



En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE)

Présentée et soutenue le 9 janvier 2018 par : THIBAUT LUNET

Stratégies de parallélisation espace-temps pour la simulation numérique des écoulements turbulents

JURY

ROLAND MASSON MARTIN GANDER MARC MASSOT DANIEL RUPRECHT QIQI WANG SERGE GRATTON JULIEN BODART XAVIER VASSEUR

Université de Nice Sophia Antipolis Université de Genève École Polytechnique University of Leeds Massachussets Institute of Technology Toulouse-INP-IRIT ISAE-SUPAERO, DAEP ISAE-SUPAERO, DISC

Président du jury Rapporteur Rapporteur Juré Juré Directeur de thèse Encadrant de thèse Encadrant de thèse

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées Unité de Recherche : ISAE-SUPAERO, DAEP & CERFACS, Parallel Algorithms project Directeur(s) de Thèse : Serge GRATTON, Julien BODART et Xavier VASSEUR Rapporteurs : Martin GANDER, Roland MASSON et Marc MASSOT

À François JULLIEN de POMMEROL,

un grand-père bienveillant et réaliste, qui m'a toujours raconté que partir dans une filière scientifique ouvrait toutes les portes ... pas sûr qu'il se doutait que j'irai jusqu'à ouvrir celle-là ...

Résumé

Actuellement, la taille des simulations numériques de problèmes physiques atteint des dimensions parfois gigantesques, afin de correctement prédire des phénomènes de plus en plus complexes. C'est le cas en particulier de certaines simulations d'écoulements turbulents en mécanique des fluides numérique (CFD), dont les tailles ont atteint récemment plus de 10¹¹ degrés de liberté (Lee & Moser, 2013). Afin de continuer à développer notre capacité de calcul et de simulation, de nouveaux super-calculateurs verront le jour dans les prochaines décennies, autorisant jusqu'à 10¹⁵ FLOPS. Cependant, les algorithmes de parallélisation classiques (décomposition de domaine en espace) deviennent de moins en moins efficaces lorsqu'utilisés dans ce contexte massivement parallèle. La communauté du calcul numérique intensif étudie donc depuis une dizaine d'années la possibilité d'intégrer la parallélisation en temps comme nouveau degré de parallélisme.

L'objectif de cette thèse est d'effectuer un état de l'art des méthodes de parallélisation en temps existantes à l'heure actuelle, afin de cerner des stratégies de parallélisation espace-temps pouvant être appliquées efficacement à un solveur CFD explicite dans un contexte de calcul massivement parallèle. Après un premier travail de sélection, le choix se porte donc sur l'algorithme Parareal avec grossissement spatial.

Dans un premier temps, cet algorithme de parallélisation en temps est étudié sur un modèle simplifié des équations de Navier-Stokes (équation d'advection) par le biais d'une analyse de Fourier et de multiples expériences numériques. Ceci permet d'identifier les paramètres jouant majoritairement sur la précision et la convergence de l'algorithme, ainsi que de quantifier leurs effets.

Puis, dans une deuxième partie, une attention particulière est portée sur la définition de deux problèmes turbulents canoniques tridimensionnels pouvant servir de cas tests pour des algorithmes de parallélisation en temps : la décroissance d'une turbulence homogène isotrope et l'écoulement de canal turbulent. Des études détaillées sont menées sur les deux problèmes considérés, afin de déterminer à la fois le gain en termes de performance apporté par Parareal avec grossissement spatial dans le cadre d'une parallélisation combinée espace-temps, et sa capacité à correctement représenter les propriétés physiques des écoulements turbulents.

Enfin, cette thèse se conclut en esquissant des conclusions quant aux gains que peut apporter la parallélisation en temps pour la CFD, et en proposant une description de ses perspectives au regard des architectures de calcul haute performance de demain.

<u>Mots-clés</u> : Calcul Haute Performance, Mécanique des Fluides Numérique, Parallélisation en Temps, Écoulements Turbulents, Parareal.

Remerciements

À mes encadrants ...

Il faudrait sûrement plus que quelques mots pour remercier ces trois personnes qui, durant trois ans, ont eu à la fois la motivation, le courage et la patience, de m'indiquer là où il fallait que j'aille et là où il valait mieux que j'arrête de m'entêter, de prendre part à nos réunions dont la durée syndicale minimale était équivalente à trois pauses déjeuner (françaises), et d'élever le niveau scientifique (et grammatical) de mes écrits aux relents parfois quelque peu abscons.

Tout au long de cette thèse, j'ai pu m'inspirer et apprendre d'eux, chacun m'apportant à sa manière une part de son expérience pour ma prochaine étape dans ce monde, fascinant mais parfois cruel, qu'est celui de la recherche scientifique.

Mon premier, dans l'ordre alphabétique, fut aussi la première personne que j'ai rencontré aux origines de la thèse, et qui me fit confiance tout en sachant que la mécanique des fluides et moi faisions au moins trois. Pourvu d'un grand sens des réalités couplé d'une certaine tranche de franchise, il n'a jamais hésité à utiliser des tournures comme "on comprend rien" ou encore "ça a l'air chiant" au vu de mes premières naïves tentatives de rédaction d'article. Certes, ces remarques (très) occasionnelles n'étaient jamais faciles à entendre, mais elles me permirent de réaliser qu'il est, et sera, toujours possible de remettre la qualité de mon travail en question. Non pas par perfectionnisme mal contenu, mais plutôt par envie de rendre ce que je fais le plus accessible possible à toute personne ne partageant pas les méandres de mon propre cerveau. Car quelle est la valeur d'un travail de recherche si personne n'a envie de le lire ou ne peut le comprendre?

Rien que pour m'avoir appris ceci, je lui en serai toujours reconnaissant. Mais plus que cela, il m'a aussi transmis son envie de savoir, de comprendre, et de chercher à expliquer; que derrière tout résultat, il y a une explication justifiable et plausible; qu'il est parfois possible de se tromper dans notre interprétation des choses, mais que s'arrêter à de simples observations par peur d'avoir tort ne sera jamais suffisant.

Enfin, c'est aussi lui qui m'a fait réaliser qu'au lieu d'utiliser ses propres faiblesses pour justifier la médiocrité de son travail et attendre la pitié de ceux qui le jugeraient, peut-être vaut-il mieux s'en servir comme d'une motivation pour chercher à les dépasser, de sorte qu'à la fin du parcours, l'on puisse être heureux et fier de tout le chemin parcouru.

Mon deuxième, fut un directeur de thèse à l'emploi du temps souvent chargé et parfois quelque peu contraignant. Cela ne l'a cependant pas empêché d'être d'une grande disponibilité durant ces trois ans, que ce soit pour participer à des vidéo-(ou juste audio)-conférences depuis les quatre coins d'Europe, ou pour m'accueillir pendant quelques petites heures dans son bureau. Si son humour ou son idée du moyen de déplacement rapide m'ont parfois laissé quelque peu perplexe, il m'a cependant énormément marqué par son côté passionné par la science, assumant totalement et fièrement son originalité.

Et plus que cet amour du savoir, c'est surtout cet entrain à le partager et à le développer qui m'a inspiré et m'inspire encore aujourd'hui; c'est comme si pour lui, l'enseignement n'est pas juste une charge, ou une distribution de son savoir à des cerveaux que l'on espère le moins hermétiques

possible, mais plutôt un défi continuel qui change avec chaque élève, et des opportunités permanentes de découvrir de futures petites pépites scientifiques. Il fait clairement partie des quelques professeurs qui m'ont donné envie de partir dans la même voie, notamment pour essayer d'arriver un jour à enseigner comme eux le font.

Enfin, s'il y a un dernier aspect du personnage que j'ai pu apprécier, c'est bien son humanité et sa sensibilité; sa capacité à sentir quand la mise en valeur de la part de travail bien faite vaut mieux que la critique de celle à améliorer; sa facilité à adroitement parsemer quelques graines d'humour dans un océan de sérieux intellectuel. C'est le genre de directeur qui ne se contente pas d'une relation nécessaire et codifiée avec ses doctorants : il cherche aussi à les connaître, les mettre en confiance, et à comprendre ce qu'ils ont derrière la tête, même lorsque c'est très embrouillé au dedans. Bref, je ne peux que lui souhaiter de rester ainsi par la suite, pour que d'autres aient aussi cette chance de le connaître et de travailler avec lui.

En ce qui concerne mon dernier, il m'a fallu quelques temps pour apprendre à le connaître. Sa nature particulièrement discrète en est probablement la raison, mais pas uniquement : si l'on prend le temps de creuser au delà de cette réserve polie et délicate, on peut découvrir une personnalité riche et dotée d'une attention aux autres qui se fait rare de nos jours, d'un goût prononcé du travail bien fait, sachant tout à fait manier occasionnellement l'humour et l'ironie.

S'il y a quelque-chose qui m'a marqué chez lui, c'est bien sa rigueur, à la fois scientifique mais surtout grammaticale, capable de corriger l'anglais d'un canadien anglophone ayant trop longtemps vécu en France. Et au delà de ça, c'est aussi quelqu'un qui m'a très bien montré comment faire pour organiser mes idées, rendre ma "prose" la plus claire possible, tout en restant cohérent et rigoureux. Si d'aventure j'ai réussi à élever la qualité d'écriture et de clarté de ce manuscrit, c'est principalement grâce à lui.

Enfin, cette personne m'a aussi redonné une nouvelle définition de l'expression "compter sur quelqu'un". Un engagement pour lui en est véritablement un, et ce ne furent ni les vacances loin d'internet, ni les horaires de non-travail, qui ne l'ont empêché de m'apporter ses commentaires et corrections, au fur et à mesure que je lui fournissais, goutte à goutte, mes premiers efforts de rédaction. S'il y a quelques qualités que j'aimerai apprendre de lui, c'est bien son dévouement à ses étudiants pour leur donner toutes les billes possibles pour la suite, et sa capacité à les encourager sans cesse pour qu'ils tiennent jusqu'au bout du chemin.

Bref, je ne sais pas si mon baratin fut vraiment la meilleure manière de leur exprimer ma reconnaissance, mais à tous les trois : MERCI!

À tous les autres ...

Parce que je suis quand même quelques fois sorti de mon bureau pendant ces trois ans de thèse ...

Tout d'abord à mes parents, Catherine et Patrick Lunet, qui se sont dit un jour "Tiens, ça serait chouette d'en avoir un autre", et tous mes frères et sœurs : Anne, qui m'a suffisamment enguirlandé pour que je fasse un minimum attention à ma santé; Pierre, qui m'a convaincu qu'au final, ça valait le coup de venir à Supaero; Bénédicte, qui a partagé avec moi l'inconvénient de ne faire partie ni des aînés, ni des petits derniers; Étienne, qui m'a fait découvrir un style de musique à la fois répétitif et entraînant, qui a animé une grande partie de mes séances de travail; et Charlotte, qui n'a jamais manqué d'histoires toutes plus rocambolesques les unes que les autres pour me changer les idées. Je n'oublie pas non plus les grands parents, Mamie, Bon-Papa et Mère-Grand, qui ont toujours fait l'effort d'écouter le titre de ma thèse jusqu'au bout pour en retenir au moins trois mots, et la foultitude de cousines, cousins, tantes et oncles, belle-soeur et beau-frère, et Nina, ce petit bijou de petite nièce, arrivée juste à la fin comme un mignon petit lutin sur la bûche de Noël. Merci aussi à tous ceux qui ont croisé ma route, que ce soit avant ou pendant ces trois années, et en particulier :

à Rémi, grand bricoleur infatigable, qui m'a donné, très jeune, le goût d'apprendre et d'expérimenter, et surtout qui m'a fait découvrir le plaisir de construire;

à Olivier, qui m'a introduit dans ce monde fascinant des arts vivants, et m'a totalement fait confiance à des moments où je n'en avais que très peu rien qu'en moi-même;

à Marc, mon inéluctable binôme, toujours disponible pour se prendre une petite fessée à Mario-Kart, ou un petit snack sur les bords de la Garonne ou sur la place du Capitole,

à Hélène, grande styliste de l'impossible, seule personne au monde capable de me faire changer ma garde robe en une après-midi de shopping, mais aussi toujours disponible pour prêter une oreille attentive et attentionnée,

à Livia, ma stagiaire/collaboratrice/artiste-peintre/accessoiriste/assistante aux décors préférée,

à Florian, qui m'a tant émerveillé les oreilles par l'exotisme de son accent du sud-ouest-maispas-n'importe-lequel, et surtout sa manière de dire "pain" dès qu'il n'y a pas de chocolat dedans,

à Nicolas, si souvent dispo pour une petite soirée de travail, qui se transformait inéluctablement en soirée bière/petit-repas/végétation-sur-canapé,

à toute cette bande de Supaero & Co., que j'ai eu le plaisir de fréquenter pendant ces six années et demi passées à Toulouse, et tous ceux qui s'y sont rajoutés au fur et à mesure des années,

à toute l'équipe Winterreise, sans cesse composée et recomposée, et tous les fabuleux artistes gravitant autour que j'ai eu le plaisir de rencontrer,

à toute la Compagnie Secouriste Sainte Barbe, et en particulier tous les jeunes, parents et chefs du (magnifique) peloton Toulousain,

à tous les membres du DAEP, chercheurs, doctorants et post-doc, avec qui j'ai partagé la moitié de mon temps, et quelques raclettes bien arrosées (à l'eau, bien sûr);

à tous les membres de l'équipe Algo du Cerfacs, avec qui j'ai partagé l'autre moitié de mon temps, et de chouettes pauses café en anglo-franco-coréano-écossais;

Et enfin, merci à mes trois rapporteurs, Martin Gander, Marc Massot, et Roland Masson, d'avoir relu et commenté ce travail de thèse, et plus particulièrement au premier pour m'avoir donné l'opportunité de travailler avec lui par la suite, et les jurés, Daniel Ruprecht et Qiqi Wang, pour avoir accepté de venir participé à ma soutenance.

Notations

Abréviations usuelles

- *e.g.* : *exempli gratia* (par exemple)
- *cf.* : *confer* (se reporter à)
- *i.e.* : *id est* (c'est-à-dire)
- Chap.: Chapitre(s)
- Sec. : Section(s)
- Ap. : Appendice(s)
- Fig. : Figure(s)
- Déf. : Définition(s)
- Lem. : Lemme(s)
- Prop. : Propriété(s)
- Obs.: Observation(s)

Symboles mathématiques

- \mathbb{R}^p : l'espace vectoriel réel de dimension p
- \mathbb{C}^p : l'espace vectoriel complexe de dimension p
- $\mathcal{R}(z)$: la partie réelle d'un nombre complexe z
- $\mathcal{I}(z)$: la partie imaginaire d'un nombre complexe z
- $\mathcal{M}_p(\mathbb{R})$: l'ensemble des matrices carrées de taille $p \times p$ à coefficients réels
- $\mathcal{M}_p(\mathbb{C})$: l'ensemble des matrices carrées de taille $p \times p$ à coefficients complexes
- \mathbb{I}_p : la matrice identité de l'espace $\mathcal{M}_p(\mathbb{R})$
- $\mathcal{P}(\mathbb{X}):$ l'ensemble des polynômes, de degré $d \in \mathbb{N}$
- $\mathcal{Q}(\mathbb{X})$: l'ensemble des fractions rationnelles, de degrés de numérateur et de dénominateur $d, d' \in \mathbb{N}$

Table des matières

1	Intr	oduction	9
	1.1	Vers de nouveaux besoins algorithmiques	10
		1.1.1 État des lieux des moyens de calcul HPC actuels	10
		1.1.2 Recommandations au vu des projections futures	14
	1.2	Calcul haute performance pour la CFD	17
		1.2.1 Contexte de la CFD	17
		1.2.2 Le solveur Hybrid	21
	1.3	Emergence de la parallélisation en temps	23
		1.3.1 Un nouveau concept de parallélisme.	23
		1.3.2 Stratégie de parallélisation combinée espace-temps	25
	1.4	Synthèse	26
2	Syn	thèse des principaux algorithmes actuels de parallélisation en temps et étude	
	d'ap	oplicabilité	27
	2.1	Parareal, le cadet célèbre	28
		2.1.1 A la base de l'algorithme : la méthode de tirs multiples (<i>Multiple Shooting</i>)	28
		2.1.2 Description de PARAREAL	30
		2.1.3 Variantes	31
		2.1.4 Applications, forces et faiblesses	33
	2.2	PARAEXP, la solution "idéale" (dans le cas linéaire)	35
		2.2.1 Description	35
		2.2.2 Applications, forces et faiblesses	37
	2.3	RIDC, l'intégration temporelle alternative	38
		2.3.1 Description	38
		2.3.2 Applications, forces et faiblesses	39
	2.4	PFASST, la complexité au service de l'efficacité	40
		2.4.1 Description	40
		2.4.2 Applications, forces et faiblesses	40
	2.5	MGRIT, la solution "universelle"?	42
		2.5.1 Description	42
		2.5.2 Applications, forces et faiblesses	43
	2.6	Vers un premier choix d'algorithme	44
		2.6.1 Critères de sélection	44
		2.6.2 Synthèse concernant les différents algorithmes présentés	45
		2.6.3 Justification du choix de Parareal	46
	2.7	Synthèse	47
3	Ana	lyse de PARAREAL avec des méthodes temporelles explicites pour des problèmes	
	liné	aires fortement advectifs	48
	3.1	Motivations générales	49
	3.2	Analyse linéaire de Fourier pour Parareal	50

		2.2.1 Easteur d'amplification d'un intégrateur temporal linéaira	Λ
		3.2.1 Facteur d'amplification de Dupuper.	ן ר
		3.2.2 Facteur d'amplification de la disentitien enstiele	3 7
		3.2.3 Prise en compte de la discretisation spatiale	3
		3.2.4 Prise en compte du changement de grille	/
	3.3	Etude des formes explicites de PARAREAL pour les problemes advectifs	4
		3.3.1 Cas scalaire	С
		3.3.2 PARAREAL avec grille spatiale unique	1
		3.3.3 PARAREAL avec grossissement spatial	3
		3.3.4 Conclusions générales autour de l'analyse de Fourier	4
	3.4	Variations autour de l'advection pure linéaire 85	5
		3.4.1 Motivations et mise en place de l'étude 85	5
		3.4.2 Equation d'advection-diffusion	5
		3.4.3 Equation de Burgers avec terme visqueux	0
		3.4.4 Conclusions générales sur l'étude de convergence	3
	3.5	Synthèse	4
4	Ana	lyse de Parareal appliqué aux simulations d'écoulements tridimensionnels tur-	
	bule	ents 90	5
	4.1	Analyse de performance	7
		4.1.1 Modèle de scalabilité	7
		4.1.2 Modèle de communication	5
	4.2	Introduction à l'analyse de convergence	8
		4.2.1 Equations de Navier-Stokes compressible	9
		4.2.2 Challenges pour l'intégration temporelle	9
		4.2.3 Choix des problèmes tests	2
	4.3	Décroissance d'une Turbulence Homogène Isotrope	3
		4.3.1 Description du problème	3
		4.3.2 Définition des solveurs fin et grossier pour Parareal	6
		4.3.3 Analyse de convergence de PARAREAL pour la décroissance de Turbulence	
		Homogène Isotrope	7
	4.4	Écoulement de Canal Turbulent	5
		4 4 1 Description du problème 12	5
		4.4.2 Définition du solveur fin et grossier pour Parareal	9
		443 Analyse de convergence de PARAREAL pour l'Écoulement de Canal Turbulent 13	1
	4.5	Synthèse	9
F	Con	ducions 14	1
3		Déalisations logisielles 14	1
	5.1		ו ר
	5.Z	Synthese	2
	5.5	Perspectives	5
Α	Inté	gration temporelle par Spectral Deferred Correction 14	5
	A.1	Description détaillée	5
	A.2	Applications à des problèmes advectifs avec intégration temporelle explicite 14	7
		A.2.1 Cadre de l'étude	7
		A.2.2 Instabilités des versions explicites des SDC	8
		A.2.3 <i>Sweep</i> SDC d'ordre élevé	9
		A.2.4 Stabilité et performance des SDC avec <i>sweep</i> d'ordre élevé	1
		A.2.5 Synthèse	2

 B.1 Méthodes de Runge-Kutta explicites B.2 Symboles de Fourier des discrétisations spa C Interpolation d'ordre variable pour maillage rés C.1 Maillages périodiques uniformes C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimensio C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-p. C.4 Application pour des champs solutions obto 	153 tiales 153 s cartésiens tridimensionnels structu- 155 nensionnelle 155 onnels 156						
 B.2 Symboles de Fourier des discrétisations spa C Interpolation d'ordre variable pour maillage rés C.1 Maillages périodiques uniformes C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimension C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-p. C.4 Application pour des champs solutions obtempted 	tiales						
 C Interpolation d'ordre variable pour maillage rés C.1 Maillages périodiques uniformes C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimension C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-p. C.4 Application pour des champs solutions obtemps de conditions de conditions	s cartésiens tridimensionnels structu- 155 ensionnelle 155 onnels 156						
 rés C.1 Maillages périodiques uniformes C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimension C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-prise c.4 Application pour des champs solutions obtemps control des champs control des champs solutions obtemps control des champs control des control	155 nensionnelle 155 onnels 156						
 C.1 Maillages périodiques uniformes C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimensio C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-p C.4 Application pour des champs solutions obto 							
 C.1.1 Interpolation point milieu monodin C.1.2 Extension aux maillages tridimension C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-pour des champs solutions obtendes des constructions des constructions	nensionnelle 155 onnels 156 157 bériodiques. 158						
 C.1.2 Extension aux maillages tridimension C.2 Extension aux maillages non-uniformes C.3 Prise en compte de conditions limites non-periode C.4 Application pour des champs solutions obtention 	onnels						
C.2 Extension aux maillages non-uniformesC.3 Prise en compte de conditions limites non- C.4 Application pour des champs solutions obte							
C.3 Prise en compte de conditions limites non-jC.4 Application pour des champs solutions obte	périodiques						
C.4 Application pour des champs solutions obt							
	enus avec le solveur Hybrid 159						
D Glossaire de mécanique des fluides	lossaire de mécanique des fluides						
Spectre énergétique d'un écoulement turbulent							
E.1 Turbulence Homogène Isotrope							
E.2 Écoulement de Canal Turbulent							
Validation des Simulations Numériques Directes pour l'Écoulement de Canal Tur-							
bulent	164						
F.1 Loi de maillage dans la direction y							
F.2 Accumulation des échantillons statistiques							
F.3 Validation du maillage spatial							
	teur d'amplification de PARAREAL avec						
G Approximation de commutativité pour le fac							
G Approximation de commutativité pour le fac grossissement spatial	169						
G Approximation de commutativité pour le fac grossissement spatial G.1 Besoin de l'approximation	169						
G Approximation de commutativité pour le fac grossissement spatial G.1 Besoin de l'approximation G.1.1 Relation de récurrence de Pascal &	169						
G Approximation de commutativité pour le fac grossissement spatial G.1 Besoin de l'approximation G.1.1 Relation de récurrence de Pascal & G.1.2 Inconvénients de l'utilisation de la	169algèbre non-commutative170ormule exacte171						
 G Approximation de commutativité pour le fac grossissement spatial G.1 Besoin de l'approximation	169algèbre non-commutative169formule exacte170171171						
 G Approximation de commutativité pour le fac grossissement spatial G.1 Besoin de l'approximation G.1.1 Relation de récurrence de Pascal & G.1.2 Inconvénients de l'utilisation de la r G.2 Validation de l'approximation G.2.1 Comparaison avec les résultats nun 	169algèbre non-commutative169formule exacte170						

CHAPITRE 1

Introduction

Table des matières

1.1	Vers de nouveaux besoins algorithmiques 10						
	1.1.1	État des lieux des moyens de calcul HPC actuels	10				
	1.1.2	Recommandations au vu des projections futures	14				
1.2	Calcu	l haute performance pour la CFD	17				
	1.2.1	Contexte de la CFD	17				
	1.2.2	Le solveur Hybrid	21				
1.3	1.3 Emergence de la parallélisation en temps						
	1.3.1	Un nouveau concept de parallélisme.	23				
	1.3.2	Stratégie de parallélisation combinée espace-temps	25				
1.4	Synth	tèse	26				

"Ne cherchez pas à avoir une armée trop nombreuse, la trop grande quantité de monde est souvent plus nuisible qu'elle n'est utile." [Tzu, 2016, Article VI, §.20]

Cette mise en garde de Sun Tzu, écrivain et stratège chinois (543 av. J.-C. – 495 av. J.-C.) prend un sens tout particulier lorsqu'on la transpose au contexte actuel du calcul haute performance (**High-Performance Computing, HPC**) pour la simulation et prédiction des phénomènes physiques. En effet, les moyens de calcul intensif d'aujourd'hui sont majoritairement basés sur la multiplication du nombre de "travailleurs" (cœurs de calcul) plutôt que sur l'amélioration de leur "performance individuelle" (nombre d'opérations par seconde par cœur). De plus, cet aspect semble être la tendance générale pour la planification des architectures de calcul haute performance de demain.

Depuis leurs origines, les techniques permettant de tirer profit des architectures HPC ont été continuellement confrontées à des limitations, caractérisées par une perte d'efficacité lorsque le nombre de "tâches" considérées (processus de calcul) devient trop important. Les préoccupations autour de ces limitations sont aujourd'hui principalement motivées par des raisons énergétiques, et toujours plus omniprésentes, car il devient de plus en plus délicat de résoudre de manière efficace, les problèmes à complexité croissante qui intéressent la communauté scientifique. Au vu des ressources de calcul colossales qui seront mises en place dans les années à venir, il apparaît nécessaire d'étudier et de développer dès à présent des algorithmes à même de tirer parti des prochaines architectures HPC.

"Celui qui, prudent, se prépare à affronter l'ennemi qui n'est pas encore; celui-là même sera victorieux." [Tzu, 2016, Article III, §.41]

Ce chapitre cherche à présenter plus en détail le contexte HPC actuel et les projections que l'on peut déjà réaliser pour les années à venir (Sec. 1.1). On se focalisera ensuite sur un pan des applications du HPC, celui de la simulation numérique des écoulements (*Computational Fluid Dynamics*, CFD), et du cadre plus particulier que l'on veut considérer dans cette thèse : les écoulements turbulents (Sec. 1.2), qui monopolisent une majeure partie des ressources pour la communauté CFD. Enfin, on introduira en Sec. 1.3 la thématique de ce travail de thèse : la parallélisation en temps.

1.1 Vers de nouveaux besoins algorithmiques

Par la suite, on désignera par **processus** l'action réalisant un calcul ("tâche") et par **cœur** le composant, ou unité de calcul ("travailleur"), permettant d'effectuer un (ou plusieurs) processus.

1.1.1 État des lieux des moyens de calcul HPC actuels

Depuis leur genèse¹, les architectures HPC dites "de calcul massivement parallèle" n'ont cessé d'évoluer, tant par leurs types de composants de base (*Computational Processing Unit* (CPU), *Graphics Processing Unit* (GPU), ...), que par une augmentation substantielle de leur nombre de cœurs de calcul. Afin de pouvoir utiliser au mieux ces machines "multi-cœurs", la communauté scientifique a développé des **modèles de programmation** permettant d'effectuer des calculs en utilisant plusieurs processus en parallèles, et ainsi d'accélérer les programmes (ou codes) de calcul.

Parmi les nombreux concepts ayant été proposés, deux modèles se sont depuis imposés :

- *Modèle à Mémoire Distribuée* (MMD) : l'espace mémoire permettant de stocker les données du calcul est séparé distinctement pour chaque processus. Ces derniers échangent des données dès que nécessaire au cours du calcul. Ce modèle est principalement spécifié aujourd'hui par le biais de la norme (ou standard) MPI (*Message Passing Interface*) [Gropp, 1998].
- *Modèle à Mémoire Partagée* (MMP) : l'ensemble des données du problème est disponible sur une mémoire commune accessible à un ensemble de processus. Ce modèle est principalement spécifié aujourd'hui par le biais de la norme OpenMP [Dagum et Menon, 1998].

Ces deux concepts mêlent avantages et inconvénients. Tandis que le modèle MMD permet de traiter facilement des problèmes de très grande taille, en répartissant l'utilisation de la mémoire sur de nombreuses unités de calcul, il induit un échange de données qui s'intensifie avec le nombre de processus considérés. Le MMP permet quant à lui de s'affranchir de ces communications entre processus, mais induit une certaine latence au cours des accès à la mémoire. De plus, il devient inapplicable lorsque la taille des problèmes est trop importante par rapport à la mémoire physique disponible sur un même composant.

Bien que des modèles hybrides MMD + MMP soient de plus en plus souvent envisagés pour optimiser l'utilisation des ressources de calcul, la plupart des codes actuels permettant de résoudre des problèmes de grande taille reposent sur le modèle MMD [Bergman *et al.*, 2008, Sec. 4.3]. On se focalise ici sur ce modèle de programmation, et en particulier sur une technique de mise en œuvre largement utilisée aujourd'hui dans ce cadre : la parallélisation en espace.

La parallélisation spatiale par mémoire distribuée. Cette technique consiste à décomposer le domaine spatial du problème en plusieurs sous-domaines. Un processus (ou un ensemble de processus) est alloué à chaque sous-domaine, comme représenté en Fig. 1.1, et résout le problème sur sa propre partie de l'espace. Les calculs peuvent ainsi être effectués en parallèle, sous réserve d'une communication régulière des données aux interfaces.



FIGURE 1.1 – Schéma d'une parallélisation spatiale appliquée à un domaine spatial 2D.

^{1.} La première machine dénommée supercalculateur, Atlas, fut installée en 1962 à l'université de Manchester. Elle fut accompagnée en 1964 par une autre génération de machines parallèles américaines, les CDC 6600, conçus par Seymour Cray.

On définit alors l'accélération parallèle, ou speedup associé à cette parallélisation comme étant :

$$\mathcal{S}(N_{proc}) = \frac{T_{calc}^1}{T_{calc}^{N_{proc}}},\tag{1.1}$$

avec $T_{calc}^{N_{proc}}$ le temps de calcul pour résoudre le problème avec N_{proc} processus. Lorsque la répartition de charge ² est égale pour chaque processus et que le temps de communication est négligeable face au temps de calcul d'un processus sur son sous-domaine, alors le *speedup* est idéal, *i.e.* $S(N_{proc}) \simeq N_{proc}$.

Cependant, si le temps de communication devient non négligeable face au temps de calcul de chaque processus, $S(N_{proc})$ devient alors inférieur à N_{proc} . Dans des situations plus extrêmes, on peut même observer que pour $N_{proc} < N'_{proc}$, $S(N_{proc}) > S(N'_{proc})$: la parallélisation spatiale perd alors tout intérêt. Afin de normaliser et mesurer sa performance, on définit également l'efficacité parallèle :

$$\mathcal{E}(N_{proc}) = \frac{\mathcal{S}(N_{proc})}{N_{proc}}.$$
(1.2)

Cette dernière, idéalement proche de 1, est donc un facteur important à prendre en compte, et fournit une manière normalisée de représenter le *speedup* d'un calcul parallèle. Ce dernier dépend en particulier du code de calcul utilisé; la manière de l'évaluer dans un cadre plus général est présentée dans le paragraphe suivant.

Notion de scalabilité (ou passage à l'échelle). Pour un code de calcul parallèle donné, on définit sa scalabilité par l'évolution de son *speedup* (ou efficacité parallèle) lorsque l'on augmente le nombre de processus utilisés pour résoudre un problème. Celle-ci peut être observée de deux manières :

Définition 1.1.1 : Scalabilité Faible – Évolution de l'efficacité parallèle lorsque N_{proc} augmente, en gardant une charge constante par processus. (e.g. Fig. 1.2 (a)).

Définition 1.1.2 : Scalabilité Forte — Évolution de l'efficacité parallèle lorsque N_{proc} augmente, en gardant une charge constante pour le problème global (e.g. Fig. 1.2 (b)).



FIGURE 1.2 – Scalabilités faible et forte : exemples de tests pour une parallélisation spatiale sur un domaine 2D.

Ces deux définitions caractérisent chacune un certain niveau de performance du code de calcul considéré. Alors que la scalabilité faible représente sa capacité à pouvoir être appliqué efficacement à des problèmes de taille de plus en plus grande, la scalabilité forte permet d'estimer à quel point le temps de restitution pour un problème de taille fixe peut être diminué, en utilisant le plus de processus possible.

Ces deux propriétés dépendent donc du code de calcul considéré, mais aussi des performances propres de la machine massivement parallèle utilisée.

^{2.} Par charge on désigne la quantité d'opérations devant être effectuées, qui peut en général se rapporter au nombre de degrés de liberté considérés.

Les architectures de calcul actuelles. La conception et les performances d'un super-calculateur reposent actuellement sur trois éléments de bases :

- 1. L'**unité de calcul**, ou "cœur" de calcul (*core*) : il s'agit du composant permettant d'effectuer les opérations numériques de base (addition, multiplication, ...) et de gérer la manipulation des données en mémoire (accès, remplacement, déplacement, ...). Plusieurs cœurs sont regroupés au sein d'un processeur (autour de 10 ou 20 pour pour les architectures actuelles), dont les CPU sont aujourd'hui la forme la plus répandue. La réalisation d'une action élémentaire par cœur est mesurée en temps de cycle du processeur, donné à son tour par une fréquence d'horloge f_{proc} . Elle permet en particulier d'évaluer le nombre d'opérations numériques de base (mesuré généralement en *FLoating Point Operations* (FLOP)) pouvant être effectué par le processeur au cours d'un cycle. Les processeurs sont regroupés en **nœuds de calculs** (*nodes*), eux-même regroupés en rangs (*rack*) au sein du super-calculateur.
- 2. La **mémoire** : c'est l'espace permettant de stocker les données du calcul. Elle est répartie en plusieurs niveaux hiérarchiques, chacun étant caractérisé par une taille (ou capacité) de stockage et une vitesse d'accès. Selon les niveaux considérés, le temps d'accès diminue avec la taille mémoire disponible. On distinguera en particulier la mémoire vive (*Random-Access Memory*, RAM), l'espace disponible pour un nœud de processeurs au cours d'un calcul.
- 3. Le **réseau de communication** : celui-ci permet aux différents cœurs de communiquer des données entre eux. Ces communications se font à une certaine vitesse, ou débit. Ce dernier peut varier selon les positions relatives de chaque cœur (processeur, nœud, rang, ...), et les caractéristiques matérielles des connexions.

La vitesse de calcul offerte par un super-calculateur (mesurée généralement en FLOP par seconde (FLOP/s)) dépend donc de la combinaison de ces trois éléments principaux. De manière générale, on distingue la performance crête idéale (proportionnelle au nombre de cœurs disponibles et à leur fréquence d'horloge) et la performance effective (mesurée en pratique par des tests). Pour exemple, les caractéristiques des cinq architectures HPC les plus rapides à ce jour sont données en Tab. 1.1.

TABLE 1.1 – Caractéristiques principales des 5 super-calculateurs les plus rapides au 6 juin 2017³.DDM : Date de Dernière Modification, N_{cores} : nombre de cœurs de calcul, f_{proc} : fréquence des CPU, R_{max} : performance effective mesurée avec le LINPACK, R_{peak} : performance crête idéale, $C_e/C_{e,Tlse}$: consommation électrique annuelle ramenée à celle de la ville de Toulouse (France) en 2016⁴.

Super-calculateur	(DDM)	Pays	N_{cores}	f_{proc} [GHz]	R_{max} [TFlop/	$[s](R_{max}/R_{peak}) \mid 0$	$C_e/C_{e,Tlse}$
Sunway TaihuLigh	nt (2016)	Chine	10,649,600	1.45	93,014.6	(75%)	10%
Tianhe-2	(2013)	Chine	3,120,000	2.2	33,862.7	(62%)	12%
Piz Daint	(2017)	Suisse	361,760	2.6	19,590.0	(77%)	1.6%
Titan	(2012)	USA	560,640	2.2	17,590.0	(65%)	5.6%
Sequoia	(2012)	USA	1,572,864	1.6	17,173.2	(85%)	5.4%

Les super-calculateurs présentés appartiennent à la classe des architectures *Peta-scale*, c'est-àdire pouvant effectuer de l'ordre de 10¹⁵ FLOP/s⁵. Les capacités de calcul considérées sont donc gigantesques, et reposent principalement sur l'utilisation d'un grand nombre de cœurs, de l'ordre du million pour les architectures les plus importantes. On parle de "parallélisme massif", et il devient de plus en plus difficile pour les codes de calcul parallèles de maintenir une bonne efficacité lorsqu'utilisés avec un nombre aussi important de cœurs.

^{3.} Données obtenues à partir de https://www.top500.org/list/2017/06.

^{4.} Sur l'année 2016, EDF a fourni 1,283,751 MWh d'électricité aux 453,317 habitants de Toulouse, selon https://www.agence-france-energie.fr/edf/toulouse-31000.

^{5.} A titre de comparaison, si chaque personne sur Terre était relativement douée en calcul mental et pouvait effectuer une opération par seconde, il faudrait un jour et demi à l'ensemble de la population mondiale (~ 7.5 milliards de personnes) pour effectuer 1 Peta-Flop.

L'exemple d'utilisation pris généralement comme référence correspond au test de performance LINPACK⁶. Ce dernier exécute de manière très optimisée des factorisations de matrices pleines sur plusieurs processeurs en parallèle. Bien que ce programme permette d'utiliser de manière quasioptimale les ressources d'un super-calculateur, la performance obtenue (R_{max}) n'atteint pas le niveau de performance théorique de la machine (R_{peak}). Or, très peu de codes de calculs effectuent uniquement des opérations aussi simples et optimisées que celles du LINPACK, et de ce fait les pertes de performances pour des parallélisations massives peuvent être bien plus importantes selon les applications considérées.

En particulier, selon les opérations mathématiques effectuées, les processeurs peuvent être utilisés individuellement à différents niveaux d'efficacité. Pour exemple, un produit matrice-matrice peut facilement utiliser près de 100% des cycles d'un processeur pour effectuer des opérations mathématiques élémentaires (additions, multiplications), alors qu'un produit matrice-vecteur, nécessitant bien plus de manipulations mémoire, n'en utilise qu'autour de 15 - 25%⁷ dans les meilleurs des cas. Ceci a par ailleurs motivé le développement du modèle *Roofline* [Williams *et al.*, 2009]. Ce dernier se base sur une représentation intuitive des performances des processeurs selon les applications considérées, et peut être utilisé pour optimiser leur conception et utilisation.

Considérations énergétiques. L'une des observations au vu de la dernière colonne de Tab. 1.1 concerne la consommation énergétique considérable des architectures *Peta-scale*. À quelques exceptions près, l'énergie électrique servant à alimenter ces super-calculateurs a non seulement un coût financier énorme, mais aussi une empreinte environnementale qui devient non négligeable. De ce fait, des pertes de performance de l'ordre de quelques pour cents peuvent représenter des sur-coûts considérables, et vice-versa. Il est donc important de prévoir et d'optimiser au mieux les performances des architectures de calcul pour les applications considérées. On mentionnera l'existence d'une variante énergétique du modèle *Roofline* [Choi *et al.*, 2013], qui permet d'estimer le ratio performance/énergie propre à une application particulière.

L'efficacité parallèle définie en (1.2) peut aussi être utilisée pour obtenir une première estimation du coût énergétique du calcul. Pour cela, on pose P_{proc} la puissance électrique nécessaire pour le calcul d'un processus, et $P_{comm}(N_{proc})$ la puissance requise par les communications induites par la parallélisation avec N_{proc} processus. Bien que la valeur de P_{proc} varie en fonction de la charge d'opérations effectuées, on peut en première approximation l'assimiler à la puissance moyenne requise pour activer un cœur de calcul (voir [Woo et Lee, 2008] pour un modèle plus détaillé). En utilisant cette modélisation simplifiée, le ratio entre consommation énergétique avec et sans parallélisation s'écrit :

$$R_{Energ.} = \frac{T_{calc}^{N_{proc}} N_{proc} P_{proc} + P_{comm}(N_{proc})}{T_{calc}^{1} P_{proc}} = \frac{1}{\mathcal{E}(N_{proc})} + \frac{P_{comm}(N_{proc})}{T_{calc}^{1} P_{proc}}.$$
 (1.3)

Si le coût énergétique des communications est négligeable face au coût de calcul avec un seul processus, $R_{Energ.}$ se réduit à l'inverse de l'efficacité parallèle. Avec cette modélisation, résoudre le problème $S(N_{proc})$ fois plus vite coûte donc au minimum $1/\mathcal{E}(N_{proc})$ fois plus cher en énergie. Or, toutes les applications n'acceptent pas nécessairement les mêmes ratios (gain en temps)/(coût énergétique). Pour exemple, les applications de recherche en mécanique des fluides numérique s'autorisent rarement des efficacités parallèles inférieures à 70-80%, car plusieurs jours (voir semaines) peuvent être envisagés pour l'obtention des résultats.

De manière générale, l'efficacité parallèle minimale autorisée dépend du nombre de cœurs à disposition, et d'une définition d'un temps de restitution, raisonnable au vu de la communauté concernée.

^{6.} Le LINPACK correspond à une série de tests de performance, mis en place par [Dongarra *et al.*, 1979], pour évaluer les performances des machines massivement parallèles.

^{7.} Ces performances sont généralement obtenues par l'utilisation de bibliothèques optimisées comme BLAS-2 (produit matrice vecteur) ou BLAS-3 (produit matrice matrice) [Dongarra *et al.*, 1990]. Elles dépendent naturellement des caractéristiques des processeurs et implémentations, compilateurs, etc ... utilisés.

1.1.2 Recommandations au vu des projections futures

Alors qu'à l'origine des architectures HPC, les développements technologiques pour l'informatique leur étaient principalement dédiés, ces derniers sont aujourd'hui majoritairement pilotés par le marché technologique de masse (ordinateurs personnels, *smartphones*, jeux-vidéo, ...). Les fabricants HPC ont donc suivi son évolution pour concevoir les architectures de calcul massivement parallèles actuelles. Au vu des évolutions récentes, la communauté HPC essaye actuellement de déterminer les tendances technologiques qui marqueront les prochaines architectures. Déjà aujourd'hui, on peut évaluer leurs caractéristiques principales avec une certaine confiance, et dès à présent identifier les évolutions nécessaires des technologies et techniques actuelles pour préparer l'utilisation des architectures *Exascale* de demain (pouvant effectuer jusqu'à 10^{18} FLOP/s).

Cahier des charges des machines Exascale. De nombreuses projections ont été réalisées afin d'identifier les différentes exigences auxquelles seront soumises les machines *Exascale*. On se contentera ici de les résumer en quatre *challenges* principaux, à partir de descriptions plus détaillées données par [Bergman *et al.*, 2008, Chap. 1, p2] et [Dongarra *et al.*, 2014, Sec. 1] :

(1) Avoir une consommation énergétique équivalente à celle des architectures *Peta-scale* actuelles :

même si l'on dispose actuellement du savoir technologique permettant de construire des machines *Exascale*, celles-ci n'ont pas encore été conçues à cause du coût énergétique trop élevé qu'elles engendreraient, au vu des technologies actuelles. En particulier, il conviendra de gérer correctement le coût de transport de données entre des processus situés physiquement à des endroits très éloignés au sein de la même machine.

(2) Optimiser la taille et le temps d'accès mémoire :

la taille des problèmes augmentant, il conviendra aussi d'augmenter la capacité de stockage mémoire du système global. En particulier les différentes manipulations sur chaque niveau de mémoire (lecture, écriture, ...) devront être plus efficaces, pour pouvoir suivre l'augmentation de la rapidité des calculs du système global.

(3) Supporter l'utilisation d'une très large quantité de processus :

la rapidité des processus seuls atteignant une valeur plateau, il devient nécessaire de pouvoir les démultiplier afin d'augmenter les performances du système global.

(4) Être résilient aux erreurs :

l'augmentation du nombre de composants utilisés pour résoudre les problèmes va nécessairement augmenter la probabilité d'une défaillance de l'un des composants au cours du calcul. De ce fait, les prochaines architectures (et/ou algorithmes) devront être conçues pour pouvoir supporter et éventuellement rattraper les défaillances apparaissant dans un ou plusieurs processus.

Caractéristiques des prochaines architectures. Toutes ces conditions décrites précédemment doivent être retranscrites dans le contexte technologique actuel, où l'on peut déjà entre-apercevoir des limitations dans les technologies futures. En particulier, on soulignera les points suivants :

(A) Convergence de f_{proc} vers une valeur plus faible :

la fréquence des processeurs a atteint une valeur maximale seuil autour de 4 GHz dans les années 2000⁸. Cependant, comme discuté dans [Bergman *et al.*, 2008, Sec. 6.2.1.5], la taille des puces atteignant sa limite minimale, l'énergie nécessaire pour refroidir le processeur commençait à atteindre des proportions inenvisageables. De ce fait, les développeurs n'ont actuellement comme seule solution pour augmenter le ratio performance/coût énergétique que de réduire leur fréquence. Une valeur nominale pour f_{proc} a été indiquée par [Dongarra *et al.*, 2011]

^{8.} A titre indicatif, un historique est donné à travers le lien http://www.cs.columbia.edu/~sedwards/classes/2012/3827-spring/advanced-arch-2011.pdf.

autour de 2 GHz, et s'observe déjà au sein des architectures HPC actuelles détaillées en Tab. 1.1.

(B) Augmentation considérable du nombre de processeurs disponibles :

ceci est directement lié au *challenge* (3) du paragraphe précédent. En particulier, ce nombre doit augmenter d'autant plus pour compenser la baisse de fréquence des processeurs détaillée précédemment, rendue nécessaire par le challenge (1). Ce nombre est évalué autour de $10^8 - 10^9$ par [Dongarra *et al.*, 2011, Sec. 4.3.1.1].

(C) Augmentation du coût relatif de transfert et de manipulation des données :

ceci est dû à la baisse nécessaire du coût énergétique requis par les processeurs pour supporter leur démultiplication, et à la multiplication des processus parallèles pour accélérer les calculs. De ce fait, le coût énergétique d'un calcul *Exascale* sera majoritairement dominé par celui nécessaire pour déplacer les données au sein du super-calculateur. Ainsi, dans le modèle de ratio énergétique développé en (1.3), l'importance du terme comprenant le coût de communication va nécessairement augmenter face au terme considérant uniquement l'efficacité parallèle des codes de calcul.

(D) Utilisation d'architectures de processeurs hybrides :

cet aspect, que l'on n'avait encore pas mentionné jusqu'à présent, correspond à une caractéristique importante des architectures HPC, actuelles et prochaines.

Depuis les années 70⁹, les GPU, qui sont un type d'unité de calcul différent de celui des CPU, ont été développés dans le but d'accélérer la génération et visualisation d'images par ordinateurs. Ce type de processeur a connu et connaît encore aujourd'hui un développement phénoménal ¹⁰. Ceci a permis de pouvoir concevoir aujourd'hui des processeurs de calcul comprenant un très grand nombre de GPU, ayant chacun une fréquence moins élevée que des CPU (autour de 1.3 GHz pour les modèles les plus récents ¹¹), mais induisant une consommation énergétique plus faible que celle des CPU. C'est pourquoi depuis quelques années l'utilisation de *General-Purpose* GPU (GPGPU) est concrètement envisagée pour la conception des machines massivement parallèles [Keckler *et al.*, 2011], et leur rôle apparaît prédominant pour la conception des prochaines machines *Exascale* [Gao et Zhang, 2016].

Cela peut déjà se voir sur deux des cinq machines parallèles les plus rapides de Tab. 1.1. Tout d'abord, Piz Daint (3^{ème}) est composée de 21% de nœuds CPU et de 79% de nœuds GPU, ce qui porte le nombre de cœurs GPU à 94.2% du nombre global ¹². Ceci explique notamment sa faible consommation énergétique. Sunway TaihuLight (1^{er}) quant à lui, est composé entièrement de *Computing Processing Element* (CPE), décomposés en groupes de 64 par *Management Processing Element* (MPE), eux-mêmes rassemblés en groupes de 4 par nœud ¹³. Ce type de processeurs (CPE), développé et conçu intégralement en Chine, est un concept intermédiaire entre GPU et CPU.

Pour conclure, on peut aussi mentionner le développement de nouvelles technologies optimisant l'utilisation des processeurs pour des applications particulières, comme les *Field Programmable Gate Arrays* (FPGA). On observe déjà d'importants gains en terme de performance avec l'utilisation de ce type de composants [Chiou *et al.*, 2007], et ces derniers seront très probablement inclus dans la conceptions des futures architectures de calcul.

12. http://www.cscs.ch/computers/piz_daint/index.html

^{9.} Les premières puces électroniques dédiées à des applications graphiques ont été développées dans les années 70, cependant le terme GPU n'a été popularisé qu'en 1999, par la société NVIDIA, lorsqu'elle mis sur le marché le *GeForce* 256, présenté comme le "premier GPU au monde".

^{10.} Ce développement a été à l'origine principalement motivé par l'industrie du jeu vidéo, et plus tard celle de la visualisation 3D en général.

^{11.} Les GPU Tesla P100 de NVidia, utilisés notamment au sein du supercalculateur Piz Daint, tournent à une fréquence de 1.328 GHz et sont rassemblés en groupes de 64 GPU par nœud, appelés aussi *Streaming Multiprocessors*.

^{13.} http://www.netlib.org/utk/people/JackDongarra/PAPERS/sunway-report-2016.pdf

Après avoir résumé les caractéristiques des super-calculateurs *Exascale*, il convient aussi de présenter les caractéristiques des problèmes pour lesquels ils seront utilisés.

Caractéristiques des applications Exascale. Une catégorisation générale des applications typiques qui seraient portées sur les machines *Exascale* a été définie par [Bergman *et al.*, 2008, Sec. 5.6.1]. Ces différentes catégories se définissent par rapport aux applications actuelles, et sont décrites ci-dessous :

- (I) Applications résolvant un problème mille fois plus grand.
- (II) Applications résolvant le même problème, mais mille fois plus rapidement.
- (III) Applications résolvant le même problème, mais sur mille fois plus de pas de temps.
- (IV) Applications résolvant le même problème, mais à une résolution mille fois plus grande.

En particulier, on peut aussi considérer des catégories construites par combinaison d'une ou plusieurs de ces catégories (*e.g.* $10 \times$ plus vite, avec $10 \times$ plus de pas de temps, et une résolution $10 \times$ plus grande). Chacune de ces catégories nécessite des propriétés de performance particulières des codes de calcul sur les prochaines architectures. En particulier, la catégorie (I) nécessite une bonne **scalabilité faible**, tandis que les catégories (II, III, IV) dépendent principalement de la **scalabilité forte** du solveur utilisé.

Axes de recherche pour préparer l'ère *Exascale.* Au vu des différents challenges, caractéristiques et applications de l'ère *Exascale* décrits dans les paragraphes précédents, il est déjà possible de discerner des axes de recherche et de développement à approfondir pour préparer l'utilisation des prochains super-calculateurs. On peut les résumer (voir [Dongarra *et al.*, 2011, Sec. 4]) en différents axes de recherche majeurs portant sur :

- (a) les systèmes logiciels :
 - ⇒ systèmes d'exploitation, d'écriture & lecture de données, environnements externes, ...
- (b) les environnements de développement : ⇒ modèles de programmation, compilateurs, structures et langages de programmation, ...
- (c) les logiciels applicatifs :
 ⇒ algorithmes des codes de calcul, logiciels d'analyse, de visualisation et gestion de données, ...
- (d) les domaines transverses :
 ⇒ résilience, gestion énergétique, optimisation de performances, facilité de programmation, ...

lci, on se focalisera principalement sur l'axe (c), et plus particulièrement sur les **recommandations de développement pour les algorithmes des codes de calculs**.

Parmi l'ensemble des axes de recherche possibles,

"Scalability is perhaps the most obvious driver for algorithms." [Dongarra et al., 2011, Sec. 4.3.1.1, §.2]

Le terme scalabilité doit être ici interprété de manière générale, *i.e.* à la fois en terme de vitesse de calcul et en terme énergétique. Au vu du nombre de processus qui seront à disposition des applications *Exascale*, **un accroissement de la scalabilité (forte et/ou faible) sera particulièrement bénéfique**, non seulement pour que les codes de calcul soient plus efficaces, mais aussi pour qu'ils minimisent la consommation énergétique. En effet :

"One of the most challenging demands is power; algorithms that minimize power use need to be developed." [Dongarra *et al.*, 2011, Sec. 4.3.1.1, §.4]

En particulier, la consommation énergétique devant être principalement dominée par les communications de données entre processus (*cf.* Sec. 1.1.2 §.2, (C)), les **algorithmes minimisant les volumes et fréquence d'échange des communications** pourraient être privilégiés.

Enfin, on mentionnera ce dernier point de développement :

"To make effective use of likely exascale hardware, methods that make more efficient use of memory [...], as well as the development of more predictive analytic performance models, will be key." [Dongarra *et al.*, 2011, Sec. 4.3.1.2, §.2]

Ainsi, les méthodes optimisant l'utilisation de la mémoire (*e.g.* discrétisations spatiales d'ordre élevé [Dongarra *et al.*, 2014, Sec. 4.3.3], ...) pourraient être privilégiées. Enfin, on conclura par le dernier point de cette citation, qui met en évidence la nécessité de **modéliser et prédire la performance des méthodes** utilisées. Ceci devra être fait non seulement pour les stratégies de parallélisation actuelles, mais aussi pour les nouvelles que l'on cherchera à développer.

1.2 Calcul haute performance pour la CFD

1.2.1 Contexte de la CFD

La mécanique des fluides numérique (CFD) correspond à l'ensemble des méthodes et applications cherchant à résoudre numériquement les équations de Navier-Stokes, régissant le comportement d'un fluide (liquide ¹⁴ ou gazeux). Généralement, son objectif est de comprendre et prédire les interactions (internes et/ou externes) du fluide dans un cadre physique particulier. Elle constitue l'un des domaines de recherche prédominants au sein de la communauté scientifique actuelle ¹⁵.

Historiquement, les premiers travaux que l'on peut comparer à la CFD moderne sont dûs à Lewis Fry Richardson. Dans un livre qu'il publie entre les deux guerres mondiales [Richardson, 1922], l'auteur décrit le principe de division de l'espace en une multitude d'éléments discrets (cellules, mailles), principe dénommé par la suite discrétisation, grâce auquel il résout une forme simplifiée des équations de Navier-Stokes afin d'obtenir une prévision météorologique. Bien que les résultats ne soient pas à la hauteur des attentes exprimées, son travail est universellement reconnu comme pionnier au sein de la communauté scientifique (voir [Hunt, 1998] pour plus d'informations).



FIGURE 1.3 – Quand HPC et CFD se combinent : simulation numérique d'un écoulement fluide autour d'un modèle d'avion de transport, avec représentation de coupes du maillage spatial tri-dimensionnel utilisé. La visualisation à elle seule a nécessité l'utilisation d'un supercalculateur, voir https://www.nasa.gov/aero/aeronautical-simulation.html pour plus d'informations. Image provenant de https://www.nasa.gov/sites/default/files/hi_lift_mesh_transport_sim.jpg.

^{14.} De ce fait, le terme Hydrodynamic Simulations est parfois utilisé pour décrire des applications en CFD.

^{15.} Pour l'année 2017 seule, environ 18000 publications scientifiques sont liées aux mots clefs "Computational Fluid Dynamics" selon https://scholar.google.com.

Depuis sa genèse, la CFD n'a eu de cesse d'évoluer dans ses techniques de discrétisation (différences finies, volumes finis, éléments finis, ...) ainsi que dans ses champs d'applications (Météorologie, Aéronautique & Spatial, Biologie & Médecine pour en citer quelques-uns). En particulier, le développement quelques années plus tard des ressources de calcul informatiques a rapidement été mis à profit pour les applications CFD, à tel point que le HPC est aujourd'hui indispensable pour de nombreuses applications dans des cadres de recherche ou industriels (*cf.* Fig. 1.3 pour exemple). Ce besoin d'une capacité croissante de calcul est motivé non seulement par la plus grande complexité physique des problèmes considérés, mais aussi par la particularité des méthodes développées jusqu'à aujourd'hui pour résoudre les équations de Navier-Stokes.

Différentes approches pour résoudre le même problème. C'est aussi à Richardson que l'on doit la célèbre phrase :

"Big whirls have little whirls that feed on their velocity, and little whirls have lesser whirls and so on to viscosity - in the molecular sense." [Richardson, 1922, p.66]

Adaptée d'un poème satirique de Jonathan Swift, elle décrit avec un certain lyrisme le comportement d'un écoulement turbulent, et sera à la base des premiers grands développement pour la théorie de la turbulence dans les années 1940 [Kolmogorov, 1941].

Car de fait, la turbulence est une composante physique majeure résultant de la résolution des équations de Navier-Stokes. Celle-ci se caractérise par le développement de structures tourbillonnaires de grandes et petites tailles, avec un niveau énergétique qui décroît avec la taille des structures considérées. Comme le sous-entend Richardson, cette turbulence est principalement caractérisée par un transfert d'énergie des grandes échelles vers les plus petites. Une fois cette énergie transférée vers les plus petites échelles (dites de "Kolmogorov"), elle est dissipée par diffusion moléculaire ¹⁶.

FIGURE 1.4 – Approches principales pour la résolution des écoulements turbulents. Représentation d'une répartition classique d'énergie turbulente en fonction des différentes échelles.

Afin de représenter fidèlement ce processus, il est donc nécessaire de résoudre ou modéliser avec précision l'ensemble de ces échelles, ce qui a motivé le développement des trois approches principales, représentées en Fig. 1.4. Chacune de ces méthodes se distingue par la plage des échelles de la turbulence qui sont modélisées.

Dans l'approche RANS (*Reynolds Averaged Navier-Stokes*), l'ensemble des structures de la turbulence est modélisé. On résout ainsi une forme moyennée des équations, plutôt que la forme originale. Cette approche permet donc de résoudre des problèmes stationnaires, mais aussi instationnaires (Unsteady RANS, ou URANS), si les échelles de temps étudiées se distinguent de celles de la turbulence.

^{16.} Ce processus, appelé "cascade énergétique" [Pope, 2000], sera revu plus en détail dans le Chap. 4.

Les deux autres approches résolvent quant à elles la forme originale des équations de Navier-Stokes, et permettent ainsi de représenter l'évolution temporelle de la turbulence. Alors que l'approche LES (*Large Eddy Simulation* ou Simulation aux Grandes Échelles) résout uniquement la partie supérieure des échelles turbulentes jusqu'à une certaine taille, l'approche DNS (*Direct Numerical Simulation*) cherche à résoudre l'ensemble des structures de l'écoulement.

Des problèmes de grande taille pour des applications particulières. Comme on peut le voir en Fig. 1.5, les différentes approches mentionnées ne permettent pas de représenter un écoulement turbulent avec le même niveau de détail, et de ce fait nécessitent des niveaux de résolution différents pour le maillage spatial.

FIGURE 1.5 – Différence entre les trois approches principales en CFD : visualisation d'un jet turbulent simulé par DNS (gauche), LES (milieu) et RANS (droite). Extrait de [Maries et al., 2012].

L'approche RANS, qui reste la moins coûteuse des trois, est déjà largement utilisée à la fois pour la recherche et dans le monde industriel. Cependant, les limitations apportées par la modélisation de la turbulence favorisent l'utilisation de méthodes LES dès qu'une représentation plus fidèle de l'écoulement est requise. Le développement des ressources HPC a par ailleurs favorisé la généralisation des simulations LES en recherche mais aussi dans l'industrie, au prix d'une résolution spatiale bien plus contraignante.

Cependant, la LES souffre également de difficulté de modélisation, ici localisée aux plus petites échelles de la turbulence. En particulier, la structure de la turbulence change selon le contexte physique (proche ou loin d'une paroi, en présence éventuelle de chocs, ...), et il reste encore difficile de correctement construire les modèles LES pour les rendre à la fois génériques et précis. L'approche DNS est de plus en plus utilisée depuis quelques décennies, pour comprendre les mécanismes fondamentaux de la turbulence et construire des modèles (voir [Orszag et Patterson Jr, 1972, Kim *et al.*, 1987] pour de premiers exemples). La résolution spatiale requise pour une DNS est cependant proportionnelle à la taille des plus petites structures. De ce fait, une DNS requiert une taille de problème d'autant plus importante que le nombre de Reynolds augmente [Pope, 2000], puisque la gamme d'échelle associée s'élargit.

Projections pour les problèmes futurs. Les hauts niveaux de résolution apportés par les approches LES/DNS permettent aujourd'hui de comprendre et décrire la turbulence avec un degré de précision et de représentation qu'il est parfois difficile d'obtenir expérimentalement. C'est pourquoi les simulations LES/DNS, qualifiées de "haute fidélité", sont parfois les seules approches permettant d'étudier les nouveaux problèmes qui intéressent la communauté scientifique (plus grandes complexités, plus grands nombres de Reynolds).

Référence	Problème considéré	Taille de maillage (points)
[Ishihara <i>et al.</i> , 2009]	Turbulence homogène isotrope	$\sim 69 \cdot 10^9$
[Pirozzoli et Bernardini, 2013]	Couche limite turbulente sur plaque plane	$\sim 34 \cdot 10^9$
[Larsson <i>et al.</i> , 2013]	Interaction choc-turbulence	$\sim 2.5 \cdot 10^9$
[Lee et Moser, 2015]	Écoulement de canal turbulent	$\sim 120 \cdot 10^9$

En Tab. 1.2 est présentée une liste des plus grandes DNS ayant été réalisées ces dernières années. On peut remarquer la taille considérable (en nombre de points¹⁷) des problèmes traités, et déjà entrapercevoir la taille des problèmes de demain, qui dépassera sûrement prochainement le billion de points (10¹²).

FIGURE 1.6 – Estimation d'une tendance HPC pour les problèmes CFD. Points rouges : 3D homogènes, points noirs : 2D homogènes, points verts : géométrie complexe. En violet ($\mathcal{O}(..)$) : nombre de degrés de liberté par cœur. Tendance pour les géométries complexes extrapolée à partir de celle estimée pour les géométries simples.

Par ailleurs, on peut examiner d'un point de vue HPC les caractéristiques des "grands" problèmes considérés jusqu'à présent, ce qui est proposé en Fig. 1.6. Bien que le nombre de degrés de liberté des problèmes augmente conjointement avec la puissance globale de calcul des systèmes utilisés, la charge allouée à chaque processus (nombre de degrés de liberté par cœur) semble diminuer progressivement. Ainsi, la taille des problèmes augmente, mais les ressources de calcul allouées (nombre de processus disponibles) augmentent encore plus vite. Cette tendance est d'autant plus marquée avec l'arrivée des nouveaux types d'accélérateurs (GPGPU, XEON PHI), où la problématique de saturation de la parallélisation spatiale (scalabilité forte) risque d'être très limitante pour les futures simulations CFD à haute fidélité.

^{17.} Cette taille peut aussi être exprimée en nombre de degrés de liberté, ce qui correspond au nombre de points multiplié par le nombre de variables considérées.

1.2.2 Le solveur Hybrid

Dans le cadre de la thèse, on utilise un solveur CFD permettant de résoudre les équations de Navier-Stokes dans leur forme compressible. Ce dernier a été choisi au vu des caractéristiques suivantes :

- possibilité de traiter des problèmes canoniques,
- utilisation de méthodes classiques utilisées en CFD,
- parallélisation en espace déjà implémentée et optimisée,
- possibilité de calculs DNS ou LES.

Le choix s'est donc porté sur le solveur HYBRID, que l'on décrit dans cette section, tout en explicitant l'objectif visé, d'appliquer une stratégie de parallélisation différente de celle déjà utilisée.

Méthodes numériques. Écrit en C++ par [Larsson *et al.*, 2007], HYBRID est aujourd'hui développé et maintenu par une communauté d'utilisateurs répartis au sein de différents instituts de recherche (*University of Maryland, University of Purdue, University of Southern California, ISAE-Supaero*). Le code utilise des maillages structurés pour résoudre les équations de Navier-Stokes pour un fluide compressible. En l'absence de discontinuités, HYBRID utilise des schémas différences finies centrés d'ordre élevé (ordre 6 par défaut), tandis que des schémas WENO [Jiang et Shu, 1996] sont utilisés dans les zones de choc. L'intégration temporelle est effectuée avec un schéma Runge-Kutta explicite d'ordre 4. Plus de détails sur les discrétisations spatiales et temporelles sont donnés dans [Johnsen *et al.*, 2010]. HYBRID peut être utilisé avec des techniques de type LES ou DNS pour la résolution d'écoulements turbulents.

Applications. Le solveur a été développé dans le but d'étudier des problèmes de turbulence fondamentale, comme l'interaction choc/turbulence [Larsson et Lele, 2009, Larsson *et al.*, 2013], des écoulements turbulents supersoniques de canal [Trettel et Larsson, 2016] ou encore des ondes acoustiques et thermoacoustiques dans un fluide compressible [Migliorino et Scalo, 2017]. Bien qu'il ait été originellement conçu pour résoudre des écoulements fluides en présence de chocs, HYBRID permet aussi de résoudre avec une grande précision des problèmes avec des solutions régulières, comme montré par [Johnsen *et al.*, 2010].

1.2.2.a Stratégie de parallélisation actuelle d'Hybrid.

Le parallélisme au sein d'HYBRID est basé sur une décomposition de domaine en espace, et un modèle de programmation à mémoire distribuée utilisant le standard MPI. Le code utilise des communications non-bloquantes pour échanger les données aux interfaces, ce qui permet un recouvrement des communications et du calcul.

Propriétés de scalabilité. De par les méthodes numériques qu'il utilise (maillages structurés, discrétisation spatio-temporelle explicite), ce solveur a montré d'excellentes propriétés de scalabilité forte et faible. Ces dernières ont été examinées en détails par [Bermejo-Moreno *et al.*, 2013a], avec des tests réalisés sur le super-calculateur Sequoia (*cf.* Tab. 1.1), inter-connecté provisoirement avec le super-calculateur voisin Vulcain, portant le nombre total de cœurs disponibles à près de 2 millions.

En Fig. 1.7 sont représentés les résultats des tests de scalabilité faible d'HYBRID, en considérant un nombre de points par cœur élevé (autour de 2 millions). Les résultats montrent d'excellentes propriétés de scalabilité au vu du nombre de processus utilisés, avec une efficacité parallèle restant supérieure à 80%. À charge maximale, la taille des problèmes considérés reste cependant particulièrement grande. En comparaison, l'étude de scalabilité forte restreint donc la taille du problème, comme présenté en Fig. 1.8.

Cores	N_x	N_y	N_z	N	Blocks	Efficiency
32	256	512	512	$67,\!108,\!864$	$2 \times 4 \times 4$	1.00
64	512	512	512	$134,\!217,\!728$	$4 \times 4 \times 4$	1.00
128	512	512	1,024	$268,\!435,\!456$	$4 \times 4 \times 8$	1.00
256	512	1,024	1,024	$536,\!870,\!912$	$4 \times 8 \times 8$	1.00
512	1,024	1,024	1,024	1,073,741,824	$8 \times 8 \times 8$	1.00
1,024	1,024	1,024	2,048	2,147,483,648	$8 \times 8 \times 16$	1.00
2,048	1,024	2,048	2,048	$4,\!294,\!967,\!296$	$8 \times 16 \times 16$	1.00
4,096	2,048	2,048	2,048	$8,\!589,\!934,\!592$	$16 \times 16 \times 16$	1.00
8,192	2,048	2,048	4,096	$17,\!179,\!869,\!184$	$16 \times 16 \times 32$	1.00
16,384	2,048	4,096	4,096	$34,\!359,\!738,\!368$	$16 \times 32 \times 32$	1.00
32,768	4,096	4,096	4,096	68,719,476,736	$32 \times 32 \times 32$	0.99
$65,\!536$	4,096	4,096	8,192	$137,\!438,\!953,\!472$	$32 \times 32 \times 64$	0.98
524,288	8,192	8,192	16,384	1,099,511,627,776	$64 \times 64 \times 128$	0.98
786,432	8,192	12,288	16,384	$1,\!649,\!267,\!441,\!664$	$64{\times}128{\times}96$	0.94
1,048,576	8,192	16,384	16,384	$2,\!199,\!023,\!255,\!552$	$64{\times}128{\times}128$	0.97
$1,\!572,\!864$	12,288	16,384	$16,\!384$	$3,\!298,\!534,\!883,\!328$	$128{\times}128{\times}96$	0.93
$1,\!966,\!080$	15,360	16,384	16,384	$4,\!123,\!168,\!604,\!160$	$128 \times 96 \times 160$	0.83

FIGURE 1.7 – Résultats de scalabilité faible d'HYBRID pour une répartition de 128³ points par cœur. Extrait de [Bermejo-Moreno et al., 2013a].

Pour l'étude de scalabilité forte effectuée par [Bermejo-Moreno *et al.*, 2013a], la taille de problème considérée (autour de 550 milliards de points) est globalement représentative des problèmes à considérer dans les années à venir. On peut remarquer que le *speedup* d'Hybrid a un comportement proche de l'idéal jusqu'à l'utilisation de près d'un million et demi de cœurs, mais commence à se détériorer pour un nombre de processus aux alentours de deux millions.

FIGURE 1.8 – Speedup d'HYBRID pour un test de scalabilité forte avec un problème de taille fixée à 8193³ points. Figure extraite et légèrement modifiée ¹⁸ à partir de [Bermejo-Moreno et al., 2013a].

1.2.2.b Objectifs d'une nouvelle stratégie de parallélisation.

Les résultats de scalabilité laissent ainsi entrevoir une limitation pour l'utilisation d'HYBRID dans un cadre *Exascale*, et mettent en évidence deux aspects pour lesquels HYBRID pourrait bénéficier d'une nouvelle stratégie de parallélisation.

Résoudre des problèmes de plus grande taille dans des temps de restitution raisonnables. Alors que l'utilisation de près de deux millions de cœurs pour résoudre le problème de taille 8193³

^{18.} Des résultats correspondant à l'utilisation d'hyperthreading ont été retirés.

présente une efficacité parallèle peu détériorée, cette résolution nécessiterait plusieurs jours (ou semaines, ou mois ...) de calcul sur l'ensemble du super-calculateur. Ceci reste donc peu envisageable dans le contexte actuel. Bien qu'une machine *Exascale* permettrait de disposer de plus de cœurs, l'accélération des calculs induite serait limitée par la perte de scalabilité forte de la parallélisation spatiale utilisée avec encore plus de processus. De ce fait, l'emploi d'une stratégie de calcul parallèle plus efficace permettrait éventuellement de réduire le coût de résolution de ce type de problème.

Résoudre les problèmes actuels plus rapidement. Certains problèmes considérés dans la communauté CFD n'utilisent pas un nombre de points aussi important en espace, mais doivent être résolus sur un nombre considérable de pas de temps pour obtenir une évaluation statistique précise de la turbulence (*e.g.* [Bermejo-Moreno *et al.*, 2013b]). Le nombre maximum de processus utilisés avec la parallélisation spatiale, limité par la scalabilité forte du code de calcul, restera donc bien en-dessous du nombre de processus à disposition. De ce fait, augmenter la scalabilité forte du solveur avec une stratégie de parallélisation alternative pourrait permettre de réduire significativement les temps de restitution, et ainsi rendre moins onéreuses des applications nécessitant de ré-itérer plusieurs fois la résolution de problème à taille équivalente (*e.g.* optimisation de forme, quantification d'incertitudes, ...).

En résumé, un solveur actuel CFD tel qu'HYBRID pourrait donc potentiellement bénéficier d'une stratégie de parallélisation différente de l'utilisation exclusive de la parallélisation spatiale, ce qui permettrait de repousser sa limite de scalabilité forte dans le futur. Le développement de nouvelles stratégies de parallélisation n'est d'ailleurs pas un concept nouveau : depuis plusieurs années déjà, la communauté scientifique étudie des degrés de parallélisme moins classiques que ceux apportés par la décomposition en espace. En particulier, l'utilisation d'une stratégie basée sur une parallélisation temporelle, présentée en section suivante, est une alternative intéressante.

1.3 Emergence de la parallélisation en temps

Dans la plupart des phénomènes physiques simulés par ordinateur, l'objectif est de représenter avec précision l'évolution de quantités physiques (solutions), à partir d'un état initial (solution initiale)¹⁹. De par la complexité des problèmes étudiés, les méthodes numériques à notre disposition ne permettent de calculer précisément cette évolution que progressivement, un pas de temps après l'autre. Pour un problème donné (généralement, basé sur des équations aux dérivées partielles), l'ensemble des pas de temps utilisés pour le résoudre, variant selon l'application considérée, est appelé **domaine temporel**. C'est donc sur ce domaine que porte la décomposition introduite par la parallélisation en temps.

1.3.1 Un nouveau concept de parallélisme.

Principes généraux. Tout comme sa cousine spatiale, la parallélisation en temps décompose donc le domaine temporel en plusieurs sous-domaines, et attribue la résolution du problème sur chaque sous-domaine à un processus particulier. Une représentation schématique est donnée en Fig. 1.9.

Cependant, cette décomposition ne se fait pas aussi naturellement que pour la parallélisation spatiale : afin de pouvoir calculer la solution à un pas de temps donné, il est nécessaire de connaître la solution au pas de temps précédent. Par conséquent il faut repenser la résolution temporelle du problème différemment, en particulier développer des algorithmes permettant cette décomposition.

La première mention de parallélisation en temps au sein de la communauté scientifique est due à [Nievergelt, 1964]. Son idée, dont sont quelque peu dérivées la plupart des solutions actuelles, est la suivante :

^{19.} Certains problèmes d'intérêt peuvent aussi être indépendants du temps, mais on ne les considèrera pas ici.

FIGURE 1.9 – Représentation schématique de la parallélisation temporelle.

- (E1) donner une solution initiale approchée et peu coûteuse à chaque sous-domaine temporel,
- (E2) résoudre parallèlement le problème sur chaque sous-domaine,
- (E3) corriger la solution finale de chaque sous-domaine.

"By introducing redundancy in computation, we try to recast an essentially serial algorithm into a form in which it consists of several subtasks which can be performed in parallel, i.e. can be computed without knowledge of the results of other subtasks." [Nievergelt, 1964, Introduction, §.3]

Bien que le nombre d'opérations effectuées puisse être supérieur à celui d'une résolution séquentielle classique (de part (E1) et (E3)), le fait de pouvoir répartir les tâches de l'étape (E2) sur plusieurs processus permet donc à terme d'obtenir un temps de restitution plus court, d'où un certain *speedup*, toutefois limité par les étapes n'étant pas effectuée en parallèle.

La méthode présentée par Nievergelt ne peut cependant être appliquée qu'à des problèmes simples, ce dernier la présentant plutôt comme un prototype en vue du développement de solutions futures. Ainsi, à une époque où on commence à peine à voir apparaître les premières machines parallèles, Nievergelt envisage déjà les problématiques de parallélisme massif actuel :

"It is believed that more general and improved versions of these methods will be of great importance when computers capable of executing many computations in parallel become available." [Nievergelt, 1964, Conclusion, §.1]

En effet, cinquante ans plus tard, la communauté HPC envisage concrètement l'apport potentiel de ce type de méthode [Dongarra *et al.*, 2014, Sec. 4.3.2], au vu des nombreuses solutions développées à l'heure actuelle, certaines pouvant être appliquées à une large gamme de problèmes intéressant la communauté scientifique. Les principales solutions seront décrites plus en détails dans le prochain chapitre.

Une caractéristique particulière. Une première remarque peut être faite quant aux communications induites par la parallélisation en temps, qui comprennent les données aux interfaces du domaine temporel, c'est-à-dire l'intégralité des champs spatiaux d'une solution à un temps donné. Chaque communication représente ainsi un volume de données considérable si l'on résout un problème de grande taille, qui peut avoir un impact non-négligeable dans un contexte *Exascale*. Cependant, il est possible de profiter de deux aspects intéressants de ce type de communication.

Tout d'abord, comme représenté en Fig. 1.9, celles-ci se font uniquement dans une direction entre chaque processus, qui n'a au plus que deux "voisins", à l'inverse des communications spatiales où chaque processus envoie et reçoit des données de l'ensemble de ses "voisins" (dont le nombre peut être particulièrement important pour des maillages tri-dimensionnels, éventuellement non-structurés). De ce fait, ces communications plus "organisées" peuvent être plus facilement optimi-sées.

Enfin, la décomposition temporelle peut être choisie de manière à ce que le temps de communication entre processus soit bien inférieur au temps de calcul sur chaque sous-domaine. Tout ceci est bien entendu conditionné par la solution de parallélisation en temps utilisée, ce qui pousse à choisir et/ou développer des méthodes parallèles en temps minimisant la fréquence des communications temporelles.

1.3.2 Stratégie de parallélisation combinée espace-temps

L'utilisation de la parallélisation en temps pour développer des méthodes plus efficaces dans un cadre *Exascale* n'a pas nécessairement intérêt à être exclusive. En effet, comme montré en Sec. 1.2.2, la parallélisation spatiale permet déjà d'obtenir des *speedup* quasi idéaux, et ce même avec un nombre de processus important. Il convient donc d'optimiser au mieux l'efficacité parallèle pouvant être fournie par la décomposition de domaine spatial, puis d'utiliser la parallélisation temporelle en "sur-couche" dans une configuration bien choisie.

FIGURE 1.10 – Schéma d'une stratégie de parallélisation espace-temps.

On considère ainsi une stratégie parallèle basée sur l'utilisation de ces deux décompositions, que l'on représente en Fig. 1.10. Pour un problème de taille donnée, si le nombre de processus disponibles est trop important pour maintenir l'efficacité de la parallélisation spatiale, on limite la décomposition en espace, et on applique la décomposition de domaine en temps pour démultiplier le nombre de processus utilisés. Ceci permettrait à terme d'améliorer la scalabilité forte du solveur, comme représenté en Fig. 1.11.

FIGURE 1.11 – Objectif de l'utilisation de la parallélisation en temps pour la scalabilité forte d'un solveur.

Ce type de stratégie requiert néanmoins une certaine performance de l'algorithme de parallélisation en temps utilisé. Comme on le verra plus en détail par la suite, il apparaît actuellement peu réalisable d'obtenir une efficacité idéale (ou proche) de cette parallélisation espace-temps au vu des problèmes complexes considérés. Cependant, la parallélisation spatiale se heurtera nécessairement à une chute importante de son efficacité dans un contexte *Exascale*. C'est pourquoi la parallélisation spatio-temporelle, permettant de garder un certain niveau d'efficacité, pourrait permettre de mieux utiliser les prochaines ressources disponibles. En particulier, les algorithmes de parallélisation en temps ont un potentiel (qui reste à évaluer) de réduire la quantité des données communiquées entre processus, ce qui risque d'être d'autant plus profitable dans le contexte *Exascale*, où le coût énergétique d'un calcul sera dominé non pas par son efficacité parallèle mais plutôt par le coût de communication des données.

On conclura donc en affirmant que la parallélisation en temps n'est pas un concept idéal, mais elle risque fortement d'être un passage obligé au vu des prochaines architectures de calcul massivement parallèles.

1.4 Synthèse

Ce chapitre nous a permis d'introduire le contexte actuel autour du calcul haute performance, et les projections que l'on peut déjà réaliser quant aux caractéristiques des architectures de calcul massivement parallèles qui seront à disposition dans les années à venir. Nous avons ensuite mis en évidence le besoin de développer de nouvelles stratégies de parallélisation, en particulier dans le cadre de la simulation numérique des écoulements en mécanique des fluides (CFD). Enfin, nous avons introduit le concept de parallélisation en temps, qui apporte un degré de parallélisme supplémentaire dont les solveurs actuels pourraient bénéficier dans un contexte de calcul Exascale.

Les chapitres suivants cherchent ainsi à apporter plus de lumière quant à l'apport d'une stratégie parallèle basée sur la parallélisation en espace et en temps pour la CFD. Tout d'abord, en Chap. 2 est proposée une comparaison des solutions de parallélisation en temps actuelles, afin de sélectionner un premier candidat pour notre étude. Puis, deux chapitres sont consacrés à l'étude de cette solution dans le cadre applicatif considéré. En Chap. 3 est présentée une analyse de la solution parallèle retenue sur des problèmes académiques simples mais représentatifs en relation avec la CFD, afin de mieux comprendre les mécanismes mis en jeu au sein de cette stratégie. Puis, en Chap. 4, cette stratégie est appliquée et évaluée pour la résolution de problèmes tridimensionnels réalistes.

CHAPITRE 2

Synthèse des principaux algorithmes actuels de parallélisation en temps et étude d'applicabilité

Table des matières

2.1	PARAREAL, le cadet célèbre						
	2.1.1 A la base de l'algorithme : la méthode de tirs multiples (<i>Multiple Shooting</i>). 28					
	2.1.2 Description de PARAREAL	30					
	2.1.3 Variantes	31					
	2.1.4 Applications, forces et faiblesses	33					
2.2	PARAExp, la solution "idéale" (dans le cas linéaire)	35					
	2.2.1 Description	35					
	2.2.2 Applications, forces et faiblesses	37					
2.3	RIDC, l'intégration temporelle alternative	38					
	2.3.1 Description	38					
	2.3.2 Applications, forces et faiblesses	39					
2.4	1 PFASST, la complexité au service de l'efficacité						
	2.4.1 Description	40					
	2.4.2 Applications, forces et faiblesses	40					
2.5	MGRIT, la solution "universelle"?	42					
	2.5.1 Description	42					
	2.5.2 Applications, forces et faiblesses	43					
2.6	Vers un premier choix d'algorithme	44					
	2.6.1 Critères de sélection	44					
	2.6.2 Synthèse concernant les différents algorithmes présentés	45					
	2.6.3 Justification du choix de PARAREAL	46					
2.7	Synthèse	47					

Ce chapitre tend à présenter les principaux algorithmes de parallélisation en temps existants au moment de la rédaction du manuscrit. Il se base en grande partie sur le travail de synthèse effectué par [Gander, 2015], renforcé par l'ajout de contributions plus récentes. En plus de décrire de manière concise le fonctionnement de ces algorithmes, ce chapitre fait une synthèse de leurs applications, en distinguant les problèmes de nature parabolique et hyperbolique, linéaire ou non.

Cinq algorithmes, considérés comme les plus significatifs, y sont décrits. Ce choix a été basé sur les critères suivants :

- 1. simplicité de mise en œuvre et compatibilité avec des solveurs d'intégration en temps explicites,
- 2. impact de l'algorithme dans la littérature scientifique, et nombre d'applications aux équations de Navier-Stokes,
- 3. performances sur les problèmes modèles tests.

Ce choix et le détail de ces critères seront discutés en Sec. 2.6. Cela permettra de dégager un premier choix d'algorithme qui sera à la base des stratégies de parallélisation espace-temps étudiées dans la suite de ce manuscrit.

Problème considéré : Pour l'ensemble des sections de ce chapitre, on s'intéresse au problème d'évolution :

$$\frac{dU}{dt} = f(U(t), t), \quad U(0) = U_I, \quad t \in [0, T],$$
(2.1)

avec $f : \mathbb{R}^p \times \mathbb{R}^+ \to \mathbb{R}^p$ un opérateur a priori non-linéaire, $U(t) \in \mathbb{R}^p$ une solution dépendante du temps $t, U_I \in \mathbb{R}^p$ la condition initiale, p le nombre total de degrés de liberté et $T \in \mathbb{R}^+$ le temps final d'intégration. L'Équation Différentielle Ordinaire (EDO) (2.1) peut être obtenue à partir de la discrétisation spatiale d'un système d'Équations aux Dérivées Partielles (EDP) par le biais de la "méthode des lignes" [Schiesser, 2012].

2.1 PARAREAL, le cadet célèbre

Cette section décrit l'un des algorithmes de parallélisation en temps les plus connus dans la communauté scientifique, introduit par [Lions *et al.*, 2001] dans une brève contribution aux Compte-Rendus de l'Académie de Sciences de Paris. Bien qu'il ne soit pas considéré comme le plus efficace pour la plupart des problèmes, il fut à l'origine d'un engouement notable de la communauté scientifique pour les méthodes parallèles en temps dans les années 2000. Il est donc important de le présenter et d'en expliquer sa genèse. C'est l'objet de la Sec. 2.1.1, l'algorithme PARAREAL étant lui-même décrit en Sec. 2.1.2. En Sec. 2.1.3 seront présentées ses principales variantes, et une discussion sera menée autour des applications et performances de l'ensemble des algorithmes présentés en Sec. 2.1.4.

2.1.1 A la base de l'algorithme : la méthode de tirs multiples (Multiple Shooting)

L'algorithme PARAREAL trouve son origine dans l'introduction par [Nievergelt, 1964] du *Multiple-Shooting* pour résoudre une EDO. Ce dernier propose de considérer que l'ensemble des solutions du problème à des temps fixés sont les inconnues d'un problème plus global, qui sont approximées de manière itérative. Bien que l'algorithme de résolution qu'il propose soit plus coûteux par rapport à un algorithme d'intégration séquentiel classique, le fait de pouvoir exécuter une partie des calculs en parallèle le rend plus rapide.

Trente ans plus tard, l'idée est reprise et développée par [Chartier et Philippe, 1993], qui étendent la méthode du *Multiple Shooting* à un système d'EDO. Leur algorithme, qu'ils dénomment par ailleurs *Parallel-Shooting*, sera à la base de PARAREAL quelques années plus tard.

Base du *Multiple-Shooting.* On considère donc le système d'EDO (2.1). L'intervalle de temps [0,T] est décomposé en N tranches temporelles $[t_{n-1},t_n]$, avec $n \in [\![1,N]\!]$. On les appellera par la suite des *time-slices*. Dans le cadre d'un calcul parallèle, N est le nombre de processeurs dédié à la parallélisation en temps uniquement. Par la suite, on notera U_n l'approximation de U au temps t_n , *i.e.*, $U_n \approx U(t_n)$. Les conditions qui garantissent la continuité temporelle de la solution aux extrémités de chaque *time-slice* s'écrivent

$$U_0 = U_I, \tag{2.2}$$

$$\forall n \in \llbracket 1, N \rrbracket, \ U_n = \underset{t_{n-1} \to t_n}{\mathcal{S}} (U_{n-1})$$
(2.3)

avec $\mathcal{S}_{t_{n-1} \to t_n} : U_{n-1} \to U_n$ l'opérateur permettant d'obtenir la solution au temps t_n à partir de celle au temps initial t_{n-1} . Le vecteur des solutions initiales $\mathbf{U} = (U_0, U_1, ..., U_N)$ devient alors l'inconnu du problème suivant :

$$F(\mathbf{U}) = \begin{pmatrix} U_0 - U_I \\ \vdots \\ U_n - \frac{\mathcal{S}}{t_{n-1} \to t_n} (U_{n-1}) \\ \vdots \end{pmatrix} = 0$$
(2.4)

Ce problème, a priori non-linéaire, peut donc être vu comme un problème de point fixe. Il est résolu par la méthode de Newton, ce qui produit l'algorithme itératif suivant :

$$\mathbf{U}^{k+1} = \mathbf{U}^k - J_F^{-1}(\mathbf{U}^k) \ F(\mathbf{U}^k), \ k \in \mathbb{N},$$
(2.5)

 J_F désignant la Jacobienne de F définie en (2.4), et pouvant s'écrire ainsi :

$$J_{F}(\mathbf{U}^{k}) = \begin{bmatrix} I_{p} & & & \\ \ddots & \ddots & & \\ & -\frac{\partial_{t_{n-1} \to t_{n}}}{\partial U} \Big|_{U_{n-1}^{k}} & I_{p} & \\ & & \ddots & \ddots \end{bmatrix} \in \mathcal{M}_{(N+1)p}(\mathbb{R}).$$
(2.6)

En multipliant (2.5) par J_F , il est possible d'écrire une relation de récurrence entre chaque itéré :

$$U_0^{k+1} = U_I,$$

$$\forall n \in [\![1, N]\!], \quad U_n^{k+1} = \frac{\mathcal{S}}{t_{n-1} \to t_n} (U_{n-1}^k) + \frac{\partial \frac{\mathcal{S}}{t_{n-1} \to t_n}}{\partial U} \bigg|_{U_{n-1}^k} (U_{n-1}^{k+1} - U_{n-1}^k).$$
(2.7)

Spécialisation de l'algorithme. La forme de l'itération (2.7) fait donc intervenir deux termes principaux. Le premier correspond à l'évaluation de l'opérateur S sur l'un des itérés U_{n-1}^k , qui peut s'effectuer pour chaque $n \in [\![1, N]\!]$ de manière indépendante, donc parallèle.

Le calcul du deuxième terme de (2.7) correspond à une évaluation du produit de la dérivée (ou Jacobienne) de S :

$$\frac{\partial \sum_{t_{n-1} \to t_n}}{\partial U} \bigg|_{U_{n-1}^k} \in \mathcal{M}_p(\mathbb{R}),$$
(2.8)

avec le vecteur des différences entre deux itérations $(U_{n-1}^{k+1} - U_{n-1}^k)$. L'évaluation de ce terme représente la partie cruciale de l'algorithme, car son coût conditionne l'efficacité parallèle de la méthode (ce point sera détaillé par la suite), et la précision avec laquelle il est évalué influe directement sur la convergence.

[Chartier et Philippe, 1993] proposent de calculer ce terme en assemblant une Jacobienne par différences finies, ce qui rend leur algorithme particulièrement coûteux. L'idée de [Lions *et al.*, 2001], bien qu'obtenue distinctement de celle de [Chartier et Philippe, 1993], propose une forme similaire à celle du *Multiple-Shooting*, mais avec un coût de calcul moindre : l'évaluation de la Jacobienne de l'opérateur S est remplacée par une différence finie basée sur l'utilisation d'un opérateur grossier (et donc moins coûteux). Cela donne alors naissance à PARAREAL, qui est décrit dans la section suivante.

2.1.2 Description de PARAREAL

ລ ເ

On reprend ici le problème (2.1) et la décomposition en *time-slices* décrite dans le premier paragraphe de la Sec. 2.1.1. On note par $\mathcal{F}_{t_{n-1}\to t_n}^{\delta_t}(U_{n-1})$ le résultat d'une intégration approchée de (2.1) sur la time slice $[t_{n-1}, t_n]$ à partir d'une solution initiale donnée U_{n-1} . Cette intégration se fait par le biais d'un intégrateur temporel "précis" \mathcal{F} utilisant un pas de temps "fin" δ_t . De manière similaire, on définit un autre intégrateur temporel "grossier" \mathcal{G} , qui utilise quant à lui un pas de temps grossier $\Delta_t >> \delta_t$. Cette relation entre les deux pas de temps rend donc \mathcal{G} beaucoup moins coûteux que \mathcal{F} , à défaut d'augmenter l'erreur d'approximation de la solution.

L'algorithme PARAREAL consiste tout d'abord à calculer une première solution initiale U_n^0 pour chaque *time-slice* en utilisant \mathcal{G} de la manière suivante :

$$\forall n \in [\![1, N]\!], \ U_n^0 = \mathcal{G}_{t_{n-1} \to t_n}^{\Delta_t}(U_{n-1}^0), \text{ avec } U_0^0 = U_I.$$
 (2.9)

Puis, une itération corrective est appliquée sur chaque time-slice :

$$\forall n \in [\![1, N]\!], \forall k \in [\![1, K]\!], \ U_n^k = \mathcal{F}_{t_{n-1} \to t_n}^{\delta_t}(U_{n-1}^{k-1}) + \mathcal{G}_{t_{n-1} \to t_n}^{\Delta_t}(U_{n-1}^k) - \mathcal{G}_{t_{n-1} \to t_n}^{\Delta_t}(U_{n-1}^{k-1}),$$
(2.10)

avec U_n^k l'approximation de U au temps t_n à la $k^{ième}$ itération de PARAREAL. On retrouve dans (2.10) la formule du *Multiple-Shooting* (2.7) (voir [Gander et Vandewalle, 2007, Remark 2.1]), avec les correspondances suivantes :

$$\mathcal{S}_{t_{n-1} \to t_n}(U_{n-1}^{k-1}) := \mathcal{F}^{\delta_t}_{t_{n-1} \to t_n}(U_{n-1}^{k-1}),$$
(2.11)

$$\frac{\partial \mathcal{S}}{\partial U} (U_{n-1}^k) (U_{n-1}^k - U_{n-1}^{k-1}) := \mathcal{G}^{\Delta_t}_{t_{n-1} \to t_n} (U_{n-1}^k) - \mathcal{G}^{\Delta_t}_{t_{n-1} \to t_n} (U_{n-1}^{k-1}).$$
(2.12)

Une représentation graphique de l'itération PARAREAL est donnée en Fig. 2.1. On peut y observer l'application du solveur fin qui peut s'effectuer de manière indépendante, donc parallèle, sur chaque *time-slice*. Cependant, le parallélisme induit par PARAREAL reste contraint par la nature séquentielle

FIGURE 2.1 – Schéma d'une itération PARAREAL. Représentation pour les time-slices $[t_{n-2}, t_{n-1}, t_n, t_{n+1}]$, entre l'itération k - 1 et k.

de (2.10), en particulier, $\forall n \in [\![1, N]\!]$ le calcul de U_n^k nécessite une évaluation de \mathcal{G} sur U_{n-1}^k , obtenue une fois la *time-slice* précédente mise à jour. C'est pourquoi PARAREAL ne peut avoir une efficacité parallèle intéressante qu'à condition que l'application de \mathcal{G} soit très peu onéreuse relativement à \mathcal{F} , et que le nombre total d'itérations utilisées K reste faible. En particulier, on peut souligner que par construction, PARAREAL converge exactement après N itérations vers la solution obtenue avec une application séquentielle de \mathcal{F} sur [0,T] [Gander et Vandewalle, 2007]. Cependant, une telle utilisation rend l'algorithme moins intéressante que \mathcal{F} seul, et ce quel que soit N.

Bien que ce concept ne soit pas spécifique à PARAREAL, il est possible d'envisager un "redémarrage" (*restart*) de l'algorithme, dans le cas où une seule application de PARAREAL ne peut pas couvrir l'ensemble de l'intervalle d'intégration temporelle [0, T]. Ce dernier est alors décomposé en N_r sous-intervalles, sur lesquels PARAREAL est appliqué successivement en prenant comme solution initiale pour un sous-intervalle celle obtenue après l'ensemble des \mathcal{K} itérations de PARAREAL sur le sous-intervalle précédent. Cette utilisation, que l'on peu aussi qualifier de *windowing*, a par ailleurs été utilisée par [Eghbal *et al.*, 2017] dans un cadre applicatif.

La grande force de PARAREAL réside dans sa simplicité d'application. Ici, \mathcal{F} et \mathcal{G} peuvent être considérés comme de simples boîtes noires, desquelles PARAREAL nécessite uniquement de manipuler les entrées et sorties. Cet algorithme a rapidement été considéré comme une sorte de *framework* autour duquel plusieurs variantes ont été déclinées. Les principales sont présentées en Sec. 2.1.3.

2.1.3 Variantes

Il faudrait sûrement plus qu'une sous-section pour décrire l'ensemble des variantes qui ont été proposées pour PARAREAL depuis son introduction dans la communauté scientifique. Ici, on se limitera aux deux principales. La première consiste en une adaptation de PARAREAL dans le cadre d'utilisation exclusive d'intégrateurs temporels explicites, tandis que la deuxième propose une amélioration de la formule d'itération (2.10) par la réutilisation de solutions déjà calculées.

D'autres variantes, non décrites dans ce manuscrit, méritent d'être mentionnées. En particulier, celle de [Duarte *et al.*, 2011], qui proposent une version de PARAREAL basée sur la décomposition d'opérateurs au sein des solveurs fin et grossier, celles de [Rao et Sandu, 2012, Chen *et al.*, 2015] basées sur l'utilisation d'opérateurs adjoints, ou encore celle de [Dai et Maday, 2013] qui proposent de projeter la solution de PARAREAL après chaque itération sur un sous-espace contenant la solution fine, amélioré après chaque itération. Certains se sont aussi penché sur une définition alternative du solveur grossier, comme [Hernandez, 2014], qui utilise un intégrateur \mathcal{G} implicite avec itérations internes tronquées, tandis que d'autres ont développé des variantes adaptées à un problème particulier, comme [Audouze *et al.*, 2009] pour la parallélisation en temps de problèmes de dynamique moléculaire. Enfin, une dernière variante plus récente [Fischer *et al.*, 2017] propose une version approchée de l'itération PARAREAL utilisant des communications MPI compressées entre processeurs, ce qui permet de réduire le coût de communication de l'algorithme.

2.1.3.a Variante basée sur l'utilisation d'intégrateurs temporels explicites

À l'origine, PARAREAL a été appliqué dans un cadre d'intégration temporelle principalement implicite. Dans ce contexte, aucune restriction n'est présente sur le pas de temps Δ_t afin de garantir la stabilité de l'intégration. Cependant, cela n'est plus le cas dans le cadre d'une utilisation exclusive de solveurs explicites, ceux-ci étant soumis à des conditions de stabilité de type Courant-Friedrich-Levy (CFL) ou Fourier, qui imposent une valeur maximale du pas de temps d'intégration.

Dans l'optique de construire un propagateur grossier explicite, [Ruprecht et Krause, 2012] ont proposé de réduire l'ordre de précision des discrétisations temporelle et/ou spatiale. Il a été aussi suggéré originalement par [Lions *et al.*, 2001] d'utiliser moins de degrés de liberté pour la grille spatiale de \mathcal{G} . Dans cette configuration, on note $\hat{\mathcal{G}}$ le nouveau solveur grossier correspondant, ainsi que \hat{p} son nombre de degrés de liberté associé. Si $\hat{p} < p$, on introduit $\hat{\mathcal{R}}$ et $\hat{\mathcal{I}}$, qui représentent respectivement les opérateurs de restriction et d'interpolation entre les espaces fins et grossiers.

L'étape de prédiction de PARAREAL (2.9) devient alors :

$$\forall n \in [\![1, N]\!], \ U_n^0 = \widehat{\mathcal{I}}_{\substack{t_{n-1} \to t_n}} (\widehat{\mathcal{R}}(U_{n-1}^0)), \ U_0^0 = U_I.$$
(2.13)

De manière similaire, l'itération PARAREAL décrite en (2.10) s'écrit à présent :

$$\forall n \in \llbracket 1, N \rrbracket, \forall k \in \llbracket 1, \widehat{K} \rrbracket, \ U_n^k = \mathcal{F}_{t_{n-1} \to t_n}^{\delta_t} (U_{n-1}^{k-1}) + \widehat{\mathcal{I}} \left(\widehat{\mathcal{G}}_{t_{n-1} \to t_n}^{\Delta_t} (\widehat{\mathcal{R}}(U_{n-1}^k)) - \widehat{\mathcal{G}}_{t_{n-1} \to t_n}^{\Delta_t} (\widehat{\mathcal{R}}(U_{n-1}^{k-1})) \right).$$

$$(2.14)$$

On notera que le choix d'appliquer $\hat{\mathcal{I}}$ avant ou après soustraction des champs grossiers dans (2.14) ne change en aucun cas l'algorithme si $\hat{\mathcal{I}}$ est un opérateur linéaire. Dans le cadre d'une interpolation non-linéaire (voir [Ryavec, 1982] pour exemple), ce choix induirait deux algorithmes différents.

2.1.3.b Améliorations de PARAREAL par réutilisation des solutions des itérations précédentes

Partons d'une observation que l'on peut faire en analysant (2.10) : la solution $\mathcal{F}_{t_{n-1} \to t_n}^{\delta_t}(U_{n-1}^{k-1})$ cal-

culée avec le solveur fin n'est utilisée que pour une seule itération k, puis laissée de côté. C'est ce qui a motivé l'introduction de variantes de PARAREAL permettant de pouvoir réutiliser cette information "coûteuse" pour les itérations suivantes de l'algorithme. Implicitement, ces variantes modifient le solveur \mathcal{G} en y ajoutant des informations obtenues grâce à \mathcal{F} , ce qui contribue à améliorer la précision globale du solveur grossier, ainsi que les propriétés de convergence de PARAREAL [Gander et Petcu, 2008].

La première variante de ce genre a été introduite par [Farhat *et al.*, 2006], et utilise la projection des solutions obtenues aux itérations précédentes sur un sous-espace de Krylov. Une deuxième version plus récente, introduite par [Chen *et al.*, 2014], reprend cette première idée mais en se basant plutôt sur des méthodes de base réduite. Les deux algorithmes sont brièvement présentés dans les deux paragraphes suivants, à partir d'une description plus complète donnée dans [Chen *et al.*, 2014]. Les applications de ces méthodes ainsi que leur efficacité seront discutées en détails en Sec. 2.1.4.

Réutilisation des solutions par sous-espace de Krylov. L'idée derrière la variante de PARA-REAL proposée par [Farhat *et al.*, 2006] est de projeter la solution U_n^k sur le sous-espace de Krylov¹ engendré par l'ensemble des solutions fines disponibles à l'itération k. On note ce sous-espace de la manière suivante :

$$\mathbf{S}^{k-1} = \operatorname{span}\left\{U_{n-1}^{j}, \ n \in [\![1, N]\!], \ j \in [\![1, k-1]\!]\right\}.$$
(2.15)

Une base orthogonale $\{\mathbf{s}_1, ..., \mathbf{s}_r\}$ est construite à partir de cette famille de vecteurs, en utilisant une méthode de décomposition QR [Golub et Van Loan, 2012]. Dès lors, on peut décomposer tout vecteur solution $U \in \mathbb{R}^p$ en une composante notée $\mathbb{P}^{k-1}U \in \mathbf{S}^{k-1}$ et sa partie orthogonale notée $\mathbb{P}^{k-1}_{\perp}U \in \mathbf{S}^{k-1}_{\perp}$. Ceci nous permet de remplacer le solveur \mathcal{G} par un nouveau solveur \mathcal{K} , défini comme suit :

$$\mathcal{K}_{t_{n-1}\to t_n}^{\Delta_t}(U_{n-1}^{k-1}) = \mathcal{G}_{t_{n-1}\to t_n}^{\Delta_t} \left(\mathbb{P}_{\perp}^{k-1}U_{n-1}^{k-1}\right) + \mathcal{F}_{t_{n-1}\to t_n}^{\delta_t} \left(\mathbb{P}^{k-1}U_{n-1}^{k-1}\right)$$
(2.16)

Bien que (2.16) fasse intervenir une évaluation de \mathcal{F} , le coût de \mathcal{K} reste équivalent à celui de \mathcal{G} , si le problème (2.1) considéré est linéaire. En effet, dans cette configuration, on peut écrire

$$\mathcal{F}^{\delta_t}_{t_{n-1}\to t_n}\left(\mathbb{P}^{k-1}U_{n-1}^{k-1}\right) = \mathcal{F}^{\delta_t}_{t_{n-1}\to t_n}\left(\sum_{i=1}^r \alpha_i \mathbf{s}_i\right) = \sum_{i=1}^r \alpha_i \mathcal{F}^{\delta_t}_{t_{n-1}\to t_n}\left(\mathbf{s}_i\right),\tag{2.17}$$

les $\mathcal{F}^{\delta_t}_{t_{n-1} \to t_n}(\mathbf{s}_i)$ étant déjà disponibles par le biais de \mathbf{S}^{k-1} , dont la dimension est incrémentée à chaque itération.

On dénommera par la suite cette variante PARAREAL-KSE (*Krylov-Subspace-Enhanced*). Il a été montré par [Gander et Petcu, 2008] qu'au fur et à mesure que la dimension de S^{k-1} augmente, \mathbb{P}^{k-1} et \mathcal{K} tendent à devenir l'opérateur identité et \mathcal{F} respectivement, ce qui par conséquent améliore la

^{1.} Dans un contexte linéaire, la famille S^{k-1} s'avère être un sous-espace de Krylov engendré à partir du vecteur solution initiale et de la matrice jacobienne du solveur fin.

Voir http://www.cl.eps.manchester.ac.uk/medialand/maths/archived-events/workshops/www.mims.manchester.ac.uk/ events/workshops/spacetime/talks/gander.pdf en *slide* 53 pour plus de détails.
convergence de PARAREAL. Cependant, ceci inclut aussi un coût de calcul supplémentaire dans la partie séquentielle de l'itération (2.10), ainsi qu'un coût de stockage croissant à chaque itération. De plus, l'adaptation de cette variante pour un problème non-linéaire, disponible au sein de l'algorithme PITA (*Parallel Implicit Time-integration Algorithm* [Farhat *et al.*, 2006]) n'a pas été présentée à ce jour pour une formulation du type PARAREAL, et est considérée comme particulièrement complexe à mettre en œuvre (*cf.* [Ruprecht et Krause, 2012], [Chen *et al.*, 2014]).

Réutilisation des solutions fines par base réduite. Afin de réduire le coût de stockage induit par l'algorithme précédent, [Chen *et al.*, 2014] proposent d'utiliser un sous-espace de S^{k-1} , obtenu grâce à une base réduite. Cette dernière est classiquement calculée par l'algorithme *Proper Orthogonal Decomposition* (POD), technique se basant sur une décomposition en valeurs singulières (SVD) de la matrice dont les colonnes sont composées des vecteurs de S^{k-1} . Afin d'éviter un coût trop important utilisé par la POD, [Chen *et al.*, 2014] ont même introduit une autre approche, l'*Empirical Interpolation Method* (EIM) permettant d'obtenir une base réduite à moindre coût. En reprenant les mêmes notations que pour le paragraphe précédent, un nouveau solveur grossier $\hat{\mathcal{K}}$ est ainsi défini comme suit :

$$\widehat{\mathcal{K}}_{t_{n-1}\to t_n}^{\Delta_t}(U_{n-1}^{k-1}) = \mathcal{F}_{t_{n-1}\to t_n}^R \left(\mathbb{P}^{k-1} U_{n-1}^{k-1} \right),$$
(2.18)

avec \mathcal{F}^R un modèle réduit du solveur \mathcal{F} obtenu grâce à la POD ou EIM utilisée sur \mathbf{S}^{k-1} . Une description plus détaillée est donnée dans [Chen *et al.*, 2014], en particulier l'utilisation de modèles réduits pour des problèmes non-linéaires. On remarquera que (2.18) est obtenue en négligeant le premier terme de droite dans (2.16). De plus, pour un problème linéaire, une décomposition similaire à (2.17) est utilisée et vient remplacer \mathcal{F}^R dans (2.18). On dénommera par la suite cette variante PARAREAL-RBE (*Reduced-Basis-Enhanced*).

2.1.4 Applications, forces et faiblesses

Cette section se veut être un bref résumé de l'ensemble des études et applications de PARAREAL ainsi que de ses algorithmes dérivés.

Une prédilection pour les problèmes paraboliques. Dès son introduction dans l'article original de [Chartier et Philippe, 1993], le *Multiple-Shooting* est qualifié de "méthode permettant de résoudre des EDO dissipatives". Bien que l'algorithme puisse être aussi appliqué à d'autres types d'EDO, son application et étude de convergence sont restreintes à des problèmes paraboliques.

De même, PARAREAL fut appliqué pour la première fois à l'équation de la chaleur (linéaire et non-linéaire) par [Lions *et al.*, 2001], et sa convergence y fut étudiée dans ce cadre, avec l'utilisation d'un solveur implicite. Des études de convergence plus générales furent notamment menées par [Bal, 2005] et [Gander et Vandewalle, 2007] pour des problèmes linéaires, en particulier les équations de diffusion et d'advection. Ces études ont été complétées par une analyse de convergence pour des problèmes non-linéaires [Gander et Hairer, 2008], ainsi qu'une étude focalisée sur l'application de PARAREAL à des problèmes hyperboliques [Gander, 2008]. Cette dernière met en évidence les difficultés de convergence de PARAREAL pour ce type de problèmes, et en particulier sa non-convergence pour l'équation d'onde.

Des instabilités pour les problèmes hyperboliques. Cet aspect de PARAREAL a rapidement été remarqué. En particulier, [Bal, 2005] met en évidence qu'une solution initiale de l'équation d'advection ayant d'importantes composantes haute fréquence peut induire des instabilités dans l'algorithme, celui-ci amplifiant ces dernières composantes et produisant de ce fait une solution non-physique. Ces instabilités sont par ailleurs particulièrement marquées si les méthodes de discrétisation utilisées ne sont pas assez diffusives. Des analyses de stabilité de PARAREAL ont par la suite été menées par [Staff et Rønquist, 2005, Wu, 2016, Ruprecht, 2017]. Alors que [Staff et Rønquist, 2005]

obtiennent des conditions de stabilité assez souples pour des problèmes purement paraboliques, celles associées à un problème purement hyperbolique restent moins bien connues. [Wu, 2016] arrive cependant à en dériver dans le cadre de l'utilisation de PARAREAL avec la méthode d'Euler implicite. Enfin, une contribution importante de [Ruprecht, 2017] est d'observer qu'une erreur de phase entre \mathcal{F} et \mathcal{G} est à l'origine des instabilités de l'algorithme pour les hautes fréquences. Ce dernier préconise, similairement à [Bal, 2005], l'utilisation de schémas de discrétisation spatiale diffusifs afin de faciliter la convergence de PARAREAL, restreinte dans ce cas aux composantes à basses et moyennes fréquences de la solution.

Une multiplication des applications. Malgré ses défauts de convergence dans certaines situations, PARAREAL fut rapidement appliqué à une grande variété de problèmes, du fait de sa simplicité de mise en œuvre. Un résumé des principales applications est proposé en Tab. 2.1.

TABLE 2.1 – Résumé non exhaustif des applications de PARAREAL réalisées jusqu'à ce jour par ordre
chronologique. SP : utilisation combinée de parallélisation spatiale et temporelle. INS : Navier-StokesIncompressible. RANS : Reynolds Averaged Navier-Stokes. DES : Detached Eddy Simulation. TG : utilisation de
la variante Two-Grid de PARAREAL, décrite en Sec. 2.1.3.a. FV : Volumes Finis. FE : Éléments finis.

Référence	Type de problème	Méthodes numériques	TG	SP
[Lions <i>et al.</i> , 2001]	Équation de la chaleur	-		
[Baffico <i>et al.</i> , 2002]	Dynamique moléculaire	-		
[Bal et Maday, 2002]	American put	-		
[Maday et Turinici, 2003]	Contrôle Quantique	-		
[Trindade et Pereira, 2004]	2D INS	Explicite, Upwind		
[Fischer et al., 2005]	2D INS	Implicite, FE/spectrale		
[Trindade et Pereira, 2006]	2D INS	Implicite, Upwind FV		\checkmark
[Farhat <i>et al.</i> , 2006]	Dynamique des Structures	-		
[Mercerat <i>et al.</i> , 2009]	Équation d'onde	Galerkin discontinu	\checkmark	
[Samaddar <i>et al.</i> , 2010]	Turbulence 2D de Plasma	Implicite, spectrale		
[Ruprecht et Krause, 2012]	Advection 1D et Eq. d'ondes	Explicite, <i>Centered/Upwind</i> FV		
[Dai <i>et al.</i> , 2013]	Systèmes Hamiltoniens	-		
[Krause et Ruprecht, 2014]	Eq. Burger visqueux 2D	Explicite, Upwind FV		\checkmark
[Ruprecht, 2014]	Advection-Diffusion 1D	Implicite, Centered FV	\checkmark	
[Croce <i>et al.</i> , 2014]	2D,3D INS	Explicite, Upwind FV		\checkmark
[Randles et Kaxiras, 2014]	Écoulement laminaire de tube 3D	Lattice Boltzmann	\checkmark	\checkmark
[Kreienbuehl et al., 2015b]	Transport poreux	-		
[Kreienbuehl et al., 2015c]	2D INS	Implicite, FE		
[Kreienbuehl et al., 2015a]	Eq. gravitationnelle d'Einstein	-		
[Steiner <i>et al.</i> , 2015]	2D INS	Implicite/explicite, Upwind FV		
[Ruprecht et al., 2016]	Eq. chaleur 2D, coeff. non constants	_		
[Eghbal <i>et al.</i> , 2017]	3D RANS, DES	Implicite, Upwind FV	✓	

On peut tout d'abord remarquer que, malgré sa prédilection pour les problèmes paraboliques, PARAREAL a été très peu appliqué à des équations de diffusion. Ceci peut s'expliquer par l'existence antérieure de méthodes multi-grille parallèles en temps (*cf.* Sec. 2.5), ainsi que l'apparition postérieure d'algorithmes plus efficaces pour ce type de problèmes (*cf.* Sec. 2.4).

Un effort important pour l'application aux équations de Navier-Stokes. On notera que de nombreuses applications se sont focalisées sur les équations de Navier-Stokes pour un fluide incompressible. De manière générale, il a été remarqué une détérioration de la convergence de PARAREAL avec l'augmentation du nombre de Reynolds [Croce *et al.*, 2014, Steiner *et al.*, 2015, Kreienbuehl *et al.*, 2015c], et plus particulièrement l'apparition d'instabilités numériques hautes fréquences dans la solution de PARAREAL, l'empêchant de converger. Pour résoudre ce problème, il a été souvent suggéré d'ajouter de la diffusion numérique. Deux solutions ont ainsi été proposées;

la première consiste à utiliser des schémas de discrétisation de type *upwind*, apportant une diffusion importante pour les hautes fréquences [Mercerat *et al.*, 2009, Ruprecht, 2017]; la deuxième propose d'augmenter la diffusion apportée par le modèle de turbulence pour des problèmes de CFD [Kreienbuehl *et al.*, 2015c, Eghbal *et al.*, 2017]. Ce type de solution, bien qu'améliorant la convergence de l'algorithme pour les composantes à basse fréquence des champs calculés, a aussi pour inconvénient de supprimer ses composantes hautes fréquences, ce qui peut dégrader certaines propriétés physiques de la solution (*e.g.* "laminarisation" d'un écoulement fluide normalement turbulent dans [Kreienbuehl *et al.*, 2015c, Eghbal *et al.*, 2017]).

Des variantes pour mieux traiter les problèmes advectifs. Bien qu'elle trouve son origine au sein de PITA de [Farhat *et al.*, 2006], la variante PARAREAL-KSE n'a été formalisée et étudiée que plus tard par [Gander et Petcu, 2008] pour des problèmes linéaires. À notre connaissance, peu d'applications ont vu le jour à l'heure actuelle, on mentionnera néanmoins le travail de [Ruprecht et Krause, 2012] qui montrent les meilleures propriétés de convergence de PARAREAL-KSE vis-à-vis de la version originale de PARAREAL pour des problèmes purement advectifs (équation d'advection, équation d'onde). Cependant, les auteurs mettent aussi en évidence un surcoût en terme de calcul et de stockage mémoire, argument utilisé par [Chen *et al.*, 2014] pour justifier l'introduction de PARAREAL-RBE (en plus de pouvoir appliquer cette dernière à des problèmes non-linéaires). PARAREAL-RBE a été appliqué à de nombreux problèmes (équation d'advection, équation d'onde, Burgers avec terme visqueux, Kuramoto-Sivashinsky, ...), cependant aucune autre application n'a été recensée hors de l'article original à notre connaissance. On peut aussi mentionner que pour l'équation de Burgers avec terme visqueux, [Chen *et al.*, 2014] remarquent que la convergence de PARAREAL-RBE est moins bonne que celle de PARAREAL pour le cas considéré, ce qu'ils imputent à une limitation induite par la représentation du solveur fin par un modèle réduit.

Des performances parallèles mitigées. En ce qui concerne PARAREAL, quelques travaux de recherche se sont principalement focalisés sur les performances parallèles de l'algorithme. Alors que [Ruprecht, 2015] s'est intéressé au choix du standard de programmation (MPI ou OpenMP) pouvant être utilisé pour implémenter l'algorithme, [Aubanel, 2011] en a examiné les différentes implémentations algorithmiques possibles pour déterminer celle pouvant optimiser la performance parallèle de PARAREAL. Une attention plus particulière sur le *speedup* de PARAREAL sera donnée dans le Chap. 4.

Il est généralement ressorti de ces études que l'efficacité de l'algorithme était inéluctablement bornée par l'inverse du nombre d'itérations, *i.e.* :

$$\mathcal{E}(N) < \frac{1}{K}.$$
(2.19)

Or le nombre d'itérations requises dépend principalement du problème et du critère de convergence choisis, qui varient considérablement au sein de la littérature "applicative" de PARAREAL. On peut néanmoins mentionner pour exemples [Samaddar *et al.*, 2010] ayant obtenu des efficacités parallèles maximum autour de 10% pour une turbulence de plasma 2D, [Trindade et Pereira, 2004] ayant obtenu des efficacités autour de 35% pour des écoulements incompressible 2D (*Taylor Green Vortex*), ou encore [Croce *et al.*, 2014] qui montre des augmentations du *speedup* par rapport à la parallélisation spatiale montant parfois jusqu'à 50% pour un écoulement incompressible 3D, sous réserve de l'utilisation d'un nombre important de *time-slices*.

2.2 PARAEXP, la solution "idéale" (dans le cas linéaire ...)

2.2.1 Description

Cet algorithme, à l'instar des algorithmes itératifs basés sur le *Multiple Shooting*, permet une résolution directe d'une EDO dans un cadre linéaire. Introduit initialement par [Gander et Güttel, 2013], il se base sur l'hypothèse que l'opérateur f dans (2.1) est la somme d'une partie linéaire indépendante du temps et d'un terme source temporel explicite. Le système d'EDO peut alors s'écrire sous la forme suivante :

$$\frac{dU}{dt} = AU + g(t), \ U(0) = U_I, \ t \in [0, T],$$
(2.20)

avec $A \in \mathcal{M}_p(\mathbb{R})$ et $g : \mathbb{R}^+ \to \mathbb{R}^p$ le terme source, éventuellement coûteux à calculer. Une décomposition en *time-slices* similaire à celle de Sec. 2.1.1 est effectuée. Puis, l'algorithme s'exécute en deux étapes. La première consiste à résoudre le problème non homogène (P1) sur chaque sous-intervalle temporel :

(P1)
$$\forall n \in [\![1, N]\!], \ \frac{dV_n}{dt} = AV_n + g(t), \ V_n(t_{n-1}) = 0, \ t \in [t_{n-1}, t_n].$$
 (2.21)

Puis le problème homogène (P2) est résolu sur des sous-intervalles de temps se recouvrant :

(P2)
$$\forall n \in [\![1, N]\!], \quad \frac{dW_n}{dt} = AW_n, \quad W_n(t_{n-1}) = V_{n-1}(t_{n-1}), \quad t \in [t_{n-1}, T].$$
 (2.22)

Enfin, la solution finale est obtenue par sommation :

$$U(t) = V_n(t) + \sum_{k=1}^n W_k(t) \text{ avec } n \text{ tel que } t \in [t_{n-1}, t_n].$$
(2.23)

Une représentation de la solution de chaque problème est donnée en Fig. 2.2.



FIGURE 2.2 – Représentation de PARAEXP, extraite de [Gander et Güttel, 2013].

Ici, la résolution de (P1) peut s'effectuer sur N processeurs en parallèle par le biais d'un intégrateur temporel classique. Cependant, résoudre (P2) de la même manière coûterait aussi cher que résoudre le problème de base (2.20), c'est pourquoi [Gander et Güttel, 2013] proposent d'utiliser une approximation de la matrice exponentielle par projection sur un sous-espace de Krylov. Plus d'informations sur ce genre de techniques pour des problèmes linéaires et non-linéaires sont données par [Schulze *et al.*, 2009]. En particulier, ils préconisent l'utilisation d'une variante de la méthode d'Arnoldi à dénominateur restreint (basé sur les travaux de [Moret et Novati, 2004]), qui profite de l'aspect linéaire homogène de (P2) pour l'intégrer à un coût très faible sur chaque intervalle [t_{n-1}, t_n]. Enfin, un choix optimal de la longueur de chaque sous-intervalle, ainsi qu'une répartition adaptée des calculs sur chaque processeur permet d'utiliser PARAEXP avec une efficacité parallèle nominale.

Cependant, deux aspects de l'algorithme méritent d'être soulignés. En premier lieu, l'utilisation de la méthode d'Arnoldi à dénominateur restreint pour résoudre (P2) nécessite d'inverser la matrice A au début des calculs, et de la stocker. Enfin, comme spécifié en début de section, cet algorithme ne s'applique qu'à des problèmes linéaires à terme source dépendant uniquement du temps.

Adaptation aux problèmes non-linéaires Récemment, [Kooij *et al.*, 2017] ont proposé une autre forme de l'algorithme PARAEXP, qui se distingue de la version originale par deux points majeurs.

Tout d'abord, le terme source est approximé par une décomposition en valeurs singulière (SVD) afin de pouvoir intégrer le problème (P1) en utilisant aussi une méthode de Krylov. L'intégrateur en question est appelé *Exponential block Krylov method (EBK)* [Botchev, 2013] et permet notamment d'intégrer le problème (2.20) quand le terme source g dépend aussi de la solution U.

Enfin, l'algorithme utilise un processus itératif pour résoudre une éventuelle partie non-linéaire du problème. Profitant des caractéristiques de la méthode EBK, le terme source $g_k(t) = g(t, U_k(t))$ est évalué explicitement à partir de l'itéré $U_k(t)$. [Kooij *et al.*, 2017] proposent aussi, dans le but d'accélérer la convergence vers la solution du problème non-linéaire, de lui rajouter un terme contenant la Jacobienne du terme source, de manière similaire à une méthode de Newton-Raphson. Cela permet d'obtenir une formule de récurrence entre l'itéré k et k + 1, ne nécessitant que la résolution d'un problème linéaire (les auteurs soulignent que cette relation de récurrence est similaire à ce qui peut être obtenu avec des méthodes de type *Waveform Relaxation* [Lelarasmee, 1982]). Le problème linéaire correspondant à l'itération k est ensuite résolu avec PARAEXP pour obtenir une solution corrigée $U_{k+1}(t)$ sur l'ensemble de l'intervalle [0, T]. Une forme similaire de l'algorithme pour un problème non-linéaire a aussi été proposée et étudiée par [Gander *et al.*, 2017], sans l'utilisation d'EBK ou de Jacobienne du terme source.

2.2.2 Applications, forces et faiblesses

Un important potentiel pour les problèmes linéaires. Dès son introduction par [Gander et Güttel, 2013], PARAEXP fut appliqué avec succès non seulement à des problèmes paraboliques (équation de la chaleur avec terme source oscillant) mais aussi à des problèmes fortement hyperboliques (équation d'onde, équation d'advection-diffusion 2D avec diffusion faible). L'efficacité parallèle de l'algorithme s'est avérée être globalement bonne pour les deux premiers problèmes (autour de 70 - 80%), mais légèrement moins importante pour le problème d'advection diffusion 2D (autour de 60%).

Du fait de son aspect récent et de sa restriction à des problèmes linéaires, peu d'application de PARAExP sont recensées aujourd'hui. À notre connaissance, celui-ci a été seulement utilisé pour résoudre des problèmes d'électromagnétisme par [Merkel *et al.*, 2016] et [Merkel *et al.*, 2017]. L'efficacité parallèle obtenue avec PARAExP n'a pas été spécifiée dans ces dernière publications. La version non-linéaire de l'algorithme proposée par [Gander *et al.*, 2017] a été appliquée à une forme non-linéaire de l'équation d'onde, et a montré une convergence se dégradant avec l'augmentation du caractère non-linéaire du problème.

Des perspectives pour les problèmes non-linéaires. La variante de PARAEXP proposée par [Kooij *et al.*, 2017], bien que n'ayant été que peu étudiée à l'heure actuelle, mérite d'attirer l'attention sur deux points.

Tout d'abord, celle-ci a été comparée à la version originale de PARAEXP sur le problème d'advection-diffusion. Les résultats ont montré une très bonne efficacité parallèle (autour de 90%), celle-ci ne décroissant pas avec le nombre de processeurs utilisés dans la direction temporelle (ce qui n'était pas le cas pour la version originale de PARAEXP).

Enfin, elle fut appliquée à l'équation de Burgers visqueux, et bien que les résultats en terme d'efficacité parallèle (non spécifiés dans l'article) n'étaient pas aussi satisfaisants que pour le problème linéaire, ceux-ci laissent imaginer un potentiel important de la méthode pour d'autres problèmes linéaires hyperboliques.

2.3 RIDC, l'intégration temporelle alternative

2.3.1 Description

L'algorithme *Revisionist Integral Deferred Correction* (RIDC), fut introduit initialement par [Christlieb *et al.*, 2010b]. Il se base sur une résolution de l'ODE (2.1) utilisant la méthode dite de *Spectral Deferred Correction* (SDC)². Elle fut introduite initialement par [Dutt *et al.*, 2000], et propose une nouvelle manière de résoudre une EDO, se démarquant d'une intégration séquentielle classique.

Base de l'intégration temporelle par SDC. On considère la représentation du système d'EDO (2.1) par sa forme intégrale de Picard :

$$U(t) = U(0) + \int_0^t f(U(\tau), \tau) d\tau.$$
 (2.24)

Ici, on notera par $\tilde{U}(t)$ une approximation de la solution exacte U(t) dans le domaine temporel. On définit l'erreur e et le résidu r:

$$e(t) = U(t) - \tilde{U}(t),$$
 (2.25)

$$r(t) = U(0) + \int_0^t f(\tilde{U}(\tau), \tau) d\tau - \tilde{U}(t).$$
(2.26)

En dérivant puis assemblant (2.25) et (2.26), on peut obtenir une EDO sur l'erreur :

$$e(t)' = r(t)' + f\left(\tilde{U}(t) + e(t), t\right) - f\left(\tilde{U}(t), t\right).$$
(2.27)

Le principe général des SDC consiste donc à résoudre (2.27) au lieu de (2.1). Ceci permet d'estimer un terme d'erreur qui est ajouté à une solution approchée pour la corriger. Ce processus de correction, appelé *sweep* (anglais pour balayage), est répété plusieurs fois. Il permet notamment d'augmenter progressivement l'ordre de la méthode, et par conséquent la précision de la solution obtenue.

Description générale d'un sweep SDC. On effectue tout d'abord une décomposition de l'intervalle [0,T] en sous-intervalles $[t_{n-1},t_n]$ avec $n \in [\![1,N]\!]$. Le pas de temps $\Delta t_n = t_n - t_{n-1}$ correspond ici au pas de temps d'une méthode d'intégration temporelle classique. Puis, chaque sous-intervalle est à nouveau décomposé en M tranches, comme détaillé en Fig. 2.3. Les extrémités de ces tranches sont notées τ_m^n , avec $(m,n) \in [\![1,M]\!] \times [\![1,N]\!]$.

FIGURE 2.3 – Discrétisation de l'intervalle [0, T] pour les SDC.

La manière dont sont choisis les τ_m^n peut varier. Pour RIDC, ceux-ci sont répartis uniformément entre t_{n-1} et t_n , et la dénomination *Integral Deferred Correction* (IDC) est donc préférée. Dans le cadre d'une répartition non uniforme des τ_m^n suivant une règle de quadrature Gaussienne, la dénomination SDC est utilisée. Pour la suite, on se restreindra à cette dernière dénomination pour décrire l'ensemble des méthodes.

Une fois les τ_m^n fixés, on note \tilde{U}_m^k l'approximation de $U(\tau_m^n)$ pour un n fixé, après avoir effectué k sweeps, $k \in [\![0, K]\!]$. (2.27) est discrétisée en utilisant une méthode d'intégration temporelle simple

^{2.} Par la suite, on utilisera la dénomination "les SDC" pour parler de cette méthode.

(*e.g. Forward Euler* (FE), *Backward Euler* (BE), ...) pour obtenir une relation entre les différents itérés. Par exemple, l'utilisation de FE comme intégrateur donne la formule suivante :

$$\tilde{U}_{m+1}^{k+1} = \tilde{U}_m^{k+1} + \Delta \tau_m^n \left[f\left(\tilde{U}_m^{k+1}, \tau_m^n\right) - f\left(\tilde{U}_m^k, \tau_m^n\right) \right] + \int_{\tau_m^n}^{\tau_m^n + 1} f\left(\tilde{U}^k(\tau), \tau\right) d\tau.$$
(2.28)

Le terme intégral de (2.28) est calculé par une quadrature numérique de la forme :

$$\int_{\tau_m^n}^{\tau_{m+1}^n} f\left(\tilde{U}^k(\tau), \tau\right) d\tau = \sum_{l=1}^M w_l^m f(\tilde{U}_l^k, \tau_l^n),$$
(2.29)

avec w_l^m les poids de quadrature. Plus de détails sont donnés en App. A.

Principe de RIDC. Contrairement aux intégrations temporelles séquentielles classiques, qui résolvent une ODE un pas de temps après l'autre, les SDC utilisent plutôt une solution peu précise et peu chère du problème sur l'ensemble de l'intervalle [0, T], qui est corrigée à chaque *sweep*. Ceci est illustré en Fig. 2.4. RIDC profite de cet aspect des SDC, en plus d'ajouter une légère variation à la



FIGURE 2.4 – Comparaison d'une intégration temporelle séquentielle classique et d'une intégration par SDC. Figure générée par l'auteur.

méthode originale. Tout d'abord, les points τ_m^n sont distribués uniformément dans N tranches temporelles, et l'on effectue M sweeps, avec $M \ll N$. Le stencil choisi pour la quadrature numérique dans (2.28) comprend (M + 1) points et sa position reste toujours la même par rapport à l'indice m.

Puis, comme l'on peut voir dans (2.28), si une solution est déjà calculée pour un (k, m) donné, alors elle peut dès lors être calculée pour (k - 1, m - 1), (k - 2, m - 2), etc... RIDC répartit donc chaque *sweep* sur M processeurs, ceux-ci calculant en parallèle dans une configuration de type *pipeline*. Plus de détails sont donnés dans [Christlieb *et al.*, 2010b, Christlieb et Ong, 2011] pour la version utilisant FE et BE, respectivement.

2.3.2 Applications, forces et faiblesses

Les avantages d'une parallélisation induite par la méthode. Bien que RIDC ait été introduit il y a quelques années déjà, et permette de traiter des problèmes linéaires ou nonlinéaires, peu d'applications ont encore été recensées à l'heure actuelle. L'algorithme fut appliqué à des ODE non-linéaires ainsi qu'à un problème 1D de mouvements d'ions et électrons dans [Christlieb *et al.*, 2010b], l'équation d'advection diffusion 1D et de Brusselator dans [Christlieb et Ong, 2011], et enfin l'équation de Burger visqueux dans [Ong *et al.*, 2012]. Pour ce dernier problème, les speed-up obtenus sont quasi-idéaux pour un faible nombre de processeurs (\sim 4), en comparant à la version séquentielle de RIDC. De manière générale, il a été montré que la version parallèle de RIDC permet d'obtenir une solution avec une précision d'ordre élevée, cela au prix d'une intégration peu précise de type FE ou BE. En particulier, sa comparaison avec des méthodes Runge-Kutta implicites d'ordre élevée dans [Ong *et al.*, 2012] a montré un très bon potentiel de la méthode, dans des situations où une parallélisation spatiale n'est pas possible ou très coûteuse.

Plusieurs faiblesses notables. L'une des inconnues majeures de RIDC et des méthodes SDC de manière générale concerne l'application de leur version explicite à des problèmes à caractère fortement hyperbolique. Bien que la première version de RIDC présentée par [Christlieb *et al.*, 2010b] soit explicite, ce sont les versions implicites ou semi-implicites qui ont été appliquées aux problèmes d'advection/diffusion et Burger visqueux dans [Christlieb et Ong, 2011, Ong *et al.*, 2012]. L'utilisation exclusive de méthodes explicites au sein des SDC pour résoudre des problèmes advectifs est étudiée plus en détails en App. A. Une version permettant d'améliorer la stabilité des *sweeps* SDC y est présentée, et sa comparaison avec des méthodes ERK classique y est faite. Elle met en évidence que le prix d'un *sweep* explicite est comparable à celui d'une intégration avec une méthode ERK, et par conséquent rend un calcul parallélisé avec RIDC aussi coûteux que si résolu séquentiellement avec une méthode classique.

De plus, l'aspect des communications "temporelles" entre processeurs n'a pas été abordé à notre connaissance. En particulier, on peut noter que les processus doivent communiquer un champ entier après chaque *sweep*, ce qui pour des problèmes de grande taille induirait des volumes de communications considérables. Ce problème peut cependant être contourné en considérant une autre approche de parallélisation au sein des SDC, comme proposé récemment par [Speck, 2017].

2.4 PFASST, la complexité au service de l'efficacité

2.4.1 Description

L'algorithme PARALLEL FULL APPROXIMATION SCHEME IN SPACE-TIME (PFASST), introduit par [Emmett et Minion, 2012], provient d'une convergence d'idées entre PARAREAL, décrit en Sec. 2.1.2, les SDC présentées en Sec. 2.3.1, et la méthode *Full Approximation Scheme* (FAS) bien connue au sein de la littérature multi-grille (voir [Briggs *et al.*, 2000, Hackbusch, 1984]). Une des idées motivant PFASST est de modifier le solveur fin dans PARAREAL par un *sweep* SDC, pour rendre le coût d'une itération plus faible et par conséquent améliorer la performance parallèle de l'algorithme. Cette approche avait déjà été étudiée par [Minion, 2010], et avait montré un potentiel intéressant. A cela s'ajoute l'introduction d'un niveau de grille grossière, utilisé dans le cadre d'une procédure de correction FAS. Plusieurs niveaux de grille grossière peuvent être introduits, comme détaillé dans [Emmett et Minion, 2014, Speck *et al.*, 2015].

Au vu de l'ensemble des méthodes utilisées au sein de l'algorithme, PFASST est considéré dans la communauté scientifique comme :

"One of the most complex methods in [the parallel-in-time integration] field" [Bolten et al., 2016, Abstract].

C'est pourquoi on ne détaillera pas ici tous les rouages de la méthode, et on redirigera le lecteur vers [Emmett et Minion, 2012]. Il peut être noté cependant que, tout comme RIDC, PFASST se base principalement sur l'utilisation des SDC pour résoudre un système d'EDO. Cependant, à l'instar de RIDC, il se base sur une répartition non-uniforme des points τ_m^n (cf. Fig. 2.3) pour utiliser des méthodes de quadrature Gaussienne.

2.4.2 Applications, forces et faiblesses

A la recherche d'une plus grande efficacité parallèle. Une pratique courante dans la littérature associée à PFASST est d'introduire la méthode en citant la limitation intrinsèque de l'efficacité de PARAREAL bornée par l'inverse de son nombre total d'itérations (*cf.* 2.1.4, §6). PFASST a donc été conçu afin de contourner cette limitation, et ses premières applications (équations de Burgers visqueux 3D, équations de Kuramoto-Slivashinsky 1D) ont en effet montré qu'il gardait de bonnes performances parallèles (efficacité autour de 50%) bien qu'effectuant un nombre raisonnable d'itérations (autour de 4).

Peu d'applications de l'algorithme ont été recensées à ce jour (voir pour exemples [Speck *et al.*, 2012, Speck *et al.*, 2014]). Cependant, la plupart ont mis en évidence le potentiel de PFASST pour compléter une parallélisation spatiale arrivant à saturation. L'exemple le plus



FIGURE 2.5 – Scalabilité de PFASST sur l'équations de la chaleur en 3D. PMG (Parallel Multi-Grid) : solveur multi-grille parallèle en espace. Le speed-up se base sur la résolution du problème sans PFASST sur 32 cœurs de calcul. Carré : utilisation de la parallélisation en espace sur 128 cœurs. Diamant : utilisation de la parallélisation en espace sur 1024 cœurs. Extrait de [Speck et al., 2014].

marquant est donné en Fig. 2.5, pour la résolution de l'équations de la chaleur en 3D par [Speck *et al.*, 2014]. On peut y voir que PFASST permet d'obtenir de bonnes performances parallèles pour des configurations où l'efficacité d'une parallélisation exclusivement en espace chuterait. De plus, la scalabilité forte de l'algorithme reste bien conservée, et ce malgré l'augmentation importante du nombre de cœurs de calcul dans la dimension temporelle (jusqu'à 64, soit 65536 au total).

Un cousin des méthodes multi-grille. Il a été remarqué que PFASST peut sous certaines conditions être assimilé à une méthode multi-grille, et sa convergence pour des problèmes de diffusion et d'advection linéaire a ainsi étudiée de manière similaire par [Bolten *et al.*, 2016]. Cet aspect lui permettait déjà d'être facilement combiné avec des méthodes multi-grille en espace [Minion *et al.*, 2015], et a permis récemment de développer une théorie sur la convergence linéaire de l'algorithme, encore peu étudiée auparavant [Bolten *et al.*, 2017]. On relèvera aussi un effort actuel ³ cherchant à intégrer des techniques d'accélération de convergence propres à PFASST au sein de MGRIT, algorithme de parallélisation en temps basé sur des techniques multi-grille, qui sera décrit en Sec. 2.5.

De grandes incertitudes pour les problèmes hyperboliques. La majeure partie des applications de PFASST a considéré des problèmes paraboliques (*e.g.* équation de la chaleur), avec une utilisation de solveurs implicites ou semi-implicites au sein des SDC. Bien qu'une étude récente du comportement de PFASST ait porté sur l'équation d'advection linéaire 1D [Bolten *et al.*, 2016], celle-ci considérait l'utilisation d'un schéma de discrétisation spatiale *Upwind* d'ordre 3, qui de

^{3.} http://www.birs.ca/events/2016/5-day-workshops/16w5030/videos/watch/201612011409-Friedhoff.html

part l'ajout d'une diffusion numérique notable au problème, lui donne des propriétés paraboliques. Ce talon d'Achille de PFASST est un sujet de recherche actuel, et s'étend même jusqu'à l'utilisation de manière générale de méthodes implicites pour paralléliser en temps des problèmes advectifs. Pour exemple, le titre d'une *slide* de conclusion de *M*. Minion lors du 5^{ème} Parallel-in-Time Integration Workshop⁴ (décembre 2016) :

> "Is there any hope for implicit methods for advective problems? Random thoughts from a depressed grumpy old guy."

De plus, l'utilisation de méthodes explicites au sein des SDC est soumise aux mêmes contraintes que RIDC pour les problèmes advectifs, et les conclusions de l'étude effectuée en App. A s'appliquent aussi dans le cadre de PFASST.

2.5 MGRIT, la solution "universelle"?

2.5.1 Description

Les méthodes multi-grille, précurseures de la parallélisation en temps. Dès les années 80, le concept de parallélisation en temps fut intégré dans la communauté multi-grille, comme le souligne la citation suivante :

"several processors may be employed to solve at several time-steps simultaneously, a technique which we shall refer to as **time-parallelism**" [Horton, 1991, Abstract].

Les premières méthodes dans cette catégorie furent proposées par [Hackbusch, 1984], qui se basèrent sur des techniques multi-grille, et [Lubich et Ostermann, 1987], qui s'inspirèrent de techniques de *Waveform Relaxation* [Lelarasmee, 1982]. La méthode de [Hackbusch, 1984] fut appliquée par [Burmeister et Horton, 1991] aux équations de Navier-Stokes 2D pour un fluide incompressible, à la même époque où [Murata *et al.*, 1991] en proposa une nouvelle variante et démontra ses performances sur ces mêmes équations.

Depuis, de nombreux travaux furent effectués dans cette direction. On peut citer pour exemples [Vandewalle et Horton, 1994, Vandewalle et Van de Velde, 1994, Horton et Vandewalle, 1995] et [Neumüller, 2013]. Cependant, ces méthodes restèrent principalement cantonnées à la communauté multi-grille, et ne pouvaient que difficilement être étendues à d'autres domaines. Ce fut l'une des raisons qui a motivé l'introduction d'une solution plus modulable avec MGRIT.

Description de MGRIT. Introduit par [Friedhoff *et al.*, 2013, Falgout *et al.*, 2014a], l'algorithme *MultiGrid Reduction In Time* s'inscrit dans la continuité des méthodes présentées précédemment, et applique des techniques multi-grille parallèles en espace à la dimension temporelle. Comme re-



FIGURE 2.6 – Description du fonctionnement de MGRIT sur un problème exemple. Source : https://computation.llnl.gov/projects/parallel-time-integration-multigrid

^{4.} http://www.birs.ca/events/2016/5-day-workshops/16w5030/videos/watch/201612011034-Minion.html

présenté en Fig. 2.6, on part d'une solution initiale approchée à la fois en temps et en espace, puis plusieurs niveaux de grille temporelle (au moins 2) sont considérés, sur lesquels peuvent être appliqués différents cycles multi-grille. Enfin, la solution est obtenue par convergence de l'algorithme après un certain nombre d'itérations. À cela peuvent être associés, comme pour PFASST, plusieurs niveaux de grilles spatiales afin de réduire le coût des itérations. Enfin, la méthode peut aussi être utilisée pour résoudre des problèmes non-linéaires. Le fonctionnement de cette dernière présentant une certaine complexité, on ne détaillera pas ici les rouages de la méthode, ceux-ci sont expliqués plus complètement dans [Falgout *et al.*, 2017].

Une bibliothèque implémentant l'algorithme, XBRAID, est actuellement développée au *Lawrence Livermore National Laboratory*⁵. Son application est rendue la plus modulaire possible : l'utilisateur doit juste définir quelques fonctions principales, puis choisit l'ensemble des paramètres propres à MGRIT pour résoudre son problème.

2.5.2 Applications, forces et faiblesses

Une solution simple pour tous types de problèmes. Il a été montré que MGRIT, utilisé avec deux niveaux de grille temporelle et un cycle multi-grille classique, était en fait un équivalent de PARAREAL (voir [Gander et Vandewalle, 2007]). C'est donc sans surprise que les premiers problèmes sur lesquels fut testé l'algorithme furent des équations de diffusion 1D, 2D et 3D [Friedhoff *et al.*, 2013, Falgout *et al.*, 2014a, Falgout *et al.*, 2016]. Cependant, l'algorithme a rapidement été appliqué à des problèmes variés, profitant de la modularité de la bibliothèque XBRAID. On peut citer comme exemple les applications aux équations RANS incompressibles 2D [Falgout *et al.*, 2014b], à l'équation d'advection linéaire avec ou sans diffusion [Dobrev *et al.*, 2016, Howse *et al.*, 2017], aux *Power Systems* [Lecouvez *et al.*, 2016], et à l'équation d'advection diffusion avec termes de bord oscillants [Günther *et al.*, 2017].

À nouveau, une faiblesse pour les problèmes hyperboliques. Comme de nombreux algorithmes similaires, MGRIT converge difficilement pour des problèmes purement hyperboliques comme l'équation d'advection linéaire [Dobrev *et al.*, 2016]. Ces derniers montrent cependant que l'utilisation de cycles multi-grille plus complexes (utilisant une relaxation de type FCF détaillé dans [Falgout *et al.*, 2014a]) permet néanmoins d'obtenir de meilleures performances. Récemment, l'utilisation d'un grossissement spatial adaptatif au sein de MGRIT par [Howse *et al.*, 2017] a montré des résultats encourageants pour l'équation d'advection linéaire 1D. Toutefois, le besoin d'un travail de recherche supplémentaire a été souligné afin d'améliorer les performances de cette stratégie.

Performances parallèles et stockage mémoire. Tout comme les autres algorithmes itératifs, le gain parallèle de MGRIT est fortement dépendant du nombre d'itérations nécessaires pour atteindre la convergence. On se focalise ici sur l'application de MGRIT à un écoulement 2D compressible résolu avec les équations RANS par [Falgout *et al.*, 2014b]. Dans ce papier, les *speed-up* obtenus sont uniquement comparés par rapport à une résolution séquentielle du problème. Les valeurs obtenues (respectivement 0.64, 2.63, et 7.53) en utilisant 256, 1024 et 4096 processeurs, traduisent des efficacités parallèles particulièrement faibles (autour de 0.25% pour chaque valeur). C'est pourquoi la solution MGRIT est plutôt préconisée dans un cadre où l'utilisation de parallélisation spatiale n'est pas envisageable.

Un deuxième point qui mérite d'être souligné correspond à un aspect propre aux méthodes multi-grilles. Les maillages que l'on considère comprennent non seulement l'ensemble des points utilisés pour discrétiser le domaine spatial, mais aussi ceux requis pour discrétiser le domaine temporel. De ce fait, il est nécessaire de stocker autant de champs spatiaux que de nombres de pas de temps utilisés pour intégrer le problème. Bien que l'aspect parallèle de MGRIT permette de répartir

^{5.} http://computation.llnl.gov/projects/parallel-time-integration-multigrid

cette charge mémoire sur différents processeurs, il peut s'agir néanmoins de volumes de donnée considérables à stocker en mémoire pour des problèmes à très haute résolution.

2.6 Vers un premier choix d'algorithme

2.6.1 Critères de sélection

On revient ici sur les critères donnés en introduction du chapitre, sur lesquels s'est basée la sélection des algorithmes présentés précédemment.

On soulignera tout d'abord, au vu de l'ensemble des algorithmes de parallélisation en temps recensés dans ce chapitre, qu'aucun n'a été appliqué à notre connaissance sur les équations de Navier-Stokes dans leur forme compressible. De ce fait, l'application de l'un d'entre eux à ce type de problème constitue déjà une nouveauté. De plus, cette thèse cherche tout d'abord à évaluer le potentiel de la parallélisation en temps pour un solveur CFD, avant d'essayer de développer de nouvelles stratégies. Cela influe donc sur les différents critères de sélection ainsi que leur ordonnancement, qui sont donnés au sein des paragraphes suivants.

Simplicité de mise en œuvre et compatibilité avec des solveurs d'intégration en temps explicites. Ce premier critère se base sur l'objectif global de la thèse : appliquer un algorithme de parallélisation en temps à un solveur CFD déjà existant. De ce fait, toute solution nécessitant un effort d'implémentation important et complexe au sein du solveur serait moins intéressante qu'une solution plus simple d'application, du point de vue de la communauté CFD. En particulier, l'utilisation d'une intégration temporelle implicite est à proscrire dans le cadre de l'utilisation du solveur HYBRID (décrit en Sec. 1.2.2), ce dernier reposant actuellement sur des schémas d'intégration exclusivement explicites.

Impact de l'algorithme dans la littérature scientifique, et nombre d'applications aux équations de Navier-Stokes. Bien qu'elle soit apparue il y a plus de 50 ans, la parallélisation en temps reste encore à l'heure actuelle un concept émergent : 24 publications annuelles liées au sujet en moyenne sur les 6 dernières années ⁶. À titre de comparaison, le nombre de publications sur l'année 2017 reliées aux mots clefs "scheme(s)", "compressible" et "Navier-Stokes" est estimé aux alentours de 5310⁷. D'où l'importance de choisir un algorithme ayant eu un impact notable sur la communauté scientifique, celle-ci servant ainsi de filtre pour distinguer les solutions les plus prometteuses. En particulier, il est préférable que l'algorithme en question ait été déjà appliqué aux équations de Navier-Stokes, ou en tout cas étudié dans ce contexte.

Performances sur les problèmes modèles tests. Bien que les performances d'un algorithme de parallélisation en temps soient un critère qu'il ne faut pas négliger, il peut être dangereux de le considérer comme argument majeur, au vu des arguments suivants.

Tout d'abord, la plupart des algorithmes qui ont été décrits ici sont des processus itératifs, c'est pourquoi leur performance est étroitement liée à leur convergence pour un problème donné. À cela s'ajoute une grande variété de paramètres différents pouvant être utilisés pour chaque algorithme. De ce fait, il peut être délicat d'évaluer la performance d'un algorithme aux résultats qu'il a montrés sur un certain problème dans une configuration particulière.

De plus, la définition de la performance parallèle pour un algorithme peut être sujet à discussion, comme souligné par M. Emmett lors d'un *workshop* organisé par le Centre International de Mathématiques et d'Informatique (CIMI) de Toulouse (2016)⁸ :

^{6.} selon www.parallelintime.org/references/index.html le 10/08/2017.

^{7.} selon https://scholar.google.com le 10/08/2017.

^{8.} http://inpact.inp-toulouse.fr/CIMI_Semester/Wksp_parallel/talk_emmett.pdf, p.14.

Parallel efficiency

Please don't do this!

```
if (number of mpi ranks > 1) then {
   do fancy adaptive multi-grid/level/model time-parallel
} else {
   do stupid non-adaptive backward Euler time-stepping
}
```

Ainsi, une attention particulière doit être portée sur la comparaison des performances de parallélisation en temps avec celles des méthodes considérées actuellement comme les plus performantes pour les problèmes considérés.

L'ensemble des critères décrits ci-dessus sera donc synthétisé et discuté en Sec. 2.6.2 pour l'ensemble des algorithmes décrits dans ce chapitre. Cela permettra de faire un premier tri dans l'ensemble des solutions, pour justifier enfin le choix final fait en Sec. 2.6.3.

2.6.2 Synthèse concernant les différents algorithmes présentés

Les nombres de citations indiqués correspondent à celles référencés par Google Scholar au 23/08/2017.

PARAREAL. [Cité 496 fois depuis son introduction en 2001]

L'impact scientifique, son nombre d'applications aux équations de Navier-Stokes (*cf.* Tab. 2.1) ainsi que son bas niveau de complexité en font un très bon candidat. En particulier, la version de PARAREAL avec grossissement spatial peut être facilement appliquée dans un cadre explicite, bien que nécessitant un travail de recherche supplémentaire. Cependant, on notera que les difficultés de convergence associées à PARAREAL pour les problèmes hyperboliques sont un bémol majeur à prendre en compte. Il en est de même pour son efficacité parallèle se réduisant dramatiquement pour des nombres d'itérations élevés.

PARAREAL avec réutilisation des solutions fines.

Les variantes de PARAREAL décrites en Sec. 2.1.3.b, souffrent en premier lieu de leur faible impact sur la communauté scientifique (PARAREAL-KS cité 62 fois depuis son introduction en 2006, PARAREAL-RB 11 fois depuis 2014). De plus, bien que l'application de PARAREAL-KS à des problèmes hyperboliques par [Ruprecht et Krause, 2012] ait montré ses avantages par rapport à PARAREAL, sa restriction à des problèmes linéaires rend son application inadéquate aux équations de Navier-Stokes. PARAREAL-RB, quant à lui, fait appel à des méthodes complexes dont l'efficacité pour les problèmes visés reste encore à étudier.

PARAEXP. [Cité 32 fois depuis son introduction en 2013]

Bien qu'il soit un excellent candidat pour les problèmes hyperboliques linéaires, l'application de PARAEXP dans un contexte non-linéaire reste encore peu étudiée. Certes, la version introduite par [Kooij *et al.*, 2017] laisse entrevoir un certain potentiel de la méthode. Cependant son introduction très récente ainsi que son niveau de complexité n'en font à l'heure actuelle pas un candidat de prédilection. En particulier, les travaux récents sur l'application des méthodes de Krylov pour résoudre les équations de Navier-Stokes en régime incompressible [Kooij *et al.*, 2016] nécessiteraient au préalable d'être étendus à leur formulation en régime compressible.

RIDC. [Cité 49 fois depuis son introduction en 2010]

Cet algorithme paraît intéressant en premier abord du fait de la parallélisation temporelle naturelle induite par la méthode SDC. Cependant, deux points importants le disqualifient pour les problèmes que l'on veut considérer.

Tout d'abord, la décomposition du calcul en *sweep* successifs n'est intéressante que dans un cadre implicite, où la taille du pas de temps peut être ainsi choisie aussi grande que voulu pour rendre le *sweep* le moins cher possible. Ceci n'est pas réalisable dans un contexte explicite, et rend de ce fait le coût d'un seul *sweep* équivalent à celui d'une méthode d'intégration temporelle explicite classique (voir Ap.A).

Le deuxième point concerne le volume de communication induit par la parallélisation temporelle. En effet, chaque processeur doit communiquer à un autre un champ entier d'une solution corrigée à chaque pas de temps, comme détaillé en Sec. 2.3. De ce fait, les volumes de données qui doivent être communiqués dans un cadre parallèle (de l'ordre de pNT/δ_t) sont considérables

PFASST. [*Cité 98 fois depuis son introduction en 2012*]

Bien que sa grande complexité puisse susciter un sentiment mélangé de peur et d'appréhension, PFASST a été attentivement considéré pendant la première partie de la thèse. C'est d'ailleurs l'existence de cet algorithme qui a motivé l'étude des SDC dans un cadre explicite pour des problèmes advectifs (présentée en Ap. A). En effet, le cadre de son utilisation par [Speck *et al.*, 2014, Speck *et al.*, 2012] coïncide exactement avec celui que l'on veut considérer, où la parallélisation en temps est associée à une parallélisation massive en espace. Cependant, la faible compatibilité des SDC avec des méthodes explicites ainsi que ses difficultés de convergence remarquées pour des problèmes advectifs simples l'écartent des choix possibles.

MGRIT. [Cité 50 fois depuis son introduction en 2014]

De par l'impact actuel important d'XBRAID sur la communauté scientifique, MGRIT peut être considéré comme une solution particulièrement intéressante. Son aspect peu intrusif facilite son application, comme détaillé par [Günther *et al.*, 2017]. Cependant, les applications récentes à des problèmes hyperboliques ont montré qu'il présentait des faiblesses similaires à PARAREAL ou PFASST, et qu'un travail de recherche dans cette direction était encore requis. De plus, le fait de considérer des maillages spatiaux-temporels, gardant en mémoire les champs de solution pour chaque pas de temps, s'avère être un aspect rédhibitoire, si l'on veut considérer des problèmes de grande taille dans les trois dimensions spatiales.

2.6.3 Justification du choix de PARAREAL

Parmi l'ensemble des solutions présentées dans ce chapitre, les algorithmes PARAREAL avec grossissement spatial et MGRIT ont été identifiés comme solutions potentielles actuellement applicables à la parallélisation en temps de simulations numériques effectuées avec HYBRID. Le choix s'est finalement porté sur la première solution au vu des arguments suivants :

(A) : équivalence de PARAREAL et MGRIT pour deux niveaux de grille temporelle (cf. Sec. 2.5.2),

- (B) : possibilité d'appliquer PARAREAL à HYBRID vu comme "boîte noire",
- (C) : prise de connaissance récente de la bibliothèque XBRAID,
- (D) : volonté de considérer à long terme des problèmes de très grande taille.

De par (A), PARAREAL peut être considéré comme un point de départ d'une étude qui pourrait, en fonction des résultats, être étendue à MGRIT par la suite. Le fait de choisir PARAREAL plutôt que MGRIT est aussi justifié par (B). Enfin, comme le souligne (C), il ne faut pas oublier que MGRIT reste une solution nouvelle, qui n'a été mise à disposition de la communauté scientifique que très récemment. Finalement, l'aspect stockage mémoire des champs pour chaque pas de temps pour MGRIT a aussi été un facteur déterminant face à (D).

Au vu des éléments bibliographiques détaillés dans ce chapitre, le choix de PARAREAL comme point de départ pour étudier la parallélisation temporelle pour la simulation d'écoulements turbulents a introduit **deux problématiques majeures**, qui ont été à la base des travaux de recherche effectués dans le cadre de la thèse. Ces deux points sont détaillés dans les deux paragraphes suivants.

Comment se comporte PARAREAL avec grossissement spatial appliqué à des problèmes fortement advectifs avec l'utilisation exclusive de solveurs explicites? Utiliser des solveurs explicites au sein de PARAREAL reste encore à notre connaissance un champ de recherche important à développer. En particulier, l'influence du changement de grille pour le solveur grossier peut être étudiée plus en détails, et comparée avec les facteurs dont l'on reconnaît aujourd'hui l'influence sur la convergence de PARAREAL (*e.g.* ratio entre les forces advectives et diffusives, non-linéarité, ...). Par conséquent on se penchera en premier lieu sur l'application de PARAREAL-TG à une version très simplifiée des équations de Navier-Stokes : le problème mono-dimensionnel d'advection-diffusion. Bien que cela reste un modèle très simplifié, il permet d'avoir un premier aperçu de l'influence de nombreux paramètres sur la convergence de PARAREAL avec grossissement spatial (*e.g.* schéma de discrétisation, taille des *time-slices*, nombre d'itérations, ...). L'étude pour ce problème ainsi qu'à certaines de ses variantes (advection avec faible diffusion, équations de Burgers visqueux) permet donc non seulement de tirer de premières conclusions sur son application, mais aussi d'identifier ses défauts et de proposer d'éventuelles améliorations. Tout ceci fera donc l'objet du Chap. 3.

Comment converge l'algorithme appliqué à un écoulement fluide 3D turbulent, et quelles sont les performances parallèles que l'on peut en tirer? Bien que PARAREAL ait été de nombreuses fois appliqué aux équations de Navier-Stokes, aucune ou très peu d'applicatiosn sur leur forme compressible n'ont été recensées à notre connaissance, en particulier pour la simulation d'écoulements turbulents en utilisant un solveur CFD explicite. Plus particulièrement, le gain éventuel de performance que PARAREAL peut apporter à une parallélisation spatiale dans ce cadre est un sujet méritant une attention particulière. Ainsi en Chap. 4 seront détaillées deux applications de PARAREAL à propos de deux problèmes turbulents tri-dimensionnels canoniques. Une attention particulière sera portée sur la description rigoureuse de ces problèmes tests, afin qu'ils puissent être éventuellement utilisés plus tard comme *benchmark* pour d'autres algorithmes de parallélisation en temps.

2.7 Synthèse

Dans ce chapitre, nous avons cherché à donner un résumé des principaux algorithmes actuels de parallélisation en temps, dans le but de sélectionner une première solution pour l'étude d'une stratégie de parallélisation spatio-temporelle pour la simulation d'écoulements turbulents avec un solveur CFD explicite. Pour chacun des algorithmes présentés (PARAREAL, PARAEXP, RIDC, PFASST, MGRIT), une description synthétique de leur fonctionnement, ainsi qu'un résumé de leur applications, forces et faiblesses ont été prodigués.

Parmi l'ensemble des solutions présentées, certaines ont été écartées en raison de leur faible compatibilité avec des solveurs explicites (PFASST, RIDC) ou encore de la difficulté qui en découlerait de les appliquer à des problèmes non-linéaires (PARAEXP). Finalement, le choix s'est porté sur l'algorithme PARAREAL avec grossissement spatial, ce qui a introduit deux problématiques majeures pour la suite de ce travail, qui seront traitées chacune dans les chapitres suivants.

CHAPITRE 3

Analyse de PARAREAL avec des méthodes temporelles explicites pour des problèmes linéaires fortement advectifs

Table des matières

3.1	Motivations générales		49
3.2	Analy	yse linéaire de Fourier pour Parareal	
	3.2.1	Facteur d'amplification d'un intégrateur temporel linéaire	50
	3.2.2	Facteur d'amplification de PARAREAL	53
	3.2.3	Prise en compte de la discrétisation spatiale	53
	3.2.4	Prise en compte du changement de grille	57
3.3	Étude	des formes explicites de PARAREAL pour les problèmes advectifs	64
	3.3.1	Cas scalaire	65
	3.3.2	PARAREAL avec grille spatiale unique	71
	3.3.3	PARAREAL avec grossissement spatial	78
	3.3.4	Conclusions générales autour de l'analyse de Fourier	84
3.4	Varia	tions autour de l'advection pure linéaire	85
	3.4.1	Motivations et mise en place de l'étude	85
	3.4.2	Equation d'advection-diffusion	86
	3.4.3	Equation de Burgers avec terme visqueux	90
	3.4.4	Conclusions générales sur l'étude de convergence	93
3.5	Synth	èse	94

Dans ce chapitre, on se focalise principalement sur l'équation d'advection-diffusion linéaire monodimensionnelle périodique :

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2}, \quad u(x,0) = u_0(x), \quad x \in [0,1], \quad t \in [0,T],$$
(3.1)

avec $c \in \mathbb{R}^*_+$ le coefficient d'advection, $\nu \in \mathbb{R}_+$ le coefficient de diffusion tel que $\nu \ll c$, $u_0 : [0, 1] \to \mathbb{R}$ et $T \in \mathbb{R}_+$. On considère la discrétisation spatiale uniforme $\Omega_F : [0 = x_0, ..., x_{p-1} = 1 - \delta_x]$, où $\delta_x = 1/p$ est le pas spatial et p le nombre de degrés de liberté. Dès lors, le problème (3.1) peut être reformulé en utilisant la dite "méthode des lignes" [Schiesser, 2012] en le système d'ODE suivant :

$$\frac{dU}{dt} = (A+Z)U, \ U(0) = U_0 \in \mathbb{R}^p, \ t \in [0,T],$$
(3.2)

avec $A, Z \in \mathcal{M}_p(\mathbb{R})$ et $U(t) \in \mathbb{R}^p$, résultant de la discrétisation respective des opérateurs d'advection et de diffusion, incluant les coefficients c et ν . Dans le cadre d'une intégration explicite de ce problème avec un pas de temps δ_t , on utilise le nombre de Courant-Friedrich-Levy (CFL) :

$$CFL = \frac{c\delta_t}{\delta_x},\tag{3.3}$$

pour exprimer des conditions sur la taille maximale du pas de temps pour une intégration numérique "stable" du problème, i.e. sans amplification non physique de la solution numérique.

Ce chapitre analyse donc l'utilisation exclusive de **schémas de discrétisations temporelles explicites** au sein de PARAREAL, appliqué au problème (3.2), avec comme préoccupation principale la détermination des conditions requises pour une intégration temporelle numériquement stable.

En premier lieu, les enjeux et objectifs de cette étude sont détaillés en Sec. 3.1. En Sec. 3.2 est présentée la méthodologie permettant d'effectuer une analyse linéaire de Fourier pour PARAREAL appliqué à (3.2) sans terme diffusif (i.e. $\nu = 0$). Celle-ci est appliquée en Sec. 3.3, afin de dégager des résultats importants concernant la convergence des formes explicites de PARAREAL appliquées à l'équation d'advection pure. Ces résultats sont finalement complétés en Sec. 3.4 avec des applications pour lesquelles le terme diffusif de (3.2) n'est pas négligé, et d'autres où l'on considère la forme non-linéaire de (3.1) (équation de Burgers visqueux).

3.1 Motivations générales

Comme déjà mentionné dans le chapitre précédent, de nombreuses études de convergence de PARAREAL ont déjà été effectuées, en particulier à propos des problèmes hyperboliques comme l'advection pure. De manière générale, le problème majeur identifié correspond à des instabilités de l'algorithme, provoquant une amplification numérique non physique de la solution, et de surcroît une convergence très lente de PARAREAL. Cependant, de nombreuses zones d'ombre demeurent encore à notre connaissance, en particulier sur les points suivants :

Influence des schémas de discrétisation spatiaux-temporels explicites. Bien que de nombreuses études aient porté sur l'utilisation d'intégrateurs temporels implicites au sein de PARAREAL ([Bal, 2005, Wu, 2016, Ruprecht, 2017]), l'étude de l'utilisation d'intégrateurs explicites n'a été que très peu réalisée. À notre connaissance, on ne recense que l'analyse de [Gander, 2008] qui considère l'utilisation exclusive d'un intégrateur de type Euler explicite, et étudie plus particulièrement l'influence des conditions de bords utilisées pour le maillage spatial sur la convergence de PARA-REAL. On notera que des applications de PARAREAL à l'équation d'advection ont été effectuées dans un cadre explicite par [Ruprecht et Krause, 2012], principalement pour illustrer les instabilités de l'algorithme pour le problème d'advection. Ces derniers ont cependant mentionné que :

"The occurrence of the instability does depend on different factors, for example the actual schemes used for the coarse and fine propagators as well as the order of the underlying spatial discretization" [Ruprecht et Krause, 2012, p.76-77, Sec. 3.1, §4]

Une analyse récente effectuée par [Ruprecht, 2017] a ainsi étudié l'influence du type de discrétisation spatiale (centré ou *upwind*) pour des schémas d'ordres peu élevés (1 à 2). Cependant, celle-ci fut effectuée en considérant un intégrateur fin exact, ainsi qu'un schéma d'Euler implicite pour le solveur grossier¹. Par conséquent, l'influence de schémas d'intégration explicites pour les solveurs fin et grossier reste encore assez méconnue. Il en est de même pour leur combinaison avec différentes discrétisations spatiales, en particulier avec des schémas d'ordres élevés. L'intérêt pour ces derniers est motivé par le fait qu'ils sont déjà largement utilisés pour pouvoir résoudre avec précision des problèmes advectifs multi-échelle (*e.g.* écoulements turbulents, *cf.* Sec. 1.2.1), et que leur utilisation risque fortement d'être favorisée dans un cadre de calcul *Exascale* (*cf.* Sec. 1.1.2, §.4).

^{1.} Ce choix d'utiliser un intégrateur implicite est principalement motivé par la simplicité que cela induit pour construire le solveur grossier : ce dernier n'étant soumis à aucune condition de type *CFL*, le pas de temps peut être choisi aussi grand que voulu pour réduire le coût d'intégration de G.

Influence du changement de grille spatial. L'utilisation d'une grille spatiale plus grossière est à notre connaissance la solution la plus intéressante pour construire le solveur grossier de PARAREAL dans un cadre d'intégration explicite. En effet, si l'on considère déjà un maillage grossier obtenu en ne gardant qu'un point du maillage fin sur deux dans chaque direction, et que l'on garde une même condition *CFL* pour les deux solveurs (*i.e.* un pas de temps grossier $\Delta_t = 2\delta_t$), on peut obtenir un ratio de coût de 4 pour un problème 1D, et 16 pour un problème 3D. Il est par conséquent difficile d'envisager de tels ratios par simple réduction de l'ordre des discrétisations spatiale et temporelle.

Cependant, l'influence du changement de grille n'a été que rapidement étudié par [Ruprecht, 2014], qui a principalement pointé que l'utilisation de schémas d'interpolation d'ordre élevé améliorait la convergence de l'algorithme. De plus, bien que certains outils existent déjà pour étudier la stabilité et la convergence de méthodes numériques basées sur plusieurs niveaux de grille, aucun n'a encore été adapté et appliqué dans le contexte particulier de PARAREAL.

Dans les trois sections suivantes on cherche donc à apporter une compréhension plus approfondie des deux points mentionnés précédemment. On considère l'utilisation d'intégrateurs explicites de type Runge-Kutta (ERK) de différents ordres, combinés avec des discrétisations spatiales de type différences finies² centrées et upwind d'ordre préférablement élevé.

3.2 Analyse linéaire de Fourier pour PARAREAL

Cette analyse repose sur l'utilisation du facteur d'amplification d'un intégrateur temporel. Ce concept, globalement connu dans la communauté scientifique, sera rapidement rappelé en Sec. 3.2.1. Son expression pour PARAREAL sera donnée en Sec. 3.2.2. Son utilisation pour analyser l'influence des schémas de discrétisation spatiale sera détaillée en Sec. 3.2.3. Enfin, l'extension pour l'analyse de PARA-REAL avec grossissement spatial, qui constitue la contribution majeure de cette section, sera donnée en Sec. 3.2.4.

3.2.1 Facteur d'amplification d'un intégrateur temporel linéaire

Préliminaires. On considère le problème (3.2) ré-écrit en négligeant sa partie diffusive ($\nu = 0$) :

$$\frac{dU}{dt} = AU, \ U(0) = U_0.$$
 (3.4)

La restriction de l'analyse à ce cas particulier de (3.2) est souvent utilisée pour étudier la stabilité et la convergence des méthodes numériques de manière générale. En effet, une représentation correcte et stable du processus physique associé (advection pure) est un critère de validation incontournable des méthodes de discrétisation, avant de les appliquer à des problèmes plus généraux.

Définition 3.2.1 : Facteur d'amplification — La solution du problème (3.4), obtenue avec un intégrateur temporel linéaire (explicite ou implicite) (e.g. Runge-Kutta ou multi-step) après s itérations temporelles, δ_t désignant le pas de temps, s'écrit :

$$\widetilde{U}(s\delta_t) = g(\delta_t A)^s U_0, \tag{3.5}$$

où $g \in Q(X)$ est une fonction rationnelle à coefficients réels appelée **facteur d'amplification** *du schéma d'intégration.*

^{2.} On peut souligner que pour un problème 1D périodique avec maillage spatial uniforme, les formulations de type différences finies, volumes finis et éléments finis sont équivalentes.

Dans (3.5), $g(\delta_t A) \in \mathcal{M}_p(\mathbb{R})$ est la matrice de l'opérateur (ou solveur) effectuant l'intégration temporelle de (3.4) sur un pas de temps δ_t . On utilisera le terme Jacobienne pour la désigner par la suite ³. Cet opérateur peut être analysé en utilisant la propriété suivante :

Proposition 3.2.1 : *Diagonalisation de l'opérateur spatial* $-Si A \in \mathcal{M}_p(\mathbb{R})$ *est issue d'une même discrétisation spatiale de type différences finies sur l'ensemble des points d'un maillage uniforme périodique, alors elle peut être diagonalisée dans l'espace de Fourier suivant :*

$$A = W_p D W_p^{-1}, \tag{3.6}$$

avec $W_p \in \mathcal{M}_p(\mathbb{C})$ la matrice de transformée de Fourier discrète (Discrete Fourier Transform, DFT), et $D \in \mathcal{M}_p(\mathbb{R})$ diagonale. Chaque coefficient diagonal de D (valeurs propres de A) est associé à un nombre d'onde, ou mode de Fourier de W_p .

Démonstration. L'utilisation d'une discrétisation uniforme sur un maillage périodique implique que A est une matrice circulante, *i.e.* chacune de ses lignes correspond à la ligne voisine du dessus, décalée d'un élément. La proposition découle directement de cette propriété, plus de détails sont donnés dans [Gray *et al.*, 2006, Chap.3].

En notant $\hat{U} = W_p^{-1}U$ la transformée de Fourier de U, la combinaison de la Déf. 3.2.1 et de Prop. 3.2.1 permet d'écrire :

$$\widetilde{\widetilde{U}}(s\delta_t) = g(\delta_t D)^s \widehat{U}_0.$$
(3.7)

Dès lors, l'analyse du système linéaire découplé (3.7) permet d'étudier l'action du schéma spatiotemporel sur chaque mode de Fourier présent dans la solution initiale (amplification, amortissement, décalage de phase). En particulier, cette analyse permet d'obtenir des conditions de stabilité et d'examiner la précision des schémas numériques. Cette méthode, communément appelée analyse de Fourier ou analyse de Von Neumann, est une cousine très proche de l'analyse locale de Fourier (*Local Fourier Analysis*, LFA) utilisée pour les méthodes multi-grille.

Facteur d'amplification dans le cas général. On peut calculer le facteur d'amplification d'une méthode d'intégration linéaire en considérant le problème suivant, appelé problème autonome ou équation de Dahlquist :

$$u'_{\lambda}(t) = \lambda \ u_{\lambda}(t), \quad \lambda \in \mathbb{C}, \quad u_{\lambda}(t) \in \mathbb{C}, u_{\lambda}(0) = 1,$$
(3.8)

dont la solution analytique s'écrit :

$$\forall t \in \mathbb{R}_+, \ u_\lambda(t) = e^{\lambda t}.$$
(3.9)

En se basant sur Déf. 3.2.1 et en prenant A = 1 dans le cas scalaire, g est obtenu en considérant la solution numérique \tilde{u}_{λ} de (3.8) calculée après un pas de temps unitaire $\delta_t = 1$, soit :

$$\forall \lambda \in \mathbb{C}, g(\lambda) = \tilde{u}_{\lambda}(1). \tag{3.10}$$

^{3.} Le terme Jacobienne est utilisé pour désigner ici la dérivée de l'opérateur d'intégration par rapport à un vecteur solution. Dans le cas linéaire ici considéré, la Jacobienne est donc indépendante de la solution initiale, et est représentée exactement par la matrice $g(\delta_t A)$. Bien que les définitions soient similaires, on distingue la Jacobienne ici définie, associée à un solveur, de la Jacobienne du problème (3.4) (*i.e.* A), associée à l'opérateur du membre de droite de (3.4).

Observation 3.2.1 : *Symétrie du facteur d'amplification* — Du fait que g soit une fraction rationnelle à coefficients réels, pour toute discrétisation temporelle linéaire d'un problème réel, le facteur d'amplification a pour propriété particulière de présenter une symétrie par rapport à l'axe réel du plan complexe, i.e. :

$$\forall \hat{z} \in \mathbb{C}, \ g(\hat{z}) = g(\mathcal{R}(\hat{z}) - \mathcal{I}(\hat{z})). \tag{3.11}$$

Définition 3.2.2 : Condition de stabilité pour un schéma temporel – L'utilisation d'un schéma d'intégration avec un pas de temps δ_t est considéré stable pour un λ donné si :

$$|g(\lambda\delta_t)| \le 1. \tag{3.12}$$

Si cette condition n'est pas respectée, l'intégration est considérée comme numériquement instable, i.e. induisant une amplification non physique de la solution à chaque pas de temps.

Définition 3.2.3 : Contour de stabilité d'un schéma temporel — Pour un schéma d'intégration linéaire, on définit le contour de stabilité :

$$\mathcal{S}_g = \{ \hat{z} \in \mathbb{C} \ \text{tel que } |g(\hat{z})| = 1 \}.$$

$$(3.13)$$

Pour des méthodes explicites, l'intégration numérique est par définition stable lorsque $\lambda \delta_t$ est contenu à l'intérieur de S_q (ou à l'extérieur pour les méthodes implicites).

Définition 3.2.4 : *Précision d'un schéma temporel* — On définit la précision du schéma numérique linéaire pour un pas de temps δ_t donné par :

$$E_{\lambda} = |g(\lambda \delta_t) - e^{\lambda \delta_t}|, \qquad (3.14)$$

et de même manière que pour la stabilité on définit le contour de précision de g associé à une tolérance χ comme :

$$\mathcal{E}_g(\chi) = \{ \hat{z} \in \mathbb{C} \text{ tel que } |g(\hat{z}) - e^{\hat{z}}| = \chi \}.$$
(3.15)

Facteur d'amplification des méthodes Runge-Kutta explicites. Les méthodes Runge-Kutta sont largement documentées dans la littérature (voir [Butcher, 2016, Chap. 3]). En particulier, on les représente par leur tableau de Butcher :

avec $B \in \mathcal{M}_{N_s}(\mathbb{R})$, $b, c \in \mathbb{R}^{N_s}$, et N_s le nombre d'étapes de la méthode Runge-Kutta. Leur facteur d'amplification s'écrit ainsi :

$$g(\hat{z}) = 1 + \hat{z}b^T (\mathbb{I}_{N_s} - \hat{z}B)^{-1}e, \qquad (3.16)$$

avec $e = (1, 1, ..., 1)^{\mathsf{T}} \in \mathbb{R}^{N_s}$. Pour des méthodes Runge-Kutta explicites (ERK) d'ordre o_{RK} , grâce à la forme triangulaire de B, g peut s'écrire sous une forme polynômiale simplifiée, donnée dans [Wang et Spiteri, 2007, Corollary 3.3] :

$$g(\hat{z}) = 1 + \sum_{l=1}^{o_{RK}} \frac{\hat{z}}{l!} + \sum_{l=o_{RK}+1}^{N_s} \hat{z}^l b^T B^{l-1} e.$$
(3.17)

3.2.2 Facteur d'amplification de PARAREAL

Une formulation du facteur d'amplification de PARAREAL a déjà été obtenue par [Staff et Rønquist, 2005, Sec. 3.2]. Cependant, cette expression considère que le propagateur grossier n'effectue qu'une seule itération par *time-slice* de PARAREAL. C'est pourquoi on présente ici l'expression dans un cas plus général :

Proposition 3.2.2 — On considère l'application de PARAREAL au problème autonome (3.8) utilisant les paramètres suivants pour les propagateurs fin et grossier respectivement :

1. Pas de temps : δ_t et Δ_t ,

2. Facteur d'amplification des schémas d'intégration temporelle : $g_{\mathcal{F}}$ et $g_{\mathcal{G}}$,

3. Nombre de pas de temps par time-slice : $s_{\mathcal{F}}$ et $s_{\mathcal{G}}$.

On pose les notations suivantes :

$$R = g_{\mathcal{G}}(\lambda \Delta_t)^{s_{\mathcal{G}}},$$

$$r = g_{\mathcal{F}}(\lambda \delta_t)^{s_{\mathcal{F}}}.$$

Dans ces conditions, le facteur d'amplification de PARAREAL à la fin de la time-slice n à l'itération k s'écrit :

$$g_{\mathcal{P}}^{n,k}(r,R) = \sum_{i=0}^{k} \binom{n}{i} (r-R)^{i} R^{n-i}.$$
(3.18)

Démonstration. Ceci peut être prouvé en utilisant une relation de récurrence de Pascal utilisée au sein d'une algèbre commutative, on référera à [Staff et Rønquist, 2005, Sec. 3.2] pour plus de détails.

Proposition 3.2.3 – Pour le problème autonome, la condition de stabilité de PARAREAL s'écrit : Parareal stable pour la $n^{ième}$ time-slice à l'itération $k \Leftrightarrow |g_{\mathcal{D}}^{n,k}(r,R)| \leq 1.$ (3.19)

Démonstration. La propriété s'obtient directement par application de Déf. 3.2.2 à Prop. 3.2.2.

Observation 3.2.2 – Pour un problème linéaire, PARAREAL revient tout simplement à calculer

$$r^{n} = ((r-R) + R)^{n} = \sum_{i=0}^{n} \binom{n}{i} (r-R)^{i} R^{n-i},$$
(3.20)

pas à pas, en ajoutant à chaque itération un terme de la somme binômiale de Newton dans (3.20). La convergence est rapide lorsque |R| >> |r-R|, ^a et l'algorithme nécessite alors k = n itérations pour compléter la somme de Newton, i.e. pour converger exactement à la fin de la $n^{ième}$ time-slice.

a. Ce comportement s'explique par une étude plus en détails de la somme binômiale de Newton pour des nombres complexes dans le cas général, qui n'est pas détaillée ici.

3.2.3 Prise en compte de la discrétisation spatiale

On présente ici comment combiner le facteur d'amplification de PARAREAL avec l'analyse de Von Neumann pour étudier l'influence des schémas de discrétisation spatio-temporelle sur la stabilité et la convergence de l'algorithme. Introduite initialement par [Crank et Nicolson, 1947], la méthode de Von Neumann est un outil très connu dans la littérature. Par soucis de clarté de l'exposé et pour préparer les développements de la Sec. 3.2.4, on résume tout d'abord ses grands principes en Sec. 3.2.3.a avant de la détailler pour PARAREAL en Sec. 3.2.3.b.

3.2.3.a Analyse de Von Neumann d'une discrétisation spatio-temporelle

Symbole de Fourier d'une discrétisation spatiale. On considère maintenant (3.4), et la forme diagonale de A donnée en Prop. 3.2.1. On note ω_{κ} la pulsation associée au nombre d'onde κ , définie comme suit :

$$\omega_{\kappa} = \kappa \frac{2\pi}{p\delta_x} \operatorname{avec} \kappa \in \llbracket 0, p \rrbracket.$$
(3.21)

On normalise les notations en notant la pulsation réduite $\theta_{\kappa} = \omega_{\kappa}L/p = \omega_{\kappa}\delta_x \in [0, 2\pi[$ avec L (= 1 dans le contexte de l'étude) la taille du domaine spatial. θ_{κ} prend ainsi des valeurs discrètes dans $[0, 2\pi[$. Cependant on considérera par la suite le cas continu, qui survient quand $p \to \infty$.

Un vecteur propre de V_{κ} de A associé au nombre d'onde κ s'écrit :

$$V_{\kappa} = [1, e^{\iota\theta_{\kappa}}, e^{2\iota\theta_{\kappa}}, ..., e^{(p-1)\iota\theta_{\kappa}}]^{\mathsf{T}},$$
(3.22)

avec $\iota^2 = -1$, et la valeur propre associée dépend du schéma de discrétisation utilisé pour construire A. De manière générale, les valeurs propres s'écrivent :

$$\lambda_{\kappa} = -c \frac{z_{adv}(\theta_{\kappa})}{\delta_x},\tag{3.23}$$

avec c le coefficient d'advection défini dans (3.1). $z_{adv}(\theta_{\kappa})$ est appelé le symbole de Fourier du schéma différences finies, et tend à approximer le symbole de Fourier exact qui serait obtenu par une discrétisation spectrale : $z_{adv,exact}(\theta_{\kappa}) = \iota \theta_{\kappa}$. Plus de détails sont donnés en Ap. B.2.

Condition *CFL*. De manière générale, on notera qu'en plus d'être 2π -périodique, le symbole de Fourier présente la propriété suivante :

$$z_{adv}(-\theta_{\kappa}) = \mathcal{R}(z_{adv}(\theta_{\kappa})) - \mathcal{I}(z_{adv}(\theta_{\kappa})).$$
(3.24)

Par conséquent, en profitant de la même propriété de symétrie présentée par le facteur d'amplification (*cf.* Obs. 3.2.1), on peut restreindre l'analyse à $\theta_{\kappa} \in [0, \pi]$. On parlera par la suite de basses fréquences pour $\theta_{\kappa} \to 0$ et hautes fréquences pour $\theta_{\kappa} \to \pi$.

Proposition 3.2.4 : Condition de stabilité CFL — Une discrétisation spatio-temporelle donnée, utilisée avec un nombre CFL fixé, est considérée stable si :

$$\forall \theta_{\kappa} \in [0, \pi], \ |g(-CFL \ z_{adv}(\theta_{\kappa}))| \le 1.$$
(3.25)

Démonstration. L'utilisation de Prop. 3.2.1 permet d'analyser la stabilité en considérant séparément chaque valeur propre de la matrice diagonale *D*, *i.e.* :

$$\forall \lambda_{\kappa}, \ |g(\delta_t \lambda_{\kappa})| \le 1. \tag{3.26}$$

L'utilisation de (3.3) et (3.23) permet d'écrire $\delta_t \lambda_{\kappa} = -CFL z_{adv}(\theta_{\kappa})$, et de retrouver ainsi la proposition pour θ_{κ} associé à la valeur propre λ_{κ} .

Caractérisation de l'erreur induite par la discrétisation spatio-temporelle. Il est possible d'écrire le facteur d'amplification d'une méthode intégrant "exactement" le problème (3.4) avec un schéma de discrétisation spatiale spectral utilisant un pas de temps avec un *CFL* donné sous la forme :

$$g_{exact}(\theta_{\kappa}) = e^{-\iota\theta_{\kappa}CFL}.$$
(3.27)

L'erreur de discrétisation spatio-temporelle d'un schéma quelconque, représentée par son symbole de Fourier z_{adv} et le facteur d'amplification de la méthode temporelle g, peut donc s'écrire :

$$E(\theta_{\kappa}) = |g(-CFL \ z_{adv}(\theta_{\kappa})) - e^{-\iota\theta_{\kappa}CFL}|.$$
(3.28)

De manière générale, on distingue cette erreur, produite par la combinaison des discrétisations spatiale et temporelle, de l'erreur induite par le schéma de discrétisation spatial seul, qui s'écrit :

$$E_{adv}(\theta_{\kappa}) = |z_{adv}(\theta_{\kappa}) - \iota \theta_{\kappa}|.$$
(3.29)

Ces erreurs peuvent être caractérisées de deux manières principales. L'ensemble des termes dont on étudie la différence sont des nombres complexes, qui peuvent par conséquent être décomposés de la manière suivante (*e.g.* pour $g_{exact}(\theta_{\kappa})$):

$$g_{exact}(\theta_{\kappa}) = \varrho e^{\iota \varphi}, \tag{3.30}$$

avec $\rho \in \mathbb{R}_+$ et $\varphi \in [0, 2\pi[$ respectivement l'amplitude et la phase du nombre complexe. On distingue donc deux types d'erreur :

- l'erreur d'amplification, portant sur *ρ*. Elle s'apparente à une erreur sur l'amplitude des ondes (ou composantes fréquentielles) au sein de la solution. Dans le cadre d'une intégration temporelle stable, celle-ci est aussi qualifiée d'erreur de diffusion.
- l'erreur de phase, portant sur φ . Elle s'apparente à une erreur sur la vitesse de propagation des ondes composant la solution.

Propriétés générales des discrétisations spatiales. Les propriétés des schémas de discrétisation sont déjà bien connues dans la littérature. Ici on en rappellera les plus importantes, dont la connaissance nous sera utile pour la suite [Olver, 2014, Chap. 5] :

- les discrétisations upwind ont des valeurs propres distribuées sur une bulle à gauche de l'axe imaginaire, tangente à ce dernier en l'origine. Cette bulle correspond à un cercle pour le schéma upwind d'ordre 1 (U1), et s'aplatit sur l'axe imaginaire avec l'augmentation de l'ordre. Ces schémas induisent principalement une erreur de diffusion sur les hautes fréquences.
- les discrétisation centrées ont des valeurs propres purement imaginaires situées sur [-ιπ, ιπ]. Ces schémas induisent uniquement une erreur de phase pour les hautes fréquences. Leur combinaison n'est stable qu'avec des méthodes (ERK ou autre) incluant une partie de l'axe imaginaire dans leur contour de stabilité. En particulier, leurs combinaisons avec *Forward Euler* ou n'importe quelle méthode ERK d'ordre 2 en 2 étapes sont inconditionnellement instables.

Contours d'amplification. Pour le problème d'advection, la principale source d'erreur correspond à une erreur de diffusion apportée par les schémas numériques. Bien qu'elle soit souvent pointée du doigt, la discrétisation spatiale n'est pas la seule responsable : en effet, l'erreur dépend aussi du schéma temporel, ainsi que de la position des valeurs propres λ_{κ} à l'intérieur (ou l'extérieur) de son contour de stabilité. De ce fait, l'influence du *CFL* est aussi à prendre en compte. Dans le but de représenter cette dépendance, on définit le contour d'amplification d'une discrétisation spatio-temporelle, normalisé par le nombre *CFL*, comme étant :

$$\mathcal{S}_{g,z_{adv}}(v) = \{(\theta_{\kappa}, CFL) \text{ tel que } |g(-CFL \ z_{adv}(\theta_k))|^{1/CFL} = v\}.$$
(3.31)

Des iso-surfaces de $S_{g,z_{adv}}(v)$ sont représentées en Fig. 3.1, ces dernières étant dénommées contours d'amplification par abus de langage. Cette représentation, introduite initialement par [Lunet *et al.*, 2017b]⁴, prodigue un certain nombre d'informations sur chaque combinaison spatio-temporelle. En particulier, elle indique la limite de stabilité *CFL*, ainsi que la fréquence qui sera amplifiée une fois la limite de stabilité franchie. De plus, l'amplitude de $g(-CFL z_{adv}(\theta_k))$ étant

^{4.} Le terme "contour de stabilité" est utilisé dans [Lunet *et al.*, 2017b] pour dénommer ce que l'on appelle ici "contour d'amplification". Cette différence résulte d'une prise de conscience récente que la dernière dénomination était plus adaptée. La normalisation par le nombre *CFL*, qui permet de comparer l'amplification à un temps final équivalent, constitue une nouveauté introduite dans le cadre de la thèse.



FIGURE 3.1 – Contours d'amplification (3.31) pour deux discrétisations spatio-temporelles équivalentes en terme de coût de calcul. U1 : discrétisation spatiale upwind d'ordre 1, C2 : discrétisation spatiale centrée d'ordre 2, HEUN2 : méthode ERK d'ordre 2 en deux étapes, RK21 : méthode ERK d'ordre 1 en deux étapes. Une description plus complète de ces méthodes est donnée en Ap. B. Chaque niveau de couleur représente une plage de valeur pour $S_{g,z_{adv}}$ comprise dans un intervalle de taille 0.5, avec $S_{g,z_{adv}}(1)$ représenté en tirets rouges. La ligne en tirets grise représente la limite maximum de CFL stable.

théoriquement unitaire pour le problème d'advection, on peut y observer l'erreur de diffusion pour chaque nombre d'onde de la solution à un *CFL* stable donné. Enfin, on peut aussi observer l'erreur de diffusion apportée uniquement par le schéma de discrétisation spatial pour *CFL* \rightarrow 0 (*i.e.* avec une erreur due à l'intégration temporelle négligeable).

3.2.3.b Facteur d'amplification de PARAREAL pour le problème d'advection

Proposition 3.2.5 — On considère l'application de PARAREAL au problème d'advection monodimensionnel (3.4), utilisant le même maillage spatial périodique pour chaque propagateur temporel \mathcal{F} et \mathcal{G} . On note les paramètres suivants pour chaque propagateur :

- 1. Nombres $CFL : CFL_{\mathcal{F}}$ et $CFL_{\mathcal{G}}$ (pouvant s'écrire en fonction de $CFL_{\mathcal{F}}$),
- 2. Symboles de Fourier : $z_{adv,F}$ et $z_{adv,G}$,
- 3. Facteurs d'amplification : $g_{\mathcal{F}}$ et $g_{\mathcal{G}}$,
- 4. Nombres de pas de temps par time-slice : $s_{\mathcal{F}}$ et $s_{\mathcal{G}}$,
- 5. $\theta_{\kappa} \in [0, \pi]$ les nombres d'onde représentés par la grille spatiale.

On pose ensuite les notations suivantes :

$$R(\theta_{\kappa}) = g_{\mathcal{G}}(-CFL_{\mathcal{G}}z_{adv,\mathcal{G}}(\theta_{\kappa}))^{s_{\mathcal{G}}},$$

$$r(\theta_{\kappa}) = g_{\mathcal{F}}(-CFL_{\mathcal{F}}z_{adv,\mathcal{F}}(\theta_{\kappa}))^{s_{\mathcal{F}}}.$$

Dans ces conditions, le facteur d'amplification de PARAREAL à la fin d'une time-slice n à l'itération k s'écrit ainsi :

$$g_{\mathcal{P}}^{n,k}(CFL_{\mathcal{F}},\theta_{\kappa}) = g_{\mathcal{P}}^{n,k}(CFL_{\mathcal{F}},R(\theta_{\kappa}),r(\theta_{\kappa})) = \sum_{i=0}^{k} \binom{n}{i} (r(\theta_{\kappa})-R(\theta_{\kappa}))^{i}R(\theta_{\kappa})^{n-i}.$$
 (3.32)

Démonstration. La proposition peut être obtenue simplement en appliquant directement Prop. 3.2.2

Définition 3.2.5 – À partir de Prop. 3.2.5, on définit les contours d'amplification normalisés de PARAREAL :

$$\mathcal{S}_{\mathcal{P}}(v) = \{ (\theta_{\kappa}, CFL_{\mathcal{F}}) / |g_{\mathcal{P}}^{n,k}(CFL_{\mathcal{F}}, \theta_{\kappa})^{1/(nCFL_{\mathcal{F}}s_{\mathcal{F}})}| = v \},$$
(3.33)

et l'erreur de parallélisation pour un CFL_{\mathcal{F}} donné :

$$E_n^k(\theta_\kappa) = |g_{\mathcal{P}}^{n,k}(CFL_{\mathcal{F}}, \theta_\kappa) - r(\theta_\kappa)^{ns_{\mathcal{F}}}|.$$
(3.34)

3.2.4 Prise en compte du changement de grille

On étend ici l'analyse de von Neumann effectuée en Sec. 3.2.3.a à la variante de PARAREAL avec grossissement spatial, dans le but d'obtenir comme pour Prop.3.2.5 l'expression de son facteur d'amplification pour le problème d'advection linéaire. De manière générale, on considère un nombre p pair de points pour la grille spatiale du propagateur fin, et une grille grossière construite en ne gardant qu'un point sur deux de la grille fine Ω_F , soit $\Omega_G : [x_0, x_2, ..., x_{p-2}]$ comprenant $\hat{p} = p/2$ points.

On retrouve ici des concepts très proches de ceux utilisés pour l'Analyse Local de Fourier (LFA) des méthodes multi-grille. Cependant, l'approche présentée ici est différente, car basée sur la méthodologie von Neumann. C'est pourquoi, pour faciliter la compréhension, on développe l'ensemble des calculs présentés dans les sections suivantes.

3.2.4.a Notations

Propagateur fin. On reprend (3.4) en utilisant les notation suivantes :

$$\frac{dU}{dt} = A_{\mathcal{F}}U,\tag{3.35}$$

avec $A_{\mathcal{F}} \in \mathcal{M}_{2\hat{p}}(\mathbb{R})$ et \hat{p} le nombre de degrés de liberté de la grille grossière. La pulsation réduite du maillage fin s'écrit : $\theta_{\kappa} = \kappa \pi / \hat{p} \in [0, 2\pi[$. Une fois le schéma d'intégration choisi pour le propagateur fin, la Jacobienne de \mathcal{F} peut s'écrire sous la forme :

$$M_{\mathcal{F}} = W_{2\widehat{p}} \begin{bmatrix} D_{\mathcal{F}} & 0\\ 0 & D_{\mathcal{F}}^* \end{bmatrix} W_{2\widehat{p}}^{-1},$$
(3.36)

avec $D_{\mathcal{F}} = diag[r(\theta_{\kappa})]$ pour $\theta_{\kappa} \in [0, \pi[$, et $r(\theta_{\kappa})$ défini de la même manière qu'en Prop. 3.2.5. $D_{\mathcal{F}}^*$ est défini de manière similaire pour $\theta_{\kappa} \in [\pi, 2\pi[$.

Propagateur grossier. Pour $\hat{\mathcal{G}}$, on introduit les représentations matricielles des opérateurs de transfert R et I pour la restriction et l'interpolation resp., matrices rectangulaires de taille respective $\hat{p} \times 2\hat{p}$ et $2\hat{p} \times \hat{p}$. On écrit la Jacobienne de $\hat{\mathcal{G}}$ sous la forme suivante :

$$M_{\widehat{\mathcal{G}}} = W_{2\widehat{p}}\widehat{I}D_{\widehat{\mathcal{G}}}\widehat{R}W_{2\widehat{p}}^{-1}$$
(3.37)

avec $D_{\mathcal{G}} = diag[R(\widehat{\theta_{\kappa}})], \widehat{\theta_{\kappa}} = 2\theta_{\kappa}$ et $\theta_{\kappa} \in [0, \pi[. \hat{I} \text{ et } \hat{R} \text{ sont appelées transformées de Fourier des opérateurs de transfert, et définies ainsi :$

$$\widehat{I} = W_{2\widehat{p}}^{-1} I W_{\widehat{p}},\tag{3.38}$$

$$\widehat{R} = W_{\widehat{p}}^{-1} R W_{2\widehat{p}}.$$
(3.39)

3.2.4.b Transformées de Fourier des opérateurs de transfert

Lemme 3.2.1 – Les transformées de Fourier pour l'injection et l'interpolation linéaire peuvent s'écrire pour l'ensemble des pulsations réduites $\theta_{\kappa} \in [0, \pi]$ de la grille fine :

$$\hat{R} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \\ & \ddots & \\ & 1/\sqrt{2} & & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} D_R & D_R \end{bmatrix}, \quad (3.40)$$

$$\widehat{I} = \begin{bmatrix} \sqrt{2} & & & \\ & \frac{1}{\sqrt{2}} \left(1 + \cos \left(\theta_{\kappa} \right) \right) & & \\ & & \ddots \\ 0 & & & \ddots \\ & \frac{1}{\sqrt{2}} \left(1 + \cos \left(\pi + \theta_{\kappa} \right) \right) & & \\ & & \ddots \end{bmatrix} = \begin{bmatrix} D_{I} \\ D_{I}^{*} \end{bmatrix},$$
(3.41)

avec $D_R, D_I, D_I^* \in \mathcal{M}_{\widehat{p}}(\mathbb{R}).$

Démonstration. On remarque tout d'abord que les matrices des opérateurs de transfert sont obtenues par concaténation (horizontale ou verticale) de deux matrices carrées de taille \hat{p} . Pour prendre en compte cet aspect, ainsi que les conditions aux bords périodiques, on pose la notation suivante :

$$i \stackrel{p}{=} j \Leftrightarrow (i-j) \equiv 0 \pmod{\widehat{p}},$$
(3.42)

où $a \equiv b \pmod{c}$ représente la congruence de a et b modulo c, pour des entiers relatifs. On définit les opérateurs de transfert avec les indices i (ligne) et j (colonne), partant chacun de 0:

$$\begin{aligned} \forall (i,j) \in [0,\hat{p}-1] \times [0,2\hat{p}-1], & R_{i,j} = 1 \text{ si } 2i \stackrel{p}{=} j, \\ & 0 \text{ sinon}, \end{aligned} \tag{3.43} \\ \forall (i,j) \in [0,2\hat{p}-1] \times [0,\hat{p}-1], & I_{i,j} = 1 \text{ si } i \stackrel{\widehat{p}}{=} 2j, \\ & 1/2 \text{ si } i - 1 \stackrel{\widehat{p}}{=} 2j \text{ ou } i - 1 \stackrel{\widehat{p}}{=} 2(j-1) \end{aligned}$$

0 sinon.

$$(W_{\widehat{p}})_{i,j} = \frac{1}{\sqrt{\widehat{p}}} \omega_{\widehat{p}}^{ij}, \text{ avec } \omega_{\widehat{p}} = e^{\frac{-2\iota\pi}{\widehat{p}}},$$
(3.45)

et pour ceux de son inverse :

$$(W_{\widehat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{\widehat{p}}} \omega_{\widehat{p}}^{-ij}.$$
 (3.46)

(3.44)

Les propriétés suivantes de $\omega_{\widehat{p}}$ seront utilisées par la suite :

$$\omega_{2\widehat{p}} = \sqrt{\omega_{\widehat{p}}},\tag{3.47}$$

$$\sum_{k=0}^{\widehat{p}-1} \omega_{\widehat{p}}^{lk} = \widehat{p} \text{ si } l \stackrel{\widehat{p}}{=} 0 \text{ sinon } 0, \tag{3.48}$$

$$\omega_{\widehat{p}}^{i} + \omega_{\widehat{p}}^{-i} = 2\cos\left(\frac{2i\pi}{\widehat{p}}\right). \tag{3.49}$$

^{5.} Ou plus directement, https://en.wikipedia.org/wiki/DFT_matrix.

Opérateur de restriction. Pour déduire $RW_{2\hat{p}}^{-1}$, on utilise (3.43) et (3.47) pour obtenir :

$$(RW_{2\hat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{2\hat{p}}} \sum_{k=0}^{2\hat{p}-1} R_{i,k} \omega_{2\hat{p}}^{-kj} = \frac{1}{\sqrt{2\hat{p}}} \omega_{2\hat{p}}^{-2ij} = \frac{1}{\sqrt{2\hat{p}}} \omega_{\hat{p}}^{-ij},$$
(3.50)

puis, en utilisant (3.48) :

$$(W_{\widehat{p}}RW_{2\widehat{p}}^{-1})_{i,j} = \frac{1}{\widehat{p}\sqrt{2}}\sum_{k=0}^{\widehat{p}-1}\omega_{\widehat{p}}^{ik}\omega_{\widehat{p}}^{-kj} = \frac{1}{\widehat{p}\sqrt{2}}\sum_{k=0}^{\widehat{p}-1}\omega_{\widehat{p}}^{(i-j)k} = 1/\sqrt{2} \text{ si } i \stackrel{\widehat{p}}{=} j \text{ sinon } 0.$$
(3.51)

Ceci prouve ainsi (3.40).

Opérateur d'interpolation. $IW_{\widehat{p}}^{-1}$ peut s'écrire ainsi :

$$(IW_{\hat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{\hat{p}}} \sum_{k=0}^{\hat{p}-1} I_{i,k} \omega_{\hat{p}}^{-kj}.$$
(3.52)

En utilisant (3.44) et (3.47), on distingue deux cas :

• *i* est pair :

$$(IW_{\hat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{\hat{p}}}\omega_{\hat{p}}^{-ij/2} = \frac{1}{\sqrt{\hat{p}}}\omega_{2\hat{p}}^{-ij}.$$
(3.53)

• *i* est impair :

$$(IW_{\widehat{p}}^{-1})_{i,j} = \frac{1}{2\sqrt{\widehat{p}}} \left(\omega_{\widehat{p}}^{-(i-1)j/2} + \omega_{\widehat{p}}^{-(i-1)j/2-j} \right) = \frac{1}{2\sqrt{\widehat{p}}} \left(\omega_{2\widehat{p}}^{-(i-1)j} + \omega_{2\widehat{p}}^{-(i-1)j} \omega_{\widehat{p}}^{-j} \right).$$
(3.54)

Puis, on applique le second produit matrice vecteur pour obtenir :

$$(W_{2\widehat{p}}IW_{\widehat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{2\widehat{p}}} \sum_{k=0}^{2\widehat{p}-1} \omega_{2\widehat{p}}^{ik} (IW_{\widehat{p}}^{-1})_{kj},$$

$$= \frac{1}{\sqrt{2\widehat{p}}} \left[\sum_{l=0}^{\widehat{p}-1} \omega_{2\widehat{p}}^{i2l} (IW_{\widehat{p}}^{-1})_{2lj} + \sum_{l=0}^{\widehat{p}-1} \omega_{2\widehat{p}}^{i(2l+1)} (IW_{\widehat{p}}^{-1})_{(2l+1)j} \right],$$

$$= \frac{1}{\sqrt{2\widehat{p}}} \left[S_p + S_i \right].$$
(3.55)

Après cela, on calcule chaque somme S_p et S_i séparément, en utilisant (3.47) et (3.48) :

$$S_{p} = \frac{1}{\sqrt{\hat{p}}} \sum_{l=0}^{\hat{p}-1} \omega_{\hat{p}}^{il} \omega_{2\hat{p}}^{-2lj} = \frac{\hat{p}}{\sqrt{\hat{p}}} = \sqrt{\hat{p}} \text{ si } i \stackrel{\hat{p}}{=} j \text{ sinon } 0.$$

$$S_{i} = \frac{1}{2\sqrt{\hat{p}}} \sum_{l=0}^{\hat{p}-1} \omega_{\hat{p}}^{il} \omega_{2\hat{p}}^{i} \left(\omega_{2\hat{p}}^{-2lj} + \omega_{2\hat{p}}^{-2lj} \omega_{\hat{p}}^{-j} \right) = \frac{1}{2\sqrt{\hat{p}}} \sum_{l=0}^{\hat{p}-1} \omega_{\hat{p}}^{(i-j)l} \omega_{2\hat{p}}^{i} \left(1 + \omega_{\hat{p}}^{-j} \right),$$

$$= \frac{\omega_{2\hat{p}}^{i} \left(1 + \omega_{\hat{p}}^{-j} \right) \hat{p}}{2\sqrt{\hat{p}}} \text{ si } i \stackrel{\hat{p}}{=} j \text{ sinon } 0.$$
(3.56)
(3.56)
(3.56)

Enfin, en utilisant (3.49) puis additionnant S_p et S_i , on obtient pour $i \stackrel{\widehat{p}}{=} j$:

$$(W_{2\widehat{p}}IW_{\widehat{p}}^{-1})_{i,j} = \frac{1}{\sqrt{2\widehat{p}}} \left[\frac{\widehat{p}}{\sqrt{\widehat{p}}} + \frac{\cos\left(\frac{i\pi}{\widehat{p}}\right)\widehat{p}}{\sqrt{\widehat{p}}} \right] = \frac{1}{\sqrt{2}} \left(1 + \cos\left(\frac{i\pi}{\widehat{p}}\right) \right).$$

L'utilisation de la notation $\theta_{\kappa} = \kappa \pi / \hat{p}$ et l'observation que $\kappa = i$ permettent d'obtenir (3.41), ce qui conclut la preuve.

Observation 3.2.3 – On peut noter la propriété suivante de ces opérateurs, essentielle pour la suite :

$$\bar{R} I = D_R (D_I + D_I^*) = \mathbb{I}_p.$$
 (3.58)

Cette dernière traduit que, après une interpolation linéaire puis une restriction de type injection, tout champ reste inchangé.

3.2.4.c Approximation de commutativité

Non commutativité des opérateurs fin et grossier. Avant d'introduire le facteur d'amplification de PARAREAL avec grossissement spatial, une importante observation peut être faite en analysant la forme des différents opérateurs fin et grossier réécrits dans la base de Fourier. On rappelle l'expression pour la Jacobienne de \mathcal{F} donnée en Sec. 3.2.4.a :

$$M_{\mathcal{F}} = W_{2\hat{p}} \begin{bmatrix} D_{\mathcal{F}} & 0\\ 0 & D_{\mathcal{F}}^* \end{bmatrix} W_{2\hat{p}}^{-1},$$
(3.59)

puis en utilisant le Lemme 3.2.1, on écrit la Jacobienne de $\widehat{\mathcal{G}}$ sous la forme :

$$M_{\widehat{\mathcal{G}}} = W_{2\widehat{p}} \begin{bmatrix} D_I \\ D_I^* \end{bmatrix} D_{\widehat{\mathcal{G}}} \begin{bmatrix} D_R & D_R \end{bmatrix} W_{2\widehat{p}}^{-1}.$$
(3.60)

Alors que les Jacobiennes des solveurs grossier et fin commutaient précédemment lorsque ces derniers partageaient la même grille spatiale, ce n'est plus le cas ici puisque, après les calculs et utilisation du caractère diagonal des opérateurs mis en jeu, nous obtenons :

$$M_{\widehat{\mathcal{G}}}M_{\mathcal{F}} = W_{2\widehat{p}} \begin{bmatrix} D_I & \\ & D_I^* \end{bmatrix} \begin{bmatrix} D_{\mathcal{F}}D_{\widehat{\mathcal{G}}} & (D_{\mathcal{F}}^*)D_{\widehat{\mathcal{G}}} \\ D_{\mathcal{F}}D_{\widehat{\mathcal{G}}} & (D_{\mathcal{F}}^*)D_{\widehat{\mathcal{G}}} \end{bmatrix} \begin{bmatrix} D_R & \\ & D_R \end{bmatrix} W_{2\widehat{p}}^{-1},$$
(3.61)

et

$$M_{\mathcal{F}}M_{\widehat{\mathcal{G}}} = W_{2\widehat{p}} \begin{bmatrix} D_I \\ D_I^* \end{bmatrix} \begin{bmatrix} D_{\mathcal{F}}D_{\widehat{\mathcal{G}}} & D_{\mathcal{F}}D_{\widehat{\mathcal{G}}} \\ (D_{\mathcal{F}}^*)D_{\widehat{\mathcal{G}}} & (D_{\mathcal{F}}^*)D_{\widehat{\mathcal{G}}} \end{bmatrix} \begin{bmatrix} D_R \\ D_R \end{bmatrix} W_{2\widehat{p}}^{-1}.$$
(3.62)

Observation 3.2.4 – La non commutativité des opérateurs grossiers et fin n'est pas surprenante :

- si \mathcal{F} est appliqué avant $\hat{\mathcal{G}}$, la solution donnée à $\hat{\mathcal{G}}$, bien que restreinte à la grille grossière, contient une partie des informations liée aux composantes à hautes fréquences de la solution initiale sur grille fine, qui ont été propagées par \mathcal{F} ,
- si $\hat{\mathcal{G}}$ est appliqué avant \mathcal{F} , la restriction sur grille grossière appliquée à la solution initiale supprime toute information liée aux hautes fréquences, ce qui fait que ces dernières n'ont aucune influence sur la solution finale, au contraire de la situation précédente.

Problème et solution proposée. De par cette non-commutativité de $M_{\mathcal{F}}$ et $M_{\widehat{\mathcal{G}}}$, l'expression de la matrice Jacobienne de PARAREAL est non seulement très difficile à obtenir analytiquement, mais aussi particulièrement inexploitable pour analyser l'influence de l'algorithme sur la solution initiale (plus de détails sont donnés en Ap. G). C'est pourquoi nous proposons de négliger la non-commutativité des opérateurs, et d'utiliser l'approximation suivante :

$$M_{\widehat{G}}M_{\mathcal{F}} \simeq M_{\mathcal{F}}M_{\widehat{G}}.$$
(3.63)

Par la suite, celle-ci sera utilisée de telle sorte que tout produit matriciel incluant l fois $M_{\widehat{\mathcal{G}}}$ et m fois $M_{\mathcal{F}}$ placées dans un ordre quelconque, avec $(l,m) \in \mathbb{N}^2$, soit approximé par $M^l_{\mathcal{F}}M^m_{\widehat{\mathcal{G}}}$. Par abus de langage, on désignera par *"facteur d'amplification de PARAREAL avec grossissement spatial"* l'équivalent de celui obtenu sans changement de grille, en suivant l'approximation détaillée précédemment. Une validation de cette approximation est donnée en Sec. 3.3.3.a et Ap. G.

3.2.4.d Facteur d'amplification de PARAREAL avec grossissement spatial

Proposition 3.2.6 — On considère l'application de PARAREAL avec grossissement spatial au problème d'advection mono-dimensionnel (3.4), sur des grilles spatiales périodiques uniformes, utilisant une restriction de type injection et une interpolation linéaire pour transférer la solution sur chaque niveau de grille respectif. On considère les paramètres suivants pour chaque propagateur :

- 1. Nombres CFL : $CFL_{\mathcal{F}}$ et $CFL_{\widehat{G}}$ (pouvant s'écrire en fonction de $CFL_{\mathcal{F}}$),
- 2. Symboles de Fourier : $z_{adv,\mathcal{F}}$ et $z_{adv,\widehat{\mathcal{G}}}$,
- 3. Facteurs d'amplifications : $g_{\mathcal{F}}$ et $g_{\widehat{G}}$,
- 4. Nombres de pas de temps par time-slice : $s_{\mathcal{F}}$ et $s_{\widehat{G}}$.

On pose les notations suivantes :

$$R(\theta_{\kappa}) = g_{\widehat{\mathcal{G}}}(-CFL_{\widehat{\mathcal{G}}}z_{adv,\widehat{\mathcal{G}}}(\theta_{\kappa}))^{s_{\widehat{\mathcal{G}}}},$$

$$r(\theta_{\kappa}) = g_{\mathcal{F}}(-CFL_{\mathcal{F}}z_{adv,\mathcal{F}}(\theta_{\kappa}))^{s_{\mathcal{F}}}.$$

Dans ces conditions, le facteur d'amplification de PARAREAL avec grossissement spatial à la fin d'une time-slice n à l'itération k (différent de n) s'écrit :

$$\forall \theta_{\kappa} \in [0,\pi] \ g_{\mathcal{P},TG}^{k,n}(\theta_{\kappa}) = g_{IR}(\theta_{\kappa}) \ g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \ g_{TG}(\theta_{\kappa}), \tag{3.64}$$

avec g_{IR} le facteur d'amplification des opérateurs de transfert, qui dans ce cadre s'écrit :

$$g_{IR} = \frac{1 + \cos(\theta_{\kappa})}{2},\tag{3.65}$$

 $g_{\mathcal{P}}$ défini en Prop 3.2.5 et g_{TG} le facteur d'amplification induit par le grossissement spatial :

$$g_{TG}(\theta_{\kappa}) = 1 + \frac{\hat{U}_0(\pi - \theta_{\kappa})}{\hat{U}_0(\theta_{\kappa})}.$$
(3.66)

Lorsque PARAREAL est redémarré N_r fois, le facteur d'amplification s'écrit ainsi :

$$g_{\mathcal{P},TG}^{k,n}(CFL_{\mathcal{F}},\theta_{\kappa}) = g_{IR}(\theta_{\kappa}) \ g_{\mathcal{P}}^{k,n}(CFL_{\mathcal{F}},R(2\theta_{\kappa}),r(\theta_{\kappa}))^{N_{r}} \ g_{TG}(\theta_{\kappa}).$$
(3.67)

Démonstration. En repartant de (3.60) et en utilisant (3.58), on peut facilement montrer que :

$$\forall l \in \mathbb{N}_*, \ M_{\widehat{\mathcal{G}}}^l = W_{2\widehat{p}} \begin{bmatrix} D_I \\ D_I^* \end{bmatrix} D_{\widehat{\mathcal{G}}}^l \begin{bmatrix} D_R & D_R \end{bmatrix} W_{2\widehat{p}}^{-1}, \tag{3.68}$$

ce qui nous permet de calculer le produit des Jacobiennes fine et grossière élevées à des puissances strictement positives arbitraires :

$$\forall (l,m) \in \mathbb{N} \times \mathbb{N}^*, \ M_{\mathcal{F}}^m M_{\widehat{\mathcal{G}}}^l = W_{2\widehat{p}} \begin{bmatrix} D_{\mathcal{F}}^m & \\ & (D^*)_{\mathcal{F}}^m \end{bmatrix} \begin{bmatrix} D_I \\ D_I^* \end{bmatrix} D_{\widehat{\mathcal{G}}}^l \begin{bmatrix} D_R & D_R \end{bmatrix} W_{2\widehat{p}}^{-1},$$

$$= W_{2\widehat{p}} \begin{bmatrix} D_I & \\ & D_I^* \end{bmatrix} \begin{bmatrix} D_{\mathcal{F}}^m D_{\widehat{\mathcal{G}}}^l \\ (D_{\mathcal{F}}^*)^m D_{\widehat{\mathcal{G}}}^l \end{bmatrix} \begin{bmatrix} D_R & D_R \end{bmatrix} W_{2\widehat{p}}^{-1}.$$

$$(3.69)$$

Or, par utilisation de l'approximation de commutativité décrite en Sec. 3.2.4.c et par construction, PARAREAL correspond à une combinaison linéaire de $M_{\mathcal{F}}^m M_{\hat{c}}^l$ *i.e.* :

$$M_{\mathcal{P}}(k,n) = g_{\mathcal{P}}^{k,n}(M_{\mathcal{F}}, M_{\widehat{\mathcal{G}}}) = \sum_{i=0}^{k} \binom{n}{i} (M_{\mathcal{F}} - M_{\widehat{\mathcal{G}}})^{i} M_{\widehat{\mathcal{G}}}^{n-i} = \sum_{l,m} \beta_{l,m}(k,n) M_{\mathcal{F}}^{m} M_{\widehat{\mathcal{G}}}^{l},$$
(3.70)

avec k le nombre d'itérations, n l'indice de la *time-slice*, les $\beta_{l,m}(k, n)$ des coefficients dépendant de k et n, et $g_{\mathcal{P}}$ défini en Prop. 3.2.2.

Ainsi, en utilisant la commutativité de chaque bloc diagonal, on peut écrire la Jacobienne de PARAREAL pour chaque itération $k \neq n$ et *time-slice* n:

$$\begin{split} M_{\mathcal{P}}(k,n) &= W_{2\widehat{p}} \begin{bmatrix} D_{I} \\ D_{I}^{*} \end{bmatrix} \begin{bmatrix} \frac{g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}, D_{\widehat{G}}) & g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}, D_{\widehat{G}})}{g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}^{*}, D_{\widehat{G}}) & g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}^{*}, D_{\widehat{G}})} \end{bmatrix} \begin{bmatrix} D_{R} \\ D_{R} \end{bmatrix} W_{2\widehat{p}}^{-1}, \\ &= W_{2\widehat{p}} \begin{bmatrix} \frac{D_{I}g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}, D_{\widehat{G}})D_{R} & D_{I}g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}, D_{\widehat{G}})D_{R}}{D_{I}g_{\mathcal{P}}^{k,n}(D_{\mathcal{F}}^{*}, D_{\widehat{G}})D_{R}} \end{bmatrix} W_{2\widehat{p}}^{-1}, \\ &= W_{2\widehat{p}}T_{\mathcal{P}}(k,n)W_{2\widehat{p}}^{-1}, \end{split}$$
(3.71)

avec

$$g_{\mathcal{P}}^{k,n}(D_1, D_2) = \sum_{i=0}^k \binom{n}{i} (D_1 - D_2)^i D_2^{n-i}.$$
(3.72)

Lorsque la même grille spatiale était utilisée pour les deux propagateurs, la forme diagonale de la Jacobienne de PARAREAL dans l'espace de Fourier nous avait permis d'en déduire directement le facteur d'amplification pour chaque nombre d'onde. Ce n'est cependant plus le cas pour PARAREAL avec grossissement spatial, la Jacobienne ayant une forme désormais tridiagonale dans l'espace de Fourier. Cette dernière, que l'on note $T_{\mathcal{P}}(k, n)$, est représentée graphiquement en Fig. 3.2.



FIGURE 3.2 – Représentation graphique de la Jacobienne de PARAREAL avec grossissement spatial dans l'espace de Fourier.

On considère donc l'application de PARAREAL avec grossissement spatial à une solution initiale dans l'espace de Fourier, représentée par le produit matrice vecteur suivant :

$$\hat{U}_{n}^{k} = T_{\mathcal{P}}(k, n)\hat{U}_{0}.$$
(3.73)

Plus spécialement, on analyse chaque composante de \hat{U}_n^k associée à un nombre d'onde donné θ_{κ} , composante que l'on note $\hat{U}_n^k(\theta_{\kappa})$. Le produit matrice vecteur (3.73) est représenté graphiquement

en Fig. 3.2, un calcul nous donnant (pour $N_r = 1$) :

$$\forall \theta_{\kappa} \in [0, \pi], \quad \hat{U}_{n}^{k}(\theta_{\kappa}) = \frac{1 + \cos(\theta_{\kappa})}{\sqrt{2}} g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \frac{1}{\sqrt{2}} \hat{U}_{0}(\theta_{\kappa}) + \frac{1 + \cos(\theta_{\kappa})}{\sqrt{2}} g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \frac{1}{\sqrt{2}} \hat{U}_{0}(\theta_{\kappa} + \pi),$$

$$= \frac{1 + \cos(\theta_{\kappa})}{2} g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \left[\hat{U}_{0}(\theta_{\kappa}) + \hat{U}_{0}(\theta_{\kappa} + \pi) \right].$$

$$(3.74)$$

Cette expression peut être réduite à une forme plus compacte, en considérant la symétrie de $\hat{U}_n^k(\theta_\kappa)$ par rapport à π , soit $\hat{U}_n^k(\theta_\kappa + \pi) = \hat{U}_n^k(\pi - \theta_\kappa)$. Cela nous permet d'écrire :

$$\forall \theta_{\kappa} \in [0, \pi], \ \hat{U}_{n}^{k}(\theta_{\kappa}) = \left[\frac{1 + \cos(\theta_{\kappa})}{2}\right] g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \left[\hat{U}_{0}(\theta_{\kappa}) + \hat{U}_{0}(\pi - \theta_{\kappa})\right],$$

$$= \left[\frac{1 + \cos(\theta_{\kappa})}{2}\right] g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \left[1 + \hat{U}_{0}(\pi - \theta_{\kappa})/\hat{U}_{0}(\theta_{\kappa})\right] \hat{U}_{0}(\theta_{\kappa}).$$

$$(3.75)$$

De cela on identifie :

$$g_{IR}(\theta_{\kappa}) = \frac{1 + \cos(\theta_{\kappa})}{2} \quad \text{et} \quad g_{TG}(\theta_{\kappa}) = 1 + \frac{\hat{U}_0(\pi - \theta_{\kappa})}{\hat{U}_0(\theta_{\kappa})}, \tag{3.76}$$

conduisant finalement à (3.64).

L'expression du facteur d'amplification pour PARAREAL redémarré peut être obtenu simplement en utilisant (3.71) et (3.58) respectivement, puis en réappliquant le même raisonnement que précédemment, ce qui complète la preuve.

Observation 3.2.5 — Contrairement à la Prop. 3.2.5, le facteur d'amplification de PARAREAL avec grossissement spatial dépend ici de la solution initiale, de par l'influence de g_{TG} .

Observation 3.2.6 — On peut aussi considérer la permutation d'indices suivante, qui couple les fréquences de la manière suivante :

$$\hat{U}' = \begin{bmatrix} \dots \\ \vdots \\ \hat{U}(\theta_{\kappa}) \\ \hat{U}(\pi + \theta_{\kappa}) \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = P \begin{bmatrix} \dots \\ \hat{U}(\theta_{\kappa}) \\ \vdots \\ \vdots \\ \hat{U}(\pi + \theta_{\kappa}) \\ \vdots \\ \vdots \end{bmatrix} = P\hat{U},$$
(3.77)

avec P orthogonale, soit $P^{-1} = P^t$. On peut ainsi ré-écrire (3.73) dans une base de Fourier réordonnée :

$$\hat{U'}_{n}^{\kappa} = PT_{\mathcal{P}}(k,n)P^{t}\hat{U'}_{0} = T'_{\mathcal{P}}(k,n)\hat{U'}_{0}, \qquad (3.78)$$

avec $T'_{\mathcal{P}}(k,n)$ une matrice diagonale par bloc, chacun des blocs étant associé à un $\theta_{\kappa} \in [0,\pi]$ et s'écrivant ainsi :

$$B_{\theta_{\kappa}} = \left[\begin{array}{c|c} g_{IR}(\theta_{\kappa}) \ g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) & g_{IR}(\theta_{\kappa}) \ g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa})) \\ \hline g_{IR}(\theta_{\kappa} + \pi) \ g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa} + \pi)) & g_{IR}(\theta_{\kappa} + \pi) \ g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa} + \pi)) \end{array} \right].$$

$$(3.79)$$

En particulier, si \hat{p} est impair, ce bloc se réduit à une composante 1×1 pour certains indices. Cette formulation permet ainsi d'extraire des informations de convergence par le rayon spectral $\rho(B_{\theta_{\kappa}})$ de chaque bloc. Ce dernier peut se calculer directement en observant que $B_{\theta_{\kappa}}$ est singulière, donc

une de ses deux valeurs propres λ_1 , λ_2 est nulle, ce qui induit $\rho(B_{\theta_{\kappa}}) = \max |\lambda_i| = |\lambda_1 + \lambda_2| = |trace(B_{\theta_{\kappa}})|$, soit :

$$\rho(B_{\theta_{\kappa}}) = g_{IR}(\theta_{\kappa}) |g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa}))| + g_{IR}(\theta_{\kappa} + \pi) |g_{\mathcal{P}}^{k,n}(R(2\theta_{\kappa}), r(\theta_{\kappa} + \pi))|$$
(3.80)

Cet ordonnancement est par ailleurs communément utilisé dans la communauté multi-grille pour la LFA (cf. [Trottenberg et al., 2001, Chap. 4]).

Proposition 3.2.7 : Facteur d'amplification des opérateurs de transfert avec interpolation d'ordre élevée — En conservant l'injection comme opérateur de restriction, on peut aussi exprimer g_{IR} pour des schémas d'interpolation d'ordres élevés impairs. Il s'écrit, pour l'ordre allant de 1 à 7 :

$$g_{IR,o1}(\theta_{\kappa}) = \frac{1 + \cos(\theta_{\kappa})}{2},\tag{3.81}$$

$$g_{IR, o3}(\theta_{\kappa}) = \frac{8 + 9\cos(\theta_{\kappa}) - \cos(3\theta_{\kappa})}{16},$$
(3.82)

$$g_{IR,o5}(\theta_{\kappa}) = \frac{128 + 150\cos(\theta_{\kappa}) - 25\cos(3\theta_{\kappa}) + 3\cos(5\theta_{\kappa})}{256},$$
(3.83)

$$g_{IR,o7}(\theta_{\kappa}) = \frac{1024 + 1225\cos(\theta_{\kappa}) - 245\cos(3\theta_{\kappa}) + 49\cos(5\theta_{\kappa}) - 5\cos(7\theta_{\kappa})}{2048}.$$
 (3.84)

Démonstration. Le fait de garder l'injection permet de conserver la validité de (3.58). De ce fait, le raisonnement utilisé pour l'interpolation d'ordre 1 peut être tenu dans le cadre de l'utilisation d'interpolations d'ordres élevés. Cependant, les calculs ne seront pas détaillés ici, les formules pouvant être directement obtenues par analogie avec les symboles de Fourier d'opérateurs de transfert utilisés dans la littérature multi-grille (*cf.* [Wienands et Joppich, 2004, Sec. 6.2.4]).

3.3 Étude des formes explicites de PARAREAL pour les problèmes advectifs

Dans cette section on applique l'analyse précédemment détaillée en Sec. 3.2 à des problèmes advectifs de complexité croissante. En premier lieu, l'analyse est appliquée à des problèmes scalaires en Sec. 3.3.1, pour identifier les caractéristiques principales de PARAREAL utilisé avec des méthodes explicites. L'application est ensuite étendue au problème mono-dimensionnel d'advection pure en Sec. 3.3.2, et l'utilisation de PARAREAL avec grossissement spatial est étudiée en Sec. 3.3.3. Différentes méthodes ERK (décrites plus en détails en Ap. B) sont utilisées :

- FE : méthode de Forward Euler, ou encore ERK d'ordre 1 en une étape,
- *RK21* : *méthode ERK d'ordre* 1 *en deux étapes*,
- HEUN2 : méthode ERK d'ordre 2 en deux étapes,
- *RKC4* : *méthode ERK d'ordre* 4 *en quatre étapes*.

Les contributions principales de cette section sont représentées par les différentes observations données tout au long de l'analyse, qui sont synthétisées pour les trois parties principales de l'étude en Sec. 3.3.1.e, Sec. 3.3.2.e et Sec. 3.3.3.d respectivement, et de manière plus globale en Sec. 3.3.4.

3.3.1 Cas scalaire

Dans l'ensemble de cette sous-section, on se focalisera sur l'application de PARAREAL au problème autonome suivant :

$$u'(t) = \lambda u(t), \ \lambda = -\iota a - d, \ u(0) = 1, \ t \in [0, T], \ (a, d) \in \mathbb{R}^*_+ \times \mathbb{R}_+,$$
(3.85)

avec $\iota^2 = -1$.

3.3.1.a Motivation de l'étude

On considère le problème original (3.2), et plus particulièrement les valeurs propres de la matrice (A + Z). En reprenant le formalisme de Sec. 3.2.3.a, celles-ci s'écrivent pour une discrétisation spatiale "exacte" (*i.e.* spectrale) sous la forme :

$$\lambda = -c \frac{\iota \theta_{\kappa}}{\delta_x} - \nu \frac{\theta_{\kappa}^2}{\delta_x^2}.$$
(3.86)

En utilisant une définition simplifiée du nombre de Reynolds $Re = c/\nu$, et en normalisant les valeurs propres par $\frac{c}{\delta r}$, on peut écrire :

$$\Lambda(\theta_{\kappa}) = \frac{\delta_x \lambda}{c} = -\iota \theta_{\kappa} - \frac{1}{Re} \frac{\theta_{\kappa}^2}{\delta_x}.$$
(3.87)

Or, en plus de réduire l'influence du terme diffusif dans (3.1), l'augmentation du nombre de Reynolds a pour effet de réduire la taille des plus petites échelles spatiales du problème (ce point sera revu plus en détails dans le Chap. 4). Pour notre exemple, on choisit une loi simple inspirée de [Pope, 2000, Sec. 9.1] où la plus petite échelle spatiale du problème η est liée au Re selon :

$$\eta = Re^{-3/4}.$$
(3.88)

Le maillage spatial utilisé à un Re donné doit donc être suffisamment fin pour représenter l'échelle η . Pour simplifier le raisonnement on choisit $\delta_x = \eta$, ce qui permet d'obtenir la représentation, donnée en Fig. 3.3, des valeurs propres normalisées $\Lambda(\theta_{\kappa})$ dans le plan complexe en fonction de Re.



FIGURE 3.3 – Valeurs propres normalisées Λ pour le problème d'advection-diffusion linéaire (3.2).

On rappelle que plus la partie réelle négative d'une valeur propre est importante, plus la composante fréquentielle associée à cette valeur propre sera dissipée lors de la résolution du problème. Des valeurs propres purement réelles négatives sont donc caractéristiques d'un problème diffusif (équation de type parabolique). À l'opposé, les valeurs propres purement imaginaires correspondent à des composantes fréquentielles qui seront "transportées" au cours du problème, sans perte d'énergie. Ces valeurs propres sont donc caractéristiques d'un problème advectif (équation de type hyperbolique) [Polyanin et Nazaikinskii, 2015].

On observe en Fig. 3.3 que l'augmentation de Re va contribuer à rapprocher les valeurs propres haute fréquence $(\theta_{\kappa} \to \pi)$ vers l'axe imaginaire. Cependant, même à un nombre de Reynolds très élevé, celles-ci ne seront jamais entièrement sur l'axe imaginaire (sauf pour le cas $Re = \infty$, qui correspond à l'advection pure). De plus, leur partie imaginaire (liée à leur fréquence associée) restera identique. De ce fait, étudier (3.85) en considérant différentes valeurs de a et d permet d'avoir un premier aperçu du comportement de l'algorithme pour des valeurs propres représentant un problème dont la caractéristique hyperbolique devient prédominante ($d \to 0$), et associées à des basses ou haute fréquences ($a \to 0$ ou $a \to \pi$ respectivement).

3.3.1.b Paramètres du problème

Sauf indication contraire, on considérera les paramètres suivants pour (3.85), inspirés du problème proposé par [Chen *et al.*, 2015]. On fixe T = 100, et on décompose l'intervalle temporel global en N = 100 *time-slices*. Suivant [Chen *et al.*, 2015], on utilise en premier lieu *Forward-Euler* pour les solveurs temporels fin et grossier ($g_{\mathcal{G}}(\hat{z}) = g_{\mathcal{F}}(\hat{z}) = 1 + \hat{z}$), avec un pas de temps $\delta_t = 0.001$ et $\Delta_t = 0.5$ respectivement, ce qui nous donne $s_{\mathcal{F}} = 1000$ et $s_{\mathcal{G}} = 2$. Le choix d'un pas de temps relativement faible permet d'assimiler le solveur fin à un intégrateur exact. De ce fait, la convergence de PARAREAL est essentiellement pilotée par le schéma d'intégration du solveur grossier (comme on le verra plus tard en Fig. 3.4), ce qui permet d'observer directement son influence.

On définit l'erreur commise par PARAREAL à l'itération k comme étant

$$\epsilon_k = \frac{1}{N} \sum_{n=1}^{N} \left| u_n^k - u(nT/N) \right|,$$
(3.89)

celle-ci pouvant être explicitée dans le cadre spécifié ci-dessus en utilisant Prop. 3.2.2 et (3.9) :

$$\epsilon_k = \frac{1}{N} \sum_{n=1}^N \left| \sum_{i=0}^k \binom{n}{i} \left[\left((1+\lambda\delta_t)^{s_{\mathcal{F}}} - (1+\lambda\Delta_t)^{s_{\mathcal{G}}} \right)^i (1+\lambda\Delta_t)^{s_{\mathcal{G}}(n-i)} \right] - e^{\frac{\lambda nT}{N}} \right|.$$
(3.90)

3.3.1.c Stabilité de PARAREAL avec diminution de la partie diffusive

Un premier exemple. On examine en premier lieu la convergence de PARAREAL pour $\lambda = -d - \iota$, avec $d \in \{0, 0.1, 0.5, 2\}$. L'erreur ϵ_k est représentée en Fig. 3.4a. On peut y observer une évolution typique de la convergence de PARAREAL : à partir d'un niveau initial (fixé par le solveur grossier), l'erreur décroit jusqu'à atteindre un plateau qui traduit la convergence de l'algorithme (*e.g.* pour d = 2.0). Cependant ce comportement ne s'observe pas dans toutes les situations, car on peut aussi voir une divergence initiale de PARAREAL pour les valeurs propres les moins diffusives (*e.g.* pour $d \in \{0, 0.1, 0.5\}$), ce qui est décrit comme le résultat d'une instabilité de l'algorithme par [Chen *et al.*, 2015, Sec. 2]. Le fait de changer l'intégrateur fin, comme représenté en Fig. 3.4b, a un impact négligeable sur l'évolution de la convergence. La seule différence entre les deux résultats correspond à l'erreur finale lorsque PARAREAL converge, qui est tout simplement l'erreur effectuée par le solveur fin. Ceci confirme l'hypothèse faite précédemment, et nous pousse à examiner l'influence de la méthode temporelle utilisée au sein du solveur grossier.

Comprendre les instabilités de PARAREAL. Avant d'aller plus loin, on peut utiliser le facteur d'amplification de PARAREAL, en particulier son contour de stabilité à chaque itération, pour mieux comprendre les résultats présentés en Fig. 3.4a. Le contour de stabilité de PARAREAL pour la 50^{ème}



FIGURE 3.4 – Influence du caractère advectif du problème scalaire sur la convergence de PARAREAL. Erreur ϵ_k donnée par (3.89) en fonction du nombre d'itérations k pour le problème autonome, avec $\lambda = -d - \iota$, $\mathcal{G} = FE$.

time-slice est représenté en Fig. 3.5 pour plusieurs itérations, de même que les valeurs propres λ associées à chaque d.



FIGURE 3.5 – Évolution de la stabilité de PARAREAL en fonction des valeurs propres du problème autonome. Contour de stabilité de PARAREAL représenté à différentes itérations k, pour la time-slice n = 50, $\mathcal{F} = FE, s_{\mathcal{F}} = 1000, \mathcal{G} = FE, s_{\mathcal{G}} = 2.$

Chaque valeur propre correspond donc à un point fixe dans le plan complexe, autour duquel évolue le contour de stabilité de PARAREAL au cours des itérations. On peut faire les observations suivantes :

- 1. Les valeurs propres pour $d \in \{0, 0.1\}$ sont situées à l'extérieur du contour de stabilité de \mathcal{G} (qui est celui de PARAREAL pour k = 0). De ce fait, la divergence initiale de PARAREAL est due principalement à une instabilité numérique (amplification) induite par \mathcal{G} et non pas à l'algorithme lui-même.
- 2. Pour les premières itérations, le contour de PARAREAL se rétracte dans sa partie réelle négative. La valeur propre pour d = 0.5 se retrouve alors légèrement hors de la zone stable pour

quelques itérations, ce qui explique ainsi la divergence initiale légère de PARAREAL pour cette valeur propre.

- 3. Simultanément à sa rétractation pour les première itérations, le contour s'aplatit et s'étend sur l'axe imaginaire, ce qui a pour effet de ramener après un certain nombre d'itérations les valeurs propres initialement instables ($d \in \{0, 0.1\}$) dans la zone stable, faisant ainsi converger l'algorithme.
- 4. Pour les dernières itérations, le contour de stabilité de PARAREAL ne se rétracte plus et s'étend dans toutes les directions, en restant toujours aplati sur l'axe imaginaire, ce qui permet la convergence de toutes les valeurs propres après un certain nombre d'itérations.

Ceci montre donc deux comportements paradoxaux de PARAREAL : la rétractation du contour de stabilité pour les premières itérations provoque des instabilités pour les valeurs propres en limite de stabilité pour \mathcal{G} , mais son extension le long de l'axe imaginaire, après plusieurs itérations, permet une intégration numérique stable pour des valeurs propres qui initialement n'étaient pas dans le contour de stabilité de \mathcal{G} .

Enfin, on peut déjà conclure que selon les valeurs propres considérées, le choix de la méthode d'intégration de G est d'une importance primordiale, celle-ci fixant le contour de stabilité initial de PARAREAL, influant de ce fait sur sa convergence.

Observation 3.3.1 — Pour le problème autonome, le contour de stabilité de PARAREAL évolue en fonction des itérations en partant de la forme de celui du solveur grossier. Pour les premières itérations, il se rétracte dans sa partie réelle négative tout en s'aplatissant et s'étendant sur l'axe imaginaire. Puis, pour les dernières itérations, le contour s'étend dans toutes les directions, en restant aplati sur l'axe imaginaire.

3.3.1.d Influence de l'intégrateur temporel grossier

Du choix judicieux de l'intégrateur selon les valeurs propres du problème. On choisit maintenant un intégrateur grossier ayant pour caractéristique d'inclure dans son contour de stabilité une partie de l'axe imaginaire. Le choix s'est porté sur la méthode RKC4, avec $s_{\mathcal{G}} = 1$, et l'évolution de ϵ_k est représentée en Fig. 3.6.a.



FIGURE 3.6 – Influence de l'intégrateur temporel grossier sur la convergence de PARAREAL selon le caractère advectif du problème scalaire. Erreur ϵ_k donnée par (3.89) en fonction du nombre d'itérations k pour le problème autonome, avec $\lambda = -d - \iota$, $\mathcal{F} = HEUN2$.
On observe une bien meilleure convergence de PARAREAL pour les valeurs faibles de d, ce qui s'explique en examinant le contour de stabilité de PARAREAL représenté en Fig. 3.7. L'ensemble des valeurs propres, même pour d = 0, reste à l'intérieur du contour de stabilité à chaque itération et ne souffre pas de sa rétractation. Cependant, on observe que le taux de convergence n'est pas le même pour chaque valeur propre, ce dernier étant le plus faible pour la valeur propre la plus diffusive.



FIGURE 3.7 – Évolution de la stabilité de PARAREAL pour un schéma d'intégration temporel grossier incluant une partie de l'axe imaginaire dans son contour de stabilité. Ce dernier est représenté pour PARAREAL à différentes itérations k, pour la time-slice n = 50, $\mathcal{F} = FE$, $s_{\mathcal{F}} = 1000$, $\mathcal{G} = RKC4$, $s_{\mathcal{G}} = 1$.

Cette dépendance des valeurs propres selon le solveur utilisé est illustrée plus particulièrement avec la Fig. 3.6.b. Dans cette dernière, la méthode de HEUN2 est choisie pour \mathcal{G} , avec $s_{\mathcal{G}} = 2$. On remarquera que les deux intégrateurs grossiers choisis effectuent le même nombre d'étapes par *time-slice* (1 × 4 pour RKC4, 2 × 2 pour HEUN2), et de ce fait ont le même coût. Or leur convergence n'est pas équivalente selon la valeur propre considérée : alors que RKC4 favorise la convergence des valeurs propres "advectives" au détriment des valeurs propres "diffusives", HEUN2 induit l'effet inverse.

Ceci nous permet de conclure que le choix du solveur grossier doit être fait en considérant l'aspect du problème (et des valeurs propres associées) : pour des problèmes plutôt hyperboliques il est préférable de choisir un intégrateur dont le contour de stabilité englobe une partie de l'axe imaginaire, tandis que pour des problèmes plutôt paraboliques, l'utilisation d'une méthode explicite dont le contour s'étend vers la partie réelle négative du plan complexe sera mieux adaptée.

Observation 3.3.2 — En fonction du choix du schéma d'intégration temporel pour le solveur grossier, la convergence de PARAREAL n'est pas la même selon les caractéristiques des valeurs propres du problème autonome résolu. En particulier, pour des valeurs propres situées proches de l'axe imaginaire, l'utilisation d'un intégrateur temporel dont le contour de stabilité englobe une partie de l'axe imaginaire est préférable.

Influence du solveur grossier selon la localisation des valeurs propres sur l'axe imaginaire. On considère maintenant $\lambda = -\iota a$, avec $a \in \{0.1, 0.5, 1, 2\}$. On qualifiera de "basses fréquences" les valeurs propres pour $a \to 0$, et de "hautes fréquences" celles pour $a \to 2$. On choisit en premier lieu la même configuration que précédemment avec $\mathcal{G} = \text{RKC4}$, et on représente ϵ_k en Fig. 3.8a.

On y observe de manière générale une meilleure convergence pour les valeurs propres basses fréquences. De plus, on peut observer que pour a = 2.0, la convergence de PARAREAL est bien plus lente que la moyenne. Ceci peut se comprendre en regardant le contour de précision de PARAREAL représenté en Fig. 3.9.



FIGURE 3.8 – Influence de $s_{\mathcal{G}}$ sur la convergence de PARAREAL pour des valeurs propres purement imaginaires du problème autonome. Erreur ϵ_k donnée par (3.89) en fonction du nombre d'itérations k, avec $\lambda = -\iota a$, $\mathcal{F} = HEUN2, s_{\mathcal{F}} = 1000, \mathcal{G} = RKC4.$

Là encore, le poids de \mathcal{G} se fait sentir, car pour k = 0 (contour associé à \mathcal{G}), les valeurs propres hautes fréquences ne sont pas situées dans le contour de précision, et de ce fait intégrées de façon imprécise. Puis, au fur et à mesure des itérations, PARAREAL va les inclure dans son contour de précision dont on observe l'expansion sur l'axe imaginaire. Bien que cette expansion soit rapide pour les premières itérations (k = 1, 8) et permette d'englober la valeur propre pour a = 1 dès la première itération, elle devient beaucoup plus lente au fur et à mesure des itérations. De ce fait, la valeur propre pour a = 2 n'est incluse dans le contour de précision qu'après un plus grand nombre d'itérations, ce qui explique sa convergence initiale lente.

Ce défaut de convergence pour les valeurs propres hautes-fréquences peut néanmoins être corrigé au prix de l'augmentation de $s_{\mathcal{G}}$ (*i.e.* réduction du pas de temps grossier), comme on peut le voir en Fig. 3.8.b. Cependant, cela inclut nécessairement un coût de calcul plus important pour \mathcal{G} . Afin d'éviter ce surcoût, on peut envisager l'utilisation de méthodes d'intégration temporelle maximisant l'inclusion de l'axe imaginaire dans leur contour de précision (*cf.* Fig. 3.9). À cet effet, il est possible qu'un travail de recherche supplémentaire soit nécessaire.

Observation 3.3.3 — Pour des valeurs propres imaginaires pures du problème autonome, l'utilisation d'un solveur grossier dont le contour de stabilité inclut une partie de l'axe imaginaire permet une convergence rapide de PARAREAL. Pour des valeurs propres situées proches de l'origine du plan complexe, la convergence s'obtient dès les premières itérations. Pour des valeurs propres situées loin de l'origine, la convergence est significativement bien plus lente.

3.3.1.e Synthèse de l'étude du cas scalaire

L'application de PARAREAL avec des intégrateurs explicites sur le problème autonome (3.8) a été étudiée, en se focalisant plus particulièrement sur des valeurs propres du problème proches ou situées sur l'axe imaginaire.

En premier lieu, l'évolution caractéristique du contour de stabilité de PARAREAL (rétraction pour les premières itération, aplatissement sur l'axe imaginaire) a été mise en évidence et étudiée (Obs. 3.3.1). Puis, nous avons observé à quel point le choix de la méthode temporelle utilisée pour le solveur grossier pouvait être plus ou moins adapté en fonction de la proximité des valeurs propres considérées avec l'axe imaginaire (Obs. 3.3.2). Enfin, une convergence effective de PARAREAL



FIGURE 3.9 – Contour de précision de PARAREAL $\mathcal{E}_g(0.1)$ (3.15) pour le problème autonome, représenté après plusieurs itérations k pour la time-slice n = 50, $\mathcal{F} = FE$, $s_{\mathcal{F}} = 1000$, $\mathcal{G} = RKC4$, $s_{\mathcal{G}} = 1$. Toute valeur propre située à l'intérieur du contour est intégrée avec une erreur $E_\lambda < 0.1$, cf. (3.14).

pour des valeurs propres imaginaires pures a été observée lorsque le contour de stabilité du solveur grossier inclut une partie de l'axe imaginaire (Obs. 3.3.3).

De manière générale, la convergence et la stabilité de PARAREAL sont conditionnées par la position des valeurs propres relativement à l'axe imaginaire et l'origine du plan complexe. Or, dans le cadre de la résolution numérique du problème d'advection (3.4), la position des valeurs propres du problème dépend non seulement du type de discrétisation spatiale utilisée, mais aussi du *CFL* choisi. Le but de la section suivante est donc d'apporter plus d'éclairage sur cet aspect.

3.3.2 PARAREAL avec grille spatiale unique

Dans cette section, on considère l'application de PARAREAL sans grossissement de grille au problème d'advection pure mono-dimensionnelle (3.4). On note par U[o] et C[o'] respectivement les schémas de discrétisation upwind d'ordre o et centrés d'ordre o'. Une description plus complète est donnée en Ap. B.

3.3.2.a Motivations de l'étude

On choisit en premier lieu d'écarter le grossissement spatial afin d'étudier tout d'abord l'influence des schémas de discrétisation spatiale, lorsque ceux-ci sont combinés avec des méthodes ERK. Le solveur fin \mathcal{F} est construit en utilisant une intégration temporelle RKC4, ainsi que les schémas de discrétisation spatiale d'ordre élevé U5 et C6⁶. Les contours d'amplification des deux solveurs fins sont donnés en Fig. 3.10.

On notera que ces deux discrétisations se distinguent principalement par l'erreur de diffusion induite pour les hautes fréquences. Celle-ci est très importante pour le schéma *upwind*, à l'instar du schéma centré qui n'apporte de la diffusion que pour les fréquences intermédiaires ($\theta_{\kappa} \sim 0.6\pi$).

Ces deux types de schémas sont très largement utilisés dans la littérature. Alors que les schémas centrés sont favorisés pour des problèmes où une représentation précise de l'ensemble des composantes fréquentielles est requise (*e.g.* écoulements turbulents, [Larsson *et al.*, 2007]), les schémas *upwind* sont utilisés en présence de forts gradients (ou chocs), pour éviter des amplifications trop importantes des composantes à hautes fréquences. En particulier, les schéma WENO

^{6.} Ces deux schémas ont par ailleurs un coût de calcul équivalent, en terme de taille de stencil de discrétisation utilisé.



FIGURE 3.10 – Contours d'amplification (3.31) des deux discrétisations spatio-temporelles utilisées pour \mathcal{F} . Chaque niveau de couleur représente une plage de valeur pour $S_{g,z_{adv}}$ comprise dans un intervalle de taille 0.5, avec $S_{g,z_{adv}}(1)$ représenté en tirets rouges. La ligne en tirets grise représente la limite maximum de CFL stable.

[Jiang et Shu, 1996], utilisés dans HYBRID pour traiter les régions présentant des chocs, sont le produit d'une combinaison de schémas *upwind* d'ordre variable, qui est adaptée en fonction de la régularité de la solution. Pour des solutions régulières, un schéma WENO est par ailleurs équivalent à un schéma *upwind* du même ordre [Wang et Spiteri, 2007, Motamed *et al.*, 2011].

3.3.2.b Paramètres de l'étude

Utilisation de PARAREAL. On considère ici l'utilisation de PARAREAL dans un cadre proche d'une parallélisation espace-temps. Le nombre de *time-slices* N est donc un facteur multiplicatif du nombre de processus N_{proc}/N , a priori important, utilisé pour la parallélisation en espace, avec N_{proc} le nombre total de processus utilisés. De ce fait, on restreindra notre étude à un nombre faible de *time-slices*, *i.e.* N = 4, et ce jusqu'à la fin de l'étude. La justification de ce choix est par ailleurs examinée plus en détail en Chap. 4.

Construction du solveur grossier. Afin de construire le solveur grossier, on utilise une réduction d'ordre de la discrétisation spatio-temporelle de \mathcal{F} , de manière similaire à [Ruprecht et Krause, 2012]. Pour simplifier l'étude, un même nombre *CFL* est choisi pour \mathcal{F} et \mathcal{G} . Deux types de combinaisons spatio-temporelle sont étudiées pour \mathcal{G} :

- Discrétisation *upwind* : HEUN2 & U1. Le contour d'amplification de cette combinaison est donné en Fig. 3.1.a, et cette combinaison est stable jusqu'à CFL = 1.
- Discrétisation centrée : RK21 & C2. Le contour d'amplification de cette combinaison est donné en Fig. 3.1.b, et cette combinaison est stable jusqu'à *CFL* = 1.

3.3.2.c Utilisation de la discrétisation *upwind* pour G

Contours d'amplification de PARAREAL. On fixe en premier lieu \mathcal{F} =RKC4-U5, et on examine les contours d'amplification de PARAREAL, utilisés au cours des itérations. Deux tailles de *time-slice* $s = \{1, 8\}$ sont étudiées, et les différents contours d'amplification sont représentés en Fig. 3.11.

On remarque tout d'abord que pour s = 1, le contour d'amplification est modifié de telle manière qu'une instabilité (*i.e.* amplification) apparaît pour des composantes hautes fréquences $(\theta_{\kappa} \sim 0.8\pi)$ à *CFL* = 1, ce qui n'était initialement pas le cas pour le solveur \mathcal{G} seul. Ceci peut être



FIGURE 3.11 – Utilisation de PARAREAL avec des discrétisations spatiales de type upwind pour \mathcal{F} et \mathcal{G} . Contours d'amplification (3.33) de PARAREAL en fonction de s et k, N = 4, $\mathcal{G} = HEUN2-U1$, $\mathcal{F} = RKC4-U5$, $s_{\mathcal{G}} = s_{\mathcal{F}} = s$. Limite de stabilité de \mathcal{G} en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

expliqué par la rétractation du contour de stabilité de PARAREAL observé en Sec. 3.3.1.c, qui a donc pour effet de rajouter une contrainte sur la condition *CFL* de PARAREAL. De plus, on observe au fur et à mesure des itérations un décalage de la composante fréquentielle la plus amplifiée vers les basses fréquences.

Néanmoins, l'augmentation de s permet de réduire cet effet, au détriment d'une diffusion plus importante pour les composantes basses fréquences. L'apport de cette augmentation de s sur la stabilité de PARAREAL est cependant conditionné à l'utilisation d'un schéma spatial *upwind* pour \mathcal{F} . En Fig. 3.12 sont représentés les contours d'amplification de PARAREAL en reprenant la même configuration avec s = 8, mais en choisissant \mathcal{F} =RKC4-U6. On y observe que l'erreur de diffusion est moins importante On peut y observer une modification similaire du contour d'amplification à celle pour s = 1. A l'instar du cas précédent, les composantes amplifiées à *CFL* = 1 restent les mêmes au fur et à mesure des itérations, cette amplification touche une plage de *CFL* augmentant avec k.



FIGURE 3.12 – Utilisation de PARAREAL avec une discrétisation spatiale de type upwind pour \mathcal{G} et centrée pour \mathcal{F} . Contours d'amplification (3.33) de PARAREAL en fonction et $k, N = 4, \mathcal{G} = HEUN2-U1, \mathcal{F} = RKC4-C6,$ $s_{\mathcal{G}} = s_{\mathcal{F}} = 8$. Limite de stabilité de \mathcal{G} en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

Enfin, si on se place à un *CFL* n'étant pas touché par l'instabilité apportée par PARAREAL, on peut observer une diffusion moins importante pour les hautes fréquences, due à l'apport de la discrétisation centrée du solveur fin. Cependant, cela ne traduit pas nécessairement une erreur moins importante pour les basses fréquences, comme on le voit dans le paragraphe suivant. **Observation 3.3.4** — Dans le cadre de l'utilisation de discrétisations spatiales de type upwind pour G, la rétraction du contour de stabilité de PARAREAL durcit la contrainte CFL originalement imposée par G. L'augmentation de la taille des time-slices permet de diminuer l'impact de cette contrainte, au prix d'une diffusion plus importante sur les basses fréquences.

Erreur de PARAREAL. On examine désormais l'erreur de PARAREAL sur chaque composante fréquentielle, considérant un *CFL* (= 0.8) non touché par l'amplification d'une composante fréquentielle due à l'algorithme. On représente en Fig. 3.13 l'erreur E_N^k , définie en (3.34), pour le solveur grossier (k = 0) et les deux premières itérations ($k = \{1, 2\}$). Pour les deux solveurs fins considérés, la convergence de PARAREAL s'effectue des basses fréquences vers les hautes fréquences, comme cela a déjà été remarqué avec des solveurs implicites par [Ruprecht, 2017, Observation 1].



FIGURE 3.13 – Influence du schéma de discrétisation spatiale de \mathcal{F} avec l'utilisation d'un schéma de type upwind pour \mathcal{G} . Erreur de PARAREAL (3.34) en fonction des itérations $k, N = 4, \mathcal{G} = \text{HEUN2-U1}, s_{\mathcal{G}} = s_{\mathcal{F}} = 8$, CFL = 0.8.

Cependant, la convergence de PARAREAL reste ici cantonnée à une plage basse fréquence. Pour \mathcal{F} =RKC4-U5 (Fig. 3.13.a), l'erreur hautes fréquences est négligeable, de par la diffusion apportée par le solveur fin (et grossier) sur ces composantes. Ce n'est plus le cas pour \mathcal{F} =RKC4-C6 (Fig. 3.13.b), et de plus PARAREAL ne corrige pas l'erreur du solveur grossier pour les hautes fréquences, malgré la diffusion moins importante pour ces composantes (*cf.* Fig. 3.12).

Observation 3.3.5 — Dans le cadre de l'utilisation de schémas de discrétisation de type upwind, la convergence de PARAREAL s'effectue des basse fréquences vers les hautes. En particulier, la correction apportée par PARAREAL sur l'erreur du solveur grossier pour les hautes fréquences est négligeable.

3.3.2.d Utilisation d'une discrétisation centrée pour G

Contours d'amplification de PARAREAL. On fixe désormais \mathcal{F} =RKC4-C6, et on examine les contours d'amplification de PARAREAL au cours des itérations. Comme précédemment, deux tailles de *time-slice* $s = \{1, 8\}$ sont étudiées, et les différents contours d'amplification sont représentés en Fig. 3.14.

Deux phénomènes particuliers peuvent y être observés. En premier lieu, on observe comme pour les discrétisations *upwind* une modification du contour d'amplification amplifiant une plage de fréquence non amplifiée initialement par le solveur \mathcal{G} à CFL = 1, ici pour $\theta_{\kappa} \sim 0.4\pi$. La plage de



FIGURE 3.14 – Utilisation de PARAREAL avec des discrétisations spatiales centrées pour \mathcal{F} et \mathcal{G} . Contours d'amplification (3.33) de PARAREAL en fonction de s et k, N = 4, $\mathcal{G} = RK21-C2$, $\mathcal{F} = RKC4-C6$, $s_{\mathcal{G}} = s_{\mathcal{F}} = s$. Limite de stabilité de \mathcal{G} en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

CFL touchée augmente avec les itérations, et l'augmentation de *s* contribue à réduire l'impact de cette modification sur la stabilité de PARAREAL.

Un deuxième aspect qui semble être propre aux discrétisations centrées est cette amplification touchant les hautes fréquences, atteignant une plage de fréquence de plus en plus basse avec la diminution du *CFL*. On observe par ailleurs une disparition/atténuation de cette amplification pour k = 3, selon la valeur de s considérée.

Techniquement, l'utilisation de schémas centrés pour G rend PARAREAL instable pour la plupart des itérations, avec les configurations étudiées ici. D'autres résultats (non présentés ici) montrent que ce comportement est caractéristique de l'utilisation d'une discrétisation spatiale centrée pour G, ce qui est en accord avec les résultats déjà obtenus par [Ruprecht, 2017, Observation 2] avec des méthodes implicites.

Il est cependant difficile d'expliquer avec précision l'origine de cette instabilité. La localisation des valeurs propres sur l'axe imaginaire ne semble pas être une raison totalement satisfaisante, au vu des éléments donnés en Sec. 3.3.1, qui montraient non seulement une intégration stable avec PARA-REAL pour des valeurs propres imaginaires pures, mais aussi un aspect bénéfique de la réduction du pas de temps (*i.e.* du *CFL*) du solveur grossier. Bien que cet aspect n'ait pas été étudié ici en détail, il semblerait que l'erreur de phase entre \mathcal{G} et \mathcal{F} joue un rôle prédominant dans l'origine de cette amplification, au regard de ce qui est déjà mentionné par [Ruprecht, 2017, Observation 3]. De par la complexité des mécanismes mis en jeu au sein de PARAREAL, la compréhension complète des instabilités mentionnées reste donc actuellement un sujet peu maîtrisé, requérant un travail de recherche supplémentaire. En particulier, une extension de l'analyse réalisée jusqu'à présent à l'erreur de phase pourrait éventuellement apporter plus d'informations.

Finalement, on notera que les instabilités discutées ici correspondent à de faibles amplifications des plages de fréquences identifiées, relativement à l'amplification qui résulterait d'un dépassement de la limite *CFL*. Par conséquent, on peut espérer raisonnablement que la prise en compte de l'opérateur de diffusion au sein de l'équation d'advection-diffusion, ainsi qu'un choix de *s* assez large, pourrait permettre d'obtenir avec PARAREAL une solution stable du problème (3.1) (*i.e.* qui garderait des valeurs "physiques"). Cet aspect sera étudié plus en détail avec des applications numériques dans les sections suivantes de ce chapitre.

Observation 3.3.6 — Dans le cadre de l'utilisation de discrétisations spatiales centrées pour \mathcal{G} , PARAREAL induit une contrainte CFL plus contraignante que celle imposée par \mathcal{G} , cette contrainte augmentant avec le nombre d'itérations. On observe en plus une amplification des hautes fréquences, qui a pour particularité de toucher des fréquences de plus en plus basses au fur et à mesure que le CFL diminue. L'amplitude de cette amplification reste faible pour des CFL "opérationnels" (proches de 1), et l'augmentation de la taille des time-slices permet de réduire son impact, plus particulièrement pour les premières itérations.

Erreur de PARAREAL. On examine désormais l'erreur de PARAREAL sur chaque composante fréquentielle, en considérant les valeurs $CFL \in \{0.4, 0.9\}$. L'erreur E_N^k est représentée pour l'initialisation (k = 0) et les premières itérations ($k \in \{1, 2\}$) en Fig. 3.15, respectivement.

Tout comme pour les schémas *upwind*, on observe une convergence de PARAREAL des basses fréquences vers les hautes, exclusivement sur un domaine basse fréquence. La diminution du *CFL* améliore cette convergence, car en effet, le solveur grossier commet une erreur moins importante du fait que l'intervalle temporel d'intégration est plus court. Cependant, on peut observer une importante erreur apparaissant au niveau des hautes fréquences, qui s'amplifie de manière globale pour ces fréquences au fur et à mesure des itérations. En particulier, la diminution du *CFL* contribue à amplifier ce phénomène.



FIGURE 3.15 – Influence du CFL avec des discrétisations spatiales centrées pour \mathcal{F} et \mathcal{G} . Erreur de PARAREAL (3.34) en fonction des itérations k, N = 4, $\mathcal{G} = RK21$ -C2, $\mathcal{F} = RKC4$ -C6, $s_{\mathcal{G}} = s_{\mathcal{F}} = 8$, CFL = 0.8.

Observation 3.3.7 — Dans le cadre de l'utilisation de schémas de discrétisation spatiale centrés, la convergence de PARAREAL s'effectue des basse fréquences vers les hautes. En particulier, la diminution du CFL améliore la convergence pour les basses fréquences, tout en augmentant l'erreur au niveau des hautes fréquences, pour lesquelles PARAREAL ne converge pas.

3.3.2.e Synthèse de l'étude de PARAREAL avec grille spatiale unique

L'application de PARAREAL avec des intégrateurs explicites au problème d'advection linaire (3.4) a été étudiée, en se focalisant plus particulièrement sur l'influence des schémas de discrétisation spatiaux utilisés au sein des différents solveurs.

L'étude a permis de dégager des résultats de stabilité et de convergence, en distinguant l'utilisation de discrétisations spatiales de type *upwind* (Obs. 3.3.4 & Obs. 3.3.5) et centrées (Obs. 3.3.6 & Obs. 3.3.7). En particulier, on a observé pour les deux types de discrétisation un durcissement de la contrainte *CFL* propre au solveur grossier. Pour les discrétisations spatiales centrées, une amplification des hautes fréquences a été mise en évidence, et a pour particularité de toucher une gamme de fréquence plus importante avec la diminution du *CFL*. Les deux effets mentionnés précédemment, qui correspondent à de faibles amplifications de certaines plages fréquentielles, voient leur impact réduit avec l'augmentation de la taille des *time-slices*. Enfin, similairement à ce qui a déjà été observé par [Ruprecht, 2017] avec des solveurs implicites, la convergence de PARAREAL en fonction des itérations s'effectue principalement pour les basses fréquences, en commençant par les plus basses.

La stratégie utilisée pour construire le solveur grossier consistait en une réduction d'ordre des schémas de discrétisation spatio-temporelle, le même maillage étant conservé pour \mathcal{F} et \mathcal{G} . C'est pourquoi l'analyse du grossissement spatial au sein de PARAREAL est effectuée dans la section suivante.

3.3.3 PARAREAL avec grossissement spatial

Dans cette section, on étudie l'application de PARAREAL avec grossissement spatial au problème d'advection pure mono-dimensionnelle (3.1). Le maillage pour le solveur grossier est obtenu par sélection d'un point sur deux du maillage fin, similairement à ce qui a été utilisé en Sec. 3.2.4.

3.3.3.a Validation numérique du modèle théorique

En premier lieu, des simulations numériques de PARAREAL sont effectuées pour évaluer le modèle théorique de la Prop. 3.2.6. Le facteur d'amplification étant désormais dépendant de la solution initiale, on introduit les deux suivantes :

$$u_0(x) = \sum_{i=0}^2 a_i \sin\left(\frac{2N_x \pi x}{d_i}\right), \text{ avec } a_i \in \{10^{-4}, 10^{-2}, 1\}, \ d_i \in \{5, 10, 20\},$$
(3.91)

• GAUSS :
$$u_0(x) = e^{-\frac{(x-1/2)^2}{\sigma^2}}, \text{ avec } \sigma \in \mathbb{R}^*_+. \tag{3.92}$$

On applique PARAREAL avec grossissement spatial au problème (3.4) avec une interpolation linéaire d'ordre 1, en utilisant pour \mathcal{F} et $\widehat{\mathcal{G}}$ la discrétisation spatio-temporelle RKC4-C6. On prend un nombre de *time-slices* N = 4, avec une taille fixée telle que $s_{\widehat{\mathcal{G}}} = 8$.



FIGURE 3.16 – Validation de la modélisation analytique de PARAREAL avec grossissement spatial vis-à-vis des résultats numériques. Spectres de Fourier de la solution initiale et finale obtenues avec PARAREAL, N = 4, $s_{\widehat{G}} = 8$, RKC4-C6, interpolation linéaire, restriction de type injection.

On représente en Fig. 3.16 les spectres de Fourier des solutions finales de PARAREAL, obtenues à la fois avec la formule analytique de Prop. 3.2.6 et numériquement. On observe pour les deux solutions initiales une bonne similarité entre les deux méthodes pour les composantes à basse fréquence ($\theta_{\kappa} \leq 0.5\pi$). De manière générale, le modèle théorique nous permet de bien retranscrire le comportement de PARAREAL avec grossissement spatial, comme le montre une étude plus détaillée donnée en Ap. G.

L'approche analytique permet néanmoins d'obtenir une bonne estimation de l'impact de l'algorithme sur la solution initiale. En particulier, on remarque la génération de composantes hautes fréquences pour la solution SINUS (Fig. 3.16a), inexistantes à l'origine dans la solution initiale, et dont les positions fréquentielles sont symétriques par rapport aux basses fréquences. Cet aspect, qui est lié au changement de grille dans PARAREAL, sera étudié plus en détail en Sec. 3.3.3.c.

La deuxième remarque que l'on peut faire, en regardant Fig. 3.16b, correspond à l'amplification des moyennes et hautes fréquences pour les deux solutions, analytique ou numérique. Ceci s'avère être un effet de l'amplification propre de PARAREAL dans ce contexte de grossissement spatial, et sera étudié plus en détail en Sec. 3.3.3.b.

3.3.3.b Contours d'amplification de PARAREAL avec grossissement spatial

On s'intéresse ici à l'influence de PARAREAL seul, représenté par le facteur d'amplification $g_{\mathcal{P}}^{k,n}(...)$ dans Prop. 3.2.6, et en particulier des différents types de schéma de discrétisation (centré ou *upwind*) utilisés pour les solveurs grossier et fin.



FIGURE 3.17 – Utilisation de discrétisations spatiales centrées pour \mathcal{F} et $\hat{\mathcal{G}}$ au sein de PARAREAL avec grossissement spatial. Contours d'amplification (3.33) de PARAREAL en fonction de $s_{\hat{\mathcal{G}}}$ et k, N = 4, RKC4-C6 pour \mathcal{F} et $\hat{\mathcal{G}}, s_{\mathcal{F}} = 2s_{\hat{\mathcal{G}}}$. Limite de stabilité de $\hat{\mathcal{G}}$ en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

On choisit en premier lieu les mêmes méthodes de discrétisation pour \mathcal{F} et $\hat{\mathcal{G}}$, à savoir RKC4-C6. En Fig. 3.17, on représente les différents contours d'amplification de PARAREAL pour les premières itérations et deux tailles de *time-slices* $s_{\widehat{\mathcal{G}}} \in \{1, 8\}$. On observe en premier lieu une importante amplification touchant l'ensemble du domaine hautes fréquences pour $s_{\widehat{\mathcal{G}}} = 1$, celle-ci étant d'autant plus importante pour les bas *CFL*. L'augmentation du nombre d'itérations atténue partiellement cette amplification, tandis que le fait de prendre une taille de *time-slice* $s_{\widehat{\mathcal{G}}}$ plus grande a un impact bénéfique bien plus important. On remarque cependant qu'une amplification persiste pour les *CFL* les plus bas.

Bien que cette amplification soit similaire à celle propre aux discrétisations spatiales centrées observée en Sec. 3.3.2.d, elle s'avère être provoquée par la forme particulière de g_{TG} , comme on peut le voir en Fig. 3.18. Sur cette figure sont représentés les contours d'amplification de PARAREAL après 2 itérations, en changeant la discrétisation spatiale pour un schéma *upwind*, successivement pour le solveur grossier puis pour le solveur fin. L'amplification à bas *CFL*, bien que moins importante que précédemment, reste présente, même si la discrétisation spatiale utilise exclusivement des schémas *upwind*. Là encore, l'augmentation de $s_{\widehat{G}}$ atténue radicalement l'amplification, sans nécessairement la faire disparaître à bas *CFL*.



FIGURE 3.18 – Influence de l'utilisation de discrétisation spatiale de type upwind sur PARAREAL avec grossissement spatial. Contours d'amplification (3.33) de PARAREAL en fonction de s, avec différentes discrétisations spatiales pour \mathcal{F} , N = 4, $\hat{\mathcal{G}} = RKC4$ -U5, $s_{\mathcal{F}} = 2s_{\hat{\mathcal{G}}}$, k = 2. Limite de stabilité de $\hat{\mathcal{G}}$ en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

Enfin, nous avons considéré jusqu'à présent uniquement l'utilisation de discrétisations spatiales d'ordre élevé pour $\hat{\mathcal{G}}$. En complément on regarde en Fig. 3.19 l'utilisation successive de deux discré-

tisations spatiales d'ordre plus faible. En Fig. 3.19a, on utilise une discrétisation C2 pour $\hat{\mathcal{G}}$, ce qui permet d'augmenter la limite de stabilité *CFL*⁷, de $\hat{\mathcal{G}}$ à 2.83, laissant envisager une meilleure stabilité de PARAREAL. Cependant, cela n'apporte aucun aspect bénéfique, l'amplification à bas *CFL* touchant même une plus large plage de fréquence. En deuxième possibilité on considère l'utilisation de U3 pour $\hat{\mathcal{G}}$, afin d'ajouter plus de diffusion dans les composantes basses fréquences. L'intérêt reste cependant limité, comme on peut le voir en Fig. 3.19b. Par ailleurs, cette utilisation ne permet pas de supprimer l'amplification à bas *CFL*.



FIGURE 3.19 – Influence de la réduction d'ordre pour la discrétisation spatiale de $\hat{\mathcal{G}}$ sur PARAREAL. Contours d'amplification de PARAREAL avec grossissement spatial, N = 4, F = RKC4-C6, $s_{\mathcal{F}} = 2s_{\widehat{\mathcal{G}}} = 16$, k = 2. Limite de stabilité de \mathcal{F} en tirets gris, se référer à la légende de Fig. 3.10 pour plus d'informations quant à l'interprétation.

Observation 3.3.8 – L'utilisation de PARAREAL avec grossissement spatial induit une erreur d'amplification pour les hautes fréquences, celle-ci étant d'autant plus importante que le CFL utilisé est faible. L'utilisation de schémas de type upwind pour \mathcal{G} et/ou \mathcal{F} permet d'atténuer ces amplifications, sans les faire disparaître totalement. L'utilisation de schémas différents pour \mathcal{F} et \mathcal{G} ne semble pas avoir un apport bénéfique notable. Enfin, l'augmentation de la taille des time-slices atténue considérablement les amplifications basse fréquence, qui sont faibles pour des tailles de time-slices suffisamment grandes.

3.3.3.c Impact du grossissement spatial sur la solution de PARAREAL

On étudie maintenant l'impact du changement de grille seul sur la solution de PARAREAL, représenté par les facteurs g_{TG} et g_{IR} dans Prop. 3.2.6. Comme indiqué en (3.66), l'expression de g_{TG} dépend de la solution initiale, en particulier de la répartition de ses composantes fréquentielles entre basses et hautes fréquences. Ainsi on représente ce facteur en Fig. 3.20a pour différentes solutions initiales du type GAUSS, décrit en (3.92), conjointement avec les spectres de Fourier de ces solutions. Le facteur g_{IR} quant à lui ne dépend que de l'ordre de l'interpolation utilisée (la restriction étant l'injection), et est représenté pour différents ordres en Fig. 3.20b.

De manière générale, on observe que g_{TG} agit tout particulièrement sur les composantes hautes fréquences, tandis que des composantes basses fréquences sont peu influencées par ce facteur. En particulier, g_{TG} amplifie les hautes fréquences, et cette amplification est proportionnelle à la différence d'amplitude entre la basse fréquence considérée, et sa "contraposée" haute fréquence, située symétriquement sur l'axe fréquentiel par rapport à $\theta_{\kappa} = \pi/2$. C'est donc ce facteur qui est

^{7.} Cette limite est obtenue par analyse de Von Neumann pour cette combinaison particulière.



(a) Spectres de Fourier des solutions initiales GAUSS (traits pleins) et des facteurs g_{TG} associés (tirets), pour différentes valeurs de σ dans (3.92).



FIGURE 3.20 – Facteurs d'amplification induits par le changement de grille dans PARAREAL avec grossissement spatial.

responsable de l'apparition des composantes à hautes fréquences absentes de la solution initiale que l'on avait déjà observées en Fig. 3.16a. Cette duplication, qui se traduit par une amplification des hautes fréquences, est d'autant plus importante lorsque la différence d'énergie entre hautes et basses fréquences de la solution initiale est grande ($\sigma = 1 \rightarrow 10$). De plus, la plage de hautes fréquences amplifiées est d'autant plus large qu'une plage plus importante de basses fréquences est présente dans la solution initiale ($\sigma = 10 \rightarrow 100$).

Cet effet du changement de grille est cependant compensé par le facteur g_{IR} , qui permet de manière générale de diffuser les hautes fréquences ($\theta_{\kappa} > \pi/2$), en apportant le moins possible de diffusion sur les basses fréquences. Comme indiqué en Fig. 3.20b, cet effet est d'autant plus important que l'ordre d'interpolation est élevé. Dans le cadre d'une interpolation "parfaite" (d'ordre infini⁸), l'interpolation diffuse totalement les basses fréquences, pour conserver parfaitement l'amplitude des composantes à basses fréquences.

Cet impact du changement de grille au sein de PARAREAL permet de tirer deux premières conclusions :

- De par l'utilisation d'une grille grossière pour $\widehat{\mathcal{G}}$, PARAREAL ne peut éviter une amplification des composantes à hautes fréquences de la solution que si le processus d'interpolation diffuse suffisamment ces dernières, *i.e.* l'ordre d'interpolation est assez élevé.
- L'ordre d'interpolation optimal dépend nécessairement du type de solution initiale considérée, et en particulier de la largeur de son spectre de composantes basse fréquence.

En particulier, dans le cadre d'une utilisation de discrétisation centrée, on avait observé dans la section précédente que PARAREAL pouvait aussi amplifier une plage non négligeable de basses fréquences. De ce fait, un compromis avec le choix d'un ordre d'interpolation assez faible permettrait donc à terme de lutter contre ce type d'amplification.

Observation 3.3.9 — Le changement de grille dans PARAREAL induit une amplification des hautes fréquences, qui dépend principalement des composantes basses fréquences de la solution initiale. Cet effet est contre-balancé par l'impact de la méthode d'interpolation, qui atténue les hautes fréquences et conserve les basses fréquences. En particulier, plus l'ordre d'interpolation est élevé, plus cet effet de l'interpolation est marqué.

^{8.} Sur un maillage uniforme mono-dimensionnel, celle-ci peut être assimilée à une interpolation par transformée de Fourier.

3.3.3.d Synthèse de l'étude de PARAREAL avec grossissement spatial

Cette étude a porté sur l'application de PARAREAL avec grossissement spatial, basé sur l'utilisation d'intégrateurs explicites, au problème d'advection linaire (3.4). Une attention particulière a été portée sur l'analyse séparée des différents facteurs influençant la stabilité de PARAREAL, en se focalisant plus particulièrement sur l'influence des schémas de discrétisation spatiaux utilisés au sein des différents solveurs.

En premier lieu, l'étude s'est concentrée sur le facteur d'amplification propre à PARAREAL, et a mis en évidence une amplification basse fréquence augmentant avec la diminution du *CFL* (Obs. 3.3.8). Bien qu'elle soit similaire à celle apportée par les schémas centrés pour PARAREAL sans grossissement spatial (Obs. 3.3.6), elle reste présente malgré l'utilisation de schémas de type *upwind*, si bien qu'elle semble être liée au changement de grille spatiale. Cette amplification est cependant particulièrement atténuée lorsque des tailles suffisamment grandes de *time-slices* sont utilisées.

Enfin, on a mis en évidence une amplification haute fréquence apportée par le grossissement spatial, dépendant exclusivement des composantes fréquentielles de la solution initiale. Cette amplification peut être contrée par l'influence du schéma d'interpolation, qui a pour effet de diffuser les hautes fréquences tout en conservant les basses (Obs. 3.3.9).

3.3.4 Conclusions générales autour de l'analyse de Fourier

Dans cette section, la méthodologie d'analyse de Fourier détaillée en Sec. 3.2 a été appliquée à PARA-REAL avec ou sans grossissement spatial, afin d'étudier sa stabilité et convergence pour des problèmes advectifs simples.

Synthèse. En Sec. 3.3.1, l'étude focalisée sur le problème autonome a mis en évidence l'importance des propriétés de stabilité du solveur grossier utilisé, tout en soulignant que certaines méthodes d'intégration temporelle étaient plus adaptées pour résoudre avec PARAREAL des problèmes à caractère advectif.

Puis en Sec. 3.3.2 on s'est focalisé sur l'application de PARAREAL sans grossissement spatial au problème d'advection linéaire mono-dimensionnel. Cette étape nous a permis d'étudier l'influence des schémas de discrétisation spatiale sur la stabilité et la convergence de PARAREAL, ainsi que de mettre en évidence et caractériser des instabilités propres à l'utilisation de discrétisations spatiales centrées.

Enfin en Sec. 3.3.3 l'étude précédente a été étendue à PARAREAL avec grossissement spatial, pour lequel on a mis en évidence des instabilités, similaires aux précédentes avec des schémas centrés, mais persistante malgré l'utilisation d'autres types de schémas de discrétisation. Aussi, l'étude a aussi mis en évidence une amplification, dépendante de la solution initiale, inhérente à la stratégie de changement de grille⁹. Cette dernière est cependant contrebalancée par le processus d'interpolation, dont l'ordre de précision apparaît avoir une importance primordiale sur la stabilité et la convergence de PARAREAL avec grossissement spatial.

Perspectives. Dans la littérature, les problèmes advectifs ont régulièrement été présentés comme le talon d'Achille de PARAREAL, les instabilités de ce dernier étant principalement montrées du doigt. L'analyse présentée nous a donc permis de comprendre un peu mieux ces instabilités, les caractériser et les évaluer pour des problèmes advectifs simples dans le cadre de l'utilisation de méthodes d'intégration temporelle explicite au sein de PARAREAL. Par ailleurs, l'analyse de Fourier qui a été présentée et appliquée est un outil pouvant être utilisé dans un cadre plus large, non nécessairement restreint à des méthodes explicites.

^{9.} Ce type d'amplification basse fréquence après l'utilisation de l'opérateur d'interpolation est généralement bien connu pour les solveurs de type multi-grille. Ceux-ci utilisent par ailleurs des étapes de lissage après chaque interpolation afin de supprimer cette amplification parasite de la solution au niveau des basses fréquences.

À la question : "PARAREAL avec grossissement spatial peut-il être numériquement stable pour l'advection pure ?", notre réponse est négative. Cependant de nombreux paramètres peuvent être correctement choisis ou optimisés afin de minimiser l'influence de ces instabilités. La taille des *time-slices*, les schémas de discrétisation, l'ordre d'interpolation, ... : tant d'éléments peuvent avoir un impact bénéfique, auxquels on peut ajouter l'apport d'un terme diffusif ou encore l'influence de la nonlinéarité pour des problèmes plus généraux. C'est pourquoi l'on peut garder de bons espoirs dans le candidat PARAREAL avec grossissement spatial, ceux-ci nécessitant cependant d'être corroborés par des applications plus "concrètes" sur les problèmes qui nous intéressent. C'est donc ce que l'on cherche à faire dans la section suivante.

3.4 Variations autour de l'advection pure linéaire

L'analyse de Fourier de PARAREAL développée en Sec. 3.2 nous a permis d'étudier en Sec. 3.3 l'application des formes explicites de PARAREAL, avec et sans grossissement spatial sur le problème d'advection pure. Nous avons pu en tirer de premières conclusions permettant de comprendre et caractériser la stabilité ainsi que la convergence de PARAREAL au niveau des "composantes fréquentielles" d'une solution initiale. Dans la continuité de l'analyse précédente, cette section tend à effectuer une analyse plus "pratique" de la convergence de PARAREAL, que l'on introduit plus en détails en Sec. 3.4.1. En particulier, l'objectif est d'élargir l'étude précédente au problème d'advection avec faible diffusion (3.1) en Sec. 3.4.2, et à sa forme non-linéaire (équation de Burgers avec terme visqueux) en Sec. 3.4.3.

Les contributions principales de cette section sont représentées par les différentes observations données tout au long de l'étude, qui sont synthétisées en Sec. 3.4.4.a.

3.4.1 Motivations et mise en place de l'étude

Choix des méthodes numériques. On se focalise ici sur la forme explicite de PARAREAL avec grossissement spatial (*cf.* Sec. 3.2.4), cette dernière étant la forme que l'on veut appliquer à Hybrid pour la simulation d'écoulements turbulents. Une interpolation d'ordre variable (1, 3 et 7, voir Ap. C) est utilisée, et on choisit de garder la restriction de type injection tout au long de cette section ¹⁰. Comme précédemment, on utilise N = 4 *time-slices* pour PARAREAL.

Afin de se rapprocher du type de discrétisation utilisée par le solveur HYBRID, on utilise au sein de \mathcal{F} une discrétisation spatiale centrée d'ordre 6 pour les termes d'advection et de diffusion, et l'intégration temporelle est réalisée avec le schéma RKC4. L'étude linéaire ayant mis en évidence que l'utilisation d'une discrétisation différente pour $\widehat{\mathcal{G}}$ n'apportait pas de bénéfice important à la convergence de PARAREAL, on choisit donc d'utiliser les mêmes méthodes de discrétisation pour \mathcal{F} et $\widehat{\mathcal{G}}$.

Critère d'erreur pour l'analyse de convergence, et paramètres étudiés. Plusieurs types de critères sont utilisés dans la littérature pour évaluer la convergence de PARAREAL pour la résolution d'un problème physique. L'utilisation d'un critère portant sur la différence des champs entre deux itérations de PARAREAL (defect) a souvent été préférée [Samaddar et al., 2010, Ruprecht et Krause, 2012, Ruprecht et al., 2016], car pouvant être calculée à faible coût en même temps que l'application de PARAREAL. A l'opposé de cette approche "opérationnelle", la convergence peut aussi être observée en regardant l'erreur vis-à-vis de la solution fine. Bien que cette méthode nécessite le calcul de cette dernière, elle reste largement préférée pour examiner la convergence de PARAREAL dans un contexte d'étude (*e.g.* [Fischer *et al.*, 2005, Reynolds-Barredo *et al.*, 2012, Kreienbuehl *et al.*, 2015c, Eghbal *et al.*, 2017] entre autres ...).

^{10.} Ce choix de garder l'injection comme type de restriction est motivé principalement par des tests à plus grande échelle qui sont détaillés dans le chapitre suivant.

On choisit donc d'évaluer la convergence en regardant l'erreur relative entre la solution fine et celle de PARAREAL à l'itération k pour le temps final T, que l'on définit ainsi :

$$E_{T,L_2}^k = \frac{\left\| U_{\mathcal{P}}^k(T) - U_{\mathcal{F}}(T) \right\|_2}{\left\| U_{\mathcal{F}}(T) \right\|_2},\tag{3.93}$$

avec $||U||_2$ la norme L_2 (dite Euclidienne) du vecteur $U \in \mathbb{R}^p$. Cette erreur est extraite pour $k \in \{0, 1, 2, 3\}$. Pour k = 0, on obtient l'erreur relative du solveur grossier (après interpolation) par rapport au solveur fin. Pour k = 4 (non représentée), cette erreur est nulle de par les propriétés de PARAREAL, qui permet de retrouver la solution fine dès lors que k > N.

Dans les sous-sections suivantes, on étudie l'influence des paramètres suivants sur la convergence de PARAREAL (*i.e.* l'évolution de E_{T,L_2}^k):

- importance du terme advectif : ce dernier a souvent été montré du doigt dans la littérature, comme un facteur dégradant la convergence de PARAREAL lorsque son importance augmente [Croce et al., 2014, Steiner et al., 2015]. Son influence dans un cadre explicite est donc un facteur primordial que l'on examinera en se basant sur la définition d'un nombre de Reynolds particulier au problème considéré.
- taille des time-slices : celle-ci a une importance cruciale, tant sur le plan de la convergence (e.g. [Fischer et al., 2005, Sec 5.2] et [Eghbal et al., 2017, Sec.4]), et la stabilité (cf. Sec. 3.3) de PARAREAL. C'est pourquoi, on cherche à examiner ici son impact sur la convergence, en relation avec les premiers résultats de stabilité obtenus en Sec. 3.3.2 et Sec. 3.3.3.
- 3. régularité de la solution : en Sec. 3.3.3.c, on a vu que la convergence de PARAREAL est influencée par la solution initiale, et plus particulièrement son contenu fréquentiel. L'étude de ce paramètre, que l'on a choisi de faire varier en prenant une solution initiale de type GAUSS avec différents coefficients σ dans (3.92), nécessite donc une attention particulière.

Ces trois paramètres sont étudiés séparément pour les deux problèmes considérés, et une synthèse globale des différentes observations est donnée en Sec. 3.4.4.a. Leur influence est étudiée en considérant systématiquement les deux aspects suivants :

- (A) **l'ordre d'interpolation** : comme observé en Sec. 3.3.3.c, ce dernier a un impact majeur sur PARAREAL avec grossissement spatial. On considère donc pour l'ensemble des tests trois ordres d'interpolation o_I : un ordre "faible" ($o_I = 1$), "moyen" ($o_I = 3$) et "élevé" ($o_I = 7$). Sans nécessairement vouloir déterminer une approche optimale, l'étude cherche principalement à observer l'impact de l'ordre d'interpolation utilisé dans le but de tirer des conclusions qualitatives sur la convergence de PARAREAL, dans la continuité du travail réalisé par [Ruprecht, 2014].
- (B) la résolution spatiale : à notre connaissance, très peu d'études ont porté sur cet aspect à tel point que son influence sur la convergence de PARAREAL reste encore peu comprise. Or la résolution spatiale a une importance majeure dans la mise en place du solveur \mathcal{F} , surtout dans le cadre de simulation numérique directe d'écoulements turbulents. Celle de $\hat{\mathcal{G}}$ étant pilotée par celle de \mathcal{F} dans le cadre considéré, la convergence de PARAREAL en est nécessairement influencée. Cette étude cherche donc à donner un premier aperçu de son influence en considérant deux niveaux de résolution ("haut" & "bas") pour chaque paramètre étudié.

3.4.2 Equation d'advection-diffusion

Configuration du problème. On résout ici numériquement le problème (3.1) sur un maillage périodique uniforme, contenant N_x points ($N_x \in \{100, 500\}$). Sauf spécification contraire, on utilise les paramètres suivants pour l'ensemble des simulations :

- $\nu = 0.0005 \text{ dans}$ (3.1),
- $s_{\widehat{\mathcal{G}}} = 8$ pas de temps par *time-slices* pour $\widehat{\mathcal{G}}$,

• $\sigma = 8$ pour la solution initiale de type GAUSS en (3.92).

Pour simplifier l'étude, on fixe $s_{\mathcal{F}} = 2s_{\widehat{\mathcal{G}}}$ et CFL = 1. En Fig. 3.21 sont représentées les différentes solutions du problème considéré.



FIGURE 3.21 – Solutions initiales et finales pour l'équation d'advection-diffusion avec le solveur \mathcal{F} , $N_x = 500$, CFL = 1, avec δ_t le pas de temps du solveur fin.

Influence du terme advectif. On utilise ici la définition simplifiée du nombre de Reynolds $Re = c/\nu$ associée au problème (3.1). La solution finale obtenue avec le solveur fin est représentée en Fig. 3.21b ($T = 64\delta_t$). On représente en Fig. 3.22 l'évolution de la convergence de PARAREAL pour $Re \in \{2000, 10000, 20000\}$.



(a) Haute résolution spatiale, $N_x = 500$.

(b) Basse résolution spatiale, $N_x = 100$.

FIGURE 3.22 – Influence du terme advectif sur la convergence de PARAREAL avec grossissement spatial pour l'équation d'advection-diffusion, avec différents ordres d'interpolation. E_{T,L_2}^k en fonction de k, traits pleins : Re = 2000, tirets : Re = 10000, pointillés : Re = 20000.

En premier lieu, on observe en Fig. 3.22a que pour chaque Re considéré, la convergence est majoritairement influencée par l'ordre d'interpolation utilisé, et ce plus particulièrement pour la première itération. Pour k = 0, on peut observer le ratio entre l'erreur induite par le solveur grossier et l'erreur d'interpolation. Cette dernière domine principalement pour $o_I = 1$, tandis qu'elle apparaît négligeable pour $o_I \in \{3, 7\}$. Cependant, bien que les niveaux d'erreur à k = 0 soient similaires pour ces deux cas, une réelle convergence de la première itération n'est obtenue que pour $o_I = 7$. Il est intéressant de noter que cet apport de l'ordre d'interpolation élevé concerne surtout les premières itérations de PARAREAL, les niveaux d'erreur étant équivalents pour les dernières itérations avec $o_I \in \{3, 7\}$.

L'augmentation du Re induit effectivement une dégradation de la convergence, mais ce uniquement dans les situations où l'algorithme converge ($k \in \{1, 2, 3\}$ pour $o_I = 7$ et $k \in \{2, 3\}$ pour $o_I \in \{1, 3\}$). Comme on peut le voir en Fig. 3.22b, cet effet est moins marqué à basse résolution spatiale, en particulier pour la première itération. Il peut être noté qu'à haute résolution spatiale et Re élevé, la convergence présente un plateau pour $o_I = 7$ et $k \ge 1$, ce qui indique que dans cette situation la deuxième et troisième itérations n'apportent aucun bénéfice à la précision de PARAREAL.

Enfin, on peut observer qu'à basse résolution spatiale, l'augmentation du Re a un effet moins important que précédemment, la convergence étant principalement pilotée par l'ordre d'interpolation. De plus, on observe une convergence pour $o_I = 3$ dès la première itération, ce qui n'était pas le cas à haute résolution spatiale.

Observation 3.4.1 — Pour le problème d'advection avec faible diffusion, la convergence de PARA-REAL avec grossissement spatial est majoritairement dominée par l'ordre d'interpolation utilisé, dont l'augmentation améliore grandement la convergence. L'importance du terme advectif impacte légèrement la convergence, cet impact étant minimisé par la réduction de la résolution spatiale.

Influence de la taille des time-slices. On considère ici plusieurs valeurs pour le nombre de pas de temps par time-slices $s_{\hat{G}} = \{8, 16, 32\}$. Afin de comparer E_{T,L_2}^k au même temps final, l'algorithme PARAREAL est redémarré avec $N_r = \{4, 2, 1\}$ respectivement. En Fig. 3.21b est représentée la solution finale obtenue avec le solveur fin ($T = 256\delta_t$). On représente en Fig. 3.23 la convergence de PARAREAL pour les différentes tailles de time-slices considérées.



(a) Haute résolution spatiale, $N_x = 500$.

(b) Basse résolution spatiale, $N_x = 100$.

FIGURE 3.23 – Influence de la taille des time-slices sur la convergence de PARAREAL avec grossissement spatial pour l'équation d'advection-diffusion, avec différents ordres d'interpolation. E_{T,L_2}^k en fonction de k, traits pleins : $s_{\widehat{G}} = 32$, tirets : $s_{\widehat{G}} = 16$, pointillés : $s_{\widehat{G}} = 8$.

On observe pour les deux niveaux de résolution spatiale que l'influence de $s_{\widehat{\mathcal{G}}}$ n'est pas la même selon l'ordre d'interpolation considéré. Pour $o_I \in \{1,3\}$, la diminution de $s_{\widehat{\mathcal{G}}}$ entraîne une dégradation de la convergence. Cependant la tendance est inversée pour $o_I = 7$, plus particulièrement à basse résolution spatiale (Fig 3.23b). De plus, on peut voir en Fig. 3.23a que $s_{\widehat{\mathcal{G}}} = 16$ correspond à un optimum pour la convergence à k = 1 entre les trois valeurs examinées.

Ce phénomène peut être expliqué par l'amplification de PARAREAL observé pour les hautes fréquences en Sec. 3.3.3 pour des petites tailles de *time-slices*. Bien que la convergence à basse fréquence (qui concerne l'ensemble des composantes fréquentielles dominantes de la solution) soit améliorée en prenant une valeur faible de $s_{\hat{G}}$, l'amplification des hautes fréquences agit comme facteur dégradant pour la convergence de PARAREAL. L'utilisation d'une plus grande valeur pour $s_{\hat{G}}$ permet donc de réduire cette amplification, et améliorer ainsi la convergence. Cependant, pour un schéma d'interpolation d'ordre élevé, la diffusion apportée sur les hautes fréquences étant bien plus importante, la convergence de PARAREAL est alors dominée par celle des basses fréquences, favorisée par une minimisation de $s_{\hat{G}}$.

On peut donc postuler l'existence d'un ordre d'interpolation minimal, à partir duquel la diminution de $s_{\widehat{G}}$ bénéficie à la convergence. Or, bien que le bénéfice d'une augmentation de $s_{\widehat{G}}$ soit clair pour $o_I = 7$ et $N_x = 100$, il l'est beaucoup moins pour $N_x = 500$ avec l'interpolation du même ordre. Ainsi, cet ordre d'interpolation "minimal" permettant de profiter d'une diminution de $s_{\widehat{G}}$ semble aussi dépendre de la résolution spatiale utilisée.

Enfin, on conclura que les variations de convergence induites par le changement de la taille des *time-slices* restent de manière générale faibles, et ce plus particulièrement à haute résolution spatiale.

Observation 3.4.2 — Pour le problème d'advection avec faible diffusion, la convergence de PARA-REAL avec grossissement spatial est faiblement dépendante de la taille des time-slices utilisée. En particulier, l'utilisation de plus petites time-slices dégrade la convergence pour des ordres d'interpolation faibles, et l'améliore pour des ordres plus élevés.

Influence de la régularité de la solution. Afin d'obtenir une solution initiale de moins en moins régulière, on utilise dans (3.92) différentes valeurs pour $\sigma \in \{8, 16, 32\}$. Les solutions initiales correspondantes, représentées en Fig. 3.21a, présentent des gradients de solution de plus en plus élevés au fur et à mesure que σ augmente, et de ce fait un spectre énergétique de plus en plus large à basse fréquence (*cf.* Fig. 3.20a). La convergence de PARAREAL pour ces différentes solutions initiales est représentée en Fig. 3.24.



(a) Haute résolution spatiale, $N_x = 500$.

(b) Basse résolution spatiale, $N_x = 100$.

FIGURE 3.24 – Influence de la régularité de la solution sur la convergence de PARAREAL avec grossissement spatial pour l'équation d'advection-diffusion, avec différents ordres d'interpolation. E_{T,L_2}^k en fonction de k, traits pleins : $\sigma = 8$, tirets : $\sigma = 16$, pointillés : $\sigma = 32$.

La première remarque que l'on peut faire concerne l'augmentation de l'erreur relative du solveur grossier par rapport au solveur (k = 0) quand σ augmente. Ainsi, plus la résolution spatiale du solveur fin est "mauvaise" du point de vue de la solution initiale considérée, plus l'écart des solutions entre solveur fin et grossier sera important. Cependant, on observe en Fig. 3.24a que l'augmentation de σ joue globalement peu sur le taux de convergence de PARAREAL.

Enfin, l'utilisation d'une basse résolution spatiale (Fig. 3.24b) montre que l'influence de l'ordre d'interpolation est d'autant moins importante dès lors que la résolution spatiale ainsi que la régularité de la solution initiale diminuent. Ce comportement laisse à penser qu'il existe un ordre d'interpolation maximal après lequel une augmentation de l'ordre n'apporte plus rien à la convergence de PARAREAL. Au vu des résultats présentés, cet ordre d'interpolation maximal serait principalement déterminé par la résolution spatiale ainsi que la solution initiale considérée.

Observation 3.4.3 — Pour le problème d'advection avec faible diffusion, la régularité de la solution a un impact important sur l'erreur commise par le solveur grossier, tandis qu'elle influe très légèrement sur la convergence de PARAREAL avec grossissement spatial, lorsque la résolution spatiale est suffisamment importante. Pour des solutions de moins en moins régulières, et lorsque la résolution spatiale diminue, le bénéfice de l'augmentation de l'ordre d'interpolation est amoindri.

3.4.3 Equation de Burgers avec terme visqueux

Configuration du problème. On considère maintenant l'équation de Burgers avec un terme de diffusion (dénommée Burgers visqueux) :

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}, \quad u(x,0) = u_0(x), \quad x \in [0,1], \quad t \in [0,T],$$
(3.94)

dont (3.1) peut être vu comme la forme linéarisée. On choisit le coefficient de diffusion ν tel que $\nu \ll \max_x(u_0)$. La même discrétisation spatiale que pour le problème linéaire est utilisée, pour obtenir le système semi-discrétisé suivant :

$$\frac{dU}{dt} = \frac{1}{2}A U^2 + Z U, \quad U(0) = U_0 \in \mathbb{R}^p, \quad t \in [0, T],$$
(3.95)

avec $A, Z \in \mathcal{M}_p(\mathbb{R})$ définies de la même manière que pour (3.2), et $U^2 \in \mathbb{R}^p$ le vecteur des éléments de U élevés terme à terme au carré.

On résout numériquement le problème (3.94) sur un maillage périodique uniforme, contenant N_x points ($N_x \in \{250, 500\}$). Sauf spécification contraire, on utilise les paramètres suivants pour l'ensemble des simulations :

- $\nu = 0.0005$ dans (3.94),
- $s_{\widehat{G}} = 8$ pas de temps par *time-slices* pour $\widehat{\mathcal{G}}$,
- $\sigma = 8$ pour la solution initiale de type GAUSS en (3.92).

Comme pour Sec. 3.4.2, on fixe $s_{\mathcal{F}} = 2s_{\widehat{\mathcal{G}}}$ et $CFL = \max_x(u_0)\delta_t/\delta_x = 1$. En Fig. 3.25 sont représentées les différentes solutions du problème considéré.

Influence du terme advectif. On utilise ici la définition simplifiée du nombre de Reynolds $Re = \max_x(u_0)/\nu$ associée au problème (3.94). La solution finale obtenue avec le solveur fin est représentée en Fig. 3.25b ($T = 64\delta_t$). En Fig. 3.26 est représentée l'évolution de la convergence de PARAREAL pour $Re \in \{2000, 10000, 20000\}$.

On remarque tout d'abord que l'augmentation de Re induit une augmentation de l'erreur du solveur grossier par rapport au fin (k = 0), ce qui ne s'observait pas pour le problème linéaire. A partir de ce niveau d'erreur initial, le taux de convergence reste globalement inchangé à haute résolution, ce dernier étant principalement piloté par l'ordre d'interpolation. On retrouve par ailleurs une convergence améliorée au fur et à mesure que o_I augmente.

Le passage en basse résolution spatiale réduit non seulement le taux de convergence, mais aussi l'impact de l'ordre d'interpolation à chaque Re considéré. Ainsi, l'augmentation de o_I n'a un intérêt bénéfique sur la convergence que pour les dernières itérations de PARAREAL. On peut lier ce comportement à celui observé pour le problème linéaire pour des solutions de moins en moins régulières (*i.e.* pour lesquelles la résolution spatiale est de moins en moins adaptée pour représenter correctement la solution).



FIGURE 3.25 – Solutions initiales et finales pour l'équation de Burgers visqueux avec le solveur \mathcal{F} , $N_x = 500$, CFL = 1, avec δ_t le pas de temps du solveur fin.



FIGURE 3.26 – Influence du terme advectif sur la convergence de PARAREAL avec grossissement spatial pour l'équation de Burgers visqueux, avec différents ordres d'interpolation. $E_{TL_2}^k$ en fonction de k, traits pleins : Re = 2000, tirets : Re = 10000, pointillés : Re = 20000.

Observation 3.4.4 — Pour l'équation de Burgers avec faible diffusion, le taux de convergence de PARAREAL avec grossissement spatial est piloté par l'ordre d'interpolation utilisé, dont l'augmentation améliore la convergence, cet effet étant beaucoup moins marqué à basse résolution spatiale. L'importance du terme advectif a un impact principalement sur l'erreur du solveur grossier, et semble faiblement influer sur la convergence de PARAREAL à haute résolution spatiale, et bien plus à basse résolution.

Influence de la taille des time-slices. On utilise ici plusieurs valeurs pour le nombre de pas de temps par time-slices $s_{\hat{G}} = \{16, 32, 64\}$. Afin de comparer E_{T,L_2}^k au même temps final, l'algorithme PARAREAL est redémarré avec $N_r = \{4, 2, 1\}$ respectivement. En Fig. 3.25b est représentée la solution finale obtenue avec le solveur fin $(T = 512\delta_t)$. On montre en Fig. 3.27 la convergence de PARAREAL pour les différentes tailles de time-slices considérées.

Comme pour le problème linéaire, on peut observer une amélioration de la convergence au fur et à mesure que la taille des *time-slices* augmente. Cependant, contrairement au cas linéaire, on n'observe pas d'inversion de cette tendance pour o_I élevé, et ce quelle que soit la résolution spatiale



(a) Haute résolution spatiale, $N_x = 500$.

(b) Basse résolution spatiale, $N_x = 250$.

FIGURE 3.27 – Influence de la taille des time-slices sur la convergence de PARAREAL avec grossissement spatial pour l'équation de Burgers visqueux, avec différents ordres d'interpolation. E_{T,L_2}^k en fonction de k, traits pleins : $s_{\mathcal{G}} = 64$, tirets : $s_{\mathcal{G}} = 32$, pointillés : $s_{\mathcal{G}} = 16$.

utilisée.

Ici encore, l'utilisation d'une basse résolution spatiale réduit considérablement l'apport d'un ordre d'interpolation élevé. Enfin, on peut souligner que contrairement au cas linéaire, l'augmentation de la taille des *time-slices* a un impact bien plus marqué.

Observation 3.4.5 — Pour l'équation de Burgers avec faible diffusion, la taille des time-slices a un impact important sur la convergence, et ce plus particulièrement pour des ordres d'interpolation élevés. En particulier, l'augmentation de la taille des time-slices améliore la convergence, et ce parfois bien plus que l'augmentation de l'ordre d'interpolation.

Influence de la régularité de la solution. Comme pour Sec. 3.4.2, on utilise dans (3.92) différentes valeurs pour $\sigma \in \{8, 16, 32\}$. Les solutions initiales correspondantes sont représentées en Fig. 3.25a. La convergence de PARAREAL pour ces différentes solutions initiales est représentée en Fig. 3.28.





(b) Basse résolution spatiale, $N_x = 250$.



Comme pour le problème linéaire, la convergence se dégrade pour une solution initiale de moins en moins régulière. On observe déjà à haute résolution spatiale que plus σ est élevé (*i.e.* moins la solution est régulière), moins l'augmentation de l'ordre d'interpolation n'a d'impact. Cet effet est d'autant plus marqué lorsque la résolution spatiale est réduite.

Il est intéressant de noter que l'on retrouve un type de convergence similaire selon σ à basse résolution spatiale que pour des variations de Re en Fig. 3.26b. Ceci peut s'expliquer par le fait que la solution finale du problème de Burgers visqueux est de moins en moins régulière au fur et à mesure que le terme de diffusion devient négligeable. De ce fait, jouer sur la régularité de la solution finale en agissant sur le terme de diffusion a un effet similaire à celui observé ici en jouant sur la régularité de la solution initiale.

Observation 3.4.6 — Pour l'équation de Burgers avec faible diffusion, la diminution de la régularité de la solution entraîne non seulement une diminution du bénéfice apporté par des ordres d'interpolation plus élevés, mais a aussi un impact dégradant sur la convergence de PARAREAL.

3.4.4 Conclusions générales sur l'étude de convergence

3.4.4.a Synthèse des résultats

Dans cette section, une étude de convergence de PARAREAL avec grossissement spatial a été effectuée, en se basant sur l'erreur relative de la solution finale de PARAREAL par rapport à la solution fine. Nous avons considéré successivement les équations mono-dimensionnelles d'advection-diffusion et celle de Burgers avec terme visqueux, soumises à une faible diffusion. L'influence de plusieurs paramètres a été étudiée, en relation avec l'ordre d'interpolation utilisé pour le changement de grille au sein de PARAREAL ainsi que la résolution spatiale utilisée pour le solveur fin.

Influence du terme advectif. Dans la littérature, il a été observé à plusieurs reprises ([Croce *et al.*, 2014, Steiner *et al.*, 2015] entre autres) que la convergence se détériorait lorsque le ratio entre les forces advectives et diffusives augmentait. Ici, l'effet est plus contrasté : pour le problème linéaire, on observe bien cette détérioration de la convergence lorsque le terme advectif prédomine, cependant celle-ci reste majoritairement dominée par l'ordre d'interpolation utilisé (Obs. 3.4.1). De plus, cette détérioration est de moins en moins visible lorsque la résolution spatiale est réduite. En revanche pour le problème non-linéaire, on observe que le niveau de diffusion choisi va plutôt fixer l'erreur initiale du solveur grossier par rapport au fin, tandis que la convergence est majoritairement pilotée par l'ordre d'interpolation choisi (Obs. 3.4.4).

Influence de la taille des time-slices. De manière générale pour une même solution initiale, l'erreur du solveur grossier par rapport au fin augmente avec la taille des *time-slices*. Il paraît donc naturel de s'attendre à ce que PARAREAL converge mieux lorsque la taille de ces dernières est minimisée. Cependant, le comportement de PARAREAL est bien plus complexe. Pour le problème linéaire, la réduction de la taille des *time-slices* n'est bénéfique à la convergence que si l'ordre d'interpolation est supérieur à une certaine valeur, dépendante de la résolution spatiale utilisée (Obs. 3.4.2). Ceci n'est cependant pas observé pour le problème non-linéaire, où l'augmentation de la taille des *time-slices* améliore la convergence de manière générale (Obs. 3.4.5).

Influence de la régularité de la solution. On a observé dans le cadre linéaire que pour des solutions initiales de moins en moins régulières, l'erreur du solveur grossier par rapport au fin augmentait, ce qui influait sur le niveau global de l'erreur de PARAREAL (Obs. 3.4.3). Le taux de convergence apparaît être principalement piloté par l'interpolation, lorsque la résolution spatiale est suffisante. Pour le problème non-linéaire, on retrouve ce comportement avec les solutions initiales les plus régulières, qui s'atténue très vite dès que la résolution spatiale ou la régularité de la solution initiale diminuent (Obs. 3.4.6). De manière générale, le lien entre la régularité de la solution et la résolution spatiale utilisée par le solveur fin a un rôle primordial au sein de la convergence de PARAREAL.

3.4.4.b Perspectives pour des problèmes de plus grandes échelles

Cette première étude de PARAREAL avec grossissement spatial nous a donc permis d'obtenir de premières conclusions sur la convergence de l'algorithme pour des problèmes avec composante advective forte, et représentant une simplification des équations de Navier-Stokes que l'on veut résoudre numériquement par la suite. Au vu de ce qui avait été souligné en conclusion du Chap. 2, on peut dès à présent tirer quelques conclusions et perspectives globales.

Influence négative du terme advectif, mais ... Bien que l'impact d'un terme advectif important puisse dégrader la convergence de PARAREAL, d'autres paramètres peuvent avoir une bien plus grande influence, parfois bénéfique, sur la convergence. En particulier, l'ordre d'interpolation utilisé pour le changement de grille, la taille des *time-slices*, ainsi que la résolution spatiale semble être des facteurs critiques pour l'optimisation de la convergence de PARAREAL pour un problème donné. Il apparaît donc nécessaire de développer une méthodologie permettant d'optimiser la convergence de PARAREAL en choisissant au mieux ces différents paramètres, et ce dans un cadre plus général.

Convergence possible, même en présence de solutions peu régulières. Le caractère advectif du problème et l'utilisation de discrétisations spatiales centrées sont deux critères qui semblent "dangereux" pour la stabilité de PARAREAL. Ainsi, l'étude effectuée en Sec. 3.3 avait donc déjà cherché à caractériser ces instabilités, afin de déterminer les facteurs en étant à l'origine et de trouver des stratégies permettant de les prendre en compte. Dans la continuité de cette démarche, l'étude présentée ici a donc montré que pour des paramètres numériques bien choisis (méthode d'intégration temporelle et d'interpolation, taille de *time-slices*, ...), il est tout à fait possible d'obtenir des erreurs de parallélisation très faibles, et ce même pour des problèmes linéaires et des solutions présentant un fort degré de non-régularité (*cf.* Fig. 3.25b et Fig. 3.27). Ces observations nous procurent donc une certaine confiance quant à l'application de PARAREAL avec grossissement spatial pour des problèmes turbulents de plus grande échelle, application qui sera examinée plus en détail dans le chapitre suivant.

Besoin de développer des critères d'erreur plus spécifiques. Enfin, dans le cadre d'une application pratique, il apparaît nécessaire de développer des critères de convergence spécifiques à PARAREAL avec grossissement spatial. En particulier, l'influence de la condition initiale considérée et la résolution spatiale doivent être prises en compte. En première approche, on peut envisager l'élaboration d'un critère basé sur les différences, calculées avec des normes discrètes, entre solutions fine et grossière (disponibles pour chaque *time-slice* à la fin de la première itération PARAREAL) et l'erreur effectuée par le changement de grille (qui peut être obtenue par des étapes successives de restriction et d'interpolation d'une même solution fine).

3.5 Synthèse

Dans ce chapitre, une étude préliminaire a été menée sur des problèmes advectifs simples, afin de caractériser les difficultés de convergence de l'algorithme pour ce type de problèmes, déjà connues dans la littérature. En particulier, cette étude s'est focalisée sur l'utilisation de méthodes d'intégration temporelle explicites au sein de PARAREAL, ainsi que sur l'influence du changement de grille sur la convergence de PARAREAL avec grossissement spatial. Cet aspect de l'algorithme étant peu exploré dans la littérature, ce travail propose donc une contribution pour l'étude de PARAREAL dans un cadre explicite. En Sec. 3.2, nous avons introduit une méthodologie pour l'analyse de Fourier, permettant d'étudier l'influence des schémas d'intégration temporelle, des discrétisations spatiales, et des opérateurs de changement de grille utilisés. En particulier, la méthodologie permettant l'étude du changement de grille au sein de PARAREAL avec grossissement spatial constitue une des contributions principales de ce travail.

Puis, l'étude de Fourier a été appliquée pour des problèmes advectifs simples à complexité croissante en Sec. 3.3. En particulier, l'étude a mis en évidence l'importance du choix de la méthode d'intégration temporelle utilisée pour le solveur grossier, ainsi que les instabilités pouvant être apportées par l'utilisation de schémas de discrétisation spatiale centrés et le changement de grille au sein de PARAREAL avec grossissement spatial. Différents paramètres ayant des vertus curatives (taille des sous-domaines temporelles, ordre d'interpolation) ont cependant été identifiés.

Enfin, une étude de convergence portant sur des problèmes plus généraux (advection-diffusion, équation de Burgers) a été menée en Sec. 3.4 dans le but de compléter l'analyse de Fourier précédente. La contribution majeure de cette étude a été d'illustrer qu'une convergence de PARAREAL avec grossissement spatial était possible pour des problèmes où l'influence des forces diffusives restait très faible vis-à-vis de celle des forces advectives. En effet, l'étude a aussi montré les impacts, parfois très bénéfiques, que pouvaient avoir différents paramètres sur la convergence de PARAREAL avec grossissement spatial (ordre d'interpolation, résolution spatiale, régularité de la solution, taille des sous-domaines temporels, ...).

CHAPITRE 4

Analyse de Parareal appliqué aux simulations d'écoulements tridimensionnels turbulents

Table des matières

4.1	Analyse de performance				
	4.1.1	Modèle de scalabilité			
	4.1.2	Modèle de communication			
4.2	Introduction à l'analyse de convergence				
	4.2.1	Equations de Navier-Stokes compressible			
	4.2.2	Challenges pour l'intégration temporelle			
	4.2.3	Choix des problèmes tests 112			
4.3	4.3 Décroissance d'une Turbulence Homogène Isotrope				
	4.3.1	Description du problème			
	4.3.2	Définition des solveurs fin et grossier pour Parareal			
	4.3.3	Analyse de convergence de PARAREAL pour la décroissance de Turbulence			
		Homogène Isotrope			
4.4	Écoulement de Canal Turbulent 125				
	4.4.1	Description du problème			
	4.4.2	Définition du solveur fin et grossier pour Parareal			
	4.4.3	Analyse de convergence de PARAREAL pour l'Écoulement de Canal Turbulent 131			
4.5	Synth	èse			

Ce chapitre cherche à effectuer une évaluation du gain de performance d'une part, ainsi qu'une analyse d'erreur d'autre part pour l'application de PARAREAL avec grossissement spatial, dans un contexte de simulation numérique directe d'écoulements turbulents, réalisée avec le solveur CFD compressible HYBRID. Le choix de se restreindre à des DNS sera discuté plus en détail en Sec. 4.2. L'analyse présentée ici constitue l'une des contributions principales de ce travail. Elle est décomposée en deux grandes parties.

Dans un premier temps, la Sec. 4.1 présente l'analyse de performance pour l'utilisation de PARAREAL avec grossissement spatial, en apport de la parallélisation spatiale déjà développée et optimisée au sein d'HYBRID. On établit un modèle de performance de l'algorithme prenant en compte le speedup obtenu par décomposition spatiale du problème, et les coûts de communication et de transfert entre niveaux de grille. Ce modèle, généralisable à d'autres types de problème et/ou solveur, est appliqué à HYBRID dans plusieurs configurations d'étude.

Dans un second temps, on analyse la convergence de PARAREAL avec grossissement spatial pour deux situations canoniques d'écoulements turbulents. L'étude de convergence est ensuite détaillée pour la décroissance d'une turbulence homogène isotrope en Sec. 4.3 et pour un écoulement de canal turbulent en Sec. 4.4.

En particulier, le travail en Sec. 4.1.1.b et Sec. 4.3 correspond à celui effectué pour la soumission (acceptée) de [Lunet et al., 2017a].

Contexte de l'étude. Comme il a été mentionné précédemment en Chap. 1 et Chap. 2, les communications induites par une parallélisation temporelle basée sur PARAREAL sont peu fréquentes, à l'opposé de celles engendrées par la décomposition spatiale. C'est pourquoi cette étude a été menée en simulant l'application de PARAREAL (*i.e.* en effectuant les calculs liés à chaque *time-slice* séquentiellement). Les solutions des solveurs fin et grossier en revanche sont obtenues par une utilisation d'HYBRID avec la parallélisation spatiale dont le solveur est doté¹. Cette approche de "simulation" de l'algorithme PARAREAL est par ailleurs plus cohérente avec l'objectif de ce travail, qui porte plus sur l'étude de stratégies de parallélisation spatio-temporelle, que sur leur développement et mise en production.

4.1 Analyse de performance

On modélise ici les performances de la parallélisation espace-temps à partir de tests de scalabilité du solveur CFD, ainsi que d'une mesure des temps de communication induits par la décomposition temporelle, tout en prenant en compte les performances de la décomposition de domaine issue de la parallélisation spatiale. Cette modélisation permet d'évaluer avec une précision raisonnable le gain pouvant être obtenu avec PARAREAL, sans nécessairement procéder à une implémentation complète de l'algorithme au sein du solveur considéré. Cette méthodologie, généralisable aux solveurs du même type, est détaillée et appliquée à HYBRID en Sec. 4.1.1.

Enfin, on présente en Sec. 4.1.2 une approche qui cherche à évaluer les volumes de communication entre processus en évaluant la quantité de données transférées au sein du système global. Un modèle dit "de communication" est ainsi développé dans ce but, et appliqué à HYBRID. Cette modélisation nous permet d'examiner les transferts de données effectués pour différentes stratégies de parallélisation espacetemps, dans l'optique d'obtenir une première évaluation énergétique des coûts de communication de données.

Description du système de calcul utilisé. L'ensemble des tests de performance a été réalisé sur le supercalculateur EOS de CALMIP² (Toulouse, France). Cette plate-forme est équipée de 612 nœuds de calcul, chacun composé de deux processeurs INTEL IVY BRIDGE de fréquence d'horloge 2.8 GHz, avec 10 cœurs par processeurs. Chaque nœud dispose de 64 GB de mémoire RAM, et l'ensemble est interconnecté au travers d'un réseau non-bloquant de type *fat tree*, avec une bande passante de 6.98 Gbytes/s. Le code HYBRID a été compilé par le biais du compilateur INTEL 14.0.2.144, avec la bibliothèque INTEL MPI 4.1.3.049.

4.1.1 Modèle de scalabilité

4.1.1.a Développement du modèle

Speedup de PARAREAL dans le cas général. En premier lieu, on cherche à modéliser le *speedup* apporté par PARAREAL indépendamment de la parallélisation spatiale. Comme souligné par [Arteaga *et al.*, 2015, Aubanel, 2011, Berry *et al.*, 2012, Ruprecht, 2015], l'algorithme est assez flexible pour s'associer à différents modèles de programmation. On choisit ici de se restreindre à un modèle de programmation à mémoire distribuée (le lecteur est invité à consulter [Ruprecht, 2015] pour une discussion incluant d'autres types de stratégie). On note N_{proc} le nombre de processus

^{1.} À cet effet, une bibliothèque Рүтном a été développée, pour manipuler les entrées et sorties d'Нувпо et simuler PARAREAL. Le solveur utilisé devient donc un paramètre d'entrée de l'algorithme.

^{2.} https://www.calmip.univ-toulouse.fr/spip.php?article388.

total associés à la parallélisation espace-temps, et N le nombre de *time-slices* considérés. Dans cette configuration, chaque *time-slice* et son sous-domaine spatial associé sont alloués à un processus MPI.

On se focalise tout d'abord sur l'algorithme PARAREAL original, et on note $C_{\mathcal{F}}$ le coût d'intégration sur une *time-slice* en utilisant $\mathcal{F}_{t_{n-1}\to t_n}^{\delta_t}$, et $C_{\mathcal{G}}$ le coût correspondant pour $\mathcal{G}_{t_{n-1}\to t_n}^{\Delta_t}$. Du fait de l'utilisation exclusive de schémas d'intégration temporelle explicites pour les deux solveurs, on peut supposer que $C_{\mathcal{F}}$ et $C_{\mathcal{G}}$ sont directement proportionnels à la taille des *time-slices*. Ainsi on écrit :

$$C_{\mathcal{F}} = T_{\mathcal{F}} N_{\delta_t, \mathcal{F}}, \quad C_{\mathcal{G}} = T_{\mathcal{G}} N_{\Delta_t, \mathcal{G}}, \tag{4.1}$$

avec $T_{\mathcal{F}}$ le temps de calcul requis pour une intégration temporelle fine sur un pas de temps et $N_{\delta_t,\mathcal{F}}$ le nombre de pas de temps effectué (des notations similaires sont utilisées pour $C_{\mathcal{G}}$).

Comme recommandé par [Ruprecht, 2015], on se focalise sur une implémentation optimisée de PARAREAL, discutée à la fois par [Minion, 2010, Sec. 5] et [Aubanel, 2011, Sec. 4], en utilisant une configuration de type *pipeline* (Fig. 4.1) pour permettre un recouvrement partiel des calculs. On désigne par OP (*Original Pipeline*) cette implémentation. En particulier, le coût d'une propagation grossière pour une itération dans (2.10) est réduit de $NC_{\mathcal{G}}$ à $C_{\mathcal{G}}$. Dans cette configuration, l'estimation du *speedup* théorique de PARAREAL utilisant K itérations est donnée par [Minion, 2010] :

$$\mathcal{S}(N) = \frac{NC_{\mathcal{F}}}{NC_{\mathcal{G}} + K(C_{\mathcal{F}} + C_{\mathcal{G}})} = \frac{1}{(1 + \frac{K}{N})\frac{C_{\mathcal{G}}}{C_{\mathcal{F}}} + \frac{K}{N}}.$$
(4.2)

A partir de (4.2), il est aussi possible de retrouver la borne de l'efficacité de PARAREAL en fonction de son nombre d'itérations :

$$\mathcal{E}(N) = \frac{\mathcal{S}(N)}{N} = \frac{1}{(N+K)\frac{C_{\mathcal{G}}}{C_{\mathcal{F}}} + K} \le \frac{1}{K}.$$
(4.3)

Prise en compte du temps de communications entre time-slices. La projection du speedup (4.2) a été obtenue en négligeant le temps nécessaire aux communications entre time-slices, dénommées par la suite "communications temporelles". Afin de compléter ce modèle de performance, on choisit d'inclure ce coût dans l'analyse, après l'introduction d'une légère modification de la forme originale de type pipeline de PARAREAL, qui permet de réduire le coût induit par les communications temporelles. Sur la Fig. 4.1.a est représenté le diagramme d'exécution de la version originale. Comme l'étape de prédiction (2.9) est effectuée de manière séquentielle, N - 1 communications entre time-slices sont requises durant cette phase (rectangles fins bariolés à gauche dans Fig. 4.1a). Afin d'éviter ce décalage, on considère l'implémentation représentée en Fig. 4.1b, pour laquelle la solution initiale à l'étape de prédiction est calculée indépendamment par chaque processus. Cette modification supprime ainsi des communications dispensables entre time-slices au prix d'une redondance des calculs. Ce type d'optimisation peut engendrer des économies de communication d'autant plus importantes que N est important. On désigne par MP (*Modified Pipeline*) ce type d'implémentation.

On note alors $C_{\mathcal{T}}^t$ le coût de communication de l'ensemble des champs de la solution globale (fine) d'une *time-slice* vers une autre. Le coût total des communications temporelles pour K itérations de PARAREAL est donné par :

$$C_{\mathcal{T},K} = C_{\mathcal{T}}^{t} \sum_{k=1}^{K} (N-k) = C_{\mathcal{T}}^{t} \frac{K(2N-K-1)}{2}.$$
(4.4)

Ainsi, le speedup de l'implémentation de type pipeline présentée en Fig. 4.1.b pour PARAREAL s'écrit :

$$\mathcal{S}_T(N) = \frac{1}{(1+\frac{K}{N})\frac{C_{\mathcal{G}}}{C_{\mathcal{F}}} + \frac{K}{N} + \frac{C_{\mathcal{T},K}}{N C_{\mathcal{F}}}}.$$
(4.5)



FIGURE 4.1 – Diagramme d'exécution de PARAREAL de type pipeline (a) et de type pipeline modifié (b). Trois itérations de PARAREAL (K = 3) sur quatre time-slices (N = 4) sont représentées. La fin de chaque itération de PARAREAL est indiquée par des tirets. Les time-slices sont numérotées de 0 à 3, chaque ligne correspondant à un processus de PARAREAL.

Prise en compte du passage sur grille spatiale grossière. On note maintenant $C_{\widehat{\mathcal{G}}}$ le coût d'intégration sur une *time-slice* du solveur $\widehat{\mathcal{G}}^{\Delta_t}$, basé sur l'utilisation d'une grille spatiale grossière. Une adaptation directe de (4.5) pour \widehat{K} itérations de PARAREAL avec grossissement spatial permet d'obtenir la projection suivante pour le *speedup* :

$$\widehat{\mathcal{S}}_{T}(N) = \frac{1}{(1+\frac{\widehat{K}}{N})\frac{C_{\widehat{\mathcal{G}}} + C_{\widehat{\mathcal{R}}} + C_{\widehat{\mathcal{I}}}}{C_{\mathcal{F}}} + \frac{\widehat{K}}{N} + \frac{C_{\mathcal{T},\widehat{K}}}{N C_{\mathcal{F}}}},$$
(4.6)

où $C_{\widehat{\mathcal{R}}}(C_{\widehat{\mathcal{I}}})$ représentent le coût de l'opérateur de restriction (interpolation) du vecteur solution sur chaque sous-domaine spatial. On peut affirmer que le coût de communication temporelle peut être négligé si :

$$\frac{C_{\mathcal{T},\widehat{K}}}{C_{\mathcal{F}}} \ll (N+\widehat{K})\frac{C_{\widehat{\mathcal{G}}}+C_{\widehat{\mathcal{R}}}+C_{\widehat{\mathcal{I}}}}{C_{\mathcal{F}}} + \widehat{K}.$$
(4.7)

Ce point sera discuté plus en détail en Sec. 4.1.1.c.

Speedup de la parallélisation spatio-temporelle. La combinaison des développements précédents permet de déduire le *speedup* de la parallélisation espace-temps :

$$\mathcal{S}_{S,T}(N_{proc}) = \mathcal{S}_S(N_s) \,\widehat{\mathcal{S}}_T(N), \tag{4.8}$$

avec $S_S(N_s)$ le *speedup* de la parallélisation spatiale du solveur fin \mathcal{F} sur N_s processus, *i.e.* :

$$\mathcal{S}_S(N_s) = \frac{T^s_{\mathcal{F},serial}}{T^s_{\mathcal{F},N_s}}.$$
(4.9)

La parallélisation espace-temps apporte donc un gain effectif de *speedup* par rapport à la parallélisation spatiale exclusive si la condition suivante est satisfaite :

$$\mathcal{S}_{S,T}(N_{proc}) > \mathcal{S}_S(N_{proc}). \tag{4.10}$$

4.1.1.b Estimation du gain de parallélisation spatio-temporelle pour HYBRID

Mise en place de l'étude. On cherche ici à obtenir une première estimation du gain parallèle apporté par PARAREAL avec grossissement spatial. On choisit pour simplifier l'analyse de résoudre

avec le solveur fin \mathcal{F} un problème périodique sur un maillage 3D uniforme, contenant un même nombre de points N_L dans chaque direction de l'espace. Par ailleurs, on considère un problème de taille modérée $N_L = 80$, ce afin d'évaluer l'efficacité parallèle de la décomposition spatiale en limite de scalabilité forte (*cf.* Déf. 1.1.2 en Chap. 1).

Le maillage utilisé pour le solveur $\hat{\mathcal{G}}$ est construit à partir du maillage fin, en ne gardant qu'un point sur deux dans chaque direction. Cette stratégie de grossissement, aussi appelée *vertex coarsening* (voir [Trottenberg *et al.*, 2001]), est identique à celle utilisée pour les problèmes 1D étudiés en Chap. 3. Afin de garder une même condition *CFL* pour \mathcal{F} et $\hat{\mathcal{G}}$, on choisit de prendre :

$$\Delta_t = 2\delta_t,\tag{4.11}$$

ce qui permet d'évaluer le ratio théorique du coût entre les deux solveurs à $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}} = 16$.

Évaluation des performance parallèles. Dans un premier temps, on choisit de négliger les coûts de calcul associés aux opérateurs de transfert ($C_{\widehat{\mathcal{R}}}$ et $C_{\widehat{\mathcal{I}}}$) et aux communications temporelles ($C_{T,\widehat{K}}$) dans (4.6). Ces coûts seront pris en compte en Sec. 4.1.1.c. La scalabilité obtenue pour \mathcal{F} et $\widehat{\mathcal{G}}$ est représentée en Fig. 4.2.



FIGURE 4.2 – Performances parallèles mesurées pour Hybrid (solveurs fin et grossier), et modélisées pour PARAREAL avec grossissement spatial pour $\widehat{K} \in \{1, 2\}$ itérations, $N_L = 80$. Les résultats pour $C_{\mathcal{F}}/C_{\widehat{G}} = \infty$ ne sont représentés qu'en Fig. 4.2b.

Le solveur fin montre de bonnes propriétés de scalabilité jusqu'à l'utilisation de 160 processus. Passé cette limite, la performance sature en raison du faible nombre de points par cœur considéré (1600 pour $N_{proc} = 320$). La détérioration des performances est d'autant plus forte pour le solveur grossier $\hat{\mathcal{G}}$, le nombre de points par cœurs considéré (200 pour $N_{proc} = 320$) étant encore plus faible. Il est important de noter que cette décroissance précoce va nécessairement détériorer la valeur du ratio $C_{\mathcal{F}}/C_{\widehat{G}}$ par rapport à sa valeur théorique.

On fixe le nombre de processeurs alloués à la parallélisation spatiale à $N_{proc}/N = 160$, et on considère une parallélisation temporelle utilisant $N = \{2, 4, 8, 16\}$ time-slices. La modélisation du speedup et de l'efficacité parallèle de PARAREAL après $\widehat{K} \in \{1, 2\}$ itérations est donnée respectivement en Fig. 4.2a et Fig. 4.2b respectivement.

Limites inhérentes à PARAREAL. Ces limites sont matérialisées sur la Fig. 4.2b, en considérant un coût du solveur grossier négligeable par rapport à celui du solveur fin, *i.e.* $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}} = \infty$. Dans ces conditions, le nombre de *time-slices* utilisées n'a pas d'influence, et l'utilisation d'une seule itération

permet de "prolonger" les bénéfices déjà obtenus par la parallélisation spatiale exclusive. Cependant, ce succès est limité dès la deuxième itération de PARAREAL. Pour $N_{proc} = 320$ et $N_{proc} = 640$ notamment, le *speedup* "idéal" de la parallélisation espace-temps est inférieur à celui obtenu par parallélisation spatiale seule. Cet effet résulte directement de (4.3), et montre l'importance d'une minimisation du nombre d'itérations au sein de PARAREAL, pour obtenir une efficacité parallèle élevée.

On peut néanmoins mettre ces résultats en relief dans le cas où bien plus de processus seraient à disposition (e.g. $N_{proc} \sim 10000$). Au vu des propriétés de scalabilité forte d'Hybrid, la parallélisation spatiale sur cette quantité de processus aurait une efficacité parallèle négative, ou serait tout simplement incompatible avec la décomposition (ce qui est le cas pour $\hat{\mathcal{G}}$ avec $N_L = 40$ et $N_{proc} = 1280$, d'où l'absence de donnée pour ce point de fonctionnement). De ce fait, un nombre important de *time-slices* devrait être utilisé, et la parallélisation espace-temps serait profitable même avec l'utilisation que l'efficacité parallèle de PARAREAL reste positive (accélération effective des calculs). Cette dernière condition, qui porterait à la fois sur \hat{K} et N, est par ailleurs en grande partie fixée par le ratio $C_{\mathcal{F}}/C_{\hat{\mathcal{G}}}$, responsable de la décroissance de l'efficacité parallèle avec l'augmentation du nombre de *time-slices*, comme on le voit par la suite.

Prise en compte du coût du solveur grossier. On examine maintenant la scalabilité de la parallélisation spatio-temporelle, en considérant à la fois le ratio théorique $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}} = 16$, et le ratio effectif $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}} = 9.4$, mesuré lors des tests effectués. Sans surprise, on observe une détérioration de l'efficacité parallèle, assez conséquente dès la première itération. Il est aussi intéressant de noter que l'écart entre l'itération $\widehat{K} = 1$ et $\widehat{K} = 2$ réduit avec l'augmentation des *time-slices* N. En effet, la part de temps de calcul associée à l'étape de propagation grossière (séquentielle) dans l'algorithme PARAREAL augmente avec N. Son coût est cependant moins important lorsque \widehat{K} augmente, relativement au coût de l'itération (deux propagations grossières et une propagation fine pour $\widehat{K} = 1$, trois propagations grossières et deux propagations fines pour $\widehat{K} = 2$).

On peut donc conclure que pour le problème considéré, la parallélisation spatio-temporelle a les meilleurs chances de "prolonger" la scalabilité spatiale si l'on se restreint à $\widehat{K} = 1$ ou $\widehat{K} = 2$. Cependant, au vu des niveaux d'efficacité relativement bas pour le cas $\widehat{K} = 2$, on cherchera à privilégier, lorsque possible, l'utilisation d'une seule itération dans le reste de l'étude.

Point de départ de la parallélisation temporelle. On étudie désormais l'influence sur le speedup de la parallélisation combinée, du nombre de processus utilisé pour la parallélisation spatiale. A cet effet on introduit le gain relatif de la parallélisation spatio-temporelle comme étant :

$$\sigma_{S,T} = \frac{\mathcal{S}_{S,T}(N_{proc}) - \mathcal{S}_{S}(N_{proc})}{\mathcal{S}_{S}(N_{proc})}.$$
(4.12)

Une valeur positive de $\sigma_{S,T}$ indique une parallélisation espace-temps favorable dans la situation étudiée. On note $\sigma_{S,T}^{\star}$ le gain relatif lorsque sont négligés les coûts des communications temporelles et des opérateurs de transfert. La valeur idéale de $\sigma_{S,T}^{\star}$ est obtenue pour des tailles de *time-slice* suffisamment grandes, *i.e.* $C_{\widehat{\mathcal{R}}}$, $C_{\widehat{\mathcal{I}}}$ et $C_{\mathcal{T},K}$ sont négligeables devant $C_{\mathcal{F}}$ et $C_{\widehat{\mathcal{G}}}$, la valeur de ces deux derniers étant proportionnelle à la taille des *time-slices* (4.1).

On représente en Tab. 4.1 les valeurs de $\sigma_{S,T}^*$ en fonction de N_{proc}/N , pour $\widehat{K} = 1$. On remarque qu'à nombre de sous-domaines spatiaux fixé, l'augmentation de N améliore le gain $\sigma_{S,T}^*$, conformément au comportement observé en Fig. 4.2. On peut également observer que la parallélisation spatio-temporelle est d'autant plus intéressante à considérer que le nombre de processus total N_{proc} est important.

Détermination d'un nombre de *time-slices* **préférentiel pour l'étude.** Comme observé précédemment, le gain apporté par PARAREAL avec grossissement spatial est majoritairement dicté

TABLE 4.1 – Gain relatif de la parallélisation spatio-temporelle $\sigma_{S,T}^{\star}$ après une itération de PARAREAL avec grossissement spatial (en négligeant $C_{\widehat{\mathcal{R}}}$ et $C_{\widehat{\mathcal{I}}}$). N_{proc} désigne le nombre total de processus utilisé, la parallélisation spatiale étant effectuée sur N_{proc}/N sous-domaines. Pour chaque valeur de N_{proc}/N , on indique les ratios $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}}$ correspondant. On considère respectivement $N \in \{2, 4, 8, 16\}$ time-slices pour chaque diagonale du tableau.

N _{proc}	160	320	640	1280	$C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}}$
$N_{proc}/N = 80$	-18%	-4%	+10%	+49%	11.6
$N_{proc}/N = 160$	-	+1%	+ 24 %	+82%	9.4
$N_{proc}/N = 320$	-	_	-1.1%	+54%	6.8

par le nombre d'itérations \widehat{K} , qui doit être minimisé. Cette minimisation est conditionnée par la convergence de l'algorithme, qui est analysée en Sec. 4.3 et Sec. 4.4 pour des cas d'applications avec HYBRID.

Néanmoins, au vu du problème considéré (résolution des équations de Navier-Stokes), une convergence en un faible nombre d'itérations sera nécessairement associée à un nombre réduit de *time-slices*. En effet, l'aspect chaotique de la turbulence, rend plus difficile la convergence de PARAREAL [Wang *et al.*, 2013]. On reviendra plus en détails sur ce point en Sec. 4.2.

Au vu des résultats présentés en Tab. 4.1, on choisit de focaliser notre étude sur la valeur N = 4. Les valeurs des gains obtenus pour l'analyse précédente sont indiqués en gras dans le Tab. 4.1. Ce type de stratégie (augmentation de N_{proc} tout en fixant N) a par ailleurs déjà été utilisé avec un certain succès par [Croce *et al.*, 2014], pour la parallélisation spatio-temporelle avec PARAREAL d'un solveur CFD 2D pour des écoulements de fluide incompressible.

On peut souligner qu'à ce stade, l'analyse de performance effectuée jusqu'à présent n'écarte en aucun cas l'intérêt d'utiliser un nombre de *time-slices* supérieur. Le choix de se restreindre principalement à N = 4 a été effectué dans le but de simplifier l'ensemble des études de convergence présentées dans ce manuscrit (Sec. 3.3, Sec. 3.4, Sec. 4.3 et Sec. 4.3), tout en maintenant une certaine uniformité et cohérence.

4.1.1.c Prise en compte du temps de communication et d'interpolation

On inclut ici dans l'analyse le coût des communications temporelles (C_{τ}^t) , et également celui des opérateurs de transfert $(C_{\hat{\tau}}$ et $C_{\hat{R}})$, en se basant sur la formule de *speedup* (4.6). On choisit ici de se limiter à une restriction de type injection, ainsi qu'une interpolation d'ordre 7³.

Le coût lié à l'opérateur de restriction peut être négligé, car aucune opération flottante n'est requise, *i.e.* $C_{\hat{\mathcal{R}}} = 0$. De plus, il est possible de relier le coût de l'opérateur d'interpolation à celui d'un pas de temps du solveur grossier, ce qui permet d'écrire :

$$C_{\widehat{\mathcal{I}}} \le \frac{15}{39} \times \frac{15}{11} \frac{C_{\widehat{\mathcal{G}}}}{N_{\Delta t,\widehat{\mathcal{G}}}}.$$
(4.13)

Le détail des calculs est donné en Ap. C. Par la suite, on considère la borne supérieure dans (4.13) comme une estimation raisonnable de $C_{\hat{\tau}}$.

Afin d'estimer le coût (en temps de calcul) d'une communication temporelle $C_{\mathcal{T}}^t$ dans PARAREAL, on adopte la stratégie de mesure suivante. En premier lieu, on recrée une décomposition spatiale identique à celle réalisée dans HYBRID avec N_{proc}/N processus. On exécute ensuite un échange standard par communications MPI de l'ensemble des variables physiques d'importance d'une *time-slice* à une autre (au total, seules 2 *time-slices* sont requises), en mesurant le temps de communication

^{3.} Ce choix de se restreindre à ce type de restriction est motivé par les résultats de convergence présentés en Sec. 4.3. L'ordre d'interpolation correspond à la valeur maximale considérée pour l'analyse de convergence en Sec. 4.3 et Sec. 4.4.

sur le système EOS. Cette procédure est répétée vingt fois et C_T^t est choisi comme le temps de communication maximal obtenu sur l'ensemble des tests. On réfère le lecteur au code source C++ ⁴ pour plus de détails.

Le gain relatif de la parallélisation espace-temps $\sigma_{S,T}$ (cf. (4.12)) est ainsi donné en Tab. 4.2. On

TABLE 4.2 – Influence de $C_{\mathcal{T},1}$ et $C_{\widehat{\mathcal{I}}}$ sur le gain relatif de la parallélisation espace-temps $\sigma_{S,T}$ après une itération de PARAREAL avec grossissement spatial ($\widehat{K} = 1$).

Le cas (a) inclut uniquement $C_{\mathcal{T},1}$, tandis que le cas (b) considère à la fois $C_{\mathcal{T},1}$ et $C_{\widehat{\mathcal{I}}}$ dans (4.12). (N = 4, injection et interpolation d'ordre 7 comme opérateurs de transfert).

$N_{proc} \mid \mathbf{Cas} \mid \sigma_{S,T}, N_{\Delta_t,\widehat{\mathcal{G}}} = 20 \mid \sigma_{S,T}, N_{\Delta_t,\widehat{\mathcal{G}}} = 100 \mid \sigma_{S,T}^{\star}$								
640	(a)	18%	20%	21%				
	(b)	17%	20%	21%				
1280	(a)	43%	47%	48%				
	(b)	41%	46%	48%				

peut y voir que les gains relatifs $\sigma_{S,T}$ ont une valeur relativement proche de la valeur idéale $\sigma_{S,T}^{\star}$ obtenue en Sec. 4.1.1.b, et ce même pour des tailles de *time-slice* relativement petites ($N_{\Delta_t,\widehat{\mathcal{G}}} = 20$). En particulier, l'augmentation de ce dernier paramètre permet d'obtenir très rapidement une valeur du gain relatif presque égale à la valeur idéale associée ($N_{\Delta_t,\widehat{\mathcal{G}}} = 100$). Finalement, la comparaison des lignes (a) et (b) en Tab. 4.2 révèle que l'influence de $C_{\widehat{\mathcal{T}}}$ est marginale.

Au vu des observations précédentes, on choisit donc de négliger les coûts de communication et d'interpolation pour la suite de l'étude.

4.1.1.d Généralisation de l'analyse de performance

Afin de considérer l'application de PARAREAL avec grossissement spatial aux cas plus complexes qui seront étudiés en Sec. 4.3 et Sec. 4.4, il est intéressant d'étendre l'analyse des performances de la parallélisation espace-temps dans un contexte de décomposition spatiale plus général. En particulier, on s'intéresse ici à des problèmes de plus grande taille, et pour lesquels une décomposition spatiale uniforme n'est pas nécessairement réalisable.

Problèmes de plus grande taille avec décomposition spatiale uniforme. On s'intéresse tout d'abord au même type de problème qu'en Sec. 4.1.1.b (3D périodique uniforme), avec des tailles de maillage de plus en plus grande ($N_L \in \{160, 320\}$). Ces valeurs pour N_L seront par ailleurs considérées dans l'analyse de convergence en Sec. 4.3.

Afin de comparer les différentes efficacités parallèles, on utilise un temps de calcul normalisé, défini ainsi pour l'intégration temporelle des équations sur un pas de temps :

$$T_w(N_{proc}, p) = \frac{N_{proc} T_{N_{proc}}}{p}$$
(4.14)

avec $T_{N_{proc}}$ le temps de calcul avec parallélisation spatiale et p le nombre de degrés de liberté considérés ($p = 5N_L^3$ ici). Cette définition, inspirée de [Bermejo-Moreno *et al.*, 2013a], permet de normaliser les efficacités parallèles par une même valeur :

$$\mathcal{E}'(N_{proc}) = \mathcal{E}(N_{proc}) \frac{T_w(N_{proc,\min}, p_{\max})}{T_w(N_{proc}, p)}$$
(4.15)

avec ici $p_{\rm max}=5\times 320^3$ et $N_{proc,{\rm min}}=20.$

^{4.} Disponible sous le lien https://gitlab.com/tlunet/parallel-in-time.

On représente en Fig. 4.3a les efficacités parallèles obtenues pour $N_L \in \{80, 160, 320\}$, en utilisant jusqu'à ${}^5 N_{proc} = 1280 = 20 \times 2^6$. Afin d'étendre l'analyse faite en Sec. 4.1.1.b à des plus grandes valeurs pour N_{proc} , les résultats pour $N_{proc} > 1280$ sont extrapolés à partir de ceux obtenus pour des maillages de plus petite taille.



FIGURE 4.3 – Performances parallèles pour des problèmes de grande taille avec décomposition spatiale homogène. N_L indiqué en légendes pour le maillage du solveur fin \mathcal{F} . Traits pleins : parallélisation spatiale seule. Tirets-points : données extrapolées à partir des résultats pour les maillages plus grossiers. Pointillés : modélisation de l'efficacité de la parallélisation spatio temporelle basée sur PARAREAL avec grossissement spatial pour $N \in \{2, 4, 8, 16\}$ time-slices et $\widehat{K} = 1$ itération.

Comme observé en Sec. 4.1.1.b, l'efficacité de la parallélisation spatio-temporelle est étroitement liée au ratio $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}}$, qui décroît avec l'augmentation du nombre de processeurs dédié à la parallélisation spatiale. On le représente pour $N_L = \{80, 160, 320\}$ en Fig. 4.3b. On observe une évolution similaire pour chaque taille de problème. Bien que ce ratio soit faible $(C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}} < 8)$ pour les couples $(N_L, N_{proc}/20) \in \{(80, 2^4), (160, 2^6), (320, 2^8)\}$, on observe un gain en efficacité induit par l'utilisation de PARAREAL avec grossissement spatial pour $N_L \in \{160, 320\}$, similaire à celui observé pour $N_L = 80$. Même si ces modélisations ont été obtenues à partir de données extrapolées, elles nous confirment que l'étude précédente effectuée pour $N_L = 80$ est généralisable à des tailles de problèmes plus importantes.

Problèmes de plus grande taille avec décomposition spatiale non-uniforme. Finalement, on considère un problème périodique dans les directions x et z, avec des conditions limites non périodiques dans la direction y. En particulier, en notant N_{L_x} , N_{L_y} et N_{L_z} le nombre de points de maillage associé à chaque direction, on fixe pour $\mathcal{F}: (N_{L_x}, N_{L_y}, N_{L_z}) = (420, 255, 420)$; et pour $\hat{\mathcal{G}}: (N_{L_x}, N_{L_y}, N_{L_z}) = (210, 127, 210)$. Ce type de problème sera par ailleurs utilisé dans l'analyse de convergence en Sec. 4.4.

Comme précédemment, on normalise l'efficacité parallèle, mais en utilisant ici $p_{\text{max}} = 420 \times 255 \times 420$. On représente en Fig. 4.4a l'efficacité parallèle des solveurs fin et grossier, en extrapolant les données pour $N_{proc} > 1240$. La valeur obtenue pour le ratio $C_{\mathcal{F}}/C_{\widehat{G}}$ est représentée en Fig. 4.4b.

On observe que pour ce type de décomposition spatiale, l'efficacité de la parallélisation spatiale seule chute plus rapidement que pour les configurations considérées précédemment. Par ailleurs, le ratio de coût entre le solveur fin et grossier reste globalement élevé, ce qui explique le gain potentiellement important avec une unique itération de PARAREAL (*cf.* Fig. 4.4a), et ce dès l'utilisation de deux *time-slices*.

^{5.} Ce choix est contraint par le nombre maximum de cœurs pouvant être utilisés pour un même calcul au sein du système EOS $N_{proc,max} = 1800$.




(b) Ratio de coût de calcul entre \mathcal{F} et $\widehat{\mathcal{G}}$.

FIGURE 4.4 – Performances parallèles pour un problème de grande taille avec décomposition spatiale non-uniforme. $(N_{L_x}, N_{L_y}, N_{L_z})$ indiqués en légendes pour le maillage du solveur fin \mathcal{F} . Traits pleins : parallélisation spatiale seule. Tirets-points : données extrapolées à partir des résultats pour les maillages plus grossiers. Pointillés : modélisation de l'efficacité de la parallélisation spatio temporelle basée sur PARAREAL avec grossissement spatial pour $N \in \{2, 4, 8, 16\}$ time-slices et $\widehat{K} = 1$.

4.1.1.e Conclusions concernant l'analyse de scalabilité

L'analyse effectuée en Sec. 4.1.1.b a permis d'effectuer une première analyse du gain d'une parallélisation spatio-temporelle basée sur PARAREAL avec grossissement spatial. Il a été observé que ce gain est principalement contraint par :

- le nombre d'itérations \widehat{K} , dont l'augmentation entraîne une dégradation importante des performances parallèles,
- le ratio $C_{\mathcal{F}}/C_{\widehat{\mathcal{G}}}$, qui doit être maintenu à un niveau élevé pour garantir un niveau de scalabilité suffisant lorsque le nombre de *time-slices* N augmente.

Entre autres, les choix effectués dans ce contexte (grossissement spatial) pour le solveur grossier induisent une décroissance de ce ratio avec l'augmentation du nombre de processus dédiés à la parallélisation spatiale.

Par ailleurs, on a montré que le temps consacré à la communication et au changement de grille ne dégrade que très légèrement les performances de la parallélisation espace-temps. De plus, ces pertes de temps peuvent être rendues négligeables par l'augmentation de la taille des *time-slices*.

Enfin, l'extension de l'étude à des problèmes de plus grande taille laisse entrevoir des gains d'efficacités parallèles similaires pour la parallélisation espace-temps. En particulier, on peut espérer des gains d'efficacité plus importants, comparés à une parallélisation spatiale exclusive, dans le cas de problèmes n'autorisant pas une décomposition spatiale uniforme.

4.1.2 Modèle de communication

Comme il a été mentionné en Chap. 1, les prochaines architectures de calcul massivement parallèles seront caractérisées par une consommation d'énergie dominée par celle liée aux communications entre processus. De ce fait, le coût énergétique des communications est un facteur important que l'on peut modéliser. Bien qu'il soit actuellement difficile de l'évaluer dans un cas général car très dépendant de l'architecture utilisée, on peut cependant commencer par estimer le volume de données transférées au cours des calculs sur l'ensemble du super-calculateur. C'est l'objet du modèle développé en Sec. 4.1.2.a qui permet d'évaluer cette quantité de données transférées, pour différentes stratégies de parallélisation spatio-temporelles basées sur PARAREAL avec grossissement spatial. Ce modèle est ensuite appliqué au solveur HYBRID en Sec. 4.1.2.b, et une discussion quant à sa généralisation est menée en Sec. 4.1.2.c.

4.1.2.a Développement du modèle

Afin de simplifier la modélisation, on se place dans la configuration du problème considéré en Sec. 4.1.1.b, *i.e.* 3D homogène périodique avec un maillage contenant N_L points dans chaque direction spatiale. On note le nombre de degrés de liberté du problème $p = N_L^3 N_{var}$, avec N_{var} le nombre de variables considérées ($N_{var} = 5$ pour les équations de Navier-Stokes pour un fluide compressible).

On note $N_{p,comm}$ le nombre de variables (degrés de liberté) transférées durant l'ensemble de la simulation, et N_{step} le nombre de pas de temps requis par le solveur fin \mathcal{F} pour simuler l'ensemble du domaine temporel prescrit par l'utilisateur. Ceci nous permet de définir le "Volume de Communication Normalisé" (VCN) comme étant :

$$Q = \frac{N_{p,comm}}{pN_{step}}.$$
(4.16)

On se propose de déterminer cette valeur dans le cadre d'une parallélisation spatiale exclusive d'une part, et de parallélisations spatio-temporelles basées sur les deux implémentations de PARAREAL données en Sec. 4.1.1.b d'autre part : *Original Pipeline* (OP, *cf.* Fig. 4.1.a) et *Modified Pipeline* (MP, *cf.* Fig. 4.1.b).

VCN pour la parallélisation spatiale seule. Dans le cadre d'une parallélisation basée uniquement sur une décomposition de domaine en espace, les échanges sont de type "halo" et le VCN s'écrit :

$$Q_S = \frac{N_{proc} Q_{S,proc}(N_L, N_{proc}) N_{comm/step}}{p},$$
(4.17)

avec $Q_{S,proc}$ le volume de données envoyées par un seul processeur lors d'une communication spatiale, et $N_{comm/step}$ le nombre de communications requises lors de l'intégration temporelle sur un pas de temps. En particulier, $Q_{S,proc}$ s'exprime en fonction de la décomposition de domaine spatiale utilisée pour le problème de taille N_L avec N_{proc} processus, mais également de la discrétisation spatiale utilisée au sein du solveur.

VCN des solveurs \mathcal{F} et $\widehat{\mathcal{G}}$ pour la parallélisation spatio-temporelle. On considère une parallélisation spatio-temporelle sur N_{proc} processus, basée sur l'utilisation de PARAREAL avec grossissement spatial, avec N le nombre de *time-slices*. Dans ce cadre, on peut considérer dans notre cas $N_{step} = 2N_{\Delta_t,\widehat{\mathcal{G}}}N$. En appliquant (4.17) au solveur fin, on évalue le VCN induit par la parallélisation spatiale de \mathcal{F} pour une *time-slice* :

$$Q_{\mathcal{F}} = \frac{(N_{proc}/N) (N_{step}/N) Q_{S,proc}(N_L, N_{proc}/N) N_{comm/step}}{p N_{step}},$$
$$= \frac{N_{proc} Q_{S,proc}(N_L, N_{proc}/N) N_{comm/step}}{p N^2},$$
(4.18)

et de même pour $\widehat{\mathcal{G}}$:

$$\mathcal{Q}_{\widehat{\mathcal{G}}} = \frac{(N_{proc}/N) (N_{step}/N/2) Q_{S,proc}(N_L/2, N_{proc}/N) N_{comm/step}}{p N_{step}},$$
$$= \frac{N_{proc} Q_{S,proc}(N_L/2, N_{proc}/N) N_{comm/step}}{2p N^2}.$$
(4.19)

Volume de communication pour PARAREAL OP. En se référant à Fig. 4.1.a, on évalue le VCN pour l'ensemble des *time-slices* après *K* itérations de PARAREAL (avec ou sans grossissement spatial) :

$$\mathcal{Q}_{T,OP}(K) = \left[\frac{K(2N - K - 1)}{2} + N\right] \frac{p}{pN_{step}} = \frac{K(2N - K - 1) + 2N}{4N_{\Delta_t,\widehat{\mathcal{G}}}N}.$$
 (4.20)

On en déduit le volume normalisé de données transférées pour l'utilisation de PARAREAL avec grossissement spatial après \widehat{K} itérations :

$$\mathcal{Q}_{\mathcal{P},OP} = N\mathcal{Q}_{\widehat{\mathcal{G}}} + \sum_{k=1}^{\widehat{K}} (N-k)\mathcal{Q}_{\mathcal{F}} + \sum_{k=1}^{\widehat{K}} (N-k-1)\mathcal{Q}_{\widehat{\mathcal{G}}} + \mathcal{Q}_{T,OP}(\widehat{K}),$$
$$= N\mathcal{Q}_{\widehat{\mathcal{G}}} + \frac{\widehat{K}(2N-\widehat{K}-1)}{2}\mathcal{Q}_{\mathcal{F}} + \frac{\widehat{K}(2N-\widehat{K}-3)}{2}\mathcal{Q}_{\widehat{\mathcal{G}}} + \mathcal{Q}_{T,OP}(\widehat{K}).$$
(4.21)

Volume de communication pour PARAREAL MP. On se réfère maintenant à Fig. 4.1.b pour obtenir :

$$Q_{T,MP}(K) = \left[\frac{K(2N - K - 1)}{2}\right] \frac{p}{pN_{step}} = \frac{K(2N - K - 1)}{4N_{\Delta_t,\widehat{g}}N}.$$
(4.22)

Puis on en déduit :

$$\mathcal{Q}_{\mathcal{P},MP} = \sum_{n=1}^{N} n \mathcal{Q}_{\widehat{\mathcal{G}}} + \sum_{k=1}^{\widehat{K}} (N-k) \mathcal{Q}_{\mathcal{F}} + \sum_{k=1}^{\widehat{K}} (N-k-1) \mathcal{Q}_{\widehat{\mathcal{G}}} + \mathcal{Q}_{T,MP}(\widehat{K})$$
(4.23)

$$=\frac{N(N+1)}{2}\mathcal{Q}_{\widehat{\mathcal{G}}}+\frac{\widehat{K}(2N-\widehat{K}-1)}{2}\mathcal{Q}_{\mathcal{F}}+\frac{\widehat{K}(2N-\widehat{K}-3)}{2}\mathcal{Q}_{\widehat{\mathcal{G}}}+\mathcal{Q}_{T,MP}(\widehat{K}).$$
 (4.24)

4.1.2.b Applications au solveur Hybrid

Mise en place de l'analyse. On applique ici le modèle de communication à HYBRID. On fixe $N_{comm/step} = 4$ (utilisation d'un schéma d'intégration Runge-Kutta en 4 étapes), et on reproduit la décomposition spatiale utilisée dans HYBRID pour déterminer la taille de chaque sous-domaine spatial ⁶. $Q_{S,proc}$ est ensuite déterminé en multipliant le nombre de points sur chaque face du sous-domaine par N_{var} et la demi-taille du stencil utilisé pour la discrétisation spatiale (*e.g.* 3 pour les schémas centrés d'ordre 6 utilisés dans HYBRID). On représente en Fig. 4.5 les VCN pour différentes stratégies de parallélisation spatio-temporelle, en distinguant l'implémentation OP (Fig. 4.5a) et MP (Fig. 4.5b). Afin de généraliser les résultats à différentes tailles de problème, on représente Q en fonction de N_{proc}/p , *i.e.* une abscisse correspond à un nombre donné de degrés de liberté (ou charge) par processus pour la parallélisation spatiale seule.

Analyse. Pour les deux implémentations de PARAREAL considérées, on observe une réduction du VCN par rapport à la parallélisation spatiale seule pour la première itération avec N = 4. De plus, la taille des *time-slices* n'a que très peu d'influence sur le VCN associé à chaque stratégie, et ce même pour des valeurs très faibles de $N_{\Delta t, \widehat{\mathcal{G}}}$. Ceci nous permet d'affirmer qu'au sein d'une parallélisation spatio-temporelle, le VCN est majoritairement dominé par les communications spatiales engendrées par \mathcal{F} et $\widehat{\mathcal{G}}$. Cet aspect a un effet notable pour l'implémentation MP, où $\widehat{\mathcal{G}}$ est calculé de manière redondante. Globalement, le VCN y est nécessairement plus élevé, et augmente avec le nombre de *time-slices*.

Enfin, on peut aussi noter que l'utilisation de deux itérations induit une importante augmentation de Q, et pour N = 4 le VCN de la parallélisation spatio-temporelle atteint une valeur comparable à celui de la parallélisation spatiale seule (et ce pour les deux implémentations de PARAREAL).

^{6.} Il s'agit de la même décomposition que celle utilisée pour la mesure du temps d'une communication temporelle, pour plus d'informations voir le code source disponible sous le lien https://gitlab.com/tlunet/parallel-in-time.



(a) Implémentation Original Pipeline

(b) Implémentation Modified Pipeline

FIGURE 4.5 – Comparaison des VCN pour différentes stratégies parallèles. Tirets : "grandes" time-slices avec $N_{\Delta_t,\widehat{G}} = 100$, pointillés : "petites" time-slices avec $N_{\Delta_t,\widehat{G}} = 5$.

4.1.2.c Conclusions concernant l'analyse des volumes de communication.

Le modèle de communication développé a donc permis d'évaluer le volume des communications engendrées sur l'ensemble du système par une parallélisation espace-temps utilisant PARAREAL avec grossissement spatial, puis de le comparer à celui associé à une parallélisation spatiale seule. Il a été remarqué que le volume de communication induit par les transferts de données entre les *time-slices* est marginal. Ainsi, la plus grande partie du volume de communications induites par la parallé-lisation spatiale des solveurs fin et grossier. L'utilisation d'un nombre faible d'itérations est donc nécessaire pour engendrer une réduction effective du volume de données transférées vis-à-vis d'une paralléli-sation spatiale seule avec le même nombre de processus.

Cette étude est à mettre en relation avec le coût de transfert de données selon le niveau de parallélisation considéré. En effet, transférer une unité de donnée n'a pas le même coût selon le bus ⁷ de transfert considéré. Celui-ci peut en effet être par exemple interne au nœud ou au contraire induire une communication inter-rack du super-calculateur. Selon la configuration, le coût éner-gétique n'est pas le même [Bergman *et al.*, 2008, Sec. 6.5]. De ce fait, il apparaît nécessaire de restreindre la parallélisation spatiale dans des configurations où les communications entre processus sont les moins coûteuses (*e.g.* décomposition des tâches sur des nœuds communs, ou des GPU), tandis que les canaux de communication les plus coûteux peuvent être laissés pour les communications temporelles. Dans la perspective d'une étude des coûts de communication plus détaillée, le modèle développé dans cette section peut être par ailleurs complété par des informations relatives au coût énergétique associé à chaque type de communication, dans le but de modéliser et/ou d'optimiser le coût énergétique d'une stratégie de parallélisation espace-temps basée sur PARAREAL.

4.2 Introduction à l'analyse de convergence

Simuler avec précision l'évolution des écoulements turbulents représente encore aujourd'hui un challenge en raison de la large gamme d'échelles qu'il faut traiter pour capturer les phénomènes physiques. Avant d'analyser la convergence de PARAREAL, on introduit donc la physique générale des problèmes qui nous intéressent. À cet effet, les équations de Navier-Stokes pour un fluide compressible sont présentées en Sec 4.2.1, tandis que le contexte et les défis liés à leur intégration temporelle sont

^{7.} Ici, le bus désigne un canal de communication entre deux processus.

présentés en Sec. 4.2.2. On introduit enfin en Sec. 4.2.3 les deux problèmes choisis pour cette étude de la convergence de PARAREAL, et les motivations sous-jacentes à ce choix.

4.2.1 Equations de Navier-Stokes compressible

On considère un fluide compressible évoluant dans les trois directions de l'espace, représenté par :

- sa densité $\rho \in \mathbb{R}^*_+$,
- sa vitesse $\boldsymbol{u} = (u, v, w) \in \mathbb{R}^3$,
- son énergie totale $e_T = e + \frac{u \cdot u}{2} \in \mathbb{R}^*_+$, avec *e* l'énergie spécifique interne du fluide.

Chaque variable peut être exprimée en tout point $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$, et toutes les quantités discutées par la suite sont dépendantes du temps $t \in \mathbb{R}_+$, sauf mention contraire. En particulier, l'indice 0 référera à un état initial à t = 0. Les équations décrivant l'évolution temporelle des cinq variables d'état s'écrivent :

$$\frac{\partial \rho}{\partial t} + \nabla .(\rho \boldsymbol{u}) = 0, \tag{4.25}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla .(\rho \boldsymbol{u} \boldsymbol{u} + p \boldsymbol{\delta}) = \nabla .\boldsymbol{\tau}, \qquad (4.26)$$

$$\frac{\partial \rho e_T}{\partial t} + \nabla . (\boldsymbol{u}(\rho e_T + \boldsymbol{p})) = \nabla . (\boldsymbol{u} \cdot \boldsymbol{\tau} - \boldsymbol{q}), \qquad (4.27)$$

avec p la pression du fluide, δ le tenseur identité, τ le tenseur des tensions visqueuses et q le flux de chaleur. e, p et ρ sont liés par l'équation d'état d'un gaz parfait :

$$p = (\gamma - 1)\rho e, \tag{4.28}$$

avec γ le coefficient adiabatique constant du gaz, tandis que τ et q sont modélisés avec les hypothèses de Stokes :

$$\boldsymbol{\tau} = 2\mu \boldsymbol{S} - \frac{2}{3}\mu(\nabla \boldsymbol{.}\boldsymbol{u})\boldsymbol{\delta}, \qquad (4.29)$$

$$\boldsymbol{q} = -k_c \nabla T, \tag{4.30}$$

avec $S = (\nabla u + (\nabla u)^T)/2$ le tenseur des déformations, μ la viscosité dynamique, k_c la constante de conductivité thermique, $T = (\gamma - 1)e/R$ la température et R la constante des gaz parfaits. La dépendance de μ vis-à-vis de la température est représentée par la loi exponentielle :

$$\mu/\mu_{ref} = (T/T_{ref})^{3/4}, \qquad (4.31)$$

et le ratio entre μ et k_c est déterminé par le nombre de Prandtl :

$$Pr = \frac{\gamma R\mu}{(\gamma - 1)k},\tag{4.32}$$

propriété du fluide et considéré constant, ce qui permet de fermer l'ensemble des équations.

4.2.2 Challenges pour l'intégration temporelle

La résolution des équations aux dérivées partielles en section précédente s'articule généralement autour de deux éléments principaux : une discrétisation spatiale et une méthode d'intégration temporelle. Ici nous nous limiterons à l'étude de l'intégration temporelle, et de sa pertinence vis-à-vis de la problématique des écoulements turbulents. **Caractériser et quantifier la turbulence.** La turbulence est un état bien connu des écoulements fluides régis par les équations de Navier-Stokes. En particulier, celle-ci se manifeste lorsque les échelles caractéristiques du terme d'advection sont bien plus importantes que celles associées à la diffusion visqueuse. En général, ce rapport d'échelle est évalué de manière adimensionnelle par le biais d'un nombre de Reynolds défini par rapport aux échelles caractéristiques du problème considéré.

Un écoulement turbulent est par nature multi-échelles : il est peuplé de structures de tailles et niveaux d'énergie différents. La gamme d'échelle présente dans l'écoulement est directement liée au rapport entre :

- 1. les échelles caractéristiques des grandes structures dites énergétiques qui produisent la turbulence (*e.g.* épaisseur d'une couche limite cisaillée),
- 2. les échelles caractéristiques des structures dites dissipatives liées à la viscosité du fluide. On désigne généralement par η la taille caractéristique des plus petites échelles dissipatives présentes dans l'écoulement (échelle de Kolmogorov). À cette grandeur spatiale est associé un temps caractéristique τ_{η} , qui est d'autant plus court que η est petite.

Ce ratio se matérialise par un nombre de Reynolds (*e.g.* Reynolds de turbulence Re_t , ou Re_λ [Pope, 2000, Chap. 6]), proportionnel à l'écart de taille entre les plus grandes et plus petites échelles turbulentes. En particulier, on distingue ce type de définition du nombre de Reynolds de celle basée sur une vitesse d'écoulement moyen et une longueur caractéristique du problème. Cette dernière définition est généralement utilisée pour prédire une transition d'un état laminaire à turbulent.

État de l'art pour les méthodes d'intégration temporelles. Les méthodes choisies pour intégrer les équations de Navier-Stokes se classent en deux grandes catégories : implicite ou explicite.

Les méthodes implicites, inconditionnellement stables, n'ont pas de limitation sur la taille du pas de temps utilisé. Néanmoins, elles requièrent la résolution d'un système non-linéaire à chaque pas de temps, dont la taille est identique à celle du problème considéré. De ce fait, le coût de calcul d'un seul pas de temps est bien plus important que celui d'un pas de temps pour une méthode explicite. Elle sont généralement favorisées pour des problèmes paraboliques (terme visqueux dominant). Actuellement, et plus particulièrement pour des écoulements turbulents, des méthodes semiimplicites, pour lesquelles seuls les termes visqueux sont traités implicitement⁸, sont généralement favorisées (pour exemple, l'intégration temporelle utilisée par [Lee et Moser, 2015], développée initialement par [Kim et Moin, 1985]),

Les méthodes explicites, quant à elles, sont particulièrement simples d'application, et permettent d'intégrer une solution sur un pas de temps de manière bien plus efficace. En particulier, paralléliser en espace un solveur explicite peut être parfois bien plus simple en comparaison avec un solveur implicite ⁹. Cependant, la taille du pas de temps est limitée par des conditions de stabilité numérique (condition CFL), qui peuvent être parfois extrêmement restrictives. Des intégrations temporelles explicites sont généralement utilisées pour les équations de Navier-Stokes pour des fluides compressibles ([Johnsen *et al.*, 2010]), et de manière générale, favorisées pour des problèmes à caractère hyperbolique.

Le temps caractéristique des petites échelles, métronome de l'intégration temporelle. Indépendamment des conditions de stabilité numérique, la turbulence impose aussi que le pas de temps d'intégration δ_t soit de l'ordre du temps caractéristique des petites échelles τ_{η} [Choi et Moin, 1994]. En particulier, l'utilisation de pas de temps bien plus large par le biais d'une

^{8.} En particulier, les méthodes semi-explicites sont contraintes, tout comme les méthodes explicites, à une condition CFL pour une intégration numérique stable

^{9.} On peut noter que certains solveurs implicites requièrent uniquement des itérations avec évaluation explicite des dérivées spatiales. Pour ce genre de méthode, la parallélisation spatiale a un même degré de complexité que pour les méthodes explicites.

méthode entièrement implicite, bien que stable numériquement, peut induire une importante erreur d'intégration (qui peut se traduire par une re-"laminarisation" d'un écoulement turbulent en raison d'un excès de dissipation [Choi et Moin, 1994]). C'est pourquoi les méthodes explicites, très efficaces pour de petites tailles de pas de temps, sont majoritairement favorisées pour les calculs LES ou DNS en CFD.

Le chaos, détracteur d'une intégration temporelle précise. Un autre challenge pour l'intégration temporelle, lié à la turbulence, provient de son caractère chaotique lié à la forte non-linéarité des équations de Navier-Stokes. Celui-ci se manifeste par une grande sensibilité aux perturbations : une petite variation entre deux solutions initiales entraîne une grande différence entre les solutions finales [Pope, 2000, Chap. 1]. Pour évaluer cette sensibilité, on considère deux simulations identiques, différenciées par une infime différence entre leurs états initiaux. Pour un système chaotique, cette différence va augmenter en suivant un taux de croissance de la forme $e^{\lambda_L t}$, avec λ_L le coefficient (maximal) de Lyapunov (introduit par [Eckmann et Ruelle, 1985]). Ce dernier se définit pour un système chaotique représenté par une fonction d'évolution $\Phi(\xi, t)$:

$$\lambda_L = \lim_{t \to \infty} \lim_{\epsilon \to 0} \frac{1}{t} \log \left(\frac{\|\Phi(\xi + \epsilon \xi', t) - \Phi(\xi, t)\|}{\|\epsilon \xi'\|} \right).$$
(4.33)

avec $\|.\|$ une norme définie pour le problème considéré.

Pour certains écoulements turbulents, des travaux ont rapidement été effectués pour estimer λ_L de manière expérimentale [Wolf *et al.*, 1985] ou par le biais de simulations [Vastano et Moser, 1991]. Les différentes observations ont abouti au postulat d'une valeur de λ_L inversement proportionnelle ¹⁰ au temps caractéristique τ_{η} des plus petites échelles de la turbulence [Wang *et al.*, 2013]. Cette grandeur détermine ainsi l'échelle de temps à partir de laquelle l'erreur de discrétisation va modifier de manière significative la solution. Cette échelle de temps est de fait conditionnée par le nombre de Reynolds, et s'accroît avec ce dernier. C'est pourquoi les méthodes d'intégration temporelles doivent être suffisamment robustes pour permettre d'obtenir de "bonnes" solutions physiques au temps final, malgré l'amplification de l'erreur d'intégration induite par le caractère de plus en plus chaotique de la turbulence.

En particulier, ce caractère chaotique implique une manière différente d'appréhender l'erreur d'intégration temporelle. On peut par exemple considérer deux simulations intégrées temporellement avec des pas de temps respectifs δ_t et $\delta_t/2$, de telle sorte que δ_t soit suffisamment petit par rapport au temps caractéristique des petites échelles τ_{η} (*i.e.* $\delta_t < \tau_{\eta}$). La différence entre les champs des deux simulations, bien qu'infime initialement, va donc être amplifiée par l'aspect chaotique de la turbulence, et générer deux solutions complètement différentes au bout d'un nombre suffisamment grand de pas de temps d'intégration temporelle. Pourtant, de par les tailles de pas de temps choisies, chacune des simulations est suffisamment bien résolue temporellement pour correctement représenter l'évolution de la turbulence, jusque pour les plus petites échelles. C'est pourquoi il semble judicieux de considérer des critères d'erreur statistiques sur la turbulence obtenue, plutôt que sur une différence point à point avec une solution de référence.

Entre solutions d'hier ... Bien que de nombreux efforts aient été consacrés à la discrétisation spatiale, les méthodes d'intégration temporelle utilisées en CFD sont restées relativement classiques. Pour exemple, le problème de plus grande taille traité par DNS à ce jour [Lee et Moser, 2015] a été intégré avec une méthode développée il y a plus de trente ans, tandis que le solveur HYBRID, pour intégrer un problème de plus de quatre billions de points (10¹²) [Bermejo-Moreno *et al.*, 2013a], utilise une méthode mise au point il y a plus d'un siècle [Runge, 1895]. De manière générale, le bon

^{10.} On peut mentionner un travail récent basé sur l'utilisation de simulations numériques directes d'une turbulence homogène isotrope [Mohan *et al.*, 2017], qui a mis en évidence une légère dépendance de $\lambda_L \tau_{\eta}$ vis-à-vis du nombre de Reynolds.

comportement de ces méthodes dans des situations variées d'écoulements n'appelle pas au développement de méthodes alternatives au sein de la communauté CFD.

Les méthodes explicites ont néanmoins connu quelques développement notables : l'optimisation pour des problèmes spécifiques comme l'acoustique de [Bogey et Bailly, 2004] en est un exemple. L'amélioration de l'efficacité de stockage mémoire [Kennedy *et al.*, 2000], ou l'adaptation pour certains types de schémas de discrétisation spatiales [Parsani *et al.*, 2013] sont d'autres améliorations intéressantes. Cependant, le formalisme utilisé (du type Runge-Kutta, ou *multi-step*) a peu évolué depuis.

... et challenges de demain. L'émergence de nouveaux problèmes plus complexes, multiéchelles, ainsi que l'arrivée de nouvelles architectures de calcul, poussent actuellement la communauté scientifique à explorer d'autres approches. Pour exemple, l'intégration temporelle asynchrone pour des problèmes présentant une forte non-homogénéité spatiale [Seny *et al.*, 2014], la règle de balayage (*swept-rule*) permettant de réduire la fréquence des communications dans le cadre d'une parallélisation spatiale [Alhubail et Wang, 2016], ou encore l'utilisation de l'approche SDC pour une intégration temporelle multi-échelle d'écoulements compressibles réactifs [Emmett *et al.*, 2014]. D'autres stratégies combinent les dimensions spatiales et temporelles, comme les techniques adaptatives en temps et espace [Duarte *et al.*, 2013], ou encore les méthodes basées sur la décomposition des opérateurs pour la résolution de problèmes multi-variables et multiéchelles [Descombes et Massot, 2004].

L'intégration parallèle en temps se situe donc dans cette catégorie de méthodes cherchant à utiliser une nouvelle approche pour l'intégration temporelle, et dont les propriétés restent peu connues pour la simulation d'écoulements turbulents. Il convient donc d'évaluer sa capacité à représenter l'évolution de la turbulence sur l'ensemble de ses échelles, et son comportement vis-à-vis du caractère chaotique d'un écoulement turbulent. Cette thèse cherche à apporter des éléments de réponses à ces questions au travers de deux problèmes tests très connus de la communauté CFD, puisqu'ils incorporent des mécanismes fondamentaux de la turbulence à moindre coût.

4.2.3 Choix des problèmes tests

A notre connaissance, la convergence de PARAREAL avec grossissement spatial pour la simulation d'écoulements turbulents a été peu étudiée, et plus particulièrement dans le cadre de simulations numériques directes (DNS). C'est pourquoi les deux problèmes tests choisis pour l'étude ont pour caractéristique principales d'être simples et très documentés, tout en présentant les caractéristiques essentielles des écoulements turbulents.

Restriction à des DNS d'écoulements faiblement compressibles. Même si le code HYBRID peut calculer des écoulements fortement compressibles, on limite l'étude pour des configurations à faible nombre de Mach (autour de 0.3) puisque la compressibilité, à l'opposé du nombre de Reynolds, a très peu d'effet direct sur la turbulence.

Dans un cadre *Exascale* et comme souligné dans le Chap. 1, ce sont les techniques de type LES ou DNS qui exploiteront le plus ce degré de liberté supplémentaire pour le parallélisme. L'objectif de l'étude étant dans un premier temps d'examiner l'effet d'une intégration temporelle parallèle sur la physique de la turbulence, on choisit d'écarter l'utilisation de la LES afin d'isoler les erreurs attribuées à l'algorithme, de l'erreur induite par la modélisation de la turbulence. De ce fait, on se focalise exclusivement sur une résolution de type DNS pour les écoulements turbulents considérés.

Décroissance d'une turbulence homogène isotrope. La décroissance d'une Turbulence Homogène Isotrope (THI) a été étudiée par de nombreux auteurs, en évoluant de l'analyse de problèmes de taille modérée [Rogallo, 1981, Vincent et Meneguzzi, 1991] vers des configurations bien plus grandes en fonction de la puissance grandissante des supercalculateurs [Kaneda *et al.*, 2003].

On peut considérer ce problème comme la configuration la plus simple incorporant les principaux mécanismes turbulents fondamentaux [Pope, 2000], de par ses propriétés d'homogénéité et d'iso-tropie, permettant une représentation statistique de la turbulence.

Dans le contexte de simulation numérique directe, la taille du problème est essentiellement pilotée par la gamme d'échelles de l'écoulement turbulent considéré, à partir de laquelle est réglée la résolution spatiale. L'augmentation du nombre de Reynolds (et donc de la taille du problème) implique la résolution d'un spectre d'échelles plus large, et rend la simulation plus représentative des configurations industrielles à haut nombre de Reynolds.

Écoulement de canal turbulent. L'Écoulement de Canal Turbulent (ECT) a été largement étudié par DNS depuis [Kim *et al.*, 1987], dans le but de caractériser l'influence des parois rigides sur les propriétés d'un écoulement turbulent. La taille des problèmes considérés n'a cessé d'évoluer, de quelques dizaines de millions de points [Moser *et al.*, 1999], à des centaines de milliards récemment [Lee et Moser, 2015], dans le but d'atteindre des nombres de Reynolds toujours plus proches de ceux d'applications industrielles. Ce problème apporte un degré supplémentaire par rapport à la configuration de THI, car il présente un écoulement moyen et une direction non homogène. Dans cette direction, normale à la paroi, la physique de la turbulence se caractérise par deux comportements bien distincts, selon que l'on se trouve proche ou loin de la paroi. Par ailleurs, à la différence de la THI, ce problème est statistiquement stationnaire.

4.3 Décroissance d'une Turbulence Homogène Isotrope

4.3.1 Description du problème

L'ensemble des notations, variables et grandeurs définies à partir d'ici sera utilisé exclusivement pour la Sec. 4.3, la plupart de ces définitions n'étant valides que dans le cadre du problème considéré. L'ensemble des définitions et propriétés présentées dans cette section provient d'une description plus complète donnée par [Pope, 2000, Chap. 6 & Sec. 9.1], que l'on résumera ici pour introduire la description des résultats obtenus avec PARAREAL. Un glossaire regroupant l'ensemble des notations définies est disponible en Ap. D.

4.3.1.a Type d'écoulement et domaine considéré.

La décroissance de Turbulence Homogène Isotrope (THI) est obtenue par la résolution des équations (4.25)-(4.27) sans terme source, sur un maillage tridimensionnel périodique uniforme dans chaque direction (représenté en Fig. 4.6a), à partir d'un état initial donné. Ce dernier peut être obtenu en fixant les champs initiaux des variables d'état (ρ_0 , u_0 , T_0), et les paramètres R, γ , Pr, μ_{ref} et T_{ref} . Les champs sont construits à partir d'un processus aléatoire qui génère un champ turbulent artificiel où l'énergie cinétique turbulente est principalement concentrée aux grandes échelles tourbillonnaires. La méthodologie associée à la construction de ces champs initiaux est décrite en détail dans [Johnsen *et al.*, 2010, Ap. A] et [Ristorcelli et Blaisdell, 1997].

Une fois les champs initiaux construits, les équations (4.25)-(4.27) sont intégrées pour observer la décroissance énergétique des structures tourbillonnaires de l'écoulement. Une représentation de l'énergie cinétique turbulente à un temps donné est présentée en Fig. 4.6b. On peut y observer les propriétés d'homogénéité et d'isotropie du problème. En particulier, l'utilisation concourante d'un maillage uniforme périodique permet de réduire l'étude de la THI en décroissance à celle de l'évolution temporelle de quantités moyennées en espace ¹¹, que l'on note ainsi :

$$\langle f \rangle = \frac{1}{L^3} \iiint_V f(x, y, z) \, \mathrm{d}x \mathrm{d}y \mathrm{d}z. \tag{4.34}$$



FIGURE 4.6 – Décroissance d'une Turbulence Homogène Isotrope (THI).

4.3.1.b Grandeurs caractéristiques de l'écoulement.

La solution turbulente de THI est caractérisée par deux nombres sans dimension, qui varient au cours du temps, et pilotent l'évolution de la turbulence :

• le nombre de Reynolds basé sur l'échelle de Taylor (échelle de décorrélation spatiale) :

$$Re_{\lambda} = \frac{\langle \rho \rangle u' \lambda}{\langle \mu \rangle},$$
(4.35)

• le nombre de Mach turbulent :

$$M_t = \sqrt{3} \frac{u'}{\langle c \rangle},\tag{4.36}$$

avec $u' = \sqrt{(\langle uu \rangle + \langle vv \rangle + \langle ww \rangle)/3}$, $c = \sqrt{\gamma RT}$ la vitesse du son dans le fluide, et λ la micro-échelle de Taylor ¹².

 Re_{λ} détermine la largeur du spectre d'énergie cinétique turbulente, et M_t indique l'influence des effets de compressibilité, qui peuvent introduire des discontinuités (*e.g. shocklets*) au sein des champs de solution pour des M_t élevés.

4.3.1.c Énergie, dissipation et spectre de turbulence.

Énergie cinétique turbulente et dissipation. Afin de décrire le type d'écoulement représenté en Fig. 4.6b, on définit l'énergie cinétique turbulente k_e et la dissipation ϵ comme suit :

$$k_e = \frac{3}{2} {u'}^2, \ \epsilon = \langle \frac{\mu}{\rho} \boldsymbol{S}. \boldsymbol{S} \rangle.$$
 (4.37)

12. Pour une turbulence homogène isotrope, λ peut s'écrire dans une formulation simplifiée $\lambda = u' \sqrt{15 \frac{\langle \mu \rangle}{\epsilon \langle \rho \rangle}}$

^{11.} Plus précisément, cette description est possible car la moyenne volumique permet d'approcher très finement la moyenne d'ensemble, que l'on calculerait en se basant sur un nombre statistiquement représentatif de conditions initiales différentes.

L'empreinte de k_e et ϵ étant majoritairement située respectivement dans les grandes et petites échelles, ces deux quantités sont des indicateurs intéressants de l'évolution de la turbulence au cours d'une simulation de THI. Il est possible de dériver une équation de transport pour k_e , qui se réduit dans le cas de la THI à :

$$\frac{\mathrm{d}k_e}{\mathrm{d}t} = -\epsilon. \tag{4.38}$$

On peut observer pour la simulation de THI en décroissance (*cf.* [Johnsen *et al.*, 2010, Sec. 3.5]) qu'après une courte phase transitoire, ces deux grandeurs décroissent exponentiellement, comme montré en Fig. 4.7a.



(a) Évolution temporelle de $k_e/k_{e,0}$ et $\epsilon/k_{e,0}$.

(b) Évolution du spectre d'énergie.

FIGURE 4.7 – Évolutions des variables d'importance lors de la décroissance de THI, avec $\tau_{\lambda_0} = (\lambda/u')_{t=0}$, $N_L = 80$, $Re_{\lambda_0} = 46$.

Spectre d'énergie cinétique de la turbulence. Il est possible en THI de décrire plus en détail la concentration énergétique au sein des différentes échelles de l'écoulement, en utilisant le spectre de densité spectrale d'énergie $E(\kappa)$ du champ turbulent. Ce dernier représente le contenu énergétique associé à un nombre d'onde ¹³ κ donné. On y associe conceptuellement des structures tourbillonnaires de grande/petite taille, représentées respectivement par les petits/grands nombres d'onde. Aussi, on a la relation :

$$k_e = \sum_{\kappa} E(\kappa). \tag{4.39}$$

Le calcul de $E(\kappa)$ nécessite une intégration non triviale de coquilles sphériques à norme de nombre d'onde donné, explicitée en Ap. E.

Une représentation du spectre d'énergie pour une simulation de THI est donnée en Fig. 4.7b. Le spectre du champ initial synthétique présente un pic énergétique dans les grandes échelles, conformément au processus d'initialisation utilisé. Après une phase transitoire pendant laquelle la turbulence initialement artificielle devient "physique" (finissant autour de $t \sim \tau_{\lambda_0}$ pour Fig. 4.7), la **cascade énergétique de la turbulence**¹⁴ se met en place : l'énergie des grands tourbillons est continuellement transférée vers les tourbillons de taille inférieure, jusqu'à dissipation des plus petits tourbillons par viscosité moléculaire. Dès lors, k_e est réparti sur l'ensemble des échelles turbulentes, et l'on observe leur décroissance énergétique ($t/\tau_{\lambda_0} = 1 \rightarrow 4$).

^{13.} Ou encore, norme d'un ensemble de vecteurs d'onde tridimensionnel.

^{14.} On distingue cette cascade énergétique "physique" de celle s'effectuant dès l'instant initial, qui consiste en un "remplissage" énergétique du spectre. On parle pour cette dernière de "reconditionnement" de turbulence synthétique.

4.3.1.d Échelles et temps caractéristiques.

L'évolution de la décroissance de la THI peut être associée à divers échelles et temps caractéristiques. On décrit ici les principales, qui sont utilisées dans la suite de l'étude.

De par les propriétés d'homogénéité et d'isotropie de l'écoulement considéré, l'échelle de Kolmogorov peut se définir en THI de la manière suivante :

$$\eta = \left(\frac{\langle \mu \rangle^3}{\epsilon \langle \rho \rangle^3}\right)^{1/4},\tag{4.40}$$

tandis que le temps caractéristique associé s'écrit :

$$\tau_{\eta} = \left(\frac{\langle \mu \rangle}{\epsilon \langle \rho \rangle}\right)^{1/2}.$$
(4.41)

Ce dernier est par ailleurs lié à une échelle de temps caractéristique basée sur l'échelle de Taylor, classiquement utilisé en THI :

$$\tau_{\lambda} = \frac{\lambda}{u'} = \sqrt{15}\tau_{\eta}.$$
(4.42)

Par la suite, on utilisera cette dernière échelle exprimée à l'instant initial (τ_{λ_0}) pour normaliser les valeurs de temps.

4.3.2 Définition des solveurs fin et grossier pour Parareal

4.3.2.a Critère de bonne résolution DNS pour le solveur fin.

Condition sur le plus grand nombre d'onde du maillage spatial. Dans le cadre d'une DNS, une condition nécessaire pour le maillage spatial est de pouvoir représenter correctement l'ensemble des structures tourbillonnaires associées aux différentes échelles de longueur détaillées en Sec. 4.3.1.d, en particulier celle des plus petites structures dissipatives η . Si l'on considère le plus grand nombre d'onde pouvant être représenté sur un maillage cartésien uniforme de taille N_L^3 :

$$\kappa_{\max}(N_L) = \frac{\pi N_L}{L},\tag{4.43}$$

on peut dériver une condition nécessaire à la bonne résolution des structures de taille η (*cf.* [Pope, 2000, Chap. 9]) :

$$\forall t, \ \eta \ \kappa_{\max}(N_L) \ge \alpha, \tag{4.44}$$

avec α un coefficient dépendant du solveur, et plus spécifiquement des schémas de discrétisation spatio-temporelle. Des DNS récentes de THI avec des discrétisations spectrales ont utilisé $\alpha = 1$ [Kaneda *et al.*, 2003]. En prenant compte du fait qu'HYBRID utilise une discrétisation spatiale centrée d'ordre élevé (*cf.* Sec. 1.2.2), nous choisissons un critère plus restrictif avec $\alpha = 1.5$ pour la résolution du solveur fin \mathcal{F} , en accord avec les expressions de nombres d'onde modifiés pour les schémas du 6^{ème} ordre en espace (*e.g.* [Lele, 1992]).

Fin de phase transitoire. L'écoulement fluide traverse une phase transitoire pendant laquelle il évolue d'une turbulence synthétique (solution initiale) à une turbulence physique. C'est pourquoi nous choisissons d'imposer le critère (4.44) après ce transitoire, que l'on note t_{ϕ} . Afin de proposer une estimation de ce temps caractéristique, on analyse l'évolution du spectre d'énergie $E(\kappa)$ défini en Sec. 4.3.1.c. On définit ici t_{ϕ} comme le temps à partir duquel la densité spectrale d'énergie décroît sur l'ensemble des échelles de turbulence, *i.e.* :

$$t_{\phi} = \min\left\{t > 0 \mid \forall \kappa, \ \delta E(\kappa) < 0\right\},\tag{4.45}$$

avec $\delta E(\kappa)$ le taux de croissance de $E(\kappa)$ en fonction du temps. Ce critère nous permet de proposer une condition systématique pour fixer le nombre maximum Re_{λ_0} d'une THI en décroissance pouvant être convenablement représentée par un maillage de taille N_L^3 donnée. On choisit la valeur maximale de Re_{λ_0} satisfaisant la condition (4.44) à t_{ϕ} , *i.e.* :

$$Re_{\lambda_0,\max} = \max\left\{Re_{\lambda_0} \mid \forall t > t_{\phi}, \ \eta \ \kappa_{\max}(N_L) \ge \alpha\right\}.$$
(4.46)

Une procédure d'essai-erreur est ainsi utilisée pour obtenir le respect de la condition (4.44) à t_{ϕ} , et permet de spécifier un état initial cohérent pour le solveur \mathcal{F} complètement caractérisé par Re_{λ_0} , en considérant le maillage à N_L^3 points.

4.3.2.b Solveur grossier de PARAREAL avec grossissement spatial

Grille spatiale grossière. Profitant du fait que la discrétisation spatiale dans HYBRID est basée sur des maillages cartésiens structurés, on utilise un grossissement spatial géométrique (aussi appelé *vertex-centered coarsening* en multi-grille [Trottenberg *et al.*, 2001]) pour définir le maillage du solveur grossier $\hat{\mathcal{G}}$. Dans cette configuration, le maillage grossier est construit en considérant un point sur deux du maillage fin dans les trois directions spatiales, ce qui implique un ratio de taille de maille de 2 entre les deux maillages. Un exemple de représentation est donné pour une direction en Fig. 4.8. L'utilisation d'un tel grossissement spatial permet donc de régler simplement le pas de



FIGURE 4.8 – Répartition des points de maillages fin et grossier dans la direction x pour $N_L = 8$, conditions aux bords de type périodique.

temps de $\widehat{\mathcal{G}}$ comme étant :

$$\Delta_t = 2\delta_t,\tag{4.47}$$

ce qui permet de garder un nombre de CFL identique pour \mathcal{F} et $\widehat{\mathcal{G}}$.

Opérateurs de transfert. Pour l'opérateur de restriction $\widehat{\mathcal{R}}$, nous avons considéré à la fois l'injection et le *full-weighting* tridimensionnel [Trottenberg *et al.*, 2001, Sec. 2.9].

Pour l'opérateur d'interpolation $\widehat{\mathcal{I}}$, deux types de méthodes sont sélectionnés :

- interpolation basée sur des transformées de Fourier [Russell et al., 2015],
- interpolation d'ordre variable pour maillages structurés.

Cette dernière est basée sur des produits tensoriels successifs d'opérateurs d'interpolation monodimensionnels, ce qui rend son application à la fois peu coûteuse et simple d'implémentation. Élevée à l'ordre 1 et 3, elle est équivalente respectivement à une interpolation trilinéaire [Trottenberg *et al.*, 2001, Sec. 2.9] et tri-cubique [Lekien et Marsden, 2005]. Elle est décrite plus en détails en Ap. C.

4.3.3 Analyse de convergence de PARAREAL pour la décroissance de Turbulence Homogène Isotrope.

Dans cette section on analyse la convergence de PARAREAL avec grossissement spatial pour le problème de THI, ainsi que l'influence des différents paramètres en jeu. Afin de rester cohérent avec le reste de l'étude (cf. Sec. 4.1.1.b §6), on fixe le nombre de time-slices N = 4. Au sein d'Hybrid, on choisit d'utiliser des schémas de discrétisation spatiale d'ordre 6 et la méthode RKC4 pour l'intégration temporelle. Pour l'ensemble des simulations, on fixe $M_t = 0.3$. Ces dernières ont été effectuées sur le système de calcul EOS. Les temps de restitution varient de quelques minutes, avec parallélisation spatiale sur 20 cœurs pour le plus petit problème considéré ($N_L = 80$, $\sim 2.5 \, 10^6$ degrés de liberté) jusqu'à une dizaine d'heures de calcul sur 320 cœurs pour le plus gros ($N_L = 640$, $\sim 1.3 \, 10^9$ degrés de liberté).

En premier lieu, on introduit l'analyse de convergence sur deux illustrations préliminaires en Sec. 4.3.3.a. Les critères d'erreur utilisés pour l'étude sont introduits en Sec. 4.3.3.b, et des analyses paramétriques de la convergence sont menées en Sec. 4.3.3.c, Sec. 4.3.3.d et Sec. 4.3.3.e.

Précision quant à l'utilisation du terme "convergence". Comme il l'a été mentionné en Sec. 4.2 §4, l'aspect chaotique de la turbulence induit qu'une faible erreur d'intégration temporelle peut provoquer une grande différence dans les solutions finales obtenues, sans pour autant altérer la physique de la turbulence. Lorsqu'un nombre donné $\widehat{K} < N$ d'itérations de PARAREAL est effectué, une erreur est nécessairement présente dans la solution de PARAREAL. Il semble donc illusoire de tenter d'obtenir un champ de solution strictement identique à celui d'une solution de référence obtenue avec \mathcal{F} seul. C'est pourquoi dans toute la suite de l'étude (y compris en Sec. 4.4), par *"la solution de PARAREAL converge vers la solution fine"* on sous-entend *"les propriétés statistiques de la solution de PARAREAL convergent vers celles de la solution fine"*.

4.3.3.a Un premier aperçu de la convergence de PARAREAL

On applique ici PARAREAL à un problème de faible taille ($N_L = 80$, $Re_{\lambda_0} = 46$) et on analyse l'écoulement après plusieurs itérations successives. On utilise l'injection comme opérateur de restriction, et on fixe $N_{\Delta_t,\widehat{\mathcal{G}}} = 20$. La décroissance de la THI est simulée entre les temps physiques $t_{\phi} = 1.4\tau_{\lambda_0}$ et $t_{end} = 4.4\tau_{\lambda_0}$, ce qui correspond à la partie la plus marquée de la décroissance (*cf.* Fig. 4.7a). On utilise respectivement l'interpolation la plus grossière (trilinéaire, ordre 1) et la plus fine (par transformée de Fourier). Les spectres de densité spectrale d'énergie correspondants sont représentés en Fig. 4.9 pour le solveur fin, le solveur $\widehat{\mathcal{G}}$ seul, et les premières itérations de PARA-REAL.





(b) Interpolation par transformée de Fourier



De par la résolution spatiale insuffisante de $\hat{\mathcal{G}}$, une sur-estimation de l'énergie pour les petites échelles est visible pour $\kappa = 0.5\kappa_{max}$. Ce comportement, typique d'une DNS sous-résolue, résulte d'une suppression partielle des mécanismes de dissipation physique agissant aux plus petites échelles. Pour le premier cas en Fig. 4.9a, l'effet de l'interpolation est particulièrement souligné à l'itération $\widehat{K} = 0$, pour laquelle l'énergie des échelles intermédiaires est particulièrement dissipée, alors qu'un pic d'énergie est observable pour les plus petites échelles. La première itération ($\widehat{K} = 1$) permet de supprimer ce pic énergétique dans les hautes fréquences, et apporte une nette amélioration dans les échelles intermédiaires.

Dans le deuxième cas en Fig. 4.9b, on observe un comportement opposé à l'itération $\widehat{K} = 0$: alors que le niveau d'énergie des petites échelles est globalement sous-estimé, on observe une surestimation de l'énergie des échelles intermédiaires, identique à celle observée pour $\widehat{\mathcal{G}}$. La deuxième itération quant à elle, permet d'obtenir pour l'ensemble des échelles un niveau d'énergie très proche de celui du solveur fin. Il est intéressant de noter que dans les deux cas observés, la deuxième itération ($\widehat{K} = 2$) a un apport très faible (relativement à celui de la première itération) sur la précision de la solution.

Observation 4.3.1 — Pour le problème de THI, le spectre d'énergie de la solution de PARAREAL avec grossissement spatial converge vers celui de la solution fine, de manière pilotée par les opérateurs de transfert utilisés pour le changement de grille. En particulier, l'apport de la deuxième itération sur la précision est peu important, et ce plus particulièrement pour les petites échelles de la turbulence.

4.3.3.b Sélection de critères d'erreur

Afin de mieux quantifier la qualité des solutions obtenues avec PARAREAL, on utilise l'erreur relative sur la densité spectrale d'énergie e_{rel} , définie ainsi :

$$e_{rel}(\kappa) = \frac{E(\kappa) - E_{\mathcal{F}}(\kappa)}{E_{\mathcal{F}}(\kappa)},$$
(4.48)

où $\tilde{E}(\kappa)$ correspond au spectre de la solution approchée ($\hat{\mathcal{G}}$ ou une itération PARAREAL) et $E_{\mathcal{F}}(\kappa)$ le spectre de la solution de référence calculée uniquement avec \mathcal{F} . Alors qu'une valeur négative pour $e_{rel}(\kappa)$ indique une dissipation d'énergie, une valeur positive témoigne plutôt d'une amplification.

On complète la quantification d'erreur en considérant l'erreur relative sur l'énergie intégrée sur l'ensemble des échelles (k_e) :

$$e_{k_e} = \frac{k_e - k_{e,\mathcal{F}}}{k_{e,\mathcal{F}}},\tag{4.49}$$

et sur la dissipation :

$$e_{\epsilon} = \frac{\tilde{\epsilon} - \epsilon_{\mathcal{F}}}{\epsilon_{\mathcal{F}}}.$$
(4.50)

Ces deux scalaires sont donc représentatifs de l'erreur statistique à deux niveaux : e_{k_e} caractérise les grandes échelles, tandis que e_{ϵ} permet plutôt d'évaluer un comportement sur les petites échelles de l'écoulement.

Les trois indicateurs e_{rel} , e_{k_e} et e_{ϵ} sont donc combinés pour étudier l'influence de plusieurs paramètres sur la qualité de la solution : les opérateurs de transfert en Sec. 4.3.3.c, le nombre de Reynolds en Sec. 4.3.3.d et la taille des *time-slices* en Sec. 4.3.3.e.

4.3.3.c Influence des opérateurs de transfert

Opérateur de restriction On s'intéresse en premier lieu à l'impact de la méthode de restriction utilisée sur la convergence de PARAREAL pour les deux premières itérations. L'erreur relative sur le spectre d'énergie est représentée en Fig. 4.10.

On observe une augmentation de l'erreur de diffusion des grandes vers les petites échelles, et ce quelle que soit la méthode utilisée. L'apport de la deuxième itération est relativement faible, et la restriction *Full-Weighting* apporte généralement une plus grande erreur de diffusion sur l'ensemble des échelles.



FIGURE 4.10 – Influence de la restriction sur l'erreur de PARAREAL. Erreur $e_{rel}(\kappa)$ (4.48) en fonction de la méthode de restriction pour la dernière time-slice. $N_L = 80, N = 4, N_{\Delta_{\star},\widehat{G}} = 20$, interpolation trilinéaire.

Observation 4.3.2 — Pour le problème de THI, la méthode de restriction utilisée pour PARA-REAL avec grossissement spatial a peu d'influence sur la convergence du spectre énergétique. En particulier, l'utilisation du Full-Weighting apporte une plus grande diffusion sur l'ensemble des échelles vis-à-vis de l'injection.

Au vu des observations précédentes, on favorise l'utilisation de l'injection pour PARAREAL avec grossissement spatial durant la suite de l'étude (celle-ci ayant par ailleurs le coût le plus faible entre les deux méthodes à disposition).

Opérateur d'interpolation On s'intéresse maintenant à l'impact de l'opérateur d'interpolation sur la solution de PARAREAL. L'erreur relative sur le spectre d'énergie est représentée en Fig. 4.11, tandis que les erreurs relatives sur l'énergie et la dissipation sont données en Tab 4.3.



FIGURE 4.11 – Influence de l'interpolation sur l'erreur de PARAREAL. Erreur $e_{rel}(\kappa)$ (4.48) en fonction de la méthode d'interpolation pour la dernière time-slice. $N_L = 80, N = 4, N_{\Delta_t,\widehat{G}} = 20$, injection pour l'opérateur de restriction.

De manière générale, on observe un effet bénéfique de l'ordre d'interpolation, qui permet d'améliorer la convergence de PARAREAL aux petites échelles. Ainsi, les méthodes d'ordre élevé permettent de réduire considérablement l'erreur sur les petites échelles, et on peut même postuler une sorte de convergence vers l'erreur de $\hat{\mathcal{G}}$ seul lorsque l'ordre augmente (Fourier pour $\widehat{K} = 1$, Fig. 4.11a). Il est aussi intéressant de noter que pour $o_I = 7$, la deuxième itération de PARAREAL n'améliore pas la qualité de la solution pour les petites échelles, contrairement à ce que l'on observe pour les autres méthodes d'interpolation. On peut donc envisager l'existence d'une valeur optimale pour l'ordre d'interpolation, permettant d'obtenir la convergence de PARAREAL après une seule itération, qui semble ici être proche de la valeur $o_I = 7$

Cependant, l'augmentation de l'ordre ne permet pas de corriger l'erreur de diffusion commise par le solveur grossier dans les grandes échelles ($\kappa = 0.2\kappa_{max}$) pour la première itération (Fig. 4.11a). Cette erreur, induite par la discrétisation spatiale plus grossière de $\hat{\mathcal{G}}$, n'est corrigée que pour la deuxième itération (Fig. 4.11b), sous réserve que l'ordre d'interpolation soit suffisamment élevé.

Aussi, on remarque que la deuxième itération bénéficie majoritairement aux grandes échelles, contrairement aux petites échelles dont l'erreur sur l'énergie évolue peu. Ce comportement est confirmé par le Tab. 4.3, où l'erreur relative sur l'énergie e_{k_e} est en moyenne divisée par deux, tandis que la convergence est plus lente pour la dissipation. Enfin, c'est sans surprise que l'on observe une erreur relative plus importante pour la dissipation comparée à celle de l'énergie, ce qui est cohérent avec la meilleure capacité de $\hat{\mathcal{G}}$ à représenter les grandes échelles par rapport aux petites.

TABLE 4.3 – Influence de l'opérateur d'interpolation pour l'erreur relative sur k_e et ϵ pour PARAREAL avec grossissement spatial (N = 4, $N_{\Delta_t,\hat{G}} = 20$, injection pour l'opérateur de restriction).

	e_{k_e}		e_ϵ	
Interpolation	$\widehat{K}=1$	$\widehat{K}=2$	$ \widehat{K}=1$	$\widehat{K}=2$
Ordre 1	-6.6%	-4.5%	-17%	-12%
Ordre 3	-1.5%	-0.69%	-5.5%	-3.5%
Ordre 7	-0.85%	-0.44%	-2.8%	-2.2%
Fourier	-0.79%	-0.48%	-2.1%	-2.1%
$\hat{\mathcal{G}}$ seul	-0.83%		-2.7%	

Observation 4.3.3 — Pour le problème de THI, l'augmentation de l'ordre d'interpolation améliore le comportement de PARAREAL avec grossissement spatial pour l'ensemble des échelles de la turbulence. Cependant, l'erreur de diffusion induite par le solveur grossier dans les grandes échelles est exclusivement corrigée par l'augmentation du nombre d'itérations de PARAREAL.

Au vu des résultats précédents, on favorise pour la suite de l'étude l'utilisation de l'interpolation d'ordre 7 au sein de PARAREAL avec grossissement spatial.

4.3.3.d Influence du nombre de Reynolds Re_{λ}

On considère maintenant une augmentation du nombre de Reynolds du problème, effectuée en considérant des tailles de maillage de plus en plus grandes. Les Re_{λ_0} considérés pour chaque taille de maillage ($N_L \in \{80, 160, 320, 640\}$) sont déterminés en suivant la procédure détaillée en Sec. 4.3.2.a. D'un point de vue physique, l'augmentation du Re_{λ} a donc pour effet d'augmenter l'intervalle entre les grandes échelles et les plus petites, faisant apparaître au sein de l'écoulement des structures de plus en plus fines, comme représenté en Fig. 4.12. L'influence de Re_{λ} sur l'erreur relative pour le spectre énergétique est représentée en Fig. 4.13, et les erreurs relatives sur l'énergie et la dissipation sont données en Tab. 4.4. La même taille de *time-slice* est gardée pour cette analyse, et le pas de temps δ_t de \mathcal{F} est réduit par deux pour chaque augmentation de la résolution spatiale. On observe ainsi une diminution du ratio entre l'intervalle temporel d'une *time-slice* et le temps caractéristique des petites échelles ($\Delta_{TS}/\tau_{\eta,\phi}$, 8^{ème} colonne de Tab. 4.4).



FIGURE 4.12 – Influence de la montée en Reynolds sur la structure du champ turbulent. Coupes 2D du domaine et représentation de ρw pour les différentes solutions initiales pour PARAREAL.



FIGURE 4.13 – Influence de la montée en Reynolds sur l'erreur de PARAREAL. Erreur $e_{rel}(\kappa)$ (4.48) en fonction de Re_{λ} pour la dernière time-slice. $N = 4, N_{\Delta_t,\widehat{\mathcal{G}}} = 20$, injection et interpolation d'ordre 7 pour les opérateurs de transfert. L'erreur pour $\widehat{\mathcal{G}}$ seul est indiquée en tirets, avec les symboles correspondants.

On observe tout d'abord un comportement assez proche concernant l'erreur sur le spectre d'énergie cinétique pour l'ensemble des Re_{λ} , et ce tout particulièrement pour l'erreur de diffusion sur les grandes échelles ($\kappa = 0.2\kappa_{max}$, Fig. 4.13a). Celle-ci est par ailleurs corrigée de manière identique pour chaque Re_{λ} après la deuxième itération (Fig. 4.13b), ce que l'on retrouve dans le comportement de l'erreur relative sur l'énergie e_{k_e} en 8^{ème} et 9^{ème} colonne de Tab. 4.4.

On peut noter cependant une légère évolution particulière de l'erreur relative e_{rel} pour les échelles intermédiaires : l'augmentation de Re_{λ} induit une sur-estimation de l'énergie sur ces échelles à $\widehat{K} = 1$ ($\kappa \approx 0.4\kappa_{max}$), faisant "tendre" légèrement l'erreur de PARAREAL vers celle effectuée par $\widehat{\mathcal{G}}$ seul. Cette légère amplification est par ailleurs atténuée par la deuxième itération de PARAREAL.

Enfin, il est intéressant de se focaliser sur l'erreur de dissipation pour les différents Re_{λ} considérés, donnée dans les deux dernières colonnes de Tab. 4.4. Alors que l'erreur e_{ϵ} est réduite par la deuxième itération pour $Re_{\lambda_0} \in \{46, 124\}$, on observe que la tendance s'inverse pour $Re_{\lambda_0} \in \{322, 744\}$. Cette "divergence" de PARAREAL pour la dissipation peut s'expliquer par une erreur à la première itération caractérisée principalement par un équilibrage entre les gains et pertes d'énergie sur les différentes échelles. En effet, e_{ϵ} est obtenue à partir de valeurs intégrales de $\kappa^2 E(\kappa)$, et est par conséquent réduite grâce à l'amplification aux échelles intermédiaires apportée par l'augmentation de Re_{λ} . De ce fait, bien que e_{ϵ} diminue ave Re_{λ} pour $\widehat{K} = 1$, elle n'est pas nécessairement représentative d'une représentation correcte sur l'ensemble des échelles de la turbulence, comme

TABLE 4.4 – Influence du nombre de Reynolds Re_{λ} sur l'erreur relative pour k_e et ϵ de PARAREAL avec grossissement spatiale (N = 4, $N_{\Delta_t,\hat{\mathcal{G}}} = 20$, injection et interpolation d'ordre 7 pour les opérateurs de transfert). $Re_{\lambda_{\phi}}$ (Re_{λ_E}) est le nombre de Reynolds au début (à la fin) de la simulation avec PARAREAL. $\Delta_{TS}/\tau_{n,\phi}$ correspond au temps d'intervalle de time-slice, normalisé par τ_n évalué à t_{ϕ} .

							$ e_i$	k_e	6	2ϵ
N_L	Re_{λ_0}	$t_{\phi}/\tau_{\lambda,0}$	$t_{end}/ au_{\lambda,0}$	$Re_{\lambda_{\phi}}$	Re_{λ_E}	$\left \Delta_{TS} / \tau_{\eta,\phi} \right $	$ \widehat{K}=1$	$\widehat{K}=2$	$ \widehat{K}=1$	$\widehat{K}=2$
80	46	1.4	4.4	26	16	3.2	-0.85%	-0.44%	-2.8%	-2.2%
160	124	2.7	4.1	41	30	2.2	-0.68%	-0.37%	-2.0%	-1.7%
320	322	3.4	4.1	55	46	1.9	-0.60%	-0.34%	-1.5%	-1.6%
620	744	3.4	3.7	80	71	1.7	-0.41%	-0.24%	-0.9%	-1.3%

en témoigne la Fig. 4.13.

Observation 4.3.4 — Pour le problème de THI, l'augmentation du nombre de Reynolds (proportionnel au ratio des grandes et petites échelles turbulentes) a une faible influence sur la convergence de PARAREAL pour les premières itérations. Concernant le spectre d'énergie cinétique, l'augmentation du Reynolds fait "tendre" l'erreur de PARAREAL vers celle effectuée par le solveur grossier, cet effet étant cependant atténué par la deuxième itération.

4.3.3.e Influence de la taille des time-slices

On s'intéresse ici à l'influence de la taille des *time-slices* utilisées pour la simulation. Ce paramètre est non seulement important d'un point de vue des performances parallèles de l'algorithme (examinées en Sec. 4.1), mais peut également avoir un impact important sur la précision de la solution, comme il l'a été observé pour des problèmes plus simples en Chap. 3.

L'augmentation du nombre de Reynolds, étudiée précédemment, est ainsi utile pour étudier ce paramètre. En effet, à bas Reynolds, le nombre de pas de temps requis pour l'intégration temporelle de la décroissance de la THI est faible. L'utilisation de plus grandes *time-slices* n'a par conséquent pas d'intérêt particulier, du fait que l'énergie cinétique turbulente de la solution tombe à un niveau relativement faible, qui est de fait correctement représenté par $\hat{\mathcal{G}}$ seul. Pour des Reynolds plus élevés, l'utilisation d'un plus grand nombre de pas de temps est requise pour la simulation globale, et PARA-REAL peut donc être appliqué avec de plus grandes *time-slices*, tandis que le niveau d'énergie reste au-delà (du point de vue des contraintes DNS) des capacités de bonne résolution de $\hat{\mathcal{G}}$.

On étudie donc l'influence de $N_{\Delta_t,\widehat{\mathcal{G}}}$ sur l'erreur relative pour l'énergie après $\widehat{K} \in \{1,2\}$ itérations de PARAREAL avec grossissement spatial. La modification de ce paramètre changeant aussi le temps physique t_{end} , on décide de redémarrer l'algorithme après les \widehat{K} itérations un nombre suffisant de fois pour que t_{end} soit commun à l'ensemble de $N_{\Delta_t,\widehat{\mathcal{G}}}$ étudiés. On considère une simulation avec $Re_{\lambda_0} = 322$, avec $N_{\Delta_t,\widehat{\mathcal{G}}} \in \{10, 20, 40\}$, ce qui implique de redémarrer PARAREAL respectivement 3, 1 et 0 fois. Le même pas de temps δ_t est utilisé pour l'ensemble des simulations, avec N = 4, ce qui implique $t_{end} = 4.8\tau_{\lambda,0}$.

L'effet principal de la variation de $N_{\Delta_t,\widehat{\mathcal{G}}}$ sur e_{rel} est représenté en Fig. 4.14. On peut observer, de manière générale pour chaque itération, une réduction de l'erreur de diffusion pour les petites échelles, accompagnée de l'apparition d'une légère erreur d'amplification pour les échelles intermédiaires ($\kappa \approx 0.4\kappa_{max}$), qui se voit plus distinctement pour $\widehat{K} = 1$ (Fig. 4.14a). Ce comportement peut être comparé à celui observé avec l'augmentation du nombre de Reynolds en Sec. 4.3.3.d, rapprochant l'erreur de PARAREAL vers celle de $\widehat{\mathcal{G}}$ aux échelles intermédiaires. Ici encore, on observe que cette tendance est liée au ratio $\Delta_{TS}/\tau_{\eta,\phi}$ (2^{ème} colonne de Tab. 4.5), qui semble, tout comme la méthode d'interpolation, piloter l'erreur de PARAREAL sur une large gamme échelles. On attribue donc cet effet à l'impact de la restriction/interpolation sur les petites échelles, minimisé ainsi



FIGURE 4.14 – Influence de la taille des time-slices sur l'erreur de PARAREAL. Erreur $e_{rel}(\kappa)$ (4.48) en fonction de $N_{\Delta_t,\widehat{\mathcal{G}}}$ pour la dernière time-slice. $N_L = 320, N = 4, N_{\Delta_t,\widehat{\mathcal{G}}} = 20$, injection et interpolation d'ordre 7 pour les opérateurs de transfert.

par l'augmentation de $N_{\Delta_t,\widehat{\mathcal{G}}}$, qui permet ainsi de mieux représenter sur une *time-slice* l'évolution temporelle des petites échelles.

TABLE 4.5 – Influence de $N_{\Delta_t,\widehat{\mathcal{G}}}$ sur l'erreur relative pour k_e et ϵ après les premières itérations de PARAREAL avec grossissement spatial. ($Re_{\lambda_0} = 322$, injection et interpolation d'ordre 7 pour les opérateurs de transfert). $\Delta_{TS}/\tau_{\eta,\phi}$ correspond au temps d'intervalle de time-slice, normalisé par τ_{η} évalué à t_{ϕ} .

		e_{k_e}		e_{ϵ}	
$N_{\Delta_t,\hat{\mathcal{G}}}$	$\Delta_{TS}/\tau_{\eta,\phi}$	$\widehat{K} = 1$	$\widehat{K}=2$	$ \widehat{K}=1$	$\widehat{K}=2$
10	0.95	-0.24%	-0.002%	-3.3%	-1.8%
20	1.9	-0.36%	-0.09%	-2.7%	-2.0%
40	3.8	-0.49%	-0.14%	-3.4%	-2.5%
$\hat{\mathcal{G}}$ seul		-0.63%		-4.2%	

D'un point de vue statistique, l'augmentation de $N_{\Delta_t,\widehat{\mathcal{G}}}$ n'affecte pas significativement e_{ϵ} , qui semble être presque indépendant de la taille des *time-slices*. D'un autre côté, l'amplification observée pour $\kappa \approx 0.4\kappa_{max}$ a sans surprise un effet détériorant sur e_{k_e} , cet effet augmentant avec $N_{\Delta_t,\widehat{\mathcal{G}}}$. Cependant, on observe que les niveaux d'erreur pour l'énergie et la dissipation, résumés en Tab. 4.5, sont peu élevés de manière générale. On note aussi que certains niveaux d'erreur très faibles (*e.g.* e_{k_e} pour $\widehat{K} = 2$ et $N_{\Delta_t,\widehat{\mathcal{G}}} = 10$) bénéficient d'un effet de compensation associé à la métrique utilisée.

Observation 4.3.5 — Pour le problème de THI, l'augmentation de la taille des time-slices a un effet principalement bénéfique sur l'énergie des petites échelles, tandis qu'elle ajoute une plus grande diffusion sur les échelles intermédiaires. Cet effet peut contrebalancer d'éventuelles amplifications énergétiques dans les échelles intermédiaires, induites par le solveur grossier.

4.3.3.f Conclusions concernant l'analyse de convergence pour la décroissance de THI.

On synthétise donc ici l'ensemble des observations effectuées dans l'analyse effectuée en Sec. 4.3.3.c à Sec. 4.3.3.e, en les résumant en trois conclusions principales.

Opérateurs de transfert et taille des *time-slices*, les paramètres importants. Les opérateurs de transfert utilisés pour PARAREAL avec changement de grille ont un impact majeur sur la qualité de la solution. En particulier, il a été observé que l'utilisation d'une méthode d'interpolation d'ordre élevé est nécessaire pour minimiser l'erreur induite par le changement de grille sur les grandes échelles turbulentes, et permettre ainsi une bonne représentation de l'énergie des petites échelles. Aussi, l'utilisation de l'injection comme opérateur de restriction semble optimale.

Alors qu'elle influence peu les grandes échelles, l'augmentation de la taille des *time-slices* permet une meilleure prédiction de l'énergie aux petites échelles, tandis qu'elle induit une diffusion plus importante pour les échelles intermédiaires.

Un léger bénéfice de la deuxième itération de PARAREAL. De manière générale, la deuxième itération de PARAREAL avec grossissement spatial permet de corriger une atténuation énergétique vers les grandes échelles apportée par la discrétisation du solveur grossier. Cependant, son impact est particulièrement minimisé sur les petites échelles pour des ordres d'interpolation élevés ou des tailles de *time-slice* suffisamment grandes.

Une influence mineure de l'augmentation du nombre de Reynolds. L'augmentation du nombre de Reynolds, réalisée en considérant des problèmes de taille croissante, a un impact mineur sur la convergence de PARAREAL avec grossissement spatial. Cela justifie ainsi la mise à l'échelle de cette étude pour de bien plus grandes valeurs du nombre de Reynolds et de la taille du problème. On peut noter néanmoins que l'erreur sur le spectre énergétique de PARAREAL tend vers celle du solveur grossier avec l'augmentation du Reynolds, effet qui reste cependant lié à une diminution de la taille des *time-slices* par rapport au temps caractéristique des petites échelles.

4.4 Écoulement de Canal Turbulent

4.4.1 Description du problème

L'ensemble des notations, variables et grandeurs définies à partir d'ici sera utilisé exclusivement pour la Sec. 4.4, la plupart de ces définitions n'étant valides que dans le cadre du problème considéré. L'ensemble des définitions et propriétés présentées dans cette section provient d'une description plus complète donnée par [Pope, 2000, Sec. 9.2.1] et [Modesti et Pirozzoli, 2016], que l'on résumera ici pour introduire la description des résultats obtenus avec PARAREAL. Un glossaire regroupant l'ensemble des notations définies est disponible en Ap. D.

4.4.1.a Type d'écoulement et domaine considéré.

L'Écoulement de Canal Turbulent (ECT) est obtenu par la résolution des équations (4.25)-(4.27) sur un domaine tridimensionnel périodique dans les directions (x, z) et avec des conditions limites non périodiques pour la direction y (paroi rigide)¹⁵. Une représentation du domaine de calcul est donnée en Fig. 4.6a. Par la suite, on considérera que les points situés sur les parois ont comme coordonnée y = 0 ou y = 2h, avec h la demi-hauteur du canal. Une visualisation du champ de vitesse à un temps arbitraire est donnée en Fig. 4.15b.

Afin d'obtenir un écoulement moyen stationnaire dans la direction axiale (x), un terme source est ajouté à (4.26) et (4.27) pour compenser les pertes par frottement et maintenir le débit. Cela revient imposer un gradient de pression moyen constant et résoudre avec une pression fluctuante qui devient périodique, technique couramment utilisée pour ce type d'écoulement.

^{15.} En particulier, la discrétisation spatiale utilise une technique de type "miroir", qui permet d'appliquer les mêmes schémas de discrétisation dans la direction y près des parois et au milieu du canal.



(a) Domaine de calcul.

(b) Vitesse axiale de l'écoulement u, normalisée par la vitesse volumique axiale moyenne u_b , $Re_{\tau} = 590$.

FIGURE 4.15 – Problème d'Écoulement de Canal Turbulent.

De par la propriété d'homogénéité dans les directions x et z, et la symétrie du problème en y, son étude peut être réduite à celle des profils moyennés, qui s'écrivent :

$$\overline{f}(y) = \frac{1}{2L_xL_z} \left[\int_x \int_z f(x, y, z) dx dz + (\varsigma) \int_x \int_z f(x, 2h - y, z) dx dz \right] \text{ avec } y \in [0, h],$$
(4.51)

avec $\varsigma = 1$ et $\varsigma = -1$ pour des champs respectivement symétriques (*e.g.* u, w, u^2 , v^2 , w^2) et antisymétriques (*e.g.* v, uv). En particulier, on utilisera l'indice f_w pour désigner un champ moyenné à la paroi (*i.e.* $f_w = \overline{f}(0)$).

De par son aspect stationnaire et l'hypothèse d'ergodicité, on peut encore restreindre l'étude statistique du problème à une moyenne temporelle des variables, définie comme suit :

$$\langle f \rangle_t (..) = \frac{1}{N_{stat}} \sum_{i=0}^{N_{stat}-1} f(.., t_{stat} + i\Delta_{stat}),$$
 (4.52)

avec Δ_{stat} l'intervalle de temps utilisé pour stocker les N_{stat} échantillons statistiques, et t_{stat} le temps initial d'échantillonnage.

4.4.1.b Grandeurs caractéristiques de l'écoulement.

Dans un contexte de résolution des équations de Navier-Stokes compressibles, l'écoulement de canal turbulent peut être caractérisé par deux nombres sans dimension, statistiquement stationnaires, qui pilotent l'évolution de la turbulence :

• le nombre de Reynolds associé à la vitesse débitante :

$$Re_b = 2\frac{\rho_b u_b h}{\mu_w},\tag{4.53}$$

• le nombre de Mach associé à la vitesse débitante :

$$M_b = \frac{u_b}{c_w} = \frac{u_b}{\sqrt{\gamma r T_w}},\tag{4.54}$$

avec ρ_b et u_b des quantité volumiques calculées ainsi :

$$\rho_b = \frac{1}{2hL_x L_z} \int_V \rho dV, \tag{4.55}$$

$$u_b = \frac{1}{2\rho_b h L_x L_z} \int_V \rho u dV. \tag{4.56}$$

À partir de la vitesse u_b on peut définir l'échelle de temps associée à l'advection d'une particule fluide sur la longueur du canal :

$$T_c = \frac{L_y}{u_b}.\tag{4.57}$$

On peut également définir la vitesse de frottement à la paroi :

$$u_{\tau} = \sqrt{\frac{\tau_w}{\rho_w}},\tag{4.58}$$

elle même définie à partir du frottement à la paroi au_w , qui s'écrit :

$$\tau_w = \mu \frac{\partial u}{\partial y} \Big|_{y=0}.$$
(4.59)

Cette échelle de vitesse, associée à la viscosité, permet de définir une échelle de longueur :

$$\delta^+ = \frac{\mu_w}{\rho_w u_\tau}.\tag{4.60}$$

Le jeu d'échelle (u_{τ}, δ^+) est couramment utilisé sous la dénomination d'"unités de paroi" et permet de définir un nouveau nombre de Reynolds :

$$Re_{\tau} = \frac{h}{\delta^+},\tag{4.61}$$

qui permet de caractériser la turbulence se développant au voisinage de la paroi. On peut noter que par définition, celui-ci est proportionnel à la racine carrée du frottement à la paroi τ_w .

4.4.1.c Profils de vitesse moyenne et tensions de Reynolds.

Profil moyen de vitesse axiale. Dans un écoulement d'ECT, la première variable d'intérêt est le profil moyen $U = \langle \overline{u} \rangle_t$, généralement normalisé en unités de paroi :

$$U^{+}(y^{+}) = \frac{\langle \overline{u} \rangle_{t} (y^{+})}{u_{\tau}}, \qquad (4.62)$$

avec $y^+ = y/\delta^+$, et u_τ calculé sur l'ensemble des champs échantillons moyennés en temps. Ces



(a) Normalisation en unités de paroi.

(b) Normalisation en unités volumiques.

FIGURE 4.16 – Profils moyens de vitesse axiale pour des écoulements de canal turbulent à différents Re_{τ} , d'après les données de [Moser et al., 1999]. On représente en tirets-pointillés les lois caractéristiques du profil normalisé en unités de paroi : loi linéaire près de la paroi $(y^+ < 5) U^+ = y^+$ et loi logarithmique en milieu de canal $U^+ = \frac{1}{\kappa} \ln(y^+) + C^+$, avec $\kappa \simeq 0.41$ la constante de Von Kármán, et $C^+ \simeq 5.0$.

profils sont donnés pour différents Re_{τ} à partir de données DNS disponibles dans la littérature en

Fig. 4.16. On peut y voir l'effet de l'augmentation du nombre de Reynolds, qui provoque la réduction de l'épaisseur de la couche visqueuse au voisinage de la paroi (Fig. 4.16b). Enfin on peut observer en Fig. 4.16a l'intérêt de la normalisation en unités de paroi, qui permet d'adimensionner les effets visqueux en proche paroi. On peut également distinguer deux zones particulières de l'écoulement : la **couche interne** proche de la paroi (y+ < 10), où les effets visqueux prédominent, et l'écoulement dans la **couche externe** (y > 0.1h, *i.e.* $y^+ > 60$ dans le cas où $Re_{\tau} = 590$).

Tensions de Reynolds. Ces dernières permettent d'obtenir une représentation plus fine du comportement de la turbulence dans un ECT. Elles apparaissent explicitement dans les équations de Navier-Stokes moyennées, dans la décomposition du terme d'advection. On les calcule en utilisant le profil de tension de Reynolds pour deux variables f et g, défini ainsi :

$$(fg)' = <\overline{f'g'} >_t \tag{4.63}$$

avec $f' = f - \overline{f}$ la fluctuation de f (et de manière similaire pour g).

Dans le cadre de cette étude, on se restreint à l'étude du premier terme diagonal des tensions de Reynolds :

$$u^{\prime 2} = \langle \overline{(u^2)} \rangle_t - \langle \overline{(u)} \rangle_t^2, \tag{4.64}$$

et au terme croisé :

$$(uv)' = \langle \overline{uv} \rangle_t - \langle \overline{u} \rangle_t \langle \overline{v} \rangle_t . \tag{4.65}$$

Ces deux termes sont généralement normalisés en unités de paroi pour obtenir $u'^+ = \sqrt{u'^2}/u_{\tau}$ et $(uv)'^+ = (uv)'/u_{\tau}^2$. Des représentations de ces profils en fonction de Re_{τ} sont données en Fig. 4.17 à partir de la base de données de [Moser *et al.*, 1999]. Trois zones caractéristiques peuvent être



(a) Coordonnée normalisée en unité de paroi.

(b) Coordonnée normalisée en unité volumique.

FIGURE 4.17 – Profils de tensions de Reynolds pour des écoulements de canal turbulents, d'après les données de [Moser et al., 1999]. Les mêmes Re_{τ} sont représentés que pour Fig 4.16 : $Re_{\tau} = 180$ (pointillés), $Re_{\tau} = 395$ (tirets), $Re_{\tau} = 590$ (lignes pleines).

distinguées pour ces profils :

- 1. une zone proche de la paroi où les effets visqueux prédominent (valeurs faibles pour u'^+ et $(uv)'^+$, couche interne),
- 2. un pic de u'^+ dans la zone de production maximale de turbulence (10 < y + < 20),
- 3. une décroissance de u'^+ et $(uv)'^+$ au fur et à mesure que l'on s'éloigne de la zone de production maximum (couche externe).

La normalisation de y en unité de paroi (y^+ , représentation logarithmique) permet ainsi de distinguer plus en détails l'évolution de ces profils dans la couche interne. **Spectres d'énergie cinétique turbulente.** Afin d'analyser le contenu spectral aux différentes échelles de l'écoulement, il est possible d'extraire la densité spectrale d'énergie cinétique turbulente moyennée dans la direction transverse pour une coordonnée y donnée. Plus d'informations quant à la procédure permettant leur calcul sont données en Ap. E.

4.4.2 Définition du solveur fin et grossier pour Parareal

Afin d'étudier la convergence de PARAREAL pour le problème d'ECT, on choisit de reproduire la DNS effectuée par [Moser et al., 1999] pour $Re_{\tau} = 590$. On présente donc en Sec. 4.4.2.a le maillage utilisé pour le solveur fin, ainsi que la grille grossière construite à partir de ce dernier. Puis, en Sec. 4.4.2.b on détaille la mise en place de PARAREAL pour l'analyse de convergence.

4.4.2.a Caractéristiques du maillage spatial

Maillage spatial pour \mathcal{F} . On considère le même domaine spatial que [Moser *et al.*, 1999], *i.e.* $L_x = 2L_z = 2\pi$. Le même nombre de points de maillage $N_{L_x} = N_{L_z} = 420$ est utilisé dans les directions axiale et transverse, ce qui induit $\delta_x^+ = 8.8 = 2\delta_z^+$, où $\delta_x^+ = \delta_x/\delta^+$. Pour la direction y, on fixe $N_{L_y} = 255$, en utilisant une loi de répartition non uniforme, décrite en Ap. F. Celle-ci permet d'obtenir une taille de maille $\delta_{y,w}^+ = 0.8$ près de la paroi et $\delta_{y,c}^+ = 9.1$ en milieu de canal. Le pas de temps est fixé à une valeur constante $\delta_t = 9.54 \ 10^{-5}T_c$, afin d'atteindre un *CFL* proche de la limite de stabilité pour l'intégration temporelle explicite effectuée avec la méthode RKC4 (*CFL* $\simeq 1.55$).

Le nombre de points de discrétisation est donc légèrement supérieur à celui utilisé par [Moser *et al.*, 1999], en raison de la différence des schémas numériques utilisés (discrétisation spectrale pour [Moser *et al.*, 1999]). HYBRID utilisant quant à lui des méthodes différences finies d'ordre 6, le maillage spatial considéré pour \mathcal{F} est donc proche d'une résolution équivalente. Une validation plus détaillée des résultats obtenus avec le maillage fin est donnée en Ap. F, par comparaison des profils de vitesse moyenne et des fluctuations de \mathcal{F} , ainsi que des spectres de turbulence, vis-à-vis des données DNS de [Moser *et al.*, 1999] obtenues pour des écoulements de fluide incompressible.

Maillage spatial pour $\hat{\mathcal{G}}$. On utilise la même stratégie de grossissement qu'en Sec. 4.3 pour les directions x et z (*vertex coarsening*). Pour la direction y cependant, le maillage non-uniforme nous pousse à utiliser une forme particulière de grossissement, représentée en Fig. 4.18. Cette définition du maillage grossier est rendue possible sous la condition que $(N_{Ly} - 1)/2$ soit un nombre impair, ce qui est le cas pour le maillage fin considéré.



FIGURE 4.18 – Répartition des points de maillages fin et grossier dans la direction y pour $N_{Ly} = 11$, conditions aux bords de type paroi.

Opérateurs de transfert Pour l'ensemble des applications de PARAREAL dans la suite de l'étude, on utilise l'injection comme opérateur de restriction. Une méthode d'interpolation polynômiale d'ordre variable est utilisée pour le passage de la grille grossière à la grille fine, plus d'informations sont données en Ap. C.

4.4.2.b Solution initiale et mise en place de PARAREAL.

On décide d'appliquer PARAREAL pour la simulation d'un champ turbulent complètement développé, *i.e.* statistiquement stationnaire. Comme il est peu envisageable de générer des champs initiaux ayant un caractère turbulent réaliste, l'application de PARAREAL se fait en deux phases.

Obtention d'une solution stationnaire. On génère un état initial à partir d'un profil laminaire, perturbé par un signal aléatoire de faible amplitude. La viscosité de référence μ_{ref} est fixée afin d'obtenir une valeur cible pour $Re_b = 21842$ (correspondant à un écoulement de canal avec $Re_{\tau} = 590$). Puis, l'intégration temporelle est effectuée pour simuler l'évolution de l'écoulement, qui connait dans un premier temps une évolution transitoire. Durant cette période transitoire T_{lt} (passage d'un état laminaire à turbulent), la turbulence se développe au sein de l'écoulement, avant d'atteindre un état où les grandeurs sont statistiquement stationnaires. En particulier, Re_{τ} évolue pour se rapprocher d'un état statistiquement stationnaire, dans lequel il est proche de la valeur cible ($Re_{\tau} = 590$). On donne son évolution en Fig. 4.19a. Afin de se débarrasser d'un maximum d'effets liés à la période transitoire, on simule l'écoulement de t = 0 à $t = 19T_c$. Cette étape n'ayant pas d'intérêt physique particulier, les calculs sont effectués avec $\hat{\mathcal{G}}$ seul, ce qui minimise le coût de calcul pour cette période.

Application de PARAREAL On applique PARAREAL au problème d'ECT avec comme champ initial la solution obtenue avec $\hat{\mathcal{G}}$ à la fin de la période transitoire. Afin d'accumuler un nombre suffisant d'échantillons statistiques, on intègre le problème sur une fenêtre temporelle de taille $T = 7.6T_c$ (la validation des résultats statistiques obtenus sur cette fenêtre temporelle est donnée en Ap. F).





FIGURE 4.19 – Evolution de Re_{τ} en fonction du temps pour les différentes phases de simulation.

En Fig. 4.19b, on donne l'évolution de Re_{τ} durant cette phase. On peut voir que celui-ci oscille autour d'une valeur moyenne constante pour les deux solveurs considérés. Cet intervalle représente un grand nombre de pas de temps du solveur fin (80000), qui n'est jamais couvert durant l'étude par une application *unique* de PARAREAL. C'est pourquoi, selon le nombre et la taille des *time-slices* considérées, on redémarre l'algorithme N_r fois pour couvrir l'ensemble de l'intervalle temporel de la phase d'application de PARAREAL.

Afin de calculer les profils de vitesse et de tension de Reynolds moyennés en temps, on effectue une moyenne statistique sur l'ensemble des solution obtenues à la fin de chaque time-slice, plus la solution initiale. Ainsi, le nombre d'échantillons utilisé est donc $N_rN + 1$ ¹⁶.

^{16.} Comme on peut le voir en Fig. 4.19b, le solveur fin présente une courte phase transitoire durant laquelle la solution initiale, calculée avec $\hat{\mathcal{G}}$, se reconditionne. C'est pourquoi on considère l'application de PARAREAL sur un domaine temporel suffisamment long, afin de minimiser l'influence de cette phase transitoire sur les résultats statistiques.

Erreur de $\hat{\mathcal{G}}$ **seul.** Bien que cet aspect soit analysé plus en détails par la suite, on peut déjà observer sur la Fig. 4.19b l'erreur de $\hat{\mathcal{G}}$ seul qui s'exprime par une sur-évaluation du Re_{τ} , *i.e.* du frottement τ_w à la paroi. Celle-ci est due à la résolution spatiale du maillage grossier, qui ne peut représenter correctement les structures turbulentes de l'écoulement proche paroi. Même s'il peut paraître faible, ce niveau d'erreur n'est pas satisfaisant dans le cadre d'une DNS.

4.4.3 Analyse de convergence de PARAREAL pour l'Écoulement de Canal Turbulent

Dans cette section, on analyse la convergence de PARAREAL avec grossissement spatial pour le problème d'ECT à $Re_{\tau} = 590$, ainsi que l'influence des différents paramètres de l'algorithme. Afin de rester cohérent avec le reste de l'étude, on fixe le nombre de time-slices à N = 4. Au sein d'Hybrid, on choisit d'utiliser des schémas de discrétisation spatiale d'ordre 6, et la méthode RKC4 pour l'intégration temporelle. Pour l'ensemble des simulations, on fixe $M_b = 0.3$. Ces dernières ont été effectuées sur le système de calcul EOS. Les temps de restitution pour une itération de PARAREAL avec grossissement spatial, pour le problème considéré, se situent autour d'une trentaine d'heures de calcul avec une parallélisation spatiale sur 320 cœurs.

Tout d'abord, on introduit l'analyse de convergence avec un premier exemple en Sec. 4.4.3.a. Puis, en Sec. 4.4.3.b on définit les grandeurs nous permettant d'évaluer la qualité de la solution. Ces derniers sont enfin utilisés pour l'étude de l'influence des paramètres principaux sur l'utilisation de PARAREAL en Sec. 4.4.3.d, Sec. 4.4.3.e et Sec. 4.4.3.f.

Note quant à la normalisation des résultats. Pour l'ensemble de cette étude, on choisit de considérer les profils moyens de vitesse et de tension de Reynolds normalisés en unités volumiques, *i.e.* U/u_b , u'^2/u_b^2 et $(uv)'/u_b^2$, avec u_b commun à l'ensemble des résultats. Ceci nous permet ainsi de comparer chaque solution à un niveau comparable, puisque les unités de paroi dépendent de la solution. Cependant, on conserve la possibilité d'utiliser la normalisation y^+ (associée à une représentation logarithmique), en se basant sur la valeur de δ^+ obtenue avec \mathcal{F} , afin de distinguer la couche interne visqueuse de la couche externe.

4.4.3.a Un premier aperçu de la convergence de PARAREAL

On considère ici l'application de PARAREAL sur le problème d'ECT, et on analyse la précision de la première itération sur les profils de vitesse moyenne et tensions de Reynolds. En s'appuyant sur les premiers résultats de convergence de PARAREAL pour le problème de THI (*cf.* Sec. 4.3), on choisit d'utiliser une méthode d'ordre 7 pour l'opérateur d'interpolation, et on fixe $N_{\Delta_t,\widehat{\mathcal{G}}} = 50$. Les profils de U, u' et (uv)' sont représentés en Fig.4.20 pour la première itération de PARAREAL, avec ceux du solveur fin et de $\widehat{\mathcal{G}}$ seul.

En premier lieu, on observe après la première itération de PARAREAL une légère sous-estimation du profil pour U et $y^+ > 10$, comparée à la solution fine, tandis qu'une bonne concordance est obtenue pour de faibles valeurs de y^+ . Aussi, bien que l'on puisse distinguer une légère amélioration pour la solution PARAREAL vis-à-vis de la solution obtenue avec $\hat{\mathcal{G}}$ seul (autour de $y^+ = 100$), les deux profils restent globalement très comparables. C'est en observant les variables représentant des caractéristiques plus fines de la turbulence, que l'on peut observer des différences plus nettes entre les solutions. En Fig. 4.20b et Fig. 4.20c, on représente les profils obtenus pour u'^2 et (uv)'respectivement. Pour ces deux grandeurs, la convergence de PARAREAL dépend particulièrement de la plage de coordonnée y^+ considérée. On distingue une convergence différente, selon que l'on considère l'écoulement de couche interne $(0 < y^+ < 60,$ *i.e.*0 < <math>y/h < 0.1) ou l'écoulement de couche externe $(60 < y^+ < 590,$ *i.e.*<math>0.1 < y/h < 1).

Pour la couche interne, la première itération de PARAREAL permet d'obtenir une meilleure estimation de u' et -(uv)', comparée à celle de $\widehat{\mathcal{G}}$ seul. Cela se distingue en particulier pour le pic



FIGURE 4.20 – Évolution des profils moyens de vitesse axiale et tensions de Reynolds pour la première itération PARAREAL. $N_{\Delta_t,\widehat{G}} = 50$, N = 4, $N_r = 200$, interpolation d'ordre 7.

de u'^2 vers $10 < y^+ < 20$. En revanche pour l'écoulement de couche externe ($y^+ > 60$), on peut observer une sur-estimation des tensions de Reynolds, à la fois pour $\hat{\mathcal{G}}$ et pour la solution de PARA-REAL avec $\hat{K} = 1$. Finalement, on observe pour l'ensemble des profils d'erreur relativement faibles. Afin de pouvoir les examiner plus en détails, on définit donc dans la section suivante les grandeurs permettant de les observer plus finement.

Observation 4.4.1 — Pour l'écoulement de canal turbulent, la qualité de la solution obtenue après une itération de PARAREAL n'est pas la même selon la zone d'écoulement considérée. En particulier, PARAREAL semble plus apte à résoudre la turbulence pour l'écoulement de la couche interne que celui de la couche externe.

4.4.3.b Observation de l'erreur pour les profils moyennés

Afin de mieux quantifier la qualité des solutions obtenues avec PARAREAL, on définit l'erreur relative pour un profil moyen donné, définie par exemple pour U:

$$e_{rel,U}(y) = \frac{\widetilde{U}(y) - U_{\mathcal{F}}(y)}{U_{\mathcal{F}}(y)},\tag{4.66}$$

où \tilde{U} correspond au profil moyen de la solution approchée (avec $\hat{\mathcal{G}}$ ou PARAREAL) et $U_{\mathcal{F}}$ celui de la solution de référence calculée uniquement avec \mathcal{F} . La coordonnée y peut être normalisée en unité

de paroi (y+), ou par la demi-hauteur du canal (y/h), selon la représentation souhaitée. Alors qu'une valeur négative pour l'erreur relative indique une sous-estimation pour la grandeur observée, une valeur positive témoigne en revanche d'une sur-estimation. Une même définition de l'erreur relative est utilisée pour u'^2 et (uv)'. À la différence des critères d'erreur définis pour la THI (*cf.* SEc. 4.3.3.b), l'erreur n'est ici pas basée sur un contenu spectral mais intégrée sur l'ensemble du spectre, par soucis de simplicité.

4.4.3.c Surestimation énergétique de l'écoulement turbulent pour la couche externe

On étudie ici l'impact de PARAREAL sur l'écoulement de couche externe, en se focalisant sur l'estimation des tensions de Reynolds. A cet effet, on représente l'évolution de l'erreur relative par rapport au solveur fin pour u'^2 et -(uv)' en fonction des itérations en Fig. 4.21, en normalisant y par la demi-hauteur du canal.

On remarque en premier lieu que pour certaines plages de y, effectuer plusieurs itérations de PARAREAL ne corrige pas l'erreur faite dans l'estimation des tensions de Reynolds. C'est le cas pour u'^2 avec $y/h \in [0.2, 1]$ et pour -(uv)' avec $y/h \in [0.2, 0.4]$. Bien que l'on observe une nette amélioration de l'erreur au centre du canal pour -(uv)' après la deuxième itération, les niveaux d'erreur de PARAREAL pour les tensions de Reynolds restent globalement supérieurs à ceux du solveur $\hat{\mathcal{G}}$ seul pour y > 0.1h. Cependant, cette sur-estimation de u'^2 et (uv)' dans la couche externe est aussi présente pour $\hat{\mathcal{G}}$, ce qui invite à une étude plus approfondie.



FIGURE 4.21 – Sur-estimation par $\widehat{\mathcal{G}}$ et PARAREAL de l'intensité de la turbulence dans la couche externe. Erreur relative (4.66) pour u' et -(uv)' en fonction des itérations. $N_{\Delta_t,\widehat{\mathcal{G}}} = 50, N = 4, N_r = 200,$ interpolation d'ordre 7.

D'un point de vue physique, la sur-estimation des tensions de Reynolds est induite par une amplification de l'énergie cinétique turbulente sur les basses fréquences. On peut l'observer sur les spectres d'énergie cinétique turbulente moyens, extraits dans une coupe de l'écoulement de la couche externe (y = 0.6h), donnés en Fig. 4.22. Que ce soit pour E_{uu} en Fig. 4.22a (représentatif de u'^2) ou pour E_{vv} en Fig. 4.22b (représentatif avec E_{uu} de -(uv)'), on observe un pic énergétique pour $\hat{\mathcal{G}}$ par rapport à \mathcal{F} sur le même mode à basse fréquence ($\kappa_z = 2$). Ce mode est amplifié par la première itération de PARAREAL, et bien que la deuxième itération permette de rabaisser son amplitude au même niveau que celui de $\hat{\mathcal{G}}$, un second mode est alors amplifié à son tour ($\kappa_z = 1$).

Le défaut de convergence dans cette zone n'est donc pas attribué à PARAREAL, mais plutôt au comportement du solveur $\hat{\mathcal{G}}$. En effet, celui-ci semble introduire dans l'algorithme une amplification de l'énergie cinétique turbulente aux basses fréquences spatiales, que PARAREAL ne peut atténuer au cours de ses premières itérations (il a même tendance au contraire à amplifier cette erreur). L'erreur de PARAREAL est ainsi plus importante que celle de $\hat{\mathcal{G}}$ seul pour une bonne partie du domaine de



FIGURE 4.22 – Amplification des modes basse fréquence dans la couche externe en fonction des itérations PARAREAL. Densité spectrale d'énergie turbulente moyennée la direction transverse en fonction de \widehat{K} , pour y = 0.6h. $N_{\Delta_r,\widehat{G}} = 50$, N = 4, $N_r = 200$, interpolation d'ordre 7.

simulation, il apparaît donc nécessaire de rechercher une solution curative. Comme on l'a déjà remarqué pour le problème de THI, la méthode d'interpolation ainsi que la taille des *time-slices* sont deux paramètres de l'algorithme qui peuvent avoir un impact non négligeable sur l'amplification ou l'amortissement de l'énergie associées aux échelles de l'écoulement. C'est pourquoi on s'attache dans les deux sections suivantes à étudier l'influence de ces deux paramètres sur la convergence de l'algorithme.

Observation 4.4.2 — Pour l'écoulement de couche externe, la convergence de PARAREAL souffre d'une amplification d'un mode basse fréquence apporté par le solveur grossier. Cela induit une sur-estimation de l'intensité de la turbulence, que PARAREAL n'arrive pas à corriger en quelques itérations.

4.4.3.d Influence de l'ordre d'interpolation

On s'intéresse ici à l'impact de la méthode d'interpolation sur la solution obtenue avec PARA-REAL. À cet effet on considère différents ordres d'interpolation $o_I \in \{1, 3, 5, 7\}$, et la taille des *time-slices* est gardée constante avec $N_{\Delta_t,\widehat{\mathcal{G}}} = 50$. On représente en Fig. 4.23 l'erreur relative pour U, u'^2 et -(uv)' après une itération de PARAREAL. Enfin, on utilise la normalisation de y en unité de paroi, afin de distinguer plus facilement les niveaux d'erreur pour l'écoulement de couche limite.

En premier lieu, on observe sans surprise les conséquences particulièrement dégradantes de l'utilisation de l'interpolation linéaire ($o_I = 1$) sur la qualité de la solution, et ce pour l'ensemble des solutions considérées. L'augmentation de l'ordre d'interpolation réduit considérablement l'erreur, et permet notamment de retrouver pour U des niveaux d'erreur bien en deçà de celui de $\hat{\mathcal{G}}$ seul et ce, tout particulièrement pour l'écoulement de couche interne ($y^+ < 10$).

Cependant, on observe que lorsque l'ordre d'interpolation augmente, l'erreur de PARAREAL se rapproche de celle effectuée par $\hat{\mathcal{G}}$ seul pour l'écoulement dans la couche interne ($y^+ < 10$), et ce pour l'ensemble des grandeurs représentées. On peut expliquer ce comportement en regardant les spectres d'énergie cinétique turbulente pour l'écoulement au voisinage de la paroi, représentés en Fig. 4.24. On peut y observer que $\hat{\mathcal{G}}$ induit de manière générale une amplification sur l'ensemble des échelles vis-à-vis de la solution de référence, qui induit donc la sur-estimation observée précédemment sur les profils en Fig. 4.23. Pour l'interpolation d'ordre 1, les échelles concernées par cette amplification sont totalement atténuées, ce qui explique la sous-estimation des tensions de Reynolds dans cette zone. L'augmentation de o_I réduit cette atténuation, au point de reproduire partiellement l'amplification de $\hat{\mathcal{G}}$ pour les ordres les plus élevés.



FIGURE 4.23 – Influence de la méthode d'interpolation sur l'erreur de PARAREAL. Erreur relative pour U, u'^2 et -(uv)' en fonction de l'ordre d'interpolation. $\widehat{K} = 1, N_{\Delta_t,\widehat{\mathcal{G}}} = 50, N = 4, N_r = 200.$

Si l'on se focalise désormais sur les erreurs relatives en Fig. 4.23 dans la couche externe $(y^+ > 60)$, on peut voir que l'ordre d'interpolation a un impact tout aussi important. L'utilisation d'une méthode d'ordre élevé $(o_I = 7)$ ou faible $(o_I = 1)$ induit une sur-estimation pour (uv)' plus importante que pour $\hat{\mathcal{G}}$, tandis que l'erreur est beaucoup plus faible pour $o_I \in \{3, 5\}$. Pour u'^2 , l'évolution de l'erreur est moins perceptible, cependant on observe également une meilleure estimation par PARAREAL pour les ordres d'interpolation "intermédiaires".

De manière générale, l'interpolation induit un amortissement sur les échelles représentées à la fois par $\hat{\mathcal{G}}$ et \mathcal{F} , qui diminue avec l'ordre de la méthode utilisée. Cet effet peut ainsi, pour certains ordres, contrebalancer l'amplification apportée par $\hat{\mathcal{G}}$, à la fois pour l'écoulement de couche interne et celui de la couche externe, et améliorer significativement l'estimation des profils moyens de vitesse et d'intensité turbulente de PARAREAL, et ce dès la première itération.

Il est donc possible d'envisager comme solution curative l'utilisation d'un ordre d'interpolation choisi de manière optimale afin d'apporter le taux de dissipation adapté pour les échelles amplifiées. Cette approche a cependant un inconvénient majeur : la baisse de l'ordre d'interpolation a aussi pour effet d'induire une plus grande amplification des basses fréquences, induite originalement par le changement de grille. C'est notamment ce qui provoque l'apparition d'oscillations pour l'erreur relative sur u'^2 que l'on peut voir dans la couche externe (*cf.* Fig. 4.23b). De ce fait, une stratégie conduisant à réduire l'ordre d'interpolation ne semble pas satisfaisante, et nous pousse à étudier l'influence de la taille des *time-slices*.



FIGURE 4.24 – Impact de l'ordre d'interpolation sur l'énergie cinétique turbulente près de la paroi. Densité spectrale d'énergie turbulente moyennés dans la direction transverse en fonction de o_I , pour $y^+ = 3.7$. $\widehat{K} = 1, N_{\Delta_I,\widehat{G}} = 50, N = 4, N_r = 200.$

Observation 4.4.3 — Pour l'écoulement de canal turbulent, la méthode d'interpolation influe grandement sur la qualité du profil de vitesse moyenne et des tensions de Reynolds estimés par PARAREAL. Alors que des ordres d'interpolation ni trop faibles, ni trop élevés, permettent de réduire dans une certaine mesure l'erreur de PARAREAL par rapport à celle de $\hat{\mathcal{G}}$, leur augmentation fait tendre l'erreur de parallélisation vers celle du solveur grossier pour l'écoulement de la couche interne, et amplifie la sur-estimation de l'intensité de la turbulence dans la couche externe. Aussi, l'utilisation d'une interpolation linéaire induit un niveau d'erreur de la solution de PARAREAL bien au-delà de celui pour $\hat{\mathcal{G}}$ seul.

4.4.3.e Influence de la taille des time-slices

On s'intéresse désormais à l'impact de la taille des time-slices sur la solution obtenue avec PARAREAL. On considère différentes valeurs pour $N_{\Delta_t,\widehat{\mathcal{G}}} \in \{50, 100, 200, 400\}$, ce qui implique de redémarrer PARAREAL avec $N_r \in \{200, 100, 50, 25\}$ respectivement. Une interpolation d'ordre 7 est utilisée pour l'ensemble des simulations. On représente en Fig. 4.25 l'erreur relative pour U, u'^2 et -(uv)' après la première itération de PARAREAL. La normalisation des abscisses en unités de paroi (y^+) est utilisée.

En premier lieu, on observe que l'augmentation des *time-slices* va de manière générale déplacer les niveaux d'erreur pour la couche interne vers des valeurs négatives. Cet effet est particulièrement visible pour U en Fig. 4.25 : alors que l'erreur de sur-estimation proche de la paroi est réduite avec l'augmentation de $N_{\Delta_t,\widehat{g}}$, on voit apparaître simultanément une sous-estimation de plus en plus importante pour les zones où la valeur du profil était originalement sous-estimée ($10 < y^+ < 100$, zone dite *"buffer"*).

Cependant, l'augmentation des *time-slices* a aussi un effet bénéfique sur les tensions de Reynolds, à la fois dans les couches interne et externe. En particulier, la sur-estimation de u'^2 et (uv)' est significativement réduite par l'augmentation de $N_{\Delta_t,\widehat{\mathcal{G}}}$, ce qui permet d'obtenir des niveaux d'erreur globalement moins importants que ceux de $\widehat{\mathcal{G}}$ seul, pouvant être observés avec u'^2 pour $N_{\Delta_t,\widehat{\mathcal{G}}} = 200$ en Fig. 4.25b.

Néanmoins, l'effet bénéfique de l'augmentation de la taille des *time-slices* semble plafonner au delà d'une certaine valeur. En effet, pour la plus grande valeur de $N_{\Delta_t,\widehat{\mathcal{G}}} = 400$, on observe non seulement une augmentation de l'erreur pour u'^2 dans la couche externe, comparée à des plus petites valeurs pour $N_{\Delta_t,\widehat{\mathcal{G}}}$, mais aussi dans la couche interne pour U. Ceci laisse donc envisager l'existence



FIGURE 4.25 – Influence de la taille des time-slices sur l'erreur de PARAREAL. Erreur relative pour U, u'^2 et -(uv)' en fonction de $N_{\Lambda_+ \widehat{G}}$. $\widehat{K} = 1$, N = 4, interpolation d'ordre 7.

d'une taille de *time-slice* optimale permettant de minimiser l'erreur de PARAREAL.

L'augmentation de la taille des *time-slices* peut donc être considérée comme une solution curative pour contrebalancer les amplifications apportées par $\hat{\mathcal{G}}$, à la fois dans la couche interne et la couche externe. Bien qu'aucune configuration optimale n'ait été déterminée, nous avons pu identifier une configuration particulière de PARAREAL permettant d'obtenir une bonne représentation de la turbulence pour les différentes zones d'écoulement ($\widehat{K} = 1$, $o_I = 7$, $N_{\Delta_t,\widehat{\mathcal{G}}} = 200$). En particulier, cette configuration permet d'obtenir une bonne estimation du profil de vitesse dans la couche interne (moins de 0.5% d'erreur), et donc du frottement à la paroi, et de réduire par deux l'erreur d'amplification sur u'^2 du solveur $\widehat{\mathcal{G}}$ dans la couche externe.

Observation 4.4.4 — Pour l'écoulement de canal turbulent, l'augmentation de la taille de time-slice permet de réduire l'erreur de PARAREAL dans la plupart des zones d'écoulement. En particulier, cet effet contribue dans une certaine mesure à contrebalancer l'amplification de l'intensité de la turbulence apportée par $\hat{\mathcal{G}}$, que ce soit pour l'écoulement de la couche interne ou celui de la couche externe. L'effet bénéfique de l'augmentation de la taille des time-slices semble cependant limité à une valeur seuil, au delà de laquelle la solution de PARAREAL commence à se dégrader dans certaines zones de l'écoulement.

4.4.3.f Apport de la deuxième itération de PARAREAL

Dans les deux sections précédentes, on s'est restreint à l'utilisation d'une seule itération pour PARAREAL avec grossissement spatial. On considère donc ici l'utilisation d'une deuxième itération, afin d'étudier le bénéfice sur la qualité de la solution. On considère successivement l'utilisation d'une interpolation d'ordre 3 et d'ordre 7, tandis que la taille des *time-slices* est gardée constante avec $N_{\Delta_t,\widehat{\mathcal{G}}} = 50$. On représente en Fig. 4.26 les profils d'erreur relative pour u'^2 en fonction des itérations, et ce pour l'utilisation des deux méthodes d'interpolation différentes.



FIGURE 4.26 – Mise en évidence du faible bénéfice de la deuxième itération. Erreur relative pour u'^+ en fonction de \widehat{K} , pour deux méthodes d'interpolation différentes. $N_{\Delta_t,\widehat{G}} = 50$, N = 4, $N_r = 200$.

En premier lieu, on observe que l'itération supplémentaire ($\hat{K} = 2$) permet une correction effective pour le champ de vitesse dans la couche interne, tandis qu'elle ne semble pas particulièrement bénéfique à l'écoulement dans la couche externe, et ce dès $y^+ > 10$. De plus, on observe que le gain de précision obtenu en changeant l'ordre d'interpolation (de 7 à 3) est bien plus important que celui obtenu en effectuant la deuxième itération dans les deux configurations étudiées.

Ces résultats mettent donc en évidence l'intérêt limité d'exécuter deux itérations pour cette formulation de l'algorithme. À noter qu'ici, seul un changement de l'ordre d'interpolation a été considéré dans l'optique d'améliorer la qualité de la solution. Or, au vu de l'étude menée en Sec. 4.4.3.e, il peut aussi sembler judicieux de modifier la taille des *time-slices*.

Observation 4.4.5 — Pour l'écoulement de canal turbulent, la deuxième itération corrige principalement la qualité de la solution pour l'écoulement de la couche interne, tandis qu'elle n'a que peu d'impact sur l'écoulement de couche externe. Aussi, le bénéfice apporté est bien inférieur à celui issu d'un choix optimal de la méthode d'interpolation ou de la taille des time-slices.

4.4.3.g Conclusions concernant l'analyse de convergence pour l'Écoulement de Canal Turbulent.

Une convergence de PARAREAL différente selon les zones d'écoulement considérées. Pour l'écoulement de canal turbulent, on observe que la convergence de PARAREAL vers une solution aux bonnes propriétés physiques est différente selon la zone de l'écoulement considérée, ce qui est à mettre en regard des différences connues sur la structure de la turbulence dans ces zones. Pour l'écoulement de couche interne (proche paroi), les propriétés physiques du solveur fin sont correctement retrouvées, tandis qu'on observe dans la couche externe une sur-estimation des profils de tension de Reynolds. Cette sur-estimation n'est par ailleurs pas corrigée par la deuxième itération de PARAREAL.

L'influence parasite du solveur grossier. De manière générale, on observe des difficultés de PARAREAL à correctement représenter l'écoulement dans la couche externe. Un examen des spectres de turbulence explique son origine par l'amplification d'un mode basse fréquence provoquée par le solveur grossier, que PARAREAL n'arrive pas à corriger avec un faible nombre d'itérations. Cependant, ce problème de précision de PARAREAL dans la couche externe peut être corrigé dans le cas présent en faisant varier l'ordre d'interpolation ou la taille des *time-slices*.

Méthode d'interpolation et taille des *time-slices*, les pilotes principaux de la précision. Alors que la deuxième itération de PARAREAL apporte un léger gain de précision pour la représentation de l'écoulement proche de la paroi, l'ordre d'interpolation et la taille des *time-slices* sont les paramètres influençant majoritairement la qualité de la solution.

Leur impact permet notamment de contrebalancer l'amplification des basses fréquences apportées par $\hat{\mathcal{G}}$, et induit une meilleure précision pour l'écoulement proche de la paroi. Pour la méthode d'interpolation, on observe cet effet bénéfique lorsque celle-ci apporte un taux mesuré de diffusion numérique, de par l'utilisation d'un ordre d'interpolation peu élevé. Cependant, la réduction d'ordre peut aussi entraîner une plus grande amplification des basses fréquences apportées par le changement de grille, et dégrader ainsi la solution de PARAREAL.

Par ailleurs, l'augmentation des *time-slices* permet dans une certaine mesure d'améliorer la qualité globale de la solution, quand le problème le permet. Cet intérêt semble limité par l'existence d'une valeur maximale pour la taille des *time-slices*, au-delà de laquelle la solution de PARAREAL se dégrade partiellement.

4.5 Synthèse

Dans ce chapitre, nous avons étudié une stratégie de parallélisation espace temps basée sur PARA-REAL avec grossissement spatial pour la simulation numérique directe d'écoulement turbulents avec le solveur HYBRID.

En Sec. 4.1, une analyse de performance a été menée, afin d'étudier les apports d'une telle stratégie dans un cadre de calcul massivement parallèle. Une étude basée sur des tests de scalabilité du solveur HYBRID a mis en évidence que cette stratégie de parallélisation espace-temps pouvait prolonger la scalabilité forte du solveur, à la condition qu'un faible nombre d'itérations de PARAREAL soit considéré. Puis, un modèle de communication a été développé pour évaluer les volumes de données transférées entre processus avec cette stratégie, en ayant pour perspective l'utilisation de ce modèle pour mesurer les coûts énergétiques de communication induits.

Dans un deuxième temps, on s'est focalisé sur la précision avec laquelle PARAREAL avec grossissement spatial peut simuler un écoulement turbulent pour deux problèmes canoniques : la décroissance d'une turbulence homogène isotrope, et l'écoulement de canal turbulent. Une introduction rapide à la physique générale de la turbulence a tout d'abord été donnée en Sec. 4.2. Pour les deux cas d'étude, traités respectivement en Sec. 4.3 et Sec. 4.4, on s'est attaché à décrire la physique particulière associée au cas test, les grandeurs principales d'intérêt, et la mise en place de PARAREAL avec grossissement spatial.

Pour la décroissance de turbulence homogène isotrope, on a montré que l'utilisation de méthode d'interpolation d'ordre suffisamment élevé ainsi qu'une taille assez grande des sous-domaines temporels permettait à PARAREAL d'obtenir une représentation relativement précise de la turbulence sur l'ensemble de ses échelles. L'étude a aussi mis en évidence un faible impact de l'augmentation du nombre de Reynolds de turbulence sur la précision de PARAREAL.

Pour l'écoulement de canal turbulent, PARAREAL a montré une bonne capacité à représenter correctement l'écoulement au voisinage des parois, mais présente aussi des difficultés pour prédire correctement les propriétés de l'écoulement dans la couche externe. Il a été observé que cette difficulté résultait d'une sur-estimation de l'énergie cinétique turbulente du solveur grossier dans cette zone de l'écoulement. Enfin, on a observé que l'ordre d'interpolation et la taille des sous-domaines pouvaient être utilisés pour minimiser l'erreur de PARAREAL, notamment dans la zone de difficulté dans la couche externe. A cet effet, un travail de recherche supplémentaire quant à l'origine de ce problème du solveur grossier s'avère nécessaire. Enfin, au vu des résultats, nous avons postulé l'existence d'une taille de time-slice optimale permettant de minimiser l'erreur de PARAREAL.
CHAPITRE 5

Conclusions

Table des matières

5.1	Réalisations logicielles	141
5.2	Synthèse	142
5.3	Perspectives	143

Ce chapitre a pour but de résumer les conclusions majeures apportées par ce travail de thèse, et les perspectives qui en découlent.

En Sec. 5.1, on présente tout d'abord les contributions n'ayant pas été mentionnées jusqu'à présent, mais qui ont pris une part importante à l'étude présentée dans ce manuscrit. Une synthèse des résultats généraux est ensuite proposée en Sec. 5.2, et ceux-ci sont mis en perspective en Sec. 5.3.

5.1 Réalisations logicielles

Bibliothèque de calcul pour l'étude des méthodes d'intégration temporelle. Utilisée en particulier pour obtenir les résultats du Chap. 3, une bibliothèque de calcul dénommée CASPER (*Combination of Algorithms for Sequential and Parallel solution of PDEs - Rapid prototyping*) a été développée dans le but d'étudier l'application de méthodes d'intégrations temporelles alternatives ou nouvelles (SDC, intégration exponentielle par méthode de Krylov, PARAREAL...) à différents types de problèmes classiques (équation d'advection-diffusion, de Burgers, système de Lorentz ...). Elle compte au moment de la rédaction du manuscrit environ 13000 lignes de code PYTHON, et son implémentation modulaire lui donne le potentiel d'être facilement extensible, à la fois vis-à-vis des problèmes pouvant être traités, et pour l'implémentation d'autres méthodes d'intégration temporelle. Il est envisagé pour la suite de continuer le développement de cette bibliothèque, afin de faciliter sa mise à disposition pour la communauté scientifique¹ et de l'étendre à d'autres types de problèmes ou méthodes numériques.

Bibliothèque d'interfaçage d'HYBRID pour l'application de PARAREAL. Comme mentionné dans le Chap. 4, l'algorithme PARAREAL n'a pas été implanté au sein du solveur HYBRID, mais plutôt au sein d'une bibliothèque permettant de manipuler les entrées et sorties du solveur CFD, ainsi que ses paramètres de simulation. Cette bibliothèque, dénommée PYHYB, comprend au moment de la rédaction plus de 15000 lignes de code PYTHON, au sein desquelles sont implémentés les opérateurs de transfert (interpolation, restriction) ainsi que les différents outils de post-traitement utilisés pour l'étude. En particulier, la bibliothèque est suffisamment modulaire pour que son cœur (interface d'HYBRID) puisse être facilement modifié, afin d'envisager l'application de PARAREAL (avec ou sans grossissement spatial) à d'autres solveurs CFD.

^{1.} Le code source est actuellement disponible sous le lien https://gitlab.com/tlunet/casper.

5.2 Synthèse

Identification des solutions actuelles de parallélisation en temps pour des solveurs CFD explicites. La première partie de ce travail de thèse a été consacrée à la sélection d'une première solution, pour appliquer une stratégie de parallélisation espace-temps, à la simulation d'écoulements turbulents tri-dimensionnels avec un solveur CFD explicite. Bien que de nombreuses solutions soient disponibles, et parfois facilement applicables dans le cadre d'une intégration temporelle implicite, on a remarqué que peu d'algorithmes actuels peuvent être facilement utilisés dans un cadre d'intégration temporelle explicite. Parmi l'ensemble des solutions recensées, l'algorithme PARAREAL avec grossissement spatial est actuellement la solution la plus envisageable pour les applications considérées.

Compréhension plus approfondie du comportement de PARAREAL pour les problèmes advectifs. Un premier objectif de la thèse a été d'étudier l'utilisation de méthodes d'intégration explicites au sein de PARAREAL pour des problèmes advectifs simples, ceci afin de caractériser les difficultés de convergence de l'algorithme, déjà connues dans la littérature. Cette analyse a aussi permis d'étudier l'influence des schémas de discrétisation spatiale, ainsi que celle du changement de grille induit par la stratégie de grossissement spatial. Pour ce dernier point, une modélisation a été spécialement développée (*cf.* Prop. 3.2.6).

Plusieurs points importants ont été mis en évidence :

- une influence majeure de la méthode d'intégration temporelle utilisée pour le solveur grossier. En particulier, une méthode d'intégration judicieusement choisie permet d'obtenir une convergence effective de PARAREAL, et ce même pour des problèmes à caractère hyperbolique (*cf.* Sec. 3.3.1).
- des instabilités linéaires faibles apportées par l'utilisation de discrétisations spatiales centrées pour le solveur grossier, qui s'amplifient avec la diminution du nombre *CFL* (*cf.* Sec. 3.3.2).
- une amplification caractéristique des composantes basse fréquence induite par la stratégie de grossissement spatial, dépendant fortement de la solution initiale considérée (*cf.* Sec. 3.3.3).

Enfin, deux solutions curatives principales pour contrer les instabilités linéaires de PARAREAL ont été mises en évidence : l'augmentation de l'ordre d'interpolation, et de la taille des sous-domaines temporels.

Validation de la stratégie de grossissement spatial au sein de PARAREAL. Une étape importante de ce travail a été d'évaluer la construction du solveur grossier pour PARAREAL. L'utilisation du grossissement spatial s'avère être une solution particulièrement intéressante, au vu des points suivants :

- une convergence effective de PARAREAL avec grossissement spatial pour des problèmes monodimensionnels fortement advectifs, éventuellement non-linéaires (*cf.* Sec. 3.4),
- un ratio de coût entre solveur fin et grossier permettant un gain (parfois important) de l'efficacité parallèle de la parallélisation spatio-temporelle, comparée à la parallélisation spatiale seule (*cf.* Sec. 4.1.1.b et Sec. 4.1.1.d).
- une représentation assez précise des grandes échelles des écoulements turbulents (*cf.* Sec. 4.3 et Sec. 4.4).

De manière générale, nous avons mis en évidence la nécessité d'utiliser des schémas d'interpolation d'ordre élevé, afin de minimiser l'erreur induite par le changement de grille, par rapport à l'erreur due à l'intégration sur la grille grossière. Enfin, il a été observé qu'un opérateur de restriction simple (injection) était suffisant pour ce type de stratégie.

Un apport prometteur de la parallélisation espace temps dans un cadre de calcul massivement parallèle. Une étude de performance de PARAREAL avec grossissement spatial appliqué à HYBRID pour différents problèmes nous a permis d'évaluer le gain que peut apporter ce type de stratégie, vis-à-vis d'une parallélisation spatiale seule. En particulier, un modèle de communication a été développé pour estimer le volume de données transférées globalement par les différentes stratégies de parallélisation espace temps. Les deux points suivants ont été mis en évidence :

- une extension de l'efficacité parallèle du solveur avec l'utilisation d'un grand nombre de processeurs, pour lequel la parallélisation spatiale seule perdrait totalement son efficacité.
- une réduction effective du volume de communication engendré globalement par la parallélisation sur le système de calcul.

Aussi, il a été souligné que ces deux gains étaient conditionnés par l'utilisation d'un faible nombre d'itérations de PARAREAL (autour de une ou deux).

Faisabilité de la parallélisation en temps pour la simulation d'écoulements turbulents. Le point d'orgue de la thèse a été d'obtenir un premier résultat de la convergence d'un algorithme de parallélisation en temps pour deux problèmes canoniques d'écoulements turbulents. Les deux cas test choisis (décroissance d'une Turbulence Homogène Isotrope & Écoulement de Canal Turbulent) ont été choisis dans le but de représenter des phénomènes caractéristiques de la turbulence.

L'étude a mis en évidence les points suivants, concernant PARAREAL avec grossissement spatial :

- une bonne capacité à représenter l'ensemble des échelles de la turbulence, et ce malgré l'utilisation d'un solveur grossier ne pouvant pas correctement représenter la totalité du spectre de la turbulence.
- une faible dépendance vis-à-vis de la gamme d'échelles considérées, qui laisse envisager la parallélisation spatio-temporelle pour des cas à des nombres de Reynolds plus élevés et des problèmes de plus grande taille.
- une intégration temporelle stable pour des écoulements à caractère fortement advectif, avec pourtant l'utilisation de méthodes de discrétisations spatiales non-dissipatives.
- une bonne capacité à représenter la turbulence proche paroi, et ce malgré une perte de précision du solveur grossier dans ces zones d'écoulement.

En particulier, il a été observé que la solution de PARAREAL avec grossissement spatial obtenue après une seule itération sur un nombre faible de *time-slices*, permettait d'obtenir une bonne représentation des propriétés physiques des écoulements turbulents considérés, sous condition d'un choix judicieux de paramètres (interpolation et taille de sous-domaine).

5.3 Perspectives

Quelques pistes d'extensions pour l'étude effectuée. Afin de délimiter le contour de l'étude, certains choix de paramètres ont été effectués au préalable. Il pourrait donc être intéressant d'étendre cette étude à des configurations plus diverses, comme par exemple étudier l'influence du nombre de *time-slices* sur la qualité de la solution obtenue avec PARAREAL.

Aussi, pour la résolution des écoulements turbulents, nous avons considéré des méthodes numériques identiques pour les solveurs fin et grossier. Au vu des défauts présentés par le solveur grossier dans certains cas, on peut envisager l'utilisation de méthodes correctrices pour influer sur la convergence de PARAREAL. Pour exemple, on peut explorer l'utilisation d'une modélisation différente (*e.g.* modèle LES pour corriger les défauts du solveur grossier observés pour l'écoulement de canal turbulent) ou bien des schémas numériques différents (*e.g.* ordre de discrétisation plus faible pour réduire le coût du solveur grossier). Enfin, l'interpolation d'ordre élevé présentait durant cette étude le double avantage de la simplicité d'implémentation et du faible coût en raison du contexte de simulation (maillages cartésiens structurés). Il est donc nécessaire d'envisager des opérateurs de changement de grille permettant d'effectuer des interpolations d'ordre élevé pour d'autres types de maillages (non cartésiens, non structurés, ...). Dans cette perspective, l'utilisation de discrétisations spatiales de type "différences spectrales" [Wang, 2007], qui fournit par construction plusieurs représentations à des ordres différents et sur des maillages non-structurés, est une alternative intéressante.

Parallélisation temporelle mono-itération pour la simulation des écoulements turbulents : entre rêve, nécessité, et réalité ... Afin de maximiser les performances de la parallélisation spatio-temporelle, l'utilisation d'un faible nombre d'itérations semble indispensable. De plus, nous avons aussi observé dans de nombreuses situations qu'une seule itération de PARAREAL permettait d'obtenir une représentation correcte des propriétés de la turbulence. En revanche, la deuxième itération n'améliorait pas significativement la qualité de la solution. On peut donc espérer l'émergence de méthodes d'intégration temporelle parallèle, à itération unique et précision raisonnable. Cet objectif n'est encore que partiellement atteint avec PARAREAL.

C'est pourquoi plusieurs pistes d'amélioration des solutions existantes doivent aussi être explorées. En particulier, pour PARAREAL, l'utilisation du grossissement spatial est très intéressante pour obtenir une solution initiale locale à chaque sous-domaine temporel. Cependant elle représente aussi un défaut de l'algorithme lorsqu'elle est incluse dans le processus de correction itératif. La simplicité du formalisme du *multiple-shooting* est néanmoins un grand avantage pour l'application à des solveurs CFD déjà existants. Il faut donc imaginer des variations de l'algorithme de *multiple-shooting*, distinctes de PARAREAL, et qui pourraient rendre les étapes de corrections plus efficaces.

Développement de moyens plus adaptés pour caractériser l'erreur de parallélisation tem-

porelle. Un point évident qui ressort de l'étude est l'apparition inéluctable d'une nouvelle erreur d'intégration, plus importante que celle considérée habituellement pour les discrétisations temporelles classiques. Dans le cas présent, l'aspect chaotique de la turbulence joue un rôle prédominant sur l'évaluation de cette erreur, et une comparaison point à point vis-à-vis d'un champ de référence apparaît non seulement insuffisante, mais aussi trop exigeante. C'est pourquoi il semble nécessaire de développer des critères d'erreur portant sur une évaluation des propriétés statistiques des solutions étudiées (*e.g.* énergie cinétique turbulente ou dissipation d'un écoulement turbulent), que l'on pourrait regrouper en critères de "qualité physique". En particulier, l'algorithme PARAREAL apporte régulièrement au cours des calculs une information liée à l'évaluation de deux solutions finales (fine et grossière), en partant d'une même solution initiale. Cette information, peu exploitée en pratique jusqu'à présent, serait intéressante pour l'élaboration de critères d'erreur. À terme, ce genre de critères pourrait être utilisé pour :

- obtenir des critères d'arrêt sur le nombre d'itérations de l'algorithme,
- juger le choix des paramètres utilisés pour l'intégration temporelle parallèle (*e.g.* type d'interpolation, taille de *time-slice*),
- optimiser ces derniers grâce à une expérience acquise sur des problèmes similaires,
- améliorer les réglages de l'algorithme en cours de calcul.

ANNEXE A

Intégration temporelle par Spectral Deferred Correction

Table des matières

A.1	Descri	ption détaillée
A.2	Appli	cations à des problèmes advectifs avec intégration temporelle explicite147
	A.2.1	Cadre de l'étude
	A.2.2	Instabilités des versions explicites des SDC
	A.2.3	Sweep SDC d'ordre élevé 149
	A.2.4	Stabilité et performance des SDC avec <i>sweep</i> d'ordre élevé
	A.2.5	Synthèse

L'objectif de cette annexe est de fournir une description détaillée des méthodes de type Spectral Deferred Correction (SDC), afin de présenter ensuite son étude pour l'application à des problèmes advectifs, en se restreignant à des formes explicites des SDC.

À cet effet, les mécanismes derrière l'intégration par SDC sont tout d'abord présentés en Sec. A.1. Puis, une rapide étude est menée en Sec. A.2, dans le but de comparer l'efficacité des méthodes SDC avec une méthode d'intégration temporelle classique de type Runge-Kutta.

A.1 Description détaillée

Dans cette section, on prolonge la description des SDC donnée en Sec. 2.3.1 afin de rentrer plus en détail dans le fonctionnement de la méthode. À nouveau, on considère un système d'EDO sous sa forme de Picard :

$$U(t) = U(0) + \int_0^t f(U(\tau))d\tau.$$
 (A.1)

On rappelle la définition de l'erreur e et du résidu r :

 $e(t) = U(t) - \tilde{U}(t), \tag{A.2}$

$$r(t) = U(0) + \int_0^t f(\tilde{U}(\tau))d\tau - \tilde{U}(t),$$
(A.3)

avec $\hat{U}(t)$ l'approximation de la solution exacte U(t) dans le domaine temporel, ainsi que l'équation différentielle sur l'erreur :

$$e(t)' = r(t)' + f(\tilde{U}(t) + e(t)) - f(\tilde{U}(t)).$$
 (A.4)

Description d'un sweep SDC. On considère l'application des SDC entre deux temps discrets t_n et t_{n+1} , à partir d'une solution initiale U_n . On représente la division de $[t_{n-1}, t_n]$ en sous-intervalles de la forme $[t_{n-1} = \tau_1^n, ..., \tau_m^n, ..., \tau_M^n = t_n]$ en Fig. A.1. On note \tilde{U}_m^k la solution obtenue au temps t_m

FIGURE A.1 – Intervalles de discrétisation pour les SDC

après k sweep, avec $k \in [[0, K]]$. Une étape d'initialisation permet d'obtenir chaque $U_m^{k=0}$ en utilisant un modèle approximatif peu coûteux (*Forward Euler* (FE), *Backward Euler* (BE), ou tout simplement une recopie de la solution initiale).

Puis, le *sweep* peut être effectué; (A.2) est discrétisée pour obtenir l'erreur au temps $\tau_{m,n}$ après le $k^{\text{ème}}$ sweep, soit :

$$\forall m \in [\![1, M]\!], \ \forall k \in [\![0, K-1]\!], \ e_m^k = \tilde{U}_m^{k+1} - \tilde{U}_m^k.$$
 (A.5)

Il en va de même pour (A.3), ce qui permet d'écrire le résidu discret sous la forme :

$$\forall m \in [\![1, M]\!], \ \forall k \in [\![0, K]\!], \ r_m^k = U_n + \int_{t_n}^{\tau_m^n} f\left(\tilde{U}^k(\tau)\right) d\tau - \tilde{U}_m^k.$$
 (A.6)

Puis une discrétisation temporelle est appliquée à (A.4) pour chaque sous-intervalle de temps $\Delta \tau_m^n = \tau_{m+1}^n - \tau_m^n$. Pour exemple, l'utilisation de la méthode BE permet d'obtenir la relation suivante :

$$e_{m+1}^{k} = e_{m}^{k} + r_{m+1}^{k} - r_{m}^{k} + \Delta \tau_{m}^{n} \left[f\left(\tilde{U}_{m+1}^{k} + e_{m+1}^{k} \right) - f\left(\tilde{U}_{m+1}^{k} \right) \right].$$
(A.7)

Enfin, en utilisant (A.6) et (A.5), puis en réarrangeant les termes de (A.7), on obtient la formule de mise à jour pour \tilde{U}_m^k (ou *sweep*) :

$$\tilde{U}_{m+1}^{k+1} = \tilde{U}_m^{k+1} + \Delta \tau_m^n \left[f\left(\tilde{U}_{m+1}^{k+1}\right) - f\left(\tilde{U}_{m+1}^k\right) \right] + \int_{\tau_m^n}^{\tau_{m+1}^n} f\left(\tilde{U}^k(\tau)\right) d\tau.$$
(A.8)

Si une discrétisation de type FE est utilisée, la formule devient alors :

$$\tilde{U}_{m+1}^{k+1} = \tilde{U}_m^{k+1} + \Delta \tau_m^n \left[f\left(\tilde{U}_m^{k+1}\right) - f\left(\tilde{U}_m^k\right) \right] + \int_{\tau_m^n}^{\tau_m^n+1} f\left(\tilde{U}^k(\tau)\right) d\tau.$$
(A.9)

Une représentation graphique des deux formules (A.8) et (A.9) est donnée en Fig. A.2. Les deux



FIGURE A.2 – Représentation graphique d'un sweep SDC avec une discrétisation de (A.4), en utilisant (a) : Forward Euler, (b) : Backward Euler. M = 6.

formes sont appelées respectivement par la suite SDC-BE et SDC-FE.

Quadrature numérique. Les formules présentées dans le paragraphe précédent pointent la nécessité de calculer une intégrale de $f(\tilde{U}^k(\tau))$ au cours de chaque *sweep*. Celle-ci est calculée sur l'intervalle $[\tau_m^n, \tau_{m+1}^n]$ par quadrature numérique. À cet effet, on utilise une représentation polynômiale de la solution temporelle sur l'ensemble des points τ_m^n , ce qui permet d'écrire :

$$\int_{\tau_m^n}^{\tau_{m+1}^n} f\left(\tilde{U}^k(\tau)\right) d\tau \simeq \sum_{l=1}^M w_l^m f\left(U_l^k\right).$$
(A.10)

Les coefficients de quadrature w_l^m sont déterminés grâce à des méthodes classiques utilisant les polynômes de Lagrange (voir [Gil *et al.*, 2007]). La répartition des points τ_m^n , appelés aussi points de quadrature, est donc un paramètre important pouvant jouer sur la précision et la stabilité de la méthode. Une étude plus détaillée de ce paramètre peut être trouvée dans [Layton et Minion, 2005].

Discrétisation temporelle d'un sweep. Comme observé précédemment, un autre paramètre important des SDC correspond à la discrétisation de la dérivée temporelle de l'erreur utilisée dans (A.4). Il a été montré par [Dutt *et al.*, 2000] que chaque *sweep* ajoute à l'ordre global de la solution celui de la méthode choisie. Ainsi, lorsqu'une discrétisation d'ordre 1 est utilisée, *k sweep* permettent (sous certaines conditions) d'obtenir une précision d'ordre *k*. D'autres types de discrétisation que BE ou FE peuvent être utilisés. [Minion, 2003] propose notamment une forme semi-implicite à partir d'une décomposition de *f* en deux termes, l'un traité implicitement avec BE, l'autre explicitement avec FE. [Christlieb *et al.*, 2009] proposent même d'utiliser des méthodes de type Runge-Kutta d'ordres élevés, ce qui permet d'augmenter la stabilité et la précision de la méthode, au prix d'évaluations supplémentaires de *f*. Comme on le verra par la suite, cet aspect a une grande importance, tant sur le coût de la méthode, que sa précision et stabilité.

A.2 Applications à des problèmes advectifs avec intégration temporelle explicite

Dans cette section, on effectue une rapide étude de l'intégration temporelle avec les SDC du problème suivant :

$$\frac{\partial u}{\partial t} = -\frac{\partial u}{\partial x}, \quad u(x,0) = u_0(x), \quad x \in [0,1], \quad t \in [0,T], \tag{A.11}$$

résolu sur un maillage spatial périodique uniforme. Ce problème simplifié nous permet d'effectuer une rapide étude d'applicabilité des SDC, afin de comparer ses performances avec les méthodes classiques déjà utilisées pour ce genre de problème. A cette effet, on choisi la méthode de Runge-Kutta classique d'ordre 4 (RKC4, cf. B.1) comme référence. De par cette comparaison n'ayant pas à notre connaissance été réalisée, l'étude fait donc partie des contributions de cette thèse¹.

Enfin, une nouvelle forme de sweep SDC, dite "d'ordre élevé", est introduite en Sec. A.2.3, et constitue aussi une des contributions novatrices de cette étude.

A.2.1 Cadre de l'étude

Intégration temporelle explicite avec discrétisation spatiale centrée. Bien que de nombreux travaux aient été effectués sur les méthodes SDC dans leurs formes implicites ou semi-implicites, très peu se sont focalisés sur l'utilisation de formes explicites des SDC. On peut mentionner cependant le travail de [Christlieb *et al.*, 2010b], qui s'intéresse plus particulièrement à la méthode

^{1.} Il est important de mentionner que les méthodes SDC ont déjà été comparées avec des méthodes Runge-Kutta explicite par [Ketcheson et bin Waheed, 2014]. Cependant, ces derniers ont comparé la stabilité des méthodes en prenant comme critère la portion d'axe réel inclus dans le contour de stabilité [Ketcheson et bin Waheed, 2014, Sec. 4.1], ce qui permet de discriminer les méthodes pour des problèmes purement diffusifs. Dans le cadre de cette étude cependant, ce sont des problèmes à nature advective qui nous intéressent.

SDC-FE, en considérant une répartition uniforme de points de quadrature. Ces derniers mettent aussi en évidence une perte de l'inclusion de l'axe imaginaire dans le contour de stabilité de SDC-FE, sous certaines conditions.

Or, lorsque des schémas de discrétisation spatiale de type centré sont requis (*e.g.* pour minimiser l'erreur de diffusion sur l'ensemble des échelles du problème), cette inclusion de l'axe imaginaire dans le contour de stabilité est primordiale. En effet, lorsque cette condition n'est pas remplie, l'intégration temporelle est instable avec toute discrétisation centrée utilisée pour résoudre le problème (A.11). C'est pourquoi on se focalise tout particulièrement sur cette condition pour la suite de l'étude.

La stabilité vue comme pilote principale de l'efficacité. Si la condition d'inclusion décrite précédemment est remplie, la longeur de la partie de l'axe imaginaire inclue dans le contour de stabilité détermine notamment le *CFL* maximum pour lequel l'intégration temporelle est stable, avec une discrétisation spatiale centrée donnée. Or, le coût d'intégration pour un pas de temps reste généralement constant au cours d'une intégration temporelle explicite. En particulier, il est à peu de choses près proportionnel au nombre d'évaluations requise pour l'opérateur f (dérivée spatiale dans notre cas).

Ainsi, maximiser le pas de temps utilisé pour l'intégration temporelle permet donc de minimiser le coût du calcul, jusqu'à un pas de temps maximal déterminé par la condition de stabilité *CFL* maximale. En particulier, cette dernière est dans notre cas directement liée à la portion de l'axe imaginaire inclue dans le contour de stabilité. C'est pourquoi on se restreint ici à l'étude du contour de stabilité, afin d'en déduire une première estimation de l'efficacité des formes explicites des SDC pour le problème étudié.

A.2.2 Instabilités des versions explicites des SDC.

On se focalise ici sur l'utilisation de SDC-FE, en utilisant une répartition des points de quadrature de type "Legendre", classiquement utilisée dans la littérature (voir [Minion, 2003] pour plus de détails). L'initialisation SDC, quant à elle, est effectuée par simple recopie de la solution initiale pour l'ensemble des points de quadrature. En Fig. A.3, on représente les contours de stabilité de SDC-FE, tout en faisant varier le nombre de sweep K et le nombre de points de quadrature M.



FIGURE A.3 – Contours de stabilité de SDC-FE, pour différentes valeurs de K et M.

On peut tout d'abord y observer que tout premier *sweep* SDC-FE est instable avec des discrétisations centrées, de par la non-inclusion de l'axe imaginaire dans le contour de stabilité, et ce même en changeant le nombre de points de quadrature. L'utilisation de *sweeps* supplémentaires permet éventuellement d'inclure une partie de l'axe imaginaire, mais ce uniquement après plusieurs *sweeps*, ce qui augmente de ce fait le coût d'intégration.

Cependant, cette instabilité de SDC-FE avec les discrétisations centrées est particulièrement problématique. En effet, l'une des raisons d'être des SDC, en particulier dans le cadre de la parallélisation temporelle, est de rendre le coût d'un *sweep* le plus faible possible, comparé à un solveur fin standard, afin de remplacer ce dernier, et de réduire ainsi le coût d'une itération de l'algorithme de parallélisation temporelle (*cf.* les motivations à l'origine PFASST en Sec. 2.4). De ce fait, le sweep SDC-FE unique est inadapté pour ce type de stratégie avec l'utilisation de schémas de discrétisation spatiale centrées. C'est pourquoi il est nécessaire de rechercher d'autres formes explicites SDC, permettant d'obtenir une stabilité avec des schémas centrées dès le premier *sweep*.

A.2.3 Sweep SDC d'ordre élevé

Solutions existantes et idée du *sweep* d'ordre élevé. Afin de résoudre ce problème mentionné précédemment, quelques travaux ont proposé de changer l'intégration temporelle pour la dérivée de l'erreur dans (A.4). En particulier, [Christlieb *et al.*, 2010a] proposent d'utiliser des méthodes de Runge-Kutta d'ordre élevé pour améliorer la stabilité. Cependant, ceci entraîne une augmentation du nombre d'évaluations de f, et donc du coût de calcul pour un *sweep*.

C'est pourquoi dans le cadre de la thèse, une nouvelle approche est proposée, permettant d'augmenter l'ordre de la discrétisation du *sweep* utilisé, en gardant le même nombre d'évaluations de l'opérateur f que pour SDC-FE. Si l'on revient à la Fig. A.2, on remarque qu'au cours d'un *sweep* pour le point de quadrature τ_m , seule l'information du champ \tilde{U}^{k+1} au point τ_{m-1} est utilisée, alors qu'elle est aussi disponible pour les points de quadrature précédents. Ceci nous motive donc pour présenter une nouvelle forme de *sweep*, dite d'ordre élevé, permettant de mettre cette "information" non utilisée à profit.

Sweep explicite d'ordre 2. On considère ici une différence finie d'ordre 2 pour discrétiser e'(t) et r'(t) dans (A.4). Pour $m \le 2$, l'approximation de e'(t) sur un maillage irrégulier s'écrit :

$$e'(\tau_m) \approx \alpha_- e_{m-1} + \alpha_0 e_m + \alpha_+ e_{m+1}, \tag{A.12}$$

et de manière similaire pour $r'(\tau_m)$. En posant $\Delta_- = \tau_m - \tau_{m-1}$ et $\Delta_+ = \tau_{m+1} - \tau_m$, les différents coefficients α s'expriment ainsi :

$$\alpha_{-} = \frac{-\Delta_{+}}{\Delta_{+}\Delta_{-} + \Delta_{-}^{2}}, \quad \alpha_{0} = \frac{\Delta_{+} - \Delta_{-}}{\Delta_{+}\Delta_{-}}, \quad \alpha_{+} = \frac{\Delta_{-}}{\Delta_{+}\Delta_{-} + \Delta_{+}^{2}}.$$
 (A.13)

Ceci permet d'écrire (A.4) sous la forme discrète suivante :

$$\alpha_{-}e_{m-1} + \alpha_{0}e_{m} + \alpha_{+}e_{m+1} = f(\tilde{u}_{m}^{k} + e_{m}) - f(\tilde{u}_{m}^{k}) + \alpha_{-}r_{m-1} + \alpha_{0}r_{m} + \alpha_{+}r_{m+1},$$
(A.14)

forme pouvant être réécrite en utilisant (A.6), et le fait que $\alpha_{-} + \alpha_{0} + \alpha_{+} = 0$, pour obtenir :

$$\tilde{u}_{m+1}^{k+1} = \frac{-\alpha_0 \tilde{u}_m^{k+1} - \alpha_- \tilde{u}_{m-1}^{k+1} + f(\tilde{u}_m^{k+1}) - f(\tilde{u}_m^k) + \alpha_+ \int_{t_m}^{t_{m+1}} f(\tilde{U}^k) - \alpha_- \int_{t_{m-1}}^{t_m} f(\tilde{U}^k)}{\alpha_+}.$$
 (A.15)

Les termes liés aux intégrales sont calculés par quadrature numérique, comme spécifié en (A.10). Cette formule de *sweep* de type *Second Order* (SO), ne peut donc être calculée qu'à partir du troisième point de quadrature (m = 3), et est par conséquent associée à la formule de SDC-FE pour m = 2 uniquement. Une représentation graphique du *sweep* est donnée en Fig. A.4, et on note par la suite cette forme SDC-SO.

Sweep d'ordre élevé. L'idée du dernier paragraphe peut très bien être généralisée à des ordres supérieurs à 2. En effet, supposons que le point de quadrature τ_m soit calculé, et que l'on cherche à estimer les dérivées $e'(\tau_m)$ et $u'^k(\tau_m)$. On considère ainsi l'approximation par différences finies de la dérivée utilisant (m + 1) valeurs temporelles d'une variable V donnée :

$$V'(\tau_m) \approx \sum_{i=1}^{m+1} \alpha_i V(\tau_i).$$
(A.16)



FIGURE A.4 – Représentation graphique d'un sweep SDC-SO, M = 6.

Ceci nous permet de réécrire (A.4) de la manière suivante :

- Evaluation de f(U)

$$\sum_{i=1}^{m+1} \alpha_i e(\tau_i) = f(\tilde{u}_m^{k+1}) - f(\tilde{u}_m^k) + \sum_{i=1}^{m+1} \alpha_i r(\tau_i).$$
(A.17)

La formule de dérivation (A.16) est exacte pour chaque polynôme de degré m passant par les (m+1) points $V(t_i)$, ceci pouvant être exprimé en utilisant les polynômes de Lagrange :

$$P_m(t) = \sum_{i=1}^{m+1} V(\tau_i) L_i(t),$$
(A.18)

Valeur de U

avec $L_i(t)$ le $i^{\text{ème}}$ polynôme de Lagrange sur $[\tau_1, ..., \tau_{m+1}]$, défini ainsi :

$$L_i(t) = \prod_{j=1, \ j \neq i}^{m+1} \frac{t - t_j}{t_i - t_j}.$$
(A.19)

Or, si (A.16) est exacte pour un polynôme de degré m, cela signifie qu'elle est exacte pour :

$$P'_{m}(t) = \sum_{i=1}^{m+1} V(\tau_{i}) L'_{i}(t),$$
(A.20)

et ce, pour n'importe quelle combinaison de $V(\tau_i)$. Ceci nous permet donc d'identifier les coefficients de dérivation dans (A.16) comme étant :

$$\alpha_i = L'_i(t_m) = \frac{\sum_{j=1, \ j \neq i}^{m+1} \prod_{k=1, \ k \neq \{i,j\}}^{m+1} (t_m - t_k)}{\prod_{j=1, \ j \neq i}^{m+1} (t_i - t_j)}.$$
(A.21)

Comme la dérivée discrète est nulle pour un polynôme constant, on a nécessairement la relation :

$$\sum_{i=1}^{m+1} \alpha_i = 0.$$
 (A.22)

De retour à (A.17), on peut désormais la développer en utilisant (A.6) :

$$\sum_{i=1}^{m+1} \alpha_i e(t_i) = f(\tilde{u}_m^{k+1}) - f(\tilde{u}_m^k) + \sum_{i=1}^{m+1} \alpha_i \tilde{u}_1^K + \sum_{i=1}^{m+1} \alpha_i \int_{\tau_1}^{\tau_i} f(\tilde{u}^k) - \sum_{i=1}^{m+1} \alpha_i \tilde{u}_i^k$$
(A.23)

Puis, en utilisant (A.22) pour supprimer la première somme du terme de droite, (A.5) pour faire apparaître \tilde{u}^{k+1} , et en enlevant le terme de la dernière somme contenant l'intégrale entre τ_1 et τ_1 , on peut écrire :

$$\sum_{i=1}^{m+1} \alpha_i \tilde{u}_i^{k+1} = f(\tilde{u}_m^{k+1}) - f(\tilde{u}_m^k) + \sum_{i=2}^{m+1} \alpha_i \int_{\tau_1}^{\tau_i} f(\tilde{u}^k).$$
(A.24)

En posant $I_i^j = \int_{\tau_i}^{\tau_j} f(\tilde{u}^k)$, on a :

$$\sum_{i=2}^{m+1} \alpha_i \int_{\tau_1}^{\tau_i} f(\tilde{u}^k) = \sum_{i=2}^{m+1} \alpha_i \sum_{j=1}^{i-1} I_j^{j+1}$$
$$= \sum_{i=2}^{m+1} \alpha_i I_1^2 + \sum_{i=3}^{m+1} \alpha_i I_2^3 + \dots + \sum_{i=m}^{m+1} \alpha_i I_{m-1}^m + \alpha_{m+1} I_m^{m+1}$$
(A.25)

$$= \sum_{i=1}^{i} I_i^{i+1} \sum_{j=i+1}^{i} \alpha_j.$$
 (A.26)

Puis, en injectant cette dernière relation dans (A.24), on obtient la formule de *sweep* d'ordre élevé pour le point de quadrature τ_{m+1} :

$$\tilde{u}_{m+1}^{k+1} = \frac{1}{\alpha_{m+1}} \left[-\sum_{i=1}^{m} \alpha_i \tilde{u}_i^{k+1} + f(\tilde{u}_m^{k+1}) - f(\tilde{u}_m^k) + \sum_{i=1}^{m} I_i^{i+1} \sum_{j=i+1}^{m+1} \alpha_j \right].$$
(A.27)

Bien entendu, cette dernière n'est applicable pour l'ordre m qu'à partir du point de quadrature τ_{m+1} , ce qui induit une augmentation de l'ordre progressive en fonction des points de quadrature.

A.2.4 Stabilité et performance des SDC avec sweep d'ordre élevé

Intérêt du sweep d'ordre élevé. On étudie ici l'intérêt des formes SDC explicite d'ordre élevé, et plus particulièrement la forme SDC-SO et SDC-TO (*Third Order*), sur les contours de stabilité des SDC. Comme précédemment, on utilise une répartition des points de quadrature de type Legendre. Les contours de stabilité pour le premier *sweep* sont représentés en Fig. A.5, en comparaison avec le contour de stabilité de la méthode Runge-Kutta classique d'ordre 4 (RKC4).



(a) Variation des formes SDC, pour M = 5 et K = 1. (b) Variation de M, pour SDC-SO et K = 1.

FIGURE A.5 – Apport des sweep d'ordre élevé sur le contour de stabilité des SDC.

On peut observer tout d'abord en Fig. A.5a que l'utilisation d'un *sweep* d'ordre 2 induit certe une rétraction du contour de stabilité selon l'axe réel, mais aussi permet une inclusion de l'axe imaginaire dès le premier *sweep*. En particulier, le contour de stabilité est similaire à celui de la méthode RKC4, ce qui induit un *CFL* maximum très proche pour les deux méthodes, pour une méthode de discrétisation spatiale donnée. Avec M = 5, le nombre d'évaluation de f requises par SDC-SO sur un pas de temps est égal à celui requis par RKC4, c'est pourquoi on peut affirmer en première estimation que les deux méthodes ont un coût similaire.

Comme on peut le voir en Fig. A.5a, l'augmentation de l'ordre du *sweep* ne permet pas d'accroîte l'inclusion de l'axe imaginaire dans le contour de stabilité : pour SDC-TO, ce dernier non seulement se rétracte d'autant plus, mais aussi perd l'inclusion une partie de l'axe imaginaire. Enfin, on peut

aussi envisager de faire varier le nombre de points de quadrature pour SDC-SO, comme représenté en Fig. A.5b. Bien que l'augmentation de M permette un acroissement du contour de stabilité de SDC-SO, et en particulier une inclusion plus importante de l'axe imaginaire dans son contour, elle induit par la même occasion une augmentation du nombre d'évaluation de f, donc du coût d'un *sweep*. De ce fait, le gain apporté par l'augmentation de la limite *CFL* est contre-balancé par cette augmentation du coût du *sweep*.

Comparaison avec une méthode Runge-Kutta classique. Comme observé précédemment, l'utilisation des SDC dans leur forme explicite optimisée (SO) peut au mieux avoir un coût équivalent à celui de méthodes classiques comme RKC4, et ce pour un seul *sweep*. De ce fait, les SDC perdent leur intérêt dans un contexte de parallélisation en temps : alors que l'utilisation d'un *sweep* SDC-BE peut être rendu extrêmement peu couteûse en prenant un pas de temps suffisamment grand, non restreint à une condition *CFL*, cela n'est pas le cas pour un *sweep* SDC explicite.

De manière générale, le coût (en terme de nombre d'évaluation de f) des méthodes Runge-Kutta est globalement équivalent à celui de la forme SDC étudiée. Enfin, si l'on prend en compte le nombre de champs devant être stockés en mémoire pour les SDC (proportionnel à MK) bien plus important que celui requis par des méthodes Runge-Kutta explicite classique, les SDC n'apparaissent donc pas comme une alternative intéressante pour l'intégration temporelle de problèmes dominés par l'advection.

A.2.5 Synthèse

Cette courte étude nous a donc permis d'étudier l'intérêt d'utiliser une forme explicite des SDC, dans un cadre particulier (problème fortement advectif, utilisation de discrétisations spatiales centrées), ce qui n'a été que très peu exploré à ce jour dans la littérature existante.

Une instabilité des formes classiques SDC-FE avec des schémas de discrétisation centrée a été mise en évidence, puis une nouvelle forme de SDC avec *sweep* d'ordre élevé a été présentée, dans le but de résoudre ces problèmes d'instabilité sans augmenter substantiellement les coûts de calcul. En particulier, la forme SDC-SO présente d'intéressantes propriétés de stabilité avec des schémas centrés. Cependant, la comparaison d'un *sweep* SDC-SO avec la méthode classique RKC4 a montré que les deux méthodes étaient globalement équivalentes. Ceci empêche donc les SDC d'être utilisées avec succès dans une stratégie cherchant à réduire de manière importante le coût d'une itération d'un algorithme parallèle en temps (*e.g.* PFASST) dans un contexte exclusivement explicite.

ANNEXE **B**

Compléments pour l'analyse de Von Neumann

Table des matières

B.1	Méthodes de Runge-Kutta explicites	153
B.2	Symboles de Fourier des discrétisations spatiales	153

B.1 Méthodes de Runge-Kutta explicites

Les tables de Butcher des méthodes ERK utilisées dans le manuscrit sont les suivantes :

Forward Euler (FE)	RI	K ord	re 2 (ŀ	HEUN	2)	RK or	ordre 1 (RK21)								
0		0				0									
1		1	1			3/4	3/4								
, , , , , , , , , , , , , , , , , , ,			1/2	1/2			-1/3	4/3							
	F	RK ord	dre 4 (RKC4)										
	0														
	1/2	1/2													
	1/2	0	1/2												
	1	0	0	1											
		1/6	1/3	1/3	1/6										

Les contours de stabilité pour chaque méthode sont représentés en Fig. B.1.

B.2 Symboles de Fourier des discrétisations spatiales

Les symboles de Fourier des discrétisations spatiales différences finies classiques sont donnés en Tab. B.1. La représentation des symboles de Fourier dans le plan complexe est donnée en Fig. B.1 pour quelques exemples de schémas.

Dénomination	Expression $rac{\partial u}{\partial x}\simeq$	Symbole de Fourier
Upwind ordre 1	$\frac{u_i - u_{i-1}}{\Delta_x}$	$1 - e^{-\iota\theta_{\kappa}}$
$\operatorname{Centr\acute{e}} \operatorname{ordre} 2$	$\frac{u_{i+1}-u_{i-1}}{2\Delta_x}$	$\frac{e^{\iota\theta_{\kappa}} - e^{-\iota\theta_{\kappa}}}{2} = \iota.sin(\theta_{\kappa})$
Upwind ordre 3	$\frac{2u_{i+1}+3u_i-6u_{i-1}+u_{i-2}}{6\Delta_x}$	$\frac{2e^{\iota\theta_{\kappa}}+3-6e^{-\iota\theta_{\kappa}}+e^{-2\iota\theta_{\kappa}}}{6}$
${\rm Centr\acute{e}} \ {\rm ordre} \ 4$	$\frac{-u_{i+2}+8u_{i+1}-8u_{i-1}+u_{i-2}}{12\Delta_x}$	$\frac{-e^{2\iota\theta\kappa}+8e^{\iota\theta\kappa}-8e^{-\iota\theta\kappa}+e^{-2\iota\theta\kappa}}{12}$
Upwind ordre 5	$\frac{-3u_{i+2}+30u_{i+1}+20u_i-60u_{i-1}+15u_{i-2}-2u_{i-3}}{60\Delta_x}$	$\frac{-3e^{2\iota\theta_{\kappa}}+30e^{\iota\theta_{\kappa}}+20-60e^{-\iota\theta_{\kappa}}+15e^{-2\iota\theta_{\kappa}}-2e^{-3\iota\theta_{\kappa}}}{60}$
$Centr\acute{\mathrm{e}} \ ordre \ 6$	$\frac{u_{i+3}-9u_{i+2}+45u_{i+1}-45u_{i-1}+9u_{i-2}-u_{i-3}}{60\Delta_x}$	$\frac{e^{3\iota\theta_{\kappa}}-9e^{2\iota\theta_{\kappa}}+45e^{\iota\theta_{\kappa}}-45e^{-\iota\theta_{\kappa}}+9e^{-2\iota\theta_{\kappa}}-e^{-3\iota\theta_{\kappa}}}{60\Delta_{x}}$

TABLE B.1 – Expressions et symboles de Fourier des schémas différences finies upwind et centrés d'ordre 1 à 6.



FIGURE B.1 – Spectres de stabilité des méthodes ERK (gauche) et symboles de Fourier de quelques discrétisations spatiales (U1 : upwind ordre 1, C2 : centré ordre 2, U5 : upwind ordre 5, C6 : centré ordre 6).

ANNEXE C

Interpolation d'ordre variable pour maillages cartésiens tridimensionnels structurés

Table des matières

C.1	Mailla	ges périodiques uniformes 155				
	C.1.1	Interpolation point milieu monodimensionnelle				
	C.1.2	Extension aux maillages tridimensionnels				
C.2	Exten	sion aux maillages non-uniformes				
C.3	Prise e	en compte de conditions limites non-périodiques				
C.4	Application pour des champs solutions obtenus avec le solveur Hybrid 159					

La méthode que l'on décrit ici se base sur des processus d'interpolation mono-dimensionnels qui sont étendus à un espace tridimensionnel. On peut apparenter ce processus à une succession de produits tensoriels utilisant des formules d'interpolation 1D. Pour simplifier la compréhension de la méthode, on décrira tout d'abord son fonctionnement pour des maillages uniformes périodiques en Sec. C.1, puis son extension à des maillages non uniformes en Sec. C.2, et enfin la prise en compte de conditions limites de type Dirichlet ou Neumann en Sec. C.3.

C.1 Maillages périodiques uniformes

C.1.1 Interpolation point milieu monodimensionnelle

Soit un maillage grossier $\Omega_{\mathcal{G}} = (x_i)_{0 \le i < N_L/2}$ construit en ne gardant qu'un point sur deux d'un maillage uniforme fin $\Omega_{\mathcal{F}}$ de taille N_L , avec N_L pair. L'ensemble des points de $\Omega_{\mathcal{F}}$ non communs avec $\Omega_{\mathcal{G}}$ est noté $\Omega_{\mathcal{F} \setminus \mathcal{G}} = (x_{i+1/2})_{0 \le i < N_L/2}$. On note $u_i = u(x_i)$ la valeur d'un champ pour un point de maillage donné. Dès lors, l'interpolation d'ordre o_I des valeurs de u sur $\Omega_{\mathcal{F} \setminus \mathcal{G}}$ à partir des valeurs de u sur $\Omega_{\mathcal{G}}$ s'écrit :

$$\forall i \in [\![0, N_L/2 - 1]\!], \ u_{i+1/2} = \frac{1}{\beta} \sum_{l=0}^{N_{st} - 1} \alpha_l (u_{i-l} + u_{i+1+l}), \beta \neq 0,$$
(C.1)

avec $N_{st} = (o_I + 1)/2$ la demi-largeur du stencil utilisé, α_l et β les coefficients d'interpolation. Ces derniers sont résumés en Tab. C.1 pour différents ordres d'interpolation.

Dans ce type de configuration, le traitement des bords ne nécessite aucune altération de (C.1), profitant des conditions périodiques pour obtenir des *stencils* de taille N_{st} aux bords, quel que soit l'ordre d'interpolation utilisé.

TABLE C.1 – Coefficients d'interpolation point milieu monodimensionnelle, pour différents ordres o_I allant de 1 à 15.

O_I	$ N_{st} $	α_l	β
1	1	{1}	2
3	2	$\{9, -1\}$	16
5	3	$\{150, -25, 3\}$	256
7	4	$\{1225, -245, 49, -5\}$	2048
9	5	$\{39690, -8820, 2268, -405, 35\}$	65536
11	6	$\{320166, -76230, 22869, -5445, 847, -63\}$	524288
13	7	$\{5153148, -1288287, 429429, -122694, 26026, -3549, 231\}$	8388608
15	8	$\{41409225, -10735725, 386486, -1254825, 325325, -61425, 7425, -429\}$	67108864

C.1.2 Extension aux maillages tridimensionnels

La formule présentée en (C.1), permet donc d'interpoler un champ dans une direction donnée. Comme mentionné par [Trottenberg et al., 2001, Sec. 2.3.4, Remark 2.3.3, p.43], il est tout a fait possible d'implémenter des procédures efficaces d'interpolation multi-dimensionnelles, basées sur des utilisations récursives de processus d'interpolation mono-dimensionnelle. Afin de présenter cette généralisation, on considère tout d'abord l'extension pour deux dimensions spatiales.

Soit donc un maillage 2D uniforme periodique, avec i et j les indices de maillage grossier utilisés respectivement pour les directions x et y, en reprenant les conventions de notation précédentes, avec N_{L_x} et N_{L_y} le nombre de points de maillage en x et y respectivement. On note $I_{d,l}(u)$ l'interpolation de u selon la direction d, en utilisant la formule (C.1) avec les valeurs des champs situées sur la ligne perpendiculaire à d, cette ligne ayant pour indice de maillage grossier l. L'interpolation sur l'ensemble du plan (x, y) se fait en trois étapes, représentées en Fig. C.1 :

FIGURE C.1 – Schéma des étapes d'interpolation point milieu uniforme dans un plan 2D(x, y).



(1) Interpolation selon la direction x, à partir des valeurs sur le maillage grossier :

$$u_{i+1/2,j} = I_{x,j}(u),$$
 (C.2)

(2) Interpolation selon la direction y, à partir des valeurs sur le maillage grossier :

$$u_{i,j+1/2} = I_{y,i}(u),$$
 (C.3)

(3) Moyenne des interpolations selon les directions x et y avec les valeurs des champs interpolées issues des étapes (1) et (2) :

$$u_{i+1/2,j+1/2} = \frac{I_{x,j+1/2}(u) + I_{y,i+1/2}(u)}{2}.$$
(C.4)

L'extension à des maillages 3D se fait donc par simple généralisation du processus décrit cidessus. On notera par ailleurs que cette méthode est tout à fait généralisable dans le cadre de maillage structurés non-uniformes, du moment que l'interpolation d'ordre variable selon chaque direction soit bien définie.

C.2 Extension aux maillages non-uniformes

On considère maintenant une répartition non uniforme pour les deux maillages périodiques $\Omega_{\mathcal{G}}$ et $\Omega_{\mathcal{F}}$, comme représenté en Fig. C.2. On considère une valeur à interpoler u_I , de coordonnée x_I . S'il



FIGURE C.2 – Interpolation sur un maillage 1D non uniforme, exemple d'un stencil pour une interpolation d'ordre 3.

existe un indice i du maillage $\Omega_{\mathcal{G}}$ tel que $x_i = x_I$, aucune interpolation n'est nécessaire, et $u_I = u_i$.

Dans le cas contraire, il existe un unique stencil de points du maillage grossier, de demi-largeur $N_{st} = (o_I + 1)/2$, encadrant u_I . On note i_0 l'indice de maillage grossier du premier point du stencil. De par les propriétés des espaces vectoriels polynomiales, on peut montrer qu'il existe un unique polynôme P d'ordre o_I prenant pour valeur u_i pour $i \in [[i_0, i_0 + o_I]]$. On calcule donc u_I en évaluant ce polynôme en x_I , ce qui nous permet d'écrire :

$$u_I = \sum_{i=i_0}^{i_0+o_I} c_i u_i,$$
(C.5)

avec c_i les coefficients d'interpolation dépendant des $o_I + 1$ points $(x_{i_0}, ..., x_{i_0+o_I})$ et de la position relative de x_I^{-1} . Afin de les déterminer, on pose $U = (u_{i_0}, ..., u_{i_0+o_I})^{\intercal}$ (vecteur colonne), et on exprime P comme suit :

$$P(x) = a_0 + a_1 x + \dots + a_{o_I} x^{o_I}.$$
(C.6)

Les coefficients polynomiaux sont donc obtenus en résolvant le système linéaire :

$$\begin{bmatrix} 1 & x_{i_0} & x_{i_0}^2 & \dots & x_{i_0}^{o_I} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{i_0+o_I} & x_{i_0+o_I}^2 & \dots & x_{i_0+o_I}^{o_I} \end{bmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{o_I} \end{pmatrix} = U,$$
(C.7)

^{1.} On peut noter que les formules d'interpolation données en Sec. C.1 sont en fait un cas particulier des formules que l'on présente dans cette section.

On désigne par M_V la matrice de Vandermonde issue de (C.7), ce qui permet d'écrire $u_I = P(x_I)$ comme le résultat du produit scalaire :

$$u_{I} = \begin{pmatrix} 1 \\ \vdots \\ x_{I}^{o_{I}} \end{pmatrix} \cdot \begin{pmatrix} a_{0} \\ \vdots \\ a_{o_{I}} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ x_{I}^{o_{I}} \end{pmatrix} \cdot M_{V}^{-1} U$$
(C.8)

En notant $C_{V,inv}(i)$ la $i^{\text{ème}}$ colonne de M_V^{-1} , par identification des coefficients c_i dans (C.8), on peut écrire :

$$c_{i} = \begin{pmatrix} 1 \\ \vdots \\ x_{I}^{o_{I}} \end{pmatrix} . C_{V,inv}(i), \text{ avec } M_{V}C_{V,inv}(i) = e_{i},$$
(C.9)

En pratique, il est possible d'obtenir $C_{V,inv}(i)$ de deux façons :

- Calcul explicite de M_V^{-1} . En particulier, celle-ci peut être obtenue analytiquement en utilisant [Higham, 2002, Algorithm 22.1].
- Inversion du système linéaire :

$$M_V C_{V,inv}(i) = e_i, \tag{C.10}$$

avec e_i correspondant au $i^{ième}$ vecteur de la base unitaire standard de \mathbb{R}^{o_I+1} . Cette approche est généralement préférée à une inversion explicite de M_V pour minimiser l'erreur numérique.

Considérations numériques Il est assez connu qu'en fonction des valeurs de $(x_{i_0}, ..., x_{i_0+o_I})$, M_V peut avoir un conditionnement élevé, ce qui peut induire une erreur numérique conséquente dans le calcul des c_i [Higham, 2002, Chap. 22]. Deux indicateurs ont été utilisés dans le cadre de la thèse afin de vérifier la précision des calculs :

• Erreur d'inversion :

$$\epsilon_{inv} = \left\| M_V M_V^{-1} - \mathbb{I}_{o_I + 1} \right\|_2 \text{ où } \epsilon_{inv}(i) = \left\| M_V C_{V,inv}(i) - e_i \right\|_2,$$
(C.11)

• Erreur de consistance :

$$\epsilon_{cons} = \left| \sum_{i=i_0}^{i_0+o_I} c_i \right|, \tag{C.12}$$

qui traduit l'exactitude de la formule d'interpolation pour un champ constant.

En pratique, une tolérance fixée à 10^{-8} a été utilisée pour ces deux erreurs dans le cadre de la thèse. Il a été remarqué que les calculs étaient particulièrement sensibles à la différence $|x_{i_0+o_I} - x_{i_0}|$. On utilise donc la méthodologie suivante : les c_i ne dépendant pas des coordonnées des points de maillage mais de leur position relative, on peut appliquer au préalable la transformation suivante pour $(x_{i_0}, ..., x_{i_0+o_I})$:

$$x^* = 2b \frac{x - x_{i_0}}{x_{i_0 + o_I} - x_{i_0}} - b.$$
(C.13)

En pratique, on a remarqué que b = 1 était une valeur bien adaptée, en accord avec les différentes bornes de conditionnement pour M_V détaillées par [Higham, 2002, Chap. 22].

C.3 Prise en compte de conditions limites non-périodiques.

La méthode présentée précédemment ne s'applique que lorsque l'on dispose suffisamment de points de maillage grossier autour du point interpolé. Alors que ceci est toujours garanti pour des maillages périodiques, ce n'est plus nécessairement le cas si l'on considère des conditions limites



FIGURE C.3 – Exemple de déplacement de stencil pour une interpolation d'ordre 3, pour des maillages non périodiques.

de type Dirichlet ou Neumann. Ainsi nous proposons ici de déplacer le stencil d'interpolation au voisinage des parois, comme représenté en Fig. C.3.

Dans le solveur HYBRID, les valeurs des champs ne sont pas explicitées à la frontière pour les directions non-périodiques. Deux configurations sont donc envisageables pour l'interpolation :

- les points les plus proches de la frontière ne sont pas "inclus" dans leur stencil, comme représenté en Fig. C.3a,
- utilisation de la valeur du champ à la frontière, comme représenté en Fig. C.3b.

Dans les deux cas, cela revient à borner l'indice i_0 du stencil à l'indice minimal (ou maximal) du maillage. Dans le deuxième cas, il est nécessaire de construire la valeur du champ à la frontière, que l'on note u_W . Selon les champs interpolés, deux formulations peuvent être envisagées :

Condition de type Dirichlet, une valeur constante u_C est fixée à la frontière :

$$u_W = u_C. \tag{C.14}$$

Condition de type Neumann homogène, une dérivée nulle du champ est imposée à la frontière :

$$\left(\frac{\partial u}{\partial x}\right)_{\text{Frontière}} = 0. \tag{C.15}$$

Pour simplifier, on considère la frontière du côté "gauche" du maillage, le raisonnement pour la frontière du coté "droit" pouvant être obtenu par symétrie. On note i_W l'indice de maillage grossier à la frontière. L'évaluation de (C.15) avec une différence finie d'ordre 1 donne :

$$\left(\frac{\partial u}{\partial x}\right)_{\text{Frontière}} \simeq \frac{u_{i_W+1} - u_{i_W}}{h_0} = 0, \tag{C.16}$$

avec $h_0 = x_{i_W+1} - x_{i_W}$. On en déduit la valeur du champ à la frontière :

$$u_W = u_{i_W} = u_{i_W+1}.$$
 (C.17)

C.4 Application pour des champs solutions obtenus avec le solveur Hybrid

Dans le cadre de la thèse, la méthode d'interpolation décrite précédemment a été appliquée à des champs tri-dimensionnels, comprenant les cinq variables $(\rho, \rho u, \rho v, \rho w, e_T)$.

Prise en compte des conditions aux bords pour le cas du canal turbulent Pour les résultats présentés en Sec. 4.4, les valeurs des champs aux frontières ont été utilisées. Une condition de Dirichlet homogène a été appliquée pour les champs de vitesse, avec $\rho u = \rho v = \rho w = 0$ à chaque frontière, et une condition de Neumann a été utilisée pour ρ et e_T .

Modélisation du coût d'interpolation pour le solveur Hybrid Pour l'ordre le plus élevé utilisé durant la thèse ($o_I = 7$), les étapes d'interpolation mono-dimensionnelle induisent une complexité similaire à celle de l'évaluation des gradients spatiaux avec une formule différence finie d'ordre 6 (15 et 11 FLOPs sont nécessaires, respectivement). De ce fait, le coût d'interpolation peut être relié au coût du calcul requis pour un pas de temps du solver grossier.

Pour HYBRID, on compte 39 évaluations de gradient pour l'ensemble des équations de Navier-Stokes (3 pour (4.25), 18 pour (4.26), 6 pour (4.27), 9 pour (4.29) et 3 pour (4.30)). Comme l'intégration temporelle Runge-Kutta explicite d'ordre 4 requiert 4 évaluations de ces dérivées, on estime à 156 le nombre total d'évaluations de gradient par pas de temps du solveur grossier.

Enfin, pour l'interpolation tridimensionnelle on compte 12 évaluations de gradient par champ, *i.e.* 60 pour les 5 variables des équations de Navier-Stokes.

ANNEXE D

Glossaire de mécanique des fluides

Décroissance d'une Turbulence Homogène Isotrope.

- Re_{λ} : nombre de Reynolds basé sur l'échelle transverse de Taylor,
- M_t : nombre de Mach basé sur les fluctuations turbulentes,
- λ : micro-échelle de Taylor,
- k_e : énergie cinétique turbulente moyenne,
- ϵ : dissipation moyenne,
- $E(\kappa)$: densité spectrale d'énergie cinétique turbulente associée au nombre d'onde tridimensionnel κ ,
- η : longueur caractéristique des structures dissipatives (échelle de Kolmogorov),
- au_η : temps caractéristique associé à l'échelle de Kolmogorov,
- au_{λ} temps caractéristique associé à l'échelle de Taylor,
- t_{ϕ} : temps de fin de période transitoire.

Ecoulement de Canal Turbulent.

- Re_{τ} : nombre de Reynolds basé sur le frottement pariétal,
- *Re_b* : nombre de Reynolds basé sur la vitesse débitante,
- M_b : nombre de Mach basé sur la vitesse débitante,
- ρ_b : densité volumique débitante,
- u_b : vitesse volumique débitante,
- T_c : temps caractéristique d'advection d'une particule sur la longueur du canal,
- u_{τ} : vitesse de frottement à la paroi,
- au_w : frottement à la paroi,
- δ^+ : échelle de longueur associée à la viscosité à la paroi,
- U: vitesse axiale moyenne,
- u'^2 , (uv)': tensions de Reynolds.

ANNEXE E

Spectre énergétique d'un écoulement turbulent

Table des matières

E.1	Turbulence Homogène Isotrope	• •	 •	•	•	•	 •	•	162										
E.2	Écoulement de Canal Turbulent	•	 •	•	•	 •	•	•	 •		•	 •	•	•	•	 •	•	163	

Dans cette annexe sont décrites les méthodologies permettant d'extraire les densités spectrales d'énergie turbulente pour les écoulements des deux problèmes tests tridimensionnels présentés dans cette thèse. On se focalise tout d'abord en Sec. E.1 sur la décroissance de Turbulence Homogène Isotrope, présentée en Sec. 4.3, puis sur l'Ecoulement de Canal Turbulent en Sec. E.2, ce dernier problème étant détaillé en Sec. 4.4.

E.1 Turbulence Homogène Isotrope

La méthodologie suivante se base sur les propriétés d'isotropie du champ turbulent, ainsi que l'utilisation d'un maillage structuré à maille uniforme dans les trois directions spatiales. Elle est largement utilisée dans la littérature, et la description donnée ici est principalement inspirée de [Jamme, 1998, Ap. B].

On note $\hat{u}(\kappa_x, \kappa_y, \kappa_z)$ la transformée de Fourier discrète (*Discrete Fourier Transform, DFT*) de u(x, y, z), en utilisant respectivement 1 $(1/N_L^3)$ pour normaliser la DFT (inverse). On définit ici κ_x , κ_y et κ_z les nombres d'onde monodimensionnel de la manière suivante :

$$\kappa_{[.]} = \left\{ n \frac{2\pi}{L} \text{ avec } n \in \mathbb{Z} \mid -\frac{N_L}{2} + 1 \le n \le \frac{N_L}{2} \right\},\tag{E.1}$$

auxquels on associe la norme vectorielle :

$$\kappa = \|\boldsymbol{\kappa}\| = \sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}.$$
(E.2)

L'espace de Fourier 3D est discrétisé en "coquilles" sphériques d'épaisseur $d\kappa^{-1}$. Ceci permet d'obtenir la densité spectrale d'énergie associée à une valeur de κ donnée en effectuant une somme cumulée sur l'ensemble des points situés à l'intérieur de la coquille sphérique :

$$E(\kappa_i) = \frac{1}{2N_L^6} \sum_{\kappa \in [\kappa_i, \kappa_i + d\kappa]} |\hat{\boldsymbol{u}}(\kappa_x, \kappa_y, \kappa_z)|^2.$$
(E.3)

Enfin, dans le cadre d'une représentation avec une échelle logarithmique, il est préférable de représenter E en fonction de la norme du nombre d'onde tridimensionnel au centre de la coquille sphérique, *i.e.* $\kappa' = \kappa + d\kappa/2$.

^{1.} Généralement , $d\kappa$ est fixé de telle manière que $\frac{2\pi d\kappa}{L} = 1$. Ce choix a par ailleurs été fait pour tout les résultats présentés dans cette thèse.

E.2 Écoulement de Canal Turbulent

Description générale. Pour ce problème, trois aspects principaux sont à prendre en compte :

- 1. Non-uniformité du maillage dans la direction y: le calcul du spectre de densité spectrale d'énergie étant basé sur des DFTs ne pouvant être réalisées que sur des maillages uniformes, les spectres sont donc calculés dans des plans (x, z) parallèles à la paroi.
- 2. Non-isotropie de l'écoulement : la présence d'un écoulement moyen non-nul dans la direction x induit une perte de l'isotropie, par conséquent on se focalise plutôt sur des spectres 1D concernant chaque direction des plans 2D (x,z), et pour chaque composante de la vitesse, séparément. En particulier, on distingue les deux directions suivante :
 - *Streamwise* (*x*, sens de l'écoulement), que l'on note par la suite avec l'indice *stw*.
 - *Spanwise* (*z*, sens perpendiculaire à l'écoulement), noté avec l'indice *spw*.

Enfin, les spectres calculés pour u, v, et w, sont représentés par la suite avec les indices uu, vv et ww, respectivement.

- 3. Étude des propriétés d'un champ moyenné : l'aspect statistiquement stationnaire de l'écoulement permet la construction d'une moyenne spatio-temporelle de spectres, et ce en trois étapes. Pour exemple, les spectres *spanwise* moyennés sont calculés de la manière suivante :
 - Moyenne spatiale sur l'ensemble des N_x lignes de direction z composant le plan (x,z) choisi,
 - Pour une coordonnée de plan $y \in [-1, 1]$ choisie, moyenne sur les deux plans de coordonnée y et -y.
 - Moyenne temporelle sur l'ensemble des échantillons temporels utilisés (cf. Sec. F.2).

Ainsi dans ce cas, un spectre pour une variable donnée est calculé à partir de la moyenne de $2N_{stat}N_x$ spectres 1D dans la direction z, N_{stat} étant le nombre d'échantillons temporels.

Calcul en pratique. On détaille ici le calcul de la densité spectrale moyenne associée à la fluctuation de la composante de vitesse u dans la direction *spanwise*, $E_{uu,spw}$. Une méthodologie analogue est utilisée pour les autres variables et directions.

On note par $\hat{u}_{i,y}(\kappa_z, t)$ la DFT 1D de $u'(t) = u(t) - \overline{u}(t)$ à l'instant t, sur une ligne d'indice i dans la direction x et de coordonnée y, en utilisant respectivement 1 ($1/N_z$) pour normaliser la DFT (inverse). La densité spectrale spatiale moyenne à l'instant t est calculée de la manière suivante :

$$E_{uu,spw}(\kappa_z,t) = \frac{1}{N_x} \sum_{i=1}^{N_x} \frac{|\hat{u}_{i,y}(\kappa_z,t)|^2 + |\hat{u}_{i,-y}(\kappa_z,t)|^2}{2}.$$
(E.4)

En définissant par $[t_1, ..., t_{N_{stat}}]$ l'ensemble des temps d'échantillonnage, on obtient ainsi :

$$E_{uu,spw}(\kappa_z) = \frac{1}{N_{stat}} \sum_{s=1}^{N_{stat}} E_{uu,spw}(\kappa_z, t_s).$$
(E.5)

Enfin, $E_{uu,spw}$ s'exprime en fonction de $\kappa_z \in [0, 1, ..., \kappa_{z,max}]$. Dans le cadre d'une représentation logarithmique, il est donc préférable de représenter $E_{uu,spw}$ en fonction de $\kappa_z + 0.5$.

ANNEXE **F**

Validation des Simulations Numériques Directes pour l'Écoulement de Canal Turbulent

Table des matières

F.1	Loi de maillage dans la direction y	4
F.2	Accumulation des échantillons statistiques 16	5
F.3	Validation du maillage spatial 160	6

Dans le cadre de la thèse, des simulations numériques directes d'écoulement de canal turbulent ont été effectuées, reproduisant le cas d'étude de [Moser et al., 1999] pour $Re_{\tau} = 590$. Dans cette annexe, on détaille l'ensemble des éléments nécessaires à la mise en place des DNS réalisées avec le solveur HYBRID.

En Sec. F.1, on décrit la loi d'évolution de taille de maille utilisée pour la répartition non-uniforme des points de grille dans la direction y. Puis, on valide en Sec. F.2 la procédure utilisée pour l'accumulation des statistiques de l'écoulement turbulent. Enfin, on évalue en Sec. F.3 les résultats obtenus avec HYBRID (solveur fin \mathcal{F} et grossier seul $\hat{\mathcal{G}}$), en prenant ceux de [Moser et al., 1999] comme référence. Plus d'informations sur \mathcal{F} et $\hat{\mathcal{G}}$ sont données en Sec. 4.4.2.a.

F.1 Loi de maillage dans la direction y

Afin de construire le maillage dans la direction y (normale aux parois), on considère une répartition uniforme de N_{L_y} points $\xi \in]-1, 1[$. Les coordonnées $y \in]-1, 1[$ sont déterminées en utilisant la loi suivante, dite "Gauss" :

$$y_{Gauss}(\xi) = \int_0^{\xi} e^{-\frac{|x|^a}{g}} dx \left[\int_0^1 e^{-\frac{|x|^a}{g}} dx \right]^{-1},$$
(F.1)

avec $g, a \in \mathbb{R}^*_+$. Pour le maillage en y utilisé pour \mathcal{F} , on a fixé g = 0.47 et a = 2.7.

Cette loi a pour objectif d'optimiser la répartition des points vis-à-vis de deux lois utilisées dans [Moser *et al.*, 1999, Trettel et Larsson, 2016] :

$$y_{Sin}(\xi) = -\cos\left(\frac{\pi(\xi+1)}{2}\right),\tag{F.2}$$

$$y_{Tan}(\xi) = \frac{\tan(g\xi)}{\tan(g)}, \ g \in \mathbb{R}_+^*.$$
(F.3)

En particulier, la loi Gauss permet de ne pas accumuler des points de maillage dans la zone très proche de la paroi¹ (comme la loi Tan), tout en conservant une bonne taille de maille en milieu de

^{1.} Cette condition est primordiale pour des solveurs compressibles explicites, afin d'éviter d'imposer des pas de temps de trop petite taille, les CFL les plus importants durant la simulation étant obtenus le plus près de la paroi.

canal (comme la loi Sin). Une comparaison des maillages obtenus pour les différentes lois avec un même nombre de points $N_{L_y} = 255$ est donnée en Fig F.1.



FIGURE F.1 – Évolution de la taille de maille en fonction de la position relativement à la paroi, pour différentes lois d'évolution de maille avec le même nombre de points dans la direction y ($N_{L_y} = 255$). Le coefficient de la loi Tan est g = 1.89, et pour la loi Gauss les coefficients choisis sont g = 0.47 et a = 2.7. Les tailles sont représentées pour une maille sur deux.

F.2 Accumulation des échantillons statistiques

Afin d'extraire les profils moyens de vitesse et tension de Reynolds, une moyenne temporelle statistique est réalisée sur un nombre d'échantillons N_{stat} extrait à des intervalles de temps régulier $\Delta_{TS} = 2\delta_t N_{\Delta_t,\widehat{\mathcal{G}}}$, avec δ_t le pas de temps du solveur fin et $N_{\Delta_t,\widehat{\mathcal{G}}}$ le nombre de pas de temps de $\widehat{\mathcal{G}}$ par *time-slice*. L'interval total d'intégration [0,T] étant gardé fixe pour chaque valeur de $N_{\Delta_t,\widehat{\mathcal{G}}}$ considérée, cela induit donc une diminution du nombre d'échantillons avec l'augmentation de la taille des *time-slices*.

Afin d'évaluer la convergence statistique des moyennes temporelles, on note la moyenne partielle pour un vecteur donné ψ :

$$\forall s \in [1, N_{stat}], \ M_s(\psi) = \frac{1}{s} \sum_{i=1}^{s} \psi_i,$$
 (F.4)

avec $(\psi_i)_{1 \le i \le N_{stat}}$ l'ensemble des échantillons. On définit alors la variation relative de la moyenne partielle comme étant :

$$\forall s \in [2, N_{stat}], \ V_s(\psi) = \frac{\|M_s(\psi) - M_{s-1}(\psi)\|_2}{\|M_{s-1}(\psi)\|_2}.$$
(F.5)

Cette dernière nous permet donc d'évaluer l'apport relatif de chaque échantillon s ajouté à la moyenne partielle déjà calculée.

En Fig. F.2, on représente V_s pour le solveur fin et $\hat{\mathcal{G}}$ seul, pour les configurations où N_{stat} est le plus élevé ($N_{\Delta_t,\widehat{\mathcal{G}}} = 50$) et le plus faible ($N_{\Delta_t,\widehat{\mathcal{G}}} = 400$), et ce pour les variables principales de l'écoulement ($\overline{u}, \overline{v}, \overline{w}, \overline{uv}$).

On peut y observer que V_s semble converger pour l'ensemble des variables, vers une valeur qui est au maximum de l'ordre de 1%. En particulier, les valeurs finales apparemment élevées pour \overline{v} et \overline{w} sont à mettre en relation avec les valeurs nulles de ces moyennes à convergence statistique au fait que pour un nombre suffisant d'échantillons considérés, les moyennes temporelles de ces variables sont très proches de 0.



FIGURE F.2 – Évolution de V_s pour les variables principales de l'écoulement, dans plusieurs configurations d'étude. Un ordonnancement aléatoire est utilisé pour l'ensemble des échantillons.

F.3 Validation du maillage spatial

Afin de valider le maillage fin utilisé pour la DNS avec Hybrid, on compare les résultats obtenus avec ceux de la base de donnée de [Moser *et al.*, 1999] pour $Re_{\tau} = 590$. Afin de donner un premier aperçu de la convergence en maillage pour Hybrid, on ajoute à la comparaison les résultats obtenus avec $\hat{\mathcal{G}}$ seul. Les moyennes temporelles pour Hybrid ont été obtenues à partir de $N_{stat} = 101$ échantillons ($N_{\Delta t, \hat{\mathcal{G}}} = 400$).

En Fig. F.3 sont représentés les profils normalisés en unités de paroi pour U^+ , u'^+ et $-(uv)'^+$. Une comparaison des densités spectrales d'énergie cinétique turbulente moyennés dans la direction transverse de l'écoulement est donnée en Fig. F.4, pour deux valeurs de y^+ (proche de la paroi, et vers le milieu du canal).

Les résultats sont globalement particulièrement satisfaisant, comparés à ceux de la DNS de [Moser *et al.*, 1999]. On note néanmoins une légère sur-estimation du pic de $-(uv)'^+$ pour Hybrid (solveur fin et $\hat{\mathcal{G}}$ seul) par rapport à cette dernière référence. Enfin, on observe une bonne coïcidence entre les courbes de densité spectrale d'énergie obtenues pour \mathcal{F} et par [Moser *et al.*, 1999]. En particulier, les fins de spectre ($\kappa_{[..]} > 10^2$) pour Hybrid avec le solveur fin confirment que l'on a bien réalisé une DNS.



FIGURE F.3 – Profils moyens normalisés en unités de paroi pour $Re_{\tau} = 590$. Résultats pour le solveur fin (trait, cercles), pour le solveur $\hat{\mathcal{G}}$ seul sans interpolation (tirets, carrés) et résultats pour la DNS de [Moser et al., 1999] (pointillés, triangles).



FIGURE F.4 – Densités spectrales d'énergie cinétique turbulente moyennés dans la direction transverse pour $Re_{\tau} = 590$. Résultats pour le solveur fin (trait, cercles), pour le solveur $\hat{\mathcal{G}}$ seul sans interpolation (tirets, carrés) et résultats pour la DNS de [Moser et al., 1999] (pointillés, triangles).

ANNEXE G

Approximation de commutativité pour le facteur d'amplification de PARAREAL avec grossissement spatial

Table des matières

G.1	Besoi	n de l'approximation 169
	G.1.1	Relation de récurrence de Pascal & algèbre non-commutative 170
	G.1.2	Inconvénients de l'utilisation de la formule exacte
G.2	Valida	ation de l'approximation 171
	G.2.1	Comparaison avec les résultats numériques
	G.2.2	Lien avec l'analyse de Fourier pour les méthodes multi-grilles

Dans cette annexe, on examine plus en détail l'approximation de commutativité faite en Sec. 3.2.4 afin d'obtenir une expression pour le facteur d'amplification de PARAREAL avec grossissement spatial. Tout d'abord, on met en évidence en G.1 le besoin d'une telle approximation, en analysant l'étude pouvant être faite en prenant une formulation exacte. Puis en Sec. G.2, on valide l'approximation et l'expression obtenue pour le facteur d'amplification, en comparant sur un large panel d'applications numériques. Pour finir, ce type d'approximation est mis en parallèle avec des techniques classiques

G.1 Besoin de l'approximation

utilisées pour l'analyse de Fourier des méthodes multi-grille.

Comme observé en Sec. 3.2.4.c, la non-commutativité des opérateurs $M_{\mathcal{F}}$ et $M_{\widehat{\mathcal{G}}}$ ne permet pas d'utiliser la formule binomiale de la Prop. 3.2.2, ce qui induit :

$$M_{\mathcal{P}}(k,n) \neq g_{\mathcal{P}}^{k,n}(M_{\mathcal{F}},M_{\widehat{\mathcal{G}}}) = \sum_{i=0}^{k} \binom{n}{i} (M_{\mathcal{F}}-M_{\widehat{\mathcal{G}}})^{i} M_{\widehat{\mathcal{G}}}^{n-i}.$$
 (G.1)

Il est donc nécessaire de repartir de la relation de récurrence d'une itération de l'algorithme pour obtenir l'expression analytique exacte de la solution numérique de PARAREAL avec grossissement spatial, dans l'espace de Fourier.

On détaille donc en Sec. G.1.1 la formule exacte, et on explique en Sec. G.1.2 pourquoi cette formulation est inadaptée pour analyser le comportement de l'algorithme, d'où la nécessité d'utiliser l'approximation de commutativité.

Relation de récurrence de Pascal & algèbre non-commutative G.1.1

Définition G.1.1 – Soit $M_1, M_2 \in \mathcal{M}_p(\mathbb{C})$, ne commutant pas et $V_0 \in \mathbb{C}^p$. On définit par $\widehat{g}_{\mathcal{P}}^{k,n}(M_1, M_2)$ l'opérateur matriciel permettant d'obtenir :

$$V_n^k = \widehat{g}_{\mathcal{P}}^{k,n}(M_1, M_2)V_0$$

avec la suite $(V)_n^k$ vérifiant :

$$\forall (k,n) \in \mathbb{N} \times \mathbb{N}^*, \ V_n^k = M_1 V_{n-1}^k + (M_2 - M_1) V_{n-1}^{k-1}, \tag{G.2}$$

(G.3)

 $\forall k \in \mathbb{N}, \ V_0^k = V_0, \\ \forall n \in \mathbb{N}^*, \ V_n^0 = M_2^n V_0.$ (G.4)

Observation G.1.1 – Si M_1 et M_2 commutent, ou k = 0, alors $\hat{g}_{\mathcal{P}}^{k,n}(M_1, M_2) = g_{\mathcal{P}}^{k,n}(M_1, M_2)$, avec $g_{\mathcal{P}}^{k,n}$ défini en Prop. 3.2.5.

Proposition G.1.1 – Soit l'application de PARAREAL avec grossissement spatial au problème d'advection mono-dimensionnel (3.4), sur des grilles spatiales périodiques uniformes, utilisant une restriction de type injection et une interpolation linéaire pour transférer la solution sur chaque niveau de grille respectif. On considère les paramètres suivants pour chaque propagateur :

- 1. Nombres CFL : CFL_F et CFL_{\hat{c}} (pouvant s'écrire en fonction de CFL_F),
- 2. Symboles de Fourier : $z_{adv,\mathcal{F}}$ et $z_{adv,\widehat{\mathcal{G}}}$:
- 3. Facteurs d'amplifications : $g_{\mathcal{F}}$ et $g_{\widehat{G}}$,
- 4. Nombres de pas de temps par time-slice : $s_{\mathcal{F}}$ et $s_{\widehat{\mathcal{C}}}$.

En posant les notations :

$$R(\theta_{\kappa}) = g_{\widehat{\mathcal{G}}}(-CFL_{\widehat{\mathcal{G}}}z_{adv,\widehat{\mathcal{G}}}(\theta_{\kappa}))^{s_{\widehat{\mathcal{G}}}},$$

$$r(\theta_{\kappa}) = g_{\mathcal{F}}(-CFL_{\mathcal{F}}z_{adv,\mathcal{F}}(\theta_{\kappa}))^{s_{\mathcal{F}}},$$

on définit les matrices (2×2) suivantes :

$$\begin{split} \widehat{M}_{\mathcal{F}}(\theta_{\kappa}) &= \begin{bmatrix} r(\theta_{\kappa}) \\ r(\theta_{\kappa} + \pi) \end{bmatrix}, \\ \widehat{M}_{\widehat{\mathcal{G}}}(\theta_{\kappa}) &= \begin{bmatrix} g_{IR}(\theta_{\kappa})R(2\theta_{\kappa}) & g_{IR}(\theta_{\kappa})R(2\theta_{\kappa}) \\ g_{IR}(\theta_{\kappa} + \pi)R(2\theta_{\kappa}) & g_{IR}(\theta_{\kappa} + \pi)R(2\theta_{\kappa}) \end{bmatrix}, \end{split}$$

avec g_{IR} défini en Prop. 3.2.7. Dans ces conditions, la solution de PARAREAL avec grossissement spatial pour le nombre d'onde θ_{κ} à la fin d'une time-slice n à l'itération k (différent de n) s'écrit :

$$\forall \theta_{\kappa} \in [0,\pi], \quad \hat{U}_{n}^{k}(\theta_{\kappa}, CFL_{\mathcal{F}}) = \begin{bmatrix} 1 & 0 \end{bmatrix} \widehat{g}_{\mathcal{P}}^{k,n}(\widehat{M}_{\widehat{\mathcal{G}}}, \widehat{M}_{\mathcal{F}}) \begin{bmatrix} \widehat{U}_{0}(\theta_{\kappa}) \\ \widehat{U}_{0}(\theta_{\kappa} + \pi) \end{bmatrix}. \tag{G.5}$$

Démonstration. À partir de l'expression de $M_{\mathcal{F}}$ et $M_{\widehat{G}}$ dans l'espace de Fourier, on effectue le changement de variable détaillé en Obs. 3.2.6 couplant les fréquences associées à θ_{κ} et $\theta_{\kappa} + \pi$, ce qui permet d'écrire les opérateurs fin et grossier dans l'espace de Fourier sous forme diagonale par blocs, ces derniers étant respectivement $M_{\mathcal{F}}(\theta_{\kappa})$ et $M_{\widehat{\mathcal{C}}}(\theta_{\kappa})$.

Puis, l'expression de $\hat{U}_n^k(\theta_{\kappa}, CFL_{\mathcal{F}})$ est obtenue par simple application de PARAREAL sur chaque bloc, en utilisant Def. G.1.1.

G.1.2 Inconvénients de l'utilisation de la formule exacte

Tout d'abord, la Prop. G.1.1 donne une formulation matricielle de la solution de PARAREAL dans l'espace de Fourier, qui semble peu exploitable. Pour exemple, le calcul de $\hat{U}_n^k(\theta_{\kappa}, CFL_{\mathcal{F}})$ pour k = 1 et n = 2 donne :

$$\hat{U}_{2}^{1}(\theta_{\kappa}, CFL_{\mathcal{F}}) = R(2\theta_{\kappa})g_{IR}(\theta_{\kappa})(2r(\theta_{\kappa}) - R(2\theta_{\kappa}))\hat{U}_{0}(\theta_{\kappa}) + R(2\theta_{\kappa})g_{IR}(\theta_{\kappa})(r(\theta_{\kappa}) + r(\theta_{\kappa} + \pi) - R(2\theta_{\kappa}))\hat{U}_{0}(\theta_{\kappa} + \pi).$$
(G.6)

Bien que cette formule corresponde à une application "simple" de PARAREAL (très peu d'itérations et de *time-slices*), elle ne permet pas d'obtenir une formulation "simple" du facteur d'amplification comme pour la Prop. 3.2.6. Et ceci est d'autant plus marqué dans le cadre général, la complexité de la formule augmentant de manière exponentielle avec k et n.

G.2 Validation de l'approximation

L'approximation de commutativité décrite en Sec. 3.2.4.c permet de :

- simplifier la forme générale de la Jacobienne de PARAREAL dans l'espace de Fourier, afin d'obtenir une formulation scalaire simple,
- de rendre les deux bandes supérieures de cette dernière matrice Jacobienne exactes, permettant ainsi de faire apparaître le facteur g_{TG}.

Grâce à cela on obtient une modélisation du facteur d'amplification de PARAREAL avec grossissement spatial, ce dernier pouvant être utilisé pour analyser les différents mécanismes de l'algorithme (cf. Sec. 3.3.3).

Cependant, comme il l'a été observé en Sec. 3.3.3.a, le modèle présente certaines différences notables avec les résultats numériques. L'objectif de cette section est donc de compléter les arguments déjà introduits en Sec. 3.3.3.a pour justifier la pertinence du modèle. En premier lieu, on le compare en Sec. G.2.1 avec des résultats numériques en faisant varier différents paramètres de l'algorithme sur une plage plus large. Puis, on complète l'argumentation en liant cette approximation à une pratique couramment effectuée pour l'analyse de Fourier des méthodes multi-grilles.

G.2.1 Comparaison avec les résultats numériques

On reprend la solution initiale de type Gaussienne utilisée en Sec. 3.3.3.a, mais en faisant cette fois varier les schémas d'interpolation utilisés (ordre $o_I \in \{1, 7\}$) ainsi que la taille des *time-slices* $(s_{\widehat{G}} \in \{10, 20, 50\})$. On représente en Fig. G.1 les spectres de Fourier pour les solutions obtenues avec le modèle et numériquement, respectivement.

Comme il l'a déjà été observé précédemment en Sec. 3.3.3.a, les résultats modélisés et théoriques concordent particulièrement bien sur les basses fréquences de la solution. On observe aussi une nette différenciation pour les hautes fréquences, en particulier pour le pic d'amplification induit par le changement de grille sur les plus hautes fréquences. Cependant, malgré ces différences, le modèle représente globalement bien l'évolution moyenne du spectre de Fourier de la solution, que ce soit pour la diffusion plus importante apportée sur les hautes fréquences par l'augmentation de l'ordre d'interpolation ($o_I = 7$) ou encore la démultiplication des pics d'amplification fréquentielle induite par l'augmentation de la taille des *time-slices* ($s_{\widehat{G}} = 50$).



FIGURE G.1 – Validation de la modélisation analytique de PARAREAL avec grossissement spatial vis-à-vis des résultats numériques. Spectres de Fourier de la solution initiale et finale obtenues avec PARAREAL, N = 4, RKC4-C6, restriction de type injection, $CFL_{\mathcal{F}} = CFL_{\widehat{\mathcal{G}}} = 1$. Différentes valeurs sont utilisées pour la taille des time-slices ($s_{\widehat{\mathcal{G}}} \in \{10, 20, 50\}$) et l'ordre d'interpolation ($o_I \in \{1, 7\}$).

Observation G.2.1 — De manière générale, la modélisation du facteur d'amplification basée sur l'approximation de commutativité permet d'obtenir une bonne approximation du comportement de l'algorithme sur l'ensemble des composantes fréquentielles de la solution, malgré une certaine démarcation sur les hautes fréquences. En particulier, ces dernières différences se caractérisent principalement par une sur-estimation par le modèle des amplifications apportées par l'algorithme.

G.2.2 Lien avec l'analyse de Fourier pour les méthodes multi-grilles

L'approximation de commutativité peut aussi être comparée à l'analyse à deux grilles simplifiée, présentée notamment par [Trottenberg *et al.*, 2001, Remark 4.6.1]. Dans cette dernière, on étudie le comportement de l'opérateur de correction de grille grossière, noté K_h^{2h} , en utilisant l'arrangement de la base de Fourier décrit en Obs. 3.2.6. Pour un problème mono-dimensionnel, K_h^{2h} peut ainsi s'écrire comme une matrice (2 × 2 au plus) :

$$K_h^{2h} = \begin{bmatrix} k_{1,1} & k_{1,2} \\ k_{2,1} & k_{2,2} \end{bmatrix},$$
(G.7)

avec $k_{1,1}$ et $k_{2,2}$ représentant respectivement les basses et hautes fréquences, tandis que $k_{2,1}$ et $k_{2,2}$ représentent le couplage entre hautes et basses fréquences. À la différence de l'Analyse Locale de Fourier (*Local Fourier Analysis* - LFA), qui étudie l'ensemble des termes exacts de K_h^{2h} , l'analyse à deux grilles simplifiée néglige l'influence de K_h^{2h} pour les hautes fréquences et leur couplage avec les basses fréquences, *i.e.* $k_{2,2} = k_{2,1} = k_{1,2} = 0$.

L'approximation de commutativité peut donc être vue comme une amélioration de l'analyse à deux grilles simplifiée, de par les deux points suivants :

- l'influence des hautes fréquences n'est pas négligée, et le terme exact est utilisé pour $k_{2,2}$,
- pour une fréquence donnée de la solution après correction, le terme de couplage hautes-basses fréquences est modélisé comme étant égal au terme de correction sur cette même fréquence de la solution avant correction, *i.e.* $k_{1,2} = k_{1,1}$ et $k_{2,1} = k_{2,2}$.

De ce fait, l'approximation de commutativité permet donc d'obtenir une méthode à mi-chemin entre l'analyse à deux grilles simplifiée et la LFA, cette dernière étant difficilement applicable et exploitable dans le cadre de l'utilisation de PARAREAL avec grossissement spatial.

Bibliographie

- [Alhubail et Wang, 2016] ALHUBAIL, M. et WANG, Q. (2016). The swept rule for breaking the latency barrier in time advancing PDEs. *J. Comp. Phys.*, 307:110–121.
- [Arteaga *et al.*, 2015] ARTEAGA, A., RUPRECHT, D. et KRAUSE, R. (2015). A stencil-based implementation of Parareal in the C++ domain specific embedded language STELLA. *Appl. Math. Comp.*, 267:727–741.
- [Aubanel, 2011] AUBANEL, E. (2011). Scheduling of tasks in the Parareal algorithm. *Parallel Comput.*, 37(3):172–182.
- [Audouze *et al.*, 2009] AUDOUZE, C., MASSOT, M. et VOLZ, S. (2009). Symplectic multi-time step Parareal algorithms applied to molecular dynamics. Rapport technique, HAL-00358459.
- [Baffico *et al.*, 2002] BAFFICO, L., BERNARD, S., MADAY, Y., TURINICI, G. et ZÉRAH, G. (2002). Parallelin-time molecular-dynamics simulations. *Physical Review E*, 66(5):057701.
- [Bal, 2005] BAL, G. (2005). On the convergence and the stability of the Parareal algorithm to solve partial differential equations. *In* KORNHUBER, R. et et AL, éditeurs : *Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering*, volume 40, pages 426–432. Springer.
- [Bal et Maday, 2002] BAL, G. et MADAY, Y. (2002). A "Parareal" time discretization for non-linear PDE's with application to the pricing of an american put. *Lecture Notes in Computational Science and Engineering*, 23:189–202.
- [Bergman et al., 2008] BERGMAN, K., BORKAR, S., CAMPBELL, D., CARLSON, W., DALLY, W., DENNEAU, M., FRANZON, P., HARROD, W., HILL, K., H., J. et al. (2008). Exascale computing study : Technology challenges in achieving exascale systems. Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep, 15.
- [Bermejo-Moreno et al., 2013a] BERMEJO-MORENO, I., BODART, J., LARSSON, J., BARNEY, B., NICHOLS, J. et JONES, S. (2013a). Solving the compressible Navier-Stokes equations on up to 1.97 million cores and 4.1 trillion grid points. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, page 62. ACM.
- [Bermejo-Moreno *et al.*, 2013b] BERMEJO-MORENO, I., LARSSON, J., BODART, J. et VICQUELIN, R. (2013b). Wall-modeled large-eddy simulations of the HIFiRE-2 scramjet. *CTR Annual Research Briefs*.
- [Berry *et al.*, 2012] BERRY, L. A., ELWASIF, W., REYNOLDS-BARREDO, J., SAMADDAR, D., SANCHEZ, R. et NEWMAN, D. E. (2012). Event-based Parareal : A data-flow based implementation of Parareal. *J. Comp. Phys.*, 231(17):5945–5954.
- [Bogey et Bailly, 2004] BOGEY, C. et BAILLY, C. (2004). A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *J. Comp. Phys.*, 194(1):194–214.
- [Bolten *et al.*, 2016] Воlten, M., Moser, D. et Speck, R. (2016). A multigrid perspective on the parallel full approximation scheme in space and time. *Numer. Linear Algebra Appl.*, pages e2110–n/a. e2110 nla.2110.

- [Bolten *et al.*, 2017] BOLTEN, M., MOSER, D. et SPECK, R. (2017). Asymptotic convergence of the parallel full approximation scheme in space and time for linear problems. *arXiv preprint arXiv*:1703.07120.
- [Botchev, 2013] BOTCHEV, M. A. (2013). A block Krylov subspace time-exact solution method for linear ordinary differential equation systems. *Numer. Linear Algebra Appl.*, 20(4):557–574.
- [Briggs et al., 2000] BRIGGS, W., HENSON, V. E. et MCCORMICK, S. (2000). A Multigrid Tutorial. SIAM.
- [Burmeister et Horton, 1991] BURMEISTER, J. et HORTON, G. (1991). Time-parallel multigrid solution of the Navier-Stokes equations. *In Multigrid methods III*, pages 155–166. Springer.
- [Butcher, 2016] BUTCHER, J. C. (2016). Numerical methods for ordinary differential equations. John Wiley & Sons.
- [Chartier et Philippe, 1993] CHARTIER, P. et PHILIPPE, B. (1993). A parallel shooting technique for solving dissipative ODE's. *Computing*, 51(3-4):209–236.
- [Chen *et al.*, 2015] CHEN, F., HESTHAVEN, J. S., MADAY, Y. et NIELSEN, A. S. (2015). An adjoint approach for stabilizing the Parareal method. Rapport technique, EPFL-ARTICLE-211097.
- [Chen *et al.*, 2014] CHEN, F., HESTHAVEN, J. S. et ZHU, X. (2014). On the use of reduced basis methods to accelerate and stabilize the Parareal method. *In Reduced Order Methods for Modeling and Computational Reduction*, pages 187–214. Springer.
- [Chiou et al., 2007] CHIOU, D., SUNWOO, D., KIM, J., PATIL, N. A., REINHART, W., JOHNSON, D. E., KEEFE, J. et ANGEPAT, H. (2007). FPGA-Accelerated Simulation Technologies (FAST) : Fast, full-system, cycle-accurate simulators. In Proceedings of the 40th Annual IEEE/ACM international Symposium on Microarchitecture, pages 249–261. IEEE Computer Society.
- [Choi et Moin, 1994] Сноі, H. et Moin, P. (1994). Effects of the computational time step on numerical solutions of turbulent flow. *J. Comp. Phys.*, 113(1):1–4.
- [Choi et al., 2013] Сної, J. W., BEDARD, D., FOWLER, R. et VUDUC, R. (2013). A roofline model of energy. In Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on, pages 661–672. IEEE.
- [Christlieb et Ong, 2011] CHRISTLIEB, A. et ONG, B. (2011). Implicit parallel time integrators. J. Sci. Comput., 49(2):167–179.
- [Christlieb *et al.*, 2009] CHRISTLIEB, A., ONG, B. et QIU, J.-M. (2009). Comments on high-order integrators embedded within integral deferred correction methods. *Comm. App. Math. and Comp. Sci.*, 4(1):27–56.
- [Christlieb *et al.*, 2010a] CHRISTLIEB, A., ONG, B. et QIU, J.-M. (2010a). Integral deferred correction methods constructed with high order Runge-Kutta integrators. *Mathematics of Computation*, 79(270):761–783.
- [Christlieb et al., 2010b] CHRISTLIEB, A. J., MACDONALD, C. B. et ONG, B. W. (2010b). Parallel highorder integrators. SIAM J. Sci. Comput., 32(2):818–835.
- [Crank et Nicolson, 1947] CRANK, J. et NICOLSON, P. (1947). A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *In Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press.
- [Croce *et al.*, 2014] CROCE, R., RUPRECHT, D. et KRAUSE, R. (2014). Parallel-in-space-and-time simulation of the three-dimensional, unsteady Navier-Stokes equations for incompressible flow. *In* Воск, H., HOANG, X., RANNACHER, R. et SCHLÖDER, J., éditeurs : *Modeling, Simulation and Optimization of Complex Processes-HPSC 2012*, pages 13–23. Springer.
- [Dagum et Menon, 1998] DAGUM, L. et MENON, R. (1998). OpenMP : an industry standard API for shared-memory programming. *IEEE computational science and engineering*, 5(1):46–55.

- [Dai *et al.*, 2013] DAI, X., LE BRIS, C., LEGOLL, F. et MADAY, Y. (2013). Symmetric Parareal algorithms for Hamiltonian systems. *ESAIM : Mathematical Modelling and Numerical Analysis*, 47(3):717–742.
- [Dai et Maday, 2013] DAI, X. et MADAY, Y. (2013). Stable Parareal in time method for first-and second-order hyperbolic systems. *SIAM J. Sci. Comput.*, 35(1):A52–A78.
- [Descombes et Massot, 2004] DESCOMBES, S. et MASSOT, M. (2004). Operator splitting for nonlinear reaction-diffusion systems with an entropic structure : singular perturbation and order reduction. *Numerische Mathematik*, 97(4):667–698.
- [Dobrev *et al.*, 2016] DOBREV, V., KOLEV, T., PETERSSON, N. et SCHRODER, J. (2016). Two-level convergence theory for multigrid reduction in time (MGRIT). Rapport technique, LLNL-JRNL-692418.
- [Dongarra *et al.*, 2011] DONGARRA, J., BECKMAN, P., MOORE, T., AERTS, P., ALOISIO, G., ANDRE, J., BAR-KAI, D., BERTHOU, J., BOKU, T., BRAUNSCHWEIG, B. *et al.* (2011). The international exascale software project roadmap. *Internat. J. HPC Appl.*, 25(1):3–60.
- [Dongarra *et al.*, 2014] DONGARRA, J., HITTINGER, J., BELL, J., CHACON, L., FALGOUT, R., HEROUX, M., HOVLAND, P., NG, E., WEBSTER, C. et WILD, S. (2014). Applied mathematics research for exascale computing. Rapport technique LLNL-TR-651000, Lawrence Livermore National Laboratory, USA.
- [Dongarra et al., 1979] DONGARRA, J., MOLER, C., BUNCH, J. et STEWART, G. (1979). LINPACK users' guide. SIAM.
- [Dongarra et al., 1990] DONGARRA, J. J., DU CROZ, J., HAMMARLING, S. et DUFF, I. S. (1990). A set of level 3 basic linear algebra subprograms. ACM Transactions on Mathematical Software (TOMS), 16(1):1–17.
- [Donzis *et al.*, 2008] DONZIS, D., YEUNG, P. et SREENIVASAN, K. (2008). Dissipation and enstrophy in isotropic turbulence : Resolution effects and scaling in direct numerical simulations. *Phys. Fluids*, 20(4):045108.
- [Duarte *et al.*, 2013] DUARTE, M., DESCOMBES, S., TENAUD, C., CANDEL, S. et MASSOT, M. (2013). Timespace adaptive numerical methods for the simulation of combustion fronts. *Combustion and Flame*, 160(6):1083–1101.
- [Duarte *et al.*, 2011] DUARTE, M., MASSOT, M. et DESCOMBES, S. (2011). Parareal operator splitting techniques for multi-scale reaction waves : Numerical analysis and strategies. *ESAIM : Mathematical Modelling and Numerical Analysis*, 45(5):825–852.
- [Dutt *et al.*, 2000] DUTT, A., GREENGARD, L. et ROKHLIN, V. (2000). Spectral deferred correction methods for ordinary differential equations. *BIT*, 40(2):241–266.
- [Eckmann et Ruelle, 1985] ECKMANN, J.-P. et RUELLE, D. (1985). Ergodic theory of chaos and strange attractors. *Reviews of modern physics*, 57(3):617.
- [Eghbal *et al.*, 2017] EGHBAL, A., GERBER, A. G. et AUBANEL, E. (2017). Acceleration of unsteady hydrodynamic simulations using the Parareal algorithm. *J. Comput. Sci.*, 19:57–76.
- [Emmett et Minion, 2012] EMMETT, M. et MINION, M. (2012). Toward an efficient parallel in time method for partial differential equations. *Comm. App. Math. and Comp. Sci.*, 7(1):105–132.
- [Emmett et Minion, 2014] EMMETT, M. et MINION, M. L. (2014). Efficient implementation of a multilevel parallel in time algorithm. In Domain Decomposition Methods in Science and Engineering XXI, pages 359–366. Springer.
- [Emmett *et al.*, 2014] EMMETT, M., ZHANG, W. et BELL, J. B. (2014). High-order algorithms for compressible reacting flow with complex chemistry. *Combust. Theory Model.*, 18(3):361–387.
- [Falgout *et al.*, 2016] FALGOUT, R., S., F., KOLEV, T. V., S.P., M., J.B., S. et S., V. (2016). Multigrid methods with space-time concurrency. Rapport technique, LLNL-JRNL-678572.
- [Falgout *et al.*, 2014a] FALGOUT, R. D., FRIEDHOFF, S., KOLEV, T. V., MACLACHLAN, S. P. et SCHRODER, J. B. (2014a). Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):C635–C661.
- [Falgout et al., 2014b] FALGOUT, R. D., KATZ, A., KOLEV, T. V., SCHRODER, J. B., WISSINK, A. et YANG, U. M. (2014b). Parallel time integration with multigrid reduction for a compressible fluid dynamics application. Rapport technique LLNL-JRNL-663416, Lawrence Livermore National Laboratory, USA.
- [Falgout et al., 2017] FALGOUT, R. D., MANTEUFFEL, T. A., O'NEILL, B. et SCHRODER, J. B. (2017). Multigrid reduction in time for nonlinear parabolic problems : A case study. SIAM Journal on Scientific Computing, 39(5):S298–S322.
- [Farhat *et al.*, 2006] FARHAT, C., CORTIAL, J., DASTILLUNG, C. et BAVESTRELLO, H. (2006). Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. *Internat. J. Numer. Methods Engrg.*, 67(5):697–724.
- [Fischer et al., 2017] FISCHER, L., GOTSCHEL, S. et WEISER, M. (2017). Lossy data compression reduces communication time in hybrid time-parallel integrators. Rapport technique, ZIB-Report : ISSN1438-0064.
- [Fischer et al., 2005] FISCHER, P. F., HECHT, F. et MADAY, Y. (2005). A Parareal in time semi-implicit approximation of the Navier-Stokes equations. In KORNHUBER, R. et et AL, éditeurs : Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, volume 40, pages 433–440. Springer.
- [Friedhoff et al., 2013] FRIEDHOFF, S., FALGOUT, R., KOLEV, T., MACLACHLAN, S. et SCHRODER, J. (2013). A multigrid-in-time algorithm for solving evolution equations in parallel. In Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States.
- [Gander, 2008] GANDER, M. (2008). Analysis of the Parareal algorithm applied to hyperbolic problems using characteristics. *Bol. Soc. Esp. Mat. Apl.*, 42:21–35.
- [Gander et Hairer, 2008] GANDER, M. et HAIRER, E. (2008). Nonlinear convergence analysis for the Parareal algorithm. *In* WIDLUND, O. B. et KEYES, D. E., éditeurs : *Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering*, volume 60, pages 45–56. Springer.
- [Gander et Petcu, 2008] GANDER, M. et PETCU, M. (2008). Analysis of a Krylov subspace enhanced Parareal algorithm for linear problems. *In ESAIM : Proceedings*, volume 25, pages 114–129. EDP Sciences.
- [Gander, 2015] GANDER, M. J. (2015). 50 years of time parallel time integration. In CARRARO, T., GEIGER, M., KÖRKEL, S. et RANNACHER, R., éditeurs : Multiple Shooting and Time Domain Decomposition Methods, pages 69–114. Springer.
- [Gander et Güttel, 2013] GANDER, M. J. et GÜTTEL, S. (2013). Paraexp : A parallel integrator for linear initial-value problems. *SIAM J. Sci. Comput.*, 35(2):C123–C142.
- [Gander *et al.*, 2017] GANDER, M. J., GÜTTEL, S. et PETCU, M. (2017). A nonlinear ParaExp algorithm. Rapport technique, Manchester University MIMS EPrint : 2017.17.
- [Gander et Vandewalle, 2007] GANDER, M. J. et VANDEWALLE, S. (2007). Analysis of the Parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578.
- [Gao et Zhang, 2016] GAO, Y. et ZHANG, P. (2016). A survey of homogeneous and heterogeneous system architectures in High Performance Computing. *In Smart Cloud (SmartCloud), IEEE International Conference on*, pages 170–175. IEEE.
- [Gil *et al.*, 2007] GIL, A., SEGURA, J. et TEMME, N. M. (2007). *Numerical methods for special functions*. SIAM.
- [Golub et Van Loan, 2012] GOLUB, G. H. et VAN LOAN, C. F. (2012). *Matrix computations*, volume 3. JHU Press.
- [Gray *et al.*, 2006] GRAY, R. M. *et al.* (2006). Toeplitz and circulant matrices : A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239.

- [Gropp, 1998] GROPP, W. (1998). *MPI : The complete reference. Vol. 2, The MPI-2 extensions.* MIT Press.
- [Günther *et al.*, 2017] GÜNTHER, S., GAUGER, N. R. et SCHODER, J. B. (2017). A non-intrusive parallelin-time adjoint solver with the XBraid library. *arXiv preprint arXiv :1705.00663*.
- [Hackbusch, 1984] НАСКВUSCH, W. (1984). Parabolic multi-grid methods. In Proc. of the sixth int'l. symposium on Computing methods in applied sciences and engineering, VI, pages 189–197. North-Holland Publishing Co.
- [Hernandez, 2014] HERNANDEZ, O. M. (2014). Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time. Thèse de doctorat, Université Pierre et Marie Curie-Paris VI.
- [Higham, 2002] HIGHAM, N. J. (2002). Accuracy and stability of numerical algorithms. SIAM.
- [Horton, 1991] HORTON, G. (1991). Time-parallel multigrid solution of the Navier-Stokes equations. In Applications of Supercomputers in Engineering II, pages 435–445. Springer.
- [Horton et Vandewalle, 1995] HORTON, G. et VANDEWALLE, S. (1995). A space-time multigrid method for parabolic partial differential equations. *SIAM J. Sci. Comput.*, 16(4):848–864.
- [Howse *et al.*, 2017] Howse, A., DE STERCK, H., MACLACHLAN, S., FALGOUT, R. et SCHRODER, J. (2017). Multigrid reduction in time with adaptive spatial coarsening for the linear advection equation. Rapport technique, Lawrence Livermore National Laboratory (LLNL), Livermore, CA.
- [Hoyas et Jiménez, 2006] HOYAS, S. et JIMÉNEZ, J. (2006). Scaling of the velocity fluctuations in turbulent channels up to $Re_{\tau} = 2003$. *Phys. Fluids*, 18(1):011702.
- [Hunt, 1998] HUNT, J. C. (1998). Lewis Fry Richardson and his contributions to mathematics, meteorology, and models of conflict. *Annual Review of Fluid Mechanics*, 30(1):xiii-xxxvi.
- [Ishihara *et al.*, 2009] Ізнінака, Т., Gотон, T. et Kaneda, Y. (2009). Study of high-Reynolds number isotropic turbulence by direct numerical simulation. *Annual Review of Fluid Mechanics*, 41:165–180.
- [Jamme, 1998] JAMME, S. (1998). Étude de l'interaction entre une turbulence homogène isotrope et une onde de choc. Thèse de doctorat, Institut National Polytechnique de Toulouse, France.
- [Jiang et Shu, 1996] JIANG, G.-S. et SHU, C.-W. (1996). Efficient implementation of weighted ENO schemes. J. Comp. Phys., 126(1):202-228.
- [Johnsen et al., 2010] JOHNSEN, E., LARSSON, J., BHAGATWALA, A. V., CABOT, W. H., MOIN, P., OLSON, B. J., RAWAT, P. S., SHANKAR, S. K., SJÖGREEN, B., YEE, H. et al. (2010). Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves. J. Comp. Phys., 229(4):1213–1237.
- [Kaneda *et al.*, 2003] КАNEDA, Y., ISHIHARA, T., YOKOKAWA, M., ITAKURA, K. et UNO, A. (2003). Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Phys. Fluids*, 15(2):L21–L24.
- [Keckler *et al.*, 2011] KECKLER, S. W., DALLY, W. J., KHAILANY, B., GARLAND, M. et GLASCO, D. (2011). GPUs and the future of parallel computing. *IEEE Micro*, 31(5):7–17.
- [Kennedy *et al.*, 2000] KENNEDY, C., CARPENTER, M. et LEWIS, R. (2000). Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Appl. Numer. Math.*, 35(3):177–219.
- [Ketcheson et bin Waheed, 2014] KETCHESON, D. et bin WAHEED, U. (2014). A comparison of highorder explicit Runge-Kutta, extrapolation, and deferred correction methods in serial and parallel. *Comm. App. Math. and Comp. Sci.*, 9(2):175–200.
- [Kim et Moin, 1985] KIM, J. et MOIN, P. (1985). Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, 59(2):308–323.

- [Kim *et al.*, 1987] KIM, J., MOIN, P. et MOSER, R. (1987). Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.*, 177:133–166.
- [Kolmogorov, 1941] Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *In Dokl. Akad. Nauk SSSR*, volume 30, pages 299–303.
- [Kooij *et al.*, 2016] KOOIJ, G., BOTCHEV, M. et GEURTS, B. (2016). A Krylov-based exponential time integrator of the incompressible Navier-Stokes equation. Talk during ECCOMAS Congress 2016.
- [Kooij *et al.*, 2017] KOOIJ, G., BOTCHEV, M. et GEURTS, B. (2017). A block Krylov subspace implementation of the time-parallel ParaExp method and its extension for nonlinear partial differential equations. *J. Comp. & Appl. Math.*, 316:229–246.
- [Krause et Ruprecht, 2014] KRAUSE, R. et RUPRECHT, D. (2014). Hybrid space-time parallel solution of Burgers' equation. *In Domain Decomposition Methods in Science and Engineering XXI*, pages 647–655. Springer.
- [Kreienbuehl *et al.*, 2015a] KREIENBUEHL, A., BENEDUSI, P., RUPRECHT, D. et KRAUSE, R. (2015a). Time parallel gravitational collapse simulation. *arXiv preprint arXiv :1509.01572*.
- [Kreienbuehl *et al.*, 2015b] КREIENBUEHL, A., NAEGEL, A., RUPRECHT, D., SPECK, R., WITTUM, G. et KRAUSE, R. (2015b). Numerical simulation of skin transport using Parareal. *Comput. Vis. Sci.*, 17(2):99–108.
- [Kreienbuehl et al., 2015c] KREIENBUEHL, A., NAEGEL, A., RUPRECHT, D., VOGEL, A., WITTUM, G. et KRAUSE, R. (2015c). Parareal convergence for 2d unsteady flow around a cylinder. arXiv preprint arXiv :1509.04252.
- [Larsson *et al.*, 2013] LARSSON, J., BERMEJO-MORENO, I. et LELE, S. K. (2013). Reynolds-and Machnumber effects in canonical shock-turbulence interaction. *J. Fluid Mech.*, 717:293.
- [Larsson *et al.*, 2007] LARSSON, J., LELE, S. et MOIN, P. (2007). Effect of numerical dissipation on the predicted spectra for compressible turbulence. *CTR Annual Research Briefs*, pages 47–57.
- [Larsson et Lele, 2009] LARSSON, J. et LELE, S. K. (2009). Direct numerical simulation of canonical shock/turbulence interaction. *Phys. Fluids*, 21(12):126101.
- [Layton et Minion, 2005] LAYTON, A. T. et MINION, M. L. (2005). Implications of the choice of quadrature nodes for Picard integral deferred corrections methods for ordinary differential equations. *BIT*, 45(2):341–373.
- [Lecouvez et al., 2016] LECOUVEZ, M., FALGOUT, R. D., WOODWARD, C. S. et TOP, P. (2016). A parallel multigrid reduction in time method for power systems. In Power and Energy Society General Meeting (PESGM), 2016, pages 1–5. IEEE.
- [Lee et Moser, 2015] LEE, M. et MOSER, R. (2015). Direct numerical simulation of turbulent channel flow up to $Re_{\tau} \simeq 5200$. J. Fluid Mech., 774:395–415.
- [Lekien et Marsden, 2005] LEKIEN, F. et MARSDEN, J. (2005). Tricubic interpolation in three dimensions. *Internat. J. Numer. Methods Engrg.*, 63(3):455–471.
- [Lelarasmee, 1982] LELARASMEE, E. (1982). *The waveform relaxation method for time domain analysis of large scale integrated circuits : Theory and applications.* Electronics Research Laboratory, College of Engineering, University of California.
- [Lele, 1992] LELE, S. (1992). Compact finite difference schemes with spectral-like resolution. J. Comp. Phys., 103(1):16-42.
- [Lions *et al.*, 2001] LIONS, J.-L., MADAY, Y. et TURINICI, G. (2001). A "Parareal" in time discretization of PDE's. *C. R. Math. Acad. Sci. Paris*, 332(7):661–668.
- [Lubich et Ostermann, 1987] LUBICH, C. et OSTERMANN, A. (1987). Multi-grid dynamic iteration for parabolic equations. *BIT Numerical Mathematics*, 27(2):216–234.

- [Lunet *et al.*, 2017a] LUNET, T., BODART, J., GRATTON, S. et VASSEUR, X. (2017a). Time-parallel simulation of the decay of homogeneous turbulence using Parareal with spatial coarsening. *Comput. Vis. Sci.*, (Accepted).
- [Lunet *et al.*, 2017b] LUNET, T., LAC, C., AUGUSTE, F., VISENTIN, F., MASSON, V. et ESCOBAR, J. (2017b). Combination of WENO and explicit Runge-Kutta methods for wind transport in the Meso-NH model. *Monthly Weather Review*, 145(9):3817–3838.
- [Maday et Turinici, 2003] MADAY, Y. et TURINICI, G. (2003). Parallel in time algorithms for quantum control : Parareal time discretization scheme. *International Journal of Quantum Chemistry*, 93(3): 223–228.
- [Maries *et al.*, 2012] MARIES, A., HAQUE, A., YILMAZ, S. L., NIK, M. B. et MARAI, G. E. (2012). Interactive exploration of stress tensors used in computational turbulent combustion. *In New Developments in the Visualization and Processing of Tensor Fields*, pages 137–156. Springer.
- [Mercerat *et al.*, 2009] MERCERAT, D., GUILLOT, L. et VILOTTE, J.-P. (2009). Application of the Parareal algorithm for acoustic wave propagation. *In AIP Conference Proceedings*, volume 1168, pages 1521–1524. AIP.
- [Merkel *et al.*, 2016] MERKEL, M., NIYONZIMA, I. et SCHÖPS, S. (2016). An application of ParaExp to electromagnetic wave problems. *In Electromagnetic Theory (EMTS), 2016 URSI International Symposium on*, pages 121–124. IEEE.
- [Merkel *et al.*, 2017] MERKEL, M., NIYONZIMA, I. et SCHÖPS, S. (2017). ParaExp using Leapfrog as integrator for high-frequency electromagnetic simulations. *arXiv preprint arXiv :1705.08019*.
- [Migliorino et Scalo, 2017] MIGLIORINO, M. T. et SCALO, C. (2017). Universal scaling of acoustic and thermoacoustic waves in compressible fluids. *arXiv preprint arXiv :1704.00084*.
- [Minion, 2003] MINION, M. (2003). Semi-implicit spectral deferred correction methods for ordinary differential equations. *Comm. Math. Sci.*, 1(3):471–500.
- [Minion, 2010] MINION, M. (2010). A hybrid Parareal spectral deferred corrections method. *Comm. App. Math. and Comp. Sci.*, 5(2):265–301.
- [Minion *et al.*, 2015] MINION, M. L., SPECK, R., BOLTEN, M., EMMETT, M. et RUPRECHT, D. (2015). Interweaving PFASST and parallel multigrid. *SIAM J. Sci. Comput.*, 37(5):S244–S263.
- [Modesti et Pirozzoli, 2016] MODESTI, D. et PIROZZOLI, S. (2016). Reynolds and Mach number effects in compressible turbulent channel flow. *Internat. J. Heat Transf. Fluid Flow*, 59:33–49.
- [Mohan *et al.*, 2017] Монан, Р., Fitzsimmons, N. et Moser, R. D. (2017). Scaling of Lyapunov exponents in homogeneous isotropic turbulence. *arXiv preprint arXiv :1707.05864*.
- [Moret et Novati, 2004] MORET, I. et NovATI, P. (2004). RD-rational approximations of the matrix exponential. *BIT Numer. Math.*, 44(3):595–615.
- [Moser *et al.*, 1999] MOSER, R., KIM, J. et MANSOUR, N. (1999). Direct numerical simulation of turbulent channel flow up to Re_{τ} = 590. *Phys. Fluids*, 11(4):943–945.
- [Motamed *et al.*, 2011] MOTAMED, M., MACDONALD, C. B. et RUUTH, S. J. (2011). On the linear stability of the fifth-order WENO discretization. *J. Sci. Comput.*, 47(2):127–149.
- [Murata *et al.*, 1991] Миката, S., SATOFUKA, N. et Kushiyama, T. (1991). Parabolic multi-grid method for incompressible viscous flows using a group explicit relaxation scheme. *Computers & Fluids*, 19(1):33–41.
- [Neumüller, 2013] NEUMÜLLER, M. (2013). *Space-Time Methods : Fast Solvers and Applications*. Monographic Series TU Graz : Computation in Engineering and Science.
- [Nievergelt, 1964] NIEVERGELT, J. (1964). Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733.
- [Olver, 2014] OLVER, P. J. (2014). Introduction to partial differential equations. Springer.

- [Ong et al., 2012] ONG, B., MELFI, A. et CHRISTLIEB, A. (2012). Parallel semi-implicit time integrators. arXiv preprint arXiv :1209.4297.
- [Orszag et Patterson Jr, 1972] ORSZAG, S. A. et PATTERSON JR, G. (1972). Numerical simulation of three-dimensional homogeneous isotropic turbulence. *Physical Review Letters*, 28(2):76.
- [Parsani *et al.*, 2013] PARSANI, M., КЕТСНЕSON, D. I. et DECONINCK, W. (2013). Optimized explicit Runge-Kutta schemes for the spectral difference method applied to wave propagation problems. *SIAM J. Sci. Comput.*, 35(2):A957–A986.
- [Pirozzoli et Bernardini, 2013] PIROZZOLI, S. et BERNARDINI, M. (2013). Probing high-Reynoldsnumber effects in numerical boundary layers. *Phys. Fluids*, 25(2):021704.
- [Polyanin et Nazaikinskii, 2015] POLYANIN, A. D. et NAZAIKINSKII, V. E. (2015). Handbook of linear partial differential equations for engineers and scientists. CRC press.
- [Pope, 2000] POPE, S. B. (2000). Turbulent Flows. Cambridge University Press.
- [Randles et Kaxiras, 2014] RANDLES, A. et KAXIRAS, E. (2014). Parallel in time approximation of the Lattice Boltzmann Method for laminar flows. *J. Comp. Phys.*, 270:577–586.
- [Rao et Yip, 2000] RAO, K. R. et YIP, P. C. (2000). *The transform and data compression handbook*, volume 1. CRC press.
- [Rao et Sandu, 2012] RAO, V. et SANDU, A. (2012). An adjoint based implementation of the Parareal algorithm. *Procedia Computer Science*, 9:1021–1029.
- [Reynolds-Barredo et al., 2012] REYNOLDS-BARREDO, J. M., NEWMAN, D. E., SANCHEZ, R., SAMADDAR, D., BERRY, L. A. et ELWASIF, W. R. (2012). Mechanisms for the convergence of time-parallelized, Parareal turbulent plasma simulations. J. Comp. Phys., 231(23):7851–7867.
- [Richardson, 1922] RICHARDSON, L. F. (1922). Weather prediction by numerical process. Cambridge University Press.
- [Ristorcelli et Blaisdell, 1997] RISTORCELLI, J. et BLAISDELL, G. A. (1997). Consistent initial conditions for the DNS of compressible turbulence. *Phys. Fluids*, 9(1):4–6.
- [Rogallo, 1981] ROGALLO, R. S. (1981). Numerical experiments in homogeneous turbulence. Rapport technique, NASA Aimes Research Center, NASA Technical Memorandum 81315.
- [Runge, 1895] RUNGE, C. (1895). Über die numerische Auflösung von Differentialgleichungen. *Ma-thematische Annalen*, 46(2):167–178.
- [Ruprecht, 2014] RUPRECHT, D. (2014). Convergence of Parareal with spatial coarsening. *PAMM*, 14(1):1031-1034.
- [Ruprecht, 2015] RUPRECHT, D. (2015). A shared memory implementation of pipelined Parareal. *arXiv preprint arXiv :1509.06935*.
- [Ruprecht, 2017] RUPRECHT, D. (2017). Wave propagation characteristics of Parareal. *arXiv preprint arXiv :1701.01359*.
- [Ruprecht et Krause, 2012] RUPRECHT, D. et KRAUSE, R. (2012). Explicit parallel-in-time integration of a linear acoustic-advection system. *Comput. & Fluids*, 59:72–83.
- [Ruprecht *et al.*, 2016] RUPRECHT, D., SPECK, R. et KRAUSE, R. (2016). Parareal for diffusion problems with space-and time-dependent coefficients. *In Domain Decomposition Methods in Science and Engineering XXII*, pages 371–378. Springer.
- [Russell et al., 2015] RUSSELL, F. P., WILKINSON, K. A., KELLY, P. H. et SKYLARIS, C.-K. (2015). Optimised three-dimensional Fourier interpolation : An analysis of techniques and application to a linear-scaling density functional theory code. Comput. Phys. Commun., 187:8–19.
- [Ryavec, 1982] RYAVEC, C. (1982). A non-linear interpolation formula. J. Math. Anal. Appl., 87(2):468–473.

- [Samaddar et al., 2010] SAMADDAR, D., NEWMAN, D. E. et SÁNCHEZ, R. (2010). Parallelization in time of numerical simulations of fully-developed plasma turbulence using the Parareal algorithm. J. Comp. Phys., 229(18):6558–6573.
- [Schiesser, 2012] SCHIESSER, W. E. (2012). The Numerical Method of Lines : Integration of Partial Differential Equations. Elsevier.
- [Schulze *et al.*, 2009] SCHULZE, J. C., SCHMID, P. J. et SESTERHENN, J. L. (2009). Exponential time integration using Krylov subspaces. *Internat. J. Numer. Methods Fluids*, 60(6):591–609.
- [Seny *et al.*, 2014] SENY, B., LAMBRECHTS, J., TOULORGE, T., LEGAT, V. et REMACLE, J.-F. (2014). An efficient parallel implementation of explicit multirate Runge-Kutta schemes for discontinuous Galerkin computations. *J. Comp. Phys.*, 256:135–160.
- [Speck, 2017] Speck, R. (2017). Parallelizing spectral deferred corrections across the method. *arXiv preprint arXiv* :1703.08079.
- [Speck *et al.*, 2014] SPECK, R., RUPRECHT, D., EMMETT, M., BOLTEN, M. et KRAUSE, R. (2014). A spacetime parallel solver for the three-dimensional heat equation. *In Parallel Computing : Accelerating Computational Science and Engineering (CSE). Advances in Parallel Computing*, pages 263–272.
- [Speck *et al.*, 2015] Speck, R., Ruprecht, D., Emmett, M., Minion, M., Bolten, M. et Krause, R. (2015). A multi-level spectral deferred correction method. *BIT*, 55(3):843–867.
- [Speck et al., 2012] SPECK, R., RUPRECHT, D., KRAUSE, R., EMMETT, M., MINION, M., WINKEL, M. et GIBBON, P. (2012). A massively space-time parallel N-body solver. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, page 92. IEEE Computer Society Press.
- [Staff et Rønquist, 2005] STAFF, G. A. et Rønquist, E. M. (2005). Stability of the Parareal algorithm. In KORNHUBER, R. et et AL, éditeurs : Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, volume 40, pages 449–456. Springer.
- [Steiner et al., 2015] STEINER, J., RUPRECHT, D., SPECK, R. et KRAUSE, R. (2015). Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number. In Abdulle, A., DE-PARIS, S., KRESSNER, D., NOBILE, F. et PICASSO, M., éditeurs : Numerical Mathematics and Advanced Applications ENUMATH 2013, volume 103, pages 195–202. Springer.
- [Trettel et Larsson, 2016] TRETTEL, A. et LARSSON, J. (2016). Mean velocity scaling for compressible wall turbulence with heat transfer. *Phys. Fluids*, 28(2):026102.
- [Trindade et Pereira, 2004] TRINDADE, J. et PEREIRA, J. (2004). Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow. *Internat. J. Numer. Methods Fluids*, 45(10):1123–1136.
- [Trindade et Pereira, 2006] TRINDADE, J. et PEREIRA, J. (2006). Parallel-in-time simulation of twodimensional, unsteady, incompressible laminar flows. *Numerical Heat Transfer, Part B : Fundamentals*, 50(1):25–40.
- [Trottenberg *et al.*, 2001] TROTTENBERG, U., OOSTERLEE, C. W. et SCHÜLLER, A. (2001). *Multigrid*. Academic Press Inc., New-York.
- [Tzu, 2016] Tzu, S. (2016). *L'art de la guerre*. Les éditions Pulsio.
- [Vandewalle et Horton, 1994] VANDEWALLE, S. et HORTON, G. (1994). Multicomputer-multigrid solution of parabolic partial differential equations. In Multigrid Methods IV : Proceedings of the Fourth European Multigrid Conference, P. Hemker and P. Wesseling, eds, volume 116, pages 97– 109. Springer.
- [Vandewalle et Van de Velde, 1994] VANDEWALLE, S. et Van de VELDE, E. (1994). Space-time concurrent multigrid waveform relaxation. *Annals of Numerical Mathematics*, 1(1-4):335–346.
- [Vastano et Moser, 1991] VASTANO, J. A. et MOSER, R. D. (1991). Short-time Lyapunov exponent analysis and the transition to chaos in Taylor-Couette flow. *J. Fluid Mech.*, 233:83–118.

- [Vincent et Meneguzzi, 1991] VINCENT, A. et MENEGUZZI, M. (1991). The spatial structure and statistical properties of homogeneous turbulence. *J. Fluid Mech.*, 225:1–20.
- [Wang *et al.*, 2013] WANG, Q., GOMEZ, S. A., BLONIGAN, P. J., GREGORY, A. L. et QIAN, E. Y. (2013). Towards scalable parallel-in-time turbulent flow simulations. *Phys. Fluids*, 25(11):110818.
- [Wang et Spiteri, 2007] WANG, R. et SPITERI, R. J. (2007). Linear instability of the fifth-order WENO method. *SIAM J. Numer. Anal.*, 45(5):1871–1901.
- [Wang, 2007] WANG, Z. (2007). High-order methods for the Euler and Navier-Stokes equations on unstructured grids. *Progress in Aerospace Sciences*, 43(1):1–41.
- [Wienands et Joppich, 2004] WIENANDS, R. et JOPPICH, W. (2004). *Practical Fourier analysis for multigrid methods*. CRC press.
- [Williams *et al.*, 2009] WILLIAMS, S., WATERMAN, A. et PATTERSON, D. (2009). Roofline : an insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76.
- [Wolf *et al.*, 1985] WOLF, A., SWIFT, J. B., SWINNEY, H. L. et VASTANO, J. A. (1985). Determining Lyapunov exponents from a time series. *Physica D : Nonlinear Phenomena*, 16(3):285–317.
- [Woo et Lee, 2008] Woo, D. H. et LEE, H.-H. S. (2008). Extending Amdahl's law for energy-efficient computing in the many-core era. *Computer*, 41(12).
- [Wu, 2016] WU, S.-L. (2016). Convergence analysis of the Parareal-Euler algorithm for systems of ODEs with complex eigenvalues. *J. Sci. Comput.*, 67(2):644–668.

AUTEUR : Thibaut LUNET

TITRE : Stratégies de parallélisation espace-temps pour la simulation numérique des écoulements turbulents

DIRECTEUR DE THÈSE : Serge GRATTON

LIEU ET DATE DE SOUTENANCE : ISAE-SUPAERO (Toulouse), le 9 janvier 2018

Cette thèse étudie l'application de méthodes de parallélisation en temps pour la simulation numérique directe des écoulements turbulents. Après une étude préliminaire, on choisit de se focaliser sur l'algorithme Parareal avec grossissement spatial. Le comportement de l'algorithme est étudié en premier lieu sur l'équation d'advection, comme simplification des équations de Navier-Stokes, par une analyse de Fourier et une série d'expériences numériques, afin d'en cerner les mécanismes et paramètres dimensionnants.

L'algorithme est ensuite étudié dans un contexte HPC, à l'aide du code de simulation massivement parallèle Hybrid. Deux situations d'écoulements turbulents tridimensionnels sont à l'étude : la décroissance d'une turbulence homogène isotrope et l'écoulement de canal turbulent. Ce travail propose une première mesure de l'efficacité de la parallélisation combinée espace-temps, ainsi qu'une évaluation précise de la capacité de l'algorithme à représenter les propriétés physiques de la turbulence.

MOTS-CLES : Calcul Haute Performance, Mécanique des Fluides Numérique, Parallélisation en Temps, Écoulement Turbulent, Parareal avec Grossissement Spatial.

DISCIPLINE ADMINISTRATIVE : Domaine Mathématiques, Mathématiques appliquées

INTITULE ET ADRESSE DE L'U.F.R. OU DU LABORATOIRE :

- ISAE-SUPAERO, DAEP,
 10 avenue Edouard Belin, BP 54032, F-31055 Toulouse Cedex 4, France,
- CERFACS, Parallel Algorithms project, 42 avenue Gaspard Coriolis, F-31057 Toulouse Cedex 1, France.

TITLE : Space-time parallel strategies for the numerical simulation of turbulent flows

This thesis aims at studying the application of time-parallel integration methods for the Direct Numerical Simulation of turbulent flows. After a preliminary study, we choose to focus on the Parareal algorithm with spatial coarsening. The behavior of the algorithm is first studied on the advection equation, as a simplified model for the Navier-Stokes equations, using a Fourier analysis and numerical experiments, to understand its mechanisms and identify the relevant parameters.

The algorithm is then studied in a HPC context, using the massively parallel CFD simulation code Hybrid. Two tri-dimensional turbulent flow problems are investigated : the decay of an Homogeneous Isotropic Turbulence and the Turbulent Channel Flow. This work offers a first evaluation of combined space-time parallel efficiency, and analyse the algorithm's abilities to correctly reproduce the physical properties of turbulence.

KEY-WORDS : High Performance Computing, Computational Fluid Dynamics, Time Parallelization, Turbulent Flow, Parareal with Spatial Coarsening.