UNIVERSITÉ DE MONTRÉAL

DÉVELOPPEMENT D'UNE MÉTHODE D'ACCÉLÉRATION PAR GRILLES VIRTUELLES RÉCURSIVES POUR L'ASSEMBLAGE DE MAILLAGES CHIMÈRES

ALEXANDRE PIGEON
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE AÉROSPATIAL)
MARS 2015

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉVELOPPEMENT D'UNE MÉTHODE D'ACCÉLÉRATION PAR GRILLES VIRTUELLES RÉCURSIVES POUR L'ASSEMBLAGE DE MAILLAGES CHIMÈRES

présenté par : PIGEON Alexandre
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. CAMARERO Ricardo, Ph. D., président
M. LAURENDEAU Éric, Ph. D., membre et directeur de recherche
M. SERMEUS Kurt, Ph. D., membre

REMERCIEMENTS

J'aimerais remercier avant tout mon directeur de maîtrise, le professeur Éric Laurendeau, pour son soutien tout au long de ce projet de deux ans. Le professeur Laurendeau constitue selon moi un des piliers de mon développement académique et professionnel. Il a su orienter de manière précise ma formation pour m'apporter des compétences tant techniques que générales. Grâce à son désir de repousser les limites du développement et de l'innovation, j'ai eu la chance de travailler sur un projet stimulant à l'état de l'art des connaissances modernes. Éric, je te remercie grandement de m'avoir accepté au sein de ton équipe de recherche.

J'aimerais aussi remercier mes parents, Gilles et Rachel, qui m'ont non seulement encouragé à travers mon parcours, mais aussi supporté tout au long de celui-ci. De mon plus jeune souvenir, mes parents ont toujours tout fait pour m'amener à réaliser mon plein potentiel et je leur en serai pour cela éternellement reconnaissant. Leur support tant moral qu'économique me permet aujourd'hui d'évoluer dans un domaine qui me passionne et pour lequel ils ont contribué à bien me préparer. Gilles et Rachel, vous représentez pour moi un modèle parental excellent que je m'assurerai de répliquer dans mon futur.

Je remercie aussi ma sœur, Kateryne, qui s'est montré une motivatrice incomparable dans plusieurs étapes de ma vie. Kat, ton soutien m'a souvent apporté énergie et réconfort et je t'en suis très reconnaissant. À travers notre complicité, tu as assurément contribué à m'amener là où je suis aujourd'hui.

Je tiens à remercier Thibaut Deloze pour son support incroyable tout au long de ce projet. Grâce à l'explication de plusieurs concepts clés, Thibaut a grandement contribué à faire de ce projet un succès. Il a su m'apprendre énormément sur le développement numérique et je dois une partie de mes compétences aux innombrables discussions que nous avons eues au cours des deux dernières années. Thibaut, ta bonne humeur inégalée et contagieuse amenait dans nos bureaux une ambiance unique et incomparable. Tu es devenu pour moi plus qu'un collègue de travail, tu es un grand ami.

J'aimerais finalement remercier l'ensemble de mes collègues de travail et plus spécialement Simon et Antoine. Merci Simon de m'avoir accompagné tard le soir durant l'écriture de ce mémoire et de m'avoir aidé à bien formater ce document ! Antoine, tu as su m'apprendre rapidement beaucoup de techniques de programmation très efficaces et je t'en suis très reconnaissant.

RÉSUMÉ

Les technologies numériques permettant de calculer des écoulements fluides sont aujourd'hui très répandues dans les domaines de la recherche et de l'industrie. Elles permettent de réduire les coûts et les risques associés à la conception d'aéronefs modernes. Quoique très efficaces, ces technologies nécessitent encore d'être améliorées pour être capable de modéliser correctement les conditions du fluide et les caractéristiques géométriques du modèle à l'étude. Pour faire face à la problématique géométrique, le concept de superposition de grilles fut introduit vers la fin des années 1980. Mieux connue comme la méthode chimère, cette approche offre au concepteur une flexibilité accrue lors de la phase de génération de grille ce qui permet d'augmenter la qualité des grilles produites et conséquemment des résultats d'une simulation. La méthode est aussi très bien adaptée aux simulations d'éléments en mouvement relatif puisqu'elle élimine la nécessité de générer un maillage à chaque fois qu'un élément solide est déplacé permettant ainsi de réduire significativement le temps de calcul de ce type de simulation. La méthode chimère est couramment utilisée au sein de groupes de recherche et est aujourd'hui suffisamment mature pour être introduite dans les pratiques industrielles. L'utilisation de la technique requiert d'assembler un système de grilles superposées de sorte à établir un lien de communication entre l'ensemble des grilles du système. Cette communication existe grâce aux zones de superpositions dans lesquelles l'information est transmise d'une grille à l'autre via des interpolations. À la suite de l'assemblage, chacune des cellules du système de grilles obtient un des trois statuts mutuellement exclusifs suivants : calculée, interpolée ou invalide. L'assemblage produit peut ensuite être utilisé par un solveur de fluide capable de traiter la structure de communication ainsi générée.

Le présent projet contient deux objectifs principaux. D'abord, la démonstration de la technologie sera réalisée grâce à une implémentation de la méthode chimère au sein de la plateforme de recherche NSCODE. Cette première phase vise à développer les outils de base de la technique de façon à minimiser les risques de développement. Ainsi, le module aura la capacité de traiter des géométries bidimensionnelles et sera basé sur une architecture mémoire simple. La deuxième phase du projet consiste à démontrer que la méthode est adaptée aux requis industriels en améliorant le programme développé. Pour cela un module indépendant, capable d'opérer avec un solveur fluide industriel, sera développé. Ce module offrira trois capacités supplémentaires par rapport à la première phase ; la faculté d'analyser des géométries tridimensionnelles, une architecture mémoire partagée et le traitement de fichiers d'entrée et de sortie standardisés.

Dans la première phase d'implémentation, les techniques permettant de déterminer le statut des cellules sont développées. La procédure retenue implique de parcourir l'ensemble des superpositions et d'établir une relation de domination entre les cellules superposées. Les cellules invalides, contenues à l'intérieur de géométries solides, sont identifiées grâce à une procédure d'élimination. Une technique répandue au sein de la communauté et basée sur l'utilisation de grilles virtuelles est implémentée afin d'accélérer le processus d'assemblage. Les limites de cette technique sont repoussées en ajoutant une composante de récursivité. Afin d'accélérer davantage l'assemblage, le programme est parallélisé sous mémoire partagée grâce à l'interface OpenMP. L'ensemble du module est intégré au solveur de fluide NSCODE afin de valider les capacités d'assemblage du programme.

La deuxième phase de développement fait usage des mêmes techniques de détection et d'accélération que durant la première phase en considérant l'ajout de la troisième dimension. L'implémentation est basée sur une architecture de calcul à mémoire distribuée permettant d'offrir un gain en performance lors de calculs réalisés en parallèle. Le processus de communication MPI est choisi pour établir la communication entre les différents processeurs. Un algorithme pouvant assigner les tâches aux différents processeurs de manière automatique est développé. Le fait que les calculs soient effectués de manière simultanée nécessite l'utilisation d'une procédure de test supplémentaire pour assembler de manière adéquate le système. Le standard d'entrée et de sortie CGNS, largement répandu dans la communauté, est utilisé pour offrir au programme une compatibilité avec d'autres modules utilisant ce standard.

L'assemblage des grilles est validé en deux et en trois dimensions sur des cas simples et sur des configurations complexes. La précision des assemblages produits par les deux modules est validée comme étant de second ordre. L'intégration du module bidimensionnel au sein de NSCODE préserve l'ordre de précision et les caractéristiques de convergence de ce dernier. Les résultats prédis par la méthode chimère sur une géométrie multiéléments sont en accord avec les valeurs expérimentales. Les deux implémentations démontrent une capacité d'assemblage parallèle et offrent de bonnes performances lorsqu'elles sont utilisées avec plusieurs processeurs. La méthode d'accélération des grilles virtuelles montre d'excellents résultats tant sur des cas académiques que sur des systèmes de grilles représentatifs des géométries industrielles. L'approche récursive de cette technique accélère radicalement le processus d'assemblage sur les cas académiques, mais requiert quelques améliorations pour maintenir cette efficacité sur les cas industriels.

Le programme d'assemblage bidimensionnel est simple et fonctionnel pour des cas académiques. Il est adapté aux besoins de recherche et de développement. Le programme tridimensionnel est quant à lui plus complet, plus robuste, plus performant et bien adapté

aux besoins d'assemblage complexe. Trois aspects limitent toutefois la performance de l'algorithme. D'abord, la technique d'identification des cellules invalides est restrictive et peut causer des problèmes dans l'assemblage généré. Ensuite, il manque à l'algorithme la faculté de répartir de manière uniforme la charge de travail lorsque des calculs parallèles sont effectués. Finalement, la technique d'accélération par grilles virtuelles récursives doit être améliorée pour être performante sur les cas industriels. Des axes de développement sont proposés pour traiter ces limites et pour amener plus de fonctionnalités à l'implémentation.

ABSTRACT

Computational fluid dynamics is nowadays widespread in research domains and industrial practices. The technology allows for drastic cost and risk reduction during the design phase of modern aircrafts. Though they provide great capabilities, the technology still requires improvements regarding its ability to accurately model the fluid state as well as the geometric characteristics of the physical body being studied. To deal with the issue of the geometry, the concept of grid overlapping was introduced at the end of the 80's to provide an increased flexibility during the grid generation process allowing for an increase in the quality of the produced grids and simulations results. The method is especially well suited to simulations where solid elements are in relative motion since it eliminates the need to generate a new grid each time a solid body is displaced which reduces significantly the time required to perform those simulations. Also known as the chimera approach, the technique is well known amongst research groups and is now sufficiently mature to be introduced into industrial practices. The technique requires to assemble a system of overlapping grids in a way to establish a communication link between the grids. This communication can only exists in overlap zones in which information is transferred from one grid to another via interpolations. Following the assembly process, each cell of the component grids retains one of the following three mutually exclusive status: computed, interpolated or invalid. The grid assembly can then be used by a flow solver equipped with the capability of using the communication structure.

The current project entails two main objectives. First the demonstration of the technology will be performed by implementing the chimera technique into the research platform NSCODE. This first phase intends to develop the basic functionalities of the technique in order to minimize the associated risks. The module will have two-dimensional capabilities and will be based on a simple memory architecture, a shared memory. The second phase of the project aims at demonstrating the maturity of the technology for industrial needs and will thus improve on the first phase. For this, an independent module, capable of operating in conjunction with industrial flow solvers, will be developed. The module will offer the following three major upgrades: three-dimensional capabilities, a distributed memory architecture and the ability to treat standardized input and output files.

In the first implementation phase, the techniques used to determine the status of the mesh cells will be developed. The chosen procedure involves searching for all possible overlaps and establish a dominance relation between the overlapping cells. The invalid cells, contained inside a solid element geometry, are identified via an elimination procedure. To acceler-

ate the assembly process, the technique of virtual grids, well known by the community, is implemented. The limits of this technique are further pushed by considering a novel approach based on recursive virtual grids. Parallelization of the program is achieved via the shared memory OpenMP interface. The entire module is then integrated into the flow solver NSCODE to validate the assembling capabilities of the program.

The second development phase extends the detection and acceleration techniques of the first phase to their three-dimensional version. The program is implemented on a distributed memory architecture allowing for an increase in performance when run on parallel computers. The MPI standard is chosen as the communication interface between the processing units. An automatic load balancing algorithm is developed. The synchronism of the assembly procedure on parallel computers requires further testing to ensure proper assembly of the overlapping grids. The CGNS input/output standard, widely used by the aerospace community, is applied to provide compatibility with complementary programs using the standard.

The assembled overlapping system of grid is validated in two and three dimensions on simple test cases as well as complex configurations. The accuracy of the assembled grids are verified to be of second order. The integration of the bidimensional module preserves the order of accuracy and the convergence characteristics of NSCODE. The numerical results of the simulation of a multi-element geometry with the chimera the technique are in agreement with experimental values. Both overset grid assembly module provide parallel computation capabilities and exhibit good scalability characteristics. The virtual grid acceleration method shows excellent results both on academic test cases and on complex systems of grids representative of industrial configurations. The recursive approach of this technique shows radical improvements in terms of computation time on academic test cases but still requires refinements to yield the same efficiency of industrial geometries.

The two-dimensional assembler is simple and functional for academic test cases and simple geometries. It's well suited for research and development purposes. The three-dimensional program is more robust, provides added functionalities and performance and is very well suited to complex assembly needs. Still, three aspects limit its performance. First, the procedure used to identify invalid cells is restrictive and can cause the assembly to contain holes in specific conditions. Second, the program currently lacks the ability to appropriately balance the load on the processors when parallel computations are performed. Finally, the recursive virtual grid acceleration technique must be refined so that it retains its efficiency on industrial configurations. Future development suggestions are proposed to deal with those limitations and to offer increased functionalities to the program.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

CFD	Computational Fluid Dynamics
CGNS	CFD General Notation System
MDA	Mcdonnell Douglas Airfoil
MPI	Message Passing Interface
ADT	Alternating Digital Tree

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

L'analyse numérique des fluides, communément appelée en anglais «Computational Fluid Dynamics (CFD)» a connu un important essor au cours des dernières décennies. Cette technologie permet de résoudre des systèmes d'équations impliquant divers phénomènes physiques tels les écoulements fluides, la transmission de chaleur et les réactions chimiques. La CFD est aujourd'hui couramment utilisée dans plusieurs domaines de recherche et milieux industriels : aéronautique, ingénierie maritime, combustion, procédés chimiques, météorologie, etc. En réduisant le besoin de résultats expérimentaux, cette technologie permet de diminuer le nombre de tests physiques souvent coûteux durant les phases de conception. De plus, la CFD réduit radicalement le temps requis à l'évaluation de différents concepts, permet d'étudier des systèmes passé leurs limites d'opération et offre une excellente capacité à effectuer des optimisations.

1.1 Concepts de base

1.1.1 Équations

Les techniques CFD permettent de résoudre les équations de Navier-Stokes. Plus globalement, elles résolvent un système d'équations de transport. Ces équations sont dérivées en appliquant les lois de conservation de la masse, de la quantité de mouvement et de l'énergie aux fluides. La forme générale d'une telle équation est la suivante (H K Versteeg, 2007) :

$$\frac{\partial}{\partial t}(\rho\phi) + \mathrm{div}(\rho\vec{u}\phi) = \mathrm{div}(\Gamma \mathrm{grad}\phi) + S_\phi \tag{1.1}$$

où ρ représente la densité du fluide, $\vec{u}$ le vecteur de vitesse, Γ le coefficient de diffusion, S_ϕ le terme source et ϕ constitue une propriété générale du fluide. Il existe une multitude de techniques permettant de résoudre ce système d'équations, mais toutes passent par les deux mêmes étapes : un calcul physique et un calcul temporel, chacun requérant une discrétisation. On parlera alors respectivement de discrétisation spatiale et temporelle.

1.1.2 La méthode des volumes finis

Les diverses techniques permettant la discrétisation spatiale se divisent en trois catégories principales : les différences finies, les éléments finis et les méthodes spectrales. Une sous branche des différences finies, les volumes finis, s'est démarquée et constitue aujourd'hui la

base de la majorité des codes CFD commerciaux (H K Versteeg, 2007) et est la méthode utilisée au sein du présent projet.

La discrétisation spatiale implique premièrement de générer une grille, aussi appelée un maillage. Deux types fondamentaux de grilles existent, les grilles cartésiennes et les grilles adaptées au domaine de simulation. Vu leur précision supérieure, les grilles adaptées sont plus communes (Blazek, 2001) et seront celles utilisées dans le cadre du présent projet. Les grilles peuvent aussi être structurées ou non structurées. Sur une grille structurée, chaque point est identifié par ses indices topologiques i,j,k. Il existe ainsi une correspondance très simple entre un élément et ses voisins puisque la topologie est représentative de l'organisation de la mémoire dans l'ordinateur. Ceci rend les grilles structurées très simples à utiliser au sein des schémas de résolution (Blazek, 2001). En contrepartie, ce type de grille peut être très difficile à réaliser sur des géométries complexes. Plusieurs options existent pour réduire les contraintes reliées au processus de génération. Celle d'intérêt au présent document est la technique appelée «multibloc» qui permet de découper le domaine de simulation en plusieurs blocs et d'effectuer la génération des grilles de manière indépendante sur ceux-ci. Cette technique est aussi intéressante d'une perspective de parallélisation puisqu'elle divise le domaine de simulation. Le deuxième type de grille, non structuré, offre la meilleure flexibilité quant au traitement de géométries complexes, requiert très peu de temps à générer et a l'avantage de nécessiter moins de points que les grilles structurées. Par contre, les grilles non structurées augmentent la complexité du solveur, le temps de calcul et utilisent davantage de mémoire.

L'étape clé de la méthode des volumes finis consiste à intégrer l'équation de transport sur l'ensemble des volumes de contrôle de manière à obtenir un système algébrique d'équations discrétisées. Le système d'équations est ensuite résolu grâce à des techniques itératives. L'intégration de l'équation sur les volumes de contrôle exprime clairement les lois de conservation des propriétés du fluide au sein du volume de contrôle (H K Versteeg, 2007). Cette relation claire entre l'algorithme numérique et l'interprétation physique distingue la technique des volumes finis des autres techniques CFD et rend sa compréhension plus simple.

1.1.3 La technique chimère

Les méthodes CFD possèdent une sensibilité majeure vis-à-vis des erreurs de discrétisation spatiale. Il a été démontré que pour une même géométrie, deux maillages distincts peuvent produire des résultats substantiellement différents (Mavriplis, 2005). De plus, l'utilisation des techniques CFD a montré à travers le temps que la qualité de la grille avait un impact important sur la validité des résultats produits. Une grande partie des développements académiques et industriels visent conséquemment à produire des grilles de qualité élevée. La

technique chimère s'inscrit dans cette optique.

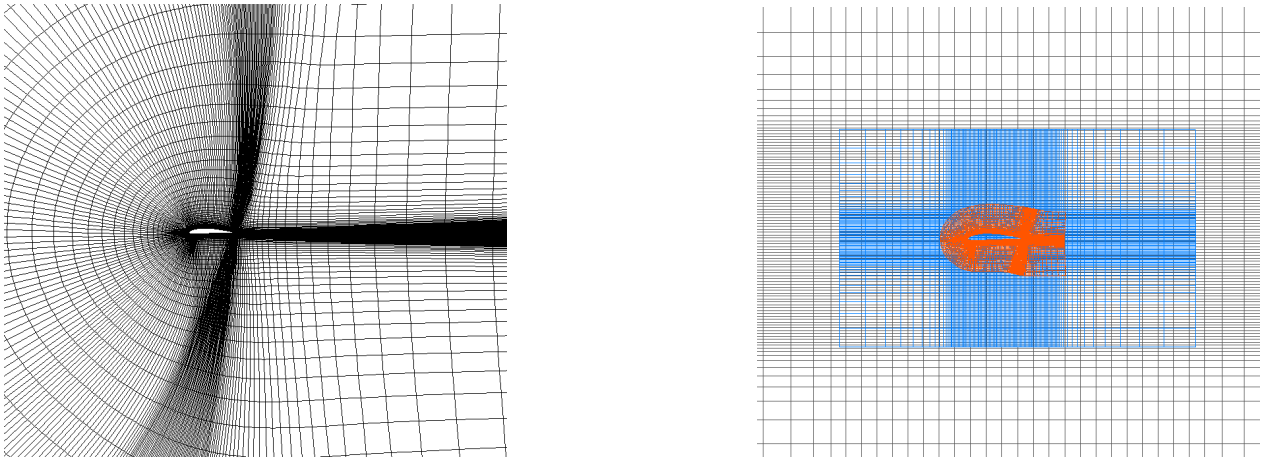
En présence de géométries complexes, le problème des grilles structurées est la difficulté à les générer de manière à ce qu'elles aient une bonne qualité sur l'ensemble du domaine. Le problème émerge de par la nature ordonnée de la grille qui limite sa capacité à s'adapter aux géométries. Malgré une flexibilité accrue apportée par la technique multibloc, la qualité du maillage produit reste discutable. Pour faire face à ce manque de flexibilité, la technique chimère fut conceptualisée vers la fin des années 1980 par la NASA (Chan, 2009). L'idée fondamentale derrière ce concept est de décomposer une géométrie complexe en plusieurs composantes simples. Cette décomposition permet de générer des grilles indépendantes pour chacun des éléments et de les assembler ensuite de manière à être traitable par un solveur. La technique peut donc être vue comme localement structurée, mais globalement non structurée (Prewitt *et al.*, 2000).

Avantages

La technique chimère comporte de multiples avantages. Premièrement, grâce à l'allègement des contraintes imposées au maillage, elle permet de réduire significativement le temps investi à la génération d'une grille de bonne qualité. En outre, la décomposition en élément simple facilite également la création automatique de maillages. Ces aspects sont capitaux considérant que 50% du temps investi sur un projet CFD industriel est dévoué à la définition du domaine de simulation et de la grille (H K Versteeg, 2007). Ces aspects ont une répercussion positive sur la qualité des résultats produits d'une simulation. Ainsi, la technique chimère offre l'avantage significatif de permettre l'analyse d'écoulements aérodynamiques pratiquement impossibles à réaliser sans celle-ci ce qui permet de mieux diriger la conception d'un aéronef.

Deuxièmement, la majorité des géométries modernes contiennent des régions ayant des requis de résolution très différents. Sans la technique chimère, les maillages structurés souffrent d'un problème systématique ; les régions de forte densité sont propagées à travers l'ensemble du domaine ce qui augmente inutilement le nombre de points du maillage. Ce phénomène est démontré sur la Figure 1.1(a) (extraite de (Rumsey, 2014)). On y voit la propagation vers le haut et le bas de la zone de haute densité située près du bord de fuite. Au contraire, le système de grilles superposées sur la Figure 1.1(b) ne présente une forte densité qu'aux endroits voulus. Cet aspect permet de réduire le temps requis pour effectuer des simulations aérodynamiques ce qui se traduit aussi par une réduction des coûts de développement.

Troisièmement, la technique chimère est très bien adaptée aux simulations où des éléments solides sont en mouvement relatif. En effet, une des techniques utilisées pour effectuer ces types de simulation consiste à régénérer une grille à chaque fois qu'un élément est déplacé.



(a) Grille standard (b) Grille chimère

Figure 1.1 Grille du profil NACA4412 sans chimère (gauche) et avec chimère (droite).

Cette approche est coûteuse en temps. L'alternative commune consiste plutôt à déformer le maillage. Quoique rapide, cette approche génère plusieurs difficultés numériques (Chandar *et al.*, 2013). La méthode chimère ne requiert quant à elle aucun changement de grille. Il ne suffit que de déplacer l'élément solide en question et de réassembler les grilles. Comme indiqué par Wissink et Meakin (1997), c'est la technique chimère qui a démontré le plus grand succès sur les simulations d'éléments en mouvement arbitraire.

1.2 Définitions

À travers le processus d'assemblage, la méthode chimère fait intervenir quatre différents types de traitement. Ces derniers sont brièvement présentés ici afin de clarifier le reste de la discussion.

- *Cellule calculée* : Les cellules calculées sont celles utilisées pour résoudre l'écoulement.
  - *Cellule dominante* : Les cellules dominantes sont des cellules calculées, mais aussi superposées aux cellules d'autres maillages.
- *Cellule interpolée* : Les cellules interpolées, aussi appelées cellules réceptrices, ne sont pas utilisées par résoudre l'écoulement. Ce type de cellule implique une superposition avec des cellules calculées. Les cellules interpolées sont dominées et reçoivent l'information de leurs cellules donneuses.
- *Cellule invalide* : Les cellules invalides sont celles situées à l'intérieur de géométries solides d'autres maillages. Elles sont retirées des calculs afin de ne pas générer de contamination numérique.

1.3 Éléments de la problématique

1.3.1 Assemblage des grilles superposées

Par leur nature indépendante, les maillages constituant le système de grilles ne contiennent aucune information concernant la superposition sur les autres maillages du système. Or, les grilles superposées sont utilisées en conjonction dans le solveur. Chacune résout une partie de l'écoulement et transmet de l'information de cet écoulement aux autres grilles. Pour cette raison, il faut créer une information d'interconnectivité pour l'ensemble des grilles constituant le système. Le problème d'assemblage des grilles superposées consiste à traiter le système de maillage de manière à ce que les maillages dominants soient identifiés dans les régions de superposition (Roget et Sitaraman, 2013a). À la suite de l'assemblage, chacune des cellules a un des trois statuts mutuellement exclusifs suivants : calculée, interpolée, invalide. Aussi, en un endroit où plusieurs cellules se superposent, seules les cellules du maillage dominant sont utilisées pour la résolution des équations. Toutes les autres sont réceptrices et reçoivent l'information via interpolation des cellules dominantes qui les superposent.

La procédure implique de générer une structure permettant au solveur d'utiliser de manière appropriée le système de grilles et d'effectuer le transfert des données des maillages dominants aux maillages dominés. Puisqu'il n'existe pas de correspondance géométrique exacte entre une cellule réceptrice et les cellules donneuses, les données transférées doivent être évaluées à l'emplacement géométrique de la cellule réceptrice. Ceci est accompli en effectuant une interpolation des données de plusieurs cellules donneuses situées proche de la cellule réceptrice. Ces interpolations impliquent toutefois une erreur sur les données qui sont transférées. Pour que l'assemblage du système de grille soit compatible avec les techniques numériques utilisées au sein d'un solveur, l'erreur d'interpolation doit être d'ordre équivalent ou supérieur à celle des schémas de discrétisation constituant le solveur. Suite à l'assemblage, la structure de communication intergrille contient pour chaque maillage la liste des cellules réceptrices, leurs cellules donneuses ainsi que les poids d'interpolation de ces dernières. La structure contient aussi l'ensemble des points invalides qui doivent être retirés des calculs.

1.3.2 Efficacité

Un des buts d'un code d'assemblage chimère est d'effectuer des simulations avec éléments solides en mouvement relatif. Ces simulations requièrent de réaliser l'assemblage à chaque fois qu'un élément est déplacé. Pour que la méthode chimère reste avantageuse par rapport à l'alternative commune consistant à recréer un maillage, la procédure d'assemblage ne doit prendre qu'une fraction du temps attribué à la résolution de l'écoulement. Pour y parvenir,

il est nécessaire d'optimiser les fonctions les plus lourdes de l'algorithme. Celles-ci sont la recherche de donneurs et la recherche des cellules invalides (Zagaris *et al.*, 2010).

1.3.3 Automatisation

Avec l'évolution des capacités de la CFD, les géométries d'intérêt se sont grandement complexifiées. La même tendance s'est observée au niveau de la complexité du système de maillages superposés utilisé pour représenter adéquatement ces géométries. Jusqu'à récemment, les implémentations de la technologie chimère nécessitaient une importante contribution de la part de l'usager pour effectuer adéquatement le découpage des zones invalides et pour définir les zones pouvant être donneuses (Rogers *et al.*, 2003). L'expérience a montré que pour des configurations complexes tridimensionnelles, la définition manuelle des zones invalides peut être extrêmement difficile et demande une très grande expertise technique (Wang et Parthasarathy, 2000). Ce processus exige une connaissance complète de la géométrie à l'étude et nécessite plusieurs tentatives d'essai-erreur. Dans une perspective d'utiliser un code d'assemblage chimère durant plusieurs phases de conception et par des utilisateurs ayant des niveaux de connaissances différents de la technologie, il est impératif que l'implémentation chimère ne demande que très peu de contribution de la part de l'utilisateur. Cette considération est encore plus importante vis-à-vis des simulations à éléments en mouvement relatif.

1.3.4 Parallélisation

Les phénomènes physiques aujourd'hui modélisés sont très complexes et requièrent un important temps de calcul. De plus, le développement actuel de la puissance de calcul rend accessible les simulations CFD possédant maintenant cent millions de points (Chan, 2009). Il est impossible de réaliser d'aussi grosses simulations en effectuant les calculs de manière séquentielle. Dans cette optique, une vaste partie des codes CFD font désormais usage de parallélisation et les grands joueurs de l'industrie et de la recherche se sont outillés de superordinateurs (Sermeus *et al.*, 2007). La parallélisation permet de diviser les tâches et de les distribuer sur plusieurs éléments participants au calcul appelés nœuds. En exploitant la puissance d'une multitude de nœuds, on peut ainsi accélérer une simulation jusqu'à trente fois (Prewitt *et al.*, 2000). Quoiqu'attirante, une parallélisation efficace d'un code implique deux étapes délicates. La première consiste à analyser le schéma numérique utilisé pour identifier les sections pouvant être exécutées simultanément et pour avoir une compréhension totale des dépendances entre les données. La deuxième étape vise la conception d'une stratégie parallèle capable de distribuer de la manière la plus uniforme que possible la charge de

travail à travers l'ensemble des nœuds et capable d'effectuer le calcul de manière parallèle. La conception doit viser à réduire au minimum les communications entre les nœuds puisque celles-ci diminuent l'efficacité globale de la procédure. Ces considérations peuvent conduire à des changements dans i) l'organisation temporelle des tâches, ii) l'organisation des données, iii) les algorithmes utilisés pour effectuer le calcul (Chawla et Weertunga, 1995). Ceci implique souvent une réorganisation importante du code accompagnée par une augmentation de la complexité.

1.4 Objectifs de recherche

Le présent projet est réalisé sous la direction de Éric Laurendeau, professeur agrégé à l'École Polytechnique de Montréal. Le professeur Laurendeau a développé un code de recherche, NSCODE, utilisé pour tester différentes techniques numériques novatrices applicables au domaine de l'aérospatial. Le code est bidimensionnel et constitue une plateforme pour développer d'éventuels algorithmes tridimensionnels. Plusieurs logiciels d'assemblage chimère existent aujourd'hui, mais soit ils sont protégés par des closes de non-distribution ou bien leur utilisation demande de payer une licence onéreuse. Par ailleurs, les codes commerciaux sont souvent moins efficaces que ceux dédiés à la solution de problèmes précis.

Les codes bidimensionnels et tridimensionnels font usage de méthodes numériques similaires et présentent alors de fortes similitudes. L'implémentation de la technique chimère dans un code ou l'autre implique donc pratiquement les mêmes défis. Une différence notable existe toutefois entre les deux types de codes. En effet, vu l'envergure des calculs réalisés en trois dimensions, ces codes font généralement usage d'une mémoire distribuée alors que NSCODE, puisque bidimensionnel, utilise une mémoire partagée. La différence entre les mémoires partagée et distribuée est abordée dans la section 4. Suivant ce contexte, le présent projet vise à faire une démonstration de la maturité de la technologie chimère pour des applications industrielles modernes du domaine aéronautique. Le projet contient précisément les trois buts suivants :

1. Implémenter la technologie chimère au sein du code de recherche NSCODE ;
2. Développer un module de traitement chimère adapté à un code industriel tridimensionnel ;
3. Évaluer les défis des implémentations parallèles de la méthode chimère bidimensionnelle et tridimensionnelle ;

1.5 Plan du mémoire

Ce document se divise en trois sections principales. D'abord, une revue de la littérature entourant les pratiques communes liées à la technique chimère sera faite. On y présentera un aperçu de l'historique de la technique ainsi que ses applications modernes avant de détailler les étapes permettant l'assemblage de grilles superposées. Cette section expliquera aussi les algorithmes capables d'accélérer l'assemblage des grilles et présentera les implémentations parallèles retenues pour concevoir le programme final. La deuxième section consiste en l'implémentation bidimensionnelle de la méthode chimère au sein de NSCODE. Une introduction aux fondements du code sera faite et la présentation détaillée de l'implémentation suivra. On y montrera les traitements topologiques et les approches utilisées pour identifier le statut des cellules des maillages. Cette section abordera aussi les techniques d'accélération et de parallélisation utilisées pour réduire le temps d'assemblage et l'intégration à NSCODE. La troisième section consiste en l'implémentation tridimensionnelle. On y présentera les modifications apportées aux algorithmes bidimensionnels ainsi qu'une librairie standard utilisées à des fins de standardisation des données d'entrée et de sortie. Cette section abordera aussi la parallélisation sous mémoire distribuée et le concept de répartition de la charge suivi des résultats de l'implémentation. La conclusion du document présentera la synthèse des travaux avec les limitations des solutions proposées et proposera des améliorations possibles pour les travaux futurs.

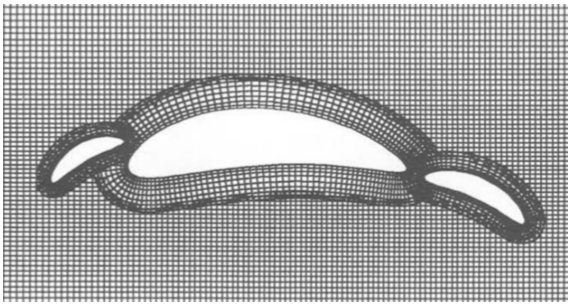
CHAPITRE 2 REVUE DE LITTÉRATURE

2.1 Historique de la technique chimère

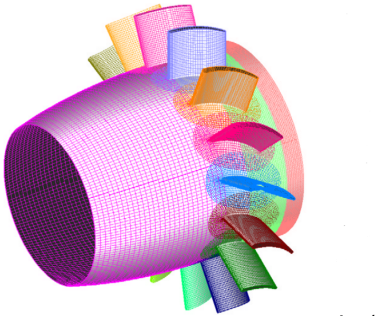
La technique chimère, aujourd'hui bien établie au sein de la communauté CFD, fait partie de la gamme de techniques visant à décomposer le domaine de simulation en éléments plus simples. Plusieurs des concepts liés à la simulation sur grilles superposées tirent leur origine des idées avant-gardistes de Joseph Steger, un chercheur de la NASA durant les années 1970 et 1980. La première implémentation de la technique fut réalisée en 1986 avec le code d'assemblage de grille PEGASUS par Benek (1986) et le solveur OVERFLOW. Ces codes furent initialement utilisés pour effectuer des analyses de profils multiéléments, de montée en orbite de la navette spatiale puis, à travers plusieurs améliorations, appliqués aux analyses d'écoulements de turbomachines, d'avions de chasse et de séparation de missiles (Chan, 2009). La Figure 2.1 montre l'évolution de la complexité des géométries traitées par la technique chimère. Cette complexité ainsi que la résolution des grilles utilisées ont rapidement nécessité une parallélisation de la méthode, accomplie pour la première fois par Barszcz *et al.* (1993). La technique a été utilisée pour étudier des phénomènes aussi variés que la combustion, le débit sanguin, les écoulements à vitesse élevés, pulmonaires et autour des navires, l'acoustique ainsi que dans le domaine aéronautique (Deloze, 2011). La NASA est restée un précurseur dans le développement de la technique chimère vu la complexité des géométries étudiées par l'organisme et une grande partie des développements de la technique chimère sont inspirés des concepts mis de l'avant par l'organisme. La technologie a aujourd'hui atteint un niveau de maturité suffisant pour être introduite dans les pratiques industrielles. Dans ce contexte, la performance en terme de temps de calcul de l'algorithme est devenue cruciale pour permettre l'utilisation de la technique durant les phases de conception. À ce fait, la technique chimère a récemment été implémentée sur une architecture de calcul GPU par Soni *et al.* (2012).

La procédure d'assemblage utilisée par la bonne majorité ((Wang et Parthasarathy, 2000), (Rogers *et al.*, 2003), (Deloze, 2011), (Chandar *et al.*, 2013) et (Roget et Sitaraman, 2013a)) des implémentations modernes de la technique chimère se divise en deux parties :

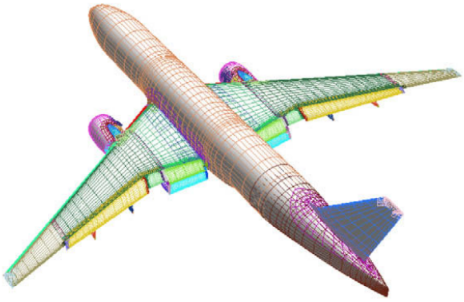
1. l'identification des cellules invalides à l'intérieur de géométries solides. Cette étape est communément appelée *holecutting* au sein de la communauté. Dans un but de clarté, le terme holecutting sera utilisé durant la présentation du présent projet.
2. l'identification des cellules interpolées et de leurs donneuses.



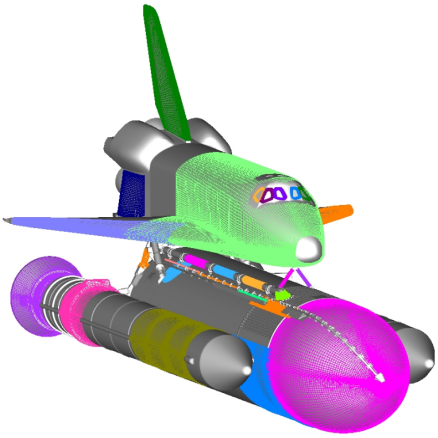
(a) Grille utilisée par Chesshire et Henshaw (1990) pour calculer un écoulement autour d'un profil multiélément (extraite de (Chesshire et Henshaw, 1990)).



(b) Grille chimère d'une turbopompe (extraite de (Chan *et al.*, 2002)).



(c) Grille d'un avion de transport civil en configuration atterrissage (extraite de (Chan, 2009)).



(d) Grille de la navette spatiale incluant les fusées et le réservoir à essence. La quasi-totalité des composantes est modélisée (extraite de (Chan, 2009)).

Figure 2.1 Évolution des géométries traitées par la technique chimère.

2.2 Holecutting

L'étape de holecutting consiste à identifier les points situés à l'intérieur de géométries solides. L'ensemble des cellules identifiées durant cette procédure représente un trou, d'où le nom holecutting. Cette identification permet d'éviter que des cellules soient utilisées pour effectuer un calcul là où le fluide ne peut se rendre. Quoique relativement simple à effectuer sur des géométries bidimensionnelles, la difficulté présentée par les géométries tridimensionnelles est considérable. L'étape de holecutting est une des plus délicates de l'assemblage chimère ce qui en fait un des axes de recherche les plus importants aujourd'hui.

Une des méthodes simple à utiliser consiste à faire usage de solides prédéfinis. Ceci implique

de définir analytiquement les solides en question ce qui peut représenter beaucoup de travail pour un cas traitant plusieurs éléments. Cette approche offre l'avantage d'être très précise et rapide, mais elle atteint vite ses limites lorsque la géométrie ne peut être facilement décrite analytiquement. Cette méthode a été utilisée avec succès par Deloze (2011) pour étudier la chute d'une sphère libre dans un tube vertical. Cette méthode n'est clairement pas adaptée à la complexité des géométries étudiées dans le domaine aéronautique.

La deuxième technique envisagée, une procédure plus généralisée que l'évaluation analytique qui fut développée au sein de la première version de PEGASUS par Benek (1986), est illustrée à la Figure 2.2. (a) En définissant C comme la surface solide, (b) la première étape consiste à construire le vecteur normal $\vec{N}$ à chaque point P_C définissant C. On définit ensuite un point temporaire, P_O, par la moyenne des points P_C. (c) On définit un cercle de recherche centré sur P_O de rayon R_{max} où R_{max} est la distance maximale entre P_O et les points P_C. Pour chaque point de la grille, on compare la magnitude du vecteur $\vec{r}$, le vecteur de position relative d'un point test P par rapport à P_O. Si la magnitude du vecteur $\vec{r}$ est supérieure à R_{max}, le point n'est pas contenu dans le cercle de recherche et est donc négligé. Dans le cas contraire, une analyse supplémentaire est nécessaire. (d) On calcule le produit scalaire des vecteurs $\vec{N}$ et $\vec{R_P}$ où $\vec{N}$ est la normale au point P_C le plus proche du point testé P, et $\vec{R_P}$ le vecteur de position du point P_C au point P. Si le résultat est strictement négatif, le point P est à l'intérieur de la géométrie solide, sinon il est à l'extérieur.

Cette technique est relativement simple à implémenter et peut traiter des géométries impossibles à définir de manière analytique ce qui représente une importante généralisation du processus. Elle est particulièrement adaptée aux formes convexes, mais permet de traiter plusieurs géométries du domaine de l'aéronautique. La technique comporte toutefois trois désavantages importants. D'abord, elle peut causer problème pour les géométries discontinues où la normale $\vec{N}$ devient difficile à définir correctement de manière automatique (Rogers *et al.*, 2003). De telles géométries se présentent fréquemment dans le domaine aéronautique. On peut penser ici par exemple au bord de fuite d'un profil ou une géométrie comme celle du profil MDA dont les caractéristiques géométriques sont décrites dans (Liao *et al.*, 2007). Deuxièmement, la technique requiert de trouver le point P_C le plus proche du point testé, une procédure connue comme étant longue. Dernièrement, la technique nécessite de tester en détail tous les points contenus dans le cercle de recherche. Pour des géométries allongées comme un profil d'aile ou un fuselage, cet ensemble peut représenter des millions de points. Autrement dit, la technique ne permet pas de circonscrire de manière appropriée la recherche.

Pour automatiser davantage l'étape de holecutting, la troisième technique ici présentée a été utilisée au sein de Beggar (Prewitt *et al.*, 2000). L'algorithme consiste à identifier un

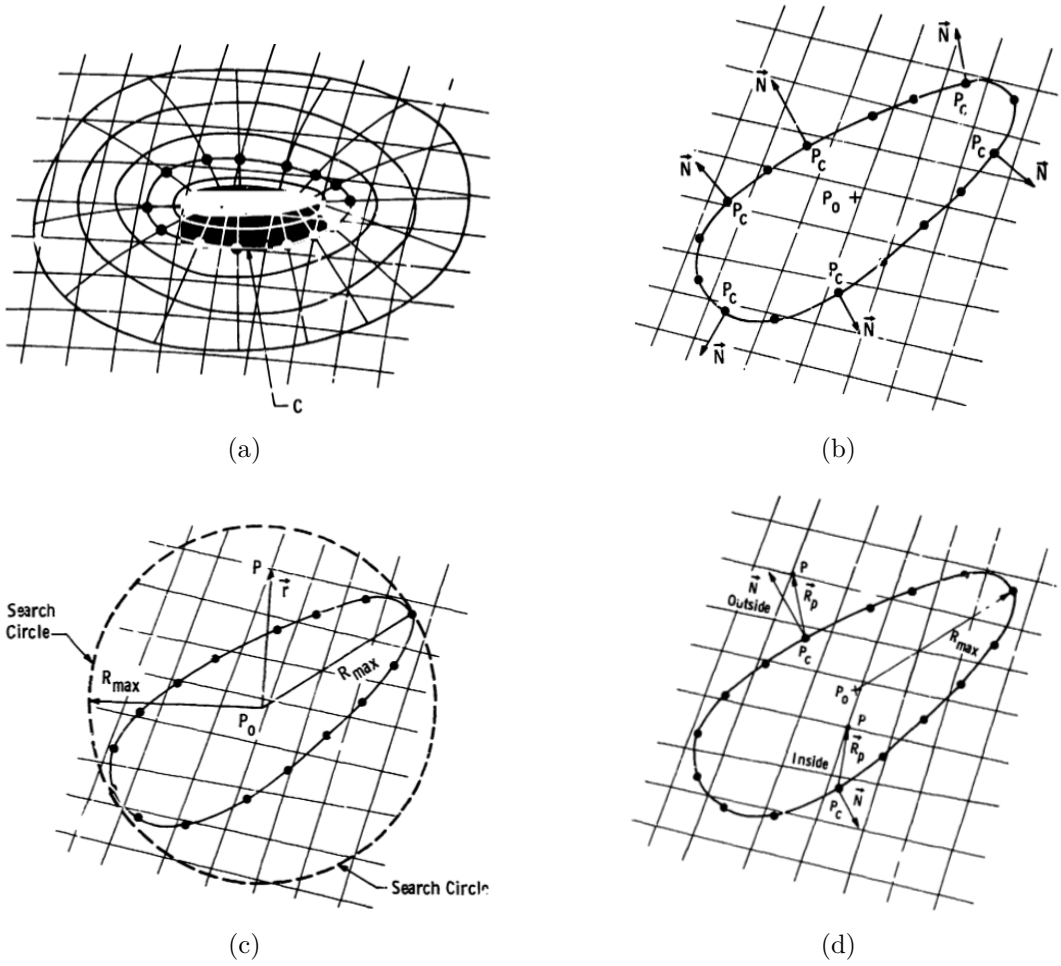


Figure 2.2 Technique de holecutting de la première version de PEGASUS (extraite de (Benek, 1986).

périmètre fermé définissant la géométrie solide pour ensuite le remplir. Dans cet algorithme, les faces de la géométrie solide sont utilisées pour définir, sur la grille coupée, le périmètre du trou. Les cellules de la grille coupée contenant les points définissant la géométrie solide doivent premièrement être identifiées. Un test de comparaison permet ensuite de déterminer si ces cellules sont à l'intérieur ou à l'extérieur du trou. Grâce à un processus récursif, le périmètre définissant la surface solide sur la grille coupée est identifié. Ce périmètre est ensuite inondé en identifiant tous les points contenus à l'intérieur de celui-ci comme des points invalides.

Cette procédure offre l'avantage significatif de pouvoir identifier les points invalides sans nécessiter d'effectuer un test sur l'ensemble des points de la grille coupée réduisant ainsi de beaucoup le temps d'exécution. De plus, la méthode est suffisamment générale pour pouvoir traiter des géométries à fortes courbures présentant même des discontinuités. En contrepartie, cette approche nécessite l'implémentation d'une structure de recherche complexe, la

cartographie polygonale, et requiert que la géométrie traitée soit fermée.

Finalement, la technique moderne offrant le plus de flexibilité et d'automatisation est l'approche «X-rays». Cette technique a démontré une très grande efficacité durant les simulations à éléments en mouvement où la procédure de holecutting doit être recommencée à chaque déplacement de la géométrie (Chan *et al.*, 2012). La technique «X-rays» étant complexe et coûteuse, elle ne répond pas à notre besoin de rapidité.

2.3 Identification des cellules interpolées et donneuses

Cette étape sert à associer les cellules réceptrices à leurs donneuses respectives. Elle va effectivement créer le lien de communication au sein du système de grille. Deux types de cellules interpolées existent : les cellules frontière et les cellules dominées. Sur la Figure 2.3, en assumant que le maillage du volet est dominant, les cellules du profil principal couvertes par celles du volet sont des cellules dominées. Aussi, puisque la grille du volet doit clairement recevoir de l'information du profil principal, les cellules rouges situées au bord externe de cette grille sont des cellules interpolées. L'identification de ces dernières est très simple à réaliser en analysant la topologie. La difficulté consiste à identifier les cellules dominées. Pour ce faire, il est nécessaire dans un premier temps de déterminer la superposition des cellules. Deux approches sont envisageables pour y parvenir. Deloze (2011) propose de décomposer une cellule en éléments basiques et de sommer les volumes créés par ces éléments. Une superposition est identifiée lorsque cette somme équivaut au volume de la cellule testée. Cette approche est simple d'implémentation, mais peut impliquer une importante quantité de tests et donc un long temps de calcul. Une alternative consiste à procéder par une méthode de gradient (Roget et Sitaraman, 2013a). Cette approche, aussi appelée «stencil-walk», créer un parcours entre un point initial et un point test et identifie une superposition grâce aux arêtes des cellules. Quoiqu'efficace, cette approche est plus complexe à implémenter surtout au sein d'un programme à mémoire partagée. Une fois la superposition établie, il est nécessaire de choisir, entre les deux cellules qui définissent doublement un même espace, celle qui sera dominante. Ce choix est réalisé en faisant intervenir un critère de dominance. La littérature montre qu'il existe trois types de ce critère.

Le premier critère utilisé pour permettre de déterminer que la cellule superposée est réceptrice est le critère hiérarchique, implémenté dans la première version de PEGASUS (Benek, 1986), qui consiste simplement à assigner un ordre hiérarchique aux différentes grilles du système. Les cellules superposées à un maillage de plus grande hiérarchie sont interpolées. Puisque ce critère n'est pas associé à une cellule en particulier, mais plutôt à un maillage entier, cette approche est très simple à implémenter et est très rapide d'exécution. Par contre, elle requiert

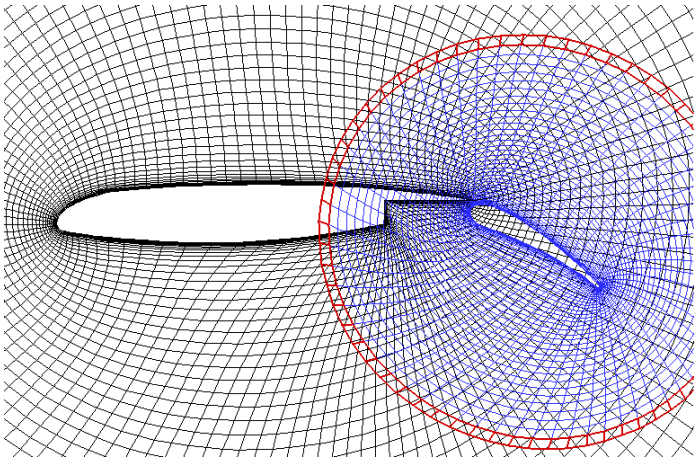


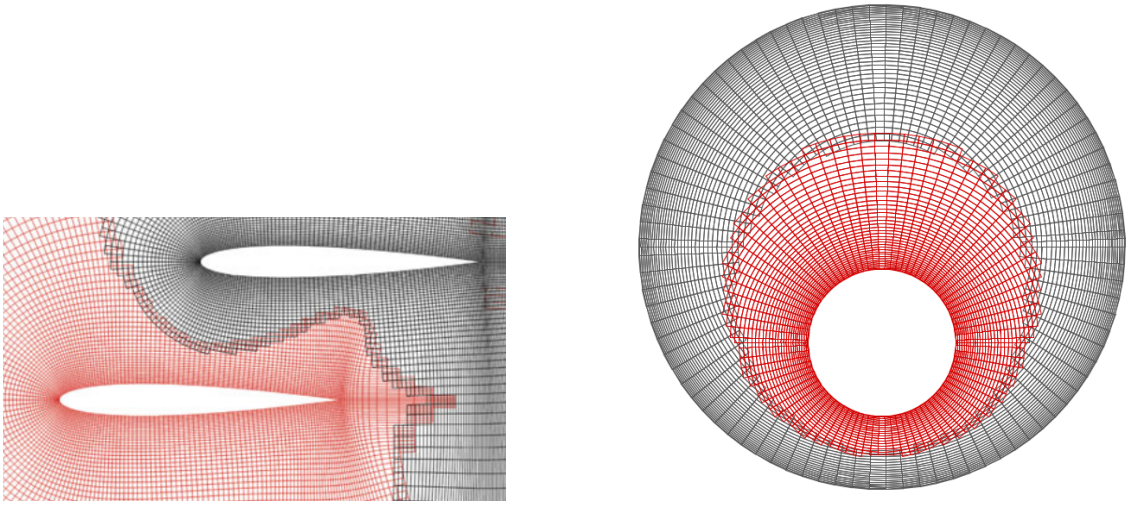
Figure 2.3 Cellules frontières

une contribution de la part de l'usager qui doit assigner les valeurs hiérarchiques à chaque grille avant l'exécution ce qui peut devenir très complexe pour un système contenant plusieurs grilles. Aussi, cette approche limite grandement les capacités d'assemblage et n'assemblera pas correctement deux grilles de hiérarchie équivalente.

Pour parer à ce problème, plusieurs ((Chandar *et al.*, 2013), (Liao *et al.*, 2007)) utilisent la propriété de résolution spatiale. L'idée consiste ici à définir les plus petites cellules, capables de résoudre l'écoulement avec plus de précision, comme les cellules dominantes et les cellules plus grandes comme réceptrices. Cette approche présente une avancée importante par rapport au critère hiérarchique puisque la flexibilité du critère permet d'assembler des grilles de hiérarchie équivalente. Les fondements de cette approche l'on amené à devenir un standard très répandu jusqu'à l'introduction récente d'un nouveau critère, décrit au paragraphe suivant.

Le critère de résolution spatiale peut mener à un assemblage des grilles douteux tel qu'illustré à la Figure 2.4(a). Ce comportement peu prévisible fortement lié à la conception du maillage et au positionnement relatif des grilles n'est pas souhaitable. Deloze (2011) a élaboré un nouveau critère basé sur la distance à la paroi. L'idée derrière cette approche veut que les cellules proches d'une géométrie solide définie par leur propre grille soient les mieux adaptées à résoudre les phénomènes physiques près des couches limites. Le critère produit en règle générale des assemblages plus uniformes que ceux du critère spatial (Figure 2.4(b)).

Pour compléter le lien de communication intergrilles, il faut déterminer les cellules qui vont fournir l'information aux cellules interpolées. Dans le cas simple où seulement deux maillages sont superposés, l'identification des cellules donneuses est relativement simple. Dans ce cas, dès qu'une cellule est interpolée, les cellules dominantes auxquelles cette cellule est superposée sont les donneuses. Le cas se complexifie lorsque plusieurs grilles sont superposées. Dans ce



(a) Critère de résolution spatiale (extraite de (Chandar *et al.*, 2013)

(b) Critère de distance à la paroi (extraite de (Deloze, 2011)

Figure 2.4 Illustration des critères de dominance

cas, il est nécessaire d'identifier les cellules superposées présentant le meilleur critère de dominance.

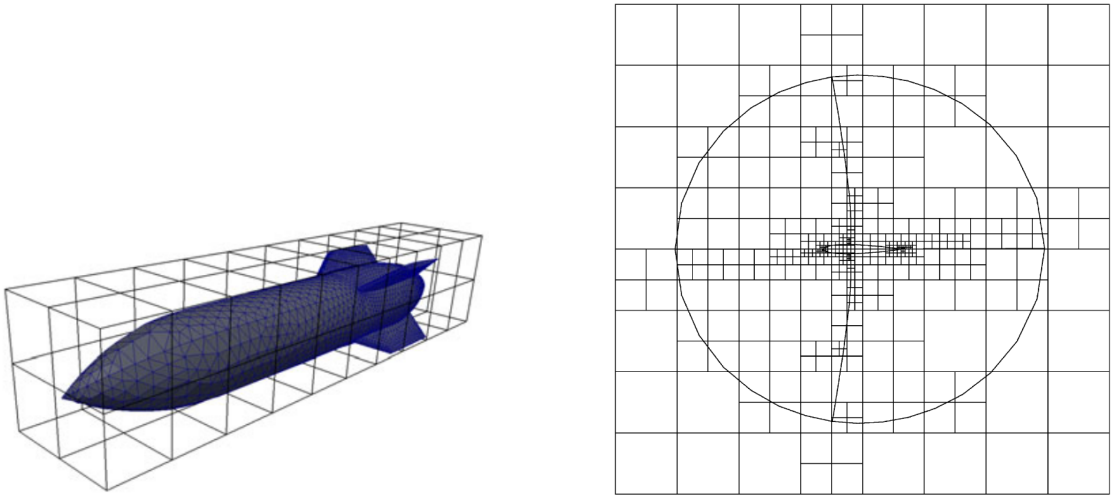
2.4 Techniques d'accélération

L'identification d'une seule cellule interpolée implique d'effectuer un test de superposition sur certaines des cellules du maillage superposé. L'approche naïve consistant à effectuer ces tests sur l'ensemble des points du maillage superposé devient rapidement trop coûteuse vu la quantité de points à tester (Soni *et al.*, 2012). Pour optimiser cette procédure, des techniques d'accélération doivent être utilisées. La technique la plus simple consiste à construire une boîte ceinturant chacun des maillages du système (Chesshire et Henshaw, 1990). Des tests de superpositions sont effectués seulement si les boîtes contours des maillages testés se superposent. La boîte peut être simplement cartésienne ou peut être orientée pour réduire son volume comme dans (Roget et Sitaraman, 2013a). Cette approche permet d'éviter les tests inutiles, mais n'accélère en rien la recherche des cellules donneuses puisque la quantité de tests effectués sur les grilles superposées n'est pas réduite.

Pour réduire le nombre de tests, deux approches ont été comparées par (Roget et Sitaraman, 2013a). La première consiste à réaliser une cartographie inverse des coordonnées physiques d'une grille ce qui permet de classer des cellules dans des partitions physiques restreintes. La deuxième utilise un arbre de recherche de type ADT (de l'anglais *Alternating Digital Tree*) qui permet de classer les points les uns par rapport aux autres. Comme les auteurs ont montré

que la technique de cartographie inverse était plus efficace, cette dernière est retenue et fait ci-bas l'objet d'une analyse plus détaillée.

La cartographie inverse consiste à trier les points de la grille physique à l'intérieur d'une grille virtuelle cartésienne. Cette grille est dite virtuelle dans le sens où elle n'a pas d'interprétation physique et n'est pas utilisée dans le solveur CFD. Elle peut être plutôt vue comme une technique de partition spatiale utilisée dans le but de grouper les points physiques d'une grille. Une version bien connue de cette approche est l'arbre de recherche «octree» (ou «quadtree» en trois dimensions) qui permet de trier des points dans une structure récursive. Plusieurs auteurs ((Zagaris *et al.*, 2010), (Deloze, 2011), (Chandar *et al.*, 2013), (Roget et Sitaraman, 2013a)) ont adapté la technique pour permettre un nombre variable de partitions dans chacune des directions (Figure 2.5(a)) et ainsi éviter de recourir à la récursivité. Prewitt *et al.* (2000) ont implémenté avec succès de manière complémentaire le «quadtree» (Figure 2.5(b)) et un arbre de cartographie polygonale («polygonal mapping tree» en anglais).



(a) Grille virtuelle englobant un missile (extraite de Zagaris *et al.* (2010))

(b) Partition de type «octree» (extraite de Prewitt *et al.* (2000))

Figure 2.5 Techniques de partition de l'espace

Une autre approche, la technique de recherche par gradient, communément appelée «stencil-walk» au sein de la communauté, est utilisée par plusieurs ((Roget et Sitaraman, 2013a), (Rogers *et al.*, 2003), (Chandar *et al.*, 2013)). Cette approche consiste à localiser les cellules interpolées et leurs donneuses en parcourant, grâce à un critère d'intersection, plusieurs donneuses potentielles à partir d'un point de départ sur lequel le gradient peut être évalué. Pour obtenir une bonne efficacité, l'algorithme nécessite que le point de départ soit suffisamment proche de la cellule donneuse. Cet algorithme est conséquemment fréquemment utilisé conjointement avec les techniques faisant usage de grilles virtuelles. Roget et Sitaraman

(2013a) présentent une revue détaillée de cet algorithme.

2.5 Parallélisation

2.5.1 Organisation de la mémoire

La parallélisation d'un code est liée à l'organisation de la mémoire. Deux types d'architecture mémoire existent du point de vue du programmeur : les mémoires distribuées et les mémoires partagées. Les ordinateurs ayant une organisation distribuée de la mémoire consistent en un nombre de processeurs appelés nœuds et d'un lien d'interconnectivité supportant l'échange d'information à travers ces nœuds. Les données d'un programme sont enregistrées dans la mémoire locale d'un ou de plusieurs de ces nœuds. La mémoire est privée, ce qui implique que seul le nœud local peut accéder à sa mémoire locale. Quand un nœud nécessite l'information de la mémoire d'un autre nœud, un transfert de données doit être effectué. Les ordinateurs ayant une organisation partagée de la mémoire consistent en un nombre de processeurs interconnectés à une seule mémoire. Des données peuvent être échangées entre les processeurs grâce à l'utilisation de variables partagées. Les mémoires partagées offrent un avantage significatif sur les mémoires distribuées puisque la communication par variables partagées est très facile (Rauber et Rünger, 2012) et évite de répliquer certaines données comme le nécessite l'architecture distribuée. Par contre, ce type d'architecture mémoire est limité à environ une douzaine de processeurs d'où la nécessité de faire appel aux mémoires distribuées pour une plus grande performance. Ces mémoires peuvent utiliser des centaines voire des milliers de processeurs.

2.5.2 Communications

Au sein d'un code programmé pour être exécuté sur un ordinateur à mémoire partagée, les communications entre les processeurs sont aujourd'hui entièrement cachées par des interfaces simples. Au contraire, pour être exécuté sur un ordinateur à mémoire partagée, un programme doit détailler explicitement l'ensemble des communications nécessaires au bon déroulement de celui-ci. Cette communication est établie grâce à l'utilisation d'une interface de communication. L'interface aujourd'hui le plus répandu est MPI (Message Passing Interface) (Barney, 2014). Cette interface, par sa standardisation, présente l'avantage d'être supporté sur pratiquement toutes les plateformes de calcul de hautes performances en plus d'offrir une excellente portabilité. L'interface fournit de plus une vaste gamme de fonctionnalités implémentées de manière efficace. Pour ces raisons la discussion suivante est centrée sur l'utilisation de ce standard.

À travers MPI, les communications entre processeurs peuvent être soit synchronisées ou asynchrones. Les communications synchronisées, comme le nom l'indique, sont des communications devant être réalisées de manière simultanée par les processeurs impliqués. Inversement, les communications asynchrones permettent la poursuite du programme même si la communication n'est pas établie. Dans le cadre de l'exécution parallèle d'un assembleur de grilles, les communications asynchrones sont utilisées puisqu'elles offrent un meilleur gain en performance (Wissink et Meakin, 1997).

2.5.3 Gains en performance

L'accélération d'un programme exécuté en parallèle est limitée par les portions séquentielles de celui-ci. La *scalabilité* d'un programme représente sa faculté à maintenir une performance adéquate lorsqu'il est exécuté en parallèle sur plusieurs processeurs. Comme indiqué dans (Chevance, 2001) «Le terme scalabilité est la francisation du terme anglais *scalability*. Scalability dérive du verbe *to scale* qui signifie augmenter ou réduire en accord avec un certain ratio. Il n'y a pas d'équivalent français communément admis». Le terme scalabilité sera donc utilisé dans le présent document. Pour obtenir une bonne scalabilité, les tâches du programme doivent être exécutées de manière simultanée. On peut ainsi exécuter parallèlement des tâches de natures différentes, communément appelé la parallélisation fonctionnelle, ou réaliser des tâches identiques, aussi appelé la parallélisation géométrique. Les codes d'assemblage de grilles superposées sont performants lorsque parallélisés géométriquement ((Wissink et Meakin, 1997), (Prewitt *et al.*, 2000), (Zagaris *et al.*, 2010)).

2.5.4 Balancement de charge

Un premier aspect à considérer pour atteindre une bonne scalabilité est la répartition de la charge à travers les processeurs ce qui implique la division uniforme du problème à l'étude. Deux types de division sont répandus : les divisions grossière et fine. La division grossière consiste à décomposer le problème selon les blocs de la grille. Cette approche est relativement simple à utiliser, mais requiert une attention particulière de la part du concepteur de maillage vis-à-vis l'exécution parallèle. En effet, les maillages générés doivent être découpés pour permettre leur répartition uniforme sur plusieurs processeurs. La division fine consiste en l'étape supplémentaire de permettre des subdivisions des blocs de la grille à l'intérieur même du programme d'assemblage de manière à offrir une répartition encore plus optimale (Zagaris *et al.*, 2010). Cette approche est évidemment plus complexe à implémenter, mais offre davantage de flexibilité. Wissink et Meakin (1997) présentent une méthode relativement simple et efficace de balancement de charge basée sur la division fine. L'approche vise à

subdiviser les grilles de manière à réduire au maximum les communications entre les grilles.

Le problème de répartition de charge est difficile à traiter pour trois raisons principales (Zagaris *et al.*, 2010) :

1. La répartition de la charge doit prendre en compte le solveur du fluide. Comme ce dernier requiert plus de temps de calcul que l'assemblage chimère, la division du problème est généralement dictée par les requis du solveur.
2. L'algorithme d'assemblage chimère comporte plusieurs phases, chacune caractérisée par une charge différente ce qui rend complexe l'élaboration d'une stratégie de répartition de charge globale.
3. Lors de simulations à éléments en mouvement, la charge est dynamiquement modifiée ce qui peut réduire la scalabilité. Pour garder une charge balancée, plusieurs auteurs font aujourd'hui usage d'algorithmes de répartition de charge dynamique ((Prewitt *et al.*, 2000), (Zagaris *et al.*, 2010), (Wissink et Meakin, 1997)).

CHAPITRE 3 MÉTHODE CHIMÈRE BIDIMENSIONNELLE

3.1 Solveur NSCODE

NSCODE est un logiciel CFD bidimensionnel utilisé à des fins de recherche dont le développement a été initié par le professeur Éric Laurendeau en 2012. Le logiciel est utilisé comme plateforme pour concevoir, développer, implémenter et tester diverses techniques numériques à l'état de l'art appliquées au domaine de l'aéronautique. Dans une perspective d'alignement technologique, les développements de NSCODE sont intimement liés à la structure du code CFD de Bombardier Aéronautique, FANSC. À ce titre, les deux codes sont basés sur des maillages structurés, utilisent des techniques de résolution similaires et sont écrits dans le même langage de programmation, le C.

Les fonctionnalités de NSCODE sont multiples. Dans un souci de concision, seuls les aspects directement liés aux travaux présentés dans ce document seront détaillés. Pour une description plus complète des différentes fonctionnalités de NSCODE, le lecteur est référé à Pigeon *et al.* (2014).

3.1.1 Équations

La physique des fluides est décrite par les équations de Navier-Stokes qui expriment les lois de conservations de masse, de la quantité de mouvement et d'énergie (Blazek, 2001). Les trois lois de conservation sont exprimées sous leur forme intégrale sur un volume de contrôle Ω de la manière suivante :

L'équation de bilan de masse :

$$\frac{\partial}{\partial t}\int_{\Omega}\rho d\Omega + \oint_{\partial\Omega}\rho\left(\vec{v}\cdot\vec{n}\right)dS = 0 \tag{3.1}$$

L'équation de bilan de quantité de mouvement :

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\vec{v}d\Omega + \oint_{\partial\Omega}\rho\vec{v}(\vec{v}\cdot\vec{n})dS = \int_{\Omega}\rho\vec{f_e}d\Omega - \oint_{\partial\Omega}p\vec{n}dS + \oint_{\partial\Omega}\left(\vec{\vec{\tau}}\cdot\vec{n}\right)dS \tag{3.2}$$

L'équation de bilan d'énergie :

$$\frac{\partial}{\partial t}\int_{\Omega}\rho E d\Omega + \oint_{\partial\Omega}\rho H(\vec{v}\cdot\vec{n})dS = \oint_{\partial\Omega}k(\nabla T\cdot\vec{n})dS + \int_{\Omega}\left(\rho\vec{f_e}\cdot\vec{v}+\dot{q}_h\right)d\Omega + \oint_{\partial\Omega}\left(\vec{\vec{\tau}}\cdot\vec{v}\right)\cdot\vec{n}dS \tag{3.3}$$

— ρ est la densité du fluide ;
— $\vec{v}$ est le vecteur de vitesse du fluide ;
— $\vec{n}$ est le vecteur normal à la face dS du volume de contrôle ;
— $\vec{f_e}$ est le vecteur des forces appliquées au fluide ;
— p est la pression ;
— $\vec{\vec{\tau}}$ est le tenseur de contraintes visqueuses ;
— E est l'énergie totale par unité de masse du fluide ;
— H est l'enthalpie totale par unité de masse du fluide ;
— k le coefficient de conductivité thermique du fluide ;
— T la température statique absolue ;
— $\dot{q}_h$ est le flux de chaleur par unité de masse ;

En réorganisant les équations 3.1, 3.2 et 3.3, on obtient la formulation suivante :

$$\frac{\partial}{\partial t}\int_\Omega \vec{W} d\Omega + \oint_{\partial\Omega} (\vec{F_C} - \vec{F_V}) dS = \int_\Omega \vec{Q} d\Omega \tag{3.4}$$

où $\vec{W}$ représente le vecteur des variables conservatrices, $\vec{F_C}$ les flux convectifs, $\vec{F_V}$ les flux visqueux et $\vec{Q}$ les termes sources à l'intérieur d'un volume fini Ω. NSCODE résout ces équations en deux dimensions en ne considérant aucun terme source, ce qui annule le terme de droite. Les flux convectifs et visqueux sont évalués à travers l'ensemble des surfaces dS du volume. Les variables conservatrices et les flux sont détaillés ainsi :

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad \vec{F_C} = \begin{bmatrix} \rho V_n \\ \rho u_n V + n_x p \\ \rho v V_n + n_y p \\ \rho H V_n \end{bmatrix} \quad \vec{F_V} = \begin{bmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} \\ \eta_x \tau_{yx} + \eta_y \tau_{yy} \\ \eta_x \Theta_x + \eta_y \Theta_y \end{bmatrix},$$

où

$$\Theta_i = u\tau_{ix} + v\tau_{iy} + k\frac{\partial T}{\partial t} \tag{3.5}$$

avec

— u et v sont les vitesses du fluide dans chacune des deux dimensions ;
— V_n est le vecteur de vitesse normal à la surface dS de l'élément ;

3.1.2 Algorithme de résolution

En discrétisant spatialement l'équation 3.4 on obtient la forme suivante exprimant les lois de conservation sur un volume de contrôle dénoté par i,j et k de volume V :

$$V_{i,j,k}\frac{\partial \vec{W}_{i,j,k}}{\partial t} + \sum_{m=1}^{4}(\vec{F_c} - \vec{F_v})_m \Delta S_m = 0 \tag{3.6}$$

Le deuxième terme, une fonction non linéaire des variables conservatrices, est appelé le résidu. Pour le cas stationnaire, il consiste en la somme des flux à travers toutes les faces du volume de contrôle. Pour les simulations instationnaires, NSCODE utilise une discrétisation temporelle basée sur la méthode d'avancement à pas de temps dual. Le système 3.6 est résolu via une procédure itérative basée sur un pseudo pas de temps (de l'anglais «pseudo time step»). Trois algorithmes de résolution sont disponibles :

— le schéma explicite Runge-Kutta tel qu'implémenté par Jameson *et al.* (1981) ;
— le schéma implicite par point tel qu'implémenté par Pierce et Giles (1997) ;
— le schéma implicite LUSGS tel qu'implémenté par Cagnone *et al.* (2011) ;

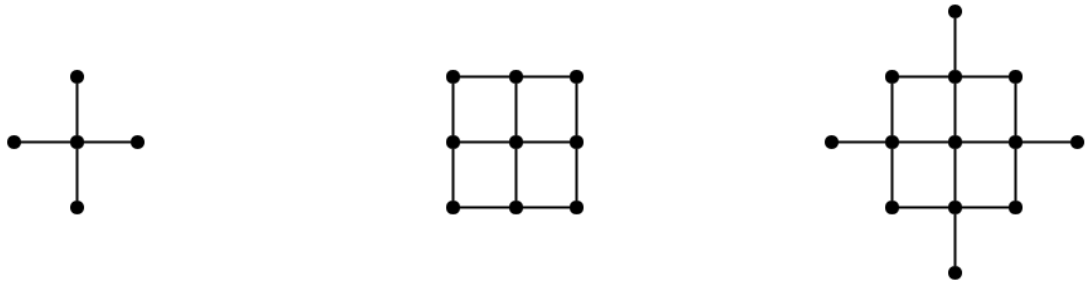
Ces algorithmes de résolution sont accélérés grâce aux techniques de multigrille (Blazek, 2001) et de lissage des résidus (Jameson et Baker, 1983).

3.1.3 Discrétisation spatiale

NSCODE utilise un schéma centré pour effectuer le calcul des flux. Les volumes de contrôle sont donc identiques aux cellules formées par le maillage et les variables représentant l'écoulement sont enregistrées aux centres des cellules. Le concept du schéma centré consiste à calculer les flux convectifs à une face du volume de contrôle à partir de la moyenne arithmétique des variables conservatrices situées de chaque côté de la face. Pour pouvoir calculer les flux sur les quatre faces constituant le volume de contrôle, le schéma nécessite l'information des cellules voisines immédiates de la cellule où les flux sont calculés tel que démontré à la Figure 3.1(a). Cet ensemble de cellules impliquées dans le calcul des flux est appelé molécule de calcul. Pour pouvoir calculer les flux dissipatifs, l'évaluation de la seconde dérivée implique une molécule telle que montrée à la figure 3.1(b).

Puisque la formulation mathématique du schéma centré permet un découplage de la solution (Blazek, 2001), l'ajout d'un flux artificiel dissipatif est nécessaire. Deux formulations du flux dissipatif sont implémentées dans NSCODE, le schéma de dissipation scalaire Jameson *et al.* (1981) et le schéma de dissipation matriciel Swanson et Turkel (1992). Ces schémas requièrent deux données de chaque côté de la face où le flux dissipatif est additionné ce qui étend la

molécule tel que montré à la Figure 3.1(c).



(a) (b) (c)

Figure 3.1 Molécules de calcul des flux

3.1.4 Conditions limites

La molécule montrée à la figure 3.1(c) implique que chacune des cellules doit avoir deux voisins. Comme cette condition n'est pas respectée au bord du domaine, là où les conditions limites sont appliquées, NSCODE fait usage de deux rangées de cellules fictives ajoutées au bord du maillage physique. Ces deux rangées sont communément appelées les *halos* puisqu'elles ceinturent le domaine. Les conditions limites sont appliquées durant l'étape de résolution en manipulant les données contenues dans les halos. Durant la procédure de résolution itérative, les valeurs enregistrées dans les halos sont fréquemment mises à jour pour correctement imposer les conditions limites. NSCODE supporte trois types de conditions limites :

— condition de paroi
— condition de symétrie
— condition de champ éloigné
— condition de connexion

3.1.5 Caractéristiques topologiques

NSCODE a la capacité d'effectuer des simulations sur des maillages multiblocs. Ces types de maillages décomposent le domaine de simulation de manière à pouvoir mieux s'adapter à la géométrie étudiée. Ces décompositions entraînent l'utilisation de la condition limite de connexion. Comme son nom l'indique, la condition de connexion permet de relier numériquement deux blocs contigus.

3.2 Fonctions d'interpolation

L'implémentation de la méthode d'assemblage chimère requiert l'utilisation d'interpolation durant deux phases de la méthode, soit l'identification des cellules interpolées et l'identification des cellules donneuses. Pour mieux organiser la discussion de ces deux phases, les deux fonctions d'interpolation utilisées sont préalablement décrites ici.

3.2.1 Interpolation tétravolumique

Comme son nom l'indique, l'interpolation tétravolumique utilise des volumes pour définir les poids d'interpolation. Tel que montré à la Figure 3.2, l'interpolation est basée sur trois valeurs, soit celles des trois cellules formant un volume capable de contenir le point d'interpolation Q. Le volume identifié n'est pas nécessairement le plus petit, mais bien le premier identifié par la procédure de test. Le poids assigné à une valeur d'interpolation est fonction du volume opposé. La valeur interpolée v_Q est calculée de la manière suivante :

$$v_Q = \sum_{i=1}^{3} \frac{V_i}{V_{tot}} v_i \tag{3.7}$$

où en faisant référence à la Figure 3.2

$$V_1 = V_{B2,B4,Q} \qquad V_2 = V_{B1,B4,Q} \qquad V_4 = V_{B1,B2,Q} \tag{3.8}$$

et V_{tot} est la somme des trois volumes et v_i est la valeur au point B_i.

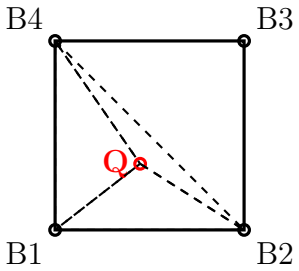


Figure 3.2 Volume utilisé pour l'interpolation tétravolumique

La méthode est intéressante pour sa simplicité d'implémentation et son temps de calcul minime. Par contre, comme il sera montré à la section 4.4.6, d'autres fonctions d'interpolation offrent une meilleure précision.

3.2.2 Interpolation bilinéaire

L'interpolation bilinéaire, tirée de (Zagaris *et al.*, 2010), consiste à faire usage des coordonnées naturelles. Cette approche définie un système de coordonnées local, $\vec{\Phi_B} = (\xi_B, \eta_B)$, en fonction de la cellule évaluée qui constitue l'ensemble B dans la présente analyse. Au sein du système créé, la variation des bases ξ_B et η_B est limitée à [0,1]. Pour effectuer l'interpolation, on doit déterminer les coordonnées ξ_B et η_B du point d'interpolation ce qui implique une transformation des coordonnées cartésienne vers les coordonnées naturelles de la cellule. Les coordonnées cartésiennes sont liées aux coordonnées naturelles par la relation suivante :

$$F(\vec{\Phi_B}) = \begin{bmatrix} x_Q(\xi_B, \eta_B) \\ y_Q(\xi_B, \eta_B) \end{bmatrix} = \sum_{i=1}^{4} N_i(\xi_B, \eta_B)\vec{Bi} \tag{3.9}$$

où $\vec{Bi}$ est le vecteur des coordonnées des cellules de l'ensemble B et les coefficients N_i sont définis par :

$$N_1 = (1-\xi_B)(1-\eta_B) \qquad N_2 = \xi_B(1-\eta_B) \qquad N_3 = (1-\xi_B)\eta_B \qquad N_4 = \xi_B\eta_B \tag{3.10}$$

La transformation inverse qui permet de déterminer les coordonnées ξ_B et η_B d'un point test implique de résoudre le système 3.9 via des techniques numériques. Une technique commune et efficace consiste à utiliser la méthode de Newton-Raphson. En définissant $\vec{Q}$ comme le vecteur des coordonnées du point testé, on minimise la fonction $F(\vec{\Phi_B}) - \vec{Q}$ grâce à la procédure itérative suivante :

$$\vec{\Phi_B}^{\,m+1} = \vec{\Phi_B}^{\,m} - [\mathbf{J}^m]^{-1}\left(F\left(\vec{\Phi_B}^{\,m}\right) - \vec{Q}\ \right) \tag{3.11}$$

où m est le compteur d'itération. La matrice jacobienne du système 3.9 est définie comme :

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x_Q}{\partial \xi_B} & \dfrac{\partial x_Q}{\partial \eta_B} \\ \dfrac{\partial y_Q}{\partial \xi_B} & \dfrac{\partial y_Q}{\partial \eta_B} \end{bmatrix} \tag{3.12}$$

La méthode Newton-Raphson converge quadratiquement si un bon point de départ est fourni. La valeur $\xi_B = \eta_B = 0.5$ correspondant au centre paramétrique de la cellule est utilisé comme valeur initiale.

Comme il sera montré à la section 4.4.6, l'interpolation bilinéaire a l'avantage de fournir

une meilleure précision que l'interpolation tétravolumique (voir section 4.4.6). Par contre, son implémentation est plus complexe et son utilisation requiert un processus itératif ce qui nécessite plus de temps de calcul.

3.3 Assemblage des grilles

L'assemblage des grilles est une procédure impliquant plusieurs comparaisons entre les données et diverses manipulations et modifications de celles-ci. Pour faciliter la communication de l'implémentation, la procédure qui est détaillée dans cette section est schématisée à la Figure 3.3. Ce schéma montre comment obtenir, pour chaque cellule, soit le statut *interpolée* (section en vert), *invalide* (en rouge) ou *calculé*. Une cellule est calculée si elle atteint la sortie du diagramme sans passer par les sections vertes ou bleues.

3.3.1 Groupes chimère

Comme il sera expliqué dans les prochaines sections, plusieurs portions de l'algorithme consistent à effectuer une analyse comparative entre des maillages appartenant à deux géométries distinctes. Ainsi, on définit un *groupe chimère* comme l'ensemble formé par des blocs reliés de manière contigüe. La Figure 3.4 illustre cette définition (les maillages des blocs orange et bleus sont masqués pour ne pas encombrer la visualisation). Les trois blocs orange associés au profil, les deux blocs bleus près du profil et le maillage de fond forment trois groupes chimère distincts.

L'identification des groupes chimère est automatisée dans l'implémentation bidimensionnelle. Cette automatisation est facilement réalisable puisque les coordonnées d'une grille contigüe sont enregistrées dans un même fichier. Ainsi, lors de la lecture du fichier de maillage, l'ensemble des blocs décrits au sein du fichier consiste en un bloc chimère. Ceci permet une identification simple des groupes chimère mais restreint en contrepartie l'organisation des données.

3.3.2 Condition limite chimère

Une nouvelle condition limite est introduite par l'utilisation de grilles superposées. En effet, les cellules situées en bordure de grilles devant recevoir l'information d'autres grilles sont identifiées par la condition limite de type chimère. La Figure 3.5 montre ces conditions limites sur le même système de grille que la Figure 3.4.

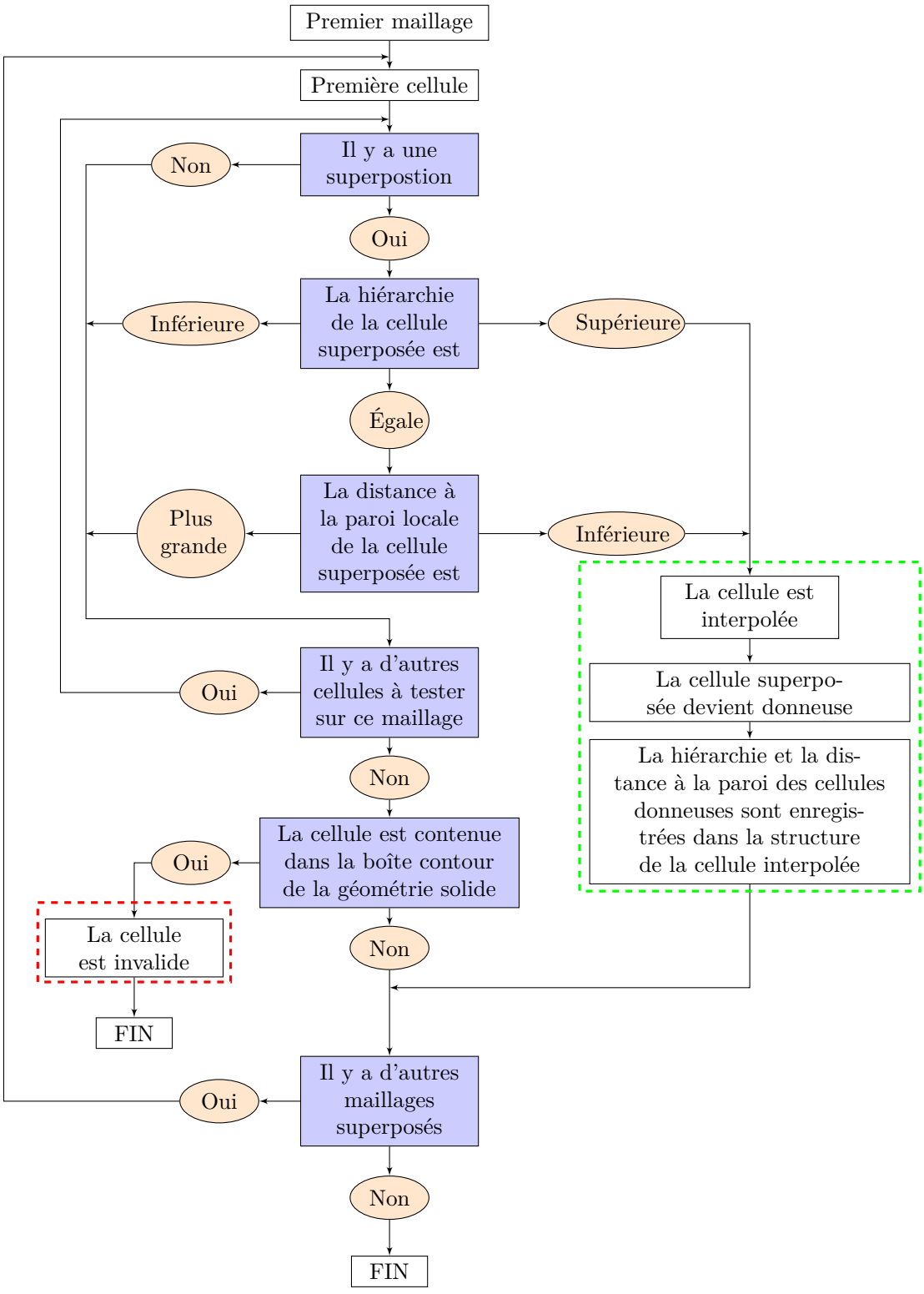


Figure 3.3 Arbre de décision représentant le processus d'assemblage des grilles

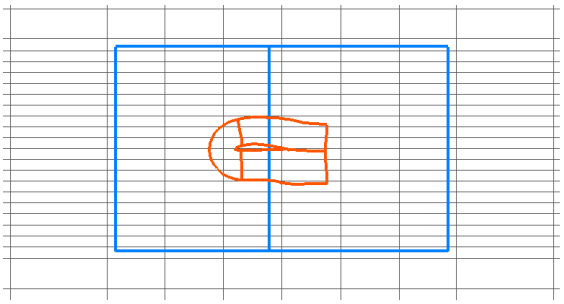


Figure 3.4 Trois groupes chimère

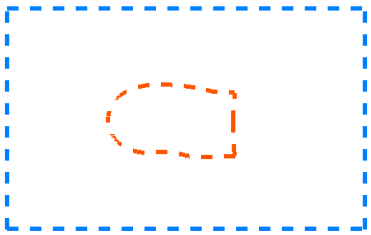


Figure 3.5 Conditions limites chimères

3.3.3 Identification des superpositions

La première étape de l'assemblage du système de grilles consiste à identifier les superpositions de cellules. Cette identification est nécessaire pour déterminer quels maillages seront utilisés pour résoudre l'écoulement et quelles cellules seront réceptrices des données.

Définition d'une superposition

La superposition d'une cellule d'un maillage A sur un maillage B se produit quand le centre de la cellule du maillage A est contenu à l'intérieur du quadrilatère formé par le centre de quatre cellules adjacentes du maillage B. La Figure 3.6(a) illustre cette définition. Dans cette figure où le point Q est superposée aux quatre cellules $B1$, $B2$, $B3$ et $B4$, les lignes tiretées représentent le maillage et les points vides, le centre des cellules. Pour clarifier la discussion suivante, quand il sera question de superposition, «la cellule superposée» réfèrera à Q et «les cellules adjacentes» à B définit comme l'ensemble contenant $B1$, $B2$, $B3$ et $B4$.

Un traitement spécial doit être fait pour détecter de manière appropriée les superpositions situées entre le centre des cellules au bord d'un maillage et les points qui définissent le périmètre de ce maillage. Cette situation est issue du fait que l'ensemble des quadrilatères formés par le centre des cellules adjacentes d'un maillage ne recouvre pas entièrement ce maillage. Pour y parer, on enregistre dans la première rangée de cellules fictives la définition du contour du maillage tel que montré à la Figure 3.6(b) où les points spéciaux, en bleu, sont obtenus par la moyenne des deux points de la grille. La Figure 3.6(b) illustre la définition sur une grille orthogonale mais cette définition n'est pas affectée si la grille n'est pas orthogonale. En effet, puisque les points spéciaux sont obtenus à partir du maillage, une modification de celui-ci entraîne aussi une modification des points spéciaux.

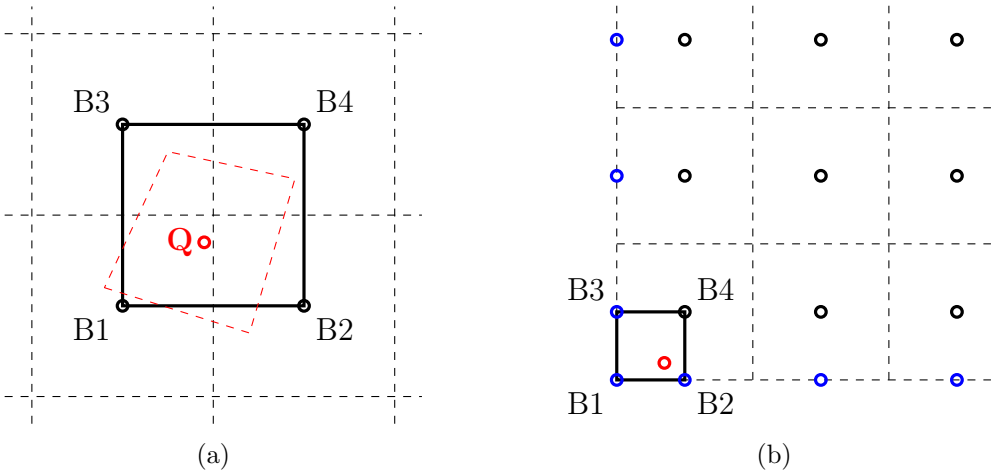


Figure 3.6 - (a) Définition d'une superposition (b) Traitement des centres de cellules au bord du maillage

Détection d'une superposition

L'approche utilisée pour effectuer la détection des superpositions est illustrée à la Figure 3.7. La technique, simple à conceptualiser, consiste à effectuer un test de comparaison d'aire. L'aire du quadrilatère formé par les cellules adjacentes est calculée puis comparée à la somme des aires des quatre triangles formés par une paire de points adjacents et le point Q sur lequel le test est effectué. Lorsque la différence des deux aires est en deçà d'une tolérance de 1×10^{-12}, une superposition est identifiée. Cette approche offre l'avantage d'être générale, rapide et simple à implémenter.

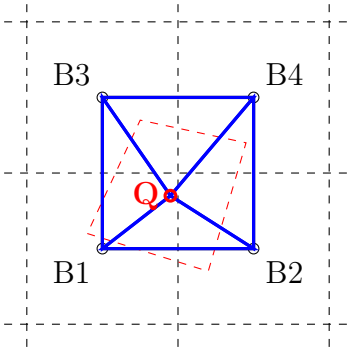


Figure 3.7 Détection d'une superposition

3.3.4 Identification des cellules interpolées

Une fois les superpositions identifiées, il faut déterminer, parmi les cellules présentant une superposition, celles qui sont réceptrices pour construire le lien de communication au sein du système de grille. Comme il a été montré dans la section 2, trois critères ont été développés pour identifier ces cellules.

Le critère primaire utilisé au sein de la présente implémentation est celui de la hiérarchie. L'utilisation de ce critère implique un certain travail de la part de l'usager pour hiérarchiser les grilles. Par contre, considérant que l'implémentation est au stade de démonstration, l'outil développé se veut donner une forme de contrôle à l'utilisateur pour permettre d'effectuer divers tests. Ceci justifie l'utilisation du critère hiérarchique.

Le deuxième critère de dominance choisi est la distance à la paroi locale. Ce critère est basé sur la distance à la paroi à l'intérieur d'un groupe chimère. Une cellule superposée est identifiée comme interpolée si cette distance est supérieure à celle des cellules auxquelles elle est superposée. Cette approche nécessite de calculer la distance à la paroi locale sur l'ensemble des groupes chimère du système de grille. Le calcul est effectué pour chacune des cellules en testant l'ensemble des points définissant une géométrie solide de manière à trouver la valeur minimale. Cette procédure est longue et, quoique plusieurs techniques efficaces existent, cet aspect ne fait pas l'objet des présents travaux. Ce calcul ne sera donc pas optimisé. Le lecteur intéressé à l'accélération de ce calcul est référé à l'article suivant (Roget et Sitaraman, 2013b).

Le critère de résolution spatiale a aussi été implémenté, mais n'est pas utilisé puisqu'il impose des contraintes lors du processus de génération de maillages. En effet, pour que ce critère génère un assemblage adéquat, la conception du maillage nécessite de générer de petites cellules aux endroits où l'on veut que le maillage en question soit dominant. Cette pratique est opposée aux concepts fondamentaux de la méthode chimère qui vise à réduire les contraintes appliquées au maillage. Contrairement au critère hiérarchique, le critère de la distance à la paroi implique une comparaison de la valeur contenue dans la cellule superposée et celles des cellules de l'ensemble B. La valeur de la distance à la paroi sur le maillage contenant les cellules adjacentes est obtenue en effectuant une interpolation à l'endroit de Q. Puisque la détermination des cellules interpolées n'est pas critique à l'ordre de précision des méthodes numériques du solveur, la précision des fonctions d'interpolation ne constitue pas une référence sur laquelle baser le choix. Puisque les deux fonctions d'interpolation ont montré à travers des tests qu'elles identifiaient les mêmes cellules interpolées, la fonction d'interpolation tétravolumique est retenue pour son faible temps de calcul.

3.3.5 Identification des cellules donneuses

Pour que l'information puisse être correctement communiquée d'une grille à l'autre, les cellules interpolées précédemment identifiées doivent être associées à des cellules dominantes qui seront leurs donneuses. Les cellules donneuses sont définies comme l'ensemble de quatre cellules adjacentes dominantes contenant la cellule interpolée.

Au moment de déterminer les cellules interpolées, les cellules dominantes n'ont pas encore été identifiées. Ainsi pour déterminer correctement les cellules donneuses, la cellule interpolée doit identifier les superpositions avec les cellules de toutes les grilles du système et effectuer un test de dominance avec celles-ci. Dès qu'une cellule est interpolée, elle enregistre la hiérarchie et la distance à la paroi locale de ses potentielles donneuses. Cette partie de l'algorithme correspond à l'encadré vert de la Figure 3.3. En suivant le diagramme, on voit que même après avoir été identifiée comme interpolée, la cellule en question poursuit les tests avec les autres grilles superposées. Ces tests seront désormais faits avec la hiérarchie et la distance à la paroi enregistrée. Cette approche garantit que les donneuses soient des cellules dominantes.

Pour que l'ordre du schéma numérique utilisé dans le solveur soit préservé à travers l'utilisation de la technique chimère, la fonction d'interpolation choisie doit être d'ordre équivalent au dit schéma (Chesshire et Henshaw, 1990). De ce fait, les interpolations tétravolumique et bilinéaire peuvent être utilisées pour calculer les poids d'interpolation des cellules donneuses. Une démonstration détaillée de l'ordre de précision des deux techniques est traitée dans la section résultat du présent chapitre.

3.3.6 Identification des cellules invalides

Les techniques de holecutting présentées à la section 2 ne sont pas adaptées aux applications visées par les présents travaux. La technique analytique n'est pas assez généralisée pour l'application considérée, et les algorithmes de remplissage utilisé dans Beggar et X-rays ne répondent pas aux attentes en termes de complexité d'implémentation. La technique implémentée dans la première version de PEGASUS est simple, mais nécessite trop de temps pour les applications industrielles visées. Pour ces raisons et pour sa simplicité et sa précision, l'algorithme présenté dans (Chandar *et al.*, 2013) sera utilisé.

Contrairement aux techniques communes qui tentent d'identifier les cellules invalides, l'algorithme proposé fonctionne par élimination en exploitant la présence de la grille définissant la géométrie solide. La procédure consiste d'abord à générer une boîte cartésienne ceinturant la géométrie solide analysée. Par la suite, les cellules invalides sont facilement identifiées comme celles étant contenues dans la boîte en question et ne détectant aucune superposition avec

le maillage qui défini la géométrie solide. En se référant à la Figure 3.3, on voit que le processus de recherche de superposition est interrompu lorsqu'une cellule est identifiée comme invalide. De plus, des cellules précédemment identifiées comme interpolées peuvent subir un changement de statut et devenir invalides.

Cette approche offre l'avantage d'être simple à conceptualiser et à implémenter. Typiquement, les étapes de holecutting et d'identification de superpositions sont distinctes au sein des codes d'assemblage chimère. La formulation de l'approche utilisée permet d'exploiter la procédure de détection de superposition de manière à combiner ces deux étapes en une seule ce qui permet de réduire le temps requis pour assembler les grilles. La capacité de traiter des géométries complexes présentant des discontinuités rend cette approche très bien adaptée aux géométries du domaine aéronautique et sa simplicité s'insère parfaitement dans le cadre d'une démonstration technologique. Malgré ces nombreux avantages, son utilisation requiert certaines considérations lors de l'étape de génération des maillages. Celles-ci seront abordées lors de la présentation des résultats.

3.3.7 Méthode d'accélération

Deux méthodes de recherche ont été couvertes dans la revue de littérature, soit les grilles virtuelles et la technique de recherche par gradient. Cette dernière technique a montré une bonne efficacité, mais est complexe d'implémentation, surtout dans une perspective d'exécution parallèle (Roget et Sitaraman, 2013a). Au contraire, les techniques utilisant des grilles virtuelles se prêtent facilement au parallélisme. Pour ces raisons et dans le but de circonscrire la démarche académique, seules les techniques de grilles virtuelles sont explorées dans ce document.

Une grille virtuelle cartésienne est premièrement générée pour chaque bloc du système de grille. L'ensemble des cellules formées de quatre points adjacents, tel $B1$, $B2$, $B3$ et $B4$ de la Figure 3.6(a), sont ensuite classées dans les divisions virtuelles formées par la grille. Pour clarifier le reste de la discussion, ces divisions virtuelles seront appelées *boîtes* de par leur fonction. Le classement d'une cellule doit être fait dans toutes les boîtes superposées par celle-ci. La manière la plus simple d'y parvenir est de créer une enveloppe cartésienne autour de la cellule à classer et d'enregistrer la cellule dans toutes les boîtes superposées par cette enveloppe telle qu'illustrée à la Figure 3.8. Cette façon de procéder est simple et rapide mais implique qu'une cellule puisse être classée dans une boîte où elle ne devrait pas. Ce classement inapproprié n'est en aucun cas problématique, il ne fait que légèrement augmenter le coût en mémoire.

La technique d'accélération est très sensible au nombre de boîtes de la grille virtuelle. En

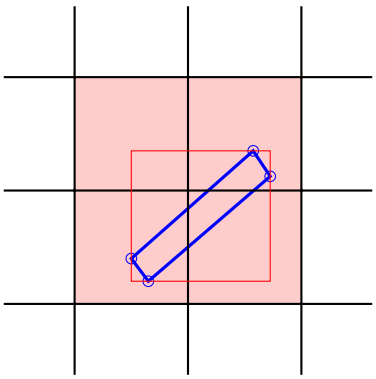


Figure 3.8 Classification d'une cellule physique (bleu) dans la grille virtuelle (noire)

définissant N_x et N_y comme le nombre de divisions dans les directions x et y, l'approche consistant à spécifier N_x et N_y séparément offre beaucoup de flexibilité pour optimiser la grille. En fait, elle offre tellement de flexibilité qu'il est difficile de déterminer que les paramètres N_x et N_y choisis sont vraiment optimaux. En outre, les paramètres ne sont pas basés sur les caractéristiques physiques du maillage ce qui implique un réajustement de ceux-ci lorsque le maillage est densifié, comme dans une analyse de convergence de grille. La présente implémentation fait plutôt usage d'un seul critère basé sur les caractéristiques du maillage, soit le nombre moyen de points par boîte. Ceci présente l'avantage significatif de réduire le temps nécessaire à l'optimisation de la grille virtuelle en plus de s'adapter automatiquement aux changements de densité.

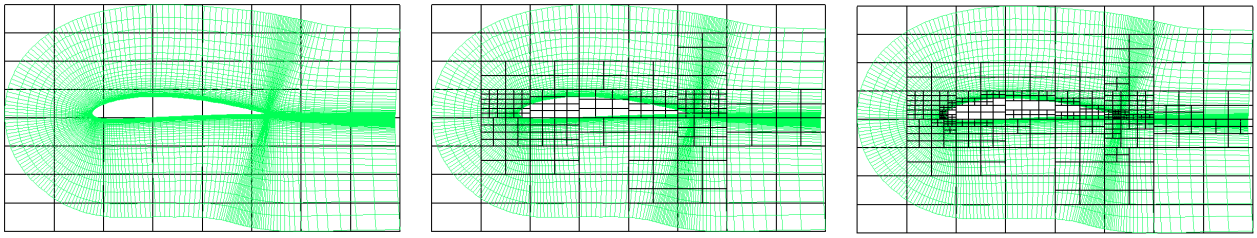
Le paramètre utilisé représente le nombre moyen de points par boîte. Ainsi le nombre de division N de la grille virtuelle est obtenue comme suit :

$$N_x = N_y = \sqrt{\frac{Npts}{\beta}} \tag{3.13}$$

où $Npts$ est le nombre de points du maillage et β est le nombre moyen de points désiré.

La technique de la grille virtuelle est essentielle pour obtenir une performance raisonnable en termes de temps. Le fait de partitionner spatialement les données permet de réduire radicalement le nombre de tests de superposition nécessaires à un assemblage adéquat. Plus une grille virtuelle est fine, plus le temps requis pour classer les points est important, mais plus le processus d'assemblage s'en trouve accéléré. Le paramètre représentant le nombre moyen de points par boîte peut donc être optimisé, mais même à sa valeur optimale certaines boîtes contiendront beaucoup plus de cellules que d'autres vu les importantes variations de densité des maillages près des parois.

L'efficacité de la technique étant diminuée dans ces zones, une approche novatrice axée sur le concept de la grille virtuelle a été considérée durant l'implémentation. La technique proposée vise à subdiviser ces boîtes qui contiennent une importante quantité de cellules. L'algorithme ne subdivise que les boîtes contenant plus de points qu'un nombre prédéterminé. Cette approche récursive subdivise une boîte selon les mêmes paramètres N_x et N_y mentionnés plus haut et classe les points de la boîte subdivisée dans la grille virtuelle récursive. La Figure 3.9 montre trois grilles virtuelles (en noir) générées à partir d'un même maillage (en vert). On y voit en (a) la grille initiale non récursive être subdivisée une première fois en (b) puis une deuxième fois en (c). Pour clarifier les discussions suivantes, la grille virtuelle de la Figure 3.9(a) sera dénotée la grille initiale non récursive et ses subdivisions seront dénotées les grilles récursives de niveaux i où i représente le niveau de récursivité. La démarche récursive permet effectivement d'adapter la technique de la grille virtuelle aux changements de densité d'un maillage. L'approche est similaire conceptuellement à l'arbre de recherche octree mais se distingue par le fait que le nombre de subdivisions est variable.



(a) grille virtuelle non récursive (b) grille récursive de niveau 1 (c) grille récursive de niveau 2

Figure 3.9 Raffinement de la grille virtuelle

3.3.8 Parallélisation

Comme le module d'assemblage est développé en fonction d'une intégration à NSCODE, sa parallélisation et l'organisation de sa mémoire sont basées sur l'architecture de celui-ci. NSCODE utilise une mémoire partagée et est parallélisé grâce à l'application de programmation OpenMP. Cette application définit un modèle portable fournissant de bonnes propriétés de scalabilité sur une interface simple et flexible. OpenMP est facile d'utilisation dans le présent contexte pour deux raisons. Premièrement, l'application est basée sur une mémoire partagée ce qui confère à l'implémentation parallèle tous les avantages mentionnés à la section 2. NSCODE n'ayant pas les capacités d'effectuer des calculs sur mémoire distribuée, les considérations relatives à la communication et la duplication des données ne sont pas présentes dans l'implémentation, simplifiant ainsi le processus de parallélisation. Il ne suffit que d'indiquer les sections du code exécutées en parallèle et celles devant spécifiquement être

séquentielles. Deuxièmement, le problème de répartition de la charge est entièrement pris en charge par OpenMP. Il suffit d'indiquer via l'interface les divisions de charge pour permettre à l'application de les assigner aux processeurs de manière automatique. Dans la présente implémentation, la charge est divisée selon les blocs du système de grilles à assembler.

3.3.9 Intégration à NSCODE

L'intégration d'un module d'assemblage chimère au sein de NSCODE est relativement simple. En effet, puisque les deux logiciels interviennent de manière séquentielle, il ne suffit que d'utiliser l'information générée par le module d'assemblage chimère durant le processus de résolution du fluide. Cette utilisation de l'information implique deux modifications mineures au sein de NSCODE. D'abord, il faut s'assurer que les cellules invalides et interpolées ne soient pas mises à jour par les procédures de résolution du code. Ensuite, à chaque fois que NSCODE effectue une mise à jour des conditions limites, les cellules interpolées doivent aussi être mises à jour avec l'information de leurs donneuses.

Les valeurs interpolées peuvent être utilisées de deux manières. La majorité des implémentations utilisent la valeur telle quelle. Une alternative proposée utilise un facteur de pondération entre la valeur enregistrée dans la cellule, qui est représentative de l'écoulement tel que résolu sur la cellule interpolée, et la valeur interpolée. Fujii (1995) a montré que cette approche pouvait améliorer la convergence du calcul. En contrepartie, ce concept nécessite une interpolation et un calcul sur l'ensemble des cellules. Dans NSCODE, la valeur interpolée est utilisée tel quel.

3.4 Résultats

Cette section valide l'ensemble des fonctionnalités implémentées. Afin de créer une logique de présentation, les fonctionnalités seront exposées dans un ordre différent de celui de la section précédente.

3.4.1 Identification des cellules invalides

Puisqu'aucun critère ne permet de valider les capacités de détection des géométries solides, cette validation est faite de manière visuelle. Une des manières simple et efficace consiste à générer une grille cartésienne fine et de la superposer à d'autres grilles contenant des géométries solides. La Figure 3.10 montre les résultats du holecutting suite à l'assemblage de quatre grilles superposées sur le profil MDA. Les grilles consistent en trois géométries solides représentant le profil MDA et un maillage cartésien fin. Pour bien visualiser les résultats, les

trois maillages définissant les géométries solides sont masqués. Seuls les bords qui définissent les géométries solides sont affichés en orange. Les cellules du maillage cartésien identifiées comme étant invalides sont bleues. La Figure 3.10 montre que toutes les cellules situées à l'intérieur des géométries solides sont adéquatement identifiées validant ainsi la fonctionnalité de holecutting de l'implémentation bidimensionnelle.

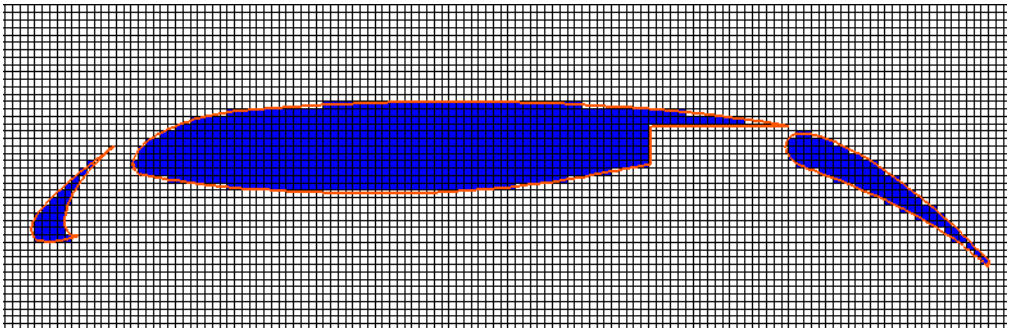
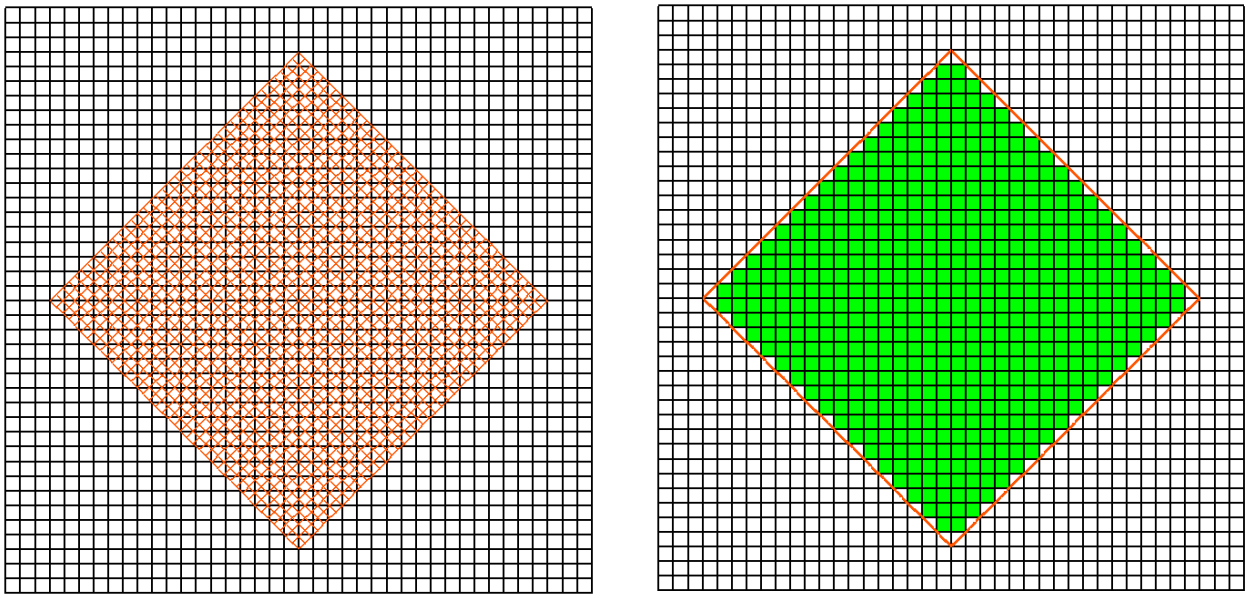


Figure 3.10 Détection de géométries solides sur le profil MDA

3.4.2 Identification des cellules interpolées

Comme pour les cellules invalides, la validation des cellules interpolées est faite de manière visuelle. La validation du critère hiérarchique est effectuée avec deux maillages cartésiens superposés tels que montrés à la Figure 3.11(a) où le maillage oblique domine hiérarchiquement le maillage noir. La Figure 3.11(b) montre les cellules interpolées du maillage dominé en vert et les contours du maillage oblique en orange. On voit que toutes les cellules superposées au maillage oblique sont correctement identifiées comme interpolées validant ainsi l'implémentation du critère hiérarchique.

La validation du critère de la distance à la paroi locale est effectuée avec deux grilles identiques du profil NACA0012 obtenue de Vassberg et Jameson (2010) créées pour une résolution de fluide eulérien. Une telle grille est montrée à la Figure 3.12(a). Les grilles, de hiérarchie équivalente, sont décalées l'une par rapport à l'autre à un angle de 45°. L'assemblage de ces grilles est montré à la Figure 3.12(b) où les cellules invalides et les cellules interpolées sont cachées. Seules les cellules calculées sont affichées. La Figure 3.12(b) valide que le critère de distance à la paroi a été correctement implémenté. Par le fait même, le calcul de la distance à la paroi local s'en trouve aussi validé.



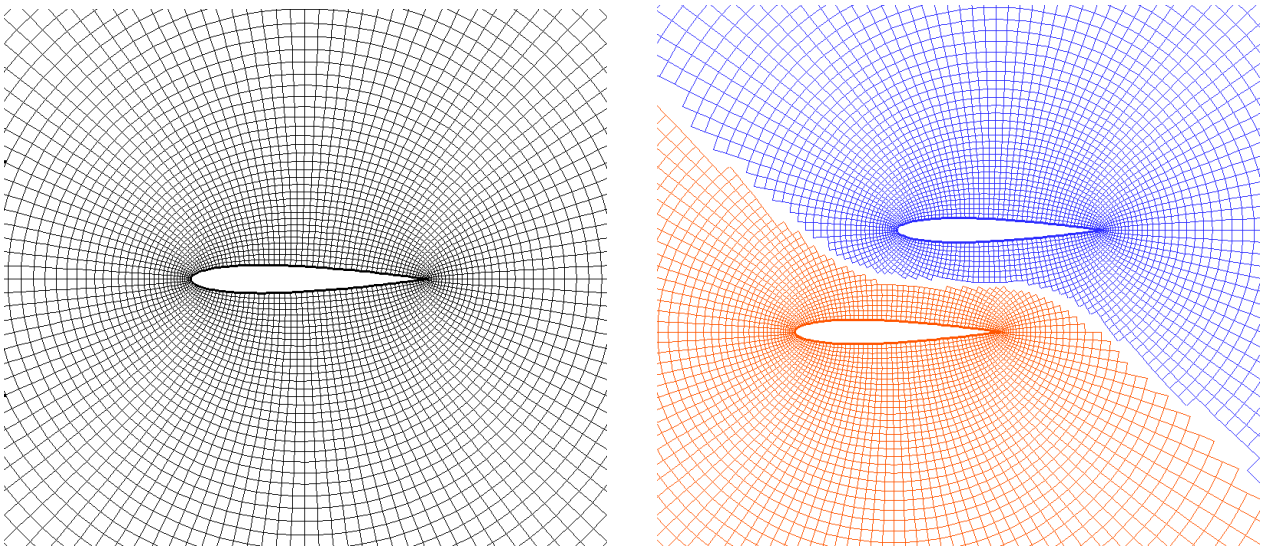
(a) Système de grilles utilisé pour valider le critère hiérarchique

(b) Les cellules interpolées sont identifiées en vert

Figure 3.11 Validation du critère hiérarchique

3.4.3 Zone tampon

Comme le montre la Figure 3.12(b), l'identification des cellules réceptrices va généralement laisser des trous dans l'assemblage chimère aux régions où les grilles doivent s'échanger l'in-



(a)

(b)

Figure 3.12 Critère de distance à la paroi

formation. Ces zones sont le résultat du positionnement relatif des cellules des maillages en question. Un assemblage contenant de telles zones n'est pas problématique en soit, mais implique que des cellules donneuses sont aussi réceptrices. Ce système implicite nécessite un traitement spécial coûteux en temps lors de l'interpolation (Chesshire et Henshaw, 1990) et peut diminuer la précision de la solution (Rogers *et al.*, 2003). Pour éviter la problématique, l'algorithme crée un système explicite où les valeurs interpolées sont basées sur des cellules dominantes. La procédure consiste à changer le statut des premières rangées de cellules formant la frontière entre les zones des cellules calculées et celles interpolées comme montré à la Figure 3.13. La largeur de la bande de superposition assure que les interpolations implicites sont limitées. Puisque le nombre de rangées ajoutées est fonction du schéma numérique, ce nombre est un paramètre déterminé par l'usager avant l'assemblage.

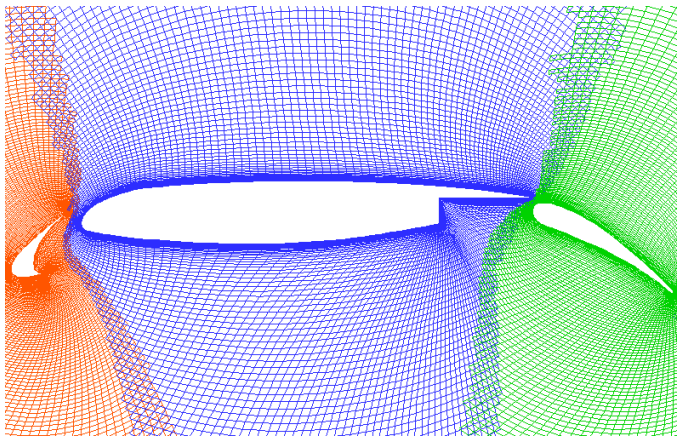


Figure 3.13 Ajout de la zone tampon sur le système de grilles du profil MDA

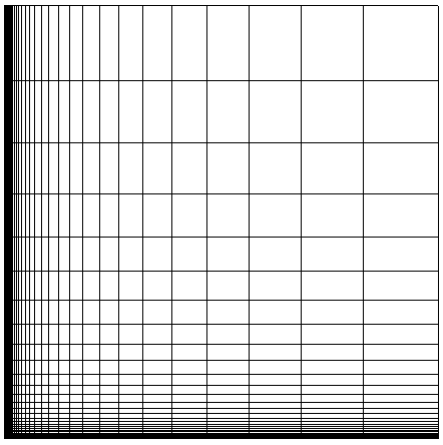


Figure 3.14 Grille cartésienne

3.4.4 Méthode d'accélération

Le système de grille présenté à la Figure 3.11(a) fut assemblé sur trois niveaux de densité différents. Le tableau 3.1 montre le temps nécessaire à l'assemblage avec et sans l'utilisation d'une grille virtuelle. On y voit que les méthodes d'accélération sont 600 fois plus rapides que l'algorithme de base pour un nombre de points de 8.19 million. On constate donc que la méthode d'accélération est absolument nécessaire pour réaliser des assemblages dans un temps raisonnable et qu'il est inconcevable de tenter d'assembler des systèmes de grilles modernes sans celle-ci.

La technique des grilles virtuelles est maintenant appliquée à un cas canonique. Deux maillages cartésiens représentatifs de la discrétisation d'une couche limite sont superposés puis assem-

Tableau 3.1 Temps requis en secondes pour l'assemblage avec et sans grille virtuelle.

Nb de points	Avec	Sans
1.28×10^3	0.03	0.16
5.12×10^4	0.15	1.89
2.05×10^5	0.89	73.2
8.19×10^6	6.16	3687

blés. Les maillages, montrés à la Figure 3.14, contiennent 160 cellules dans chacune des directions. Pour montrer l'effet des grilles virtuelles récursives, les deux maillages identiques sont superposés sans décalage ce qui créer des zones de fortes concentrations de mailles. On cherche ici à déterminer la performance de la méthode récursive vis-à-vis deux paramètres :

1. le nombre de points moyen par boîtes dénoté β
2. le niveau de récursivité des grilles virtuelles

Il est nécessaire de définir un nombre critique de points qui déclenchera la génération d'une grille virtuelle récursive. Ce nombre est soumis à deux contraintes. D'une part, il doit être suffisamment élevé pour éviter d'effectuer le processus récursif aux endroits où il n'est pas nécessaire. Inversement, il doit être suffisamment petit pour éviter d'avoir trop de points dans une boîte. Ce nombre critique pourrait faire l'objet d'une optimisation, mais l'expérience avec le code a montré que la valeur de 100 est souvent optimale. Cette valeur est donc retenue.

La Figure 3.15 montre le temps requis pour assembler les maillages en fonction du paramètre β pour différents niveaux de récursivité (noter que les deux échelles sont logarithmiques). La courbe noire correspond au temps d'assemblage obtenu grâce à l'utilisation de grilles virtuelles non récursives. Les courbes titrées «Niveau i» représentent les temps d'assemblage obtenus avec l'utilisation de grilles virtuelles récursives de niveau i. Le graphique comporte trois points d'intérêt. Premièrement, on voit d'abord que sur une grille virtuelle sans récursivité la zone optimale du paramètre β est très petite. Autrement dit, de très faibles variations de ce critère amènent à des temps très élevés. On observe en effet que les temps d'assemblage peuvent être rapidement doublés, voir quadruplés, si le paramètre β est loin de sa zone optimale. Au contraire, l'algorithme récursif offre une plage optimale très large, un aspect idéal pour une application où l'utilisateur ne désire pas investir du temps pour identifier les paramètres optimaux. Deuxièmement, le fait d'utiliser un nombre critique fixé à 100 assure que pour un paramètre β faible le processus récursif n'est pas déclenché. Ceci implique que l'utilisation de grilles récursives de haut niveau garantit d'obtenir le temps d'assemblage le plus faible. Troisièmement, l'algorithme récursif montre une performance largement supérieure en termes de temps. En effet, le temps d'assemblage au niveau 9 est réduit de 70% par rapport à la

grille sans récursivité de paramètre β optimal.

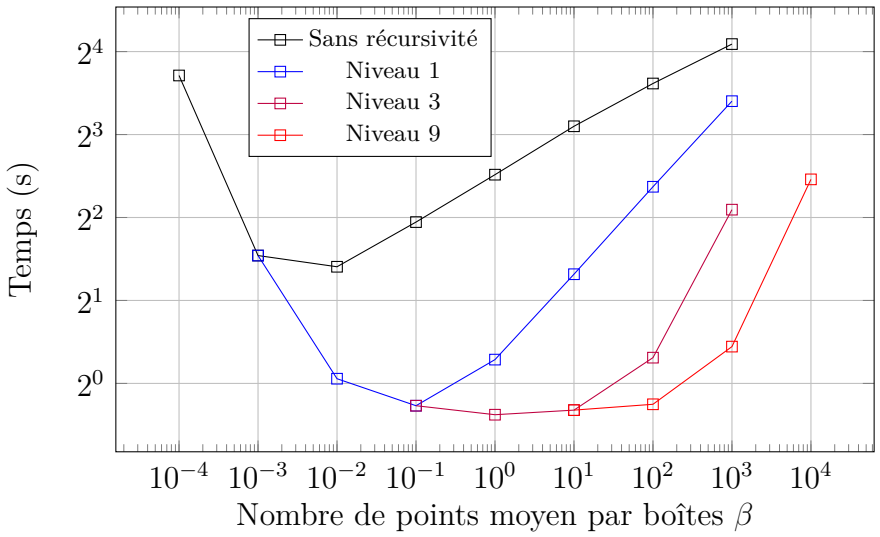


Figure 3.15 Temps d'assemblage du cas académique en fonction du paramètre β.

La technique est ensuite appliquée à un système de grille représentatif des géométries du domaine de l'aéronautique. Le cas test consiste en le système de grille du profil MDA tel que présenté à la Figure 3.13. Les résultats sont montrés à la Figure 3.16. Reprenons les trois mêmes observations que pour le cas précédant. D'abord, on note que la zone optimale de la grille virtuelle sans récursivité est encore une fois très restreinte et que l'algorithme récursif maintient sa capacité à fournir une plage optimale très large. Ensuite, dans la zone de faible β, le temps d'assemblage avec et sans grilles virtuelles récursives est équivalent. Finalement, l'algorithme récursif offre sur cette configuration un gain en performance de 10%. Ce gain, considérablement moindre que le cas académique, est explicable par le fait qu'il y a moins de superposition sur le système du profil MDA. En effet, quoique non visible sur la Figure 3.13, le maillage du corps principal (bleu) s'étend jusqu'à 50 fois la corde du profil alors que ceux du bec et du volet sont très localisés (environ une corde). Cette observation implique que le processus récursif est appliqué, pour le maillage du profil principal, dans une région où il n'y a pas de superposition. Or, puisque les grilles virtuelles servent à identifier les superpositions, la construction d'une telle grille est inutile dans les zones où il n'y en a pas. Ce point constitue une limite de la technique telle qu'implémentée actuellement. Pour y parer, il sera nécessaire d'identifier grossièrement les endroits où aucune superposition n'est possible pour éviter d'y effectuer la classification des points dans les grilles virtuelles. Ce point fera l'objet d'une discussion dans la conclusion des présents travaux.

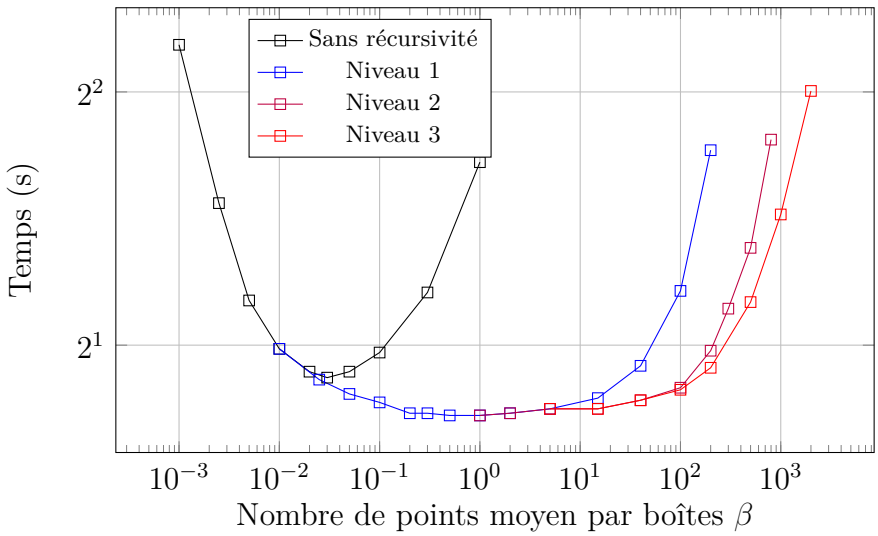


Figure 3.16 Temps d'assemblage du profil MDA en fonction du paramètre β.

3.4.5 Ordre de précision

Avant de pouvoir intégrer le module d'assemblage à NSCODE, l'ordre de précision doit être déterminé. Cette procédure va permettre de valider que l'association entre les cellules interpolées et leurs donneuses est adéquate. L'étude est réalisée sur deux grilles superposées utilisant un critère hiérarchique qui permet de s'assurer qu'une seule grille soit systématiquement dominante. L'évaluation de l'ordre de précision de l'assemblage des grilles est réalisé grâce à une étude de raffinement de maillage effectuée sur une famille de six assemblages de grilles (voir Tableau 3.2) comme celui montré à la Figure 3.11(a) où le maillage oblique domine hiérarchiquement le maillage noir. Les valeurs minimales et maximales des coordonnées dans les deux directions sont respectivement de -20 et 20. Les deux grilles les plus grossières contiennent 40 cellules sur chacun de leurs côtés. En subdivisant systématique la taille des cellules par deux, les deux grilles les plus fines contiennent 1280 cellules sur chacun de leurs côtés.

L'ordre de précision est calculé pour les fonctions d'interpolation tétravolumique et bilinéaire en mesurant l'erreur d'interpolation obtenue à la suite d'un assemblage des deux maillages. Pour pouvoir calculer cette erreur d'interpolation, une fonction analytique basée sur les coordonnées physiques est calculée aux cellules du maillage dominant. La fonction non linéaire utilisée est :

$$f(x,y) = cos\left(\frac{\pi x}{30}\right) * sin\left(\frac{\pi y}{30}\right) \tag{3.14}$$

Cette fonction est bien adaptée aux grilles puisque sa variation est basse par rapport à la taille des cellules.

Suite à l'assemblage, l'interpolation de cette fonction est effectuée sur les cellules réceptrices du maillage dominé. L'erreur d'interpolation locale est calculée comme la différence entre les valeurs interpolée et analytique à l'endroit de la cellule réceptrice. L'erreur d'interpolation d'intérêt ici consiste en la somme des erreurs d'interpolation locales. La procédure permettant de calculer l'ordre de précision est extraite de (Vassberg et Jameson, 2010).

L'équation qui permet de calculer l'ordre de convergence d'un schéma général est :

$$(F_f - F_m){h_g}^p + (F_g - F_f){h_m}^p + (F_m - F_g){h_f}^p = 0 \tag{3.15}$$

où F est le résultat discret dont on veut connaître l'ordre de précision et f, m et g sont les indices décrivant une grille fine, moyenne et grossière respectivement. L'indice d'échelle d'une grille est dénoté par h et la valeur estimée de l'ordre de précision est p. Pour que l'équation 3.15 soit correctement résolue, $h_f < h_m < h_g$ et la séquence $[F_g, F_m, F_f]$ doit être monotone. La famille de grille générée suit un ratio de raffinement $\omega = 2$ donc :

$$h_g = \omega h_m = \omega^2 h_f = 2h_m = 4h_f \tag{3.16}$$

On note aussi que

$$(F_f - F_g) = (F_f - F_m) + (F_m - F_g) \tag{3.17}$$

En combinant les équations 3.15 et 3.17 et en manipulant, on obtient la relation R décrivant la variation des valeurs discrètes obtenues entre chaque changement de maillage :

$$R = \frac{F_f - F_m}{F_m - F_g} = \frac{\omega^p - 1}{\omega^{2p} - \omega^p} = \frac{2^p - 1}{2^{2p} - 2^p} \tag{3.18}$$

Pour résoudre p, on utilise la méthode de Newton :

$$p^{n+1} = p^n - \frac{G(p)}{G_p(p)} \tag{3.19}$$

où

$$G(p) = (4^p - 2^p)R - 2^p + 1 = 0 \tag{3.20}$$

et

$$G_p(p^n) = \ln(4)4^p R - \ln(2)2^p(R+1) \tag{3.21}$$

L'estimation de F est faite pour $h \to 0$ comme suit :

$$F^* = F_f + \frac{F_f - F_m}{2^p - 1} \tag{3.22}$$

Le tableau 3.2 montre les résultats de l'étude. La première colonne indique le nombre de cellules sur chacun des côtés des grilles, la deuxième indique le nombre total de cellules de chacun des deux maillages. Les trois rangées suivantes indiquent respectivement l'erreur d'interpolation $F = \sqrt{\sum \left((f_i^* - f_i)/\Omega_i\right)^2}$ où f^* représente la valeur interpolée de la fonction f , la variation des valeurs discrètes R et l'ordre de précision estimé p pour la fonction d'interpolation tétravolumique. Les trois mêmes valeurs sont aussi présentées pour la fonction d'interpolation bilinéaire. On note d'abord la forte similarité entre les résultats des deux fonctions d'interpolation, particulièrement sur les grilles les plus fines. Les résultats montrent que la précision de l'assemblage produit est de deuxième ordre pour les deux fonctions ce qui prouve du même coup que l'identification des cellules donneuses est réalisée correctement. Les résultats sont d'autant plus probants lorsqu'on considère que les ordres de précision p du tableau 3.2 s'approchent asymptotiquement de la valeur de deux. Les résultats sont illustrés à la Figure 3.17 où les données discrètes sont représentées par les symboles pleins. Les droites représentent les extrapolations vers $h \to 0$. La proximité des données discrètes à cette droite montre que les résultats obtenus sont indépendants du maillage validant l'ordre de précision calculé. Puisque l'ordre de précision de la fonction d'interpolation bilinéaire est le plus proche de deux sur les grilles plus grossières, cette fonction d'interpolation est retenue.

Tableau 3.2 Erreur d'interpolation et ordre de convergence estimé de l'implémentation bidimensionnelle

Maillage		Tétravolumique			Bilinéaire		
NC	Npts	F	R	p	F	R	p
40	1600	8.53E-3	-	-	6.74E-3	-	-
80	6400	1.98E-3	-	-	1.98E-3	-	-
160	25600	5.31E-4	0.221	1.56	5.31E-4	0.304	1.72
320	102400	1.37E-4	0.271	1.88	1.37E-4	0.271	1.88
640	409600	3.51E-5	0.260	1.95	3.51E-5	0.260	1.95
1280	1638400	8.85E-6	0.257	1.96	8.85E-6	0.257	1.96
-	-	-1.94E-7	-	-	-1.94E-7	-	-

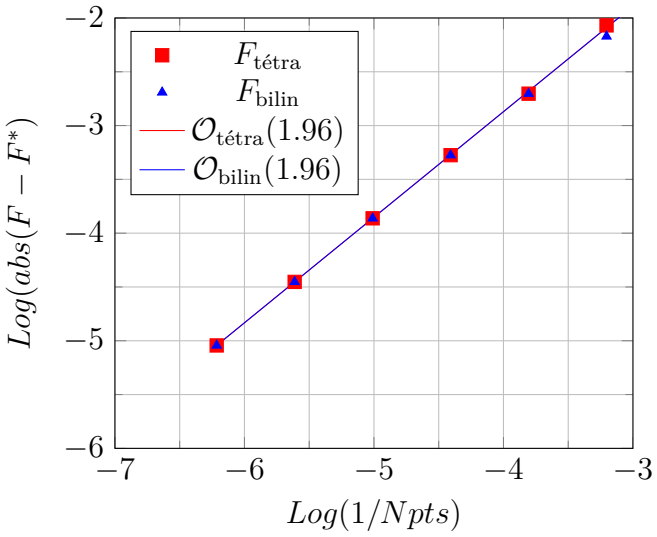


Figure 3.17 Convergence de l'erreur d'interpolation en fonction du nombre de points

3.4.6 Parallélisation

Pour correctement tester la performance parallèle de l'implémentation bidimensionnelle, la répartition de la charge du cas testé doit être parfaite. En effet, comme il fut mentionné à la Section 2, une mauvaise répartition de charge est souvent le facteur principal causant les pertes de performances d'une implémentation d'assemblage chimère. Ainsi, pour valider que le programme présente un gain de performance lorsqu'il est lancé sur plusieurs processeurs, le système de grilles à assembler requiert deux caractéristiques importantes. D'abord, chacun des blocs composant le système doit nécessiter autant de calculs que n'importe quel autre. Ensuite, tous les blocs doivent être de même hiérarchie pour éviter que les filtres expliqués précédemment n'affectent le balancement de la charge. Pour répondre à ces deux critères, deux grilles cartésiennes identiques de trois blocs chacune sont superposées l'une sur l'autre sans décalage. Chacun des trois blocs est formé de 160×160 cellules. Le critère de discrimination utilisé est la distance à la paroi. Une condition limite de paroi est appliquée à une des faces de la grille A et à la face opposée sur la grille B. Pour assurer une bonne répartition, le nombre de processeurs utilisés pour le calcul doit être un diviseur de 6. Ainsi, les assemblages sur 1, 2, 3 et 6 processeurs sont ici considérés. La machine utilisée pour réaliser les tests de parallélisation exploite un processeur Intel Core i7-3930K hexa-coeur à 3.20GHz.

Un examen détaillé de ces quatre assemblages a montré que les résultats obtenus sur 1,2,3 et 6 processeurs sont rigoureusement identiques. L'implémentation parallèle est donc réussie du point de vue du traitement des données puisqu'elle produit des résultats indépendants du nombre de processeurs. Les temps d'assemblage sont présentés au Tableau 3.3. La dernière

Tableau 3.3 Temps d'assemblage et facteur d'accélération en fonction du nombre de processeurs

Nb procs	Temps (s)	Accélération
1	186.2	1.00
2	93.0	2.00
3	63.4	2.94
6	32.8	5.68

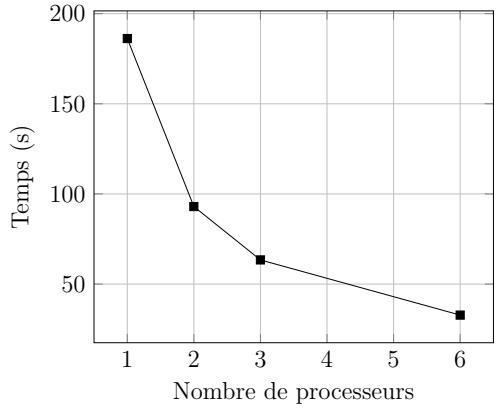


Figure 3.18 Temps requis pour assembler les grilles de 3 blocs

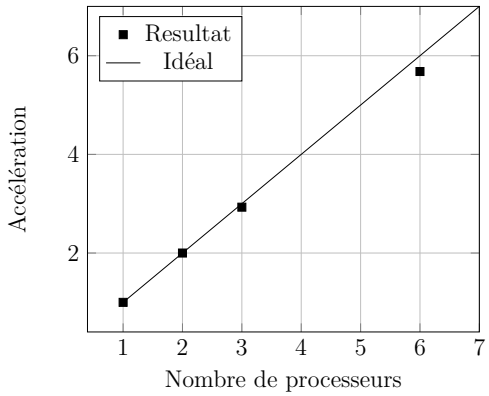


Figure 3.19 Accélération de l'assemblage en fonction du nombre de processeurs

colonne de ce tableau représente le facteur d'accélération obtenu en divisant le temps d'assemblage par celui obtenu avec un processeur. Les deux colonnes du tableau sont respectivement représentées graphiquement aux Figures 3.18 et 3.19. La Figure 3.18 montre que le temps d'assemblage diminue systématiquement lorsque davantage de processeurs sont appliqués au calcul démontrant ainsi les capacités de calculs parallèles. Le troisième point d'intérêt, la scalabilité, est représenté à la Figure 3.19 où la droite indique la performance optimale qui serait obtenue si le programme était exécuté de manière entièrement parallèle. On voit que le gain en performance diminue légèrement au fur et à mesure que des processeurs sont ajoutés au calcul. Ce phénomène est mathématiquement représenté par la loi d'Amdahl (Amdahl, 1967) :

$$A(n) = \frac{T(1)}{T(n)} = \frac{1}{S + \frac{1}{n}(1 - S)} \tag{3.23}$$

où A représente le facteur d'accélération, T le temps de calcul, n le nombre de processeurs et S la fraction séquentielle du programme. Selon cette loi, l'implémentation ci-présentée offre un taux de parallélisme de 98.9% ce qui permet un facteur d'accélération maximal d'environ 90. Ces résultats démontrent que l'implémentation parallèle bidimensionnelle présente

d'excellentes caractéristiques de scalabilté.

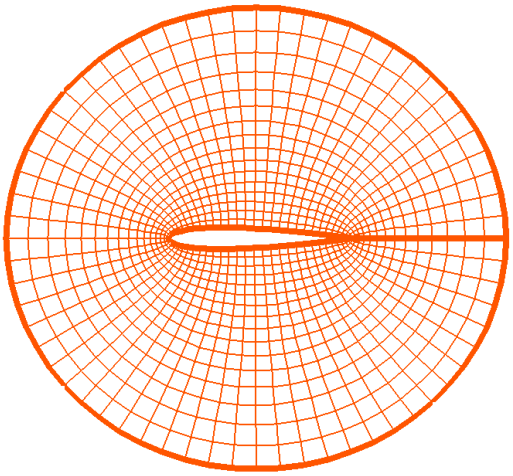
3.4.7 Intégration à NSCODE

L'intégration du module d'assemblage chimère au sein de NSCODE permet d'effectuer une validation plus détaillée de l'implémentation. Ainsi, l'implémentation chimère, une fois utilisée dans NSCODE, devrait premièrement préserver l'ordre de précision du solveur et deuxièmement produire des résultats similaires à ceux obtenus sans la méthode chimère.

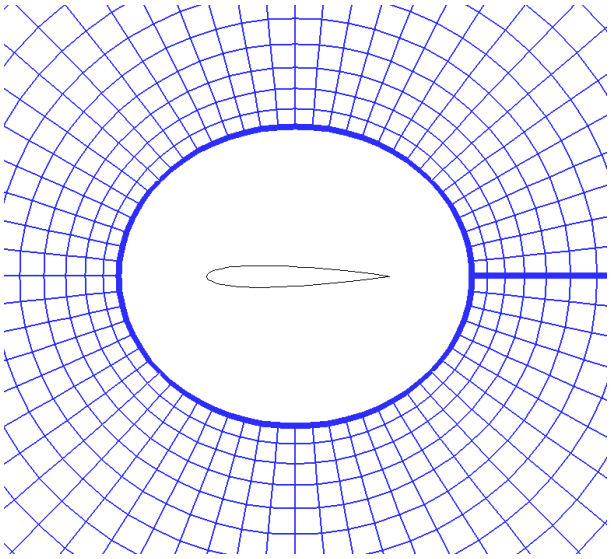
Pour évaluer le premier point, une étude de convergence de grille similaire à celle effectuée par Vassberg et Jameson (2010) est réalisée sur les grilles du profil NACA0012 fournies par les auteurs. Ces grilles ont la caractéristique d'être parfaitement orthogonales puisqu'elles sont obtenues analytiquement. Les six grilles utilisées sont successivement raffinées selon un facteur de deux, la plus grossière contenant 64 cellules dans les directions tangentielle et normale et la plus fine en contenant 2048. La grille de 128 cellules est illustrée à la Figure 3.12(a). Puisque les grilles fournies ne présentent pas de superpositions, elles sont traitables par NSCODE sans l'utilisation du module chimère. Afin de valider l'implémentation chimère, l'ensemble des grilles fournies doivent être découpées de manière à créer une superposition. Cette découpe est montrée à la Figure 3.20 où on y voit en (a) la grille définissant le profil et en (b) la grille éloignée. Le fait que la grille éloignée n'ait aucune information de la géométrie solide brise effectivement le lien avec le profil. Cette communication doit être construite par le module chimère en superposant les deux grilles illustrées en (c) où la zone de superposition est identifiée en vert. Pour créer le lien entre les deux grilles, des conditions limites de type chimère sont appliquées à la frontière externe de la grille définissant la géométrie et à la frontière interne de la grille éloignée.

Les schémas numériques de NSCODE sont de premier ordre dans des conditions d'écoulement transsoniques. Pour valider le deuxième ordre de l'assemblage chimère, les simulations sont donc réalisées dans le domaine subsonique. De plus, les grilles fournies sont adaptées aux simulations eulériennes. Ainsi, les conditions d'écoulement sont $M = 0.5, \alpha = 1.25°$ (Euler). La Figure 3.20(d) montre les lignes de coefficient de pression (de 0.990 à 1.006 par incrément de 0.04) obtenues sur la grille superposée de 64 cellules. Par la continuité des lignes de pression dans la zone de superposition, la figure montre une bonne communication des variables de l'écoulement à travers les deux grilles.

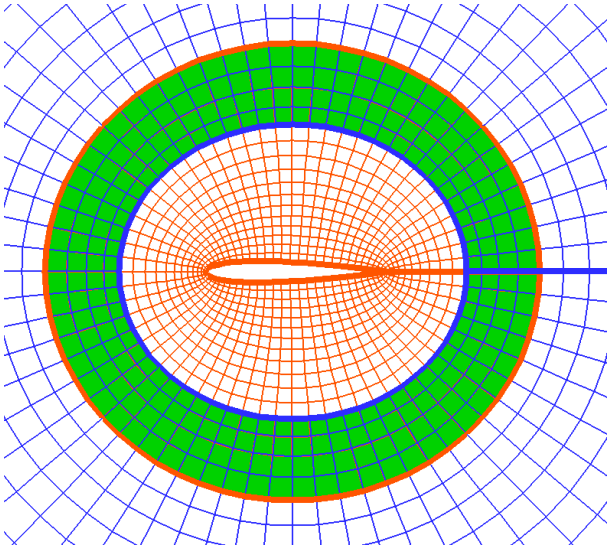
L'ordre de précision du solveur de NSCODE est calculé pour les résultats obtenus avec et sans la technique chimère afin de valider son indépendance par rapport à l'assemblage chimère. Les résultats pour les coefficients de portance et de traînée sont montrés respectivement aux Figures 3.21(a) et 3.21(b). Pour les deux figures, la convergence du coefficient calculé



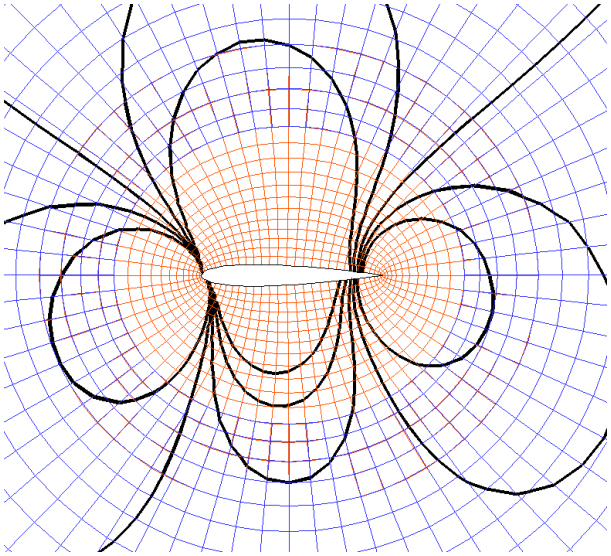
(a) Grille proche paroi



(b) Grille éloignée (le profil est montré à titre de référence seulement)



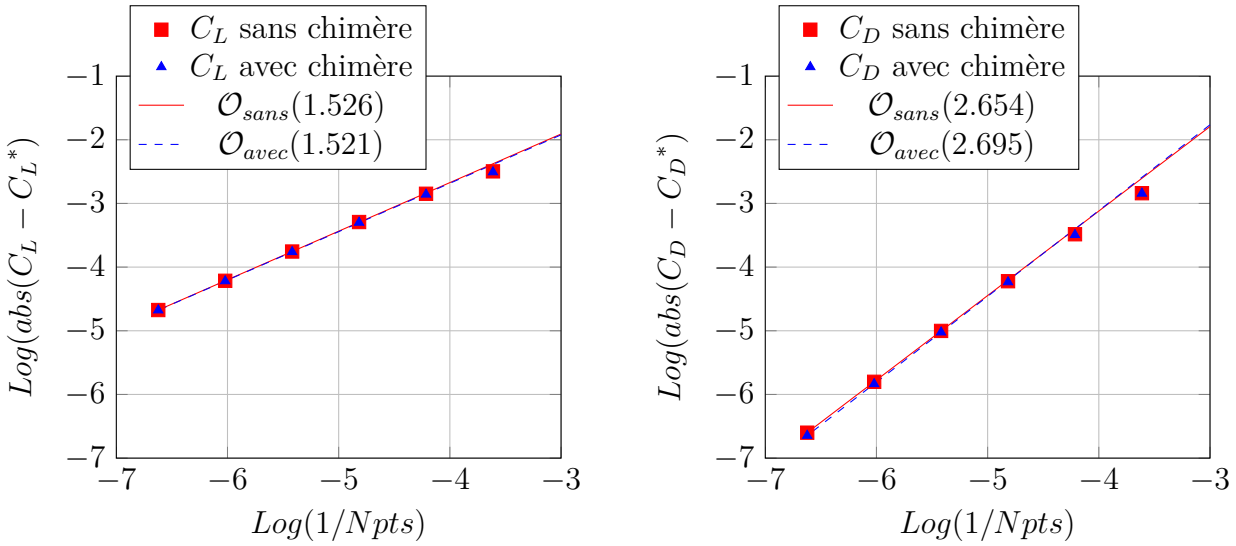
(c) Superposition des grilles proche paroi et éloignées. La zone verte représente la zone de superposition



(d) Lignes de pression obtenues suite au calcul

Figure 3.20 Visualisation de l'intégration NSCODE

(points discrets) et l'ordre de précision obtenu (droite) sont montrés avec et sans l'assemblage chimère. On voit que dans les ordres de précision des deux méthodes sont très similaires. Ceci valide la compatibilité de l'assemblage chimère avec le solveur de NSCODE. De plus, ces graphiques permettent de déterminer les propriétés de convergence des grilles des simulations. Ainsi, les points discrets situés sur la droite représentant l'ordre de convergence sont des résultats indépendants de la grille utilisée. Cette indépendance est souhaitable puisqu'elle



(a) Coefficient de portance

(b) Coefficient de traînée

Figure 3.21 Convergence de l'erreur en fonction du nombre de points

indique qu'une grille capture adéquatement les phénomènes physiques en jeu. Sur la Figure 3.21, on voit que les résultats obtenus sans la méthode chimère sont tous alignés sur les droites et que cette caractéristique désirable est préservée par l'assemblage chimère.

La Figure 3.22 montre une comparaison du résidu en fonction du nombre d'itérations des simulations avec et sans la technique chimère effectuées sur la grille de 64x64 cellules. Cette comparaison est non seulement valide, mais aussi d'intérêt puisque les maillages utilisés sont très similaires. On constate que le processus de résolution n'est pas affecté par l'assemblage chimère montrant ainsi que les techniques mathématiques utilisées dans NSCODE gardent leur efficacité.

La Figure 3.21 valide l'ordre de précision de la méthode chimère, mais cette information à elle seule ne permet pas de valider que les résultats obtenus avec la méthode sont cohérents avec ceux calculés sans celle-ci. Pour ce faire, une comparaison de la différence entre les deux coefficients calculés est faite. La Figure 3.23 montre cette différence pour les six simulations. On voit premièrement que la différence entre les deux coefficients est très basse, soit moins de 0.01% même sur la grille la plus grossière. Deuxièmement, la figure montre que cette erreur diminue systématiquement au fur et à mesure que la grille est raffinée indiquant que la différence entre les coefficients est due à la discrétisation physique.

Pour valider davantage l'intégration à NSCODE, une simulation est réalisée sur une géométrie complexe multiélément. Le profil MDA est choisi puisqu'il s'agit d'un cas test très répandu au sein de la communauté. La simulation est effectuée avec et sans la technique chimère aux

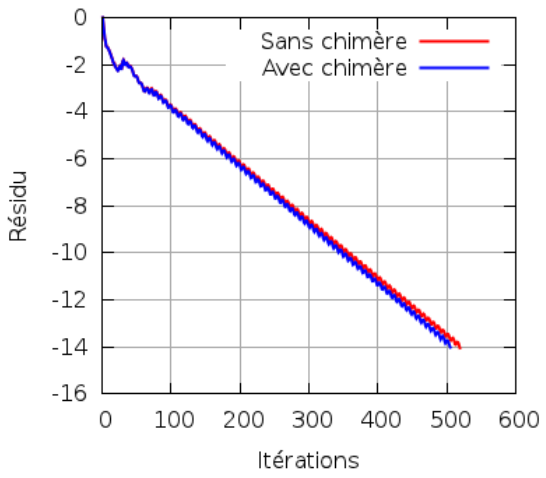


Figure 3.22 Valeur du résidu en fonction du nombre d'itérations avec et sans la méthode chimère

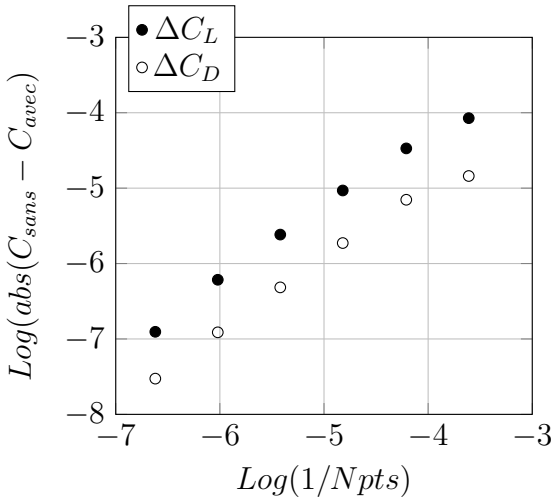


Figure 3.23 Différences entre les coefficients obtenus avec et sans la méthode chimère en fonction du nombre de points

conditions suivantes : $M = 0.2, \alpha = 8°, Re = 9.0E6$, Spalart-Allmaras. La grille chimère utilisée consiste en une grille en «O» pour chacun des éléments et a été réalisé de manière automatique grâce au logiciel maison NSGRID (Hasanzadeh *et al.*, 2015). L'assemblage de grilles chimère obtenu a précédemment été montré à la Figure 3.13. La grille utilisée pour

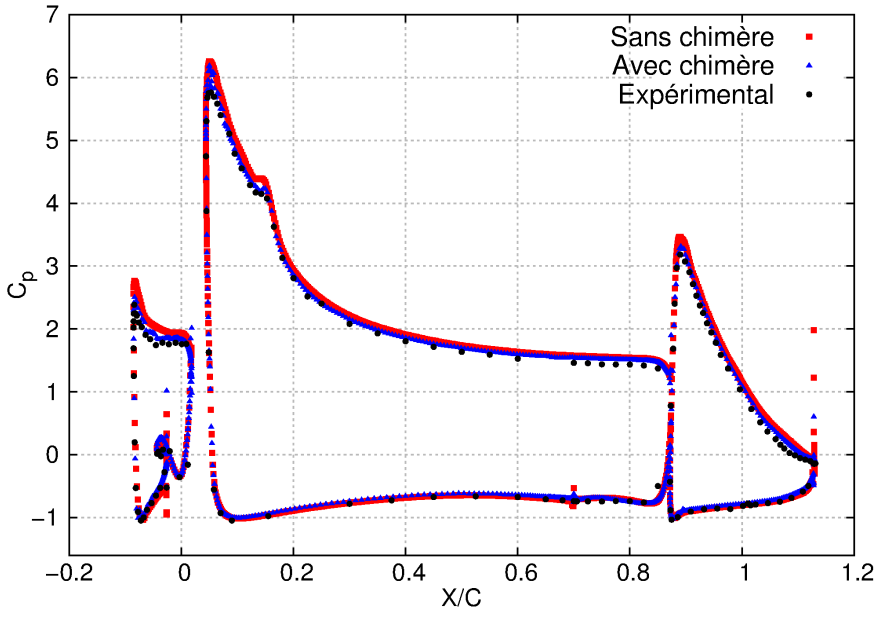


Figure 3.24 Coefficient de pression sur le profil MDA. $M0.2, alpha = 8°, Re9.0E6$

effectuer la simulation sans la technique chimère n'est pas montrée. Il suffit ici de dire que la présence des trois éléments solides et la discontinuité présentée par le profil principal mènent à une grille de faible qualité présentant plusieurs cellules étirées et inclinées.

Les coefficients de pression obtenus sont comparés aux valeurs expérimentales (Cagnone *et al.*, 2011) à la Figure 3.24. On observe d'abord que les deux résultats numériques sont concordants avec les données expérimentales et que la zone de recompression est mieux prédite par la grille chimère. Cette observation est bien visible sur l'extrados du bec. Outre les résultats de simulations, l'utilisation de la méthode chimère pour analyser le profil multiélément s'est avéré avantageuse durant le processus de génération de maillage. En effet, le maillage produit présente une meilleure discrétisation spatiale pour un temps de création bien plus faible.

CHAPITRE 4 MÉTHODE CHIMÈRE TRIDIMENSIONNELLE

4.1 Solveurs tridimensionnels

Les logiciels CFD tridimensionnels, tel FANSC créé au début des années 2000 par Bombardier Aéronautique (Sermeus *et al.*, 2007), sont développés pour effectuer la conception, les analyses et la certification des aéronefs. Pour réaliser ces tâches en un temps raisonnable, les simulations réalisées grâce à ces logiciels nécessitent l'utilisation de super ordinateurs à mémoire distribuée. Outre cette différence majeure et le fait qu'ils résolvent des écoulements tridimensionnels, les solveurs tridimensionnel présentent très peu de différence avec NSCODE. Entre autres, ils utilisent un système d'équations, des algorithmes de résolution ainsi qu'un schéma de discrétisation spatiale généralement très similaires à ceux de NSCODE. De plus, les codes tridimensionnels font aussi usage de la méthode de décomposition multibloc.

4.2 Fonctions d'interpolation

Comme pour l'implémentation bidimensionnelle, l'implémentation tridimensionnelle utilise les fonctions d'interpolation tétravolumique et trilinéaire. Leur extension en trois dimensions est simple puisque les deux méthodes gardent conceptuellement la même forme.

L'interpolation tétravolumique utilise des tétraèdres basés sur quatre points plutôt que des triangles à trois points. Comme en deux dimensions, le volume identifié n'est pas nécessairement le plus petit, mais bien le premier identifié par la procédure de test.

L'interpolation trilinéaire ajoute quant à elle une coordonnée naturelle supplémentaire ce qui défini le système $\vec{\Phi_B} = (\xi_B, \eta_B, \psi_B)$.

$$F(\vec{\Phi_B}) = \begin{bmatrix} x_Q(\xi_B, \eta_B, \psi_B) \\ y_Q(\xi_B, \eta_B, \psi_B) \\ z_Q(\xi_B, \eta_B, \psi_B) \end{bmatrix} = \sum_{i=1}^{8} N_i(\xi_B, \eta_B, \psi_B)\vec{B}i \tag{4.1}$$

où les coefficients N_i sont définis par :

$$\begin{array}{ll} N_1 = (1-\xi_B)(1-\eta_B)(1-\psi_B) & N_5 = (1-\xi_B)(1-\eta_B)\psi_B \\ N_2 = \xi_B(1-\eta_B)(1-\psi_B) & N_6 = \xi_B(1-\eta_B)\psi_B \\ N_3 = (1-\xi_B)\eta_B(1-\psi_B) & N_7 = (1-\xi_B)\eta_B\psi_B \\ N_4 = \xi_B\eta_B(1-\psi_B) & N_8 = \xi_B\eta_B\psi_B \end{array} \tag{4.2}$$

Le centre paramétrique de la cellule de valeur $\xi_B = \eta_B = \psi_B = 0.5$ est utilisé comme valeur initiale.

4.3 Assemblage des grilles

4.3.1 Entrée et sortie CGNS

Un requis propre à l'implémentation tridimensionnelle consiste à offrir une compatibilité avec le standard CGNS. Le standard CGNS (CFD General Notation System) fournit une approche générale, portable et flexible vis-à-vis la lecture et le stockage de données de CFD à travers un logiciel gratuit à sources ouvertes (open-source). Initialement créé par Boeing et la NASA, il facilite le partage de fichiers de données entre différents collaborateurs et la qualité d'archivage des données. Le standard est d'ailleurs une pratique recommandée de l'«American Institute of Aeronautics and Astronautics», un organisme influant du domaine aéronautique. Pour ces motifs, l'implémentation tridimensionnelle doit être capable de traiter des données d'entrées et de générer une sortie sous ce format.

Le système CGNS est constitué de deux parties : 1) une description du format standardisé pour produire des données, et 2) des librairies capables de lire, écrire et modifier les données enregistrées dans ce format [1]. L'accès aux données est réalisé grâce à une librairie de fonctions compatible avec le langage de programmation utilisé dans le présent projet.

Groupe chimère

Contrairement aux fichiers de l'implémentation bidimensionnelle, sous le format CGNS, l'ensemble des informations concernant un système de grille est contenu au sein d'un même fichier unique. La technique utilisée pour l'implémentation bidimensionnelle qui consiste à déterminer les groupes chimère grâce aux différents fichiers composant le système de grille n'est plus valide. On doit maintenant plutôt utiliser la définition d'un groupe chimère pour effectuer l'identification. Cette définition stipule que les blocs formant un groupe chimère ont la propriété d'être reliés de manière contigüe ce qui nous permet d'utiliser l'information topologique pour faire l'identification. Ainsi, dans l'implémentation tridimensionnelle, l'identification des groupes chimère est automatisée via un outil capable de parcourir les blocs contigus grâce à une recherche des conditions limites de connexion pour assembler ces blocs en un groupe chimère.

1. http ://cgns.sourceforge.net/FAQs.html

Identification des superpositions

La technique de détection d'une superposition telle qu'utilisée en deux dimensions doit être conceptuellement modifiée pour être utilisable en trois dimensions. Il est nécessaire de construire les 18 tétraèdres définis par les 8 points formant une cellule structurée en trois dimensions. Le test de superposition avec le point d'intérêt est ensuite réalisé sur chacun de ces tétraèdres en effectuant une comparaison de volume. Tel que montré à la Figure 4.1, le tétraèdre est composé des points $C1$, $C2$, $C3$ et $C4$ et le point Q est le point testé. Les quatre volumes formés par le point test et trois points du tétraèdre sont additionnés puis comparés au volume du tétraèdre. Une superposition est identifiée dès que la différence entre ces deux volumes est inférieure à une tolérance de 1×10^{-12}.

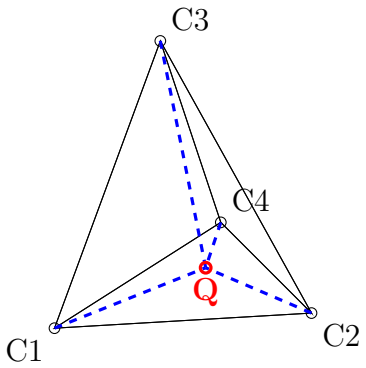


Figure 4.1 Test de superposition en trois dimensions

Identification des cellules interpolées

Pour identifier les cellules interpolées, les critères de hiérarchie et de distance à la paroi sont utilisés de la même manière que dans l'implémentation bidimensionnelle. Puisque le calcul de la distance à la paroi locale est basé sur l'ensemble des coordonnées définissant une géométrie solide, il est nécessaire de communiquer à l'ensemble des membres du groupe chimère les coordonnées de cette géométrie puisque cette information est désormais distribuée à travers plusieurs processeurs.

Comme pour l'algorithme bidimensionnel, la comparaison des distances à la paroi de deux grilles est réalisée avec l'interpolation tétravolumique puisque celle-ci requiert moins de temps que l'interpolation trilinéaire et puisque plusieurs tests ont montré que l'assemblage produit était le même qu'avec l'interpolation trilinéaire. La procédure utilisée pour identifier les cellules donneuses est inchangée par rapport à l'implémentation bidimensionnelle.

Identification des cellules invalides

La technique d'identification des cellules invalides détaillée à la Section 3.3.6 est étendue à sa version tridimensionnelle. Pour préserver les capacités de l'algorithme à correctement détecter ces cellules sous mémoire distribuée, il est nécessaire de communiquer à l'ensemble des membres d'un groupe chimère les paramètres définissant la boîte cartésienne qui ceinture la géométrie solide définie par le groupe.

4.3.2 Parallélisation

Le processus d'assemblage des grilles tel que présenté par l'implémentation bidimensionnelle est affecté par le fait que la mémoire soit distribuée. La principale complication qui en émerge est la nécessité d'effectuer des transferts de données d'un processeur à l'autre pour effectuer les tests de superposition. L'algorithme présenté à la section 3 ne contient pas de dépendance entre les données ce qui lui permet d'être exécuté de manière parallèle efficacement.

L'approche parallèle considérée tire son origine des travaux de Barszcz *et al.* (1993) et se divise en trois principales étapes. Ces étapes impliquant des communications entre deux grilles du système sont :

1. L'envoi des requêtes
2. Le traitement des requêtes
3. La réception des résultats

L'envoi des requêtes entre deux grilles consiste à identifier les points d'une grille pouvant présenter une superposition avec une autre grille. Il est nécessaire à ce moment de minimiser la quantité de points à analyser pour réduire au minimum les communications et la charge de travail. Pour ce faire, plusieurs filtres sont mis en place pour éliminer les points garantis de ne pas être superposés à l'autre grille. Le premier filtre est appliqué sur les maillages d'un même groupe chimère qui ne nécessitent évidemment pas d'analyser les superpositions entre eux puisqu'ils sont par définition contigus. Le deuxième filtre est la hiérarchie relative des deux grilles en question. Selon ce critère, une grille de hiérarchie supérieure est garantie d'être dominante par rapport à l'autre grille ce qui élimine la nécessité de faire les tests de superpositions. Le troisième filtre appliqué consiste à utiliser une boîte cartésienne ceinturant un maillage. Ainsi lorsqu'un maillage A détermine lesquelles de ses cellules doivent être transmises à un maillage B pour être traitées, seules les cellules contenues dans la boîte cartésienne du maillage B sont envoyées. Cette procédure permet de réduire grandement le nombre de tests de superposition devant être réalisé à l'étape 2.

Le traitement des requêtes consiste à effectuer les tests de superposition présentés au schéma

3.3. Cette procédure n'est pas affectée par la distribution de la mémoire ni par la parallélisation du code.

La dernière étape, la réception des résultats, doit prendre en compte l'effet de parallélisation pour être bien fonctionnelle. Le fait que les calculs soient effectués de manière simultanée implique de réaliser une procédure de test simple lors de la réception des données. Le problème justifiant ces tests tire son origine de l'état initial et final des données sur le processeur qui effectue l'envoi des données. Le problème est mieux illustré sous forme d'exemple. Supposons qu'une grille A soit superposé à deux grilles, les grilles B et C, et que chacune de celles-ci est enregistrée sur un processeur distinct. Les points à analyser de A sont envoyés à B et C puis les deux processeurs associés à ces grilles font l'analyse de superposition. Dépendamment des géométries des grilles, le statut d'une cellule de la grille A peut être différent sur les grilles B et C. Lors de la réception des données sur le processeur A, il est donc nécessaire de comparer le statut proposé par les deux grilles pour correctement réaliser l'assemblage puisque l'analyse proposée par une grille peut être invalidée par celle d'une autre. Cette comparaison est complexe et inclue plusieurs items de vérification. Un exemple d'intérêt démontrant bien la problématique se trouve dans l'identification des cellules donneuses. Les cellules donneuses proposées par deux grilles doivent être comparées sur la base des critères de discrimination pour correctement identifier celles-ci.

La procédure détaillée ci-haut est tirée de l'implémentation proposée par Wissink et Meakin (1997) faisant usage de communications asynchrones. Ce type de communications est nécessaire à la bonne scalabilité du programme puisqu'elle permet aux différents processeurs d'effectuer le traitement des requêtes de manière simultanée. Ainsi, les trois étapes ci-haut sont toutes réalisées de manière asynchrone. Un processeur procédera donc d'abord à l'envoi des requêtes à l'ensemble des processeurs contenant une grille pouvant présenter une superposition. Par la suite, il vérifiera si des requêtes lui ont été envoyées et dans le cas échéant les complètera toutes avant de recevoir les résultats de ses envois. Les deux dernières étapes sont réalisées de manière continuelle par tous les processeurs tant que l'ensemble de ceux-ci n'a pas reçu tous les résultats nécessaires.

Méthode d'accélération

Le même concept qu'en deux dimensions est utilisé pour l'accélération de l'assemblage tridimensionnel. La méthode utilise des grilles virtuelles récursives cartésiennes tridimensionnelles au sein desquelles les cellules sont classées. Ni la construction ni l'utilisation de la technique d'accélération ne sont affectées par la parallélisation du code puisque ces deux tâches sont

réalisées localement. Les paramètres des boîtes sont cette fois définis selon :

$$N_x = N_y = N_z = \left(\frac{Npts}{\beta}\right)^{1/3} \tag{4.3}$$

4.3.3 Balancement de charge

Contrairement aux mémoires partagées qui permettent de séparer un problème parallèle au moment de l'exécution, l'usage d'une mémoire distribuée implique de diviser les tâches avant de commencer le calcul. Pour ce faire, il est nécessaire de développer une procédure permettant de répartir la charge à travers l'ensemble des processeurs participants au calcul. Cette répartition adéquate de la charge est cruciale à la scalabilité de l'algorithme parallèle. À travers cette procédure, on tente de charger au minimum l'ensemble des processeurs disponibles.

Pour garder l'implémentation au stade démonstratif, le programme présenté utilise une division grossière telle que présentée à la section 2.5.4 pour répartir la charge sur les différents processeurs. Selon cette approche, la meilleure répartition de charge consiste à diviser géométriquement le problème de sorte à distribuer les grilles vers chacun des processeurs de la manière la plus uniforme que possible. Ceci est réalisé en identifiant la grille qui contient le plus de points et en l'associant au processeur avec la charge la plus faible. Cette procédure est répétée jusqu'à ce que l'ensemble des grilles aie été associé à un processeur.

Cette approche est évidemment simple à implémenter, mais peut limiter la capacité de l'algorithme à répartir adéquatement la charge si le système de grilles fourni ne contient pas suffisamment de divisions par rapport au nombre de processeurs utilisés. Ce problème peut être évité lors du processus de génération de maillage en divisant manuellement les grilles, mais ceci allonge le temps nécessaire à la préparation des simulations. Le solveur FANSC inclut un module ayant la capacité de subdiviser les blocs pour mieux répartir la charge sur l'ensemble des processeurs. Il est intéressant de considérer que le programme ici présenté est entièrement compatible avec ce genre de module.

4.4 Résultats

4.4.1 Entrée et sortie CGNS

La validation des capacités de l'implémentation à traiter les fichiers de type CGNS est réalisée grâce au système de grilles superposées de la géométrie DLRF6. Cette géométrie qui représente un fuselage et une aile tridimensionnels a été largement utilisée par la communauté,

notamment durant le troisième atelier de prédiction de la traînée (en anglais Drag Prediction Workshop). Ce maillage de douze blocs a été obtenu du site web de la NASA [2] et respecte le standard CGNS. Ces douze blocs modélisent :

1. le fuselage
2. le nez fuselage
3. la queue fuselage
4. la jonction de l'aile avec le fuselage
5. l'aile où le bord de fuite est droit
6. l'aile où le bord de fuite a une flèche
7. le bout de l'aile
8. le bord de fuite de l'aile
9. le bord de fuite de la jonction de l'aile avec le fuselage
10. un maillage de champ éloigné proche du fuselage
11. un maillage de champ éloigné proche de l'aile
12. un maillage de champ éloigné englobant l'ensemble des autres blocs

La validation comporte trois portions. Premièrement, il est nécessaire de montrer qu'un fichier CGNS fourni en entrée peut être lu, traité et inséré dans la mémoire du présent programme, puis réécrit en sortie uniquement à partir des informations contenues dans cette mémoire. Cette première étape valide les capacités de lecture et d'écriture des points physiques du maillage ainsi que des conditions limites qui y sont appliquées. Cette validation est relativement simple à réaliser en considérant que le programme à la capacité de lire en entrée et d'écrire en sortie les données dans un format PLOT3D. Il suffit alors de créer deux fichiers d'entrée, un en format PLOT3D et un en format CGNS, équivalents et de comparer les fichiers de sortie. La comparaison de ces deux fichiers a montré que l'implémentation produit un fichier de sortie équivalent ce qui valide cette capacité.

La deuxième étape consiste à montrer que l'information topologique contenue dans le fichier CGNS est adéquatement utilisée pour effectuer le prétraitement nécessaire à l'assemblage des grilles. Ce prétraitement consiste entre autres à construire les blocs chimère, calculer la distance à la paroi locale et identifier les cellules frontière réceptrices. La Figure 4.2 montre la distance à la paroi calculée sur la grille représentant le fuselage. On constate que les distances calculées sont bonnes ce qui valide la lecture des conditions limites du fichier CGNS. Une procédure de validation similaire a été réalisée pour l'assemblage des groupes chimères et

2. ftp ://cmb24.larc.nasa.gov/outgoing/DPW3/

pour les cellules frontières. Les résultats visualisés grâce au logiciel Tecplot ont montré que le traitement topologique était adéquat.

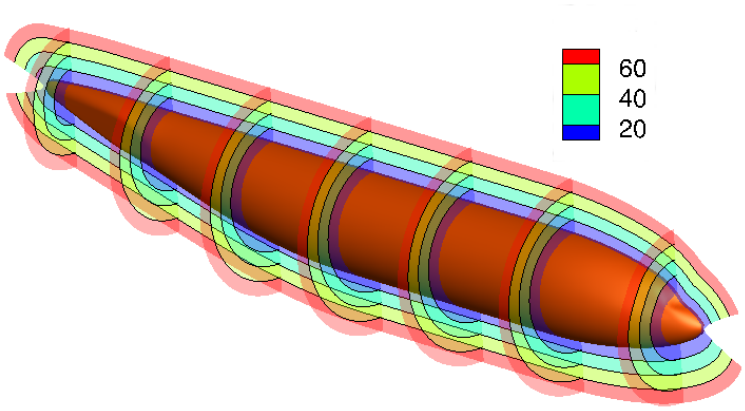


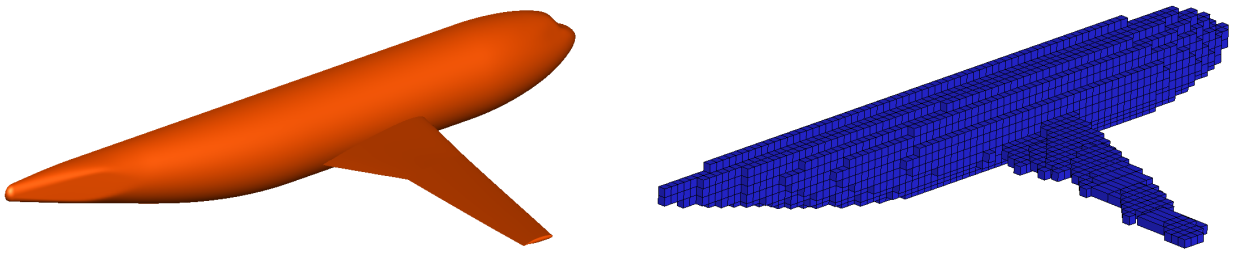
Figure 4.2 Distance à la paroi sur la grille du fuselage du DLRF6 (note : pour clarifier l'image l'affichage est limité à une distance de 80)

La troisième et dernière étape consiste à démontrer que l'enregistrement des informations créées par l'assemblage des grilles est correctement réalisé au sein du fichier de sortie. Cette portion de la validation implique donc une lecture du fichier de sortie CGNS pour valider que la communication entre les grilles est adéquatement établie. Cette validation reste à compléter.

4.4.2 Identification des cellules invalides

Comme dans le cas bidimensionnel, la validation de la capacité à détecter les géométries solides est faite de manière visuelle. Pour montrer les capacités de l'algorithme sur un cas complexe, la géométrie du DLRF6 est de nouveau utilisée. La géométrie solide telle que définie par les conditions limites est montrée à la Figure 4.3(a). La Figure 4.3(b) montre les cellules identifiées comme invalides suite à l'assemblage. On voit que la forme de la géométrie solide est répliquée par les cellules invalides validant ainsi que leur identification est fonctionnelle dans l'implémentation tridimensionnelle.

Une importante limitation de la technique de détection des cellules invalides existe et devient critique sur les configurations à trois dimensions. On rappelle au lecteur que la technique d'identification utilisée procède par élimination en exploitant deux aspects : 1) la présence de la grille qui définit la géométrie solide, 2) la boîte cartésienne ceinturant la géométrie solide. Ces deux éléments sont illustrés à la Figure 4.4 où la boîte cartésienne est identifiée en lignes tiretées rouges et les parois solides en noir. On y voit que le maillage ne couvre pas



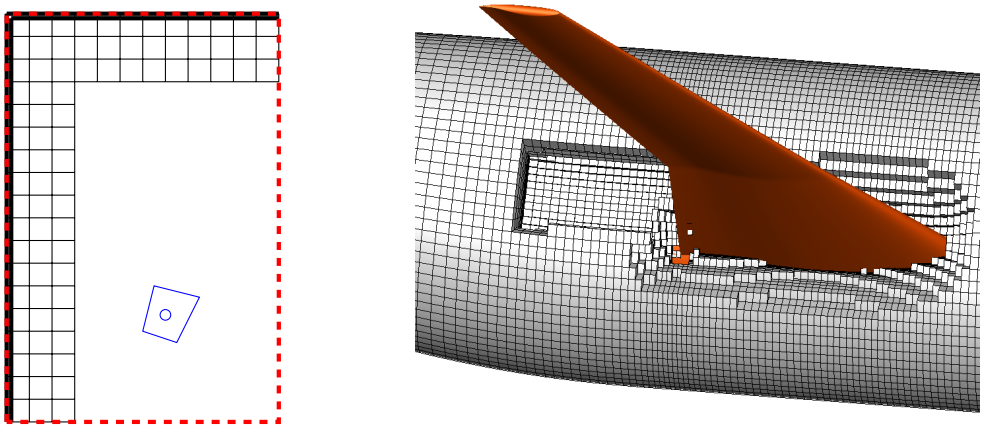
(a) géométrie solide (b) cellules invalides

Figure 4.3 Identification des cellules invalides sur le maillage du DLRF6

complètement la boîte cartésienne rouge ce qui consitue l'origine du problème. En effet, on voit que la cellule bleue, en passant à travers le processus d'assemblage de la Figure 3.3, est identifiée incorrectement comme une cellule invalide. La situation est problématique dans le sens où ces types de configuration mènent à des trous dans l'assemblage des grilles ce qui implique que la résolution du fluide n'est pas faite en ces endroits.

Sur le système de grilles du DLRF6, un seul bloc présente cette problématique. Il s'agit d'un bloc très localisé qui couvre le bord de fuite tout au long de l'envergure de l'aile. Ce bloc, présenté à la Figure A.1 de l'Annexe A, est utilisé pour obtenir une haute résolution à l'endroit du bord de fuite d'où sa proximité à la géométrie qu'il défini. Le fait que ce maillage soit très localisé fait en sorte qu'il ne couvre pas la boîte contour de sa géométrie solide ce qui créer un trou dans l'assemblage des grilles. Le problème est démontré à la Figure 4.4(b) où on voit que le maillage du fuselage présente un trou à l'arrière de l'aile. Pour pouvoir valider de manière appropriée l'implémentation tridimensionnelle, ce bloc a été retiré du système de grilles. Les assemblages présentés plus tôt sont valides puisque l'ensemble de la géométrie de l'avion est entièrement défini par les maillages restants. Autrement dit, la géométrie solide définie par le bloc problématique est aussi définie par un autre bloc qui ne présente pas le problème identifié.

Cette situation peut être évitée en s'assurant, à travers l'ajout de cellules supplémentaires, que le maillage définissant une géométrie solide couvre la totalité de la boîte contour de la géométrie solide. En deux dimensions, cet ajout ne constitue pas un problème significatif puisqu'il n'implique qu'un faible nombre de cellules. Par contre, en trois dimensions cette nécessité n'est pas désirable puisque l'ajout de cellules est plus important ce qui peut augmenter significativement la taille des maillages et le temps de calcul nécessaire à l'assemblage. De plus, ce concept impose une contrainte supplémentaire lors du processus de génération de maillage.



(a) illustration d'une mauvaise identification d'une cellule invalide

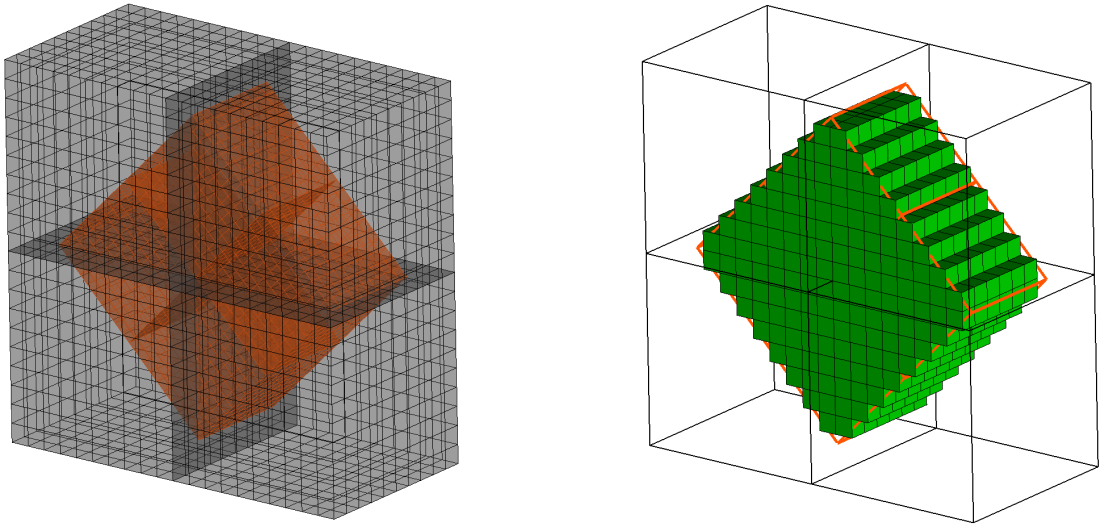
(b) trou de cellules invalides suite à l'assemblage du système de grille du DLRF6

Figure 4.4 Identification innapropriée de cellules invalides

4.4.3 Identification des cellules interpolées

La validation des cellules interpolées est aussi faite visuellement. On vise ici à valider l'implémentation des critères hiérarchique et de distance à la paroi locale. Le cas du DLRF6 ne représente pas un cas optimal pour valider le premier critère puisque la conception du système de grilles a été davantage orientée vers une utilisation du critère de distance à la paroi locale. Ainsi pour valider en trois dimensions le critère hiérarchique, la grille bidimensionnelle utilisée à cette fin est recréée ici en version tridimensionnelle. Le système de grille superposé est montré à la Figure 4.5(a) où le maillage oblique domine hiérarchiquement le maillage noir. La Figure 4.5(b) montre les cellules interpolées du maillage dominé en vert et les contours du maillage oblique en orange. On y voit que les cellules superposées au maillage dominant sont correctement identifiées comme dominées, et donc interpolées, ce qui valide l'implémentation du critère hiérarchique.

La validation du critère de distance à la paroi locale est réalisée sur le maillage du DLRF6 puisqu'il s'agit d'un cas tridimensionnel complexe idéal pour bien valider ce critère. Le maillage utilisé a été développé de manière à réduire au minimum les superpositions inutiles puisque celles-ci ne contribuent pas au calcul du fluide. De ce fait, seulement deux blocs comportent une forte superposition. Il s'agit du bloc de forme cylindrique qui définit le fuselage et du bloc de configuration «C» qui définit l'aile à sa jonction avec le fuselage. Ces deux blocs sont tous deux de même hiérarchie et doivent donc déterminer le statut de leurs cellules uniquement grâce à la distance à la paroi locale. Puisque ces deux blocs ne font pas partie d'un même groupe chimère, leur distance à la paroi ne considère que l'élément



(a) Système de grilles utilisé pour valider le critère hiéarachique

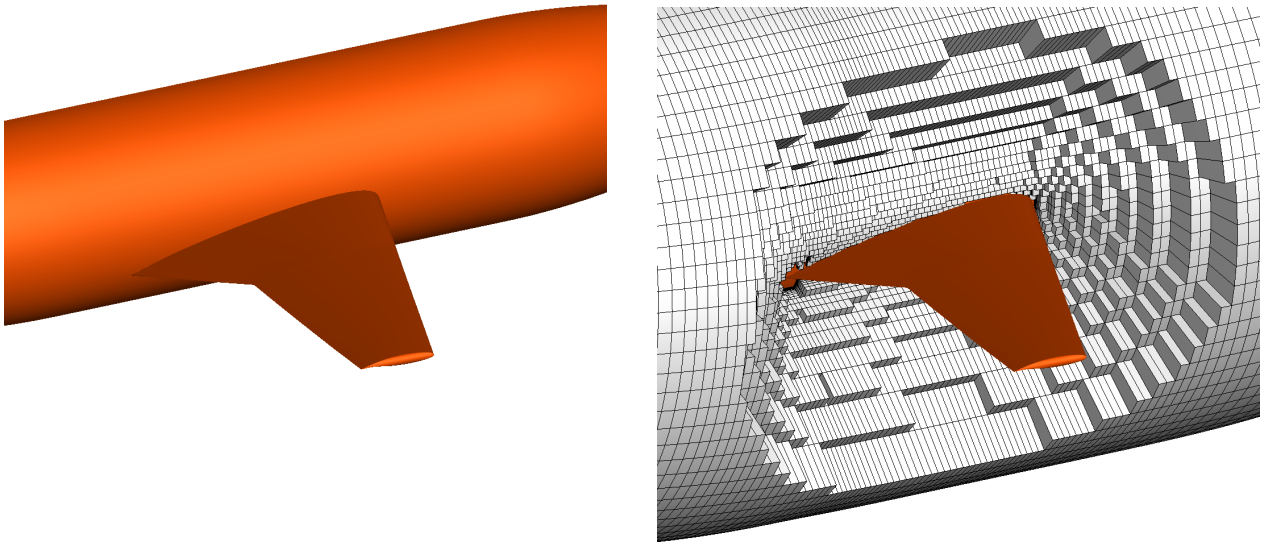
(b) Les cellules interpolées sont identifiées en vert

Figure 4.5 Validation du critère hiérarchique

qu'ils définissent. Ainsi, la distance à la paroi du bloc définissant le fuselage ne prend pas en compte la présence de l'aile comme il est visible à la Figure 4.3. La même situation s'applique à l'aile. Ainsi lors de l'assemblage des blocs on s'attend à ce que la découpe des grilles dans cette région soit faite de manière conique autour de la jonction de l'aile et du fuselage. La Figure 4.6(a) montre la géométrie solide proche de cette jonction et la Figure 4.6(b) montre les cellules calculées du maillage du fuselage. On y voit que le maillage, originalement continu, est effectivement découpé de manière conique par la grille qui définit l'aile. Quoique non montrée, la grille définissant l'aile forme le négatif du trou observé dans le maillage du fuselage. Cette démonstration valide l'implémentation du critère de la distance à la paroi en trois dimensions.

4.4.4 Zone tampon

Afin d'éviter la création d'un système d'interpolation implicite, l'algorithme implémenté ajoute une zone tampon comme dans le cas bidimensionnel. Pour généraliser l'implémentation à des schémas numériques d'ordre variés, le nombre de rangées définissant la zone tampon est un paramètre ajustable par l'utilisateur avant l'assemblage. La Figure 4.7 montre l'ajout de la zone tampon, large d'une cellule, sur le maillage qui définit le fuselage. Les cellules composant la zone tampon y sont identifiées en vert. En comparant cette figure à la Figure 4.6(b), on voit que l'ajout de ces cellules est correctement effectué dans les trois dimensions validant ainsi cet outil dans sa version tridimensionnelle.



(a) (b)

Figure 4.6 Validation du critère de la distance à la paroi locale

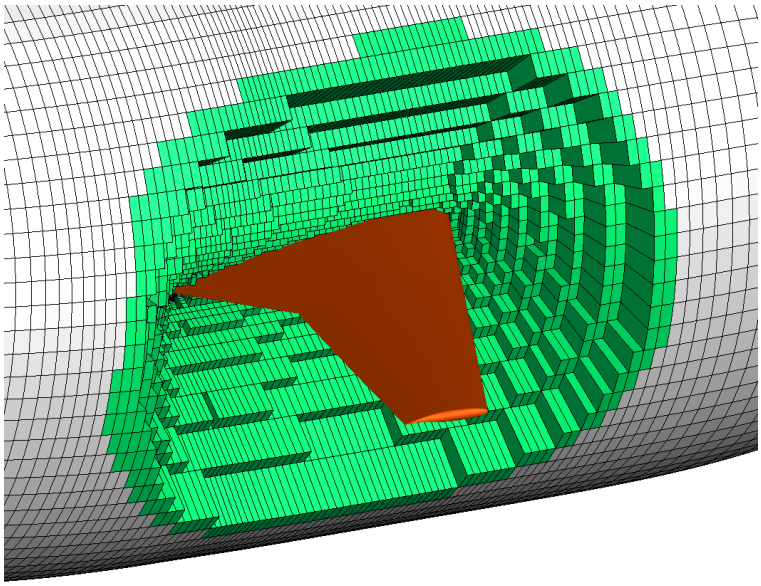


Figure 4.7 Ajout de la zone tampon sur le maillage définissant le fuselage

4.4.5 Méthode d'accélération

La technique des grilles virtuelles est appliquée à un cas canonique. Les maillages qui ont été utilisés pour valider la méthode d'accélération bidimensionnelle sont étendus à leur version tridimensionnelle. Les maillages sont générés de manière à bien représenter une couche limite dans les trois dimensions créant ainsi d'importants changements de densité des cellules. Les

maillages contiennent 20 cellules dans chacune des directions. À nouveau, ces maillages sont superposés sans décalage et on étudie la performance de l'algorithme en termes de temps par rapport au paramètre β et au niveau de récursivité des grilles virtuelles. Le nombre critique permettant de déclencher les grilles récursives est le même, 100, puisque des tests ont montré que ce paramètre était aussi optimal en trois dimensions.

La Figure 4.8 montre le temps requis pour assembler les maillages en fonction du paramètre β pour différents niveaux de récursivité (noter que les deux échelles sont logarithmiques). La courbe noire et les courbes titrées «Niveau i» correspondent au temps d'assemblage respectivement obtenu grâce à l'utilisation de grilles virtuelles non récursives et avec l'utilisation de grilles virtuelles récursives de niveau i. Le graphique montre que les propriétés obtenues dans l'implémentation bidimensionnelle sont préservées en trois dimensions. On voit donc que la zone optimale sans grilles récursives est restreinte, mais que les grilles récursives permettent d'élargir cette zone. À nouveau, l'utilisation d'un nombre critique fixé à 100 assure aux grilles récursives de haut niveau de fournir le temps d'assemblage le plus faible. La comparaison des temps d'assemblage à β optimal montre une réduction de 70% lorsque les grilles de récursivité de niveau 9 sont utilisées.

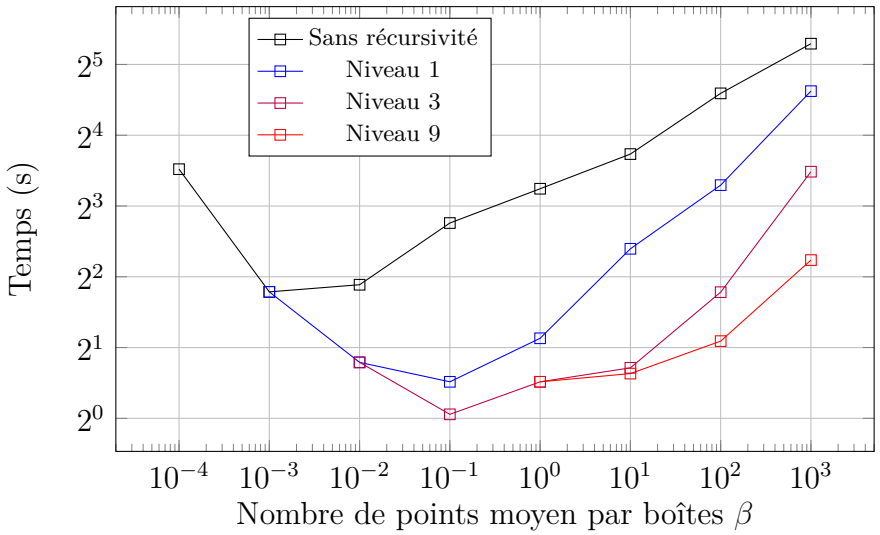


Figure 4.8 Temps d'assemblage en fonction du paramètre β.

La méthode d'accélération récursive ne présente pas une bonne performance sur l'assemblage des grilles du cas industriel du DLRF6. Ceci s'explique simplement par le fait que ce système de grille comporte très peu de superpositions. Ainsi, la classification des points du maillage dans les grilles récursives est seulement bénéfique pour une faible portion de l'assemblage. Les gains apportés durant le processus d'assemblage ne sont pas suffisants pour compenser

le temps requis pour mettre en place les grilles récursives. Ce problème avait été identifié plus tôt dans l'implémentation bidimensionnelle. On a vu que sur le profil MDA les gains en temps dû à l'utilisation de grilles récursives sont minimes. Le problème du DLRF6 est le même, mais accentué par le très faible taux de superposition. Dans ce cas, à β optimal, l'utilisation de grilles récursives amène au mieux à une perte de performance de 20%.

4.4.6 Ordre de précision

L'ordre de précision de l'implémentation tridimensionnel est maintenant vérifié selon la même approche qui fut précédemment présentée. La procédure permet de valider que l'association entre les cellules interpolées et leurs donneuses est adéquate. L'étude est réalisée sur deux grilles superposées utilisant un critère hiérarchique qui permet de s'assurer qu'une seule grille soit systématiquement dominante. L'évaluation implique à nouveau une étude de raffinement de maillage effectuée sur une famille de cinq assemblages de grilles comme celui montré à la Figure 4.5(a) où le maillage oblique domine hiérarchiquement le maillage noir. Les valeurs minimales et maximales des coordonnées dans les trois directions sont respectivement de -20 et 20. Les deux grilles les plus grossières contiennent 20 cellules sur chacun de leurs côtés. En subdivisant systématique la taille des cellules par deux, les deux grilles les plus fines contiennent 320 cellules sur chacun de leurs côtés.

L'ordre de précision est calculé pour les fonctions d'interpolation tétravolumique et trilinéaire. La fonction analytique non linéaire utilisée pour calculer l'erreur d'interpolation est :

$$f(x,y) = cos\left(\frac{\pi x}{30}\right) * sin\left(\frac{\pi y}{30}\right) * cos\left(\frac{\pi z}{30}\right) \tag{4.4}$$

Cette fonction est bien adaptée aux grilles puisque sa variation est basse par rapport à la taille des cellules. Suite à l'assemblage, l'interpolation de cette fonction est effectuée sur les cellules réceptrices du maillage dominé. La définition de l'erreur d'interpolation et la procédure permettant de calculer l'ordre de précision sont les mêmes qu'à la section 3.4.5.

Le tableau 4.1 montre les résultats de l'étude. Contrairement à l'implémentation bidimensionnelle, on note d'abord une forte dissimilitude entre les résultats des deux fonctions d'interpolation. On voit dans les colonne F que la fonction tétravolumique produit une erreur d'interpolation beaucoup plus élevée que la fonction trilinéaire. Par contre, le fait que l'erreur des deux fonctions d'interpolation diminue systématiquement lorsque les grilles sont raffinées montre que l'identification des cellules donneuses est réalisée correctement. Les résultats indiquent que la précision de l'assemblage produit est de deuxième ordre pour la fonction trilinéaire et près du premier ordre pour la fonction tétravolumique. On peut expliquer ce

Tableau 4.1 Erreur d'interpolation et ordre de convergence estimé de l'implémentation tridimensionnelle

Maillage		Tétravolumique			Trilinéaire		
NC	Npts	F	R	p	F	R	p
20	4000	5.51E+1	-	-	3.63E-1	-	-
40	32000	2.46E+1	-	-	8.57E-2	-	-
80	256000	1.32E+1	0.375	1.41	2.19E-2	0.230	2.30
160	2048000	7.01E-0	0.544	0.88	5.49E-3	0.257	1.96
320	16384000	-	-	-	1.38E-3	0.251	1.99
-	-	-3.86E-1	-	-	1.44E-6	-	-

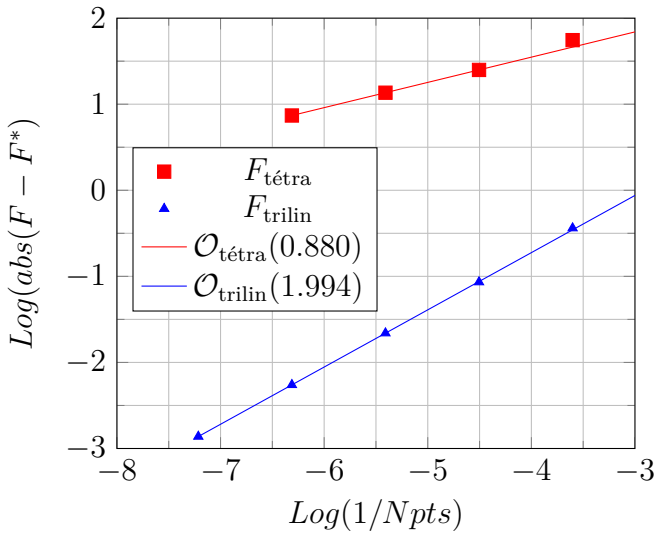


Figure 4.9 Représentation visuelle des résultats du Tableau 4.1

comportement par le nombre de points utilisés pour effectuer l'interpolation. En effet, l'interpolation bilinéaire/trilinéaire passe de quatre points en deux dimensions à huit points en trois dimensions alors que l'interpolation tétravolumique passe de trois points en deux dimensions à quatre points en trois dimensions. Cette faible augmentation du nombre de points n'est pas suffisante pour garder l'ordre de précision puisque trop peu d'information est utilisée pour réaliser l'interpolation. Les résultats sont illustrés à la Figure 4.9 où les données discrètes sont représentées par les symboles pleins. Les droites représentent les extrapolations vers $h \to 0$. La proximité des données discrètes à cette droite montre que les résultats obtenus sont encore une fois indépendants du maillage, validant les ordres de précision calculés. Puisque la fonction d'interpolation trilinéaire est d'ordre deux, c'est celle qui est retenue.

4.4.7 Parallélisation

La procédure de la validation de l'implémentation parallèle est similaire à celle en deux dimensions. Puisque les contraintes quant à la répartition de la charge sont les mêmes, un système de grilles similaire est utilisé pour cette validation. On génère donc deux grilles cartésiennes identiques de 16 blocs chacune, superposées l'une sur l'autre sans décalage. Chacun des 16 blocs est formé de $80 \times 80 \times 40$ cellules et comme auparavant, le critère de discrimination utilisé est la distance à la paroi. Une condition limite de paroi est appliquée à une des faces de la grille A et à la face opposée sur la grille B. Pour assurer une bonne répartition, le nombre de processeurs utilisés pour le calcul doit être un diviseur de 32. Ainsi, les assemblages sur 1, 2, 4, 8, 16 et 32 processeurs sont ici considérés. La machine utilisée pour réaliser les tests de parallélisation exploite des noeuds de deux processeurs Intel Westmere-EP X5650 hexa-coeur à 2.667GHz.

Tableau 4.2 Temps d'assemblage et facteur d'accélération en fonction du nombre de processeurs

Nb procs	Temps (s)	Accélération
1	51.4	1.00
2	25.2	2.04
4	12.8	4.00
8	7.18	7.16
16	3.83	13.4
32	2.03	25.3

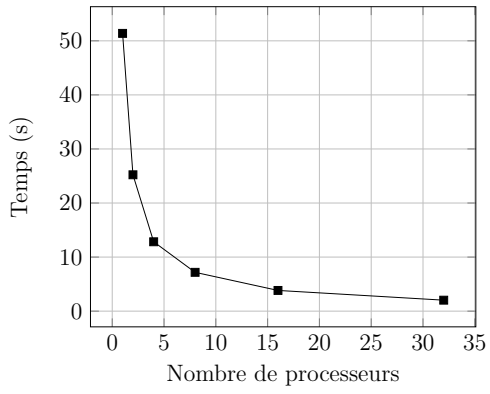


Figure 4.10 Temps requis pour assembler les grilles de 16 blocs

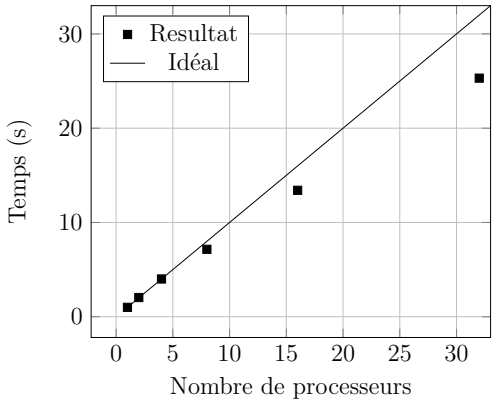


Figure 4.11 Accélération de l'assemblage en fonction du nombre de processeurs

Comme pour l'implémentation bidimensionnelle, un examen détaillé de ces six assemblages a montré que les résultats obtenus sur 1, 2, 4, 8, 16 et 32 processeurs sont rigoureusement

identiques démontrant ainsi le succès de l'implémentation du point de vue du traitement des données. Les temps d'assemblage sont présentés au Tableau 4.2 et présentés graphiquement aux Figures 4.10 et 4.11. Les deux figures montrent des tendances très similaires à l'implémentation bidimensionnelle. À nouveau, le temps d'assemblage diminue systématiquement lorsque davantage de processeurs sont appliqués au calcul, démontrant ainsi les capacités de calculs parallèles. La scalabilité du programme est représentée à la Figure 4.11 où le facteur d'accélération est comparé à la droite qui indique la performance qui serait obtenue si le programme était exécuté de manière entièrement parallèle. On voit que le gain en performance diminue légèrement lorsque des processeurs sont ajoutés au calcul. Selon la loi d'Amdahl, l'implémentation ci-présentée offre un taux de parallélisme de 99.2% ce qui permet un facteur d'accélération maximal d'environ 120. Ces résultats montrent que l'implémentation parallèle tridimensionnelle présente aussi d'excellentes caractéristiques de scalabilté.

CHAPITRE 5 CONCLUSION

Cette section offre un sommaire des travaux exécutés et présente un regard critique sur les limites de la solution proposée. De possibles avenues futures sont proposées et évaluées en terme de priorité et de complexité.

5.1 Synthèse des travaux

Une démonstration de la maturité de la technologie chimère pour des applications industrielles modernes du domaine aéronautique a été réalisée. Les difficultés rencontrées lors du traitement de géométries complexes ont été présentées. La discussion a détaillé les motifs économiques et techniques justifiant l'utilisation de la méthode et la manière avec laquelle la superposition de grilles permet d'aborder les difficultés en question. Les principaux éléments à considérer durant l'implémentation de la technique chimère et l'approche envisagée pour gérer ces éléments ont été abordés avant de présenter les objectifs du projet. Les objectifs consistaient à développer la technologie chimère d'abord en version bidimensionnelle au sein du code de recherche NSCODE et ensuite en version tridimensionnelle dans un module indépendant de sorte qu'il puisse être arrimé à n'importe quel solveur de fluide tridimensionnel ayant les capacités chimères. Les travaux visaient aussi à identifier les défis de telles implémentations. Ces défis seront présentés aux Sections 5.2 et 5.3 ci-bas.

5.1.1 Implémentation bidimensionnelle

La démonstration de la technologie fut d'abord réalisée par son implémentation au sein de NSCODE, la plateforme de recherche du laboratoire du professeur Éric Laurendeau. Le concept de groupe chimère présenté au début de cette section a permis de réduire significativement le temps de calcul nécessaire à l'assemblage d'un système de grilles. Pour pouvoir transférer l'information d'un maillage à un autre, une nouvelle condition limite a été introduite, la condition limite chimère. La méthode utilisée pour détecter les superpositions de cellules a été présentée. Deux critères ont été mis en place pour permettre l'identification des cellules interpolées : le critère hiérarchique et le critère de distance à la paroi locale. Ces deux critères ont permis d'automatiser le processus d'assemblage de grilles en minimisant le traitement requis par l'utilisateur. Il a été montré comment la combinaison de ces deux critères peut mener à des assemblages adaptés aux phénomènes physiques pour lesquels les grilles sont utilisées. Un algorithme parcourant l'ensemble des grilles superposées et permet-

tant d'identifier les cellules donneuses a été implémenté. Il a été expliqué pourquoi les cellules donneuses devaient préférablement être des cellules calculées, ce qui a mené à créer une zone tampon utile pour produire un système d'interpolation explicite. La communication entre les grilles a été vérifiée grâce à une étude de raffinement de maillage qui a démontré que les interpolations tétravolumique et bilinéaire produisaient des assemblages de deuxième ordre. Une technique de base permettant d'identifier les cellules invalides en exploitant la présence du maillage définissant une géométrie solide a été présentée. Les résultats ont montré que, malgré sa simplicité, la technique était adaptée aux géométries complexes du domaine de l'aéronautique en deux dimensions.

La technique de grilles virtuelles commune a été appliquée, mais ses limitations dans le cas de maillage présentant de forts changements de densité ont poussé à considérer une technique novatrice visant à utiliser le principe des grilles virtuelles de manière récursive. Les résultats sur le cas canonique ont montré un gain de temps marginal sur l'assemblage. Lorsqu'appliquée au cas industriel, la méthode récursive s'est montrée plus flexible que l'approche standard, mais n'a fourni qu'un faible gain en temps dû au plus faible taux de superposition. La parallélisation de l'implémentation bidimensionnelle a été réalisée avec l'interface de mémoire partagée OpenMP. Le programme a montré une bonne performance parallèle notamment en fournissant des caractéristiques de scalabilité intéressantes.

L'intégration du module d'assemblage avec NSCODE a consisté en une légère modification des routines de mise à jour des variables de l'écoulement. Il a été montré que les assemblages produits par le programme préservaient l'ordre de précision de NSCODE et assuraient une bonne communication des variables de l'écoulement dans les zones superposées. Il a aussi été montré que l'utilisation de grilles superposées pouvait mener à des résultats plus proches des valeurs expérimentales grâce à l'utilisation de maillages mieux adaptés à la géométrie.

5.1.2 Implémentation tridimensionnelle

Un module tridimensionnel a été développé en tant que module indépendant d'un solveur de fluide. Ce module, distinct du programme bidimensionnel, visait à présenter la performance de l'algorithme d'assemblage de grilles superposées pour des applications davantage axées sur les besoins industriels. Ces besoins consistent notamment en la compatibilité avec un standard de données d'entrée et de sortie, les capacités d'assemblage en trois dimensions et la parallélisation sur mémoire distribuée. Le module développé est notamment compatible avec le solveur de fluide de Bombardier Aéronautique, FANSC, et utilise majoritairement les mêmes concepts et approches que l'implémentation bidimensionnelle. Les différences entre les deux programmes émergent de la considération de la troisième dimension et du mode de

parallélisation. L'utilisation du standard CGNS a nécessité une adaptation de la procédure d'identification des groupes chimères puisque l'information de la topologie et du maillage est contenue au sein d'un seul fichier CGNS. Cette identification est faite de manière automatique en se basant sur l'information topologique de connexions inter blocs. Le code présenté a la capacité d'utiliser en entrée et de produire en sortie un fichier selon le format CGNS. La validation de l'écriture des informations de communications entre les grilles au sein du fichier CGNS de sortie n'a pas été réalisée. D'autre part, comme la mémoire de l'implémentation tridimensionnelle est partagée, il a été nécessaire de communiquer les coordonnées des parois solides aux processeurs faisant partie d'un même groupe chimère pour correctement calculer la distance à la paroi et les propriétés des géométries solides. Sauf quelques modifications mineures, les techniques de détection du statut des cellules et des cellules donneuses sont inchangées. L'implémentation des critères de hiérarchie et de distance à la paroi locale a été validée sur la géométrie du DLRF6 tout comme l'ajout de la zone tampon en trois dimensions. La communication adéquate entre les grilles a été à nouveau vérifiée grâce à une étude de raffinement de maillage à travers laquelle il a été observé que les fonctions d'interpolation tétravolumique et trilinéaire produisaient respectivement des assemblages de premier et de deuxième ordre. Pour cette raison, la fonction trilinéaire est retenue dans le programme final.

La technique d'accélération des grilles virtuelles récursives a été développée vers sa version tridimensionnelle puis testée sur un cas canonique où les capacités d'accélération de l'algorithme récursif se sont avérées très bonnes. L'utilisation de l'approche récursive sur le système de grilles de la géométrie DLRF6 n'a pas présenté une réduction du temps d'assemblage, mais plutôt une augmentation de celui-ci puisque le système présente un très faible taux de superposition.

La parallélisation sous mémoire distribuée a affecté considérablement la procédure d'assemblage des grilles. Dans cette implémentation, l'assemblage est réalisé en trois phases distinctes : l'envoi des requêtes, le traitement des requêtes et la réception des résultats. Pour limiter l'analyse nécessaire, trois filtres ont été utilisés lors de l'envoi des requêtes. Le traitement des requêtes est inchangé par rapport à sa version bidimensionnelle, mais l'étape de réception des résultats a nécessité quelques considérations. Ainsi, la procédure d'identification des cellules donneuses et des cellules invalides a été modifiée pour s'adapter au synchronisme du processus d'assemblage. Par ailleurs, l'algorithme de répartition de charge implémenté consiste à assigner de manière successive les blocs contenant le plus de points aux processeurs les moins chargés. Cette approche est suffisante pour répartir la charge si les divisions du problème sont égales. Les tests ont montré que lorsque la charge est bien balancée, l'implémentation parallèle tridimensionnelle présente d'excellentes caractéristiques de scalabilité ce qui lui permet d'être utilisée sur une grande quantité de processeurs.

5.2 Limitations de la solution proposée

Tel que présenté à la Section 4.4.2, l'aspect le plus limitatif de l'implémentation tridimensionnelle est sa technique d'identification des cellules invalides. En effet, dans les zones de superposition, si le maillage d'une géométrie solide ne contient pas la totalité de sa boîte contour, une identification erronée des cellules invalides se produit. Un assemblage avec des cellules invalides erronées implique que le fluide n'est pas calculé à un endroit ou il devrait l'être, ce qui fausse les résultats de la simulation numérique. La technique utilisée dans les implémentations souffre de sa simplicité, c'est-à-dire du fait qu'elle exploite la présence de la grille qui définit la géométrie solide. Le problème peut être évité en créant un maillage suffisamment vaste pour englober de manière cartésienne la géométrie solide. Cette approche est très peu souhaitable pour les maillages tridimensionnels puisque cet ajout implique une importante augmentation du temps de calcul autant pour l'assemblage des grilles que pour le calcul du fluide. La limitation existe aussi au sein de l'implémentation bidimensionnelle, mais est beaucoup moins contraignante puisque l'ajout de cellules en deux dimensions n'implique qu'une faible augmentation du temps de calcul. Dans un contexte de recherche utilisant la plateforme NSCODE, la solution consistant à ajouter des cellules est appropriée puisque seulement deux dimensions sont considérées. Pour un contexte industriel, il serait par contre préférable d'implémenter une approche plus générale.

La deuxième importante limitation, spécifique à l'implémentation tridimensionnelle, est la capacité à distribuer la charge de travail de manière uniforme sur l'ensemble des processeurs participants au calcul. Cet aspect est d'intérêt puisque dans sa configuration actuelle, le programme ne peut pas nécessairement garantir une bonne performance en termes de temps lorsque le calcul est effectué sur plusieurs processeurs puisque seules les divisions de charge selon les blocs du maillage sont permises. On constate rapidement deux problèmes avec ce type de division. D'abord, le nombre de processeurs utilisés pour le calcul est limité au nombre de blocs constituant le système ce qui implique de porter une attention particulière durant le processus de génération de maillage pour décomposer le maillage en une multitude de blocs. Cette considération est contraire à la technique chimère qui vise à entre autres à éliminer les contraintes lors du processus de génération de maillages. Le deuxième problème de la technique est que le fait de diviser manuellement le maillage en plusieurs blocs ne garantit pas que la charge sera bien répartie. Pour illustrer ceci, on peut considérer le cas extrême d'un bloc d'un million de cellules devant être assemblé avec un bloc de mille cellules avec deux processeurs. Clairement, la charge de travail associée au bloc d'un million de cellules devrait être répartie sur les deux processeurs. Or, l'algorithme actuel ne permet pas une telle répartition. Un module de partitionnement de bloc, comme celui intégéré dans FANSC,

règlerait le problème.

La troisième limitation se trouve dans l'implémentation de la méthode d'accélération par grilles virtuelles qui présente deux points d'intérêt à la présente discussion. Premièrement, le fait que le nombre de divisions soit le même dans toutes les directions limite l'adaptabilité de la grille virtuelle au maillage à assembler. À titre d'exemple, on peut considérer le cas extrême d'un maillage cartésien de 10000×10 cellules pour lequel on impose la valeur de 10 points en moyenne par boîtes. Dans ce cas, N_x et N_y serait 100. Or, puisqu'il n'y a que dix cellules dans la deuxième direction, cette division de la grille virtuelle est inappropriée. Deuxièmement, comme il a été constaté en deux dimensions sur l'assemblage du profil MDA, mais surtout en trois dimensions sur la géométrie DLRF6, le faible taux de superposition d'un système de grilles réduit la performance de l'algorithme des grilles virtuelles puisqu'une partie du temps investi à construire ces grilles est gaspillé. En effet, puisque les grilles virtuelles sont utilisées pour accélérer la recherche de superposition, leur construction dans les zones où il n'y a pas de superpositions n'amène pas de réduction du temps de calcul, mais contribue plutôt à l'augmentation de ce temps. Cet effet est d'autant plus important lorsque les grilles virtuelles récursives sont utilisées. Cette limitation est importante puisque pour réduire les requis en mémoire et le temps de calcul d'une simulation d'un fluide, il est préférable de générer des systèmes de grilles avec un faible taux de superposition.

5.3 Améliorations futures

Six axes de développement considérés prioritaires par l'auteur sont maintenant présentés. Ces développements se rassemblent sous trois branches principales :

1. Fonctionalité d'assemblage
   - Technique d'identification des cellules invalides
   - Interactions de solide à solide
2. Performance de l'algorithme
   - Balancement de charge
   - Méthode des grilles virtuelles
   - Réduction des communications
3. Qualité de l'assemblage
   - Procédure de vérification de l'assemblage

5.3.1 Fonctionnalités d'assemblage

Technique d'identification des cellules invalides

L'axe de développement le plus important consiste en l'amélioration de la technique d'identification des cellules invalides pour des applications industrielles. Selon les besoins d'analyse, deux des techniques présentées à la Section 2 sont considérables. La technique de remplissage utilisée dans (Prewitt *et al.*, 2000) et (Rogers *et al.*, 2003) est capable de traiter les géométries complexes du domaine aéronautique et semble raisonnablement simple d'implémentation. Si le besoin est axé sur la performance en termes de temps, l'algorithme «X-rays» tel que présenté par (Chan *et al.*, 2012) est plutôt suggéré, mais implique une implémentation plus complexe.

Interactions solide à solide

La capacité à traiter les interactions de solide à solide, ou autrement dit, de permettre qu'un solide en pénètre un autre est d'intérêt, surtout dans un cadre industriel. Cette fonctionnalité est souhaitable puisque les interactions de solide à solide apparaissent fréquemment lors de simulations d'éléments en mouvement. Rappelons que la réalisation de ce type de simulation est un des principals facteurs de motivation pour l'utilisation de la méthode chimère. Dans la présente implémentation, un système de grilles constitué d'un profil et d'un aileron est correctement assemblé si les deux solides ne sont pas en contact. Par contre s'il existe un point de contact entre les solides, comme c'est le cas sur un aéronef physique, l'assemblage produit contiendra des erreurs et ne permettra pas de réaliser la simulation de manière adéquate. La méthode la plus répandue pour traiter ce type d'interaction est la technique «patch grid» dont Schwarz *et al.* (2014) en font une bonne présentation.

5.3.2 Performance

Balancement de charge

Pour garantir une bonne performance lors de calculs parallèles, un algorithme de balancement de charge basé sur une division fine doit être implémenté. La division fine a été identifiée par la vaste majorité des auteurs du domaine comme la technique de choix pour bien répartir la charge. Elle permet de subdiviser un bloc afin d'en assigner chacune des subdivisions à un processeur. L'algorithme proposé par Wissink et Meakin (1997) est suggéré puisque l'approche est très simple d'implémentation et offre une répartition de la charge assez uniforme. L'idée consiste à déterminer le nombre de points moyen par processeurs à partir du nombre

de points total du système et du nombre de processeurs. On détermine ensuite le nombre de processeurs à appliquer à chacun des blocs du système et on subdivise les blocs selon des axes qui réduisent au minimum les communications requises dans un système à mémoire partagée. Les auteurs proposent aussi dans le même article une technique pour adapter la charge lors de simulations avec éléments en mouvement.

Méthode des grilles virtuelles

Suivant les limitations précédemment identifiées, la méthode des grilles virtuelles devrait être modifiée de sorte à permettre un nombre différent de divisions dans chacune des directions. Cet aspect est particulièrement crucial dans le cas de grilles représentant des extrusions, comme une aile tridimensionnelle par exemple. Typiquement, pour que le temps de résolution du fluide soit acceptable, le maillage d'une aile tridimensionnelle va contenir beaucoup de points dans un plan (le plan du profil) et relativement peu selon l'envergure. Cette configuration présente précisément les caractéristiques qui font en sorte que trop de divisions sont générées selon l'axe de l'envergure ce qui réduit quelque peu l'efficacité de la méthode des grilles virtuelles. À la connaissance de l'auteur, la littérature ne présente aucune technique permettant de déterminer de manière automatique le nombre de divisions dans chacune des directions. Une approche basée sur un échantillonnage de la répartition spatiale des points du maillage est envisageable.

Dans un deuxième temps, la technique des grilles virtuelles récursives doit être adaptée aux faibles taux de superposition. En effet, une réduction importante du temps d'assemblage peut être obtenue si les grilles récursives sont construites uniquement dans les endroits de superpositions. Pour identifier ces endroits, (Zagaris *et al.*, 2010) proposent de définir grossièrement une zone de superposition basée sur les coordonnées minimales et maximales d'un bloc tel que présenté à la Figure 5.1. Cette approche, simple d'implémentation, permettrait à la technique des grilles virtuelles récursives d'offrir une performance plus proche de celle observée sur les cas canoniques.

Réduction des communications

En observant la Figure 5.2, où les cellules calculées et interpolées sont respectivement bleues et vertes, et en considérant la molécule de calcul d'une cellule calculée, on voit que seulement quelques cellules interpolées sont nécessaires proche de la frontière. En effet, même si les cellules interpolées loin de la frontière reçoivent de l'information de leurs donneuses, celle-ci n'est jamais utilisée pour calculer les flux dans le solveur du fluide. Cette situation implique une perte d'efficacité due au transfert inutile de ces informations. Pour identifier les cellules

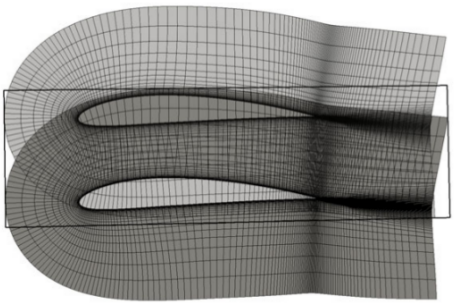


Figure 5.1 Zone de superposition approximée

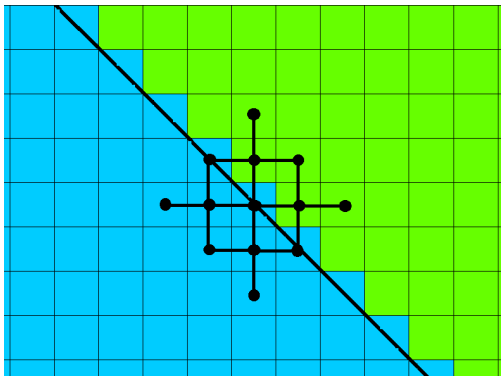


Figure 5.2 Frontière des cellules calculées et interpolées. La droite oblique représente un maillage dominant

utiles au calcul, une procédure quasi identique à celle permettant de créer la zone tampon peut être utilisée. En effet, en faisant grandir de manière contrôlée une bande vers l'intérieur de la zone d'interpolation, on peut facilement garantir l'intégralité de la molécule de calcul en négligeant toutes les cellules interpolées au-delà de ladite bande.

5.3.3 Qualité de l'assemblage

Finalement, une procédure de vérification de l'assemblage devrait aussi être développée. Une telle procédure devrait d'abord vérifier que le système construit est bel et bien explicite, ce qui implique de vérifier que chaque cellule donneuse est dominante. Ensuite, il est nécessaire d'identifier les cellules orphelines. Ces cellules sont des cellules interpolées n'ayant pas trouvé de donneuses. Cette situation, assez commune, peut se présenter par exemple si une cellule d'une frontière de condition limite chimère se retrouve à l'intérieur d'une géométrie solide. Une troisième portion de la vérification devrait vérifier l'intégrité des molécules de calcul. Pour chaque cellule calculée, il faut s'assurer qu'aucune des cellules composant la molécule de calcul ne soit une cellule invalide puisque les flux s'en trouvent affectés, ce qui altère les résultats de la simulation numérique. Une telle situation peut se produire par exemple lorsque deux éléments solides sont très rapprochés. Ces trois procédures de vérification sont très simples à implémenter puisqu'elles n'impliquent que de simples tests.

RÉFÉRENCES

AMDAHL, G. M., "Validity of the single processor approach to achieving large scale computing capabilities", dans *AFIPS spring joint computer conference*, 1967.

BARNEY, B., "Message passing interface (mpi)", nov 2014, https ://computing.llnl.gov/tutorials/mpi/.

BARSZCZ, E., WEERATUNGA, S. K., et MEAKIN, R. L., "Dynamic overset grid communication on distributed memory parallel processors", *Journal of Computational Physics*, 1993.

BENEK, J. A., "Chimera : A grid-embedding technique", Rapp. tech., April 1986, aEDC-TR-85-64.

BLAZEK, J., *Computational Fluid Dynamics : Principles and Applications*. Elsevier, 2001.

CAGNONE, J., SERMEUS, K., NADARAJAH, S. K., et LAURENDEAU, E., "Implicit multigrid schemes for challenging aerodynamics simulations on block-structured grids", *Computers & Fluids*, vol. 44, pp. 314–327, 2011.

CHAN, W. M., "Overset grid technology development at nasa ames research center", *Computers & Fluids*, vol. 38, pp. 496–503, 2009.

CHAN, W. M., III, R. J. G., ROGERS, S. E., et BUNING, P. G., "Best practices in overset grid generation", dans *32nd AIAA Fluid Dynamics Conference*, 2002.

CHAN, W. M., KIM, N., et PANDYA, S. A., "Advances in domain connectivity for overset grids using the x-rays approach", dans *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.

CHANDAR, D. D., SITARAMAN, J., et MAVRIPLIS, D., "A gpu-based incompressible navier-stokes solver on moving overset grids", *International Journal of Computational Fluid Dynamics*, vol. 27, no. 6–7, pp. 268–282, 2013.

CHAWLA, K. et WEERTUNGA, S., "Overset grid applications on distributed memory mimd computers", dans *33rd AIAA Aerospace Sciences Meeting and Exhibit*, 1995.

CHESSHIRE, G. et HENSHAW, W., "Composite overlapping meshes for the solution of partial differential equations", *Journal of Computational Physics*, vol. 90, pp. 1–64, 1990.

CHEVANCE, R. J., “Serveurs multiprocesseurs et sgbd parallélisés”, août 2001, http ://www.techniques-ingenieur.fr/.

DELOZE, T., “Couplage fluide-solide appliqué à l’étude de mouvement d’une sphère libre dans un tube vertical”, Thèse de doctorat, Université de Strasbourg, 2011.

FUJII, K., “Unified zonal method based on the fortified solution algorithm”, *Journal of Computational Physics*, vol. 118, pp. 92–108, 1995.

H K VERSTEEG, W. M., *Computational Fluid Dynamics : The Finite Volume Method*. Pearson Education Limited, 2007.

HASANZADEH, K., LAURENDEAU, E., et PARASCHIVOIU, I., “Adaptative curvature control grid generation algorithms for complex glaze ice shapes rans simulations”, dans *53rd AIAA Aerospace Sciences Meeting*, 2015.

JAMESON, A. et BAKER, T. J., “Solution of the euler equations for complex configurations”, *AIAA Paper 83-1929*, 1983.

JAMESON, A., SCHMIDT, W., et TURKEL, E., “Numerical solutions of the euler equations by finite volume methods using runge-kutta time-stepping schemes”, *AIAA Paper 81-1259*, 1981.

LEVESQUE, A. T., PIGEON, A., DELOZE, T., et LAURENDEAU, E., “An overset grid 2d/infinite swept wing urans solver using recursive cartesian virtual grid method”, 2015.

LIAO, W., CAI, J., et TSAI, H. M., “A multigrid overset flow solver with implicit hole cutting method”, *Computational Methods Applied to Mechanical Engineering*, vol. 196, pp. 1701–1715, 2007.

MAVRIPLIS, D. J., “Grid resolution study of a drag prediction workshop configuration using the nsu3d unstructured mesh solver”, dans *23rd Applied Aerodynamics Conference, June 6-9 2005, Toronto, Canada*, 2005.

PIERCE, N. A. et GILES, M. B., “Preconditioned multigrid methods for compressible flow calculations on stretched meshes”, *Journal of Computational Physics*, vol. 136, pp. 425–445, 1997.

PIGEON, A., LEVESQUE, A.-T., et ÉRIC LAURENDEAU, “Two-dimensional navier-stokes flow solver developments at École polytechnique de montréal”, dans *22nd Annual Conference of the CFD Society of CanadaProgress in Aerospace Sciences*, 2014.

PREWITT, N. C., BELK, D. M., et SHYY, W., "Parallel computing of overset grids for aerodynamic problems with moving objects", *Progress in Aerospace Sciences*, vol. 36, pp. 117–172, 2000.

RAUBER, T. et RÜNGER, G., *Parallel Programming for Multicore and Cluster Systems*. Springer, 2012.

ROGERS, S. R., SUHS, N. E., et DIETZ, W. E., "Pegasus 5 : An automated preprocessor for overset-grid computational fluid dynamics", *AIAA Journal*, vol. 41, no. 6, pp. 1037–1045, June 2003.

ROGET, B. et SITARAMAN, J., "Robust and efficient overset grid assembly for partitioned unstructured meshes", *Journal of Computational Physics*, 2013.

ROGET, B. et SITARAMAN, J., "Wall distance search algorithm using voxelized marching spheres", *Journal of Computational Physics*, vol. 241, pp. 76–94, 2013.

RUMSEY, C., "Test/validation cases", oct 2014, http ://cfl3d.larc.nasa.gov/.

SCHWARZ, T., SPIERING, F., et KROLL, N., "Grid coupling by means of chimera interpolation techniques", dans *Second Symposium "Simulation of Wing and Nacelle Stall", June 22nd - 23rd, 2010, Braunschweig, Germany*, 2014.

SERMEUS, K., LAURENDEAU, E., et PARPIA, F., "Parallelization and performance optimization of bombardier multiblock structured navier-stokes solver on ibm eserver cluster 1600", dans *45th AIAA Aerospace Sciences Meeting and Exhibit, 8-11 January 2007, Reno, Nevada*, 2007.

SONI, K., CHANDAR, D. D., et SITARAMAN, J., "Development of an overset grid computational fluid dynamics solver on graphical processing units", *Computers & Fluids*, vol. 58, pp. 1–14, 2012.

SWANSON, R. et TURKEL, E., "On central difference and upwind schemes", *Journal of Computational Physics*, vol. 101, pp. 292–306, 1992.

VASSBERG, J. C. et JAMESON, A., "In pursuit of grid convergence for two-dimensional euler solutions", *Journal of Aircraft*, vol. 47, no. 4, 2010.

WANG, Z. J. et PARTHASARATHY, V., "A fully automated chimera methodology for multiple moving body problems", *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 919–938, 2000.

WISSINK, A. M. et MEAKIN, R. L., “On parallel implementations of dynamic overset grid methods”, dans *Proceedings of the 1997 ACM/IEEE conference on supercomputing*, 1997.

ZAGARIS, G., BODONY, D. J., BRANDYBERRY, M. D., CAMPBELL, M. T., SHAFFER, E., et FREUND, J. B., “A collision detection approach to chimera grid assembly for high fidelity simulations of turbofan noise”, dans *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

ANNEXE A

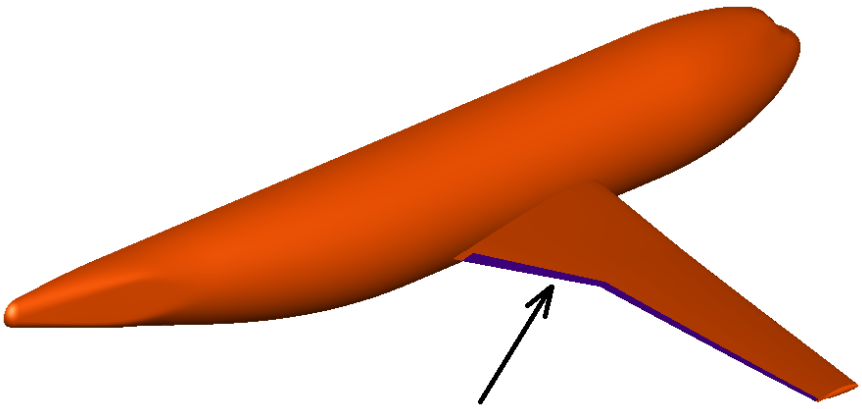


Figure A.1 Bloc (identifié en noir) définissant le bord de fuite de l'aile de la géométrie DLRF6

ANNEXE B

Une papier scientifique traitant des travaux présentés dans ce mémoire a été publié dans le Journal de l'AIAA (American Institue of Aeronautics and Astronautics). Cette publication traite d'un module d'assemblage chimère bidimensionnel intégré à NSCODE utilisant la technique de grilles vrituelles récursives. Le module d'assemblage est vérifié grâce à des fonctions analytiques et en comparant les résultats obtenus avec et sans la méthode chimère. Le document présente par la suite une optimisation de la position du volet sur le profil 30P30N dans un écoulement bidimensionnel à aile fléchée infinie. La référence à cette publication est présente à la section «Références» du présent document (Levesque *et al.*, 2015).