



HAL
open science

Méthode explicite à pas de temps local pour la simulation des écoulements turbulents instationnaires

Guillaume Jeanmasson

► **To cite this version:**

Guillaume Jeanmasson. Méthode explicite à pas de temps local pour la simulation des écoulements turbulents instationnaires. Systèmes dynamiques [math.DS]. Université de Bordeaux, 2019. Français. NNT : 2019BORD0458 . tel-02872204

HAL Id: tel-02872204

<https://theses.hal.science/tel-02872204>

Submitted on 17 Jun 2020

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

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

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE
DOCTEUR DE
L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE
SPÉCIALITÉ : MATHÉMATIQUES APPLIQUÉES

Par Guillaume JEANMASSON

**METHODE EXPLICITE A PAS DE TEMPS LOCAL POUR LA
SIMULATION DES ECOULEMENTS TURBULENTS
INSTATIONNAIRES**

Sous la direction de : Luc MIEUSSENS

Co-directeur : Ivan MARY

Soutenue le 19 décembre 2019

Membres du jury :

M. CREUSE Emmanuel	Pr., Université Polytechnique Hauts-de-France	Examineur
M. REMACLE Jean-François	Pr., Université Catholique de Louvain	Rapporteur
M. LAMBALLAIS Éric	Pr., Université de Poitiers	Président du jury
Mme. BEAUGENDRE Héloïse	Maître de Conférences, Université de Bordeaux	Examineur
M. MIEUSSENS Luc	Pr., Université de Bordeaux	Examineur
M. MARY Ivan	Ingénieur de recherche, ONERA	Examineur
M. BRENNER Pierre	Ingénieur, ArianeGroup	Invité

Titre : METHODE EXPLICITE A PAS DE TEMPS LOCAL POUR LA SIMULATION DES ECOULEMENTS TURBULENTS INSTATIONNAIRES

Résumé : La simulation instationnaire d'écoulements turbulents (LES : Large Eddy Simulation, par exemple) reste à l'heure actuelle coûteuse en terme de temps de calcul. Un travail sur les méthodes numériques d'intégration temporelle des équations de la mécanique des fluides peut rendre de type de simulation plus accessible. Les méthodes d'intégration temporelle explicites présentent des propriétés intéressantes telles qu'une bonne précision et une bonne compatibilité avec les techniques HPC (parallélisation, vectorisation...). Néanmoins, le pas de temps de ces méthodes est fortement limité : il est choisi de manière à respecter la contrainte CFL la plus restrictive sur le maillage. Ceci rend ces méthodes généralement très coûteuses. Dans le cas des méthodes explicites à pas de temps local, le pas de temps varie sur le maillage en s'adaptant à différentes contraintes CFL locales. Le pas de temps est plus optimal sur l'ensemble du maillage, ce qui réduit les temps de simulation par rapport à une méthode explicite à pas de temps global (uniforme). Les schémas à pas de temps local de la littérature sont essentiellement utilisés sur des cas-test académiques, et on recense peu d'applications de ces schémas en CFD. L'objectif de cette thèse est de montrer que ce type de schéma peut être utilisé de manière efficace pour la LES. Pour atteindre cet objectif, deux nouveaux schémas à pas de temps local ont d'abord été mis en place. Deux simulations LES ont ensuite été réalisées à l'aide de notre schéma à pas de temps local le plus performant. Ces deux simulations ont démontré la bonne précision et l'efficacité de notre schéma à pas de temps local pour la simulation d'écoulements turbulents.

Mots clés : Méthodes de Runge-Kutta explicites, Pas de temps local, Simulations instationnaires, Ecoulements turbulents, LES, Calcul parallèle

Title : Explicit local time stepping method for unsteady turbulent flows

Abstract : Unsteady simulations of turbulent flows (LES: Large Eddy Simulation, for instance) still present an important computational cost. Improvement of numerical time integration methods could reduce the computational effort needed by these simulations. Explicit time integration methods present attractive properties such that accuracy and good compatibility with HPC (parallelization, vectorization...). Nevertheless, the time step used by these methods is strongly restricted: it is chosen to respect the most severe CFL condition over the mesh. This makes explicit time integration methods very expensive in terms of computational cost. Explicit local time stepping methods use a non uniform time step to satisfy several local CFL conditions over the mesh. The time step is more optimal over the mesh, which leads to a reduction of computational cost with respect to an explicit time integration method with a uniform time step. Most of the local time stepping schemes proposed in the literature are applied to academic test cases, and very few applications are performed in CFD. This thesis's goal is to show the ability of explicit local time stepping schemes to carry out efficient and accurate simulations of turbulent flows. To reach this goal, two new explicit local time stepping schemes have been developed. Then, two LES have been carried out with our most efficient local time stepping scheme. Both cases have demonstrated the accuracy and the efficiency of our local time stepping scheme for the computations of turbulent flows.

Keywords : Explicit Runge-Kutta methods, Local time stepping, Unsteady simulations, Turbulent flows, LES, Parallel computing

IMB – Institut de Mathématiques de Bordeaux UMR 5251
Université de Bordeaux 351, cours de la Libération - F 33 405 TALENCE

Table des matières

1	Méthodes d'intégration temporelle pour la simulation numérique en mécanique des fluides	12
1.1	Discrétisation spatiale des équations : la méthode volumes finis	13
1.1.1	La méthode volumes finis (VF) dans le cas multidimensionnel	13
1.1.2	La méthode volumes finis dans le cas unidimensionnel	15
1.1.3	Principe de conservation	18
1.2	Exigences concernant les méthodes d'intégration temporelle	19
1.2.1	Caractéristiques des écoulements turbulents	20
1.2.2	Trois critères essentiels pour la simulation des écoulements turbulents	21
1.3	Les méthodes d'intégration temporelle explicites	23
1.3.1	Présentation générale des méthodes explicites	23
1.3.2	Méthodes explicites à un pas : les méthodes de Runge-Kutta explicites	23
1.3.3	Les méthodes explicites multipas : les méthodes d'Adams-Bashforth explicites	29
1.3.4	Stabilité concernant l'intégration d'une équation différentielle	30
1.3.5	Stabilité concernant l'intégration d'une EDP discrétisée en espace	31
1.3.6	Précision des méthodes explicites	33
1.4	Les méthodes d'intégration temporelle implicites	34
1.4.1	Présentation générale des méthodes implicites	34
1.4.2	Les méthodes implicites à un pas	34
1.4.3	Une méthode implicite multipas : la méthode de Gear	37
1.4.4	Stabilité des méthodes implicites	37
1.4.5	Précision des méthodes implicites	39
1.4.6	Algorithme de Newton pour la résolution du système non-linéaire	40
1.4.7	Algorithme de Newton adaptatif	42
1.5	Bilan comparatif des méthodes implicites et explicites	43
1.6	Les méthodes IMplicite/EXplicite (IMEX)	44
1.7	Les méthodes explicites à pas de temps variable en espace	46
1.7.1	Méthodes explicites à pas de temps local asynchrones	47
1.7.2	Méthodes explicites à pas de temps local	49
2	Schémas à pas de temps local de Constantinescu et Sandu	55
2.1	Méthodes de Runge-Kutta Partitionnées	55
2.1.1	Lien entre schémas à pas de temps local et méthodes RKP	56

2.1.2	Conditions d'ordre pour les méthodes RKP	60
2.2	Schémas à pas de temps local de Constantinescu et Sandu	61
2.2.1	Construction des méthodes RK_S et RK_F	61
2.2.2	Ordre temporel des schémas de Constantinescu et Sandu	64
2.2.3	Conservativité des schémas de Constantinescu et Sandu	67
2.3	Schéma de Constantinescu et Sandu basé sur la méthode RK2 (CSRK2)	68
2.3.1	Discussions autour de la méthode RK_S	69
2.3.2	Estimation du coût algorithmique du schéma CSRK2	74
2.3.3	Généralisation à N pas de temps du schéma CSRK2	75
2.3.4	Discussion sur le choix du rapport entre les pas de temps	76
2.4	Schéma de Constantinescu et Sandu basé sur une méthode RK3 (CSRK3)	77
2.4.1	Construction des méthodes RK_S et RK_F	77
2.4.2	Discussions autour de la méthode RK_S	78
2.4.3	Estimation du coût algorithmique du schéma CSRK3	78
2.4.4	Ordre temporel et conservativité du schéma CSRK3	79
2.4.5	Généralisation à N pas de temps du schéma CSRK3	79
3	Nouveaux schémas à pas de temps local	80
3.1	Nouveau schéma à pas de temps local d'ordre deux (NRK2)	80
3.1.1	Construction des méthodes RK_S et RK_F	80
3.1.2	Ordre du schéma NRK2	82
3.1.3	Correction de flux pour assurer la conservativité du schéma NRK2	83
3.1.4	Discussion autour de la méthode RK_S	86
3.1.5	Estimation du coût algorithmique du schéma NRK2	89
3.1.6	Différences avec le schémas à pas de temps local de Tang et Warnecke	90
3.1.7	Généralisation du schéma NRK2 à N pas de temps	91
3.2	Nouveau schéma à pas de temps local d'ordre trois (NRK3)	92
3.2.1	Construction des méthodes RK_S et RK_F	92
3.2.2	Détermination des coefficients α^S et α^F pour assurer l'ordre trois	95
3.2.3	Correction de flux pour assurer la conservativité du schéma NRK3	103
3.2.4	Discussions autour des méthodes RK_S et RK_F	104
3.2.5	Estimation du coût algorithmique du schéma NRK3	108
3.2.6	Différences avec les schémas à pas de temps local d'ordre trois de la littérature	109
3.2.7	Généralisation du schéma NRK3 à N pas de temps	111
3.3	Implémentation du schéma NRK3	111
3.3.1	Notion de raccords entre zones	112
3.3.2	Notion d'évènement	113
3.3.3	Paramètres q^* et θ_i	116
3.3.4	Pilotage des évènements	117
3.3.5	Algorithme du schéma NRK3 à N pas de temps	120

4	Validation des schémas proposés sur des cas tests académiques	122
4.1	Équations de la mécanique des fluides	122
4.2	Rappels sur la discrétisation spatiale Volumes Finis (VF)	124
4.3	Flux convectifs	126
4.3.1	Schéma AUSM+(P) senseur	126
4.3.2	Schéma de Roe	129
4.4	Flux visqueux	131
4.5	Advection d'un tourbillon 2D	133
4.5.1	Présentation du cas-test	133
4.5.2	Présentation du domaine de calcul	134
4.5.3	Étude de la stabilité, du gain en temps de calcul et de la conservativité des schémas à pas de temps local	135
4.5.4	Résultats des simulations	139
4.5.5	Vérification de l'ordre temporel	141
4.5.6	Vérification de l'ordre spatial	142
4.6	Tube à choc de Sod	143
4.6.1	Présentation du cas-test et du domaine de calcul	143
4.6.2	Étude numérique de la stabilité et du speedup des schémas à pas de temps local	144
4.6.3	Résultats des simulations	146
4.6.4	Vérification de l'ordre temporel	148
4.7	Tourbillon de Taylor-Green 3D	148
4.7.1	Présentation du cas-test et du domaine de calcul	149
4.7.2	Étude numérique de la stabilité, du speedup et de l'erreur sur la conservativité	151
4.7.3	Résultats des simulations	153
4.7.4	Vérification de l'ordre temporel	153
4.8	Bilan des cas-test académiques	154
5	Démonstration sur des configurations de type industriel	156
5.1	LES d'un écoulement transitionnel autour du profil SD7003	156
5.1.1	Description du cas	156
5.1.2	Paramètres de la simulation et maillage	158
5.1.3	Découpage du maillage pour le schéma NRK3	160
5.1.4	Schémas d'intégration temporelle utilisés sur ce cas	166
5.1.5	Présentation des résultats	167
5.1.6	Comparaison des temps de calcul et bilan de ce cas-test	172
5.2	Parallélisation OpenMP/MPI du schéma NRK3	173
5.2.1	Parallélisation MPI du schéma NRK3	176
5.2.2	Parallélisation OpenMP du schéma NRK3	181
5.3	LES d'un écoulement 3D autour d'un cylindre par une méthode de frontières de immergées	186
5.3.1	Quelques généralités sur les méthodes de frontières immergées	187
5.3.2	Présentation du cas et maillage	188

5.3.3	Schémas d'intégration temporelle utilisés sur ce cas	190
5.3.4	Résultats des simulations	191
5.3.5	Comparaison des temps de calcul et bilan de ce cas-test	195
A	Conditions d'ordre trois pour une méthode de Runge-Kutta partitionnée (RKP)	200
A.1	Développement de Taylor de u_S^{n+1}	200
A.2	Développement de Taylor de la solution exacte $u_S(t + \Delta t)$	204
A.3	Conditions d'ordre trois	204
B	Tableau de Butcher représentant deux applications successives d'une méthode de pas de temps $\Delta t/2$	206
C	Les différentes étapes du schéma NRK3 écrit sous forme "Low storage"	210

Remerciements

A l'heure où ce travail de thèse prend fin, il est temps pour moi de remercier toutes les personnes qui m'ont accompagné de près ou de loin dans cette aventure.

Je voudrais en premier lieu remercier Luc Mieussens, mon directeur de thèse, pour son soutien, ses compétences et ses remarques toujours constructives sur mes travaux. Nos discussions ont toujours permis de définir une ligne claire et précise pour l'avancement de cette thèse. Je remercie tout particulièrement Ivan Mary, mon co-directeur de thèse, sans qui ce travail n'aurait jamais vu le jour. Sa disponibilité, sa bonne humeur et ses compétences dans de nombreux domaines de la CFD ont largement contribué à la réalisation de ce travail de thèse. Je lui dois beaucoup, et je l'en remercie.

Je remercie également tous les membres de l'unité DEFI pour ces trois années et demi passées dans l'équipe : T-H, Jean-Marie, Didier, Jacques, Xavier, Thomas, Stéphanie, Christelle, Christophe, Sam. Leur bonne humeur et leur sympathie ont rendu les moments de pausé café et déjeuner particulièrement agréables. J'ai une pensée particulière pour Thomas et Jacques, fervents supporter du PSG, qui n'ont jamais perdu le sourire malgré les nombreux "chambrages" suite aux éliminations du PSG en ligue des champions. Ne perdez pas espoir, la ligue des champions est peut-être pour cette année (ou pas)!

Je remercie Nicolas Alferez, avec qui j'ai partagé mon bureau pendant deux ans. Ses compétences et sa joie de vivre ont rendu ces deux années de collaboration très instructives et agréables.

Merci aux membres du jury qui ont bien voulu évaluer mes travaux. Je remercie tout particulièrement les deux rapporteurs, Monsieur Jean-François Remacle et Monsieur Eric Lamballais.

Je remercie tous les doctorants et stagiaires rencontrés tout au long de cette thèse pour avoir créé une ambiance chaleureuse et détendue.

Enfin, merci à ma famille pour son soutien et sa patience.

Introduction

La simulation numérique instationnaire s'impose progressivement comme un moyen efficace pour la compréhension des phénomènes complexes et fortement dépendants du temps. Des simulations instationnaires sont par exemple utilisées pour tenter de prédire les changements climatiques futurs [1], pour comprendre la propagation des tsunamis [2] ou bien le fonctionnement du champ magnétique terrestre [3]. Dans le domaine de l'aéronautique, ces simulations peuvent par exemple être utilisées pour la prédiction du phénomène de décrochage, responsable de la perte des performances aérodynamiques d'un avion [4], ou bien pour l'étude de la propagation acoustique du bruit généré par les aéronefs [5]. La compréhension des phénomènes aérodynamiques est d'une importance capitale pour améliorer le confort, les performances, la sécurité des engins aéronautiques. En particulier, les simulations instationnaires de type LES (Large Eddy Simulation) apparaissent comme un outil de plus en plus fiable pour l'étude et la compréhension des écoulements turbulents. Cette technique est un compromis entre la simulation de toutes les échelles spatio-temporelles de la turbulence (Direct Numerical Simulation, DNS) et la modélisation physique des effets de la turbulence (modèles RANS) : il s'agit de calculer directement les plus grosses structures turbulentes (les plus énergétiques) et de modéliser les plus petites. Les modèles RANS, majoritairement utilisés dans l'industrie, permettent de réaliser des simulations rapides mais leur fonctionnement est assez aléatoire : généralement, ils donnent de très bons résultats mais principalement sur les cas pour lesquels ils ont été "calibrés". Quant à la DNS, elle requiert une puissance de calcul qui dépasse les capacités des supercalculateurs actuels ; aussi cette technique est limitée à des simulations académiques. L'utilisation de simulations instationnaires de type LES (et DNS) peut permettre une meilleure description de la turbulence, et peut également servir à enrichir des bases de données sur les écoulements turbulents en vue d'améliorer les modèles RANS.

Les simulations instationnaires d'écoulements turbulents (de type LES par exemple) restent tout de même assez coûteuses. En effet, le maillage doit être suffisamment fin pour capter les structures turbulentes les plus énergétiques. De plus, les phénomènes turbulents étant fortement aléatoires, un traitement statistique (moyenne temporelle) de l'écoulement est nécessaire. Le calcul doit être réalisé sur un temps physique assez long pour obtenir des analyses statistiques de qualité. Il est donc important de pouvoir réduire au maximum le coût de calcul des simulations instationnaires. Deux approches sont envisageables pour atteindre cet objectif. La première approche consiste à travailler sur la modélisation physique des phénomènes turbulents. En effet, une modélisation physique fiable et précise permet de réduire le nombre cellules dans le maillage, et donc de diminuer les temps de simu-

lation. En LES, l'effort de modélisation porte sur l'amélioration des modèles sous-maille [6]. Une meilleure modélisation de la turbulence peut également être obtenue avec des méthodes hybrides RANS/LES [7]. La seconde approche consiste à améliorer les méthodes numériques de discrétisation spatiale et/ou d'intégration temporelle. Une idée intéressante serait d'améliorer la précision des méthodes de discrétisation spatiale pour pouvoir utiliser moins de points sur le maillage. C'est par exemple le cas des méthodes de type Discontinuous Galerkin (DG), où la solution sur chaque cellule du maillage peut être construite comme un polynôme d'ordre élevé (supérieur à deux). Ces méthodes donnent de très bons résultats et connaissent un développement important ces dernières années en mécanique des fluides [8, 9]. Néanmoins, la supériorité des méthodes DG par rapport aux méthodes Volumes Finis, d'ordre moins élevé et nécessitant un nombre de points plus important sur le maillage, n'a pas encore été clairement démontrée [10]. En revanche, les méthodes numériques d'intégration temporelle semblent représenter une source de gains potentiels plus intéressante. Au cours des dernières décennies, de nouvelles méthodes d'intégration temporelle ont d'ailleurs permis d'améliorer l'efficacité des simulations instationnaires (méthodes IMplicite/EXplicite [11, 12, 13, 14, 15], méthodes explicites à pas de temps variable en espace [16, 17, 18, 19]). Cette thèse s'oriente donc vers l'amélioration des méthodes numériques d'intégration temporelle en vue de réaliser des simulations instationnaires fiables d'écoulements turbulents.

On distingue généralement deux types de méthodes d'intégration temporelle : les méthodes explicites et implicites. Les méthodes explicites sont considérées comme très précises et sont dotées d'une bonne compatibilité avec le Calcul Haute Performance (HPC) : calcul parallèle, optimisation des accès mémoire, vectorisation... Par contre, elles sont souvent coûteuses en temps de calcul. En effet, le pas de temps des méthodes explicites est fortement limité par une contrainte de stabilité numérique, appelée condition *CFL*. Cette condition *CFL* est définie en chaque cellule et est proportionnelle au ratio entre la vitesse locale de l'écoulement et la taille de la cellule. Or, ce ratio est bien souvent non uniforme sur le maillage. D'une part, parce que la vitesse de l'écoulement varie dans l'espace. D'autre part, parce que la grande diversité des échelles spatiales de l'écoulement nécessite de faire varier la taille des cellules sur le maillage. Le pas de temps des méthodes explicites est choisi de manière à respecter la condition *CFL* la plus restrictive. Ce choix est d'autant plus pénalisant que le nombre de cellules présentant une condition *CFL* très restrictive est faible devant le reste des cellules, dont la condition *CFL* plus lâche. Ainsi, à cause d'une minorité de cellules, le pas de temps est restreint et est non optimal sur une grande partie du maillage. Cette restriction sur le pas de temps impose d'effectuer un grand nombre d'itérations temporelles pour atteindre un temps physique donné, et cela engendre des temps de calcul élevés.

Les méthodes implicites s'affranchissent de cette contrainte de stabilité numérique sur le pas de temps, ce qui les rend généralement moins coûteuses en temps de calcul. Par contre, ces méthodes sont généralement moins précises que les méthodes explicites et également moins compatibles avec les techniques de calcul parallèle.

Au cours des dernières décennies, certains auteurs ont proposé de nouvelles méthodes

d'intégration temporelle pour remédier aux inconvénients des méthodes explicites et implicites. On peut par exemple citer les méthodes IMplicit/EXplicit (IMEX), évoquées précédemment. Ces méthodes se composent d'un schéma implicite pour intégrer les cellules présentant la condition CFL la plus restrictive et d'un schéma explicite pour l'intégration des cellules présentant une condition CFL moins restrictive. Les méthodes IMEX donnent de bons résultats pour la simulation instationnaire [11, 12, 13, 14, 15], mais la partie implicite de ces méthodes peut potentiellement limiter leur compatibilité avec les techniques HPC. Aussi, cette thèse s'oriente vers des méthodes purement explicites : les méthodes explicites à pas de temps local. Ces méthodes ont été inventées pour gommer le principal défaut des méthodes explicites : les temps de calcul élevés. Les méthodes explicites à pas de temps local exploitent le fait que la condition CFL est bien souvent non-uniforme sur le domaine de calcul. Le pas de temps varie sur le domaine de calcul pour respecter différentes conditions CFL locales : de "petits" pas de temps sont utilisés sur les cellules présentant une condition CFL très restrictive, et des pas de temps "plus grands" sont appliqués sur les cellules dont la condition CFL est moins restrictive. Ainsi, le pas de temps est plus optimal sur l'ensemble du maillage, ce qui réduit le nombre global d'opérations à effectuer, et diminue le temps de calcul par rapport à une méthode explicite à pas de temps global (uniforme). Les méthodes explicites à pas de temps local peuvent donc permettre de corriger la "lenteur" des méthodes explicites à pas de temps global tout en conservant leurs avantages (précision, bonne compatibilité avec les techniques HPC). Ces méthodes pourraient donc apparaître comme "presque idéales" pour la simulation instationnaire, d'où le choix de les étudier dans cette thèse.

L'étude de la bibliographie sur les schémas à pas de temps local a révélé deux inconvénients pour ce type de schémas : leur ordre de précision temporel est souvent limité, et ils sont souvent non conservatifs. De plus, les schémas à pas de temps local de la littérature sont souvent appliqués à la résolution de problèmes de type académiques (équation de Burgers, équations d'Euler 1D ou 2D...) [20, 21, 16, 17, 22, 23]. Quelques simulations "de plus grande envergure" ont été réalisées, mais la littérature est peu fournie. On peut notamment citer Sény *et al* [24], qui ont réalisé une simulation de l'hydrodynamique au niveau de la grande barrière de corail à l'aide de schémas à pas de temps local. On peut également citer Itam [25] qui a réalisé une simulation RANS/LES à l'aide d'un schéma explicite à deux pas de temps. Il semble donc nécessaire de montrer que ce type de schéma peut être utilisé pour réaliser des simulations à caractère industriel : 3D, instationnaires de type LES. Au vu de cette étude bibliographique, un objectif a été défini pour cette thèse : réaliser des simulations 3D, instationnaires sur des configurations de type industriel avec un schéma à pas de temps local. Pour mener à bien cet objectif, deux nouveaux schémas à pas de temps local d'ordre deux et trois ont été proposés.

Ce manuscrit a donc pour but de présenter les travaux menés pour construire, implémenter et évaluer ces nouveaux schémas. Ce manuscrit se décompose en cinq chapitres.

Dans le chapitre 1, une présentation des méthodes d'intégration temporelle implicites et explicites est d'abord réalisée. Ensuite, différentes méthodes dérivant des méthodes explicites et implicites sont décrites. Bien entendu, parmi ces méthodes, une attention particulière est portée à la présentation des méthodes explicites à pas de temps local.

Dans le chapitre 2, un type de schémas à pas de temps local de la littérature est présenté en particulier : il s'agit des schémas de Constantinescu et Sandu [16]. Ces schémas présentent des propriétés très intéressantes (ordre deux, conservativité) ; ils ont donc été logiquement choisis comme points de comparaison pour les deux nouveaux schémas à pas de temps local proposés dans cette thèse.

Dans le chapitre 3, les deux nouveaux schémas à pas de temps local sont présentés. Dans un premier temps, un schéma à pas de temps local précis à l'ordre deux a été mis en œuvre. Dans un second temps, un schéma précis à l'ordre trois (mieux adapté aux niveaux de précisions requis pour la simulation instationnaire) a été mis en place. Dans la section 3.3 de ce chapitre, l'implémentation du schéma d'ordre trois est détaillée.

Dans le chapitre 4, les nouveaux schémas à pas de temps local sont confrontés aux schémas de Constantinescu et Sandu [16] sur trois cas-test académiques. A l'issue de ces cas-test, le nouveau schéma à pas de temps local d'ordre trois a été choisi pour réaliser deux simulations instationnaires (LES) :

- un écoulement transitionnel autour du profil SD7003,
- un écoulement autour d'un cylindre (en conjonction avec une méthode de frontières immergées)

Dans le chapitre 5, les résultats obtenus sur ces deux cas avec notre schéma à pas de temps local sont soigneusement comparés à des résultats issus de schémas explicites et implicites. On montre également dans ce chapitre des résultats de parallélisation MPI et OpenMP obtenus avec le schéma à pas de temps local d'ordre trois.

Communication/Publication

Ce travail de thèse a fait l'objet d'une communication lors d'un congrès :

G.Jeanmasson, I. Mary, L. Mieussens. Explicit local time stepping scheme for the unsteady simulation of turbulent flows. *Tenth International Conference on Computational Fluid Dynamics (ICCFD10)*. Barcelona, Spain, July 9-13, 2018.

Ce travail de thèse a également fait l'objet d'une publication dans une revue :

G.Jeanmasson, I. Mary, L. Mieussens. On some explicit local time stepping finite volume schemes for CFD *Journal of Computational Physics*, 397 :108818,2019

Chapitre 1

Méthodes d'intégration temporelle pour la simulation numérique en mécanique des fluides

Les écoulements de fluide sont modélisés par des équations aux dérivées partielles (équations de Navier-Stokes ou d'Euler). Comme des solutions analytiques à ces équations sont bien souvent impossibles à obtenir, les scientifiques ont recours à des méthodes numériques. Celles-ci reposent sur la transformation du problème continu en un problème discret et permettent d'obtenir une solution approchée des équations aux dérivées partielles (EDP).

La transformation du problème continu en problème discret requiert dans un premier temps de discrétiser le domaine spatial sur lequel les équations sont définies. L'espace est alors fragmenté en un nombre fini d'éléments (également appelés cellules ou mailles), sur lesquels sont définies les inconnues spatiales. Il s'agit de l'étape de maillage du domaine spatial. A l'issue de cette étape de maillage, une méthode de discrétisation spatiale peut être appliquée aux EDP. On distingue trois grandes méthodes de discrétisation spatiale : la méthode volumes finis (VF), la méthode différences finies (DF) et la méthode éléments finis (EF). Dans ce mémoire, on s'intéresse particulièrement à la résolution numérique des équations de la mécanique des fluides, et la méthode VF est la méthode de discrétisation spatiale la plus populaire dans ce domaine de la physique. En effet, les EDP régissant la mécanique des fluides sont des lois de conservation, elles traduisent de manière infinitésimal le principe de conservation suivant : la variation temporelle d'une grandeur contenue dans un volume est reliée à la différence entre les quantités sortante et entrante dans le volume. Or, la méthode VF respecte parfaitement ce principe au niveau discret. C'est la raison pour laquelle cette méthode est largement répandue dans les codes de calcul résolvant les équations de la mécanique des fluides. C'est d'ailleurs la méthode de discrétisation spatiale qui a été utilisée dans cette thèse. Pour ces deux raisons, seule la méthode VF est présentée dans ce mémoire. Précisons que malgré la popularité de la méthode VF, les méthodes DF et EF (de type Galerkin-Discontinu) sont également utilisées et donnent de bons résultats [26, 27, 8, 9]. Après application de la méthode de discrétisation spatiale, le problème spatio-temporel est transformé en problème purement temporel : une méthode numérique d'intégration temporelle est alors utilisée pour obtenir une solution numérique

à l'équation de conservation initiale.

La première partie de ce chapitre est dédiée à la présentation de la méthode de discrétisation spatiale volumes finis. Ensuite, une présentation des principales méthodes d'intégration temporelle est réalisée.

1.1 Discrétisation spatiale des équations : la méthode volumes finis

1.1.1 La méthode volumes finis (VF) dans le cas multidimensionnel

On considère un domaine Ω de \mathbb{R}^3 , muni du repère $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$. Les variables d'espace dans les directions \vec{e}_1 , \vec{e}_2 et \vec{e}_3 sont notées x , y et z . La variable temporelle est désignée par t . On considère également une grandeur vectorielle U évoluant dans le domaine Ω :

$$U = \begin{pmatrix} u_1(x, y, z, t) \\ u_2(x, y, z, t) \\ \cdot \\ \cdot \\ \cdot \\ u_n(x, y, z, t) \end{pmatrix} \quad (1.1)$$

avec $(x, y, z, t) \in \Omega \times \mathbb{R}^+$. L'évolution de U est supposée régie par une EDP du type loi de conservation, qui s'écrit :

$$\frac{\partial U}{\partial t}(x, y, z, t) + \nabla \cdot F(U(x, y, z, t)) = 0, \quad (x, y, z, t) \in \Omega \times \mathbb{R}^+ \quad (1.2)$$

avec des conditions initiales (C.I) et des conditions aux limites (C.L). On considère un maillage du domaine Ω : celui-ci est décomposé en un nombre fini de volumes élémentaires (cellules) Ω_i . Par commodité, nous supposons que les cellules sont invariantes au cours du temps. La méthode VF requiert d'intégrer l'équation (1.2) sur chaque volume élémentaire Ω_i de frontière Γ_i :

$$\iiint_{\Omega_i} \frac{\partial U}{\partial t}(x, y, z, t) d\Omega_i + \iint_{\Gamma_i} \nabla \cdot F(U(x, y, z, t)) d\Omega_i = 0. \quad (1.3)$$

En appliquant le théorème de Green-Ostrogradski, l'équation (1.3) devient :

$$\iiint_{\Omega_i} \frac{\partial U}{\partial t}(x, y, z, t) d\Omega_i + \iint_{\Gamma_i} F(U(x, y, z, t)).n_i d\Gamma_i = 0, \quad (1.4)$$

où n_i est le vecteur unité, normal à Γ_i et pointant vers l'extérieur de Ω_i . Le volume élémentaire Ω_i étant invariant au cours du temps, on peut "sortir" la dérivée temporelle de l'intégrale dans le premier terme de (1.4) :

$$\iiint_{\Omega_i} \frac{\partial U}{\partial t}(x, y, z, t) d\Omega_i = \frac{d}{dt} \iiint_{\Omega_i} U(x, y, z, t) d\Omega_i. \quad (1.5)$$

On désigne par U_i la valeur moyenne de U dans Ω_i :

$$U_i(t) = \frac{1}{|\Omega_i|} \iiint_{\Omega_i} U(x, y, z, t) d\Omega_i, \quad (1.6)$$

où $|\Omega_i|$ est le volume de l'élément Ω_i . La valeur U_i est la solution numérique sur la cellule Ω_i . En utilisant l'équation (1.6), la relation (1.5) devient :

$$\iiint_{\Omega_i} \frac{\partial U}{\partial t}(x, y, z, t) d\Omega_i = |\Omega_i| \frac{d}{dt} U_i(t). \quad (1.7)$$

Le volume élémentaire Ω_i peut être un cube, comme sur la figure 1.1. Cette figure fait également apparaître la face $\Gamma_{i,l}$ du cube, de normale sortante unitaire $n_{i,l}$, et de surface $S_{i,l}$. L'intégrale sur toute la surface extérieure du cube Ω_i est donc égale à la somme des intégrales sur chaque face $\Gamma_{i,l}$ du cube :

$$\iint_{\Gamma_i} F(U(x, y, z, t)).n_i d\Gamma_i = \sum_l \iint_{\Gamma_{i,l}} F(U(x, y, z, t)).n_{i,l} d\Gamma_{i,l} \quad (1.8)$$

En considérant U constant sur chaque face $\Gamma_{i,l}$, l'intégration de $F(U)$ sur la face $\Gamma_{i,l}$ revient à multiplier le terme constant $F(U)$ par la surface $S_{i,l}$:

$$\sum_l \iint_{\Gamma_{i,l}} F(U(x, y, z, t)).n_{i,l} d\Gamma_{i,l} = \sum_{l=1}^6 F(U).n_{i,l} S_{i,l}. \quad (1.9)$$

Un des enjeux principaux d'une méthode VF est de trouver une bonne approximation de $F(U)$ sur les faces [28]. En effet, les seules données disponibles dans le formalisme VF sont les solutions numériques U_i , qui ne sont que des valeurs moyenne de U dans les cellules. Les approximations de $F(U)$ aux faces sont appelés flux numériques. Ils peuvent dépendre de la solution numérique sur la cellule Ω_i mais également de la solution numérique sur plusieurs cellules "voisines" de Ω_i . La discrétisation spatiale complète de l'équation (1.2) par la méthode VF s'écrit donc :

$$\frac{d}{dt}U_i + \frac{1}{|\Omega_i|} \sum_l F_{i,l}(U_{i-\alpha}, \dots, U_i, \dots, U_{i+\beta}) \cdot n_{i,l} S_{i,l} = 0, \quad (1.10)$$

où $F_{i,l}(U_{i-\alpha}, \dots, U_i, \dots, U_{i+\beta})$ est le flux numérique sur la face l de la cellule Ω_i , dépendant de U_i et de $\alpha + \beta$ solution numériques voisines de Ω_i (α et β étant des entiers). Les différentes cellules dont dépend le flux numérique est le stencil du flux numérique. Il existe dans la littérature une large gamme de flux numériques.

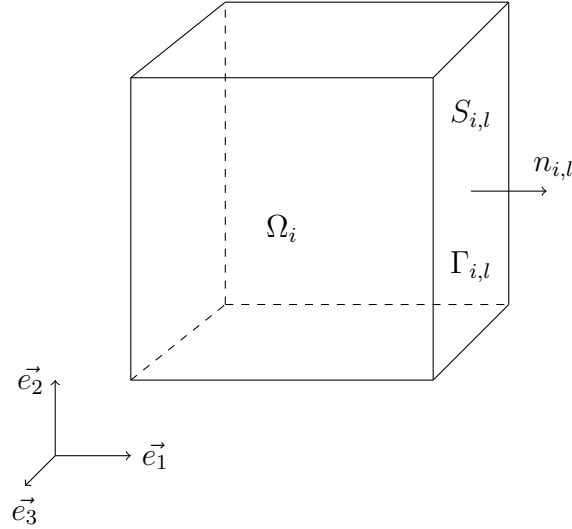


FIGURE 1.1 – Représentation d’un volume élémentaire Ω_i et des différentes données au niveau de sa face l .

Pour des raisons de clarté, la plupart des schémas numériques présentés dans cette thèse sont basés sur la méthode VF appliquée dans un cas scalaire (unidimensionnel). Une brève présentation de la méthode VF dans ce cas particulier est donc fournie, et les notations qui seront utilisées tout au long de ce mémoire sont introduites.

1.1.2 La méthode volumes finis dans le cas unidimensionnel

On considère maintenant un espace unidimensionnel muni de la variable d’espace x et de la variable temporelle t . On considère également une grandeur scalaire $u(x, t)$ évoluant dans un domaine Ω . L’évolution de u est supposée régie par une EDP scalaire, unidimensionnelle, de type loi de conservation, qui s’écrit :

$$\frac{\partial u}{\partial t}(x, t) + \frac{\partial F(u(x, t))}{\partial x} = 0 \quad (x, t) \in \Omega \times \mathbb{R}^+, \quad (1.11)$$

avec des conditions initiales (C.I) et des conditions aux limites (C.L). Dans le cas unidimensionnel, les cellules constituant le maillage sont des segments, comme le montre la figure 1.2. On suppose que le maillage est constitué de N cellules réelles (en noir) et de deux cellules fictives (en rouges) utilisées pour les conditions aux limites. Chaque cellule

réelle i , de milieu x_i , est délimitée par les points $x_{i+\frac{1}{2}}$ et $x_{i-\frac{1}{2}}$. La longueur de la cellule i est Δx_i . La solution numérique sur la cellule i est notée u_i . La méthode VF est appliquée uniquement sur les cellules réelles (cellules 1 à N). Le rôle des cellules fictives 0 et $N + 1$ est de permettre le calcul des flux aux bords du domaine de calcul. Le remplissage des cellules fictives dépend des conditions aux limites du problème.

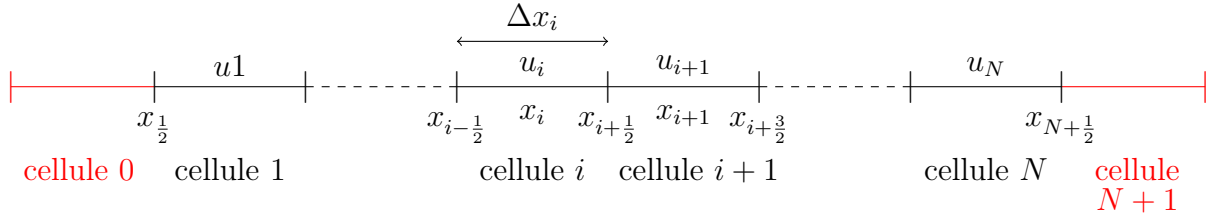


FIGURE 1.2 – Maillage unidimensionnel utilisé pour appliquer la méthode VF dans le cas scalaire, avec les cellules réelles (en noir) et les cellules fictives (en rouge).

L'intégration de l'équation de conservation scalaire (1.11) sur la cellule i donne :

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial u}{\partial t}(x, t) dx + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x} F(u(x, t)) dx = 0 \quad (x, t) \in \Omega \times \mathbb{R}^+. \quad (1.12)$$

L'intégration du terme en dérivée spatiale fait apparaître les flux numériques $F_{i+\frac{1}{2}}$ et $F_{i-\frac{1}{2}}$ aux bords $x_{i+\frac{1}{2}}$ et $x_{i-\frac{1}{2}}$ de la cellule i . On introduit également la valeur moyenne u_i sur cette cellule :

$$u_i = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t) dx. \quad (1.13)$$

Ceci permet d'obtenir la forme discrétisée en espace de l'équation de conservation scalaire (1.11) :

$$\frac{d}{dt} u_i + \frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_{i-\alpha}, \dots, u_i, \dots, u_{i+\beta}) - F_{i-\frac{1}{2}}(u_{i-1+\alpha}, \dots, u_{i-1}, \dots, u_{i-1+\beta}) \right) = 0 \quad (1.14)$$

où les flux numériques peuvent dépendre de $\alpha + \beta$ solutions numériques sur les cellules "voisines" de la cellule i . Cependant, pour alléger les notations, on admettra sauf mention contraire que les flux numériques aux interfaces dépendent uniquement de deux solutions numériques, situées de part et d'autre des interfaces :

$$\frac{d}{dt} u_i = -\frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right). \quad (1.15)$$

Pour des raisons de clarté et dans l'optique d'alléger les notations, on introduit D_i , le bilan des flux sur la cellule i :

$$D_i(u_{i-1}, u_i, u_{i+1}) = \frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right). \quad (1.16)$$

L'équation discrétisée en espace (1.15) peut donc se réécrire sous la forme suivante, qui sera fréquemment utilisée au cours de ce mémoire :

$$\frac{d}{dt} u_i = -D_i(u_{i-1}, u_i, u_{i+1}). \quad (1.17)$$

Le second membre D_i dépend de trois solutions numériques : nous supposons donc que le schéma numérique possède un **stencil** égal à trois. La relation (1.17) est valable sur chaque cellule i . Un système différentiel dont la taille est égale au nombre de cellules sur le maillage est donc obtenu :

$$\frac{dU}{dt} = -D(U), \quad (1.18)$$

avec $U = (u_1, \dots, u_i, u_{i+1}, \dots, u_N)^T$, le vecteur des solutions numériques sur chaque cellule et $D = (D_1, \dots, D_i, D_{i+1}, \dots, D_N)^T$, le vecteur du bilan des flux sur chaque cellule.

Cette approche qui consiste à discrétiser spatialement une équation aux dérivées partielles pour se ramener à un système d'équations différentielles, est appelée méthode des lignes. Une méthode numérique d'intégration temporelle peut être utilisée pour résoudre le système (1.18) et ainsi obtenir une solution numérique (approchée) de l'équation de conservation initiale.

1.1.2.1 Remarques sur les flux numériques

Différentes méthodes existent pour calculer les flux numériques. Dans la suite de ce mémoire, nous serons amenés à utiliser des flux centrés et décentrés. Une brève présentation de ces deux types de flux est donc fournie ci-dessous.

- Dans le cas d'un flux centré, la solution à l'interface $x_{i+\frac{1}{2}}$ est approchée par la moyenne arithmétique des solutions numériques de part et d'autre de cet interface. Le flux numérique centré est l'évaluation de F en cette valeur approchée :

$$F_{i+\frac{1}{2}}(u_i, u_{i+1}) = F\left(\frac{u_i + u_{i+1}}{2}\right) \quad (1.19)$$

- Dans le cas d'un flux décentré, la solution à l'interface $x_{i+\frac{1}{2}}$ dépend du signe de la vitesse à laquelle la solution u est transportée. Pour une équation d'advection à vitesse a constante, le second membre $F(u(x, t))$ de l'équation de conservation a pour expression :

$$F(u(x, t)) = au(x, t) \quad (1.20)$$

Si la vitesse a est positive, la solution u_i sur chaque cellule i est transportée de la gauche vers la droite. Par conséquent, la solution à l'interface $x_{i+\frac{1}{2}}$ provient de la cellule de gauche : il s'agit donc de la solution u_i . Par conséquent, si $a > 0$ le flux décentré a pour expression :

$$F_{i+\frac{1}{2}}(u_i, u_{i+1}) = F(u_i) = au_i \quad (1.21)$$

Dans la suite, on présente un principe important d'une méthode VF : le principe de conservation, évoqué dans la présentation de ce chapitre.

1.1.3 Principe de conservation

Supposons que l'on ait un domaine hermétique contenant une certaine quantité de matière. La quantité de matière doit rester constante au cours du temps car aucun échange n'est possible avec l'environnement extérieur (il n'y a aucune perte, ni aucun gain). On va montrer que la méthode VF utilisée en conjonction avec la méthode d'intégration temporelle Euler explicite vérifie cette propriété. On dit alors que les méthodes VF et Euler explicite sont conservatives. On utilise pour cette démonstration la forme discrétisée en espace de l'équation de conservation scalaire :

$$\frac{d}{dt}u_i = -\frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right) \quad (1.22)$$

La forme discrétisée en espace (1.22) est intégrée temporellement à l'aide de la méthode Euler explicite. On considère un instant t^n quelconque et un pas de temps de temps Δt . La méthode d'Euler explicite nécessite d'intégrer (1.22) entre t^n et $t^{n+1} = t^n + \Delta t$:

$$\int_{t^n}^{t^{n+1}} \frac{d}{dt}u_i dt = -\frac{1}{\Delta x_i} \int_{t^n}^{t^{n+1}} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right) dt \quad (1.23)$$

Par définition de l'intégrale, le membre de gauche de l'équation (1.23) devient :

$$\int_{t^n}^{t^{n+1}} \frac{d}{dt}u_i dt = u_i^{n+1} - u_i^n, \quad (1.24)$$

où u_i^{n+1} et u_i^n sont les solutions numériques sur la cellule i aux temps t^{n+1} et t^n . Le membre de droite de (1.23) est approché par la méthode des rectangles (l'intégrande est évaluée au temps t^n puis multipliée par Δt) :

$$\int_{t^n}^{t^{n+1}} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right) dt \approx \Delta t \left(F_{i+\frac{1}{2}}(u_i^n, u_{i+1}^n) - F_{i-\frac{1}{2}}(u_{i-1}^n, u_i^n) \right) \quad (1.25)$$

Le schéma VF intégré temporellement avec la méthode Euler explicite s'écrit donc :

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i^n, u_{i+1}^n) - F_{i-\frac{1}{2}}(u_{i-1}^n, u_i^n) \right) \quad (1.26)$$

La forme intégrée temporellement étant obtenue, on va maintenant prouver que la méthode VF utilisée conjointement avec la méthode d'intégration temporelle d'Euler explicite est conservative. La quantité totale de la grandeur u dans le domaine de calcul Ω s'écrit :

$$q = \int_{\Omega} u \, dx. \quad (1.27)$$

On peut approcher la quantité q par la quantité \tilde{q} qui s'écrit :

$$\tilde{q} = \sum_{i=1}^N u_i \Delta x_i \approx q. \quad (1.28)$$

En sommant l'équation (1.26) sur toutes les cellules i du domaine de calcul, on obtient :

$$\underbrace{\sum_{i=1}^N u_i^{n+1} \Delta x_i}_{\tilde{q}^{n+1}} = \underbrace{\sum_{i=1}^N u_i^n \Delta x_i}_{\tilde{q}^n} - \sum_{i=1}^N \left(F_{i+\frac{1}{2}}(u_i^n, u_{i+1}^n) - F_{i-\frac{1}{2}}(u_{i-1}^n, u_i^n) \right) \quad (1.29)$$

où \tilde{q}^{n+1} et \tilde{q}^n désignent respectivement la quantité de la grandeur u dans le domaine de calcul Ω aux temps t^{n+1} et t^n . Dans l'équation (1.29), un changement d'indice dans les termes somme peut être réalisé :

$$\tilde{q}^{n+1} = \tilde{q}^n - \sum_{i=1}^N F_{i+\frac{1}{2}}(u_i^n, u_{i+1}^n) + \sum_{i=0}^{N-1} F_{i+\frac{1}{2}}(u_i^n, u_{i+1}^n) \quad (1.30)$$

Ainsi, les termes de flux s'annulent deux à deux, sauf pour les termes de flux aux bords du domaine : $F_{\frac{1}{2}}$ et $F_{N+\frac{1}{2}}$. Or ces flux sont égaux à zéro puisqu'on suppose que le domaine de calcul est isolé (il n'y a pas de flux à travers les bords du domaine). La quantité totale de la grandeur u est donc inchangée entre les instants t^n et t^{n+1} :

$$\tilde{q}^{n+1} = \tilde{q}^n \quad (1.31)$$

Par conséquent le schéma VF utilisé en conjonction avec la méthode Euler explicite est conservatif. Néanmoins, l'utilisation de certaines méthodes d'intégration temporelle peut amener à la perte de cette propriété de conservativité. Dans la revue des grandes méthodes d'intégration temporelle réalisée ci-après, une attention particulière sera portée au caractère conservatif (ou non) des méthodes.

1.2 Exigences concernant les méthodes d'intégration temporelle

Dans cette thèse, on s'intéresse particulièrement à la simulation numérique des écoulements de fluides turbulents. Dans cette partie, une description des écoulements turbulents est d'abord effectuée, puis on s'appuie sur cette description pour déterminer trois grands critères que doivent satisfaire les méthodes d'intégration temporelle pour simuler correctement et efficacement ce type d'écoulement.

1.2.1 Caractéristiques des écoulements turbulents

Les écoulements turbulents présentent différentes caractéristiques, énumérées ci-dessous.

- On estime qu'un écoulement devient turbulent lorsque le nombre de Reynolds Re caractérisant l'écoulement est supérieur à 2000 [29]. Le nombre de Reynolds est un nombre sans dimension défini par :

$$Re = \frac{\rho V L}{\mu}, \quad (1.32)$$

où ρ est la masse volumique du fluide, V une vitesse caractéristique du fluide, L une dimension caractéristique de l'écoulement et μ la viscosité dynamique du fluide.

- La turbulence a un caractère fortement instationnaire, aléatoire. Par exemple, si on place une sonde de pression en un point de l'écoulement, le signal de pression enregistré aura l'aspect montré par la figure 1.3 : des oscillations irrégulières mais qui se produisent autour d'une valeur moyenne [29].
- Un écoulement turbulent est extrêmement sensible aux conditions initiales (caractéristique des systèmes chaotiques).
- Un écoulement turbulent est tridimensionnel et rotationnel (les écoulements turbulents sont constitués de tourbillons).
- Les grands tourbillons (associés aux petites longueurs d'onde) extraient de l'énergie cinétique à l'écoulement moyen. Il y a ensuite un transfert d'énergie des grands tourbillons vers les tourbillons plus petits (associés aux grandes longueurs d'onde), jusqu'à ce que la viscosité moléculaire dissipe l'énergie de ces derniers. On appelle échelle de Kolmogorov la taille des tourbillons dont l'énergie cinétique est totalement dissipée par la viscosité. Le spectre d'énergie cinétique de la turbulence comprend d'abord une zone de production aux petits nombres d'onde, une zone de transfert d'énergie des petits nombres d'onde vers les grands, et enfin une zone de dissipation aux grands nombres d'onde. Ceci est représenté par la figure 1.4.

Pour simuler des écoulements turbulents, une première approche consiste à résoudre directement les équations de la mécanique des fluides (équations de Navier-Stokes) sur un maillage très fin afin de capter la totalité des échelles spatiales de la turbulence, et de simuler l'ensemble du spectre d'énergie montré sur la figure 1.4. Cette approche est appelée **DNS (Direct Numerical Simulation)**. Les mailles doivent être de taille inférieure à la plus petite échelle spatiale turbulente. Ceci génère un nombre de points de maillage très élevé, ce qui rend cette approche actuellement inutilisable sur des configurations industrielles car elle requiert une puissance de calcul qui dépasse les capacités des supercalculateurs actuels. Le lecteur pourra se reporter à [30] pour plus d'informations sur la DNS.

Une seconde approche est l'approche **RANS (Reynolds Average Navier-Stokes)**. Elle consiste à modéliser l'ensemble du spectre d'énergie de la turbulence : chaque grandeur (vitesse, pression...) est écrite comme la somme d'une valeur moyenne et d'une valeur fluctuante. Cette décomposition est injectée dans les équations de Navier-Stokes, et un

opérateur de moyenne est appliqué. Des inconnues supplémentaires apparaissent alors dans le système des équations moyennées ; différents modèles RANS existent pour "fermer" le système. Les modèles RANS permettent de réaliser des simulations rapides mais ils donnent généralement de très bons résultats, mais principalement sur les cas pour lesquels ils ont été "calibrés". Pour davantage de précisions concernant les différents modèles RANS existant, le lecteur peut se référer à [31].

Enfin, l'approche **LES (Large Eddy Simulation)** est une approche intermédiaire entre les deux précédentes. Elle consiste à simuler directement les grandes échelles turbulentes (les plus énergétiques, responsable d'une part importante de quantité de mouvement) et à modéliser les plus petites. Des modèles "sous-maille" sont utilisés pour la modélisation des plus petites échelles. Le lecteur peut se reporter [6] pour davantage de précisions sur la LES et les différents modèles sous-maille. C'est une approche prometteuse, qui permet de décrire plus précisément la structure de l'écoulement que l'approche RANS puisqu'elle donne l'avantage de représenter les grandes structures instationnaires. Néanmoins, cette approche reste encore assez coûteuse ; un travail d'optimisation de la méthode d'intégration temporelle est réalisé dans cette thèse pour tenter de rendre cette technique plus abordable.

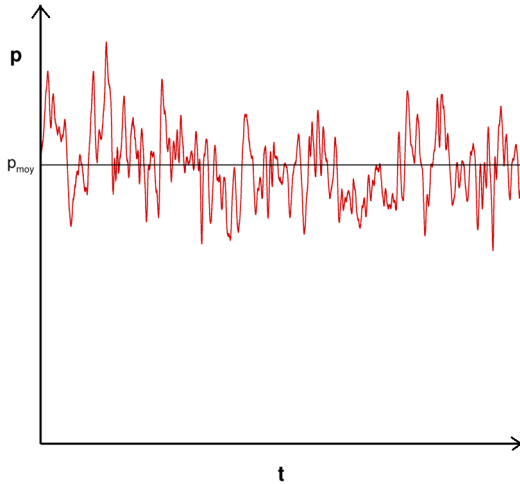


FIGURE 1.3 – Allure d'un signal de pression enregistré en un point d'un écoulement turbulent.

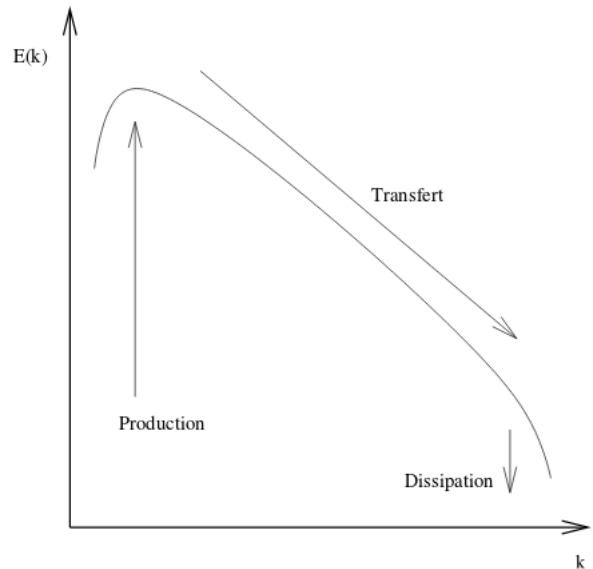


FIGURE 1.4 – Spectre de l'énergie cinétique turbulente $E(k)$ en fonction du nombre d'onde k .

1.2.2 Trois critères essentiels pour la simulation des écoulements turbulents

Nous avons déterminé trois critères qui doivent être respectés par la méthode d'intégration temporelle pour une simulation efficace des écoulements turbulents.

— Critère de précision :

Comme le montre le signal de pression de la figure 1.3, un écoulement turbulent a

un caractère fortement aléatoire et complexe. La méthode d'intégration temporelle doit donc être suffisamment précise pour décrire ce comportement

— Critère de coût CPU :

La méthode d'intégration temporelle doit fournir une solution numérique dans un délai de temps raisonnable. Les simulations instationnaires de type LES (DNS) nécessitent de calculer directement certaines (toutes les) échelles turbulentes. Un maillage assez fin est donc nécessaire, ce qui tend à augmenter les temps de simulation. La méthode d'intégration temporelle doit donc être la plus performante possible pour minimiser le temps de calcul. Ce critère est d'autant plus important que les écoulements turbulents doivent être simulés sur des temps assez longs. En effet, la turbulence étant un phénomène très aléatoire, il est judicieux d'effectuer des moyennes statistiques de l'écoulement au cours du temps. Pour obtenir des grandeurs moyennes de qualité, qui pourront être comparées aux grandeurs moyennes obtenues de manière expérimentale, les moyennes statistiques doivent être accumulées sur une durée assez importante.

— Critère de compatibilité avec le HPC (High Performance Computing) :

Il existe à l'heure actuelle plusieurs techniques, dites techniques HPC, permettant d'accélérer un code informatique. Il y a d'une part les techniques de **parallélisation MPI (Message Passing Interface) et OpenMP** qui permettent d'exécuter un code informatique sur plusieurs cœurs de calcul "en parallèle". Les différentes parties d'un code informatique sont exécutées simultanément par plusieurs cœurs de calcul. Cela réduit le temps d'exécution par rapport au cas où un seul cœur exécute les différentes parties du code séquentiellement (les unes après les autres). Il y a d'autre part les **techniques de vectorisation** qui permettent aux cœurs de calcul d'effectuer plusieurs opérations simultanément. La vectorisation n'est possible que sur des données contiguës en mémoire. Enfin, il existe des **techniques d'optimisation des accès mémoires**. L'idée est de faire en sorte que le maximum de données auxquelles le processeur accède soient présentes dans sa mémoire cache L1 (zone mémoire à accès rapide) plutôt que dans la RAM (zone mémoire à accès plus lent). Toutes ces techniques peuvent permettre d'obtenir des accélérations spectaculaires sur le temps d'exécution d'un code informatique. Si une méthode d'intégration temporelle est intrinsèquement rapide mais qu'elle est peu compatible avec les techniques HPC, elle risque d'être dépassée par une méthode intrinsèquement moins rapide mais mieux adaptée à ces techniques.

Dans la revue des grandes méthodes d'intégration temporelle réalisée ci-après, une attention particulière sera portée au respect (ou non) de ces trois critères par les différentes méthodes d'intégration temporelle.

1.3 Les méthodes d'intégration temporelle explicites

1.3.1 Présentation générale des méthodes explicites

Nous avons vu dans la partie 1.1.2 de ce mémoire que l'étape de discrétisation spatiale d'une équation aux dérivées partielles permet de se ramener au système d'équations différentielles (1.18). On suppose que l'intervalle de temps T sur lequel on résout le système (1.18) est discrétisé en plusieurs instants t^n séparés par un pas de temps constant Δt : $t^n = n\Delta t$ (n un entier). Des estimations temporelles successives de la solution, notées U^n , sont calculées aux instants discrets successifs t^n . Une telle discrétisation temporelle est illustrée par la figure 1.5.

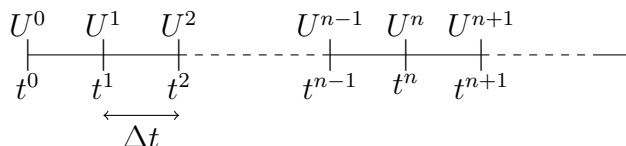


FIGURE 1.5 – Discrétisation de l'intervalle de temps

Cette discrétisation donne l'opportunité d'écrire des combinaisons linéaires de U^n et de $D(U^n)$ pour approcher le système (1.18) et calculer U^{n+1} :

$$\frac{1}{\Delta t} \sum_{k=0}^K a_k U^{n+1-k} = - \sum_{k=0}^K \alpha_k D(U^{n+1-k}). \quad (1.33)$$

Il existe deux grandes familles de méthodes d'intégration temporelle : **les méthodes implicites** et **les méthodes explicites**.

Dans le cas des méthodes explicites, le terme α_0 est forcément nul : le second membre D ne dépend pas de U^{n+1} . Par conséquent, U^{n+1} apparaît seulement dans le membre de gauche de l'équation (1.33), et se calcule directement à partir des solutions aux temps précédents. Dans le cas des méthodes explicites à un pas, seule la solution au temps t^n intervient dans le calcul de la solution au temps t^{n+1} . Dans ce cas, on a $a_k = \alpha_k = 0$ pour $k \geq 2$ dans (1.33). Éventuellement, des approximations à différents instants intermédiaires entre t^n et t^{n+1} peuvent être utilisées (méthodes de Runge-Kutta). Dans le cas des méthodes multipas, la solution au temps t^n ainsi que différentes solutions calculées à des temps antérieurs (t^{n-1} , t^{n-2} ...) sont utilisées pour le calcul de U^{n+1} .

1.3.2 Méthodes explicites à un pas : les méthodes de Runge-Kutta explicites

Les méthodes de Runge-Kutta explicites sont basées sur le calcul de différentes estimations de la solution entre t^n et t^{n+1} . La solution numérique au temps t^{n+1} est construite

par combinaison linéaire de la solution au temps t^n et des différentes estimations calculées. Pour la définition de ces méthodes, on considère l'équation modèle suivante :

$$\frac{du}{dt} = f(t, u) \quad (1.34)$$

Une définition générale des méthodes de Runge-Kutta explicites est donnée dans [32] :

Définition 1.1. Soient a_{ij} , b_i , c_i ($1 \leq i, j \leq q$) des réels. Soit u^n la solution numérique au temps t^n de l'équation (1.34). Une méthode de Runge-Kutta explicite à q étapes appliquée entre t^n et $t^{n+1} = t^n + \Delta t$ pour la résolution de l'équation (1.34) s'écrit :

$$u^{n,i} = u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K^{n,j}, \quad (1.35)$$

$$t^{n,i} = t^n + c_i \Delta t, \quad (1.36)$$

$$K^{n,i} = f(u^{n,i}, t^{n,i}), \quad (1.37)$$

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^q b_i K^{n,i}. \quad (1.38)$$

Les termes $u^{n,i}$ sont les différentes estimations de la méthode de Runge-Kutta entre t^n et t^{n+1} . Le terme $t^{n,i}$ représente le temps (compris entre t^n et t^{n+1}) auquel l'estimation $u^{n,i}$ est calculée. Le terme u^{n+1} est la solution au temps t^{n+1} .

Chaque terme $K^{n,i}$ dépend uniquement des termes calculés précédemment. On s'en convainc facilement en écrivant quelques termes K :

$$\left\{ \begin{array}{l} K^{n,1} = f(u^n, t^n + c_1 \Delta t) \\ K^{n,2} = f(u^n + \Delta t a_{21} K^{n,1}, t^n + c_2 \Delta t) \\ K^{n,3} = f(u^n + \Delta t (a_{31} K^{n,1} + a_{32} K^{n,2}), t^n + c_3 \Delta t) \\ \cdot \\ \cdot \\ \cdot \\ K^{n,q} = f(u^n + \Delta t (a_{q1} K^{n,1} + a_{q2} K^{n,2} + \dots + a_{qq-1} K^{n,q-1}), t^n + c_q \Delta t) \end{array} \right. \quad (1.39)$$

Remarque 1.1. La matrice A contenant les coefficients a_{ij} et les vecteurs b et c contenant respectivement les coefficients b_i et c_i peuvent être rangés dans un tableau de Butcher [32].

Un exemple de tableau de Butcher est donné sur le tableau 1.1 ci-dessous :

La matrice A des méthodes de Runge-Kutta explicite est triangulaire inférieure : $a_{ij} = 0$ si $j > i$. Le tableau de Butcher donne la formule des différentes estimations de la méthode

c	A	→		c_1	0					
				c_2	a_{21}					
		c_3	a_{31}	a_{32}						
		\cdot	\cdot							
		\cdot	\cdot							
		\cdot	\cdot							
		c_q	a_{q1}	a_{q2}	\cdot	\cdot	\cdot	$a_{q,q-1}$		
			b_1	b_2	\cdot	\cdot	\cdot	b_{q-1}	b_q	

TABLE 1.1 – Tableau de Butcher d’une méthode de Runge-Kutta explicite à q étapes

de Runge-Kutta. La formule d’une estimation $u^{n,i}$ ($1 \leq i \leq q$) est donnée par la i – eme ligne du tableau de Butcher :

$$u^{n,i} = u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} K^{n,j}, \quad (1.40)$$

$$K^{n,j} = f(u^{n,j}, t^n + c_j \Delta t). \quad (1.41)$$

L’expression de u^{n+1} est donnée par la dernière ligne du tableau de Butcher :

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^q b_i K^{n,i}. \quad (1.42)$$

Le coefficient a_{11} de la première ligne étant nul, on a d’après la formule (1.40) :

$$u^{n,1} = u^n.$$

Le coefficient c_1 est donc également nul puisque $u^{n,1}$ est une estimation au temps t^n . Cette première étape n’a pas beaucoup d’intérêt. Aussi, à partir du chapitre 2, cette première étape sera "court-circuitée" : on désignera directement par $u^{n,1}$ la solution de l’équation (1.34) au temps t^n .

Il est possible d’évaluer de façon théorique la précision d’une méthode d’intégration temporelle en calculant son ordre. Une définition de l’ordre d’une méthode d’intégration temporelle, tirée de [32], est présentée ci-dessous.

Définition 1.2. Soit u^{n+1} la solution numérique obtenue par application d’une méthode d’intégration temporelle à (1.34) entre les instants t^n et t^{n+1} . Soit $u(t^{n+1})$ le développement

de Taylor à t^{n+1} de la solution exacte. Une méthode d'intégration temporelle est d'ordre p si :

$$\|u(t^{n+1}) - u^{n+1}\| \leq K \Delta t^{p+1}, \quad \text{avec } K \text{ une constante.} \quad (1.43)$$

Autrement dit, $u(t^{n+1})$ et u^{n+1} doivent coïncider jusqu'au terme proportionnel à Δt^p dans le développement de Taylor. Plus l'ordre d'une méthode de Runge-Kutta est élevé, plus la méthode est considérée comme précise.

L'ordre d'une méthode de Runge-Kutta est conditionné par un certain nombre de relations portant sur ses coefficients. Tout d'abord, les méthodes de Runge-Kutta explicites vérifient la condition suivante, dite condition de consistance [32] :

$$c_i = \sum_{j=1}^q a_{ij}, \quad \text{pour } 1 \leq i \leq q. \quad (1.44)$$

Cette condition assure que chaque solution intermédiaire de la méthode de Runge-Kutta est d'ordre un [32]. Ensuite, la méthode de Runge-Kutta doit respecter des conditions, dites conditions d'ordre, portant sur ses coefficients pour être d'un ordre p donné. Dans le tableau 1.2, les conditions d'ordre un, deux et trois d'une méthode de Runge-Kutta sont listées. Nous serons amenés à les utiliser dans la suite de ce mémoire.

Condition d'ordre 1	Conditions d'ordre 2	Conditions d'ordre 3
$\sum_{i=1}^q b_i = 1$	$\sum_{i=1}^q b_i = 1$ $\sum_{i=1}^q b_i c_i = \frac{1}{2}$	$\sum_{i=1}^q b_i = 1$ $\sum_{i=1}^q b_i c_i = \frac{1}{2}$ $\sum_{i=1}^q b_i c_i^2 = \frac{1}{3}$ $\sum_{1 \leq i, j \leq q} b_i a_{ij} c_j = \frac{1}{6}$

TABLE 1.2 – Conditions d'ordre 1,2 et 3 pour les méthodes de Runge-Kutta explicites

Enfin, nous concluons ce paragraphe en présentant quelques méthodes de Runge-Kutta explicites que nous serons amenés à utiliser dans la suite de mémoire. Les méthodes sont appliquées au système différentiel (1.18).

La méthode Runge-Kutta 1 (RK1) :

Cette méthode est également connue sous le nom de méthode Euler explicite. Son expression et son tableau de Butcher sont donnés ci-dessous :

$$\begin{array}{l}
 U^{n,1} = U^n \\
 U^{n+1} = U^{n,1} - \Delta t D(U^{n,1})
 \end{array}
 \qquad
 \begin{array}{c|c}
 0 & 0 \\
 \hline
 & 1
 \end{array}$$

TABLE 1.3 – Tableau de Butcher de RK1

La méthode RK1 est d'ordre un [32]. A notre connaissance, cette méthode n'est pas utilisée pour réaliser des simulations instationnaires. Probablement à cause de sa faible précision et de sa stabilité très restreinte, qui limite grandement le pas de temps.

La méthode Runge-Kutta 2 (RK2) de ce mémoire :

Plusieurs méthodes de Runge-Kutta d'ordre deux à deux étapes existent. La méthode que nous désignons par RK2 dans ce mémoire est la méthode de Heun. Son expression et son tableau de Butcher sont donnés ci-dessous :

$$\begin{array}{l}
 U^{n,1} = U^n \\
 U^{n,2} = U^{n,1} - \Delta t D(U^{n,1}) \\
 U^{n+1} = U^{n,1} - \frac{\Delta t}{2} (D(U^{n,1}) + D(U^{n,2}))
 \end{array}
 \qquad
 \begin{array}{c|cc}
 0 & & 0 \\
 1 & & 1 \\
 \hline
 & \frac{1}{2} & \frac{1}{2}
 \end{array}$$

TABLE 1.4 – Tableau de Butcher de RK2

La méthode RK2 est d'ordre deux [32]. Cette méthode semble peu utilisée pour réaliser des simulations instationnaires de "grandes envergures". Dans [33], elle a permis de réaliser une simulation instationnaire mais "seulement" dans un contexte 2D.

La méthode Runge-Kutta 3 (RK3) de ce mémoire :

Comme pour la méthode RK2, plusieurs méthodes RK3 (méthodes de Runge-Kutta d'ordre trois à trois étapes) existent. La méthode RK3 utilisée dans ce mémoire est la méthode RK3 proposée par Lowery and Reynolds [34]. Cette méthode, d'ordre trois, est une méthode de Runge-Kutta dite "low storage". Une méthode "low storage" est construite pour que chaque nouvelle évaluation du second membre D et de la solution intermédiaire $U^{n,i}$ "écrase" la précédente. L'objectif est d'optimiser le coût mémoire de la méthode. Pour plus d'informations sur les méthodes "low storage", le lecteur peut se référer à [35]. La méthode RK3 utilisée dans ce mémoire s'écrit :

$$\begin{aligned}
U^{n,1} &= U^n \\
K^{n,1} &= \Delta t D(U^{n,1}) \\
U^{n,2} &= U^{n,1} - \beta_1 K^{n,1} & \beta_1 &= 1/2 \\
K^{n,2} &= \gamma_2 K^{n,1} + \Delta t D(U^{n,2}) & \gamma_2 &= -0.6830127 \\
U^{n,3} &= U^{n,2} - \beta_2 K^{n,2} & \beta_2 &= 0.9106836 \\
K^{n,3} &= \gamma_3 K^{n,2} + \Delta t D(U^{n,3}) & \gamma_3 &= -4/3 \\
U^{n+1} &= U^{n,3} - \beta_3 K^{n,3} & \beta_3 &= 0.3660254
\end{aligned}$$

Bien entendu, cette méthode peut aussi s'écrire sous la forme d'une méthode de Runge-Kutta classique, et un tableau de Butcher peut y être associé :

$U^{n,1} = U^n$			
$D^{n,1} = D(U^{n,1})$ $U^{n,2} = U^n - a_{21} \Delta t D^{n,1}$	$a_{21} = \beta_1$	0	0
$D^{n,2} = D(U^{n,2})$ $U^{n,3} = U^n - \Delta t(a_{31} D^{n,1} - a_{32} D^{n,2})$	$a_{31} = \beta_1 + \beta_2 \gamma_2$ $a_{32} = \beta_2$ $c_2 = \frac{1}{2}$	c_2	a_{21}
$D^{n,3} = D(U^{n,3})$ $U^{n+1} = U^n - \Delta t(b_1 D^{n,1} - b_2 D^{n,2} - b_3 D^{n,3})$	$b_1 = a_{31} + \beta_3 \gamma_2 \gamma_3$ $b_2 = \beta_2 + \beta_3 \gamma_3$ $b_3 = \beta_3$ $c_3 = 0.7886751$	c_3	$a_{31} \quad a_{32}$
			$b_1 \quad b_2 \quad b_3$

TABLE 1.5 – La méthode RK3 de ce mémoire écrite sous forme classique avec son tableau de Butcher

Les méthodes de RK3 sont beaucoup plus utilisées que les méthodes RK2 ou RK1 pour réaliser des simulations instationnaires, grâce à leur meilleure précision et leur meilleure stabilité (qui permet d'accroître le pas de temps, et donc de réduire le coût CPU). La méthode RK3 de ce mémoire a par exemple été utilisée par I.Bennaceur [36, 37] pour réaliser des simulations LES de la diffusion d'une onde acoustique au sein d'une couche de cisaillement turbulente. Une autre méthode RK3 (différente de celle présentée dans ce mémoire) a été utilisée dans [38] pour réaliser des simulations LES.

Une méthode Runge-Kutta 4 (RK4) :

Cette méthode de Runge-Kutta d'ordre quatre à quatre étapes est probablement la méthode de Runge-Kutta la plus connue et la plus utilisée, grâce à sa bonne précision. Son expression et son tableau de Butcher sont donnés ci-dessous :

$$\begin{aligned}
 U^{n,1} &= U^n & 0 & \left| \begin{array}{c} 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \right. \\
 U^{n,2} &= U^{n,1} - \frac{\Delta t}{2} D(U^{n,1}) \\
 U^{n,3} &= U^{n,1} - \frac{\Delta t}{2} D(U^{n,2}) \\
 U^{n,4} &= U^{n,1} - \Delta t D(U^{n,3}) \\
 U^{n+1} &= U^{n,1} - \frac{\Delta t}{6} (D(U^{n,1}) + 2 D(U^{n,2}) \\
 &\quad + 2 D(U^{n,3}) + D(U^{n,4}))
 \end{aligned}$$

TABLE 1.6 – Tableau de Butcher de RK4

Elle a notamment été utilisée dans [39] pour réaliser une DNS.

1.3.3 Les méthodes explicites multipas : les méthodes d'Adams-Bashforth explicites

Les méthodes d'Adams-Bashforth explicites sont de la forme :

$$U^{n+1} = U^n - \Delta t \sum_{i=1}^k \beta_k D(U^{n+1-k}) \quad (1.45)$$

En particulier, la méthode d'Adams-Bashforth d'ordre 2 s'écrit :

$$U^{n+1} = U^n - \Delta t \left(\frac{3}{2} D(U^n) - \frac{1}{2} D(U^{n-1}) \right) \quad (1.46)$$

La méthode d'Adams-Bashforth d'ordre deux (1.46) est la méthode d'Adams-Bashforth présentant le meilleur compromis entre stabilité et précision. En effet, les méthode d'Adams-Bashforth d'ordre supérieur voient leur domaine de stabilité se réduire drastiquement, ce qui impose de fortes limitations sur leur pas de temps et allonge considérablement les temps de calcul [32].

La méthode d'Adams-Bashforth d'ordre deux (1.46) est utilisée pour réaliser des simulations instationnaires. Par exemple, elle a été utilisée dans [40, 41] pour réaliser une DNS d'un écoulement turbulent dans un canal. Dans [42], elle a permis d'effectuer une simulation LES d'un jet turbulent.

1.3.4 Stabilité concernant l'intégration d'une équation différentielle

De manière générale, l'analyse de stabilité d'une méthode d'intégration temporelle permet d'estimer le pas de temps maximal qui peut être utilisé par la méthode. Pour analyser la stabilité d'une méthode d'intégration temporelle, une analyse de stabilité linéaire est très souvent pratiquée. L'analyse de stabilité linéaire est réalisée en appliquant la méthode d'intégration temporelle à l'équation test de Dahlquist :

$$\frac{du}{dt} = \lambda u, \quad \text{avec } \lambda \in \mathbb{C} \text{ et } u \in \mathbb{R}. \quad (1.47)$$

Certes, l'analyse de stabilité sur un cas simple et linéaire tel que l'équation de Dahlquist n'est pas complètement représentative de la stabilité de la méthode dans un cas non linéaire. Néanmoins, dans de nombreuses applications, l'analyse de stabilité linéaire suffit à donner des informations valides sur le comportement de la méthode dans le cas non linéaire [43]. Pour plus d'informations concernant l'analyse de stabilité linéaire et l'origine de l'équation test de Dahlquist, le lecteur peut se référer à [32].

La méthode Euler explicite appliquée à l'équation test de Dahlquist (1.47) s'écrit :

$$\begin{aligned} u^{n+1} &= u^n + \Delta t \lambda u^n \\ &= (1 + z)u^n \quad (\text{en posant } z = \lambda \Delta t) \\ &= R(z)u^n, \end{aligned} \quad (1.48)$$

où $R(z)$ est la fonction de stabilité de la méthode. Cette fonction permet d'accéder au domaine de stabilité de la méthode, dont une définition donnée ci-dessous est tirée de [32].

Définition 1.3. *Le domaine de stabilité d'une méthode d'intégration temporelle est l'ensemble S des complexes z tels que $|R(z)| \leq 1$:*

$$S = \{z \in \mathbb{C} : |R(z)| \leq 1\} \quad (1.49)$$

D'après la définition 1.3, le domaine de stabilité de la méthode Euler explicite est l'ensemble des nombres complexes z vérifiant $|1+z| \leq 1$. Il s'agit donc du disque de diamètre 1 et centré sur le réel -1 . Sur la figure 1.6, on a représenté (à l'aide d'un programme informatique) le domaine stabilité de la méthode Euler explicite et celui de la méthode RK2, dont la fonction de stabilité s'écrit : $R(z) = 1 + z + \frac{z^2}{2}$.

La figure 1.6 montre que le domaine de stabilité de la méthode RK2 est plus grand que celui de la méthode Euler explicite. Dans le cas des méthodes de Runge-Kutta, l'augmentation de l'ordre et du nombre d'étapes va généralement de pair avec une augmentation de la stabilité. Cette figure 1.6 montre donc que les pas de temps utilisés par la méthode RK2 peuvent être plus grands que ceux utilisés par la méthode Euler explicite.

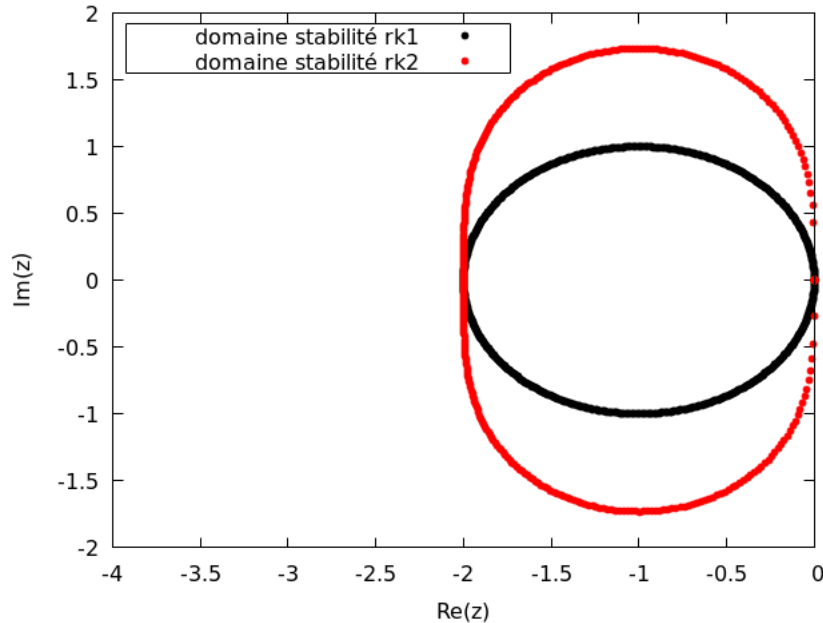


FIGURE 1.6 – Domaine de stabilité des méthodes Euler explicite (RK1) et RK2

1.3.5 Stabilité concernant l'intégration d'une EDP discrétisée en espace

Nous avons vu au début de ce mémoire que la discrétisation spatiale d'une EDP mène à un système différentiel. Une analyse de stabilité similaire à celle réalisée dans le cas des équations différentielles exigerait de trouver une relation récurrence entre le vecteur des solutions au temps t^{n+1} et le vecteur des solutions au temps t^n . Cette tâche peut se révéler assez difficile puisqu'elle nécessite de diagonaliser le système différentiel. Heureusement, Von Neumann a publié dans les années 1950 une méthode simple et efficace pour analyser la stabilité d'une méthode d'intégration temporelle appliquée à une EDP discrétisée en espace. A l'image de l'étude de stabilité sur les équations différentielles, l'étude de stabilité concernant les EDP est réalisée sur un problème linéaire, plus simple que les problèmes "réels", généralement non linéaires. L'étude de stabilité sur un problème simple et linéaire permet tout de même d'accéder à des informations importantes sur le comportement général de la méthode d'intégration temporelle. De plus, la stabilité dans le cas linéaire est une condition nécessaire pour la stabilité dans le cas non linéaire [44]. A titre d'exemple, on choisit de baser notre étude de stabilité sur une équation d'advection scalaire à coefficient a constant et positif :

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (x, t) \in \mathbb{R} \times \mathbb{R}^+ \quad (1.50)$$

Cette équation (1.50) est discrétisée spatialement sur le maillage de la figure 1.2 par le biais de la méthode VF. Un schéma flux décentrés est utilisé pour le calcul des flux numériques. Le système différentiel suivant est donc obtenu :

$$\frac{du_i}{dt} = -\frac{1}{\Delta x} (au_i - au_{i-1}), \quad 1 \leq i \leq N. \quad (1.51)$$

La méthode Euler explicite appliquée à (1.51) s'écrit :

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x} (au_i^n - au_{i-1}^n), \quad 1 \leq i \leq N. \quad (1.52)$$

La méthode de Von Neumann repose sur la transformée de Fourier discrète [43]. On pose donc :

$$u_i^n = \hat{u}^n e^{Jki\Delta x} \quad 1 \leq i \leq N, \quad (1.53)$$

où \hat{u}^n est l'amplitude, $J = \sqrt{-1}$, et $k\Delta x$ est l'angle de phase associé au nombre d'onde k . Le schéma (1.52) devient :

$$\hat{u}^{n+1} e^{Jki\Delta x} = \hat{u}^n e^{Jki\Delta x} - \frac{a\Delta t}{\Delta x} (\hat{u}^n e^{Jki\Delta x} - \hat{u}^n e^{Jk(i-1)\Delta x}). \quad (1.54)$$

Après quelques simplifications, la relation (1.54) devient :

$$\hat{u}^{n+1} = \hat{u}^n - \hat{u}^n \frac{a\Delta t}{\Delta x} (1 - e^{-Jk\Delta x}). \quad (1.55)$$

Le facteur d'amplification h du schéma, défini par $h = \frac{\hat{u}^{n+1}}{\hat{u}^n}$, s'écrit :

$$h = 1 - \frac{a\Delta t}{\Delta x} (1 - e^{-Jk\Delta x}). \quad (1.56)$$

D'après Von Neumann, le schéma est stable si $|h| \leq 1$ [43]. Un rapide calcul montre que $|h| \leq 1$ si :

$$-2a \frac{\Delta t}{\Delta x} (1 - \cos(k\Delta x)) \left(1 - \frac{a\Delta t}{\Delta x}\right) \leq 0, \quad (1.57)$$

Soit :

$$\frac{a\Delta t}{\Delta x} \leq 1. \quad (1.58)$$

La condition (1.58) est appelée condition CFL (Courant, Friedrichs, Lewy). Cette condition de stabilité ne porte plus uniquement sur le pas de temps Δt (comme c'était le cas pour l'étude de stabilité concernant les équations différentielles), mais aussi sur la vitesse de transport a de la solution et sur le pas d'espace Δx . On remarque notamment que le pas de temps Δt est d'autant plus restreint que plus le pas d'espace du maillage Δx est faible, et que la vitesse de transport a est grande. Le pas de temps des méthodes explicite est donc toujours limité par une condition de stabilité numérique (condition *CFL*). Ce n'est pas forcément le cas des méthodes implicites, comme on le verra par la suite.

1.3.6 Précision des méthodes explicites

On reprend le schéma numérique (1.52) que l'on écrit sous une forme équivalente :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{a}{\Delta x} (u_i^n - u_{i-1}^n) = 0 \quad (1.59)$$

Pour étudier la précision de la méthode Euler explicite, on se propose de déterminer l'équation modifiée du schéma [28]. En effet, les schémas numériques ne sont que des approximations discrètes de l'équation continue. Par conséquent, ils ne vérifient pas exactement l'équation continue mais une équation légèrement différente, appelée équation modifiée [28]. Pour déterminer l'équation modifiée du schéma, on définit l'erreur locale τ_i^n , commise en injectant une certaine fonction $u(x_i, t^n)$ dans le schéma (1.59) :

$$\tau_i^n = \frac{u(x_i, t^n + \Delta t) - u(x_i, t^n)}{\Delta t} + \frac{a}{\Delta x} (u(x_i, t^n) - u(x_i - \Delta x, t^n)), \quad (1.60)$$

On suppose que $u(x_i, t^n)$ admet des dérivées partielles par rapport à ses deux variables. Dans l'équation (1.60), les termes $u(x_i, t^n + \Delta t)$ et $u(x_i - \Delta x, t^n)$ sont obtenus par développement de Taylor autour de t^n et de x_i :

$$u(x_i, t^n + \Delta t) = u(x_i, t^n) + \Delta t \frac{\partial u}{\partial t}(x_i, t^n) + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t^n) + \mathcal{O}(\Delta t^3) \quad (1.61)$$

$$u(x_i - \Delta x, t^n) = u(x_i, t^n) - \Delta x \frac{\partial u}{\partial x}(x_i, t^n) + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta x^3) \quad (1.62)$$

En introduisant les développements de Taylor (1.61) et (1.62) dans l'équation (1.60), on obtient l'expression suivante pour l'erreur locale τ_i^n :

$$\tau_i^n = \frac{\partial u}{\partial t}(x_i, t^n) + a \frac{\partial u}{\partial x}(x_i, t^n) + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t^n) - \frac{a \Delta x}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta t^2, \Delta x^2) \quad (1.63)$$

Si $u(x_i, t^n)$ est solution de l'équation d'advection (1.50) alors les termes dominants dans l'erreur locale (1.63) sont de l'ordre de Δt et Δx . Par contre, si $u(x_i, t^n)$ est solution de l'équation :

$$\frac{\partial u}{\partial t}(x_i, t^n) + a \frac{\partial u}{\partial x}(x_i, t^n) + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t^n) - \frac{a \Delta x}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t^n) = 0, \quad (1.64)$$

alors les termes dominants dans l'erreur locale (1.63) sont de l'ordre de Δt^2 et Δx^2 . Dans ce cas, l'erreur locale (1.63) est plus faible, ce qui signifie que le schéma numérique (1.59) est plus "proche" de l'équation (1.64) que de l'équation d'advection (1.50). L'équation (1.64) est l'équation modifiée du schéma. L'équation (1.64) peut s'écrire sous une forme plus simple, en remarquant que :

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2}(x_i, t^n) &= -a \frac{\partial^2 u}{\partial t \partial x}(x_i, t^n) + \mathcal{O}(\Delta t, \Delta x), \\ \frac{\partial^2 u}{\partial t \partial x}(x_i, t^n) &= -a \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta t, \Delta x),\end{aligned}$$

soit :

$$\frac{\partial^2 u}{\partial t^2}(x_i, t^n) = a^2 \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta t, \Delta x).$$

L'équation modifiée (1.64) peut donc être réécrite sous la forme suivante :

$$\frac{\partial u}{\partial t}(x_i, t^n) + a \frac{\partial u}{\partial x}(x_i, t^n) = \frac{a}{2}(\Delta x - a\Delta t) \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta t^2, \Delta x^2). \quad (1.65)$$

L'équation modifiée (1.65) est une équation d'advection-diffusion, de coefficient de diffusion égal à $\frac{a}{2}(\Delta x - a\Delta t)$. Le schéma numérique produit donc de la diffusion numérique, proportionnelle à la valeur du coefficient de diffusion. Remarquons que la valeur du pas de temps a une influence sur le coefficient de diffusion. Dans ce cas précis, le coefficient de diffusion vaut 0 lorsque la condition CFL est égale à 1 (c'est à dire lorsque $\frac{a\Delta t}{\Delta x} = 1$), et il tend vers $\frac{a\Delta x}{2}$ lorsque Δt est très petit.

Dans la suite, nous présentons de la même manière certaines méthodes implicites et leur caractéristiques en terme de stabilité et de précision.

1.4 Les méthodes d'intégration temporelle implicites

1.4.1 Présentation générale des méthodes implicites

Dans le cas des méthodes implicites, le terme α_0 de l'équation (1.33) est non nul : le second membre dépend de U^{n+1} . L'obtention de U^{n+1} nécessite de résoudre un système d'équation qui peut être non linéaire si D est non linéaire. Cela peut s'avérer coûteux en terme de temps de calcul mais on verra dans la suite que ce coût est compensé par le fait que ces méthodes ont de bonnes propriétés de stabilité qui leur permettent d'utiliser des pas de temps beaucoup plus grand que les méthodes explicites. Comme pour les méthodes explicites, on distingue les méthodes implicites multipas et les méthodes implicites à un pas, suivant qu'on utilise ou non les solutions antérieures à t^n pour le calcul de U^{n+1} .

1.4.2 Les méthodes implicites à un pas

1.4.2.1 La méthode Euler implicite

La méthode Euler implicite est la méthode implicite à un pas la plus simple. Le premier membre du système différentiel (1.18) est approché par différence fini et le second membre

est estimé à l'instant t^{n+1} :

$$\frac{U^{n+1} - U^n}{\Delta t} = -D(U^{n+1}). \quad (1.66)$$

Cette méthode, seulement d'ordre un, ne semble pas assez précise pour pouvoir être utilisée dans un contexte de simulations instationnaires. Par exemple, dans [45], la méthode Euler implicite est comparée à plusieurs autres méthodes implicites sur un cas standard d'écoulement instationnaire autour d'un cylindre. Les auteurs montrent que la méthode Euler implicite n'est pas compétitive par rapport aux autres méthodes en terme de précision.

1.4.2.2 La méthode de Crank-Nicolson

La méthode de Crank-Nicolson est une méthode implicite à un pas qui est plus précise que la méthode Euler implicite puisqu'elle est d'ordre deux. Cette méthode appliquée au système différentiel (1.18) s'écrit :

$$\frac{U^{n+1} - U^n}{\Delta t} = -\frac{1}{2} (D(U^{n+1}) + D(U^n)). \quad (1.67)$$

Grâce à sa plus grande précision, ce schéma semble être un candidat plus crédible que le schéma Euler implicite pour effectuer des simulations instationnaires. D'ailleurs ce schéma a déjà été utilisé dans ce contexte. Il a été utilisé pour simuler une interaction choc-couche limite instationnaire [46]. Dans [47], il a été utilisé pour réaliser une simulation LES, instationnaire, d'un écoulement turbulent autour d'un profil d'aile.

1.4.2.3 Les méthodes de Runge-Kutta implicites

La précision de ces méthodes peut largement dépasser celle de la méthode de Crank-Nicolson. Il existe par exemple des méthodes de Runge-Kutta implicites d'ordre 3 [48], d'ordre 4 [49] et même d'ordre 6 [49]. Pour présenter ces méthodes de la manière la plus claire possible, on les applique à la résolution de l'équation différentielle scalaire suivante :

$$\frac{du}{dt} = f(t, u). \quad (1.68)$$

Les méthodes de Runge-Kutta implicites utilisent différentes estimations numériques $u^{n,i}$ calculées entre t^n et t^{n+1} pour obtenir la solution numérique u^{n+1} . Dans [32], une définition des méthodes de Runge-Kutta implicites est donnée :

Définition 1.4. Soient a_{ij} , b_i , c_i ($1 \leq i, j \leq q$) des réels. Soit u^n la solution numérique au temps t^n de l'équation (1.68). Une méthode de Runge-Kutta implicite à q étapes appliquée entre t^n et $t^{n+1} = t^n + \Delta t$ pour la résolution de l'équation (1.68) s'écrit :

$$u^{n,i} = u^n + \Delta t \sum_{j=1}^q a_{ij} K^{n,j}, \quad (1.69)$$

$$t^{n,i} = t^n + c_i \Delta t, \quad (1.70)$$

$$K^{n,i} = f(u^{n,i}, t^{n,i}), \quad (1.71)$$

$$u^{n+1} = u^n + \Delta t \sum_{i=1}^q b_i K^{n,i}. \quad (1.72)$$

Les termes $u^{n,i}$ sont les différentes solutions intermédiaires de la méthode de Runge-Kutta entre t^n et t^{n+1} . Le terme $t^{n,i}$ représente le temps (compris entre t^n et $t^n + \Delta t$) auquel la solution intermédiaire $u^{n,i}$ est estimée. Le terme u^{n+1} est la solution au temps t^{n+1} .

Contrairement aux méthodes de Runge-Kutta explicites, l'indice j du symbole somme dans l'équation (1.69) ne s'arrête pas à $i - 1$: il y a une dépendance entre tous les termes K . On s'en convainc facilement en écrivant quelques termes K :

$$\left\{ \begin{array}{l} K^{n,1} = f(u^n + \Delta t(a_{11}K^{n,1} + \dots + a_{1q}K^{n,q}), t^n + c_1 \Delta t) \\ K^{n,2} = f(u^n + \Delta t(a_{21}K^{n,1} + \dots + a_{2q}K^{n,q}), t^n + c_2 \Delta t) \\ \cdot \\ \cdot \\ K^{n,q} = f(u^n + \Delta t(a_{q1}K^{n,1} + \dots + a_{qq}K^{n,q-1}), t^n + c_q \Delta t) \end{array} \right. \quad (1.73)$$

Remarque 1.2. La matrice A contenant les coefficients a_{ij} et les vecteurs b et c contenant respectivement les coefficients b_i et c_i peuvent être rangés dans un tableau de Butcher [32]. Un exemple de tableau de Butcher est donné sur le tableau 1.7 ci-dessous :

c	A		c_1	a_{11}	a_{12}	\cdot	\cdot	\cdot	\cdot	a_{1q}
		\longrightarrow	c_2	a_{21}	a_{22}	\cdot	\cdot	\cdot	\cdot	a_{2q}
			c_3	a_{31}	a_{32}	\cdot	\cdot	\cdot	\cdot	a_{3q}
			\cdot	\cdot	\cdot					\cdot
			\cdot	\cdot	\cdot					\cdot
			\cdot	\cdot	\cdot					\cdot
			c_q	a_{q1}	a_{q2}	\cdot	\cdot	\cdot	\cdot	$a_{q,q-1}$
				b_1	b_2	\cdot	\cdot	\cdot	b_{q-1}	b_q

TABLE 1.7 – Tableau de Butcher d'une méthode de Runge-Kutta implicite à q étapes

L'obtention de l'ensemble des termes K nécessite de résoudre le système (1.73) de taille q . Ce système est non linéaire si f est elle-même non linéaire. Si une telle méthode est

appliquée à un système différentielle de taille N (comme c'est le cas après l'étape de discrétisation spatiale d'une équation aux dérivées partielles), l'obtention de l'ensemble des termes K est au prix de la résolution d'un système de taille $N \times q$. L'implémentation de ce type de méthode peut donc être complexe [50].

La résolution du système peut néanmoins être rendue plus simple dans certains cas particulier. Si $a_{ij} = 0$ pour $j > i$, on est dans cas particulier des méthodes DIRK (Diagonal Implicit Runge-Kutta) [32]. Dans ce cas, au lieu d'être obtenus simultanément, les termes K sont obtenus les uns après les autres en résolvant q systèmes de taille N . Ceci rend l'implémentation plus simple. D'après [50], les méthodes DIRK sont les méthodes de Runge-Kutta implicites les plus largement utilisées. Elles ont par exemple été utilisées dans [51, 45, 52, 53] pour réaliser des simulations instationnaires d'écoulements fluides.

1.4.3 Une méthode implicite multipas : la méthode de Gear

L'utilisation de méthodes implicites multipas permet d'augmenter l'ordre sans augmenter la taille du système à résoudre, comme pour les méthodes de Runge-Kutta implicites. Cependant, ces méthodes requièrent certaines solutions à des instants antérieurs pour pouvoir mettre à jour la solution numérique. Ces solutions antérieures doivent donc être stockées. Par conséquent, ces méthodes sont généralement plus coûteuses en termes de mémoire que les méthodes à un pas. Nous avons fait le choix de ne présenter qu'une seule méthode implicite multipas : la méthode de Gear. En effet, cette méthode est probablement la méthode implicite la plus populaire. Les autres méthodes implicites multipas sont beaucoup moins utilisées.

Cette méthode a été initialement proposée par Gear [54]. Il s'agit d'une méthode multipas de la famille des méthodes BDF (Backward Differentiation)[32]. La méthode de Gear appliquée au système différentiel (1.18) s'écrit :

$$\frac{3U^{n+1} - 4U^n + U^{n-1}}{2\Delta t} = -D(U^{n+1}) \quad (1.74)$$

Cette méthode est d'ordre deux [55]. Elle a été très largement utilisée dans le contexte de simulations instationnaires d'écoulements turbulents. Dans [56, 57], elle a permis de réaliser des simulations instationnaires de type DNS. Elle a également permis de réaliser des simulations LES instationnaires [58, 55, 4].

1.4.4 Stabilité des méthodes implicites

1.4.4.1 Stabilité concernant l'intégration d'une équation différentielle

On étudie la stabilité linéaire de la méthode Euler implicite. Pour ce faire, on applique la méthode Euler implicite à l'équation test de Dahlquist (1.47), il vient :

$$u^{n+1} = u^n + \Delta t \lambda u^{n+1} \quad (1.75)$$

Une relation de récurrence entre u^{n+1} et u^n est donc obtenue :

$$u^{n+1} = \frac{1}{1 - \Delta t \lambda} u^n \quad (1.76)$$

En posant $z = \lambda \Delta t$, on obtient :

$$\begin{aligned} u^{n+1} &= \frac{1}{1 - z} u^n \\ &= R(z) u^n \end{aligned} \quad (1.77)$$

D'après la définition 1.3, le domaine de stabilité de la méthode Euler implicite est constitué des complexes z tels que $|1 - z| \geq 1$. Le domaine de stabilité de cette méthode est donc constitué du plan complexe privé du disque de centre 1 et de rayon 1. La méthode Euler implicite est dite A-stable.

Définition 1.5. [32] Une méthode d'intégration temporelle est dite A-stable si son domaine de stabilité S contient le demi-plan complexe de valeur réelle négative \mathbb{C}^- :

$$S \supset \mathbb{C}^- = \{z : \operatorname{Re}(z) \leq 0\}$$

Le caractère A-stable d'une méthode d'intégration temporelle est très utile puisqu'il assure que la méthode d'intégration temporelle est stable quelque soit le pas de temps Δt .

Précisons que toutes les méthodes implicites ne sont pas A-stable. Tout comme la méthode Euler implicite, la méthode de Crank-Nicolson et la méthode de Gear sont A-stable [32]. Par contre, la méthode BDF-3 ne l'est pas [32].

1.4.4.2 Stabilité concernant l'intégration d'une EDP discrétisée en espace

Pour cette étude stabilité de linéaire, on réutilise l'équation d'advection scalaire à coefficient a constant et positif :

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \quad (x, t) \in \mathbb{R} \times \mathbb{R}^+ \quad (1.78)$$

Pour la discrétisation spatiale, on réutilise également la méthode VF avec des flux décentrés :

$$\frac{du_i}{dt} = -\frac{1}{\Delta x} (au_i - au_{i-1}) \quad 1 \leq i \leq N \quad (1.79)$$

La méthode Euler implicite appliquée à l'équation (1.79) s'écrit :

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x} (au_i^{n+1} - au_{i-1}^{n+1}) \quad 1 \leq i \leq N \quad (1.80)$$

La méthode de Von Neumann appliquée au schéma (1.80) mène à la relation suivante :

$$\hat{u}^{n+1} e^{Jki\Delta x} = \hat{u}^n e^{Jki\Delta x} - \frac{a\Delta t}{\Delta x} (\hat{u}^{n+1} e^{Jki\Delta x} - \hat{u}^{n+1} e^{Jk(i-1)\Delta x}), \quad (1.81)$$

où \hat{u}^n est l'amplitude, $J = \sqrt{-1}$, et $k\Delta x$ est l'angle de phase associé au nombre d'onde k .

Soit, après quelques simplifications :

$$\hat{u}^{n+1} = \hat{u}^n - \hat{u}^{n+1} \frac{a\Delta t}{\Delta x} (1 - e^{-Jk\Delta x}) \quad (1.82)$$

Le facteur d'amplification h du schéma, défini par $h = \frac{\hat{u}^{n+1}}{\hat{u}^n}$, s'écrit :

$$h = \frac{1}{1 + \frac{a\Delta t}{\Delta x} (1 - e^{-Jk\Delta x})} \quad (1.83)$$

Un rapide calcul montre que $|h| \leq 1$ si :

$$(1 - \cos(k\Delta x)) \frac{2a\Delta t}{\Delta x} \left(\frac{a\Delta t}{\Delta x} + 1 \right) \geq 0 \quad (1.84)$$

Soit :

$$\frac{a\Delta t}{\Delta x} \geq 0 \quad (1.85)$$

On retrouve bien le caractère A-stable de la méthode : le pas de temps n'est théoriquement pas limité, contrairement aux méthodes explicites. Cependant, en pratique, le pas de temps des méthodes implicites A-stables est quand même limité par certains facteurs. D'une part, un pas de temps très grand a une influence néfaste sur la précision de la solution numérique obtenue. D'autre part, l'utilisation d'un pas de temps trop grand peut mener à une mauvaise convergence de l'algorithme de résolution du système non linéaire.

1.4.5 Précision des méthodes implicites

Dans ce paragraphe, nous montrons à l'aide d'un exemple théorique simple que les méthodes implicites sont susceptibles de dégrader la solution numérique dans le cas où un pas de temps trop grand est utilisé. Pour cette démonstration, on reprend le schéma numérique (1.80), que l'on écrit sous une forme équivalente :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{a}{\Delta x} (au_i^{n+1} - au_{i-1}^{n+1}) = 0 \quad (1.86)$$

Pour la démonstration, on détermine l'équation modifiée du schéma (1.86). Pour ce faire, on définit l'erreur locale τ_i^n , commise en injectant une certaine fonction $u(x_i, t^n)$ dans le schéma (1.86) :

$$\tau_i^n = \frac{u(x_i, t^n + \Delta t) - u(x_i, t^n)}{\Delta t} + \frac{a}{\Delta x} (u(x_i, t^n + \Delta t) - u(x_i - \Delta x, t^n + \Delta t)). \quad (1.87)$$

Dans cette situation, il est plus simple d'effectuer des développements de Taylor autour de $t^n + \Delta t$ et de x_i :

$$u(x_i, t^n) = u(x_i, t^n + \Delta t) - \Delta t \frac{\partial u}{\partial t}(x_i, t^n + \Delta t) + \frac{\Delta t^2}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t^n + \Delta t) + \mathcal{O}(\Delta t^3) \quad (1.88)$$

$$u(x_i - \Delta x, t^n) = u(x_i, t^n) - \Delta x \frac{\partial u}{\partial x}(x_i, t^n) + \frac{\Delta x^2}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t^n) + \mathcal{O}(\Delta x^3) \quad (1.89)$$

Les relations (1.88) et (1.89) sont injectées dans l'expression de l'erreur locale (1.87), il vient :

$$\begin{aligned} \tau_i^n &= \frac{\partial u}{\partial t}(x_i, t^n + \Delta t) + a \frac{\partial u}{\partial x}(x_i, t^n + \Delta t) - \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t^n + \Delta t) \\ &\quad - \frac{a \Delta x}{2} \frac{\partial^2 u}{\partial x^2}(x_i, t^n + \Delta t) + \mathcal{O}(\Delta t^2, \Delta x^2) \end{aligned} \quad (1.90)$$

Après quelques calculs, on peut montrer que l'équation modifiée vérifiée par le schéma (1.86) s'écrit :

$$\frac{\partial u}{\partial t}(x_i, t^n) + a \frac{\partial u}{\partial x}(x_i, t^n) = \frac{a}{2} (\Delta x + a \Delta t) \frac{\partial^2 u}{\partial x^2} + \mathcal{O}(\Delta t^2, \Delta x^2) \quad (1.91)$$

Là encore, on remarque que l'équation modifiée du schéma est une équation d'advection diffusion. Néanmoins, le coefficient de diffusion obtenu avec la méthode Euler implicite, $\frac{a}{2}(\Delta x + a\Delta t)$, est différent de celui obtenu avec la méthode Euler explicite, $\frac{a}{2}(\Delta x - a\Delta t)$. Si on considère que Δx et a sont fixes et que seul Δt varie, le coefficient de diffusion obtenu avec la méthode Euler explicite est **majoré** par $\frac{a\Delta x}{2}$, tandis que le coefficient de diffusion obtenu avec la méthode Euler implicite est **minoré** par $\frac{a\Delta x}{2}$. Ainsi, la méthode Euler implicite introduit plus de diffusion numérique que la méthode Euler explicite, ce qui la rend moins précise. Le pas de temps de la méthode Euler implicite, théoriquement non limité par une contrainte de stabilité numérique, doit quand même être restreint sous peine d'introduire une quantité importante de diffusion numérique et de fournir une solution très dégradée.

1.4.6 Algorithme de Newton pour la résolution du système non-linéaire

Les systèmes obtenus par les méthodes Euler implicite, Crank-Nicolson et Gear sont les systèmes (1.66), (1.67) et (1.74), dont l'inconnue est le vecteur U^{n+1} . Si le second membre D de ces systèmes est non linéaire (ce qui est souvent le cas), alors ces systèmes sont non linéaires. Ces systèmes peuvent s'écrire sous la forme d'un problème qui consiste à trouver U^{n+1} tel que :

$$H(U^{n+1}, U^n, \dots, U^{n-k}) = 0, \quad (1.92)$$

où H provient du schéma numérique et dépend :

- de l'inconnue U^{n+1} ,
- de U^n , le vecteur contenant les solutions numériques u_i^n ,
- éventuellement de k vecteurs solution aux k temps antérieurs (cas des méthodes multipas)

Le problème (1.92) doit être résolu à chaque pas de temps. Nous présentons la méthode de Newton, qui est largement utilisée dans le cas de simulations instationnaires. Elle a été utilisée pour réaliser des simulations DNS [56, 57] et LES [58, 55, 4].

La méthode de Newton est une méthode itérative : plusieurs estimations successives $U^{n,l}$ sont calculées jusqu'à obtenir la convergence vers l'inconnue U^{n+1} . La méthode de Newton est basée sur la linéarisation de l'équation (1.92) à l'ordre un autour de $U^{n,l}$:

$$H(U^{n,l+1}, U^n, \dots, U^{n-k}) \approx H(U^{n,l}, U^n, \dots, U^{n-k}) + J(U^{n,l})\Delta U^{n,l}, \quad (1.93)$$

où $J(U^{n,l})$ est la matrice Jacobienne de H en $U^{n,l}$, et $\Delta U^{n,l} = U^{n,l+1} - U^{n,l}$. Afin d'annuler H , $U^{n,l+1}$ est construite de la façon suivante :

$$\begin{cases} J(U^{n,l})\Delta U^{n,l} = -H(U^{n,l}, U^n, \dots, U^{n-k}) \\ U^{n,l+1} = U^{n,l} + \Delta U^{n,l} \end{cases} \quad (1.94)$$

Le second membre $-H(U^{n,l}, U^n, \dots, U^{n-k})$ est le résidu de Newton, qui doit tendre vers 0. La procédure (1.94) peut être effectuée pendant un nombre N d'itérations jusqu'à obtenir une estimation $U^{n,N}$ qui vérifie :

$$\frac{\|H(U^{n,N}, U^n, \dots, U^{n-k})\|_\infty}{\|H(U^{n,1}, U^n, \dots, U^{n-k})\|_\infty} \leq \epsilon, \quad (1.95)$$

où $\|\cdot\|_\infty$ représente la norme infinie et ϵ est un réel fixé par l'utilisateur. Avec ce critère, on impose le fait que le résidu de Newton soit divisé d'un certain facteur entre la sous-itération 1 et la sous-itération N . Lorsque la condition (1.95) est vérifiée, le processus de Newton est stoppé et on a :

$$U^{n+1} = U^{n,N}$$

Itérer jusqu'à la convergence complète du processus de Newton peut mener à des temps de calcul exorbitants, d'autant plus que ce processus de Newton est appelé à chaque pas de temps. D'où la présence de cette tolérance ϵ sur le niveau de convergence. Le niveau de convergence de l'algorithme de Newton est crucial. Dans [39, 4, 55], il est clairement montré qu'un relâchement du niveau de convergence de l'algorithme de Newton entraîne une dégradation de la précision des méthodes implicites.

Revenons rapidement sur la procédure de calcul de $U^{n,l+1}$ (1.94). Le calcul de la matrice Jacobienne exacte de H peut être souvent complexe (à cause de la complexité des flux numériques par exemple). Cette matrice est donc généralement calculée de manière approchée, par exemple par différences finies [58, 59]. De la même manière, la résolution du

système linéaire est généralement faite de manière approchée : on peut par exemple faire référence à la méthode LU-SGS proposée par Jameson et Yoon [60]. Cette méthode a été utilisée avec succès sur des calculs LES dans [4, 55].

1.4.7 Algorithme de Newton adaptatif

Le coût CPU d'une itération temporelle implicite est très élevé car il faut résoudre un système non linéaire de grande taille (à l'aide d'un algorithme de Newton). Un bon moyen de réduire le temps de calcul des méthodes implicites est donc de réduire le coût numérique d'une itération temporelle, en rendant l'algorithme de Newton plus efficace. La vitesse de convergence de l'algorithme de Newton est liée à la nature du phénomène physique simulé mais également au nombre CFL, dont nous rappelons l'expression :

$$CFL = \frac{U \times \Delta t}{\Delta x}, \quad (1.96)$$

où U est la vitesse de propagation du phénomène simulé, Δt est le pas de temps et Δx le pas d'espace. Plus le nombre CFL est grand, plus le nombre d'itérations N nécessaires à la convergence de l'algorithme de Newton est élevé [55]. Or, le nombre CFL admet souvent des valeurs très hétérogènes sur le maillage. D'une part, parce que la vitesse de propagation U du phénomène étudié varie dans l'espace. D'autre part, parce que la taille du pas d'espace Δx est elle aussi très souvent non-uniforme. En effet, il est très fréquent qu'en certaines zones de l'espace, le phénomène simulé développe de forts gradient : il est alors nécessaire de diminuer localement le pas d'espace afin de correctement "capter" ces brusques variations spatiales. C'est par exemple le cas des problèmes de mécanique des fluides présentant une paroi solide [16]. Proche de la paroi solide se développe la couche limite du fluide, une zone de forts gradients qui nécessite une réduction local du pas d'espace Δx .

Généralement, le nombre d'itérations N de l'algorithme de Newton est choisi de sorte que la convergence soit assurée dans les zones les plus "difficiles", c'est à dire les zones sur lesquelles le nombre CFL est le plus élevé. Les zones de plus petit CFL sont alors convergées plus que nécessaire. Un choix plus "économique" serait d'adapter le nombre d'itérations N en fonction du nombre CFL : un faible nombre d'itérations dans les zones de petit nombre CFL et un grand nombre d'itérations dans les zones de grand nombre CFL. Cette approche apparaît d'autant plus efficace que le nombre de cellules présentant un fort CFL est petit devant le nombre total de cellules. Un tel algorithme, que nous appellerons algorithme de Newton adaptatif, a été proposé par F. Daude [55, 61]. Dans l'algorithme proposé, une valeur du critère de convergence ϵ (équation(1.95)) est fixée ; le nombre d'itérations nécessaire pour atteindre ce critère varie sur le maillage (en fonction de la difficulté de convergence). L'algorithme de Newton adaptatif a permis de réaliser avec succès une simulation LES d'un écoulement de cavité [55, 61], et s'est révélé 10 fois plus rapide que l'algorithme de Newton "classique" avec un nombre d'itérations uniforme [55]. En terme de précision, les résultats numérique obtenus avec l'algorithme de Newton adaptatif sont en bon accord avec les données expérimentales [55]. Alferez [4] a également utilisé avec succès l'algorithme de Newton adaptatif pour réaliser une simulation LES de l'écoulement autour d'une pale d'hélicoptère en décrochage. Néanmoins, l'accélération obtenue par Alferez [4]

avec l'algorithme de Newton adaptatif s'est révélée moins spectaculaire : elle n'est "que" d'un facteur 2 par rapport à l'algorithme de Newton non adaptatif.

1.5 Bilan comparatif des méthodes implicites et explicites

Ce bilan comparatif est dressé à partir des trois critères définis précédemment, à savoir le critère de **précision**, le critère de **coût CPU** et le critère de **compatibilité avec les techniques HPC**.

Concernant le critère de **précision**, l'avantage est donné aux méthodes explicites. Comme on l'a vu dans l'exemple de l'équation d'advection scalaire, la méthode explicite est intrinsèquement moins diffusive que la méthode implicite. De plus, la contrainte de stabilité qui porte sur le pas de temps limite "naturellement" la diffusion numérique des méthodes explicites. Dans [62, 39, 63, 55], la précision des méthodes implicites proposées est jugée par rapport à des solutions de références obtenues à l'aide de méthodes explicites. Ceci atteste de la bonne réputation des méthodes explicites en matière de précision dans le cas de simulations instationnaires. De plus, l'utilisation de méthodes implicites dans le cas de simulations de type LES ou DNS peut être critiquée. En effet, la philosophie des simulations DNS (LES) est de résoudre l'ensemble (une partie) des échelles turbulentes. L'utilisation d'une méthode implicite, avec un pas de temps élevé, risque de filtrer certaines échelles turbulentes, ce qui n'est pas en accord avec la philosophie de ces simulations (surtout dans le cas de la DNS). Choi et Moin [64] ont constaté que l'utilisation de pas de temps trop grands dans le cas d'une DNS d'un canal plan entraînait une perte de certaines échelles turbulentes. Malgré ces défauts, les méthodes implicites restent quand même très utilisées et fournissent des solutions numériques correctes et largement exploitables dans le cas de simulations LES [55, 4, 62].

Concernant le **critère du coût CPU**, l'avantage est donné aux méthodes implicites. Ce critère constitue d'ailleurs le principal intérêt des méthodes implicites. Comme on l'a montré précédemment, ces dernières admettent des pas de temps plus grands que les méthodes explicites. Ceci leur permet d'effectuer moins d'itérations temporelles pour atteindre un temps physique donné. Cet avantage est nuancé par le fait qu'une itération temporelle implicite est beaucoup plus chère qu'une itération explicite, à cause de la résolution du système non linéaire. Mais, en général, les méthodes implicites sont plus rapides que les méthodes explicites. Par exemple, dans [39], les auteurs montrent que leur méthode implicite est six fois plus rapide qu'une méthode explicite de type RK3, pour la simulation d'une couche limite turbulente. Lessani *et al* [65] ont montré que leur méthode implicite était plus rapide d'un facteur quatre que la méthode explicite qu'ils utilisent pour simuler des cas d'écoulement turbulent dans un canal et autour d'un cylindre.

Enfin, concernant le critère de **compatibilité avec les techniques HPC**, l'avantage est donné aux méthodes explicites. En effet, les méthodes implicites nécessitent de résoudre un grand système linéaire et il est difficile de paralléliser l'algorithme de résolution de ce grand système linéaire [66, 63]. En général, le système linéaire est décomposé en plusieurs

sous-systèmes résolus simultanément (en parallèle). Mais cela engendre des imprécisions numériques dues à la difficulté de traiter des conditions aux limites entre chaque sous-système [66]. De plus, l'efficacité de la procédure est limitée par la difficulté d'équilibrer la charge de travail entre les différents sous-systèmes (et donc entre les différents coeurs de calcul) [66]. Dans le cas des méthodes explicites, la parallélisation est beaucoup plus simple et efficace [63].

En conclusion, les méthodes explicites et implicites présentent des avantages et des inconvénients par rapport aux trois critères énoncés. Ainsi, au cours des dernières décennies, les auteurs ont tenté d'apporter des modifications à ces méthodes pour les rendre plus performantes. On peut par exemple citer l'utilisation de techniques de contrôle du pas de temps. L'objectif de ces techniques est d'améliorer la précision, la stabilité ou encore la convergence de l'algorithme de Newton (dans le cas des méthodes implicites) en contrôlant le pas de temps au cours de la simulation. Il existe une large gamme de critères pour le contrôle du pas de temps. Par exemple, le pas de temps peut être contrôlé au cours de la simulation en se basant sur un critère d'erreur numérique. L'erreur numérique commise par le schéma d'intégration temporelle est en général calculée en comparant deux solutions numériques d'ordre différent [32, 67]. Si cette erreur dépasse une certaine tolérance le pas de temps est réduit, dans le cas contraire, le pas de temps peut être augmenté. Par exemple, une méthode de contrôle du pas de temps a été appliquée à des méthodes de Runge-Kutta implicites dans [68]. Les auteurs montrent que cette technique a permis d'améliorer la précision de ces méthodes tout en assurant une meilleure convergence de l'algorithme de résolution du système non linéaire. Bien qu'intéressante, cette technique de contrôle du pas de temps ne suffit pas à gommer complètement les défauts des méthodes explicites et implicites.

Aussi, de nouvelles méthodes d'intégration temporelle comme les **méthodes IMplicite/EXplicite (IMEX)** et les **méthodes explicites à pas de temps variable en espace** ont été mises en place pour tenter de dépasser les limitations des méthodes implicites et explicites. Ces méthodes sont présentées dans la suite de ce chapitre.

1.6 Les méthodes IMplicite/EXplicite (IMEX)

Cette approche combine méthodes d'intégration implicites et explicites pour tenter de tirer profit des avantages des deux méthodes.

Historiquement, les schémas IMEX ont été conçus pour la résolution d'une loi de conservation où la dérivée spatiale porte sur deux termes F et G :

$$\frac{\partial u}{\partial t}(x, t) + \frac{\partial}{\partial x} (F(u(x, t)) + G(u(x, t))) = 0 \quad (1.97)$$

Ce type de loi de conservation se rencontre assez fréquemment. On peut par exemple citer le cas d'une équation d'advection-diffusion à coefficients constants. Pour une telle équation, les termes F et G s'écrivent :

$$F = au(x, t) \quad \text{et} \quad G = d \frac{\partial u}{\partial x}(x, t),$$

où a et d sont des réels, supposés constants, représentant respectivement la vitesse d'advection et le coefficient de diffusion. La discrétisation spatiale de l'équation (1.97) mène à un système d'équations différentielles :

$$\frac{dU}{dt} = -D_F(U) - D_G(U) \tag{1.98}$$

où $U = [u_1, u_2, \dots, u_i, \dots, u_N]$ est le vecteur contenant les solutions numériques u_i sur chaque cellule i . Les termes D_F et D_G sont les bilans de flux associés aux termes F et G . Les termes D_F et D_G peuvent imposer des conditions de stabilité très différentes au pas de temps de la méthode d'intégration temporelle [43]. La condition CFL du terme d'advection impose une limite du type :

$$\Delta t \leq \frac{\Delta x}{a}$$

La condition CFL du terme de diffusion impose une limite différente, du type :

$$\Delta t \leq \frac{\Delta x^2}{d}$$

Si a et d sont du même ordre de grandeur, le pas de temps est beaucoup plus restreint par la condition CFL du terme de diffusion que par la condition CFL du terme d'advection. On dit que le terme de diffusion est plus "raide" que le terme d'advection [43]. Pour s'affranchir de la condition de stabilité très restrictive imposée par le terme de diffusion, on peut envisager d'intégrer ce terme avec une méthode implicite. Le terme d'advection, moins raide, peut être intégré avec une méthode explicite. Une telle méthode est appelée méthode IMplicite/EXplicite (IMEX).

Nous avons vu jusqu'à maintenant que le choix de la méthode d'intégration temporelle (implicite ou explicite) était piloté par la raideur des opérateurs D_F et D_G de l'équation discrétisée en espace. Pour qualifier cette approche, nous dirons que la méthode IMEX est basée sur une **décomposition d'opérateur**. Sur un maillage "réel", le pas d'espace Δx est variable, ce qui implique que la raideur a également une dépendance spatiale. Il peut alors être utile de "rassembler" les conditions de stabilité imposées par chaque opérateur en définissant Δt tel que :

$$\Delta t \leq \min\left(\frac{\Delta x}{a}, \frac{\Delta x^2}{d}\right),$$

et de considérer que la raideur varie spatialement (sur le maillage). Les zones du maillage où le pas d'espace Δx est petit sont très contraignantes en terme de stabilité. Par conséquent, une méthode d'intégration temporelle implicite est appliquée dans ces zones. Une

méthode d'intégration temporelle explicite est utilisée dans les zones de pas d'espace plus grand. Pour qualifier cette approche, nous dirons que la méthode IMEX est basée sur une **décomposition spatiale**. La méthode IMEX basée sur une décomposition spatiale semble beaucoup mieux adaptée à la simulation d'écoulements de fluides turbulents. En effet, un certain nombre de simulations de type LES [12, 13, 14, 15] ou RANS/LES [11] ont été réalisées avec succès grâce à cette approche. Dans [69], des comparaisons en terme de coût CPU entre un schéma IMEX (basé sur une décomposition spatiale) et un schéma purement explicite sont réalisées sur plusieurs cas d'écoulements 1D et 2D. La performance du schéma IMEX par rapport au schéma purement explicite varie selon les cas. Les auteurs montrent que la méthode IMEX est d'autant plus performante que :

- le ratio entre la taille de la plus grande cellule du maillage et la taille de la plus petite cellule du maillage est grand,
- le nombre de petites cellules est faible devant le nombre de grandes cellules.

Ils observent, dans le cas le plus favorable, une réduction du temps de calcul d'un facteur en faveur du schéma IMEX.

Au vu de la littérature, la méthode IMEX basée sur une décomposition spatiale semble donc être une méthode intéressante pour la simulation instationnaire. Elle semble assez précise et permet de réduire le coût CPU de manière efficace [69]. Néanmoins, nous craignons que la compatibilité de la méthode IMEX avec les techniques HPC soit limitée par la phase d'intégration implicite. C'est pourquoi nous avons fait le choix de nous tourner vers des méthodes purement explicites qui utilisent des pas de temps variables sur le maillage pour réduire leur temps de calcul.

1.7 Les méthodes explicites à pas de temps variable en espace

Comme nous l'avons évoqué à plusieurs reprises dans ce mémoire, le pas de temps des méthodes explicites est sévèrement limité par la condition CFL. Elle est définie en chaque cellule et dépend notamment de la taille de la cellule et de la vitesse de propagation du phénomène au niveau de la cellule. Cette condition CFL est bien souvent non-uniforme sur le domaine de calcul. D'une part, parce que la vitesse de propagation du phénomène physique étudié varie dans l'espace et dans le temps. D'autre part, parce que la taille des cellules varie sur le maillage, pour répondre à des contraintes de précision (la taille des cellules est réduite dans les zones où le phénomène simulé présente de forts gradients).

Pour les méthodes explicites présentées jusqu'ici, le pas de temps est identique sur l'ensemble du domaine de calcul. Il est choisi de manière à respecter la condition CFL la plus restrictive du maillage. Certes, ce pas de temps assure la stabilité de la méthode sur l'ensemble du maillage, mais il n'est pas optimal : il pourrait être plus grand sur les cellules qui présentent une condition CFL moins restrictive. Cette situation est particulièrement défavorable lorsqu'un petit nombre de cellules présente une condition CFL beaucoup plus sévère que le reste des cellules du maillage. Ainsi, à cause d'une minorité de cellules, le pas

de temps est beaucoup plus petit que nécessaire sur la majorité des cellules. Une majorité de cellules est mise à jour beaucoup plus fréquemment que nécessaire : beaucoup de calculs "inutiles" sont donc effectués, ce qui affecte considérablement les temps de calcul.

Pour résoudre ce problème, des méthodes explicites à pas de temps variable en espace ont été imaginées par différents auteurs depuis le début des années 1980. Ces méthodes reposent sur une adaptation spatiale du pas de temps : des "petits" pas de temps sont utilisés sur les cellules présentant une condition CFL restrictive et des "grands" pas de temps sont appliqués aux cellules qui présentent une condition CFL moins sévère. Ainsi, chaque cellule est intégrée avec un pas de temps qui est plus optimal que dans le cas d'une méthode explicite à pas de temps uniforme. Le nombre de calculs "inutiles" est considérablement réduit, par conséquent ces méthodes sont généralement plus rapides que les méthodes explicites classiques (à pas de temps uniforme).

L'étude de la littérature sur met en lumière deux types de méthodes à pas de temps variable en espace.

- Il y a d'une part **les méthode explicites à pas de temps local asynchrones** qui se caractérisent par l'utilisation d'un pas de temps différent pour chaque cellule.
- Il y a d'autre part **les méthodes explicites à pas de temps local**, également appelées **méthodes multirate** dans la littérature. Contrairement aux méthode à pas de temps local asynchrones, les pas de temps locaux s'appliquent à des groupes de cellules.

Quelque soit le type de méthode choisi, les auteurs se heurtent à deux difficultés majeures :

- mettre en place une méthode d'ordre élevé en temps,
- assurer la conservativité de la méthode.

1.7.1 Méthodes explicites à pas de temps local asynchrones

Cette première stratégie consiste à intégrer chaque cellule du maillage avec son pas de temps maximal. Vu que chaque cellule présente une condition CFL différente, il est possible d'avoir autant de pas de temps différents qu'il y a de cellules dans le domaine de calcul. De plus, Il n'existe aucun lien entre les valeurs des différents pas de temps. Ceci empêche la synchronisation temporelle de la solution sur les différentes cellules, d'où l'appellation de "schéma asynchrone" pour ce type de méthode. Une synchronisation temporelle de la solution sur les différentes cellules est uniquement réalisée en fin de simulation par l'intermédiaire d'extrapolations. Omelchenko et Karimabadi [18] ont proposé un schéma basé sur cette stratégie.

Nous présentons brièvement ce schéma ; pour plus de précision le lecteur peut se référer à [18]. On considère le maillage de la figure 1.7 constitué de cinq cellules, et on se place à l'instant initial de la simulation. Dans chaque cellule, la solution est estimée au temps $t_{init} = 0$. Chaque cellule i est dotée d'un temps "cible" de mise à jour, t_i , calculé grâce au pas de temps local à la cellule : $t_i = t_{init} + \Delta t_i$. Le schéma à pas de temps local asynchrone d'Omelchenko et Karimabadi peut être présenté en quatre étapes, qui sont décrites ci-dessous.

1. Un algorithme recherche la cellule ayant le temps de mise à jour t_i le plus faible. Dans la figure 1.7, il s'agit de la cellule 3.
2. La solution sur cette cellule est mise à jour au temps t_3 en utilisant son pas de temps local Δt_3 , comme le montre la figure 1.8. Un trait plein au niveau du temps t_3 est utilisé pour signifier que la solution sur cette cellule est maintenant estimée à cet instant.
3. Sur les cellules faisant partie du stencil de la cellule 3 (ici les cellules 1 et 2) la solution est également mise à jour au temps t_3 (figure 1.9). Ainsi, l'ensemble du stencil de la cellule 3 est estimé au même instant : le calcul des flux aux interfaces entre les cellules 1 et 2 et entre les cellules 3 et 4 peut être effectué [18].
4. Enfin, un nouveau temps cible de mise à jour t_3 est calculé pour la cellule 3 (figure 1.10). L'algorithme se remet à chercher la cellule ayant le temps de mise à jour le plus faible, ici il s'agit de la cellule 1.

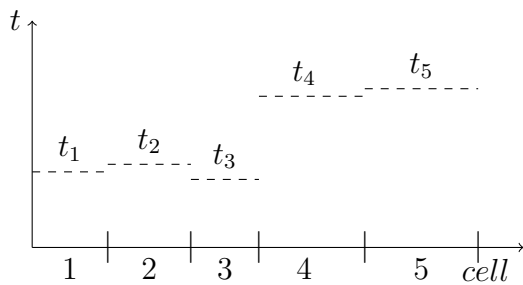


FIGURE 1.7 – Les 5 cellules du maillage ont des temps de mise à jour différents (en pointillés), basés sur leur pas de temps local

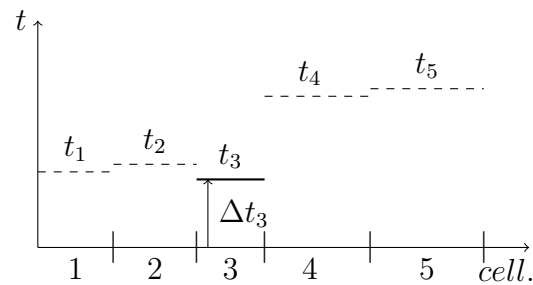


FIGURE 1.8 – La solution numérique sur la cellule 3 est maintenant mise à jour au temps t_3

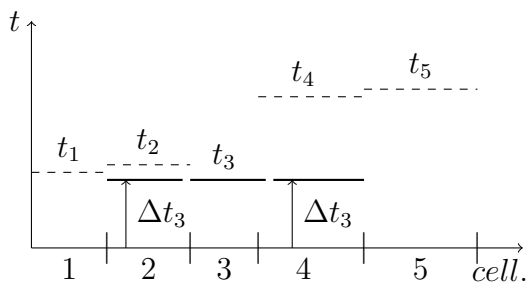


FIGURE 1.9 – La solution numérique sur les cellules 2 et 3 est mise à jour au temps t_3

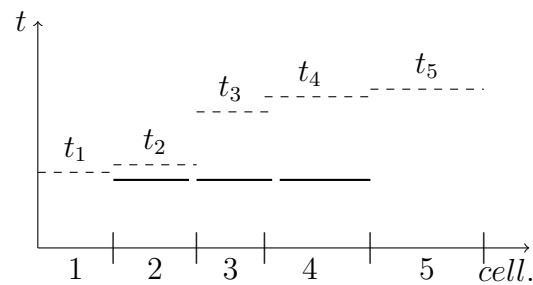


FIGURE 1.10 – Un nouveau temps cible de mise à jour, t_3 , est calculé pour la cellule 3

Le schéma proposé est conservatif, mais il est seulement d'ordre un en temps, car il est basé sur la méthode Euler explicite (RK1). Thomas Unfer [19] a développé un schéma asynchrone similaire, d'ordre un et conservatif. Mais contrairement à Omelchenko et Karimabadi qui définissent un pas de temps par cellule, Unfer définit un pas de temps par flux

d'interface [19].

Bien que très bien adapté aux maillages fortement hétérogènes que l'on peut rencontrer dans les simulations d'écoulements turbulents, ce schéma présente certains inconvénients. Tout d'abord, l'ordre un en temps est très limitant en terme de stabilité et de précision pour des calculs d'écoulements turbulents. Des schémas asynchrones d'ordre deux en temps existent [70, 71], mais l'ordre trois nous paraît meilleur en terme de stabilité et donc de performance pour nos applications. Or, à notre connaissance, aucun schéma à pas de temps local asynchrone d'ordre trois n'existe dans la littérature. Ces schémas présentent une certaine complexité algorithmique, et nous pensons que cela peut être un frein à leur développement à l'ordre trois.

D'autre part, nous doutons de la compatibilité de ce type de schéma avec certaines techniques HPC telles que les techniques d'optimisation des accès mémoire ou les techniques de vectorisation. En effet, la structure même de l'algorithme fait que le parcours des mailles en espace est aléatoire [19] : deux cellules successivement mises à jour peuvent très bien se trouver éloignées spatialement. Il y a donc un fort risque de ne pas travailler sur des données contiguës en mémoire, or les techniques d'optimisation des accès mémoire et de vectorisation requièrent de travailler sur des données contiguës en mémoire.

Les méthodes explicites à pas de temps local asynchrones peuvent être intéressantes en ce qui concerne le critère de la réduction du coût CPU grâce à la variation spatiale du pas de temps, qui permet d'optimiser la CFL en chaque maille. Cependant, à cause de la mauvaise compatibilité de ces méthodes avec les techniques HPC nous avons choisi de ne pas les retenir pour la thèse. Nous avons préféré nous tourner vers un autre type de méthodes explicites à pas de temps local.

1.7.2 Méthodes explicites à pas de temps local

Ces méthodes sont également désignées par le terme "méthodes multirate" dans la littérature. Kværnø [72] donne les caractéristiques des méthodes multirate :

- Les différents pas de temps sont appliqués à des groupes de cellules
- Les différents pas de temps sont des fractions du plus grand pas de temps Δt
- Il y a synchronisation de l'ensemble des pas de temps à la fin de chaque grand pas de temps Δt (d'où l'emploi du terme synchrone dans le titre pour désigner ce type de méthodes).

Dans la suite de ce mémoire, nous désignons par "méthodes explicites à pas de temps local" ou "méthodes multirate" l'ensemble des méthodes d'intégration temporelle qui satisfont ces trois conditions.

Dans la plupart des schémas à pas de temps local de la littérature, une méthode d'intégration temporelle de base est choisie. Cette méthode, que nous notons B et de pas de temps Δt , est utilisée pour intégrer les cellules présentant la condition CFL la moins restrictive. Les cellules présentant une condition CFL k fois plus restrictive (où k est un entier) sont intégrées avec la même méthode de base de pas de temps $\Delta t/k$, que nous notons B/k . Une telle approche "télescopique" est illustrée par la figure 1.11, où l'on considère les cas

particuliers $k = 2, 4$.

Le fait de conserver la même méthode de base en adaptant son pas de temps en fonction de la CFL locale est probablement l'approche la plus simple au niveau théorique et algorithmique pour généraliser le schéma à un nombre quelconque de pas de temps. Dans la suite, **nous dirons qu'un schéma à pas de temps local est basé sur une méthode d'intégration temporelle lorsque cette dernière constitue sa méthode de base.**

Dans la suite de cette partie, une revue bibliographique des schémas à pas de temps local de la littérature est effectuée. Les schémas sont classés en fonction de leur ordre de précision temporel. En effet, un des inconvénients des schémas à pas de temps local est que l'augmentation de leur ordre temporel peut s'avérer délicate, à cause de la gestion de l'interface entre deux zones de pas de temps différents. Ce critère constitue donc un bon critère pour différencier les schémas. Une attention particulière est également portée au caractère conservatif (ou non) des schémas.

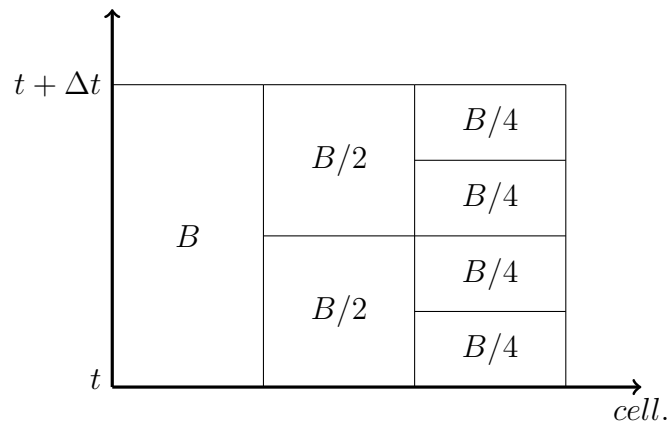


FIGURE 1.11 – Illustration de l'approche utilisée dans la majorité des schémas à pas de temps local de la littérature. L'axe des abscisses "cell." représente les différentes cellules d'un maillage 1D.

1.7.2.1 Schémas à pas de temps local d'ordre un.

Dans les années 1980, Osher et Sander [73] sont parmi les premiers à proposer un schéma explicite à pas de temps local. Leur schéma est basé sur la méthode Euler explicite et est donc d'ordre un en temps. Il est également conservatif.

Un algorithme de ce schéma est présenté pour souligner les différences avec les schémas asynchrones. Le maillage utilisé pour cet algorithme est représenté sur la figure 1.12. On se place à l'instant initial t^0 de la simulation, et on suppose que le schéma numérique utilisé a un stencil de trois. Le pas de temps $\frac{\Delta t}{2}$ est utilisé sur les cellules 1 à 3 : la solution numérique sur ces cellules doit donc être mise à jour aux temps $t^0 + \frac{\Delta t}{2}$ et $t^0 + \Delta t$. Le pas de temps Δt est utilisé sur les cellules 4 et 5 : la solution numérique sur ces cellules est mise à jour uniquement au temps $t^0 + \Delta t$. L'algorithme du schéma d'Osher et Sanders peut être décomposé en trois étapes, décrites ci-dessous.

1. Sur les cellules 1 à 3, la solution numérique est mise à jour au temps $t^0 + \frac{\Delta t}{2}$ (figure 1.13). Le stencil de la solution numérique sur la cellule 3 est représenté par des flèches en pointillés rouges.
2. Sur les cellules 1 à 3, la solution est mise à jour au temps $t^0 + \Delta t$ (figure 1.14). Remarquons le stencil particulier utilisé pour mettre à jour la solution sur la cellule 3. Ce stencil "mélange" des solutions numériques qui ne sont pas estimées au même instant : la solution sur la cellule 4, estimée au temps t^0 , et les solutions sur les cellules 2 et 3, estimées au temps $t^0 + \Delta t/2$.
3. Enfin, sur les cellules 4 et 5 la solution est mise à jour au temps $t^0 + \Delta t$ (figure 1.15). Le stencil de la solution numérique sur la cellule 4 (représenté par les flèches en pointillés rouges) est lui aussi particulier puisqu'il mélange des solutions estimées à des temps différents.

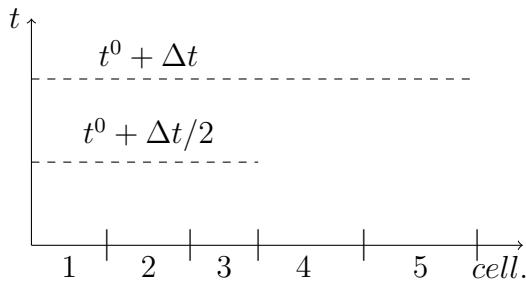


FIGURE 1.12 – Les cellules 1 à 3 sont intégrées avec un pas de temps $\Delta t/2$, les cellules 4 et 5 avec un pas de temps Δt

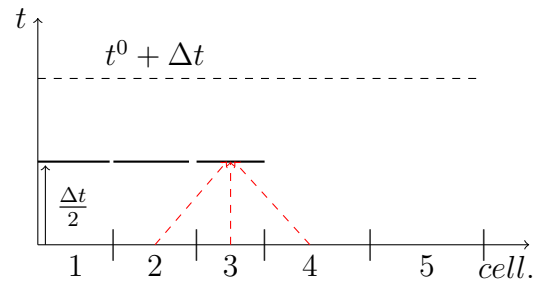


FIGURE 1.13 – Sur les cellule 1 à 3, la solution numérique est mise à jour au temps $t^0 + \Delta t/2$. Le stencil de la cellule 3 est représenté par les flèches en pointillés rouges

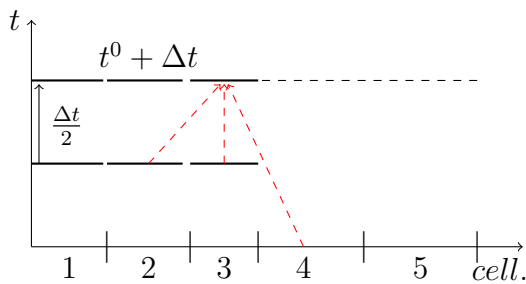


FIGURE 1.14 – Sur les cellules 1 à 3, la solution numérique est mise à jour au temps $t^0 + \Delta t$.

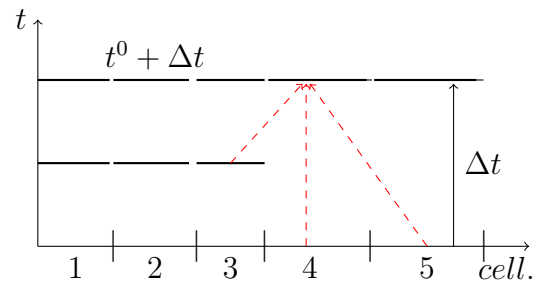


FIGURE 1.15 – Sur les cellule 4 et 5, la solution est mise à jour au temps $t^0 + \Delta t$.

Cet algorithme simple illustre le problème majeur d'un schéma à pas de temps local : **la gestion de l'interface entre zones de pas de temps différents**. Comme nous le verrons par la suite, il y a deux stratégies possibles pour la gestion de cette interface :

- On peut "mélanger" des valeurs estimées à des instants différents, comme le font Osher et Sanders. Dans la littérature, les schémas à pas de temps local qui utilisent

- cette stratégie sont qualifiés de schémas localement inconsistants [74]. Les schémas basés sur cette stratégie sont en général conservatifs [73, 16] mais d'ordre peu élevé.
- On peut aussi faire en sorte que l'ensemble du stencil soit évalué au même instant, en calculant des estimations aux temps intermédiaires manquants. Par exemple, dans la figure 1.14, on pourrait envisager le calcul d'une estimation au temps $t^0 + \Delta t/2$ sur la cellule 4, pour "alimenter" le stencil de la solution sur la cellule 3. Les schémas basés sur cette stratégie sont en général d'ordre plus élevé que les schémas basés sur la stratégie précédente, mais ils ne sont en général pas conservatifs [21, 17].

On remarque également que l'algorithme du schéma à pas de temps local d'Osher et Sanders a une structure plus "compacte" que l'algorithme du schéma asynchrone. Les cellules partageant un même pas de temps sont contiguës en espace, et donc en mémoire. Les schémas à pas de temps local semblent donc mieux adaptés aux techniques HPC que les schémas asynchrones.

Tan *et al* [23] ont développé un schéma à pas de temps local d'ordre un sur un maillage adaptatif (le maillage évolue au cours du temps en fonction de l'évolution temporelle de la solution numérique). Contrairement à Osher et Sanders, des interpolations sont effectuées aux interfaces pour que les stencils des cellules aux interfaces ne contiennent que des cellules estimées aux mêmes instants.

Les schémas d'Osher et Sanders et de Tan *et al*, seulement d'ordre un, manquent de précision pour effectuer des simulations instationnaires. Pour remédier à ce problème, différents auteurs ont développé des schémas à pas de local plus précis, à l'ordre deux, à l'ordre trois et même très récemment, à l'ordre quatre.

1.7.2.2 Schémas à pas de temps local d'ordre deux.

Au début des années 2000, Dawson et Kirby [20] et Tang et Warnecke [21] proposent des schémas à pas de temps local basés sur une méthode RK2. Le schéma de Dawson et Kirby [20] est conservatif mais il n'est pas d'ordre deux à cause du traitement réalisé aux interfaces entre zones de pas de temps différent. Le schéma de Tang et Warnecke est quant à lui bel et bien d'ordre deux, les auteurs en fournissent une justification théorique et numérique. Cependant, il n'est pas conservatif au niveau des interfaces entre pas de temps différents. En 2007, Constantinescu et Sandu [16] utilisent la théorie des méthodes Runge-Kutta Partitionnées (RKP) [32] pour construire le premier schéma à pas de temps local d'ordre deux en temps et conservatif basé sur une méthode RK2. La théorie des méthodes RKP est très utile pour mettre en place des schémas à pas de temps local. Elle est présentée en détails dans le chapitre 2 de ce mémoire. Dans [24], les auteurs proposent de nouveaux schémas à pas de temps local en appliquant la stratégie de Constantinescu et Sandu à des méthodes de Runge-Kutta d'ordre plus élevé. Mais l'ordre temporel des schémas obtenus ne peut excéder deux.

Les schémas explicites à pas de temps local mentionnés jusque là sont basés sur des méthodes de Runge-Kutta. Des schémas explicites à pas de temps local basés sur les méthodes explicites d'Adams-Bashforth existent aussi. Constantinescu et Sandu [22] ont développé un schéma à pas de temps local basé sur une méthode d'Adams-Bashforth d'ordre deux. Ce schéma présente les mêmes propriétés que leur schéma à pas de temps local basé sur la

méthode de Runge-Kutta 2 : il est conservatif mais, au maximum, d'ordre deux. Récemment, Mikida *et al* [75] ont développé un schéma à pas de temps local d'ordre 2 basé sur une méthode d'Adams-Bashforth et dédié à la simulation sur maillages recouvrants.

1.7.2.3 Schémas à pas de temps local d'ordre trois.

La littérature concernant les schémas à pas de temps local d'ordre trois est plus réduite. A notre connaissance, Kværnø [72] et Schlegel [17] ont chacun proposé un schéma à pas de temps local d'ordre trois. Dans [17], le schéma de Schlegel est utilisé pour résoudre une équation d'advection 1D. Sény a utilisé ce schéma pour réaliser des simulations plus complexes : l'hydrodynamique au niveau de la grande barrière de Corail [24], la propagation d'un tsunami [76]. Précisons que ce schéma n'est pas conservatif.

Le schéma développé par Kværnø [72] est utilisé pour résoudre les équations différentielles issues des circuits électriques.

1.7.2.4 Schémas à pas de temps local d'ordre quatre.

Dans des travaux concomitants à cette thèse, Sexton *et al* [77] et Sandu [78] (2019) ont proposé des schémas à pas de temps local d'ordre quatre. Ces schémas s'appuient sur le travail de Knoth et Wolke initialement dédié aux schémas IMEX [79], et sont basés sur une méthode de Runge-Kutta d'ordre quatre. Les schémas développés sont utilisés pour la résolution de systèmes d'équations différentielles.

1.7.2.5 Bilan sur les schémas à pas de temps local.

Les applications des schémas à pas de temps local mentionnés précédemment se limitent en général à la résolution de problèmes de type académiques. Il s'agit par exemple de la résolution de l'équation d'advection ou de l'équation de Burgers [20, 21, 16, 17, 22, 23], des équations d'Euler en 2D [21, 75] ou encore de la résolutions de systèmes d'équations différentielles [78, 77, 72]. Quelques simulations "de plus grande envergure" ont été réalisées à l'aide de schémas à pas de temps local, mais la littérature est peu fournie. On peut notamment citer Sény *et al* [24], qui ont réalisé une simulation de l'hydrodynamique au niveau de la grande barrière de corail à l'aide de schémas à pas de temps local. On peut également citer Itam [25] qui a réalisé une simulation RANS/LES à l'aide d'un schéma explicite à deux pas de temps. C'est pourquoi, dans ce travail de thèse, nous nous sommes attachés à montrer que nos schémas à pas de temps local peuvent tout à fait être utilisés pour réaliser des simulations plus complexes, de type industriel (écoulements 3D turbulents, LES).

Les schémas à pas de temps local semblent en bon accord avec les trois critères que doivent satisfaire les méthodes d'intégrations temporelle pour la simulation instationnaire (voir la section 1.2.2).

Concernant le critère de **précision**, il est vrai que la montée en ordre des schémas à pas de temps est délicate, mais elle reste possible comme le montre l'étude de la littérature. Par conséquent, on peut considérer que les schémas à pas de temps local satisfont le critère de précision, nécessaire à la réalisation de simulations instationnaires.

Concernant le critère du **coût CPU**, les schémas à pas de temps local ont un avantage indéniable sur les méthodes explicites classiques, à pas de temps uniforme. En effet, tous les articles évoqués ci-dessus rapportent que les schémas à pas de temps local permettent de réduire les temps de calcul par rapport à un schéma explicite à pas de temps global (uniforme). Par conséquent, nous considérons que les schémas à pas de temps local satisfont le critère du **coût CPU**.

Enfin, l'aspect explicite des schémas à pas de temps local nous semble être un avantage pour satisfaire le critère de **compatibilité avec les techniques HPC**. D'ailleurs, Sény *et al* [80] montrent un bon comportement de leur schéma à pas de temps local dans un contexte de parallélisation MPI.

Il semble donc que les schémas à pas de temps local soient des méthodes d'intégration temporelle bien adaptées à la simulation instationnaire de fluides turbulents. C'est la raison pour laquelle nous avons choisi de développer ce type de méthode au cours de cette thèse.

Chapitre 2

Schémas à pas de temps local de Constantinescu et Sandu

Nous décrivons dans cette partie les schémas à pas de temps local de Constantinescu et Sandu [16]. Nous pensons que ces schémas constituent des points de comparaison intéressants pour les schémas à pas de temps local que nous proposerons dans le chapitre 3. En effet, les schémas de Constantinescu et Sandu rassemblent plusieurs avantages : ils sont conservatifs, d'ordre deux et ils fonctionnent avec différentes méthodes de base [24]. Cette dernière caractéristique permet de mettre en place une large gamme de schémas, avec des propriétés de stabilité différentes.

Ce chapitre est décomposé en quatre grandes parties. Tout d'abord (section 2.1.1), une présentation de la théorie des méthodes de Runge-Kutta Partitionnées est réalisée. Cette théorie joue un rôle important car elle est utilisée pour déterminer l'ordre temporel des différents schémas à pas de temps local de ce mémoire (les schémas de Constantinescu et Sandu et les schémas que nous proposerons au chapitre 3).

Ensuite (section 2.2.1), nous présentons l'approche générale de Constantinescu et Sandu, puis (section 2.3) nous donnons une présentation détaillée du schéma de Constantinescu et Sandu basé sur la méthode RK2.

Enfin, dans la dernière partie de ce chapitre (section 2.4), nous proposons un nouveau schéma de Constantinescu et Sandu, basé sur la méthode RK3 "low storage" de ce mémoire.

Notation : Dans la suite de ce mémoire, toute solution u évaluée au temps t^n est notée $u^{n,1}$. Toute solution u évaluée au temps $t^{n+1} = t^n + \Delta t$ est notée u^{n+1} . Sauf mention contraire, la solution $u^{n,1}$ est supposée connue et est utilisée pour initialiser les différentes méthodes de Runge-Kutta décrites dans ce chapitre.

2.1 Méthodes de Runge-Kutta Partitionnées

Dans cette partie, la théorie des méthodes de Runge-Kutta Partitionnées (RKP) est présentée. Cette théorie est utilisée dans [16, 77, 78, 17] pour établir l'ordre temporel des schémas à pas de temps local.

Dans un premier temps, nous présentons le lien entre schémas à pas de temps local et méthodes RKP. Dans un second temps, les conditions d'ordre fournies par la théorie des méthodes RKP sont présentées.

2.1.1 Lien entre schémas à pas de temps local et méthodes RKP

On considère le domaine de calcul 1D représenté sur la figure 2.1 (les cellules fictives sont volontairement omises pour plus de clarté). On suppose que ce domaine est composé de deux sous-domaines présentant chacun une taille de cellule différente. La différence de taille de cellules entre les deux sous-domaines induit une différence de condition CFL. Le second sous-domaine, de CFL plus restrictive, nécessite l'utilisation d'un pas temps plus faible que dans le premier sous-domaine. Dans la littérature traitant de schémas à pas de temps local ([16, 77, 78, 17]), le second sous-domaine est qualifié de sous-domaine rapide car la solution évolue plus rapidement que dans le premier sous-domaine (à cause de son petit pas de temps). Le premier sous-domaine est qualifié de sous-domaine lent. Dans un soucis d'uniformisation avec la littérature traitant de schémas à pas de temps local, les termes anglais sont utilisés : le premier sous-domaine est nommé "Slow subdomain" et le second sous-domaine est appelé "Fast subdomain" (comme dans [16, 77, 78, 17]).

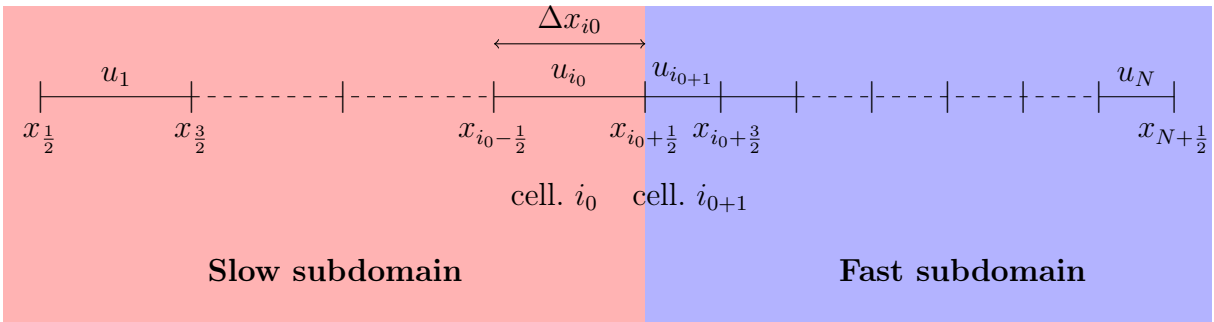


FIGURE 2.1 – Domaine de calcul 1D partagé en deux sous-domaines

On suppose que les cellules du sous-domaine Fast sont deux fois plus petites que les cellules du sous-domaine Slow. Ainsi, la condition CFL dans le sous-domaine Fast est deux fois plus restrictive que dans le sous-domaine Slow. Une approche intéressante consisterait à mettre en œuvre un schéma à pas de temps local dans lequel :

- une application de la méthode RK2 de pas de temps Δt est effectuée dans le sous-domaine Slow entre t^n et $t^n + \Delta t$,
- deux applications successives de la méthode RK2 de pas de temps $\Delta t/2$ sont effectuées dans le sous-domaine Fast entre t^n et $t^n + \Delta t$.

Nous allons tenter de construire un peu naïvement ce type de schéma, afin de mettre en évidence les difficultés rencontrées.

Tentative de construction d'un schéma à pas de temps local

Tout d'abord, on rappelle brièvement l'expression de la méthode RK2. On considère une équation différentielle du type :

$$\frac{dU}{dt} = -D(U) \quad (2.1)$$

La méthode RK2 utilisée pour intégrer l'équation (2.1) entre t^n et $t^n + \Delta t$ s'écrit :

$$\begin{aligned} \text{estimation : } U^{n,2} &= U^{n,1} - \Delta t D(U^{n,1}) \\ \text{solution à } t^n + \Delta t : U^{n+1} &= U^{n,1} - \frac{\Delta t}{2} (D(U^{n,1}) + D(U^{n,2})) \end{aligned} \quad (2.2)$$

Première étape du schéma : Cette étape est illustrée par la figure 2.2. Dans un souci de lisibilité, seules les deux dernières cellules du sous-domaine Slow (cellules $i_0 - 1$ et i_0) et les deux premières cellules du sous-domaine Fast (cellules $i_0 + 1$ et $i_0 + 2$) sont représentées. Dans le sous-domaine Slow, l'estimation est calculée avec un pas de temps Δt : elle est notée $U^{n,2}(\Delta t)$ sur la figure 2.2. Dans le sous-domaine Fast, l'estimation est calculée avec un pas de temps $\Delta t/2$: elle est notée $U^{n,2}(\Delta t/2)$ sur la figure 2.2. On suppose que le schéma numérique a un stencil de trois, illustré par les flèches en pointillés. Le calcul de ces estimations ne pose aucun problème. En effet, leur calcul ne fait intervenir que $U^{n,1}$, supposée connue partout.

Deuxième étape du schéma : Cette étape est illustrée par la figure 2.3. Elle est dédiée au calcul de la solution au temps $t^n + \Delta t/2$ (notée $U^{n+\frac{1}{2}}$), sur la première cellule du sous-domaine Fast. Cependant, comme le montre la figure 2.3, le calcul de $U^{n+\frac{1}{2}}$ nécessite de connaître $U^{n,2}(\Delta t/2)$ sur l'ensemble du stencil. Or, cette valeur numérique n'est pas connue sur la cellule i_0 (dernière cellule du sous-domaine Slow).

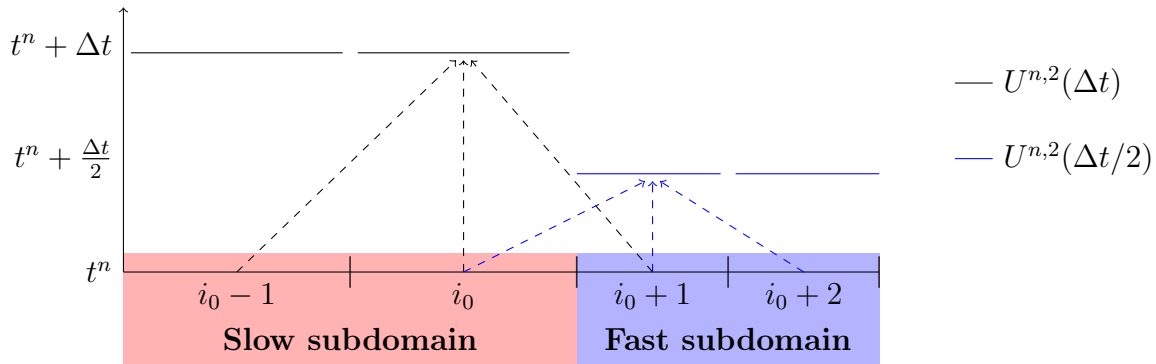


FIGURE 2.2 – Calcul de la première estimation de la méthode RK2 avec un pas de temps Δt sur les deux dernières cellules du sous-domaine Slow et avec un pas de temps $\Delta t/2$ sur les deux premières cellules du sous-domaine Fast.

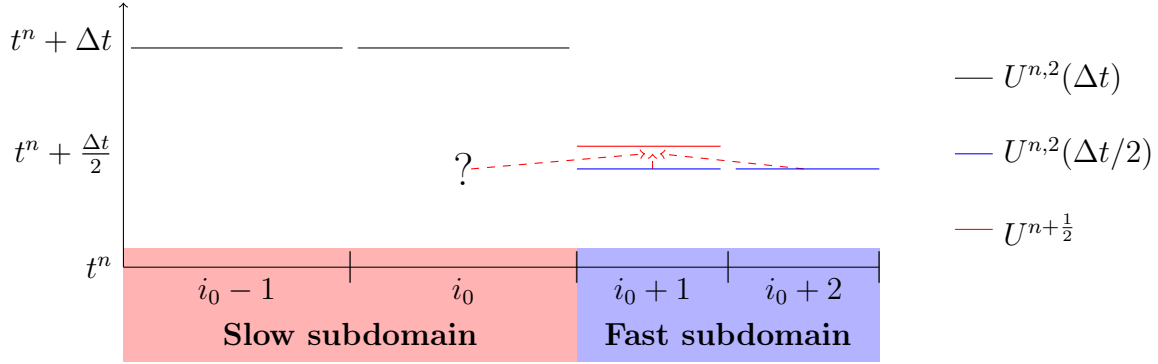


FIGURE 2.3 – Calcul de la solution au temps $t^n + \Delta t/2$, $U^{n+\frac{1}{2}}$, sur la première cellule du sous-domaine Fast

Cet exemple montre que sur la dernière cellule du sous-domaine Slow, il manque une valeur numérique pour pouvoir calculer $U^{n+\frac{1}{2}}$ sur la cellule $i_0 + 1$. Plusieurs approches sont envisageables pour résoudre ce problème :

- interpoler la valeur manquante sur la cellule i_0 ,
- utiliser l'estimation $U^{n,2}(\Delta t)$ calculée sur la cellule i_0 (le stencil de $U^{n+\frac{1}{2}}$ sera alors composé de valeurs numériques estimées à des instants différents).

Mais une nouvelle question se pose alors : **la solution numérique calculée avec une des deux approches est-elle précise à l'ordre deux ?** On souhaite en effet que les solutions numériques calculées sur chaque cellule (et notamment sur les cellules à l'interface entre les sous-domaines) soient précises à l'ordre deux.

Pour répondre à cette question, il faut introduire une description mathématique des schémas à pas de temps local, comme cela a été fait dans [16, 17, 72].

Description mathématique d'un schéma à pas de temps local

Comme montré dans la partie 1.1.2, la discrétisation spatiale d'une EDP sur le maillage de la figure 2.1 mène au système différentiel suivant :

$$\frac{dU}{dt} = -D(U), \quad (2.3)$$

avec $U = (u_1, \dots, u_{i_0}, u_{i_0+1}, \dots, u_N)^T$ et $D = (D_1, \dots, D_{i_0}, D_{i_0+1}, \dots, D_N)^T$.

Dans [16], les auteurs introduisent les vecteur U_S et U_F contenant respectivement les N_S valeurs numériques du vecteur solution du sous-domaine "Slow" et les N_F valeurs numériques du vecteur solution sous-domaine "Fast" :

$$U_S = (u_1, \dots, u_{i_0})^T, \quad U_S \in \mathbb{R}^{N_S},$$

$$U_F = (u_{i_0+1}, \dots, u_N)^T, \quad U_F \in \mathbb{R}^{N_F}.$$

La réécriture du système (2.3) en fonction de U_S et U_F mène au système partitionné suivant :

$$\begin{cases} \frac{dU_S}{dt} = -D_S(U_S, U_F) \\ \frac{dU_F}{dt} = -D_F(U_S, U_F) \end{cases} \quad (2.4)$$

Dans le cas d'un schéma à pas de temps local, la première équation du système (2.4) est intégrée avec une méthode de Runge-Kutta de pas de temps Δt que nous appelons RK_S (car elle est utilisée pour intégrer l'équation différentielle portant sur U_S). La seconde équation du système (2.4) est intégrée avec une méthode de Runge-Kutta de pas de temps inférieur à Δt , que nous appelons RK_F (car elle est utilisée pour intégrer l'équation différentielle portant sur U_F). Précisons que nous restons ici volontairement vague sur la construction des méthodes RK_S et RK_F , ce problème sera largement développé par la suite.

Le fait d'intégrer la première équation du système (2.4) avec la méthode RK_S et la seconde équation avec la méthode RK_F nous place dans le cadre d'une méthode de Runge-Kutta Partitionnée (RKP) [32]. En effet, une méthode RKP se compose de deux méthodes de Runge-Kutta différentes pour résoudre les équations d'un système différentiel. La définition 2.1 suivante, tirée de [32], est la définition mathématique d'une méthode RKP.

Définition 2.1. Soient les méthodes RK_S et RK_F à q étapes et Δt , un pas de temps commun aux deux méthodes. On désigne par (a_{ij}^S, b_i^S, c_i^S) et (a_{ij}^F, b_i^F, c_i^F) les coefficients respectifs des méthodes RK_S et RK_F . La méthode RKP composée de RK_S pour intégrer la première équation de (2.4) et de RK_F pour intégrer la seconde équation de (2.4) s'écrit :

$$\begin{aligned} U_S^{n+1} &= U_S^{n,1} - \Delta t \sum_{i=1}^q b_i^S D_S^{n,i} & U_F^{n+1} &= U_F^{n,1} - \Delta t \sum_{i=1}^q b_i^F D_F^{n,i} \\ U_S^{n,i} &= U_S^{n,1} - \Delta t \sum_{j=1}^{i-1} a_{ij}^S D_S^{n,j} & U_F^{n,i} &= U_F^{n,1} - \Delta t \sum_{j=1}^{i-1} a_{ij}^F D_F^{n,j} \\ D_S^{n,i} &= D_S(U_S^{n,i}, U_F^{n,i}) & D_F^{n,i} &= D_F(U_S^{n,i}, U_F^{n,i}) \end{aligned}$$

Remarquons qu'une méthode RKP n'est définie que pour des méthodes RK_S et RK_F présentant le même nombre d'étape q .

Étudier l'ordre temporel d'un schéma à pas de temps local revient à étudier l'ordre temporel de la méthode RKP qui lui est associée. L'étude de l'ordre temporel d'une méthode RKP se fait en vérifiant des conditions d'ordre. En effet, à l'image des méthodes de Runge-Kutta "classiques", il est possible de déterminer des conditions sur les coefficients des méthodes RK_S et RK_F pour que les solutions numériques U_S^{n+1} et U_F^{n+1} soient précises à un certain ordre n .

2.1.2 Conditions d'ordre pour les méthodes RKP

Hairer et Wanner [32] ont publié les conditions portant sur les coefficients des méthodes RK_S et RK_F pour obtenir une solution numérique du système (2.4) précise à l'ordre deux. D'une part, chacune des méthodes doit être au moins d'ordre deux. D'autre part, deux conditions de couplage entre les coefficients des deux méthodes doivent être satisfaites. Toutes ces conditions sont regroupées dans le tableau 2.1.

Les conditions d'ordre trois sont regroupées dans le tableau 2.2. Elles ont notamment été publiées dans le papier de Günther *et al* [81]. Une méthode permettant d'obtenir ces conditions d'ordre (basée sur des développements de Taylor) est présentée en annexe A. Le tableau 2.2 montre que les méthodes RK_S et RK_F doivent être d'ordre trois, et que dix conditions de couplage entre les coefficients des méthodes doivent être satisfaites.

RK_S d'ordre 2	RK_F d'ordre 2	Conditions de couplage
$\sum_{i=1}^q b_i^S = 1$	$\sum_{i=1}^q b_i^F = 1$	$\sum_{i=1}^q b_i^S c_i^F = \frac{1}{2}$
$\sum_{i=1}^q b_i^S c_i^S = \frac{1}{2}$	$\sum_{i=1}^q b_i^F c_i^F = \frac{1}{2}$	$\sum_{i=1}^q b_i^F c_i^S = \frac{1}{2}$

TABLE 2.1 – Conditions portant sur les coefficients de Runge-Kutta des méthodes RK_S et RK_F pour obtenir une solution numérique d'ordre deux

RK_S d'ordre 3	RK_F d'ordre 3	Conditions de couplage
$\sum_{i=1}^q b_i^S = 1$	$\sum_{i=1}^q b_i^F = 1$	$\sum_{i=1}^q b_i^S c_i^F = \frac{1}{2}$ (1)
$\sum_{i=1}^q b_i^S c_i^S = \frac{1}{2}$	$\sum_{i=1}^q b_i^F c_i^F = \frac{1}{2}$	$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^S = \frac{1}{6}$ (2)
$\sum_{i=1}^q b_i^S (c_i^S)^2 = \frac{1}{3}$	$\sum_{i=1}^q b_i^F (c_i^F)^2 = \frac{1}{3}$	$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^F = \frac{1}{6}$ (3)
$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^S = \frac{1}{6}$	$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^F a_{ij}^F c_j^F = \frac{1}{6}$	$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^F = \frac{1}{6}$ (4)
		$\sum_{i=1}^q b_i^S c_i^F c_i^S = \frac{1}{3}$ (5)
		$\sum_{i=1}^q b_i^F c_i^S = \frac{1}{2}$ (6)
		$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^F a_{ij}^S c_j^S = \frac{1}{6}$ (7)
		$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^F a_{ij}^S c_j^F = \frac{1}{6}$ (8)
		$\sum_{i=1}^q \sum_{j=1}^{i-1} b_i^F a_{ij}^F c_j^S = \frac{1}{6}$ (9)
		$\sum_{i=1}^q b_i^F c_i^S c_i^F = \frac{1}{3}$ (10)

TABLE 2.2 – Conditions portant sur les coefficients de RK_S et RK_F pour obtenir une solution numérique d'ordre trois. Les différentes conditions de couplage sont numérotées de (1) à (10) afin de faciliter les références à ces conditions dans le mémoire.

2.2 Schémas à pas de temps local de Constantinescu et Sandu

Constantinescu et Sandu [16] ont développé une stratégie générale pour construire des schémas à pas de temps local : n'importe quelle méthode de Runge-Kutta peut être utilisée comme méthode de base pour générer les méthodes RK_S et RK_F . Les méthodes RK_S et RK_F des schémas de Constantinescu et Sandu ont la particularité d'avoir leurs coefficients b égaux : $b_i^S = b_i^F, \forall i$. Ceci permet de satisfaire "automatiquement" certaines conditions de couplage et d'assurer l'ordre deux des schémas. Ceci permet également d'assurer la conservativité des schémas.

2.2.1 Construction des méthodes RK_S et RK_F

A l'image de l'article de Constantinescu et Sandu [16], les tableaux de Butcher seront largement utilisés pour décrire la construction des méthodes RK_S et RK_F .

On considère une méthode de Runge-Kutta de base à q étapes, de coefficients a_{ij} , b_i et c_i (avec $1 \leq i, j \leq q$). Le tableau de Butcher de cette méthode est représenté sur le tableau 2.3 où la matrice A contient les coefficients a_{ij} et les vecteur b^T et c contiennent respectivement les coefficients b_i et c_i .

Dans [16], Constantinescu et Sandu montrent que les deux applications successives de la méthode de base de pas de temps $\Delta t/2$ peuvent s'écrire sous la forme d'une méthode de Runge-Kutta de pas de temps Δt , à $2q$ étapes. Cette méthode est la méthode RK_F , décrite par le tableau de Butcher 2.4. En annexe B, nous fournissons une preuve montrant que le tableau de Butcher 2.4 décrit bien deux application successives de la méthode de base de pas de temps $\Delta t/2$. Sur ce tableau, $\mathbf{1}$ désigne le vecteur de taille q dont tous les coefficients valent 1. Par commodité, la matrice A et les vecteurs b^T et c de la méthode RK_F sont notés A^F , b^{F^T} et c^F .

Dans le sous-domaine Slow, Constantinescu et Sandu construisent une méthode à $2q$ étapes à partir de la méthode de base : il s'agit de la méthode RK_S . On rappelle que la méthode RK_S doit posséder le même nombre d'étapes que la méthode RK_F pour entrer dans le cadre des méthodes RKP. Comme le montre le tableau de Butcher 2.5, la méthode RK_S est construite en dupliquant la méthode de base, et en divisant les coefficients b par deux. Par commodité, la matrice A et les vecteurs b^T et c de la méthode RK_S sont notés A^S , b^{S^T} et c^S .

c	A
	b^T

TABLE 2.3 – Tableau de Butcher de la méthode de base

$$c^F \left\{ \begin{array}{c|cc} & \overbrace{A/2}^{A^F} & \\ \hline c/2 & & \\ (1+c)/2 & b^T/2 & A/2 \\ \hline & \underbrace{b^T/2 \quad b^T/2}_{b^{FT}} & \end{array} \right.$$

TABLE 2.4 – Tableau de Butcher de la méthode RK_F en fonction de la matrice A et des vecteurs b^T et c de la méthode de base

$$c^S \left\{ \begin{array}{c|cc} & \overbrace{A}^{A^S} & \\ \hline c & & \\ c & & A \\ \hline & \underbrace{b^T/2 \quad b^T/2}_{b^{ST}} & \end{array} \right.$$

TABLE 2.5 – Tableau de Butcher de la méthode RK_S en fonction de la matrice A et des vecteurs b^T et c de la méthode de base

Les tableaux de Butcher 2.4 et 2.5 montrent que le vecteur b^S de la méthode RK_S est identique au vecteur b^F de la méthode RK_F .

Remarque 2.1. *La méthode RK_S , constituée du même nombre d'étapes que la méthode RK_F , est à priori aussi coûteuse que cette dernière d'un point de vue algorithmique. Ceci va à l'encontre du principe d'un schéma à pas de temps local, basé sur l'utilisation de méthodes de pas de temps différents, et donc de coût algorithmique différent dans chaque sous-domaine. On montrera plus loin qu'à une certaine distance de l'interface, la méthode RK_S peut se simplifier en la méthode de base (deux fois moins coûteuse) sans modifier le schéma. Mais on considère dans un premier temps que les méthodes RK_S et RK_F sont appliquées sur l'ensemble des sous-domaines Slow et Fast. Cette hypothèse nous permet notamment de nous placer dans le cadre de la théorie des méthodes RKP.*

Remarque 2.2. *Les méthodes RK_S et RK_F du schéma de Constantinescu et Sandu n'ont pas les mêmes coefficients c_i . En effet, le vecteur c^S de la méthode RK_S s'écrit :*

$$c^S = [c, c]$$

tandis que le vecteur c^F de la méthode RK_F s'écrit :

$$c^F = [c/2, (1+c)/2]$$

Par conséquent, des solutions calculées à des instants différents peuvent être mélangées dans le stencil spatial [16]. Ceci est montré sur la figure 2.4 suivante, où le stencil de la solution numérique $U^{n+\frac{1}{2}}$ se compose de deux valeurs estimées à l'instant $t^n + \Delta t/2$ (notées $U^{n,2}(\Delta t/2)$) et d'une valeur estimée à l'instant $t^n + \Delta t$ (notée $U^{n,2}(\Delta t)$).

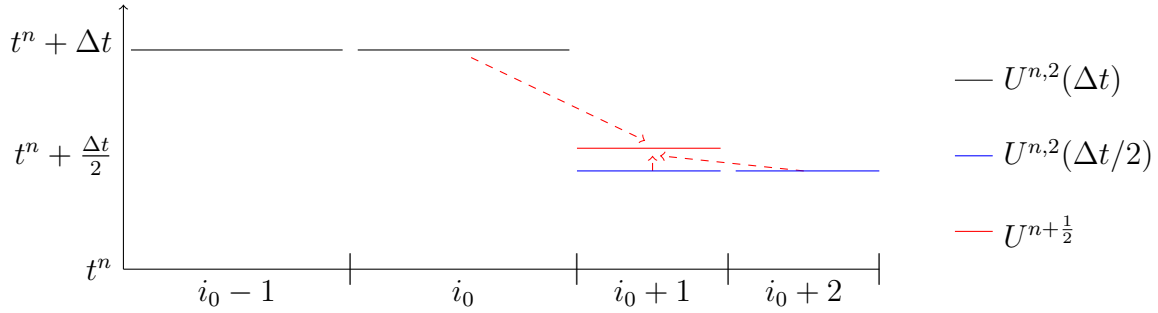


FIGURE 2.4 – Stencil de la solution numérique $U^{n+\frac{1}{2}}$.

Exemple : Illustrons la stratégie de Constantinescu et Sandu lorsque la méthode de base est la méthode RK2 définie dans la partie 1.3.2. Le tableau de Butcher de la méthode RK2 est rappelé sur le tableau 2.6. La méthode RK_F associée est décrite par le tableau de Butcher 2.7. La méthode RK_S associée est décrite par le tableau de Butcher 2.8.

c	A
	b^T

↔

0	0
1	$1 \quad 0$
	$1/2 \quad 1/2$

TABLE 2.6 – A titre d'exemple, on considère que la méthode de base est la méthode RK2 définie dans la partie 1.3.2.

$c/2$	$A/2$
$(\mathbb{1}+c)/2$	$b^T/2 \quad A/2$
	$b^T/2 \quad b^T/2$

↔

0	0
$\frac{1}{2}$	$\frac{1}{2} \quad 0$
$\frac{1}{2}$	$\frac{1}{4} \quad \frac{1}{4} \quad 0$
1	$\frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{2} \quad 0$
	$\frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4} \quad \frac{1}{4}$

TABLE 2.7 – Tableau de Butcher de la méthode RK_F , qui décrit deux applications de la méthode RK2 de pas de temps $\Delta t/2$.

c	A	↔	0	0			
c	A		1	1	0		
			0	0	0	0	
			1	0	0	1	0
	$b^T/2$ $b^T/2$						$\frac{1}{4}$ $\frac{1}{4}$ $\frac{1}{4}$ $\frac{1}{4}$

TABLE 2.8 – Tableau de Butcher de la méthode RK_S , constituée de deux applications de la méthode RK2.

Dans la suite, on discute des questions d'ordre temporel et de conservativité des schémas de Constantinescu et Sandu. Ces questions sont traitées dans un cas général où les méthodes RK_S et RK_F sont générées à partir de n'importe quelle méthode de base.

2.2.2 Ordre temporel des schémas de Constantinescu et Sandu

La proposition 2.1 suivante est tirée de [16].

Proposition 2.1. *Les schémas de Constantinescu et Sandu sont d'ordre deux si la méthode de base est d'ordre deux.*

Démonstration. Les schémas de Constantinescu et Sandu sont constitués de la méthode RK_S (décrite par le tableau de Butcher 2.5) dans le sous-domaine Slow et de la méthode RK_F (décrite par le tableau de Butcher 2.4) dans le sous-domaine Fast. Mathématiquement, cela revient à intégrer la première équation du système partitionné (2.5) avec la méthode RK_S et la seconde équation de ce système avec la méthode RK_F .

$$\begin{cases} \frac{dU_S}{dt} = -D_S(U_S, U_F) & \text{résolue avec } RK_S \\ \frac{dU_F}{dt} = -D_F(U_S, U_F) & \text{résolue avec } RK_F \end{cases} \quad (2.5)$$

On se retrouve dans le cadre de la théorie des méthodes RKP présentée dans la partie 2.1.1. On montre que la méthode RKP constituée de RK_S et RK_F est d'ordre deux en vérifiant les conditions d'ordre fournies par le tableau 2.1 (page 60).

On se donne une méthode de base de coefficients (a_{ij}, b_i, c_i) avec $1 \leq i, j \leq q$. On suppose que la méthode de base est d'ordre deux, ce qui se traduit par les relations suivantes sur les coefficients de cette méthode (d'après le tableau 1.2, page 26) :

$$\sum_{i=1}^q b_i = 1 \quad (\text{ordre 1}) \quad (2.6)$$

$$\sum_{i=1}^q b_i c_i = \frac{1}{2} \quad (\text{ordre } 2) \quad (2.7)$$

Vérifions les conditions d'ordre deux du tableau 2.1. On rappelle que les coefficients des méthodes RK_S et RK_F sont présentés sur les tableaux de Butcher 2.5 et 2.4.

Vérifions que la méthode RK_F , de coefficients (a_{ij}^F, b_i^F, c_i^F) , avec $1 \leq i, j \leq 2q$, est d'ordre deux :

$$\sum_{i=1}^{2q} b_i^F = 2 \sum_{i=1}^q \frac{b_i}{2} = \sum_{i=1}^q b_i = 1,$$

en utilisant (2.6). De plus :

$$\begin{aligned} \sum_{i=1}^{2q} b_i^F c_i^F &= \sum_{i=1}^q \frac{b_i}{2} \frac{c_i}{2} + \frac{1}{2} \sum_{i=1}^q \frac{b_i}{2} + \sum_{i=1}^q \frac{b_i}{2} \frac{c_i}{2} \\ &= \frac{1}{2} \sum_{i=1}^q b_i c_i + \frac{1}{4} \sum_{i=1}^q b_i \\ &= \frac{1}{4} + \frac{1}{4} \\ &= \frac{1}{2}, \end{aligned}$$

en utilisant (2.6) et (2.7). Donc la méthode RK_F est bien d'ordre deux.

Vérifions que la méthode RK_S de coefficients (a_{ij}^S, b_i^S, c_i^S) , avec $1 \leq i, j \leq 2q$ est d'ordre deux :

$$\sum_{i=1}^{2q} b_i^S = 2 \sum_{i=1}^q \frac{b_i}{2} = \sum_{i=1}^q b_i = 1,$$

en utilisant (2.6). De plus :

$$\sum_{i=1}^{2q} b_i^S c_i^S = 2 \sum_{i=1}^q \frac{b_i}{2} c_i = \sum_{i=1}^q b_i c_i = \frac{1}{2},$$

en utilisant (2.7). Donc la méthode RKS est bien d'ordre deux.

Vérifions que les conditions de couplage entre les méthodes RKS et RKF sont vérifiées :

Tout d'abord, on a bien :

$$\sum_{i=1}^{2q} b_i^S c_i^F = \sum_{i=1}^{2q} b_i^F c_i^F = \frac{1}{2},$$

en utilisant l'égalité des coefficients b et le fait que RKF soit d'ordre deux. De même, on a bien :

$$\sum_{i=1}^{2q} b_i^F c_i^S = \sum_{i=1}^{2q} b_i^S c_i^S = \frac{1}{2},$$

en utilisant le fait que pour tout i , $b_i^S = b_i^F$ et le fait que RKS soit d'ordre deux.

Les méthodes RK_S et RK_F vérifient toutes les conditions d'ordre deux fournies par la théorie des méthodes RKP (tableau 2.1), donc les schémas de Constantinescu et Sandu composés de RK_S et RK_F sont d'ordre deux en temps. \square

On énonce la proposition 2.2 suivante, également tirée de [16], qui concerne l'ordre trois.

Proposition 2.2. *Les schémas de Constantinescu et Sandu ne peuvent pas être d'ordre trois, même si la méthode de base est d'ordre trois.*

Démonstration. Supposons que la méthode de base soit d'ordre trois. Les coefficients de cette méthodes vérifient donc la relation d'ordre un (2.6), d'ordre deux (2.7) ainsi que les deux relations suivantes (voir tableau 1.2) :

$$\sum_{i=1}^q b_i c_i^2 = \frac{1}{3} \quad (\text{ordre 3}) \quad (2.8)$$

$$\sum_{i=1}^q \sum_{j=1}^q b_i a_{ij} c_j = \frac{1}{6} \quad (\text{ordre 3}) \quad (2.9)$$

On peut montrer que la relation de couplage (10) du tableau 2.2 (page 60), qui nécessaire à l'obtention d'une méthode RKP d'ordre trois, n'est pas vérifiée.

$$\begin{aligned} \sum_{i=1}^{2q} b_i^F c_i^F c_i^S &= \sum_{i=1}^q \frac{b_i c_i}{2} \frac{c_i}{2} + \frac{1}{2} \sum_{i=1}^q \frac{b_i}{2} c_i + \sum_{i=1}^q \frac{b_i c_i}{2} \frac{c_i}{2} \\ &= \frac{1}{2} \sum_{i=1}^q b_i c_i^2 + \frac{1}{4} \sum_{i=1}^q b_i c_i \\ &= \frac{1}{6} + \frac{1}{8} \quad (\text{en utilisant les relations (2.8) et (2.9)}) \\ &= \frac{7}{24} \\ &\neq \frac{1}{3}, \end{aligned} \quad (2.10)$$

Par conséquent, les schémas de Constantinescu et Sandu ne peuvent pas être d'ordre trois. \square

2.2.3 Conservativité des schémas de Constantinescu et Sandu

La proposition 2.3 suivante est tirée de [16].

Proposition 2.3. *Les schémas de Constantinescu et Sandu sont conservatifs.*

Démonstration. Pour cette démonstration, on intègre temporellement l'équation discrétisée en espace suivante :

$$\frac{d}{dt}u_i = -\frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right), \quad (2.11)$$

sur le domaine de calcul qui est rappelé sur la figure 2.5.

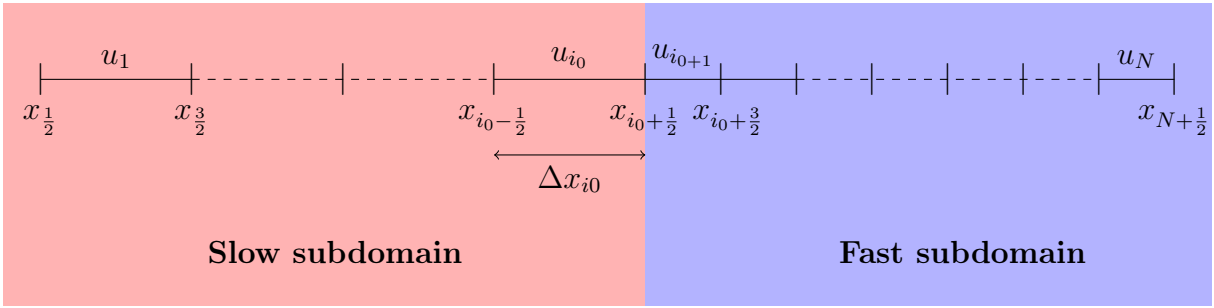


FIGURE 2.5 – Domaine de calcul 1D partagé en deux sous-domaines

D'après Constantinescu et Sandu [16], le problème de conservativité peut se poser à l'interface entre les sous-domaines, puisqu'un changement de méthode d'intégration temporelle apparaît à cette interface. Pour montrer la conservativité des schémas de Constantinescu et Sandu, il faut vérifier que le flux total calculé par la méthode RK_S à l'interface $x_{i_0+\frac{1}{2}}$ entre t^n et $t^n + \Delta t$ est égal au flux total calculé par la méthode RK_F à cette même interface et pendant le même intervalle de temps. La solution numérique calculée au temps $t^n + \Delta t$ avec la méthode RK_S sur la dernière cellule du sous-domaine Slow (cellule i_0) s'écrit :

$$u_{i_0}^{n+1} = u_{i_0}^n - \frac{\Delta t}{\Delta x_{i_0}} \sum_{j=1}^{2q} b_j^S \left(F_{i_0+\frac{1}{2}}(u_{i_0}^{n,j}, u_{i_0+1}^{n,j}) - F_{i_0-\frac{1}{2}}(u_{i_0-1}^{n,j}, u_{i_0}^{n,j}) \right), \quad (2.12)$$

où $u_i^{n,j}$ désigne la j -eme estimation de Runge-Kutta sur la cellule i . Le flux total calculé avec la méthode RK_S entre t^n et $t^n + \Delta t$ au niveau de l'interface $x_{i_0+\frac{1}{2}}$ vaut donc :

$$\Phi_{i_0+\frac{1}{2}}^{RK_S} = \sum_{j=1}^{2q} b_j^S F_{i_0+\frac{1}{2}}(u_{i_0}^{n,j}, u_{i_0+1}^{n,j}) \quad (2.13)$$

La solution numérique calculée au temps $t^n + \Delta t$ avec la méthode RK_F sur la première cellule du sous-domaine Fast (cellule i_{0+1}) s'écrit :

$$u_{i_{0+1}}^{n+1} = u_{i_{0+1}}^n - \frac{\Delta t}{\Delta x_{i_{0+1}}} \sum_{j=1}^{2q} b_j^F \left(F_{i_0+\frac{3}{2}}(u_{i_{0+1}}^{n,j}, u_{i_{0+2}}^{n,j}) - F_{i_0+\frac{1}{2}}(u_{i_0}^{n,j}, u_{i_{0+1}}^{n,j}) \right) \quad (2.14)$$

Le flux total calculé avec la méthode RKF entre t^n et $t^n + \Delta t$ au niveau de l'interface $x_{i_0+\frac{1}{2}}$ vaut donc :

$$\Phi_{i_0+\frac{1}{2}}^{RK_F} = \sum_{j=1}^{2q} b_j^F F_{i_0+\frac{1}{2}}(u_{i_0}^{n,j}, u_{i_{0+1}}^{n,j}) \quad (2.15)$$

Le schéma est conservatif s'il y a égalité des flux calculés par chaque méthode, c'est à dire si :

$$\Phi_{i_0+\frac{1}{2}}^{RK_S} = \Phi_{i_0+\frac{1}{2}}^{RK_F} \quad (2.16)$$

Au vu des expression des flux totaux (2.13) et (2.15), la condition (2.16) est satisfaite si :

$$b_i^F = b_i^S, \quad 1 \leq i \leq 2q. \quad (2.17)$$

Or la condition (2.17) est satisfaite par les schémas de Constantinescu et Sandu, donc ces derniers sont conservatifs. □

Il s'agit là d'une propriété intéressante des schémas de Constantinescu et Sandu car l'étude de la littérature faite dans la partie 1.7.2 montre que de nombreux schémas à pas de temps local ne sont pas conservatifs, comme par exemple les schémas présentés dans [21, 17, 72, 77, 78].

Nous venons de présenter de manière générale les schémas de Constantinescu et Sandu ainsi que deux propriétés de ces schémas : ils sont précis à l'ordre deux et conservatifs. A présent, nous décrivons spécifiquement le schéma de Constantinescu de Sandu basé sur la méthode RK2. Ce schéma a en effet été utilisé à plusieurs reprises au cours de cette thèse.

2.3 Schéma de Constantinescu et Sandu basé sur la méthode RK2 (CSRK2)

Le schéma de Constantinescu et Sandu basé sur la méthode RK2 est nommé CSRK2 dans tout ce mémoire. Les tableaux de Butcher des méthodes RK_S et RK_F du schéma CSRK2 ont déjà été développés, à titre d'exemple, dans la partie 2.2.1. Il s'agit des tableaux 2.8 et 2.7 (page 63).

D'après la proposition 2.1, le schéma CSRK2 est d'ordre deux (vu que sa méthode de base est d'ordre deux). Le schéma CSRK2 est également conservatif d'après la proposition 2.3.

2.3.1 Discussions autour de la méthode RK_S

Jusqu'à maintenant, nous avons considéré que la méthode RK_S était appliquée sur l'ensemble du sous-domaine Slow. Pour montrer que cette situation n'est pas en adéquation avec la philosophie d'un schéma à pas de temps local, nous commençons par une évaluation rapide du coût algorithmique de la méthode RK_S . Pour cette évaluation du coût algorithmique, on considère l'équation différentielle ci-dessous :

$$\frac{dU}{dt} = -D(U) \quad (2.18)$$

En suivant le tableau de Butcher de la méthode RK_S (rappelé ci-dessous), la méthode RK_S appliquée à la résolution de l'équation (2.18) s'écrit :

0	0					
1	1	0				$U^{n,2} = U^{n,1} - \Delta t D(U^{n,1})$
0	0	0	0			$U^{n,3} = U^{n,1}$
1	0	0	1	0		$U^{n,4} = U^{n,1} - \Delta t D(U^{n,3})$
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$		$U^{n+1} = U^{n,1} - \frac{\Delta t}{4} (D(U^{n,1}) + D(U^{n,2}) + D(U^{n,3}) + D(U^{n,4}))$

TABLE 2.9 – Tableau de Butcher de la méthode RK_S , et application de cette méthode à la résolution de l'équation (2.18).

Le tableau 2.9 montre que la méthode RK_S nécessite d'évaluer quatre bilans de flux D (l'évaluation d'un bilan de flux est généralement l'étape la plus coûteuse en temps de calcul dans un code). C'est autant que la méthode RK_F . En effet, celle-ci décrit deux applications successives de la méthode RK2 de pas de temps $\Delta t/2$, or une application de la méthode RK2 nécessite d'évaluer deux bilans de flux D , comme montré par l'équation (2.2). Comme évoqué dans la remarque 2.1 (page 62), appliquer la méthode RK_S sur l'ensemble du sous-domaine Slow n'est pas en adéquation avec la philosophie d'un schéma à pas de temps local, basée sur l'idée d'utiliser des méthodes de pas de temps différents, et donc de coût algorithmique différent dans chaque sous-domaine. Dans cette partie, on montre qu'au delà de deux cellules jouxtant l'interface entre les sous-domaines Slow et Fast, la méthode RK_S peut se simplifier en la méthode de base (la méthode RK2), deux fois moins coûteuse. On obtient alors un "véritable" schéma à pas de temps local.

Pour cette démonstration, on considère pour davantage de clarté le maillage réduit de la figure 2.6. La méthode RK_S est appliquée à la résolution de l'équation (2.19) sur les cellules 1 à 3 entre t^n et $t^n + \Delta t$. La méthode RK_F est appliquée à la résolution de l'équation (2.19) sur la cellule 4 entre t^n et $t^n + \Delta t$.

$$\frac{du_i}{dt} = -D_i(u_{i-1}, u_i, u_{i+1}) \quad (2.19)$$

L'algorithme du schéma de Constantinescu et Sandu constitué de RK_S et RK_F est présenté ci-après. Dans cet algorithme, la notation suivante est utilisée :

$$D_i^{n,j} = D(u_{i-1}^{n,j}, u_i^{n,j}, u_{i+1}^{n,j}).$$

Concernant les solutions numériques sur les cellules fictives (u_0 et u_5), on suppose sans perte de généralité que pour tout j , $u_0^{n,j} = u_1^{n,j}$ et $u_4^{n,j} = u_5^{n,j}$.

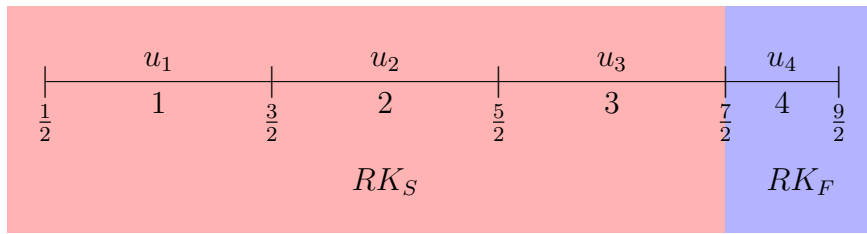


FIGURE 2.6 – Maillage 1D constitué de quatre cellules. Chaque cellule i est délimitée par les interfaces $i - \frac{1}{2}$ et $i + \frac{1}{2}$.

Au vu de l'équation (2.19), la mise à jour de la solution numérique u_4 dans le sous-domaine Fast nécessite la connaissance de u_3 , qui appartient au sous-domaine Slow. On dit que le stencil de la solution u_4 déborde d'une cellule dans le sous-domaine Slow. De même, le stencil de la solution numérique u_3 déborde d'une cellule dans le sous-domaine Fast.

Première étape du schéma de Constantinescu et Sandu constitué de RK_S et RK_F :

$$\begin{aligned} u_1^{n,2} &= u_1^{n,1} - \Delta t D_1^{n,1} \\ u_2^{n,2} &= u_2^{n,1} - \Delta t D_2^{n,1} \\ u_3^{n,2} &= u_3^{n,1} - \Delta t D_3^{n,1} \\ u_4^{n,2} &= u_4^{n,1} - \frac{\Delta t}{2} D_4^{n,1} \end{aligned}$$

Seconde étape du schéma de Constantinescu et Sandu constitué de RK_S et RK_F :

$$\begin{aligned} u_1^{n,3} &= u_1^{n,1} \\ u_2^{n,3} &= u_2^{n,1} \\ u_3^{n,3} &= u_3^{n,1} \\ u_4^{n,3} &= u_4^{n,1} - \frac{\Delta t}{4} (D_4^{n,2} + D_4^{n,1}) \end{aligned}$$

A ce stade de l'algorithme, on a les relations suivantes sur les bilans de flux :

$$\begin{aligned} D_1^{n,3} &= D_1^{n,1} \quad \text{car } u_0^{n,3} = u_0^{n,1}, u_1^{n,3} = u_1^{n,1} \text{ et } u_2^{n,3} = u_2^{n,1} \\ D_2^{n,3} &= D_2^{n,1} \quad \text{car } u_1^{n,3} = u_1^{n,1}, u_2^{n,3} = u_2^{n,1} \text{ et } u_3^{n,3} = u_3^{n,1} \\ D_3^{n,3} &\neq D_3^{n,1} \quad \text{car } u_4^{n,1} \neq u_4^{n,3} \end{aligned}$$

Troisième étape du schéma de Constantinescu et Sandu constitué de RK_S et RK_F :

$$\begin{aligned} u_1^{n,4} &= u_1^{n,1} - \Delta t D_1^{n,3} \\ u_2^{n,4} &= u_2^{n,1} - \Delta t D_2^{n,3} \\ u_3^{n,4} &= u_3^{n,1} - \Delta t D_3^{n,3} \\ u_4^{n,4} &= u_4^{n,1} - \frac{\Delta t}{4} (D_4^{n,1} + D_4^{n,2}) - \frac{\Delta t}{2} D_4^{n,3} \end{aligned}$$

On peut remarquer que :

$$\begin{aligned} u_1^{n,4} &= u_1^{n,2} \quad \text{car } D_1^{n,3} = D_1^{n,1} \\ u_2^{n,4} &= u_2^{n,2} \quad \text{car } D_2^{n,3} = D_2^{n,1} \\ u_3^{n,4} &\neq u_3^{n,1} \quad \text{car } D_3^{n,3} \neq D_3^{n,1} \end{aligned}$$

Par conséquent :

$$\begin{aligned} D_1^{n,4} &= D_1^{n,2} \quad \text{car } u_0^{n,4} = u_0^{n,2}, u_1^{n,4} = u_1^{n,2} \text{ et } u_2^{n,4} = u_2^{n,2} \\ D_2^{n,4} &\neq D_2^{n,2} \quad \text{car } u_3^{n,4} \neq u_3^{n,2} \\ D_3^{n,3} &\neq D_3^{n,1} \quad \text{car } u_4^{n,4} \neq u_4^{n,2} \end{aligned}$$

Quatrième étape du schéma de Constantinescu et Sandu constitué de RK_S et RK_F :

$$\begin{aligned} u_1^{n+1} &= u_1^{n,1} - \frac{\Delta t}{4} (D_1^{n,1} + D_1^{n,2} + D_1^{n,3} + D_1^{n,4}) \\ &= u_1^{n,1} - \frac{\Delta t}{2} (D_1^{n,1} + D_1^{n,2}) \end{aligned} \tag{2.20}$$

$$\begin{aligned} u_2^{n+1} &= u_2^{n,1} - \frac{\Delta t}{4} (D_2^{n,1} + D_2^{n,2} + D_2^{n,3} + D_2^{n,4}) \\ &= u_2^{n,1} - \frac{\Delta t}{4} (2D_2^{n,1} + D_2^{n,2} + D_2^{n,4}) \end{aligned} \tag{2.21}$$

$$u_3^{n+1} = u_3^{n,1} - \frac{\Delta t}{4} (D_3^{n,1} + D_3^{n,2} + D_3^{n,3} + D_3^{n,4}) \tag{2.22}$$

$$\begin{aligned}
u_4^{n+1} &= u_4^{n,1} - \frac{\Delta t}{4} (D_4^{n,1} + D_4^{n,2}) - \frac{\Delta t}{4} (D_4^{n,3} + D_4^{n,4}) \\
&= u_4^{n+\frac{1}{2}} - \frac{\Delta t}{4} (D_4^{n,3} + D_4^{n,4})
\end{aligned}
\tag{2.23}$$

L'équation (2.20) montre que sur la cellule 1, l'application de la méthode RK_S ou de la méthode de base mène à la même expression pour u_1^{n+1} . Sur le maillage complet de la figure 2.1, appliquer la méthode RK_S dans tout le sous-domaine slow (cellule 1 à i_0) est mathématiquement équivalent à appliquer la méthode RK_S sur les cellules $i_0 - 1$ et i_0 et la méthode de base sur les cellules 1 à $i_0 - 2$. Cela ne change rien à l'expression de la solution u_i^{n+1} dans ce sous-domaine. Par contre, au niveau algorithmique, il est beaucoup plus intéressant de remplacer la méthode RK_S par la méthode de base sur les cellules 1 à $i_0 - 2$ puisque la méthode de base nécessite seulement deux évaluations de bilans de flux, contre quatre pour la méthode RK_S . **A partir de maintenant, on désigne par CSRK2 le schéma constitué, dans le sous-domaine Slow, de la méthode de base et de la méthode RK_S . Le schéma pour lequel la méthode RK_S est appliquée sur l'ensemble du sous-domaine Slow est qualifié de schéma CSRK2 équivalent "mathématiquement"**. La figure 2.7 résume la disposition des méthodes d'intégration temporelles dans le cas du schéma CSRK2 et de son schéma équivalent "mathématiquement".

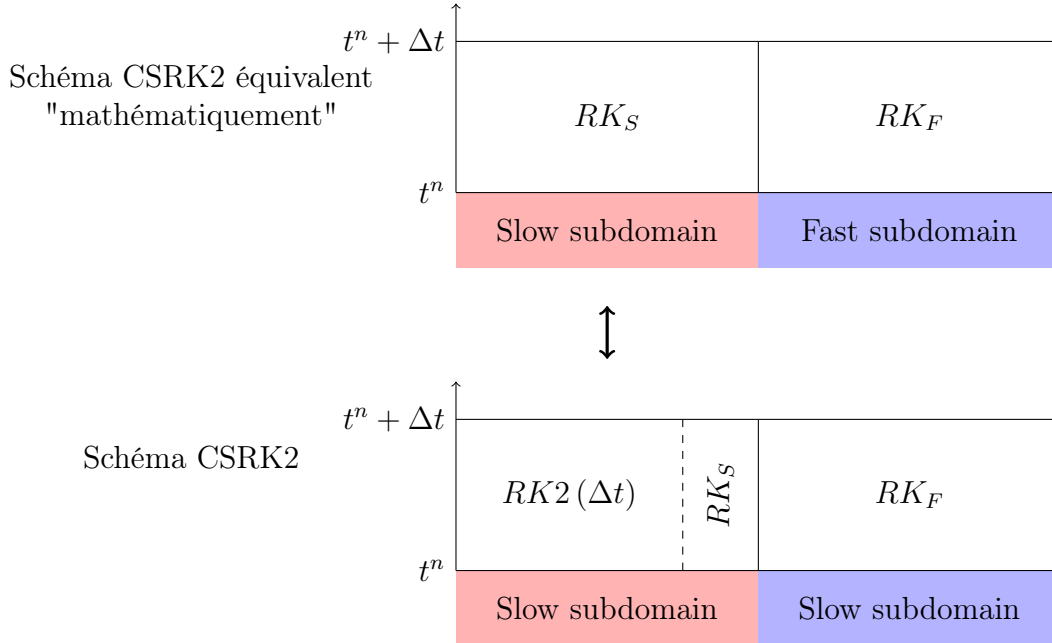


FIGURE 2.7 – Disposition des méthodes d'intégration temporelle pour le schéma CSRK2 et son schéma équivalent "mathématiquement". La méthode de base est notée $RK2(\Delta t)$.

Le schéma CSRK2 hérite des propriétés que nous avons déterminées pour son schéma équivalent mathématiquement. Le schéma CSRK2 est donc, comme son schéma équivalent, d'ordre deux et conservatif.

L'ensemble des cellules sur lesquelles la méthode RK_S est appliquée est appelé zone tampon ("Buffer zone" dans [16]). La taille de la zone tampon (le nombre de cellules sur lequel la méthode RK_S est appliquée) est une donnée qui a son importance. En effet, bien qu'utilisée dans le sous-domaine Slow, la méthode RK_S a le même coût algorithmique que la méthode RK_F . Par conséquent, si la méthode RK_S est appliquée sur un grand nombre de cellule, cela réduit l'efficacité du schéma à pas de temps local. Dans [16], Constantinescu et Sandu précisent que la taille de la zone tampon dépend du débordement d du stencil. Dans ce chapitre, chaque solution numérique a un stencil de trois. Comme le montre la figure 2.8, le stencil de la solution numérique sur la dernière cellule du sous-domaine Slow (u_{i_0}) déborde sur une cellule du sous-domaine Fast. De même, le stencil de la solution numérique sur la première cellule du sous-domaine Fast (u_{i_0+1}) déborde sur une cellule du sous-domaine Slow. Le débordement d est donc égal à 1 ici. Nous avons remarqué que la taille de la zone tampon dépendait également du nombre d'étapes q de la méthode de base. Nous avons appliqué la stratégie de Constantinescu pour différentes valeurs de q et d et obtenu la taille de la zone tampon. Les résultats sont regroupés dans le tableau 2.10.

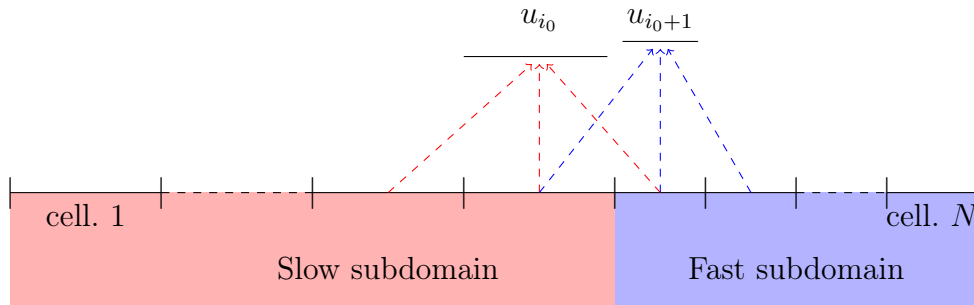


FIGURE 2.8 – Représentation du stencil des solutions numériques u_{i_0} et u_{i_0+1} , afin d'illustrer le débordement du stencil d .

q	d	Taille de la zone tampon
2	1	2
2	2	4
3	1	3
3	2	6

TABLE 2.10 – Taille de la zone tampon en fonction du nombre d'étapes q de la méthode de base et du débordement du stencil d .

Bien que nous n'en ayons pas la preuve mathématique, il semble que la taille de la zone tampon soit égale au produit du débordement du stencil d par le nombre d'étapes de la méthode de base q .

2.3.2 Estimation du coût algorithmique du schéma CSRK2

Deux opérations sont prises en compte pour estimer le coût algorithmique des schémas à pas de temps local de ce mémoire :

- **opération "B" : calcul du Bilan des flux** (on rappelle que le bilan des flux correspond au terme D_i dans l'équation (2.19)),
- **opération "V" : calcul d'une Valeur numérique** (les valeurs numériques sont les différentes estimations $U^{n,j}$ de Runge-Kutta ainsi que la solution "finale" U^{n+1} au temps t^{n+1}).

Le coût algorithmique associé à l'opération B est noté η_B . De même, le coût algorithmique associé à l'opération V est noté η_V . Dans les codes de calcul, l'opération B est généralement plus coûteuse que l'opération V :

$$\eta_B > \eta_V.$$

En effet, l'opération de calcul d'une valeur numérique nécessite généralement peu d'opérations arithmétiques (qui sont de plus assez simples) : multiplier les bilans des flux D_i par les coefficients de Runge-Kutta a_{ij} ou b_i et les additionner. Par contre, les bilans de flux D_i ont souvent une expression complexe qui demande plus d'opérations arithmétiques. L'utilisation désormais courante de techniques de reconstructions aux interface (pour augmenter la précision spatiale) et de limiteur de pente ajoute une couche supplémentaire d'opérations arithmétiques pour calculer les bilans de flux.

Le calcul du coût algorithmique C d'un schéma nécessite de compter le nombre d'opérations B et V effectuées entre t^n et $t^n + \Delta t$ sur l'ensemble du domaine de calcul. On définit le coût algorithmique C d'un schéma par la relation ci-dessous :

$$C = (\text{nombre d'opérations } B) \times \eta_B + (\text{nombre d'opérations } V) \times \eta_V \quad (2.24)$$

Le tableau 2.11 montre le nombre d'opérations B et V effectuées par chaque méthode composant le schéma CSRK2, en considérant que les sous-domaine Slow et Fast sont respectivement constitués de N_S et N_F cellules. Le nombre total d'opérations B et V effectuées sur l'ensemble du domaine de calcul est également affiché dans ce tableau. On détaille ci-dessous le nombre d'opérations B et V effectuées par chaque méthode.

- La méthode RK2 de pas de temps Δt , notée RK2 (Δt) est appliquée sur $N_S - 2$ cellules. C'est une méthode de Runge-Kutta à deux étapes qui nécessite, comme l'indique l'équation (2.2), de calculer deux bilans de flux D_i et deux valeurs numériques ($u_i^{n,2}$ et u_i^{n+1}), sur chaque cellule. Le nombre d'opérations B et V effectuées par cette méthode est donc : $2 \times (N_S - 2)$.
- La méthode RK_S est appliquée sur les deux dernières cellules du sous-domaine Slow. C'est une méthode de Runge-Kutta à quatre étapes qui nécessite de calculer quatre bilans de flux et quatre valeurs numériques sur chaque cellule, comme l'indique l'équation (2.9). Le nombre d'opérations B et V effectuées par cette méthode est donc : 4×2 .

- Enfin, la méthode RK_F décrit deux application successives de la méthode RK2 de pas de temps $\Delta t/2$: elle nécessite donc le calcul de quatre bilans de flux et de quatre valeurs numériques sur chaque cellule. Le nombre d'opérations B et V effectuées par cette méthode est donc : $4 \times N_F$

	$RK2(\Delta t)$ $N_S - 2$ cell.	RK_S 2 cell.	RK_F N_F cell.	Total $N_S + N_F$ cell.
Nbre. d'opérations B	$2 \times (N_S - 2)$	4×2	$4 \times N_F$	$2N_S + 4N_F + 4$
Nbre. d'opérations V	$2 \times (N_S - 2)$	4×2	$4 \times N_F$	$2N_S + 4N_F + 4$

TABLE 2.11 – Nombre d'opérations B et V effectuées par chaque méthode composant le schéma CSRK2, et nombre total d'opérations B et V (sur l'ensemble du domaine de calcul).

D'après la formule (2.24), le coût algorithmique du schéma CSRK2 s'écrit :

$$C_{CSRK2} = (2N_S + 4N_F + 4) \times (\eta_B + \eta_V) \quad (2.25)$$

Ce coût algorithmique peut être réécrit en faisant apparaître :

- le coût associé à un schéma "idéal" constitué de la méthode RK2 (Δt) dans tout le domaine Slow et de la méthode RK_F dans le sous-domaine Fast,
- le surcoût par rapport au schéma idéal, dû à la zone tampon.

Le coût algorithmique du schéma CSRK2 est réécrit ci-dessous :

$$C_{CSRK2} = \underbrace{(2N_S + 4N_F) \times (\eta_B + \eta_V)}_{\text{Coût du schéma idéal}} + \underbrace{4 \times (\eta_B + \eta_V)}_{\text{Surcoût dû à la zone tampon}} \quad (2.26)$$

2.3.3 Généralisation à N pas de temps du schéma CSRK2

Le schéma CSRK2 présenté jusqu'à maintenant est constitué de deux pas de temps : Δt et $\Delta t/2$. Comme mentionné dans [16], le schéma peut être généralisé à N pas de temps en appliquant de façon récursive la stratégie définie pour les pas de temps Δt et $\Delta t/2$. La figure 2.9 illustre cette approche récursive (également appelée approche télescopique dans [78, 77]) dans le cas où trois pas de temps différents sont utilisés sur le maillage ($N = 3$).

Cette généralisation est certes simple, mais elle nécessite un rapport constant entre les pas de temps associés à des sous-domaines adjacents. Sur l'exemple de la figure 2.9, il y a toujours un rapport deux entre les pas de temps adjacents.

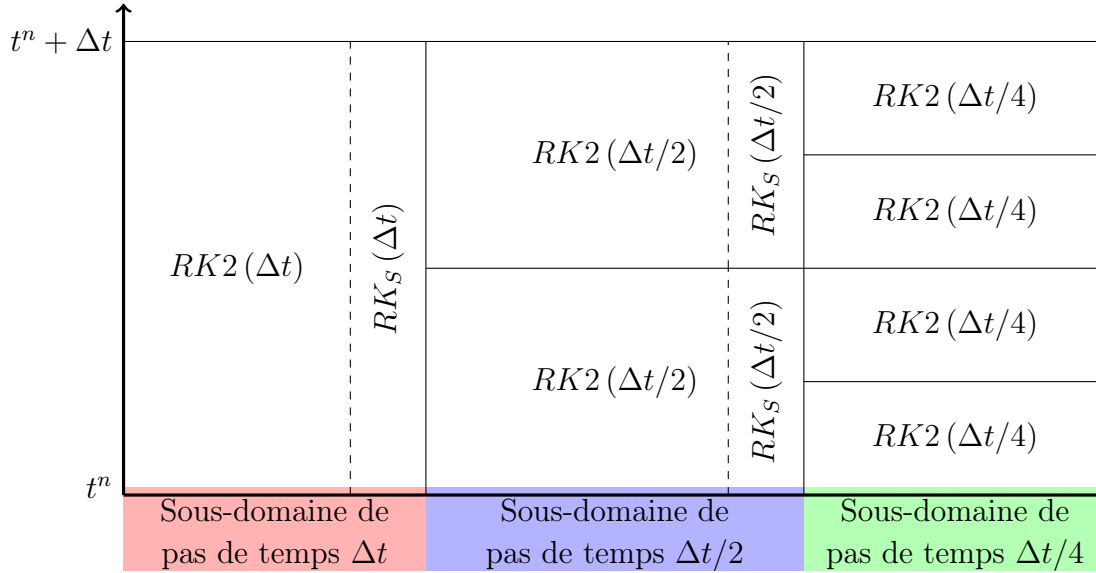


FIGURE 2.9 – Approche récursive pour généraliser le schéma CSRK2 à N pas de temps ($N = 3$ sur la figure).

2.3.4 Discussion sur le choix du rapport entre les pas de temps

Jusqu'à maintenant, nous avons seulement considéré le cas d'un rapport deux entre les pas de temps des sous-domaines Slow et Fast. Ce rapport est d'ailleurs utilisé dans la grande majorité des schémas à pas de temps local de la littérature ([16, 17, 20, 21, 73, 24]). Dans [16], Constantinescu et Sandu ont également proposé d'utiliser leur schéma avec un rapport trois entre les pas de temps des sous-domaines Slow et Fast. Un rapport cinq entre pas de temps adjacents est même proposé dans [78].

En général, un "bon" maillage évolue de manière régulière pour garantir un bon niveau de précision et de stabilité de la solution numérique. Pour des calculs LES par exemple, il est conseillé d'utiliser un rapport de taille n'excédant pas 1.05 pour les cellules adjacentes [82]. La condition CFL impose un rapport du même ordre sur les pas de temps associés à ces cellules. Par conséquent, nous ne voyons pas l'utilité de développer un schéma à pas de temps local utilisant des rapports entre pas de temps supérieurs à deux, étant donné que de tels rapports dans les tailles de cellules ne sont pas utilisés. Précisons que les auteurs ayant développé le schéma avec un rapport cinq entre pas de temps [78] travaillent sur des systèmes d'équations différentielle et ne sont pas confronté à des problématiques de maillage. Le schéma qu'ils ont développé est bien adapté à leur problème mais un peu moins au nôtre.

Comme évoqué précédemment, deux cellules adjacentes ont plutôt un rapport de taille compris entre un et deux. De plus, ce rapport est variable sur le maillage. Par conséquent, un schéma à pas de temps local capable de gérer plusieurs rapports de pas de temps compris entre un et deux (par exemple 3/2, 4/3...) semble la solution idéale pour optimiser la CFL sur tout le maillage. Liu *et al* [83, 84] ont proposé un schéma à pas de temps local de ce type. Néanmoins, leur schéma introduit une grande complexité algorithmique puisque

l'algorithme doit être capable de gérer tous les rapports possibles.

Un schéma à pas de temps local avec un rapport deux entre les pas de temps nous paraît être un bon compromis entre la complexité algorithmique et la volonté d'optimiser la condition CFL sur tout le maillage. De plus, ce rapport deux entre pas de temps est parfaitement adapté aux maillage de type Octree, pour lesquels un raffinement d'un facteur deux est systématiquement utilisé. Ce type de maillage est notamment utilisé en conjonction avec les méthodes IBM (Immersed Boundary Method) [85].

Pour toutes ces raisons, nous ne considérons que des schémas admettant un rapport deux entre les pas de temps dans ce mémoire.

2.4 Schéma de Constantinescu et Sandu basé sur une méthode RK3 (CSRK3)

2.4.1 Construction des méthodes RK_S et RK_F

Dans [24], Sény *et al* proposent d'utiliser une méthode RK3 comme méthode de base du schéma de Constantinescu et Sandu. Nous reprenons cette idée, mais en utilisant la méthode RK3 "low storage" présentée dans la partie 1.3.2, afin d'optimiser le coût mémoire. Utiliser la méthode RK3 comme méthode de base nous permet également d'obtenir un schéma plus stable (et plus performant) que le schéma CSRK2, comme on le verra au chapitre 4. Ce schéma à pas de temps local basé sur la méthode RK3 et utilisant la stratégie de Constantinescu et Sandu est désigné par CSRK3 dans toute la suite de ce mémoire.

Le tableau de Butcher de la méthode RK3 de ce mémoire est rappelé sur le tableau 2.12. La valeur numérique des coefficients de ce tableau est donnée dans la partie 1.3.2.

0	0		
c_2	a_{21}		
c_3	a_{31}	a_{32}	
	b_1	b_2	b_3

TABLE 2.12 – Tableau de Butcher de la méthode RK3 de ce mémoire, utilisée comme méthode de base du schéma CSRK3.

Les tableaux de Butcher des méthodes RK_S et RK_F du schéma CSRK3 sont les tableaux 2.13 et 2.14 ci-dessous. Ils sont obtenus en suivant la stratégie de Constantinescu et Sandu décrite par les tableaux de Butcher 2.5 et 2.4 (page 62).

0	0					
c_2	a_{21}					
c_3	a_{31}	a_{32}				
0	0	0	0			
c_2	0	0	0	a_{21}		
c_3	0	0	0	a_{31}	a_{32}	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$

TABLE 2.13 – Tableau de Butcher de la méthode RK_S du schéma CSRK3

0	0					
$\frac{c_2}{2}$	$\frac{a_{21}}{2}$					
$\frac{c_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$				
$\frac{1}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$			
$\frac{1+c_2}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{21}}{2}$		
$\frac{1+c_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$

TABLE 2.14 – Tableau de Butcher de la méthode RK_F du schéma CSRK3

2.4.2 Discussions autour de la méthode RK_S

D'après le tableau 2.10 (page 73), le schéma CSRK3 a une zone tampon de taille égale à trois cellules (ligne avec $q = 3$ et $d = 1$). On rappelle que dans les mêmes conditions, le schéma CSRK2 a une zone tampon de taille égale à deux cellules. Sur le maillage de la figure 2.5 (page 67), la méthode RK_S est donc équivalente à la méthode de base sur les cellules 1 à $i_0 - 3$. Le schéma CSRK3 est constitué dans le sous-domaine Slow :

- de la méthode de base (tableau de Butcher 2.12) sur les cellules 1 à $i_0 - 3$,
- de la méthode RK_S (tableau de Butcher 2.14) sur les cellules $i_0 - 2$ à i_0 .

Ceci permet de réduire le coût algorithmique par rapport au cas où la méthode RK_S est appliquée sur l'ensemble du sous-domaine Slow. En effet, la méthode RK_S nécessite d'évaluer six bilans de flux contre "seulement" trois pour la méthode de base.

2.4.3 Estimation du coût algorithmique du schéma CSRK3

Le tableau 2.15 montre le nombre de bilans de flux (nombre d'opérations B) et le nombre de valeurs numériques (nombre d'opérations V) calculés par chaque méthode composant le schéma CSRK3. Ce tableau montre aussi le nombre total d'opérations B et d'opérations V effectuées sur l'ensemble du domaine de calcul. On détaille ci-dessous le nombre d'opérations B et V effectuées par chaque méthode.

- La méthode RK3 de pas de temps Δt , notée RK3 (Δt) est appliquée sur $N_S - 3$ cellules. C'est une méthode de Runge-Kutta à trois étapes qui nécessite, comme montré dans la partie 1.3.2, de calculer trois bilans de flux D_i et trois valeurs numériques ($u_i^{n,2}$, $u_i^{n,3}$ et u_i^{n+1}) sur chaque cellule. Cette méthode effectue donc $3 \times (N_S - 3)$ opérations B et autant d'opérations V .
- La méthode RK_S est appliquée sur les trois dernières cellules du sous-domaine Slow. C'est une méthode de Runge-Kutta à six étapes qui nécessite de calculer six bilans

de flux et six valeurs numériques sur chaque cellule, comme l'indique son tableau de Butcher 2.13. Cette méthode effectue donc 6×3 opérations B et autant d'opérations V .

- Enfin, la méthode RK_F décrit deux application successives de la méthode RK3 de pas de temps $\Delta t/2$: elle nécessite donc le calcul de six bilans de flux et de six valeurs numériques sur chaque cellule. Cette méthode effectue donc $6 \times N_F$ opérations B et autant d'opérations V .

	$RK3(\Delta t)$ $N_S - 3$ cell.	RK_S 3 cell.	RK_F N_F cell.	Total $N_S + N_F$ cell.
Nbre. d'opérations B	$3 \times (N_S - 3)$	6×3	$6 \times N_F$	$3N_S + 6N_F + 9$
Nbre. d'opérations V	$3 \times (N_S - 3)$	6×3	$6 \times N_F$	$3N_S + 6N_F + 9$

TABLE 2.15 – Nombre d'opérations B et V effectuées par chaque méthode composant le schéma CSRK3, et nombre total d'opérations B et V (sur l'ensemble du domaine de calcul).

D'après la formule (2.24), le coût algorithmique du schéma CSRK3 s'écrit :

$$\begin{aligned}
 C_{CSRK3} &= (3N_S + 6N_F + 9) \times (\eta_B + \eta_V) \\
 &= \underbrace{(3N_S + 6N_F) \times (\eta_B + \eta_V)}_{\text{Coût algorithmique du schéma idéal}} + \underbrace{9 \times (\eta_B + \eta_V)}_{\text{Surcoût associé à la zone tampon}}
 \end{aligned} \tag{2.27}$$

2.4.4 Ordre temporel et conservativité du schéma CSRK3

Bien que la méthode de base soit d'ordre trois, le schéma de CSRK3 ne l'est pas en vertu de la proposition 2.2. La méthode de base étant d'ordre trois, elle vérifie a fortiori toutes les conditions d'ordre deux. Le schéma CSRK3 est donc d'ordre deux, d'après la proposition 2.1.

D'après la proposition 2.3, le schéma CSRK3 est conservatif.

2.4.5 Généralisation à N pas de temps du schéma CSRK3

A l'image du schéma CSRK2, le schéma CSRK3 peut se généraliser à N pas de temps en respectant la contrainte d'un rapport deux entre les pas de temps associés à chaque sous-domaine adjacent.

Chapitre 3

Nouveaux schémas à pas de temps local

Comme évoqué dans la partie 1.2.2, les simulations instationnaires nécessitent des méthodes d'intégration temporelle précises. Ainsi, un schéma à pas de temps local précis à l'ordre trois nous semble plus adapté à ce type de simulation que les schémas de Constantinescu et Sandu, "seulement" précis à l'ordre deux.

Tous les schémas à pas de temps local d'ordre trois de la littérature [17, 78, 77, 72] sont obtenus en suivant une stratégie qui diffère de celle de Constantinescu et Sandu. Plutôt que d'imposer l'égalité des coefficients b^S et b^F comme Constantinescu et Sandu, les auteurs imposent l'égalité des coefficients c^S et c^F pour satisfaire "automatiquement" certaines relations couplage de la méthode RKP (tableau 2.2, page 60). On rappelle que cette stratégie se traduit par le fait que l'ensemble du stencil d'une solution numérique est estimé au même instant. Précisons que cette stratégie n'est pas conservative. Nous proposons alors une méthode de correction de flux pour rendre les schémas conservatifs.

Afin de nous familiariser avec cette nouvelle stratégie, nous avons dans un premier temps mis en place un nouveau schéma à pas de temps local "simple", d'ordre deux. Nous avons ensuite mis en place un nouveau schéma à pas de temps local d'ordre trois.

Ce chapitre peut être divisé en deux parties. La première partie est dédiée à la présentation du nouveau schéma à pas de temps local d'ordre deux; la seconde partie présente le nouveau schéma à pas de temps local d'ordre trois.

3.1 Nouveau schéma à pas de temps local d'ordre deux (NRK2)

3.1.1 Construction des méthodes RK_S et RK_F

Le nouveau schéma à pas de temps local que nous proposons est basé sur la méthode RK2, dont le tableau de Butcher est rappelé sur le tableau 3.1.

La méthode RK_F de notre schéma est identique à celle du schéma CSRK2; son tableau de Butcher est le tableau 3.2 ci-après. On rappelle que la méthode RK_F décrite par le tableau de Butcher 3.2 est une méthode à quatre étapes qui représente une double application de la méthode de base (RK2) de pas de temps $\Delta t/2$.

0	0	
1	1	0
	1/2	1/2

TABLE 3.1 – Tableau de Butcher de la méthode de base (RK2).

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	
1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	0
	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$

TABLE 3.2 – Tableau de Butcher de la méthode RK_F .

Le tableau de Butcher 3.2 fait clairement apparaître le vecteur c^F de la méthode RK_F :

$$c^F = [0, \frac{1}{2}, \frac{1}{2}, 1] \quad (3.1)$$

La méthode RK_S de ce nouveau schéma est construite à partir de la méthode de base dont le vecteur c s'écrit :

$$c = [0, 1] \quad (3.2)$$

On souhaite que la méthode RK_S ait le même nombre d'étapes que la méthode RK_F , pour être conforme à la définition des méthodes RKP. De plus, le vecteur c^S de la méthode RK_S doit être égal au vecteur c^F (défini par l'équation (3.1)), afin de satisfaire "automatiquement" les conditions de couplage d'ordre deux (ceci est prouvé dans la partie 3.1.2). La méthode RK_S définie par le tableau de Butcher 3.3 est proposée. Elle répond aux contraintes fixées puisqu'elle contient quatre étapes comme la méthode RK_F , et son vecteur c^S est identique au vecteur c^F . Le tableau de Butcher 3.3 de la méthode RK_S montre qu'elle est constituée de la méthode de base (coefficients noirs) à laquelle s'ajoutent deux valeurs additionnelles (coefficients rouges) calculées au temps $t^n + \Delta t/2$. Les valeurs additionnelles se distinguent des valeurs de "de base" par le fait que les coefficients b qui leur sont associés sont nuls.

On appelle NRK2 le schéma constitué des méthodes RK_S et RK_F définies respectivement par les tableaux de Butcher 3.3 et 3.2. On considère pour l'instant que les méthodes RK_S et RK_F sont appliquées sur l'ensemble des cellules des sous-domaines Slow et Fast.

Remarque 3.1. Dans le tableau de Butcher 3.3 de la méthode RK_S , on remarque que les coefficients a_{ij} utilisés pour calculer les deux valeurs additionnelles sont identiques $(\frac{1}{2}, 0)$. Il s'agit donc d'une seule et même valeur additionnelle, utilisée deux fois.

0	0			
$\frac{1}{2}$	$\frac{1}{2}$	0		
$\frac{1}{2}$	$\frac{1}{2}$	0	0	
1	1	0	0	0
	$\frac{1}{2}$	0	0	$\frac{1}{2}$

TABLE 3.3 – Tableau de Butcher de la méthode RK_S , composée de la méthode de Heun et de valeurs additionnelles (calculées à l’aide des coefficients rouge).

3.1.2 Ordre du schéma NRK2

Proposition 3.1. *Le schéma NRK2 est d’ordre deux.*

Démonstration. Le schéma NRK2 est constitué de la méthode RK_S (tableau de Butcher 3.3) pour intégrer l’ensemble des cellules du sous-domaine Slow, et de la méthode RK_F (tableau de Butcher 3.2) pour intégrer l’ensemble des cellules du sous-domaine Fast. Mathématiquement, cela revient à intégrer le système partitionné suivant :

$$\begin{cases} \frac{dU_S}{dt} = -D_S(U_S, U_F) & \text{résolue avec } RK_S \\ \frac{dU_F}{dt} = -D_F(U_S, U_F) & \text{résolue avec } RK_F \end{cases} \quad (3.3)$$

en utilisant la méthode RK_S pour intégrer la première équation du système et la méthode RK_F pour intégrer la seconde équation de ce système. On se retrouve dans le cas des méthodes RKP, et il faut vérifier les conditions d’ordre deux fournies par le tableau 2.1 (page 60) : les méthodes RK_S et RK_F doivent être d’ordre deux, et les conditions de couplage entre les deux méthodes doivent être satisfaites.

Vérifions que la méthode RK_S est d’ordre deux

Pour cela, on vérifie les conditions d’ordre de la première colonne du tableau 2.1 :

$$\sum_{i=1}^4 b_i^S = \frac{1}{2} + 0 + 0 + \frac{1}{2} = 1$$

$$\sum_{i=1}^4 b_i^S c_i^S = \frac{1}{2} \times 0 + 0 + 0 + \frac{1}{2} \times 1 = \frac{1}{2}$$

Donc la méthode RK_S est bien d’ordre deux.

Vérifions que la méthode RK_F est d’ordre deux

Pour cela, on vérifie les conditions d'ordre de la deuxième colonne du tableau 2.1 :

$$\sum_{i=1}^4 b_i^F = 4 \times \frac{1}{4} = 1$$

$$\sum_{i=1}^4 b_i^F c_i^F = \frac{1}{4} \times 0 + \frac{1}{4} \times \frac{1}{2} + \frac{1}{4} \times \frac{1}{2} + \frac{1}{4}$$

$$= \frac{1}{2}$$

Donc la méthode RK_F est bien d'ordre deux.

Vérifions que les conditions de couplage sont satisfaites (troisième colonne du tableau 2.1)

L'égalité des coefficients c^S et c^F assure que les conditions de couplage sont satisfaites :

$$\sum_{i=1}^4 b_i^S c_i^F = \sum_{i=1}^4 b_i^S c_i^S = \frac{1}{2}, \quad (3.4)$$

De même :

$$\sum_{i=1}^4 b_i^F c_i^S = \sum_{i=1}^4 b_i^F c_i^F = \frac{1}{2}, \quad (3.5)$$

Toutes les conditions d'ordre du tableau 2.1 sont satisfaites donc la méthode RKP constituée de RK_S et RK_F est d'ordre deux. Par conséquent, le schéma constitué des méthodes RK_S et RK_F (c'est à dire le schéma NRK2) est d'ordre deux. \square

3.1.3 Correction de flux pour assurer la conservativité du schéma NRK2

Non conservativité du schéma NRK2

Dans ce paragraphe, on montre que le schéma NRK2 n'est pas conservatif à l'interface entre les sous-domaines Slow et Fast. Pour cette démonstration, le schéma NRK2 est appliqué sur la dernière cellule du sous-domaine Slow (cellule i_0 sur le maillage de la figure 2.5, page 67) et sur la première cellule du sous domaine Fast (cellule i_{0+1} sur le maillage de la figure 2.5), pour la résolution de l'EDP discrétisée en espace :

$$\frac{du_i}{dt} = -\frac{1}{\Delta x_i} D_i(u_{i-1}, u_i, u_{i+1}). \quad (3.6)$$

On rappelle l'expression du bilan des flux sur la cellule i , D_i :

$$D_i(u_{i-1}, u_i, u_{i+1}) = \frac{1}{\Delta x_i} \left(F_{i+\frac{1}{2}}(u_i, u_{i+1}) - F_{i-\frac{1}{2}}(u_{i-1}, u_i) \right) \quad (3.7)$$

Pour alléger la présentation, les notations suivantes sont introduites :

$$D_i^{n,j} = D_i(u_{i-1}^{n,j}, u_i^{n,j}, u_{i+1}^{n,j}), \quad (3.8)$$

$$F_{i+\frac{1}{2}}^{n,j} = F_{i+\frac{1}{2}}(u_i^{n,j}, u_{i+1}^{n,j}), \quad (3.9)$$

où $u_i^{n,j}$ est la j -ième estimation de Runge-Kutta entre t^n et $t^n + \Delta t$ sur la cellule i .

Les solutions numériques sur les cellules i_0 et i_{0+1} sont supposées connues au temps t^n . On les note $u_{i_0}^{n,1}$ et $u_{i_{0+1}}^{n,1}$. D'après le tableau de Butcher 3.2 de la méthode RK_F , la solution numérique sur la cellule $i_0 + 1$ au temps $t^n + \Delta t$ s'écrit :

$$u_{i_{0+1}}^{n+1} = u_{i_{0+1}}^{n,1} - \frac{\Delta t}{4\Delta x_{i_{0+1}}} (D_{i_{0+1}}^{n,1} + D_{i_{0+1}}^{n,2} + D_{i_{0+1}}^{n,3} + D_{i_{0+1}}^{n,4}) \quad (3.10)$$

D'après le tableau de Butcher 3.3 de la méthode RK_S , la solution numérique sur la cellule i_0 au temps $t^n + \Delta t$ s'écrit :

$$u_{i_0}^{n+1} = u_{i_0}^{n,1} - \frac{\Delta t}{2\Delta x_{i_0}} (D_{i_0}^{n,1} + D_{i_0}^{n,4}) \quad (3.11)$$

D'après les relations (3.7) et (3.10), le flux total calculé par la méthode RK_F entre t^n et $t^n + \Delta t$ au niveau de l'interface $x_{i_0+\frac{1}{2}}$ vaut :

$$\phi_{i_0+\frac{1}{2}}^{RK_F} = \frac{1}{4} \left(F_{i_0+\frac{1}{2}}^{n,1} + F_{i_0+\frac{1}{2}}^{n,2} + F_{i_0+\frac{1}{2}}^{n,3} + F_{i_0+\frac{1}{2}}^{n,4} \right) \quad (3.12)$$

De même, d'après les relations (3.7) et (3.11), le flux total calculé par la méthode RK_S entre t^n et $t^n + \Delta t$ au niveau de l'interface $x_{i_0+\frac{1}{2}}$ vaut :

$$\phi_{i_0+\frac{1}{2}}^{RK_S} = \frac{1}{2} \left(F_{i_0+\frac{1}{2}}^{n,1} + F_{i_0+\frac{1}{2}}^{n,4} \right) \quad (3.13)$$

Au niveau de l'interface $x_{i_0+\frac{1}{2}}$, le flux total calculé par la méthode RK_F entre t^n et $t^n + \Delta t$ (relation (3.12)) est différent du flux total calculé par la méthode RK_S (relation (3.13)) durant ce même intervalle de temps. Par conséquent, le schéma NRK2 n'est pas conservatif au niveau de cette interface. Une correction de flux est proposée afin de rendre le schéma conservatif.

Etape de correction de flux

L'étape de correction de flux est réalisée une fois que la solution numérique au temps $t^n + \Delta t$ est calculée sur l'ensemble du domaine de calcul. En utilisant l'équation (3.13), la solution numérique $u_{i_0}^{n+1}$ peut s'écrire :

$$u_{i_0}^{n+1} = u_{i_0}^{n,1} - \frac{\Delta t}{\Delta x_{i_0}} \left[\phi_{i_0+\frac{1}{2}}^{RK_S} - \frac{1}{2}(F_{i_0-\frac{1}{2}}^{n,1} + F_{i_0-\frac{1}{2}}^{n,4}) \right]. \quad (3.14)$$

Cette solution numérique est corrigée dans le but d'obtenir un schéma conservatif. L'expression de la solution numérique corrigée, notée $u_{i_0}^{n+1,c}$ est :

$$u_{i_0}^{n+1,c} = u_{i_0}^{n+1} + \frac{\Delta t}{\Delta x_{i_0}} \left(\phi_{i_0+\frac{1}{2}}^{RK_S} - \phi_{i_0+\frac{1}{2}}^{RK_F} \right) \quad (3.15)$$

En remplaçant $u_{i_0}^{n+1}$ par son expression (3.14), la relation (3.15) se réécrit :

$$u_{i_0}^{n+1,c} = u_{i_0}^{n,1} - \frac{\Delta t}{\Delta x_{i_0}} \left[\phi_{i_0+\frac{1}{2}}^{RK_F} - \frac{1}{2}(F_{i_0-\frac{1}{2}}^{n,1} + F_{i_0-\frac{1}{2}}^{n,4}) \right]. \quad (3.16)$$

Dans l'expression de la solution corrigée (3.16), le flux total calculé par la méthode RK_S à l'interface $x_{i_0+\frac{1}{2}}$ a été remplacé par le flux total calculé par la méthode RK_F à cet interface.

La solution numérique $u_{i_0+1}^{n+1}$ reste inchangée et s'écrit :

$$u_{i_0+1}^{n+1} = u_{i_0+1}^{n,1} - \frac{\Delta t}{\Delta x_{i_0}} \left[\frac{1}{4}(F_{i_0+\frac{3}{2}}^{n,1} + F_{i_0+\frac{3}{2}}^{n,2} + F_{i_0+\frac{3}{2}}^{n,3} + F_{i_0+\frac{3}{2}}^{n,4}) - \phi_{i_0+\frac{1}{2}}^{RK_F} \right] \quad (3.17)$$

Le flux total à l'interface $x_{i_0+\frac{1}{2}}$, $\phi_{i_0+\frac{1}{2}}^{RK_F}$, apparaît maintenant dans l'expression de la solution numérique sur la cellule i_0 (équation (3.16)) et sur la cellule $i_0 + 1$ (équation (3.17)), donc le schéma est conservatif.

Cette méthode de correction de flux a été utilisée par Krivodonova [86], dans le contexte d'une discrétisation spatiale de type Galerkin Discontinu. Une stratégie similaire a également été proposée par Schlegel *et al* [17] : le flux calculé par la méthode RK_F à l'interface $x_{i_0+\frac{1}{2}}$ est imposé pour le calcul de chaque estimation de la méthode RK_S .

Cette étape de correction de flux est optionnelle, l'utilisateur peut choisir, ou non, de l'effectuer. Plusieurs raisons nous ont poussé à rendre cette étape optionnelle :

- Cette étape introduit un surcoût algorithmique puisqu'il faut stocker des flux d'interface et effectuer la correction. Il nous apparaît judicieux de laisser à l'utilisateur le choix de s'affranchir de ce surcoût algorithmique s'il souhaite optimiser son temps de calcul.
- Certaines méthodes spatiales sont par nature non-conservatives, comme par exemple la méthode IBM (Immersed Boundary Method), que nous avons utilisée au chapitre

5. Dans ce cas, nous ne voyons pas l'intérêt de rendre la méthode d'intégration temporelle conservative, sachant que la méthode spatiale ne l'est pas.

3.1.4 Discussion autour de la méthode RK_S

Nous avons considéré jusque-là que la méthode RK_S était appliquée sur l'ensemble des cellules du sous-domaine Slow. Cela nous a permis de nous placer dans le contexte des méthodes RKP pour déterminer l'ordre du schéma. De plus, cela a permis de montrer et de corriger la non conservativité du schéma. On montre dans cette partie que la méthode RK_S peut être remplacée par la méthode de base sur toutes les cellules du sous-domaine Slow, sauf sur celle jouxtant l'interface avec le sous-domaine Fast. Lors de cette démonstration, on considère pour plus de clarté le maillage réduit de la figure 3.1, constitué de trois cellules. La méthode RK_S est appliquée sur les cellules 1 et 2 et la méthode RK_F est appliquée sur la cellule 3. Concernant les solutions numériques sur les cellules fictives (u_0 et u_4), on suppose, sans perte de généralité, que pour tout j , $u_0^{n,j} = u_1^{n,j}$ et $u_4^{n,j} = u_3^{n,j}$.

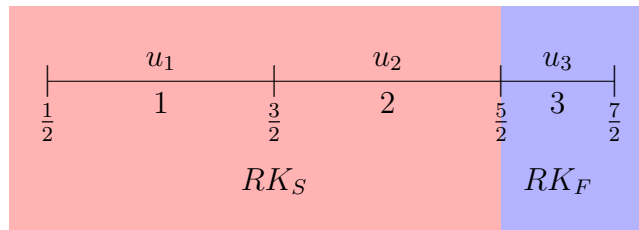


FIGURE 3.1 – Maillage 1D réduit constitué de trois cellules. Chaque cellule i est délimitée par les interfaces $i - \frac{1}{2}$ et $i + \frac{1}{2}$.

Le tableau 3.4 présente les différentes étapes du schéma NRK2 appliqué à la résolution de l'équation :

$$\frac{du_i}{dt} = -D(u_{i-1}, u_i, u_{i+1}), \quad (3.18)$$

sur le maillage réduit de la figure 3.1.

RK_S		RK_F
cellule 1	cellule 2	cellule 3
$D_1^{n,1} = D(u_0^{n,1}, u_1^{n,1}, u_2^{n,1})$ $u_1^{n,2} = u_1^{n,1} - \frac{\Delta t}{2} D_1^{n,1}$	$D_2^{n,1} = D(u_1^{n,1}, u_2^{n,1}, u_3^{n,1})$ $u_2^{n,2} = u_2^{n,1} - \frac{\Delta t}{2} D_2^{n,1}$	$D_3^{n,1} = D(u_2^{n,1}, u_3^{n,1}, u_4^{n,1})$ $u_3^{n,2} = u_3^{n,1} - \frac{\Delta t}{2} D_3^{n,1}$
$u_1^{n,3} = u_1^{n,2}$	$u_2^{n,3} = u_2^{n,2}$	$D_3^{n,2} = D(u_2^{n,2}, u_3^{n,2}, u_4^{n,2})$ $u_3^{n,3} = u_3^{n,1} - \frac{\Delta t}{4} (D_3^{n,1} + D_3^{n,2})$
$u_1^{n,4} = u_1^{n,1} - \Delta t D_1^{n,1}$	$u_2^{n,4} = u_2^{n,1} - \Delta t D_2^{n,1}$	$D_3^{n,3} = D(u_2^{n,3}, u_3^{n,3}, u_4^{n,3})$ $u_3^{n,4} = u_3^{n,3} - \frac{\Delta t}{2} D_3^{n,3}$
$D_1^{n,4} = D(u_0^{n,4}, u_1^{n,4}, u_2^{n,4})$ $u_1^{n+1} = u_1^{n,1} - \frac{\Delta t}{2} (D_1^{n,1} + D_1^{n,4})$	$D_2^{n,4} = D(u_1^{n,4}, u_2^{n,4}, u_3^{n,4})$ $u_2^{n+1} = u_2^{n,1} - \frac{\Delta t}{2} (D_2^{n,1} + D_2^{n,4})$	$D_3^{n,4} = D(u_2^{n,4}, u_3^{n,4}, u_4^{n,4})$ $u_3^{n+1} = u_3^{n,3} - \frac{\Delta t}{4} (D_3^{n,3} + D_3^{n,4})$

TABLE 3.4 – Les différentes étapes du schéma NRK2 sur le maillage réduit de la figure 3.1.

On remarque que les valeurs additionnelles calculées par la méthode RK_S sur la cellule 2 ($u_2^{n,2}$ et $u_2^{n,3}$), sont utilisées pour évaluer deux bilans des flux sur la cellule 3 ($D_3^{n,2}$ et $D_3^{n,3}$). Par contre, les valeurs additionnelles calculées sur la cellule 1 ($u_1^{n,2}$ et $u_1^{n,3}$) ne sont jamais utilisées : elles n'apparaissent dans aucun bilan des flux D . De plus, la solution numérique u_1^{n+1} obtenue avec la méthode RK_S est identique à la solution u_1^{n+1} qui aurait été obtenue en appliquant la méthode de base. Ceci est montré par le tableau 3.5 suivant.

RK_S	Méthode de base	Commentaires
$D_1^{n,1} = D_1(u_0^{n,1}, u_1^{n,1}, u_2^{n,1})$ $u_1^{n,2} = u_1^{n,1} - \frac{\Delta t}{2} D_1^{n,1}$ $u_1^{n,3} = u_1^{n,2}$ $u_1^{n,4} = u_1^{n,1} - \Delta t D_1^{n,1}$ (a)	$D_1^{n,1} = D_1(u_0^{n,1}, u_1^{n,1}, u_2^{n,1})$ $u_1^{n,2} = u_1^{n,1} - \Delta t D_1^{n,1}$ (b)	Les expression (a) et (b) sont égales,
$D_1^{n,4} = D_1(u_0^{n,4}, u_1^{n,4}, u_2^{n,4})$ (c)	$D_1^{n,2} = D_1(u_0^{n,2}, u_1^{n,2}, u_2^{n,2})$ (d)	donc (c) et (d) sont égales.
$u_1^{n+1} = u_1^{n,1} - \frac{\Delta t}{2} (D_1^{n,1} + D_1^{n,4})$	$u_1^{n+1} = u_1^{n,1} - \frac{\Delta t}{2} (D_1^{n,1} + D_1^{n,2})$	Donc u_1^{n+1} est identique pour les 2 méthodes.

TABLE 3.5 – Expression de u_1^{n+1} avec la méthode RK_S et la méthode de base

En conclusion, sur la cellule 1, la méthode RK_S peut être remplacée par la méthode de base : cela ne change rien à l'expression de la solution numérique u_1^{n+1} et cela permet d'économiser le calcul de valeurs additionnelles inutiles. **Sur le maillage complet de la figure 2.5, la méthode RK_S peut être remplacée par la méthode de base sur les cellules 1 à $i_0 - 1$ pour s'affranchir du calcul des valeurs additionnelles, sans modifier l'expression de la solution u_i^{n+1} sur ces cellules.** A partir de maintenant et dans toute la suite de ce mémoire, le schéma NRK2 désigne le schéma constitué, dans le sous-domaine Slow, de la méthode de base et de la méthode RK_S . Le schéma NRK2 constitué de la méthode RK_S dans tout le sous-domaine slow est qualifié de schéma NRK2 équivalent "mathématiquement". La disposition des méthodes d'intégration temporelle pour ces deux schémas est résumée sur la figure 3.2.

Le schéma NRK2 hérite de toutes les propriétés déterminées pour son schéma équivalent mathématiquement : il est d'ordre deux en temps et non-conservatif, mais l'application de la stratégie de correction de flux permet de le rendre conservatif.

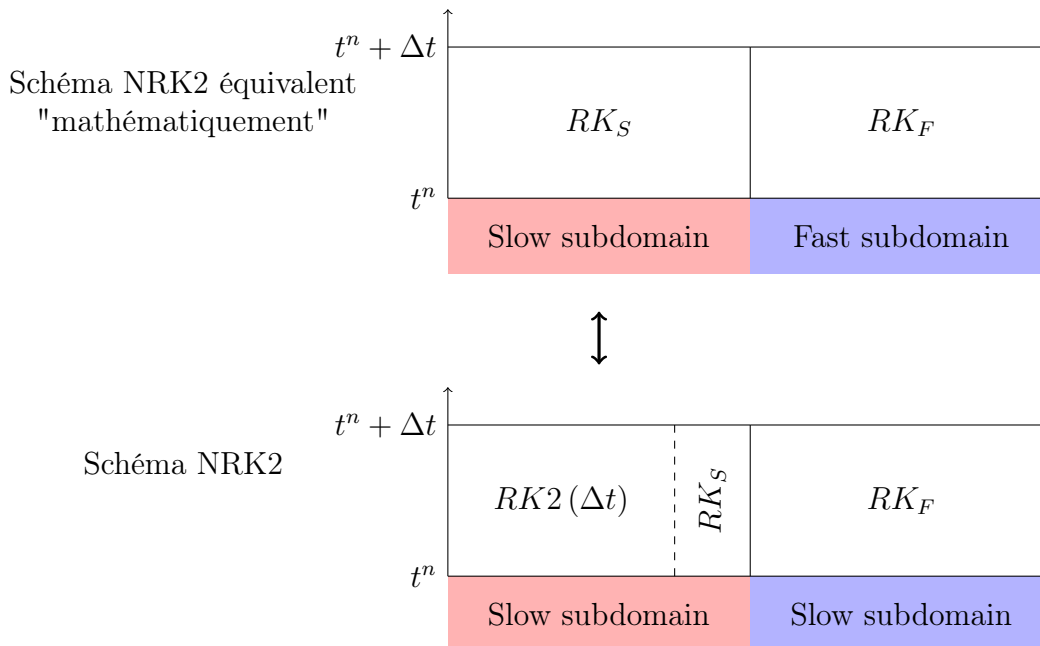


FIGURE 3.2 – Disposition des méthodes d'intégration temporelle pour le schéma NRK2 et son schéma équivalent "mathématiquement". Sur cette figure, la méthode de base est désignée par $RK2(\Delta t)$.

On aborde à présent la question du nombre de cellules sur lequel la méthode RK_S doit être appliquée (c'est à dire la taille de la zone tampon). Dans le cas présenté ici, la méthode RK_S est appliquée sur une seule cellule (la dernière cellule du sous-domaine Slow). En effet,

le stencil de la solution numérique sur la première cellule du sous-domaine Fast "déborde" d'une cellule sur le sous-domaine Slow. Il faut appliquer la méthode RK_S sur un nombre de cellule égal au "débordement".

Le tableau 3.6 présente les différentes étapes du schéma NRK2 sur le maillage complet de la figure 2.5. Le schéma est appliqué à la résolution de l'équation (3.18). Il contient cinq étapes, mais la dernière étape est optionnelle (il s'agit de l'étape de correction de flux effectuée sur la solution numérique de la cellule i_0 pour assurer la conservativité).

	Sous-domaine Slow		Sous-domaine Fast
	$RK2(\Delta t) : 1 \leq i \leq i_0 - 1$	$RK_S : i = i_0$	$RK_F : i_0 + 1 \leq i \leq N$
1	Evaluer $D_i^{n,1}$	Evaluer $D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \frac{\Delta t}{2} D_i^{n,1}$	Evaluer $D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \frac{\Delta t}{2} D_i^{n,1}$
2		$u_i^{n,3} = u_i^{n,2}$	Evaluer $D_i^{n,2}$ $u_i^{n,3} = u_i^{n,1} - \frac{\Delta t}{4} (D_i^{n,1} + D_i^{n,2})$
3	$u_i^{n,2} = u_i^{n,1} - \Delta t D_i^{n,1}$	$u_i^{n,4} = u_i^{n,1} - \Delta t D_i^{n,1}$	Evaluer $D_i^{n,3}$ $u_i^{n,4} = u_i^{n,3} - \frac{\Delta t}{2} D_i^{n,3}$
4	Evaluer $D_i^{n,4}$ $u_i^{n+1} = u_i^{n,1} - \frac{\Delta t}{2} (D_i^{n,1} + D_i^{n,2})$	Evaluer $D_i^{n,4}$ $u_i^{n+1} = u_i^{n,1} - \frac{\Delta t}{2} (D_i^{n,1} + D_i^{n,4})$	Evaluer $D_i^{n,4}$ $u_i^{n+1} = u_i^{n,3} - \frac{\Delta t}{4} (D_i^{n,3} + D_i^{n,4})$
5		Correction de flux	

TABLE 3.6 – Les différentes étapes du schéma NRK2 sur le maillage complet de la figure 2.5.

3.1.5 Estimation du coût algorithmique du schéma NRK2

Le tableau 3.7 montre le nombre de bilans de flux (nombre d'opérations B) et le nombre de valeurs numériques (nombre d'opérations V) calculés par chaque méthode composant le schéma NRK2. Ce tableau montre aussi le nombre total d'opérations B et V effectuées sur l'ensemble du domaine de calcul. On rappelle que les sous-domaine Slow et Fast sont respectivement constitués de N_S et N_F cellules. Le lecteur pourra se reporter à la partie 2.3.2 pour la justification concernant le nombre d'opérations B et V effectuées par les méthodes $RK2(\Delta t)$ et RK_F .

La méthode RK_S est appliquée sur la dernière cellules du sous-domaine Slow. Comme le montre le tableau 3.6, elle nécessite le calcul de deux bilans de flux et de trois valeurs

numériques (la valeur $u_i^{n,3}$ est égale à la valeur $u_i^{n,2}$). Cette méthode effectue donc 2×1 opérations B et 3×1 opérations V .

En utilisant la formule (2.24), le coût algorithmique du schéma NRK2 s'écrit :

$$C_{NRK2} = \underbrace{(2N_S + 4N_F) \times (\eta_B + \eta_V)}_{\text{Coût algorithmique du schéma "idéal"}} + \underbrace{\eta_V}_{\text{Surcoût dû à la zone tampon}} \quad (3.19)$$

	RK2(Δt)	RK_S	RK_F	Total
	$N_S - 1$ cell.	2 cell.	N_F cell.	$N_S + N_F$ cell.
Nbre. d'opérations B	$2 \times (N_S - 1)$	2×1	$4 \times N_F$	$2N_S + 4N_F$
Nbre. d'opérations V	$2 \times (N_S - 1)$	3×1	$4 \times N_F$	$2N_S + 4N_F + 1$

TABLE 3.7 – Nombre d'opérations B et V effectuées par chaque méthode composant le schéma NRK2, et nombre d'opérations B et V total effectuées sur le domaine de calcul.

Remarque 3.2. *L'application de la stratégie de correction de flux nécessite le calcul d'une valeur numérique supplémentaire : il s'agit de la valeur corrigée sur la dernière cellule du sous-domaine Slow. On estime donc que dans ce cas, le coût algorithmique du schéma NRK2 est donné par la relation :*

$$C_{NRK2} = (2N_S + 4N_F) \times (\eta_B + \eta_V) + 2\eta_V. \quad (3.20)$$

A première vue, le coût algorithmique de notre schéma NRK2 est donc plus faible que celui du schéma CSRK2, donné par la relation (2.26). Cela est dû au fait que la zone tampon de notre schéma NRK2 est algorithmiquement un peu moins coûteuse que celle du schéma CSRK2. Néanmoins, lorsque les nombres N_S et N_F de cellules sont grands devant un (ce qui est le cas en général), on a :

$$C_{NRK2} \approx C_{CSRK2} \approx (2N_S + 4N_F) \times (\eta_B + \eta_V),$$

et donc les deux schémas tendent vers le même coût algorithmique.

3.1.6 Différences avec le schémas à pas de temps local de Tang et Warnecke

Tang et Warnecke [21] ont proposé un schéma à pas de temps local basé sur la méthode RK2 et assurant, comme notre schéma NRK2, l'égalité des temps d'estimation. Le schéma de Tang et Warnecke [21] est très similaire au nôtre : il est constitué de la méthode RK2 de pas de temps Δt , d'une méthode RK_S et d'une méthode RK_F . La méthode RK_F du schéma de Tang et Warnecke [21] est identique à celle de notre schéma NRK2 (tableau

de Butcher 3.2). Par contre, leur méthode RK_S est légèrement différente de la nôtre. Son tableau de Butcher (tableau 3.8) et son expression sont donnés ci-dessous.

0	0				
$\frac{1}{2}$	$\frac{1}{2}$	0			$U^{n,2} = U^{n,1} - \frac{\Delta t}{2} D(U^{n,1})$
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0		$U^{n,3} = U^{n,1} - \frac{\Delta t}{4} (D(U^{n,1}) + D(U^{n,2}))$
1	1	0	0	0	$U^{n,4} = U^{n,1} - \Delta t D(U^{n,1})$
$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$U^{n+1} = U^{n,1} - \frac{\Delta t}{4} (D(U^{n,1}) + D(U^{n,4}))$

TABLE 3.8 – Tableau de Butcher de la méthode RK_S du schéma à pas de temps local de Tang et Warnecke [21]

L’expression de la méthode RK_S ci-dessus montre que cette méthode nécessite le calcul de trois bilans de flux D différents et de quatre valeurs numériques différentes. Le tableau (3.9) présente le nombre d’opération B et V effectuées par les différentes méthodes composant le schéma de Tang et Warnecke.

	RK2(Δt)	RK_S	RK_F	total
	$N_S - 1$ cell.	1 cell.	N_F cell.	$N_S + N_F$ cell.
Nbre. d’opérations B	$2 \times (N_S - 1)$	3×1	$4 \times N_F$	$2N_S + 4N_F + 1$
Nbre d’opérations V	$2 \times (N_S - 1)$	4×1	$4 \times N_F$	$2N_S + 4N_F + 2$

TABLE 3.9 – Coût algorithmique de chaque méthode pour le schéma de Tang et Warnecke [21]

En utilisant la formule (2.24), le coût algorithmique du schéma de Tang et Warnecke [21] s’écrit :

$$C_{TangWarnecke} = (2N_S + 4N_F) \times (\eta_B + \eta_V) + \eta_B + 2\eta_V \quad (3.21)$$

Le coût algorithmique de notre schéma NRK2, donné par l’équation (3.19) est donc légèrement plus faible que le coût algorithmique du schéma de Tang et Warnecke.

3.1.7 Généralisation du schéma NRK2 à N pas de temps

Le schéma NRK2 se généralise à N pas de temps en utilisant l’approche récursive décrite dans la partie 2.3.3. Un rapport deux doit être conservé entre les pas de temps de deux zones adjacentes. A cette condition, le schéma NRK2 se généralise à plusieurs pas de temps, comme décrit par la figure 2.9.

3.2 Nouveau schéma à pas de temps local d'ordre trois (NRK3)

Tout comme le schéma NRK2, ce nouveau schéma est basé sur l'égalité des coefficients c^S et c^F . Ce nouveau schéma à pas de temps local d'ordre trois que nous proposons sera noté "NRK3" dans toute la suite de ce mémoire.

3.2.1 Construction des méthodes RK_S et RK_F

La méthode de base utilisée pour mettre en place ce schéma est la méthode RK3 décrite dans la partie 1.3.2. Le tableau de Butcher de cette méthode est rappelé sur le tableau 3.10 suivant.

c	A	\longleftrightarrow	0	0
c_2	a_{21}		c_3	a_{32}
b^T			b_1	$b_2 \quad b_3$

TABLE 3.10 – Tableau de Butcher de la méthode RK3 de base.

La méthode RK_S est construite à partir de la méthode de base. Le vecteur c^S de la méthode RK_S contient donc les coefficients c_i de la méthode de base (regroupés dans le vecteur c suivant) :

$$c = [0, c_2, c_3]. \quad (3.22)$$

La méthode RK_F est construite à partir d'une double application de la méthode de base de pas de temps $\Delta t/2$. Le vecteur c^F de la méthode RK_F contient donc les coefficients c_i correspondant à cette double application de la méthode de base de pas de temps $\Delta t/2$. On rappelle que Constantinescu et Sandu [16] ont montré comment écrire sous la forme d'un tableau de Butcher deux applications successives d'une méthode de pas de temps $\Delta t/2$ (tableau de Butcher 2.4, page 62). En s'appuyant sur le tableau de Butcher 2.4, on peut écrire les coefficients c_i correspondant à la double application de la méthode RK3 de pas de temps $\Delta t/2$ (ces coefficients sont regroupés dans le vecteur \tilde{c} suivant) :

$$\begin{aligned} \tilde{c} &= [c/2, (\mathbf{1} + c)/2] \\ &= [0, \frac{c_2}{2}, \frac{c_3}{2}, \frac{1}{2}, \frac{1+c_2}{2}, \frac{1+c_3}{2}] \end{aligned} \quad (3.23)$$

Le tableau 3.11 montre les coefficients c_i des méthodes RK_S et RK_F (marqués par des cases grises), dans l'hypothèse où la méthode RK_S est uniquement constituée de la méthode de

base et la méthode RK_F est uniquement constituée de deux applications successives de la méthodes de base de pas de temps $\Delta t/2$. Évidemment, dans ce cas, la condition d'égalité des coefficients c_i^S et c_i^F n'est pas vérifiée. Pour satisfaire cette condition d'égalité :

- la méthode RK_S doit fournir cinq valeurs additionnelles correspondant aux cinq coefficients c_i présents dans la méthode RK_F mais absents dans la méthode RK_S (cases rouges du tableau 3.12)
- la méthode RK_F doit fournir deux valeurs additionnelles correspondant aux deux coefficients c_i présents dans la méthode RK_S mais absents dans la méthode RK_F (cases bleues du tableau 3.13)

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
c_2		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.11 – Coefficients c_i des méthodes RK_F et RK_S .

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
c_2		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.12 – Valeurs additionnelles fournies par la méthode RK_S .

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
c_2		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.13 – Valeurs additionnelles fournies par la méthode RK_F .

Suivant le choix de la méthode RK3 de base, il est possible de réduire le nombre de valeurs additionnelles à calculer si par exemple $c_2 = \frac{1}{2}$ ou $c_3 = \frac{1+c_2}{2}$, dans le tableau 3.11. Dans la méthode RK3 de ce mémoire, on a : $c_2 = \frac{1}{2}$ (voir la partie 1.3.2, où sont données les valeurs numériques des différents coefficients de la méthode RK3). Le tableau 3.11 devient le tableau 3.14, dans lequel le coefficient c_2 de la méthode RK_S est maintenant confondu avec le coefficient $\frac{1}{2}$ de la méthode RK_F .

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.14 – Coefficients c_i des méthodes RK_F et RK_S , en remarquant que $c_2 = \frac{1}{2}$.

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.15 – Valeurs additionnelles fournies par la méthode RK_S , en remarquant que $c_2 = \frac{1}{2}$.

$\frac{1+c_3}{2}$		
c_3		
$\frac{1+c_2}{2}$		
$\frac{1}{2}$		
$\frac{c_3}{2}$		
$\frac{c_2}{2}$		
0		
c_i	RK_F	RK_S

TABLE 3.16 – Valeur additionnelles fournies par la méthode RK_F , en remarquant que $c_2 = \frac{1}{2}$.

A présent, pour satisfaire la condition d'égalité entre les coefficients c_i des méthodes RK_F et RK_S :

- la méthode RK_S doit fournir quatre valeurs additionnelles correspondant aux quatre coefficients c_i présents dans la méthode RK_F mais absents dans la méthode RK_S (cases rouges du tableau 3.15). La méthode RK_S doit donc fournir quatre valeurs additionnelles aux quatre temps manquants $t^n + \frac{c_2}{2}\Delta t$, $t^n + \frac{c_3}{2}\Delta t$, $t^n + \frac{1+c_2}{2}\Delta t$ et $t^n + \frac{1+c_3}{2}\Delta t$. Le vecteur c^S de la méthode RK_S s'écrit donc :

$$c^S = \left[0, \frac{c_2}{2}, \frac{c_3}{2}, c_2 = \frac{1}{2}, \frac{1+c_2}{2}, c_3, \frac{1+c_3}{2}\right],$$

où les coefficients des valeurs additionnelles apparaissent en rouge.

- la méthode RK_F doit donc fournir une valeur additionnelle correspondant au coefficient c_i présent dans la méthode RK_S mais absent dans la méthode RK_F (case bleue du tableau 3.16). La méthode RK_F doit fournir une valeur additionnelle au temps $t^n + c_3\Delta t$. Ainsi, le vecteur c^F de la méthode RK_F s'écrit :

$$c^F = \left[0, \frac{c_2}{2}, \frac{c_3}{2}, \frac{1}{2}, \frac{1+c_2}{2}, c_3, \frac{1+c_3}{2}\right],$$

où le coefficient c_3 de la valeur additionnelle apparaît en bleu.

Ainsi, les deux méthodes satisfont la conditions d'égalité des coefficients c_i , puisque $c^S = c^F$. De plus, elles ont le même nombre d'étapes (sept), ce qui permet de se placer dans le cadre des méthodes RKP. Les tableaux de Butcher des méthodes RK_S et RK_F sont les tableaux 3.17 et 3.18. **On appelle NRK3 le schéma constitué des méthodes RK_S (tableau de Butcher 3.17) et RK_F (tableau de Butcher 3.18). On suppose pour l'instant**

que les méthodes RK_S et RK_F sont appliquées sur l'ensemble des sous-domaines Slow et Fast.

0	0	\longleftrightarrow	0	0						
c_2^S	a_{21}^S		$\frac{c_2}{2}$	α_1^S						
·	·		$\frac{c_3}{2}$	α_2^S	α_3^S					
c_7^S	a_{71}^S · a_{76}^S		$c_2 = \frac{1}{2}$	a_{21}	0	0				
·	·		$\frac{1+c_2}{2}$	α_4^S	α_5^S	α_6^S	α_7^S			
c_7^S	a_{71}^S · a_{76}^S		c_3	a_{31}	0	0	a_{32}	0		
	b_1^S · · b_7^S	$\frac{1+c_3}{2}$	α_8^S	α_9^S	α_{10}^S	α_{11}^S	α_{12}^S	α_{13}^S		
	b_1^S · · b_7^S		b_1	0	0	b_2	0	b_3	0	

TABLE 3.17 – Tableau de Butcher de la méthode RK_S .

0	0	\longleftrightarrow	0	0					
c_2^F	a_{21}^F		$\frac{c_2}{2}$	$\frac{a_{21}}{2}$					
·	·		$\frac{c_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$				
c_7^F	a_{71}^F · a_{76}^F		$\frac{1}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$			
·	·		$\frac{1+c_2}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{21}}{2}$		
c_7^F	a_{71}^F · a_{76}^F		c_3	α_1^F	α_2^F	α_3^F	α_4^F	α_5^F	
	b_1^F · · b_7^F	$\frac{1+c_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$	0	
	b_1^F · · b_7^F		$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	0	$\frac{b_3}{2}$

TABLE 3.18 – Tableau de Butcher de la méthode RK_F .

Sur les tableaux de Butcher 3.17 et 3.18, les coefficients "inconnus" α^S et α^F permettant le calcul des valeurs additionnelles sont déterminés de manière à satisfaire les conditions d'ordre trois.

3.2.2 Détermination des coefficients α^S et α^F pour assurer l'ordre trois

Proposition 3.2. *Le schéma NRK3 est d'ordre trois si :*

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^F c_j^F = \frac{1}{6} \quad (3.24)$$

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^S c_j^S = \frac{1}{6} \quad (3.25)$$

Démonstration. Les méthodes RK_S et RK_F sont appliquées dans les sous-domaines Slow et Fast. Mathématiquement, cela revient à résoudre le système partitionné suivant :

$$\begin{cases} \frac{dU_S}{dt} = -D_S(U_S, U_F) & \text{résolue avec } RK_S \\ \frac{dU_F}{dt} = -D_F(U_S, U_F) & \text{résolue avec } RK_F \end{cases} \quad (3.26)$$

en utilisant la méthode RK_S pour intégrer la première équation et la méthode RK_F pour intégrer la seconde. On se retrouve dans le cadre de la théorie des méthodes RKP. Pour démontrer la proposition, on vérifie les différentes conditions d'ordre trois fournies dans le tableau 2.2 (page 60). On rappelle que la méthode RK3 est d'ordre trois donc, d'après le tableau 1.2 (page 26), ses coefficients vérifient les relations suivantes :

$$\sum_{i=1}^3 b_i = 1 \quad (3.27)$$

$$\sum_{i=1}^3 b_i c_i = \frac{1}{2} \quad (3.28)$$

$$\sum_{i=1}^3 b_i (c_i)^2 = \frac{1}{3} \quad (3.29)$$

$$\sum_{i=1}^3 \sum_{j=1}^3 b_i a_{ij} c_j = \frac{1}{6} \quad (3.30)$$

Vérifions que la méthode RK_S est d'ordre trois (première colonne du tableau 2.2).

Le tableau 3.17 montre la correspondance entre les coefficients (a_{ij}^S, b_i^S, c_i^S) et les coefficients (a_{ij}^S, b_i, c_i) .

$$\sum_{i=1}^7 b_i^S = \sum_{i=1}^3 b_i = 1, \quad \text{d'après l'eq. (3.27)}$$

$$\sum_{i=1}^7 b_i^S c_i^S = \sum_{i=1}^3 b_i c_i = \frac{1}{2}, \quad \text{d'après l'eq. (3.28)}$$

$$\sum_{i=1}^7 b_i^S (c_i^S)^2 = \sum_{i=1}^3 b_i (c_i)^2 = \frac{1}{3}, \quad \text{d'après l'eq. (3.29)}$$

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^S c_j^S = \sum_{i=1}^3 \sum_{j=1}^3 b_i a_{ij} c_j = \frac{1}{6}, \quad \text{d'après l'eq. (3.30)}$$

Donc la méthode RK_S est bien d'ordre trois.

Vérifions que la méthode RK_F est d'ordre trois (deuxième colonne du tableau 2.2) :

Le tableau 3.18 montre la correspondance entre les coefficients (a_{ij}^F, b_i^F, c_i^F) et les coefficients $(a_{ij} \alpha_k^F, b_i, c_i)$.

$$\sum_{i=1}^7 b_i^F = 2 \sum_{i=1}^3 \frac{b_i}{2} = 1, \quad \text{d'après l'eq. (3.27)}$$

$$\begin{aligned} \sum_{i=1}^7 b_i^F c_i^F &= \sum_{i=1}^3 \left(\frac{b_i c_i}{2 \cdot 2} + \frac{b_i}{2} \frac{1 + c_i}{2} \right) \\ &= \sum_{i=1}^3 \left(\frac{b_i c_i}{2 \cdot 2} + \frac{b_i c_i}{2 \cdot 2} + \frac{b_i}{4} \right) \\ &= \frac{1}{8} + \frac{1}{8} + \frac{1}{4} \quad \text{d'après les eq. (3.27) et (3.28)} \\ &= \frac{1}{2} \end{aligned}$$

$$\begin{aligned} \sum_{i=1}^7 b_i^F (c_i^F)^2 &= \sum_{i=1}^3 \left(\frac{b_i}{2} \left(\frac{c_i}{2} \right)^2 + \frac{b_i}{2} \left(\frac{1 + c_i}{2} \right)^2 \right) \\ &= \sum_{i=1}^3 \left(\frac{b_i c_i^2}{8} + \frac{b_i (1 + 2c_i + c_i^2)}{8} \right) \\ &= \sum_{i=1}^3 \left(\frac{b_i c_i^2}{8} + \frac{b_i}{8} + \frac{b_i c_i}{4} + \frac{b_i c_i^2}{8} \right) \\ &= \frac{1}{24} + \frac{1}{8} + \frac{1}{8} + \frac{1}{24}, \quad \text{d'après les eq. (3.27), (3.28) et (3.29)} \\ &= \frac{1}{3} \end{aligned}$$

$$\begin{aligned}
\sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^F c_j^F &= \sum_{i=1}^3 \sum_{j=1}^3 \frac{b_i a_{ij} c_j}{2 \cdot 2 \cdot 2} + \sum_{i=1}^3 \sum_{j=1}^3 \frac{b_i b_j c_j}{2 \cdot 2 \cdot 2} + \sum_{i=1}^3 \sum_{j=1}^3 \frac{b_i a_{ij} (1 + c_j)}{2 \cdot 2 \cdot 2} \\
&= \frac{1}{48} + \sum_{i=1}^3 \frac{b_i}{2} \frac{1}{8} + \sum_{i=1}^3 \frac{b_i c_i}{2} \frac{1}{4} + \sum_{i=1}^3 \sum_{j=1}^3 \frac{b_i a_{ij} c_j}{2 \cdot 2 \cdot 2} \\
&= \frac{1}{48} + \frac{1}{16} + \frac{1}{16} + \frac{1}{48}, \quad \text{d'après les eq. (3.27), (3.28) et (3.30)} \\
&= \frac{1}{6}
\end{aligned}$$

Donc la méthode RK_F est bien d'ordre trois.

Concernant les relations de couplage (troisième colonne du tableau 2.2) :

L'égalité entre chaque coefficient c_i^S et c_i^F ($1 \leq i \leq 7$) assure que les conditions de couplage (1), (4), (5), (6), (9) et (10) du tableau 2.2 sont "automatiquement" vérifiées. En effet, dans chacune de ces relations, les coefficients c_i^S peuvent être remplacés par les coefficients c_i^F (et vice versa) ce qui permet de se ramener à une condition d'ordre trois, vérifiée, pour RK_F (ou RK_S).

Les quatre relations de couplage non vérifiées sont les relations (2), (3), (7) et (8). Mais parmi ces quatre relations, certaines sont équivalentes, grâce à l'égalité entre les coefficients c_i^S et c_i^F :

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^F c_j^S = \sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^F c_j^F \quad (\text{équivalence entre les relations (2) et (3)})$$

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^S c_j^F = \sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^S c_j^S \quad (\text{équivalence entre les relations (7) et (8)})$$

Par conséquent, toutes les relations d'ordre trois fournies dans le tableau 2.2 sont vérifiées si :

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^F c_j^F = \frac{1}{6},$$

$$\sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^S c_j^S = \frac{1}{6},$$

et dans ce cas, la méthode partitionnée constituée de RK_S et RK_F est d'ordre 3. \square

Remarque 3.3. *La stratégie de Constantinescu et Sandu, qui impose l'égalité des coefficients b_i^S et b_i^F , permet de satisfaire "automatiquement" quatre conditions de couplage de la méthode RKP d'ordre trois. Il s'agit des relations (1), (3), (6) et (7) du tableau 2.2. Mais la stratégie de Constantinescu et Sandu, contrairement à la stratégie présentée dans cette partie, n'introduit pas les degrés de liberté supplémentaires qui permettent de satisfaire les conditions de couplages restantes. Ceci explique pourquoi la stratégie présentée dans cette partie, qui consiste à imposer l'égalité des coefficients c_i^S et c_i^F , est à l'heure actuelle la mieux adaptée pour construire des schémas à pas de temps local d'ordre supérieur à deux.*

D'après la proposition 3.2 le schéma NRK3 est d'ordre trois si les relations (3.24) et (3.25) sont satisfaites. A ces relations s'ajoute la relation de consistance (1.44) de la page 26 qui doit être satisfaite par les coefficients α_i de chaque valeur additionnelle. Toutes ces relations sont rassemblées dans le système suivant :

$$\left\{ \begin{array}{l} \sum_{i=1}^7 \sum_{j=1}^7 b_i^S a_{ij}^F c_j^F = \frac{1}{6} \\ \sum_{i=1}^7 \sum_{j=1}^7 b_i^F a_{ij}^S c_j^S = \frac{1}{6} \\ \frac{c_2}{2} = \alpha_1^S \\ \frac{c_3}{2} = \alpha_2^S + \alpha_3^S \\ \frac{1+c_2}{2} = \alpha_4^S + \alpha_5^S + \alpha_6^S + \alpha_7^S \\ \frac{1+c_3}{2} = \alpha_8^S + \alpha_9^S + \alpha_{10}^S + \alpha_{11}^S + \alpha_{12}^S + \alpha_{13}^S \\ c_3 = \alpha_1^F + \alpha_2^F + \alpha_3^F + \alpha_4^F + \alpha_5^F \end{array} \right. \quad (3.31)$$

On développe les deux premières relations du système (3.31) pour faire apparaître les coefficients α_i^F et α_i^S :

$$\left\{ \begin{array}{l}
b_2 \left(\frac{b_2 c_2}{2} + \frac{b_3 c_3}{2} \right) + b_3 \left(\alpha_2^F \frac{c_2}{2} + \alpha_3^F \frac{c_3}{2} + \alpha_4^F \frac{1}{2} + \alpha_5^F \frac{1+c_2}{2} \right) = \frac{1}{6} \\
\frac{b_3}{2} \alpha_3^S \frac{c_2}{2} + \frac{b_2}{2} \left(\alpha_5^S \frac{c_2}{2} + \alpha_6^S \frac{c_3}{2} + \alpha_7^S \frac{1}{2} \right) + \\
\frac{b_3}{2} \left(\alpha_9^S \frac{c_2}{2} + \alpha_{10}^S \frac{c_3}{2} + \alpha_{11}^S \frac{1}{2} + \alpha_{12}^S \frac{1+c_2}{2} + \alpha_{13}^S c_3 \right) = \frac{1}{6} \\
\frac{c_2}{2} = \alpha_1^S \\
\frac{c_3}{2} = \alpha_2^S + \alpha_3^S \\
\frac{1+c_2}{2} = \alpha_4^S + \alpha_5^S + \alpha_6^S + \alpha_7^S \\
\frac{1+c_3}{2} = \alpha_8^S + \alpha_9^S + \alpha_{10}^S + \alpha_{11}^S + \alpha_{12}^S + \alpha_{13}^S \\
c_3 = \alpha_1^F + \alpha_2^F + \alpha_3^F + \alpha_4^F + \alpha_5^F
\end{array} \right. \quad (3.32)$$

Le système (3.32) est largement sous-déterminé. En effet, seulement deux équations (la première et la dernière) portent sur les cinq coefficients inconnus α^F . De même, seulement cinq équations (les autres) portent sur les treize coefficients inconnus α^S . Il est donc possible de fixer arbitrairement trois coefficients inconnus α^F et huit coefficients inconnus α^S .

Choix arbitraire de huit coefficients inconnus α^S

Considérons l'équation discrétisée en espace suivante :

$$\frac{d}{dt} u_i = -D_i(u_{i-1}, u_i, u_{i+1}). \quad (3.33)$$

Les huit coefficients inconnus α^S sont fixés afin de ne pas avoir à évaluer le second membre D_i de l'équation (3.33) au niveau des valeurs additionnelles.

Nous donnons un exemple pour clarifier cette démarche. D'après le tableau de Butcher 3.17 de la méthode RK_S , l'expression de la seconde valeur additionnelle est donnée par :

$$u_i^{n,3} = u_i^{n,1} - \Delta t \left(\alpha_2^S D_i(u_{i-1}^{n,1}, u_i^{n,1}, u_{i+1}^{n,1}) + \alpha_3^S D_i(u_{i-1}^{n,2}, u_i^{n,2}, u_{i+1}^{n,2}) \right). \quad (3.34)$$

Or, cette expression nécessite d'évaluer le second membre D_i au niveau des valeurs additionnelles $u_{i-1}^{n,2}$, $u_i^{n,2}$ et $u_{i+1}^{n,2}$. On fixe donc α_3^S à zéro pour ne pas avoir à évaluer le second membre au niveau de ces valeurs additionnelles, et n'utiliser que l'évaluation du second membre au niveau de la solution au temps t^n .

En suivant cette démarche pour l'ensemble des coefficients inconnus α^S , nous fixons :

$$\alpha_3^S = \alpha_5^S = \alpha_6^S = \alpha_7^S = \alpha_9^S = \alpha_{10}^S = \alpha_{12}^S = \alpha_{13}^S = 0$$

Choix arbitraire de trois coefficients inconnus α^F

Concernant le tableau de Butcher 3.18 de la méthode RK_F , nous avons choisi de fixer :

$$\alpha_3^F = \alpha_4^F = \alpha_5^F = 0$$

Résolution du système (3.32)

Après avoir fixé arbitrairement les huit coefficients inconnus α^S et les trois coefficients inconnus α^F à zéro, le système suivant est obtenu :

$$\left\{ \begin{array}{l} b_2 \left(\frac{b_2}{2} \frac{c_2}{2} + \frac{b_3}{2} \frac{c_3}{2} \right) + b_3 \alpha_2^F \frac{c_2}{2} = \frac{1}{6} \\ \frac{b_3}{2} \alpha_{11}^S \frac{1}{2} = \frac{1}{6} \\ \frac{c_2}{2} = \alpha_1^S \\ \frac{c_3}{2} = \alpha_2^S \\ \frac{1+c_2}{2} = \alpha_4^S \\ \frac{1+c_3}{2} = \alpha_8^S + \alpha_{11}^S \\ c_3 = \alpha_1^F + \alpha_2^F \end{array} \right. \quad (3.35)$$

En sachant que :

- $\frac{b_2}{2} \frac{c_2}{2} + \frac{b_3}{2} \frac{c_3}{2} = \frac{1}{8}$ (cette égalité provient de la relation d'ordre trois $\sum_{i=1}^3 b_i c_i = \frac{1}{2}$),
- $c_2 = \frac{1}{2}$,

on obtient le système résolu suivant :

$$\left\{ \begin{array}{l} \alpha_2^F = \left(\frac{2}{3} - \frac{b_2}{2}\right) \frac{1}{b_3} \approx 1.2440 \\ \alpha_1^F = c_3 - \alpha_2^F \approx -0.45534 \\ \alpha_1^S = \frac{c_2}{2} \\ \alpha_2^S = \frac{c_3}{2} \\ \alpha_4^S = \frac{1+c_2}{2} \\ \alpha_{11}^S = \frac{2}{3b_3} \approx 1.821 \\ \alpha_8^S = \frac{1+c_3}{2} - \frac{2}{3b_3} \approx -0.9270 \end{array} \right. \quad (3.36)$$

Les tableaux de Butcher 3.19 et 3.20 sont les tableaux de Butcher "finaux" des méthodes RK_S et RK_F . Ils font seulement apparaître les coefficients α^S et α^F non-nuls ; l'expression de ces coefficients est fournie dans le système (3.36).

0	0					
$\frac{c_2}{2}$	α_1^S					
$\frac{c_3}{2}$	α_2^S	0				
$\frac{1}{2}$	a_{21}	0	0			
$\frac{1+c_2}{2}$	α_4^S	0	0	0		
c_3	a_{31}	0	0	a_{32}	0	
$\frac{1+c_3}{2}$	α_8^S	0	0	α_{11}^S	0	0
	b_1	0	0	b_2	0	b_3

TABLE 3.19 – Tableau de Butcher final de la méthode RK_S

0	0						
$\frac{c_2}{2}$	$\frac{a_{21}}{2}$						
$\frac{c_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$					
$\frac{1}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$				
$\frac{1+c_2}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{21}}{2}$			
c_3	α_1^F	α_2^F	0	0	0		
$\frac{1+c_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{a_{31}}{2}$	$\frac{a_{32}}{2}$	0	
	$\frac{b_1}{2}$	$\frac{b_2}{2}$	$\frac{b_3}{2}$	$\frac{b_1}{2}$	$\frac{b_2}{2}$	0	$\frac{b_3}{2}$

TABLE 3.20 – Tableau de Butcher final de la méthode RK_F

Précisons que les conditions d'ordre trois sont vérifiées à l'issue des sept étapes des méthodes RK_S et RK_F (et donc à l'issue des deux applications de la méthode RK3 de pas de temps $\Delta t/2$). Par conséquent, la première application la méthode RK3 de pas de temps $\Delta t/2$ n'est pas d'ordre trois.

3.2.3 Correction de flux pour assurer la conservativité du schéma NRK3

A l'image du schéma NRK2, le schéma NRK3 n'est pas conservatif. Pour rendre le schéma NRK3 conservatif, la même stratégie de correction de flux que celle présentée pour le schéma NRK2 est utilisée. Nous la représentons, un peu plus brièvement, dans le contexte de ce schéma NRK3.

La méthode RK_F est appliquée à la résolution de l'équation (3.6) sur la cellule $i_0 + 1$, qui est la première cellule du sous-domaine Fast, d'après le maillage de la figure 2.5. Les notations simplificatrices fournies par la relation (3.8) sont utilisées. D'après le tableau de Butcher 3.2 de la méthode RK_F , l'expression de la solution numérique au temps $t^n + \Delta t$ sur cette cellule est donnée par :

$$u_{i_0+1}^{n+1} = u_{i_0+1}^{n,1} - \frac{\Delta t}{\Delta x_{i_0+1}} \left(\frac{b_1}{2} D_{i_0+1}^{n,1} + \frac{b_2}{2} D_{i_0+1}^{n,2} + \frac{b_3}{2} D_{i_0+1}^{n,3} + \frac{b_1}{2} D_{i_0+1}^{n,4} + \frac{b_2}{2} D_{i_0+1}^{n,5} + \frac{b_3}{2} D_{i_0+1}^{n,7} \right). \quad (3.37)$$

La relation (3.37) permet de déduire le flux calculé par la méthode RK_F à l'interface $x_{i_0+\frac{1}{2}}$ entre t^n et $t^n + \Delta t$:

$$\Phi_{i_0+\frac{1}{2}}^{RK_F} = \frac{b_1}{2} F_{i_0+\frac{1}{2}}^{n,1} + \frac{b_2}{2} F_{i_0+\frac{1}{2}}^{n,2} + \frac{b_3}{2} F_{i_0+\frac{1}{2}}^{n,3} + \frac{b_1}{2} F_{i_0+\frac{1}{2}}^{n,4} + \frac{b_2}{2} F_{i_0+\frac{1}{2}}^{n,5} + \frac{b_3}{2} F_{i_0+\frac{1}{2}}^{n,7}. \quad (3.38)$$

La méthode RK_S est appliquée à la résolution de l'équation (3.6) sur la cellule i_0 , qui est la dernière cellule du sous-domaine Slow, d'après le maillage de la figure 2.5. D'après le tableau de Butcher 3.3 de la méthode RK_S , l'expression de la solution numérique sur cette cellule au temps $t^n + \Delta t$ est donnée par :

$$u_{i_0}^{n+1} = u_{i_0}^{n,1} - \frac{\Delta t}{\Delta x_{i_0}} (b_1 D_{i_0}^{n,1} + b_2 D_{i_0}^{n,4} + b_3 D_{i_0}^{n,6}) \quad (3.39)$$

La relation (3.39) permet de déterminer le flux calculé par la méthode RK_S à l'interface $x_{i_0+\frac{1}{2}}$ entre t^n et $t^n + \Delta t$:

$$\Phi_{i_0+\frac{1}{2}}^{RK_S} = b_1 F_{i_0+\frac{1}{2}}^{n,1} + b_2 F_{i_0+\frac{1}{2}}^{n,4} + b_3 F_{i_0+\frac{1}{2}}^{n,6} \quad (3.40)$$

A l'interface $x_{i_0+\frac{1}{2}}$, le flux calculé par la méthode RK_S entre t^n et $t^n + \Delta t$ (relation (3.40)) est différent du flux calculé par la méthode RK_F (relation (3.38)) dans ce même intervalle de temps. Par conséquent, le schéma NRK3 n'est pas conservatif à cette interface.

Comme pour le schéma NRK2, l'étape de correction de flux est effectuée une fois que les solutions numériques sont calculées au temps $t^n + \Delta t$ sur les deux sous-domaines. La solution numérique sur la cellule i_0 (obtenue avec la méthode RK_S) est corrigée pour rendre conservatif le schéma NRK3. L'expression de la solution numérique corrigée est :

$$u_{i_0}^{n+1,c} = u_{i_0}^{n+1} + \frac{\Delta t}{\Delta x_{i_0}} \left(\Phi_{i_0+\frac{1}{2}}^{RK_S} - \Phi_{i_0+\frac{1}{2}}^{RK_F} \right) \quad (3.41)$$

En utilisant les relations (3.39) et (3.41), ainsi que la relation de correspondance entre bilans de flux et flux aux interfaces (3.7), la solution corrigée $u_{i_0}^{n+1,c}$ est réécrite :

$$u_{i_0}^{n+1,c} = u_{i_0}^{n,1} - \frac{\Delta t}{\Delta x_{i_0}} \left(\Phi_{i_0+\frac{1}{2}}^{RK_F} - b_1 F_{i_0-\frac{1}{2}}^{n,1} + b_2 F_{i_0-\frac{1}{2}}^{n,4} + b_3 F_{i_0-\frac{1}{2}}^{n,6} \right) \quad (3.42)$$

Dans l'expression de la solution corrigée (3.42), le flux à l'interface $x_{i_0+\frac{1}{2}}$ est le flux calculé par la méthode RK_F à cette interface : le schéma NRK3 est maintenant conservatif.

3.2.4 Discussions autour des méthodes RK_S et RK_F

Comme évoqué dans la partie 3.2.1, nous avons jusque-là considéré que le schéma NRK3 était constitué des méthodes RK_S et RK_F appliquées sur l'ensemble des cellules des sous-domaines Slow et Fast. Ceci nous a permis de nous placer dans le contexte des méthodes RKP, pour déterminer les conditions d'ordre trois du schéma NRK3. Ceci nous a également permis de montrer et de corriger la non-conservativité du schéma NRK3.

Dans cette partie, on montre que les méthodes RK_S et RK_F peuvent se simplifier en dehors des cellules i_0 et $i_0 + 1$. Le raisonnement est similaire au cas du schéma NRK2.

On reprend la figure 2.8 (page 73) qui montre le stencil des solutions numériques u_{i_0} et u_{i_0+1} . Le stencil de la solution numérique u_{i_0+1} "déborde" sur la dernière cellule du sous-domaine Slow. Par conséquent, il est nécessaire d'appliquer la méthode RK_S sur la dernière cellule du sous-domaine Slow, en particulier pour fournir toutes les valeurs additionnelles permettant de faire la transition avec la méthode RK_F . De même, il est impératif d'appliquer la méthode RK_F sur la première cellule du sous-domaine Fast, en particulier pour fournir la valeur additionnelle qui permet de faire la transition avec la méthode RK_S .

Par contre, le calcul des valeurs additionnelles sur les cellules 1 à $i_0 - 1$ du sous-domaine Slow est inutile. Sur ces cellules, la méthode RK_S peut être remplacée par la méthode de base. Comme montré par le tableau 3.21, l'expression de u_i^{n+1} n'est pas affectée par ce changement. Le remplacement de la méthode RK_S par la méthode de base permet d'économiser le calcul de valeurs additionnelles inutiles.

RK_S	Méthode de base (RK3)	Commentaires
$u_i^{n,2} = u_i^{n,1} - \Delta t \alpha_1^S D_i^{n,1}$ $u_i^{n,3} = u_i^{n,1} - \Delta t \alpha_2^S D_i^{n,1}$ $u_i^{n,4} = u_i^{n,1} - \Delta t a_{21} D_i^{n,1}$ (a)	$u_i^{n,2} = u_i^{n,1} - \Delta t a_{21} D_i^{n,1}$ (b)	(a) et (b) sont égales,
$u_i^{n,5} = u_i^{n,1} - \Delta t \alpha_4^S D_i^{n,1}$ $u_i^{n,6} = u_i^{n,1} - \Delta t(a_{31} D_i^{n,1} + a_{32} D_i^{n,4})$ (c)	$u_i^{n,3} = u_i^{n,1} - \Delta t(a_{31} D_i^{n,1} + a_{32} D_i^{n,2})$ (d)	donc (c) et (d) sont égales,
$u_i^{n,7} = u_i^{n,1} - \Delta t(\alpha_8^S D_i^{n,1} + \alpha_{11}^S D_i^{n,4})$		
$u_i^{n+1} = u_i^{n,1} - \Delta t(b_1 D_i^{n,1} + b_2 D_i^{n,4} + b_3 D_i^{n,6})$	$u_i^{n+1} = u_i^{n,1} - \Delta t(b_1 D_i^{n,1} + b_2 D_i^{n,2} + b_3 D_i^{n,3})$	donc u_i^{n+1} est identique pour les 2 méthodes.

TABLE 3.21 – Expression de u_i^{n+1} avec la méthode RK_S et la méthode de base. Les différences entre ces expressions résident dans le nom donné aux différentes estimations.

Le même raisonnement est appliqué dans le sous-domaine Fast. Le calcul des valeurs additionnelles sur les cellules $i_0 + 2$ à N est inutile. Dans cette zone, on peut remplacer la méthode RK_F par deux applications successives de la méthode de base de pas de temps $\Delta t/2$, sans affecter l'expression de la solution u_i^{n+1} sur ces cellules. Cela permet de s'affranchir du calcul de la valeur additionnelle inutile.

A partir de maintenant et dans toute la suite de ce mémoire, le schéma NRK3 désigne le schéma constitué :

- dans le sous-domaine Slow, de la méthode de base et de la méthode RK_S ,
- dans le sous-domaine Fast, de la méthode RK_F et de deux applications successives de la méthode de base de pas de temps $\Delta t/2$.

Le schéma constitué de la méthode RK_S dans tout le sous-domaine Slow et de la méthode RK_F dans tout le sous-domaine Fast est qualifié de schéma NRK3 équivalent "mathématiquement". La figure 3.3 résume la disposition des méthodes d'intégration temporelle pour le schéma NRK3 et son schéma équivalent mathématiquement.

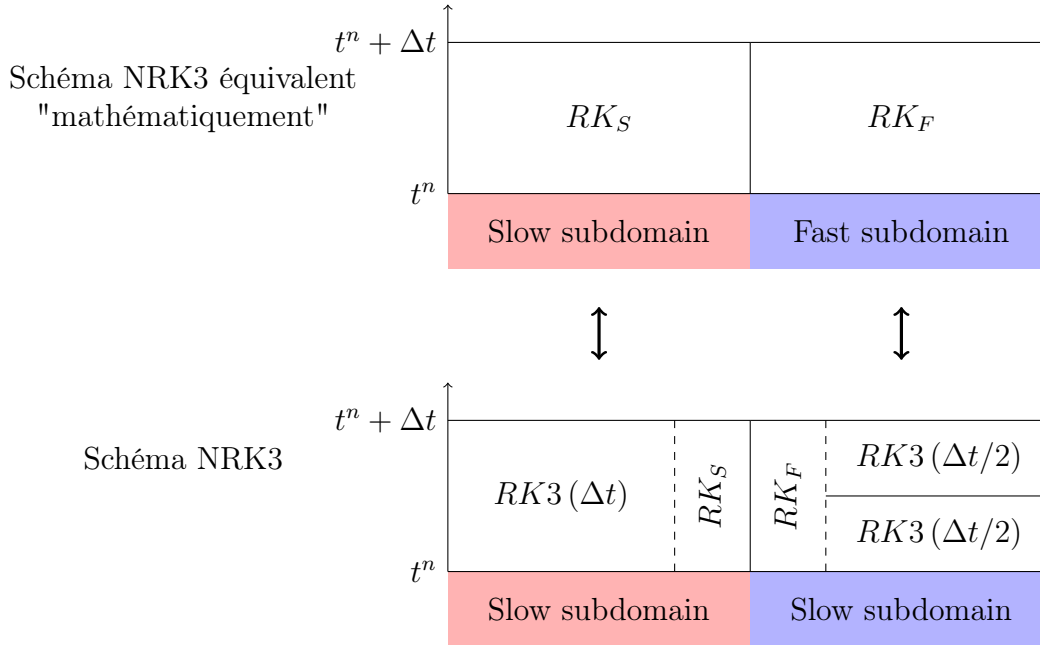


FIGURE 3.3 – Disposition des méthodes d’intégration temporelle pour le schéma NRK3 et son schéma équivalent "mathématiquement". La méthode de base de pas de temps Δt est notée $RK3(\Delta t)$, la méthode de base de pas de temps $\Delta t/2$ est désignée par $RK3(\Delta t/2)$.

Le schéma NRK3 hérite de toutes les propriétés déterminées pour son schéma équivalent mathématiquement : il est d’ordre trois en temps et non-conservatif, mais l’application de la stratégie de correction de flux permet de le rendre conservatif.

Ce schéma contient deux zones tampon : la première à la fin du sous-domaine Slow et la seconde au début du sous-domaine Fast. Comme pour le schéma NRK2, la taille des zones tampon est égale aux "débordements" des stencils.

Le schéma NRK3 se décompose en huit étapes, présentées dans le tableau 3.22 ; la dernière, optionnelle, est l’étape de correction de flux réalisée sur la solution numérique $u_{i_0}^{n+1}$, pour assurer la conservativité du schéma.

Le schéma NRK3 peut être écrit de manière équivalente en utilisant la forme "low storage" de la méthode RK3. On rappelle que la méthode RK3 programmée sous sa forme "low storage" permet de réduire de coût mémoire par rapport à sa programmation sous forme classique. Le tableau C.1 en annexe C présente les différentes étapes du schéma NRK3 avec la méthode RK3 écrite sous sa forme "low storage".

Sous-domaine Slow		Sous-domaine Fast	
	$RK3 (\Delta t) : 1 \leq i \leq i_0 - 1$	$RK_S : i = i_0$	$RK_F : i = i_0 + 1$
1	Evaluer $D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \alpha_1^S \Delta t D_i^{n,1}$	Evaluer $D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \Delta t \frac{a_{21}}{2} D_i^{n,1}$	Evaluer $D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \Delta t \frac{a_{21}}{2} D_i^{n,1}$
2	$u_i^{n,3} = u_i^{n,1} - \alpha_2^S \Delta t D_i^{n,1}$	$u_i^{n,3} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,1} + \frac{a_{32}}{2} D_i^{n,2} \right)$	$u_i^{n,3} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,1} + \frac{a_{32}}{2} D_i^{n,2} \right)$
3	$u_i^{n,4} = u_i^{n,1} - a_{21} \Delta t D_i^{n,1}$ Evaluer $D_i^{n,2}$	$u_i^{n,4} = u_i^{n,1} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + \frac{b_2}{2} D_i^{n,2} + \frac{b_3}{2} D_i^{n,3} \right)$ Evaluer $D_i^{n,4}$	$u_i^{n,4} = u_i^{n,1} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + \frac{b_2}{2} D_i^{n,2} + \frac{b_3}{2} D_i^{n,3} \right)$ Evaluer $D_i^{n,4}$
4	$u_i^{n,5} = u_i^{n,1} - \alpha_4^S \Delta t D_i^{n,1}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{a_{21}}{2} D_i^{n,4}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{a_{21}}{2} D_i^{n,4}$
5	$u_i^{n,6} = u_i^{n,1} - \Delta t (a_{31} D_i^{n,1} + a_{32} D_i^{n,2})$ Evaluer $D_i^{n,3}$	$u_i^{n,6} = u_i^{n,1} - \Delta t (\alpha_1^S D_i^{n,1} + \alpha_2^S D_i^{n,2})$	
6	$u_i^{n,7} = u_i^{n,1} - \Delta t (\alpha_8^S D_i^{n,1} + \alpha_{11}^S D_i^{n,4})$	$u_i^{n,7} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,4} + \frac{a_{32}}{2} D_i^{n,5} \right)$ Evaluer $D_i^{n,7}$	$u_i^{n,6} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,4} + \frac{a_{32}}{2} D_i^{n,5} \right)$ Evaluer $D_i^{n,6}$
7	$u_i^{n+1} = u_i^{n,1} - \Delta t (b_1 D_i^{n,1} + b_2 D_i^{n,4} + b_3 D_i^{n,6})$	$u_i^{n+1} = u_i^{n,4} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + b_2 D_i^{n,5} + \frac{b_3}{2} D_i^{n,7} \right)$	$u_i^{n+1} = u_i^{n,4} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + b_2 D_i^{n,4} + b_3 D_i^{n,6} \right)$
8	Correction de flux		

TABLE 3.22 – Les différentes étapes du schéma NRRK3

3.2.5 Estimation du coût algorithmique du schéma NRK3

Le tableau 3.23 montre le nombre de bilans de flux (Nbre. d'opérations B) et le nombre de valeurs numériques (Nbre. d'opérations V) calculés par les différentes méthodes composant le schéma NRK3. On détaille ci-dessous les formules du tableau 3.23.

- La méthode RK3 de pas de temps Δt est appliquée sur $N_S - 1$ cellules (toutes les cellules du sous-domaine Slow, sauf la dernière). Elle nécessite le calcul de trois bilans de flux et de trois valeurs numériques. Cette méthode effectue donc $3 \times (N_S - 1)$ opérations B et autant d'opérations V .
- La méthode RK_S est appliquée sur la dernière cellule du sous-domaine Slow. Comme le montre le tableau 3.22, elle nécessite le calcul de trois bilans de flux et de sept valeurs numériques (les trois valeurs de la méthode de base et quatre valeurs additionnelles). Cette méthode effectue donc 3×1 opérations B et 7×1 opérations V .
- La méthode RK_F est appliquée sur la première cellule du sous-domaine Fast. Comme le montre le tableau 3.22, elle nécessite le calcul de six bilans de flux et de sept valeurs numériques (les six valeurs de la méthode de base et une valeur additionnelle). Cette méthode effectue donc 6×1 opérations B et 7×1 opérations V .
- Les deux applications de la méthode RK3 de pas de temps $\Delta t/2$ sont effectuées sur toutes les cellules du sous-domaine Fast, sauf la première. Ces deux applications nécessitent le calcul de six bilans de flux et de six valeurs numériques. $6 \times (N_F - 1)$ opérations B sont effectuées, et autant d'opérations V .

	$RK3(\Delta t)$	RK_S	RK_F	$RK3(\Delta t/2)$	Total
	$N_S - 1$ cell.	1 cell.	1 cell.	$N_F - 1$ cell.	$N_S + N_F$ cell.
Nbre. d'opérations B	$3 \times (N_S - 1)$	3×1	6×1	$6 \times (N_F - 1)$	$3N_S + 6N_F$
Nbre. d'opérations V	$3 \times (N_S - 1)$	7×1	7×1	$6 \times (N_F - 1)$	$3N_S + 6N_F + 5$

TABLE 3.23 – Coût algorithmique de chaque méthode pour le schéma NRK3

En utilisant la formule (2.24), le coût algorithmique du schéma NRK3 s'écrit donc :

$$C_{NRK3} = \underbrace{(3N_S + 6N_F) \times (\eta_B + \eta_V)}_{\text{Coût du schéma idéal}} + \underbrace{5\eta_V}_{\text{Surcoût dû à la zone tampon}} \quad (3.43)$$

Remarque 3.4. *L'application de la stratégie de correction de flux nécessite le calcul d'une valeur numérique supplémentaire : il s'agit de la valeur corrigée sur la dernière cellule du sous-domaine Slow. On estime donc que dans ce cas, le coût algorithmique du schéma NRK3 s'écrit :*

$$C_{NRK3} = (3N_S + 6N_F) \times (\eta_B + \eta_V) + 6\eta_V \quad (3.44)$$

A première vue, le coût algorithmique de notre schéma NRK3 est donc plus faible que celui du schéma CSRK3, donné par la relation (2.27). Cela est dû au fait que les deux zones tampon de notre schéma NRK3 sont algorithmiquement un peu moins coûteuse que la zone tampon du schéma CSRK3. Néanmoins, lorsque les nombres N_S et N_F de cellules sont grands devant un (ce qui est le cas en général), on a :

$$C_{NRK3} \approx C_{CSRK3} \approx (3N_S + 6N_F) \times (\eta_B + \eta_V)$$

et donc les deux schémas tendent vers le même coût algorithmique.

3.2.6 Différences avec les schémas à pas de temps local d'ordre trois de la littérature

Contrairement aux schémas à pas de temps local d'ordre trois de la littérature [17, 72], notre méthode RK3 de base est une méthode de type "Low storage". Notre schéma NRK3 est donc potentiellement moins coûteux en terme de mémoire que les schémas présentés dans [17, 72].

Le schéma de Schlegel *et al* [17] est basé sur une méthode de Runge-Kutta d'ordre trois à quatre étapes, dont le tableau de Butcher est le tableau 3.24. On se propose de comparer le coût algorithmique de notre schéma à celui de Schlegel *et al* [17].

0	0	$U^{n,2} = U^{n,1} - \frac{1}{2}\Delta t D(U^{n,1})$
$\frac{1}{2}$	$\frac{1}{2}$	$U^{n,3} = U^{n,1} - \Delta t \left(-\frac{1}{6}D(U^{n,1}) + \frac{2}{3}D(U^{n,2}) \right)$
$\frac{1}{2}$	$-\frac{1}{6}$	$U^{n,4} = U^{n,1} - \Delta t \left(\frac{1}{3}D(U^{n,1}) - \frac{1}{3}D(U^{n,2}) + D(U^{n,3}) \right)$
1	$\frac{1}{3}$	$U^{n+1} = U^{n,1} - \Delta t \left(\frac{1}{6}D(U^{n,1}) + \frac{1}{3}D(U^{n,2}) + \frac{1}{3}D(U^{n,3}) \right)$
	$-\frac{1}{3}$	$+ \frac{1}{6}D(U^{n,4})$
	$\frac{1}{3}$	
	$\frac{1}{3}$	
	$\frac{1}{6}$	

TABLE 3.24 – Tableau de Butcher de la méthode de base du schéma de Schlegel *et al* [17] (à gauche), et application de la méthode à l'équation différentielle (2.18) (à droite).

La construction du schéma de Schlegel *et al* est très similaire à la construction de nos schémas : la méthode de base de pas de temps Δt est appliquée sur le sous-domaine Slow, et deux applications de la méthode de base de pas de temps $\Delta t/2$ sont effectuées dans le sous-domaine Fast. A l'interface (au niveau des zones tampon), des valeurs additionnelles sont calculées pour effectuer la transition. Nous ne rentrons pas dans les détails du schéma et nous ne considérons pas les zones tampons, dont la taille est petite devant les N_S cellules du sous-domaine Slow ou les N_F cellules du sous-domaine Fast : le coup algorithmique de ces zones tampon peut donc être négligé. Le tableau de Butcher 3.24 montre que la méthode de base du schéma de Schlegel *et al* nécessite le calcul de quatre bilans de flux et

de quatre valeurs numériques. Deux applications successives de cette méthode nécessitent donc le calcul de huit bilans de flux et de huit valeurs numériques. Le coût algorithmique du schéma de Schlegel est donc :

$$C_{Schlegel} \approx (4N_S + 8N_F) \times (\eta_B + \eta_V). \quad (3.45)$$

Sans considérer les zones tampons, le coût algorithmique de notre schéma NRK3 est :

$$C_{NRK3} \approx (3N_S + 6N_F) \times (\eta_B + \eta_V). \quad (3.46)$$

Notre schéma NRK3 a donc un coût algorithmique plus faible que celui de Schlegel *et al.* Néanmoins, le schéma de Schlegel *et al* a peut-être une meilleure stabilité que notre schéma NRK3 puisqu'il est basé sur une méthode de Runge-Kutta à quatre étape. Ceci permettrait au schéma de Schlegel *et al* de fonctionner avec des pas de temps supérieurs à ceux utilisés par notre schéma NRK3, ce qui compenserait son surcoût algorithmique. Par manque de temps, il n'a malheureusement pas été possible de vérifier tout ceci expérimentalement

Le schéma de Kværnø *et al* est basé sur une méthode de Runge-Kutta à trois étapes d'ordre trois, présentée sur le tableau 3.25 suivant.

0	0	$U^{n,2} = U^{n,1} - \frac{1}{2}\Delta t D(U^{n,1})$
$\frac{1}{2}$	$\frac{1}{2}$	$U^{n,3} = U^{n,1} - \frac{3}{4}\Delta t D(U^{n,2})$
$\frac{3}{4}$	0 $\frac{3}{4}$	$U^{n+1} = U^{n,1} - \Delta t (\frac{2}{9}D(U^{n,1}) + \frac{1}{3}D(U^{n,2}) + \frac{4}{9}D(U^{n,3}))$
$\frac{2}{9}$	$\frac{1}{3}$ $\frac{4}{9}$	

TABLE 3.25 – Tableau de Butcher de la méthode de base du schéma de Kværnø *et al* [72] (à gauche), et application de la méthode à l'équation différentielle (2.18) (à droite).

La construction du schéma de Kværnø *et al* est très similaire à la construction de notre schéma NRK3 : la méthode de base de pas de temps Δt est appliquée sur le sous-domaine Slow, et deux applications de la méthode de base de pas de temps $\Delta t/2$ sont effectuées dans le sous-domaine Fast. Au niveau des zones tampon, des valeurs additionnelles sont calculées pour effectuer la transition. Le tableau de Butcher 3.24 montre que la méthode de base du schéma de Kværnø *et al* nécessite le calcul de trois bilans de flux et de trois valeurs numériques. Deux applications successives de cette méthode nécessitent donc le calcul de six bilans de flux et de six valeurs numériques. Sans considérer les zones tampon, le coût algorithmique du schéma de Kværnø *et al* est donc approximativement identique à celui de notre schéma :

$$C_{Kværnø} \approx C_{NRK3} \approx (3N_S + 6N_F) \times (\eta_B + \eta_V). \quad (3.47)$$

Étant donné que les auteurs appliquent leur schéma à des systèmes d'équations différentielles, la notion de conservativité n'entre pas en jeu. Ainsi, aucune stratégie de correction flux n'est proposée pour ce schéma.

3.2.7 Généralisation du schéma NRK3 à N pas de temps

Le schéma NRK3 se généralise à N pas de temps en utilisant l'approche récursive décrite précédemment. Une condition nécessaire pour pouvoir utiliser cette approche récursive est de conserver un rapport deux entre pas de temps adjacents. La figure 3.4 illustre cette approche récursive pour un maillage sur lequel trois pas de temps différents sont appliqués : Δt , $\Delta t/2$ et $\Delta t/4$ ($N = 3$). La transition entre les méthodes RK3 de pas de temps $\Delta t/2$ et $\Delta t/4$ est assurée par les méthodes RK_S et RK_F de pas de temps $\Delta t/2$ (notées $RK_S(\Delta t/2)$ et $RK_F(\Delta t/2)$).

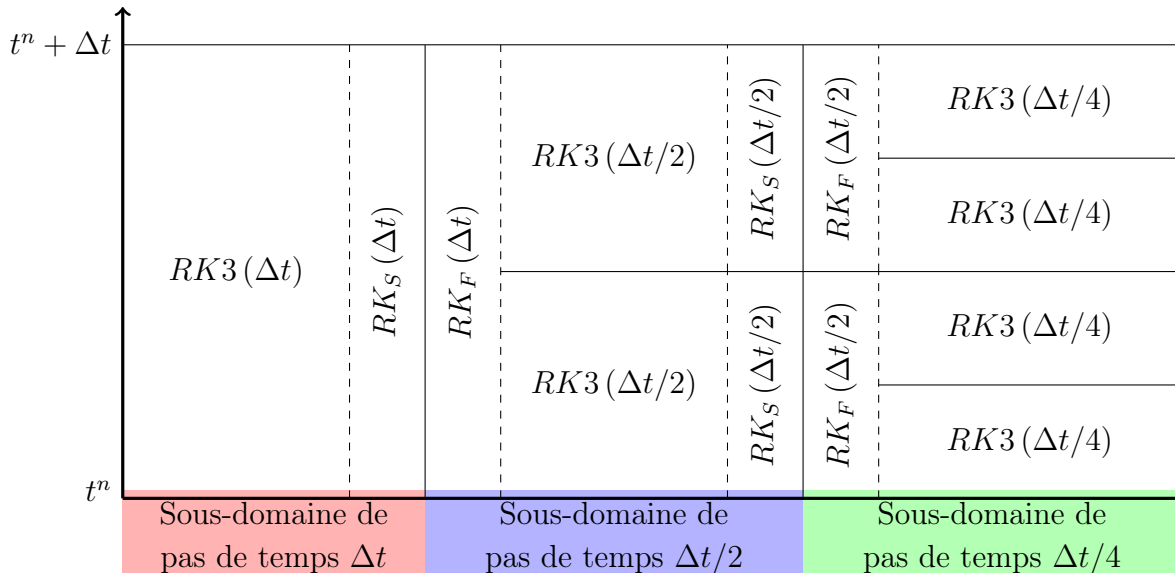


FIGURE 3.4 – Approche récursive pour généraliser le schéma CSRK2 à N pas de temps

3.3 Implémentation du schéma NRK3

Le schéma NRK3 a été implémenté dans le code de calcul FastS développé à l'ONERA. Il s'agit d'un code de calcul sur maillages structurés, optimisé pour le calcul haute performance. Il assure un parallélisme MPI et OpenMP, et est muni de techniques d'optimisation des accès mémoires (Cache Blocking...) et de techniques de vectorisation.

Dans cette partie à dominante algorithmique, le terme "sous-domaine" est remplacé par le terme "zone", mieux adapté.

Parmi les deux schémas à pas de temps local que nous proposons (NRK2 et NRK3), le schéma NRK3 est le plus complexe puisque les deux méthodes constituant le schéma (RK_S et RK_F) doivent fournir des valeurs additionnelles. Nous proposons un algorithme dans

le cas le plus complexe, c'est à dire le cas du schéma NRK3. Cet algorithme est proposé dans un contexte général. En effet, le domaine de calcul peut être 1D, 2D ou 3D, et il est constitué de N pas de temps :

$$\frac{\Delta t}{2^0}, \frac{\Delta t}{2^1}, \frac{\Delta t}{2^2}, \dots, \frac{\Delta t}{2^{N-1}}.$$

3.3.1 Notion de raccords entre zones

Dans cette partie, nous nous limitons au cas 2D pour ne pas trop complexifier les schémas et descriptions. Néanmoins, toutes les remarques faites dans ce cas 2D sont valables pour le cas 3D, et bien sûr le cas 1D.

Pour implémenter le schéma NRK3, nous avons choisi une approche **multizones**. Le maillage est découpé en Q zones structurées avec $Q \geq N$, car il est possible d'associer un même pas de temps à plusieurs zones. Chaque zone est dotée de cellules fictives afin de recevoir les informations des zones adjacentes. Cette approche est illustrée par la figure 3.5, qui représente une vue éclatée de trois zones 2D de pas de temps Δt et $\Delta t/2$.

Précisons qu'une approche **monozone** est également envisageable. Dans ce cas, le maillage est constitué d'une seule "grosse zone". Pour délimiter les zones d'application des pas de temps sur le maillage, des plages d'indices leur sont attribués. Cette approche permet de réduire le nombre des cellules fictives par rapport à une approche multizone. Vu qu'il y a moins de cellules fictives, il y a moins d'opération de copie de valeurs dans ces cellules, ce qui réduit le temps de simulation par rapport à une approche multizone. Cependant, il semble peut probable que l'on observe une réduction importante du temps de calcul puisque le coût algorithmique d'une opération de copie est relativement faible. L'approche **monozone** est plus complexe à mettre en place et n'a pas été testée dans cette thèse.

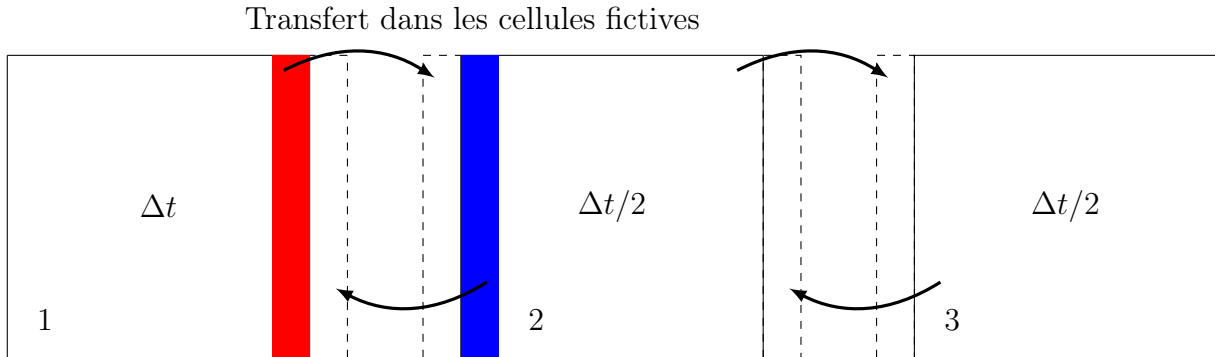


FIGURE 3.5 – Vue éclatée de trois zones de pas de temps respectif Δt et $\Delta t/2$. Ces zones sont numérotées de 1 à 3.

Pour chaque zone, il est nécessaire connaître le pas de temps de ses zones adjacentes. Considérons par exemple la zone de gauche (zone 1) sur la figure 3.5 (cette figure représente une vue éclatée d'un maillage constitué de trois zones et de deux pas de temps). Comme

indiqué sur la figure 3.5, toutes les cellules de cette zone sont intégrées avec la méthode RK3 de pas de temps Δt . Cette zone partage une frontière (un raccord) avec la zone 2, de pas de temps $\Delta t/2$. Ce raccord constitue donc une zone d'interface entre deux zones de pas de temps différent : des valeurs additionnelles doivent être calculées pour assurer une transition correcte entre les deux méthodes. Ainsi dans la zone 1, à certaines étapes de l'algorithme, des valeurs additionnelles (en rouge) sont calculées au niveau du bord correspondant au raccord (le bord droit de la zone 1). Ces valeurs additionnelles sont ensuite transférées dans les cellules fictives de la zone 2, pour "alimenter" le stencil des solutions numériques du bord de la zone 2. De même, à une certaine étape de l'algorithme, des valeurs additionnelles (en bleu) sont calculées au niveau du bord gauche de la zone 2 (le bord correspondant au raccord avec la zone 1, de pas de temps Δt) et elles sont transférées dans les cellules fictives de la zone 1. Lorsque deux zones adjacentes ont le même pas de temps (zones 2 et 3), aucune valeur additionnelle n'est calculée au niveau des bords correspondant au raccord. Seules les "classiques" estimations de la méthode RK3 sont transférées dans les cellules fictives.

A partir de cet exemple, on peut définir trois types de raccords :

- Raccord de type 1 : la zone adjacente a un pas de temps deux fois plus faible que la zone courante. Dans ce cas, la zone courante doit calculer des valeurs additionnelles (valeurs additionnelles "bleues") lors de quatre étapes de l'algorithme, au niveau du bord correspondant au raccord.
- Raccord de type 2 : la zone adjacente a un pas de temps deux fois plus important que la zone courante. Dans ce cas, la zone courante doit calculer des valeurs additionnelles (valeurs additionnelles "rouge") lors d'une étape de l'algorithme, au niveau bord correspondant au raccord.
- Raccord de type 3 : la zone adjacente a le même pas de temps que la zone courante. Dans ce cas, la zone courante ne calcule pas de valeurs additionnelle au niveau du bord correspondant à ce raccord.

3.3.2 Notion d'évènement

Pour définir la notion d'évènement utile à la mise en place notre algorithme, nous nous appuyons sur le tableau 3.26, qui représente les différentes étapes du schéma NRK3 constitué de deux pas de temps : Δt et $\frac{\Delta t}{2}$. Ce tableau est différent du tableau 3.22 initial (page 107), dans lequel on se contentait de "suivre" les étapes des tableaux de Butcher des méthodes RK_S et RK_F . Nous avons inversé les étapes 4 et 5 du tableau 3.22 initial pour faciliter la généralisation à N pas de temps, et les bilans de flux sont systématiquement évalués après le calcul des valeurs numériques. A l'instant initial, la solution $u^{n,1}$ et le bilan des flux $D^{n,1}$ sont supposés connus sur l'ensemble du domaine de calcul.

Sous-domaine Slow		Sous-domaine Fast	
	RK3 (Δt) : $1 \leq i \leq i_0 - 1$	RK5 : $i = i_0$	RK3 ($\Delta t/2$) : $i_0 + 2 \leq i \leq N$
1	$u_i^{n,2} = u_i^{n,1} - \alpha_1^S \Delta t D_i^{n,1}$ Evaluer $D_i^{n,2}$	$u_i^{n,2} = u_i^{n,1} - \Delta t \frac{a_{21}}{2} D_i^{n,1}$ Evaluer $D_i^{n,2}$	$u_i^{n,2} = u_i^{n,1} - \Delta t \frac{a_{21}}{2} D_i^{n,1}$ Evaluer $D_i^{n,2}$
2	$u_i^{n,3} = u_i^{n,1} - \alpha_2^S \Delta t D_i^{n,1}$	$u_i^{n,3} = u_3^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,1} + \frac{a_{32}}{2} D_i^{n,2} \right)$ Evaluer $D_i^{n,3}$	$u_i^{n,3} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,1} + \frac{a_{32}}{2} D_i^{n,2} \right)$ Evaluer $D_i^{n,3}$
3	$u_i^{n,4} = u_i^{n,1} - a_{21} \Delta t D_i^{n,1}$ Evaluer $D_i^{n,4}$	$u_i^{n,4} = u_i^{n,1} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + \frac{b_2}{2} D_i^{n,2} + \frac{b_3}{2} D_i^{n,3} \right)$ Evaluer $D_i^{n,4}$	$u_i^{n,4} = u_i^{n,1} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + \frac{b_2}{2} D_i^{n,2} + \frac{b_3}{2} D_i^{n,3} \right)$ Evaluer $D_i^{n,4}$
4	$u_i^{n,6} = u_i^{n,1} - \Delta t (a_{31} D_i^{n,1} + a_{32} D_i^{n,2})$ Evaluer $D_i^{n,6}$	$u_i^{n,6} = u_i^{n,1} - \Delta t (\alpha_1^S D_i^{n,1} + \alpha_2^S D_i^{n,2})$	
5	$u_i^{n,5} = u_i^{n,1} - \alpha_4^S \Delta t D_i^{n,1}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{a_{21}}{2} D_i^{n,4}$ Evaluer $D_i^{n,5}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{a_{21}}{2} D_i^{n,4}$ Evaluer $D_i^{n,5}$
6	$u_i^{n,7} = u_i^{n,1} - \Delta t \left(\alpha_8^S D_i^{n,1} + \alpha_{11}^S D_i^{n,4} \right)$	$u_i^{n,7} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,4} + \frac{a_{32}}{2} D_i^{n,5} \right)$ Evaluer $D_i^{n,7}$	$u_i^{n,6} = u_i^{n,1} - \Delta t \left(\frac{a_{31}}{2} D_i^{n,4} + \frac{a_{32}}{2} D_i^{n,5} \right)$ Evaluer $D_i^{n,6}$
7	$u_i^{n+1} = u_i^{n,1} - \Delta t \left(b_1 D_i^{n,1} + b_2 D_i^{n,4} + b_3 D_i^{n,6} \right)$ Evaluer D_i^{n+1}	$u_i^{n+1} = u_i^{n,4} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + b_2 D_i^{n,5} + \frac{b_3}{2} D_i^{n,7} \right)$ Evaluer D_i^{n+1}	$u_i^{n+1} = u_i^{n,4} - \Delta t \left(\frac{b_1}{2} D_i^{n,1} + b_2 D_i^{n,4} + b_3 D_i^{n,6} \right)$ Evaluer D_i^{n+1}
8	Correction de flux Réévaluer D_i^{n+1} au niveau de la solution corrigée		

TABLE 3.26 – Les différentes étapes du schéma NRK3

On présente maintenant les différents évènements, c'est à dire **les différents types d'opérations à effectuer sur une zone (calcul d'une estimation de la méthode RK3, calcul d'une valeur additionnelle...)**. A partir du tableau 3.26, on peut recenser six évènements différents à effectuer sur une zone de pas de temps quelconque $\frac{\Delta t}{2^i}$. On présente tout d'abord les trois premiers évènements, qui correspondent à une application "classique" de la méthode RK3 :

- **Evènement (1)** $[\frac{\Delta t}{2^i}]$: Calcul de la première estimation de la méthode RK3 avec un pas de temps $\frac{\Delta t}{2^i}$. Par exemple, entre t^n et $t^n + \frac{\Delta t}{2^i}$, cette étape s'écrit :

$$u^{n,2} = u^{n,1} - \frac{\Delta t}{2^i} a_{21} D^{n,1} \quad (3.48)$$

- **Evènement (2)** $[\frac{\Delta t}{2^i}]$: Calcul de la seconde estimation de la méthode RK3 avec un pas de temps $\frac{\Delta t}{2^i}$. Entre t^n et $t^n + \frac{\Delta t}{2^i}$, cette étape s'écrit :

$$u^{n,3} = u^{n,1} - \frac{\Delta t}{2^i} (a_{31} D^{n,1} + a_{32} D^{n,2}) \quad (3.49)$$

- **Evènement (3)** $[\frac{\Delta t}{2^i}]$: Calcul de la solution finale de la méthode RK3 avec un pas de temps $\frac{\Delta t}{2^i}$. On note cette solution $u^{n+\frac{1}{2^i}}$. Entre t^n et $t^n + \frac{\Delta t}{2^i}$, cette étape s'écrit :

$$u^{n+\frac{1}{2^i}} = u^{n,1} - \frac{\Delta t}{2^i} (b_1 D^{n,1} + b_2 D^{n,2} + b_3 D^{n,3}). \quad (3.50)$$

- **Evènement (4)** : Evaluation d'un nouveau bilan de flux

Dans le contexte d'un schéma à pas de temps local une zone de pas de temps $\frac{\Delta t}{2^i}$ peut être raccordée à des zones de pas de temps deux fois plus faible ($\frac{\Delta t}{2^{i+1}}$) ou à des zones pas de temps deux fois plus grand ($\frac{\Delta t}{2^{i-1}}$). On est donc amené à définir trois évènements supplémentaires, comme le montre le tableau 3.26 :

- **Evènement (5)** $[\frac{\Delta t}{2^i}]$: Calcul, avec un pas de temps $\frac{\Delta t}{2^i}$, d'une valeur additionnelle destinée aux zones adjacentes de pas de temps deux fois plus faible.
- **Evènement (6)** $[\frac{\Delta t}{2^i}]$: Calcul, avec un pas de temps $\frac{\Delta t}{2^i}$, d'une valeur additionnelle destinée aux zones adjacentes de pas de temps deux fois plus grand.
- **Evènement (7)** $[\frac{\Delta t}{2^i}]$: Cet évènement consiste à effectuer l'étape de correction de flux dans la zone de pas de temps $\frac{\Delta t}{2^i}$ (afin d'assurer la conservativité) et à évaluer le bilan de flux au niveau de la solution corrigée. Cet évènement est réalisé au niveau des raccords avec les zones adjacentes de pas de temps deux fois plus faible.

Ces évènements pourront être abrégés par E1, E2, E3, E4, E5, E6, E7 dans la suite. Pour davantage de clarté, on réécrit le tableau 3.26 avec les différents évènements (tableau 3.27).

Etape (q)	Zone de pas de temps Δt		Zone de pas de temps $\Delta t/2$	
	Hors zone tampon	Zone tampon	Zone tampon	Hors zone tampon
1		$E5[\Delta t]$	$E1[\frac{\Delta t}{2}]$ $E4$	
2		$E5[\Delta t]$	$E2[\frac{\Delta t}{2}]$ $E4$	
3	$E1[\Delta t]$ $E4$		$E3[\frac{\Delta t}{2}]$ $E4$	
4	$E2[\Delta t]$ $E4$		$E6[\Delta t]$	
5		$E5[\Delta t]$	$E1[\frac{\Delta t}{2}]$ $E4$	
6		$E5[\Delta t]$	$E2[\frac{\Delta t}{2}]$ $E4$	
7	$E3[\Delta t]$ $E4$		$E3[\frac{\Delta t}{2}]$ $E4$	
8		$E7[\Delta t]$		

TABLE 3.27 – Réécriture du tableau 3.26 en fonction des différents évènements.

Comme le montre le tableau 3.27, les différents évènements ne sont pas réalisés aux mêmes étapes suivant les zones (et leur pas de temps associé). Par exemple, l'évènement **E2** est réalisé aux étapes 2 et 6 dans la zone de pas de temps $\frac{\Delta t}{2}$ et à l'étape 4 dans la zone de pas de temps Δt . Dans la suite, nous décrivons la méthode mise en place pour piloter les évènements dans chaque zone, dans un contexte général où le schéma est constitué de N pas de temps différents. Pour décrire cette méthode, on s'appuiera sur le cas particulier du schéma constitué de deux pas de temps : Δt et $\frac{\Delta t}{2}$ (tableau 3.27).

3.3.3 Paramètres q^* et θ_i

Afin de décrire convenablement le pilotage des évènements dans un contexte général, nous devons introduire deux notions importantes : **le nombre d'étapes q^* de l'algorithme** ainsi que **le paramètre θ_i associé à chaque zone**.

Nombre d'étapes de l'algorithme

Avant d'effectuer les q^* étapes de l'algorithme, la solution est estimée à l'instant t^n sur l'ensemble du domaine de calcul. A l'issue des q^* étapes de l'algorithme, la solution est estimée à l'instant $t^n + \Delta t$ sur l'ensemble du domaine de calcul.

Pour déterminer le nombre d'étapes q^* de notre algorithme, on raisonne sur le pas de temps minimal $\frac{\Delta t}{2^{N-1}}$. Dans les zones munies du pas de temps minimal $\frac{\Delta t}{2^{N-1}}$, il faut réaliser 2^{N-1} applications de la méthode RK3 de pas de temps $\frac{\Delta t}{2^{N-1}}$ pour intégrer la solution entre t^n et $t^n + \Delta t$. Or, la méthode RK3 est une méthode de Runge-Kutta à trois étapes. On prévoit une quatrième étape pour le calcul de la valeur additionnelle destinée aux zones adjacentes de pas de temps deux fois plus grand. Le nombre d'étapes q^* de l'algorithme est donc :

$$q^* = 4 \times 2^{N-1} \quad (3.51)$$

Ainsi, pour un schéma constitué de deux pas de temps ($N = 2$), on a $q^* = 8$.

Paramètre θ_i associé à chaque zone

Dans toute zone de pas de temps $\frac{\Delta t}{2^i}$, il y a q^* étapes. Celles-ci peuvent être séparées en 2^i "blocs" identiques (car il y a 2^i applications de la méthode RK3). Le nombre d'étapes dans chaque bloc est noté θ_i . Ainsi, dans toute zone de pas de temps $\frac{\Delta t}{2^i}$, le nombre d'étapes vaut $q^* = 2^i \times \theta_i$, et donc :

$$\theta_i = \frac{q^*}{2^i}. \quad (3.52)$$

Dans le cas d'un schéma constitué de deux pas de temps (tableau 3.27) :

- le paramètre θ_0 associé à la zone de pas de temps Δt vaut : $\theta_0 = \frac{q^*}{2^0} = 8$,
- le paramètre θ_1 associé à la zone de pas de temps $\frac{\Delta t}{2}$ vaut $\theta_1 = \frac{q^*}{2^1} = 4$.

Le paramètre θ_i nous permet de piloter la réalisation des évènements dans chaque zone. On décrit maintenant le pilotage de ces évènements, en s'appuyant sur le tableau 3.27.

3.3.4 Pilotage des évènements

Pilotage de l'évènement (1) $\left[\frac{\Delta t}{2^i}\right]$

Dans la zone de pas de temps Δt , l'évènement (1) $\left[\frac{\Delta t}{2^i}\right]$ est réalisé à l'étape $q = 3$ (voir le tableau 3.27). Cette étape vérifie la relation :

$$q = \frac{\theta_0}{2} - 1.$$

Dans la zone de pas de temps $\frac{\Delta t}{2}$, cet évènement est réalisé aux étapes $q = 1$ et $q = 5$, qui vérifient :

$$q = \frac{\theta_1}{2} - 1,$$

$$q = \theta_1 + \frac{\theta_1}{2} - 1.$$

De manière générale, l'évènement (1) $[\frac{\Delta t}{2^i}]$ est réalisé aux étapes q vérifiant : $q = k \times \theta_i + \frac{\theta_i}{2} - 1$, ou encore :

$$\text{mod}(q, \theta_i) = \frac{\theta_i}{2} - 1, \quad (3.53)$$

où mod désigne la fonction "modulo", qui renvoie le reste de la division euclidienne de q par θ_i .

Pilotage de l'évènement (2) $[\frac{\Delta t}{2^i}]$

D'après le tableau 3.27, l'évènement (2) $[\frac{\Delta t}{2^i}]$ est réalisé à l'étape $q = 4$ dans la zone de pas de temps Δt . Cette étape q vérifie :

$$q = 0 \times \theta_0 + \frac{\theta_0}{2}$$

Dans la zone de pas de temps $\frac{\Delta t}{2}$, cet évènement est réalisé aux étapes $q = 2$ et $q = 6$, qui vérifient :

$$q = 0 \times \theta_1 + \frac{\theta_1}{2},$$

$$q = 1 \times \theta_1 + \frac{\theta_1}{2}.$$

On remarque donc que l'évènement (2) $[\frac{\Delta t}{2^i}]$ est réalisé aux étapes q vérifiant $q = k \times \theta_i + \frac{\theta_i}{2}$, ou encore :

$$\text{mod}(q, \theta_i) = \frac{\theta_i}{2} \quad (3.54)$$

Pilotage de l'évènement (3) $[\frac{\Delta t}{2^i}]$

D'après le tableau 3.27, l'évènement (3) $[\frac{\Delta t}{2^i}]$ est réalisé à l'étape $q = 7$ dans la zone de pas de temps Δt , et aux étapes $q = 3, 7$ dans la zone de pas de temps $\frac{\Delta t}{2}$. Ces étapes q vérifient : $q = k \times \theta_i - 1$, ou bien :

$$\text{mod}(q, \theta_i) = \theta_i - 1. \quad (3.55)$$

Pilotage de l'évènement (4)

L'évènement (4) est réalisé aux mêmes étapes que les évènements (1) (2) et (3). Par conséquent, cet évènement est réalisé aux étapes q vérifiant les formules (3.53), (3.54) et (3.55).

Pilotage de l'évènement (5) $[\frac{\Delta t}{2^i}]$

Comme le montre le tableau 3.27, l'évènement (5) $[\frac{\Delta t}{2^i}]$ est effectué aux mêmes étapes que les évènements $\mathbf{E1}[\frac{\Delta t}{2}]$. D'après la formule (3.53), les évènements $\mathbf{E1}[\frac{\Delta t}{2}]$ sont réalisés aux étapes q vérifiant :

$$q = k \times \theta_1 + \frac{\theta_1}{2} - 1 \quad (k = 0, 1) \quad (3.56)$$

Nous devons réécrire cette relation en faisant apparaître le paramètre θ_0 car l'évènement (5) $[\frac{\Delta t}{2^i}]$ est réalisé dans la zone de pas de temps Δt ; il faut donc qu'il soit piloté par le paramètre associé à cette zone, c'est à dire θ_0 . En sachant que $\theta_0 = 2\theta_1$, la formule (3.56) se réécrit :

$$q = k \times \frac{\theta_0}{2} + \frac{\theta_0}{4} - 1.$$

De manière générale, on réalise l'évènement (5) $[\frac{\Delta t}{2^i}]$ aux étapes q vérifiant $q = k \times \frac{\theta_i}{2} + \frac{\theta_i}{4} - 1$, ou de manière équivalente :

$$\text{mod}(q, \frac{\theta_i}{2}) = \frac{\theta_i}{4} - 1 \quad (3.57)$$

Le tableau 3.27 montre que l'évènement (5) $[\frac{\Delta t}{2^i}]$ doit également être effectué aux mêmes étapes que les évènements $\mathbf{E2}[\frac{\Delta t}{2}]$. D'après la formule (3.54), les évènements $\mathbf{E2}[\frac{\Delta t}{2}]$ sont réalisés aux étapes q vérifiant :

$$q = k \times \theta_1 + \frac{\theta_1}{2} \quad (k = 0, 1) \quad (3.58)$$

Comme précédemment, on réécrit la formule (3.58) en fonction de θ_0 :

$$q = k \times \frac{\theta_0}{2} + \frac{\theta_0}{4}.$$

De manière générale, on réalise l'évènement (5) $[\frac{\Delta t}{2^i}]$ aux étapes q vérifiant $q = k \times \frac{\theta_i}{2} + \frac{\theta_i}{4}$, ou de manière équivalente :

$$\text{mod}(q, \theta_i) = \frac{\theta_i}{4} \quad (3.59)$$

Pilotage de l'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$

L'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$ est effectué, comme le montre le tableau 3.27, à la même étape que l'évènement **E2** $[\Delta t]$. D'après la formule (3.54), cet évènement est réalisé aux étapes q vérifiant :

$$q = k \times \theta_0 + \frac{\theta_0}{2} \quad (k = 0)$$

Nous devons réécrire cette formule en faisant apparaître le paramètre θ_1 car l'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$ est réalisé dans la zone de pas de temps $\Delta t/2$; il faut donc qu'il soit piloté par le paramètre associé à cette zone, c'est à dire θ_1 . Cette relation se réécrit :

$$q = k \times 2\theta_1 + \theta_1 \tag{3.60}$$

De manière générale, on réalise l'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$ aux étapes q vérifiant $q = k \times 2\theta_i + \theta_i$, ou bien :

$$\text{mod}(q, 2\theta_i) = \theta_i \tag{3.61}$$

Pilotage de l'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$

L'évènement (6) $\lfloor \frac{\Delta t}{2^i} \rfloor$ est effectué à l'étape $q = 8$, qui vérifie $\text{mod}(q, \theta_0) = 0$ (car $\theta_0 = 8$). De manière générale, cet évènement est réalisé aux étapes q vérifiant :

$$\text{mod}(q, \theta_i) = 0 \tag{3.62}$$

3.3.5 Algorithme du schéma NRK3 à N pas de temps

Les notions développées dans les sections précédentes permettent d'implémenter le schéma NRK3 avec l'algorithme ci-après. Nous mentionnons que cet algorithme est indépendant de la dimension du problème (1D, 2D ou 3D). Cet algorithme se compose de trois boucles principales :

- Une première boucle parcourt les différentes étapes du schéma (de 1 à q^*).
- Une seconde boucle parcourt les différents pas de temps qui composent le schéma.
- Une troisième boucle parcourt les zones de pas de temps égal au pas de temps courant. Remarquons que cette troisième boucle est scindée en deux : dans un premier temps, on calcule les solutions numériques (estimations de Runge-Kutta, valeurs additionnelles...), puis dans un second temps, les bilans de flux au niveau de ces solutions numériques sont évalués. En effet, le tableau 3.27 montre que les bilans de flux sont systématiquement évalués après le calcul des solutions numériques.

Entrées : $u_i^{n,1}$ et $D_i^{n,1}$ connus sur tout le domaine de calcul, **cons** = Vrai

```

pour q = 1 to q* faire          /* Boucle sur les étapes du schéma */
  pour i = 0 to N - 1 faire    /* Boucle sur les pas de temps */
    pour toutes les zones de pas de temps  $\frac{\Delta t}{2^i}$  faire
      si mod(q,  $\theta_i$ ) =  $\frac{\theta_i}{2} - 1$  alors
        | Faire évènement (1) [ $\frac{\Delta t}{2^i}$ ]
      fin
      si mod(q,  $\theta_i$ ) =  $\frac{\theta_i}{2}$  alors
        | Faire évènement (2) [ $\frac{\Delta t}{2^i}$ ]
      fin
      si mod(q,  $\theta_i$ ) =  $\theta_i - 1$  alors
        | Faire évènement (3) [ $\frac{\Delta t}{2^i}$ ]
      fin

      pour tous les raccords de la zone faire
        si raccord de type 1 and ( $\text{mod}(q, \frac{\theta_i}{2}) = \frac{\theta_i}{4} - 1$  ou  $\text{mod}(q, \frac{\theta_i}{2}) = \frac{\theta_i}{4}$ )
          alors
            | Faire évènement (5) [ $\frac{\Delta t}{2^i}$ ]
          fin
        si raccord de type 2 and  $\text{mod}(q, 2\theta_i) = \theta_i$  alors
          | Faire évènement (6) [ $\frac{\Delta t}{2^i}$ ]
        fin
        si raccord de type 1 and  $\text{mod}(q, \theta_i) = 0$  and cons = Vrai alors
          | Faire évènement (7) [ $\frac{\Delta t}{2^i}$ ]
        fin
      fin
    fin
  fin
  pour toutes les zones de pas de temps  $\frac{\Delta t}{2^i}$  faire
    si mod(q,  $\theta_i$ ) =  $\frac{\theta_i}{2} - 1$  ou  $\text{mod}(q, \theta_i) = \frac{\theta_i}{2}$  ou  $\text{mod}(q, \theta_i) = \theta_i - 1$  alors
      | Faire évènement (4)
    fin
  fin
fin

```

Sorties : u_i^{n+1} et D_i^{n+1} sur tout le domaine de calcul.

$$u_i^{n+1,1} \leftarrow u_i^{n+1}$$

$$D_i^{n+1,1} \leftarrow D_i^{n+1}$$

Algorithme 1 : Algorithme du schéma NRK3 à N pas de temps

Chapitre 4

Validation des schémas proposés sur des cas tests académiques

Le chapitre précédent a permis de présenter et donner les caractéristiques des nouveaux schémas à pas de temps local NRK2 et NRK3. L'objectif est maintenant de valider ces schémas sur trois cas test académiques. Pour la validation, les résultats obtenus avec les schémas NRK2 et NRK3 sont comparés aux solutions issues des schémas de Constantinescu et Sandu CSRK2 et CSRK3, présentés dans le chapitre 2.

Dans un premier temps, les équations de Navier-Stokes régissant les écoulements de fluides compressibles, Newtonien et non pesants sont rappelées (seul ce type de fluide est considéré dans ce mémoire). Dans un second temps, on décrit les méthodes numériques utilisées pour résoudre numériquement ces équations sur des maillages structurés (seul ce type de maillage a été utilisé au cours de cette thèse). Enfin, les comparaisons entre les différents schémas à pas de temps local sont réalisées sur les trois cas test académiques.

4.1 Équations de la mécanique des fluides

On considère un domaine Ω de \mathbb{R}^3 muni du repère $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ et de la variable temporelle t . Les variables d'espace dans les directions \vec{e}_1 , \vec{e}_2 et \vec{e}_3 sont désignées par x_1 , x_2 et x_3 . Le système d'équations de Navier-Stokes pour un fluide Newtonien, compressible et non pesant est décrit par les relations (4.1) et (4.2). Dans la relation (4.2), la convention d'Einstein sur la sommation des indices répétés i et j est utilisée, ainsi que le symbole de Kronecker δ_{ij} (δ_{ij} vaut 1 si $i = j$, 0 sinon).

$$\frac{\partial U}{\partial t} + \nabla \cdot (F^c(U) - F^v(U)) = 0 \quad (4.1)$$

$$U = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{pmatrix} \quad F^c(U) = \begin{pmatrix} \rho u_i \vec{e}_i \\ \rho u_1 u_j \vec{e}_j + p \delta_{1j} \\ \rho u_2 u_j \vec{e}_j + p \delta_{2j} \\ \rho u_3 u_j \vec{e}_j + p \delta_{3j} \\ (\rho E + p) u_j \vec{e}_j \end{pmatrix} \quad F^v(U) = \begin{pmatrix} 0 \\ \tau_{1j} \vec{e}_j \\ \tau_{2j} \vec{e}_j \\ \tau_{3j} \vec{e}_j \\ \tau_{ij} u_i \vec{e}_i - q_j \vec{e}_j \end{pmatrix} \quad (4.2)$$

Les différentes variables liées au fluide $(\rho, u_1, u_2, u_3, \dots)$ dépendent des variables d'espace x_1, x_2, x_3 ainsi que de la variable temporelle t . La première équation du système décrit par les relations (4.1) et (4.2) est l'équation de conservation de la masse. Les deuxième, troisième et quatrième équations de ce système décrivent la conservation de la quantité de mouvement et la dernière équation du système décrit la conservation de l'énergie. Dans les relations (4.1) et (4.2), les termes F^c et F^v sont respectivement les termes de flux convectif et de flux visqueux.

Remarque 4.1. Lorsque le terme de flux visqueux F^v est nul, on obtient les équations d'Euler :

$$\frac{\partial U}{\partial t} + \nabla \cdot F^c(U) = 0. \quad (4.3)$$

Dans la relation (4.2), le terme ρ désigne la masse volumique, u_i désigne la vitesse dans la direction \vec{e}_i , p désigne la pression, E correspond à l'énergie totale spécifique, q_i désigne le flux de chaleur dans la direction \vec{e}_i et τ désigne le tenseur des contraintes visqueuses.

Le fluide étant supposé newtonien, le tenseur des contraintes s'écrit :

$$\tau_{ij} = \mu(T) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (4.4)$$

où la convention d'Einstein est utilisée sur l'indice k .

Dans l'expression du tenseur des contraintes (4.4), la viscosité du fluide $\mu(T)$ est une fonction de la température T du fluide, son expression est donnée par la loi de Sutherland :

$$\mu(T) = \mu_0 \left(\frac{T}{T_0} \right)^{1/2} \frac{1 + 110.4/T_0}{1 + 110.4/T} \quad (4.5)$$

Dans la loi de Sutherland (4.5), μ_0 et T_0 désignent respectivement une viscosité et une température de référence dont les valeurs sont : $\mu_0 = 1.711 \times 10^{-5} \text{kg.m}^{-1}.\text{s}^{-1}$, $T_0 = 273.16\text{K}$.

Le flux de chaleur q_i suit la loi de Fourier :

$$q_i = -\kappa(T) \frac{\partial T}{\partial x_i} \quad \text{avec} \quad \kappa(T) = \frac{\mu(T) C_p}{Pr}, \quad (4.6)$$

où Pr est le nombre de Prandtl ($Pr = 0.72$) et C_p la capacité calorifique à pression constante.

Enfin, l'énergie totale E s'exprime en fonction de l'énergie interne e :

$$E = e + \frac{1}{2}u_i u_i \quad (4.7)$$

L'énergie interne e est obtenue à partir de la loi des gaz parfaits. Pour un tel gaz, l'énergie interne e est donnée par la relation :

$$e = C_v T = \frac{1}{\gamma - 1} \frac{p}{\rho}, \quad (4.8)$$

où C_v est la capacité calorifique à volume constant, et $\gamma = \frac{C_p}{C_v} = 1.4$.

Remarque 4.2. Dans le cas d'un gaz parfait, la vitesse du son c s'écrit :

$$c = \sqrt{\gamma R_s T}, \quad (4.9)$$

où R_s est la constante spécifique des gaz parfaits (pour l'air, $R_s = 287 \text{ J.kg}^{-1} \text{ K}^{-1}$) et T la température.

Remarque 4.3. Les variables $(\rho, \rho u_1, \rho u_2, \rho u_3, \rho E)$ sont appelées "variables conservatives". Dans la suite, on sera également amené à travailler avec les variables primitives : (ρ, u_1, u_2, u_3, T) .

4.2 Rappels sur la discrétisation spatiale Volumes Finis (VF)

Tout d'abord, on exploite la linéarité de l'opérateur de divergence pour réécrire le système d'équations de Navier-Stokes (4.1) :

$$\frac{\partial U}{\partial t} + \nabla \cdot F^c(U) - \nabla \cdot F^v(U) = 0 \quad (4.10)$$

On considère un maillage structuré du domaine de calcul Ω . Chaque cellule du maillage peut donc être repérée par un triplet d'indice (i, j, k) . Comme expliqué dans la partie 1.1.1, la méthode VF consiste à intégrer les équations sur chaque cellule. Le système (4.10) est intégré sur une cellule Ω_{ijk} quelconque et le théorème de Green-Ostrogradski est appliqué sur les deux termes de divergence. En se servant des développements faits dans la partie 1.1.1, on aboutit à l'équation discrétisée en espace suivante :

$$\frac{d}{dt} U_{ijk} + \frac{1}{|\Omega_{ijk}|} \left(\sum_l F_{ijk,l}^c \cdot n_{ijk,l} S_{ijk,l} - \sum_l F_{ijk,l}^v \cdot n_{ijk,l} S_{ijk,l} \right) = 0. \quad (4.11)$$

Dans la relation (4.11), on rappelle que $F_{ijk,l}^c$ et $F_{ijk,l}^v$ sont les flux numériques convectif et visqueux sur la face l de la cellule Ω_{ijk} . Les termes $n_{ijk,l}$ et $S_{ijk,l}$ sont respectivement la normale sortante à la face l de la cellule Ω_{ijk} et l'aire de cette même face.

Pour un meilleur repérage des différentes faces de la cellule Ω_{ijk} , on utilise les indices $i \pm \frac{1}{2}, j, k$ sur les faces de normale colinéaire à \vec{e}_1 , les indices $i, j \pm \frac{1}{2}, k$ pour les faces de normale colinéaire à \vec{e}_2 , et les indices $ij, k \pm \frac{1}{2}$ pour les faces de normale colinéaire à \vec{e}_3 (voir figure 4.1).

Par exemple, avec les nouveaux indices, on a :

$$\begin{aligned} \sum_l F_{ijk,l}^c \cdot n_{ijk,l} S_{ijk,l} &= F_{i+\frac{1}{2},jk}^c \cdot n_{i+\frac{1}{2},jk} S_{i+\frac{1}{2},jk} + F_{i-\frac{1}{2},jk}^c \cdot n_{i-\frac{1}{2},jk} S_{i-\frac{1}{2},jk} \\ &+ F_{i,j+\frac{1}{2},k}^c \cdot n_{i,j+\frac{1}{2},k} S_{i,j+\frac{1}{2},k} + F_{i,j-\frac{1}{2},k}^c \cdot n_{i,j-\frac{1}{2},k} S_{i,j-\frac{1}{2},k} \\ &+ F_{ij,k+\frac{1}{2}}^c \cdot n_{ij,k+\frac{1}{2}} S_{ij,k+\frac{1}{2}} + F_{ij,k-\frac{1}{2}}^c \cdot n_{ij,k-\frac{1}{2}} S_{ij,k-\frac{1}{2}} \end{aligned}$$

Comme évoqué dans la partie 1.1.1, une des principales difficultés de la méthode VF est le calcul des flux aux faces puisque la solution n'est connue qu'au centre des cellules. Les deux prochaines parties sont donc dédiées au calcul des flux convectifs et visqueux.

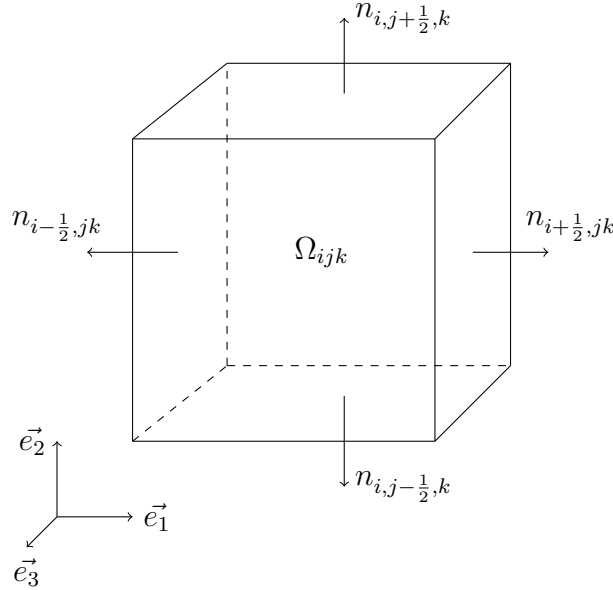


FIGURE 4.1 – Représentation d'un volume élémentaire Ω_{ijk} et des normales au niveau de ses faces, avec les nouveaux indices

4.3 Flux convectifs

Dans cette thèse, deux schémas numériques pour le calcul des flux convectifs ont été utilisés. Pour la majorité des cas test, le schéma que nous appelons "AUSM+(P) senseur" est utilisé. Ce schéma, proposé par Mary [87, 88] et Laurent [89], est une variante du schéma AUSM+(P) introduit par Edwards et Liou [90] en 1998. Comme montré dans [91, 89], le schéma "AUSM+(P) senseur" est adapté aux écoulements à faible nombre de Mach et ne présentant pas de choc (ce schéma est en effet dépourvu de propriété de capture de choc). La plupart des écoulements simulés dans cette thèse présentent ces deux caractéristiques, ce qui justifie l'utilisation de ce schéma.

Le schéma de Roe est uniquement utilisé sur le cas-test du tube à choc (résolu avec les équations d'Euler), afin de bénéficier de ses propriétés de capture de choc. Dans un souci de simplicité, ces deux schémas sont présentés dans un contexte 1D.

4.3.1 Schéma AUSM+(P) senseur

Le schéma AUSM+(P) senseur est basé sur la méthode FVS (Flux Vector Splitting), qui repose sur une décomposition du flux entre une contribution purement convective (F^{conv}) et une contribution liée à la pression (p). Ainsi, à l'interface $i + \frac{1}{2}, jk$ de la cellule représentée sur la figure 4.1, on obtient (en omettant les indices jk pour plus de clarté) :

$$F_{i+\frac{1}{2}}^c = F_{i+\frac{1}{2}}^{conv} + p_{i+\frac{1}{2}} \quad (4.12)$$

Le terme $F_{i+\frac{1}{2}}^{conv}$ du schéma AUSM+(P) senseur s'écrit :

$$F_{i+\frac{1}{2}}^{conv} = \frac{1}{2} \tilde{v}_{i+\frac{1}{2}} \left(U_{i+\frac{1}{2},G} + U_{i+\frac{1}{2},D} \right) - |v_{dis}| \frac{1}{2} \left(U_{i+\frac{1}{2},G} - U_{i+\frac{1}{2},D} \right), \quad (4.13)$$

où $U = (\rho, \rho u_1, \rho u_2, \rho u_3, \rho E + p)^T$ est le vecteur des variables conservatives. Le terme $U_{i+\frac{1}{2},G}$ (resp. $U_{i+\frac{1}{2},D}$) désigne l'état gauche (resp. droit) provenant de l'interpolation MUSCL de Van Leer [92] à l'interface d'indice $i + \frac{1}{2}$. Avec la méthode MUSCL, on suppose que la solution varie à l'intérieur de la cellule, ce qui conduit à une solution plus précise. Les états gauche et droit sont données par :

$$\begin{cases} U_{i+\frac{1}{2},G} = U_i + \frac{1}{6} (2(U_{i+1} - U_i) + (U_i - U_{i-1})) \\ U_{i+\frac{1}{2},D} = U_{i+1} - \frac{1}{6} (2(U_{i+1} - U_i) + (U_{i+2} - U_{i+1})) \end{cases} \quad (4.14)$$

Le terme de pression $p_{i+\frac{1}{2}}$, qui apparaît dans l'équation (4.12) s'écrit :

$$p_{i+\frac{1}{2}} = \frac{1}{2} (p_{i+\frac{1}{2},G} + p_{i+\frac{1}{2},D}), \quad (4.15)$$

où $p_{i+\frac{1}{2},G}$ et $p_{i+\frac{1}{2},D}$ sont les états gauche et droit de pression, provenant de l'interpolation de Van Leer.

Le terme $\tilde{v}_{i+\frac{1}{2}}$ de l'équation (4.13) décrit la vitesse normale à l'interface :

$$\tilde{v}_{i+\frac{1}{2}} = \frac{1}{2}n_{i+\frac{1}{2}} \cdot (V_{i+\frac{1}{2},G} + V_{i+\frac{1}{2},D}) - c_2(p_{i+\frac{1}{2},D} - p_{i+\frac{1}{2},G}), \quad (4.16)$$

où $n_{i+\frac{1}{2}}$ désigne la normale à la face d'indice $i + \frac{1}{2}$, c_2 une constante et V le vecteur vitesse : $V = (u_1, u_2, u_3)^T$.

Le paramètre $|v_{dis}|$ qui apparaît dans la relation (4.13) est homogène à une vitesse, et s'écrit :

$$|v_{dis}| = \Phi_{i+\frac{1}{2}} \max(|\tilde{v}_{i+\frac{1}{2}}|, c_1), \quad (4.17)$$

où c_1 est une constante numérique. Le terme $\Phi_{i+\frac{1}{2}}$ dans l'expression (4.17) a été introduit par Mary [88]. Il s'agit d'un senseur physique (d'où le nom donné au schéma) qui permet d'obtenir un schéma hybride centré/décentré. Les schémas centrés sont peu dissipatifs mais ils génèrent très facilement des oscillations parasites pouvant conduire à des instabilités numériques. L'introduction d'un décentrement permet de limiter les oscillations, ce qui assure la stabilité du schéma. Le terme $\Phi_{i+\frac{1}{2}}$ dépend de la régularité de la solution exprimée en variables primitives : $\phi = (\rho, u_1, u_2, u_3, T)^T$. La régularité de chaque variable primitive est évaluée dans les trois directions de l'espace, et au niveau de chaque face. Par exemple, dans la direction e_1^i et au niveau de la face d'indice $i + \frac{1}{2}$, on a :

$$\Delta_{\phi}^{i+\frac{1}{2}} = \begin{cases} -1 & \text{si } (\phi_{i+2} - \phi_{i+1})(\phi_{i+1} - \phi_i) < 0 \\ 1 & \text{sinon} \end{cases}$$

$$W_{\phi_m} = \begin{cases} 1 & \text{si } \Delta_{\phi_m}^{i+\frac{1}{2}} + \Delta_{\phi_m}^{i-\frac{1}{2}} < 0 \\ 0 & \text{sinon} \end{cases}$$

$$\Phi_{i+\frac{1}{2}} = \max(W_{\phi_m}) \quad 1 \leq m \leq 5$$

Lorsque des oscillations parasites au sein de la solution sont repérées, $\Phi_{i+\frac{1}{2}}$ prend la valeur 1 : le terme $|v_{dis}|$ est alors non nul et un décentrement est introduit. Si aucune oscillation n'est repérée, $\Phi_{i+\frac{1}{2}}$ prend la valeur 0 : le terme $|v_{dis}|$ est alors nul, ce qui réduit le décentrement.

Les constantes c_1 et c_2 proposées par Mary [87] valent :

$$c_1 \approx 0.04U_{ref} \quad \text{et} \quad c_2 \approx \frac{0.04}{\rho_{ref}U_{ref}},$$

où U_{ref} et ρ_{ref} sont une vitesse de référence et une densité de référence de l'écoulement.

Dans ses travaux de thèse, Laurent [89] a proposé une amélioration en remarquant qu'un décentrement était systématiquement introduit dans le calcul de la vitesse normale à l'interface $\tilde{v}_{i+\frac{1}{2}}$, que la solution soit régulière ou non à cause du terme de pression dans la relation (4.16). Elle propose alors une nouvelle définition pour que le schéma dégénère en schéma purement centré en présence d'un écoulement régulier :

$$\tilde{v}_{i+\frac{1}{2}} = \frac{1}{2}n_{i+\frac{1}{2}} \cdot (V_{i+\frac{1}{2},G} + V_{i+\frac{1}{2},D}) - c_2(p_{i+\frac{1}{2},D} - p_{i+\frac{1}{2},G}) \times \Phi_{i+\frac{1}{2}}. \quad (4.18)$$

Cette relation (4.18) est retenue comme la nouvelle définition de la vitesse $\tilde{v}_{i+\frac{1}{2}}$ du schéma AUSM+(P) senseur.

Le schéma AUSM+(P) senseur utilisé dans cette thèse est d'ordre deux [89, 4]. Dans le contexte Volumes Finis, la mise en œuvre d'un schéma d'ordre élevé (d'ordre quatre par exemple) pour calculer les flux convectifs est difficile. De plus, les schémas d'ordre élevé présentent généralement un coût numérique important. Il est généralement plus efficace de travailler avec des schéma d'ordre peu élevé (d'ordre deux par exemple) sur un maillage plus dense que de travailler avec des schémas d'ordre élevé sur un maillage plus grossier. Par exemple, dans [93], Meinke *et al* comparent un schéma d'ordre deux utilisé sur un maillage fin avec un schéma d'ordre six utilisé sur un maillage grossier dans le cas d'une simulation LES d'un jet turbulent. Le schéma d'ordre deux utilisé sur maillage fin se révèle être le plus efficace. Dans les articles [91, 61] des simulations LES ont été réalisées avec succès à l'aide du schéma Senseur d'ordre deux présenté ici.

Nous concluons cette présentation du schéma AUSM+(P) senseur en évoquant le stencil de ce schéma. Les relations (4.12), (4.13) et (4.15) montrent que le flux du schéma AUSM+(P) senseur à l'interface $i + \frac{1}{2}$ ($F_{i+\frac{1}{2}}^c$) est déterminé par les états gauche et droit à l'interface $i + \frac{1}{2}$. Or, la relation (4.14) montre que le calcul de ces états gauche et droit nécessite les solutions numériques sur deux cellules en amont de l'interface (U_{i-1} et U_i) et en aval de l'interface (U_{i+1} et U_{i+2}). Le stencil du flux au niveau d'une face d'indice $i + \frac{1}{2}$ (face verticale) est illustré sur la figure 4.2. De même, le stencil du flux au niveau d'une face d'indice $j + \frac{1}{2}$ (face horizontale) est illustré sur la figure 4.3. On en déduit le stencil pour l'évaluation du bilan des flux sur une cellule (figure 4.4).

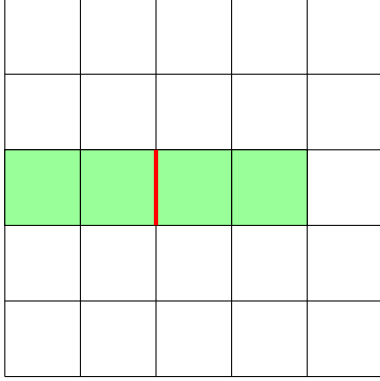


FIGURE 4.2 – Stencil (en vert) pour évaluer le flux au niveau d’une face d’indice $i + \frac{1}{2}$ (en rouge).

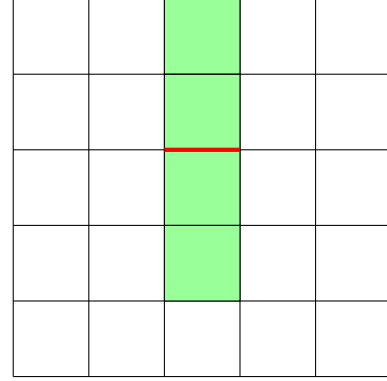


FIGURE 4.3 – Stencil (en vert) pour évaluer le flux au niveau d’une face d’indice $j + \frac{1}{2}$ (en rouge).

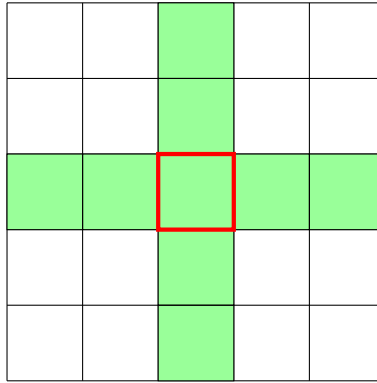


FIGURE 4.4 – Stencil (en vert) pour évaluer le bilan des flux sur une cellule (en rouge).

4.3.2 Schéma de Roe

On désigne par A la matrice Jacobienne des flux convectifs F^c :

$$A = \frac{\partial F^c}{\partial U} \quad (4.19)$$

Le flux de Roe à l’interface d’indice $i + \frac{1}{2}$ est donné par la relation suivante :

$$F_{i+\frac{1}{2}}^c = \frac{1}{2} \left(F^c(U_{i+\frac{1}{2},G}) + F^c(U_{i+\frac{1}{2},D}) \right) - |\tilde{A}_{i+\frac{1}{2}}| \left(U_{i+\frac{1}{2},D} - U_{i+\frac{1}{2},G} \right). \quad (4.20)$$

Dans l’équation (4.20), le terme $\tilde{A}_{i+\frac{1}{2}}$ correspond à la matrice jacobienne A évaluée au niveau des états de Roe $U_{G,D}$:

$$\tilde{A}_{i+\frac{1}{2}} = A(U_{G,D}), \quad (4.21)$$

avec :

$$U_{G,D} = \frac{\sqrt{\rho_{i+\frac{1}{2},G}} U_{i+\frac{1}{2},G} + \sqrt{\rho_{i+\frac{1}{2},D}} U_{i+\frac{1}{2},D}}{\sqrt{\rho_{i+\frac{1}{2},G}} + \sqrt{\rho_{i+\frac{1}{2},D}}}. \quad (4.22)$$

La matrice $\tilde{A}_{i+\frac{1}{2}}$ est diagonalisable, on note λ_p ses valeurs propres, et on note P la matrice de passage. La matrice $|\tilde{A}_{i+\frac{1}{2}}|$ est définie par :

$$|\tilde{A}_{i+\frac{1}{2}}| = P|\Delta|P^{-1}, \text{ avec } |\Delta| = \text{diag}(|\lambda_p|) \quad (4.23)$$

Pour davantage de précisions sur le schéma de Roe, le lecteur peut se reporter à [94].

Extension de la précision spatiale au second ordre

Afin d'accroître la précision du schéma de Roe à l'ordre deux, les états gauche et droit $U_{i+\frac{1}{2},G}$ et $U_{i+\frac{1}{2},D}$ sont reconstruits à l'aide de la méthode MUSCL. Ce choix conduit à une solution précise au second ordre en espace mais elle amène des oscillations numériques dans les régions de forts gradients. Pour remédier à cela, il est nécessaire de coupler la méthode MUSCL à un limiteur de pente, dont le rôle est de réduire l'ordre du schéma à un dans les régions de forts gradients, et de maintenir l'ordre deux dans les zones "régulières".

Limiteur de pente Minmod

La limitation des pentes est effectuée à l'aide du limiteur Minmod. Les états gauche et droit à l'interface d'indice $i + \frac{1}{2}$ sont donnés par :

$$\begin{aligned} U_{i+\frac{1}{2},G} &= U_i + \frac{1}{2} \times \Phi(U_i - U_{i-1}, U_{i+1} - U_i), \\ U_{i+\frac{1}{2},D} &= U_{i+1} - \frac{1}{2} \times \Phi(U_{i+1} - U_i, U_{i+2} - U_{i+1}). \end{aligned} \quad (4.24)$$

Dans la relation (4.24), la fonction Φ représente le limiteur Minmod, qui s'écrit :

$$\Phi(a, b) = \begin{cases} 0 & \text{si } ab < 0 \\ a & \text{si } |a| < |b| \text{ et } ab > 0 \\ b & \text{si } |b| < |a| \text{ et } ab > 0 \end{cases} \quad (4.25)$$

4.4 Flux visqueux

Les flux visqueux sont par nature dissipatifs. Ils sont donc classiquement discrétisés à l'aide d'un schéma centré précis à l'ordre deux. Ce schéma nécessite la connaissance des dérivées de la vitesse et de la température au niveau des faces des cellules. Le calcul des gradients au niveau des faces est basé sur l'idée de Chakravarthy [95]. Dans un premier temps, comme le montre la figure 4.5, un volume de contrôle Ω_c est défini autour de la face (sur cette figure, il s'agit d'une face verticale, d'indice $i + \frac{1}{2}$). Ensuite la formule de Green-Ostrogradski est utilisée pour évaluer le gradient au centre du volume de contrôle, c'est-à-dire au niveau de la face d'indice $i + \frac{1}{2}$. Ainsi, pour une quantité ϕ donnée, on a :

$$\frac{\partial \phi}{\partial x_i} \approx \frac{1}{\Omega_c} \int_{\Omega_c} \frac{\partial \phi}{\partial x_i} d\Omega, \quad (4.26)$$

où $\frac{\partial \phi}{\partial x_i}$ est évalué au centre du domaine Ω_c .

La formule de Green-Ostrogradski est appliquée au second membre de l'équation (4.26) :

$$\begin{aligned} \frac{1}{\Omega_c} \int_{\Omega_c} \frac{\partial \phi}{\partial x_i} d\Omega &= \frac{1}{\Omega_c} \int_{\partial \Omega_c} \phi n_i dS \quad (\text{avec } n_i \text{ la normale dans la direction } x_i) \\ &= \frac{1}{\Omega_c} \sum_{l=1}^6 \phi_l n_{i,l} S_l, \end{aligned} \quad (4.27)$$

avec $n_{i,l}$ la composante dans la direction x_i de la normale à la face l de Ω_c , S_l la surface de la face l de Ω_c et ϕ_l la quantité ϕ au centre de la face l de Ω_c . Pour les faces gauche et droite de Ω_c , la quantité ϕ_l correspond à la quantité ϕ au centre des cellules "classiques". Pour les faces haut et bas de Ω_c , la quantité ϕ_l est interpolée. L'interpolation est réalisée à partir de la grandeur ϕ sur les quatre cellules entourant les faces haut et bas. Le stencil permettant d'évaluer la dérivée de ϕ (et donc le flux visqueux sur ϕ) au niveau d'une face d'indice $i + \frac{1}{2}$ est représenté sur la figure 4.6.

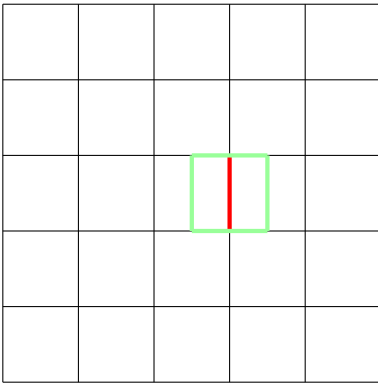


FIGURE 4.5 – Une face d'indice $i + \frac{1}{2}$ (rouge), au niveau de laquelle on évalue la dérivée, et le volume Ω_c (vert) qui l'entoure.

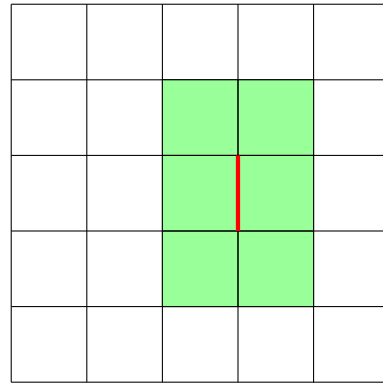


FIGURE 4.6 – Stencil (en vert) pour l'évaluation de la dérivée au niveau d'une face d'indice $i + \frac{1}{2}$ (rouge).

Le même raisonnement est appliqué pour obtenir le stencil permettant de calculer la dérivée d'une quantité ϕ au niveau d'une face d'indice $j + \frac{1}{2}$. Le stencil de la dérivée de ϕ (et donc du flux visqueux de ϕ) au niveau d'une face d'indice $j + \frac{1}{2}$ est représenté sur la figure 4.7. On en déduit le stencil permettant de calculer le bilan des flux visqueux de ϕ sur une cellule (figure 4.8).

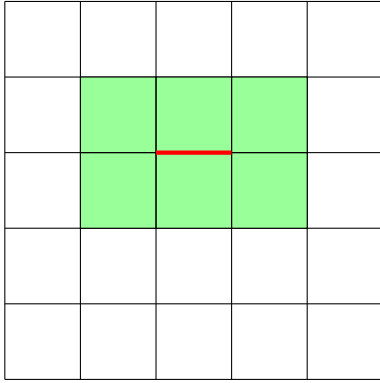


FIGURE 4.7 – Stencil (en vert) pour l'évaluation de la dérivée au niveau d'une face d'indice $j + \frac{1}{2}$ (rouge).

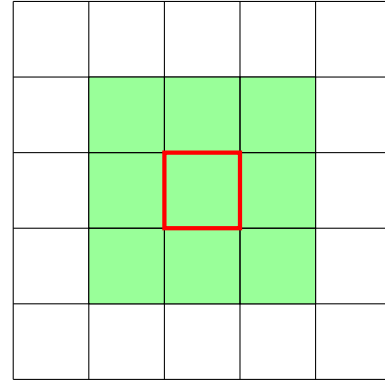


FIGURE 4.8 – Stencil (en vert) pour l'évaluation du bilan des flux visqueux sur une cellule (rouge).

Lorsque les équations de Navier-Stokes sont discrétisées avec nos schémas spatiaux, le bilan des flux est composé du bilan des flux convectifs et du bilan des flux visqueux. Dans ce cas, le stencil du bilan des flux est représenté sur la figure 4.9. Il s'agit de la somme entre le stencil du bilan des flux convectifs (figure 4.4) et le bilan des flux visqueux (figure 4.8).

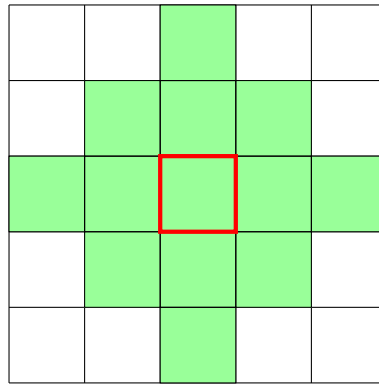


FIGURE 4.9 – Stencil (en vert) pour évaluer le bilan des flux sur une cellule (en rouge) dans le cas des équations de Navier-Stokes discrétisées avec le schéma AUSM+(P) senseur et le schéma des flux visqueux.

On remarque que ce stencil s'étend, au plus, sur deux cellules autour de la cellule sur laquelle on calcule le bilan des flux (cellule rouge). Par conséquent, si la cellule sur laquelle

on calcule le bilan des flux se situe à l'extrémité d'un sous-domaine, le débordement du stencil sur le sous-domaine voisin s'étend, au plus, sur deux cellules ($d = 2$). Pour les schémas NRK2 et NRK3, la taille de la zone tampon est donc égale à deux ; pour les schémas CSRK2 et CSRK3, la taille de la zone tampon est respectivement égale à quatre et six (voir tableau 2.10, page 73).

4.5 Advection d'un tourbillon 2D

Le cas test académique de l'advection 2D d'un tourbillon est relativement simple : il est basé sur les équations d'Euler (fluide non visqueux) et la solution est régulière. Il a été retenu pour effectuer une première évaluation des schémas à pas de temps local proposés, NRK2 et NRK3.

Pour ce cas-test, **ainsi que pour l'ensemble des cas-test présentés dans ce chapitre**, on désigne par $RK2_{dtglobal}$ le schéma RK2 "classique", à pas de temps global (pas de temps uniforme sur le domaine de calcul). De même, on désigne par $RK3_{dtglobal}$ le schéma RK3 "classique", à pas de temps global. **Sauf mention contraire, les schémas NRK2 et NRK3 sont toujours utilisés avec l'étape de correction de flux, y compris pour les études de convergence en temps et en espace.**

4.5.1 Présentation du cas-test

Le cas-test a notamment été décrit par Hu et Shu dans [96]. Pour ce cas-test, toutes les grandeurs sont données sans dimension. Le nombre de Mach M de l'écoulement est fixé à : $M = 0.7$. On considère un champ initial, dans lequel on a :

$$\begin{cases} \rho = 1, \\ u_1 = 0.7, \\ u_2 = 0, \\ p = \frac{1}{\gamma} = 0.714, \end{cases} \quad (4.28)$$

où ρ est adimensionnée par $\rho_0 = 1.2 \text{ kg.m}^{-3}$, les vitesses sont adimensionnées par la vitesse du son c à $T_0 = 300K$, et la pression p est adimensionnée par $p_0 = \rho_0 c^2$. On précise également que les longueurs sont adimensionnées par la longueur de référence $L_0 = 1 \text{ m}$ et que les durées sont adimensionnées par le temps de référence $t_0 = L_0/c$. On superpose à ce champ des perturbations simulant la présence d'un tourbillon. Les perturbations sur ρ , u_1 , u_2 et p sont données par :

$$\left\{ \begin{array}{l} \delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}}, \\ \delta u_1 = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(y_c - y), \\ \delta u_2 = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(x - x_c), \\ \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}}, \end{array} \right. \quad (4.29)$$

avec $\delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}e^{1-r^2}$, la perturbation autour de la température initiale $T = 1$, $\epsilon = 5$ l'intensité du tourbillon, $(x_c = 15, y_c = 10)$ les coordonnées du centre du tourbillon et $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$, la distance entre le point courant et le centre du tourbillon. Ainsi, la solution initiale de ce cas-test est donnée par le système :

$$\left\{ \begin{array}{l} \rho = 1 + \delta\rho \\ u_1 = 0.7 + \delta u_1 \\ u_2 = 0 + \delta u_2 \\ p = \frac{1}{\gamma} + \delta p \end{array} \right. \quad (4.30)$$

La solution exacte de ce cas-test est la simple advection du tourbillon par le champ initial.

4.5.2 Présentation du domaine de calcul

Le domaine de calcul 2D, montré sur la figure 4.10, a une longueur $L = 30$ dans la direction \vec{x} et une largeur $l = 20$ dans la direction \vec{y} . Il est divisé en trois zones de longueur $L' = 10$ et de largeur $l = 20$. Dans la première zone (zone de gauche) et dans la dernière zone (zone de droite), les pas d'espaces dans les directions \vec{x} et \vec{y} (Δx et Δy) sont uniformes : $\Delta x = \Delta y = 0.05$. On précise que ces deux zones contiennent le même nombre de cellules : $nbc_{cell_1} = 8000$. Dans la seconde zone (c'est à dire la zone du milieu), le pas d'espace Δy est uniforme : $\Delta y = 0.05$. Le pas d'espace Δx varie de manière continue, avec $\Delta x = 0.05$ aux bords gauche et droit de la zone, et $\Delta x = 0.025$ au milieu de la zone. Le nombre de cellules dans cette zones vaut $nbc_{cell_2} = 116000$. A cause du raffinement du pas d'espace Δx , la seconde zone contient des cellules présentant une condition *CFL* deux fois plus restrictive que les cellules de la première et la dernière zone. Ainsi, pour les schémas à pas de temps local, le pas de temps Δt est utilisé dans la première et la dernière la zone et le pas de temps $\frac{\Delta t}{2}$ est utilisé dans la seconde zone. Pour les schémas à pas de temps global, le pas de temps $\Delta t/2$ est utilisé dans les trois zones.

Le champ de masse volumique ρ à l'instant initial ainsi que le maillage sont visibles sur la figure 4.10. Sur cette figure, le nombre de point est volontairement réduit pour une meilleur lisibilité. De plus, des traits verticaux en gras ont été tracés au niveau des abscisses $x = 10$ et $x = 20$ pour marquer la séparation entre les différentes zones.

Les conditions aux limites en $x = 0$, $x = 30$, $y = 0$ et $y = 20$ sont des conditions de périodicité.

Le schéma spatial utilisé pour le calcul des flux convectifs est le schéma AUSM+(P) sensé présent dans la partie 4.3.1. Les calculs sont effectués jusqu'au temps final sans dimension $t^f = 35$: le tourbillon effectue un cycle complet d'advection (il revient dans sa position initiale après avoir parcouru tout le domaine de calcul), puis s'arrête à l'abscisse $x = 20$. Cette abscisse correspond, dans le cas d'un schéma à pas de temps local, à une interface entre deux zones de pas de temps différent. Nous souhaitons savoir si le tourbillon est susceptible d'être dégradé au niveau d'une telle interface.

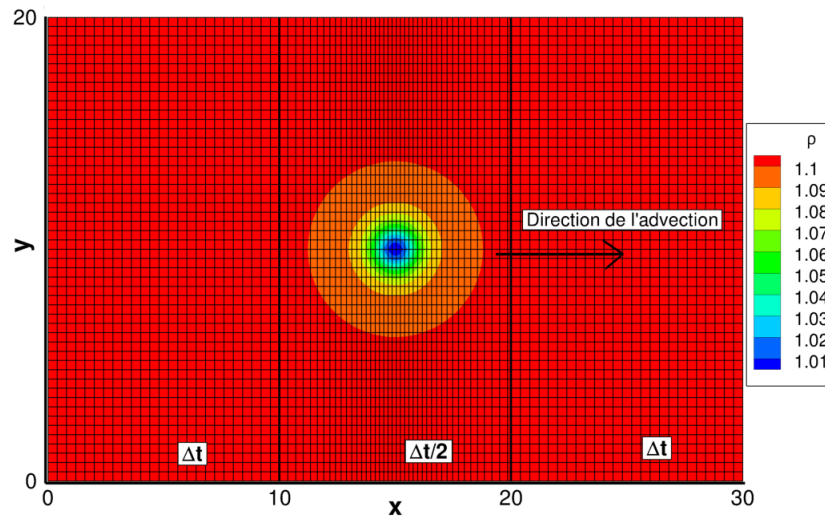


FIGURE 4.10 – Advection du tourbillon : vue du maillage (1 point sur 6 est représenté) et champ de masse volumique ρ à l'instant initial

4.5.3 Étude de la stabilité, du gain en temps de calcul et de la conservativité des schémas à pas de temps local

Avant d'analyser les résultats à proprement parler, trois caractéristiques des schémas à pas de temps local ont été étudiées :

- **la stabilité** : pour chaque schéma, on détermine numériquement la CFL maximale (et le pas de temps maximal) assurant une solution numérique stable et précise. On précise que lorsqu'on parle du pas de temps d'un schéma à pas de temps local, on fait référence au pas de temps le plus grand sur l'ensemble du maillage (celui utilisé dans la première et la dernière zone).
- **le gain en temps de calcul (speedup)** : pour chaque schéma à pas de temps local, nous avons mesuré le speedup par rapport à leur version à pas de temps global.
- **l'erreur sur la conservativité** : pour chaque schéma à pas de temps local, l'erreur commise sur la conservation des variables conservatives a été calculée.

Définition du speedup

On définit le speedup S d'un schéma A par rapport à un schéma B par :

$$S = \frac{T_B - T_A}{T_B}, \quad (4.31)$$

où T_A et T_B sont les temps de simulation associés respectivement aux schémas A et B .

Il est possible de calculer un speedup théorique des schémas à pas de temps local par rapport à leur version à pas de temps global en se basant sur les coûts algorithmiques entre t^n et $t^n + \Delta t$. Pour les schémas à pas de temps local, on néglige les zones tampon dans le but de simplifier les calculs.

Dans le cas des schémas à pas de temps local, la méthode de base de pas de temps Δt , que nous appelons RK_B , est appliquée sur chaque cellule de la première et de la dernière zone. Le coût algorithmique de cette méthode sur chaque cellule est noté \tilde{C} . On rappelle que la première et la dernière zone contiennent $nbcell_1$ cellules chacune. Dans le second sous-domaine, deux applications de la méthode RK_B de pas de temps $\Delta t/2$ sont effectuées donc le coût algorithmique par cellule est égal à $2 \times \tilde{C}$. On rappelle que cette zone contient $nbcell_2$ cellules.

Le coût algorithmique des schémas à pas de temps local est donc :

$$C_{dtlocal} = \underbrace{2 \times nbcell_1 \times \tilde{C}}_{\text{première et dernière zones}} + \underbrace{nbcell_2 \times 2 \times \tilde{C}}_{\text{seconde zone}} \quad (4.32)$$

Pour le schéma $RKB_{dtglobal}$, la méthode RK_B est appliquée partout avec un pas de temps $\Delta t/2$: deux applications de cette méthode sont effectuées entre t^n et $t^n + \Delta t$, sur l'ensemble du maillage. Le coût algorithmique du schéma $RKB_{dtglobal}$ s'écrit donc :

$$C_{dtglobal} = \underbrace{2 \times nbcell_1 \times 2 \times \tilde{C}}_{\text{première et dernière zones}} + \underbrace{nbcell_2 \times 2 \times \tilde{C}}_{\text{seconde zone}} \quad (4.33)$$

On applique la formule du speedup fournie par l'équation (4.31) aux coûts algorithmiques (4.32) et (4.33). On obtient le speedup théorique S_{theo} suivant :

$$\begin{aligned} S_{theo} &= \frac{C_{dtglobal} - C_{dtlocal}}{C_{dtglobal}} \\ &= \frac{nbcell_1}{2 \times nbcell_1 + nbcell_2} \quad (\text{en simplifiant par } 2 \times \tilde{C}) \end{aligned} \quad (4.34)$$

En utilisant les valeurs numériques de $nbcell_1$ et $nbcell_2$ données dans la description du cas-test, on obtient $S_{theo} \approx 29\%$. Sur ce cas-test, on prévoit donc une réduction du temps de calcul d'environ 29% avec les schémas à pas de temps local par rapport à leur version à pas de temps global.

Définition de l'erreur sur la conservativité

On définit l'erreur ϵ_Φ commise sur la conservation d'une variable Φ par :

$$\epsilon_\Phi = \frac{\int_\Omega \Phi(t^f) - \int_\Omega \Phi(t^0)}{\int_\Omega \Phi(t^0)}, \quad (4.35)$$

où $\int_\Omega \Phi(t^f)$ et $\int_\Omega \Phi(t^0)$ sont respectivement les intégrales sur le domaine de calcul Ω de Φ évaluée au temps final t^f et au temps initial t^0 . Vu que les conditions aux limites du domaine de calcul sont de type périodique, toute quantité conservative sortant du domaine de calcul est compensée par la même quantité entrant dans le domaine. Un schéma conservatif doit donc donner une erreur ϵ_Φ nulle (ou proche de zéro machine). Le calcul de l'erreur ϵ_Φ pour les schémas NRK2 et NRK3 permettra d'évaluer le fonctionnement de la stratégie de correction de flux, destinée à rendre ces schémas conservatifs.

Résultats

Les tableaux 4.1 et 4.2 montrent l'erreur sur la conservation des variables ρ , ρu_1 , ρu_2 et ρE pour les différents schémas à pas de temps local.

Schéma	ϵ_ρ	$\epsilon_{\rho u_1}$	$\epsilon_{\rho u_2}$	$\epsilon_{\rho E}$
CSRK2	1.5×10^{-14}	7.6×10^{-15}	4.2×10^{-15}	2.5×10^{-15}
NRK2	1.0×10^{-14}	8.3×10^{-15}	3.5×10^{-15}	6.7×10^{-15}
NRK2 sans correction de flux	4.1×10^{-11}	5.5×10^{-11}	7.8×10^{-11}	6.4×10^{-11}

TABLE 4.1 – Erreur sur la conservation des variables ρ , ρu_1 , ρu_2 et ρE pour le schéma CSRK2 et le schéma NRK2 utilisé avec et sans la technique de correction de flux.

Schéma	ϵ_ρ	$\epsilon_{\rho u_1}$	$\epsilon_{\rho u_2}$	$\epsilon_{\rho E}$
CSRK3	1.1×10^{-14}	1.1×10^{-15}	2.4×10^{-15}	3.2×10^{-15}
NRK3	1.5×10^{-14}	3.8×10^{-15}	3.0×10^{-15}	7.6×10^{-15}
NRK3 sans correction de flux	1.0×10^{-11}	1.2×10^{-10}	4.1×10^{-11}	1.7×10^{-10}

TABLE 4.2 – Erreur sur la conservation des variables ρ , ρu_1 , ρu_2 et ρE pour le schéma CSRK3 et le schéma NRK3 utilisé avec et sans la technique de correction de flux.

Les résultats obtenus sur les tableaux 4.1 et 4.2 confirment que les schémas CSRK2 et CSRK3 sont conservatifs. En effet, pour ces deux schémas, l'erreur sur la conservation des variables ρ , ρu_1 , ρu_2 et ρE est de l'ordre de 10^{-15} , c'est à dire proche du zéro machine. Ces tableaux montrent que des erreurs de l'ordre de 10^{-15} sont également obtenues avec les schémas NRK2 et NRK3 (munis de la stratégie de correction de flux). On en déduit donc que la stratégie de correction de flux fonctionne : elle permet d'assurer la conservativité des schémas. Lorsque la correction de flux n'est pas activée, l'erreur sur la conservation des variables est 10^4 fois plus grande, même si elle reste relativement faible (de l'ordre de 10^{-11}).

Les tableaux 4.3 et 4.4 affichent les résultats de stabilité et de speedup pour les schémas CSRK2, NRK2, CSRK3 et NRK3.

Schéma	$CFL_{max} (\Delta t_{max})$	Speedup par rapport à $RK2_{dtglobal}$
CSRK2	0.25 ($\Delta t_{max} = 0.0070$)	26%
NRK2	0.25 ($\Delta t_{max} = 0.0070$)	28%
NRK2 sans correction de flux	0.25 ($\Delta t_{max} = 0.0070$)	28%

TABLE 4.3 – CFL maximale (pas de temps maximal) et speedup par rapport au schéma $RK2_{dtglobal}$ pour les schémas CSRK2 et NRK2.

Schéma	$CFL_{max} (\Delta t_{max})$	Speedup par rapport à $RK3_{dtglobal}$
CSRK3	1.1 ($\Delta t_{max} = 0.029$)	25%
NRK3	1.1 ($\Delta t_{max} = 0.029$)	28%
NRK3 sans correction de flux	1.1 ($\Delta t_{max} = 0.029$)	28%

TABLE 4.4 – CFL maximale (pas de temps maximal) et speedup par rapport au schéma $RK3_{dtglobal}$ pour les schémas CSRK3 et NRK3.

Les tableaux 4.3 et 4.4 montrent que le speedup des schémas à pas de temps local par rapport à leur version à pas de temps global est proche de 29% (le speedup théorique). Ceci tend à montrer que l'implémentation des schémas à pas de temps local est correcte. Remarquons que pour les schémas NRK2 et NRK3, le même speedup est obtenu avec et

sans l'étape de correction de flux : cette dernière a une influence très faible sur les temps de calcul pour ce cas-test.

Enfin, remarquons que les schémas à pas de temps local basés sur la méthode RK3 sont beaucoup plus stables que les schémas à pas de temps local basés sur la méthode RK2. Les schémas CSRK3 et NRK3 fournissent une solution stable et précise jusqu'à $CFL = 1.1$, tandis que pour les schémas CSRK2 et NRK2, le nombre CFL ne peut excéder 0.25. Ceci rend les schémas CSRK3 et NRK3 beaucoup plus performants que les schémas CSRK2 et NRK2. Nous avons mesuré un speedup d'environ 77% pour les schémas CSRK3 et NRK3 par rapport au schéma à pas de temps global $RK2_{dtglobal}$. Ce résultat valide notre choix de ne pas nous limiter à des schémas basés sur la méthode RK2.

4.5.4 Résultats des simulations

Les figures 4.11 et 4.12 montrent respectivement le profil de la masse volumique ρ et de la vitesse dans la direction \vec{y} (u_2) à $y = 10$ et à l'instant final t^f pour chaque schéma à pas de temps local et pour une solution de référence.

La solution de référence mentionnée sur ces figures est obtenue avec le schéma $RK3_{dtglobal}$. Il a été utilisé avec le pas de temps $\Delta t = 0.0035$, ce qui conduit à un nombre CFL maximal égal 0.25. Le pas de temps $\Delta t = 0.007$ a donc été utilisé pour les schémas CSRK2 et NRK2 ; le pas de temps $\Delta t = 0.029$ ($CFL = 1.1$) a été utilisé pour les schéma CSRK3 et NRK3.

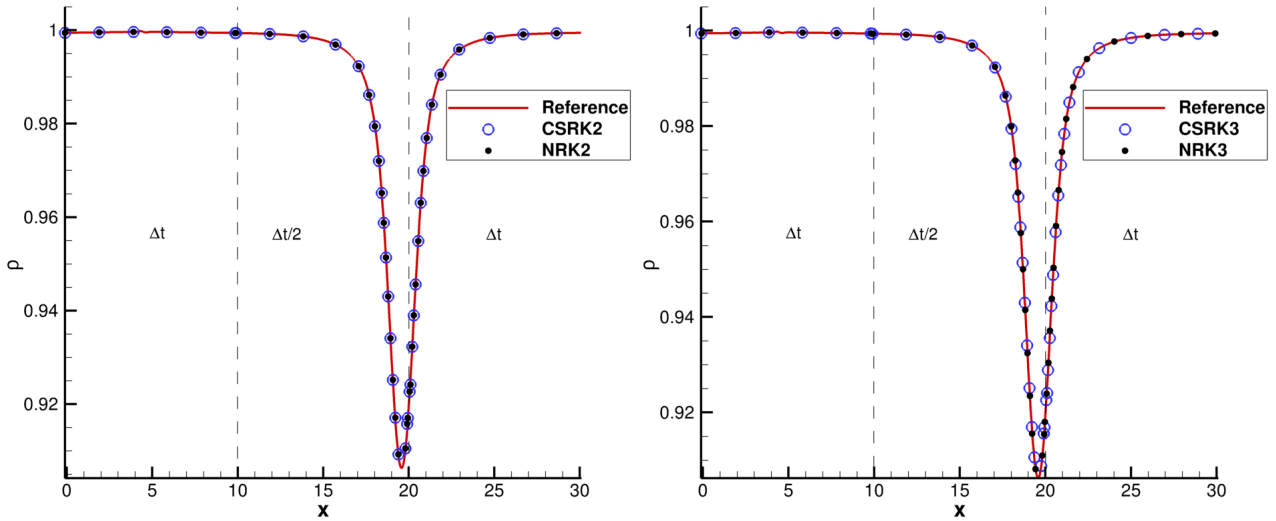


FIGURE 4.11 – Profil de la masse volumique ρ à $y = 10$ et à t^f . **A gauche** : schémas NRK2 et CSRK2. **A droite** : schémas NRK3 et CSRK3.

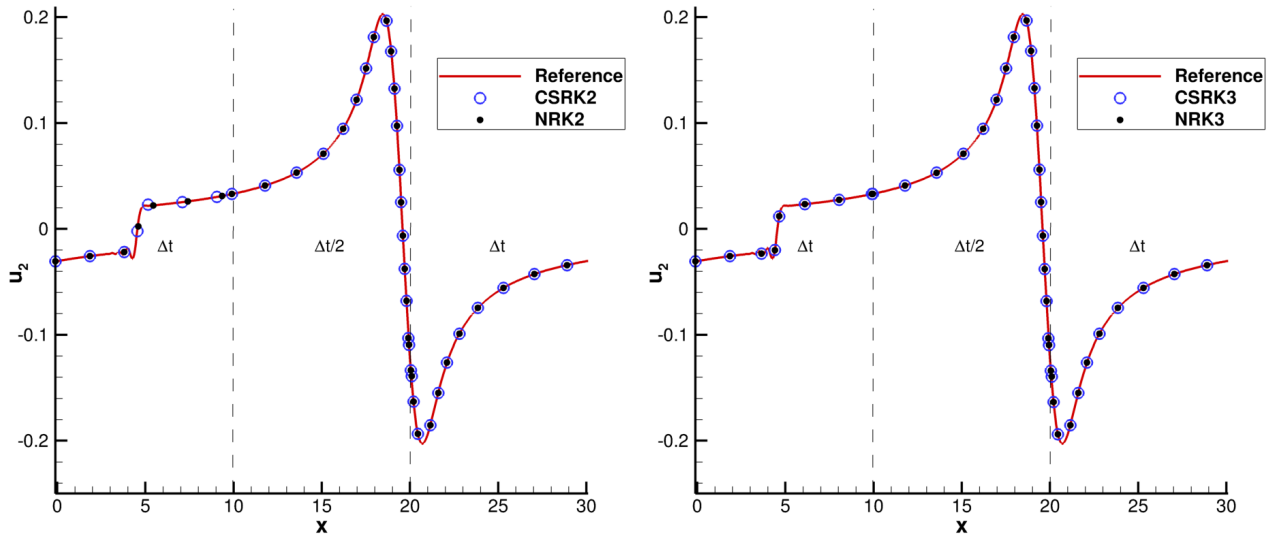


FIGURE 4.12 – Profil de la vitesse u_2 à $y = 10$ et à t^f . **A gauche** : schémas NRK2 et CSRK2. **A droite** : schémas NRK3 et CSRK3.

Les figures 4.11 et 4.12 montrent un très bon accord entre la solution de référence et les solutions issues des différents schémas à pas de temps local. Les solutions obtenues avec les schémas à pas de temps local ne semblent pas être dégradées au niveau des interfaces entre zones de pas de temps différents.

Afin de mener une étude plus poussée concernant la précision, les grandeurs $\frac{\partial \rho}{\partial x}$ et $\frac{\partial^2 \rho}{\partial x^2}$ sont calculées pour la solution de référence et pour les schémas à pas de temps local. Des résultats très similaires ont été obtenus pour tous les schémas à pas de temps local. Ainsi, seuls les résultats obtenus avec le schéma NRK3 sont présentés. La figure 4.13 montre l'évolution de $\frac{\partial \rho}{\partial x}$ (à gauche) et $\frac{\partial^2 \rho}{\partial x^2}$ (à droite) dans la région $[-17.5, 22.5]$, centrée autour de l'abscisse $x = 20$ (qui correspond à une interface entre deux zones de pas de temps différents). Cette région est particulièrement intéressante car elle est le siège d'importantes variations de la masse volumique ρ dans la direction \vec{x} , comme le montre la figure 4.11.

La figure 4.13 montre que les dérivées de la masse volumiques obtenues pour le schéma NRK3 sont conformes à celles obtenues pour la solution de référence. Ceci confirme la bonne précision du schéma NRK3 et des autres schémas à pas de temps local (pour lesquels des résultats très similaires ont été obtenus), en particulier au niveau des interfaces entre deux zones de pas de temps différents.

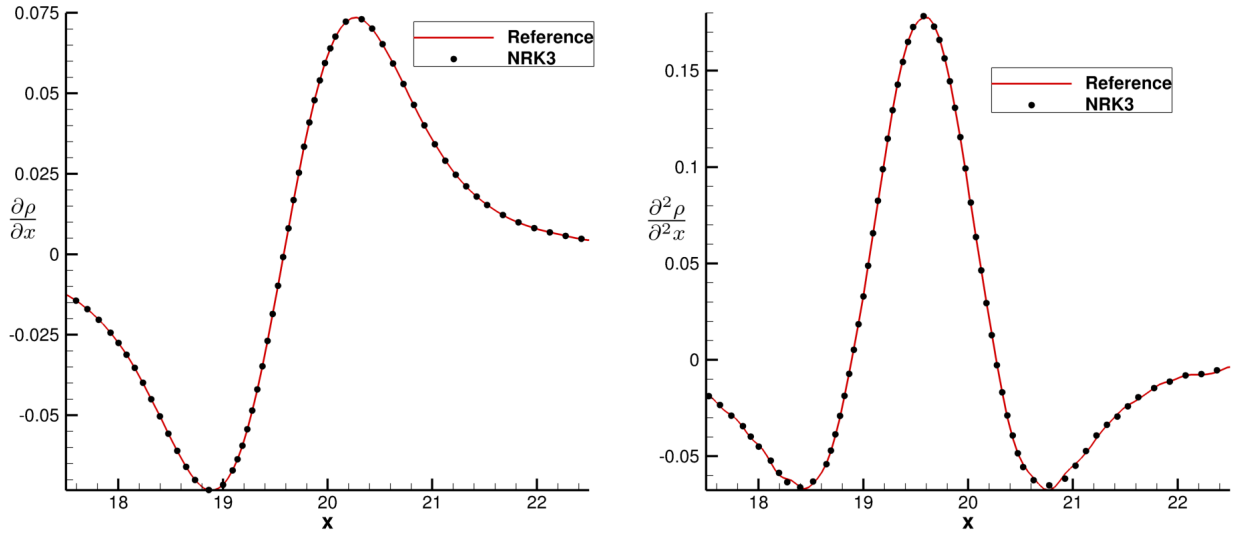


FIGURE 4.13 – Profil de $\frac{\partial \rho}{\partial x}$ (à gauche) et de $\frac{\partial^2 \rho}{\partial^2 x}$ (à droite), à $y = 10$ et à t^f .

4.5.5 Vérification de l'ordre temporel

La figure 4.14 montre le résultat de l'étude d'ordre temporel menée pour les différents schémas à pas de temps local. Pour réaliser cette étude, une solution de référence a été calculée pour chaque schéma à pas de temps local avec un pas de temps $\Delta t = \Delta t_{ref} = 10^{-4}$. Cette solution a été calculée sur le maillage décrit dans la partie 4.5.2. Trois autres simulations avec $\Delta t = 2\Delta t_{ref}$, $\Delta t = 4\Delta t_{ref}$ et $\Delta t = 8\Delta t_{ref}$ sont réalisées (le maillage reste identique). Pour chacune des trois simulations, l'erreur par rapport à la masse volumique obtenue avec la solution de référence (ρ_{ref}) est calculée en norme L_2 . Cette erreur est notée $\|\rho - \rho_{ref}\|_2$ sur la figure 4.14. Le tracé, en échelle logarithmique, de l'erreur en fonction du pas de temps donne une droite de pente égale à l'ordre temporel du schéma.

Pour chaque schéma à pas de temps local, la pente obtenue est en accord avec l'ordre théorique. Pour les schémas CSRK2, NRK2 et CSRK3, en théorie précis à l'ordre deux, on obtient une droite de pente deux. Remarquons que des erreurs quasiment identiques ont été obtenues pour les schémas CSRK2 et NRK2. Concernant le schéma NRK3, en théorie d'ordre trois, la droite obtenue est bien de pente trois.

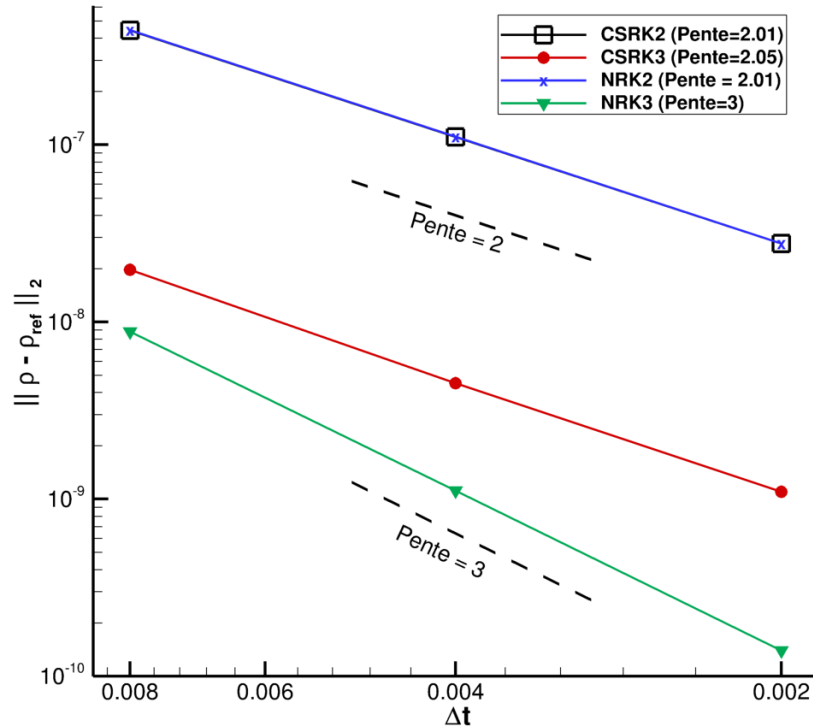


FIGURE 4.14 – Étude de l’ordre temporel des différents schémas à pas de temps local sur le cas-test de l’advection 2D d’un tourbillon.

4.5.6 Vérification de l’ordre spatial

Le but de cette étude est de vérifier que l’ordre spatial n’est pas dégradé par les schémas à pas de temps local. Pour cette étude, les pas d’espaces Δx et Δy sont uniformes dans les trois zones (il n’y a pas de raffinement spatial dans la seconde zone, comme sur le maillage précédent). Pour chacun des schémas à pas de temps local, la solution de référence est calculée avec le pas de temps $\Delta t = 9.10^{-4}$, sur un maillage de résolution $\Delta x = \Delta y = 0.013$. Trois autres simulations sont réalisées sur des maillages de résolution $(2\Delta x, 2\Delta y)$, $(3\Delta x, 3\Delta y)$, $(4\Delta x, 4\Delta y)$, en conservant le même pas de temps. Pour chacune des trois simulations, l’erreur par rapport à la masse volumique obtenue avec la solution de référence (ρ_{ref}) est calculée en norme L_2 . Cette erreur est notée $\|\rho - \rho_{ref}\|_2$ sur la figure 4.15, qui montre le résultat de l’étude de l’ordre spatial pour les différents schémas. Le tracé, en échelle logarithmique, de l’erreur en fonction du pas d’espace donne une droite dont la pente est égale à l’ordre spatial du schéma. On précise que la stratégie de correction de flux a été activée pour les schémas NRK2 et NRK3.

Des erreurs très similaires ont été obtenues pour les différents schémas. Pour chacun des schémas, la droite obtenue est de pente environ égale à deux. Ceci est cohérent avec l’ordre théorique, égal à deux, du schéma AUSM+(P) sensor.

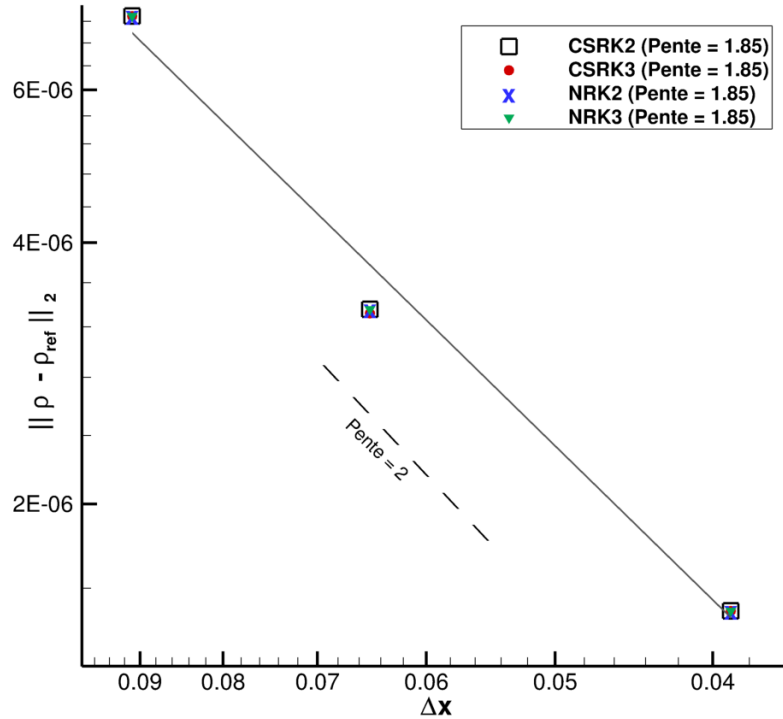


FIGURE 4.15 – Étude de l'ordre spatial des différents schémas à pas de temps local sur le cas-test de l'advection 2D d'un tourbillon.

4.6 Tube à choc de Sod

Ce cas-test, initialement proposé par Sod [97] (1978), a été sélectionné pour évaluer le comportement des schémas à pas de temps local sur un cas d'écoulement de fluide non visqueux comportant des discontinuités.

4.6.1 Présentation du cas-test et du domaine de calcul

Ce problème d'ordinaire 1D a été résolu en deux dimensions, en utilisant des conditions de périodicité dans la direction \vec{y} pour le "rendre" mono-dimensionnel. En effet, le code de calcul utilisé pour ces travaux de thèse ne résout que des écoulements 2D ou 3D. Étant donné qu'il "ne se passe rien" dans la direction \vec{y} (invariance de la solution dans cette direction), on donnera uniquement les informations relatives à la direction \vec{x} . A l'instant initial, le champ de vitesse est nul. La densité et la pression à l'instant initial sont donnés par :

- $\rho(x) = 1, p(x) = 1$ for $x \leq 9$,
- $\rho(x) = 0.125, p(x) = 0.1$ for $x > 9$,

où ρ est adimensionnée par $\rho_0 = 1.2 \text{ kg.m}^{-3}$, et la pression p est adimensionnée par $p_0 = \rho_0 c^2$, avec c la vitesse du son à $T_0 = 300K$. Les longueurs sont adimensionnées par la

longueur de référence $L_0 = 1\text{ m}$ et les durées sont adimensionnées par le temps de référence $t_0 = L_0/c$. Le domaine de calcul est rectangulaire, de longueur $L = 18$. Il est divisé en trois zones de même longueur $L = 6$. Dans la première et la dernière zone, le pas d'espace dans la direction \vec{x} est uniforme : $\Delta x = 0.01$. Dans la seconde zone, le pas d'espace Δx varie de manière progressive, avec $\Delta x = 0.01$ au niveau des bords gauche et droit de la zone, et $\Delta x = 0.005$ au milieu de la zone. Ainsi, pour les schémas à pas de temps local, le pas de temps Δt est utilisé dans la première et la dernière zone, tandis que le pas de temps $\frac{\Delta t}{2}$ est utilisé dans la seconde zone. Pour les schémas à pas de temps global, le pas de temps $\frac{\Delta t}{2}$ est utilisé sur l'ensemble du domaine de calcul. La figure 4.16 montre le profil de la masse volumique ρ à l'instant initial.

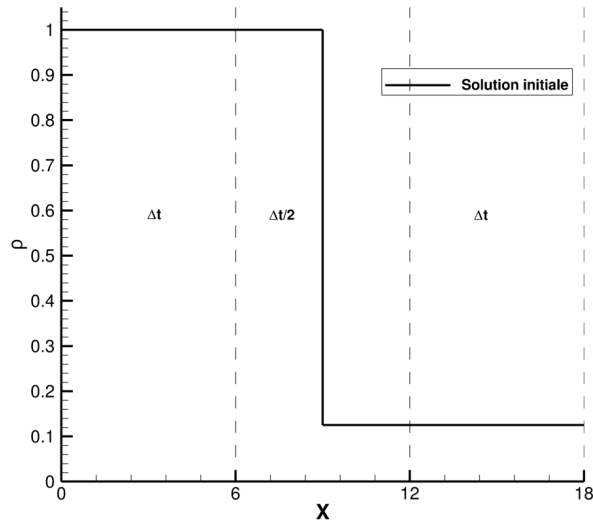


FIGURE 4.16 – Profil de la masse volumique ρ à l'instant initial.

Les conditions aux limites dans la direction \vec{x} sont des conditions de Neumann. Le schéma utilisé pour calculer les flux convectif est le schéma de Roe.

4.6.2 Étude numérique de la stabilité et du speedup des schémas à pas de temps local

Pour ce cas-test, seuls la stabilité et le speedup des schémas à pas de temps ont été étudiés. L'erreur commise sur la conservation des variables conservatives n'a pas été calculée. En effet, étant donné que les conditions aux limites sont des conditions de Neumann, les quantités sortant du domaine ne sont pas compensées par les quantités entrantes : on ne s'attend donc pas à obtenir une valeur remarquable (en l'occurrence une valeur nulle) pour l'erreur commise sur la conservation des variables.

Sur ce cas-test, le nombre de cellules dans la première et la dernière zone est : $nbc_{cell_1} = 72000$; le nombre de cellules dans la seconde zone est : $nbc_{cell_2} = 90000$. En appliquant la formule (4.34), le speedup théorique sur ce cas-test est : $S_{theo} \approx 30\%$.

Les tableaux 4.5 et 4.6 montrent que pour chaque schéma à pas de temps local, le speedup mesuré est en bon accord avec le speedup théorique d'environ 30%. Comme observé sur le premier cas-test de l'advection 2D du tourbillon, désactiver l'étape de correction de flux ne change pas le speedup.

Les tableaux 4.5 et 4.6 montrent également que les schémas à pas de temps local basés sur la méthode RK3 (CSRK3 et NRK3) ont une meilleure stabilité que les schémas à pas de temps local basés sur la méthode RK2 (CSRK2 et NRK2). En effet, les schémas CSRK3 et NRK3 fournissent une solution stable et précise jusqu'à $CFL = 0.90$, tandis que pour les schémas CSRK2 et NRK2, le nombre CFL maximal est de 0.70. Cette différence de stabilité est cependant moins marquée que dans le cas-test précédent (advection 2D d'un tourbillon). Ceci rend les schémas à pas de temps local basés sur la méthode RK3 légèrement plus performants que les schémas à pas de temps local basés sur la méthode RK2. En effet, nous avons mesuré un speedup de 38% pour les schémas CSRK3 et NRK3 par rapport au schéma $RK2_{dtglobal}$. Les schémas CSRK2 et NRK2 ne mènent qu'à un speedup de 26% ou 28% par rapport au schéma $RK2_{dtglobal}$.

	CFLmax (Δt_{max})	Speedup par rapport à $RK2_{dtglobal}$
CSRK2	0.70 ($\Delta t_{max} = 0.0032$)	26%
NRK2	0.70 ($\Delta t_{max} = 0.0032$)	28%
NRK2 sans correction de flux	0.70 ($\Delta t_{max} = 0.0032$)	28%

TABLE 4.5 – Stabilité et speedup obtenus avec les schéma NRK2 et CSRK2.

	CFLmax (Δt_{max})	Speedup par rapport à $RK3_{dtglobal}$
CSRK3	0.90 ($\Delta t_{max} = 0.004$)	25%
NRK3	0.90 ($\Delta t_{max} = 0.004$)	27%
NRK3 sans correction de flux	0.90 ($\Delta t_{max} = 0.004$)	27%

TABLE 4.6 – Stabilité et speedup obtenus avec les schéma NRK3 et CSRK3.

4.6.3 Résultats des simulations

La figure 4.17 montre le profil de la masse volumique ρ à l'instant final $t^f = 4.8$, pour chaque schéma à pas de temps local et pour une solution de référence. Entre $x = 3$ et $x = 9$, on observe la région de détente, à $x \approx 13.5$ et $x \approx 17$ on observe respectivement la discontinuité de contact et le choc. La figure 4.18 montre le profil de la vitesse dans la direction \vec{x} (u_1) au même instant t^f .

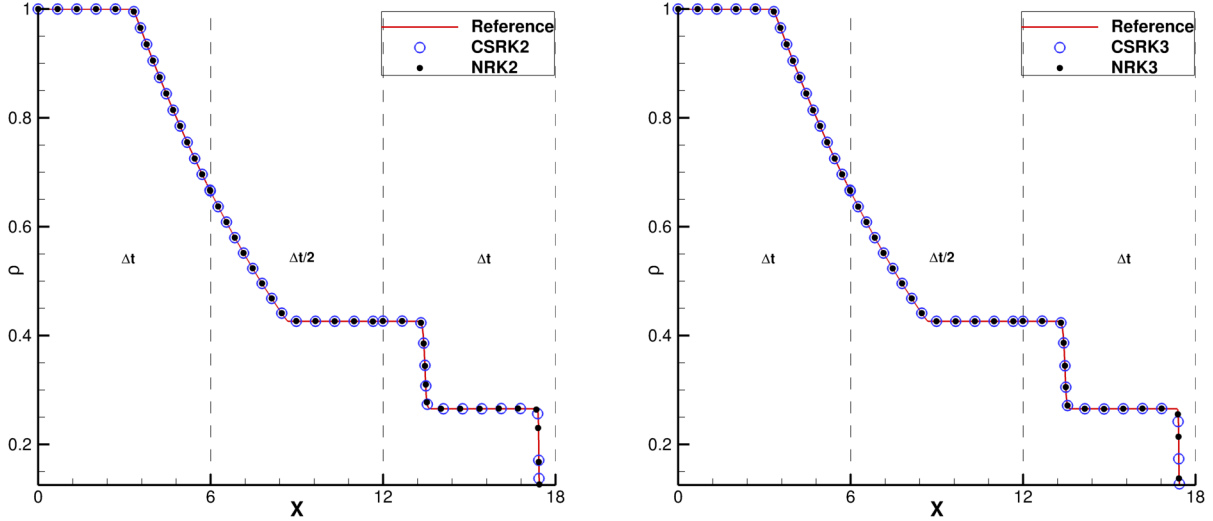


FIGURE 4.17 – Profil de la masse volumique ρ à l'instant final t^f . **A gauche** : schémas NRK2 et CSRK2. **A droite** : Schémas NRK3 et CSRK3.

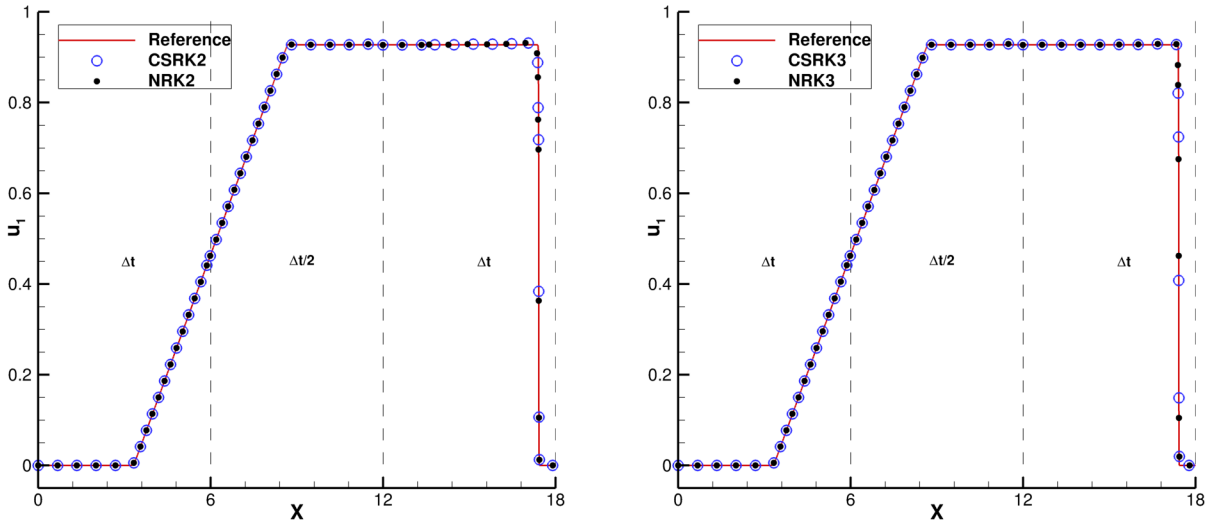


FIGURE 4.18 – Profil de la vitesse selon \vec{x} (u_1) à l'instant final t^f . **A gauche** : schémas NRK2 et CSRK2. **A droite** : Schémas NRK3 et CSRK3.

La solution de référence est obtenue avec le schéma $RK3_{dtglobal}$ de pas de temps $\Delta t = 5.10^{-4}$ (le nombre CFL maximal sur le maillage vaut $CFL = 0.2$). Le pas de temps utilisé par les schémas à pas de temps local est leur pas de temps maximal utilisable, affiché dans la colonne " $CFL_{max}(\Delta t_{max})$ " des tableaux 4.5 et 4.6.

Les figures 4.17 et 4.18 montrent un très bon accord entre les différents schémas à pas de temps local et la solution de référence.

Nous avons également voulu étudier sur ce cas-test l'influence du caractère conservatif des schémas sur la solution obtenue. Pour cela, le schéma NRK3 a été utilisé avec et sans l'étape de correction de flux. Les deux solutions ont été comparées à une solution de référence. Cette dernière est obtenue avec le schéma $RK3_{dtglobal}$ de pas de temps $\Delta t = 5.10^{-4}$ ($CFL = 0.2$). Pour le schéma NRK3, le pas de temps est fixé à sa valeur maximale utilisable, affichée dans la colonne " $CFL_{max}(\Delta t_{max})$ " du tableau 4.6. Les simulations sont réalisées jusqu'à l'instant $t = 1.7$. A cet instant, le choc de masse volumique ρ est approximativement situé au niveau de l'abscisse $x = 12$, qui correspond à une interface entre deux zones de pas différents. Le schéma NRK3 privé de correction de flux est non conservatif au niveau de cette interface, ce qui peut engendrer une mauvaise position du choc. La figure 4.19 montre le choc de masse volumique ρ pour la solution de référence et pour le schéma NRK3 utilisé avec et sans l'étape de correction de flux. Cette figure montre que schéma NRK3 utilisé sans correction de flux (donc non conservatif) prédit une position correcte du choc. Ceci rejoint les travaux de Tang et Warnecke [21], qui montrent que leur schéma à pas de temps local non conservatif fournit des solutions correctes sur plusieurs cas-test d'écoulements présentant des discontinuités.

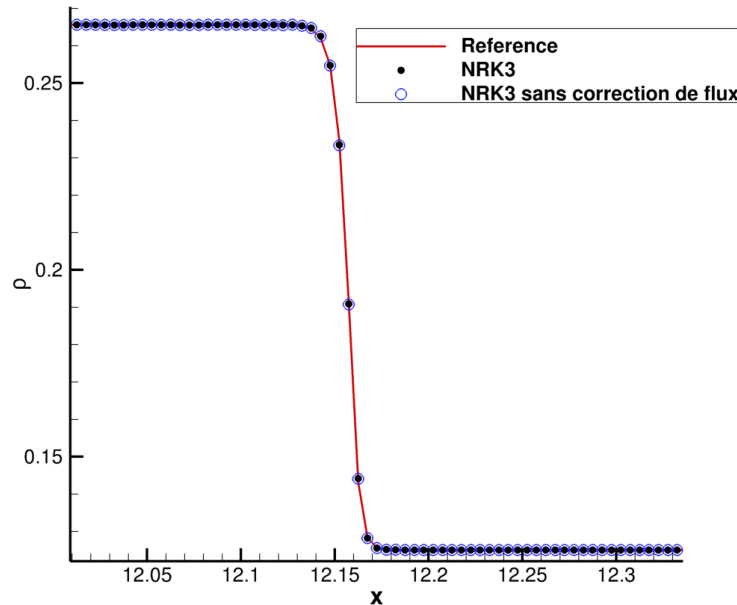


FIGURE 4.19 – Profil de la masse volumique ρ à l'instant $t = 1.7$ pour la solution de référence et le schéma NRK3 utilisé avec et sans l'étape de correction de flux.

4.6.4 Vérification de l'ordre temporel

Les résultats de l'étude de l'ordre temporel sont affichés sur la figure 4.20. Pour cette étude, nous avons suivi la même méthodologie que celle décrite dans la partie 4.5.5. Pour chaque schéma à pas de temps local, une solution de référence a été calculée avec le pas de temps $\Delta t = \Delta t_{ref} = 2.5 \cdot 10^{-4}$. Trois autres solutions sont calculées avec $\Delta t = 2\Delta t_{ref}$, $\Delta t = 4\Delta t_{ref}$ et $\Delta t = 8\Delta t_{ref}$, en gardant le même maillage. Sur ce cas-test aussi, l'ordre temporel obtenu expérimentalement est conforme à l'ordre temporel théorique pour chaque schéma. En effet, la figure 4.20 montre que la courbe d'erreur a une pente égale à deux pour les schéma CSRK2, NRK2 et CSRK3 qui sont en théorie, d'ordre deux. La courbe d'erreur a une pente égale à trois pour le schéma NRK3, en théorie d'ordre trois.

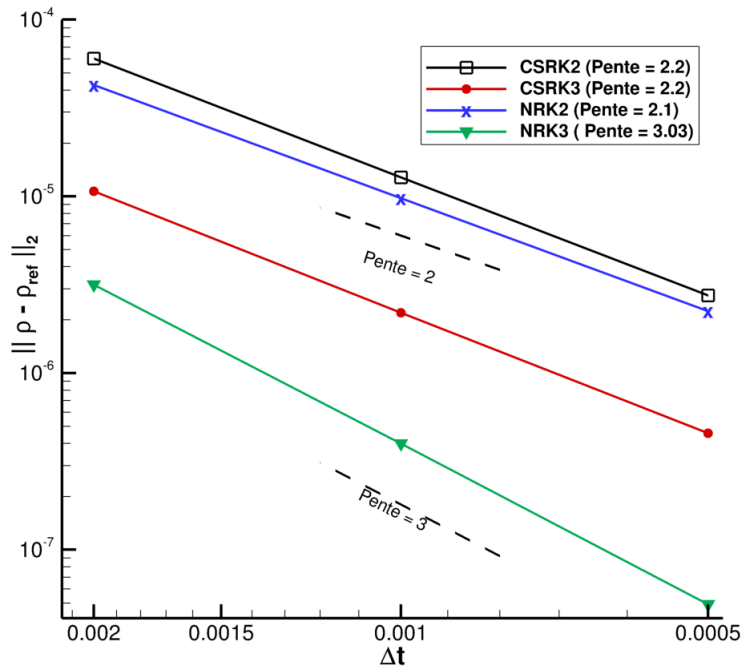


FIGURE 4.20 – Étude de l'ordre temporel des différents schémas à pas de temps local sur le cas-test du tube à choc de Sod.

4.7 Tourbillon de Taylor-Green 3D

L'objectif de ce cas-test est d'évaluer les schémas à pas de temps local sur un cas d'écoulement plus complexe que les précédents. En effet, sur ce cas-test l'écoulement est gouverné par les équations de Navier-Stokes, il est 3D et turbulent. Précisons que ce cas-test est assez populaire dans le domaine de la CFD et fait partie des cas-test du Workshop on High-Order CFD Methods [98].

4.7.1 Présentation du cas-test et du domaine de calcul

Dans un soucis de conformité avec les données du Workshop on High-Order CFD Methods [98], les variables intervenant dans ce cas-test sont dimensionnées. Les conditions initiales sont données par :

$$\left\{ \begin{array}{l} p = p_0 + \frac{\rho_0 U_0^2}{16} (\cos(2x/L_0) + \cos(2y/L_0)) (\cos(2z/L_0) + 2), \\ u_1 = U_0 \sin(x/L_0) \cos(y/L_0) \cos(z/L_0), \\ u_2 = -U_0 \cos(x/L_0) \sin(y/L_0) \cos(z/L_0), \\ u_3 = 0, \\ \rho = p/(RT_0), \end{array} \right. \quad (4.36)$$

où la température de référence, la pression de référence et la masse volumique de référence sont respectivement fixées à $T_0 = 294K$, $p_0 = 101183Pa$ et $\rho_0 = 1.2kg.m^{-3}$. Le nombre de Mach basé sur la vitesse de référence U_0 et la vitesse du son c est fixé à $M = 0.1$. La vitesse de référence U_0 vaut donc : $U_0 = 34m.s^{-1}$. Enfin, le nombre de Reynolds vaut $Re = 1600$, ce qui permet d'en déduire la longueur de référence $L_0 = 7 \times 10^{-4}m$.

A partir de la solution initiale, l'écoulement est simulé jusqu'à l'instant final $t^f = 14 L_0/U_0$. La figure 4.21 montre la topologie de l'écoulement à trois instant successifs grâce à une isosurface du critère Q . On précise que le critère Q est un indicateur des structures turbulentes, défini par la formule :

$$Q = \frac{1}{2} [(tr(\nabla V))^2 - tr(\nabla V \cdot \nabla V)], \quad (4.37)$$

avec $\vec{V} = (u_1, u_2, u_3)$ le vecteur vitesse de l'écoulement. A partir de la solution initiale (figure de gauche), qui comporte quatre tourbillons contra-rotatifs de même dimension, l'écoulement développe rapidement une large gamme d'échelles turbulentes.

Il est possible d'analyser l'évolution temporelle de plusieurs quantité : les variables primitives, l'énergie cinétique turbulente et l'enstrophie. Dans [99], les auteurs montrent que cette dernière quantité est la plus sensible à la précision de la méthode numérique. Ainsi, sur ce cas-test, on se concentre uniquement sur l'analyse de l'enstrophie ϵ , définie par :

$$\epsilon = \frac{L_0}{U_0^2 \rho_0} \int_{\Omega} \rho \vec{w} \cdot \vec{w} d\Omega, \quad (4.38)$$

où \vec{w} est la vorticit e d efinie par le vecteur : $\vec{w} = \vec{\nabla} \wedge \vec{V}$.

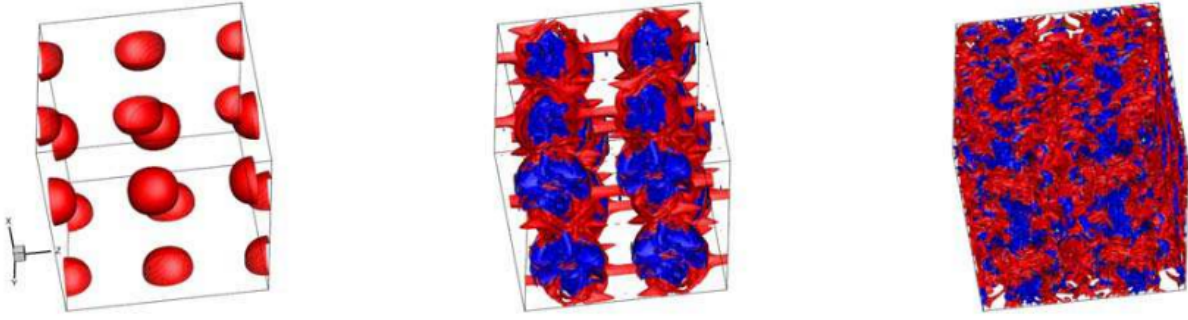


FIGURE 4.21 – Isosurface du critère $Q = -20(U_0/L_0)^2$ colorée par la fluctuation de la pression : $t = 0, 8, 14 \times L_0/U_0$ (de gauche à droite).

Le domaine de calcul est un cube de côté $2\pi L_0$ divisé en trois zones de mêmes dimensions, comme le montre la figure 4.22.

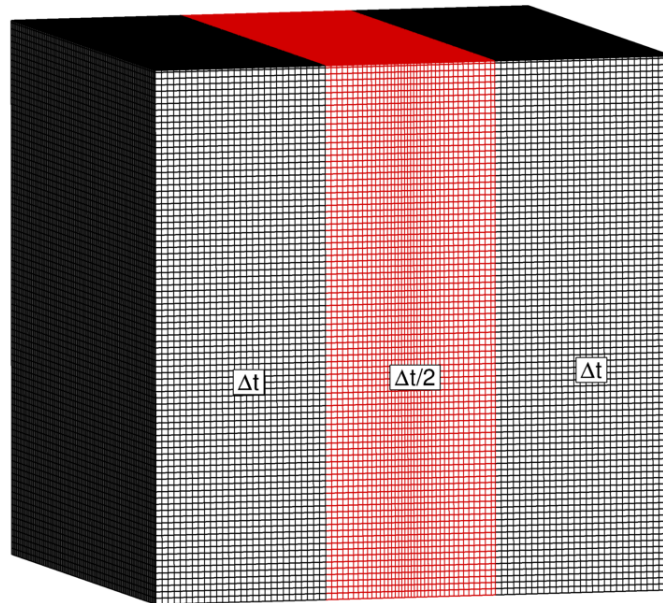


FIGURE 4.22 – Domaine de calcul et pas de temps associé à chaque zone (pour un schéma à pas de temps local).

Dans la première et la dernière zone, les pas d'espace dans les directions \vec{x} , \vec{y} et \vec{z} sont égaux : $\Delta x = \Delta y = \Delta z = 0.025 L_0$. Il y a 256 cellules dans les directions \vec{y} et \vec{z} , et 84 cellules dans la direction \vec{x} . Dans la seconde zone, on fait varier le pas d'espace Δx de manière continue. Au milieu de la zone, le pas d'espace est deux fois plus faible ($\Delta x = 0.0125 L_0$) qu'au niveau des bords ($\Delta x = 0.025 L_0$). Dans cette zone, il y a toujours 256 cellules dans les directions \vec{y} et \vec{z} , et 112 cellules dans la direction \vec{x} . Ainsi, pour les schémas à pas de

temps local, le pas de temps Δt est utilisé dans la première et la dernière zone, le pas de temps $\frac{\Delta t}{2}$ est utilisé dans la seconde zone. Pour les schéma à pas de temps global le pas de temps $\frac{\Delta t}{2}$ est utilisé sur l'ensemble du domaine de calcul. Toutes les conditions aux limites sont périodiques.

Le schéma utilisé pour le calcul des flux convectifs est la schéma AUSM+(P) senseur présenté dans la partie 4.3.1. Pour le calcul des flux visqueux, le schéma centré d'ordre deux décrit dans la partie 4.4 est utilisé.

4.7.2 Étude numérique de la stabilité, du speedup et de l'erreur sur la conservativité

Sur ce cas-test, l'erreur sur la conservation des grandeurs ρ , ρu_1 , ρu_2 , ρu_3 et ρE est calculée. Toutes les conditions aux limites sont périodiques, donc un schéma conservatif doit mener à une erreur sur la conservation de ces grandeurs théoriquement nulle (ou très proche du zéro machine en pratique). Les tableaux 4.7 et 4.8 montrent l'erreur sur la conservation des variables ρ , ρu_1 , ρu_2 , ρu_3 et ρE pour les différents schémas à pas de temps local.

Schéma	ϵ_ρ	$\epsilon_{\rho u_1}$	$\epsilon_{\rho u_2}$	$\epsilon_{\rho u_3}$	$\epsilon_{\rho E}$
CSRK2	1, 6.10 ⁻¹⁴	4, 6.10 ⁻¹⁵	4, 1.10 ⁻¹⁵	3, 5.10 ⁻¹⁵	1, 5.10 ⁻¹⁵
NRK2	1, 0.10 ⁻¹⁴	7, 7.10 ⁻¹⁵	8, 1.10 ⁻¹⁵	2, 2.10 ⁻¹⁵	6, 8.10 ⁻¹⁵
NRK2 sans correction de flux	1, 7.10 ⁻¹¹	2, 9.10 ⁻¹¹	4, 3.10 ⁻¹¹	4, 4.10 ⁻¹¹	7, 3.10 ⁻¹¹

TABLE 4.7 – Erreur sur la conservation des variables ρ , ρu_1 , ρu_2 , ρu_3 et ρE pour le schéma CSRK2 et le schéma NRK2 utilisé avec et sans la technique de correction de flux.

Schéma	ϵ_ρ	$\epsilon_{\rho u_1}$	$\epsilon_{\rho u_2}$	$\epsilon_{\rho u_3}$	$\epsilon_{\rho E}$
CSRK3	1, 0.10 ⁻¹⁴	2, 5.10 ⁻¹⁵	4, 9.10 ⁻¹⁵	5, 1.10 ⁻¹⁵	3, 9.10 ⁻¹⁵
NRK3	1, 2.10 ⁻¹⁴	2, 9.10 ⁻¹⁵	3, 8.10 ⁻¹⁵	8, 2.10 ⁻¹⁵	9, 6.10 ⁻¹⁵
NRK3 sans correction de flux	2, 6.10 ⁻¹¹	6, 7.10 ⁻¹⁰	7, 1.10 ⁻¹¹	5, 4.10 ⁻¹¹	3, 7.10 ⁻¹⁰

TABLE 4.8 – Erreur sur la conservation des variables ρ , ρu_1 , ρu_2 , ρu_3 et ρE pour le schéma CSRK3 et le schéma NRK3 utilisé avec et sans la technique de correction de flux.

Les tableaux 4.7 et 4.8 montrent que l'erreur de conservativité obtenue avec les schémas NRK2 et NRK3 est du même ordre de grandeur que celle obtenue avec les schémas conservatifs CSRK2 et CSRK3 (de l'ordre de 10⁻¹⁵). Ce résultat confirme que la stratégie de

correction de flux appliquée aux schémas NRK2 et NRK3 fonctionne dans un contexte d'écoulement 3D gouverné par les équations de Navier-Stokes. Comme sur le cas-test de l'advection 2D du tourbillon, l'erreur de conservativité est environ 10000 fois plus importante lorsque l'étape de correction de flux n'est pas utilisée.

Comme pour les cas-test précédents, le speedup théorique des schémas à pas de temps local par rapport à leur version à pas de temps global peut être calculé. Le nombre de cellules dans la première et la dernière zone est : $nbc_{cell_1} = 5505024$; le nombre de cellules dans la seconde zone est : $nbc_{cell_2} = 7340032$. En appliquant la formule (4.34), le speedup théorique $S_{theo} \approx 30\%$ est obtenu pour ce cas-test. Les tableaux 4.9 et 4.10 montrent la CFL maximale et le speedup pour les différents schémas à pas de temps local.

	CFLmax, Δt_{max}	Speedup par rapport à $RK2_{dtglobal}$
CSRK2	0.3, $\Delta t_{max} = 0.0006 \times L_0/U_0$	27%
NRK2	0.3, $\Delta t_{max} = 0.0006 \times L_0/U_0$	28%
NRK2 sans correction de flux	0.3, $\Delta t_{max} = 0.0006 \times L_0/U_0$	28%

TABLE 4.9 – CFL maximale et speedup pour les schéma CSRK2 et NRK2

	CFLmax, Δt_{max}	Speedup par rapport à $RK3_{dtglobal}$
CSRK3	0.9, $\Delta t_{max} = 0.0019 \times L_0/U_0$	26%
NRK3	0.9, $\Delta t_{max} = 0.0019 \times L_0/U_0$	28%
NRK3 sans correction de flux	0.9, $\Delta t_{max} = 0.0019 \times L_0/U_0$	28%

TABLE 4.10 – CFL maximale et speedup pour les schéma CSRK2 et NRK2

Les speedup mesurés sont proches du speedup théorique de 30%, ce qui atteste de la bonne implémentation des schémas dans un contexte 3D. A l'image des cas-test précédents, aucune différence sur le speedup n'a été observée lorsque l'étape de correction de flux est désactivée (schémas NRK2 et NRK3). Enfin, comme pour le cas de l'advection 2D du tourbillon, une importante différence de stabilité est observée entre les schémas à pas de temps local basés sur la méthode RK3 et les schémas à pas de temps local basés sur la méthode RK2. Ceci rend les schémas basés sur la méthode RK3 plus performants. Nous avons mesuré un speedup d'environ 60% pour les schémas CSRK3 et NRK3 par rapport

au schéma $RK2_{dtglobal}$. Le speedup des schémas CSRK2 et NRK2 par rapport au schéma $RK2_{dtglobal}$ n'est que de 28%.

4.7.3 Résultats des simulations

La figure 4.23 montre l'évolution temporelle de l'enstrophie pour les différents schémas à pas de temps local et pour la solution de référence. La solution de référence est obtenue à l'aide du schéma $RK3_{dtglobal}$ de pas de temps $\Delta t = 0.00015 \times L_0/U_0$. Avec ce pas de temps, le nombre CFL dans la seconde zone (le nombre CFL maximal sur le maillage) est $CFL = 0.3$. Les différents schémas à pas de temps sont utilisés avec leur pas de temps maximal, fourni dans la première colonne des tableaux 4.9 et 4.10. Cette figure montre un très bon accord entre les différents schémas à pas de temps local et la solution de référence. Ceci montre donc que nos schémas NRK2 et NRK3 peuvent être utilisés pour réaliser des simulation d'écoulements 3D turbulents.

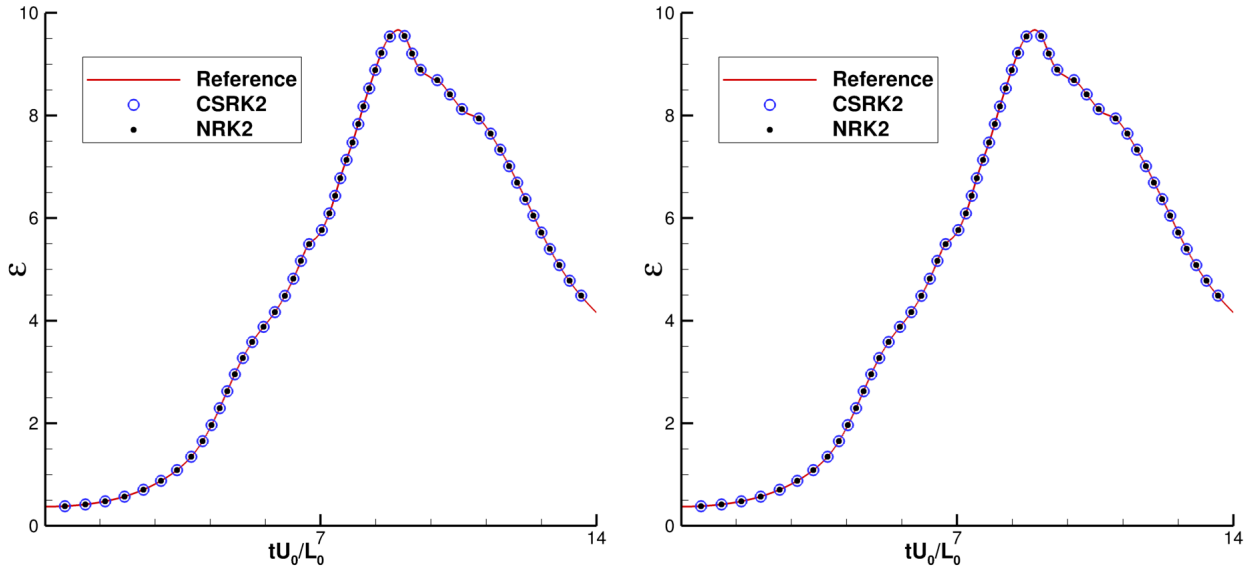


FIGURE 4.23 – Evolution temporelle de l'enstrophie ϵ pour les schémas CSRK2, NRK2 (à gauche) et les schémas CSRK3, NRK3 (à droite).

4.7.4 Vérification de l'ordre temporel

Les résultats de l'étude de l'ordre temporel sont affichés sur la figure 4.20. Pour chaque schéma à pas de temps local, une solution de référence a été calculée avec le pas de temps $\Delta t = \Delta t_{ref} = 7.10^{-5} \times L_0/U_0$. Trois autres solutions sont calculées avec $\Delta t = 2\Delta t_{ref}$, $\Delta t = 4\Delta t_{ref}$ et $\Delta t = 8\Delta t_{ref}$. Comme pour les cas-test précédents, cette étude confirme l'ordre temporel théorique de chaque schéma. En effet, comme le montre la figure 4.24, la courbe d'erreur a une pente égale à deux pour les schémas CSRK2, CSRK3 et NRK2; la courbe d'erreur a une pente égale à trois pour le schéma NRK3

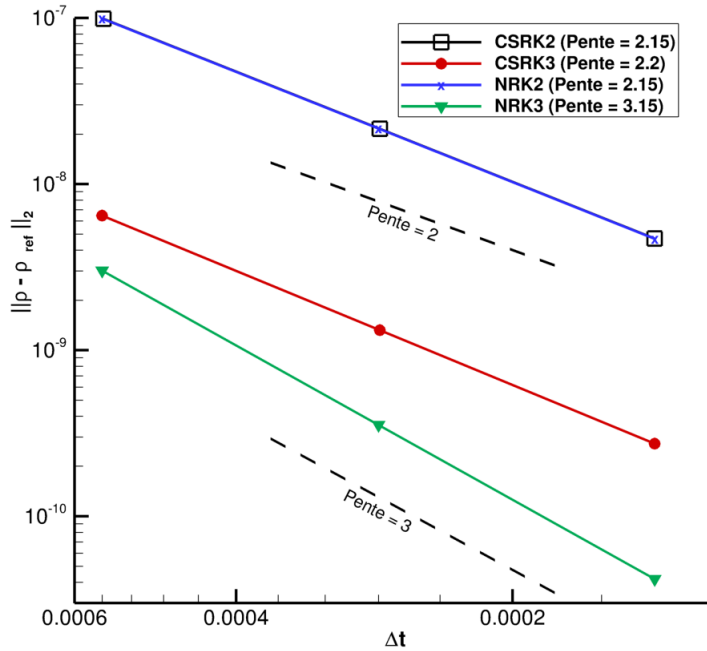


FIGURE 4.24 – Étude de l'ordre temporel des différents schémas à pas de temps local sur le cas-test du tourbillon de Taylor-Green.

4.8 Bilan des cas-test académiques

Pour réaliser des simulations plus complexes (l'objet du prochaine chapitre), nous avons choisi de ne sélectionner qu'un schéma à pas de temps local parmi les quatre testés dans ce chapitre. Réaliser ce type de simulations requiert une implémentation des schémas plus générale que pour les cas-test académiques présentés dans ce chapitre. Par exemple, pour les cas-test académiques, les schémas à pas de temps local ne sont constitués que de deux pas de temps (Δt et $\Delta t/2$) et les variations de pas de temps se font toujours dans la direction \vec{x} . Sur des cas plus complexes, un effort d'implémentation est nécessaire pour gérer un nombre quelconque N de pas de temps (Δt , $\Delta t/2$, $\Delta t/4\dots$) avec des variations de pas de temps dans les trois directions de l'espace. Nous avons décidé de réaliser cet effort d'implémentation pour le "meilleur" des quatre schémas à pas de temps local.

Sur les trois cas-test académiques présentés dans ce chapitre, tous les schémas à pas de temps local ont montré une très bonne correspondance avec les solutions de référence. Par contre, des différences importantes sur la stabilité sont relevées, en particulier lorsque le schéma des flux convectifs AUSM+(P) senseur est utilisé. Dans ce cas, nous observons que les schémas à pas de temps local basés sur la méthode RK3 (CSRK3 et NRK3) peuvent fonctionner avec un pas de temps trois fois plus grand que les schémas à pas de temps local basés sur la méthode RK2 (CSRK2 et NRK2). Ceci rend les schémas CSRK3 et NRK3 plus performants que les schémas CSRK2 et NRK2, malgré leur coût algorithmique supérieur. Nous éliminons donc les schémas CSRK2 et NRK2, jugés trop peu performants. Les

cas-test académiques révèlent que les schémas CSRK3 et NRK3 mènent à des temps de calcul similaires, par contre le schéma NRK3 est plus précis : il est précis à l'ordre trois, tandis que le schéma CSRK2 est seulement précis à l'ordre deux. **Par conséquent, le schéma NRK3 nous paraît être le meilleur schéma à pas de temps local parmi les quatre. Une implémentation générale de ce schéma (pour gérer un nombre quelconque de pas de temps, des variations de pas de temps dans les trois directions de l'espace...) est réalisée pour le rendre utilisable sur des configurations complexes. L'implémentation du schéma NRK3 dans un cas général est décrite au chapitre 3 (partie 3.3).**

Chapitre 5

Démonstration sur des configurations de type industriel

Dans ce chapitre, notre schéma à pas de temps local NRK3 est utilisé pour réaliser deux simulations LES :

- l'écoulement d'air autour d'un profil d'aile (profil SD7003)
- l'écoulement d'air autour d'un cylindre, avec une méthode de frontières immergées (en anglais Immersed Boundary Method, IBM)

Pour chaque simulation, notre schéma NRK3 est soigneusement comparé à des schémas implicite et explicite à pas de temps global en terme de précision et de temps de calcul. Une description des stratégies de parallélisation MPI et OpenMP utilisées pour notre schéma NRK3 est également réalisée.

Dans la première partie de ce chapitre, le cas d'écoulement d'air autour du profil SD7003 est présenté, et les résultats de comparaison entre les différents schémas sont donnés. Dans la seconde partie, les stratégies de parallélisation MPI et OpenMP de notre schéma sont décrites, et des test de scalabilité forte sont réalisés. Enfin, dans la dernière partie de ce chapitre, les résultats concernant le cas d'écoulement autour du cylindre par une approche IBM sont fournis.

5.1 LES d'un écoulement transitionnel autour du profil SD7003

5.1.1 Description du cas

Ce cas-test concerne un écoulement d'air autour du profil d'aile SD7003. Il s'agit d'un cas d'écoulement assez bien documenté dans la littérature [100, 101, 102].

Avec les paramètres retenus pour la simulation, l'écoulement simulé est de type transitionnel : le fluide passe d'un état laminaire à un état turbulent par le biais d'un bulbe de décollement laminaire. Dans cette partie, une description plus précise de ce type d'écoulement est réalisée, puis nous montrons en quoi sa simulation nous paraît pertinente pour

évaluer notre schéma à pas de temps local NRK3.

Tout d'abord, un bref rappel est effectué concernant les profils aérodynamiques et le vocabulaire qui leur est associé. Un profil aérodynamique est une section de l'aile d'un aéronef, dessinée pour permettre des performances (forces de portance et de traînée) optimales. La force de portance est la force exercée par l'air sur l'aile dans la direction verticale ; elle permet de faire décoller l'aéronef. La force de traînée est la force exercée par l'air sur l'aile dans la direction horizontale ; elle freine l'aéronef dans son déplacement. Comme schématisé sur la figure 5.1, la partie supérieure d'un profil aérodynamique est appelée extrados, et la partie inférieure est l'intrados.

L'écoulement rencontre le profil en amont au niveau d'une zone appelée bord d'attaque, et quitte le profil en aval au niveau du bord de fuite. On appelle corde (notée c) le segment de droite (et la distance qui y est associée) reliant le bord d'attaque et le bord de fuite (voir figure 5.1). L'angle d'incidence (noté α), est l'angle entre l'axe bord d'attaque - bord de fuite (la corde) et la direction de l'écoulement à l'infini amont.

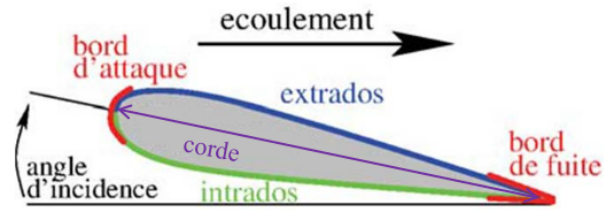


FIGURE 5.1 – Description schématique d'un profil aérodynamique

En fonction des paramètres du problème (angle d'incidence, nombre de Reynolds...), l'écoulement du fluide à l'extrados peut être ralenti par un gradient de pression adverse plus ou moins fort, qui s'oppose à l'écoulement. En présence d'un fort gradient de pression adverse, les forces de pression exercées sur la couche limite (la zone d'interface entre l'écoulement et la paroi) peuvent devenir telles que les vitesses longitudinales deviennent négatives près de la paroi. Ceci crée une zone de recirculation du fluide. Le point de départ de cette zone de recirculation est qualifié de point de décollement de la couche limite. En effet, les lignes de courant de l'écoulement (c'est à dire les trajectoires des particules fluides) se séparent littéralement de la paroi à partir de ce point. Un recollement de la couche limite plus loin en aval peut se produire.

Dans le cas où la couche limite décolle dès le bord d'attaque et parvient à recoller rapidement, la région de recirculation a une longueur bien inférieure à la corde c du profil. On parle alors de bulbe de décollement pour qualifier cette région de recirculation, qui s'étend du point de décollement au point de recollement. Lorsque l'écoulement est initialement laminaire (au point de décollement), on parle de bulbe décollement laminaire (BDL). La figure 5.2 présente le schéma d'un BDL moyenné en temps. Un BDL est souvent le siège d'une transition d'un écoulement laminaire vers un écoulement turbulent. Il s'agit d'un phénomène instationnaire, qui est sujet à un lacher tourbillonnaire, comme cela est montré sur la figure 5.3. Le lecteur intéressé peut se reporter à la thèse de Nicolas Alferez [4] pour davantage de précisions sur le BDL.

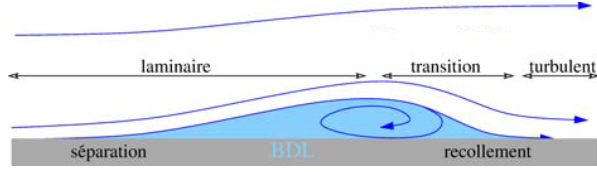


FIGURE 5.2 – Lignes de courant issues d'un champ de vitesse moyen d'un BDL (figure tirée de [89])

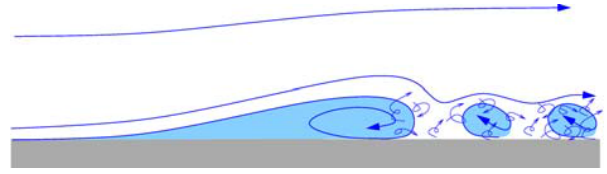


FIGURE 5.3 – Représentation instantanée d'un BDL (figure tirée de [89])

A cause du processus de transition laminaire/turbulent, l'écoulement est fortement instationnaire. Ainsi, l'utilisation d'une méthode d'intégration temporelle explicite à pas de temps local semble être une approche très intéressante pour réaliser une simulation précise et à moindre coût. De plus, Marxen *et al* [103] et Alferez [4] ont montré que l'écoulement était sensible à l'introduction de perturbations physiques de faible amplitude (environ 10^4 ou 10^5 plus faible que la vitesse de l'écoulement loin du profil). Ce cas s'avère donc particulièrement discriminant et permettra donc d'évaluer la précision de notre schéma à pas de temps local NRK3 dans un contexte LES.

5.1.2 Paramètres de la simulation et maillage

Le nombre de Mach de l'écoulement basé sur la vitesse à l'infini U_∞ (vitesse de l'écoulement loin du profil) vaut : $M = 0.1$. Le nombre de Reynolds Re basé sur la corde c et U_∞ est fixé à $Re = 60000$. Les valeurs de la vitesse U_∞ et de la corde c sont : $U_\infty = 34 \text{ m.s}^{-1}$ et $c = 0.027 \text{ m}$. L'angle d'attaque α est égal à $\alpha = 8^\circ$. Comme montré dans l'étude de Galbraith et Visbal [100], ces valeurs permettent de se placer dans le cas d'un écoulement instationnaire décrit dans la partie 5.1.1 : l'air transitionne d'un état laminaire vers un état turbulent par l'intermédiaire d'un bulbe de décollement laminaire (BDL).

Un maillage 3D structuré constitué d'environ 90 millions de cellules est utilisé. Le pas d'espace dans la direction \vec{z} est uniforme : $\Delta z = 0.002 c$. Cent cellules sont utilisées dans la direction \vec{z} , la profondeur du maillage est donc égale à $0.2 c$ (20% de la corde). En mécanique des fluides, il est commun de se baser sur les pas d'espace adimensionnés à la paroi pour évaluer la résolution d'un maillage. Le pas d'espace adimensionné Δz^+ vaut :

$$\Delta z^+ = \frac{u_\tau \Delta z}{\nu} \approx 8, \quad (5.1)$$

où ν est la viscosité cinématique de l'air et u_τ est la vitesse de friction définie par :

$$u_\tau = \sqrt{\frac{\tau}{\rho}} \quad \text{avec} \quad \tau = \mu \left(\frac{\partial u_2}{\partial y} \right)_{y=0}. \quad (5.2)$$

Dans l'équation (5.2), μ représente la viscosité dynamique de l'air. Les pas d'espace Δx^+ et Δy^+ au niveau de la paroi valent :

$$\Delta x^+ = \frac{u_\tau \Delta x}{\nu} \approx 25 \quad \Delta y^+ = \frac{u_\tau \Delta y}{\nu} \approx 1. \quad (5.3)$$

Cette résolution à la paroi est plus fine que celle utilisée dans [88] et équivalente à celle utilisée dans [91, 104]. Dans [91], une étude de sensibilité au maillage a montré que cette résolution est suffisante pour obtenir des résultats très satisfaisants. Nous estimons donc que la résolution du maillage que nous avons choisie convient à l'écoulement simulé. A partir de cette résolution à la paroi, les pas d'espace Δx et Δy sont progressivement augmentés d'un facteur qui n'excède pas 1.05.

Les conditions aux limites sont de type périodique dans la direction \vec{z} . Une coupe 2D de ce maillage est représentée sur la figure 5.4 avec un nombre réduit de point pour plus de clarté. Les conditions aux limites au niveau des bords du domaine de calcul (bords surlignés en rouge sur la figure 5.4) sont des conditions de type non-réfléchissantes [105]. La figure 5.5 est un zoom au niveau du profil SD7003 (au centre du maillage). On précise que les bords du domaine sont situés à une distance égale à $100c$ du profil. Les conditions aux limites au niveau des bords correspondants aux parois du profil (bords surlignés en bleu sur la figure 5.5) sont de type paroi adiabatique.

A l'image de ce qui est fait dans [103, 4], une perturbation de faible amplitude est imposée sur la vitesse à la paroi, juste avant le décollement (à $x/c = 0.1$). Cette perturbation s'écrit :

$$u_{pertu} = A_u \times U_\infty \times \cos\left(\frac{2\pi z}{\lambda_z}\right) \times \sin(2\pi ft),$$

où $A_u = 10^{-4}$ est l'amplitude de la perturbation, $\lambda_z = 2.7mm$ ($0.02c$) est la longueur d'onde et $f = 50000 Hz$ ($40 \times U_\infty/c$). Le but de cette perturbation est de déclencher de manière déterministe le processus de transition laminaire/turbulente. En effet, en l'absence de ce forçage, le processus de transition laminaire/turbulent est piloté par le bruit numérique provoqué par les erreurs de troncature de la méthode numérique [4]. L'objectif est donc d'homogénéiser le point de transition pour les différents schémas d'intégration temporels qui seront utilisés sur ce cas, dans le but d'assurer une comparaison correcte des différents schémas.

Comme souvent en simulation instationnaire, on ne prend pas un champ uniforme comme solution initiale. C'est l'écoulement en régime établi qui nous sert de solution initiale. Il est obtenu en réalisant deux simulations :

- Une première simulation RANS est effectuée pour obtenir le champ moyen en régime établi,
- A partir de cette solution moyennée, le schéma $RK3_{dtglobal}$ est utilisé pour atteindre le temps $t = 6 \times c/U_\infty$.

On s'assure ainsi que l'on a dépassé le régime transitoire et atteint le régime établi.

Enfin, à l'image des calculs réalisés dans [106, 100, 101, 4] aucun modèle sous-maille n'est utilisé pour cette simulation LES. Lorsque le maillage est localement trop lâche pour bien représenter les échelles dissipatives, le schéma quasi-centré AUSM+(P) senseur présenté dans la section 4.3.1 du chapitre précédent prend localement le relais pour assurer la régularisation des plus petites échelle.

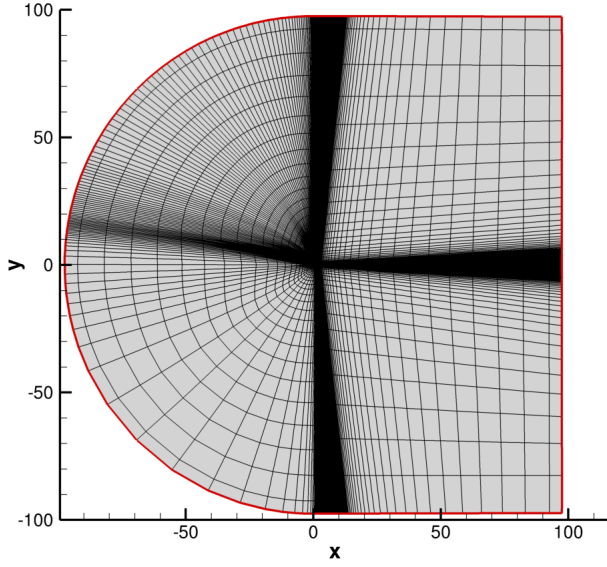


FIGURE 5.4 – Coupe 2D du maillage avec un nombre réduit de cellules.

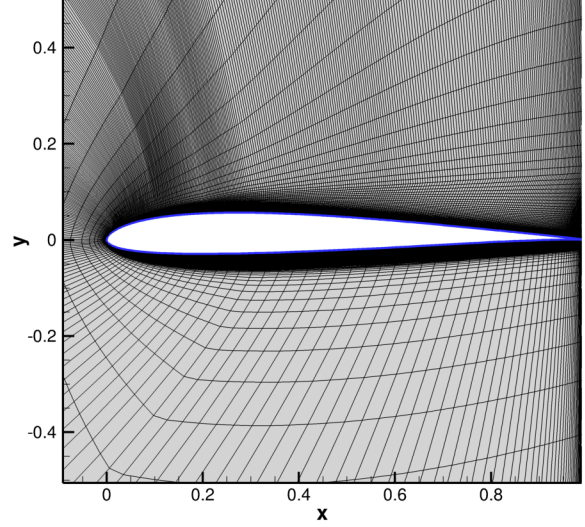


FIGURE 5.5 – Zoom au niveau du profil, avec un nombre réduit de cellules.

5.1.3 Découpage du maillage pour le schéma NRK3

Le schéma à pas de temps local NRK3 nécessite un découpage du maillage en zones de pas de temps différents. Plusieurs étapes sont nécessaires pour réaliser ce découpage.

Etape 1

Sur chaque cellule i , le pas de temps maximal admissible Δt_i est calculé à partir de la condition CFL locale. On précise que le calcul de la condition CFL locale sur chaque cellule se base sur les données de la solution initiale. Vu que la solution initiale est bi-dimensionnelle, il n'y a pas de variation de la CFL locale dans la direction \vec{z} .

Dans le cas des équations de Navier-Stokes, le pas de temps Δt_i sur chaque cellule correspond à un minimum entre un pas de temps convectif Δt_i^c et un pas de temps visqueux Δt_i^v calculés sur chaque cellule :

$$\Delta t_i = \min(\Delta t_i^c, \Delta t_i^v). \quad (5.4)$$

Le pas de temps Δt_i^c est limité par la contrainte CFL convective suivante :

$$\Delta t_i^c \leq \frac{\min(\Delta x_i, \Delta y_i, \Delta z_i)}{|\vec{V}_i| + V_{son,i}}, \quad (5.5)$$

où Δx_i , Δy_i et Δz_i sont les pas d'espace de la cellule i , $|\vec{V}_i|$ et $V_{son,i}$ sont respectivement la norme du vecteur vitesse et la vitesse du son sur cette même cellule. Exceptionnellement sur ce cas, la vitesse du son n'est pas notée c pour ne pas introduire de confusion avec la corde. En pratique, on choisit :

$$\Delta t_i^c = 0.95 \times \frac{\min(\Delta x_i, \Delta y_i, \Delta z_i)}{|\vec{V}_i| + V_{son,i}}. \quad (5.6)$$

Le pas de temps Δt_i^v est limité par la contrainte CFL visqueuse suivante :

$$\Delta t_i^v \leq \rho_i \frac{\min(\Delta x_i^2, \Delta y_i^2, \Delta z_i^2)}{2\gamma(\frac{\mu_i}{Pr} + \frac{\mu_{ti}}{Pr_t})}, \quad (5.7)$$

où μ_t est la viscosité turbulente et Pr le nombre de Prandtl (Pr_t le nombre de Prandtl turbulent). En pratique, on choisit :

$$\Delta t_i^v = 0.95 \times \rho_i \frac{\min(\Delta x_i^2, \Delta y_i^2, \Delta z_i^2)}{2\gamma(\frac{\mu_i}{Pr} + \frac{\mu_{ti}}{Pr_t})}. \quad (5.8)$$

Les pas de temps maximal et minimal sur le domaine de calcul, Δt_{max} et Δt_{min} sont déterminés :

$$\Delta t_{max} = \max_i \Delta t_i, \quad (5.9)$$

$$\Delta t_{min} = \min_i \Delta t_i. \quad (5.10)$$

Le pas de temps Δt_{min} correspond au pas de temps requis pour la cellule présentant la condition CFL la plus restrictive. Ce pas de temps serait appliqué sur l'ensemble du domaine de calcul dans le cas d'un schéma explicite classique, à pas de temps global. Nous avons obtenu les valeurs numériques suivantes pour les pas de temps Δt_{min} et Δt_{max} :

$$\Delta t_{min} = 1.25 \times 10^{-6} \times c/U_\infty, \quad (5.11)$$

$$\Delta t_{max} = 1.8 \times 10^{-5} \times c/U_\infty. \quad (5.12)$$

On remarque que le rapport entre Δt_{max} et Δt_{min} est environ égal à 14. Ce chiffre peut paraître faible, en sachant qu'il y a par exemple un rapport 1000 entre le pas d'espace Δy "loin" du profil et le pas d'espace Δy dans la couche limite. Ce rapport entre Δt_{max} et Δt_{min} est en fait limité par la valeur uniforme de pas d'espace Δz (qui est environ 14 fois plus grand que les pas d'espace minimal dans la direction \vec{y}).

Pour le schéma à pas de temps local NRK3, on cherche à déterminer le pas de temps maximal Δt que l'on va utiliser sur le maillage. On pose z tel que :

$$2^z = \frac{\Delta t_{max}}{\Delta t_{min}}. \quad (5.13)$$

On a donc :

$$z = \log \left(\frac{\Delta t_{max}}{\Delta t_{min}} \right) / \log(2) \quad (5.14)$$

$$\approx 3.7.$$

On prend la partie entière de z ; le pas de temps maximal sur le maillage (Δt) vaut donc :

$$\Delta t = 2^3 \Delta t_{min} = 8 \Delta t_{min} = 1 \times 10^{-5} \times c/U_\infty$$

Sur ce cas, quatre pas de temps différents sont donc utilisés sur le domaine de calcul : Δt , $\Delta t/2$, $\Delta t/4$ et $\Delta t/8 = \Delta t_{min}$. Cette première étape a donc permis de déterminer les différents pas de temps à utiliser sur le maillage. Il faut maintenant découper le maillage en différentes zones et répartir ces quatre pas de temps sur les zones : c'est l'objet des étapes suivantes.

Etape 2

Nous avons choisi de diviser le maillage en zones structurées de 20×20 cellules. Dans chaque zone, on détermine le pas de temps maximal permettant d'assurer la stabilité de toutes les cellule de la zone. Comme le montre la figure 5.6, les pas de temps associés aux zones peuvent prendre des valeurs bien différentes de Δt , $\Delta t/2$, $\Delta t/4$ ou $\Delta t/8$. Sur la figure 5.6, les zones sont numérotées de 1 à 8, afin de les repérer dans le texte. On précise que ce découpage est effectué dans le plan (\vec{x}, \vec{y}) uniquement puisque les variations de la *CFL* (et donc du pas de temps) n'ont lieu que dans ce plan. Le découpage obtenu en 2D est ensuite étendu dans la direction \vec{z} .

Etape 3

On vient ensuite corriger le pas de temps associé à chaque zone avec l'un des quatre pas de temps de notre schéma (Δt , $\Delta t/2$, $\Delta t/4$, $\Delta t/8$). Cette correction suit les règles suivantes, dans lesquelles Δt_z représente le pas de temps associé à la zone :

Si $\Delta t \leq \Delta t_z$, alors Δt_z prend la valeur Δt

$\forall k \in [1, 3]$, si $\Delta t/2^k \leq \Delta t_z \leq \Delta t/2^{k-1}$ alors Δt_z prend la valeur $\Delta t/2^k$

La figure 5.7 montre les pas de temps corrigés associés aux différentes zones.

Etape 4

Il peut arriver que le rapport entre les pas de temps de deux zones adjacentes soit supérieur à deux. Comme mentionné dans la partie 3.2.7, une telle situation n'est pas permise car notre schéma est conçu pour gérer des rapports entre pas de temps égaux à deux au plus. Cette situation se produit sur la figure 5.7 : la zone 5 a un pas de temps quatre fois plus grand que la zone 6, qui lui est adjacente. Dans ce cas, on crée au sein de

la zone 5 (la zone de pas de temps Δt) une zone de pas de temps $\Delta t/2$ qui vient jouxter la zone 6 (de pas de temps $\Delta t/4$). Ceci est illustré par la figure 5.8. Ainsi, sur l'ensemble du maillage, on garantit un rapport deux (au plus) entre les pas de temps des zones adjacentes. Précisons que le pas de temps $\Delta t/2$ est stable pour la nouvelle zone puisque les cellules constituant cette nouvelle zone étaient initialement intégrées avec le pas de temps Δt .

$\Delta t/0.8$	$\Delta t/3.1$	$\Delta t/5$	$\Delta t/8$
5	6	7	8
$\Delta t/0.9$	$\Delta t/1.6$	$\Delta t/3.9$	$\Delta t/6.7$
1	2	3	4

FIGURE 5.6 – **Etape 2** : Maillage découpé en zones de 20×20 cellules, avec le pas de temps associé aux différentes zones.

Δt	$\Delta t/4$	$\Delta t/8$	$\Delta t/8$
5	6	7	8
Δt	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$
1	2	3	4

FIGURE 5.7 – **Etape 3** : Correction des pas de temps associés aux différentes zones.

Δt	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	$\Delta t/8$
5	9	6	7	8
Δt		$\Delta t/2$	$\Delta t/4$	$\Delta t/8$
1	2	3	4	

FIGURE 5.8 – **Etape 4** : Création, entre les zones 5 et 6 de la zone 9 de pas de temps $\Delta t/2$. Le but est de garantir un rapport deux (au maximum) entre les pas de temps des zones adjacentes.

Etape 5

Enfin, une étape d'optimisation est réalisée. On fusionne les zones adjacentes ayant le même pas de temps pour optimiser le nombre de cellules fictives (et le transfert d'information associé à ces cellules fictives). Néanmoins, une contrainte doit être respectée : la zone résultant de la fusion de zones structurées doit être une zone structurée. Ainsi, sur l'exemple de la figure 5.8, on ne peut pas fusionner les zones 4, 8 et 7 car on obtiendrait une zone non structurée. On ne peut fusionner que les zones 7 et 8 ou les zones 8 et 4.

Δt	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	
5	9	6	7	
Δt		$\Delta t/2$	$\Delta t/4$	$\Delta t/8$
1	2	3	4	

FIGURE 5.9 – **Etape 5** : Fusion des zones 7 et 8 de la figure 5.8 pour optimiser le nombre de cellules fictives.

Le résultat de notre algorithme de découpe du maillage est visible sur la figure 5.10, qui

montre la répartition des différents pas de temps autour du profil SD7003. Logiquement, on retrouve les plus petits pas de temps à proximité de la paroi du profil, au niveau de la couche limite.

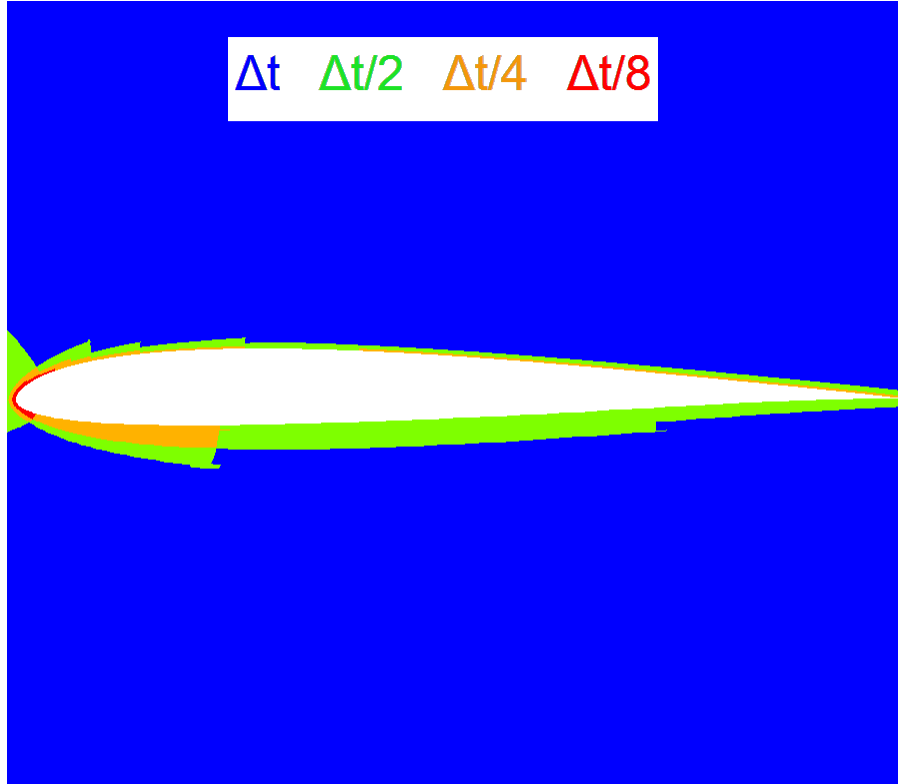


FIGURE 5.10 – Répartition des différents pas de temps autour du profil SD7003.

Le tableau 5.1 montre le pourcentage de cellules de chaque pas de temps.

Pas de temps	Δt	$\frac{\Delta t}{2}$	$\frac{\Delta t}{4}$	$\frac{\Delta t}{8}$
Nbre. relatif de cellules (%)	55	18	20	7

TABLE 5.1 – Nombre relatif de cellules pour chaque pas de temps.

En raisonnant sur le coût algorithmique entre t^n et $t^n + \Delta t$, on montre facilement que le speedup théorique S_{theo} du schéma à pas de temps local NRK3 par rapport au schéma RK3 à pas de temps global ($RK3_{dtglobal}$) est donné par :

$$S_{theo} = \frac{8 - (\pi_{\Delta t} + \pi_{\Delta t/2} \times 2 + \pi_{\Delta t/4} \times 4 + \pi_{\Delta t/8} \times 8)}{8}, \quad (5.15)$$

où $\pi_{\Delta t}$, $\pi_{\Delta t/2}$, $\pi_{\Delta t/4}$ et $\pi_{\Delta t/8}$ désignent le pourcentage de cellules de pas de temps Δt , $\Delta t/2$, $\Delta t/4$ et $\Delta t/8$, dont les valeurs numériques sont données dans le tableau 5.1. **Un speedup théorique d'environ 70% est attendu sur ce cas.**

5.1.4 Schémas d'intégration temporelle utilisés sur ce cas

Sur ce cas, notre schéma à pas de temps local NRK3 est comparé à deux autres schémas d'intégration temporelle (un schéma explicite à pas de temps global et un schéma implicite).

- Le schéma implicite utilisé pour cette comparaison est celui développé par F. Daude [55, 61]. Il est basé sur la méthode de Gear décrite dans la partie 1.4.3 pour discrétiser la dérivée temporelle. Le système non-linéaire est résolu à l'aide de l'algorithme de Newton adaptatif présenté dans la partie 1.4.7 et décrit plus précisément dans [55, 61]. Dans un premier temps, on fixe le critère de convergence de l'algorithme de Newton à $\epsilon = 0.1$ (voir eq. (1.95) pour la définition de ce critère), puis dans un second temps on fixe $\epsilon = 0.06$, pour accroître la précision. Les deux solutions ainsi obtenues sont notées "Implicite, $\epsilon = 0.1$ " et "Implicite, $\epsilon = 0.06$ " sur les figures présentées ci-après. Dans les deux simulations, le pas de temps vaut : $\Delta t = 6 \times 10^{-5} \times c/U_\infty$ (le nombre CFL maximal sur le maillage est environ égal à 30).
- Pour le schéma NRK3, on rappelle que la valeur du plus grand pas de temps sur le domaine de calcul est $\Delta t = 1 \times 10^{-5} \times c/U_\infty$. Précisons que l'étape de correction de flux a été activée sur ce cas.
- Pour le schéma $RK3_{dtglobal}$, le pas de temps vaut : $\Delta t = 1,25 \times 10^{-6} \times c/U_\infty$.

Précisons que le schéma $RK3_{dtglobal}$ et le schéma implicite, ont respectivement, été utilisés avec succès dans [107, 36, 37] et [91, 104, 61, 106] pour réaliser des simulations instationnaires d'écoulements turbulents. Ils constituent donc des points de comparaison pertinents pour notre schéma à pas de temps local NRK3.

Le schéma des flux convectif est le schéma AUSM+(P) senseur ; le schéma des flux visqueux est le schéma centré d'ordre deux décrit dans la partie 4.4 du chapitre précédent.

Les simulations sont réalisées jusqu'à l'instant final $t^f = 2 \times c/U_\infty$. La turbulence étant un phénomène aléatoire, il est judicieux d'effectuer, au cours de la simulation, des moyennes temporelles des grandeurs aérodynamiques en chaque cellule du maillage. La valeur moyennée en temps d'une grandeur aérodynamique f en un point x est notée $\langle f \rangle$. On s'intéressera en particulier aux valeurs moyennes des composantes de la vitesse : $\langle u_1 \rangle$, $\langle u_2 \rangle \dots$ On s'intéressera également aux tensions de Reynolds moyennes $\langle \rho u'_i u'_j \rangle$. Les tensions de Reynolds peuvent être vues comme des contraintes supplémentaires s'exerçant sur les particules fluides, causées par le caractère turbulent de l'écoulement. Le lecteur intéressé peut se référer à la thèse de C. Laurent [89] pour davantage de précisions sur les fondements théoriques de l'analyse statistique de la turbulence.

5.1.5 Présentation des résultats

La figure 5.11 montre une isosurface du critère Q , colorée par la norme de la vitesse. Cette figure a été obtenue au temps final t^f avec le schéma à pas de temps local NRK3. Sur cette figure, on distingue les trois phases de l'écoulement décrites par la littérature. Au niveau du bord d'attaque, l'écoulement est laminaire. Ensuite, des structures turbulentes commencent à se former au niveau de la zone de transition. Enfin, l'écoulement devient pleinement turbulent.

La figure 5.12 montre le champ moyen de la vitesse dans la direction \vec{x} (notée u_1) dans le plan (\vec{x}, \vec{y}) . Cette figure a été obtenue avec le schéma à pas de temps local NRK3 à t^f . L'image de gauche fait clairement apparaître le BDL (siège de la transition laminaire turbulent). Celui-ci se caractérise par une recirculation du fluide : la couleur bleue au sein du BDL traduit des valeurs négatives pour la vitesse u_1 . On montre également la zone de recollement. Les points 1 et 2 correspondent aux points où sont enregistrés les signaux de pression pour réaliser les spectres. Sur l'image de droite est affichée la zone mi-corde (où l'écoulement est pleinement turbulent) et le point d'enregistrement 3 du signal de pression.

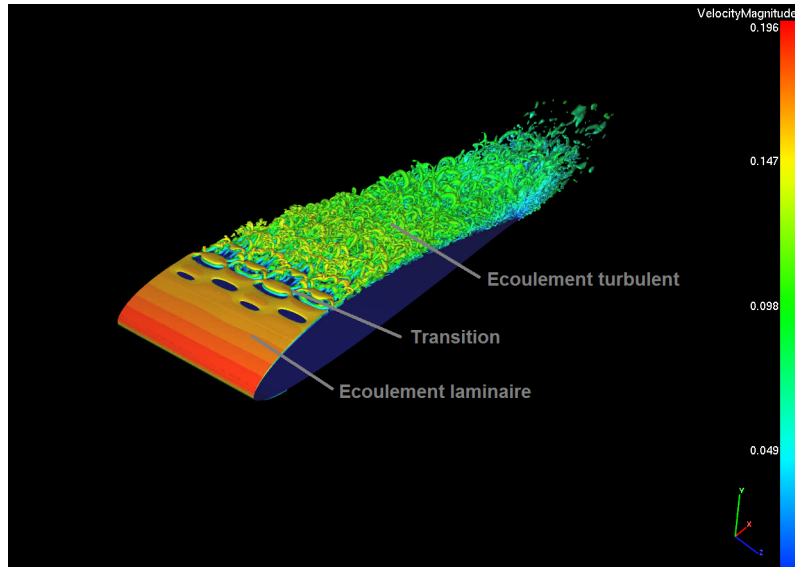


FIGURE 5.11 – Isosurface du critère $Q = 1.5(U_\infty/c)^2$ coloré par la norme de la vitesse.

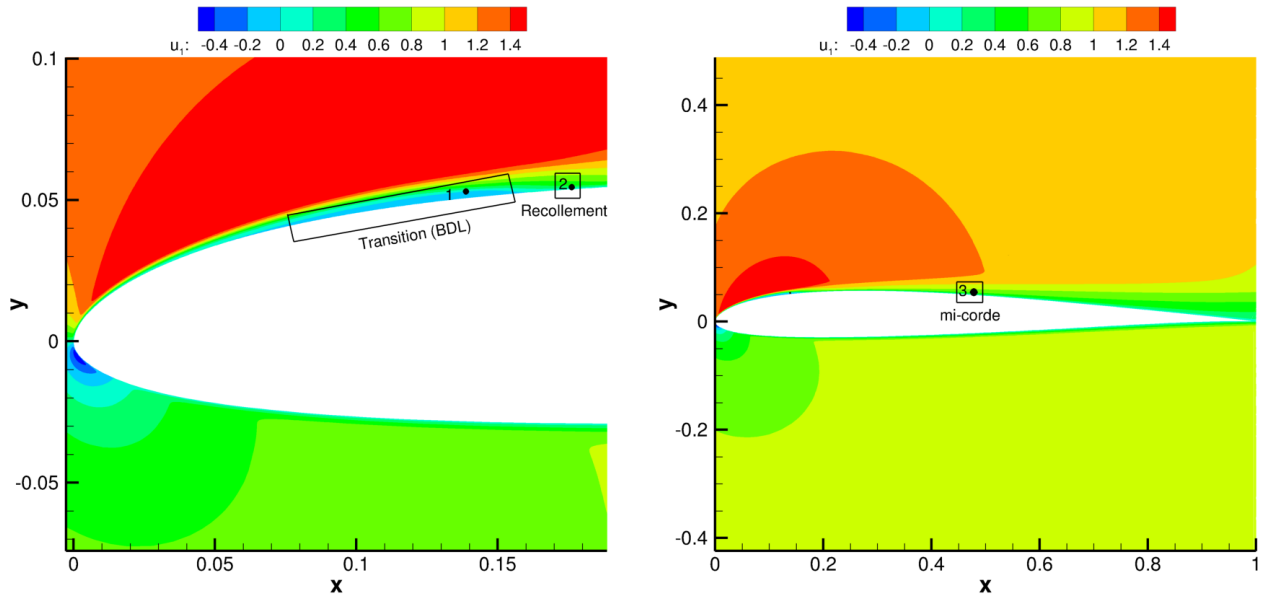


FIGURE 5.12 – Champ moyen de la vitesse dans la direction \vec{x} (u_1). A gauche : les zones de transition et de recollement. A droite : la zone mi-corde.

Vitesse moyenne dans les zones de transition, recollement et mi-corde

La figure 5.13 présentent le profil de la vitesse moyenne adimensionnée $\langle u_1 \rangle / u_e$ pour les différents schémas dans les zones de transition, recollement et mi-corde. On précise que u_e désigne la vitesse dans la direction \vec{x} au bord de la couche limite.

La figure 5.14 montre la même comparaison pour le profil de la vitesse moyenne adimensionnée $\langle u_2 \rangle / \langle v_\infty \rangle$, où u_2 désigne la vitesse dans la direction \vec{y} et v_∞ désigne la vitesse à l'infini dans la direction \vec{y} . Sa valeur numérique est : $v_\infty = 4,76 \text{ m}\cdot\text{s}^{-1}$.

Les figures 5.13 et 5.14 montrent que les différents schémas donnent des profils de vitesse moyenne très similaires. Ceci montre le bon comportement de notre schéma à pas de temps local NRK3 sur ce cas.

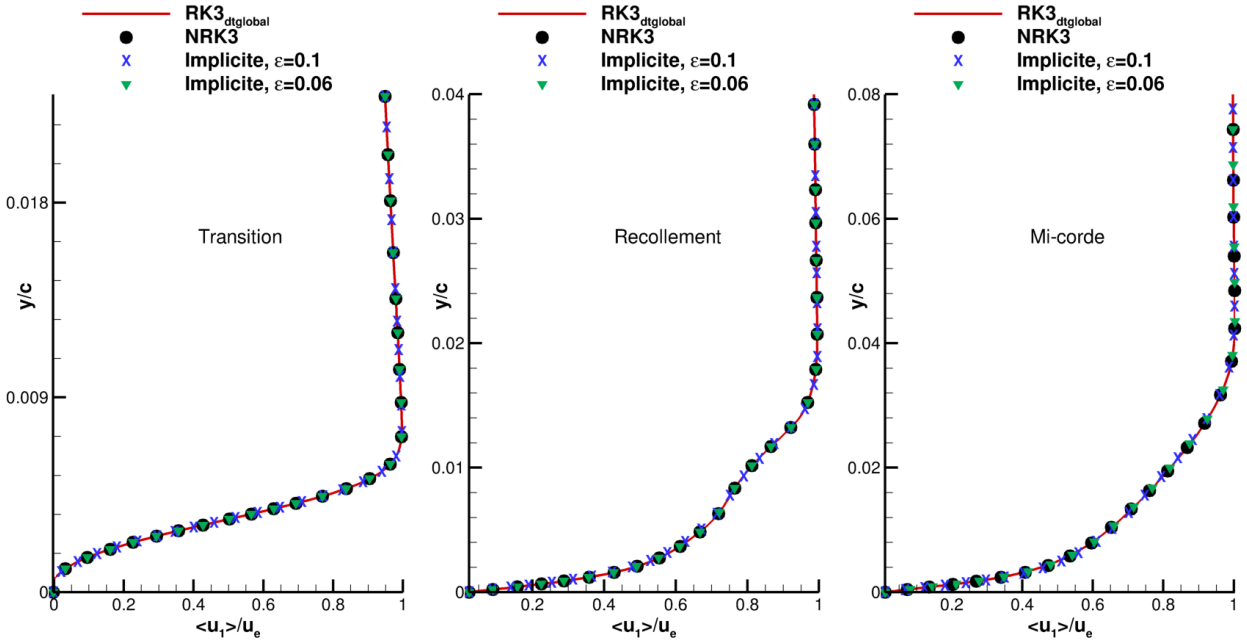


FIGURE 5.13 – Profils de la vitesse moyenne dans la direction \vec{x} dans les zones de transition, recollement et mi-corde.

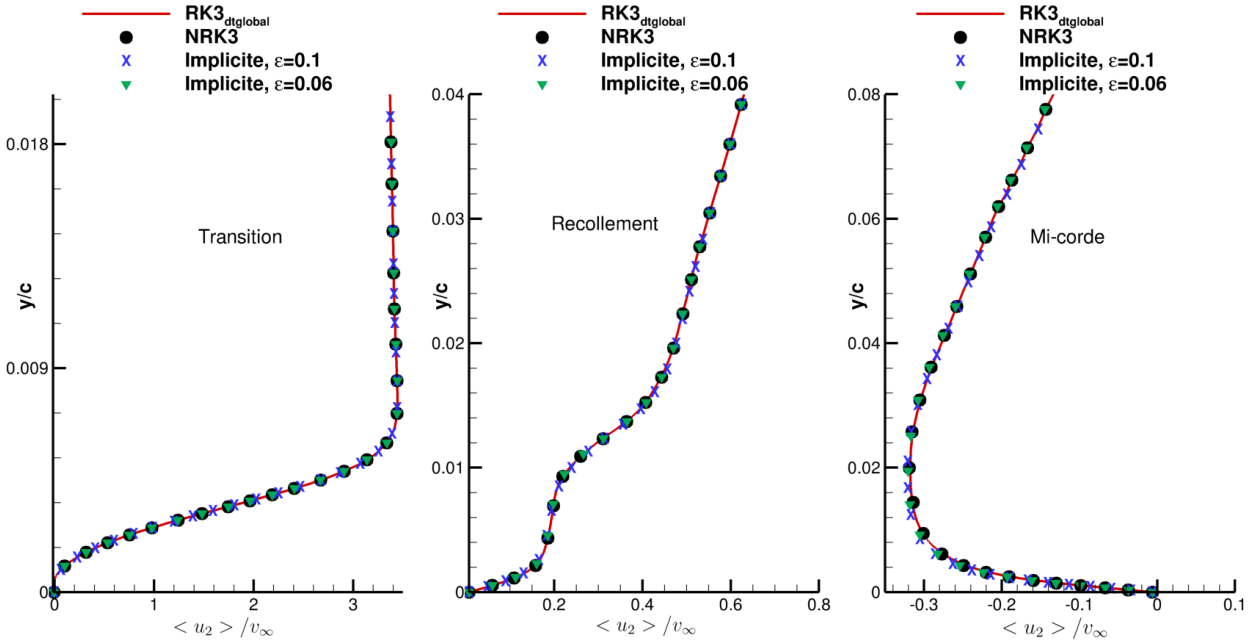


FIGURE 5.14 – Profils de la vitesse moyenne dans la direction \vec{y} dans les zones de transition, recollement et mi-corde.

Tensions de Reynolds moyennes dans les zones de transition, recollement et mi-corde

Les figures 5.15 et 5.16 montrent les profils des tension de Reynolds moyennes $\langle \rho u'_1 u'_1 \rangle$ et $\langle \rho u'_1 u'_2 \rangle$ adimensionnées par $\rho_{ref} u_e^2$. Ces figures montrent un très bon

accord entre les solutions obtenues avec notre schéma NRK3, le schéma $RK3_{dtglobal}$ et le schéma implicite avec $\epsilon = 0.06$. Le schéma implicite utilisé avec le critère de convergence $\epsilon = 0.1$, moins précis, donne une solution qui diffère légèrement des trois autres.

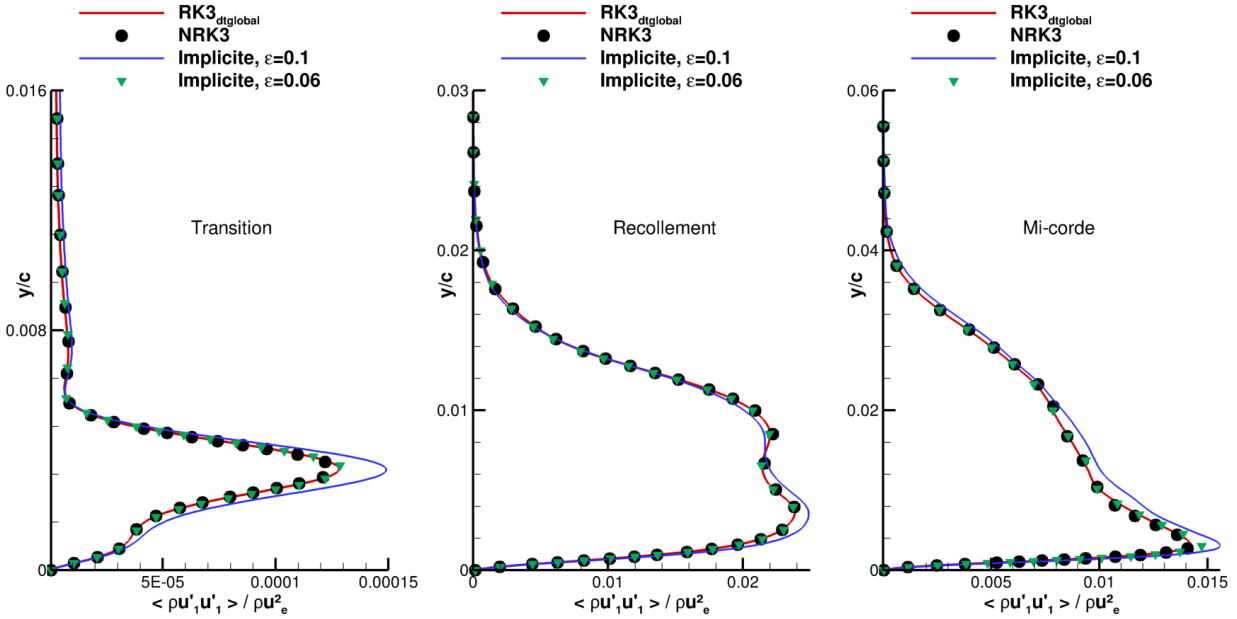


FIGURE 5.15 – Profils des tensions de Reynolds moyennes $\langle \rho u'_1 u'_1 \rangle$.

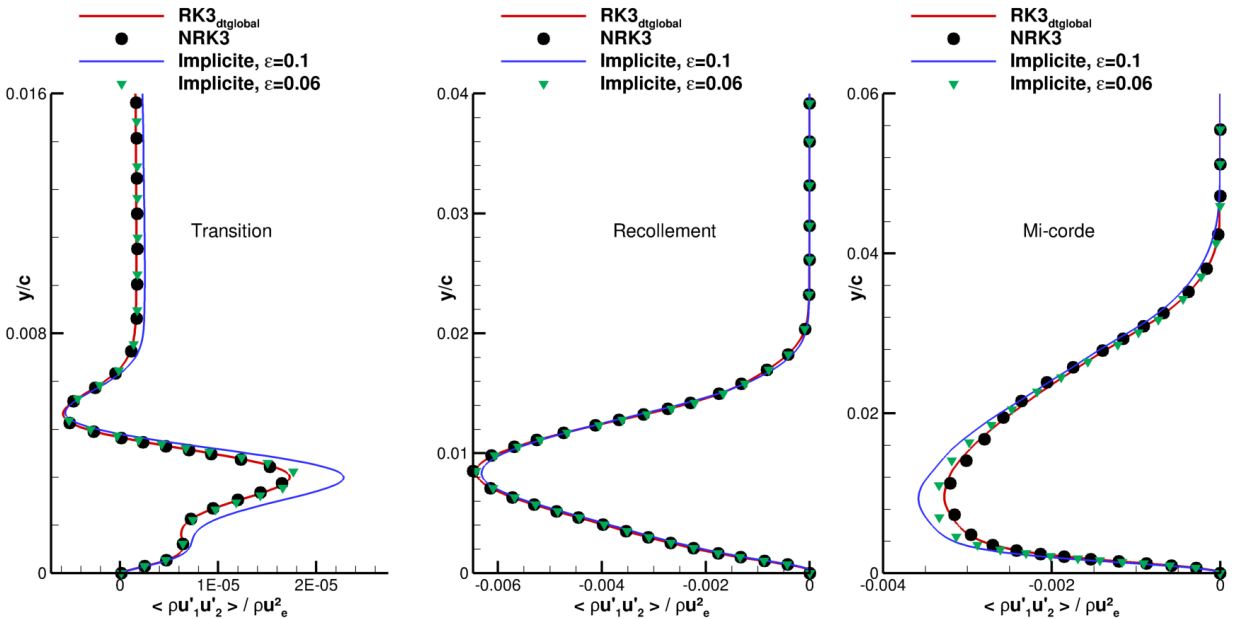


FIGURE 5.16 – Profils des tensions de Reynolds moyennes $\langle \rho u'_1 u'_2 \rangle$.

Spectres des signaux de pression aux points 1 (zone de transition), 2 (zone de recollement) et 3 (zone mi-corde, écoulement pleinement turbulent)

La figure 5.17 montre les spectres du signal de pression obtenus au point 1 avec les différents schémas. Ces spectres sont obtenus par transformée de Fourier du signal temporel de pression. Nous avons volontairement réparti les spectres sur deux images (NRK3 à gauche et le schéma implicite à droite) afin de gagner en lisibilité. La grandeur représentée en ordonnée est la densité spectrale de puissance (Power Spectral Density ou PSD en anglais). Elle est proportionnelle au carré du module de la Transformée de Fourier. La PSD est tracée en fonction du nombre de Strouhal $St = f \times c/U_\infty$, qui correspond à une fréquence adimensionnée. Les figures 5.18 et 5.19 montrent les spectres de pression obtenus avec les différents schémas aux points 2 et 3.

Les figures 5.17, 5.18 et 5.19 montrent un très bon accord entre les différents spectres obtenus.

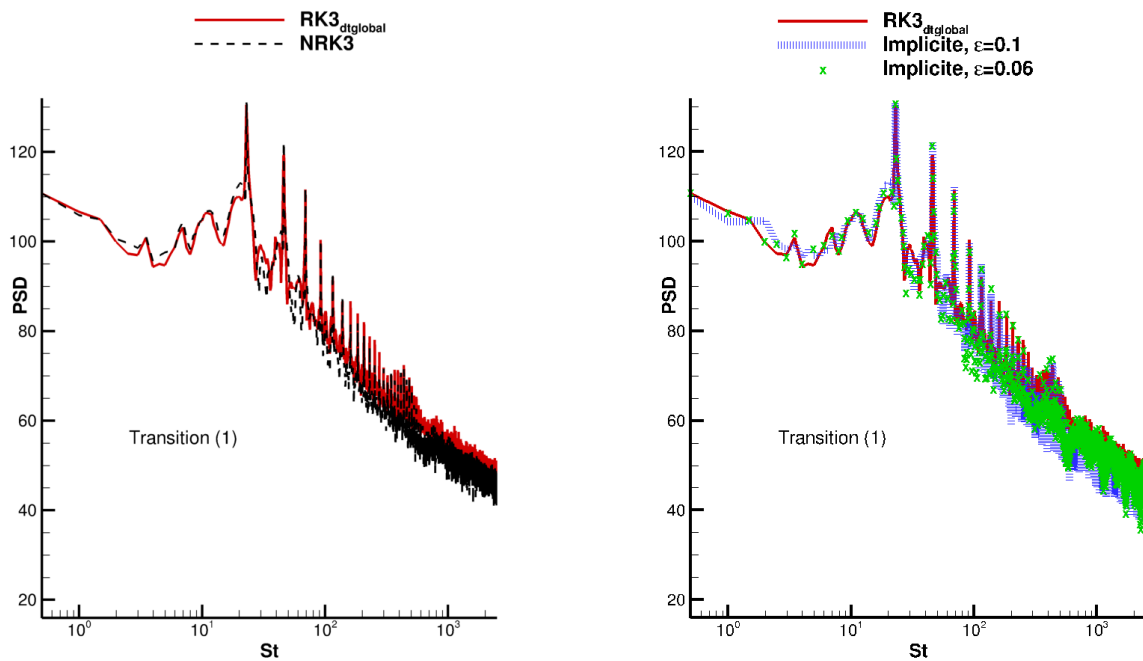


FIGURE 5.17 – Spectre du signal de pression au point d'enregistrement 1, pour les différents schémas.

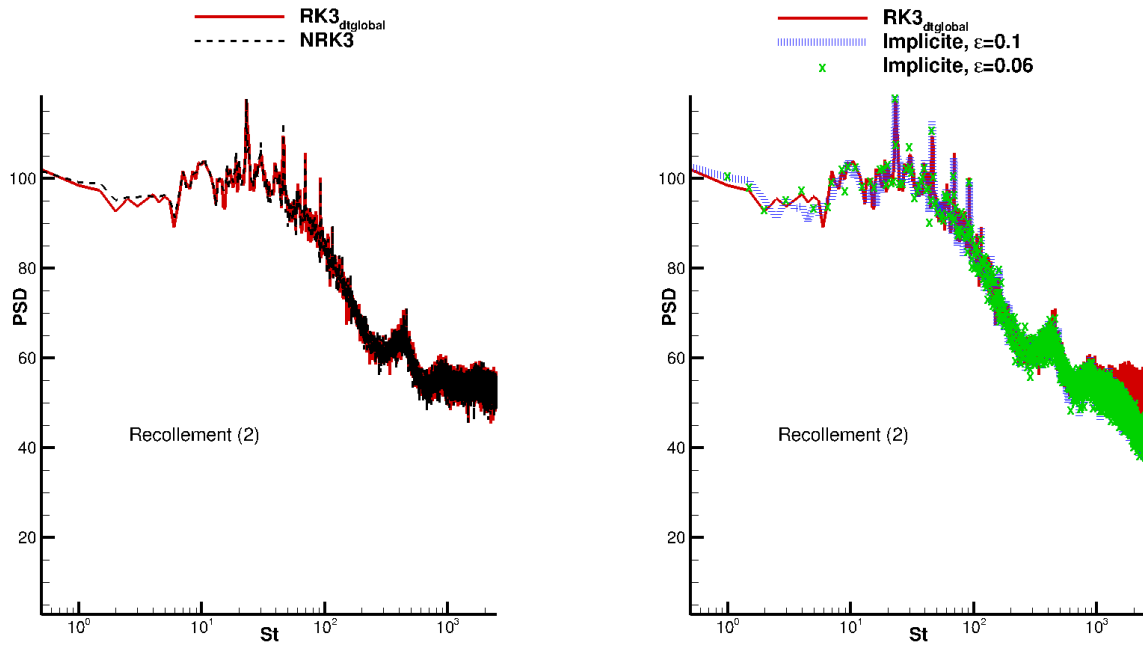


FIGURE 5.18 – Spectre du signal de pression au point d’enregistrement 2, pour les différents schémas.

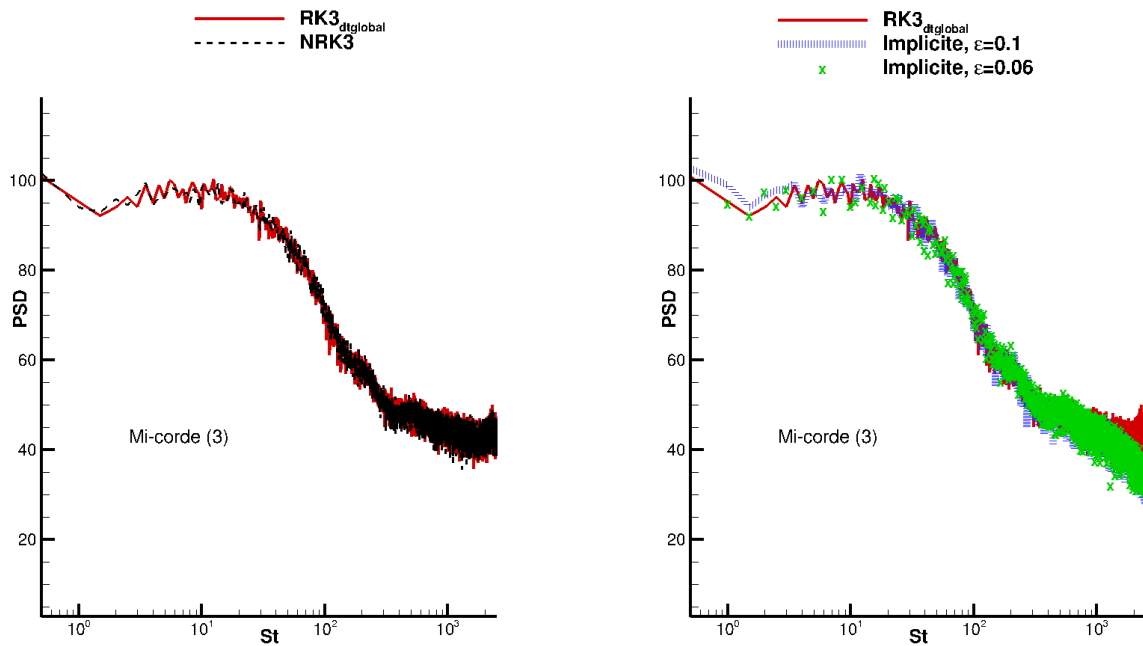


FIGURE 5.19 – Spectre du signal de pression au point d’enregistrement 3, pour les différents schémas.

5.1.6 Comparaison des temps de calcul et bilan de ce cas-test

Le tableau 5.2 montre, pour chaque schéma, les temps de simulation et le speedup par rapport à $RK3_{dtglobal}$. On précise que les simulations ont été effectuées sur un noeud Intel

Xeon Phi 7230 avec 64 threads OpenMP.

	$RK3_{dtglobal}$	NRK3	Implicite $\epsilon = 0.1$	Implicite $\epsilon = 0.06$
Temps CPU (heures)	653	280	231	510
Speedup par rapport à $RK3_{dtglobal}$ (en %)	0	57	65	22

TABLE 5.2 – Temps CPU et speedup par rapport au schéma $RK3_{dtglobal}$ sur le cas-test de l'écoulement autour du profil SD7003.

Le speedup obtenu avec notre schéma à pas de temps local NRK3 est d'environ 57%. C'est en deçà du speedup théorique de 70% déterminé dans la section 5.1.3. Ceci peut s'expliquer par le fait que ce cas présente un nombre important de zones de pas de temps différents. Il y a donc un nombre élevé de zones tampon (les interfaces entre zones de pas de temps différents). Or, les zones tampons ne sont pas prises en compte dans le calcul de speedup théorique; ceci peut donc induire des différences entre le speedup théorique et le speedup mesuré en pratique.

Le speedup obtenu avec notre schéma NRK3 est un peu moins bon que le speedup obtenu avec le schéma implicite muni du critère de convergence $\epsilon = 0.1$ (65%). Néanmoins, nous avons observé que le schéma implicite utilisé avec ce critère de convergence pouvait manquer de précision, en particulier sur le calcul des tensions de Reynolds (voir figures 5.15 et 5.16). Afin d'atteindre un niveau de précision identique à celui de notre schéma NRK3 (notamment sur le calcul des tensions de Reynolds), le critère de convergence doit être réduit à $\epsilon = 0.06$. Dans ce cas, le speedup du schéma implicite chute à 22% : le schéma implicite devient donc moins performant que notre schéma NRK3.

Finalement, ce cas-test démontre que notre schéma à pas de temps local NRK3 peut être utilisé pour réaliser des simulations de type industriel avec une très bonne précision (équivalente à celle du schéma $RK3_{dtglobal}$) et pour un coût CPU compétitif, proche de celui obtenu avec notre schéma implicite.

5.2 Parallélisation OpenMP/MPI du schéma NRK3

Comme évoqué dans la partie 1.2.2, la compatibilité avec les techniques HPC est un critère essentiel dans le choix d'une méthode d'intégration temporelle, au même titre que la précision et le coût CPU de la méthode. La méthode d'intégration temporelle doit pouvoir être utilisée de manière efficace sur une machine multinoeuds (un supercalculateur). En général, un nœud est une machine de calcul composée de plusieurs processeurs (ou sockets). Chaque processeur possède de la mémoire vive (DRAM), des cœurs de calcul et

des mémoires cache (plus petites et plus rapides d'accès que la DRAM). Chaque cœur de calcul peut accéder à l'intégralité de la DRAM attachée à son socket, mais également aux DRAM attachées aux autres sockets du noeud de calcul. Si la bande passante entre un coeur et la DRAM dépend de leur position relative, on parle de mémoire NUMA (Non Uniform Memory Access). Précisons que la mémoire DRAM n'est pas partagée entre les différents nœuds de calcul du supercalculateur, qui sont donc vus comme des entités indépendantes. Pour remédier à ce problème, les nœuds du supercalculateur sont reliés entre eux par des réseaux de communication très rapide ($\approx 100Gb.s^{-1}$) afin de pouvoir transférer des données entre les DRAM de 2 nœuds de calculs. La norme MPI (Message Passing Interface), introduite dans les années 1990, contribue au fonctionnement d'un algorithme sur ce type d'architecture, appelée architecture à mémoire distribuée.

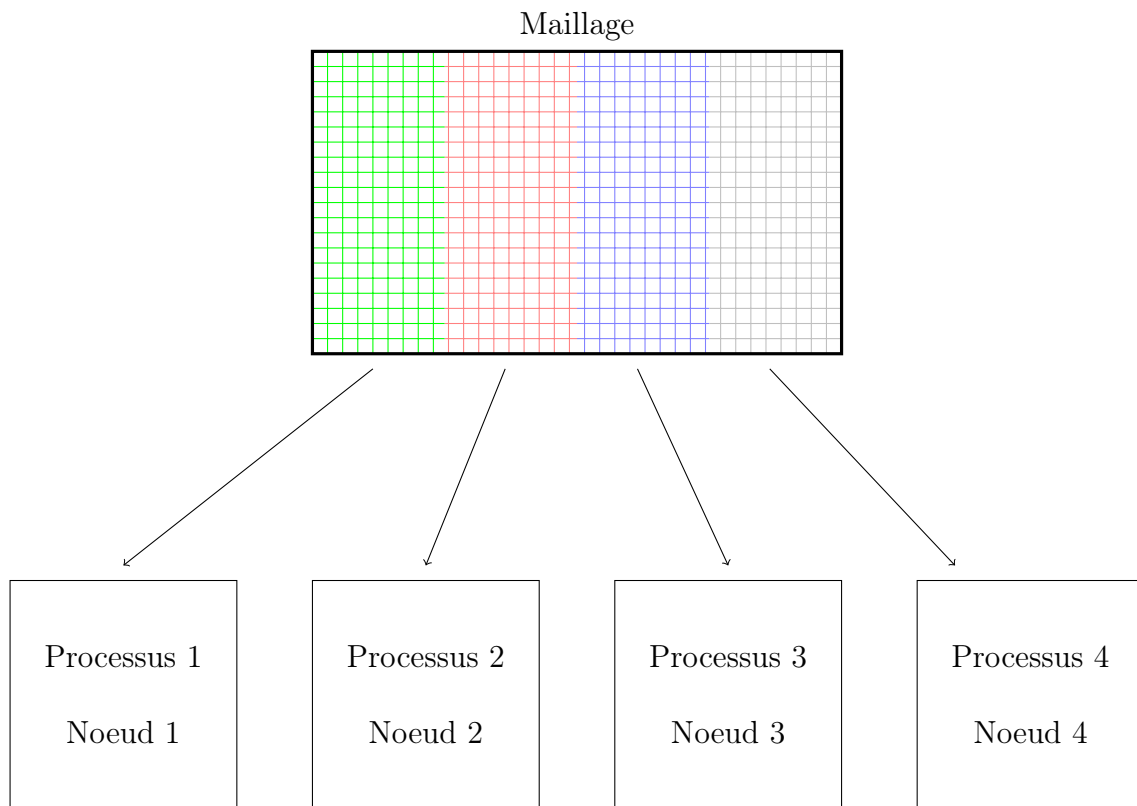


FIGURE 5.20 – Illustration de la parallélisation pure MPI.

En simulation numérique, il est commun de diviser le maillage en plusieurs zones, et de distribuer ces zones sur l'ensemble des nœuds disponibles en créant un processus MPI par zone comme le montre la figure 5.20. Chacun des processus MPI est associé à un noeud et à l'intérieur du noeud son exécution s'effectue sur le même coeur de calcul tout au long de la simulation pour des raisons d'efficacité. Ainsi, les calculs sur les différentes zones du maillage sont effectués simultanément sur les différents nœuds du supercalculateur. Ceci réduit le temps de calcul par rapport au cas séquentiel où un nœud unique calculerait l'ensemble du maillage. Pour les applications numériques utilisant des méthodes numériques à stencil, des données doivent être échangées entre les différents processus MPI via les

réseaux de communication inter-nœuds et des fonctions de la librairie MPI. Dans l'idéal, le temps de calcul obtenu grâce à la parallélisation est égal au temps de calcul séquentiel divisé par le nombre de processus MPI (nombre de nœuds). En pratique, ceci est rarement le cas pour plusieurs raisons :

- Pour communiquer les données entre les différents processus MPI, des buffers mémoires doivent être initialisés du côté du processus envoyeur, tandis que le processus récepteur doit ensuite recopier ces données dans la zone mémoire d'intérêt de l'application (les cellules fictives en général). Suivant les algorithmes, ces échanges peuvent être non bloquants, ce qui autorise le calcul de l'application à se poursuivre jusqu'à un certain point où les données attendues en réception sont vraiment nécessaires. Malgré cette technique de recouvrement du temps de transfert par du calcul, cette étape ajoute toujours un surcoût CPU, dont l'importance va dépendre de la quantité de données à transférer et du temps de calcul nécessaire à l'application entre 2 phases d'échange.
- Le découpage du maillage est rarement parfaitement équilibré entre les processus MPI, surtout lorsqu'il s'agit de découper des maillages structurés. Ainsi, les processus MPI ont des charges de travail (légèrement) différentes et cela oblige les processus les plus rapides à "attendre" les plus lents (lors des phases de communication) ce qui dégrade l'efficacité de la parallélisation.
- Il existe toujours une partie purement séquentielle dans un algorithme, dont la durée est incompressible. Cette dernière s'avère généralement très faible, mais quand le nombre de processus MPI retenu pour le calcul des séquences parallélisées de l'application devient suffisamment grand, le temps CPU la phase séquentielle devient non-négligeable et tend à dégrader l'efficacité de la parallélisation.

Depuis une décennie, une autre stratégie de parallélisation a émergé basée sur l'approche hybride MPI/OpenMP. En effet, sur chaque nœud à mémoire partagée, il est possible de distribuer le travail à réaliser grâce à l'interface de programmation OpenMP qui a été conçue à cet effet. Grâce à la norme OpenMP, le travail à effectuer peut être partagé en plusieurs tâches et la librairie OpenMP va générer des threads (processus légers), qui exécuteront ces différentes tâches simultanément sur les différents cœurs de calcul disponibles. La parallélisation OpenMP retenue dans le code de calcul de cette thèse est de type "coarse grain" : cela consiste à maximiser la taille de la phase parallèle afin de limiter le nombre de synchronisations qui deviennent pénalisantes en coût CPU quand le nombre de threads augmente. La figure 5.21 illustre cette parallélisation hybride : la zone attribuée à chaque processus MPI (nœud) est re-divisée en sous-zones, et chaque thread OpenMP assure le calcul d'une sous-zone dans une partie parallèle (au sens OpenMP). L'approche hybride permet surtout de réduire le nombre de données à échanger et de limiter l'empreinte mémoire pour les calculs réalisés sur des milliers de cœurs de calcul, car la mémoire nécessaire à la gestion des threads OpenMP est beaucoup plus petite que celle réclamée par la gestion des processus MPI.

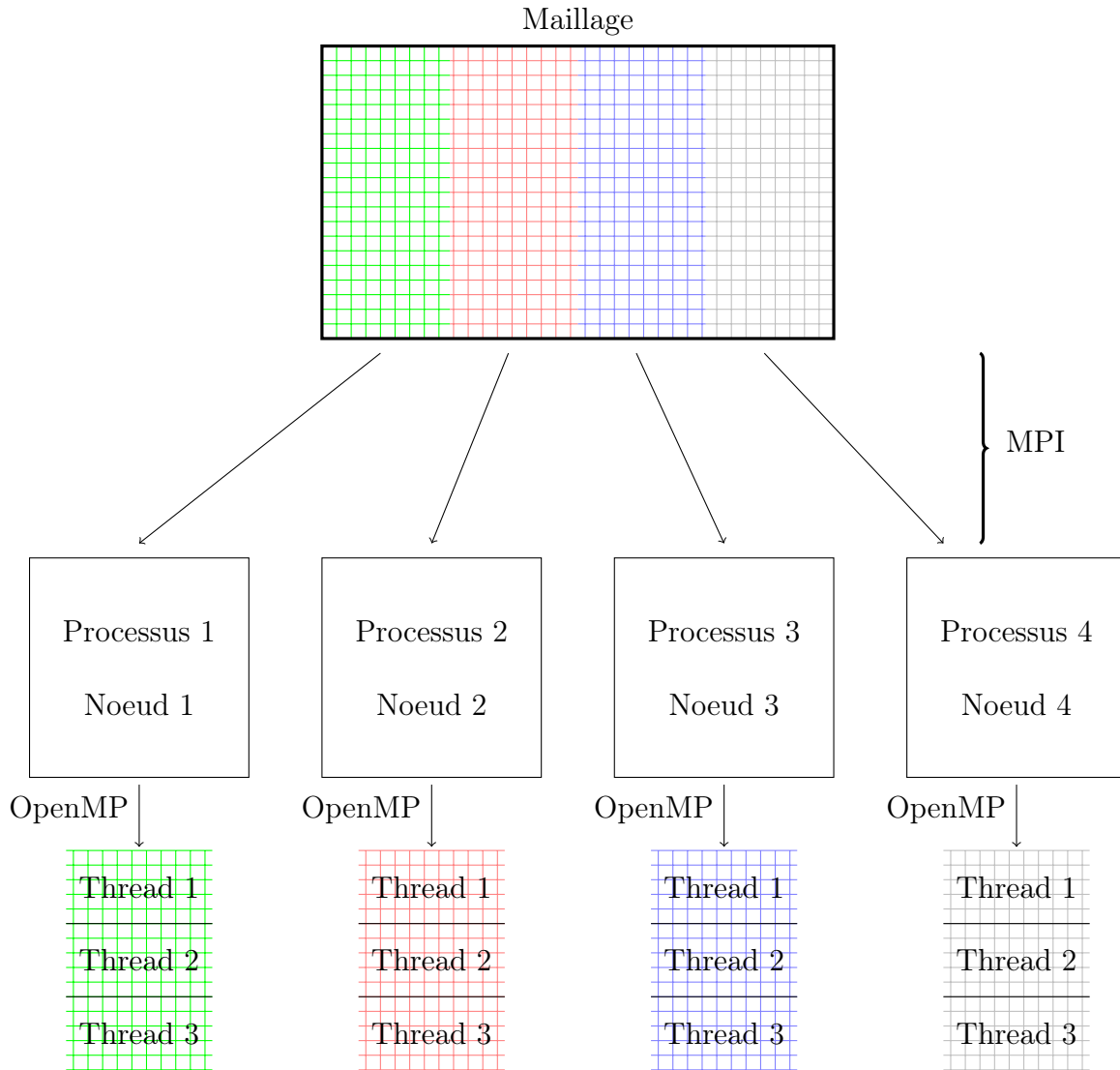


FIGURE 5.21 – Illustration de la parallélisation hybride MPI/OpenMP "coarse grain".

Dans un premier temps, nous décrivons la stratégie mise en place pour assurer une parallélisation MPI efficace de notre schéma à pas de temps local NRK3. Ensuite, nous présentons les deux stratégies mises en place pour assurer la parallélisation OpenMP de notre schéma NRK3.

5.2.1 Parallélisation MPI du schéma NRK3

Pour assurer une parallélisation MPI efficace d'un schéma explicite classique (à pas de temps global), il faut répartir de manière équilibrée les cellules du maillage entre les processus MPI. Ainsi, chaque processus se retrouve avec la même charge de travail, condition nécessaire pour obtenir une réduction efficace du temps de calcul.

Le cas d'un schéma à pas de temps local est différent puisque les cellules ne présentent

pas la même charge de travail. Pour un maillage constitué de cellules de pas de temps Δt et $\frac{\Delta t}{2}$, une application de la méthode de base est effectuée sur les cellules de pas de temps Δt , tandis que deux applications de cette méthode sont effectuées sur les cellules de pas de temps $\frac{\Delta t}{2}$, entre t^n et $t^n + \Delta t$. Les cellules de pas de temps $\frac{\Delta t}{2}$ ont donc une charge de travail deux fois plus importante que les cellules de pas de temps Δt . Si l'on se contente de répartir équitablement l'ensemble des cellules entre les processus sans se soucier de leur charge de travail, il est très probable que l'on induise un déséquilibre de la charge de travail, qui mènera à une piètre efficacité de la parallélisation.

La problématique de la parallélisation MPI d'un schéma à pas de temps local a déjà fait l'objet de plusieurs publications [76, 80, 108]. Ces publications sont unanimes concernant la meilleure stratégie à adopter : pour chaque pas de temps, il faut répartir les cellules de manière équilibrée sur les processus. **Autrement dit, les cellules de pas de temps Δt doivent être réparties équitablement sur les processus, de même pour les cellules de pas de temps $\frac{\Delta t}{2}$, et ainsi de suite jusqu'aux cellules de pas de temps minimal.** Cette stratégie de répartition est illustrée par la figure 5.22. Sur cette figure, l'image de gauche représente un maillage "séquentiel" constitué de trois pas de temps différents : Δt , $\frac{\Delta t}{2}$ et $\frac{\Delta t}{4}$, et de quatre zones de taille identique (numérotées de 1 à 4). La figure de droite montre le maillage "parallèle", constitué de six zones (numérotées de 1 à 6) qui assure une répartition équilibrée de chaque pas de temps sur deux processus MPI (P_1 et P_2). D'une part, les cellules de pas de temps Δt et $\frac{\Delta t}{4}$ sont réparties de manière équilibrée sur P_1 et P_2 . En effet, les zones de pas de temps Δt et $\frac{\Delta t}{4}$ sont découpées en deux zones structurées de taille identique et ces deux zones sont réparties sur P_1 et P_2 . D'autre part, les cellules de pas de temps $\frac{\Delta t}{2}$ sont également réparties de manière équilibrée entre les deux processus puisque la première zone de pas de temps $\frac{\Delta t}{2}$ est attribuée au processus P_1 et la seconde zone, de même taille, est attribuée au processus P_2 .

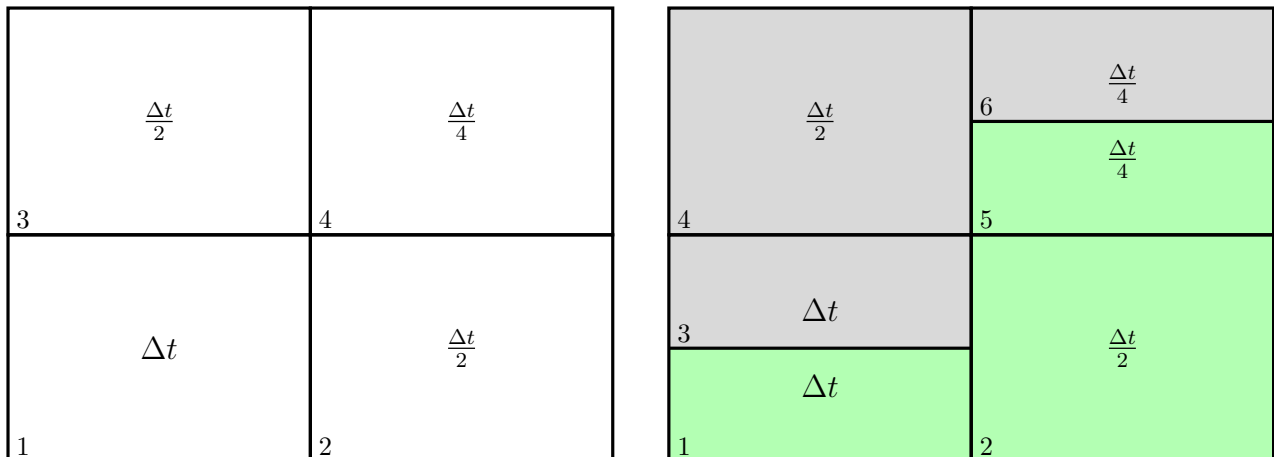


FIGURE 5.22 – Stratégie de répartition des cellules sur les processus MPI pour le schéma NRK3. **A gauche** : maillage "séquentiel" constitué de trois pas de temps différents. **A droite** : maillage "parallèle", qui assure une répartition équilibrée des cellules sur deux processus MPI : P_1 et P_2 .

Précisons que la répartition des cellules de pas de temps $\frac{\Delta t}{2^i}$ sur les processus MPI est automatique : elle est assurée par le module Distributor du logiciel Cassiopee [109] développé à l'ONERA. Le logiciel procède lui-même au découpage de certaines zones (si nécessaire) et répartit de manière équilibrée toutes les zones de pas de temps $\frac{\Delta t}{2^i}$ sur les processus MPI.

Les zones nouvellement créées par le découpage MPI sont elles-aussi dotées de cellules fictives pour recevoir les informations des zones adjacentes. Ceci est illustré par la figure 5.23, qui montre une vue éclatée du maillage "parallèle".

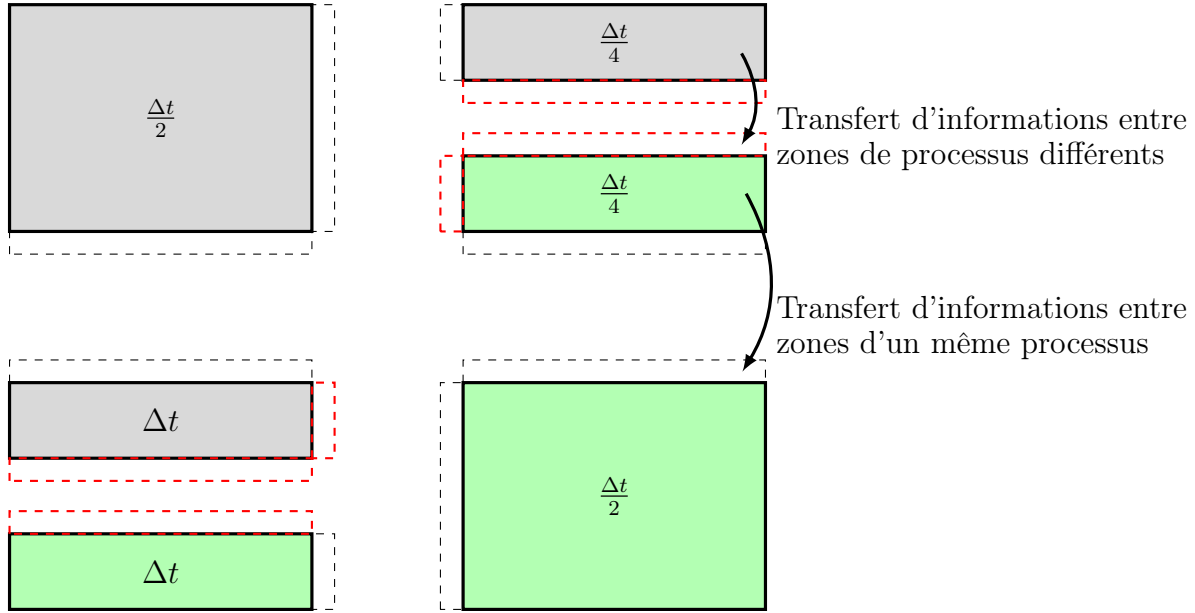


FIGURE 5.23 – Vue éclatée du maillage "parallèle" pour faire apparaître les cellules fictives (représentées par des traits pointillés) au niveau des raccords entre zones.

Comme le montre cette figure, les cellules fictives peuvent recevoir des informations provenant d'une zone adjacente qui appartient au même processus MPI qu'elles ; ces cellules fictives sont représentées en traits pointillés noirs sur la figure 5.23. Les cellules fictives peuvent également recevoir des informations provenant d'une zone adjacente appartenant à un processus MPI différent ; ces cellules fictives sont représentées en traits pointillés rouges sur la figure 5.23. Le volume de communications MPI est donc relié au nombre de cellules fictives qui reçoivent des informations provenant d'une zone adjacente de processus MPI différent. On note $ncell_{MPI}$ ce nombre de cellules. On définit le volume des communications MPI, noté Vol_{MPI} comme le rapport entre $ncell_{MPI}$ et le nombre total de cellules sur le maillage $ncell_{tot}$:

$$Vol_{MPI} = \frac{ncell_{MPI}}{ncell_{tot}}. \quad (5.16)$$

L'autre paramètre que nous utilisons pour mesurer l'efficacité de notre implémentation est le speedup S , défini comme le rapport :

$$S = \frac{T_{A,1}}{T_{A,p}} \quad (5.17)$$

où $T_{A,p}$ désigne le temps de calcul obtenu avec un certain schéma A sur p processus et $T_{A,1}$ désigne le temps de calcul obtenu avec le même schéma sur 1 processus.

Résultats

Nous avons testé la scalabilité forte de notre schéma NRK3 et du schéma $RK3_{dtglobal}$ sur le cas d'écoulement autour du profil SD7003. Le test de scalabilité forte consiste à augmenter le nombre de processus MPI p tout en gardant un nombre de cellules constant et à mesurer le speedup S pour chaque valeur de p . Contrairement au calcul de la section précédente 5.1.1 réalisé sur un noeud Intel Xeon Phi KNL sur 64 cœurs par une approche pure OpenMP, ce test est réalisé sur un noeud Intel Broadwell possédant 2 sockets de 14 cœurs. La valeur p maximale testée est donc de $p = 28$. En effet, le module Connector de la suite Cassiopee qui gère les échanges MPI entre les zones n'est pas optimisé pour ce processeur KNL à l'architecture singulière (allocation dynamique de mémoire très lente), ce qui rendrait l'exercice peu pertinent.

Précisons que le schéma $RK3_{dtglobal}$ a été choisi comme point de comparaison car il s'agit d'un schéma explicite à pas de temps global; ce type de schéma est réputé pour ses bonnes propriétés de parallélisation MPI. Les temps de calcul sont mesurés au bout de 100 itérations temporelles. Un dernier paramètre très important concernant les études de scalabilité concerne l'optimisation HPC de la partie calcul : plus une application calcule vite sur un coeur, plus la scalabilité est difficile à obtenir. Le solveur FastS, employé pour ces simulations, a été optimisé récemment [110] lors d'une collaboration avec la société Intel : avec les options numériques du calcul (schéma AUSM+(P) senseur et $RK3_{dtglobal}$), chaque coeur de calcul est capable de calculer une étape de la méthode RK3 sur environ 10^7 cellules en une seconde.

La qualité de la répartition des cellules sur les différents processus est illustrée dans les tableaux 5.3 ($RK3_{dtglobal}$) et 5.4 ($NRK3$). Pour le schéma $RK3_{dtglobal}$, la parallélisation du calcul sur 28 cœurs entraîne un accroissement du nombre de zones à calculer qui passe de 3 (1 coeur) à 31 (28 cœurs). L'équilibrage reste très bon, puisque la déviation maximale est de 2.3% (p=28) comme l'indique la colonne VarMax. Le graphe de communication reste assez léger, puisque chaque processus réalise en moyenne 5 communications point à point (communications avec d'autres processus), pour $p = 28$. Pour le schéma $NRK3$, le nombre de zones s'accroît rapidement avec le nombre de processus : pour p=28, le maillage est décomposé en 189 zones et l'équilibrage est imparfait. Pour les zones de pas de temps Δt , ce déséquilibre atteint quasiment 8 % pour p=28 et le graphe de communication est beaucoup plus complexe, puisque chaque processus réalise en moyenne 24 communications point à point.

Nb processus MPI	VarMax(%)	Nb communication point à point			Nb zones
	$\Delta t/8$	moy	max	min	
1	0	1	1	1	3
2	0,02	2	2	2	5
4	0,09	3	4	3	7
8	0,2	5	6	3	13
16	0,7	4	8	3	19
20	0,8	6	15	3	29
28	2,3	5	11	3	31

TABLE 5.3 – Caractéristiques des maillages découpés en vue du calcul MPI avec le schéma $RK3_{dtglobal}$

Nb processus MPI	VarMax (%)				Nb communication point à point			Nb zones
	Δt	$\Delta t/2$	$\Delta t/4$	$\Delta t/8$	moy	max	min	
1	0	0	0	0	1	1	1	67
2	0,04	0,9	0,2	1,2	2	2	2	71
4	0,7	0,9	0,6	0,8	4	4	4	78
8	3	0,6	0,6	2,6	8	8	8	99
16	8,7	5,9	2,2	2,4	15	16	12	124
20	3,2	1,3	1,5	6,5	18	20	15	145
28	7,8	2,4	2,5	3,5	24	28	20	189

TABLE 5.4 – Caractéristiques des maillages découpés en vue du calcul MPI avec le schéma $NRK3$

La figure 5.24 montre les résultats de l'étude de scalabilité forte pour les schémas $RK3_{dtglobal}$ et $NRK3$. Pour le schéma $RK3_{dtglobal}$, la perte de scalabilité (speedup $S \approx 22$ pour $p=28$) est principalement due à l'éviction de données du cache L3 qui augmente à l'instar de p [110]. En effet, sur la majorité des processeurs actuels, certaines mémoires cache (le niveau 3 sur les processeur Intel par exemple) sont partagées par différents cœurs de calcul. En augmentant le nombre de processus MPI, la mémoire cache disponible par processus diminue et la vitesse d'exécution du code ralentit si ce dernier a besoin de données présentes dans la DRAM, beaucoup plus lente que le cache. Pour le schéma $NRK3$, les speedup obtenus sont un peu moins bons puisqu'un speedup $S \approx 18$ est obtenu pour $p=28$. Compte tenu du moins bon découpage réalisé pour le schéma $NRK3$ (graphe plus complexe, déséquilibré), les speedup obtenus avec le schéma $NRK3$ ne sont pas si éloignés des speedup obtenus avec le schéma $RK3_{dtglobal}$. De plus, le volume des communications MPI est une autre raison qui explique la différence de scalabilité entre les deux schémas. La figure 5.25 montre l'évolution du volume des communications MPI en fonction du nombre de processus p . On remarque que le volume des communications MPI est plus important pour le schéma $NRK3$ que pour le schéma $RK3_{dtglobal}$. Pour $p = 28$, il y a notamment un rapport deux entre le volume des communications MPI de chaque schéma. Ceci s'explique par le fait que la stratégie de répartition des cellules utilisée pour le schéma $NRK3$ est plus complexe que la stratégie de répartition "classique", et tend à complexifier le graphe de communications MPI tout en augmentant la taille des données à échanger. Cette observation rejoint les travaux de Sény [80, 76] (2014), qui a montré dans un contexte de maillage non structuré que la stratégie de répartition des cellules utilisée pour les schémas à pas de temps local menait à un volume des communications MPI plus grand par

rapport à une stratégie de répartition classique. L'introduction de davantage de recouvrement calcul/transfert permettrait certainement d'améliorer la parallélisation MPI du schéma $NRK3$, mais cette dernière a été jugée satisfaisante pour le cadre de cette thèse et ce travail d'optimisation n'a pas été poussé plus loin.

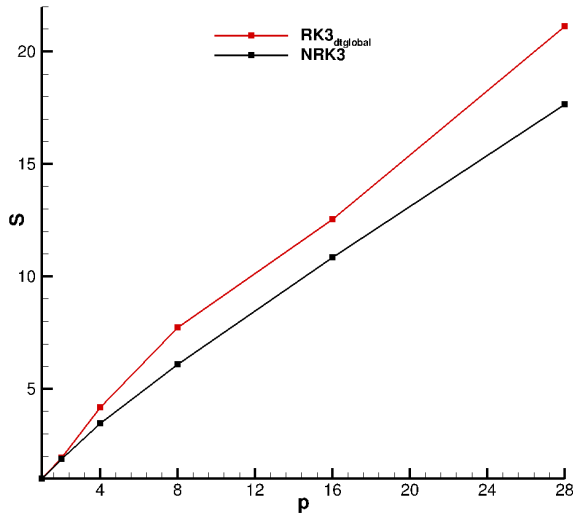


FIGURE 5.24 – Test de scalabilité forte : Speedup en fonction du nombre de processeurs p pour les schémas $RK3_{dtglobal}$ et $NRK3$.

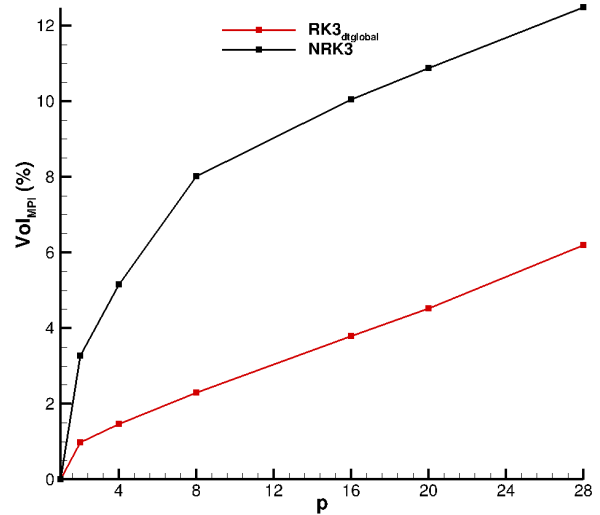


FIGURE 5.25 – Calcul du volume des communication MPI pour les schémas $RK3_{dtglobal}$ et $NRK3$.

5.2.2 Parallélisation OpenMP du schéma $NRK3$

Une autre piste prometteuse pour améliorer la parallélisation du schéma $NRK3$ réside dans l'approche OpenMP, qui nécessite moins de découpage que l'approche MPI et devrait donc limiter la quantité de donnée à transférer. Deux stratégies de parallélisation OpenMP ont été testées durant cette thèse. La première stratégie est appelée "OpenMP mode 0". Cette stratégie était déjà présente dans le code avant le début de ces travaux de thèse. La seconde est appelée "OpenMP mode 1", elle a été mise en place pour remédier à certaines difficultés posées par la stratégie "OpenMP mode 0".

OpenMP mode 0

On considère le même maillage structuré que celui présenté sur la figure 5.22 (image de gauche), constitué de quatre zones et de trois pas de temps. Ce maillage est rappelé sur la figure 5.26 (image de gauche). La stratégie OpenMP mode 0 consiste, **pour chaque zone**, à répartir les cellules de manière équilibrée sur les threads. Ceci est illustré par la figure 5.26 (image de droite), dans le cas où deux threads sont utilisés : T_1 et T_2 . Remarquons que cette stratégie de parallélisation a du sens. En effet, l'algorithme du schéma $NRK3$ (algorithme 1, page 121) montre que toutes les zones sont parcourues entre l'étape $q = 1$ et l'étape $q = q^*$ (certaines zones plus que d'autres, en fonction de leur pas de temps).

Théoriquement, si l'on parvient par exemple à diviser par deux le temps de calcul sur chaque zone (en répartissant équitablement les cellules de chaque zone sur deux threads), on s'attend à ce que le temps de calcul global soit également divisé par deux.

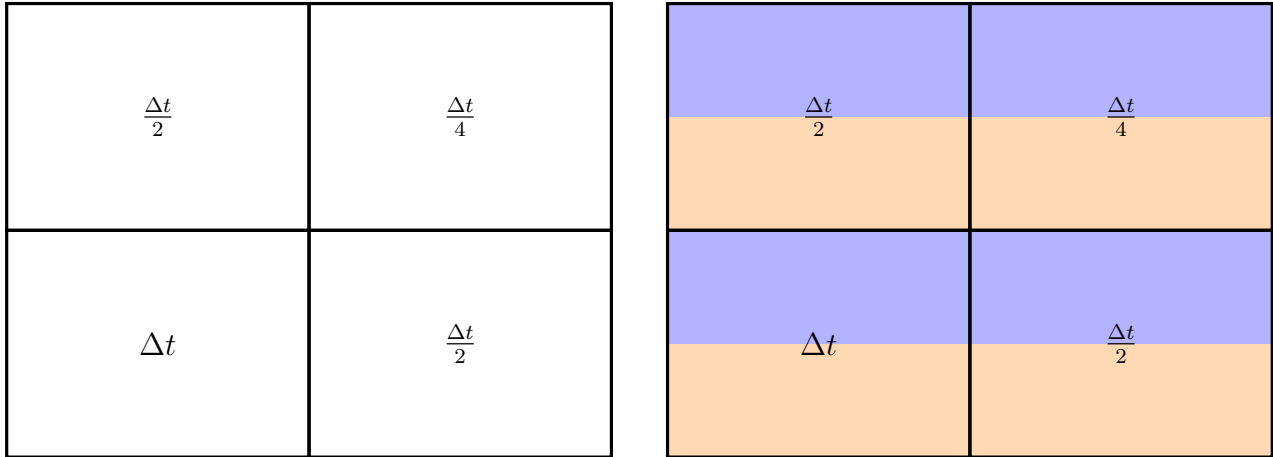


FIGURE 5.26 – Illustration de la stratégie OpenMP mode 0. **A gauche** : maillage structuré constitué de trois pas de temps différents. **A droite** : illustration de la répartition de deux threads OpenMP (Th_1 et Th_2) sur le maillage.

Précisons que contrairement au MPI, aucune nouvelle zone n'est créée lors de la répartition des cellules entre les threads (il n'y a pas de découpage des zones). Des plages d'indices structurées sont attribuées aux threads pour spécifier leur région de "travail" au sein d'une zone et des verrous sont placés dans le solveur pour gérer la synchronisation entre les threads.

Comme pour le cas de la parallélisation MPI, la scalabilité forte a été étudiée pour le schéma NRK3 et le schéma $RK3_{dtglobal}$: on augmente le nombre de threads (tout en gardant le même maillage) et on mesure pour chaque nombre de threads le speedup obtenu. Dans ce cas, le speedup S d'un schéma A est défini par :

$$S = \frac{T_{A,1}}{T_{A,Th}}, \quad (5.18)$$

où $T_{A,1}$ correspond au temps de calcul du schéma A sur un thread, et $T_{A,Th}$ désigne le temps de calcul obtenu avec Th threads.

La figure 5.27 montre le résultat de l'étude de scalabilité forte pour les schémas $RK3_{dtglobal}$ et NRK3 (noté NRK3 mode 0 car la stratégie OpenMP mode 0 est utilisée pour répartir les cellules sur les threads). L'étude de scalabilité est effectuée sur le cas de l'écoulement autour du profil SD7003. Les temps de simulation sont mesurés au bout de 100 itérations temporelles. Pour le schéma $RK3_{dtglobal}$ le maillage comporte 3 zones tandis que pour NRK3 la grille est décomposée en 67 zones (voir tableau 5.4 pour $p=1$)

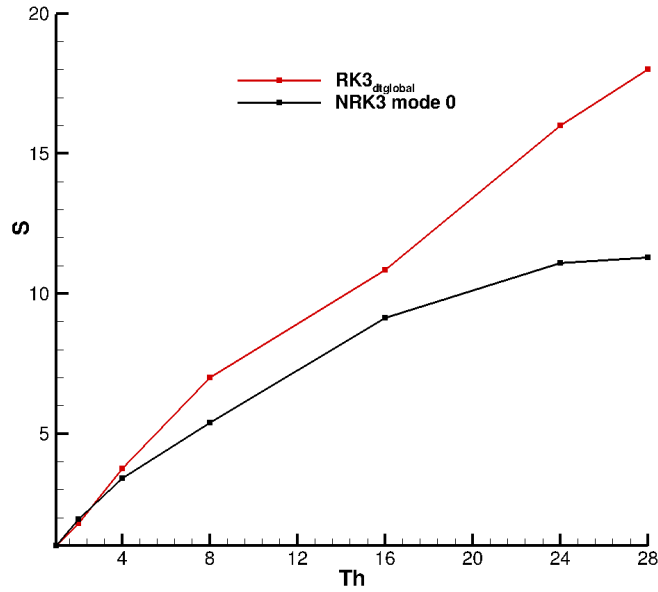


FIGURE 5.27 – Test de scalabilité forte en OpenMP : Speedup en fonction du nombre de threads Th pour les schéma $RK3_{dtglobal}$ et NRK3.

La figure 5.27 montre une différence notable entre les deux courbes de scalabilité, en particulier lorsque le nombre de threads devient supérieur à 16. Par exemple, pour un nombre de threads $Th = 28$, un speedup $S \approx 18$ est obtenu avec le schéma $RK3_{dtglobal}$, tandis qu'un speedup $S \approx 11$ est obtenu avec le schéma NRK3. Pour le schéma $RK3_{dtglobal}$ la perte de scalabilité par rapport à l'approche MPI ($S \approx 22$) s'explique en partie par l'accès NUMA à la mémoire et une performance moindre du découpeur utilisé pour l'équilibrage OpenMP. Le maillage sur lequel le schéma NRK3 est utilisé contient plusieurs zones de petite taille. Or l'algorithme qui répartit le travail entre les threads est un algorithme simplifié par rapport à celui présent dans le module Distributor : il engendre souvent un déséquilibre de l'ordre de 10% ; de plus le découpage du travail ne peut se faire que dans deux directions uniquement pour simplifier le processus de synchronisation entre threads. Il arrive qu'une zone soit trop petite pour que l'ensemble des threads puissent la calculer simultanément : dans ce cas le nombre de threads actifs pour le calcul de cette petite zone est diminué temporairement. Nous avons donc mis en place une deuxième stratégie de répartition des cellules sur les threads (appelée "OpenMP mode 1") sensée réduire les contraintes de répartition.

OpenMP mode 1

La stratégie de répartition "OpenMP mode 1" suit le même principe que la stratégie de répartition des cellules utilisée pour la parallélisation MPI : **pour chaque pas de temps, l'ensemble des cellules est réparti de manière équilibrée entre les threads**. La figure 5.28 illustre la stratégie de répartition "OpenMP mode 1" sur deux threads Th_1 et Th_2 . Sur les maillages que l'on utilise, il y a en général plusieurs zones de même pas

de temps. Ainsi, avec la stratégie OpenMP mode 1 les répartitions sont effectuées sur des ensembles de cellules plus grands que pour la stratégie OpenMP mode 0. Ceci facilite une répartition équilibrée des cellules, surtout lorsque le nombre de threads est élevé.

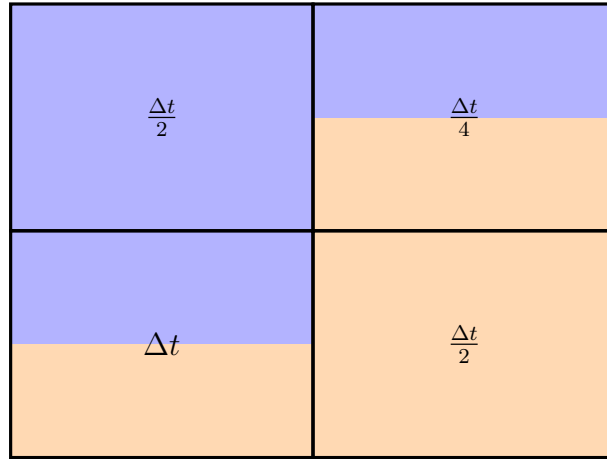


FIGURE 5.28 – Illustration de la stratégie OpenMP mode 1 pour répartir les cellules sur deux threads OpenMP : Th_1 et Th_2 .

La figure 5.29 montre le résultat de scalabilité du schéma NRK3, lorsque la stratégie de répartition OpenMP mode 1 est utilisée (courbe NRK3 mode 1). Les courbes de scalabilité du schéma $RK3_{dt_{global}}$ et du schéma NRK3 muni de la stratégie OpenMP mode 0 (NRK3 mode 0) sont rappelées.

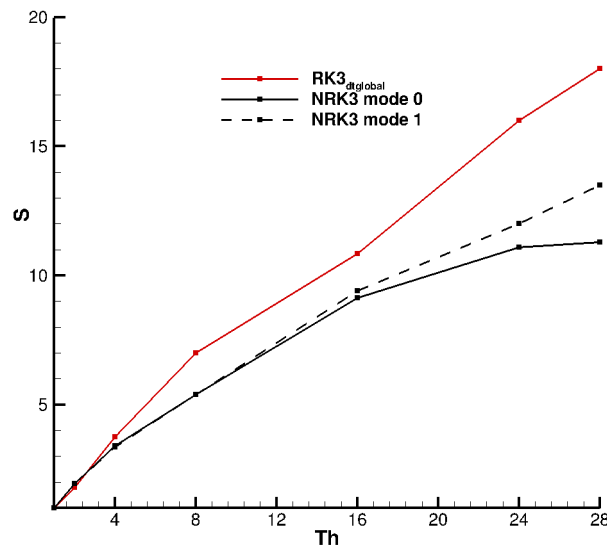


FIGURE 5.29 – Test de scalabilité forte en OpenMP : Speedup en fonction du nombre de threads Th pour les schéma $RK3_{dt_{global}}$ et NRK3 (stratégies OpenMP mode 0 et mode 1).

L'utilisation de la stratégie OpenMP mode 1 améliore légèrement les résultats de scalabilité du schéma NRK3, en particulier lorsque le nombre de threads augmente. Par exemple, pour $Th = 28$, le speedup obtenu avec le schéma NRK3 est $S \approx 14$ lorsque la stratégie OpenMP mode 1 est utilisée. On rappelle qu'un speedup $S \approx 11$ est obtenu lorsque stratégie OpenMP mode 0 est utilisée. Ces résultats décevants du mode 1 s'expliquent en partie par le fait que le solveur est optimisé pour tirer un bon parti du cache L3. Or il a été constaté que le temps de calcul par cellule pouvait être influencé significativement par la taille de la grille structurée à résoudre. Avec un code sur maillage structuré, les accès mémoire sont réguliers. Pour certaines tailles de grille, on constate une éviction des données du cache L3 beaucoup plus marquée (et donc une perte d'efficacité du solveur FastS). Ceci est illustré par la figure 5.30 qui montre le temps de calcul par cellule pour différentes tailles de grille. On distingue clairement que de faibles variations de la taille I ou J de la grille impactent l'efficacité du solveur (augmente le temps de calcul par cellule). Pour certaines tailles critiques, ce ralentissement peut atteindre 50%. Comme avec le mode 1 les threads ne travaillent pas forcément sur les mêmes zones, il peut se créer un déséquilibre important même si le nombre de cellules à mettre à jour par chacun des threads est parfaitement équilibré. Un équilibrage tenant compte de ces vitesses variable pourrait être envisagé à l'avenir pour améliorer l'efficacité de la parallélisation OpenMP.

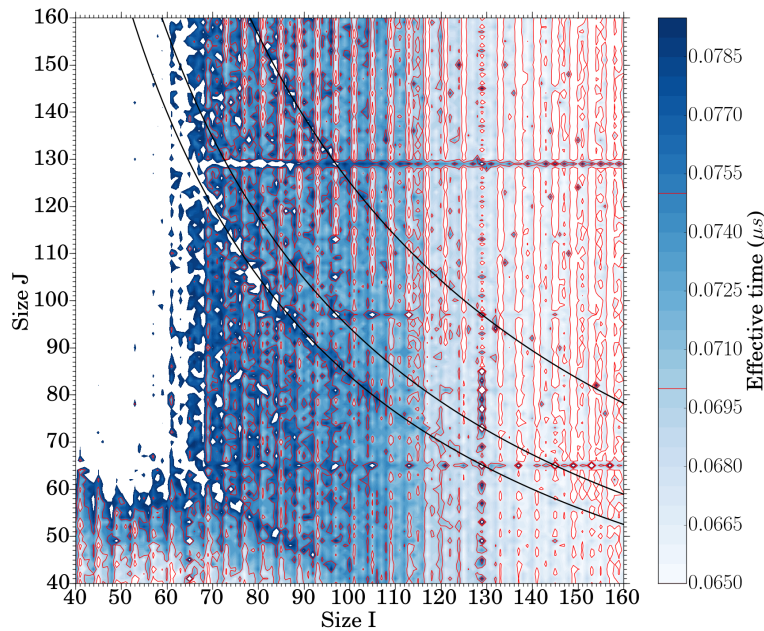


FIGURE 5.30 – Solveur FastS : variation du temps de calcul par cellule en fonction de la taille I et J de la zone structurée.

5.3 LES d'un écoulement 3D autour d'un cylindre par une méthode de frontières de immergées

La particularité des méthodes de frontières immergées (Immersed Boundary Method, IBM) réside dans le maillage utilisé : il s'agit d'un maillage cartésien de type "Octree" [111], qui "recouvre" les parois de l'obstacle autour duquel s'écoule le fluide (le maillage n'est pas adapté aux contours de l'obstacle). Précisons que pour les maillages cartésiens de type "Octree", le pas d'espace est systématiquement divisé par deux d'une zone de raffinement à l'autre, comme cela est montré sur la figure 5.31.

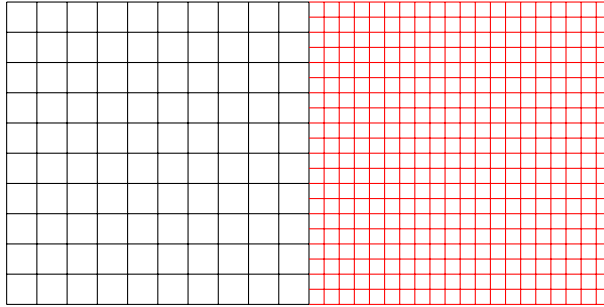


FIGURE 5.31 – Illustration d'un maillage Octree cartésien constitué de deux zones : le pas d'espace dans la zone rouge est deux fois plus faible que dans la zone noire (et ce, dans toutes les directions). Le pas d'espace est uniforme dans chaque zone.

De plus, les mailles sont isotropes (le pas d'espace est identique dans les trois directions de l'espace). Les méthodes de frontières immergées se distinguent des méthodes "classiques" (basées sur un maillage qui suit les parois de l'obstacle) sur plusieurs points :

- La génération d'un maillage cartésien Octree non adapté aux parois est beaucoup plus simple et rapide que la génération d'un maillage adapté aux parois de l'obstacle : elle peut même être automatisée, comme cela est proposé dans [111]. Ceci est particulièrement intéressant quand on sait que l'étape de maillage est généralement une étape assez longue dans le processus de mise en place d'une simulation. Grâce au maillage non adapté, les méthodes de frontières immergées permettent également de simuler des écoulements autour de géométries complexes.
- L'utilisation de mailles isotropes implique que la résolution est la même dans les trois directions de l'espace : le choix d'une résolution très fine à la paroi peut donc conduire à un nombre de points extrêmement élevé. Par conséquent, une résolution très fine à la paroi ne peut pas être utilisée dans le cas d'une méthode de frontières immergées. Ainsi, l'écoulement dans les régions de couche limite n'est pas "directement" résolu. Les méthodes de frontières immergées nécessitent l'utilisation d'une loi de parois pour modéliser l'écoulement dans ces régions. En revanche, l'emploi d'une résolution plus lâche à la paroi limite la contrainte CFL sur le pas de temps. Un pas de temps supérieur à celui utilisé dans le cas d'une méthode classique (avec un maillage adapté à l'obstacle et une résolution très fine à la paroi) peut donc être

choisi.

On se propose d'effectuer une simulation LES de l'écoulement d'air autour d'un cylindre en utilisant la méthode de frontières immergées disponible dans le code de calcul utilisé pour ces travaux de thèse. Cette méthode de frontières immergées est encore en développement et n'a pas été validée sur des cas d'écoulements autour de géométries complexes. Par contre, elle a été validée pour des géométries plus simples : écoulement 2D autour du profil NACA0012 [85], écoulement autour d'un cylindre 3D [85], écoulement autour d'un fuselage simplifié d'hélicoptère [112]. C'est la raison pour laquelle nous avons choisi d'effectuer une simulation d'écoulement autour d'un obstacle présentant une géométrie simple (cylindre).

Par ailleurs, notre schéma NRK3 nous paraît bien adapté au maillage cartésien de type Octree utilisé par la méthode de frontières immergées. En effet, vu que le pas d'espace est systématiquement divisé par deux d'une zone de raffinement à l'autre, le nombre CFL est lui multiplié par deux. Pour assurer la stabilité, il faut donc diviser le pas de temps par deux entre deux zones de raffinement, et c'est justement ce qui est fait par notre schéma NRK3.

5.3.1 Quelques généralités sur les méthodes de frontières immergées

Dans le cas des méthodes de frontières immergées, la principale difficulté se présente lorsqu'il s'agit d'imposer les conditions aux limites au niveau des parois de l'obstacle immergé. C'est l'approche utilisée pour imposer ces conditions aux limites qui différencie une méthode de frontière immergée d'une autre. On peut citer de manière non exhaustive la première approche développée par Peskin (1972) [113], et l'approche "Ghost Cell" utilisée notamment par Fadlun *et al* (2000) [114] et Tseng *et al* [115]. L'approche utilisée par la méthode de frontières immergées de notre code est une approche "Ghost cell", similaire à celle utilisée dans [114, 115].

La figure 5.32 illustre les différents type de points rencontrés dans la méthode de frontières immergées utilisée dans notre code. Il y a :

- les points calculés : il s'agit des points où les équations de Navier-Stokes sont résolues.
- les points IBC : ces points sont généralement situés sur les premières couches de cellules près de la paroi, dans la zone fluide. La solution en ces points est reconstruite (à partir de la solution aux points calculés) en se basant sur la loi de paroi de Musker [116]. Ces points permettent donc de prendre en compte la présence de la paroi.
- les points solides, sur lesquels aucun calcul n'est effectué.

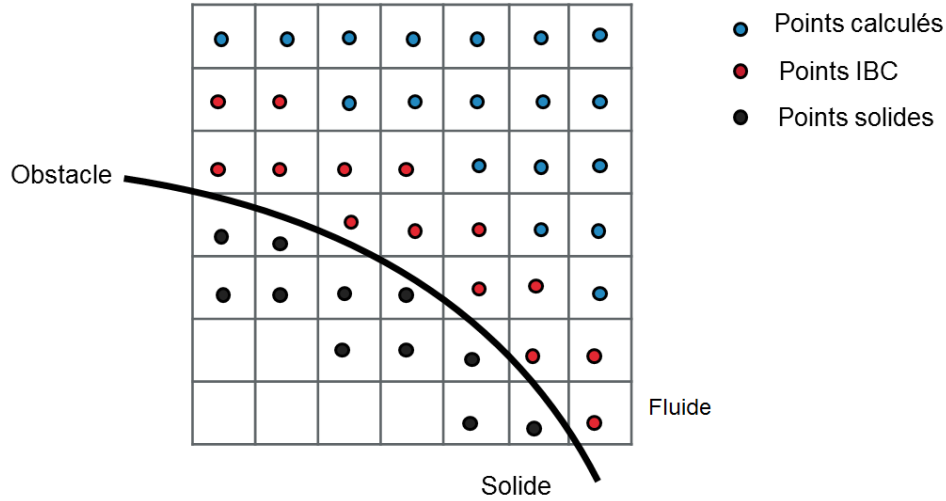


FIGURE 5.32 – Les différents types de points rencontrés dans notre méthode de frontières immergées.

5.3.2 Présentation du cas et maillage

Le cas consiste à étudier l'écoulement d'air autour d'un cylindre. Le nombre de Reynolds de cet écoulement basé sur le diamètre d du cylindre et la vitesse à l'infini U_∞ vaut : $Re = 3900$. Le nombre de Mach basé sur U_∞ vaut : $M = 0.1$. Les valeurs de U_∞ et d sont : $U_\infty = 34 \text{ m.s}^{-1}$ et $d = 1,7 \times 10^{-3} \text{ m}$. La littérature montre que pour un tel nombre de Reynolds, l'écoulement devient turbulent dans le sillage du cylindre [85, 117]. Dans [117], les auteurs précisent que dans ces conditions, des lâchers tourbillonnaires en aval du tourbillon sont effectués à la fréquence $f = 0.2 \times U_\infty/d$.

Le maillage est tridimensionnel ; il est constitué d'environ 90 millions de cellules et de 193 zones cartésiennes de type Octree. La figure 5.33 montre une vue 2D du maillage à proximité du cylindre (on a volontairement effacé les cellules recouvrant le cylindre pour plus de clarté). On peut y distinguer les zones de raffinement successifs. Près du cylindre, les pas d'espaces dans les direction \vec{x} et \vec{y} valent : $\Delta x = \Delta y = 0.003 d$. Les bords \vec{x} et \vec{y} du maillage sont situés à une distance égale à $1000d$ du centre du cylindre. A proximité de ces bords, les pas d'espace valent : $\Delta x = \Delta y = 6.0 d$. Le pas d'espace dans la direction \vec{z} est uniforme et vaut toujours : $\Delta z = 0.03 d$. Il y a 96 cellules dans la direction \vec{z} . Les conditions aux limites au niveau des bords \vec{x} et \vec{y} du maillage sont des conditions de non-réflexion [105]. Des conditions de périodicité sont utilisées dans la direction \vec{z} .

Comme pour le cas d'écoulement autour du profil SD7003, c'est l'écoulement en régime établi qui nous sert de solution initiale. Il est obtenu en deux temps :

- Une première simulation RANS est effectuée pour obtenir le champ moyen en régime établi
- A partir de cette solution moyennée, le schéma $RK3_{dtglobal}$ est utilisé pour atteindre le temps $t = 40 \times d/U_\infty$

On s'assure ainsi que l'on dépassé le régime transitoire et atteint le régime établi.

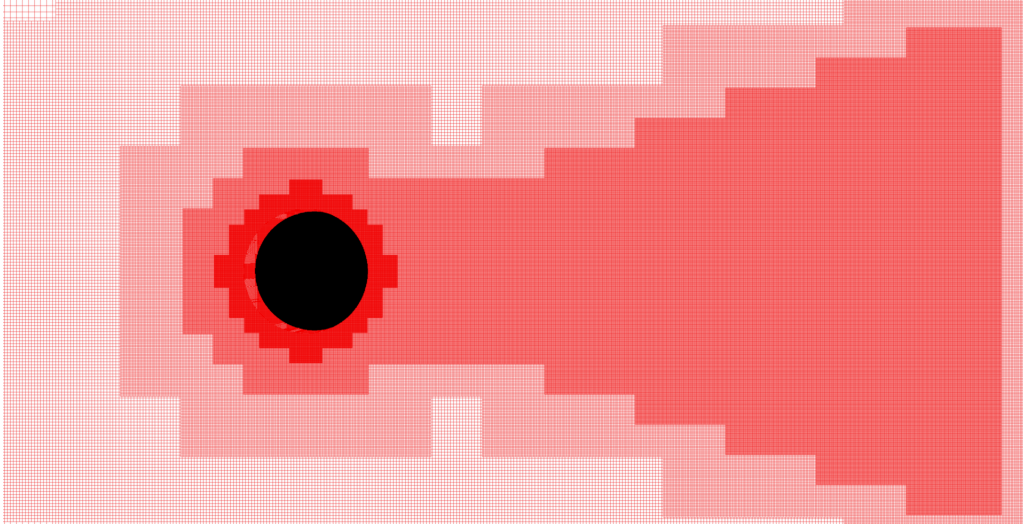


FIGURE 5.33 – Vue 2D du maillage à proximité du cylindre. Les cellules du maillage sensées recouvrir le cylindre ne sont affichées pour plus de clarté.

Détermination des différents pas de temps pour le schéma NRK3

A partir de la solution initiale, on peut calculer la condition CFL sur chaque cellule et on détermine le pas de temps minimal Δt_{min} et le pas de temps maximal Δt_{max} sur le maillage. Les pas de temps Δt_{min} et Δt_{max} valent :

$$\Delta t_{min} = 2.5 \times 10^{-4} \times d/U_{\infty}, \quad (5.19)$$

$$\Delta t_{max} = 2.5 \times 10^{-3} \times d/U_{\infty}. \quad (5.20)$$

On a donc un rapport 10 entre le pas de temps minimal et le pas de temps maximal. Ce chiffre peut paraître faible au vu du rapport, qui est de l'ordre de 2000, entre les pas d'espace Δx et Δy "loin" du cylindre et ces mêmes pas d'espace à proximité du cylindre. Le rapport entre Δt_{max} et Δt_{min} est en fait limité par la valeur uniforme du pas d'espace dans la direction \vec{z} (Δz) qui est toujours dix fois plus grand que les pas d'espace Δx et Δy minimaux.

En utilisant la formule (5.14) de la page 162, on obtient l'exposant z du pas de temps maximal : $z = 3$. Pour le schéma NRK3, le pas de temps maximal sur le domaine de calcul vaut donc :

$$\Delta t = 2^3 \Delta t_{min} = 8\Delta t_{min} = 2 \times 10^{-3} \times d/U_{\infty}.$$

Il y a en tout quatre pas de temps sur le domaine de calcul : Δt_{min} , $2 \Delta t_{min}$, $4 \Delta t_{min}$ et $8 \Delta t_{min}$. Contrairement au cas du profil SD7003, nous n'avons pas besoin de découper

le maillage en zones (en vue de leur attribuer un pas de temps). En effet, le maillage est déjà découpé en 193 zones, et le passage d'un niveau de raffinement à un autre se fait systématiquement au moyen d'une nouvelle zone, comme le montre la figure 5.31. Précisons que chaque zone est dotée de cellules fictives pour récupérer les données de la zone voisine. Il nous "suffit" donc de parcourir ces 193 zones et de leur attribuer un des quatre pas de temps en fonction de la CFL locale sur la zone. La figure 5.34 montre la répartition des différents pas de temps sur le maillage. Les pas de temps les plus faibles sont situés à proximité du cylindre et dans le sillage, où l'écoulement est turbulent.

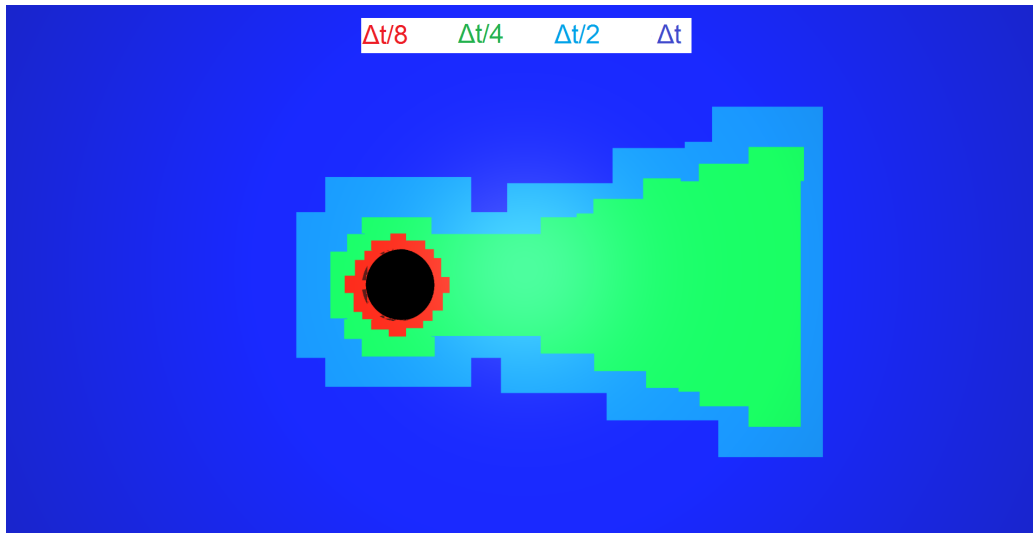


FIGURE 5.34 – Répartition des différents pas de temps sur le maillage.

Le pourcentage de cellules de chaque pas de temps est donné dans le tableau 5.5 ci-dessous.

Pas de temps	Δt	$\frac{\Delta t}{2}$	$\frac{\Delta t}{4}$	$\frac{\Delta t}{8}$
Nbre. relatif de cellules (%)	28	9	52	11

TABLE 5.5 – Pourcentage de cellules de chaque pas de temps.

En appliquant la formule (5.15) de la page 165 aux pourcentages de cellules fournis dans le tableau, **on obtient un speedup théorique d'environ 58% sur ce cas.**

5.3.3 Schémas d'intégration temporelle utilisés sur ce cas

Sur ce cas, notre schéma à pas de temps local NRK3 est comparé aux schéma $RK3_{dtglobal}$ et au schéma implicite utilisé avec le critère de convergence $\epsilon = 0.1$. On énonce ci-dessous le pas de temps utilisé par chaque schéma.

- Le pas de temps $\Delta t = 2.5 \times 10^{-4} \times d/U_\infty$ est utilisé pour le schéma $RK3_{dtglobal}$ (le nombre CFL maximal sur le maillage est $CFL = 0.95$).
- Comme évoqué précédemment, pour le schéma $NRK3$ le plus grand pas de temps sur le domaine de calcul vaut $\Delta t = 2.0 \times 10^{-3} \times d/U_\infty$ (il est huit fois plus grand que le pas de temps utilisé par le schéma $RK3_{dtglobal}$). La méthode de frontières immergées utilisée n'étant pas conservative, l'étape de correction de flux n'a pas été activée sur ce cas. En effet, il ne nous semble pas utile de rendre la méthode d'intégration temporelle conservative sachant que la méthode spatiale ne l'est pas.
- Enfin, le pas de temps utilisé par le schéma implicite est $\Delta t = 6.0 \times 10^{-3} \times d/U_\infty$. Avec ce pas de temps, le nombre CFL maximal sur le maillage est environ égal à 20.

A partir de la solution initiale ($t = 0$), les trois simulations sont réalisées jusqu'à l'instant $t^f = 80 \times d/U_\infty$. Des moyennes statistiques sont accumulées pour chaque simulation durant cet intervalle de temps. Le schéma AUSM+(P) senseur est utilisé pour le calcul des flux convectifs, et le schéma centré d'ordre deux est utilisé pour le calcul des flux visqueux.

5.3.4 Résultats des simulations

La figure 5.35 montre l'isosurface du critère $Q = 1 \times (U_\infty/d)^2$. Cette figure, obtenue à t^f avec notre schéma NRK3, confirme que l'écoulement devient turbulent en aval du cylindre. L'aspect turbulent de l'écoulement en aval du cylindre est également observable sur la figure 5.36, qui montre la norme de la vorticit  obtenue avec le schéma NRK3 à t^f . On rappelle que la vorticit  caract rise l'aspect tourbillonnaire d'un  coulement et s'écrit : $\nabla \wedge \vec{V}$, \vec{V}  tant le vecteur vitesse. Des all es de tourbillons de Von Karman sont observables en aval du cylindre. Ces tourbillons sont n anmoins rapidement dissip s   cause de l'augmentation du pas d'espace, en particulier   partir de $x/d = 6$. Nous avons enregistr  le signal de pression au niveau du point 1 (paroi sup rieure du cylindre) et au niveau du point 2 (dans le sillage du cylindre,   $x/d = 1.5$), en vue d'effectuer des analyses spectrales.

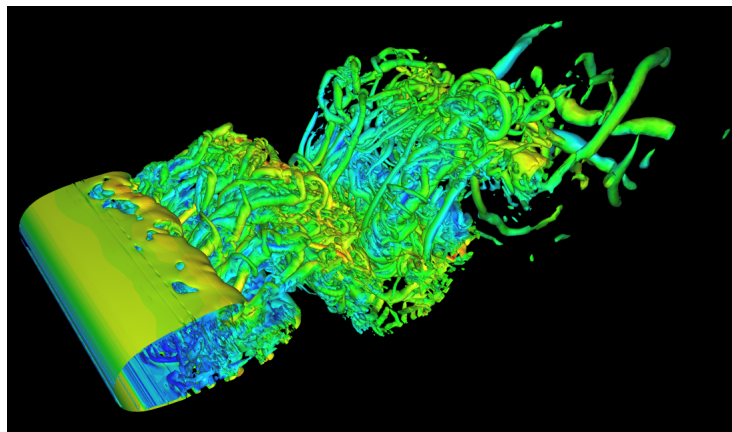


FIGURE 5.35 – Isosurface du crit re $Q = 1 \times (U_\infty/d)^2$.

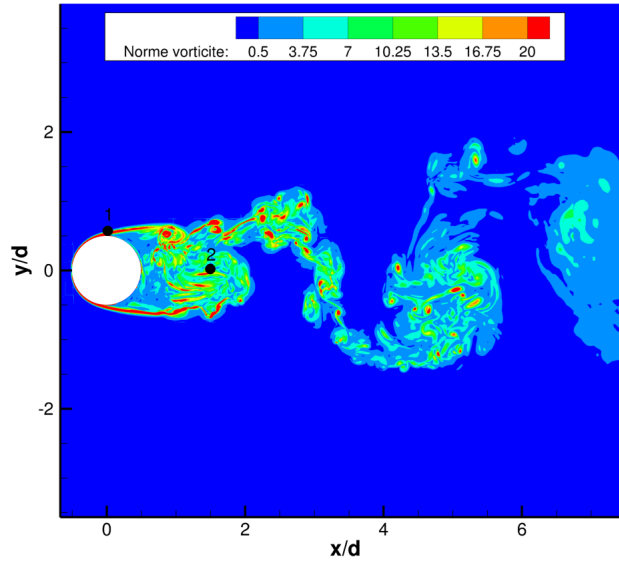


FIGURE 5.36 – Norme de la vorticité à t^f .

Les profils de vitesse moyenne dans les directions \vec{x} (u_1) et \vec{y} (u_2) sont respectivement montrés sur les figures 5.37 et 5.38. Sur chaque figure, l'image de gauche montre le profil de vitesse à $x/d = 1.06$, et l'image de droite montre le profil à $x/d = 2.02$. Les profils des tensions de Reynolds moyennes sont montrés sur les figures 5.39 et 5.40.

Sur la figure 5.37, des données expérimentales issues de [118] sont affichées. Les résultats numériques obtenus par le biais de la méthode de frontières immergées sont en bon accord avec les données expérimentales. De manière générale, on observe une bonne correspondance entre les solutions issues des trois schémas. On remarque néanmoins que les courbes obtenues avec le schéma NRK3 sont légèrement plus proches de la solution de référence (schéma $RK3_{dtglobal}$) que les courbes obtenues avec le schéma implicite. Ceci est notamment visible sur la vitesse moyenne dans la direction \vec{y} (figure 5.38) et sur les tensions de Reynolds (figures 5.39 et 5.40). On remarque également que les courbes obtenues avec notre schéma NRK3 ne sont pas "confondues" avec les courbes issues du schéma $RK3_{dtglobal}$, comme c'était le cas pour le profil SD7003.

La figure 5.41 montre les spectres de pression au niveau de la paroi du cylindre (image de gauche) et dans le sillage, à $x/d = 1.5$ (image de droite). Sur l'image de gauche, on observe un pic principal à $St = 0.2$ (la fréquence des lâchers tourbillonnaires au niveau du cylindre). Il s'agit d'un résultat classique pour ce cas-test [85]. Sur l'image de droite, un pic principal est observé à $St = 0.4$: la pression oscille à une fréquence égale à deux fois le nombre de Strouhal de l'écoulement. Ce résultat est également conforme aux observations faites dans [85].

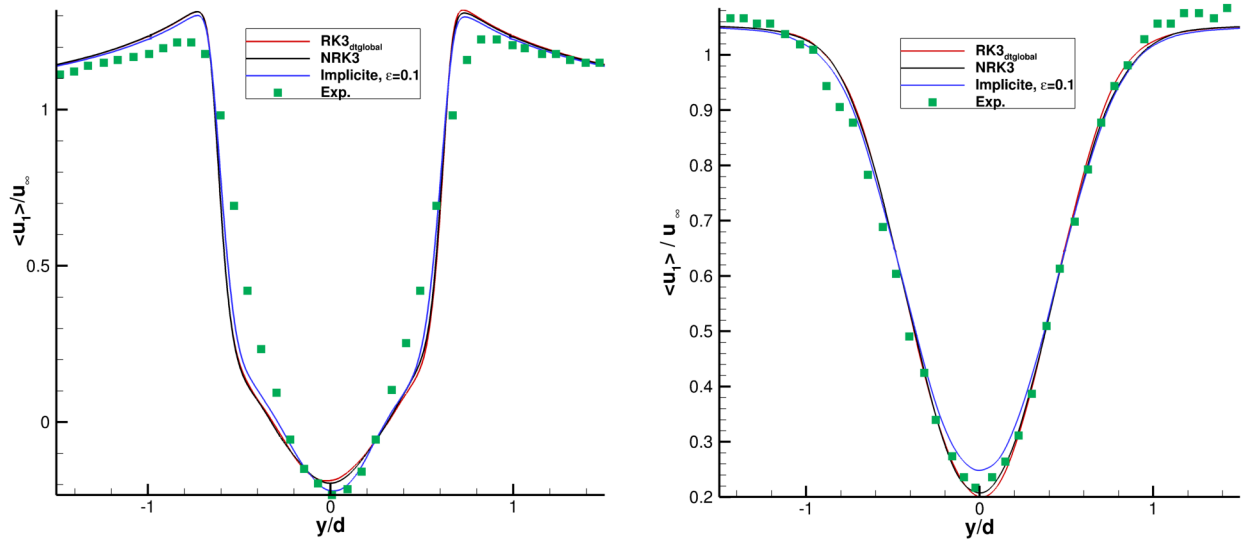


FIGURE 5.37 – Profil de la vitesse moyenne dans la direction \vec{x} à $x/d = 1.06$ (image de gauche) et $x/d = 2.02$ (image de droite).

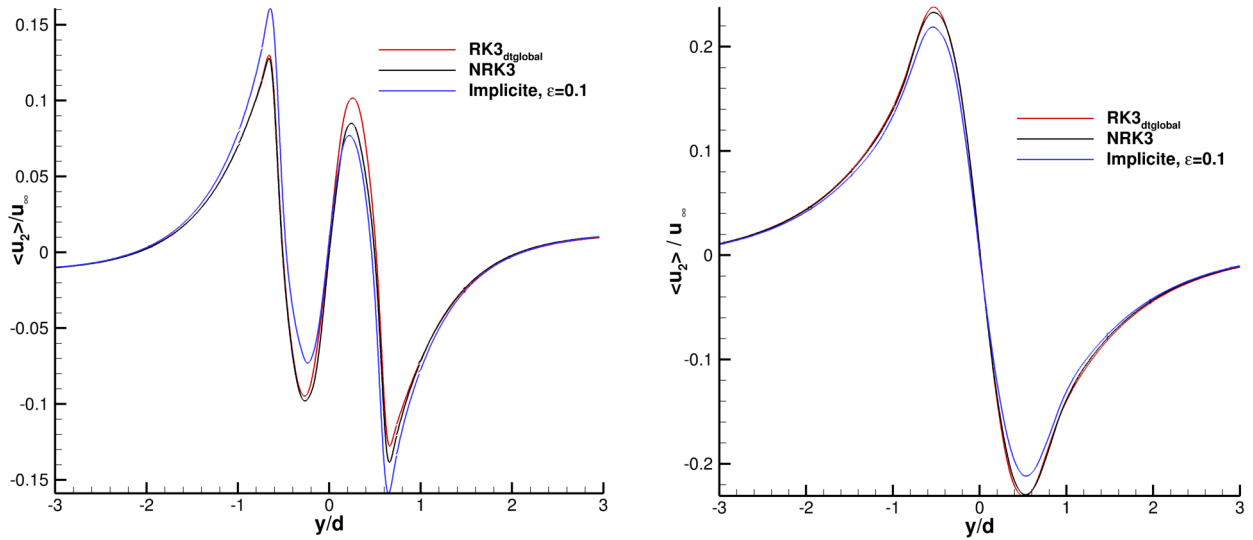


FIGURE 5.38 – Profil de la vitesse moyenne dans la direction \vec{y} à $x/d = 1.06$ (image de gauche) et $x/d = 2.02$ (image de droite).

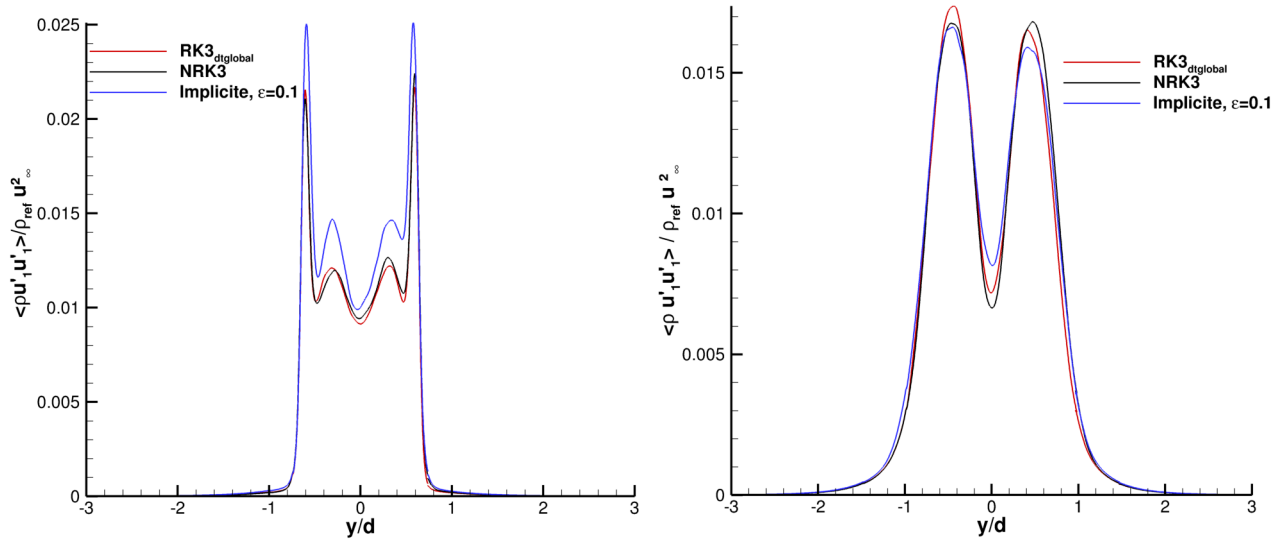


FIGURE 5.39 – Profil des tensions de Reynolds $\langle \rho u'_1 u'_1 \rangle$ à $x/d = 1.06$ (image de gauche) et $x/d = 2.02$ (image de droite).

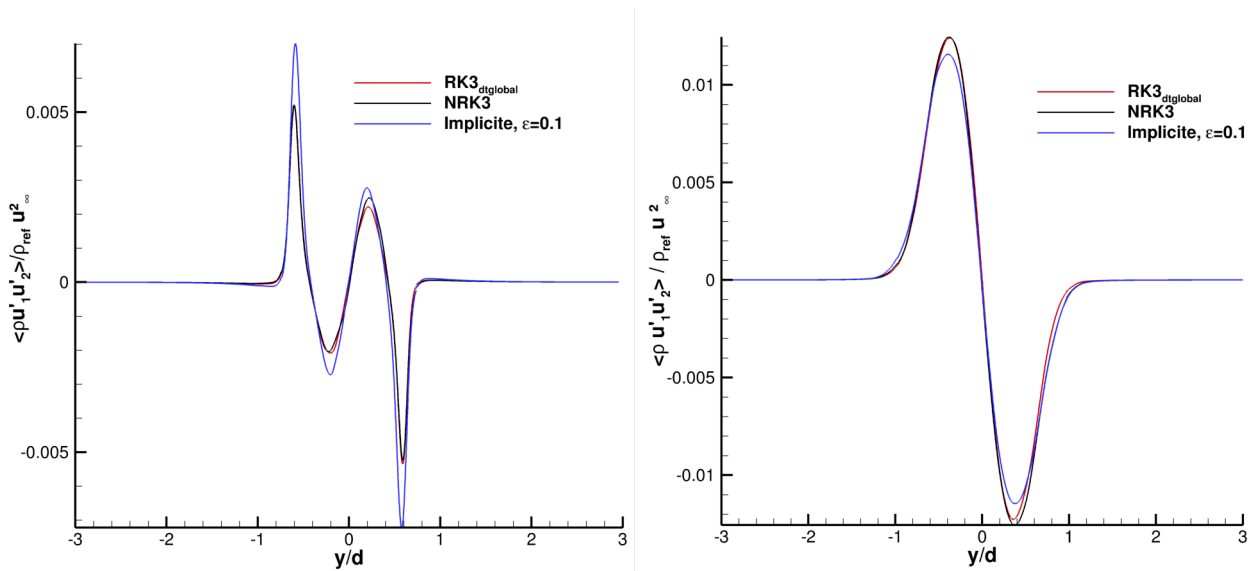


FIGURE 5.40 – Profil des tensions de Reynolds $\langle \rho u'_1 u'_2 \rangle$ à $x/d = 1.06$ (image de gauche) et $x/d = 2.02$ (image de droite).

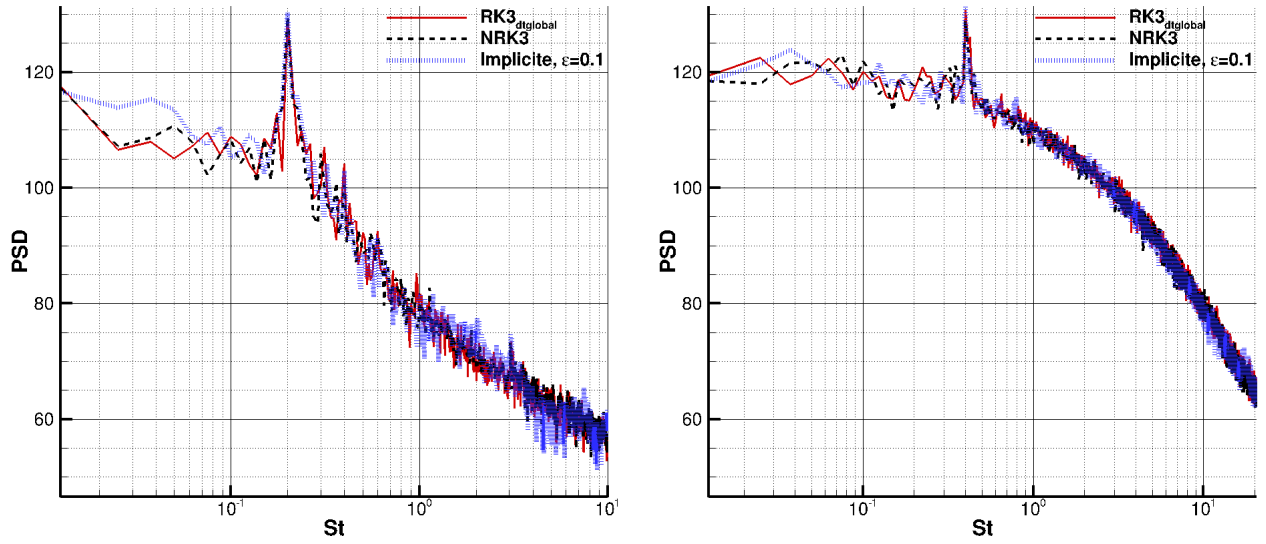


FIGURE 5.41 – Spectre du signal de pression : au niveau de la paroi du cylindre, c'est à dire au point d'enregistrement 1 (image de gauche), et à $x/d = 1.5$, c'est à dire au point d'enregistrement 2 (image de droite).

5.3.5 Comparaison des temps de calcul et bilan de ce cas-test

Le tableau 5.6 montre, pour chaque schéma, les temps de simulation et le speedup par rapport à $RK3_{dtglobal}$. On précise que les simulations ont été effectuées sur un noeud Intel Xeon(R) E5-2690 avec 24 processus MPI.

	$RK3_{dtglobal}$	NRK3	Implicite, $\epsilon = 0.1$
Temps CPU (heures)	180	88	104
Speedup par rapport à $RK3_{dtglobal}$ (en %)	0	51	42

TABLE 5.6 – Temps CPU et speedup par rapport au schéma $RK3_{dtglobal}$.

Le speedup obtenu avec notre schéma à pas de temps local NRK3 est d'environ 51%, qui est légèrement inférieur au speedup théorique de 58%. Sur ce cas, notre schéma NRK3 est le plus rapide puisqu'un speedup de "seulement" 42% est obtenu avec le schéma implicite.

Ce cas-test démontre que notre schéma à pas de temps local NRK3 est parfaitement utilisable avec une méthode de frontières immergées.

Conclusion et perspectives

La simulation numérique des phénomènes instationnaires nécessite l'utilisation d'une méthode d'intégration temporelle précise, rapide et compatible avec les techniques HPC. Nous avons montré dans le chapitre 1 qu'en général, les méthodes d'intégration temporelle implicites et explicites ne satisfont pas ces trois critères simultanément. Ainsi, au cours des dernières décennies, certains auteurs ont proposé de nouvelles méthodes d'intégration temporelle pour dépasser les limitations des méthodes implicites et explicites. Parmi ces nouvelles méthodes, les méthodes explicites à pas de temps local ont été créées pour remédier au principal défaut des méthodes explicites classiques (à pas de temps uniforme) : les temps de simulation élevés. Nous pensons que les méthodes explicites à pas de temps local peuvent véritablement remédier à la "lenteur" des méthodes explicites classiques tout en conservant leurs avantages (précision, compatibilité avec les techniques HPC), d'où notre choix de travailler sur ces méthodes. A l'issue de l'étude bibliographique concernant les schémas à pas de temps local, nous nous étions fixés l'objectif suivant : réaliser des simulations instationnaires sur des configurations de type industriel avec un schéma à pas de temps local.

Pour atteindre cet objectif, nous avons tout d'abord mis en place deux nouveaux schémas à pas de temps local. Inspirés par Constantinescu et Sandu [16], la théorie des méthodes de Runge-Kutta Partitionnées a été utilisée. Cette théorie fournit les conditions d'ordre qui doivent être satisfaites par les coefficients des méthodes de Runge-Kutta de pas de temps Δt (RK_S) et $\Delta t/2$ (RK_F) pour obtenir un schéma précis à un certain ordre p . Constantinescu et Sandu imposent l'égalité des coefficients b des méthodes RK_S et RK_F ($b_i^S = b_i^F$) pour satisfaire automatiquement les conditions d'ordre. Les schémas obtenus sont conservatifs, mais au plus d'ordre deux. Nous avons fait le choix d'imposer l'égalité des coefficients c ($c_i^S = c_i^F$) pour pouvoir mettre en place un schéma d'ordre supérieur à deux. Malheureusement, ce choix implique la perte d'égalité sur les coefficients b ($b_i^S \neq b_i^F$) et donc de la conservativité. Nous avons alors proposé une étape de correction de flux dans l'algorithme de nos schémas pour les rendre conservatifs. Nous avons commencé par proposer un nouveau schéma à pas de temps local d'ordre deux (NRK2). Ce schéma n'apporte pas de réelle plus-value par rapport aux schémas de la littérature puisqu'il possède par exemple les mêmes propriétés que le schéma proposé par Constantinescu et Sandu [16] (ordre deux, conservatif), ainsi qu'un coût algorithmique quasi-identique. Le schéma d'ordre trois que nous proposons (NRK3) apporte quelques améliorations par rapport aux schémas d'ordre trois de la littérature [72, 17]. Contrairement aux schémas proposés dans [72, 17], il est basé sur une méthode RK3 de type "low-storage" pour optimiser le coût

mémoire. Nous avons également montré qu'il avait, en théorie, un coût algorithmique inférieur à celui du schéma de Schlegel *et al* [17]. Faute de temps, cela n'a malheureusement pas pu être confirmé expérimentalement et pourrait faire l'objet d'une étude future.

Les schémas à pas de temps local proposés ont été comparés aux schémas de Constantinescu et Sandu sur trois cas-test académiques : l'advection 2D d'un tourbillon, le tube à choc de Sod et le tourbillon de Taylor-Green 3D (chapitre 4). Ces cas-test ont montré que les schémas proposés (ainsi que les schémas de Constantinescu et Sandu) fournissaient des solutions en très bon accord avec les solutions de référence. Ces cas-test ont également permis de confirmer l'ordre temporel théorique des schémas proposés ainsi que le bon fonctionnement de notre stratégie de correction de flux pour assurer la conservativité. Finalement, parmi les schémas testés, seul le schéma d'ordre trois que nous avons proposé (NRK3) a été retenu pour réaliser des simulations instationnaires sur des configurations plus complexes. En effet, ce schéma s'est révélé très performant et plus précis que les schémas de Constantinescu et Sandu (seulement d'ordre deux). Un effort d'implémentation a donc été réalisé sur ce schéma pour pouvoir l'utiliser sur des cas-test plus complexes.

Dans le chapitre 5, deux simulations LES sont réalisées à l'aide de notre schéma à pas de temps local NRK3 :

- un écoulement d'air autour du profil SD7003,
- un écoulement d'air autour d'un cylindre en conjonction avec une méthode de frontières immergées. Le rapport deux entre les différents pas de temps utilisés par notre schéma NRK3 est parfaitement adapté au maillage de type Octree utilisé sur ce cas. En effet, un rapport deux sur le pas d'espace est systématiquement utilisé pour raffiner (ou déraffiner) le maillage.

Ces simulation ont été réalisées sur des maillages d'environ 90 millions de points. Aussi, il était nécessaire de réaliser une implémentation OpenMP/MPI de notre schéma NRK3 pour obtenir des résultats dans un délai de temps raisonnable. L'implémentation MPI de notre schéma NRK3 s'est révélée satisfaisante : les speedup mesurés lors des test de scalabilité forte étaient assez proches des speedup obtenus avec le schéma RK3 à pas de temps global. De moins bons résultats ont été obtenus sur les test de scalabilité forte en OpenMP. Une étude plus poussée est nécessaire pour déterminer les raisons à cette dégradation de la scalabilité forte en OpenMP.

Concernant le cas d'écoulement d'air autour du profil SD7003, notre schéma à pas de temps local NRK3 s'est révélé aussi précis que le schéma RK3 à pas de temps global pour le calcul de différentes grandeurs liées à l'écoulement : vitesses moyennes, tensions de Reynolds moyennes, spectres de pression. De plus, une réduction du temps de calcul de 57% par rapport au schéma RK3 à pas de temps global a été obtenue. Le schéma implicite utilisé sur le même cas a permis d'obtenir une réduction de temps calcul similaire (65%). Néanmoins, le schéma implicite s'est révélé moins précis que notre schéma NRK3, en particulier sur le calcul des tensions de Reynolds moyennes.

Concernant l'écoulement d'air autour d'un cylindre réalisé en conjonction avec une méthode de frontières immergées, notre schéma NRK3 ainsi que le schéma implicite ont

fourni des solutions en bon accord avec les solutions issues du schéma RK3 à pas de temps global. Sur ce cas, notre schéma NRK3 s'est révélé être le schéma le plus rapide, avec une réduction du temps de calcul de 51% par rapport au schéma RK3 à pas de temps global. Une réduction du temps de calcul de "seulement" 42% a été obtenue avec le schéma implicite.

A travers ces deux simulations, nous avons montré qu'un schéma explicite à pas de temps local constitue une vraie alternative aux schémas explicites et implicites "classiques" pour réaliser des simulations instationnaires.

Malgré ces résultats prometteurs, certains aspects de notre schéma à pas de temps local NRK3 méritent d'être approfondis.

Au niveau algorithmique, il serait intéressant d'envisager une approche monozone plutôt qu'une approche multizone pour notre schéma. Comme expliqué dans la partie 3.3, la répartition des différents pas de temps sur le maillage est réalisée à l'aide d'une approche multizone : le maillage est découpé en zones auxquelles on attribue un pas de temps, et chaque zone est dotée de cellules fictives pour récupérer les informations contenues sur les zones adjacentes. Cette approche génère un nombre important de cellules fictives : de nombreuses opérations de copie de valeurs dans ces cellules sont donc effectuées. Toutes ces opérations de copie peuvent générer un surcoût en terme de temps de calcul. Avec une approche monozone, on s'affranchit du concept de zones munies de cellules fictives : on se contente d'utiliser des plages d'indices pour définir les régions d'application des différents pas de temps. Il y a moins de cellules fictives et donc moins d'opérations de copie dans ces cellules, ce qui permettrait de réduire le temps de calcul par rapport à une approche multizone.

Dans la partie 5.21, nous avons observé que la parallélisation MPI et OpenMP de notre schéma NRK3 était moins efficace que celle du schéma RK3 à pas de temps global. Bien qu'une analyse plus approfondie de cette perte d'efficacité soit nécessaire, nous pouvons d'ores et déjà proposer quelques pistes d'amélioration. Concernant la parallélisation MPI, il serait intéressant de repenser notre algorithme pour améliorer le recouvrement entre les calculs et les communications MPI. L'objectif serait de diminuer l'influence des communications MPI (qui ont tendance à dégrader la scalabilité) en faisant en sorte qu'un maximum de calculs soit effectué en même temps que ces communications. Le passage à une approche monozone pourrait également permettre d'améliorer l'efficacité de parallélisation MPI en réduisant le nombre de cellules fictives et donc le nombre de communications MPI. De plus, nous avons observé que sur certaines zones, le temps de calcul par cellule était un peu plus élevé que sur d'autres zones. Nous n'avons pas d'explication précise à ce phénomène, mais nous supposons que la taille et la forme des zones ont une influence sur ce temps de calcul par cellule. L'idée serait de mettre en place une stratégie d'équilibrage OpenMP qui prennent en compte ce temps de calcul par cellule variable, afin d'obtenir une répartition plus équilibrée de la charge de travail sur les threads.

Notre schéma NRK3 a été implémenté dans un contexte de maillages structurés. Une implémentation de notre schéma sur maillages non-structurés est nécessaire, notamment en prévision de simulations d'écoulement autour de géométries complexes (turbines...).

Enfin, ce travail de thèse pourrait par exemple déboucher sur une amélioration de la

méthode IMEX (IMplicite/EXplicite). Comme nous l'avons décrit dans la partie 1.6, cette méthode se compose d'un schéma implicite pour intégrer les cellules présentant une condition CFL très restrictive et d'un schéma explicite, utilisé sur les cellules où la condition CFL est plus lâche. L'idée serait de remplacer le schéma explicite par un schéma explicite à pas de temps local pour améliorer l'efficacité de la méthode IMEX.

Annexe A

Conditions d'ordre trois pour une méthode de Runge-Kutta partitionnée (RKP)

On considère le système différentielle suivant, constitué de deux équations différentielles scalaires :

$$\begin{cases} \frac{du_S}{dt} = -D_S(u_S, u_F), & u_S \in \mathbb{R} \\ \frac{du_F}{dt} = -D_F(u_S, u_F), & u_F \in \mathbb{R} \end{cases} \quad (\text{A.1})$$

Des équations scalaires sont ici considérées pour simplifier les calculs, mais sans perte de généralité. On suppose que la première équation de (A.1) est intégrée avec une méthode de Runge-Kutta RKS de coefficients (a_{ij}^S, b_i^S, c_i^S) et constituée de q étapes : $1 \leq i, j \leq q$. On suppose que la seconde équation de (A.1) est intégrée avec une méthode de Runge-Kutta RKF à q étapes de coefficients (a_{ij}^F, b_i^F, c_i^F) . On introduit également Δt , un pas de temps commun aux deux méthodes. Le vecteur u_S (resp. u_F) au temps t^n est notée $u_S^{n,1}$ (resp. $u_F^{n,1}$). La méthode RKP composée de RKS et RKF s'écrit :

$$u_S^{n+1} = u_S^{n,1} - \Delta t \sum_{i=1}^q b_i^S D_S^{n,i} \quad u_F^{n+1} = u_F^{n,1} - \Delta t \sum_{i=1}^q b_i^F D_F^{n,i}$$

$$D_S^{n,i} = D_S(u_S^{n,i}, u_F^{n,i}) \quad D_F^{n,i} = D_F(u_S^{n,i}, u_F^{n,i})$$

$$u_S^{n,i} = u_S^{n,1} - \Delta t \sum_{j=1}^{i-1} a_{ij}^S D_S^{n,j} \quad u_F^{n,i} = u_F^{n,1} - \Delta t \sum_{j=1}^{i-1} a_{ij}^F D_F^{n,j}$$

A.1 Développement de Taylor de u_S^{n+1}

On rappelle que l'expression de u_S^{n+1} est donnée par l'équation (A.2).

$$u_S^{n+1} = u_S^{n,1} - \Delta t \sum_{i=1}^q b_i^S D_S^{n,i} \quad (\text{A.2})$$

On Considère le terme $D_S^{n,i}$ de l'équation (A.2), qui s'écrit :

$$\begin{aligned} D_S^{n,i} &= D_S(u_S^{n,i}, u_F^{n,i}) \\ &= D_S\left(u_S^{n,1} - \Delta t \underbrace{\sum_{j=1}^{i-1} a_{ij}^S D_S^{n,j}}_{du_S}, u_F^{n,1} - \Delta t \underbrace{\sum_{j=1}^{i-1} a_{ij}^F D_F^{n,j}}_{du_F}\right) \end{aligned}$$

Les abréviations du_S et du_F sont introduites pour clarifier les expressions. Le développement de Taylor à l'ordre trois de $D_S^{n,i}$ autour de $(u_S^{n,1}, u_F^{n,1})$ s'écrit :

$$\begin{aligned} D_S^{n,i} &= D_S^{n,1} - \Delta t du_S \frac{\partial D_S^{n,1}}{\partial u_S} - \Delta t du_F \frac{\partial D_S^{n,1}}{\partial u_F} - \Delta t^2 du_S du_F \frac{\partial^2 D_S^{n,1}}{\partial u_S \partial u_F} \\ &\quad - \frac{\Delta t^2}{2} du_F^2 \frac{\partial^2 D_S^{n,1}}{\partial u_F^2} - \frac{\Delta t^2}{2} du_S^2 \frac{\partial^2 D_S^{n,1}}{\partial u_S^2} + \mathcal{O}(\Delta t^3) \end{aligned} \quad (\text{A.3})$$

Le travail n'est pas terminé, puisque le développement (A.3) contient encore des termes qui ne sont pas exprimés au temps t^n : il s'agit des termes $D_S^{n,j}$ et $D_F^{n,j}$ contenus dans du_S et du_F . On rappelle l'expression des termes $D_S^{n,j}$:

$$\begin{aligned} D_S^{n,j} &= D_S(u_S^{n,j}, u_F^{n,j}) \\ &= D_S\left(u_S^{n,1} - \Delta t \sum_{k=1}^{j-1} a_{jk}^S D_S^{n,k}, u_F^{n,1} - \Delta t \sum_{k=1}^{j-1} a_{jk}^F D_F^{n,k}\right) \end{aligned}$$

On effectue un développement de Taylor de $D_S^{n,j}$ à l'ordre deux autour de $(u_S^{n,1}, u_F^{n,1})$:

$$\begin{aligned} D_S^{n,j} &= D_S^{n,1} - \Delta t \left(\sum_{k=1}^{j-1} a_{jk}^S D_S^{n,k} \right) \frac{\partial D_S^{n,1}}{\partial u_S} - \Delta t \left(\sum_{k=1}^{j-1} a_{jk}^F D_F^{n,k} \right) \frac{\partial D_S^{n,1}}{\partial u_F} + \mathcal{O}(\Delta t^2) \\ &= D_S^{n,1} - \Delta t \left(\sum_{k=1}^{j-1} a_{jk}^S D_S^{n,1} \right) \frac{\partial D_S^{n,1}}{\partial u_S} - \Delta t \left(\sum_{k=1}^{j-1} a_{jk}^F D_F^{n,1} \right) \frac{\partial D_S^{n,1}}{\partial u_F} + \mathcal{O}(\Delta t^2), \end{aligned} \quad (\text{A.4})$$

vu que $D_S^{n,k} = D_S^{n,1} + \mathcal{O}(\Delta t)$.

Le même raisonnement est appliqué pour effectuer le développement de Taylor de $D_F^{n,j}$.

Dans l'équation (A.4), tous les termes sont exprimés au temps t^n . Les termes du_S et du_F s'écrivent donc :

$$du_S = \sum_{j=1}^{i-1} a_{ij}^S \left(D_S^{n,1} - \Delta t \sum_{k=1}^{j-1} a_{jk}^S D_S^{n,1} \frac{\partial D_S^{n,1}}{\partial u_S} - \Delta t \sum_{k=1}^{j-1} a_{jk}^F D_F^{n,1} \frac{\partial D_S^{n,1}}{\partial u_F} \right) + \mathcal{O}(\Delta t^2) \quad (\text{A.5})$$

$$du_F = \sum_{j=1}^{i-1} a_{ij}^F \left(D_F^{n,1} - \Delta t \sum_{k=1}^{j-1} a_{jk}^S D_S^{n,1} \frac{\partial D_F^{n,1}}{\partial u_S} - \Delta t \sum_{k=1}^{j-1} a_{jk}^F D_F^{n,1} \frac{\partial D_F^{n,1}}{\partial u_F} \right) + \mathcal{O}(\Delta t^2) \quad (\text{A.6})$$

Ce développement à l'ordre deux des termes du_S et du_F est suffisant puisque ces termes sont multipliés par Δt dans l'équation (A.3). Ainsi, un reste global $\mathcal{O}(\Delta t^3)$ est obtenu.

Tous ces développements permettent d'obtenir le développement de Taylor de u_S^{n+1} , qui apparaît dans l'équation (A.7).

$$\begin{aligned}
u_S^{n+1} &= u_S^{n,1} - \underbrace{\Delta t \sum_{i=1}^q b_i^S D_S^{n,1}}_{(1)} - \underbrace{\Delta t^2 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^S D_S^{n,1} \frac{\partial D_S^{n,1}}{\partial u_S}}_{(2)} \\
&\quad - \underbrace{\Delta t^2 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^F D_F^{n,1} \frac{\partial D_S^{n,1}}{\partial u_F}}_{(3)} \\
&\quad - \underbrace{\frac{\Delta t^3}{2} \sum_{i=1}^q b_i^S \left(\sum_{j=1}^{i-1} a_{ij}^S \right)^2 (D_S^{n,1})^2 \frac{\partial^2 D_S^{n,1}}{\partial u_S^2}}_{(4)} \\
&\quad - \underbrace{\Delta t^3 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^S \sum_{k=1}^{i-1} a_{ij}^F (D_S^{n,1})^2 \frac{\partial^2 D_S^{n,1}}{\partial u_S \partial u_F}}_{(5)} \\
&\quad - \underbrace{\Delta t^3 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^S \sum_{k=1}^{j-1} a_{jk}^S D_S^{n,1} \frac{\partial D_S^{n,1}}{\partial u_S} \frac{\partial D_S^{n,1}}{\partial u_S}}_{(6)} \\
&\quad - \underbrace{\Delta t^3 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^S \sum_{k=1}^{j-1} a_{jk}^F \frac{\partial D_S^{n,1}}{\partial u_S} \frac{\partial D_S^{n,1}}{\partial u_F} D_F^{n,1}}_{(7)} \\
&\quad - \underbrace{\frac{\Delta t^3}{2} \sum_{i=1}^q b_i^S \left(\sum_{j=1}^{i-1} a_{ij}^F \right)^2 (D_S^{n,1})^2 \frac{\partial^2 D_S^{n,1}}{\partial u_F^2}}_{(8)} \\
&\quad - \underbrace{\Delta t^3 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^F \sum_{k=1}^{j-1} a_{jk}^S \frac{\partial D_S^{n,1}}{\partial u_F} \frac{\partial D_F^{n,1}}{\partial u_S} D_S^{n,1}}_{(9)} \\
&\quad - \underbrace{\Delta t^3 \sum_{i=1}^q b_i^S \sum_{j=1}^{i-1} a_{ij}^F \sum_{k=1}^{j-1} a_{jk}^F \frac{\partial D_S^{n,1}}{\partial u_F} \frac{\partial D_F^{n,1}}{\partial u_F} D_F^{n,1}}_{(10)} \\
&\quad + \mathcal{O}(\Delta t^4)
\end{aligned} \tag{A.7}$$

A.2 Développement de Taylor de la solution exacte $u_S(t + \Delta t)$

En utilisant la formule de Taylor, on a :

$$u_S(t + \Delta t) = u_S(t) + \Delta t \frac{du_S}{dt}(t) + \frac{\Delta t^2}{2} \frac{d^2u_S}{dt^2}(t) + \frac{\Delta t^3}{6} \frac{d^3u_S}{dt^3}(t) + \mathcal{O}(\Delta t^4) \quad (\text{A.8})$$

En prenant en compte l'expression de la dérivée temporelle de u_S dans le système (A.1), on peut réécrire son développement de Taylor (équation (A.9)). Dans l'équation (A.9), les arguments des différentes fonctions ne sont pas affichés par soucis de lisibilité, mais chaque fonction est bien entendu évaluée au temps t^n .

$$\begin{aligned} u_S(t + \Delta t) = u_S(t) &- \underbrace{\Delta t D_S}_{(1)} - \underbrace{\frac{\Delta t^2}{2} \frac{\partial D_S}{\partial u_S} D_S}_{(2)} - \underbrace{\frac{\Delta t^2}{2} \frac{\partial D_S}{\partial u_F} D_F}_{(3)} \\ &- \underbrace{\frac{\Delta t^3}{6} \frac{\partial^2 D_S}{\partial u_S^2} (D_S)^2}_{(4)} - \underbrace{\frac{\Delta t^3}{3} \frac{\partial^2 D_S}{\partial u_S \partial u_F} D_S D_F}_{(5)} \\ &- \underbrace{\frac{\Delta t^3}{6} \frac{\partial^2 D_S}{\partial u_S} \frac{\partial D_S}{\partial u_S} D_S}_{(6)} - \underbrace{\frac{\Delta t^3}{6} \frac{\partial D_S}{\partial u_S} \frac{\partial D_S}{\partial u_F} D_F}_{(7)} \\ &- \underbrace{\frac{\Delta t^3}{6} \frac{\partial^2 D_S}{\partial u_F^2} (D_F)^2}_{(8)} - \underbrace{\frac{\Delta t^3}{6} \frac{\partial^2 D_S}{\partial u_F} \frac{\partial D_F}{\partial u_S} D_S}_{(9)} \\ &- \underbrace{\frac{\Delta t^3}{6} \frac{\partial D_S}{\partial u_F} \frac{\partial D_F}{\partial u_F} D_F}_{(10)} + \mathcal{O}(\Delta t^4) \end{aligned} \quad (\text{A.9})$$

A.3 Conditions d'ordre trois

Pour obtenir les conditions d'ordre trois, il faut faire correspondre le développement de Taylor de u_S^{n+1} (équation (A.7)) avec le développement de Taylor de $u_S(t + \Delta t)$ (équation (A.9)). Chaque terme (i) de l'équation (A.7) doit coïncider avec le même terme (i) de l'équation (A.9). Ceci mène aux conditions sur les coefficients des méthodes RK_S et RK_F , données par le système (A.11). Dans ce système, la relation de consistance des méthodes de Runge-Kutta, donnée dans l'équation (A.10), est utilisée.

$$c_i = \sum_{j=1}^{i-1} a_{ij} \quad (\text{A.10})$$

$$\left\{ \begin{array}{l} \text{correspondance de (1)} : \sum_{i=1}^q b_i^S = 1 \\ \text{correspondance de (2)} : \sum_{i=1}^q b_i^S c_i^S = \frac{1}{2} \\ \text{correspondance de (4)} : \sum_{i=1}^q b_i^S (c_i^S)^2 = \frac{1}{3} \\ \text{correspondance de (6)} : \sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^S = \frac{1}{6} \\ \text{correspondance de (3)} : \sum_{i=1}^q b_i^S c_i^F = \frac{1}{2} \\ \text{correspondance de (5)} : \sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S c_j^S c_j^F = \frac{1}{3} \\ \text{correspondance de (7)} : \sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^S c_j^F = \frac{1}{6} \\ \text{correspondance de (8)} : \sum_{i=1}^q b_i^S (c_i^F)^2 = \frac{1}{3} \\ \text{correspondance de (9)} : \sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^S = \frac{1}{6} \\ \text{correspondance de (10)} : \sum_{i=1}^q \sum_{j=1}^{i-1} b_i^S a_{ij}^F c_j^F = \frac{1}{6} \end{array} \right. \quad (\text{A.11})$$

Dans le système (A.11), les conditions d'ordre trois de la méthode RK_S apparaissent en noir. Les conditions écrites en rouge sont des conditions de couplage. Pour obtenir les conditions d'ordre trois de la méthode RK_F et les reste des conditions de couplage, il faut utiliser la même approche que précédemment. Il faut calculer le développement de Taylor de u_F^{n+1} et le faire coïncider avec le développement de Taylor de $u_F(t + \Delta t)$. Ces développements, très similaires à ceux qui viennent d'être effectués dans cette annexe, sont laissés au lecteur.

Annexe B

Tableau de Butcher représentant deux applications successives d'une méthode de pas de temps $\Delta t/2$

On considère une méthode de base RK_B décrite par son tableau de Butcher B.1.

c	A	0	0
\cdot	\cdot	\cdot	\cdot
c_i	$a_{i1} \quad \cdot \quad a_{i,i-1}$	c_i	$a_{i1} \quad \cdot \quad a_{i,i-1}$
\cdot	\cdot	\cdot	\cdot
c_q	$a_{q1} \quad \cdot \quad \cdot \quad a_{q,q-1}$	c_q	$a_{q1} \quad \cdot \quad \cdot \quad a_{q,q-1}$
			$b_1 \quad \cdot \quad \cdot \quad \cdot \quad b_q$

TABLE B.1 – Tableau de Butcher de la méthode de base RK_B . A gauche : écriture matricielle faisant apparaître la matrice A des coefficients a_{ij} et les vecteurs b^T et c contenant les coefficients b_i et c_i . A droite : écriture classique avec les différents coefficients de la méthode.

On considère la méthode RK_F construite à partir de la méthode de base RK_B , en suivant la stratégie de Constantinescu et Sandu. Le tableau de Butcher de la méthode RK_F écrit avec les notations matricielles est le tableau B.2. Le tableau de Butcher de la méthode RK_F écrit en notations classiques (avec les coefficients) est le tableau B.2.

$$c^F \left\{ \begin{array}{c|cc} & \overbrace{A^F} & \\ \hline c/2 & A/2 & \\ (1+c)/2 & b^T/2 & A/2 \\ \hline & \underbrace{b^T/2} & \underbrace{b^T/2} \\ & b^{FT} & \end{array} \right.$$

TABLE B.2 – Tableau de Butcher de la méthode RK_F en notations matricielles

0	0								
.									
$\frac{c_i}{2}$	$\frac{a_{i1}}{2}$.	$\frac{a_{i,i-1}}{2}$						
.									
c_q	a_{q1}	.	.	$a_{q,q-1}$					
$\frac{1}{2}$	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	0			
.									
$\frac{1+c_i}{2}$	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	$\frac{a_{i1}}{2}$.	$\frac{a_{i,i-1}}{2}$	
.									
c_q	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	a_{q1}	.	.	$a_{q,q-1}$
	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	$\frac{b_1}{2}$.	.	$\frac{b_q}{2}$

TABLE B.3 – Tableau de Butcher de la méthode RK_F en notations classiques (avec les coefficients)

Proposition B.1. *La méthode RK_F définie par le tableau de Butcher B.3 décrit deux application successives de la méthode de base RK_B (définie par le tableau de Butcher B.1) de pas de temps $\Delta t/2$.*

Démonstration. Pour cette démonstration, on considère l'équation différentielle (B.1).

$$\frac{dU}{dt} = -D(U) \tag{B.1}$$

Le tableau de Butcher de la méthode RK_F est rappelé sur le tableau B.4. Ce tableau est divisé en deux partie : une première partie constituée des coefficients rouges et une deuxième partie constituée des coefficients bleus.

0	0								
.									
$\frac{c_i}{2}$	$\frac{a_{i1}}{2}$.	$\frac{a_{i,i-1}}{2}$						
.									
c_q	a_{q1}	.	.	$a_{q,q-1}$					
$\frac{1}{2}$	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	0			
.									
$\frac{1+c_i}{2}$	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	$\frac{a_{i1}}{2}$.	$\frac{a_{i,i-1}}{2}$	
.									
c_q	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	a_{q1}	.	.	$a_{q,q-1}$
	$\frac{b_1}{2}$.	.	.	$\frac{b_q}{2}$	$\frac{b_1}{2}$.	.	$\frac{b_q}{2}$

TABLE B.4 – Tableau de Butcher de la méthode RK_F divisé en deux parties

Trivialement, la première partie du tableau de Butcher B.4 (coefficients rouges) décrit la première application de la méthode de base de pas de temps $\Delta t/2$:

1ere appli. de la méth. RK_B de pas $\Delta t/2$	Première partie du tableau B.4
$\forall i \leq q, U^{n,i} = U^{n,1} - \frac{\Delta t}{2} \sum_{j=1}^{i-1} a_{ij} D(U^{n,j})$	$\forall i \leq q, U^{n,i} = U^{n,1} - \Delta t \sum_{j=1}^{i-1} \frac{a_{ij}}{2} D(U^{n,j})$
$U^{n+\frac{1}{2}} = U^{n,1} - \frac{\Delta t}{2} \sum_{j=1}^q b_j D(U^{n,j})$	$U^{n,q+1} = U^{n,1} - \Delta t \sum_{j=1}^q \frac{b_j}{2} D(U^{n,j})$

Donc $U^{n+\frac{1}{2}} = U^{n,q+1}$. On pose $U^{n+\frac{1}{2},1} = U^{n+\frac{1}{2}}$ et on passe à la seconde application de la méthode de base RK_B de pas de temps $\Delta t/2$, et à la seconde partie du tableau de Butcher B.4 (coefficients bleus).

Montrons la propriété (P) suivante : $\forall i \leq q, U^{n+\frac{1}{2},i} = U^{n,q+i}$

Cette démonstration se fait par récurrence forte sur i .

Initialisation : La propriété (P) est vraie au rang $i = 1$ car $U^{n+\frac{1}{2},1} = U^{n,q+1}$.

Hérédité : Soit $i < q$. On suppose que pour tout $j \leq i : U^{n+\frac{1}{2},j} = U^{n,q+j}$. Démontrons la propriété (P) au rang $i + 1$.

La solution $U^{n,q+i+1}$ s'écrit :

$$\begin{aligned}
U^{n,q+i+1} &= U^{n,1} - \Delta t \sum_{j=1}^q \frac{b_j}{2} D(U^{n,j}) - \Delta t \sum_{j=1}^i \frac{a_{ij}}{2} D(U^{n,q+j}) \\
&= U^{n,q+1} - \Delta t \sum_{j=1}^i \frac{a_{ij}}{2} D(U^{n,q+j})
\end{aligned} \tag{B.2}$$

La solution $U^{n+\frac{1}{2},i+1}$ s'écrit :

$$U^{n+\frac{1}{2},i+1} = U^{n+\frac{1}{2},1} - \frac{\Delta t}{2} \sum_{j=1}^i a_{ij} D(U^{n+\frac{1}{2},j}) \tag{B.3}$$

L'hypothèse d'hérédité ainsi que l'égalité $U^{n,q+1} = U^{n+\frac{1}{2},1}$ assurent que $U^{n+\frac{1}{2},i+1} = U^{n,q+i+1}$. Par conséquent, la propriété (P) est vraie : $\forall i \leq q, U^{n+\frac{1}{2},i} = U^{n,q+i}$.

La solution U^{n+1} obtenue avec la seconde partie de la méthode RK_F s'écrit :

$$\begin{aligned}
U^{n+1} &= U^{n,1} - \Delta t \sum_{j=1}^q \frac{b_j}{2} D(U^{n,j}) - \Delta t \sum_{j=1}^q \frac{b_j}{2} D(U^{n,q+j}) \\
&= U^{n,q+1} - \Delta t \sum_{j=1}^q \frac{b_j}{2} D(U^{n,q+j})
\end{aligned} \tag{B.4}$$

La solution U^{n+1} obtenue avec la seconde application de la méthode RK3 de pas de temps $\Delta t/2$ s'écrit :

$$U^{n+1} = U^{n+\frac{1}{2},1} - \frac{\Delta t}{2} \sum_{j=1}^q b_j D(U^{n+\frac{1}{2},j}) \tag{B.5}$$

D'après la propriété (P) et l'égalité $U^{n+\frac{1}{2},1} = U^{n,q+1}$, les équations (B.4) et (B.5) sont égales. Par conséquent, la solution U^{n+1} obtenue avec la seconde partie de la méthode RK_F est égale à solution U^{n+1} obtenue avec la seconde application de la méthode RK_B de pas de temps $\Delta t/2$.

En conclusion, la méthode RK_F décrit bien deux applications successives de la méthode RK_B de pas de temps $\Delta t/2$. \square

Annexe C

Les différentes étapes du schéma NRK3 écrit sous forme "Low storage"

Le tableau figure à la page suivante.

Sous-domaine Slow		Sous-domaine Fast	
$\text{RK3} (\Delta t) : 1 \leq i \leq i_0 - 1$ $K_i^{n,1} = D_i^{n,1}$	$\text{RK}_S : i = i_0$ $K_i^{n,1} = D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \alpha_1^S \Delta t D_i^{n,1}$ $u_i^{n,3} = u_i^{n,1} - \alpha_2^S \Delta t D_i^{n,1}$	$\text{RK}_F : i = i_0 + 1$ $K_i^{n,1} = D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \Delta t \frac{\beta_1}{2} K_i^{n,1}$ $K_i^{n,2} = \gamma_2 K_i^{n,1} + D_i^{n,2}$ $u_i^{n,3} = u_i^{n,2} - \Delta t \frac{\beta_2}{2} K_i^{n,2}$ $K_i^{n,3} = \gamma_3 K_i^{n,2} + D_i^{n,3}$	$\text{RK3} (\Delta t/2) : i_0 + 2 \leq i \leq N$ $K_i^{n,1} = D_i^{n,1}$ $u_i^{n,2} = u_i^{n,1} - \Delta t \frac{\beta_1}{2} K_i^{n,1}$ $K_i^{n,2} = \gamma_2 K_i^{n,1} + D_i^{n,2}$ $u_i^{n,3} = u_i^{n,2} - \Delta t \frac{\beta_2}{2} K_i^{n,2}$ $K_i^{n,3} = \gamma_3 K_i^{n,2} + D_i^{n,3}$ $u_i^{n,4} = u_i^{n,3} - \Delta t \frac{\beta_3}{2} K_i^{n,3}$ $K_i^{n,4} = D_i^{n,4}$
$u_i^{n,2} = u_i^{n,1} - \Delta t \frac{\beta_1}{2} K_i^{n,1}$ $K_i^{n,2} = \gamma_2 K_i^{n,1} + D_i^{n,2}$	$u_i^{n,4} = u_i^{n,1} - \Delta t \frac{\beta_1}{2} K_i^{n,1}$ $K_i^{n,4} = \gamma_2 K_i^{n,1} + D_i^{n,4}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{\beta_1}{2} K_i^{n,4}$ $K_i^{n,6} = \gamma_2 K_i^{n,4} + D_i^{n,5}$	$u_i^{n,5} = u_i^{n,4} - \Delta t \frac{\beta_1}{2} K_i^{n,4}$ $K_i^{n,5} = \gamma_2 K_i^{n,4} + D_i^{n,5}$
$u_i^{n,3} = u_i^{n,2} - \Delta t \frac{\beta_2}{2} K_i^{n,2}$ $K_i^{n,3} = \gamma_3 K_i^{n,2} + D_i^{n,3}$	$u_i^{n,5} = u_i^{n,1} - \alpha_4^S \Delta t D_{i_0}^{n,1}$	$u_i^{n,6} = u_i^{n,1} - \Delta t (\alpha_1^S D_i^{n,1} + \alpha_2^S D_i^{n,2})$	
$u_i^{n,4} = u_i^{n,1} - \Delta t \frac{\beta_1}{2} K_i^{n,1}$ $K_i^{n,4} = \gamma_2 K_i^{n,1} + D_i^{n,4}$	$u_i^{n,6} = u_i^{n,4} - \Delta t \frac{\beta_2}{2} K_i^{n,5}$ $K_i^{n,6} = \gamma_3 K_i^{n,4} + D_i^{n,6}$	$u_i^{n,6} = u_i^{n,5} - \Delta t \frac{\beta_2}{2} K_i^{n,5}$ $K_i^{n,7} = \gamma_3 K_i^{n,6} + D_i^{n,7}$	$u_i^{n,7} = u_i^{n,5} - \Delta t \frac{\beta_2}{2} K_i^{n,6}$ $K_i^{n,6} = \gamma_3 K_i^{n,5} + D_i^{n,6}$
$u_i^{n,5} = u_i^{n,1} - \alpha_4^S \Delta t D_{i_0}^{n,1}$	$u_i^{n,7} = u_i^{n,1} - \Delta t \alpha_8^S D_{i_0}^{n,1}$	$u_i^{n+1} = u_i^{n,7} - \Delta t \frac{\beta_3}{2} K_i^{n,7}$ $K_i^{n+1} = D_i^{n+1}$	$u_i^{n+1} = u_i^{n,6} - \Delta t \frac{\beta_3}{2} K_i^{n,6}$ $K_i^{n+1} = D_i^{n+1}$
$u_i^{n,6} = u_i^{n,2} - \Delta t \frac{\beta_2}{2} K_i^{n,2}$ $K_i^{n,3} = \gamma_3 K_i^{n,2} + D_i^{n,3}$	$u_i^{n+1} = u_i^{n,6} - \Delta t \frac{\beta_3}{2} K_i^{n,6}$ $K_i^{n+1} = D_i^{n+1}$	$u_i^{n+1} = u_i^{n,7} - \Delta t \frac{\beta_3}{2} K_i^{n,7}$ $K_i^{n+1} = D_i^{n+1}$	
$u_i^{n,7} = u_i^{n,3} - \Delta t \frac{\beta_3}{2} K_i^{n,3}$ $K_i^{n+1} = D_i^{n+1}$	$u_i^{n+1} = u_i^{n,6} - \Delta t \frac{\beta_3}{2} K_i^{n,6}$ $K_i^{n+1} = D_i^{n+1}$	$u_i^{n+1} = u_i^{n,7} - \Delta t \frac{\beta_3}{2} K_i^{n,7}$ $K_i^{n+1} = D_i^{n+1}$	
	Correction de flux		

TABLE C.1 – Les différentes étapes du schéma NRRK3 écrites en utilisant la forme "low storage" de la méthode RK3

Bibliographie

- [1] Xiaoyan Ma, Yufu Guo, Guangyu Shi, and Yongqiang Yu. Numerical simulation of global temperature change during the 20th century with the iap/lasg goals model. *Advances in Atmospheric Sciences*, 21(2) :227–235, Apr 2004.
- [2] Stein Bondevik, Finn Løvholt, Carl Harbitz, Jan Mangerud, Alastair Dawson, and John Inge Svendsen. The storegga slide tsunami—comparing field observations with numerical simulations. *Marine and Petroleum Geology*, 22(1) :195 – 208, 2005.
- [3] Glatzmaier and Roberts. Rotation and magnetism of earth’s inner core. *Science*, 274 5294 :1887–91, 1996.
- [4] N. Alferez. *Simulation des grandes échelles du processus de décrochage par éclatement de bulbe de décollement laminaire*. PhD thesis, 2014.
- [5] L. Sanders, E. Manoha, S. Ben Khelil, and C. Francois. Lagoon : Cfd/caa coupling for landing gear noise and comparison with experimental database. *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*.
- [6] P. Sagaut. *Large-eddy simulation for incompressible flows*. 2001.
- [7] G. Pont. *Self adaptive turbulence models for unsteady compressible flows*. PhD thesis, 2015.
- [8] F. Bassi, A. Crivellini, S. Rebay, and M. Savini. Discontinuous galerkin solution of the reynolds-averaged navier–stokes and k– turbulence model equations. *Computers Fluids*, 34(4) :507 – 540, 2005. Residual Distribution Schemes, Discontinuous Galerkin Schemes and Adaptation.
- [9] P.-O. Persson, J. Bonet, and J. Peraire. Discontinuous galerkin solution of the navier–stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17) :1585 – 1595, 2009.
- [10] J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. Cfd vision 2030 study : A path to revolutionary computational aerosciences. *NASA report*, 03 2014.
- [11] Laurent Muscat, Guillaume Puigt, Marc Montagnac, and Pierre Brenner. A coupled implicit-explicit time integration method for compressible unsteady flows. *Journal of Computational Physics*, 398 :108883, 2019.
- [12] Per-Olof Persson. *High-Order LES Simulations using Implicit-Explicit Runge-Kutta Schemes*.
- [13] B. C. Vermeire and S. Nadarajah. Adaptive imex schemes for high-order unstructured methods. *Journal of Computational Physics*, 280 :261 – 286, 2015.

- [14] M. Shoeybi, M. Svärd, F. E. Ham, and P. Moin. An adaptive implicit–explicit scheme for the dns and les of compressible flows on unstructured grids. *Journal of Computational Physics*, 229(17) :5944 – 5965, 2010.
- [15] M. Shoeybi, M. Svärd, F. E. Ham, and P. Moin. An adaptive implicit–explicit scheme for the dns and les of compressible flows on unstructured grids. *Journal of Computational Physics*, 229(17) :5944 – 5965, 2010.
- [16] E. M. Constantinescu and A. Sandu. Multirate timestepping methods for hyperbolic conservation laws. *Journal of Scientific Computing*, 33(3) :239–278, Dec 2007.
- [17] M. Schlegel, O. Knoth, M. Arnold, and R. Wolke. Multirate runge–kutta schemes for advection equations. *Journal of Computational and Applied Mathematics*, 226(2) :345 – 357, 2009. Special Issue : Large scale scientific computations.
- [18] Y.A. Omelchenko and H. Karimabadi. Self-adaptive time integration of flux-conservative equations with sources. *Journal of Computational Physics*, 216(1) :179 – 194, 2006.
- [19] T. Unfer. An asynchronous framework for the simulation of the plasma/flow interaction. *Journal of Computational Physics*, 236 :229 – 246, 2013.
- [20] C. Dawson and R. Kirby. High resolution schemes for conservation laws with locally varying time steps. *SIAM J. Sci. Comput.*, 22(6) :2256–2281, June 2000.
- [21] H. Tang and G. Warnecke. A class of high resolution schemes for hyperbolic conservation laws and convection-diffusion equations with varying time and space grids. *Journal of Scientific Computing*, 24(2) :121–140, 2005.
- [22] A. Sandu and E. Constantinescu. Multirate explicit adams methods for time integration of conservation laws. *Journal of Scientific Computing*, 38(2) :229 – 249, 2009.
- [23] Z. Tan, Z. Zhang, Y. Huang, and T. Tang. Moving mesh methods with locally varying time steps. *Journal of Computational Physics*, 200(1) :347 – 367, 2004.
- [24] B. Seny, J. Lambrechts, R. Comblen, V. Legat, and J.-F. Remacle. Multirate time stepping for accelerating explicit discontinuous galerkin computations with application to geophysical flows. *International Journal For Numerical Mehtods In Fluids*, 71(41) :41–64, 2007.
- [25] E. Itam. *Simulation numérique d’écoulements autour de corps non profilés par des modèles de turbulence hybrides et un schéma multirate*. PhD thesis, 2017.
- [26] N. Takemitsu. Finite difference method to solve incompressible fluid flow. *Journal of Computational Physics*, 61(3) :499 – 518, 1985.
- [27] M. Li, T. Tang, and B. Fornberg. A compact fourth-order finite difference scheme for the steady incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 20(10) :1137–1151, 1995.
- [28] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.
- [29] J. Padet. *Fluides en Ecoulement - Méthodes et Modèles. 2 Edition*. 2011.

- [30] Parviz Moin and Krishnan Mahesh. Direct numerical simulation : A tool in turbulence research. *Annual Review of Fluid Mechanics*, 30(1) :539–578, 1998.
- [31] R. Manceau. *Modélisation statistique de la turbulence. 10ème école de Printemps mécanique des fluides numérique du CNRS, Roscoff, France.*
- [32] E.Hairer, G.Wanner, and S.P. Norsett. *Solving ordinary differential equations I : Nonstiff Problems.* Springer, Berlin, 1993.
- [33] S. Bayyuk, K. Powell, and B. VA. *A simulation technique for 2-D unsteady inviscid flows around arbitrarily moving and deforming bodies of arbitrary geometry.*
- [34] P.S. Lowery and W. C. Reynolds. Numerical simulation of a spatially-developping, forced, plane mixing layer. *Report of the Stanford University, TF26*, 1986.
- [35] J.H Williamson. Low-storage runge-kutta schemes. *Journal of Computational Physics*, 35(1) :48 – 56, 1980.
- [36] I. Bennaceur, D.C. Mincu, I. Mary, M. Terracol, L. Larchevêque, and P. Dupont. Numerical simulation of acoustic scattering by a plane turbulent shear layer : Spectral broadening study. *Computers Fluids*, 138 :83 – 98, 2016.
- [37] I. Bennaceur. *Etude numérique de la diffusion d'une onde acoustique par une couche de cisaillement turbulente à l'aide d'une simulation aux grandes échelles.* PhD thesis, 2017.
- [38] D. Drikakis, M. Hahn, A. Mosedale, and B. Thornber. Large eddy simulation using high-resolution and high-order methods. *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 367(1899) :2985–2997, 2009.
- [39] M. Pino Martín and G. V. Candler. A parallel implicit method for the direct numerical simulation of wall-bounded compressible turbulence. *Journal of Computational Physics*, 215(1) :153 – 171, 2006.
- [40] H. Kawamura, K. Ohsaka, H. Abe, and K. Yamamoto. Dns of turbulent heat transfer in channel flow with low to medium-high prandtl number fluid. *International Journal of Heat and Fluid Flow*, 19(5) :482 – 491, 1998.
- [41] H. Kawamura, H. Abe, and Y. Matsuo. Dns of turbulent heat transfer in channel flow with respect to reynolds and prandtl number effects. *International Journal of Heat and Fluid Flow*, 20(3) :196 – 207, 1999.
- [42] X. Zhou, K. H. Luo, and J. J.R Williams. Large-eddy simulation of a turbulent forced plume. *European Journal of Mechanics - B/Fluids*, 20(2) :233 – 254, 2001.
- [43] W. Hundsdorfer and J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations.* Springer, 2003.
- [44] C. Hirsch. *Numerical computation of internal and external flow.* J. Wiley and Sons, 1988.
- [45] Hester Bijl, Mark H. Carpenter, Veer N. Vatsa, and Christopher A. Kennedy. Implicit time integration schemes for the unsteady compressible navier–stokes equations : Laminar flow. *Journal of Computational Physics*, 179(1) :313 – 329, 2002.

- [46] R. Van Buuren, H. Kuerten, and B.J. Geurts. Implicit time accurate simulation of unsteady flow. *International Journal for Numerical Methods in Fluids*, 35(6) :687–720, 2001.
- [47] C.Weber and F.Ducros. Large eddy and reynolds-averaged navier - stokes simulations of turbulent flow over an airfoil. *International Journal of Computational Fluid Dynamics - INT J COMPUT FLUID DYNAMICS*, 13 :327–355, 2000.
- [48] M. Crouzeix. *Sur l'approximation des équations différentielles operationnelles linéaires par des méthodes de Runge-Kutta*. PhD thesis, 1975.
- [49] J.C. Butcher. Implicit runge-kutta processes. *Mathematics of Computation*, 18(85) :50 – 64, 1964.
- [50] C.A. Kennedy and M.H. Carpenter. *Diagonally implicit Runge-Kutta methods for ordinary differential equations, a review*.
- [51] H. Bijl. *Iterative methods for unsteady flow computations using implicit Runge-Kutta integration schemes*.
- [52] Ray Hixon and Miguel Visbal. *Comparison of High-Order Implicit Time Marching Schemes for Unsteady Flow Calculations*.
- [53] L. Wang and D.J. Mavriplis. Implicit solution of the unsteady euler equations for high-order accurate discontinuous galerkin discretizations. *J. Comput. Phys.*, 225(2) :1994–2015, August 2007.
- [54] C. W. Gear. *Numerical initial value problems in ordinary differential equations*. Prentice Hall PTR, Upper Saddle River, New Jersey, USA, 1971.
- [55] F. Daude. *Méthode d'intégration temporelle implicite pour la Simulation des Grandes Échelles : application à la réduction du bruit de cavité*. PhD thesis, 2007.
- [56] M. Rai, T. Gatski, and G. Erlebacher. Direct simulation of spatially evolving compressible turbulent boundary layers. *AIAA Paper 95-0583*, 1995.
- [57] Man Mohan Rai and Parviz Moin. Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer. *Journal of Computational Physics*, 109(2) :169 – 192, 1993.
- [58] D. P. Rizzetta and M. R. Visbal. Large-eddy simulation of supersonic boundary-layer flow by a high-order method. *International Journal of Computational Fluid Dynamics*, 18(1) :15–27, 2004.
- [59] D. P. Rizzetta, M. R. Visbal, and G. A. Blaisdell. A time-implicit high-order compact differencing and filtering scheme for large-eddy simulation. *International Journal for Numerical Methods in Fluids*, 42(6) :665–693, 2003.
- [60] A. Jameson and S.Yoon. Lower-upper implicit schemes with multiple grids for the euler equations. *AIAA Journal*, 25(7) :929–935, 1987.
- [61] F. Daude, I. Mary, and P. Comte. Self-adaptive newton-based iteration strategy for the les of turbulent multi-scale flows. *Computers and Fluids*, 100 :278 – 290, 2014.
- [62] N. Alkishriwi, M. Meinke, and W. Schröder. A large-eddy simulation method for low mach number flows using preconditioning and multigrid. *Computers Fluids*, 35(10) :1126 – 1136, 2006.

- [63] H. Dong and X. Zhong. Aiaa 2000-0275 a parallel high-order implicit algorithm for compressible navier-stokes equations. 2000.
- [64] Haecheon Choi and Parviz Moin. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics*, 113(1) :1 – 4, 1994.
- [65] Bamdad Lessani, Jan Ramboer, and Chris Lacor. Efficient large-eddy simulations of low mach number flows using preconditioning and multigrid. *International Journal of Computational Fluid Dynamics*, 18(3) :221–233, 2004.
- [66] M. J. Wright, G. V. Candler, and M. Prampolini. Data-parallel lower-upper relaxation method for the navier-stokes equations. *AIAA Journal*, 34(7) :1371–1377, 1996.
- [67] M. H. Carpenter, C. A. Kennedy, Hester Bijl, S. A. Viken, and Veer N. Vatsa. Fourth-order runge–kutta schemes for fluid mechanics applications. *Journal of Scientific Computing*, 25(1) :157–194, Oct 2005.
- [68] C. Völcker, J. B. J. P. G. Thomsen, and E. H. Stenby. Adaptive stepsize control in implicit runge-kutta methods for reservoir simulation. *IFAC Proceedings Volumes*, 43(5) :523 – 528, 2010. 9th IFAC Symposium on Dynamics and Control of Process Systems.
- [69] A. Kanevsky, M. H. Carpenter, D. Gottlieb, and J. S. Hesthaven. Application of implicit–explicit high order runge–kutta methods to discontinuous-galerkin schemes. *Journal of Computational Physics*, 225(2) :1753 – 1781, 2007.
- [70] A. Toumi, G. Dufour, R. Perrussel, and T. Unfer. Asynchronous numerical scheme for modeling hyperbolic systems. *Comptes Rendus Mathematique*, 353(9) :843 – 847, 2015.
- [71] V.A. Semiletov and S.A. Karabasov. Cabaret scheme for computational aero acoustics : Extension to asynchronous time stepping and 3d flow modelling. *International Journal of Aeroacoustics*, 13(3-4) :321–336, 2014.
- [72] A. Kvaerno and P. Rentrop. Low order multirate runge-kutta methods in electric circuit simulation. *Preprint No. 99/1, IWRMM, Universität Karlsruhe (TH)*, 1999.
- [73] S. Osher and R. Sanders. Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Mathematics of computation*, 41(164), October 1983.
- [74] W. Hundsdorfer, A. Mozartova, and V. Savcenco. *Analysis of explicit multirate and partitioned Runge-Kutta schemes for conservation laws*. CWI report. MAS-E. Centrum voor Wiskunde en Informatica, 2007.
- [75] C. Mikida, A. Klöckner, and D. Bodony. Multi-rate time integration on overset meshes. 2018.
- [76] B. Seny. *Development and implementation of a parallel explicit multirate time stepping strategy for accelerating discontinuous Galerkin computations*. PhD thesis, 2014.
- [77] Jean M. Sexton and D. Reynolds. Relaxed multirate infinitesimal step methods. *Preprint submitted to Journal of Computational and Applied Mathematics*, 08 2018.

- [78] A. Sandu. A class of multirate infinitesimal gark methods. *Technical report of the computational science laboratory (Virginia Tech)*, 08 2019.
- [79] O. Knoth and R. Wolke. Implicit-explicit runge-kutta methods for computing atmospheric reactive flows. *Applied Numerical Mathematics*, 28(2) :327 – 341, 1998.
- [80] B. Seny, J. Lambrechts, T. Toulorge, V. Legat, and J.-F. Remacle. An efficient parallel implementation of explicit multirate runge–kutta schemes for discontinuous galerkin computations. *Journal of Computational Physics*, 256 :135 – 160, 2014.
- [81] M. Günther, A. Kværnø, and P. Rentrop. Multirate partitioned runge-kutta methods. *BIT Numerical Mathematics*, 41(3) :504–514, Jun 2001.
- [82] Christophe Bogey. Grid sensitivity of flow field and noise of high-reynolds-number jets computed by large-eddy simulation. *International Journal of Aeroacoustics*, 17 :399 – 424, 2018.
- [83] L. Liu, X. Li, and F. Q. Hu. Nonuniform time-step runge–kutta discontinuous galerkin method for computational aeroacoustics. *Journal of Computational Physics*, 229(19) :6874 – 6897, 2010.
- [84] L. Liu, X. Li, and F. Q. Hu. Nonuniform-time-step explicit runge–kutta scheme for high-order finite difference method. *Computers and Fluids*, 105 :166 – 178, 2014.
- [85] T. Renaud, C. Benoit, S. Peron, I. Mary, and N. Alferez. Validation of an immersed boundary method for compressible flows. *AIAA Scitech 2019*, 2179, 2019.
- [86] L. Krivodonova. An efficient local time-stepping scheme for solution of nonlinear conservation laws. *Journal of Computational Physics*, 229(22) :8537 – 8551, 2010.
- [87] I. Mary. *Méthode de Newton approchée pour le calcul d’écoulements stationnaires comportant des zones à très faible nombre de Mach*. PhD thesis, 1999.
- [88] I. Mary and P. Sagaut. Large eddy simulation of flow around an airfoil near stall. *AIAA Journal*, 40(6) :1139–1145, 2002.
- [89] C. Laurent. *Étude d’écoulements transitionnels et hors équilibre par des approches DNS et RANS*. PhD thesis, 2012.
- [90] J.R. Edwards and M.-S. Liou. Low-diffusion flux-splitting methods for flows at all speeds. *AIAA Journal*, 36(9) :1610–1617, 1998.
- [91] N. Alferez, I. Mary, and E. Lamballais. Study of stall development around an airfoil by means of high fidelity large eddy simulation. *Flow, Turbulence and Combustion*, 91(3) :623–641, 2013.
- [92] B. Van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov’s method. *Journal of Computational Physics*, 32(1) :101 – 136, 1979.
- [93] M. Meinke, W. Schröder, E. Krause, and Th. Rister. A comparison of second- and sixth-order methods for large-eddy simulations. *Computers Fluids*, 31(4) :695 – 718, 2002.
- [94] P.L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2) :357 – 372, 1981.

- [95] S.R Chakravarthy. High resolution upwind formulation for the navier-stokes equations. *Von Karman Institute Lecture Series*, 32(1) :101 – 136, 1988.
- [96] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1) :97 – 127, 1999.
- [97] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1) :1 – 31, 1978.
- [98] <https://how5.cenaero.be>. 5th international workshop on high order cfd methods. *In 2018 AIAA SciTech Conference*, 2018.
- [99] J. R. DeBonis. Solutions of the taylor-green vortex problem using high-resolution explicit finite difference methods. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013.
- [100] M. Galbraith and M. Visbal. *Implicit Large Eddy Simulation of Low-Reynolds-Number Transitional Flow Past the SD7003 Airfoil*.
- [101] A. Uranga, P.O. Persson, M. Drela, and J. Peraire. Implicit large eddy simulation of transition to turbulence at low reynolds numbers using a discontinuous galerkin method. *International Journal for Numerical Methods in Engineering*, 87(1-5) :232–261, 2011.
- [102] A. Ducoin and J.C. Robinet. Etude numérique de la transition laminaire turbulent sur un profil d’aile. *21eme congrès français de mécanique*, 2013.
- [103] O. Marxen and Dan S. Henningson. The effect of small-amplitude convective disturbances on the size and bursting of a laminar separation bubble. *Journal of Fluid Mechanics*, 671 :1–33, 2011.
- [104] C. Laurent, I. Mary, V. Gleize, A. Lerat, and D. Arnal. Dns database of a transitional separation bubble on a flat plate and application to rans modeling validation. *Computers and Fluids*, 61 :21–30, 2012.
- [105] Kevin W Thompson. Time dependent boundary conditions for hyperbolic systems. *Journal of Computational Physics*, 68(1) :1 – 24, 1987.
- [106] J. Dandois, I. Mary, and V. Brion. Large-eddy simulation of laminar transonic buffet. *Journal of Fluid Mechanics*, 850 :156–178, 2018.
- [107] L. Larchevêque, P. Sagaut, I. Mary, O. Labbé, and P. Comte. Large-eddy simulation of a compressible flow past a deep cavity. *Physics of Fluids*, 15(1) :193–210, 2003.
- [108] M. Rietmann, D. Peter, O. Schenk, B. Uçar, and M.J. Grote. Load-balanced local time stepping for large scale wave propagation. *2015 IEEE International Parallel and Distributed Processing Symposium, IEEE CPS*, 2015.
- [109] Christophe Benoit, Stéphanie Péron, and Sâm Landier. Cassiopee : A cfd pre- and post-processing tool. *Aerospace Science and Technology*, 45 :272 – 283, 2015.
- [110] Rapport final prf champion. *Communication interne ONERA*, 2019.
- [111] Stéphanie Péron and Christophe Benoit. Automatic off-body overset adaptive cartesian mesh method based on an octree approach. *Journal of Computational Physics*, 232(1) :153 – 173, 2013.

- [112] S. Péron, C. Benoit, V. Gleize, I. Mary, and M. Terracol. *A mixed overset grid/immersed boundary approach for CFD simulations of complex geometries.*
- [113] Charles S Peskin. Flow patterns around heart valves : A numerical method. *Journal of Computational Physics*, 10(2) :252 – 271, 1972.
- [114] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1) :35 – 60, 2000.
- [115] Yu-Heng Tseng and Joel H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2) :593 – 623, 2003.
- [116] A. J. Musker. Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer. *AIAA Journal*, 17(6) :655–657, 1979.
- [117] A. G. Kravchenko and P. Moin. Numerical studies of flow over a circular cylinder at $Re=3900$. *Physics of Fluids*, 12(2) :403–417, 2000.
- [118] Lourenco L. M. and Shih C. Characteristics of the plane turbulent near wake of a circular cylinder. a particle image velocimetry study (communication privée, données publiées dans "Numerical experiments on the flow past a circular cylinder at sub-critical reynolds number", Beaudan and Moin (1994)).