en ondelette) qui est à la base de la MRA. Puis il est expliqué comment la transformée multi-échelle permet d'adapter le maillage pour optimiser la charge computationnelle. Une fois ces prérequis établis, l'algorithme typique de mise en oeuvre de la multi-résolution adaptative est décrit. Sensuit alors naturellement une présentation des différentes implémentations de la MRA, avec une attention particulière sur celle développée au CMAP au travers du logiciel Samurai. Enfin l'impact de la multi-résolution sur la qualité des solutions numériques est abordé.

A La transformée multi-échelle

Cette partie présente la transformée multi-échelle discrète. La transformée multi-échelle continue en simulation numérique, c'est bien sûr la version discrète qui est utile. Elle se veut avant tout introductive et omets ou simplifie certaines notions; plus de détails sont donnés en [16].

A.i Définition mathématique Les explications sont développées en dimensions un à des fins pédagogiques, la plupart des concepts s'entendent naturellement aux dimensions supérieures. De plus, la discrétisation de l'espace se fait selon une grille dyadique, d'autre choix pourraient être fait mais c'est un choix simple, naturel et standard. Il faut détailler cette notion.

Définition 2.2.15 (Grille dyadique)**.** Une grille dyadique ou discrétisation dyadique d'un intervalle $I \subset \mathbb{R}$ est une série de partitions de I indexées par des entiers $j \in J \subset \mathbb{N}^*$. La discrétisation de niveau j correspond à une partitions de I en 2^j intervalles (voir fig. 2.3). Ainsi à chaque changement de niveau, du niveau j vers le niveau $j+1$, la résolution de la discrétisation est doublée. Les cases de cette partition dyadique sont indexées par deux entiers : j le niveau de résolution de la grille et k l'index de la case au sein de ce niveau. En particulier les cases $2k$ et $2k+1$ du niveau $j+1$ correspondent à la case k du niveau j.

Niveau $j-1$	$k=0$				$k=1$			
Niveau j	$k=0$		$k=1$		$k=2$		$k=3$	
Niveau $j+1$	$k=0$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$

FIGURE 2.3 – Exemple de grille dyadique

Dans ce qui suit il est supposé sans perte de généralité que la discrétisation se fait sur l'intervalle $[0,1]$, ainsi le niveau j correspond à des cellules de tailles $1/2^j$ et la cellule k du niveau j est centrée en $x_k^j = \frac{k+(k+1)}{2}\frac{1}{2^j} = \frac{2k+1}{2^j}$. La notion d'ondelette se définit de la manière suivante :

Définition 2.2.16 (Ondelette)**.** Une ondelette est une fonction $\Phi \in L^2(\mathbb{R})$ à support compact de moyenne nulle. Pour qu'une ondelette soit pertinente dans le cas de la transformée en multi-échelle il est requis que la famille $\left(x \mapsto \Phi(2^j k - x)\right)_{(j,k)\in\mathbb{Z}\times\mathbb{Z}}$ forme une base de $L^2(\mathbb{R})$. En effet la transformée en ondelette sera une projection sur cette base, un peu comme la transformée de Fourrier est une projection sur les fonctions trigonométriques.

Alors la transformée en ondelette discrète peut être définie :

Définition 2.2.17 (Transformée en ondelette discrète - *Discrete Wavelet Transform, DWT*)**.** Donnée une fonction f, le coefficient γ_k^j de sa DWT sur la cellule k au niveau de résolution j est :

$$\gamma_k^j = \frac{1}{N_j} \int_{\mathbb{R}} \Phi(2^j \cdot k - t) f(t) \mathrm{d}t, \tag{2.25}$$

Où N_j est un coefficient normalisation dépendant du niveau j.

Contrairement à une transformée de Fourier, les coefficients ne dépendent pas d'une mais de deux variables. En effet, la transformée en ondelette est plus riche d'informations. Là où la transformée de Fourier ne donne qu'une information sur le contenu fréquentiel d'un signal, la transformée en ondelette donne une information sur le contenu en fréquence et sur la localisation de ce contenu fréquentiel.

A.ii La notion de détails La multi-résolution adaptative se sert de la transformée multi-échelle pour adapter le maillage, c'est à dire compresser l'information (voir B). Cela requiert l'introduction de la notion de détail. Ce concept permet de ne pas utiliser la transformée en ondelette pour quantifier le contenu absolu porté par une échelle particulière, mais plutôt à comprendre en quoi ce contenu s'éloigne de ce que les échelles supérieures pourraient laisser supposer, en quoi il est *inattendu*. Pour résumer à partir d'un niveau de résolution j, on définit un **prédicteur** polynomial, qui tente d'inférer l'allure de la fonction au niveau $j+1$. Puis le **détail** ne cherche pas naïvement à quantifier et localiser l'information contenue aux échelles du niveau $j+1$ mais plutôt, à quantifier l'écart à la prédiction polynomiale.

Le prédicteur Donné un point central (x_0, y_0), et $2s$ voisins $(x_{-s}, y_{-s}), (x_{-s+1}, y_{-s+1}) \dots (x_{s-1}, y_{s-1}), (x_s, y_s)$, un prédicteur polynomial ponctuel cherche le polynôme P de degré $2s$ passant par ces $2s+1$ points. Cela permet d'inférer des valeurs pour y en tout point x. Pour trouver $P(X) = \sum_{k=0}^{2s} a_k X^k$ revient à résoudre le système linéaire :

$$\forall j \in \{-s, \dots 0, \dots, s\} : y_j = \sum_{k=0}^{2s} a_k x_j^k \tag{2.26}$$

Ce stage se focalise sur les volumes finis, donc ce n'est pas un prédicteur ponctuel, adapté au différences finies (voir **??**), qui est utilisé mais un prédicteur sur la valeur moyenne. Il ne cherche à imposer les valeurs en chaque point mais à fixer la valeurs moyenne sur chaque cellule, cela ajoute peu de complexité puisqu'il suffit d'ajouter une intégration lors de l'établissement du système linéaire pour travailler sur les valeurs moyennes. Pour l'usage que souhaité ici, il s'agit en d'évaluer la solution sur la cellule k du niveau de résolution j (ce qui correspond à la cellule de centre $x_k^j = (k+1/2)2^{-j}$) au

niveau de résolution supérieure $j+1$, il faut donc appliquer le correcteur linéaire centré sur x_k^j en $x_\pm = \pm 2^{-(j+1)}$. En pratique cela revient à faire une combinaison linéaire des $2s$ voisins qui vient corriger la valeur en x_k^j. Le prédicteur dépend donc du nombre de voisins pris de part et d'autre, ce nombre noté s et appelé le *stencil* du prédicteur. Plus s est grand plus l'opération de prédiction est précise (mais peut éventuellement devenir bruitée[23]) et plus elle est coûteuse. Le coût exact n'est pas évident à estimer puisque qu'une combinaison linéaire quelques termes se fait en $O(1)$ sur les machines modernes, toutefois quelques subtilités détaillé en B interviennent. Ainsi les valeurs usuelles tu stencil sont généralement $s=1$ ou $s=2$

Les détails À présent que le prédicteur à été décrit le concept de détail peut enfin être abordé. On suppose que l'on dispose d'une fonction $\tilde{u}^j$ qui soit une approximation de la fonction u, au niveau de résolution j. Comme vu précédemment, le prédicteur permet d'obtenir une approximation de u au niveau $j+1$ grâce à $\tilde{u}^j$. Cette prédiction est noté $\hat{u}^{j+1}$. Les détails à la résolution $j+1$ sont alors définis comme les coefficient de la transformée multi-échelle de la différence entre cette prédiction et la vraie fonction : $u - \hat{u}^{j+1}$. Alors les coefficients de détails n'encode que ce qui n'était pas prédictible par l'interpolateur polynomial. Ce concept de détail est essentiel. En pratique on ne réalise pas l'opération comme expliqué plus haut puisque à prédicteur et ondelette fixés Φ, il existe une *ondelette duale* Ψ qui permet directement d'obtenir les détails pour la DWT sur Φ en réalisant une DWT sur Ψ ce qui accélère considérablement les calculs.

Une autre façon de voir la notion de détail est la suivante : pour un niveau de résolution j on note $V_j = \text{Vect}\left((\Phi_k^j)_k\right)$, c'est à dire l'ensemble de fonction représentables par les ondelettes de niveau j. Pour les ondelettes classiques[24] la relation suivante est vérifiée $V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_N$. Et bien alors l'espace des détails, celui accessible par l'ondelette duale est Q_{j+1} le supplémentaire de V_j dans V_{j+1}, en d'autre terme, il représente toutes les informations, les *détails* contenues dans le niveau d'approximation V_{j+1} qui n'étaient pas prise en compte par V_j (à l'échelle j ce n'était que des détails). Les coefficients de la décomposition par l'ondelette duale sont Ainsi grâce à l'ondelette duale, il est possible de calculer les coefficients de détails d_k^j qui à chaque montée en résolution n'encode que l'information qui n'était pas contenue dans la décomposition en ondelette au niveau précédent.

Les deux visions ne sont pas rigoureusement les mêmes, la première représente ce qui est réalisé en pratique lors de la MRA, la seconde est la vision standard de la théorie des ondelettes. Toutefois les deux approches ont la même motivation, ne calculer et ne mettre en valeur à chaque niveau que ce qui est nouveau, ce qui n'était pas contenu dans les niveaux précédents.

Intuition sur les détails Lorsque l'on s'intéresse aux coefficients de détails d_k^j, l'indice j de *dilatation* fixe l'échelle analysée, c'est à dire la longueur d'onde analysée. Par exemple si $j=5$, les coefficients d_k^5 donne une information sur l'information portée par les longueurs d'onde de l'ordre de $2^{-5} = 1/32$. La variable k précise l'indice de la cellule analysée. Par exemple $\|d_1^5 0\| > \|d_5^5\|$ signifie que l'information portée par l'échelle $1/32$ est plus importante au voisinage de la case 10 qu'au voisinage de la case 5. De même si $\|d_7^j\| > \|d_{14}^{j+1}\|$ cela signifie qu'au voisinage de $x = \frac{7}{2^j}$ les longueurs d'ondes $\frac{1}{2^j}$ sont plus présentes que les longueurs d'ondes $\frac{1}{2^{j+1}}$. Pour se fixer les idées, c'est comme si la transformée de Fourrier n'avait qu'une vision globale du contenu en fréquence, quelle ne voyait que la

23. Ce n'est pas détaillé ici mais le lecteur se réfèrera à la théorie de l'interpollation...
24. C'est en tout cas vrai pour les ondelettes de Haar[16].

moyenne sur le domaine de la transformée en ondelette pour chaque longueur d'onde.

$$\|TF[f](\omega = 2^j)\|^2 \sim \|\sum_k d_k^j\|^2. \tag{2.27}$$

Grâce à cette notion de détaille la décomposition multi-échelle permet une description de la solution physique où l'apport à la solution de chaque échelle, de chaque distance typique est quantifié, les coefficients d_k^j décrivant l'information contenue dans les échelles de l'ordre de 2^{-j}.

B L'adaptation

B.i La compression par décomposition multi-échelle Pour compresser une fonction (ou une image comme dans le processus `jpg`), le processus est très simple, il suffit de fixer un seuil de compression $\varepsilon > 0$, de calculer les coefficients de détails de la fonction, puis d'omettre (en pratique de retirer de la mémoire) les coefficient dont la norme est inférieure à ε. En pratique le coefficient de compression dépend du niveau étudié puisque le volume des cellules mise en jeu chute avec le niveau j (un détail de 10^{-2} à moins d'importance s'il porte sur des cellules de taille 10 que sur des cellules de taille 10^{-5}). En pratique l'algorithme serait le suivant :

1. Calculer les coefficients d_k^j.
2. Pour chaque niveau j et chaque coefficient d_k^j : Si $|d_k^j| < \varepsilon_j = 2^{-j}\varepsilon$, alors $d_k^j \leftarrow 0$, les coefficients sont seuillés.

   Pour reconstruire au niveau de résolution souhaité : utiliser les détails jusqu'au niveau j le plus fin conservé puis utiliser le prédicteur pour interpoler jusqu'au niveau désiré. Le vocabulaire est heureusement choisit, pour compresser on omet les détails négligeables l'on conserve les détails importants.

B.ii L'adaptation de maillage et l'heuristique d'Harten L'adaptation de maillage, est une variation de la compression par décomposition multi-échelles précédemment décrite. C'est est une opération de compression de solutions physiques *prudente*. Les coefficients y sont seuillés de manière moins impitoyable : certains coefficients qui devraient être écartés par la compression sont malgré tout préservés. Ce choix se fait sur la base d'intuitions physiques, la plus connue étant *l'heuristique d'Harten*, introduite par Ami Harten [11], le père de la MRA. Elle stipule que même si un coefficient de détail d_k^j devrait être supprimé, si le niveau de détail du niveau supérieur (c'est à dire $d_{\lfloor k/2 \rfloor}^{j-1}$) est particulièrement élevé, par exemple qu'il est par deux fois supérieur au seuil ε_{j-1}, alors le coefficient doit être conservé. En d'autre terme, même si la compression considère l'information à l'échelle j négligeable, l'intuition physique pose son véto puisque les échelles supérieurs sont de grandes magnitudes et que cela présage que dans les pas de temps a venir les échelles seront nécessaires à la fidèle capture des phénomènes physiques simulés. Par exemple cela peut signifier qu'un front d'onde est en train d'arriver dans les zone étudiée, il faut donc que la simulation puisse en capter toute la richesse.

C Algorithmes de simulation numérique

C.i Algorithme général L'intégration de la MRA un algorithme de simulation physique se fait de la manière suivante : tous les n_{MRA} pas de temps (éventuellement être à chaque pas de temps), la

solution physique est adaptée selon le procédé explicité ci-dessus. Les détails précédemment omis devenant nécessaires sont obtenus grâce au prédicteur. Puis la simulation se poursuite à partir de cette grille de calcul adapté. L'économie de mémoire se fait puisque tous les détails ne sont pas stockés et l'économie de calculs puisque moins de cellules sont mises en jeu lors du déroulement de lalgorithme de simulation.

Par exemple, en dimension une, si une zone est adapté avec un niveau de détail de niveau 5, la densité de cellule sur lequel il faut réaliser des est de 2^5 et la densité mémoire est de $\sum_{j=0}^{5} 2^j$. Si le niveau le plus fin de la grille est par exemple 9, il y a un gain computationnel théorique de l'ordre de $2^{9-5} = 16$ et en terme de mémoire une économie de l'ordre de $\frac{\sum_{j=0}^{5}}{2^8} > 10$. Ce gain est exponentiel avec la dimension du problème.

C.ii Le cas volumes finis Il faut détailler ce qui signifie dans le cadre des volumes finis, *poursuivre la simulation sur la grille de calcul adaptée*. Le maillage adapté est constitué de cellules de tailles différents. Il s'agit alors pour chacune de ces cellules de prédire la valeur moyenne en son sein au pas de temps suivant, c'est à dire faire évoluer le maillage. Cela se fait en évaluant un flux à partir des valeurs de la solution aux interfaces de la cellule. Tout l'objet des volumes finis est : à partir de valeurs moyennes sur les cellules voisines, estimer les valeurs ponctuelles aux interfaces. La MRA amène alors une subtilité, les schémas volume finis usuels s'appuient sur des cellules de volumes similaires de l'ordre de Δx^d. Cependant,

C.iii Le débat sur la reconstruction

D Impact sur les solutions

D.i

D.ii

E Implémentations de la multi-résolution

E.i Méthodes usuelles

Les approches *patch-based*

Les approches *cell-based*

E.ii Le logiciel Samurai

Un tructure de données originale

2.3 Objectifs

Ce stage vise à évaluer des stratégies numériques modernes pour les équations d'advection-diffusion-réaction (ADR). Ces stratégies cherchent à contrecarrer les difficultés majeures des équations d'ADR : *des opérateurs de raideurs multiples* et *une ample variété d'échelles spatiales* (voir 2.2.2).

Premier objectif : éprouver les méthodes ImEx comme alternative au *splitting* pour la gestion des raideurs multiples Le *splitting* est une méthode très populaire pour intégrer des opérateurs de raideurs différentes. Cependant des méthodes alternatives dites implicites-explicites (ImEx) présentent des avantages tangibles, comme une meilleure montée en ordre et une gestion intrinsèque du couplage des erreurs. Ces méthodes ImEx ont donc été comparées sur un cas particulier ; sur le pan théorique (domaine de stabilité) et expérimental (étude de convergence).

Second objectif : étude de l'erreur apportée par la multi-résolution adaptative lorsque utilisée pour gérer les différentes échelles spatiales La multi-résolution adaptative est très étudiée dans la communauté scientifique et par l'équipe comme outils pour palier le caractère multi-échelles des solutions des équations d'ADR. Une analyse d'erreur théorique a été conduite sur une équation de diffusion résolue par une méthode numérique usuelle à laquelle s'ajoute une étape de multi-résolution. La décision de travailler sur l'équation de diffusion a été prise pour compléter le travail réalisé par l'équipe en [4] qui se centrait sur les opérateurs d'advection. L'analyse d'erreur met en lumière un mécanisme apporté par la multirésolution-adaptative pouvant dégrader l'ordre de convergence d'une méthode numérique classique. Une expérience numérique a par la suite été menée pour entreprendre d'observer ce phénomène dans un contexte pratique.

Ces deux objectifs permettent d'apporter différents éléments de compréhension sur le comportement des stratégies de simulation des équations d'ADR.

Chapitre 3

Contribution

Ce chapitre présente les travaux du stage. La première contribution analyse deux méthodes ImEx appliquées à l'équation de Nagumo (une équation de diffusion-réaction) et les compare à une méthode de *splitting* sur les questions de stabilité et de convergence. La seconde contribution étudie l'impact de la multirésolution adaptative sur un problème de diffusion. Cette étude débute avec un volet théorique (obtention de l'équation équivalente du schéma avec multirésolution adaptative) et poursuit par un volet expérimental (étude de convergence grâce au logiciel Samurai).

3.1 Étude de méthodes ImEx sur une équation de diffusion-réaction

L'objectif est de comparer la pertinence des méthodes RK implicites-explicites au *splitting* d'opérateurs traditionnel. Pour introduire cette première étude l'équation de Nagumo est d'abord présentée comme un excellent cas test pour éprouver les méthodes de résolution des équations d'advection-diffusion-réaction. Dans un second temps les méthodes ImEx utilisées sont détaillées. Par la suite leur stabilité est évaluée dans un contexte général ; puis en se focalisant sur l'équation de Nagumo, où elles sont comparée à une méthode de séparation d'opérateur classique (splitting de Strang). Ceci permet de valider la pertinence *a priori* de ces méthodes sur les équations de réaction-diffusion, et mène naturellement à une étude de convergence expérimentale.

3.1.1 L'équation de Nagumo

L'équation de Nagumo (ou FitzHugh-Nagumo) est issue de modèles de transmission de l'information nerveuse [8]. L'étude travaille sur la forme spatiale de l'équation [12] avec un terme de réaction cubique introduisant de la non-linéarité :

$$\partial_t u = \underbrace{D\partial_{xx} u}_{\text{diffusion}} - \underbrace{ku(1-u^2)}_{\text{réaction}}. \tag{3.1}$$

A Solutions Analytiques

L'équation admet des solutions propagatives sous la forme [1] :

$$u(x-ct) = \frac{e^{\sqrt{\frac{k}{2D}}\left((x-x_0)-ct\right)}}{1+e^{-\sqrt{\frac{k}{2D}}\left((x-x_0)-ct\right)}} \tag{3.2}$$

Avec : $c = \sqrt{\frac{kD}{2}}$ et x_0 le point de départ de l'onde.
Ainsi, le produit kD fixe la vitesse et le ratio $\frac{k}{D}$ la magnitude du gradient d'espace.

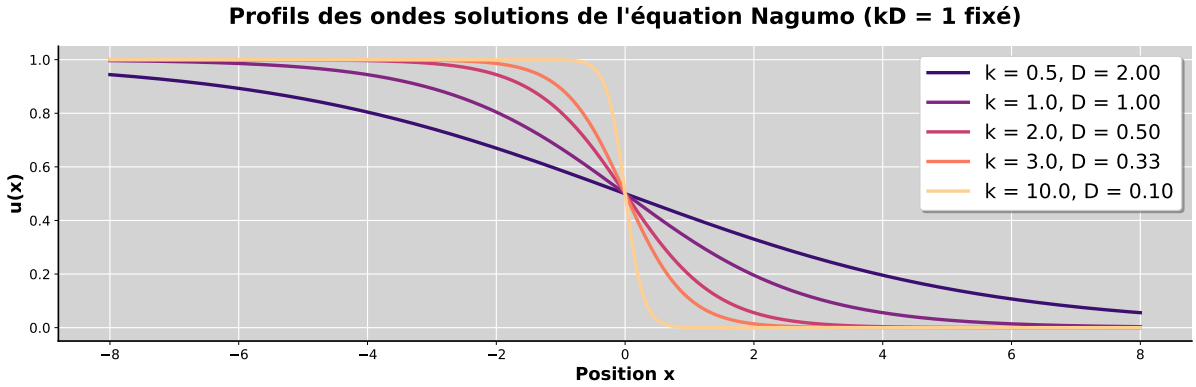


FIGURE 3.1 – Profils des ondes solutions de l'équation de Nagumo pour différents ratios k/D avec le produit $kD = 1$ fixé (c'est à dire à vitesse fixée). L'augmentation du ratio k/D accentue le gradient spatial.

B Analyse des opérateurs

Un analyse des deux opérateurs de l'EDP est nécessaire pur en saisir les enjeux. *L'opérateur de diffusion* est non-local et, discrétisé à l'ordre deux par n points et un pas Δx, les valeurs propres associées sont $\left\{\frac{2D}{\Delta x^2}\left(\cos\frac{p\pi}{n+1}-1\right) \mid p \in \{1,\dots,n\}\right\}$ [5], ainsi la raideur du terme de diffusion croit linéairement avec le coefficient de diffusion D et de manière quadratique avec la finesse du maillage $1/\Delta x$. En effet les valeurs propres sont négatives et :

$$\max_p |\cos\frac{p\pi}{n+1} - 1| \sim 2, \tag{3.3}$$

$$\min_p |\cos\frac{p\pi}{n+1} - 1| \sim \frac{1}{2}\left(\frac{\pi}{n+1}\right)^2. \tag{3.4}$$

1. C'est une sigmoïde, qui se propage à vitesse $\sqrt{\frac{kD}{2}}$.

Et donc :

$$\frac{\max_p |1-\cos\frac{p\pi}{n+1}|}{\min_p |1-\cos\frac{p\pi}{n+1}|} \approx n^2 \propto \left(\frac{1}{\Delta x}\right)^2. \tag{3.5}$$

Concernant *le terme de réaction*, en choisissant un état initial correspondant à 3.2, la solution reste entre 0 et 1. Ainsi le terme de réaction est local en espace, et ses valeurs propres sont comprises entre $-k$ et $2k$. En fonction de la valeur de u, la réaction se comporte localement (dans le temps et l'espace) comme une relaxation de temps caractéristique $\tau \sim \frac{1}{k}$ ou comme une explosion de temps caractéristique $\tau \sim \frac{1}{2k}$, en effet :

$$\text{Terme de réaction : } R(u) = ku(1-u^2), \tag{3.6}$$

$$\text{Valeurs propres de la réaction : } R'(u) = k(1-3u^2). \tag{3.7}$$

Pour les valeurs étudiés, $k \leq 20$, la réaction reste ainsi peu raide par rapport au "vraies" réactions chimiques (il faut avoir conscience de cette différence pour considérer ce cas test de la bonne manière, ici la réaction est le terme le moins raide alors que sur des "vrais" applications, ce n'est pas le cas).

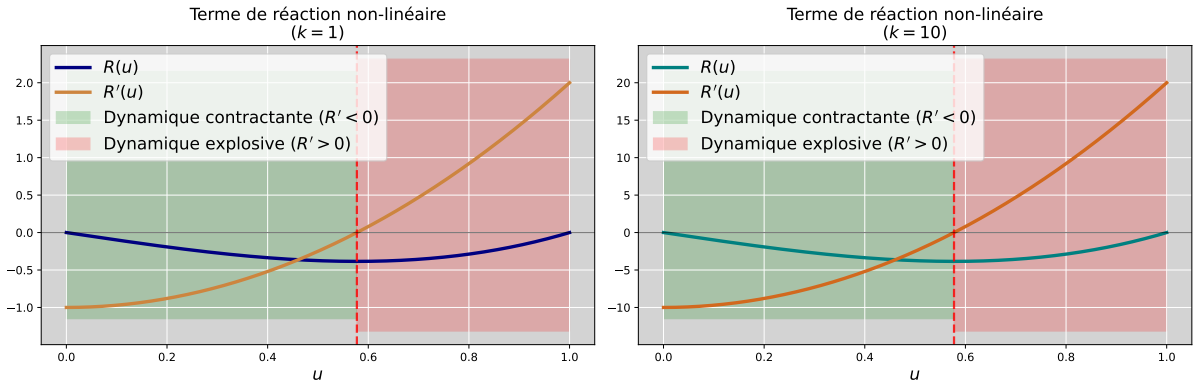


FIGURE 3.2 – Plage de valeurs du terme de réaction non-linéaire et de sa différentielle pour deux coefficients de réactions : $k = 1$ et $k = 10$.

C Conclusion sur l'équation de Nagumo

Ainsi l'équation de Nagumo, présente un terme de réaction [2], et un terme de diffusion. Cette équation fait émerger un front d'onde [3] et dispose de deux paramètre k et D pour piloter aisément les propriétés des solutions. Cela en fait donc une équation-test de choix pour étudier le comportement de diverses méthodes dédiées aux équations d'advections-réaction-diffusion.

2. à noter qu'il n'est pas raide, comparé aux termes de réaction rencontrés en combustion.
3. Cela permet de tester le comportement de la multi-résolution adaptative.

3.1.2 Les méthodes ImEx

Les méthodes ImEx étudiées sont les méthodes de Runge et Kutta additives (RK-ImEx ou RK-additive). Ces méthodes consistent à sommer plusieurs méthodes de Runge et Kutta appliquées chacune à un opérateur différent. *L'objectif est d'intégrer chaque opérateurs avec des méthodes RK différentes (implicites ou explicites), en accord les spécificité de chaque opérateur et cela, indépendamment des autres opérateurs.*

A Un exemple

Pour introduire la méthodes de Runge et Kutta additives, on commence par un exemple simple usant d'une méthode RK-ImEx d'ordre un. Cette ImEx naît de la conjugaison de deux méthodes Runge et Kutta à un étages (RK1). Cette méthode est notée ImEx111 [3]. Les méthodes RK1 servant de briques élémentaires à la RK111 sont : un schéma d'Euler explicite et un schéma d'Euler implicite. Soit une équation d'évolution faisant intervenir deux opérateurs :

- ◇ L'opérateur A^E se prêtant aux méthodes explicites (par exemple, un opérateur peu raide mais non local).
- ◇ L'opérateur A^I se prêtant aux méthodes implicites (par exemple un opérateur raide mais local).

L'équation cible serait alors de la forme :

$$\partial_t u = A^E u + A^I u. \tag{3.8}$$

A.i Résolution par approche monolithique En n'utilisant qu'une seule RK1 pour tout le problème (approche monolithique), la dynamique serait approchée d'une des façon suivante. En simulant avec un schéma d'Euler explicite monolithique, la méthode s'écrit alors :

$$u^{n+1} = u^n + \Delta t(A^E + A^I)u^n. \tag{3.9}$$

Mais si l'opérateur A^I est très raide, la stabilité risque d'imposer un pas de temps très restrictif risquant de rendre la méthode non viable. En résolvant avec un schéma d'Euler implicite monolithique, la méthode s'écrit :

$$u^{n+1} = \big(Id - \Delta t(A^E + A^I)\big)^{-1} u^n. \tag{3.10}$$

Mais si l'opérateur A^E rend l'inversion coûteuse ; par exemple s'il est non-local (impliquant la résolution d'un gros système au lieu de plusieurs petits systèmes), et/ou s'il est non linéaire (nécessite d'être réinverser à chaque pas de temps) ; alors cette méthode ne sera pas viable non plus.

A.ii Résolution par une méthode ImEx : une Runge et Kutta Additive Lorsque la méthode ImEx111 est choisie, lapproximation au pas de temps $n+1$ s'écrit en sommant une contribution issue de la méthode Euler explicite (RKE1) et une contribution issue de la méthode Euler implicite (RKI1) :

$$u^{n+1} = u^n + \Delta t(\underbrace{k_1}_{\text{RKE1}} + \underbrace{k_1'}_{\text{RKI1}}) \tag{3.11}$$

La contribution issue de la RKE1 appliquée à A^E s'écrit (Euler explicit) :

$$k_1 = A^E u^n. \tag{3.12}$$

La contribution issue de la RKI1 appliquée à A^I s'écrit (Euler implicit) :

$$k'_1 = A^I u^{n+1} \tag{3.13}$$

Ainsi :

$$\begin{aligned} u^{n+1} &= u^n + \Delta t(A^E u^n + A^I u^{n+1}), \\ \text{donc : } u^{n+1} - \Delta t A^I u^{n+1} &= u^n + \Delta t A^E u^n, \\ \text{et donc : } u^{n+1} &= (Id - \Delta t A^I)^{-1} \circ (Id + \Delta t A^E) u^n. \end{aligned} \tag{3.14}$$

Ainsi dans cette méthode seul lopérateur $Id - \Delta t A^I$ est inversé et les problèmes de raideurs sont résolus ; ce qui était lobjectif. Les traitements sur les opérateurs sont bien découplés lors de la résolution.

B Cadre mathématique général

Pour construire des méthodes plus complexes et d'ordres supérieurs introduisons le formalisme de [3] pour traiter les méthodes RK-additives. Ici, nous travaillons uniquement sur méthodes ImEx pour deux opérateurs mais théoriquement, il est possible de construire des méthodes ImEx pour traiter autant d'opérateurs que l'on le souhaite [13].

B.i Notations Une méthode ImEx additive est construite à partir d'une méthode implicite à s étages (une méthode DIRK et si possible SDIRK) et d'une méthode explicite à $s+1$ étages [4]. Pour uniformiser, le tableau de Butcher de la méthode implicite est complété par une ligne et une colonne de zéros afin que les deux méthodes s'écrivent comme si elles avaient le même nombre d'étages. Les tableaux de Butcher des deux méthodes s'écrivent alors :

Méthode RKE, $s+1$ étages :

$$\text{RKE : } \begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{b}^T \end{array} = \begin{array}{c|cccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ \tilde{c}_1 & \tilde{a}_{10} & 0 & 0 & \cdots & 0 \\ \tilde{c}_2 & \tilde{a}_{20} & \tilde{a}_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \tilde{c}_s & \tilde{a}_{s0} & \tilde{a}_{s1} & \tilde{a}_{s2} & \cdots & 0 \\ \hline & \tilde{b}_0 & \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_s \end{array} \tag{3.15}$$

4. Au besoin, la méthode explicite peut être à s étages, qui est un cas particulier d'une méthode à $s+1$ étages.

Méthode RKI (DIRK) s étages :

$$\text{RKI}: \begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cccccc} 0 & 0 & 0 & 0 & \cdots & 0 \\ c_1 & 0 & a_{11} & 0 & \cdots & 0 \\ c_2 & 0 & a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ c_s & 0 & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & 0 & b_1 & b_2 & \cdots & b_s \end{array} \tag{3.16}$$

où les coefficients $\tilde{a}_{ij}$, $\tilde{b}_i$, $\tilde{c}_i$ définissent la méthode explicite et les coefficients a_{ij}, b_i, c_i définissent la méthode implicite DIRK.

B.ii Schéma général d'une méthode RK-additive Une étape de la méthode RK-additive appliquée entre les pas de temps n et $n+1$ au système $\frac{du}{dt} = A^E u + A^I u$ s'écrit :

Calcul des approximations intermédiaires : Les calcul des approximations aux pas de temps intermédiaires $(u_i)_{i\in\{0,\dots,s\}}$ se fait grâce à la relation :

$$u_i = u^n + \Delta t \sum_{j=0}^{i-1} \tilde{a}_{ij} A^E u_j + \Delta t \sum_{j=0}^{i} a_{ij} A^I u_j, \qquad i = 0,1,\dots,s, \tag{3.17}$$

En initialisant $u_0 = u^n$.

Soit en mettant en lumière le caractère implicite de la méthode sur A^I :

$$(Id - \Delta t a_{ii} A^I) u_i = u^n + \Delta t \sum_{j=0}^{i-1} \left(\tilde{a}_{ij} A^E u_j + a_{ij} A^I u_j\right), \qquad i = 0,1,\dots,s \tag{3.18}$$

Calcul de l'approximation déinitive :

$$u^{n+1} = u^n + \Delta t \sum_{i=0}^{s} \tilde{b}_i A^E u_i + \Delta t \sum_{i=0}^{s} b_i A^I u_i \tag{3.19}$$

Cette formulation générale permet de construire des méthodes d'ordre élevé.

B.iii Ordre de convergence L'ordre d'une méthode RK-additive est évidemment borné par l'ordre des méthodes RK individuelles convoquées. Naturellement, cette borne n'est pas nécessairement atteintes ; des conditions d'ordre liant les coefficients des méthodes individuelles entre eux doivent être respectées. Le nombre de ses conditions augmente (très) rapidement avec l'ordre de la méthode et le nombre d'opérateurs à résoudre [13], le lecteur motivé se référera par exemple à [9].

3.1.3 Analyse de stabilité

L'objectif est d'appréhender la viabilité des RK-ImEx sur l'équation de Nagumo. Dans ce but, leur stabilité est étudiée. Dans un premier temps, une étude générale de la stabilité de des RK-ImEx est menée. Puis, l'étude de stabilité se centre sur l'application à l'équation de Nagumo. L'ensemble des codes utilisés pour évaluer numériquement et afficher les domaines de stabilités sont disponibles à l'adresse : https ://github.com/Ocelot-Pale/ImEx_stability_Nagumo.

A Étude de stabilité générale des RK-ImEx

Avec une méthode ImEx, les deux opérateurs de l'EDP sont découplés, c'est là l'intérêt. Cependant cela complexifie l'analyse usuelle de stabilisé. En effet la fonction de stabilité attend alors deux variables, le coefficient spectral Z_E associé à l'opérateur traité explicitement et le coefficient spectral Z_I associé à l'opérateur traité implicitement. Ainsi, pour chaque couple (Z_E, Z_I) d'indices spectraux, la fonction de stabilité prend une valeur différente, et comme les coefficients spectraux sont des nombres complexes, on ne peut plus visualiser d'un simple coup d'oeil le domaine de stabilité (comme en ... AFAIRE), puisque celui-ci se trouve dans un espace de dimension quatre [5].

A.i Calcul des fonction d'amplification Afin d'étudier la stabilité linéaire des méthodes, les fonctions d'amplifications ont été numériquement évaluées grâce à une fonction informatique. La démarche est la suivante :

1. Entrer les valeurs de (Z_E, Z_I) pour lesquelles la fonction de stabilité doit être évaluée
2. Simuler un pas schéma en partant de $u_0 = 1$ appliqué à une équation du type Dahlquist : $\partial_t u = \lambda_E u + \lambda_I u$
   (a) Construire toutes les approximations intermédiaires avec les valeurs
   (b) Construire l'approximation finale u_1
3. Évaluer la norme de U_1

Cette étude n'est pas détaillée ici par nécessité de concision, le lecteur intéresser pourra trouver les graphiques représentants les domaines de stabilité sur le Notebook en ligne.

B Étude de stabilité appliquée à l'équation de Nagumo

Nous allons particulariser la démarche suivante en la centrant sur l'équation de Nagumo. Cela vas nous permettre de comprendre comme se comportent les méthodes ImEx sur ce problème particulier.

B.i Valeurs propres mises en jeu Comme expliqué en 3.1.1 l'équation présente deux opérateurs :

- ◇ La diffusion dont le spectre d'étend de $\frac{-1}{L^2}$ à $\frac{-1}{\Delta x^2}$ (où L est la taille du domaine discrétisé).
- ◇ La réaction dont le spectre balaie continûment $-k$ jusqu'à $2k$

Pour restreindre l'analyse de stabilité il faut donc tracer le diagramme de stabilité des méthodes étudiées en prenant $Z_I \in \mathbb{R}^-$ et $Z_E \in [-k; 2k] \subset \mathbb{R}$ ce qui nous donne un espace à deux dimensions. Il faut ensuite placer des couples (Z_E, Z_I) correspondant. Lorsque l'on réalise se travaille nous trouvons les diagrammes suivant :

5. En effet la fonction de stabilité $R\mathbb{C} \times \mathbb{C} \to \mathbb{R}$ et $\dim \mathbb{C} \times \mathbb{C} = 4$.

B.ii Résultats Le lecteur est invité à prendre un peu de temps pour comprendre la logique de ces graphiques car ils sont très éclairants. Ces diagrammes permettent d'analyser respectivement la stabilité de la méthode ImEx222, de la méthode ImEx232, ainsi qu'à titre de comparaison, la stabilité d'une méthode RKE d'ordre 2 [6] et d'une méthode de splitting [7]. Chaque colonne représente l'analyse d'une méthode différente. La première ligne présente le domaine de stabilité en fonction des indices spectraux $Z_E \in \mathbb{R}$ et $Z_I \in \mathbb{R}^-$. Les points bleus représentent les couples d'indices spectraux intervenant dans la résolution de l'équation de Nagumo pour les paramètres d'équation choisis (D et k) et les paramètres de discrétisation retenus (Δt et Δx). La seconde ligne n'est qu'un zoom de la première autour de ces indices spectraux. La dernière colonne (splitting) présente une disposition différente, puisque les opérateurs sont totalement découplés. La première ligne correspond à la fonction de stabilité de la méthode explicite (avec un zoom autour des indices spectraux de la réaction). la seconde ligne représente la fonction de stabilité de la méthode implicite. Dans les deux cas, l'intervalle tracé en bleu représente la plage de valeurs d'indices spectraux balayés par chaque opérateur.

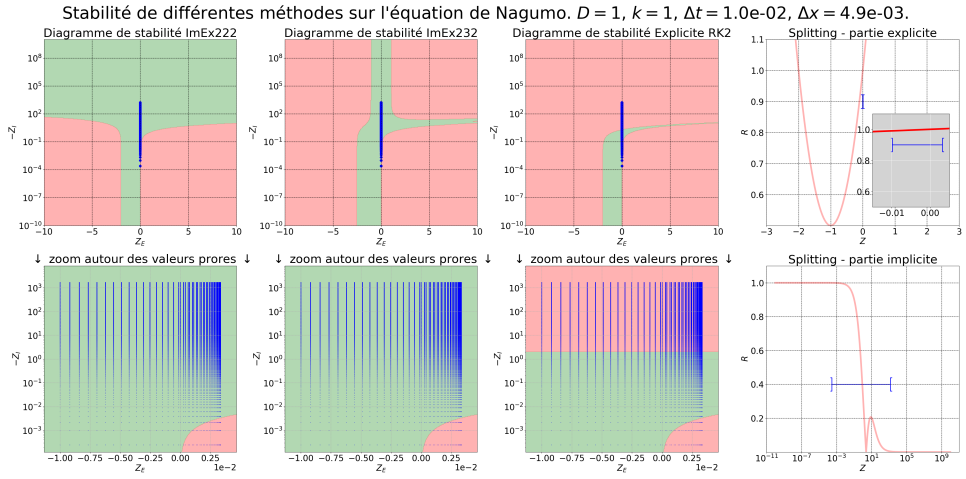
B.iii Analyse

Analyse générale Analysons les domaines de stabilité des figures en fig. 3.3, pour l'instant nous ignorons les marqueurs bleus sur les figures.

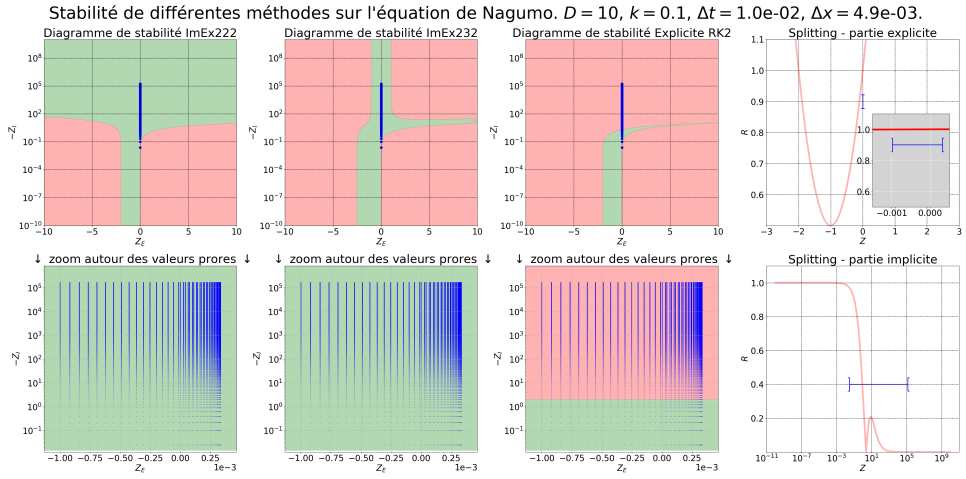
- **Méthode RKE2 :** En troisième colonne, le diagramme de stabilité d'une méthode explicite naïve RKE2, sert de référence. Le domaine de stabilité accepte des valeurs propres négatives de magnitude deux, ce qui est résultat classique des méthodes Runge et Kutta explicites d'ordre deux. Ainsi domaine de stabilité s'étend jusqu'à -2 selon l'axe portant Z_E tant que la valeur propre Z_I est négligeable. De même le domaine de stabilité s'étend jusqu'à -2 selon Z_I tant que la valeur propres Z_E est négligeable. Enfin il y a une zone intermédiaire quand Z_E et Z_I sont tous les deux de l'ordre de l'unité [8], où la raideur résultante est $Z_E + Z_I$.
- **Méthode ImEx232 :** En observant la seconde colonne, nous constatons que la méthode ImEx232 maintient un domaine de stabilité restreint (jusqu'à -2) selon l'axe Z_E, mais selon l'axe Z_I, le domaine de stabilité s'est étendu considérablement. C'est logique puisque la valeurs propre Z_E est explicitée, sont domaine pris seul n'a évolué, et la valeur propre Z_I peut être très raide (très négative) puisque la méthode explicite l'opérateur lié à Z_I.
- **Méthode ImEx222 :** Passant à la première colonne, le domaine de stabilité ImEx222 resemble beaucoup à celui de l'ImEx232. Seulement, le domaine de stabilité s'élargit considérablement selon Z_E, pourvus que Z_I soit assez grand. Cette propriété est remarquable, cela signifie que la méthode traite couple les raideurs dans sont traitement. Plus précisément, plus l'opérateur implicité est raide, plus l'opérateur explicité peut être raide [9].

Analyse selon les paramètres de l'équation k et D Analysons grace au graphiques fig. 3.3 la disposition des valeurs couples de valeurs propres mis en jeu par l'équation de Nagumo selon les

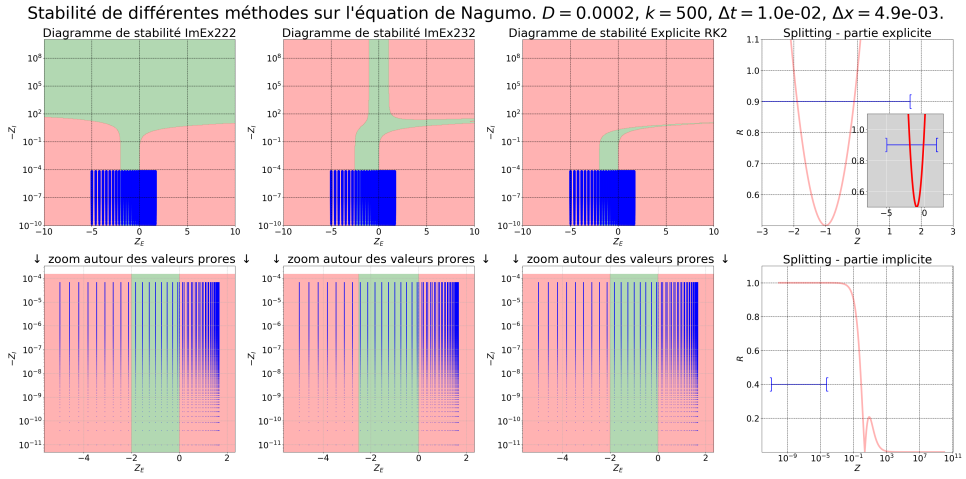
6. Celle apparaissant dans ImEx222.
7. Où l'on utilise les méthodes implicites et explicites de la méthode ImEx222 mais dans un contexte de splitting de Strang.
8. Attention à l'échelle logarithmique.
9. Cette analyse est partiellement erronée, nous verrons pourquoi au prochain paragraphe.



(a) Cas standard
D=1, k=1, dt=1.0e-03, dx=2.4e-03



(b) Cas diffusion plus raide, réaction moins raide
D=10, k=0.1, dt=1.0e-02, dx=4.9e-03



(c) Cas diffusion moins raide, réaction plus raide
D=2e-4, k=500, dt=1.0e-02, dx=4.9e-03

FIGURE 3.3 – Pour différents couples D et k, diagrammes de stabilité des méthodes ImEx et de référence sur l'équation de Nagumo.

paramètre k et D. Les paramètres de simulation : Δt et Δx sont fixés. Les jeux de valeurs choisis sont $(k,D) = (1,1)$, $(k,D) = (0.1,10)$, $(k,D) = (500, 2\,10^{-4})$. Le produit kD est maintenu égal à un, ainsi la vitesse de propagation est toujours la même. Ces couples de valeurs propres Z_E, Z_I mis en jeu par les opérateur de l'équation sont tracés en bleus [10].

- ◇ **Cas standard,** $(k,D) = (1,1)$ - fig. 3.3a :
  Dans ce cas, la raideur de la diffusion (Z_I) déstabilise la méthode RKE2 (on voit que de nombreux couples de v.p. entrent dans le zones rouges quand Z_I augmente). Pour ces valeurs de $(\Delta x, \Delta t)$ cette méthode n'est donc pas viable. C'est tout à fait normal, les méthodes imposent des pas de temps très restrictifs sur les problèmes de diffusion. En revanche, les méthodes ImEx sont tout à fait stable puisque, comme constaté précédemment, le domaine de stabilité s'étend infiniment quand $Z_I \to -\infty$. Le point notable est que certains couples de valeurs propres tombent malgré tout dans une zone instable (en bas à gauche). Mais cela n'est pas un problème car il s'agit de couples de valeurs propres ou la valeur propres [11] de l'opérateur de réaction (Z_E) est positive. Donc la méthode n'est pas instable au sens ou elle reflète simplement la dynamique explosive de la réaction. D'ailleurs si en se penchant sur le graphique de la partie explicite du splitting, on constante qu'il y a une zone (correspondant à Z_E positive) ou la fonction d'amplification est d'amplitude supérieure à un, le splitting reproduit donc fidèlement la dynamique de la réaction. Ce qui peut être un problème est l'inverse, pour les méthodes ImEx, il y a des couples de valeurs propres où Z_E est positif et où la fonction d'amplification est d'amplitude inférieure à un. Cela pourrait être un frein pour reproduire fidèlement la dynamique explosive de la réaction dans les zones concernées [12].
- ◇ **Cas diffusion raide, réaction peu raide,** $(k,D) = (0.1,10)$ - fig. 3.3b :
  Ici, $D = 10$ donc toutes les valeurs propres liées à la diffusion sont multipliées par 10 par rapport au cas précédent. De fait la méthode RK2E de référence présente des instabilités pour encore plus de couples de valeurs propres est n'est pas pas viable. Concernant les méthodes ImEx222 et ImEx232 elles sont stables, et cette fois-ci toutes les valeurs propres liées à la dynamique explosive de la réaction sont amorties.
- ◇ **Cas diffusion peu raide, réaction très raide**$(k,D) = (500, 2\,10^{-4})$ - fig. 3.3c
  Dans ce cas de figure, $k = 500$. La grande valeur du coefficient de réaction rend cette dernière très raide. Cela à pour effet de dilater selon l'axe des abscisse les indices spectraux puisque $Z_E \in [-500\Delta t, +1000\Delta t]$ alors que dans le cas $k = 1$: $Z_E \in [-\Delta t, 2\Delta t]$. Ici la méthode méthode explicite au sein des ImEx n'est plus stable pour la réaction, ainsi toutes les méthodes deviennent instables. Le splitting également devient instable car il utilise aussi la méthode RK2E pour la réaction. Le fait que la méthode explicite de l'ImEx soit instable pour l'opérateur explicité peu sembler un obstacle infranchissable, cependant ce n'est pas si simple. Pour illustrer ce point, étendons l'analyse avec le cas spécial en fig. 3.4, dans ce cas la réaction est toujours raide $k = 500$ mais la diffusion est également très raide car $D = 500$ [13] Alors la méthode ImEx222 de-

10. Pour les Z_I le spectre est discret, pour Z_E, le spectre est continu, il a donc fallut échantillonnés le long de l'axe Z_E
11. Dans cette section, nous identifions valeurs propres λ et indices spectraux $z = \lambda\Delta t$ puisque le pas de temps Δt est maintenu constant. Cette identification permet de discuter directement en termes de raideur des opérateurs.
12. Il n'est pas évident d'avoir *a priori* la bonne intuition car peut être que la diffusion calme en quelque sorte le caractère explosif de la réaction et qu'alors une fonction d'amplification d'amplitude < 1 est normal... Restons prudent sur cette analyse.
13. Jusqu'ici, la vitesse de propagation était la même dans tous les scénarios puisque kD était maintenu constant. Dans

vient stable, comme vu en B.iii, plus l'opérateur traité implicitement est raide, plus la méthode permet à l'opérateur traité explicitement d'être raide. C'est un cas remarquable ou le couplage intervenant au sein de la méthode ImEx la rend plus stable que le splitting!

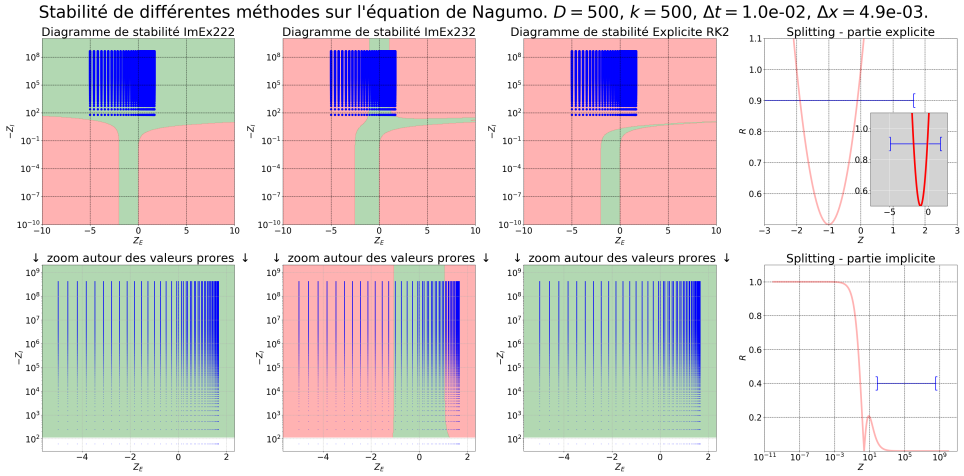


FIGURE 3.4 – Pour $k = 500$ et $D = 500$: diagrammes de stabilité des méthodes ImEx et de référence sur l'équation de Nagumo.

Analyse selon les paramètres de simulation Δt et Δx

3.1.4 Étude de la convergence

À présent que la stabilité des deux méthodes ImEx ont été comparées au *splitting* d'opérateur, il est naturel de poursuivre par une expérience numérique pour qualifier la convergence de chaque méthode et d'évaluer la pertinence des méthodes ImEx face au *splitting*.

A Expérience sur l'équation de Nagumo

A.i Présentation de l'expérience L'expérience est réalisée sur l'équation de Nagumo 1D à partir d'une solution initiale correspondant au profil de l'onde propagative de l'équation (voir 3.1.1). Succinctement, la simulation à lieu sur le domaine spatial $[-20, +20]$ entre $t = 0$ et $t = 3$. La grille spatial est divisée en 2^{13} cellules ce qui équivaut à un pas d'espace $\Delta t \approx 4.8\,10^{-3}$. Des conditions de Neumann homogènes et une vitesse de propagation adaptées permettent de maintenir le front d'onde au centre du domaine et de négliger les effets de bords afin de comparer à la solution analytique exacte d'onde propagative. Les erreurs sont calculés sur le domaine $[-5, +5]$ pour ce centrer sur l'étude du front d'onde.

A.ii Résultats Les résultats de l'expérience sont présentés en 3.5,

A.iii Conclusion

le scénario présenté ici, ce n'est plus le cas

B Expérience sur l'équation de Nagumo - Complément avec AMR

3.1.5 Conclusion

3.2 Obtention de l'équation équivalente d'une méthode de lignes avec multirésolution adaptative sur un problème de diffusion.

La multirésolution adaptative (MRA) a démontré une grande efficacité expérimentalement. Cependant, son impact sur la qualité des solutions obtenues n'est pas encore totalement compris mathématiquement. En [4], une étude de l'erreur introduite par la multirésolution adaptative a été menée. Cette étude se concentre sur les équations d'advection résolues par des schémas de type *one-step* [6] et compare l'équation équivalente des schémas avec MRA et celle des schémas sans MRA pour mettre en lumière les différences introduites par la multirésolution. La présente étude se place dans la continuité de cette démarche [14] et suit un cheminement similaire pour déterminer, grâce aux équations équivalentes, l'impact de la MRA sur une équation de diffusion résolue par une méthode des lignes d'ordre deux. La différence est donc double : d'une part la nature de l'opérateur étudié est différent (diffusion et non advection), d'autre part le type de schéma utilisés est également différent (méthode des lignes, ne couplant pas les erreurs en espace et en temps contre les schémas *one-step* traitant d'un coup d'un seul les erreurs en temps et en espace.). L'objectif est d'évaluer l'impact de la MRA dans des contextes variés. D'abord, le problème cible est présenté ainsi que le schéma de référence. Par la suite, les équations équivalentes du schéma de référence et du schéma avec multirésolution adaptative sont évaluées puis analysées. Enfin, les résultats théoriques obtenus sont éprouvés expérimentalement.

3.2.1 Cadre de l'étude

A Problème cible

Nous cherchons à résoudre le problème de diffusion suivant :

$$\partial_t u = D\partial_{xx} u. \tag{3.20}$$

Nous ignorons les problématiques de conditions de bords.

B Méthode des lignes utilisée

Pour résoudre cette équation aux dérivées partielles, nous utilisons une méthode des lignes. D'abord un schéma volume fini pour la discrétisation spatiale menant à l'équation semi-discrétisée suivante :

$$dtU(t) = \underbrace{\frac{D}{\Delta x}}_{\text{cellule}} \Big(\underbrace{\frac{U_{k+1} - 2U_k + U_{k-1}}{\Delta x}}_{\text{approx. gradients}} \Big) \tag{3.21}$$

14. À un niveau plus modeste.

Puis une méthode de Runge und Kutta explicite d'ordre deux sur lopérateur linéaire donne :

$$\begin{aligned} U_k^{n+1} &= U_k^n \\ &+ D\underbrace{\frac{\Delta t}{\Delta x}}_{\text{cellule}}\Big(\underbrace{\frac{U_{k+1}-2U_k+U_{k-1}}{\Delta x}}_{\text{approx. gradients}}\Big) \\ &+ \frac{1}{2}D^2\underbrace{\frac{\Delta t^2}{\Delta x^2}}_{\text{cellule}}\Big(\underbrace{\frac{U_{k+2}-4U_{k+1}+6U_k-4U_{k-1}+U_{k-2}}{\Delta x^2}}_{\text{approx. gradients}}\Big). \end{aligned} \tag{3.22}$$

Cela s'écrit sous la forme conservative suivante :

$$u_k^{n+1} = u_k^n + D\frac{\Delta t}{\Delta x}\Big(\Phi_{k+1/2}^n - \Phi_{k-1/2}^n\Big) + \Big(D\frac{\Delta t}{\Delta x}\Big)^2\Big(\Psi_{k+1/2}^n - \Psi_{k-1/2}^n\Big) \tag{3.23}$$

Avec :

$$\Phi_{k+1/2}^n = \frac{1}{\Delta x}(u_{k+1}^n - u_k^n), \tag{3.24}$$

$$\Phi_{k-1/2}^n = \frac{1}{\Delta x}(u_k^n - u_{k-1}^n), \tag{3.25}$$

$$\Psi_{k+1/2}^n = \frac{1}{\Delta x^2}\Big(\frac{1}{2}u_{k+2}^n - \frac{3}{2}u_{k+1}^n + \frac{3}{2}u_k^n - \frac{1}{2}u_{k-1}^n\Big),$$

$$\Psi_{k-1/2}^n = \frac{1}{\Delta x^2}\Big(\frac{1}{2}u_{k+1}^n - \frac{3}{2}u_k^n + \frac{3}{2}u_{k-1}^n - \frac{1}{2}u_{k-2}^n\Big).$$

C multirésolution adaptative

La multirésolution adaptative consiste à compresser le maillage, puis a effectuer les calculs sur le maillage compressé. Le schéma classique est le suivant :

1. Partir d'un état compressé au pas de temps n.
2. Calculer la solution au pas de temps $n+1$
3. Compresser de nouveau selon un seuil de compression ε grâce à une transformée multiéchelle.

Lors de la compression, la transformée multiéchelle représente la solution sur plusieurs niveaux de détails, du plus global, au plus local. Plus le niveau est profond, c'est à dire plus il est local, moins les détails associés portent d'information. L'opération de compression est réalisée en supprimant en chaque cellules, les niveaux dont la valeur des détails passent sous un certain seuil [15] [**postelApprox**]
.

Ce seuil ε n'est pas l'unique juge lors la compression, des heuristiques reposant sur la quantité d'information des détails de niveau supérieur sont utilisées pour ne pas seuiller systématiquement. L'objectif est en quelque sorte d'anticiper le besoin de détails [16]. La plus connue est l'heuristique d'Ami Harten [11].

Plusieurs stratégies existent pour réaliser le calcul d'un pas de temps à l'autre. Généralement, on estime les quantités au temps $n+1$ aux niveaux courants, à partir des quantité au niveau courant au

15. Typiquement $2^{\Delta l}\varepsilon$ où Δl

16. Même si la quantité d'information laisse entendre que certains détails pourraient être ignorés, l'intuition physique pose sont veto et force certains détails à être conservés par précaution, par exemple si un front front d'onde arrive.

temps n. Ensuite une opération de reconstruction-prédiction détermine le niveau de finesse requis de la solution au temps $n+1$. Il est également possible de calculer les quantités du temps $n+1$ au niveau courant, à partir des quantités au temps n **reconstruites à un niveau plus fin**. Bien que cela aie une faible efficacité computationnelle, cela réduirait les erreurs liés à la multirésolution selon la qualité du prédicteur employé comme discuté en [4]. Ici nous allons étudier théoriquement les erreurs dans un contexte similaire. Nous nous plaçons sur une cellule à un niveau de détail fixé, les flux sont calculés à partir de quantités reconstruites à un niveau de détails Δl plus fin. Le raisonnement et les ressources de calcul formel de [4] on été d'une aide précieuse.

Calcul du flux au travers de Δl niveaux :

Lorsque l'on applique le procédé de multirésolution, étant donné une cellule à un niveau de détail donné l, on cherche à faire évoluer la valeur à l'étape n vers la valeur à l'étape $n+1$. Pour ce faire, il faut évaluer les flux à partir les cellules voisines. Dès lors plusieurs choix s'offrent à nous. Où bien on utilise les cellules voisines à leurs niveaux courants, où bien on use de l'opérateur de reconstruction afin d'estimer les cellules voisines à des niveaux plus fins.

Dans un premier temps le stencil est choisi égal à 1. L'opérateur de prédiction d'un niveau à l'autre s'écrit alors :

$$\hat{u}_{2k}^{l+1} = +\frac{1}{8}u_{k-1}^{l} + u_{k}^{l} - \frac{1}{8}u_{k+1}^{l}, \tag{3.26}$$

$$\hat{u}_{2k+1}^{l+1} = -\frac{1}{8}u_{k-1}^{l} + u_{k}^{l} + \frac{1}{8}u_{k+1}^{l}. \tag{3.27}$$

Puis en notant $\hat{\hat{u}}_{(\cdot)}^{l+\Delta l}$ cet opérateur de prédiction itéré au travers de Δl niveaux [17] :

$$\begin{bmatrix} \hat{\hat{u}}^{(l+\Delta l)}_{2^{\Delta l}k-2} \\ \hat{\hat{u}}^{(l+\Delta l)}_{2^{\Delta l}k-1} \\ \hat{\hat{u}}^{(l+\Delta l)}_{2^{\Delta l}k} \\ \hat{\hat{u}}^{(l+\Delta l)}_{2^{\Delta l}k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} +1/8 & 1 & -1/8 & 0 \\ -1/8 & 1 & +1/8 & 0 \\ 0 & +1/8 & 1 & -1/8 \\ 0 & -1/8 & 1 & +1/8 \end{bmatrix}^{\Delta l}}_{\text{Matrice de passage } P \text{ pour } s=1.} \cdot \begin{bmatrix} u_{k-2}^{l} \\ u_{k-1}^{l} \\ u_{k}^{l} \\ u_{k+1}^{l} \end{bmatrix} \tag{3.28}$$

C.i Flux calculés au niveau le plus fin. On travaille sur une cellule au niveau courant l (cellules de tailles $\tilde{\Delta x} = 2^{\Delta l}\Delta x$) et lon reconstruit les états au niveau $l+\Delta l$ grâce à des flux flux au niveau fin, dont les gradients sont approximé par un pas Δx. La mise à jour conservative utilisée est donc

$$u_{k}^{n+1} = u_{k}^{n} + \underbrace{\frac{D\Delta t}{\Delta x 2^{\Delta l}}}_{\text{cellule}}\left(\hat{\hat{\Phi}}^{n}_{k+\frac{1}{2}} - \hat{\hat{\Phi}}^{n}_{k-\frac{1}{2}}\right) + \left(\underbrace{\frac{D\Delta t}{\Delta x 2^{\Delta l}}}_{\text{cellule}}\right)^{2}\left(\hat{\hat{\Psi}}^{n}_{k+\frac{1}{2}} - \hat{\hat{\Psi}}^{n}_{k-\frac{1}{2}}\right). \tag{3.29}$$

17. Au sens où l'on applique le prédicteur à des données déjà issues d'une prédiction.

Les flux sont évalués au *niveau fin* (facteurs $1/\Delta x$ et $1/\Delta x^2$ portés par les flux) à partir détats reconstruits $\hat{u}^{\,l+\Delta l}$:

$$\hat{\hat{\Phi}}^n_{k-\frac{1}{2}} = \frac{1}{\Delta x}\left(\hat{u}^{\,l+\Delta l}_{2^{\Delta l}k} - \hat{u}^{\,l+\Delta l}_{2^{\Delta l}k-1}\right), \tag{3.30}$$

$$\hat{\hat{\Phi}}^n_{k+\frac{1}{2}} = \frac{1}{\Delta x}\left(\hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)} - \hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)-1}\right), \tag{3.31}$$

$$\hat{\hat{\Psi}}^n_{k-\frac{1}{2}} = \frac{1}{\Delta x^2}\left(\tfrac{1}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}k+1} - \tfrac{3}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}k} + \tfrac{3}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}k-1} - \tfrac{1}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}k-2}\right), \tag{3.32}$$

$$\hat{\hat{\Psi}}^n_{k+\frac{1}{2}} = \frac{1}{\Delta x^2}\left(\tfrac{1}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)+1} - \tfrac{3}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)} + \tfrac{3}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)-1} - \tfrac{1}{2}\,\hat{u}^{\,l+\Delta l}_{2^{\Delta l}(k+1)-2}\right). \tag{3.33}$$

C.ii Écriture matricielle. Pour simplifier l'implémentation des calculs dans les codes de calculs formel, il est pertinent d'écrire se qui précède sous forme matricielle.
Pour les flux gauches :

$$\hat{\hat{\Phi}}^n_{k-\frac{1}{2}} = \frac{1}{\Delta x}\begin{bmatrix}0\\-1\\+1\\0\end{bmatrix}^{\top}\cdot\begin{bmatrix}\frac{1}{8} & 1 & -\frac{1}{8} & 0\\ -\frac{1}{8} & 1 & +\frac{1}{8} & 0\\ 0 & +\frac{1}{8} & 1 & -\frac{1}{8}\\ 0 & -\frac{1}{8} & 1 & +\frac{1}{8}\end{bmatrix}^{\Delta l}\cdot\begin{bmatrix}u^l_{k-2}\\u^l_{k-1}\\u^l_k\\u^l_{k+1}\end{bmatrix}, \tag{3.34}$$

$$\hat{\hat{\Psi}}^n_{k-\frac{1}{2}} = \frac{1}{\Delta x^2}\begin{bmatrix}-\frac{1}{2}\\+\frac{3}{2}\\-\frac{3}{2}\\+\frac{1}{2}\end{bmatrix}^{\top}\cdot\begin{bmatrix}\frac{1}{8} & 1 & -\frac{1}{8} & 0\\ -\frac{1}{8} & 1 & +\frac{1}{8} & 0\\ 0 & +\frac{1}{8} & 1 & -\frac{1}{8}\\ 0 & -\frac{1}{8} & 1 & +\frac{1}{8}\end{bmatrix}^{\Delta l}\cdot\begin{bmatrix}u^l_{k-2}\\u^l_{k-1}\\u^l_k\\u^l_{k+1}\end{bmatrix}. \tag{3.35}$$

Pour les flux droits :

$$\hat{\hat{\Phi}}^n_{k+\frac{1}{2}} = \frac{1}{\Delta x}\begin{bmatrix}0\\-1\\+1\\0\end{bmatrix}^{\top}\cdot\begin{bmatrix}\frac{1}{8} & 1 & -\frac{1}{8} & 0\\ -\frac{1}{8} & 1 & +\frac{1}{8} & 0\\ 0 & +\frac{1}{8} & 1 & -\frac{1}{8}\\ 0 & -\frac{1}{8} & 1 & +\frac{1}{8}\end{bmatrix}^{\Delta l}\cdot\begin{bmatrix}u^l_{k-1}\\u^l_k\\u^l_{k+1}\\u^l_{k+2}\end{bmatrix}, \tag{3.36}$$

$$\hat{\hat{\Psi}}^n_{k+\frac{1}{2}} = \frac{1}{\Delta x^2}\begin{bmatrix}-\frac{1}{2}\\+\frac{3}{2}\\-\frac{3}{2}\\+\frac{1}{2}\end{bmatrix}^{\top}\cdot\begin{bmatrix}\frac{1}{8} & 1 & -\frac{1}{8} & 0\\ -\frac{1}{8} & 1 & +\frac{1}{8} & 0\\ 0 & +\frac{1}{8} & 1 & -\frac{1}{8}\\ 0 & -\frac{1}{8} & 1 & +\frac{1}{8}\end{bmatrix}^{\Delta l}\cdot\begin{bmatrix}u^l_{k-1}\\u^l_k\\u^l_{k+1}\\u^l_{k+2}\end{bmatrix}. \tag{3.37}$$

3.2.2 Les équations équivalentes

A Calcul des équations équivalentes

Tout le calculs ont été réalisés grâce à la bibliothèque de calcul formel `Sympy` et les codes sont disponibles à l'adresse : https://github.com/Ocelot-Pale/etude_MR_RK2.

A.i Équation équivalente du schéma sans MRA Le calcul de l'équation équivalente sans MRA donne :

$$\frac{\partial u}{\partial t} = +D\frac{\partial^2 u}{\partial x^2} + \Delta x^2 \frac{D}{12}\frac{\partial^4 u}{\partial x^4} - \Delta t^2 \frac{D^3}{6}\frac{\partial^6 u}{\partial x^6} - \Delta t^3 \frac{D^4}{24}\frac{\partial^8 u}{\partial x^8}. \tag{3.38}$$

Le schéma de base est donc bien d'ordre deux en espace et en temps. En supposant une relation de stabilité du type $\lambda = \frac{D\Delta t}{\Delta x^2}$:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + \Delta x^2 \frac{D}{12}\frac{\partial^4 u}{\partial x^4} - \Delta x^4 \frac{D\lambda^2}{6}\frac{\partial^6 u}{\partial x^6} - \Delta x^6 \frac{D\lambda^3}{24}\frac{\partial^8 u}{\partial x^8}. \tag{3.39}$$

A.ii Équation équivalente du schéma avec MRA Dans le cas avec MRA, le logiciel de calcul formel donne l'équation équivalente :

$$\begin{aligned}\frac{\partial u}{\partial t} = {} & D\frac{\partial^2 u}{\partial x^2} \\ & - \frac{\Delta t}{2} D^2 \left(2^{2\Delta l} - 1\right)\frac{\partial^4 u}{\partial x^4} - \Delta t^2 \frac{D^3}{6}\frac{\partial^6 u}{\partial x^6} - \Delta t^3 \frac{D^4}{24}\frac{\partial^8 u}{\partial x^8} \\ & + 2^{2\Delta l}\frac{D\Delta x^2}{12}\frac{\partial^4 u}{\partial x^4} - 2^{2\Delta l}\frac{D\Delta l\Delta x^2}{4}\frac{\partial^4 u}{\partial x^4}\end{aligned} \tag{3.40}$$

Nous constatons que formellement le schéma est formellement d'ordre un en temps. Cela suggère donc que théoriquement, la multirésolution devrait faire perdre l'ordre de convergence temporelle de la méthode des lignes. Cependant en pratique la contrainte de stabilité impose une relation type CFL $\lambda = \frac{D\Delta t}{\Delta x^2} < 1/2$. Cela masque la perte en ordre [18] puisque cela mène à lélution équivalente :

$$\begin{aligned}\frac{\partial u}{\partial t} = {} & +D\frac{\partial^2 u}{\partial x^2} \\ & + \Delta x^2\left(\frac{2^{2\Delta l} D\lambda \frac{\partial^4 u}{\partial x^4}}{2} - \frac{2^{2\Delta l} D\Delta l \frac{\partial^4 u}{\partial x^4}}{4} - \frac{D\lambda\frac{\partial^4 u}{\partial x^4}}{2} + \frac{2^{2\Delta l} D\frac{\partial^4 u}{\partial x^4}}{12}\right) \\ & - \Delta x^6 \frac{D\lambda^3 \frac{\partial^8 u}{\partial x^8}}{24} - \Delta x^4 \frac{D\lambda^2 \frac{\partial^6 u}{\partial x^6}}{6}\end{aligned} \tag{3.41}$$

B Comparaison

Un ordre de précision a été perdu en temps. Nous essayons a présent de comprendre quel mécanisme mène à cette baisse de performances. Ma démarche consiste à comparer les équations équivalentes avec et sans multirésolution, **avant** d'appliquer la procéduer de Cauchy-Kovaleskaya.

18. Cela à été vérifié expérimentalement

B.i Sans MRA L'équation modifiée sans multirésolution, avant procédure de Cauchy-Kovaleskaya est :

$$\begin{aligned}\frac{\partial u}{\partial t} &= D\frac{\partial^2 u}{\partial x^2} \\ &+ \frac{1}{2}\underbrace{\left(D^2\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial t^2}\right)}_{\substack{\text{Se compense par}\\ \text{la procédure de}\\ \text{Cauchy-Kovaleskaya}}}\Delta t + \frac{D}{12}\frac{\partial^4 u}{\partial x^4}\Delta x^2 - \frac{1}{24}\frac{\partial^4 u}{\partial t^4}\Delta t^3 - \frac{1}{6}\frac{\partial^3 u}{\partial t^3}\Delta t^2.\end{aligned} \tag{3.42}$$

La méthode est bien d'ordre un, car à l'ordre un : $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$ et donc le terme $D^2\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial t^2}$ se compense au cours de la procédure de Cauchy-Kovaleskaya.

B.ii Avec MRA L'équation modifiée avec multirésolution, avant procédure de Cauchy-Kovaleskaya est :

$$\begin{aligned}\frac{\partial u}{\partial t} =& D\frac{\partial^2 u}{\partial x^2} \\ &+ \frac{\Delta t}{2}\underbrace{\left(2^{2\Delta l}D^2\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial t^2}\right)}_{\text{Ne se compensent plus.}} - \frac{\Delta t^3}{24}\frac{\partial^4 u}{\partial t^4} - \frac{\Delta t^2}{6}\frac{\partial^3 u}{\partial t^3} + \frac{\Delta x^2}{12}(1-3\Delta l)2^{2\Delta l}D\frac{\partial^4 u}{\partial x^4}\end{aligned} \tag{3.43}$$

Dans ce cas le terme en facteur du Δt ne se s'annule plus. En effet le terme $D^2\frac{\partial^4 u}{\partial x^4}$ est devenu au cours de la reconstruction $2^{2\Delta l}D^2\frac{\partial^4 u}{\partial x^4}$. En conséquence, la méthode perds un ordre de convergence temporel.

Ce mécanisme s'explique de la manière suivante : dans l'équation équivalente, le terme $\frac{\partial^2 u}{\partial t^2}$ apparaît indépendamment de la discrétisation spatiale [19]. La méthode des lignes initiale crée un terme *sur mesure* pour le compenser en approximant le terme spatial $D^2\frac{\partial^4 u}{\partial x^4}$. Cependant au cours du processus de compression-reconstruction, cette approximation est entachée d'un facteur $2^{2\Delta l}$. En d'autres termes le terme spatial construit pour compenser un terme temporel a été modifié par la multirésolution, alors que le terme temporel lui n'est pas affecté par la multirésolution. Ainsi, les deux termes ne se compensent plus et l'ordre est perdu.

C Conclusion sur le résultat obtenu grâce aux équations équivalentes

Il a été ici mis en lumière que la multirésolution-adaptative appliquée à méthode des lignes très simple peut théoriquement mener à un couplage des erreurs espace-temps polluant l'ordre initial de la méthode. En particulier l'étape de reconstruction-reconstruction altère des termes spatiaux qui ne compensent plus certaines erreurs temporelles.

19. Il emerge de la différence $u_k^{n+1} - u_k^n$ à k fixé.

3.2.3 Complément expérimental

A Présentation du cas test

Pour tâcher d'observer la perte d'ordre des expériences numériques ont été menées grâce au logiciel Samurai. Les expériences ont portées sur la simulation de l'équation de diffusion en 1D avec une solution initiale en forme de courbe de Gauß avec conditions de Dirichlet homogènes au bords. Pour limiter les effets de bords (qui non pris en compte dans l'analyse précédente) le domaine à été pris *grand* devant la largeur de la gaussienne initiale et le temps final assez petit pour que la diffusion n'atteigne pas le bord (qualitativement). L'erreur à été calculée comme la norme L^2 de l'erreur au temps final centrée sur la gaussienne[20]. Ce cas test est pertinent car il offre une solution lisse mais avec des gradients (donc l'AMR doit compresser à divers endroits) tout en offrant une solution analytique connue pour comparer l'erreur. En effet le problème posé sur $\mathbb{R}^+ \times \mathbb{R}$ (sans conditions de bord) :

$$\partial_t u = \partial_{xx} u, \tag{3.44}$$
$$u(t=0,x) = \frac{1}{\sqrt{4\pi a}} \exp\Big(-\frac{x^2}{4a}\Big).$$

Admet pour solution :

$$u(t=0,x) = \frac{1}{\sqrt{4\pi a(1+t)}} \exp\Big(-\frac{x^2}{4a(1+t)}\Big). \tag{3.45}$$

La solution numérique avec conditions de bord a été comparée avec la solution analytique sans conditions de bord, au vu de la taille du domaine et des temps de simulation cela ne devrait pas interférer de manière mesurable.

B Des défis expérimentaux

L'observation du mécanisme de perte d'ordre mis théoriquement à jour précédemment est une tâche ardue.

B.i Une expérience exacte impossible à reproduire Il n'est pas possible d'utiliser la méthode numérique utilisée dans l'étude théorique pour essayer de la valider expérimentalement. En effet, la méthode est une RK2 explicite, elle impose sur ce problème de diffusion une condition de stabilité du type $\Delta t \propto \Delta x^2$, ce qui en pratique donne $\Delta t \ll \Delta x$. De fait, la majorité des erreurs sont liés au pas d'espace "grand" devant le pas de temps, et donc si l'on fixe le pas d'espace pour faire converger la méthode en temps, l'erreur est déjà saturée en temps et on n'observe rien. C'est un classique de l'analyse numérique.

B.ii Une tentative infructueuse Suite à cette limite, l'expérience a été retentée avec une méthode voisine, une RK2, mais sans contrainte de stabilité. Le code de calcul Samurai a donc été relancé avec une méthode RK implicite d'ordre deux, plus précisément une SDIRK. Avec cette méthode, l'ordre

20. On calcul l'erreur autour de la gaussienne et pas sur tout le domaine car sinon elle serait artificiellement faible puisque la solution est quasi-nulle sur le reste du domaine qui a été pris grand pour éviter les effets de bord.

deux est observé et ce qu'importe les paramètre de l'AMR. Plusieurs hypothèses peuvent expliquer ce résultat :

1. Le phénomène n'est pas présent sur cette méthode implicite.
2. D'autres problèmes biaisent l'expérience (voir paragraphe suivant).
3. Les calculs ou les prémisses du développement théorique précédent des équations équivalentes sont faux.

B.iii Des biais multiples D'autres biais expérimentaux peuvent expliquer l'invisibilité du phénomène. Par exemple l'étude précédente ne prend pas en compte les conditions de bord. Une autre hypothèse est peut être que le maillage n'est compressé que localement ce qui n'altère que peut être pas la convergence globale. Enfin, dans les calculs théoriques précédents, il a été fait l'hypothèse que l'évaluation est faite au niveau le plus fin (que la solution est entièrement reconstruite pour l'évaluation des flux), ce qui n'est pas fait en pratique. Cette idée viens du fait qu'intuitivement, si l'on reconstruit jusqu'au niveau le plus fin les termes servant dans le calcul des flux, alors l'erreur devrait diminuer ; c'est ce que suggère [4]. Cependant cette fonctionnalité n'étant pas encore disponible dans le logiciel de calcul, l'expérience a été réalisée sans reconstruire ls flux au niveau le plus fin mais en prenant la valeur disponible au niveau courant de compression. Il semble peu probable que ce soit la cause de la non-observation du phénomène de perte d'ordre mais cela reste un biais potentiel. Enfin peut être qu'un bug s'est glissé dans mon implémentation mais cela semble peu probable puisque ce serait une erreur d'implémentation qu'il "améliore" l'ordre de convergence...

C ... DEPEND DE LA SUITE ...

3.2.4 Conclusion

3.3 Impact de la qualité de reconstruction des flux pour les problèmes diffusion avec AMR.

3.3.1 Présentation de l'étude

Cette troisième contribution s'inscrit directement dans la continuité de la précédente. L'objectif est ici d'étudier expérimentalement l'impact de la multirésolution et de ses modalités de mise en oeuvre sur les approximations numériques des problèmes diffusifs. Elle propose trois algorithmes d'AMR se distinguant par la qualité de lévolution des flux numériques.

D'abord les différences entre les trois algorithmes d'AMR sont mises en lumière, puis une première expérience numérique reprenant le schéma numérique de la contribution précédente (VF + ERK2) est réalisée. Face à des résultats surprenants, l'hypothèse de l'émergence d'instabilités résiduelles pour certains algorithmes a été émise. Cependant cette hypothèse semble être invalidée par une étude de stabilité linéaire réalisée grâce à Sympy. La contrainte de stabilités imposées par la méthode ERK2 n'avait jusqu'ici permises d'observer que des solutions convergées en temps (erreur spatiale dominante). Restait inconnu l'impact de la méthode d'évaluation des flux numériques dans un contexte ou les erreurs temporelles restent dominantes. Une seconde expérience à alors été réalisé avec la méthode explicite stabilité ROK4 au lieu de la méthode ERK2, permettant d'accéder à des pas de temps plus grands.

3.3.2 Présentation des trois algorithmes

La différence entre les trois algorithmes étudiés se trouve dans la manière de conjuguer les volumes finis [14] et la multirésolution adaptative. Le paradigme des volumes finis nécessite le calcul d'un *flux*, qui requiert lui l'évaluation de deux termes dépendant de la solution aux *interfaces* amont et avales des cellules. À cellule k fixée, ces termes sont notés Φ^+ et Φ^-. Le problème est que les volumes finis n'approximent que les valeurs moyennes sur les cellules et non les valeurs ponctuelles aux interfaces, nécessaires au calcul des flux. Les schéma volumes finis approximent alors les termes de flux comme fonction des valeurs moyennes sur les cellules voisines de l'interface. À titre d'exemple, pour la diffusion, le flux en amont de la cellule k est calculé comme $\frac{u_k - u_{k-1}}{\Delta x}$.

La MRA rend l'explication précédente non-univoque, la manière de calculer les flux peut être définie de plusieurs façon et c'est ceux en quoi les trois algorithmes étudiés diffèrent.

Comme la MRA défini plusieurs grilles de pas $\Delta x, 2\Delta x, 4\Delta x, ...$ le choix de la grille sur laquelle choisir cellules voisines intervenant dans le calcul du flux est ambigu. En effet, à niveau de détail l fixé, doit-ont évaluer les termes de flux à partir des cellules voisines du niveau $l, l+1, l+2...$ (voir le schéma en fig. 3.9) ? Le premier algorithme étudié (la référence en MRA), consiste à évaluer les termes de flux à partir des voisins au même niveau que la cellule étudiée. C'est à dire que si la cellule est de niveau l, les voisines de l'interface sont choisit également au niveau l. Cela revient à résoudre localement l'EDP au niveau courant de la grille, il est la norme en MRA car ne ne requiert aucun calcul supplémentaire, les valeurs sur la grille au niveau l sont directement accessibles. Le second algorithme consiste à systématiquement les valeurs de la grille la plus fine. Intuitivement c'est le plus précis, mais cela peut s'avérer très coûteux car la grille plus fine n'est pas directement accessible. Par exemple si la MRA à choisir une grille plus grossière de 4 niveau par rapport à la grille la plus fine, il faut reconstruire les valeurs au travers de 4 niveau. Enfin le troisième algorithme est un compromis

entre les deux méthode précédents, il consiste à utiliser calculer les flux à partir des valeurs un niveau en deçà du niveau courant, pour gagner un peu en précision sans pour autant s'exposer à des coûts prohibitifs. En [4], la différence théorique entre les deux premiers algorithmes a été étudiée sur des problèmes d'advection linéaires, une comparaison expérimentale entre les trois algorithmes à été réalisée sur des problèmes d'advections linéaires et non-linéaires.

3.3.3 Expérience numérique avec une méthode Runge et Kutta explicite

La première expérience numérique a donc été faire sur l'équation de diffusion avec le même schéma numérique que dans l'étude théorique précédente (*cf* 3.2) ; c'est à dire une discrétisation spatiale d'ordre deux du Laplacien intégré en temps avec une méthode Runge et Kutta explicite d'ordre 2. Il pourrait sembler plus astucieux d'utiliser une méthode implicite pour s'affranchir des problèmes de stabilité, cependant l'inversion d'un système linéaire couplé à la reconstruction des flux à des niveaux plus fin (non-standard, algos 2 et 3) est très difficile techniquement et aurait ralenti l'étude. A cause de la contrainte de stabilité $\Delta t \propto \Delta x^2$, seules des solutions convergées en temps (erreur spatiale dominante) ont pu être observées (cette contrainte sera levée par la suite en **??**).

A Résultats numériques

Il résulte expérimentalement que dans ce contexte, étonnamment, l'algorithme un (le plus grossier) offre la plus faible erreur et que plus l'algorithme reconstruit finement les flux plus l'erreur est grande. A titre d'exemple, sur une solution initiale gaussienne, avec une seuil de compression $\varepsilon = 10^{-4}$ et un maillage présentant jusqu'à 4 niveau de finesse, les erreurs L^2 au temps final $T_f = 1$ sont les suivantes :

Algo n^o	Niveau d'évaluation des flux	Erreur L^2
1	Courant	1×10^{-4}
2	Inférieur direct	2×10^{-4}
3	Plus fin	3×10^{-4}
référence	Sans AMR	2×10^{-5}

Bien sûr, l'algorithme sans AMR reste celui avec la plus petite erreur.

B Analyse et hypothèses

Ce résultat est assez surprenant puisqu'on s'attendrait à ce qu'une reconstruction plus précise du flux donne de meilleurs résultats. Dès lors plusieurs hypothèses peuvent être émises :

1. Le fait de reconstruire *déstabilise* la méthode.
2. L'usage de cellules plus grandes pour évaluer les flux augmente le caractère diffusif du schéma.
3. Peut-être que cet effet n'a lieu que sur des solutions où l'erreur spatiale domine et que les résultats seraient différents lorsque l'erreur temporelle reste dominante.

La première hypothèse est éprouvée dans la prochaine section 3.3.4 et une méthode stabilisée est utilisée en **??** pour explorer numériquement la troisième hypothèse.

3.3.4 Analyse de stabilité

Cette partie étudie la stabilité des différents algorithmes d'AMR à l'aide une analyse type Fourrier - Von Neumann. Ces résultats ont été obtenus en adaptant le code de calcul formel ayant fourni les équations équivalente en 3.2. Une petite interface graphique est disponible à cette adresse https://github.com/Ocelot-Pale/etude_MR_RK2/blob/main/code_2/stabilite_AMR_plotly.html . ... A FINIR ...

A Conclusion

Il semble que le résultat numérique surprenant précédemment présenté de vienne pas de problèmes de stabilités liées au reconstructions, en effet si ces problèmes de stabilités apparaissaient, ils devraient être plus prononcés sur la méthode où la reconstruction se fait au niveau courant. Alors que c'est l'inverse qui est observée.

3.3.5 Expérience numérique avec une méthode explicite stabilisée

Pour observer ce qui se produit lorsque l'erreur n'est pas saturée en espace mais que l'erreur temporelle intervient également, le logiciel Ponio [21] a été couplé à Samurai. Il permet d'utiliser facilement des méthodes d'intégration en temps complexe. Grâce Ponio, lexpérience précédent à été réitérée en remplaçant la méthode ERK2 par la méthode stabilisée ROCK4 [2]. Cette méthode reste explicite (ce qui permet grâce à Samurai d'étudier facilement les différentes façons d'évaluer les flux) tout en assouplissant significativement la contrainte de stabilité.

A Résultats numériques

Les résultats sont présentés en fig. **??**. Chaque graphique correspond à un paramétrage différent de l'AMR. Les colonnes correspondent à différents niveau de compression : $\varepsilon \in \{10^{-5}, 10^{-4}, 10^{-3}\}$. Les lignes correspondent à différents niveau de profondeurs du maillage : $\Delta l^{max} \in \{1,2,4\}$. Sur chaque graphique l'erreur L^2 tracée en fonction du pas de temps pour chaque méthode de calcul du flux numérique. La courbe blanche correspond à la solution sans AMR sur la grille la plus fine, elle sert de référence. La courbe marron (`flux_recons_lvl=0`) est celle de l'algorithme 1, standard en AMR - reconstruction des flux avec les cellules du niveau courant. La courbe verte (`flux_recons_lvl=-1`) représente lalgorithme 2 où les cellules servant à l'évaluation des flux sont systématiquement reconstruites au niveau le plus fin. Enfin la courbe bleu (`flux_recons_lvl=1`) représente l'algorithme 3 où la solution est reconstruite d'un niveau pour évaluer le flux numérique. Si le paramètres de compression est très restrictif $\varepsilon = 10^{-5}$ (colonne de gauche), les 4 algorithmes sont équivalents, en effet il n'y a presque jamais d'adaptation de maillage, car le seuil de compression est trop restrictif. Pour des paramètres de compression plus raisonnables ($\varepsilon = 10^{-4}$ et $\varepsilon = 10^{-3}$), les observations sont plus riches. Lorsque les erreurs temporelles de dominent pas, les trois algorithmes d'AMR améliorent la convergence - ce qui est surprenant. Cependant leur convergence s'arrête plus tôt, du fait des erreurs spatiales liées à l'AMR et pour des pas de temps assez petit, le schéma sans AMR demeure meilleur. Pour un pas de temps où l'erreur en temps domine, en figure 3.10 est tracé l'erreur au temps final pour chacune des méthodes est tracé et il semble effectivement que les méthodes avec AMR lissent "mieux" que la méthode sans AMR qui semble un peu plus bruité.

21. https://github.com/hpc-maths/ponio

Pour des pas de temps menant à une erreur spatiale dominante, les résultats sont similaires à ce qui avait été observées, plus les flux sont évaluées à partir de reconstructions fines, plus le plateau de saturation en erreur spatiales est important.

B Conclusion

Deux conclusions sont à tirer de cette expérience :

1. L'AMR tend à faire chuter plus rapidement l'erreur temporelle qu'avec une méthode non adaptée spatialement, et cela indépendamment de la méthode d'évaluation du flux numérique.
2. L'AMR mène à un plateau de convergence plus élevée que sans AMR (l'erreur spatiale domine). Ce qui est étonnant est que cette saturation de l'erreur arrive d'autant plus vite que les flux sont évaluées à partir de reconstructions fines ; même observation qu'avec la méthode ERK2.

Ces résultats ne valent a priori que sur une équation de diffusion "pure" et la relation d'ordre entre les erreurs qui parait incohérente n'est peut être que le résultat d'un "heureux effet lissant" qu'apporte l'AMR et qui est plus prononcé encore si l'on évalue les flux au niveau courant (algorithme 1). La prochaine section se propose donc de reprendre l'expérimentation sur une équation de diffusion-réaction, présentant une onde progressive ce qui s'approche un peu plus d'un contexte de simulation industrielle réaliste. Dès lors, il ne s'agit plus pour le schéma de simplement lisser mais également suivre la dynamique d'un front d'onde ! Peut-être que les résultats seront différents.

3.3.6 Extension sur une équation de diffusion-réaction

Cette section vise à observer l'impact du niveau de reconstruction des flux quand l'opérateur de diffusion est couplé à un autre opérateur. L'équation de diffusion est remplacée par l'équation de Nagumo (voir 3.1.1) qui ajoute un terme de réaction. Des ondes progressives sont solution de cette équation, le schéma numérique donc doit être en mesure de suivre la dynamique du front d'onde.

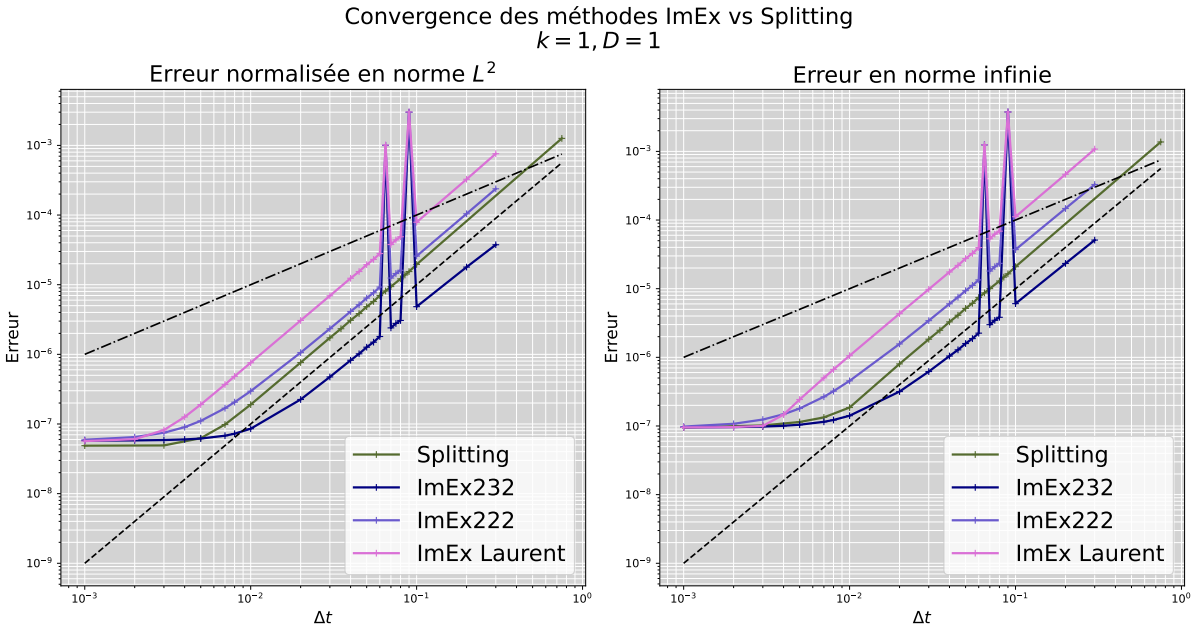
A Résumé de l'expérience

L'expérience à été réalisée dans des conditions similaires à l'expérience numérique **??**. C'est à dire :

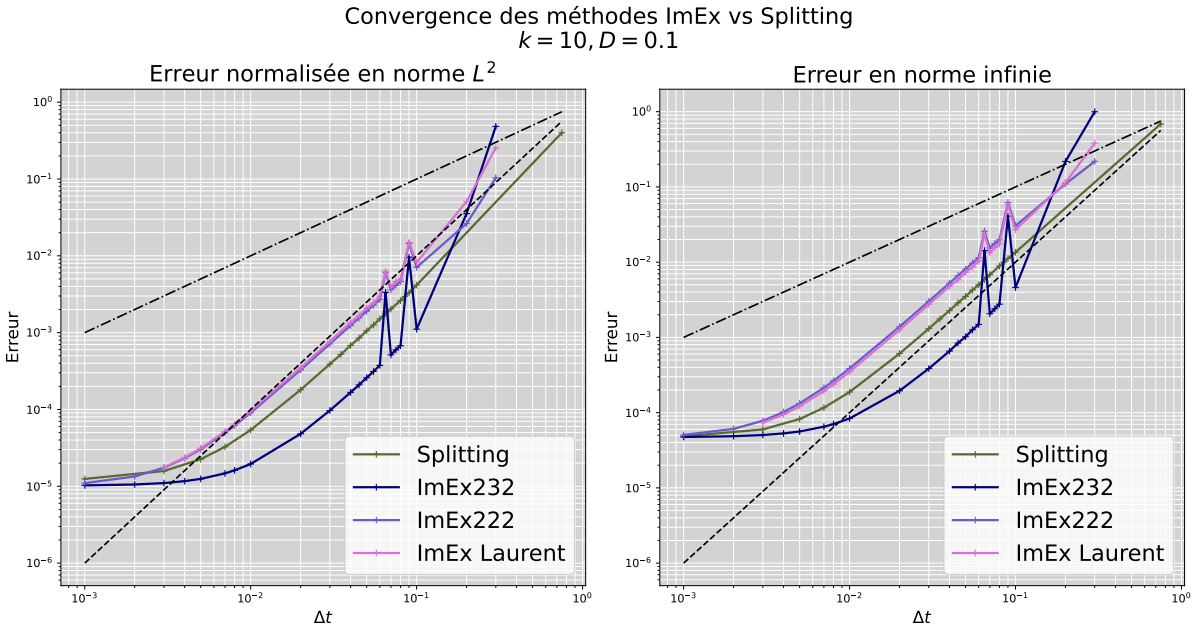
- ◇ Profil de l'onde propagative comme état initial.
- ◇ Domaine étendu avec conditions de Neumann aux limites pour limiter les effets de bord

La seule différence majeur réside dans le remplacement de la méthode Runge et Kutta ImEx par un schéma de splitting avec une méthode stabilisée explicite pour la diffusion (ROCK 2 [2]). Ce choix découle de la nécessité d'éviter toute inversion de système linéaires lorsque l'on reconstruit les flux à partir d'approximation fines car limplémentation prendrait actuellement trop de temps.

B Résultats de l'expérience


Convergence des méthodes ImEx vs Splitting
$k = 1, D = 1$
Erreur normalisée en norme L^2
Erreur en norme infinie
Erreur
Δt
Splitting
ImEx232
ImEx222
ImEx Laurent

(a)


Convergence des méthodes ImEx vs Splitting
$k = 10, D = 0.1$
Erreur normalisée en norme L^2
Erreur en norme infinie
Erreur
Δt
Splitting
ImEx232
ImEx222
ImEx Laurent

(b)

FIGURE 3.5

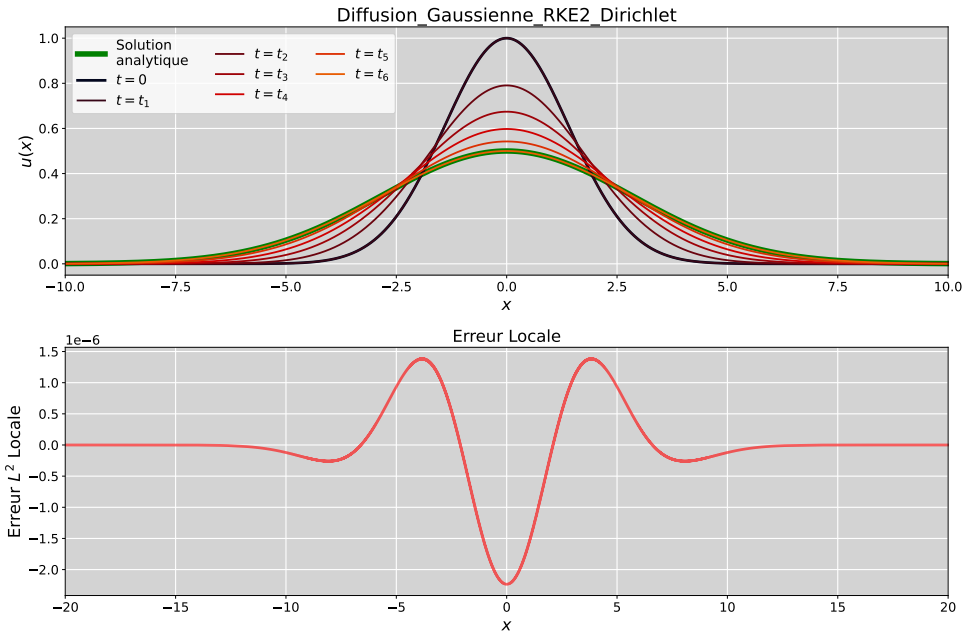


FIGURE 3.6 – Illustration d'une simulation du cas test 3.44 avec conditions de Dirichlet homogène et affichage de l'erreur locale au temps final.

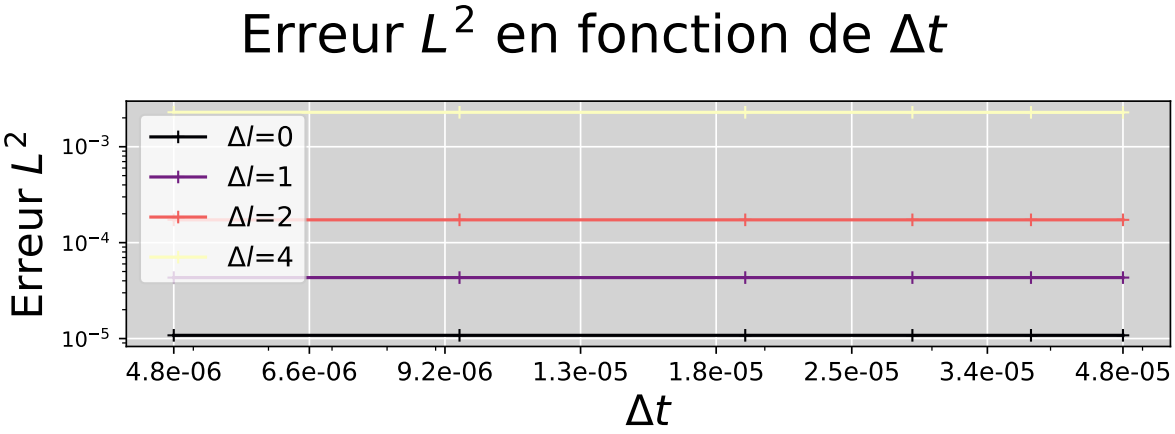


FIGURE 3.7 – Saturation de la convergence temporelle avec une méthode RKE2 sur l'équation de diffusion. L'erreur Lš stagne malgré la diminution du pas de temps, illustrant la domination de l'erreur spatiale due à la contrainte CFL $\Delta t \propto \Delta x^2$.

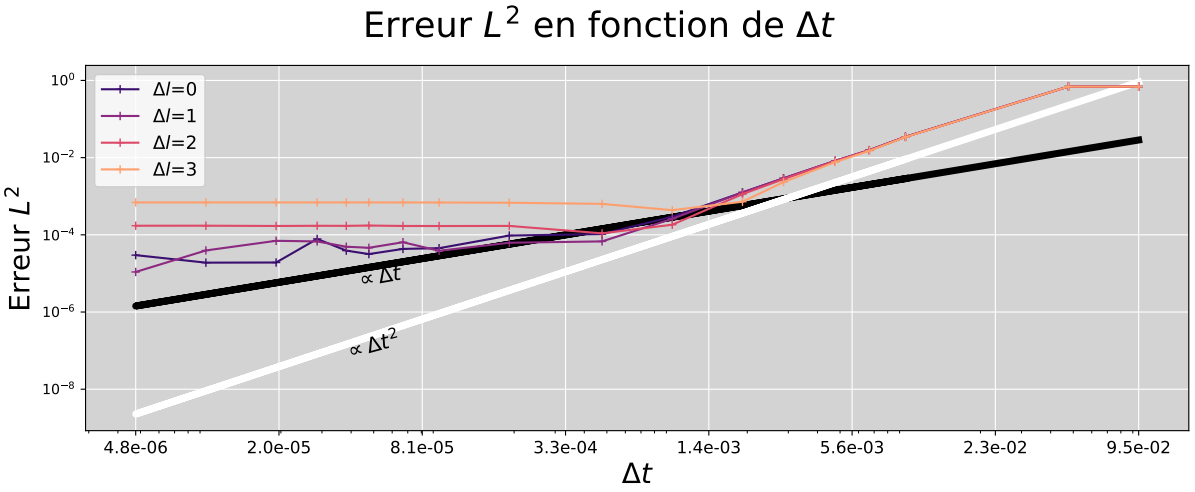


FIGURE 3.8 – Convergence temporelle d'ordre 2 avec une méthode SDIRK-RK2 sur l'équation de diffusion. L'ordre théorique est préservé indépendamment des paramètres MRA, contrastant avec nos prédictions théoriques établies pour les méthodes explicites.

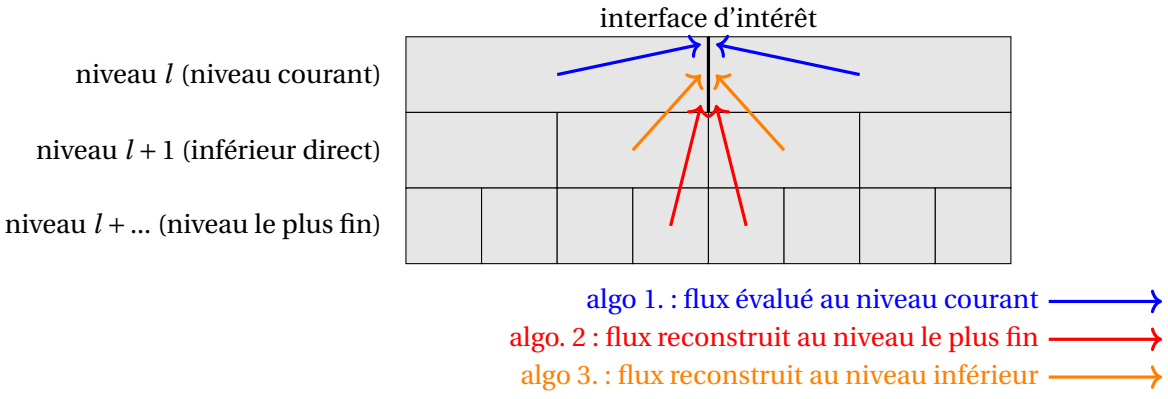


FIGURE 3.9 – Illustration des trois algorithmes évalués. L'algorithme 1 en bleu calcul le flux à partir des cellule de la grille au même niveau que l'interface étudiée. L'algorithme 2 reconstruit les valeurs de la solution sur la grille de niveau inférieur. L'algorithme 3 reconstruit au niveau le plus fin possible. Plus l'algorithme reconstruit finement, plus les valeurs moyennes sont données sur des cellules petites et plus cela s'approche d'une valeur "ponctuelle".

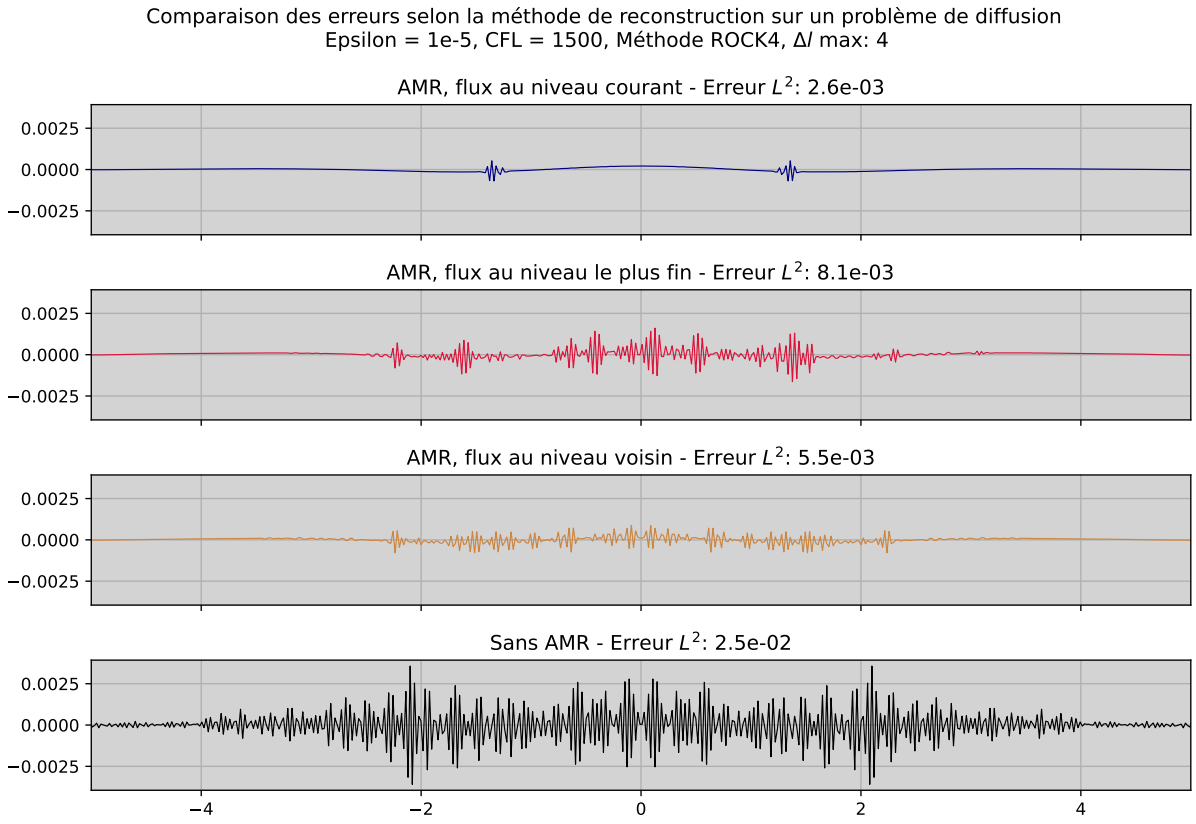


FIGURE 3.10 – Erreur au pas de temps final pour les différentes méthodes, avec une constante CFL de diffusion $D\Delta t/\Delta x^2 = 1500$, correspondant à des solutions où l'erreur temporelle reste dominante.

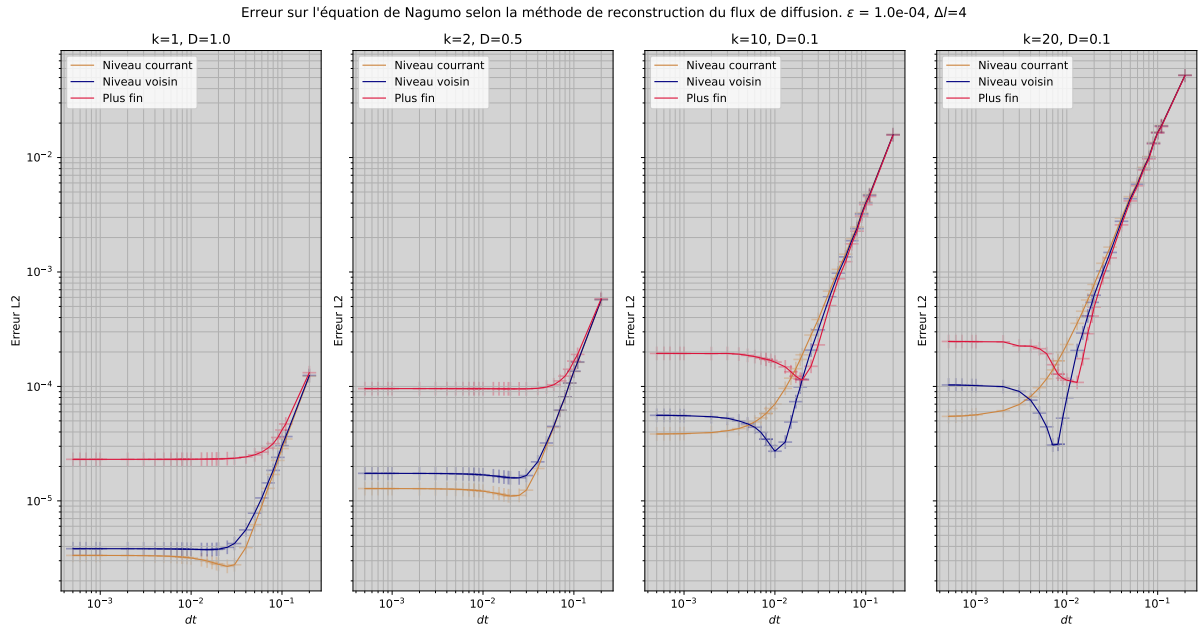


FIGURE 3.11 – Courbes de convergence de chaque méthode d'AMR pour différents paramètres de l'équation. Plus k est élevé, plus le profil de l'onde est raide et plus la réaction domine. La célérité de l'onde est néanmoins identique pour chaque jeu de paramètres puisque le produit kD reste constant d'une expérience à l'autre.

Chapitre 4

Conclusion

Bibliographie

[1] Assyr ABDULLE. “Fourth Order Chebyshev Methods with Recurrence Relation”. In : *SIAM Journal on Scientific Computing* 23.6 (2002), p. 2041-2054. DOI : 10.1137/S1064827500379549.

[2] Assyr ABDULLE et Alexei A. MEDOVIKOV. “Second order Chebyshev methods based on orthogonal polynomials”. In : *Numerische Mathematik* 90.1 (2001), p. 1-18. DOI : 10.1007/s002110100292.

[3] Uri M. ASCHER, Steven J. RUUTH et Raymond J. SPITERI. “Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations”. In : *Applied Numerical Mathematics* 25.2 (1997). Special Issue on Time Integration, p. 151-167. ISSN : 0168-9274. DOI : https://doi.org/10.1016/S0168-9274(97)00056-1. URL : https://www.sciencedirect.com/science/article/pii/S0168927497000561.

[4] BELLOTI et al. “Modified equation and error analyses on adaptative meshes for the resolution of evolutionary PDEs with Finite Volume schemes”. In : (2025).

[5] M. BOUCHET. *Le laplacien discret 1D*. Notes de cours. Agrégation externe de mathématiques 2019-2020, Leçons 144, 155, 222, 226, 233. ENS Rennes, 2020. URL : https://perso.eleves.ens-rennes.fr/~mbouc892/lapdisc1d.pdf.

[6] V. DARU et C. TENAUD. “High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations”. In : *Journal of Computational Physics* 193.2 (2004), p. 563-594. ISSN : 0021-9991. DOI : https://doi.org/10.1016/j.jcp.2003.08.023. URL : https://www.sciencedirect.com/science/article/pii/S0021999103004327.

[7] Max Pedro DUARTE. “Méthodes numériques adaptives pour la simulation de la dynamique de fronts de réaction multi-échelle en temps et en espace”. 2011ECAP0057. Thèse de doct. 2011. URL : http://www.theses.fr/2011ECAP0057/document.

[8] Richard FITZHUGH. “Impulses and Physiological States in Theoretical Models of Nerve Membrane”. In : *Biophysical Journal* 1.6 (1961), p. 445-466. ISSN : 0006-3495. DOI : https://doi.org/10.1016/S0006-3495(61)86902-6. URL : https://www.sciencedirect.com/science/article/pii/S0006349561869026.

[9] E. HAIRER. “Order conditions for numerical methods for partitioned ordinary differential equations”. In : *Numerische Mathematik* 36.4 (1981), p. 431-445. ISSN : 0945-3245. DOI : 10.1007/BF01395956. URL : https://doi.org/10.1007/BF01395956.

[10] Ernst HAIRER, Syvert P. NØRSETT et Gerhard WANNER. *Solving Ordinary Differential Equations I : Nonstiff Problems*. 2e éd. T. 8. Springer Series in Computational Mathematics. Springer Berlin, Heidelberg, 1993, p. XV, 528. ISBN : 978-3-540-56670-0. DOI : 10.1007/978-3-540-78862-1. URL : https://doi.org/10.1007/978-3-540-78862-1.

[11] Ami HARTEN. “Adaptive Multiresolution Schemes for Shock Computations”. In : *Journal of Computational Physics* 115.2 (1994), p. 319-338. ISSN : 0021-9991. DOI : https://doi.org/10.1006/jcph.1994.1199. URL : https://www.sciencedirect.com/science/article/pii/S0021999184711995.

[12] James KEENER et James SNEYD. *Mathematical Physiology*. 1re éd. Interdisciplinary Applied Mathematics. New York, NY : Springer, 1998, p. 281. ISBN : 978-0-387-22706-1. DOI : 10.1007/b98841.

[13] Christopher A. KENNEDY et Mark H. CARPENTER. “Additive RungeKutta schemes for convectiondiffusionreaction equations”. In : *Applied Numerical Mathematics* 44.1 (2003), p. 139-181. ISSN : 0168-9274. DOI : https://doi.org/10.1016/S0168-9274(02)00138-1. URL : https://www.sciencedirect.com/science/article/pii/S0168927402001381.

[14] Randall J. LEVEQUE. *Numerical Methods for Conservation Laws*. Lectures in Mathematics ETH Zürich. Basel : Birkhäuser Verlag, 1990. ISBN : 978-3-7643-2464-3. DOI : 10.1007/978-3-0348-5116-9.

[15] L. PARESCHI et G. RUSSO. *Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation*. 2010. arXiv : 1009.2757 [math.NA]. URL : https://arxiv.org/abs/1009.2757.

[16] Marie POSTEL. “Approximations multiéchelles”. Polycopié, Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie.

[17] Louis REBOUL. “Development and analysis of efficient multi-scale numerical methods, with applications to plasma discharge simulations relying on multi-fluid models”. 2024IPPAX134. Thèse de doct. 2024. URL : http://www.theses.fr/2024IPPAX134/document.

Annexes

Annexe A : Évaluation numérique des fonctions de stabilité

```python
import numpy as np

def evaluate_stab_ImEx222(ze , zi):
    GAMMA = (2-np.sqrt(2))/2
    DELTA = 1 - 1/(2*GAMMA)
    u1 = (1 + GAMMA * ze)/(1 - GAMMA * zi)
    u2 = (1 + (1-GAMMA) * zi * u1 + DELTA * ze + (1-DELTA) * ze * u1) /
        (1 - GAMMA * zi)
    return np.abs(u2)

def evaluate_stab_ImEx232(ze,zi) :
    GAMMA = (2-np.sqrt(2))/2
    DELTA = -2 * np.sqrt(2) / 3
    u1 = (1 + GAMMA * ze)/ (1 - GAMMA * zi)
    u2 = (1 + DELTA * ze + (1-DELTA) * ze *u1 + (1-GAMMA) * zi * u1) /
        (1 - GAMMA*zi)
    u_fin = 1 + (1-GAMMA)*zi*u1 + GAMMA*zi*u2 + (1-GAMMA)*ze*u1 + GAMMA*
        ze*u2
    return np.abs(u_fin)

def evaluate_stab_RKE2(ze,zi) :
    z = ze+zi
    return np.abs(1+z+(z**2)/2)

def evaluate_stab_RKI2(ze,zi) :
    ze = ze/2
    GAMMA=1 - np.sqrt(2)/2
    z = ze+zi
    u1 = 1/(1-GAMMA*z)
    u2 = (1 + (1-2*GAMMA)*z*u1)/(1-GAMMA*z)
    return np.abs(1+.5*z*(u1 + u2))


def evaluate_stab_SplittingRK2(ze,zi):
    R_e = evaluate_stab_RKE2(ze,0)
    R_i = evaluate_stab_RKI2(0,zi)
```

```
34      return np.maximum(R_e,R_i)
```