Titre: Title:	Development of a Parallel 3D Solid-Solid Overset RANS Method
Auteur: Author:	Michaël Gagnon
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Gagnon, M. (2022). Development of a Parallel 3D Solid-Solid Overset RANS Method [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10351/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10351/
Directeurs de recherche: Advisors:	Éric Laurendeau
Programme: Program:	Génie aérospatial

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Development of a Parallel 3D Solid-Solid Overset RANS Method

MICHAËL GAGNON
Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie aérospatial

Avril 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Development of a Parallel 3D Solid-Solid Overset RANS Method

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

Jean-Yves TRÉPANIER, président
Eric LAURENDEAU, membre et directeur de recherche
Bruno BLAIS, membre

ACKNOWLEDGEMENTS

I would like to thank my research supervisor, Dr. Éric Laurendeau. His guidance and confidence allowed me to grow and learn a lot throughout my thesis.

This project would not have been possible without the support of Bombardier Aerospace's Advanced group. The financial aspect of the project was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fond de Recherche Québécois de Nature et Technologie (FRQNT) and Bombardier Aerospace.

Many major breakthroughs during this project were supported by countless discussions and brainstorms with research associate Dr. Simon Bourgeault-Côté and postdoctoral associate Dr. Frédéric Plante. They provided vital feedback regarding the integration of this project in the overall laboratory structure which encompasses many other projects and features.

RÉSUMÉ

Ce présent mémoire présente les avancées récentes apportées à la technologie *overset* du laboratoire de recherche du professeur Éric Laurendeau à Polytechnique Montréal. L'approche *overset* est vastement utilisée dans l'industrie aérospatiale pour supporter la résolution de simulations hautement complexes en mécanique des fluides numérique. La méthode présentée dans ce mémoire assemble des maillages individuels contenant différentes géométries ou parties de géométries. On peut donc représenter une géométrie complexe par le biais de différents sous-maillages conçus pour les requis locaux de discrétisation. En référence à la mythologie grecque, la méthode est parfois appelée *chimère*.

La flexibilité que procure cette méthode minimise les coûts et risques associés à la génération de maillages. Un préprocesseur est typiquement utilisé pour assembler les différents maillages utilisés dans une simulation. Une connectivité est définie pour permettre aux maillages de communiquer entre eux à chaque itération dans le solveur Computational fluid dynamics (CFD). Ce préprocesseur permet de facilement ajouter, retirer ou modifier une aile horizontale lors de la simulation sur un avion complet par exemple.

Dans le cadre de cette maitrise, un préprocesseur *overset* tridimensionnel est développé. La technologie est intégrée au logiciel de mécanique des fluides numérique tridimensionnel CHApel Multi-Physics Simulation (CHAMPS) et vise à s'attaquer à des problèmes industriels d'actualité. L'approche chimère est typiquement composée d'une étape de coupure de trou qui permet l'identification des cellules hors du domaine de calcul et d'une étape de création de la connectivité entre les sous-maillages. Une première implémentation de la technologie dans le laboratoire de professeur Laurendeau permet une utilisation de l'approche avec des maillages *overset* bidimensionnel exempts de chevauchement impliquant plus d'une géométrie. Afin d'optimiser la robustesse et la flexibilité de l'approche sur des cas tridimensionnels plus complexes, une revue de littérature a été faite.

Le premier développement de ce présent projet permettant l'application de la méthode à des cas tridimensionnels complexes est une méthode de découpage beaucoup plus robuste. L'approche des rayons X permet le découpage précis autour des géométries convexes et concaves avec une précision contrôlable. La méthode génère, avec les paramètres et ressources appropriés, des résultats sans défaut sur des applications industrielles complexes.

L'application de la méthode sur des géométries complexes implique la plupart du temps des duplications de surface solide. Les maillages de type collant, aussi connu comme *collar grid* sont utilisés pour redéfinir des parties du maillage qui sont critiques et/ou mal définis. Ces maillages sont notamment utilisés pour redéfinir la région entourant les intersections entre géométries solides. Des progrès pour utiliser ce type de maillage sont implantés.

La création de la connectivité, qui est définie par un schéma d'interpolation, est supportée par une structure de type *oct-tree* qui permet une évolutivité du temps de calcul considérable sur plusieurs noeuds de calcul.

Finalement, l'étude de cas met en lumière les capacités de la technologie chimère développée. La méthode est testée sur des cas industriels d'actualité tels que ceux utilisés aux ateliers 4th Drag Prediction Workshop (DPW4) et 5th Drag Prediction Workshop (DPW5) présentés par la American Institute of Aeronautics and Astronautics (AIAA). Les résultats obtenus sont en accord avec ceux de la littérature ainsi que ceux obtenus avec l'approche traditionnelle utilisant un seul maillage. Cela montre une implémentation adéquate et robuste de la méthode compte tenu de la complexité des géométries étudiées. Bien que la génération de maillage soit habituellement facilitée par l'utilisation de l'approche chimère, générer des maillages tels que ceux utilisés à ces ateliers est hors du cadre du projet et donc des maillages disponibles en ligne seront utilisés durant sur projet.

Cette contrainte complique l'évaluation exacte des limites de la méthode telle qu'elle a été implantée dans CHAMPS, car les maillages utilisés ont été conçus pour différents schémas. La formulation *vertex-centred* pour représenter les variables plutôt que *cell-centred* comme dans CHAMPS est un exemple de différence qui peut critique. Avec l'approche *cell-centred* on a de disponible une couche de cellules de chevauchement de moins ce qui egendre parfois un manque. Ce manque de chevauchement prévient la définition adéquate de la connectivité pour des schémas de dissipation de deuxième ordre par exemple.

ABSTRACT

The present thesis presents the recent advances brought to the overset technology in the research laboratory of Professor Éric Laurendeau at Polytechnique Montréal. The overset approach is widely used in the aerospace industry to support flow resolution on complex simulations in CFD. The method presented in this thesis assembles individual grids containing different solid bodies or part of them. It is therefore possible to represent a complex geometry via different sub-grids generated to fit their local discretization requirements. In reference to the Greek mythology, the method is sometimes referred as *chimera*.

The flexibility brought by the method minimizes the risks and costs associated with the grid generation process. Typically, a preprocessor is used to build the grid to grid connectivity linking the different grids. This connectivity allows the communication of the flow variable's information at each iteration of the flow solver. The approach allows to easily add, remove or modify a horizontal stabilizer onto an aircraft configuration, for example.

The scope of this thesis is to develop a three-dimensional overset preprocessor within the CHAMPS framework, which is a three-dimensional CFD solver, and aim to tackle industrial applications. The method is usually composed of a hole cutting step that allows identification of the grid cells outside of the computational domain and a connectivity step that build the interpolation stencil linking the grids. A first implementation of the technology in Professor Eric Laurendeau's laboratory allows applications on bi-dimensional case exempt of intersection between solid bodies.

The first advancement to allow applications on more complex three-dimensional cases was to implement a more robust hole cutting method. The X-ray approach allows a precise cutting around both concave and convex bodies. With the appropriate inputs, the method generates results without default on complex industrial cases.

The application of the method on more complex geometries often implies surface duplication. The collar grid approach uses auxiliary grids to define critical or ill-defined regions of the domain. They are often defining the regions surrounding an intersection between multiple solid bodies. Considerations for the use of the collar grid are brought to the implementation in order to increase the robustness. First, the surface integration to compute the aerodynamic forces must consider the surface duplications. Then, an auxiliary method helps accelerate the hole cutting in these often fine regions.

The grid to grid connectivity, which is defined by an interpolation scheme, is supported by

an *oct-tree* type data structure that allows the optimization of the method on shared and distributed memory.

Finally, the case study puts to light the capabilities of the chimera technology as implemented during this thesis. The method is tested on industrial cases as presented at workshops hosts by AIAA such as the DPW4 and DPW5. The results obtained on industrial applications are in agreements with the literature as well as with the results obtained with the standard single grid approach. This shows a proper and robust implementation. Although the grid generation process is known to be simplified by the overset approach, generating grids of *industrial quality* is a challenge outside of this thesis scope. The overset technology developed within CHAMPS during this thesis was thus able to use the overset approach in industrial applications.

Using other laboratory's grids complicates the precise evaluation of the limits of the method developed because of the fact that they were initially designed for different solvers. The *vertex-centred* approach to represent the flow variables in contrast to the *cell-centred* approach used in CHAMPS is an example of difference that might be critical in some cases. This has an impact when using a second order dissipation scheme which requires more overlap for example.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

DPW4	4th Drag Prediction Workshop
DPW5	5th Drag Prediction Workshop
CFD	Computational fluid dynamics
NS	Navier-Stokes
RANS	Reynolds Averaged Navier-Stokes
FVM	Finite Volume Method
FDM	Finite Difference Method
FEM	Finite Element Method
CHAMPS	CHApel Multi-Physics Simulation
NSCODE	Navier-Stokes CODE
LU-SGS	Lower-Upper Symmetric Gauss–Seidel
GMRES	Generalized Minimal RESidual
RK	Runge-Kutta
GG	Green-Gauss
WLS	Weighted Least Squares
AABB	Axis-Aligned Bounding Box
CRM	Commom Research Model
MPI	Message Passing Interface
AIAA	American Institute of Aeronautics and Astronautics

CHAPTER 1 INTRODUCTION

Modern CFD technology helps engineers in design and analysis by providing numerical simulations to the equations that govern fluid motion. The technology reduces the need for real life tests (e.g., wind tunnel, flight tests) which are time and money expensive. This is of great interest in the aerospace industry to simulate flow around geometries and predict their aerodynamic properties. Chemical reactions, internal combustions and internal flows are other types of CFD applications. External compressible flows are studied in this thesis.

One approach CFD is to solve the Reynolds Averaged Navier-Stokes (RANS) equations [2, 4]. As presented by [5], various methods can be used to solve these equations, but all go through a physical and temporal computation which both require discretization. The Finite Element Method (FEM), the Finite Volume Method (FVM) and the Finite Difference Method (FDM) are popular approaches for spatial discretization. In the FVM, which is the approach adopted in this thesis, the properties are either defined at the centroid or the vertexes of the control volumes and assumed constant throughout these volumes. The spatial discretization is ensured by grid(s) generated to define the computational domain to study. The quality of those grids has a major impact on the CFD simulations ; it was shown that two different discretizations of the same domain could lead to substantially different results [6]. Producing quality grids is therefore a critical step for all CFD users as it can often represent around 50% of the overall time of the CFD process [7]. Traditionally, structured grid methods relying on arrays of quadrilateral cells (or hexahedral in 3D) are used to discretize the computational domain. Structured grids connectivity is expressed implicitly by the two- or three-dimensional arrays of grid connectivity, memory speaking. For complex geometries, however, unstructured grids based on explicit irregular connectivity provide greater flexibility. Details about the two approaches are listed in table 1.1.

From these approaches, others were developed. A hybrid structured/unstructured grid approach is sometimes used to take advantages of the two, for example. It consists of mixing zones with different grid types within the computational domain. Connectivity between the regions is then ensured with conformal matching frontiers. Ensuring conformal matching frontiers, however, can be a considerable design requirement. Being able to bypass this requirement of perfect frontiers matching would therefore make the grid generation process considerably easier and faster. The **overset** approach suggests decomposing complex components of the domain into simpler ones. Each sub-grid can then be tailored for their local requirements without being constrained by the exact definition of the frontiers. The only re-

Table 1.1: Some advantages of the structured and unstructured grid approach

Structured grids	**Unstructured grids**
Higher degree of quality and control.	Faster grid generation process.
Anisotropy is typically almost achieved implicitly because of grid lines and flow alignments with the contour of the geometry.	Better flexibility with complex geometries.
Less memory and time consuming due to implicit connectivity.	Less experience is required from the grid developer.
	Explicit connectivity is built when reading the grid.This is not as straightforward.

quirement of this approach is to have a satisfactory overlap between each grid. This provides a new considerable flexibility to the CFD users. Some of the most difficult CFD problems are now solved using this approach. The Integrated Space Shuttle Launch Vehicle (SSLV) in Figure 1.1, for example, was one of the first cases that drove the emergence of the overset technology [8] toward complex industrial applications.

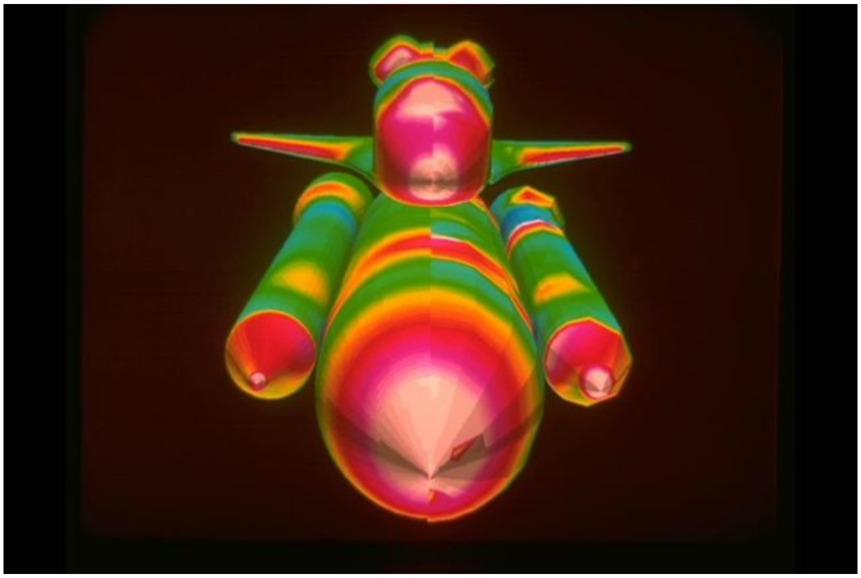


Figure 1.1 Space shuttle. Figure from [1].

1.1 Constraints

The overset technology has already been explored in the research laboratory of Professor Eric Laurendeau in large part because of the advantages presented in the previous section. Also for the sake of continuation, the overset approach is thus further developed in this thesis to test its capacities and limitations on three-dimensional industrial applications.

In regard to the current state of the technology, most of the developments were done in the Navier-Stokes CODE (NSCODE) framework. NSCODE is a two-dimensional structured flow simulation solver written in C and Python developed in Professor Éric Laurendeau's laboratory. The implementation of the overset technology within this framework showed good results on two-dimensional high-lift airfoil sections. Limitations were, however, stated in regard to multiple bodies either intersecting each other or with narrow gaps between them. Furthermore, three-dimensional capabilities are necessary to further test the robustness of the overset method on industrial cases. NSCODE is therefore currently limiting the development of the technology. This leads to the development of the technology within CHAMPS, a three-dimensional CFD software developed within Professor Éric Laurendeau's laboratory. The tool will have to be compatible with the unstructured topology of CHAMPS which is written in the Chapel HPC language. Furthermore, the technology will have to be compatible with the solver part of the CHAMPS framework.

1.1.1 More On CHAMPS

The CHAMPS software uses a finite volume approach with a second order cell-centred spatial discretization and unstructured grids to solve the RANS equations. The capabilities of this University software was verified and validated on different critical cases [9,10]. Different options are available in CHAMPS to solve a flow simulation. The Runge-Kutta (RK), Lower-Upper Symmetric Gauss–Seidel (LU-SGS) and Generalized Minimal RESidual (GMRES) solvers are available to compute the time iterations. The Green-Gauss (GG) and Weighted Least Squares (WLS) gradients are available to allow second order capabilities in the discretization of the convective and viscous fluxes.

Chapel Language

NASA's CFD Vision 2030 identify the development of highly scalable and robust algorithms in CFD applications as one of its key goals [11]. Traditionally Message Passing Interface (MPI) is used to handle the communication through distributed memory. C, C++ and Fortran are the most popular language used to optimize the performances of CFD software. The Chapel programming language, on which CHAMPS is based, offers a flexible approach to develop scalable CFD software without the need for OpenMP nor MPI libraries [12]. The performances of CHAMPS with an emphasis on the Chapel language verified on different applications and showed a good scaling on up to 256 compute nodes, totalling 1264 cores, as presented by Parenteau et al. [9]. The scalability of the implementation done in this thesis will therefore be of importance and verified.

1.2 The Overset RANS Approach

The approach was introduced by Benek et al. [13] in the 1980s for steady state problems on structured grids to *accommodate easily generated grids* [13]. The approach is also often referred as the *Chimera* approach after the Greek creature composed of different animal parts [14]. The concept is to decompose a complex geometry (or domain) into simpler components. This decomposition allows each sub-grids to be tailored for its local requirements to optimize the overall quality. Furthermore, the method allows to add, remove and modify parts of the grid assembly with ease and minimum user effort. A grid to grid connectivity must, however, be built so the flow solver can compute simulations on the global domain. The use of the method in unstructured solvers like it is the case in this thesis is not uncommon [15] [16]. In the author's experience, the unstructured approach has shown to simplify the grid generation.

Elements Of The Overset Problematic

The different grids composing an overset assembly initially have no information about each other nor about their relative position. Each grid solves its own part of the flow but have to communicate with the other grid in some way. This communication is usually ensured by interpolation in the overlapping regions, but other approaches such as domain reconstruction exists. Furthermore, control volumes outside of the computational domain must be located so they are not considered. Figure 1.2 presents a simple overset case for which the connectivity process can be fairly straightforward. As mentioned above, however, the modern aerospace applications deal with considerably more complex geometries. The overset grid in Figure 1.3, for example, involves intersections between the solid bodies which requires more precise methods to correctly, and in an acceptable lapse of time, interpret the overlaps and build the connectivity. Furthermore, interpolation in the boundary layer where gradients can be high requires robust considerations to properly capture the aerodynamic features like the shocks and recirculations.

A preprocessor can be used to build a grid to grid connectivity and store the associated information. Such preprocessors are usually composed of similar steps : (1) the control volume overlapping solid components of the domain are located and taken out of the computational domain and (2) a connectivity is built to allow information to be passed between the grids. More than one approach exists to build this connectivity, but they usually rely on either inter-

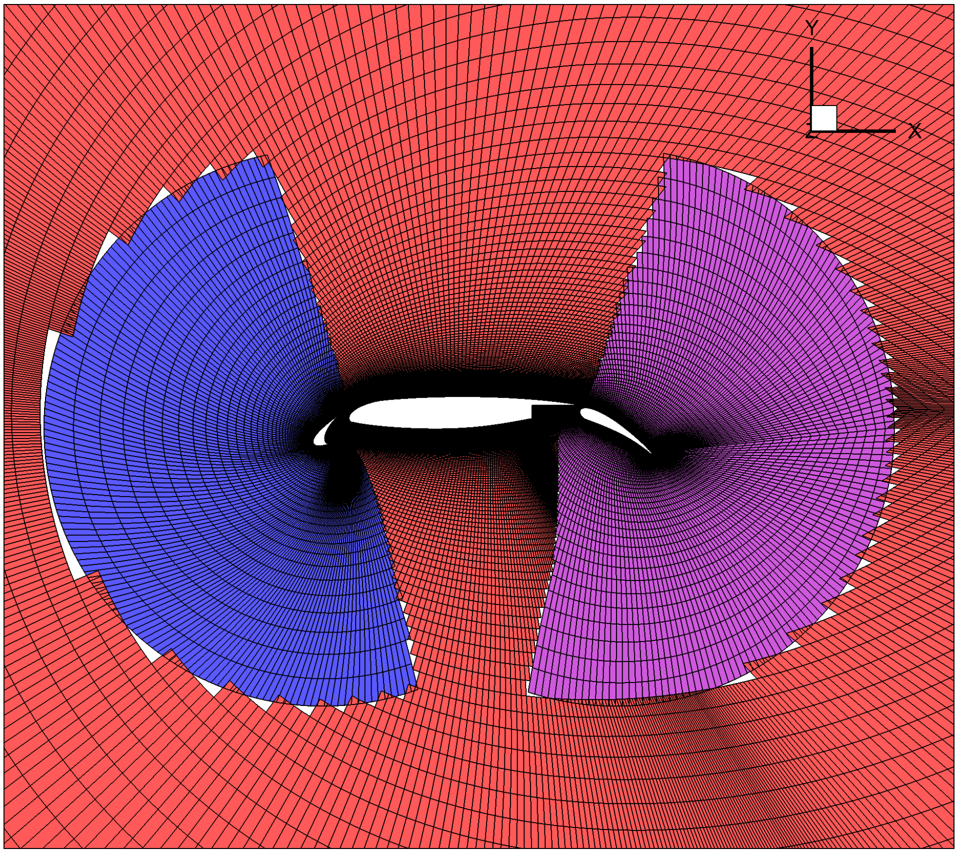


Figure 1.2 Grid of a multi-element wing section using the overset approach. The different colours represent the different overset grids. The overlapping regions have been blanked for better visualization.

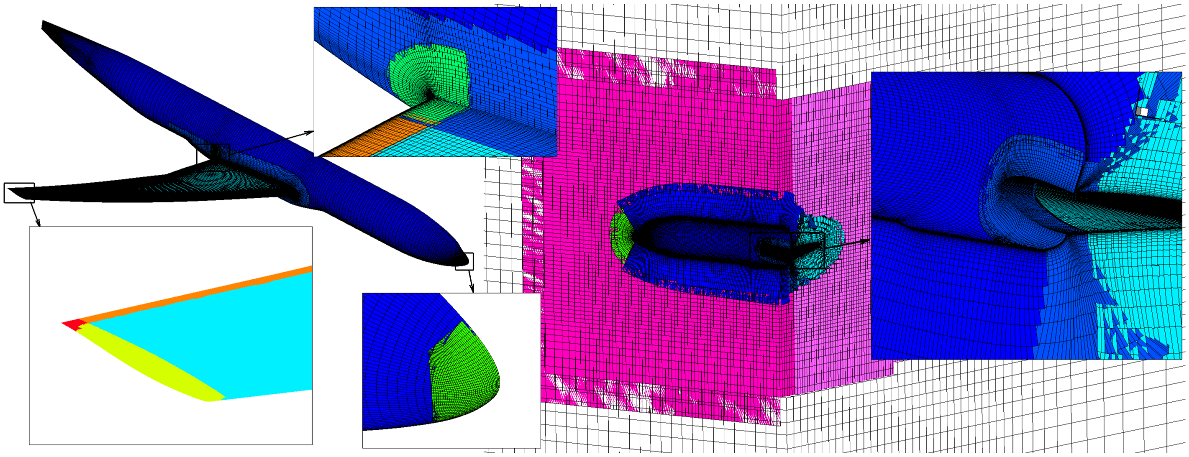


Figure 1.3 Grid of an aircraft using the overset approach. The different colours represent the different overset grids.

polation or grid reconstruction. The preprocessor interprets the grids in order to determine the following information :

- Which grid cells are computed.
- Which grid cells are part of the grid to grid connectivity.
- Which grid cells are to be taken out of the computational domain.

Definition

Here are some definitions that will be used throughout this thesis. Figure 1.4 illustrates the approach applied to a multi-element wing section composed of a slat, flap and main elements similarly to the application presented in Figure 1.2. It is evident that grid cells from Grid 1, for example, will be found within the flap, slat and main components of the airfoil. Such grid cells are referred as **invalid** and are excluded from the computational domain using a **hole cutting algorithm**. Through the rest of the domain, **computed** and **interpolated** grid cells are defined using different criteria that traditionally reflects the grid cell quality : this step is the **overset connectivity** step.

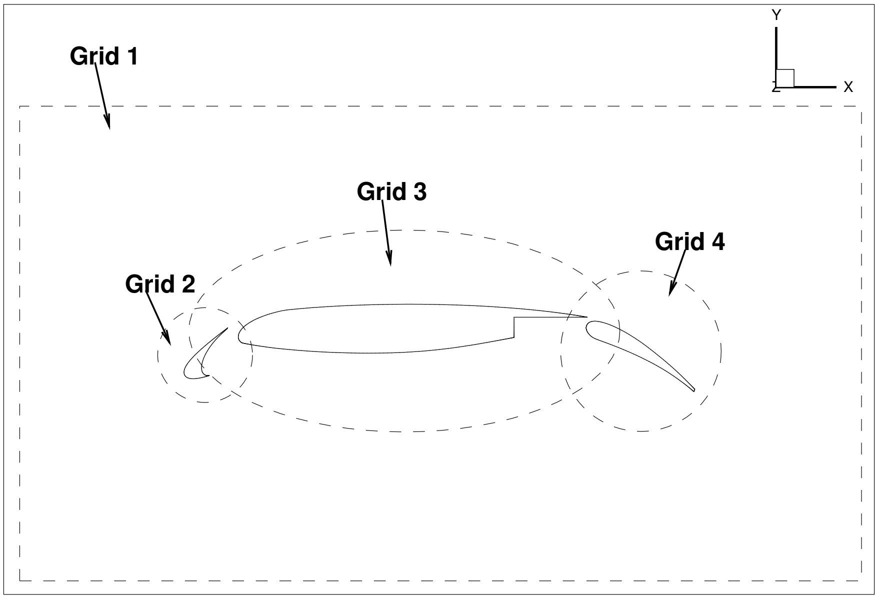


Figure 1.4 Illustration of a multi-element airfoil meshed using the overset approach.

1.3 Objectives

The development of an overset approach capable of treating solid intersections is a bottleneck in the creation of a robust method. When fully taking advantages of the overset approach, such intersections are inevitable when dealing with industrial applications. The main objective of this thesis is therefore to develop a preprocessor capable of treating the overset intersections in overset grids. From the overset technology already developed within the research's laboratory, this objective will be reached through the following steps :

- Implement a hole cutting method which is robust and efficient enough to generate valid holes around complex geometries.
- Investigate treatment of overset connectivity close to the wall (i.e., high gradient regions).
- Investigate post-treatment requirements of overset grids with solid intersections.

Events like the 4th and 5th Drag Prediction Workshops, to only name those, offers great examples of modern and relevant CFD applications in the industry. The overset grids studied at these workshops will serve as final applications of the overset technology developed in this thesis.

Since CHAMPS's creation in 2019, many students have used the software as a technological canvas for their research. Performances and robustness of the overall algorithm is therefore important in order to stay relevant for the CHAMPS software to continue using the overset approach in the future. The implementation will also have to consider the different schemes used and their stencils to correctly integrate the technology to the software.

1.4 Outline Of Thesis

A literature review is presented in the next section. The method is further presented as well as its challenges. The literature review revealed many different approaches that could provide robustness and precision to the technology. The preprocessor developed is then presented with canonic verifications. Finally, numerical results of industrial applications present the capabilities and limitations of the current implementation.

CHAPTER 2 LITERATURE REVIEW

The literature review presented in this thesis focuses on the developments of an overset method applied to complex geometries. Such overset grids generally have solid intersections which require additional treatments. The experience developed through the years have shown the overset approach as prone to automation [17–19] which will also be of consideration in this literature review.

2.1 History Of The Overset Method

Generating acceptable meshes around complex geometries has been a challenging bottleneck in the overall CFD process for a long time. Joseph Steger [20] defines two groups of methods that help mitigate these difficulties : (1) domain decomposition methods and (2) grid-adapting methods.

The overset approach is part of the first group : the domain decomposition. By decomposing the computational domain into simpler subdomains, the grid generation process becomes more straightforward. Each subdomain can be meshed separately with the only requirement on having a sufficient overlap between each of them to allow proper connectivity construction. While this *minimum overlap* varies from a solver to another, they are usually simple to ensure when generating the grids and thus the overlap can remain fairly arbitrary [17, 21]. The flexibility offered by the overset approach has shown to significantly improve the grid quality compared to the standard single grid approach [20]. Furthermore, it is possible, with this approach, to add, remove or move any subdomain by simply reprocessing the subdomains with different user inputs. In addition to save time, this allows freedom of motion between the subdomains. The standard single grid approach fails to satisfy all of the requirements for such simulations [21].

The idea behind the approach can be traced to Joseph Steger, a NASA Ames researcher, in the late 1970s and 1980s [20,21]. Since its first introduction, the overset approach has proved to be suited to provide accurate solutions when working with complex geometries [17]. The technology becomes more and more integrated in many well-known verified CFD software, like OVERFLOW[1], elsA[2] and ADflow[3] to only name those.

2.2 The Overset RANS Approach

Figure 2.1 presents the standard approach for building an overset connectivity. The invalid grid cells are located using a **hole cutting method** and the **grid to grid connectivity** is usually built by either reconstructing the grid or defining an interpolation scheme in the overlapping regions [5, 19–21, 23].

For more complex cases including **solid intersections**, further steps are needed to ensure robustness and proper connectivity. The boundary layer in Navier-Stokes flow computation is significantly disturbed with the standard implementation of the overset approach [3]. This is mainly caused by the different discretizations of the geometry by the different sub-grids in the solid overlapping regions. Furthermore, the duplication of the surface must be taken into consideration when computing the surface integration.

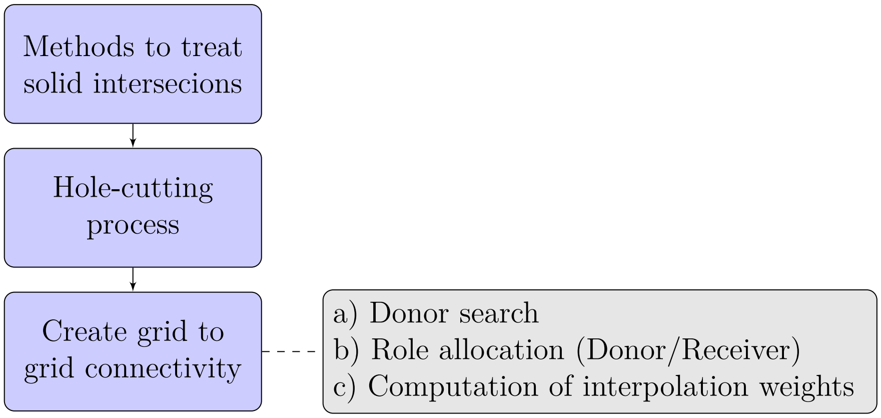

Methods to treat solid intersecions
Hole-cutting process
Create grid to grid connectivity
a) Donor search
b) Role allocation (Donor/Receiver)
c) Computation of interpolation weights

Figure 2.1 A standard overset approach

The modern overset technology is so that it is now possible to simulate transonic flow around a complete space shuttle using an overset grid with over 20 million grid points [24]. Robust-

[1] OVERFLOW 2.2 [7] is a widely used Reynolds-averaged Navier– Stokes (RANS) CFD code considered reliable and accurate for analyzing modern transport configurations. Originally developed by NASA with numerous contributions from academia and industry, it is a node-based solver specifically designed for structured overset grids where many options are available to the user, such as 2D/3D and steady/ unsteady simulations, thin layer versus full Navier–Stokes, multiple turbulence models, and a quadratic constitutive relation (QCR). [22]

[2] elsA is a cell-centred, finite volume, RANS code originally developed by ONERA and capable of using both point-matched and overset grids. [22]

[3] ADflow is a finite-volume RANS code maintained by the Multidisciplinary Design Optimization Laboratory (MDOlab) at the University of Michigan. A chimera overset grid method was implemented in ADflow. [22]

ness, automation, speed and memory consumption are the usual criteria for an ideal overset connectivity preprocessor [25] and will therefore guide this literature review.

2.3 Hole Cutting

The possibility to generate independent grids with arbitrary overlaps frequently causes situations as presented in Figure 2.2 where grid cells of a certain mesh A may be found within the solid body of a second mesh B. With a hole cutting methods, those grid cells are located, taken out of the computational domain and surrounded by interpolated grid cells. By doing so, in grid A, the invalid grid cells do not reach the stencils of the computed ones. Furthermore, these computed grid cells now have information coming from the solid component of grid B through the interpolated grid cells defined around the invalid ones. The thickness of the layer of interpolated grid cells depends on the stencils of the numerical methods used in the flow solver.

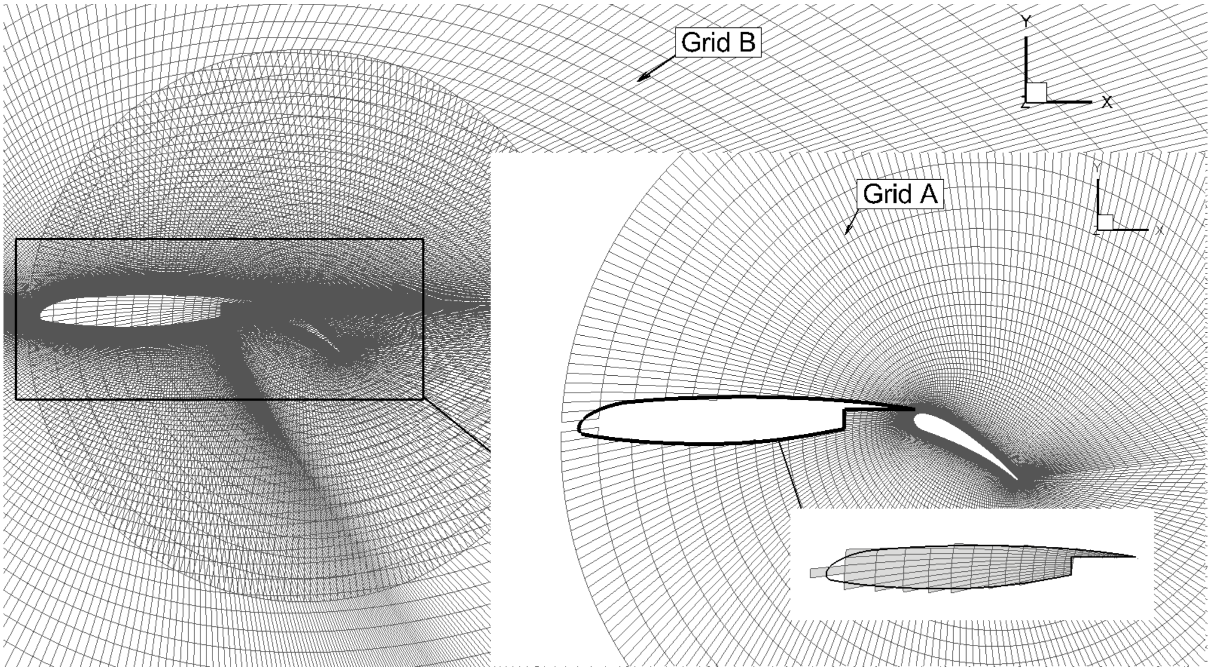


Figure 2.2 Illustration of the hole cutting process in grid A by the solid body in a grid B

This step is one of the most complex and critical processes of the overset approach [26]. Handling large and complex cases remains difficult. The memory consumption and scalability are the usual bottlenecks [26]. The hole cutting methods could be grouped in two categories : the implicit methods and the explicit methods.

Implicit Hole Cutting Method

The concept of the implicit hole cutting method could be summarized in two points :

1. The invalid grid cells inside the solid components do not need to be explicitly located or blanked from the solver to prevent contamination of the computational domain.
2. The list of blanked grid cells can be obtained in an alternative method without defining exactly the hole boundary [27].

The implicit approaches use a donor search method. For each pair of overlapping grid cells, a donor and a receiver is defined based on a user-defined criterion. The criterion traditionally reflects the quality of the grid cells like its volume or wall distance. The position of the communication interface will therefore be where overlapping grid cells have similar characteristics. The grid cells within a certain distance of the solid bodies and for which no donor was found, it is then supposed that they are within the solid body and are therefore invalid [28]. The implicit approach is thus a *cell-selecting* process that requires no hole cutting step. The implicit methods have shown to allow proper cutting of a minimum hole in many cases [26, 27, 29, 30]. The simplicity of computation and implementation of the approach are other important advantages.

This definition of the approach is, however, not really robust. Wrongly identified invalid grid cells can leave an insufficient overlap to create the connectivity. This is especially critical in cases with solid intersections or with minal overlap between the overset grids. In thoses cases, however, the implicit approach can still help position the interfaces whilst auxiliary explicit methods can be used to improved the overall robustness. More robust variations of the implicit approach are presented by Stuart E. Rogers et al. and X.H. Chang et al. [26, 30].

Optimum Overlap The position of the overset interface (i.e., the delimitation of computed and fringe grid cells) obtained with the implicit approach is automatically optimum relative to the criteria used in the donor search. The donor founds are automatically the best available [27]. An optimum hole cut process is defined by an inter-grid communication occurring between grid cells of compatible sizes. If not respected, solution accuracy and speed of convergence can both be significantly impacted. For example, a low-resolution region cannot properly capture the flow features that are formed in high resolution one [25, 30].

In sight of this, the explicit approach is chosen to help properly define the overset interface. However, one of the goal of this project is to develop an overset technology applicable to complex cases (e.g. complex geometries). Therefore, an explicit method is needed to ensure a proper hole cutting.

Explicit Hole Cutting Method

The scope of this thesis is to use overset grids with solid intersections. The needed precision of the hole cutting method with such cases increase ; every invalid grid points must be located, no more, no less. Search based methods (i.e., implicit methods) will sometimes falsely mark grid cells as invalid if the grids do not have enough overlap. In the explicit approach, the first step is to explicitly locate the grid cells within the solid bodies. Then, the donor grid cells for all the interpolated grid cells must be found. Explicit methods do not use auxiliary approximation and are therefore the most accurate methods. They can, however, be quite expensive in terms of time and memory [25, 31]. Many methods were developed [13, 31, 32] and optimized [33] through the years.

In Suggar++, a method called "direct cutter" finds the grid cells directly intersecting the surface grid to define the boundary holes [34]. The grid cells contained by the holes are then identified using a flooding algorithm. The standard flooding algorithms, however, will not work with open surfaces.

Benek, JA and Steger, JL and Dougherty, FC and Buning, PG presents a method developed in PEGASUS [20] based on wall distance and surface normals. A surface C, for example, is introduced into another subdomain. First, the normals are computed at each point defining the surface C. Then, a minimum hole is created around the body enclosing it whole. Using a scalar product of the normal vectors and the distance vector, the grid points can be positioned as either inside or outside of the solid body using the minimum hole as the first requirement of invalidity. In many cases, this approach creates an appropriate hole that allows building a proper connectivity. However, the normals can sometimes get difficult to compute automatically in the case of discontinued geometries for example [30]. Furthermore, cutting surfaces, viewed from the overlapping grid points, must be convex and closed [32] for the approach to be robust.

Object X-ray Method The X-ray hole cutting method is a robust approach to locate the grid cells within the solid bodies. The method borrows ideas from the hole-map and ray casting methods as presented by Robert Meakin [32]. With a proper data structure, the object X-ray method can be efficient enough for both static and unsteady applications with moving complex bodies which requires computing the hole cutting process each time there is a grid or body movement.

Figure 2.3 illustrates the original X-ray method [25, 35]. First, an Axis-Aligned Bounding Box (AABB) is created around the solid body. Then, rays are cast within the AABB and the pierce points on the solid surface are computed. For a certain grid cell, the entry and exit points of the surrounding rays are used to determine the inside-outside status. With a proper data structure, the method can be robust and efficient [32]. The oct-tree data is used in SUGGAR to quickly determine the grid cell close to the solid surface [25].

1. Create an axis-aligned bounding box around the solid body.
2. Cast rays in the axis-aligned bounding box.
3. Compute the pierce points of the rays in the solid body.
4. Test all points within the axis-aligned bounding box using the surrounding rays.

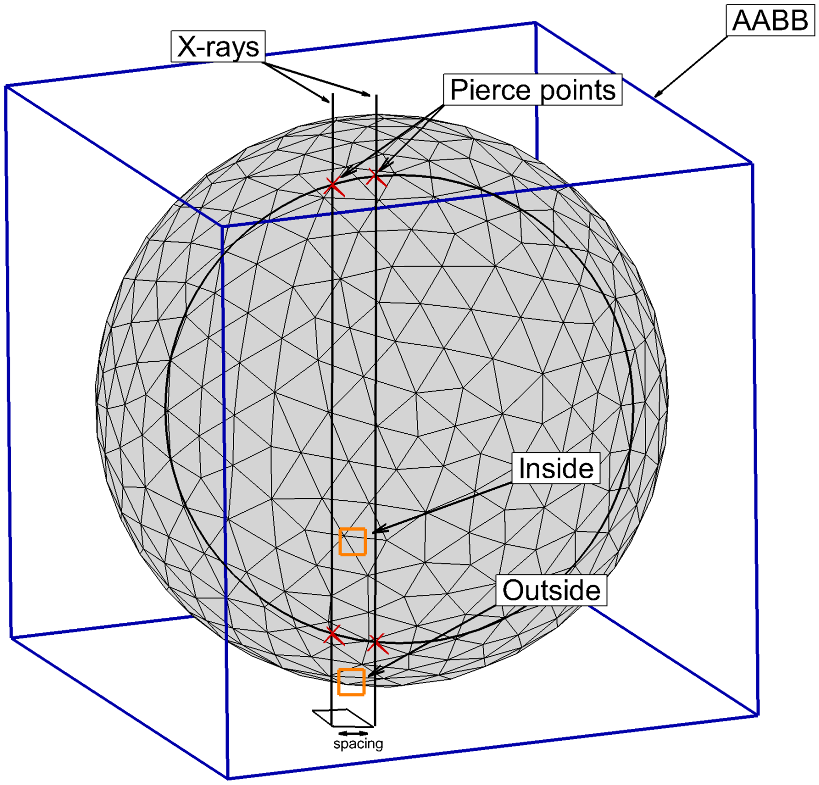


Figure 2.3 Illustration of the inside-outside test of the X-ray hole cutting method

The weaknesses of the object X-ray method were explored by Noah Kim and William M. Chan [25] with the purpose of developing a more automated and precise X-ray method. A constant offset distance from the minimum hole can be used, for example, to avoid interpolation close to the surface (i.e., in high flow gradient regions). However, a constant distance does not always generate the best overlap, especially in cases with near components or solid intersections.

William M. Chan et al. [35] suggests a dual wall distance based function to define a variable offset of the hole boundaries. An initial estimation of the interpolation boundaries using auxiliary methods, for example, can significantly decrease the density of X-ray needed to ensure robustness.

2.4 Grid To Grid Connectivity

In order for the subdomains to properly represent their part of the global domain, data must be communicated between the oversetting grids through the interpolated grid cells. There are two types of regions where grid cells are defined as interpolated [25] : (1) around the hole boundaries created in the hole cutting process and (2) at the overlapped field boundary conditions. This is illustrated in Figure 2.4 : here, for example, the domain grid gets information from the component through the hole boundary and the component grid gets information through the overset boundary condition.

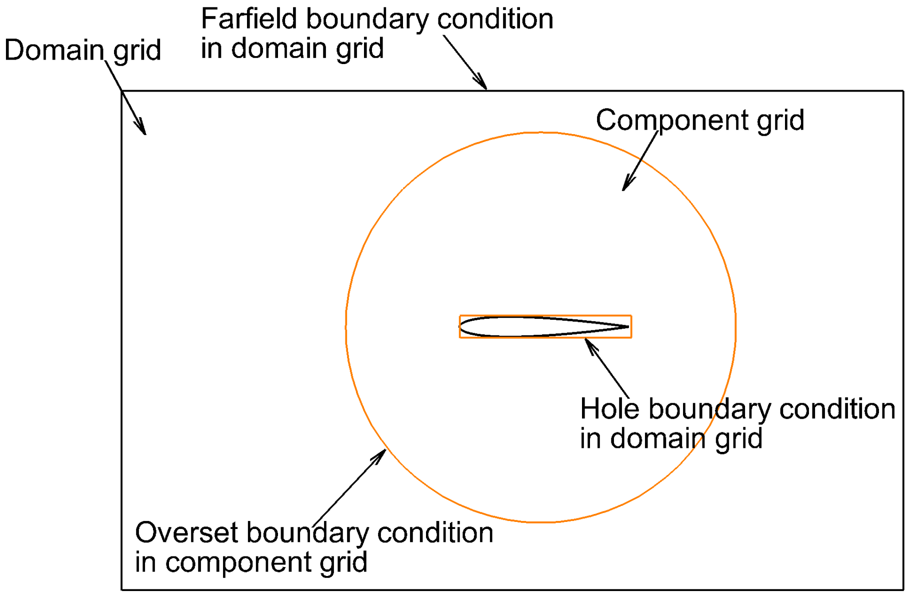


Figure 2.4 Illustration of the two types of boundary condition in overset method (in orange) : overset and hole boundaries

The interpolation methods can be grouper in two categories : non-conservative and conservative interpolation.

Non-conservative Interpolation

A downside of solution interpolation is that global conservation of fluxes is not guaranteed. It has been shown that a certain quantity of mass is generated along the fringe points boundaries, resulting in a pressure increase [36, 37]. This can even cause inconsistent boundary condition and thus failure of the simulation. One way of resolving this issue is to generate grids that perfectly match along the overset boundaries [37]. A conservative internal boundary treatment to transfer data between oversetting grids with complex geometries is, however, possible as presented by J.A. Wright and W. Shyy [38].

The actual local or global effect of this is difficult to measure, but many agree that the solutions are not degraded when the interpolation occurs in regions where sharp gradient, shock or discontinuities are not present [29]. Many non-conservative methods exist, but linear interpolation functions are the most popular due to their simplicity [29]. The Taylor approximation, the inverse distance and trilinear methods are other examples of popular non-conservative interpolation methods [20, 26]. Work on error analysis showed, however, that an interpolation of higher order of the underlying discretization may be necessary to preserve the accuracy in some region [29].

Conservative Interpolation

Cases in which shock waves or high gradients occur close to the interpolation regions, conservation is usually needed [36]. The conservative methods, however, are known to be substantially more complex than non-conservative. Furthermore, in three dimensions, it can sometimes be almost impossible to apply. An approach called the zipper grid [36] is sometimes used to create new grid cells to replace the overlaps as illustrated in Figure 2.5. The zipper grid approach first blank the grid points which are part of the overlap and then close the gap with new grid cell elements.

2.5 Solid Intersections

For viscous flow around intersecting pieces of the geometry in overset grids, it is difficult to generate proper grid to grid connectivity. The final resolution of the intersection region is therefore negatively impacted. Tests have shown that in those cases, resolution of the intersection region and geometric surface with respect to both grids had to be improved [39]. Different approaches can help solve this problem and they all share the same general idea : in order to properly define solid intersections, one of the meshes needs to have knowledge of the intersecting region of the geometry. Some approach extends the grid's surface to cover a

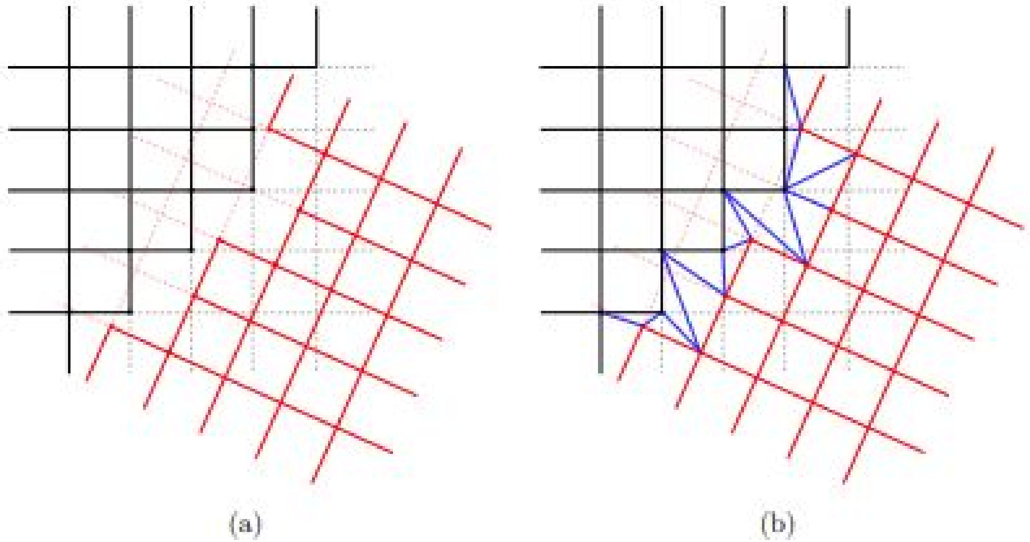


Figure 2.5 Zipper grid approach. (a) Overset grids with dominant grid cells only are not blanked. (b) A triangulation algorithm adds new grid cells to cover the region (from [2])

part of the other component [4, 40], for example.

Collar Grid Approach

The collar grid method is another approach used to improve the resolution of any section of a mesh, especially the regions around solid intersections. Collar grids are body fitted to both intersecting bodies while the intersecting grids are only body-fitted to their own piece of the body [41]. As illustrated in Figure 2.6, a hole is first created around the solid intersection, then a new grid (i.e., the collar grid) is overset to properly defines the intersection and allows proper communication between the surrounding grids.

Mismatch In Surface Discetizations The collar grids generally have a different surface discretization than the underlying overlapped grids. This mismatch has two direct effects. First, the donor-receiver pairs of grid cells have different values of wall distance (refer to figures 2.7(c-d)) which has considerable effect in the turbulence model, especially when the source term is dependent on the wall distance value. Furthermore, orphan grid points as illustrated in Figure 2.7(a-b) can be observed because no valid donor is found. A simple correction which was proposed by Schwarz [3] would be to virtually modify the interpolated cells coordinates in the interpolation stencil. He was indeed able to correct a significant

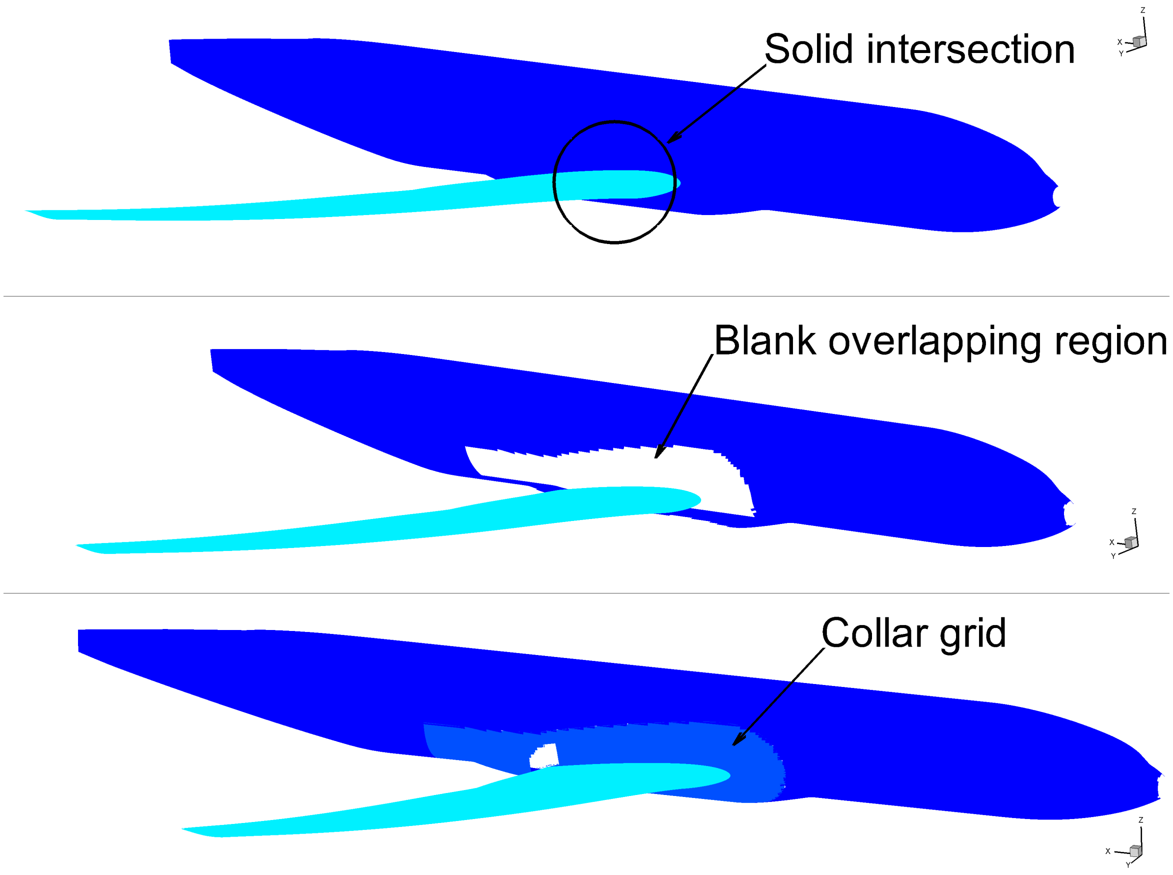
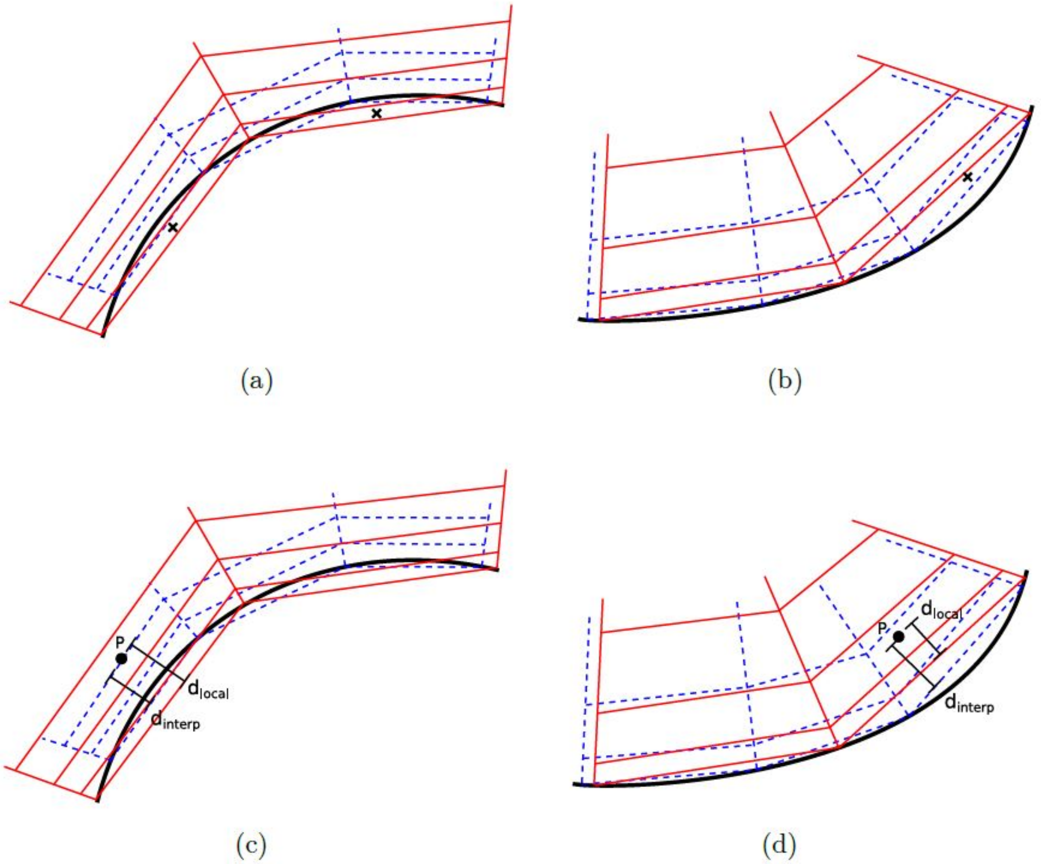


Figure 2.6 Illustration of the collar grid approach

error in the skin friction computation using his correction. The correction is based on the projection of the overlapping surfaces onto each other. Using the projection vectors, and before computing the donor search, the grid points are shifted to match their potential donor's wall distance. Because the problem occurs close to the wall, the extent of the application of the method decrease far from the wall.

Surface Integration Aerodynamics properties are often used to verify and validate new implementations. They can also be used to monitor numerical convergence of a simulation. Thus, the duplication of the surface area when using collar grids must be considered to avoid counting the same area twice in the surface integration. The zipper grid method can also be used for force and moment computation [42] : the overlapping surface elements are blanked and then a hybrid composite surface is generated using quadrilaterals and triangles to fill the hole and connect the grids. FOMOCO is a software package developed at NASA for computing flow coefficients on a collection of overset surfaces [43]. FOMOCO is a two-step process. First, the MIXSUR code blank the overlapping areal regions and close the



(a)

(b)

(c)

(d)

Figure 2.7 Surface discretization of overlapping meshes for a convex geometry (a-c) and a concave geometry (b-d), highlighting cells located outside of the computational domain (a-b) and the improper interpolation donors (c-d).

holes with triangulation then OVERINT compute the aerodynamic forces and moments over the new unique surface [44]. This method was, however, developed with the vertex-based approach in mind. Furthermore, the triangulation process is often not robust. Wigton [45] presents a solution to the later problem which avoid triangulation altogether. The algorithm, which he called Polymixsur, applies the polygon Boolean difference operator to weight each surface element. A weight between 0 and 1 is assigned to each areal element based on its relative hierarchy and its overlap by other surface elements of higher hierarchy. These weights then multiply the contribution of each surface element in the surface integration step. This algorithm was shown to be more robust and more general than the original one of FOMOCO. The bottleneck for the robustness of this approach depends on the underlying polygon Boolean method used. Most of the method to compute the weights increase in precision when refining the grids [44]. The algorithm steps are as follows [44] :

1. Find pairs of overlapping surface elements. Collision of their axis-aligned bounding box and dot product of the normals can be used to determine the validity of the pairs [44].
2. Based on the hierarchy of each grid, the relative dominance is defined.
3. Compute the polygon Boolean operations. This is the most important step of this weighting algorithm. The polygon Boolean difference removes portions of the lower-ranking overlapped surface elements.
4. The final weighting coefficient is computed as the ratio of the polygon region and the original region.

Francisco Martínez, Carlos Ogayar, Juan R. Jiménez, Antonio J. Rueda [46] presents a simple algorithm for Boolean operations on polygons. First, the edges of the polygons are subdivided at their intersection points. Using the internal parts of the edges, a new polygon is created within the overlapped surface element.

2.6 Acceleration Techniques

Speed and automation of the overset approach are two criteria of interest in this literature review. Modern CFD applications are often composed of hundreds of millions of grid cells. The implemented methods must be able to compute on multiple computational nodes in parallel and must be scalable to some extent. This is achieved through coding techniques applied in every part of the implementation. Different acceleration methods such as spatial data structure can also help to speed up the processes. A very popular technic used in CFD software is the oct-tree data structure.

2.6.1 Oct-tree Data Structure

The oct-tree data structure is an approximation of a dimensional object by a set of cubes divided recursively [47]. The donor search step and the hole cutting step both require many searches through the domain. To speed up the search process, the cubes are hierarchically organized as illustrated in Figure 2.8 to reach the information faster. The oct-tree is initialized by creating a bounding box around the grids ; this is the first level node. Then, recursively, sub-nodes are added by dividing the previous one in eight new ones until certain criteria are met [48]. The depth of the structure is usually controlled by user inputs like a maximum depth or number of grid cells within the deeper level nodes.

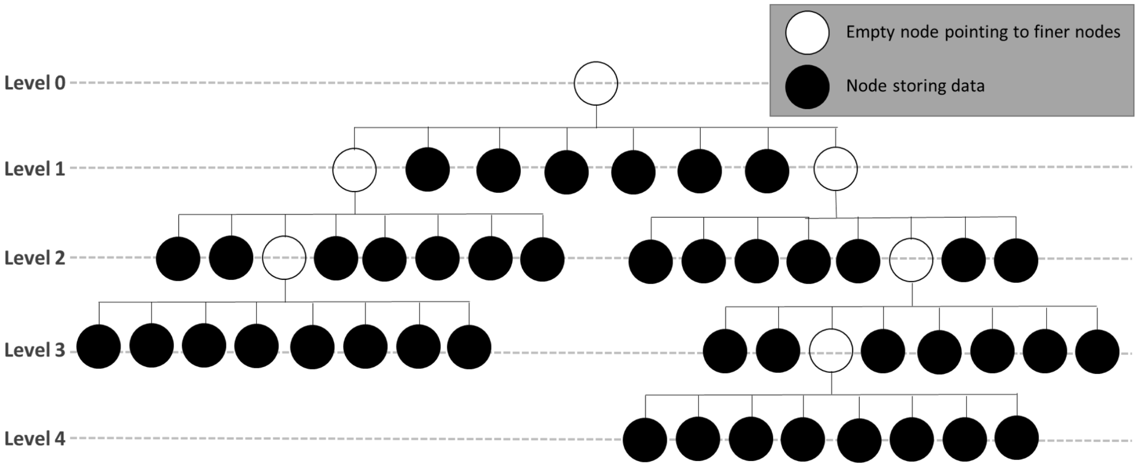


Figure 2.8 Illustration of the oct-tree data structure

The natural approach for computing the donor search becomes very expensive because of the large quantity of grid points and possible pairs of overlapping grid cells that are possible. The donor search for a specific grid cell consists of searching another overlapping grid cell that encloses its centroid. With the oct-tree data structure, the actual quantity of *containment tests* to do can substantially decrease. The oct-tree stores the grid points information in a tree manner based on their coordinates. When searching a potential donor for a grid cell, the oct-tree is passed starting from the top level following the nodes containing the grid cell. The deeper node found containing the grid cell then allows to considerably narrow down the list of grid cells to test [48]. The acceleration provided by the structure is so that the procedure to get the narrowed list of grid cells is faster than simply do the *containment tests* on every grid cells of the domain.

CHAPTER 3 METHODOLOGY

This section presents the methods used to implement the overset technology in CHAMPS. Creating the overset connectivity is essential in order to run flow simulation. This is usually done with a preprocessor and written in the grid files so it is only done once. The main purpose of an overset preprocessor is thus to analyze how the different grids overlap with each other in order to create a connectivity. Figure 3.1 is a schematization of the overall overset process as implemented in CHAMPS. The two main steps of the method are the hole cutting and donor search steps. They are the most time and memory consuming steps of the preprocessor. First, the definition of the donor search method in CHAMPS is presented. Following, the X-ray hole cutting method as developed in this thesis to tackle the challenge introduced by complex geometries is presented. Then, the definition of a more robust and parallel X-ray method, which was a considerable challenge, is presented. The interpolation stencil used in the overset method is then defined in order to respect the other numerical stencils already in the CHAMPS software.

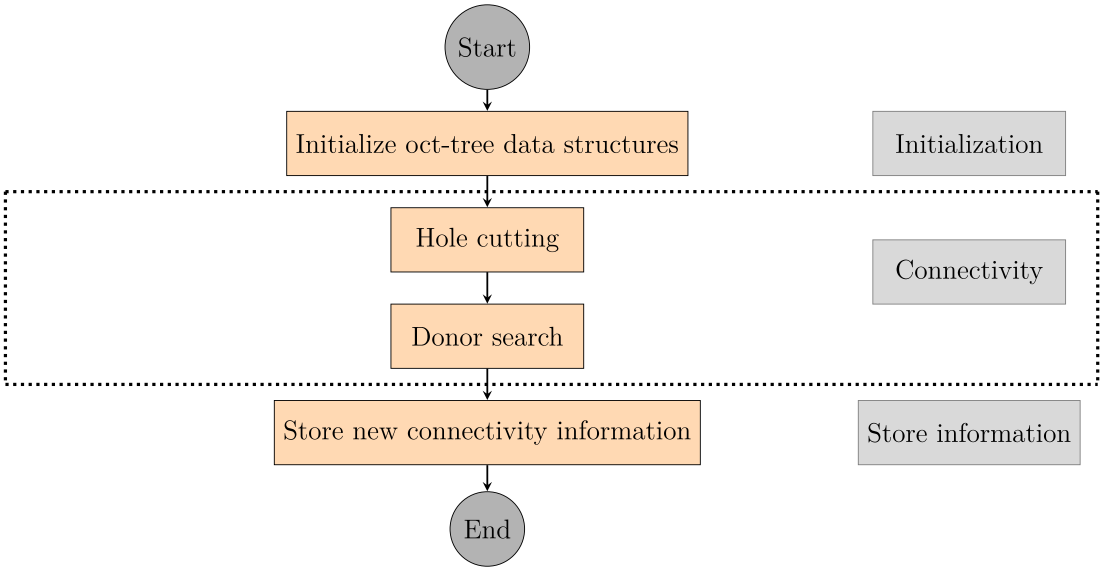


Figure 3.1 Illustration of the overall process of the overset preprocessor in CHAMPS

The oct-tree data structure implemented is also presented. The structure is built with the sole purpose of accelerating the assembly part by allowing a faster nearest neighbour search. Furthermore, the structure is implemented to manage and minimize the communication between computer nodes in the nearest neighbour search step. The considerations implemented to allow the use of collar grids on solid surfaces are then presented. The scalability of the different steps of the preprocessor in terms of computational time when using additional CPUs is also verified and discussed.

3.1 Grid To Grid Connectivity

To build the inter-grid connectivity, as presented in Figure 3.2, the hole cutting algorithm is first executed. The X-ray hole cutting method was implemented for its robustness and flexibility to work on complex geometries. Once the invalid grid cells are found, the implicit method is used to identify the donor and receiver pair of grid cells. As it will be presented, the implicit approach on its own fails to generate accurate results in some cases.

Hole cutting
Donor search
Connectivity

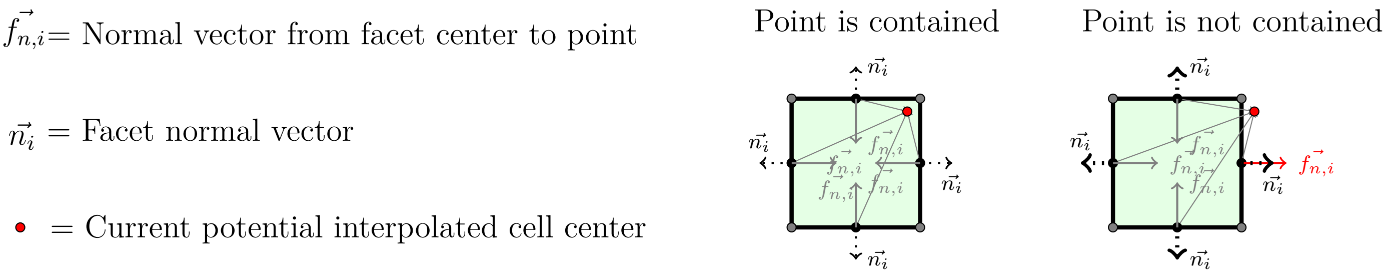
Figure 3.2 Creating the grid to grid connectivity

3.1.1 Implicit Approach

The first approach implemented in CHAMPS is the implicit hole cutting approach. As explained in the literature review section, this approach is based on the idea that computing the donor search algorithm for every grid cells of the domain will create the optimum overset interfaces and implicitly isolate the grid cells to hole cut out of the computational domain. In CHAMPS, an axis-aligned bounding boxes around the solid bodies are used to test the grid cells that are not overlaped. In other words, every grid cells inside the bounding box for which no valid donor from the component grid was found are considered invalid.

The donor algorithm consist of verifying which grid cells are overlapped by an other one. A grid cell A is considered overlapped by a second grid cell B if its centroid lies inside the grid cell B as it is illustrated in Figure 3.3. The inside-outside status is determined by the equation 3.1 : the centroid is *inside* if equation 3.1 is verified at every face of the overlapping grid cell. CHAMPS considere the standard CGNS linear grid elements [49] which are within the limits of application of this test.

$$\begin{aligned} &\text{if} \quad \vec{f_i} \cdot \vec{n_i} > 0 \quad \text{, cell center is outside.} \\ &\text{else} \quad \vec{f_i} \cdot \vec{n_i} \leq 0 \quad \text{, cell center is inside.} \end{aligned} \tag{3.1}$$



$\vec{f_{n,i}}$= Normal vector from facet center to point
$\vec{n_i}$ = Facet normal vector
● = Current potential interpolated cell center
Point is contained
$\vec{n_i}$
$\vec{f_{n,i}}$
Point is not contained
$\vec{f_{n,i}}$

Figure 3.3

In some cases, however, the implicit approach is not applicable because of a lack of overlap. Such problems were encountered in the application of the method on both two- and three-dimensional cases.

Figures 3.4 is an example of such situation. As it is shown on top of figure 3.4, the extent of the domain of the main component grid is fairly small and therefore the axis-aligned bounding box around the solid component overlaps regions outside the extent of the grid domain. As it is pointed by the red circles, this wrongly identifies invalid grid cells and thus leaves holes in the final grid. Figures 3.5 is three-dimensional example of this problem : the axis-aligned bounding box around the wing component reaches regions outside of the component grid.

Grid To Grid Communication

To communicate the information from the donor grid cells to the associated receiver grid cells, the Taylor approximation as presented in equation 3.2 is used.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x - a) \tag{3.2}$$

CHAMPS uses a cell-centred discretization of the flow variable to represent the flow. The concept of this scheme is to compute fluxes through the faces of each control volume to update the solution. The fluxes are computed on each face based on the value of the two control volume sharing the face. In the overset grids, hole cut and interpolated grid cells

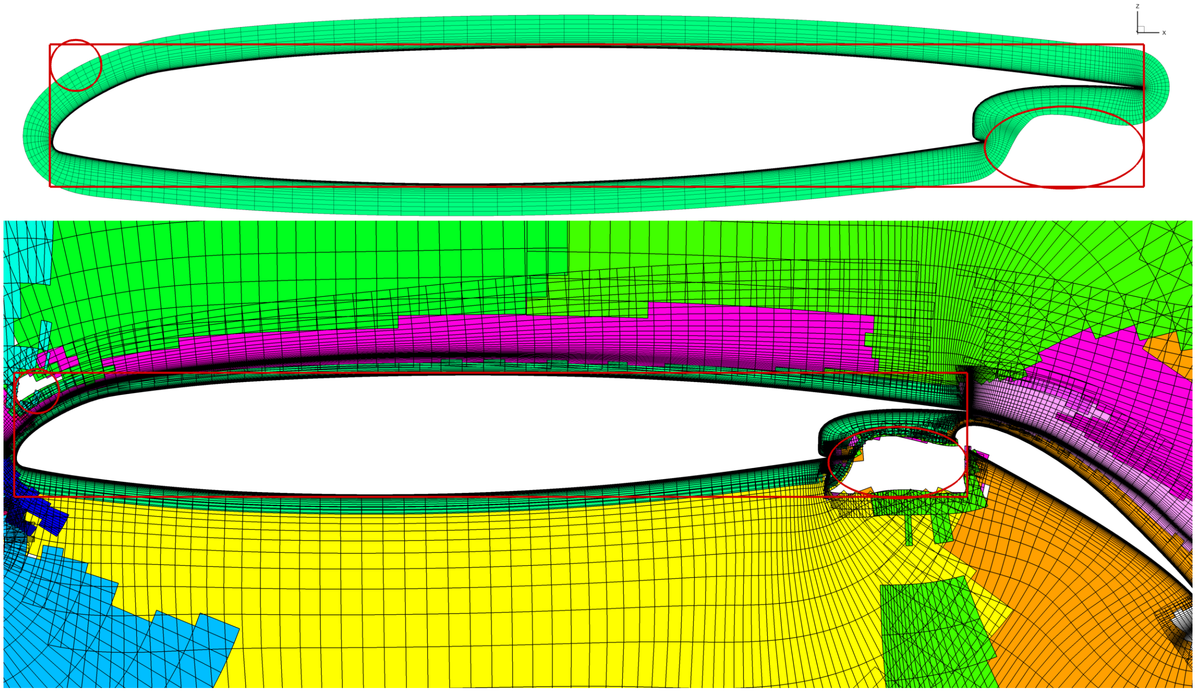


Figure 3.4 Illustration of the limitation of the implicit hole cutting approach on a two-dimensional case. Only computed grid cells are shown.

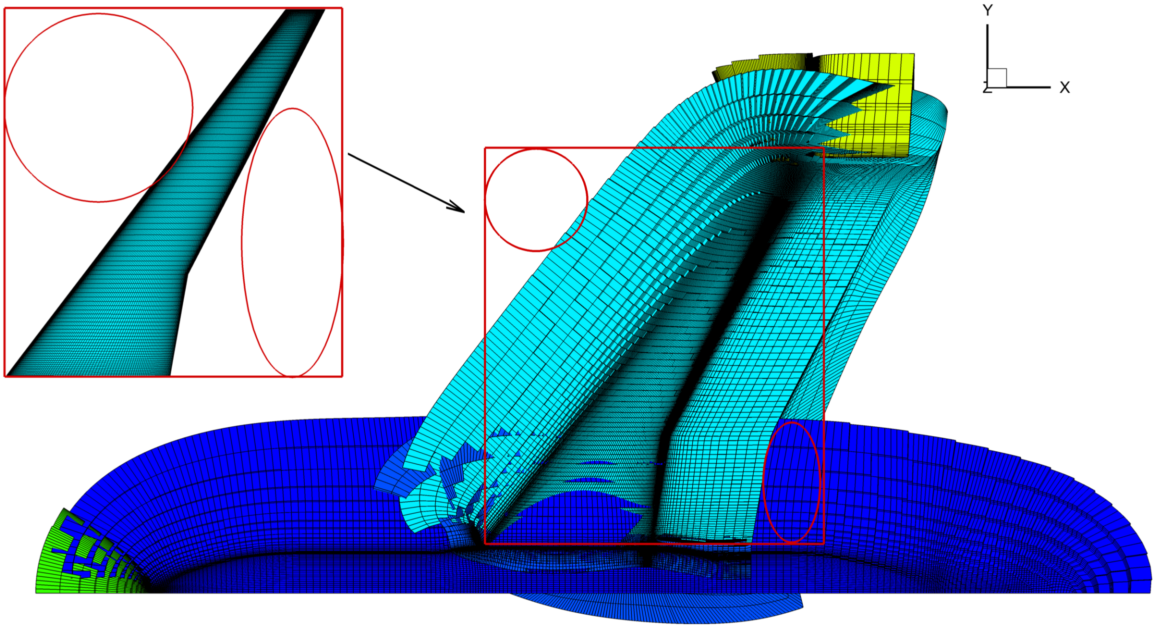


Figure 3.5 Illustration of the limitation of the implicit hole cutting approach on a three-dimensional case. Only computed grid cells are shown.

do not need to be updated. However, to correctly update the computed grid cells, the flux through all its faces must be computed which means that the values in the immediate neighbouring control volumes must also be accurate as shown in Figure 3.6. The grid cells with a crossed centroid represent the neighbours used in the stencil. Blue control volumes represent the computed neighbours. Yellow control volumes represent the interpolated neighbours. P_i and P_c indices indicate that proper primitive flow variables are available in the control volume from interpolation or computation respectively. G and L indices indicate that proper gradients and limiters are available in the control volume respectively. For a first-order flow simulation in CHAMPS, only one layer of interpolated grid cells is necessary. In second order schemes, however, the control volume sharing a face needs appropriate gradients to compute the fluxes. The computation of the gradient also implies the immediate neighbouring grid cells information which means that an additional layer is necessary to properly compute the gradients in the interpolated grid cells sharing faces with one or more computed ones. Figure 3.7 presents the stencil of a second order scheme : the control volume with a black dot in its centroid needs all its neighbours to have both values of primitives and gradients available and correct. The first left neighbour of the control volume has its primitive values interpolated and are therefore available. For the gradients to be available, they need to be reconstructed, which uses the primitive values of the next neighbours : that is why an additional layer of interpolated grid cells is added. When using limiters to limit extremum values of gradients, an additional layer is again necessary. The stencil to compute the limiter uses the neighbours gradients : Figure 3.8 presents the extent of the stencil when using limiters. When generating our own grids, this requirement is not really restrictive. Generating grids for complex cases such as a complete aircraft can, however, be a considerably challenging on its own and is not within the scope of this thesis. That being, when using grids generated by others, lack of overlap between the grids can prevent the definition of three layers of interpolated grid cells as described here. Grids designed for a vertex-centred CFD solver automatically have an additional layer of control volume compared to the cell-centred approach used in CHAMPS, for example.

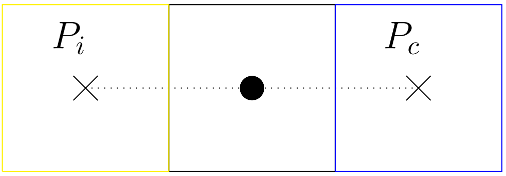


Figure 3.6 Extent of the stencils, in one dimension, used in a first order scheme simulation to update the solution in the grid cell marked with a black dotted centroid

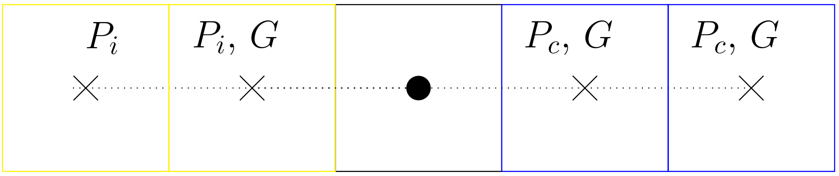


Figure 3.7 Extent of the stencils, in one dimension, used in a second order scheme simulation to update the solution in the grid cell marked with a dotted centroid

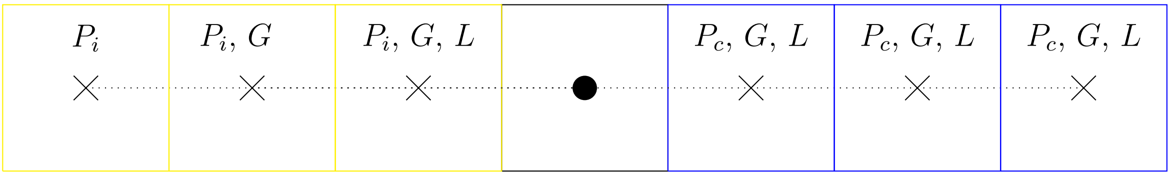


Figure 3.8 Extent of the stencils, in one dimension, used in a second order scheme simulation using a limiter to update the solution in the grid cell marked with a dotted centroid

3.1.2 X-ray Hole Cutting Method

As already mentionned, a mix of an implicit and explicit hole cutting method is implemented in CHAMPS. The implicit approach optimizes the position of the overset interfaces and the explicit method ensures robustness with complex bodies.

The X-ray method is available in CHAMPS to strengthen the hole cutting step. As explained in a previous section, the method cast rays onto the solid bodies to create a Cartesian mapping of the surfaces and then position the overlapping grid cells as inside or outside. Figure 3.9 illustrates the overall process of the method.

1. **Create Axis-Aligned Boxes :** The AABB (Axis-Aligned Bounding Box) of a component simply correspond to the box formed by the extremum's coordinates of this component. Using AABB allows speed up the method as grid points outside of those boxes necessary does not intersect the components. These AABB are computed around the simpler components of the overset grid.

2. **Cast X-rays In Axis-Aligned Boxes :** Based on a user-defined spacing, rays are cast in the AABB. Based on the user inputs, rays can be cast, in CHAMPS, in X, Y and Z direction. This can be useful to quickly adapt the method to different cases which contains symmetry plane or open surfaces. The spacing can be separately defined for each orientation which further increase the flexibility.

3. **Create Cartesian Mapping With The Pierce Points :** The intersection points

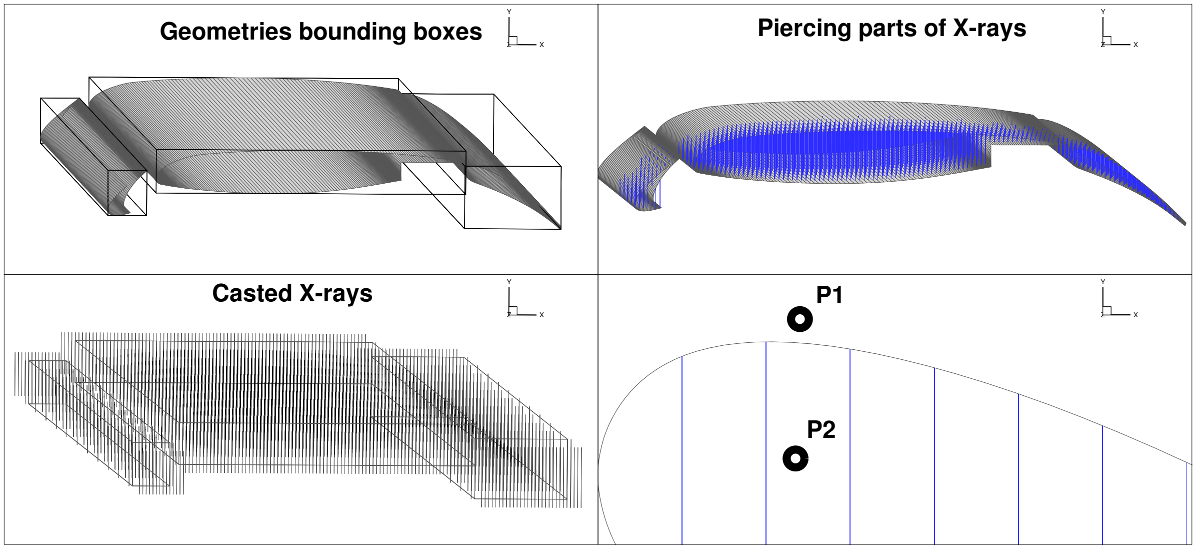


Figure 3.9 Process of the X-ray hole cutting method

of the rays in the solid components are then computed. To do so, a collision detection algorithm is used. CHAMPS uses unstructured grids and can therefore have surface elements of different types. To standardize the process, the surface elements are divided in triangular elements using the edges and the centroid. The algorithm to verify the intersection of a line segment and a triangular face element as illustrated in Figure 3.10 is defined by algorithm 3.1 : this algorithm verifies the intersection for a ray oriented parallel to the x-axis and was inspired by the work presented by Ericson [50].

4. **Detection Of The Grid Cells To Hole Cut :** Once all the collision points are found, it is possible to know each entry and exit points. If a certain overlapping grid cell is found to be within the AABB of a solid component, the four surrounding X-rays need to be located. This can be done fairly easily and quickly with the Cartesian data structure of the object X-ray method. For a distribution of rays oriented parallel to the X-axis as illustrated in Figure 3.12, the code snippet presented in algorithm 3.2 shows how to find the four surrounding rays : rays are uniformly ditributed and numbered as illustrated in figure 3.11. Once this is done, the inside-outside status is determined with the algorithm shown in equation 3.3.

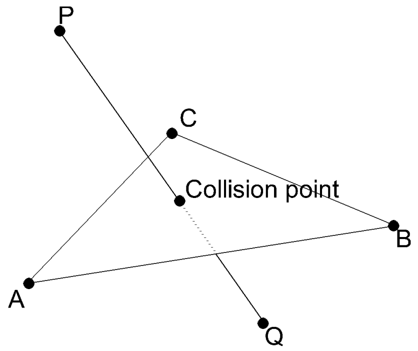


Figure 3.10 Intersection of line segments PQ against the triangular surface face ABC

Algorithm 3.1 Algorithm to verify the collision of a line segment $\vec{PQ}$ and a triangular surface ABC. This is illustrated in Figure 3.10

```
// Compute scalar and vectorial products
vector M =  vectorial_product(PQ, PC)
float u =   scalar_product(PB, M)
float v = - scalar_product(PA, M)
float w =   scalar_product(vectorial_product(PQ, PB), PA)

// Test intersection of PQ against triangle ABC
if      (checkSign(u) != checkSign(v))  then OUTSIDE
else if (checkSign(u) != checkSign(w)) then OUTSIDE
else if (u == 0.0 && v == 0.0 && w == 0.0) then OUTSIDE
else                                            INSIDE

// Barycentric coordinates
barycentric_u = (u * denom) / (u + v + w)
barycentric_v = (v * denom) / (u + v + w)
barycentric_w = (w * denom) / (u + v + w)

COLLISION = u*a + v*b + w*c // Collision point
```

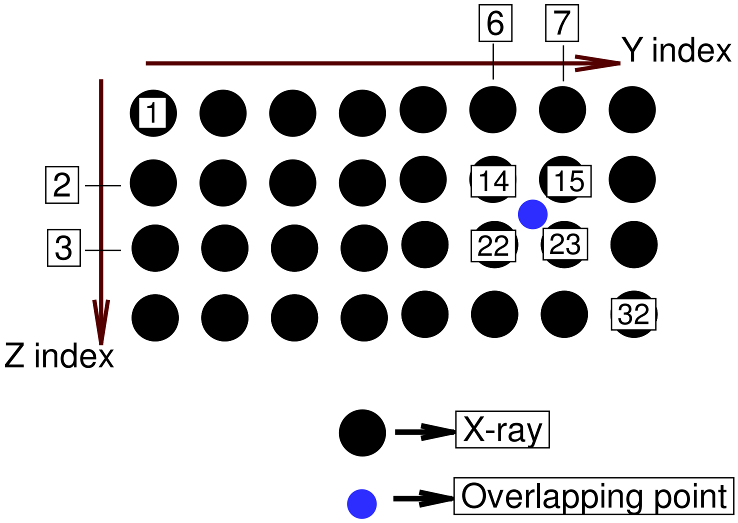


Figure 3.11 Illustration of the distribution and numbering of the rays in X-ray hole cutting method

Algorithm 3.2 Algorithm to get the index of the four rays surronding a certain point which is within the AABB of a solid component. Refer to figure 3.11 for better understanding.

```
// Get spatial position of first ray in data structure
yMin = solid_component.axisAlignedBoundingBox_.yMin_;
zMin = solid_component.axisAlignedBoundingBox_.zMin_;

// 1. Locate the surronding rays
// 1.1 Get the cartesian mapping indexes
yIndex_1 = floor((point.y_ - yMin)/dy_) + 1
zIndex_1 = floor((point.z_ - zMin)/dz_) + 1
yIndex_2 = floor((point.y_ - yMin)/dy_) + 2
zIndex_2 = floor((point.z_ - zMin)/dz_) + 2
// 1.2 Get indexes for data structure
ray_1_index = yIndex1 + (zIndex1 * numberOfRaysInY_)
ray_2_index = yIndex2 + (zIndex1 * numberOfRaysInY_)
ray_3_index = yIndex1 + (zIndex2 * numberOfRaysInY_)
ray_4_index = yIndex2 + (zIndex2 * numberOfRaysInY_)
```

$$\begin{gathered} \mathit{Blankig} = \sum_{n=1}^{4} W_i * b_i \;\; \textit{if Blankig} < \textit{0.5 then grid cell is hole cut} \\ \textit{Where } b_i \textit{ is either 1 or 0 : if the point is between points} \\ \textit{P and Q for the X-ray "i", then } b_i = 1. \\ \eta \textit{ and } \zeta \textit{ represents the adimensionalized coordinates} \\ \textit{of the point to test within the surronding rays.} \end{gathered} \tag{3.3}$$

$$W_1 = (1 - \eta)(1 - \zeta) \tag{3.4}$$

$$W_2 = (1 - \eta)\zeta \tag{3.5}$$

$$W_3 = \eta(1 - \zeta) \tag{3.6}$$

$$W_4 = \eta\zeta \tag{3.7}$$

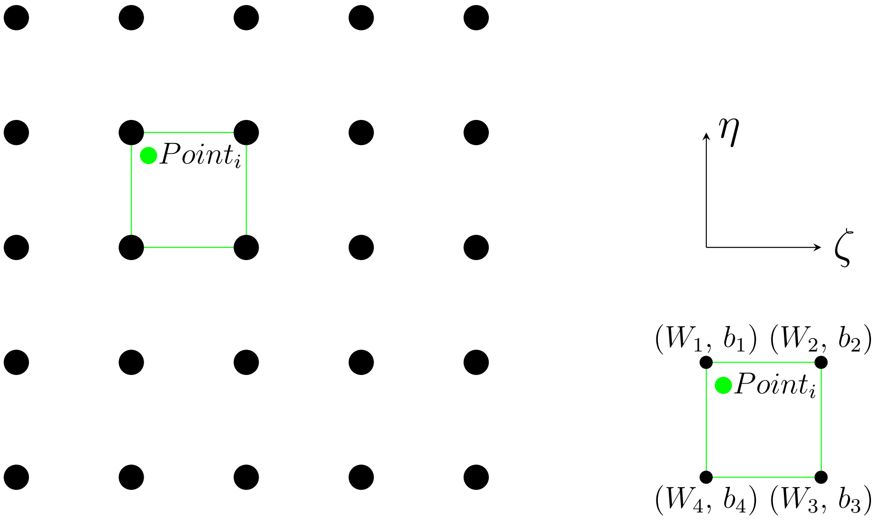


Figure 3.12 Representation of an X-ray distribution. Point $Point_i$ is surrounded by four X-ray as illustrated by the green box.

The first verification was done by simply making sure that entry and exit collision points were found for every cast rays. The verification of the implementation was done mainly visually. Figure 3.13 illustrates the process of visually verifying the X-ray method. First, it was observed that all pierce points were identified as expected as illustrated by the green box in Figure 3.13. Then, blanking the hole cut grid cells or highlighting them in different colours shows that every invalid grid cells are located (two boxes on the top of Figure 3.13).

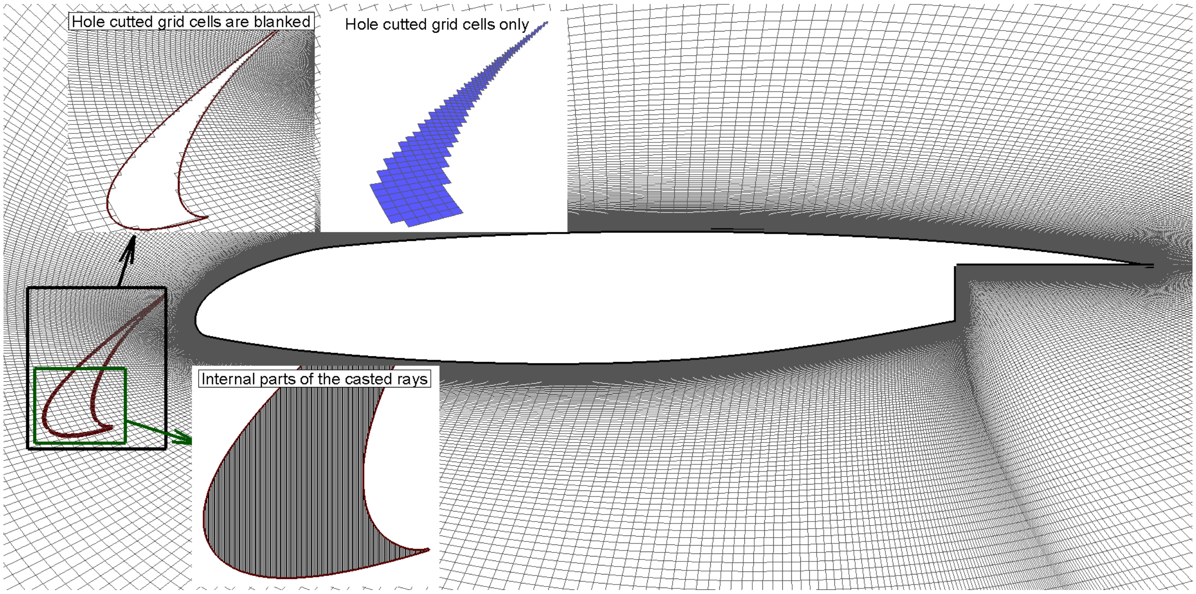

Hole cutted grid cells are blanked
Hole cutted grid cells only
Internal parts of the casted rays

Figure 3.13 X-ray hole cutting method to create hole boundaries applied to an overset case of a multi-element airfoil

More complex 3D cases are also successfully verified. The verification is again here mainly done visually as illustrated in figures 3.14 and 3.15.

Parallel X-ray Hole Cutting Method The parallelization of the hole cutting method is fairly straightforward once the pierce points are computed : each zone of the computational domain, which is already balanced on the different CPUs, verify the inside-outside status of their own local grid cells. The bottleneck is in the pierce points computation step when working on distributed memory. When working on shared memory, it is easier and faster to access the data of the different processes. On distributed memory, however, a step of consolidation on the data is done to communicate the information between the compute nodes. Chapel uses a global namespace which allows direct access to data on different memory nodes without any special consideration. whilst accessing information on other compute nodes is easy to program, continually doing so is not effective in term of time. To optimize this step, a procedure based on the idea behind the MPI_BCAST call from the MPI library is developed : the information that is necessary for each nodes is written and red once from a common data base and then stored locally for futur access. As it will be presented is section 3.4, this approach only allowed to have a *not so negative* scaling when using distributed memory. Review of the implementation could help improve the performances. Whilst this has not been a bottleneck through this project, it could become one when working with very large grids and limited computing ressources.

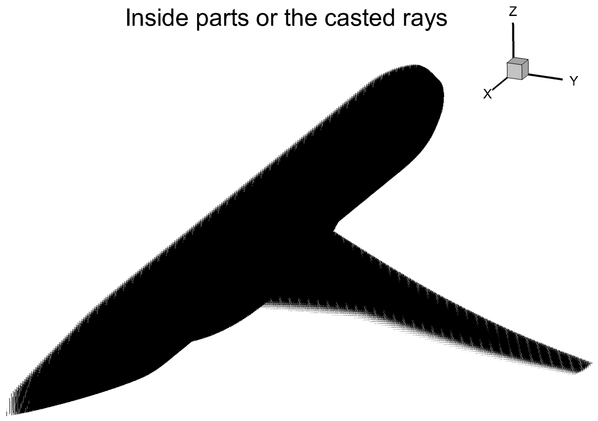


Figure 3.14 Inside part of the cast X-rays on a complete aircraft geometry

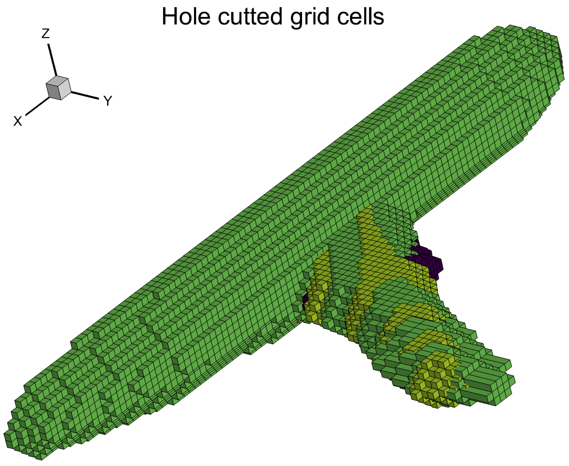


Figure 3.15 Hole cut cells from the X-ray hole cutting approach on a complete aircraft geometry

Advances To The Hole Cutting Method Around Solid Intersections

Collar grids are usually used to connect to intersecting components. The most common use would be to define the wing-fuselage intersection. The Commom Research Model (CRM) configuration was meshed using the overset approach by Boeing and the grids were made available on the DPW4's website. As shown in Figure 3.16, the two grids (i.e., the fuselage grid and wing grid) are only body fitted to their own geometry. A collar grid is applied to properly connect the two grids and allow proper resolution of the flow in this region. Figure 3.17 presents the collar grid used to assemble the fuselage and wing grids. The X-ray hole cutting algorithm would require high density of rays because of the small grid cell sizes in the region and would therefore be considerably more time and memory consuming. This was verified in CHAMPS. Considering that the collar grid was made especially for this region, we could, however, assume that the collar grid is of better quality (i.e., dominant) than the other background grids. That being said, an auxiliary distance-based method is implemented to create a rough hole around the intersection line and thus alleviate the overall process.

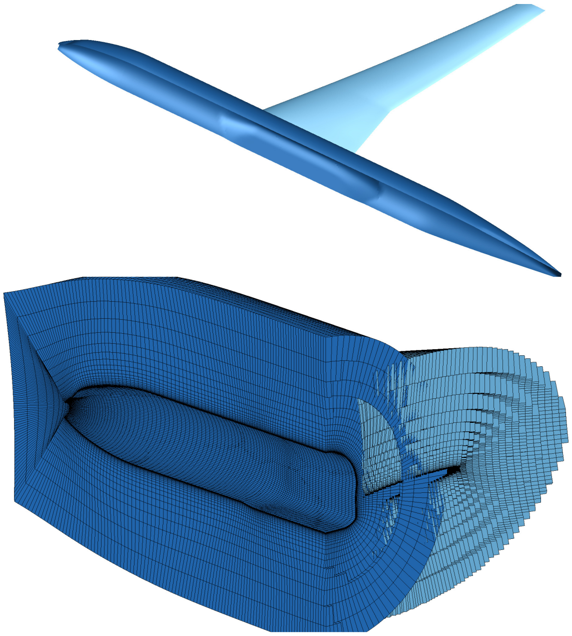


Figure 3.16 Solid intersection between a wing and fuselage. Grids were made available by Boeing for the DPW4

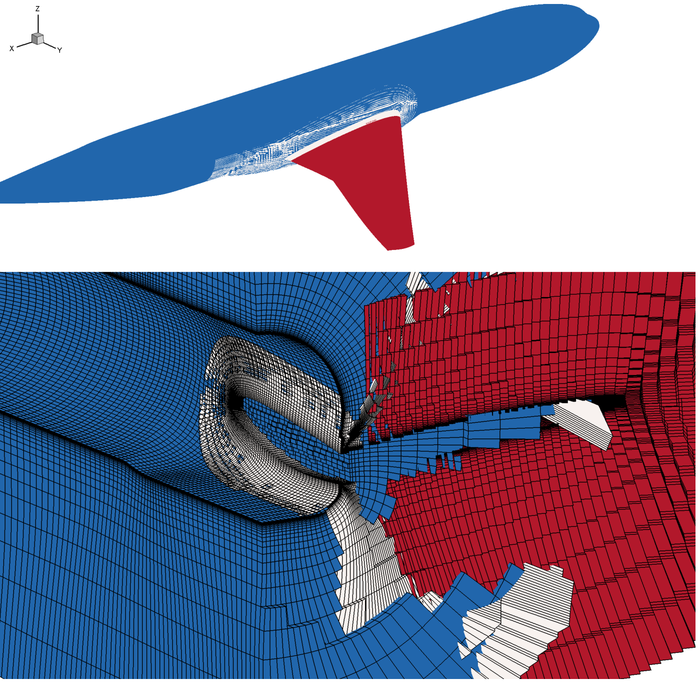


Figure 3.17 Collar grid used to define the solid intersection between a wing and fuselage. Grids were made available by Boeing for the DPW4.

The mean distance between the intersection line and the outer extremity of the different components as illustrated by the red arrow in Figure 3.18 are first computed.

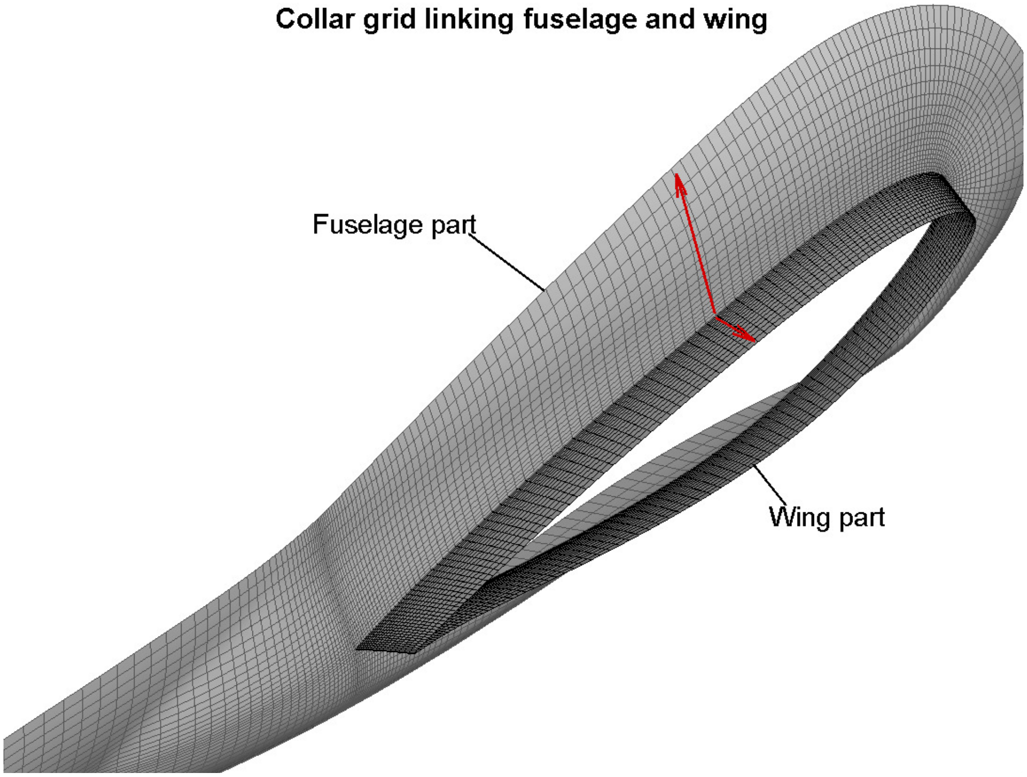


Figure 3.18 Illustration of the distances used in the distance-based hole cutting method around solid intersections

The minimum hole is then created around the intersection line. A fraction of the mean distances as presented in Figure 3.18 is used to control the extent of the hole. To control this fraction, a user input is used in CHAMPS. 70% of the mean distance is usually a good value to compute a proper hole. The best fraction will vary from a case to another since the goal is to position the hole boundary such as the fringe grid cells identified on the border have similar characteristics than their donors. Figure 3.19 present the result of using the method on a wing fuselage configuration assembled using a collar grid.

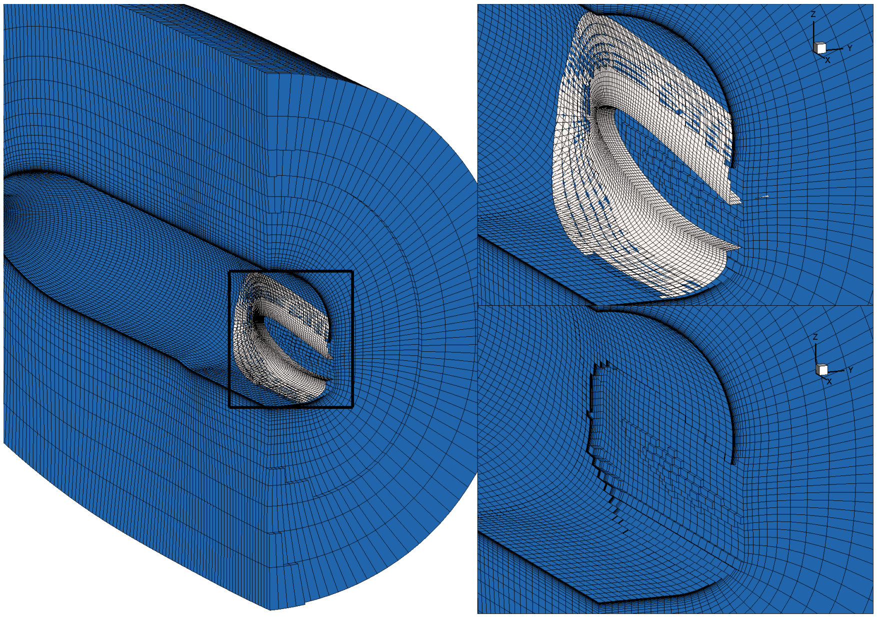


Figure 3.19 Illustration of the distance based hole cutting method around a solid intersection

3.2 Collar Grids

3.2.1 Wall Treatment For The Mismatch In Discretization With Collar Grids

In CHAMPS, the collar grid approach was chosen to treat the solid intersection in overset grid. As presented in the literature review, the collar grid method requires further considerations in order to properly build the overset connectivity. To observe the problem and apply the solution presented by Thorsten Schwarz [3], a similar overset grid is generated. First, orphan grid cells are indeed observed after assembling collar grids. As it can be observed in Figure 3.20, some grid cells in green close to the wall are within the solid body of the overset grid in black and therefore no potential donors can be found with the present method. Also, donors that are defined for grid cells close to the wall will have significantly different wall distance than their receiver counterpart which can have a negative impact especially on the turbulence model [3].

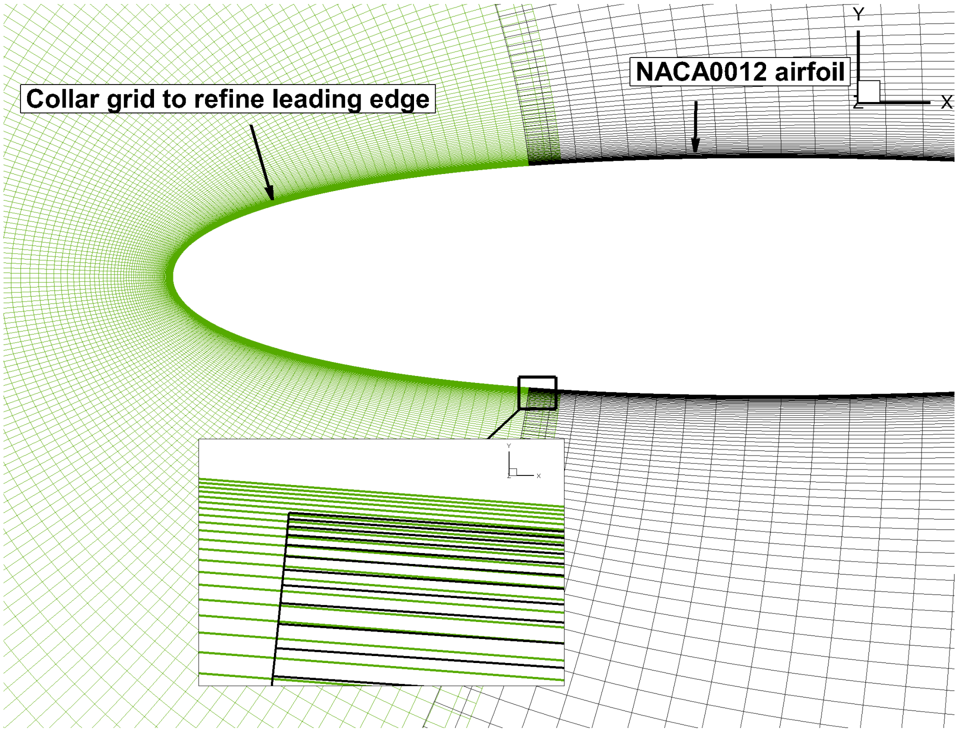


Figure 3.20 Illustration of the surface discretization mismatch under a collar grid

As illustrated in figure 3.21, grid cells can have different wall distance values depending on the grid. Thorsten Schwarz proposes to modify the coordinates of the receiver's centroid to take the different discretizations into account. The algorithm 3.3 presents this correction. The idea is to move the coordinates by *epsilon*, which represent the normal distance between the overlapping solid facets closer to the grid cell. This correction, however, is gradually less applied far from the wall.

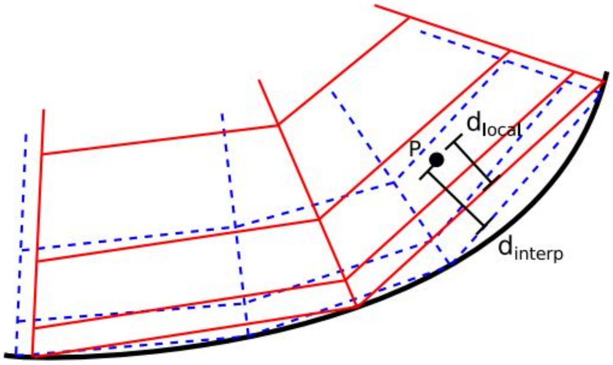


Figure 3.21 Illustration of the wall discretization mismatch proposed by Thorsten Schwarz

Algorithm 3.3 Implementation of the wall treatment proposed by Thorsten Schwarz.

```
// Get the coordinates of the grid cell to interpolate
vector P_coordinates = get_coordinates(P_index)

// Get distance between the two overlapping solid facets
vector epsilon = (d_interp - d_local)*normal

// Parameters to gradually decrease correction far from the wall
float d1 = 10.0*normL2(epsilon)
float d2 = 30.0*normL2(epsilon)

// Weight for the correction
float weight;

if      d1==d2 && d1==0 then weight = 0.0;
else if d_local >= 0 && d_local < d1 then weight = 1.0;
else if d_local >= d1 && d_local < d2 then weight = (d2 - x_ps)/(d2 - d1);
else if d_local >= d2 then weight = 0.0;

// Apply correction
NEW_COORDINATES = P_coordinates + weight*epsilon
```

Verification

Figures 3.22 and 3.23 present the reproduction of the problem observed by Thorsten Schwarz [3] in the skin friction distribution using CHAMPS. The grid used here is presented in Figure 3.20 . The flow solver was set with the following parameters : Mach=0.8, AoA=1.25, second order discretization and Spalart-Allmaras turbulence model. First, Figure 3.22 presents the problem as expected. Then, the solution is indeed able to resolve the orphan grid cell problems and to rectify the spike observed in the skin friction distribution as shown is Figure 3.23.

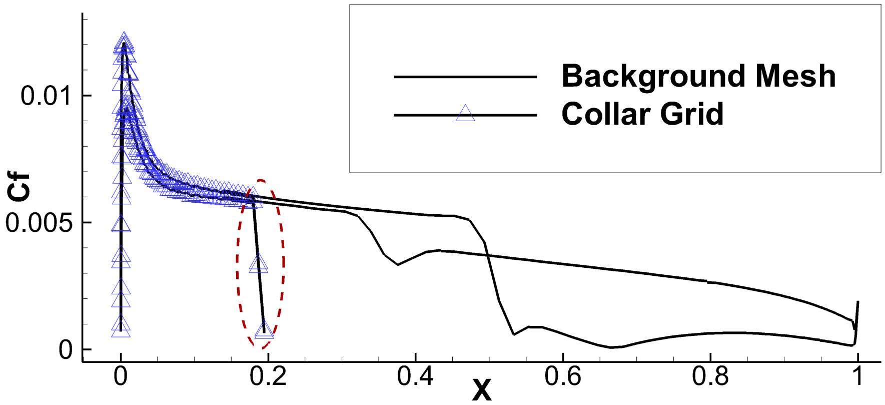


Figure 3.22 Skin friction around a NACA0012 airfoil using a collar grid on the leading edge. No correction is applied here.

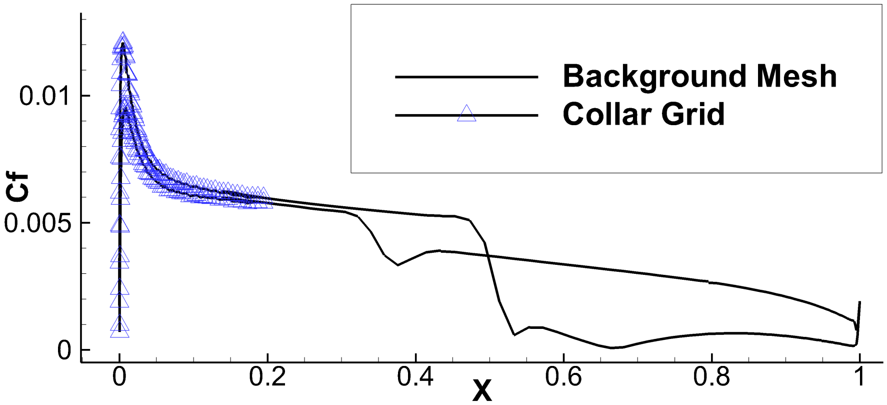


Figure 3.23 Skin friction around a NACA0012 airfoil using a collar grid on the leading edge. The wall treatment proposed by Thorsten Schwarz [3] is applied here

3.2.2 Surface Duplication

To properly compute surface integration, the overlapped surfaces must be weighted so no area is counted twice. The method implemented in CHAMPS projects the faces of the solid boundaries onto the underlying boundaries. A first loop through the boundary faces is done to determine which face is completely covered by the projection : if every node of a face is covered, the weight can be directly set to 0. A second loop is done on the faces onto which only one of the two nodes is covered : the weight of these faces are between 0 and 1. The shadow areas on those faces are computed by computing the intersection between the projected faces and the underlying ones as illustrated in Figure 3.24. Surface elements can be of different shapes and the overlap is arbitrary. Therefore, the new shadow elements can be of arbitrary shape also. To compute the area, the new surface element is divided in sub-triangles using the edges and the centroid.

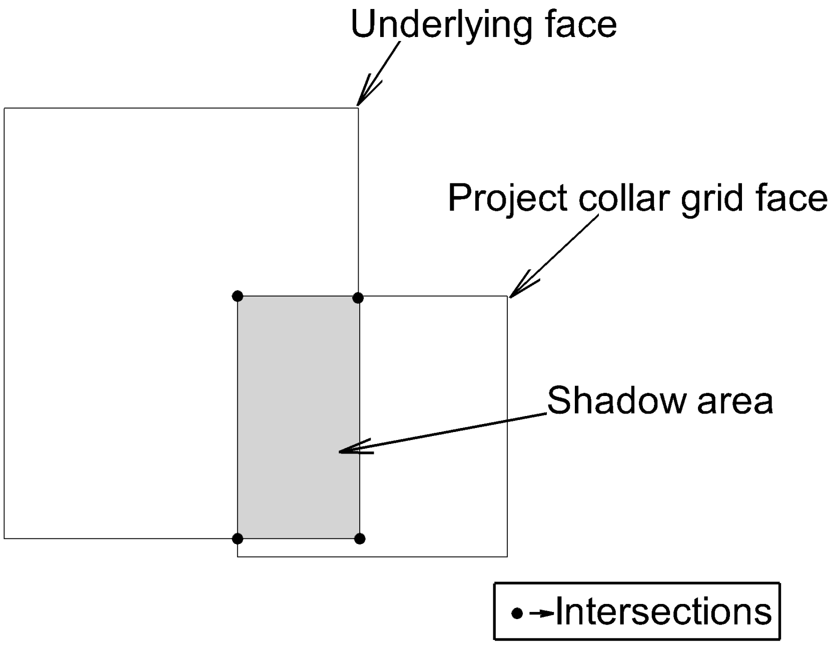


Figure 3.24 Weighting overlapped solid boundary faces by reconstructing projected faces

Verification

Integrating the faces normal around a closed body theoretically gives zero. In order to properly compute the aerodynamic coefficients, which require surface integration, both the overset communication and surface weighting need to be accurate. That being said, the first test is done using a collar grid perfectly coincident to the background discretization as presented in Figure 3.25. The collar grid is used here to assemble the multi-blocks grid without needing to create halo cells at the blocks interfaces. The results using the collar grid approach and the standard single grid approach are expected to be the same since the final grids are the same with the two approaches. Table 3.1 shows, as expected, similar results within machine accuracy using the overset and single grid approach.

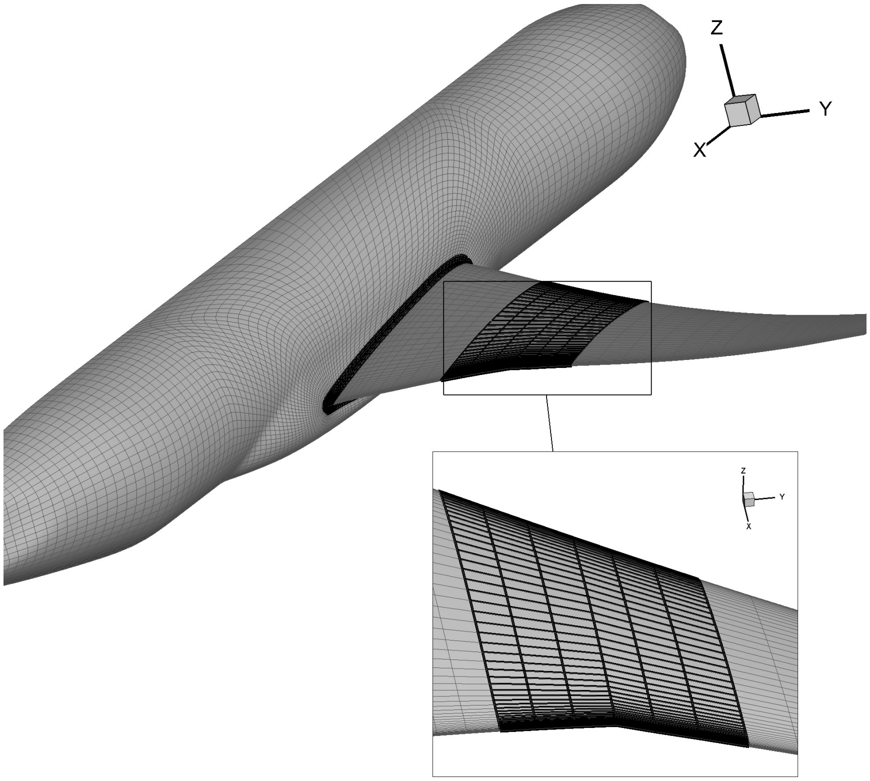


Figure 3.25 Coincident collar grids used to assemble a multiblock grid of the CRM wing fuselage configuration

Table 3.1: Simulation on the CRM wing fuselage configuration using the overset and single grid approach. Parameter : Mach=0.85, CL=0.5, Re=5e6

	Overset	Single grid	abs(Delta)
CD	0.02628	0.02628	< 1e-14
AOA	2.2450	2.2450	< 1e-14

It is more complicated to verify the algorithm with non-coincident collar grids : it is expected that the method is not perfect since the geometry is not really closed by the approach. As mentioned above, surface elements completely shadowed by the collar grid are easier to locate and thus the errors due to the approximations in the method are concentrated at the interface between the grids. In addition, such regions usually represent a small amount of the overall surface area and thus do not have a great impact on the overall solution. It is also anticipated that grid refinement will minimize this error.

The following test case presents the impact of using a non-coincident collar grid as presented in Figure 3.26 on the solution. The overset grid is composed of a cylinder on which a collar grid is applied. The collar grid used here has the same discretization as the background grid. First, it is observed that the weighting method is not perfect as the surface's integration on the overset grid with non-coincident collar grid does not return a null value in comparison to the assembly with the coincident one.

A grid convergence study is done to measure the impact of this on the solution. Similarly to the last section, results with the coincident collar grids are similar to the single grid approach within machine accuracy which is as expected because the final grids from the solver's point of view are the same. Figure 3.27 shows the grid convergence of the drag coefficient with the overset and single grid approach. The results are at first sight highly similar. Figure 3.28 presents the difference between the two approaches compared to the grid refinement. As anticipated, the results converge to the same value when refining the grids.

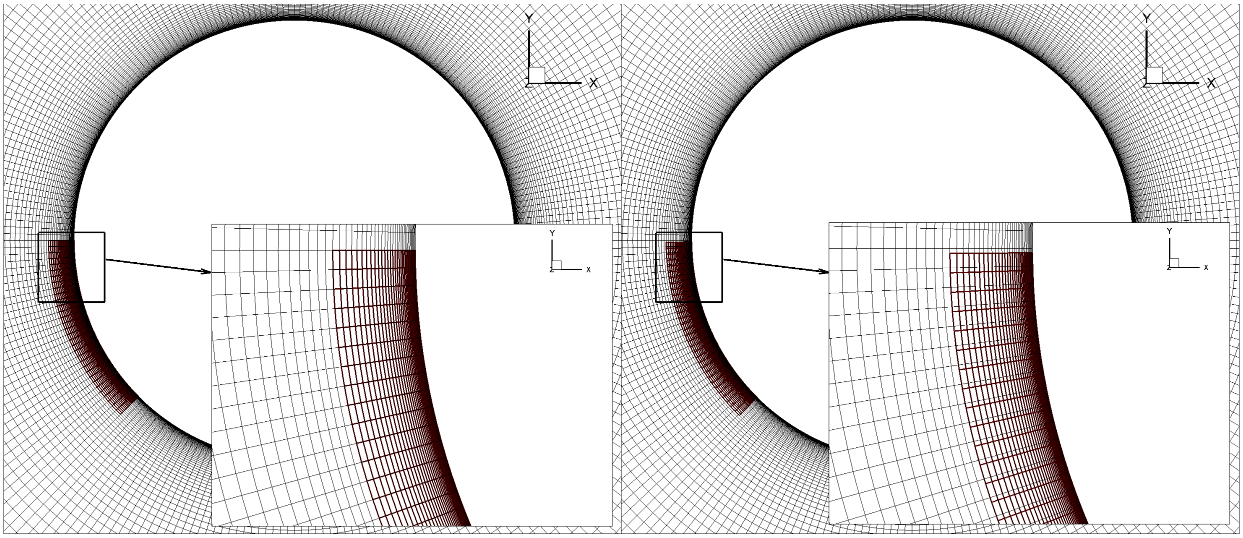


Figure 3.26 Overset grids : cylinder and collar grid. Coincident collar grid on the left and a collar grid with the same discretization but translated on the surface on the right.

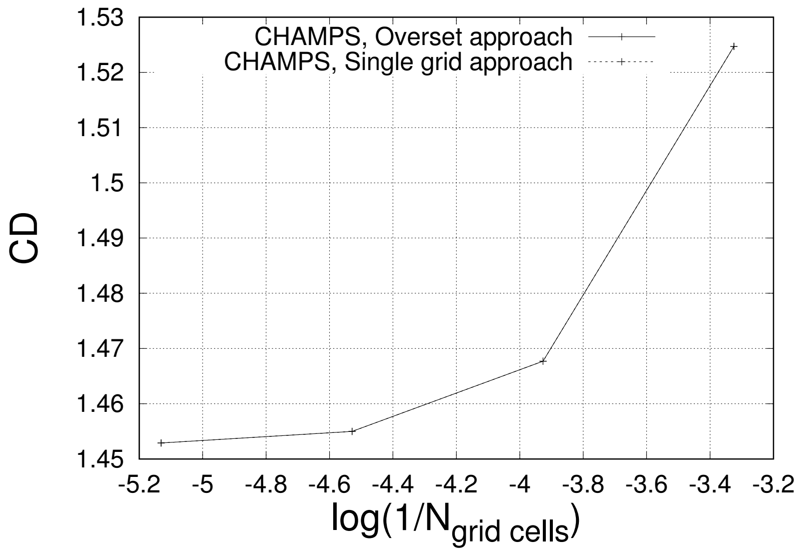


Figure 3.27 Grid convergence analysis. Drag coefficients using the overset and single grid approach

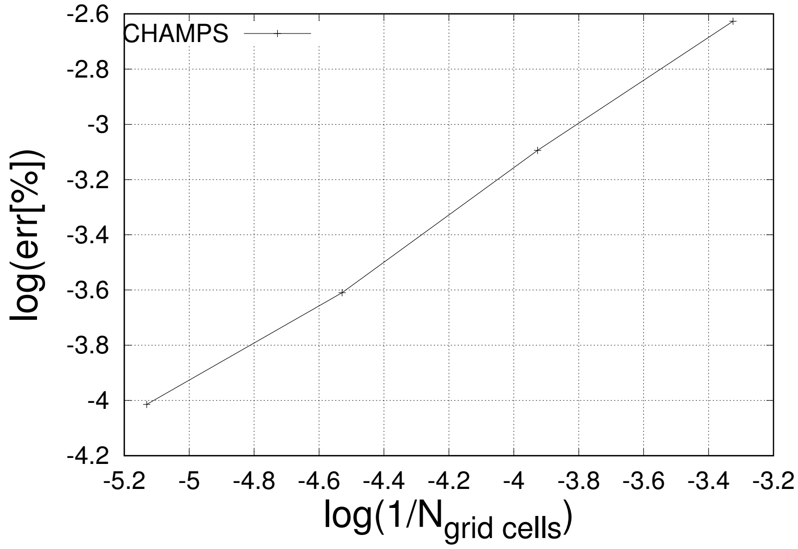


Figure 3.28 Grid convergence analysis. Comparison between the overset and single grid approach

3.3 Oct-tree Data Structure

As presented in the literature review, the oct-tree data structure is recursively constructed from a starter node. The oct-tree structure is implemented to stay coherent to the fact that

CHAMPS is a parallel CFD solver. With the overset approach, it is fairly straightforward to solve the computation on multiple nodes by assigning a different sub-grid to each compute nodes. Furthermore, CHAMPS divides the sub-grids in multiple zones to accelerate the computations even more. To stay coherent with the overall CHAMPS framework, the oct-tree data structure is therefore built on each zone in parallel.

In each zone, the first node of the oct-tree data structure is defined around the whole domain of the zone. Starting from this first level, the domain is recursively divided into eight child nodes until certain conditions are met. The user can control the depth of the structure by fixing either a maximum depth or a maximum number of grid cells that are allowed within the finer level nodes. Figure 3.29 illustrates the oct-tree structure created on the grid presented by Vassberg and Jameson [51] for different parameters.

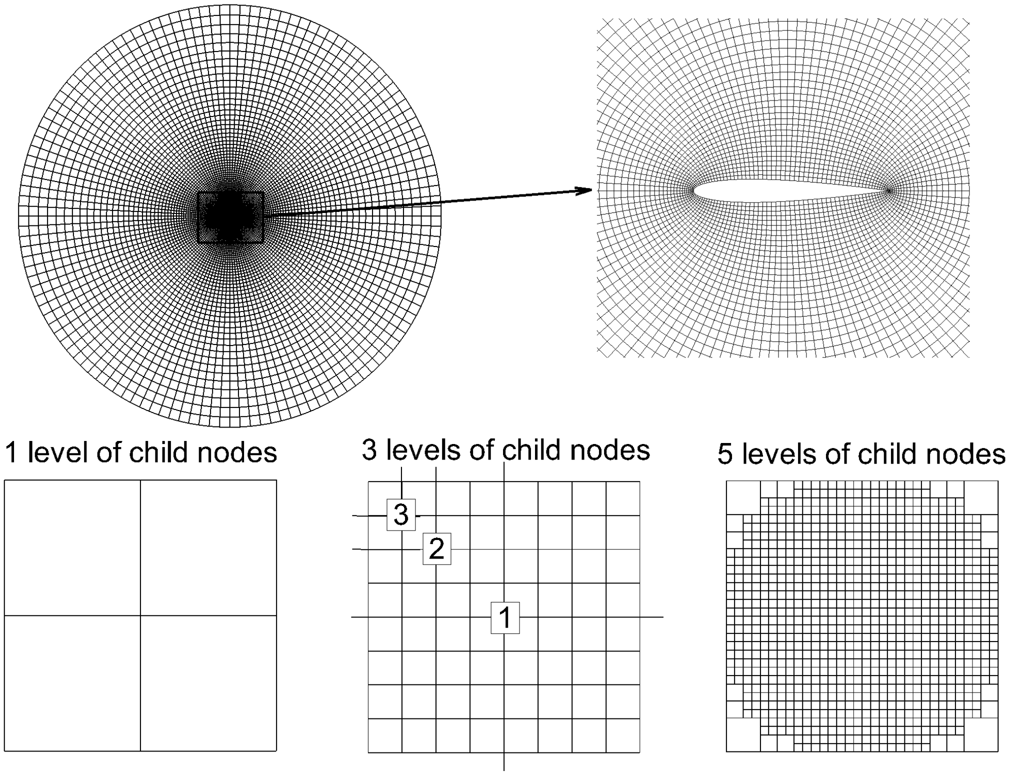


Figure 3.29 Illustration of the oct-tree structure created on a NACA0012 Euler grid. Illustration of the structure with one, three and five maximum levels of child nodes are compared.

The overall overset process requires many searches through the database to find grid cells based on their approximate location. In the donor search algorithm, for example, a certain grid cell is searched based on the coordinate of a point within it. The oct-tree data structure stores the different information in a tree manner so that given some coordinate, the structure can quickly provide a shorten list of the surrounding grid cells. Starting from the top node of the structure, the eight direct child nodes are tested to verify if it contains the given coordinate : this is fairly easy and quick to verify if a certain coordinate is within a bounding box. This is done recursively from the first node to their child nodes until the last child node containing the coordinate is found. The length of the returned list compared to the total number of grid cells depends on the depth of the oct-tree : the finer the last level is, the shorter the returned list is, but also the heavier the structure in terms of memory.

The oct-tree data structure must be used with proper parameters. For a deeper oct-tree, the cost of retrieving information increase. So whilst the oct-tree data structure decreases the number of operations necessary, it needs to be adequately parameterized so the operations to build the oct-tree and use it stay competitive compared to the time saved. The structure is controlled by two parameters : (1) a maximum depth allowed for the oct-tree and (2) a critical number of grid points in a single node. In CHAMPS, the first one is more important than the latter.

To verify the implementation, the performances of the structure are verified on a three-dimensional Cartesian grid containing 1 million grid cells. One hundred thousand points were randomly generated in the domain to compute the donor search. From an oct-tree with 1 level of nodes to an oct-tree with 6 levels, the algorithm accelerates the donor search process as presented on the log-log graph of Figure 3.30. This analysis shows that fixing those parameters in order to get finer child nodes containing around 0.03% of the total number of grid points usually gives the best results. Some results of this are presented in Figure 3.31 : in this test case, using four levels of child nodes provides the best acceleration. On the other hand, when allowing five or six levels of child nodes, the time needed to build the oct-tree becomes larger than the time saved.

3.4 Performance Of The Overset Preprocessor

The three most time consuming steps are the X-ray method, the donor search and the wall distance computation. As predicted by reading the literature, the bottleneck is in the scalability of the X-ray hole cutting algorithm. In the case of complex geometries, a large number of rays are necessary to ensure a proper hole cutting. For each of those rays, many pierce points can exist. To optimize the process, CHAMPS decomposed the sub-grids in *zones*. This is done in a similar fashion than the multi-block approach often used with the

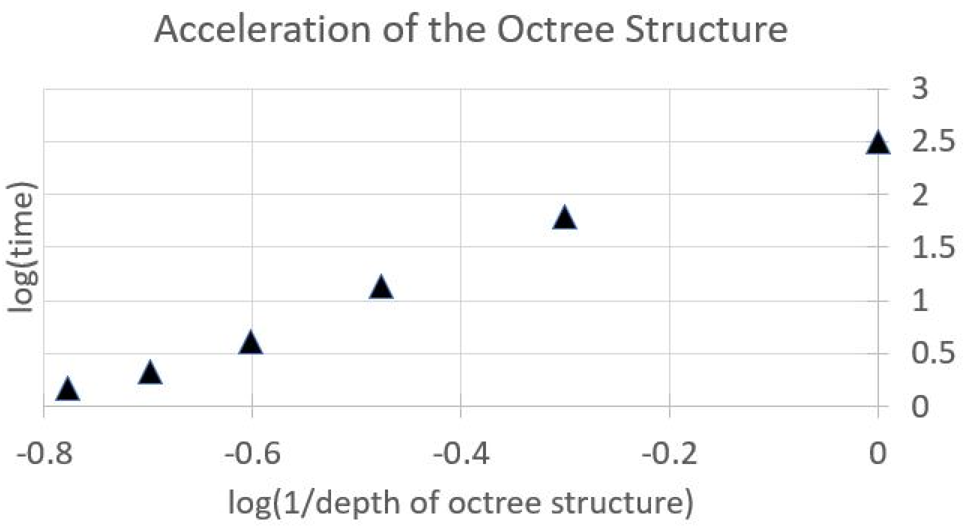
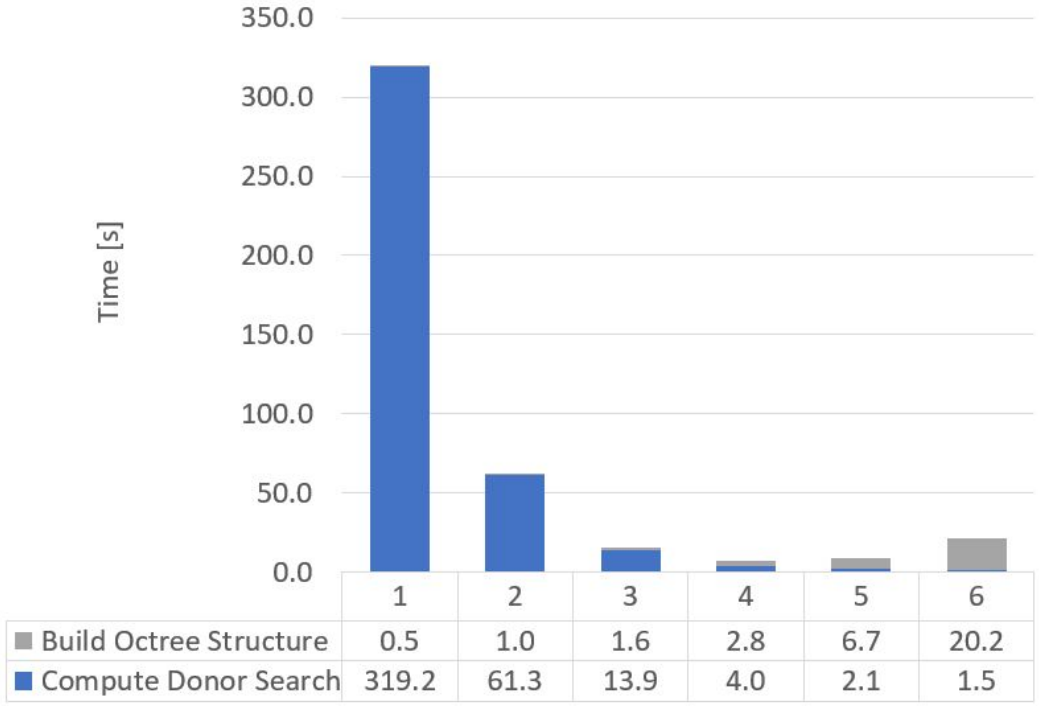


Figure 3.30 Time to compute the donor search for one hundred thousand randomly generated grid cells within a background grid composed of a million grid cells



	1	2	3	4	5	6
▨ Build Octree Structure	0.5	1.0	1.6	2.8	6.7	20.2
■ Compute Donor Search	319.2	61.3	13.9	4.0	2.1	1.5

Figure 3.31 Time (in seconds) to build oct-tree data structure and compute donor search for different depths of oct-tree structure

structured grid approach. Those zones are distributed on different tasks that can potentially be run on distributed memory. Therefore, a potentially very large number of piercing points information has to be communicated throughout the tasks.

The overset grids shown in Figure 3.32 are used to verify the performances of the overset method in CHAMPS. There is a background grid with around three million grid cells and a near-field grid containing a spherical solid body with around half a million grid cells. More than fifty thousand X-rays are casted in the solid body to compute the hole cutting. This is done on Compute Canada's cluster called *Graham* using up to 64 CPUs. These nodes are composed of 32 CPUs each. When using 64 CPUs, for example, 2 nodes of 32 CPUs are used.

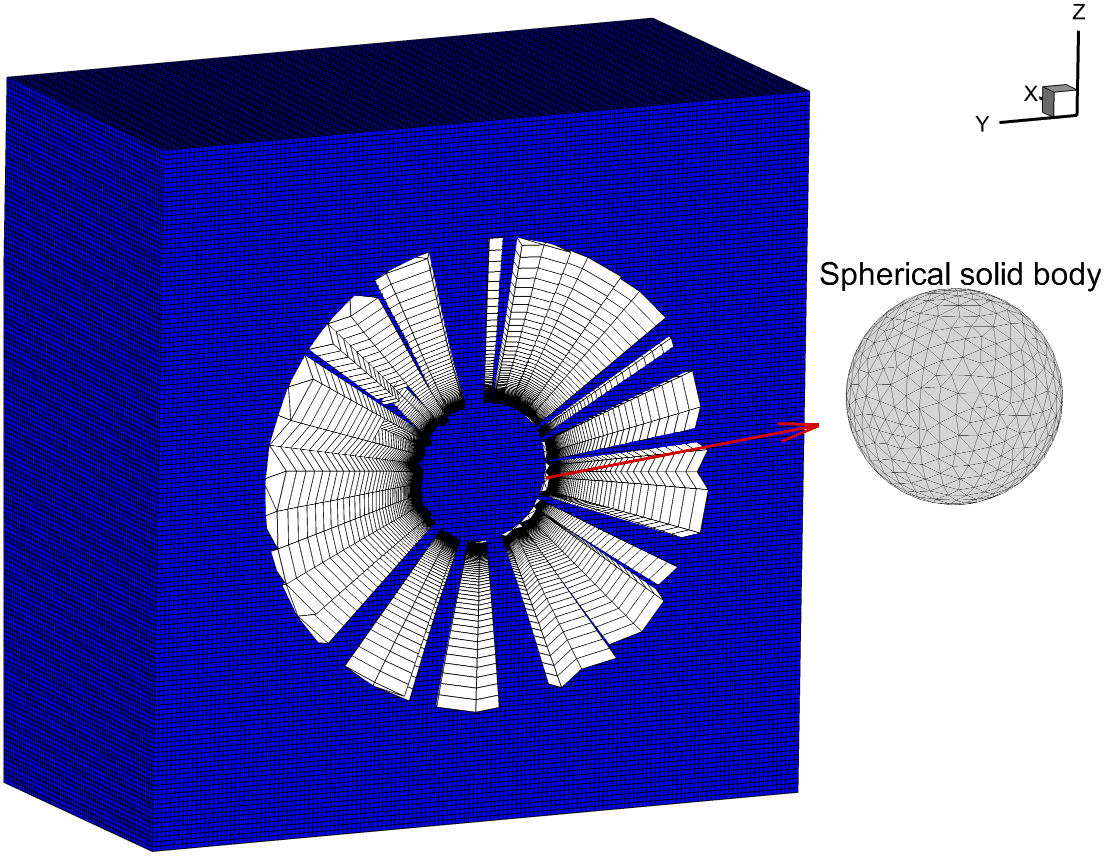


Figure 3.32 Overset grid used to verify the scalability of the overset method in CHAMPS

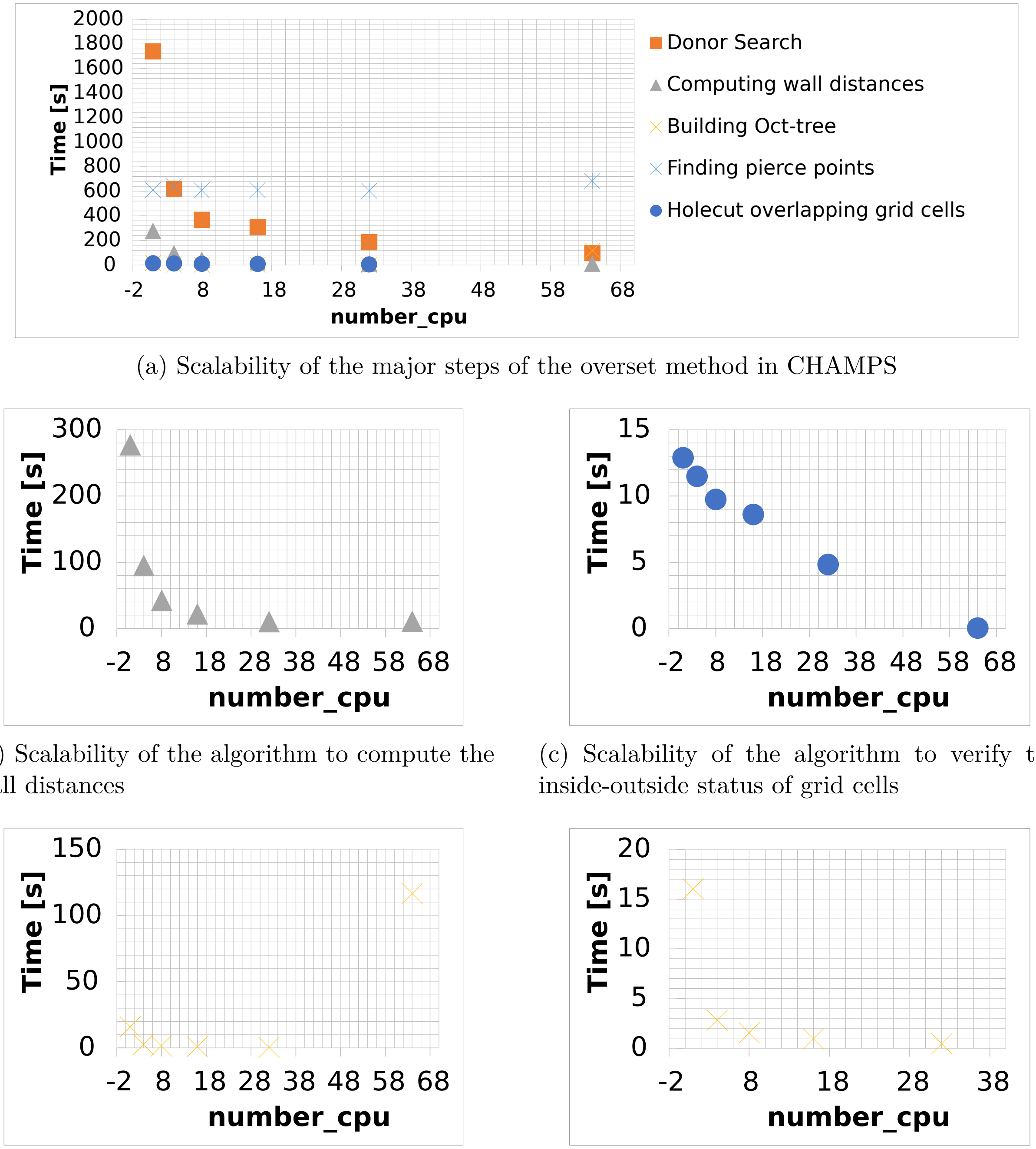
The results in Figure 3.33a show that the overall overset method can be accelerated by using more CPUs (i.e., scalable). The preprocessor developed is fully parallel, which means that every steps of the algorithm is split throughout the available CPUs. That being said, the CPUs often needs considerable amounts of information generated by other CPUs. This is straightforward in shared memory, but not as much in shared memory. The performance of the different methods on more CPUs and compute nodes differ from one and other and mainly depends on how and the amount of information shared between shared and distributed memory. The observations are as follow :

- The first observation is that the donor search (i.e. the orange squares in Figure 3.33a) is the most dominant function in terms of time. In this test case, each time the number of CPUs is doubled, the algorithm speedup is between 1.5 a 1.95.

- Because of the dominance of the pierce points computation in the hole cutting step, it was considered separately. Locating the piercing points coordinates is next in time consumption : this step does not scale when using more CPUs, and even stays close to constant as it is presented by the blue stars in Figure 3.33a. The step becomes more time expensive than the donor search when using more than 8 CPUs. From the author's point of view, the time of this step is acceptable considering the quantity of X-rays and tests that ought to be done : it did not limit the application of the overset method during this thesis.

- The algorithms to compute the wall distances and to locate the invalid grid cells show scalability similar to the donor search step when using more CPUs as shown in figures 3.33b and 3.33c.

- To avoid too much communication in the donor search step, the oct-tree data structure is duplicated on each node. Figure 3.33e shows that building the oct-tree data structure using more CPUs on a single node allows scalability. In Figure 3.33d, however, it can be observed that when using an additional compute node (i.e., distributed memory), the step becomes significantly more time consuming. This can be expected because of the relatively large amount of data that is communicated to consolidate the information through the compute nodes. Furthermore information necessary to the donor search step is also communicated and store in the oct-trees in order to speed up the donor search step.



(a) Scalability of the major steps of the overset method in CHAMPS

(b) Scalability of the algorithm to compute the wall distances

(c) Scalability of the algorithm to verify the inside-outside status of grid cells

(d) Scalability of the algorithm to build the oct-tree data structure

(e) Emphasis on the values obtained on a single compute nodes from Figure 3.33d

Figure 3.33 Verification of the scalability of the overset method in CHAMPS on up to 64 CPUs on an overset grid with about 3.5 million grid cells

CHAPTER 4 RESULTS FROM TWO-DIMENSIONAL APPLICATIONS

Two test cases are presented here. First, Koji Morinishi [52] presents numerical results for a transonic flow over two NACA0012 airfoils staggered from each other using a 105x81 structured grid. With the overset approach, two Euler grids from Vassberg and Jameson [51] are assembled by CHAMPS to represent the domain. The results are in accordance with the literature and show sharp capture of the shock between the two airfoils.

Secondly, as part of the Mesh Effects for CFD Solutions special session at AIAA Aviation 2020, a set of eighteen overset grids were custom tailored at NASA to yield wake clustering that is appropriate for this test case which is an airfoil section of the CRM [53]. Those grids provide a pertinent benchmark for two-dimensional overset applications in the industry. The results obtained with CHAMPS are in general accordance with the literature, but further analysis of the results indicate limitations with the grids.

4.1 Staggered NACA0012 Airfoils

Two 129x129 grids as presented in Figure 4.1 are used : the first grid is centred at the origin and the second one is staggered horizontally and vertically by half a chord length. The implicit approach is used for this assembly using the wall distance values as criteria in the donor search step. This means that the grid cells closer to an overlapping solid body are to be interpolated. This created a smooth interface as presented in Figure 4.1. The first two layers of interpolated grid cells close to the interface are defined as computed in order to distanciate the interfaces. This help smooth the communication and make sur to avoid implicit interpolation between two grid cells [54].

Figure 4.2 presents the isobar lines of pressure in the flow field around the airfoils. The result is smooth and in agreement with the literature. Figure 4.3 shows the pressure distribution on the two airfoils in comparison with the results of Koji Morinishi [52]. The strong shock wave is properly captured on the overset assembled mesh as shown on Figure 4.3, which indicates a proper communication stencil between the meshes. Differences between Morinishi and CHAMPS can be explained by the difference in the numerical scheme and grids.

4.2 Airfoil Section Of The CRM-HL Configuration

Figure 4.4 presents the coarser set of eighteen overset grids developed by NASA. From what the author has understood, the solver for which those grids were generated uses a vertex-

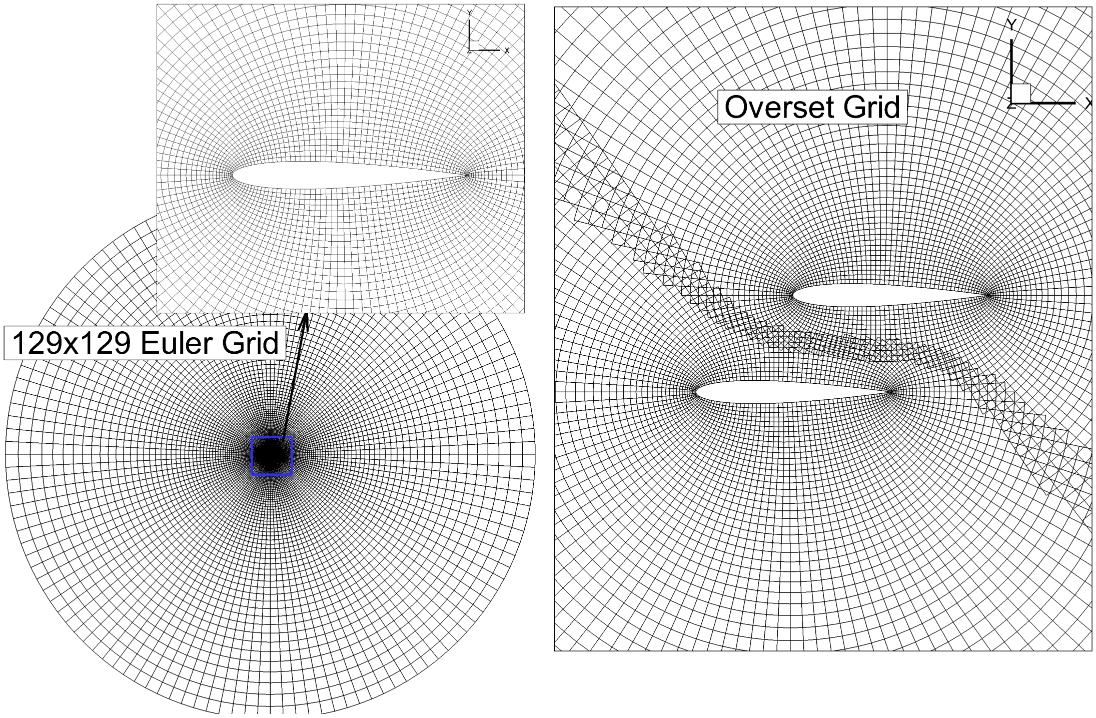


Figure 4.1 Two staggered NACA0012 airfoils. Overset grid composed of two 129x129 Euler grids.

centred approach. As already mentioned, an additional layer is available with the same grid when using the vertex-centred approach. Only two layers of grid cells were available in most overlap region to build the connectivity and as presented in a previous section CHAMPS would need three layers in order to properly compute the gradients and limiters. The gradients of the donor grid cells are therefore used in this simulation to compute the limiters. Furthermore, in this case, the interpolation occurs in high gradients regions close to the surface. As mentioned in the literature review section, this can have a negative impact on the converged solution when using a non-conservative interpolation method, which CHAMPS does. Those are two possible error sources that are kept in mind in the result analysis. However, it is anticipated that such errors will decrease with a grid refinement.

The grid convergence study presented in Figure 4.5 shows the results obtained with CHAMPS using different limiters. The results are compared with FUN3D on two families of grids using the single grid approach. First, the results are all in general agreements with the references (i.e. results on the single grid approach and the FUN3D results). The lift coefficients obtained with the overset approach without any limiter are higher than the references. From the literature review, such increase could be expected considering the *non conservation* of the interpolation method.

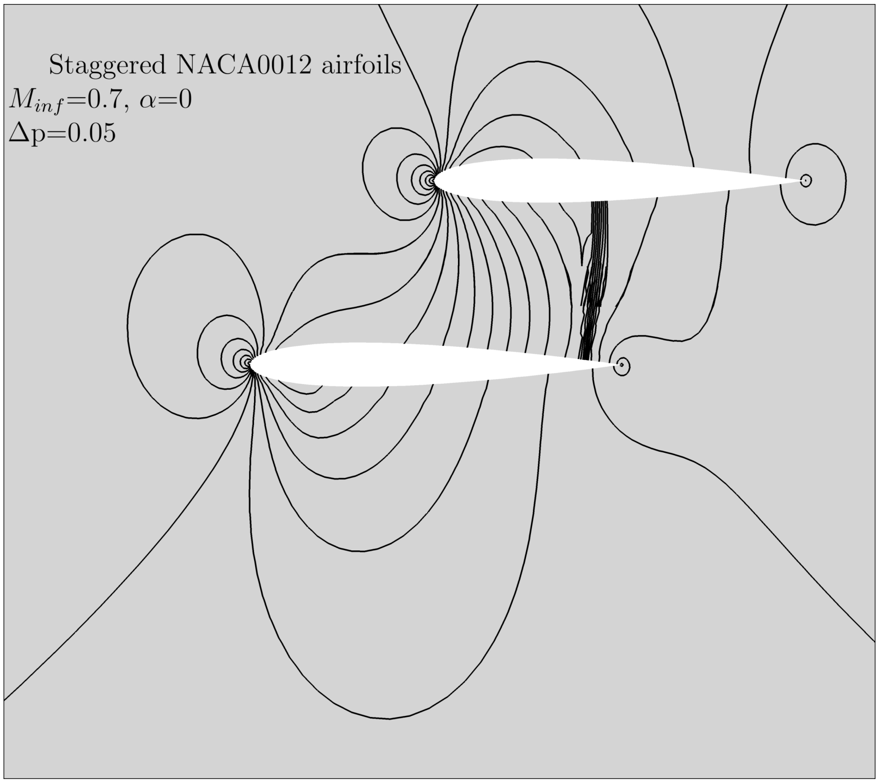


Figure 4.2 Isobar lines of pressure in the flow field around two NACA0012 airfoils staggered by half a chord length

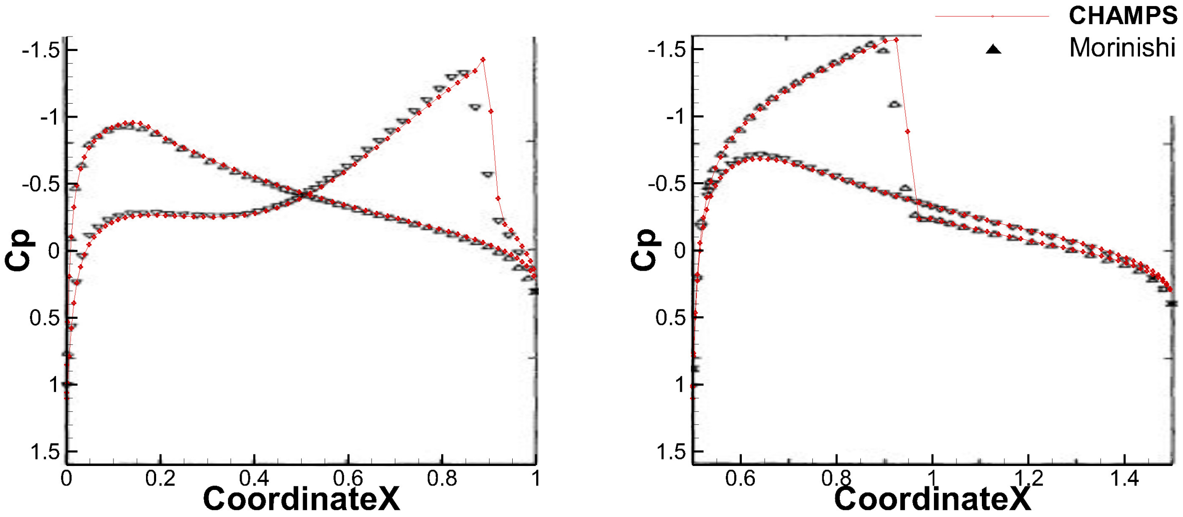


Figure 4.3 Pressure distribution on two NACA0012 airfoils staggered by half a chord length. On the left is the centred airfoil, and on the right the staggered one.

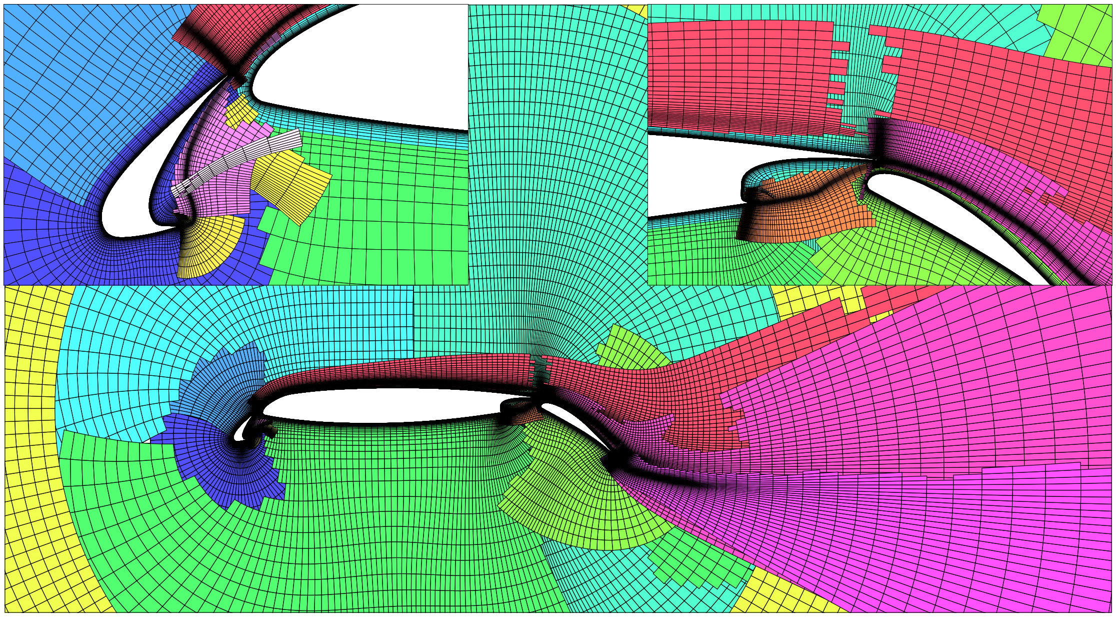


Figure 4.4 Overset grid developed at NASA of an airfoil section of CRM-HL. Overset connectivity created with CHAMPS.

Furthermore, the results obtained with limiters which are more *limiting* seems to be closer to the other results. The fact that they are more dissipating could minimize the error resulting from the non-conservation. Figures 4.6 and 4.7 presents the pressure and skin distribution respectively on the second finest set of overset grids compared to results from FUN3D [53].

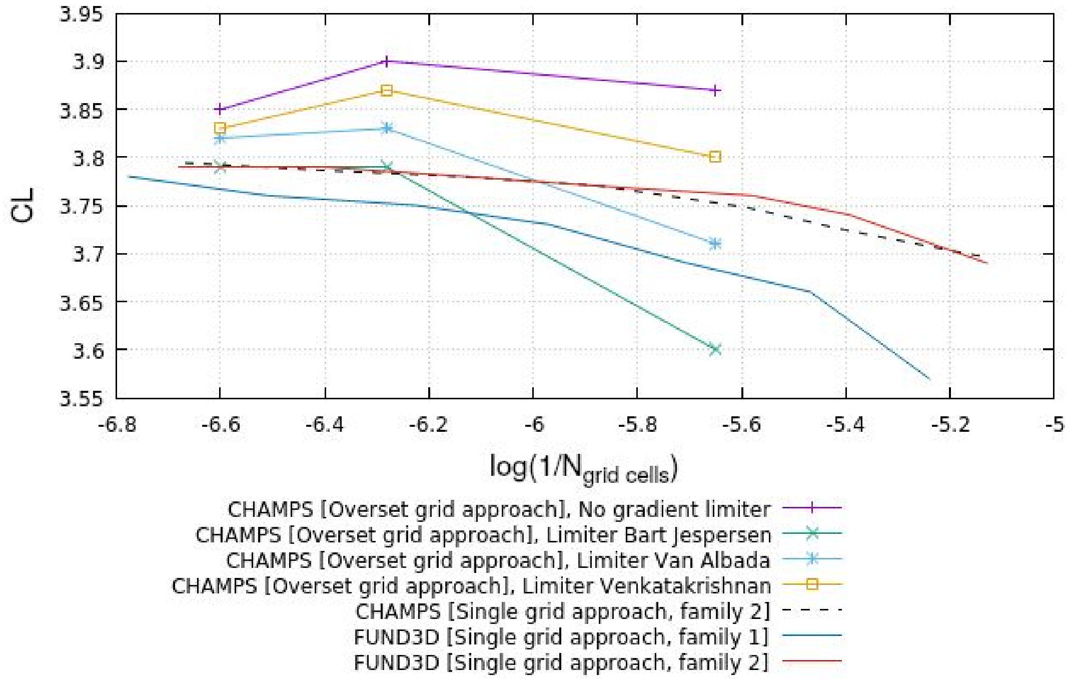


Figure 4.5 Lift coefficients obtained on different levels of the grid refinement. CHAMPS Results using the overset and single grid approach compared. Results are compared to FUN3D.

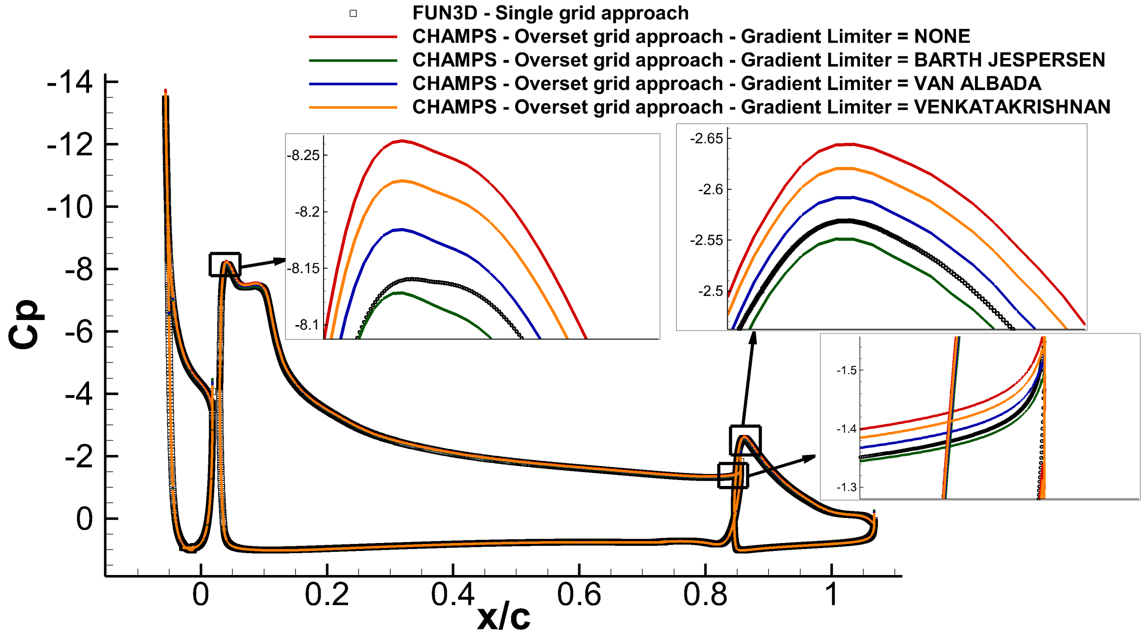


Figure 4.6 Pressure distribution on an airfoil section of the HL-CRM configuration. Results from FUN3D were generated with a single grid approach containing about 5 million grid cell and CHAMPS used overset grids composed of around 4 million grid cells.

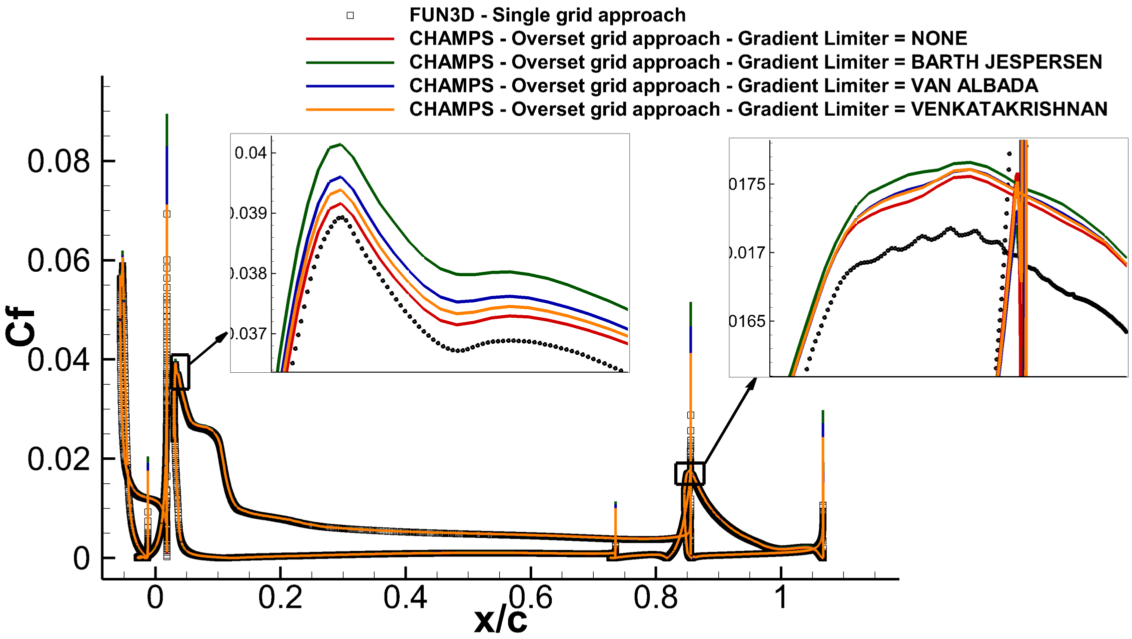


Figure 4.7 Skin friction distribution on an airfoil section of the HL-CRM configuration. Results from FUN3D were generated with a single grid approach containing about 5 million grid cell and CHAMPS used overset grids composed of around 4 million grid cells.

CHAPTER 5 RESULTS FROM THREE-DIMENSIONAL APPLICATIONS

Industrial applications are often three-dimensional and require the overset method to be able to treat solid intersections. Such intersections often, if not always, imply collar grid and surface overlaps. Using grids available on the website of the fourth and fifth drag prediction workshops [55, 56], it is possible to further verify the limits of the new overset method. Different overset cases are presented to verify the proper implementation of the solid intersection capabilities. The geometry analyzed is the **NASA Common Research Model (CRM)**. This model was created to improve the state-of-the-art in CFD applications by Langley Research Center and Ames Research Center [57]. This test case is therefore highly relevant for the present thesis.

5.1 Canonic Test Case With Coincident Collar Grids

Here, two coincident collar grids are applied on the geometry. Coincident means that the collar grid and the background grid have the same discretization in their overlapping domain. The final global grid obtains is identical to the single grid approach from the point of view of the flow solver. Flow simulations with the single grid and overset approach should therefore give the same numerical results. Figure 5.1 present the overset grids. Pressure distribution from the overset and single grid approach are compared at two positions on the wing which are both near the collar grids as shown in Figure 5.2.

5.1.1 First Order Simulation

The flow solver is set up using the parameters described in the table 5.1. Simulations with a first order discretization scheme gives the same numerical results within the accuracy to which the simulation was computed. The simulation is converged with a order of precision smaller than 1.0e-4.5 on the density residual as shown in table 5.2. Figures 5.3 and 5.4 show the pressure distributions and the convergence of the density and lift coefficient respectively.

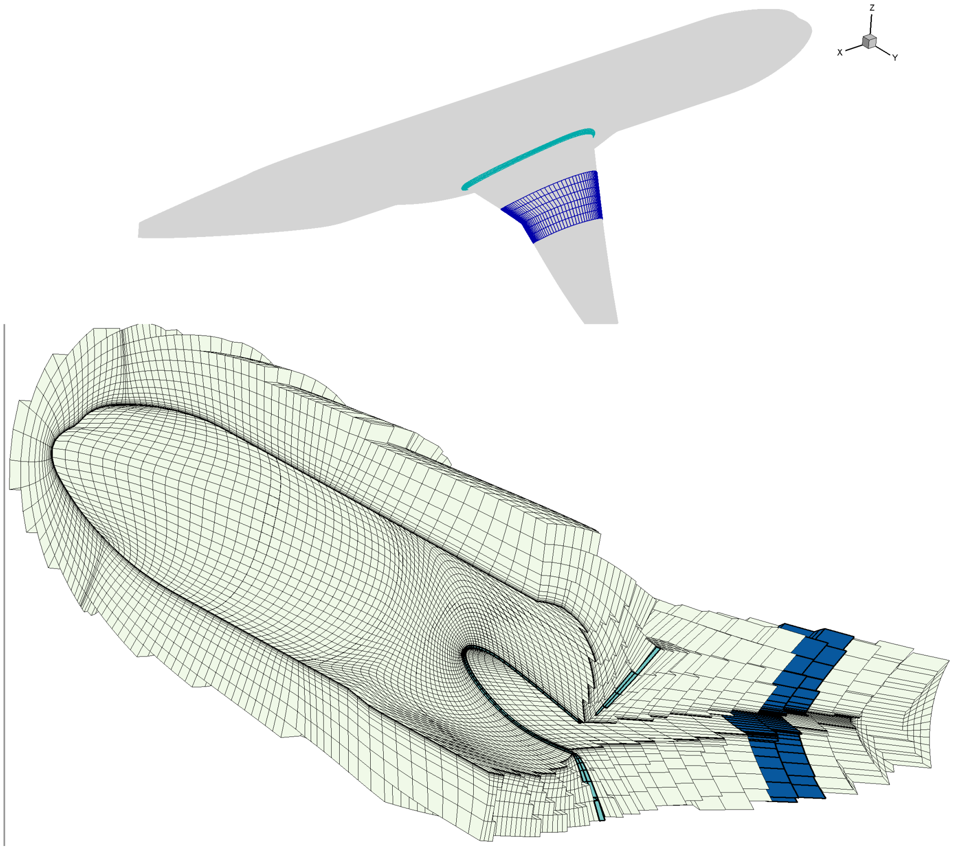


Figure 5.1 NASA Common Research Model (CRM), wing-body. Coincident collar grids on the wing-body intersection and on the wing.

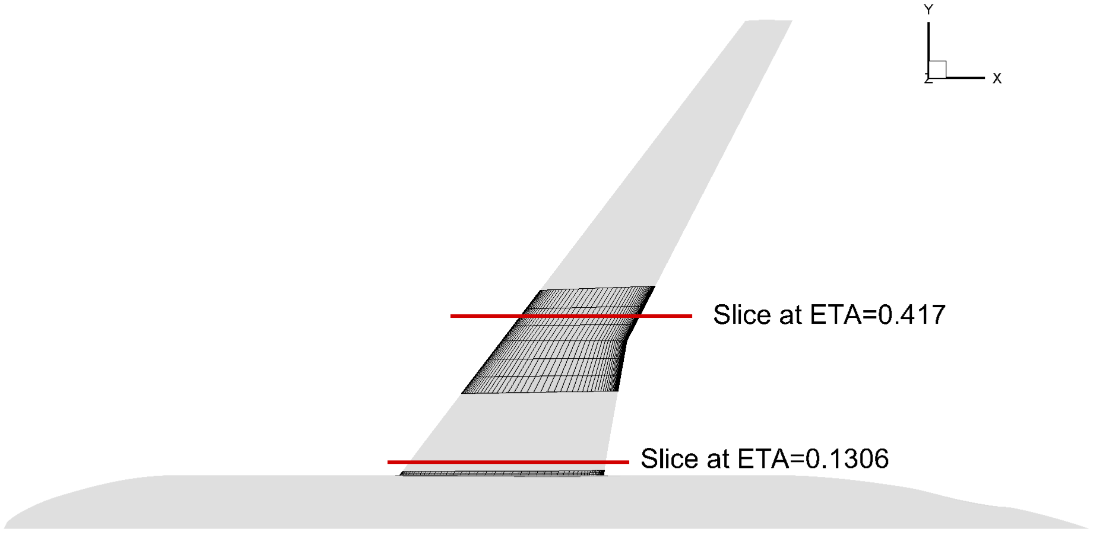


Figure 5.2 Position of the slide used for data comparison

Table 5.1: Parameters used for the simulation on the CRM Wing-Body configuration. First-order accuracy.

Parameter	Value
Mach Number	0.85
Reynolds Number	5’000’000
Angle-of-Attack	2.5
Turbulence Model	Spalart-Allmaras
Solver	LU-SGS (implicit)
Convective fluxes	Roe first order

Table 5.2: Comparison between the overset and single grid approach. Grids are presented in Figure 5.1 and flow solver parameters are described in table 5.1.

Approach	CL	CD
Overset grid	4.4874781e-01	8.425674e-02
Single grid	4.4874782e-01	8.425675e-02

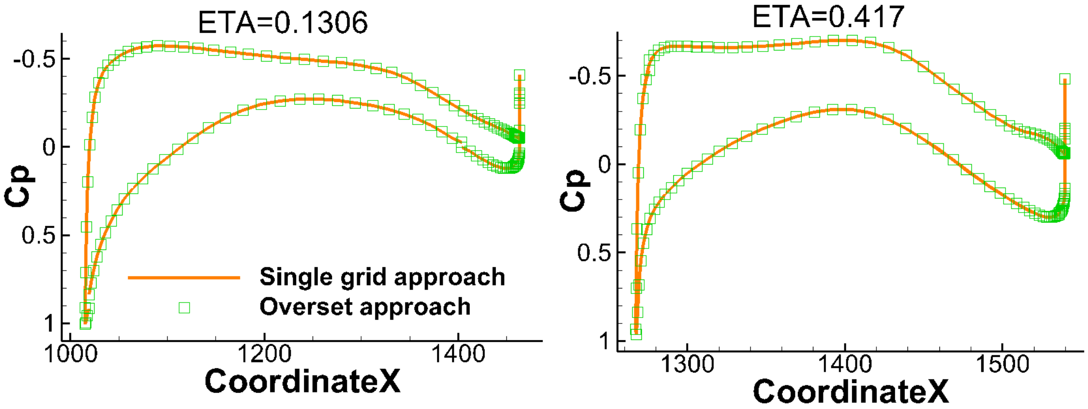


Figure 5.3 Pressure distribution using a coincident collar grid on the CRM Wing-Body

5.1.2 Second Order Simulation

On the same global overset assembly, a second order flow simulation is tested. The flow solver is set up using the parameters described in table 5.3. Again, the results obtained using the overset approach is equal to the single grid approach within expected accuracy : the difference between the two results is smaller than the simulation convergence level. Table 5.4 shows the results. Again, the simulation is converged with an order of precision smaller than 1.0e-4.5 on the density. Figures 5.5 and 5.6 shows the pressure distributions and the convergence of the density and lift coefficient respectively.

Those canonic test cases verified that the implementation and the use of the overset approach

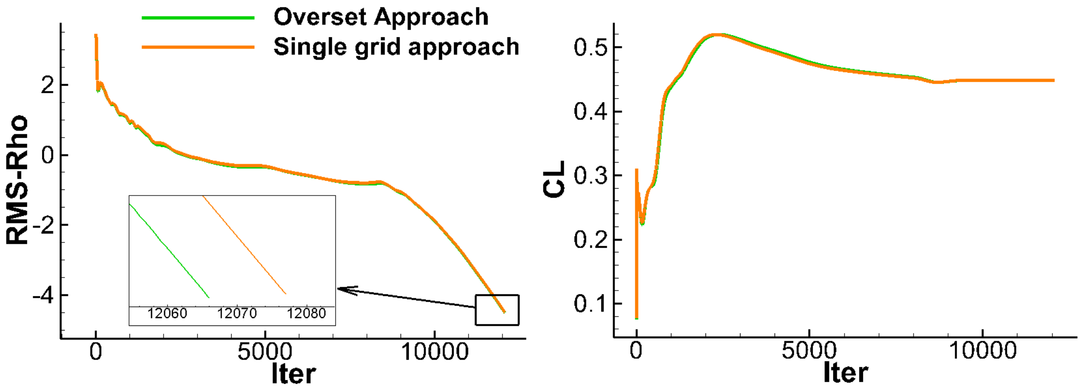


Figure 5.4 Density and lift convergence using a coincident collar grid on the CRM Wing-Body

Table 5.3: Parameters used for the simulation on the CRM Wing-Body configuration. Second-order accuracy.

Parameter	Value
Mach Number	0.85
Reynolds Number	5'000'000
Angle-of-Attack	2.5
Turbulence Model	Spalart-Allmaras
Solver	LU-SGS (implicit)
Convective fluxes	Roe second order

Table 5.4: Comparison between the overset and single grid approach. Grids are presented in Figure 5.1 and flow solver parameters are described in table 5.3.

Approach	CL	CD
Overset grid	5.365231e-01	2.8691996e-02
Single grid	5.365232e-01	2.8691998e-02

does not modify the results for a coincident collar grid. The global grid of the overset grids used here is identical, in terms of computed domain, to the grid used in the single grid approach. Hence, there is no error introduced by the overset communication and it is demonstrated that the overset process introduces no error in the computation. This is a first but important verification that gives confidence in the overall implementation for three-dimensional applications.

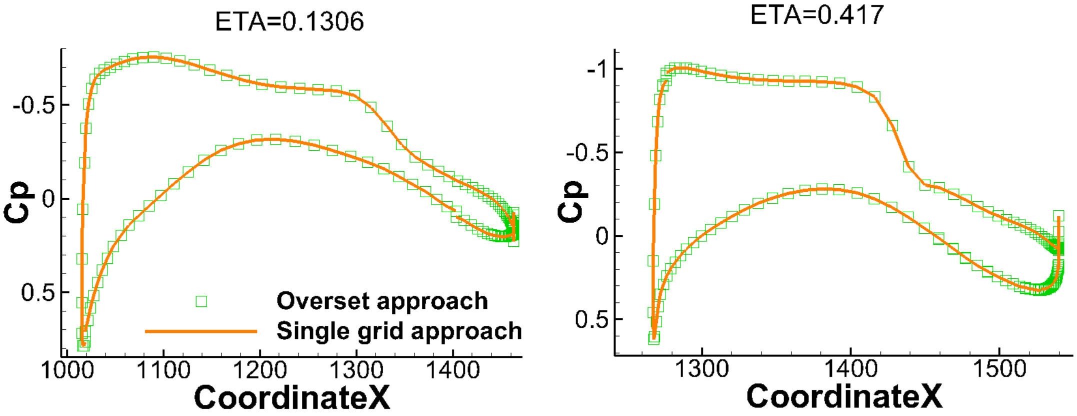


Figure 5.5 Density and lift convergence using a coincident collar grid on CRM Wing-Body

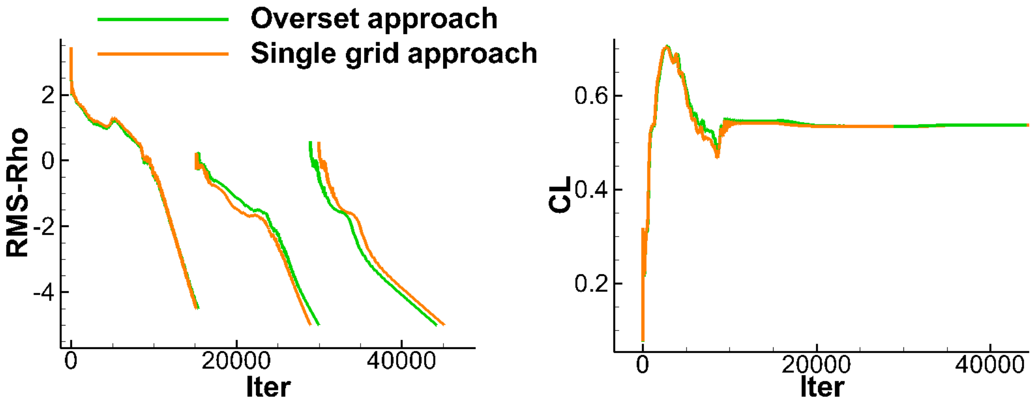


Figure 5.6 Density and lift coincident collar grid on CRM Wing-Body. The different curves represent different restarts of the simulation to finally converged without a limiter.

5.2 Industrial Application Of The Collar Grid Approach

Here, two non-coincident collar grids are applied to solve the wing-body intersection region. Such application is more frequent and therefore representative of industrial applications. Figure 5.7 presents the overset grid.

In the previous section, the grids obtained from the overset and single grid approach were identical. Therefore, identical results were expected. The application presented in this section, on the other hand, uses collar grid with different discretizations than the underlying grid. The final overset grid is now different than the single grid approach and therefore the same results cannot be expected. Moreover, the overset communication through an interpolation will introduce a numerical error. The flow solver is set up using the parameters listed in Table 5.5. Figure 5.8 present the pressure distribution obtained on the overset grid and

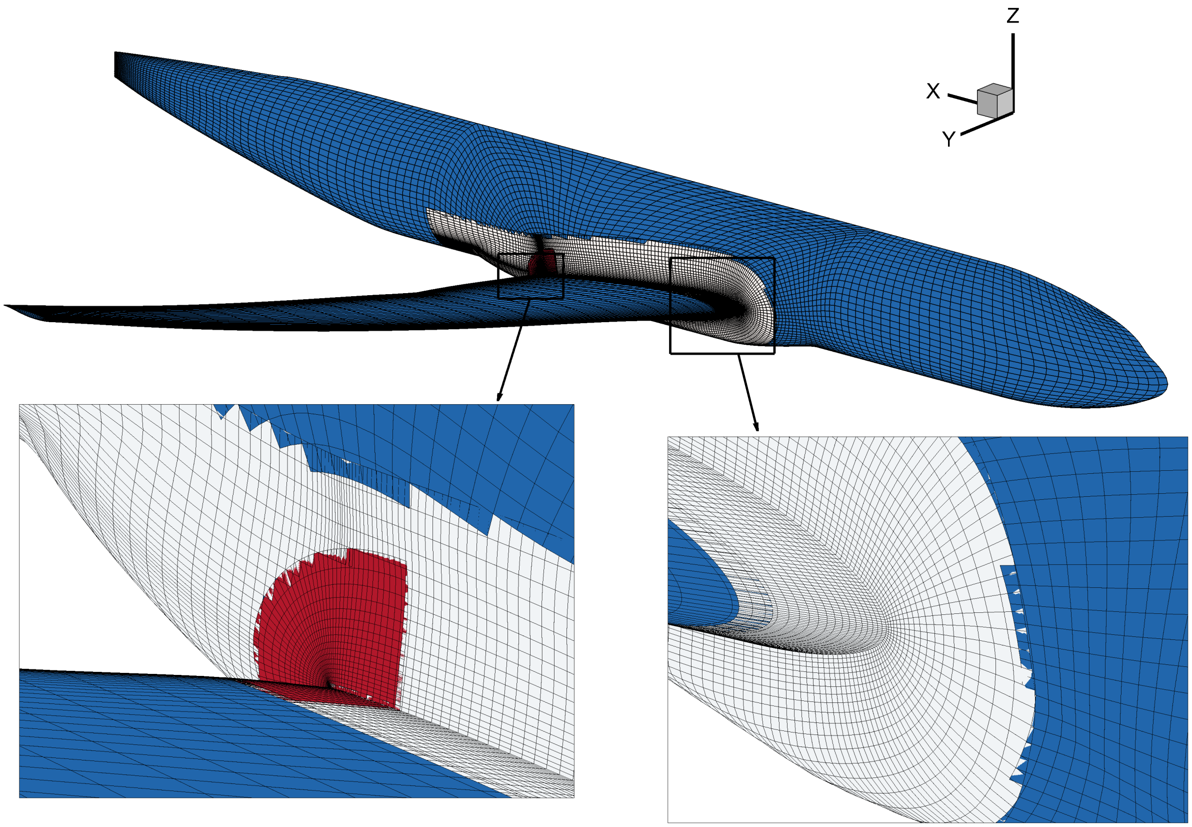


Figure 5.7 Industrial collar grid application on CRM Wing-Body configuration

the pressure distribution compared to the single grid approach. The left part of the figure illustrates where the pressure cut is done on the wing.

Table 5.5: Parameters used for the simulation on the CRM Wing-Body configuration. Second-order accuracy.

Parameter	Value
Mach Number	0.85
Reynolds Number	5’000’000
Angle-of-Attack	2.5
Turbulence Model	Spalart-Allmaras
Solver	LU-SGS (implicit)
Convective fluxes	Roe second order

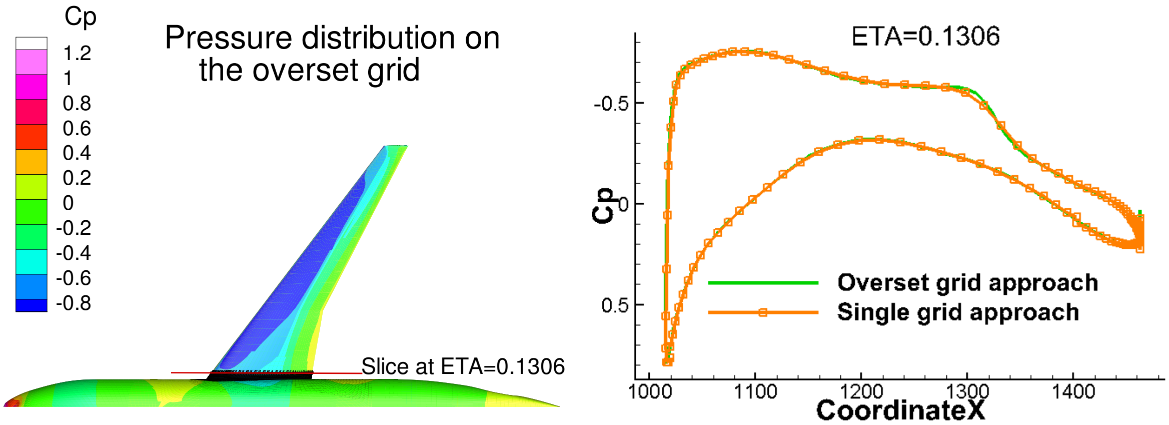


Figure 5.8 Pressure distribution : comparison between the overset and single grid approach. Grids are presented in Figure 5.7 and flow solver parameters are described in table 5.5.

The results show a good robustness of the method to use the collar grid approach. The intersection region was improved by the collar grid. As presented in Figure 5.9, the separation bubble is not captured with the single grid approach. The collar grid was tailored for this region and it can therefore be of better quality. The collar grid is finer than the underlying grid.

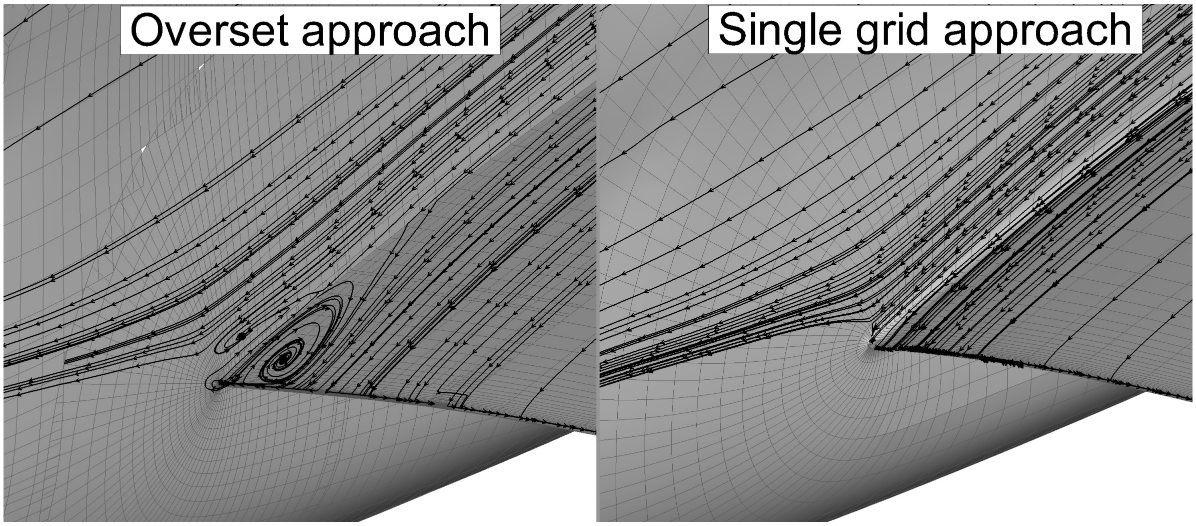


Figure 5.9 Skin friction lines showing the separation bubble in the Body-Wing intersection of CRM Wing-Body configuration when using the collar grid method. Mach=0.85, Re=5M, second order accuracy.

5.3 Full Industrial Overset Application On CRM Wing-Body-Tail Configuration

The grids used here was generated by Boeing and made available during the AIAA 4th drag prediction workshop. The geometry studied at the DPW4 was the NASA Common Research Model transonic Wing-Body-Tail. A set of 17 overset grids were generated to represent the computational domain.

For the grid connectivity step, interpolation is set at the cell center in CHAMPS. The grids only allow one layer of interpolated grid cells. As explained in a previous section, more than one layer of interpolated grid cells are necessary to properly reconstruct the limited gradients. OVERFLOW, for which the grids were initially designed, uses a vertex-center representation and therefore benefits from an additional layer [58].

5.3.1 Wing-Body And Wing-Body-Tail Configuration

One advantage of the overset approach is to easily add or remove components of a geometry. In this line of thought, the first simulation is done here using only the grids to represent the Wing-Body part of the configuration. Figures 5.10 and 5.11 present the grids before and after the creation of an overset connectivity respectively. The overset preprocessor allows adding a tail to the initial configuration as presented in Figure 5.12. This shows the flexibility provided by the overset approach.

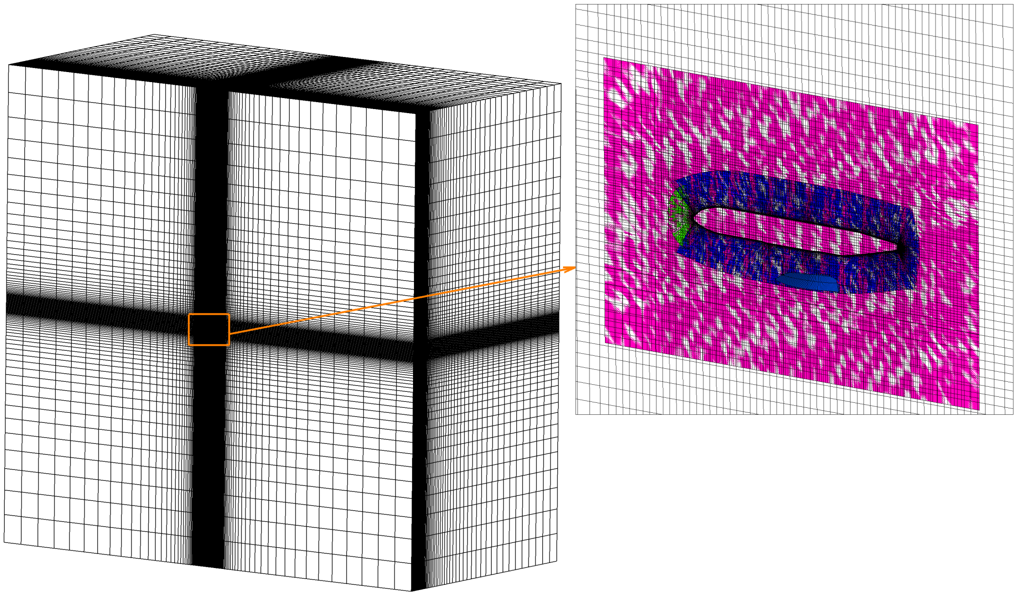


Figure 5.10 NASA Common Research Model, Wing-Body configuration. Overset grids generated by Boeing for the 4th Drag Prediction Workshop.

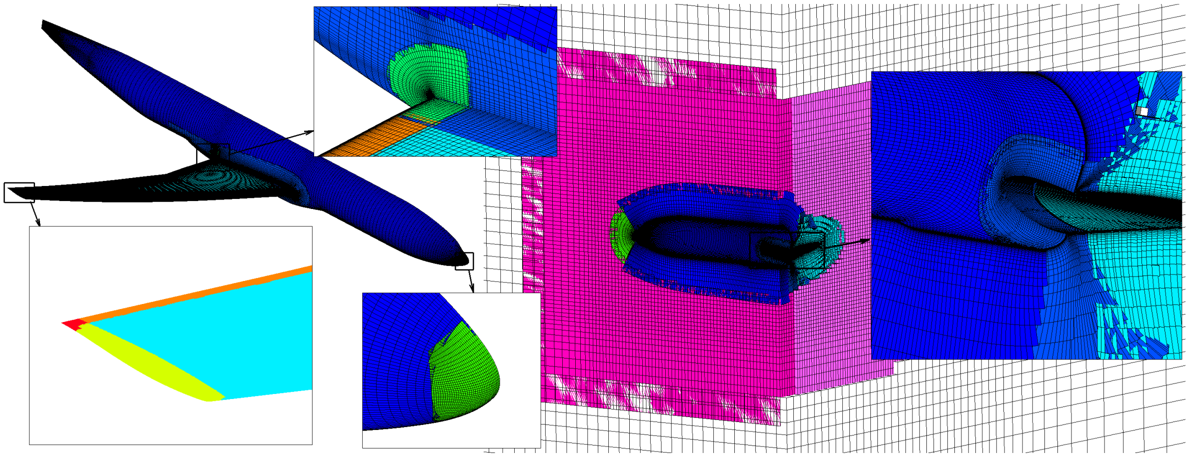


Figure 5.11 NASA Common Research Model, Wing-Body configuration. Overset connectivity generated with CHAMPS

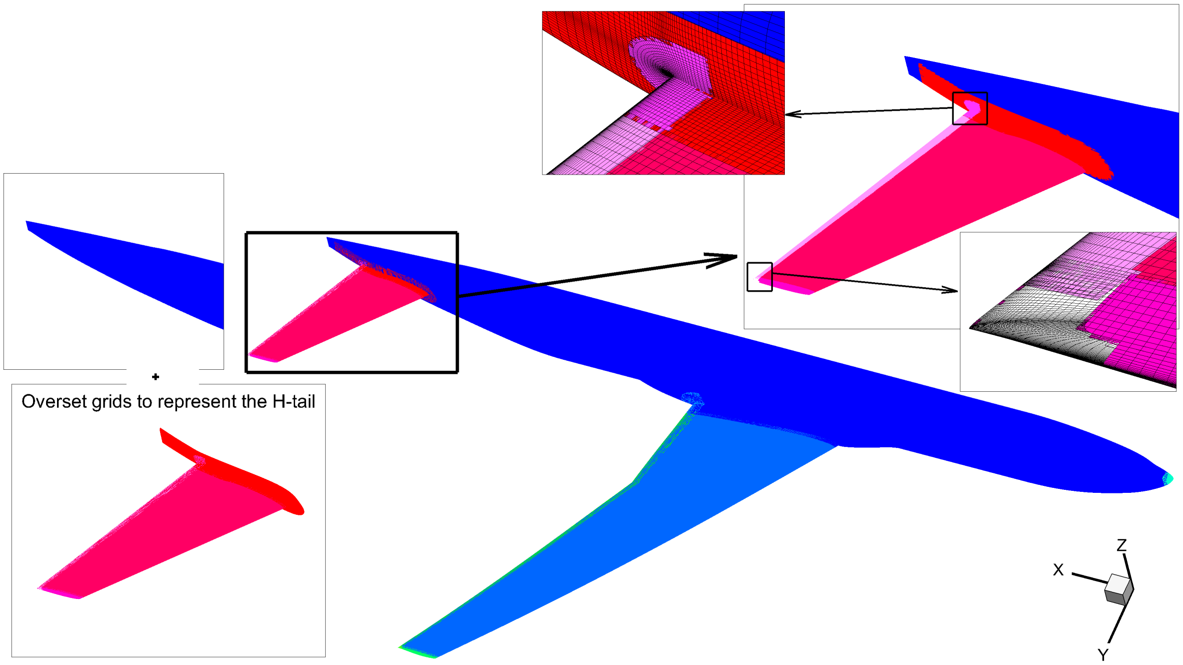


Figure 5.12 NASA Common Research Model, Wing-Body-Tail configuration. Overset connectivity generated with CHAMPS

Flow Simulation On The Wing-Body Configuration

The flow solver is set up using the parameters presented in table 5.6. Figure 5.13 presents the pressure distribution on the solid surface. It shows a smooth pressure distribution on the aircraft surface. The region surrounding the collar grids are also smooth and does not present any disturbtion. Figure 5.14 presents the convergence of the density and lift coefficient. The convergence is smooth and has adequate performances. The spikes especially apparent in the density convergence are normal and expected, and represent the moment when the angle of attack is updated : to converge toward the proper lift coefficient, CHAMPS update the angle of attack every few iterations. The pressure distributions presented in Figure 5.15 are smooth and continue throughout the surface which is composed of many collar grids. As better presented in Figure 5.16, surface segments from different grid and collar grid composed the global surface. The surface used to compute pressure distribution are therefore also from different grids and collar grids. This shows a proper implementation and effectiveness of the different methods implemented in the overset preprocessor.

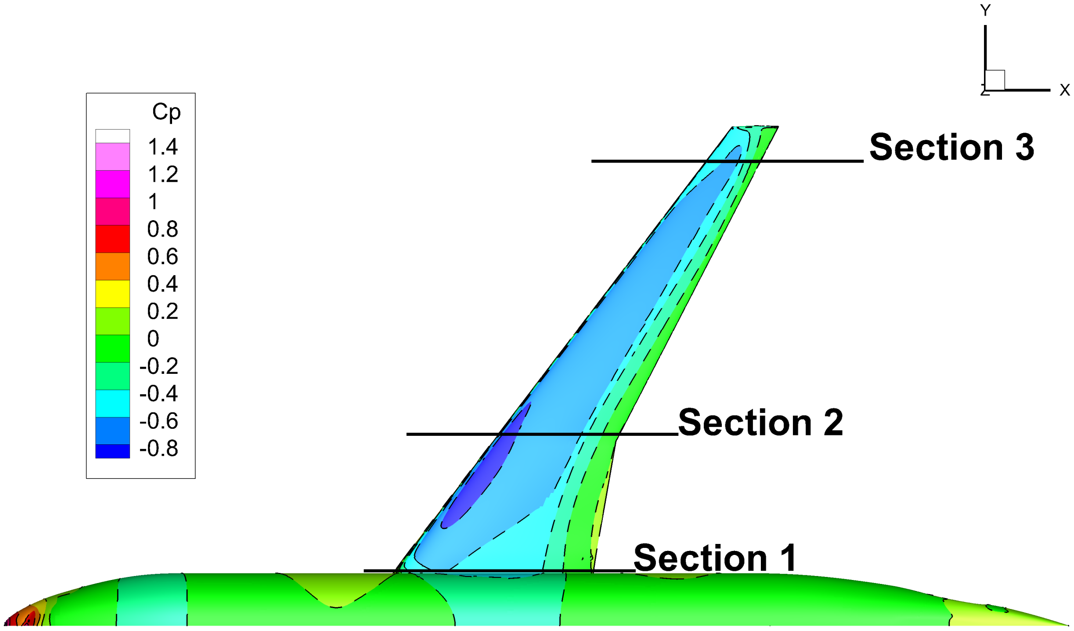


Figure 5.13 Pressure distribution on the CRM Wing-Body configuration. Flow solver parameters are presented in table 5.6

Table 5.6: Parameters used for the simulation on the CRM Wing-Body configuration using full overset approach. First-order accuracy.

Parameter	Value
Mach Number	0.85
Reynolds Number	5’000’000
Lift coefficient	0.5
Turbulence Model	Spalart-Allmaras
Solver	LU-SGS (implicit)
Convective fluxes	Roe first order

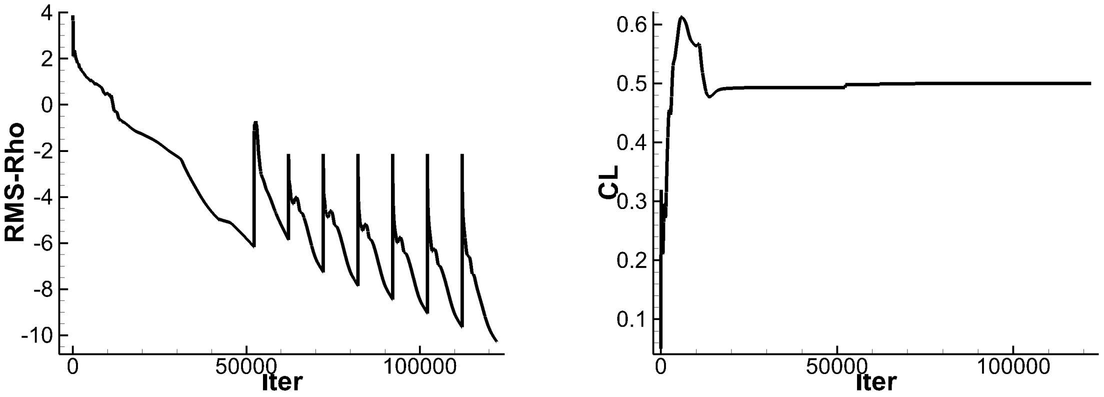


Figure 5.14 Convergence on the CRM Wing-Body using the full overset approach. On the left is the convergence of density and on the right the lift coefficient. Flow solver parameters are presented in table 5.6

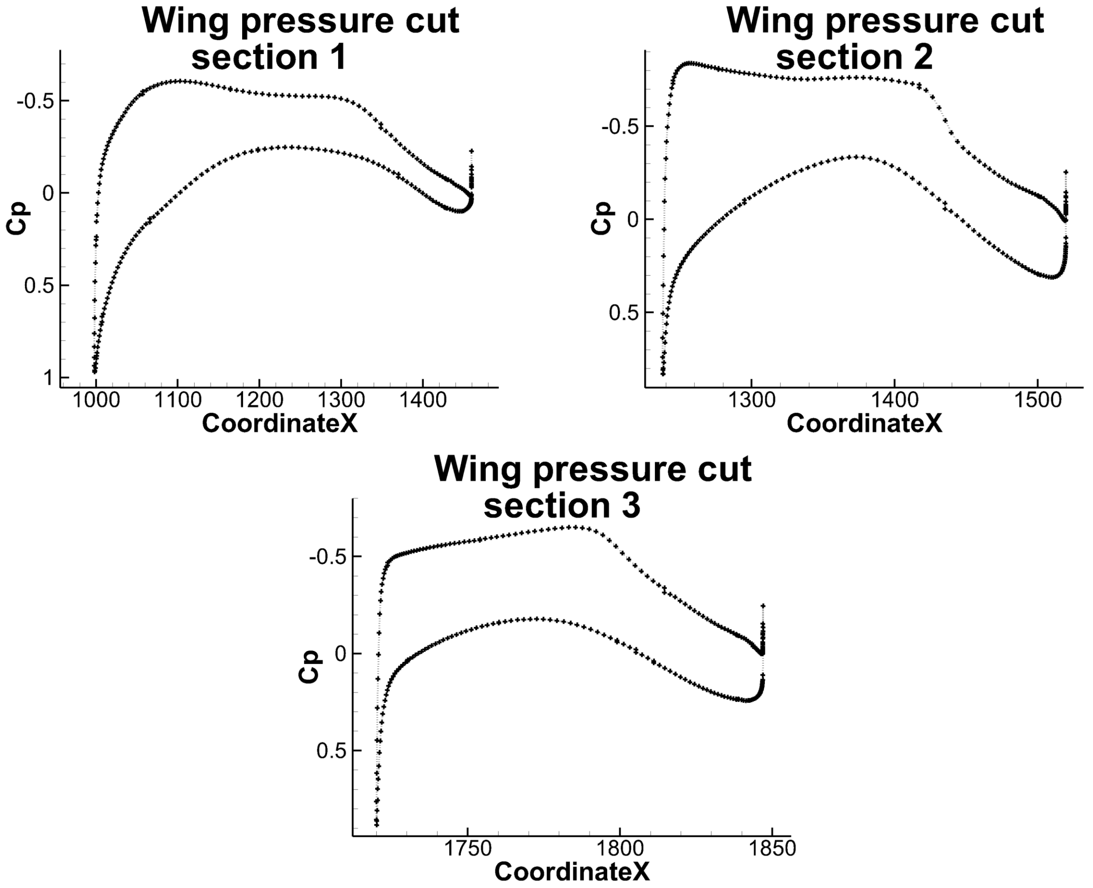


Figure 5.15 Pressure distribution for three different cut sections along the wing of the CRM Wing-Body configuration. Flow solver parameters are presented in table 5.6 and sections position illustrated in Figure 5.13

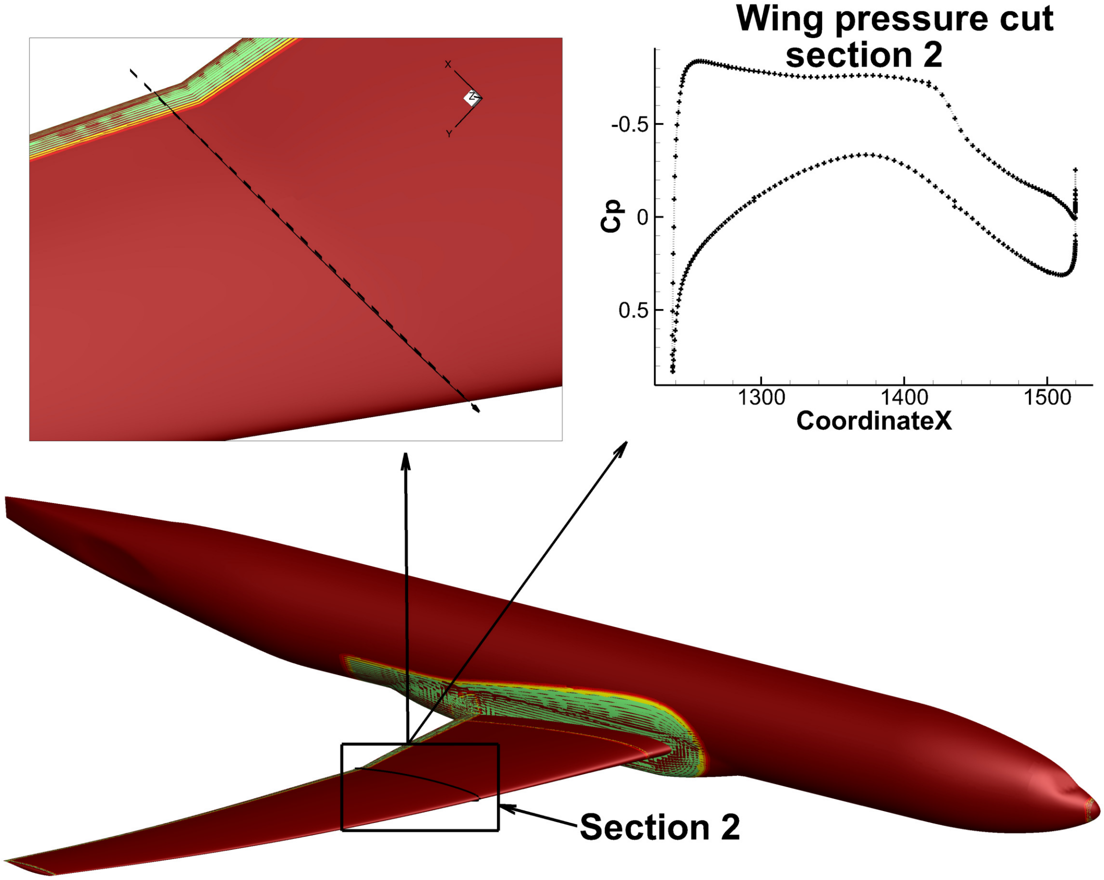


Figure 5.16 Zoom on the section of pressure cut. Green regions represent the collar grids. Flow solver parameters are presented in table 5.6

CHAPTER 6 CONCLUSION

6.1 Synthesis Of Work

This thesis presented the advances of the overset method developed in the in-house CFD solver CHAMPS. The technology developed is able to generate good results on industrial applications which was the initial objective of the project. Treating complex geometries requires robust methods to allow a proper communication of the flow information. The literature review helped design a preprocessor capable of creating a connectivity on such cases.

The methods used to allow solid intersections in overset assembly were identified using the literature :

- The X-ray hole cutting method is used. From the author's experience, the X-ray method can be valid with the proper inputs (i.e., enough rays and orientation(s)). The limit of the method is rather imposed by the resources (memory and time) available to the user.

- The flow solver of CHAMPS was reviewed in order to properly consider the different numerical stencils in the definition of the overset connectivity. Higher order methods require extended stencils and therefore additional grid cells are necessary at the overset interfaces. The scope of this thesis is not to generate overset grids but to develop the technology to execute simulations onto them. Therefore, the limitations come with using other laboratory's grids which are designed for other solvers.

- The overset connectivity close to the wall is more critical because of the high gradients in the region. To better define those regions, a special type of grids called *collar grids* are used to patch the regions.

- Initial verification using collar grids showed the necessity for treatments close to the wall. Overset connectivity close to the wall is sensible to the different discretizations of the geometry. Investigation of a correction method found in the literature corrected the observed problem and thus allowed to generate good results around solid intersections.

- Collar grids generate surface duplication which distorts the results of surface integration. Before computing the aerodynamic coefficients, an algorithm weights the overlapping regions of the geometry to consider the duplication.

The methods and their details are presented in chapter 3. Simple canonic tests are presented to verify the proper implementation of the algorithms. The limits and errors are, when possible, evaluated. Finally, the overall overset algorithm was verified with a grid convergence study of a circular cylinder. The effect of the overset method is isolated using a collar grid with the same discretization as the background grid. The hypothesis is made that an *error* is introduced by translating the collar grid. Results showed that the weighting method used with the collar grids was, as predicted, not perfect. The error can, however, be minimized using finer grids.

The application of the technology on industrial applications generated results in agreement with the single grid approach and the literature. Both two- and three-dimensional capabilities of the preprocessor were verified. The final application of the overset method is the NASA Commom Research Model geometry, which is a test case developed by NASA. Overset grids of the CRM generated at Boeing for the 4th drag prediction workshop were used to verify the capabilities of the overset technology in CHAMPS. The results show smooth pressure distribution around the aircraft configuration. Limitation imposed by those grids, however, only allowed to verify the first-order discretization stencil.

6.2 Future Work

6.2.1 Acceleration Of The X-ray Hole Cutting Method

While the applications presented in this thesis are of an industrial level, application with considerably larger grids are frequent. To eventually address those larger cases, the X-ray hole cutting algorithms needs to be reviewed to improve its scalability. Because of the large size of the data buffer communicated between memory nodes in the algorithm, other tools like external C libraries could help optimize the process. Furthermore, an auxiliary method could help optimize the number of rays used in the method. Currently, the distribution is constant through the components and thus sometime fairly dense distribution is needed in order to accommodate a single isolated zone. The distribution could therefore be defined as variable within a single component.

6.2.2 Conservative Interpolation Method

The overset grids used for the simulations on an airfoil section of the CRM-HL involve interpolation in high gradients region which is known to have a negative effect on the solution. A grid convergence study shows a slightly different residual convergence when using the overset approach in CHAMPS. The interpolation method used in CHAMPS is non-conservative and could therefore generate mass in the domain and could thus explaines these differences. The results seem, however, to converge toward the literature's results when refining the grids.

6.2.3 Minimize Impact Of Lack Of Overlap

The main difficulty encountered in the verification of the method was the lack of overlap which prevents proper connectivity definition. A solution to minimize this problem is initiated : it consists of locating an alternative donor when no valid one is initially found. The only valid solution, however, is to eventually generate custom grids for CHAMPS or find ones compatible. As mentioned in chapter 5, the cell-centred discretization used in CHAMPS takes out an entire layer of grid cell available for the overset connectivity. From an idea found in the literature [58], advances to the algorithm were therefore initiated in CHAMPS to help minimize the impact of this lack of overlap. The approach consists of enlarging the area of research in the donor search process.

REFERENCES

[1] "Nasa image and video library," https://images.nasa.gov/, (Accessed on 02/15/2022).

[2] A. Desmarais, "Extension of a 2D multiblock structured overset suite to a fully 3D multiblock unstructured flow solver," p. 109.

[3] T. Schwarz, "Development of a wall treatment for navier-stokes computations using the overset-grid technique," 09 2000.

[4] J. Guay, "Extension of the Overset Grid Preprocessor for surface conforming meshes," p. 114.

[5] A. PIGEON, "DÉveloppement d'une mÉthode d'accÉlÉration par grilles virtuelles rÉcursives pour l'assemblage de maillages chimÈres," p. 97. [Online]. Available: https://publications.polymtl.ca/1692/

[6] D. Mavriplis, "Grid resolution study of a drag prediction workshop configuration using the NSU3d unstructured mesh solver," in *23rd AIAA Applied Aerodynamics Conference.* American Institute of Aeronautics and Astronautics, Jun. 2005. [Online]. Available: https://doi.org/10.2514/6.2005-4729

[7] H. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics*, 2nd ed. Philadelphia, PA: Prentice Hall, Feb. 2007.

[8] W. Chan *et al.*, "Best practices in overset grid generation," in *32nd AIAA Fluid Dynamics Conference and Exhibit.* American Institute of Aeronautics and Astronautics, Jun. 2002. [Online]. Available: https://doi.org/10.2514/6.2002-3191

[9] M. Parenteau *et al.*, "Development of parallel CFD applications with the chapel programming language," in *AIAA Scitech 2021 Forum.* American Institute of Aeronautics and Astronautics, Jan. 2021. [Online]. Available: https://doi.org/10.2514/6.2021-0749

[10] S. Bourgault-Côté, "Simulation du givrage sur ailes en flèche par méthodes rans/eulérienne quasi stationnaires," Ph.D. dissertation, École Polytechnique de Montréal, 2015.

[11] Z. Xu *et al.*, "Towards a scalable hierarchical high-order CFD solver," in *AIAA Scitech 2021 Forum.* American Institute of Aeronautics and Astronautics, Jan. 2021. [Online]. Available: https://doi.org/10.2514/6.2021-0494

[12] "Chapel: Productive parallel programming," https://chapel-lang.org/, (Accessed on 02/10/2022).

[13] J. BENEK, J. STEGER, and F. DOUGHERTY, "A flexible grid embedding technique with application to the euler equations," in *6th Computational Fluid Dynamics Conference Danvers*. American Institute of Aeronautics and Astronautics, Jul. 1983. [Online]. Available: https://doi.org/10.2514/6.1983-1944

[14] T. Dimas, "2d flow simulation with chimera grids," 2021.

[15] J. de Laborderie *et al.*, "Numerical analysis of a high-order unstructured overset grid method for compressible les of turbomachinery," *J. Comput. Phys.*, vol. 363, pp. 371–398, 2018.

[16] D. D. Chandar, "On overset interpolation strategies and conservation on unstructured grids in openfoam," *Computer Physics Communications*, vol. 239, pp. 72–83, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465519300153

[17] S. Rogers *et al.*, "Advances in overset cfd processes applied to subsonic high-lift aircraft," in *18th Applied Aerodynamics Conference*, 2000, p. 4216.

[18] W. Chan *et al.*, "Advances towards automatic surface domain decomposition and grid generation for overset grids," in *13th Computational Fluid Dynamics Conference*, 1997, p. 1979.

[19] Z. Wang, V. Parthasarathy, and N. Hariharan, "A fully automated chimera methodology for multiple moving body problems," in *36th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 1998. [Online]. Available: https://doi.org/10.2514/6.1998-217

[20] J. Benek *et al.*, "Chimera. a grid-embedding technique," ARNOLD ENGINEERING DEVELOPMENT CENTER ARNOLD AFB TN, Tech. Rep., 1986.

[21] W. M. Chan, "Overset grid technology development at NASA ames research center," *Computers & Fluids*, vol. 38, no. 3, pp. 496–503, Mar. 2009. [Online]. Available: https://doi.org/10.1016/j.compfluid.2008.06.009

[22] J. G. Coder *et al.*, "Contributions to the sixth drag prediction workshop using structured, overset grid methods," *Journal of Aircraft*, vol. 55, no. 4, pp. 1406–1419, Jul. 2018. [Online]. Available: https://doi.org/10.2514/1.c034486

[23] T. Deloze, “Couplage fluide-solide appliqué à l’étude de mouvement d’une sphère libre dans un tube vertical,” Theses, Université de Strasbourg, May 2011. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00687119

[24] “Development of a large scale chimera grid system for the space shuttle launch vehicle,” in *31st Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Jan. 1993. [Online]. Available: https://doi.org/10.2514/6.1993-533

[25] N. Kim and W. Chan, “Automation of hole-cutting for overset grids using the x-rays approach,” in *20th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 2011. [Online]. Available: https://doi.org/10.2514/6.2011-3052

[26] X. Chang *et al.*, “A parallel implicit hole-cutting method based on background mesh for unstructured chimera grid,” *Computers & Fluids*, vol. 198, p. 104403, Feb. 2020. [Online]. Available: https://doi.org/10.1016/j.compfluid.2019.104403

[27] Y. Lee and J. Baeder, “Implicit hole cutting - a new approach to overset grid connectivity,” in *16th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 2003. [Online]. Available: https://doi.org/10.2514/6.2003-4128

[28] D. D. Chandar, J. Sitaraman, and D. J. Mavriplis, “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, Jul. 2013. [Online]. Available: https://doi.org/10.1080/10618562.2013.829915

[29] A. Sharma *et al.*, “Overset meshes for incompressible flows: On preserving accuracy of underlying discretizations,” *Journal of Computational Physics*, vol. 428, p. 109987, Mar. 2021. [Online]. Available: https://doi.org/10.1016/j.jcp.2020.109987

[30] S. E. Rogers, N. E. Suhs, and W. E. Dietz, “PEGASUS 5: An automated preprocessor for overset-grid computational fluid dynamics,” *AIAA Journal*, vol. 41, no. 6, pp. 1037–1045, Jun. 2003. [Online]. Available: https://doi.org/10.2514/2.2070

[31] I.-T. Chiu and R. Meakin, “On automating domain connectivity for overset grids,” in *33rd Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 1995. [Online]. Available: https://doi.org/10.2514/6.1995-854

[32] R. Meakin, “Object x-rays for cutting holes in composite overset structured grids,” in *15th AIAA Computational Fluid Dynamics Conference*. American

Institute of Aeronautics and Astronautics, Jun. 2001. [Online]. Available: https://doi.org/10.2514/6.2001-2537

[33] K. W. Cho, J. H. Kwon, and S. Lee, “Development of a fully systemized chimera methodology for steady/unsteady problems,” *Journal of Aircraft*, vol. 36, no. 6, pp. 973–980, Nov. 1999. [Online]. Available: https://doi.org/10.2514/2.2538

[34] R. Noack *et al.*, “Suggar++: An improved general overset grid assembly capability,” 06 2009.

[35] S. A. P. William M. Chan and N. Kim, “Advances in domain connectivity for overset grids using the x-rays approach,” Jul. 2012. [Online]. Available: https://ntrs.nasa.gov/api/citations/20130001601/downloads/20130001601.pdf

[36] K.-H. Kao and M.-S. Liou, “Advance in overset grid schemes - from chimera to DRAGON grids,” *AIAA Journal*, vol. 33, no. 10, pp. 1809–1815, Oct. 1995. [Online]. Available: https://doi.org/10.2514/3.12921

[37] B. Hubbard and H.-C. Chen, “A chimera scheme for incompressible viscous flows with application to submarine hydrodynamics,” in *Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 1994. [Online]. Available: https://doi.org/10.2514/6.1994-2210

[38] J. Wright and W. Shyy, “A pressure-based composite grid method for the navier-stokes equations,” *Journal of Computational Physics*, vol. 107, no. 2, pp. 225–238, Aug. 1993. [Online]. Available: https://doi.org/10.1006/jcph.1993.1139

[39] S. PARKS *et al.*, “Collar grids for intersecting geometric components within the chimera overlapped grid scheme,” in *10th Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 1991. [Online]. Available: https://doi.org/10.2514/6.1991-1587

[40] D. Hue *et al.*, “Validation of a near-body and off-body grid partitioning methodology for aircraft aerodynamic performance prediction,” *Computers & Fluids*, vol. 117, pp. 196–211, Aug. 2015. [Online]. Available: https://doi.org/10.1016/j.compfluid.2015.05.021

[41] T. Schwarz, F. Spiering, and N. Kroll, “Grid coupling by means of chimera interpolation techniques,” 06 2010.

[42] W. Chan and P. Buning, “Zipper grids for force and moment computation on overset grids,” in *12th Computational Fluid Dynamics Conference*. American

Institute of Aeronautics and Astronautics, Jun. 1995. [Online]. Available: https://doi.org/10.2514/6.1995-1681

[43] W. M. Chan and P. Buning, "User's manual for fomoco utilities-force and moment computation tools for overset grids," 1996.

[44] D. Boger and J. Dreyer, "Prediction of hydrodynamic forces and moments for underwater vehicles using overset grids," in *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Jan. 2006. [Online]. Available: https://doi.org/10.2514/6.2006-1148

[45] L. Wigton, "Polymixsur – boeing's replacement for mixsur," Oct. 2004.

[46] F. Martínez *et al.*, "A simple algorithm for boolean operations on polygons," *Advances in Engineering Software*, vol. 64, pp. 11–19, Oct. 2013. [Online]. Available: https://doi.org/10.1016/j.advengsoft.2013.04.004

[47] K. Yamaguchi *et al.*, "Octree-related data structures and algorithms," *IEEE Computer Graphics and Applications*, vol. 4, no. 1, pp. 53–59, 1984. [Online]. Available: https://doi.org/10.1109/mcg.1984.275901

[48] William H. Press *et al.*, *Numerical recipes 3rd edition*, 3rd ed. Cambridge, England: Cambridge University Press, Sep. 2007.

[49] "Cgns standard interface data structures - conventions," https://cgns.github.io/CGNS_docs_current/sids/conv.html#unst_3d, (Accessed on 05/15/2022).

[50] C. Ericson, *Real-Time Collision Detection*, 1st ed. CRC Press, Dec. 2004. [Online]. Available: https://www.taylorfrancis.com/books/9780080474144

[51] J. Vassberg and A. Jameson, "In pursuit of grid convergence for two-dimensional euler solutions," *Journal of Aircraft*, vol. 47, no. 4, pp. 1152 – 66, 2010, two-dimensional Euler solutions;airfoil geometry;NACA0012 equation;NACA0012 airfoil;trailing edge;Karman-Trefftz conformal transformation;standard O-mesh;far-field boundary;quadrilateral cell;nonlifting solution;flow solvers;drag;angles-of-attack;transonic flow conditions;FL082 solutions;asymptotic grid convergence;OVERFLOW;CFL3D;pitching moment;aerodynamic coefficients;computational fluid dynamics methods;. [Online]. Available: http://dx.doi.org/10.2514/1.46737

[52] K. Morinishi, "A finite difference solution of the Euler equations on non-body-fitted Cartesian grids," *Computers & Fluids*, vol. 21, no. 3, pp. 331 – 344, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/004579309290042T

[53] C. Rumsey, “Verif/2dmea: 2d multielement airfoil verification case - intro page,” Jan. 2021. [Online]. [Online]. Available: https://turbmodels.larc.nasa.gov/multielementverif.html

[54] R. Kannan and Z. Wang, “Overset adaptive cartesian/prism grid method for stationary and moving-boundary flow problems,” *AIAA journal*, vol. 45, no. 7, pp. 1774–1779, 2007.

[55] “Dpw-5.” [Online]. Available: https://aiaa-dpw.larc.nasa.gov/Workshop5/workshop5.html

[56] “Dpw-4.” [Online]. Available: https://aiaa-dpw.larc.nasa.gov/Workshop4/workshop4.html

[57] “NASA Common Research Model | providing data worldwide.” [Online]. Available: https://commonresearchmodel.larc.nasa.gov/

[58] A. Sclafani *et al.*, “Drag prediction for the NASA CRM wing-body-tail using CFL3d and OVERFLOW on an overset mesh,” in *28th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 2010. [Online]. Available: https://doi.org/10.2514/6.2010-4219