THÈSE

Pour l'obtention du grade de

Docteur de l'Université de Strasbourg

❧

Laboratoire ICube - Département de Mécanique
École doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

Discipline : Mécanique des Fluides

Multi-step icing modelling in the NSMB solver

Présentée par : ALI AL-KEBSI

Sous la direction de :

Directeur de thèse : YANNICK HOARAU, Professeur des Universités

Co-Directeur de thèse : ROBERT MOSÉ, Professeur des Universités

Composition du jury:

Rapporteur externe : Dr Heloïse Beaugendre, Université de Bordeaux

Rapporteur externe : Pr Marcello Righi, Zhaw school of engineering

Président du jury : Pr Philippe Helluy, Université de Strasboug

Examinateur : Jean-Philippe Pineau, Flight Avionics - Engeineering & Development, Thales

Date de soutenance : 31 Janvier 2020

Remerciements

Going through PhD has been a truly life-changing experience. A period where support, motivation and guidance is primordial.

I would like to thank my supervisor, HOARAU Yannick,for his guidance through each stage of the process, for his patience, motivation, and immense knowledge. I could not have imagined having a better advisor and mentor for my Ph.D study. I would also like to thank my co-supervisor, MOSE Robert for his insightful comments and encouragement.

My sincere thanks also goes to my teammates, PONCE Anthony for his technical help, LEAUTAUD Vincent for listening and engaging in profound discussions related to problems encountered throughout the development of different parts of the code. Not forgetting others, Dorian Pena for supervising my first steps within NSMB and ice simulation, MOTURI Viswa Maitreyi for her encouragement, Chao Kun Huang, BERNARD Chrystelle , DURRENBERGER Daniel, MAROUF Abderahman and Mahdi Rezaei.

I would like to thank my family : my father who i couldn't meet during the last five year, my mother who went through many hardships to allow me to become what i am, my brothers and sister for supporting me spiritually throughout writing this thesis. And lastly my dearest wife Muna, the patient, supportive, enduring mother of my two kids, who stood beside, who i gave my all to reunite with, it is my pleasure to be by your side, until the day this becomes a memory of who i was. It was worth it !

Table des matières

Table des figures

Liste des tableaux

Abstract

In-Flight ice accretion is modeled using the Level-Set novel approach. The Level-Set method alleviates concerns of complex remeshing in concurrent ice models used by the industry. It allows the simulation of very complex ice forms with all difficult topographies, namely coalescing interfaces or severely separating ones. The Level-Set method allows a straightforward multi-step simulation. The solid boundary is treated in different methods, implicitly or explicitly. The solid part, wing and ice, is defined via a characteristic level-set function in the former case, and the interface is defined as a list of Lagrangian points in the latter Therefore, the air flow boundary conditions are applied through the turbulence modeled at the wall. On the other hand, droplet transport is modeled via an IBM-LS Eulerian approach, using a discrete formulation of the interface.

Résumé

La tâche principale était d'améliorer et continuer à développer un code pour la modélisation de givrage au sien de NSMB. NSMB « Navier-Stokes Multi-Block solveur » est un solveur développé depuis 1990 par des institutes de recherche européen, un solveur parallélisé qui porte sur plusieurs modèles physique et chimique, plusieurs modèles de turbulence, et contiens des modèles de haut niveau comme ALE, maillage chimère,...etc.

Le modèle connu et utilisé souvent pour la modélisation de givrage compose un seul pas de givrage. On commence par la résolution de l'écoulement d'air. Puis, on injecte des gouttelettes d'eau et on trace leurs trajets. Ceux qui impactent l'aile, soit se givre en impacte, soit s'écoule à l'arrière et givre en aval. Une balance Thermodynamique est appliquée au bord de l'aile pour bien capturer la quantité d'eau givrée, la quantité qui peuvent s'écouler, et leurs températures. Après, on calcule la hauteur de glace dans chaque point au bord de l'aile. La procédure souvent se suive par une simple advection de surface, c.-à-d. chaque point au bord de l'aile est transporté suivant la hauteur calculée et la nouvelle géométrie déformé givrée est atteint. Dans notre modèle, on a ajouté un modèle de tracement d'interface pour les modèles diphasique, appelé Level-Set « Lignes de niveau ». Ce modèle remplace la tache de suivi des points par les anciens modèles lagrangiens. Il nous permit d'implémenter des modèles de givrage multicouches qui approche la réalité. Il permit aussi de capturer le givrage pour des géométries complexe, une tâche encore impossible avec les anciennes modèles.

Notre but est d'avoir un utile capable de simuler le givrage en transitoire. On suppose que le givre se développe couche par couche, chaque couche est quasi-stationnaire. Chaque couche vaut une nouvelle géométrie à remailler, on évite le remaillage par l'utilisation de méthode LEVEL-SET.

Dans cette disposition la géométrie solide est donnée par une fonction implicite dite LEVEL-SET qui se déforme, et bouge librement simulant l'accumulation de givre. La méthode Level-Set la plus basique – avec ordres de discrétisation bas – fonction bien pour des formations de givre lisses sinon on observe des erreurs numériques dissipatives. On a développé des méthodes aux ordres élevés -WENO5 RK3- capables de diminuer ces erreurs numériques. On a développé l'équation de réinitialisation qui permet de remettre la fonction Level-Set dans sa forme.

Pour simuler l'écoulement d'air autour de la géométrie en utilisait la méthode de pénalisation. Ce modèle ne permettait pas de prédire avec précision le profil de vitesse proche paroi, dans la couche limite turbulente. Alors il était remplacé par une méthode de reconstruction basée sur l'interpolation, en imposant une profile de vitesse avec une loi de puissance. Cette loi de puissance est souvent utilisé pour modéliser la couche limite turbulente. Ce n'est cependant pas le meilleur choix, mais le plus simple. Le modèle de turbulence de Spalart-Allmaras était aussi couplé avec cette méthode de pénalisation et reconstruction par interpolation, et est utilisé pour simuler l'aspect turbulent de l'écoulement d'air. Pour le transport des gouttelettes et leur impacte sur l'aile on utilise la méthode de reconstruction par interpolation pour simuler leur impacte sur la paroi. Or, certain gouttelettes impacte la paroi dans la zone amont mouillée, et n'impacte pas derrière vers l'aval, où le solide est sèche. Cette procédure nécessite une condition au limite adaptative, et est atteint par la méthode de reconstruction par interpolation comme pour les méthodes de limite immergée.

Les principaux modules de ce code sont : le module IBM-LS (Immersed Boundary Method coupled with Level-Set), solveur de l'écoulement d'air, et le solveur de capture de gouttelettes.

Le module IBM-LS initialise les fonctions liées à la Level-SET au début, génère les informations nécessaires pour l'IBM-LS pour les reconstructions par interpolation, et est appelé à la fin pour propager la vitesse de givrage, bouger la géométrie représentant l'accrétion de glace et réinitialiser (remettre en forme) la fonction Level-Set. Le solveur de l'écoulement d'air utilise ensuite les informations IBM-LS pour appliquer un profil de vitesse proche paroi afin de reproduire la condition de non-glissement aux bords. Enfin, le module de capture de gouttelettes transporte les gouttelettes et utilise également les informations IBM-LS pour réaliser la procédure de capture dans les zones humides.

Une analyse approfondie de l'importance des différents aspects sur les équations de Level-Set a été présentée. La méthode WENO5 couplé aux schémas RK3 a montré de bonnes performances. La réinitialisation nécessite la correction de conservation locale proposé par Sussman et Fatemi pour réduire les dégradations topologiques. Les équations Level-Set ont bien fonctionné sur des grilles superposées, chimères, aussi bien en advection qu'en réinitialisation. Ensuite, nous avons validé la méthode IBM-LS pour la reconstruction par interpolation, à l'aide des cas de tests simples. Le profil de vitesse de l'air imposé par IBM-LS a également été comparé aux autre données de littérature et a donné de meilleurs résultats qu'avec la pénalisation. Il a été conclu que l'amélioration de profil de vitesse est nécessaire et que le raffinement de la grille proche paroi pourrait améliorer les résultats. La capture des gouttelettes a également été étudiée, et les résultats en termes de taux d'impacte (efficacité de capture) étaient acceptables, avec de légers écarts. L'algorithme a ensuite été mis à l'épreuve ; l'accrétion de glace a été comparée entre deux grilles, l'une grille intégrée IBM-LS, l'autre turbulente et adaptée au corps. La forme de la glace qui en a résulté était presque similaire, si ce n'était de taux d'impacte légèrement plus grand résultant de la grille intégrée. La forme de la glace résultée par différents nombres de couches a enfin été présentée.

Au moment de soumission de manuscrit cela était la fin des travaux réalisé, une conclusion et des perspectives étaient fournis a ce stade. Néanmoins, après la soumission de manuscrit, des améliorations importantes étaient faits et présentées lors de la présentation orale. Un petit rapport était rajouté a la fin de manuscrit détaillant ces suppléments.

La première modification concerne le profil de vitesse de la forme d'une loi de puissance, celui était remplacé par un modèle de paroi turbulent. La loi de puissance est très limitant en application et nécessite un maillage très fin proche paroi. Le modèle de paroi turbulent traite la couche limite turbulente plus correctement, est plus robuste, et reproduis les effets diffusifs de tourbillons attendu des modèles RANS. Le modèle de turbulence de Spalart-Allmaras était aussi modifie pour prendre en effet le modèle de paroi turbulent. Des améliorations importantes étaient observés dans la stabilité de solveur d'écoulement d'air, l'efficacité de capture, et la forme de givre. La deuxième évolution concerne des cas teste supplémentaires. Des comparaisons des résultats de la loi de puissance, et le modèle de paroi turbulent sous différents niveaux de raffinement de maillage étaient présentés. Il était démontré que plus le maillage est raffiné plus la loi de puissance rapproche des résultats bien vérifiés. Il était aussi démontré que le modèle de paroi turbulent était efficace et s'améliore en raffinant le maillage. Une étude sur un maillage chimère, ou les bloque chimère superposent sur la géométrie, était présenté. On a démontré que la méthode en sa totalité fonctionne bien pour un maillage chimère. Le dernier cas teste présenté concerne un cas bien connu dans la littérature, étudié par différents scientifiques et avec des résultats expérimentaux a l'appui. Ce cas teste était choisi prenant en compte sa température très basse. Notre modelé actuellement limité aux situations de givrage type blanc, avait montré un petit écart dans l'épaisseur de givre pour le cas teste présenté dans le manuscrit. Un cas ou la température est très basse, assure un givrage blanc, donc permis une meilleure validation de notre modèle. C'était bien démontré que notre modèle a produit un très bon résultat, comparable avec les études précédant dans la littérature. Il était démontré aussi de nouveau l'importance de modélisation multi-couche de givrage, et pour la première fois dans le domaine, sans remaillage.

Le troisième ajout concerne le traitement des types de givrage plus complexe, qui est la racine des problématiques pour les autres codes concurrents, celui de verglas. Le modèle complet nécessite un développement rigoureux, vu le temps très limité on a opté de simplifier et implémenter un modèle 2D. À partir d'une géométrie 2D défini par une fonction LS, on reconstruit l'interface. Un traitement laborieux est ensuite fait pour pouvoir discrétiser les équations de modèle « Shallow water equation : équations de St-Venant » et pour créer une méthode numérique capable de lui résoudre sur un maillage multi-bloque, chimère et parallélisé. La méthode marchait bien en lui-même, sauf qu'il y avait des difficultés plus profondes ailleurs. L'ingrédient la plus important pour ce modèle est le coefficient de transfert thermique. Celui-là est loin d'être facile à calculer. On a essayé deux méthodes non-simples sans succès. Un travail plus approfondi est nécessaire a ce stade pour mieux estime ce coefficient.

Certaines améliorations sont fortement recommandées à ce stade. Le premier est de fournir la possibilité de l'imposition dynamique de blocs de maillage chimères à des régions d'intérêt, cartésiennes ou curvilignes. Le deuxième est de continuer la developement de model « Shallow water equation » en 3D, mais tout d'abord de bien calculer le coefficient de transfert thermique. C'est seulement à ce stade que des formes de glace très compliquées, impossibles à traiter avec les codes conventionnels, vont être étudiées. Des situations de v tridimensionnelles avec des effets de dégivrage et de détachement de la glace peuvent être étudiées en profondeur grâce à une nouvelle méthode jamais possible auparavant.

Introduction

Contexte

Humans always thrive to exploit nature in search of comfort and sense of security. From simple tools made of stone to advances in artificial intelligence and various energy sources culmination, many incredible imprints have been left by humanity. One major technological advancement of humanity is the ability to pave the skies. Roaring through the skies, thousands of airplanes connect the vast corners of earth. An average of 6000-10000 planes in the air at any moment carries more than one million passenger [84, 45]. Despite the lingering fear of plane crashes, planes as a mean of transportation is actually one of the safest modes of transportation. Accidents happen. However, the statistics show that planes crash less than any other transportation[31], when the criteria of comparison is over casualty per distance. Since their invention, security is the number one consideration. Comparing the number of accidents for the last 2 decades shows considerable improvement Fig. 1. Two of the main reasons for accidents are mechanical failures and adverse weather conditions. Aircrafts employ abundant mechanisms to foresee weather implications. However they still stumble upon unforeseen circumstances. One of which is icing.

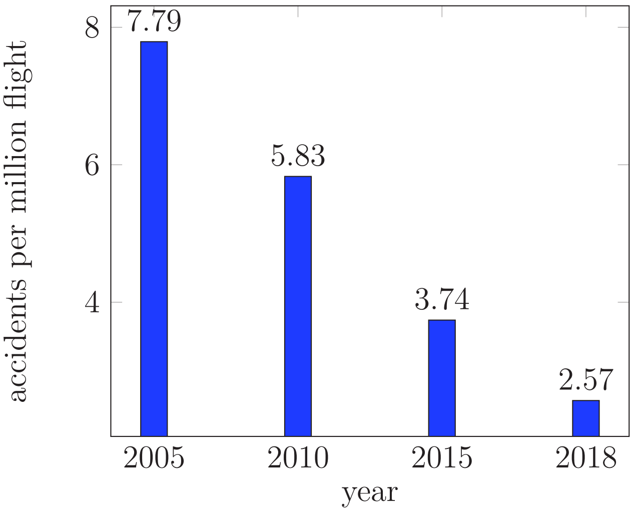


Figure 1 – Number of flight accidents per million flight. Obtained from deviding number of accidents [2] by number of flights per year[76].

The Study of icing and its effects is an important area of research in aeronautics nowadays. This is motivated by the need of secured and highly reliable design criteria for aircraft to be less sensitive to in-flight icing. Supercooled water droplets present in clouds impinge on the surface of aircraft structures. They either solidify totally on impact or partially then create a thin liquid film runback depending on the flow temperature and speed hence, creating dry rime ice or glaze wet ice respectively. Low speeds and low temperatures cause dry rime ice formations, while higher flow speeds and temperatures lead to glaze wet ice formations. Consequences of changes in geometry induced by ice accumulation range from deterioration in performance to hazardous loss of control resulting in accidents. For this reason, aircrafts are certified concerning icing and its effects before being placed on the market. Designing an adequate de-icing mechanism on the other hand, requires full knowledge of the icing phenomenon itself. Icing experimental study cannot exceed the scope of a handful of simple cases due to complexity and cost. Consequently, the use of computational fluid dynamics is inevitable.

Ice accretion simulation codes used today by aerospace industry assume the icing process to be broken up into four steps : first, single phase air flows around the wing transporting water suspended droplets ; second, droplets impinge into the surface generating a liquid or dry film ; third, the film exchanges energy with the surface ; and lastly fourth, the ice is accreted to shape the final form during a certain exposure time. Air flow is usually based on 2D inviscid panel methods coupled with an empirical boundary layer method for heat transfer calculations [54]. Following that, the droplets trajectories are traced using a Lagrangian formulation. Then, the so-called Messinger model is used to evaluate the energy exchange in the liquid film [63]. Having calculated the ice thickness, the new geometry is obtained. However, this process is usually assumed to occur on a single step (single shot) considering that the time scale of the icing process is very long compared with that of the air flow. However this one-way coupling process can be repeated for portions of the required exposure time [42] but still with decoupled time scales.

The current tendency to improve the model is to use Reynolds-Averaged Navier-Stokes (RANS) equations for the air flow such as in NASA's LEWICE3D [12], FENSAP-ICE of McGill University [8], CANICE2D-NS developed at École Polytechnique Montréal [42], and ONERA's ONICE3D [69]. To get the most of such a model based on the RANS solvers, the droplet trajectories can be modelled using an Eulerian formulation as introduced by [92]. This is the case of several icing codes ([15, 52, 56, 69, 113, 22]). The main advantage in using an Eulerian representation is the possibility to calculate the droplet field on the same mesh used by the fluid solver, allowing a better integration with the Navier-Stokes solver and making use of the available CFD acceleration methods.

To tackle full 3D configurations, traditional 1D Messinger model is being gradually replaced by a PDE formulation in [72] and [9]. Many other sub-models for the runback water film exist but are out of the scope of our current study. One of the most important factors affecting the ice accretion is the heat transfer. In FENSAP-ICE and ONICE3D it is determined with the RANS flow solver instead of the classical empirical boundary layer method.

To generate the final geometry however, mesh deformation methods have been employed for multi-step icing calculations as in [9] and [69]. Such techniques are usually easy to implement and were, therefore, favoured. However, ice formation produces stretched cells and badly intersected cells which require specific treatments. These problems are intensified for 3D configurations for which a time-consuming manual remeshing is usually needed. Therefore, inefficiency of these mesh deformation algorithms requires the introduction of a different approach. Such approach must maintain a good quality of the grid especially in the boundary layer region.

The current work is presented as part of an effort to develop a new approach in ice accretion simulation. This new approach should overcome some of the limitations, and most importantly, should inspire and propose Quasi-non-Steady ice accretion abilities. One promising candidate is the Level-Set function which was first introduced by [77] and [78]. In a Level-Set disposition, the interface is defined implicitly via a passive scalar function. This function is set to zero at the interface, positive outside and negative inside. The Level-Set have been used by [10] to evaluate the ice shedding trajectories. In this study however, the Level zero represent the evolution of the ice/air interface. Also, The negative zone would represents the accreted ice and one can solve a heat transfer equation in this subdomain. Likewise, the positive side represents the external flow zone. By respecting a very fine mesh around the interface a high quality flow is obtained. The mesh in question does not need be a body fitted mesh and distance normal to the wall can be arbitrary. Such embedded-grid method maintains the same Eulerian formulation of PDEs, overcoming meshing issues. Modeling with embedded-grids can be performed using structured grids like in [4, 68]. Wall boundary conditions can be treated by a penalization method, in which they are treated as impermeable media in the so called Brinckman-Navier-Stokes equations. It can be also treated in an IBM framework, wherein the solid is defined explicitly. In an IBM disposition the boundary condition can by applied in two ways. The first is on the continuous form of the PDE which resembles the penalization method used in a level-set framework. The second, is by imposing the required BCs on the discretized form.

However, we keep in mind the limitations imposed by these models on the flow evaluation itself. With a non body-fitted grids, orthogonality at the wall is lost, and turbulence modeling requires immense refinements at the wall. This work was done on the prospect that chimera patches can be added dynamically to refine the grid close to the wall. This part of the code, unfortunately is still being under development. Thus, we assess the applicability of the Level-Set method in producing multi-layered ice accretion. Future developments will merge the LS method with the 'to be finished' part of the code of dynamically adding chimera patches.

This thesis is organized in three parts. The first part is devoted to bibliographic study. We present in this part the icing phenomenon in aeronautics which is then divided, where each aspect is discussed in a different chapter. These aspects are namely, the droplet flow, the icing thermodynamics, and the ice geometry evolution. Following that the Level-Set approach is introduced, along with its related issues. And lastly, for the first part, we introduce the embedded grid techniques used to apply the solid boundary condition on the flow. In the second part, the numerical methods used are developed and detailed. In the first chapter of the second part we introduce NSMB solver, along with materials related to icing simulation. In the second chapter of the second part, we develop the icing module in the Level-Set framework. Finally, in the third part includes the results and discussion of different aspects of the icing module. Namely, first aspect is the Level-Set equation validation, second aspect is the boundary condition application to air-flow and droplet-flow in the Level-Set framework, and the third aspect, is multi-step icing. The third part is then finished by general conclusion and perspectives.

Première partie

Literature Review

Chapitre 1

INTRODUCTION TO ICE MODELLING

The development of effective numerical icing modeling codes is highly crucial. The study of ice accretion on aircraft structure and its effects is of a main concern for aeronautic research institutes and industry. The study of such phenomenon is motivated by the need to reduce and limit accidents caused by icing. Flight safety, is highly affected by icing to a point that aircrafts have to be certified for icing. Prediction of aircraft performance deterioration because of ice accumulation is one of the major challenges for the aviation scientific community. As stated by the National Transportation Safety Board [86], ice accumulation and its effects are a major cause of aircraft accidents.

In this chapter we will introduce the basics of modeling aircraft icing phenomenon. Firstly, the phenomenon is explained and analyzed. Following that, the different types of ice are classified. Problems resulted from each type are explained. Then, different ice models that exist in the literature are explained briefly. Each major aspect is then well developed in appropriate chapters.

1.1 Icing Phenomenon

Ice accumulates on aircraft surfaces when flying in severe weather conditions. Ice forms in different surfaces such as wings, engine air inlets, and other control surfaces. Weather conditions behind icing accumulation are : low temperatures below water freezing point and presence of dense clouds with high water droplet content. Droplets impacting against the aircraft either freeze or flow as a runback film on the surface. Depending on the surface temperature, total, or partial freezing of the impacting droplet occurs at the point of impact. When only a portion freezes on impact, the remainder flows above the surface. Depending on further conditions it could still freeze further downstream. The temperature range of aircraft icing events is between 0 $°C$ and -40 $°C$ and the altitude range varies from 90m to 9km.

Icing is more problematic in conditions favoring the formation of very large droplets (Super Large Droplets : SLD). Two basic types of clouds are remarked for icing phenomena ; the first is stratiform clouds spread out over large horizontal areas with low liquid water content. The second is cumuliform clouds of small horizontal areas but high liquid water content.

Typical forms of ice are rime ice and glaze ice. Dry rime icing forms at very low temperatures when the entire drop freezes at the point of impact. This white or opaque frost generally forms at low speeds, low water content and very low temperatures. Glaze ice forms at a temperature near the freezing point when a fraction of the water freezes at the point of impact while the remainder flows along the surface as a reflux freezing downstream. Glaze ice forms mainly at high speeds and high concentrations of liquid water. Glaze ice generates more complex forms than rime ice, such as the formation of ice horns. It generally generates much more disturbances and degradation of aerodynamic performance. A mixed ice formation may develop during a change in flying conditions when glaze and rime ice form simultaneously and mix on the bearing surface. Fig. 1.1 presents the two types of icing commonly encountered.

The main parameters in ice accretion process are the following :

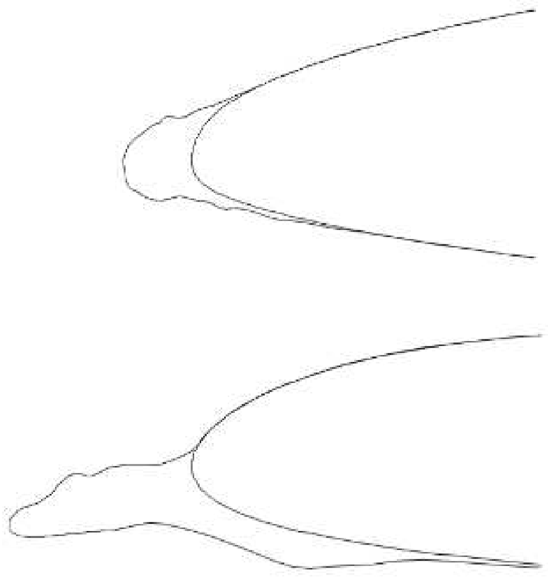


FIGURE 1.1 – Typical forms of icing simulated in NSMB. On top : rime icing on a NFL0414 airfoil. On bottom : glaze icing on a SA13112 airfoil.

* The liquid water content of the environment (LWC : Liquid Water Content$[kg/m^3]$).
* The droplet size and/or the size distribution.
* The air temperature.
* The air velocity.
* The aircraft surface roughness.

Icing research topics are mainly classified into two categories : ice accretion simulation and icing effects analysis. Ice formation on aircraft leads to changes in the aerodynamic profiles of the designed surfaces, drastically impacting their performance. Results obtained in [17] comparing moment, drag, and lift coefficients on a NACA23012 geometry for different icing shapes, showed large increase in drag coefficient associated with a decrease in lift coefficient and stall angle. Furthermore, accumulated icing can suddenly detach and impact downstream surfaces, damaging them. Additionally, ice formation on sensor probes leads to erroneous measurements of flight parameters such as speed and angle of attack leading to serious incidents.

1.2 Icing Models : Numerical Simulation

The main purpose in icing simulation is to understand ice formation physics. In other words, ice modelling attempts to predict ice forms and shapes based on flight and environmental conditions.

Many models and simulation codes have been developed by the aeronautic industry such as LEWICE3D (NASA Glenn Research Center)[12, 105], CANICE (École Polytechnique de Montréal) [87], ONERA's code [35], FENSAP-ICE (McGill University) [8], ICECREMO2 [102], NSMB-ICE (ICUBE University of Strasbourg) [81], CIRA (Italy) [66], and other codes like [56], [69], [113], [22], and [53].

1.2.1 Single-layer, step, or shot ice model

Most of aforementioned codes treat the icing process as a single layer in a quasi-steady state. A single icing step is assumed to be the result of four consecutive steps :

— Single phase air flows around the wing.
— Water droplets are transported, then they impinge against the surface.
— The liquid water film on the surface reaches a thermodynamic balance.
— The geometry is deformed by the accumulation of freezing water.

All external flow, ice accretion, and heat transfer parameters do not change during accretion ; only parameters and variables from the clean geometry are used until the end of the ice accretion. A new flow calculation is then run on the final deformed geometry with a corrected mesh to evaluate the deterioration of the aerodynamic performance.

1.2.1.1 External Airflow

The majority of codes used by the industry are based on a two-dimensional panel method to simulate the external airflow. More recently, some have been coupled with a Navier-Stokes solver replacing the traditional panel method FENSAP-ICE (McGill University) [8], LEWICE3D [12], CANICE-NS [43]. The Reynolds averaged Navier-Stokes equations are coupled with the Spalart-Allmaras turbulence model [95] and its ONERA roughness extension [5].

1.2.1.2 Water Droplet Flow

To predict the droplets trajectory two possible formulations exist : Lagrangian approach [89], and Eulerian approach [92]. In both approaches many assumptions on the dynamics of the droplets are made. The droplets as particles or as a field are transported by the airflow. They impinge on the surface and the rate of impact, called the Collection efficiency β, is calculated.

1.2.1.3 Liquid Water Film : Thermodynamic Model

Water droplets impinging on the surface either fully or partially freeze or flow as a liquid film. A thermodynamic model on the surface is used to evaluate the mass and energy balance. The rate of freezing mass is calculated at the surface and is used to generate the new geometry.

1.2.2 Multi-layer, step, or shot ice model

Icing is a transient phenomenon, since the continuous accumulation of ice on the surface affects the flow parameters progressively [3]. Icing process is assumed to take effect progressively, layer by layer quasi-steadily. Consequently, the external flow field properties are recalculated on the deforming geometry multiple times during accretion. This task requires mesh regeneration at each step (layer). The multi-step model considers the ice to form step-by-step as a piecewise function of time [102]. A criterion to reevaluate the flow is that the ice thickness in any step should not exceed 1 percent of the wing chord length [109] [108]. Wright proposed a formula to estimate the number of steps required based on the maximum amount of ice allowed over the surface during the exposure time. Verdin noted a converging behavior after a small number of steps, around five steps for rime ice and twelve for glaze ice configurations [102]. Using additional steps did not improve the ice shape. Additional steps resulted in ice formations oscillating around the converged shape. He claimed that this behaviour is due to difficulties and errors in the automatic grid generator. Another method for the multi-step icing simulation is by using a predictor-corrector algorithm. In this approach the need for remeshing at each step is avoided. The flow is solved for the clean geometry using a single step model, then reevaluated for the deformed geometry. A second calculation is then launched wherein all physical parameters are interpolated. [67, 38, 110, 102].
Many authors assessed the improvements achieved using multi-Step icing model. Most of the cases on the literature are two dimensional since the automatic 3D grid regeneration is nearly impossible without resulting in very poor grids. The few 3D cases treated using the multi-step icing are not well explained in terms of flow convergence and grid quality and are usually accompanied with poor heat transfer and ice formation calculations.

Chapitre 2

DROPLETS FLOW

As stated in the last chapter, two possible formulations exist : Lagrangian approach [89], and Eulerian approach [92]. In both approaches many assumptions on the dynamics of the droplets are made.

2.1 Assumptions

Assumptions made on the droplets to facilitate their modelling in both Lagrangian or Eulerian representation are the following.

- Droplets are small in size that assuming spherical shape, non-deformability, and incompressibility is accepted.
- Droplets are in the same inertial frame as the air when not disturbed. i.e., droplets have the air velocity at farfield boundary conditions.
- Droplets are at thermal balance with the air. Again, it means droplets have the air temperature at farfield BC.
- There is a one way coupling where droplets flow does not affect the air flow. The following ice accumulation does not affect the droplet flow as well.
- Turbulence effects on the droplet flow are neglected.
- Droplets being small do not rebound or splash on the surface, they are captured by the surface.
- There is no interaction between the droplets, no breakup, coalescence, diffusion, heat transfer, or any other form of interaction.
- The only forces applied to the droplets are 1) Drag force, 2) Gravity force, and 3) Buoyancy force.

2.2 Models

The main objective here is to predict the motion of droplets from the farfield up to impact. The main information required from the droplet solver model is an estimation of the droplet impingement rate which is related to the collection efficiency noted as β. β is a measure of the ratio of distance between adjacent droplets in the far field and the distance between their impact points on the surface.

2.2.1 Lagrangian Approach

In a Lagrangian approach the droplets are considered as particles i.e, droplets have three degrees of freedom. The equation of droplet dynamics derives from Newton's second law [89].

$$m_d \frac{d^2 \boldsymbol{r_d}}{dt^2} = \sum \boldsymbol{F} \tag{2.1}$$

where m_d is the droplet mass, $\boldsymbol{r_d}$ the displacement vector of position, and $\boldsymbol{F}$ external forces.

A cloud of particles injected randomly on the farfield are traced until impingement against the surface. Their trajectories are calculated by solving the following equation.

$$\rho_d V_d \frac{d^2 \boldsymbol{r_d}}{dt^2} = (\rho_d - \rho_a) V_d \boldsymbol{g} + \frac{1}{2} \rho_d A_d C_d d |\boldsymbol{u}_a - \boldsymbol{u}_d| (\boldsymbol{u}_a - \boldsymbol{u}_d) \tag{2.2}$$

where g is the gravitational acceleration, ρ the density, V_d the droplet volume, A_d the cross sectional area of droplets, $\boldsymbol{u}$ the velocity vector, and $C_d d$ the droplet drag coefficient. The drag coefficient is given by an empirical relation. Under the assumption of spherical droplets at low Reynolds number Re_d the following empirical relation was chosen in [89] :

$$\frac{C_d d Re_d}{24} = 1.0 + 0.197 Re_d^{0.63} + 2.6 \times 10^{-4} Re_d^{1.38} \tag{2.3}$$

where Re_d is the droplets relative Reynolds number which is defined as :

$$Re_d = \rho_a d_d |\boldsymbol{u}_a - \boldsymbol{u}_d| / \mu_a \tag{2.4}$$

where d_d is the droplet diameter.

Then , the impingement rate is to be calculated geometrically. Different Lagrangian methods exist : Ensemble-Average Method, Weighted-Ensemble Method, Area Method "streamtube method", and Full Lagrangian Methods [107]. In Full Lagrangian Methods the droplet concentration is also traced along with other variables based on a Lagrangian differential equation along the droplet path. [107] further proposed a Globally Eulerian/Locally Lagrangian Method.

2.2.2 Eulerian Approach

On the other hand, an Eulerian approach utilizes droplet field conservation equations as proposed by [92] and as implemented by [15], [52], [56], [69], [113] and [22]. An Eulerian formulation has many advantages over the traditional Lagrangian such as the direct determination of the impingement rate, simplicity with multi-body geometries and 3d simulations, and straightforward integration to flow solvers.

An Eulerian approach solves a droplet velocity and volume fraction fields. The governing equations take the form seen in Eq. 2.5.

$$\begin{cases} \frac{\partial \alpha}{\partial t} + \nabla.(\alpha \mathbf{u}_d) = 0 \\ \frac{\partial \alpha \mathbf{u}_d}{\partial t} + \nabla.\left(\alpha \mathbf{u}_d \otimes \mathbf{u}_d\right) = \mathbf{F}_D + \mathbf{F}_{BG} \\ \mathbf{F}_D = \alpha \frac{C_D Re_d}{24K}(\mathbf{u}_a - \mathbf{u}_d) \\ \mathbf{F}_{BG} = \alpha \left(1 - \frac{\rho_a}{\rho_w}\right) \frac{1}{Fr^2}\mathbf{g} \end{cases} \tag{2.5}$$

where α is the water volume fraction, $\mathbf{u}_d$ droplet velocity field, $\mathbf{u}_a$ air velocity, ρ_w water density, ρ_a air density, $\mathbf{g}$ gravity vector, $K = \rho d U_\infty / 18 L \mu$ inertia parameter, L reference length, and d droplet median diameter. Re_d is the droplets Reynolds number and is defined based on the slip velocity between the air phase and the droplets in Eq. 2.4. Fr is Froude number which is a measure of the ratio of the inertial to gravitational external force given by $Fr = U_\infty / \sqrt{Lg}$.

The drag coefficient C_D can be calculated using the equation 2.3 as suggested by [89]. Another one proposed by [90] is given in equation 2.6.

$$\begin{cases} C_D = \frac{0.44}{Re_d} & for \quad Re_d > 1000 \\ C_D = \frac{24\left(1+0.15\, Re_d^{0.657}\right)}{Re_d} & for \quad 0.1 > Re_d > 1000 \\ C_D = \frac{24}{Re_d} & for \quad Re_d < 0.1 \end{cases} \tag{2.6}$$

Because of the assumption that all droplets are captured by the surface, and the fact that the model treats the icing in quasi-steady state, special treatment is required for the droplets boundary condition. Droplets are assumed to disappear on the solid body in zones where droplets impact on the solid body. In other words, the solid acts as Neumann outlet. Howevers, in dry zones where droplets do not impact on the solid body, the solid should act as a Dirichlet BC. The droplet velocity vector is checked at the interface and the appropriate BC is applied locally accordingly. This method applied in [52] is given by equation 2.7

$$\begin{cases} if \quad \mathbf{u}_d \cdot \mathbf{n} < 0, \quad Impact\, zone \quad \to Neuman \quad \frac{\partial u}{\partial n} = 0, \frac{\partial \alpha}{\partial n} = 0 \\ \\ if \quad \mathbf{u}_d \cdot \mathbf{n} > 0, \quad Dry\, zone \quad \to Dirichlet \quad \mathbf{u} = \mathbf{0}, \alpha = \mathbf{0} \end{cases} \tag{2.7}$$

These equations 2.5 were solved using Finite Element Method in [15], and using Finite Volume Method in [52, 23, 69] in which they used the QUICK or HLLC schemes for the advection term. The time scheme usually used is an explicit Runge-Kutta of different orders. [8] used an implicit time scheme.

The collection efficiency is directly calculated using the formula in equation 2.8.

$$\beta = \alpha \mathbf{u}_d \cdot \mathbf{n} \tag{2.8}$$

2.3 Droplet Size Considerations

The aforementioned model treats the droplets as having a single constant mean valued diameter d_d. Real clouds however contain a wide range of droplet sizes. Using a mean value can be questioned as introduced in [80]. Wherein, droplets distribution assumed are considered as a distribution of spheres divided into a number of groups (bins) each with a mean valued diameter (LANGMUIR "D" dimensionless distribution) Fig. 2.1. For a distribution of droplet diameter, the distribution is split into a number of groups each with a mean diameter and each with a weight ω_i . The transport equation 2.5 is solved for each bin size separately and in the end the overall impingement rate is evaluated by the weighted average and is given by equation 2.9

$$\beta = \sum_i \omega_i \beta_i \tag{2.9}$$

2.4 Super Large droplets

Under certain meteorological conditions droplets can have large diameter up to 2mm. The aerodynamic characteristics are affected greatly in the presence of SLD. Zones downstream that are normally shadowed are now susceptible to ice accretion. The reason behind

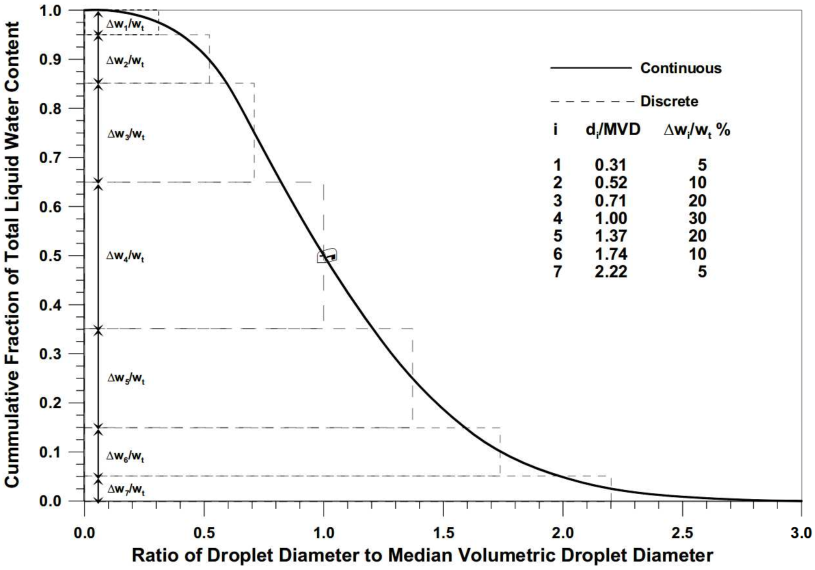


FIGURE 2.1 – LANGMUIR “D” dimensionless distribution of droplet diameter [80].

this is the assumptions usually taken in droplet solvers. Droplets are assumed to be of spherical shape, to have constant volume, to be non deformable, to not rebound or splash, to not break up. Add to that the runback liquid film assumed not to be affected by the impingement.

2.4.1 SLD Deformation Splashing and Rebounding

Large droplets are subjected to deformation near the leading edge. Many experimental and numerical studies investigated the effect of deformation on the impingement rate, heat transfer, and the resulting ice form. In case of an Eulerian droplet, the deformation of the droplets can be treated by changing the empirical relations of the drag as function of their inertia. The splashing and rebounding can be treated as well by empirical relations, wherein the boundary condition is modified to take into account the size of the droplet and its direction with respect to the surface. The droplet equations are modified to take into account these effects [49], [74], [106], [6], [71], [97], [59], [91], [88], [101], and [44].

For SLD deformation the first approach is to assume the droplets to deform into disks. This results in a modification of the drag coefficient. A more elaborate drag coefficient takes the form in equation 2.10

$$C_d = \left[C_{d,sphere} + ee\left(C_{d,disk} - C_{d,sphere}\right)\right] Re_d \tag{2.10}$$

where ee is a weight factor given by another empirical equation 2.11 and is a measure of the deformation of the droplet from a sphere to a disk, $C_{d,sphere}$ the drag coefficient of a sphere, and $C_{d,disk}$ that of a disk.

$$ee = 1 - (1 + 0.007We^{0.5})^{-6} \tag{2.11}$$

where We is the Weber number which measures the relative importance of the droplet's inertia to its surface tension. It is given by equation 2.12.

$$We = \rho_d|\mathbf{u_a} - \mathbf{u_d}|^{\mathbf{2}}\mathbf{D}/\sigma \tag{2.12}$$

where σ is the droplets surface tension approximated at $0.0756N/m$ for water.

The empirical relations used to calculate $C_{d,disk}$ and $C_{d,sphere}$ are given in equations 2.13 and 2.14.

$$\begin{cases} C_{d,sphere} = 0.1875\frac{24}{Re_d} & for \quad Re_d <= 0.01 \\ C_{d,sphere} = \frac{24}{Re_d}\left(1 + 0.1315Re^{0.82-0.05ww}\right) & for \quad 0.01 < Re_d <= 20 \\ C_{d,sphere} = \frac{24}{Re_d}\left(1 + 0.1935Re^{0.6305}\right) & for \quad 20 < Re_d <= 260 \\ C_{d,sphere} = 10^{1.6435-1.1242ww+0.1558ww^2} & for \quad 260 < Re_d <= 1500 \\ C_{d,sphere} = 10^{-2.4571+2.5558ww-0.9295ww^2+0.1049ww^3} & for \quad 1500 < Re_d <= 1.2 \times 10^4 \\ C_{d,sphere} = 10^{-1.9181+0.6370ww-0.0636ww^2} & for \quad 1.2 \times 10^4 < Re_d <= 4.4 \times 10^4 \\ C_{d,sphere} = 10^{-4.3390+1.5809ww-0.1546ww^2} & for \quad 4.4 \times 10^4 < Re_d <= 3.38 \times 10^5 \\ C_{d,sphere} = 29.78 - 5.3ww & for \quad 3.38 \times 10^5 < Re_d <= 4 \times 10^5 \\ C_{d,sphere} = 0.1ww - 0.49 & for \quad 4 \times 10^5 < Re_d <= 10^6 \\ C_{d,sphere} = 0.19 - 8 \times 10^4/Re_d & for \quad 10^6 < Re_d \end{cases} \tag{2.13}$$

$$\begin{cases} C_{d,disk} = \frac{64}{\pi Re_d}(1 + \frac{Re_d}{2\pi}) & for \quad Re_d <= 0.01 \\ \\ C_{d,disk} = \frac{64}{\pi Re_d}(1 + 10^{-0.883+0.906ww-0.025ww^2}) & for \quad 0.01 < Re_d <= 1.5 \\ \\ C_{d,disk} = \frac{64}{\pi Re_d}(1 + 0.138 Re_d^{0.792}) & for \quad 1.5 < Re_d <= 133 \\ \\ C_{d,disk} = 1.17 & for \quad Re_d > 133 \end{cases} \tag{2.14}$$

where ww is the logarithm of Re to power 10 ; $ww = log_{10}(Re_d)$.

Chapitre 3

ICING THERMODYNAMICS

As stated in the last chapter, the main information in icing is calculating the impingement efficiency. This information permits calculating the water mass impacting at each point of the surface under study.

Evaluating the mass and energy balance of the liquid film generated by the droplets on the surface with thermodynamic models permits the prediction of the iced surface. In other words, the main result to be obtained from such thermodynamic model of the liquid film is the determination of the portion of water mass freezing at different points of the surface.

In this chapter we present the different methods of calculating the heat transfer as well as the different film thermodynamic models existing in the literature.

3.1 Convective Heat Transfer Coefficient

In icing thermodynamics, accurate evaluation of the convective heat transfer h_c coefficient is essential. It is from this variable that one can evaluate the wall convective heat flux. The convective heat flux at the wall is the driving factor in ice formation. To evaluate the heat transfer coefficient many methods are applicable depending on the flow regime and type of the fluid under study. In thermodynamics h_c is usually calculated from the Nusselt number which in turn depends on the flow and the fluid thermal properties. In icing h_c is mainly influenced by the wall temperature and the ice type, glaze or rime. Furthermore, h_c is evaluated either empirically using a boundary layer method, or by solving the Navier-Stokes flow equations and evaluating the heat flux at the wall. Solving the flow using Navier-Stokes equations coupled with a turbulence model like the Spalart-Allmaras, makes it possible to add a roughness extension, ONERA's extension or BOEING's [5]. By imposing a wall temperature during the fluid calculation one can calculate the resulting wall heat convective flux. Following that, the calculation of the convective heat transfer can be done using simple heat transfer notions. h_c is defined as the ratio between the energy transfered by heat and the driving force i.e., the temperature gradient. Heat transfer occurs when the temperature at the boundary layer immediately adjacent to the wall is different from the wall temperature. To evaluate this temperature, the wall is assumed to be a perfect insulator, thus equilibrium is attained and the wall temperature becomes equal to the temperature at the boundary layer, this term is called the recovery temperature T_{rec} or the adiabatic wall temperature. For convection heat transfer in a compressible gas the following relations are usable.

$$h_c = \frac{Q_{wall}}{T_{rec} - T_{wall}} \tag{3.1}$$

$$T_{rec} = T_\infty \left(1 + r\left(\gamma - 1\right) M^2/2\right) \tag{3.2}$$

where T_∞ is the free stream temperature, or, in icing, the farfield temperature, M the Mach number, γ the isentropic exponent known as the heat capacity ratio or the adiabatic index, and r is known as the recovery coefficient. This recovery coefficient depends on many factors such as the flow regime, flow type, and the thermal properties of the fluid. For a laminar boundary layer over a flat plate $r = \sqrt{Pr}$, while for a turbulent boundary layer over a flat plate $r = \sqrt[3]{Pr}$. [69] used equation 3.2 to evaluate the recovery temperature using the values of r discussed above.

To better predict the recovery temperature, the Navier-Stokes solver can be used. By imposing zero heat flux at the surface the wall being in thermal balance with the boundary layer adjacent, is at the recovery temperature, i.e., $T_{rec} = T_{wall,adiabatic}$. The Navier-Stokes solver is then launched a second time by imposing an arbitrary temperature at the wall, from which the heat flux at the wall is evaluated. Following that, one can use equation 3.1 to evaluate h_c.

The next step is to model the mass and energy balance on the surface. Many models exist in the literature, the most important of which are presented here.

3.2 Messinger Model

The simplest model is the Messinger Model which is based on mass and energy balance [63] as given in equation 3.3. In this model, the radiation is neglected, and no conduction transfer is taken into consideration, neither in the ice accumulated nor on the water film. The model called "runback" model is applicable in 2D configurations. In this model the liquid film is generated and computed starting from the stagnation point then loops on the cells towards the trailing edge. The film is supposed to freely flow from one cell to its adjacent cell, when it does not freeze, with no liquid mass conserved locally in the cells.

$$\begin{cases} \dot{m}_{in} + \dot{m}_{imp} - \dot{m}_{ice} - \dot{m}_{evap} - \dot{m}_{out} = 0 \\ \\ \dot{q}_{in} + \dot{q}_{imp} - \dot{q}_{ice} - \dot{q}_{evap} - \dot{q}_{out} - \dot{q}_{cnv} = 0 \end{cases} \tag{3.3}$$

where $\dot{m}_{in}$ is the mass flow rate of water entering by reflux from an upstream cell, $\dot{m}_{imp}$ the impacting mass of water at the cell under study, $\dot{m}_{out}$ the mass of water coming out by reflux to a downstream cell, $\dot{m}_{evap}$ the rate of evaporation or sublimation. $\dot{q}$ is the heat transfer rate by the different mechanisms explained above, and $\dot{q}_{cnv}$ is the convective heat transfer rate. Some of these terms are available from previous calculations and some are given by empirical relations [110].

$$\dot{m}_{imp} = LWC \cdot V_\infty \beta A \tag{3.4}$$

$$\begin{cases} \dot{q}_{in} = \dot{m}_{in} C_p (T_{s-1} - T_{ref}) \\ \\ \dot{q}_{out} = \dot{m}_{out} C_p (T_s - T_{ref}) \\ \\ \dot{q}_{evp} = \dot{m}_{evp} (C_p (T_s - T_{ref}) - L_{evap}) \\ \\ \dot{q}_{ice} = \dot{m}_{ice} (C_{p,ice} (T_s - T_{ref}) - L_{fusion}) \\ \\ \dot{q}_{cnv} = h_c (T_s - T_{rec}) \\ \\ \dot{q}_{imp} = \dot{m}_{imp} (C_p (T_\infty - T_{ref}) + V_\infty^2 / 2) \end{cases} \tag{3.5}$$

where LWC is the liquid water content at the farfield in kg/m^3, β the collection efficiency at the surface cell under study, V_∞ the velocity of the air at the farfield. C_p is the specific heat capacity of water, $C_{p,ice}$ the specific heat capacity of ice, T_s the temperature at the surface cell under study, T_{s-1} the temperature of the upstream point, $T_{ref} = 273.15K$ is the water freezing temperature, L_{evap} latent heat of evaporation, L_{fusion} latent heat of fusion of water.

The freezing fraction is defined as :

$$f = \frac{\dot{m}_{ice}}{\dot{m}_{in} + \dot{m}_{imp} - \dot{m}_{evap}} \tag{3.6}$$

The mass conservation equation is then rewritten as :

$$\begin{aligned} \dot{m}_{ice} &= f(\dot{m}_{imp} + \dot{m}_{in} - \dot{m}_{evap}) \\ \dot{m}_{out} &= (1-f)(\dot{m}_{imp} + \dot{m}_{in} - \dot{m}_{evap}) \end{aligned} \tag{3.7}$$

In thermodynamic phase change problems the use of trial and error method on the type of phase is recurrent. To complete the equations, an additional closure is added. Three situations are possible : Wet for glaze ice state, Dry for rime ice, and Liquid for no icing but pure liquid film. For wet surface ($0 < f < 1$) the wall temperature T_s is equal to the freezing temperature $T_s = T_{ref} = 273.15K$. A dry surface on the other hands, for which $f = 1$, means no liquid mass flows out $\dot{m}_{out} = 0$ and is at a temperature less than the freezing point of water $T_s < T_{ref}$. Finally, a liquid surface has a temperature higher than the freezing point ; no ice is formed $f = 0$, $\dot{m}_{ice} = 0$.

To solve the Messinger equations we start by localizing the stagnation point and set $\dot{m}_{in} = 0$. Then we loop on the surface cells from this point to the trailing edge. At each cell, we assume a type of ice thus closing the system, and check if the guess is correct. We start by assuming a wet surface thus $T_s = 0$. If the freezing fraction is ($0 < f < 1$) then the guess is correct. If not, we change the type depending on the value of f. If f is negative or equal to zero we set it equal to zero. It is a liquid surface, for which $\dot{m}_{ice} = 0$. If f is higher or equal to 1 we set it equal to one. It is a Dry surface, for which $\dot{m}_{out} = 0$

3.3 Iterative Messinger Model

The Iterative Messinger Model [113] introduced the possibility to treat multiple stagnation points. The model is based on the same equations as the Messinger model. An addition to the Messinger model is made by evaluating the mass flowing out of the cells from the wall shear stress and pressure gradient. This model also allows the accumulation of liquid mass in the cells.

$$\begin{cases} \sum \dot{m}_{in} + \dot{m}_{imp} - \dot{m}_{ice} - \dot{m}_{evap} - \sum \dot{m}_{out} = 0 \\ \\ \sum \dot{q}_{in} + \dot{q}_{imp} - \dot{q}_{ice} - \dot{q}_{evap} - \sum \dot{q}_{out} - \dot{q}_{cnv} = 0 \end{cases} \tag{3.8}$$

The crucial difference here is that we don't need to start from the stagnation point. Consequently, the model can be extended easily to 3D configurations. The reflux is calculated from the wall forces, so extending the calculation to a 3D case is possible. The mass flow reflux direction depends on the forces or the friction coefficient C_f. The mass can flow in or out from all the directions. In 2D we talk about two points of the cells, while in 3D the cell is enclosed by its edges. For a constructed mesh a cell is enclosed by four edges. The mass flowing along a direction or the other is determined by the following equation :

$$\begin{aligned} \dot{m}_{out,x} &= \tfrac{C_{f,x}}{|C_{f,x}|+|C_{f,y}|} \cdot \sum \dot{m}_{in} \\ \dot{m}_{out,y} &= \tfrac{C_{f,y}}{|C_{f,x}|+|C_{f,y}|} \cdot \sum \dot{m}_{in} \end{aligned} \tag{3.9}$$

The iterative behaviour of the method is used to close the system, however the closure-three types of surface trial and error-addition used for the messinger model is still required. The iteration is required since the stagnation point is not known. We start by assuming all the points as the stagnation point for which $\dot{m}_{in} = 0$.

3.4 Extended Messinger Model

Other recognisable methods are based on solving ordinary or partial differential equations ODE.-PDE. The first ODE method was proposed by [72] then extended into 2D for multi-body geometries by [79]. This Extended Messinger Model added the ability to solve a convection heat transfer equation on ice and water substrates. The main difference from the Messinger model is the temperature gradient in the water and ice layers. The thermodynamics model is based on a Stefan problem of phase transition. The conduction in each sub layer is given by the following equations :

$$\frac{\partial T_i}{\partial t} = \frac{k_i}{\rho_i C_{pi}} \frac{\partial^2 T_i}{\partial y^2} \tag{3.10}$$

$$\frac{\partial T_w}{\partial t} = \frac{k_w}{\rho_w C_{pw}} \frac{\partial^2 T_w}{\partial y^2} \tag{3.11}$$

The mass conservation equation takes the following form :

$$\rho_i \frac{\partial h_i}{\partial t} + \rho_w \frac{\partial h_w}{\partial t} = \dot{m}_{in} + \dot{m}_{imp} - \dot{m}_{evap} \tag{3.12}$$

The change of phase is governed by the following equation :

$$\rho_i L_{fusion}\frac{\partial h_i}{\partial t} = k_i\frac{\partial T_i}{\partial y} - k_w\frac{\partial T_w}{\partial y} \tag{3.13}$$

To close the system it is supposed to be in quasi-steady state. Myers [72] showed that this is valid for thin ice formation, for ice height less than $2.4cm$ and water film height less than $3mm$.

3.5 Shallow Water Icing Model

A PDE method was proposed by [14], which is based on the Messinger model and the Shallow-Water equation SWIM. The film velocity $\mathbf{u_f}$ is taken as a function of the coordinates $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2})$ on the surface and the normal distance to the wall y. Another hypothesis consists of considering the velocity distribution in the direction normal to the wall at the film to follow a linear relation given by the following formula :

$$\mathbf{u_f}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{y}}{\mu_\mathbf{w}}\tau_\mathbf{wall}(\mathbf{x}) \tag{3.14}$$

Where μ_w is the water's dynamic viscosity, τ_{wall} the wall shear stress assumed to be the principal force responsible for the flow of the liquid film. In a shallow water model variations along the depth are neglected. Consequently, $\mathbf{u_f}$ should be a function only of $\mathbf{x}$. The equation 3.14 is thus integrated along the depth of the film to obtain a mean film velocity :

$$\mathbf{u_f}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{1}}{\mathbf{h_f}}\int_\mathbf{0}^{\mathbf{h_f}} \mathbf{u_f}(\mathbf{x}, \mathbf{y})\mathbf{dy} = \frac{\mathbf{h_f}}{\mathbf{2}\mu_\mathbf{w}}\tau_\mathbf{wall}(\mathbf{x}) \tag{3.15}$$

where h_f is the water film height. The wall shear stress τ_{wall} is evaluated by the air solver "Navier-Stokes equations".

The partial differential equations for mass and energy conservation as proposed by [9] take the following form :

$$\begin{cases} \frac{\partial}{\partial t}\rho_w h_f + \nabla.(\rho_w \bar{\mathbf{u}}_f h_f) = \dot{m}_{imp} - \dot{m}_{evap} - \dot{m}_{ice} \\ \\ \frac{\partial}{\partial t}\rho_w C_w h T_f + \nabla.(\rho_w C_w \bar{\mathbf{u}}_f h_f T) = \dot{Q} \\ \\ \dot{Q} = \dot{q}_{imp} - \dot{q}_{evp} - \dot{q}_{ice} - \dot{q}_{cnv} \end{cases} \tag{3.16}$$

where the different terms are the same for those explained in the Messinger model. T is the temperature of the liquid film. The same closure problem used with the messinger model is required here. Eq. 3.16 is complemented with four compatibility relations as in Eq. 3.17

$$\begin{cases} h \geq 0, \\ \dot{m}_{ice} \geq 0, \\ h \cdot T \geq 0, \\ \dot{m}_{ice} \cdot T \leq 0. \qquad T \text{ is in } {}^\circ\text{C} \end{cases} \tag{3.17}$$

These relations state that water depth h_f and mass of ice $\dot{m}_{ice}$ can only be positive. On the other hand $h \cdot T \geq 0$ states that the water can only exist when the film temperature is higher than the freezing temperature of water at 0°C. And the last condition is the result of the third where water freezes only at temperatures lower than 0°C.

This model have apparently three unknowns with only two equations. These unknowns can be reduced to two by defining three types of surfaces :

— A dry surface, for which $h = 0 \longrightarrow h \cdot T = 0$

— A wet surface, for which $T = 0 \longrightarrow h \cdot T = 0$

— A liquid surface, for which $\dot{m}_{ice} = 0 \longrightarrow \dot{m}_{ice} \cdot T > 0$

For each case, several terms disappear, and since the type of the surface is unknown, a process of trial and error is used. An educated guess is chosen based on initial values, then the compatibility relations are verified. If one of the compatibility relations is not satisfied we switch to another surface type as shown in Fig. 3.1.

[20] proposed another extension on the Myers model based on two PDEs for 3D glaze ice configurations. All the thermodynamic models discussed above can be extended to consider anti-icing and de-icing [41], [29]. Many researchers modelled the rivulets that appear in de-iced airwings.[29] proposed a physical and mathematical model of rivulet flow. However, he concluded that the effect of water-air interfacial waves, dynamic contact angle and water-ice interactions are very complex and need further investigation. [37] also developed another numerical approach to predict the rivulets in a runback water film. [85] proposed a microphysic model of ice deposition using random methods to decide whether a droplet would freeze instantly or flow along the film. [40] further improved on the surface roughness consideration and heat transfer by coupling the ice roughness and transfer models. Their new model greatly enhanced the prediction of ice-horns typical with glaze ice formations that were usually over-predicted by the classical models.

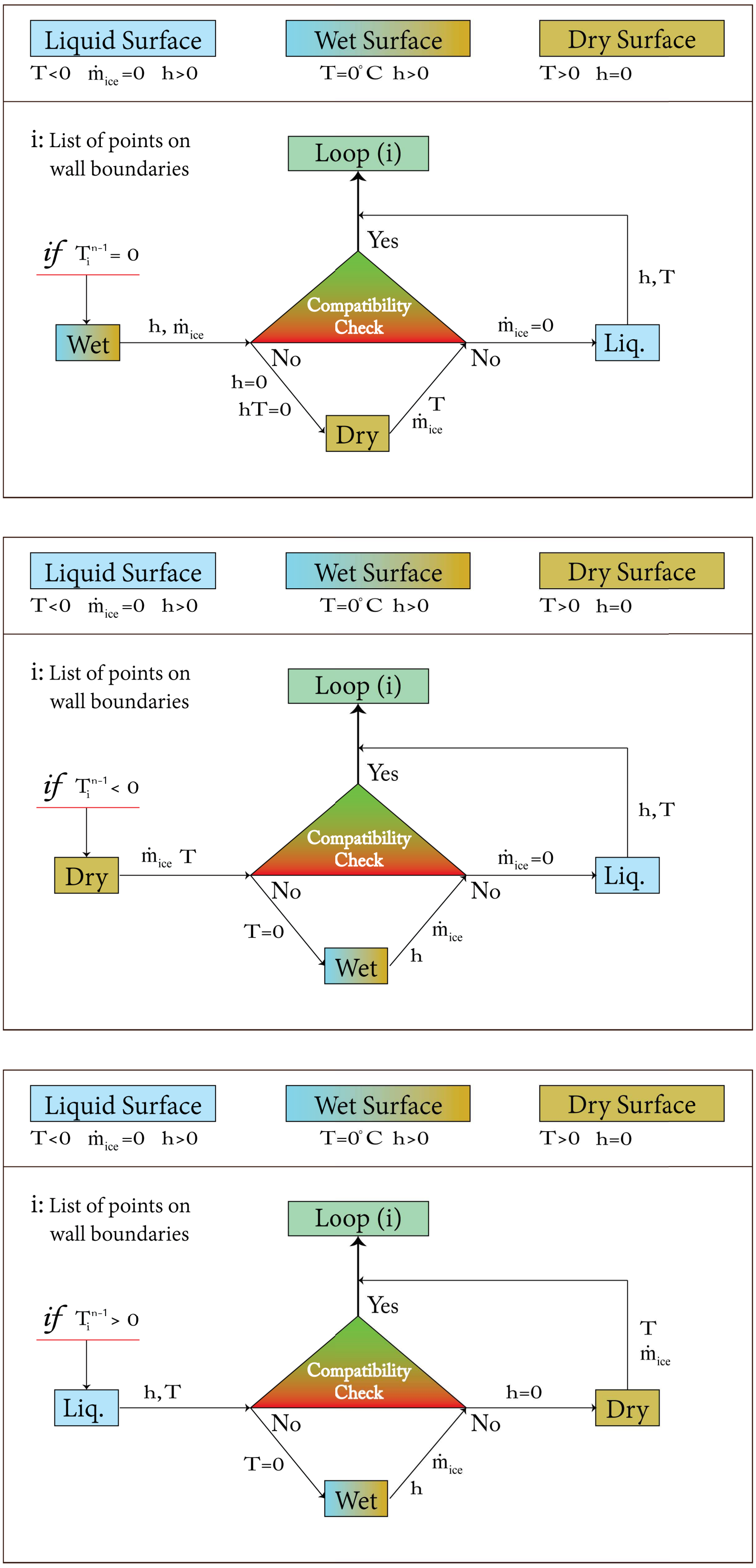


FIGURE 3.1 – Trial and error procedure at each wall node i.

Chapitre 4

ICED GEOMETRY EVOLUTION

To generate the new iced geometry many techniques exist in the literature [69], [21], [102], [43]. These techniques are usually based on a Lagrangian displacement coupled with a remeshing technique. In a single step icing configuration the result is the final iced surface, and remeshing is necessary to evaluate the aerodynamic deterioration caused by ice accumulation. In a multi-icing code the remeshing is required after each shot to resolve the whole system of equations for each shot. Each shot is assumed to be in quasi steady state with one way coupling.

4.1 Remeshing

The simplest approach is to displace the nodes in a direction normal to the wall, with a distance equal to the height of ice calculated at the node. The magnitude of ice height is taken usually as an average between the displaced face points [94]. To conserve the mass of the ice a sub-iteration method is usually used, where the height for each sub-layer is corrected to include the extension in the cell size when it is advected outside. One of the main drawback of such a Lagrangian technique is the difficulty in maintaining high quality grid. [69], [21], [102], [43] used automatic remeshing techniques for multi-layer ice simulation. However, for glazed configurations automatic remeshing was found to result in poor calculations and manual remeshing was required in most situations. The accumulation of ice on the surface results in cells stretching, thus quality close to the wall is lost if the calculation isn't started with a huge number of points. These points should be distributed according to the final iced shape to be obtained, thus knowledge of the final form is crucial before starting the first calculation. These processes are impossible to be automated. Another possibility is to remesh the surface after each shot instead of just displacing the initial mesh. This requires a full remeshing of the domain. For structured grids, there exists algebraic, elliptic and parabolic methods. Remeshing on non-structured grids is more straightforward but refining the grid close to the wall to capture the turbulent boundary layer is really difficult.

The most complex problem to discuss in remeshing techniques is their use for 3D or multi-body configurations. Another problem is the treatment of topological changes such as coalescence of two ice fronts. Referring to Fig. 4.1, we see two situations where automatic remeshing fails. In the first, two fronts blend, a difficult topology to be handled automatically. In the second, two fronts initially connected get separated, another difficult situation for automatic remeshers. It should be stated that these situations are also difficult to be handled through typical IBM methods in which the interface is given explicitly in a list of lagrangian points. A manual modification is quite necessary for such configurations. These situations are usually encountered in glaze ice formulation. Wherein, horns or protrusion are usually encountered. Under long exposure times these protrusions coalesce. Simply moving the surface nodes, generates severely distorted elements, not mentioning that the connectivity between these nodes is well not established. Different nodes from different fronts meet and negative volumes are generated, simply deleting the elements does not conserve the quantities under study. A very special manual treatment is required in such situations. Smoothing and regularization of these elements and rerunning the calculations to ensure conservation of mass at lower time steps can be hideous if not impossible for 3D configurations.

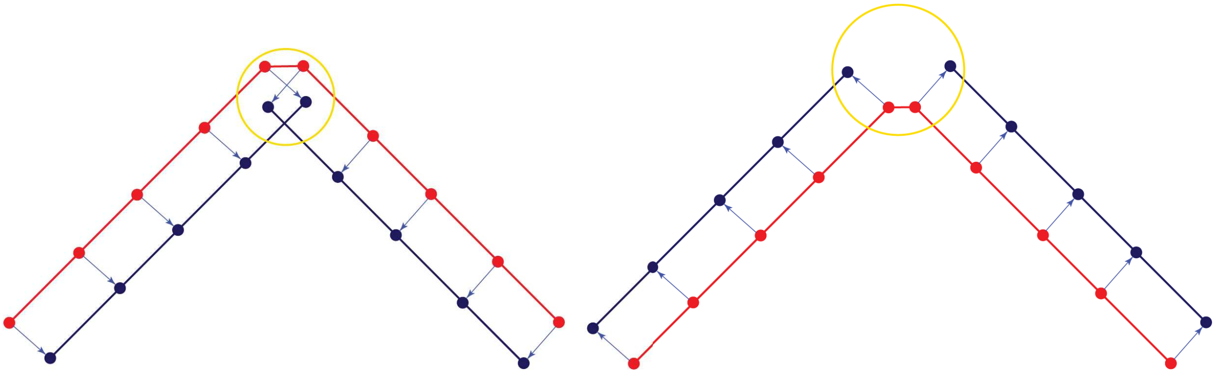


FIGURE 4.1 – Topoligical problems occuring when using remeshing.

To avoid such problems, the use of implicit interface tracing techniques solves all of these problems. Interface tracing techniques such as the Volume of Fluid "VOF" or the Level-Set are the most adapted to trace the evolution of the interface for these topological problems. The VOF method has a disadvantage when calculating the direction normal to the interface because of the precision of the reconstruction method. On the other hand the Level-Set method has most of the properties capable of solving all the problems explained before.

4.2 Level-Set Method : no Remeshing

The Level-Set method was proposed by [81] to trace the deformed geometry using the Level-Set function. An icing velocity drives the interface in the normal direction by advecting an implicit function. This method was shown to be more robust since manual remeshing was avoided using embedded grid techniques like IBM or Penalization methods [4]. To solve the flow around the deformed geometry it is sufficient to penalize the flow in the regions where the level-set function describes the solid geometry [1], [55]. This method is further explained in the following chapters.

Chapitre 5

LEVEL-SET FUNCTION

In this chapter we introduce the Level-Set function which gives an implicit definition of the interface. In two or three dimensional spaces, a lower dimensional interface can be defined by an implicit function ϕ that separates the domain into two or more parts. The interface can be designated by the zero isocontour of ϕ. By sitting a sign to such function ϕ we can discern points on the outside from points on the inside, $\phi(\mathbf{x}) > \mathbf{0}$ outside, and $\phi(\mathbf{x}) < \mathbf{0}$ inside, Fig. 5.1.

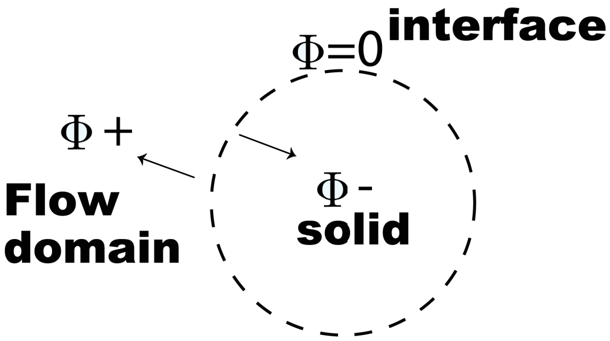


Figure 5.1 – Level-Set function representing the solid geometry.

The Level-Set approach was developed in [81] to track the ice air interface evolution. The ice/air interface is represented implicitly by the zero isocontour of the Level-Set function ϕ. The Level-Set equation is set as the signed distance function [78]. A signed distance function is given by : $|\phi(\overrightarrow{x})| = d(\overrightarrow{x})$. It is initialized in the computational domain as follows :

— $\phi = d$, in the outside zone (air)

— $\phi = -d$, in the inside zone (ice,solid)

— $\Gamma = x|\phi(\mathbf{x}, \mathbf{t}) = \mathbf{0}$, the interface

where d is the distance to the interface Γ.

The Level-Set function has many advantageous properties such as :

- The normal direction or vector $\overrightarrow{\mathbf{n}}$ is easily calculated through equation 5.1

$$\overrightarrow{\mathbf{n}} = \frac{\nabla\phi}{|\nabla\phi|} \tag{5.1}$$

- The tangential vectors can also be determined by generating an orthonormal basis with the normal $\overrightarrow{\mathbf{n}}$

- The curvature κ is also easily calculated via equation 5.2

$$\kappa = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \tag{5.2}$$

- If the Level-Set function is a signed distance as explained above it has an additional property : its gradient is a unit vector $|\nabla \phi| = 1$ and thus $\overrightarrow{\mathbf{n}} = \nabla \phi$, and $\kappa = \Delta \phi$.

The solid body is usually defined by a characteristic function χ_s. This characteristic function is a smoothed Heaviside function computed from the signed distance level-set function as in equation 5.3,5.4

$$H(\phi) = \begin{cases} 0 \qquad \phi < -\epsilon \\ 0.5\,(1 - (\phi/\epsilon) - \sin(\pi\phi/\epsilon)/\pi) \qquad |\phi| \leq \epsilon \\ 1 \qquad \phi > \epsilon \end{cases} \tag{5.3}$$

$$\chi_s = H(-\phi) \tag{5.4}$$

Another important ingredient is the Dirac delta function δ_s.

$$\delta(\phi) = \begin{cases} 0 \qquad \phi < -\epsilon \\ 0.5\,(1 + \sin(\pi\phi/\epsilon)) \quad |\phi| \leq \epsilon \\ 1 \qquad \phi > \epsilon \end{cases} \tag{5.5}$$

where ϵ is a parameter used to choose the region over which these functions are smoothed. ϵ should always have a value close to the size of the cell close to the wall. It is usually set equal to the thickness of the largest cell close to the wall, measured in the direction normal to the wall.

5.1 Level-Set Advection : Iced Surface Evolution

The idea behind using the Level-Set to follow the ice interface is to use a transport equation, a PDE. The interface Γ is advected by the accumulation of ice. Actually it is the whole signed distance Level-Set function that is advected. By introducing an ice accumulation velocity on the boundary calculated by dividing the mass rate of ice accumulation $\dot{m}_{ice}$ by the ice density. This velocity is projected in the normal direction, since ice is assumed to form normal to the wall.

$$\mathbf{v}_{ice,wall} = \frac{\dot{m}_{ice}}{\rho_{ice}}.\nabla\phi \tag{5.6}$$

The Level-Set advection called the Level-Set equation was first introduced by [77]. It defines the evolution of the implicit signed distance Level-Set advection in an externally generated velocity field. The equation in icing framework is given in 5.7.

$$\frac{\partial\phi}{\partial t} + \mathbf{v}_{ice}\nabla.\phi = 0 \tag{5.7}$$

This PDE models the evolution of the interface Γ for which $\phi(\mathbf{x}) = \mathbf{0}$. Unlike the mesh deformation techniques that rely on a Lagrangian formulation, it is an Eulerian formulation. One thing to notice is that $\mathbf{v}_{ice}$ should be defined on the whole domain in which this equation is solved, and not only on the interface. Thus the velocity calculated at the interface is propagated, as explained in section 5.3.

5.2 Level-Set Reinitialization

When the signed distance Level-Set function is advected, the signed distance property gets deteriorated. Consequently, the signed distance function needs reconstruction. Reinitializing ϕ while being advected can restore the values around the interface to signed distance. Advecting a signed distance function is numerically stable and ameliorates conservation [78]. Reinitializing ϕ at the end of each ice layer is essentially required to calculate the new solid characteristic function χ_s from the smoothed Heaviside function $H(\phi)$. To recover $|\Phi(\overrightarrow{x})| = d(\overrightarrow{x})$ we solve the reinitialization equation Eq. 5.8 proposed by [78]. The equation can be rewritten in the form Eq. 5.9 as well.

$$\frac{\partial\phi}{\partial t} + |\nabla\phi| = 1 \tag{5.8}$$

$$\frac{\partial\phi}{\partial t} + sign(\phi_0)\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\phi = sign(\phi_0) \tag{5.9}$$

At steady state ϕ does not change any more, thus $\frac{\partial\phi}{\partial t} = 0$. This means that the gradient of ϕ becomes a unit vector, which is only possible once ϕ is equal to signed distance ; i.e., $|\nabla\phi| = 1$. The equation is written in the form 5.9 to allow a single treatment for the whole domain, where the advection direction is dictated by the sign function $sign(\phi)$.

5.2.1 Sign function

The sign function $sign(\phi)$ should be set equal to one outside and minus one inside and zero at the interface. But since the Level-Set is defined implicitly on the domain, this sign function needs to be smoothed so that the cells through which the interface passes have a small value close to zero, pertaining a sign, nonetheless. Otherwise these points are going to need be used as boundary conditions of two seperate problems where each side is solved independently. While redistancing the LS function ϕ, the sign function shall not change, since the interface is stationary, which is why it is written as $sign(\phi_0)$. The numerical smeared out sign function could be written as proposed by [60].

$$sign(\phi) = \frac{\phi}{\sqrt{\phi^2 + \Delta x^2}} \tag{5.10}$$

Another suggestion made by [82] takes the following form :

$$sign(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2 \Delta x^2}} \tag{5.11}$$

He suggested that equation 5.11 gave better results than 5.10.

5.2.2 Interface Preservation Considerations

Solving the reinitialization equation ensures a signed distance function everywhere. The cells through which the interface passes, keep themselves as so. Cells outside stay outside, and cells inside stay inside. No movement of the interface globally. However, the interface is not forced to stay stationary locally. That is, the interface can move within the cell it intersects because of numerical perturbations, and because of the smoothed sign function. Actually the reinitialization equation 5.9 is not consistent with the fact that the interface should stay stationary locally.

Method of Sussman and Fatemi 1999 [98]

Many researchers suggested an additional fix to the reinitialization equation 5.9. The first method was proposed by [98]. They added a source term to equation 5.9. This source term conserves the volume at each side of the interface on the cell it passes. The modified equation takes the form :

$$\frac{\partial\phi}{\partial t} + sign(\phi_0)\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\phi = sign(\phi) + \lambda\delta(\phi_0)|\nabla\phi_0| \tag{5.12}$$

where λ : is a function that ensures that the local volume occupied to each side of the interface inside the cell it passes stays the same. After few mathematical operations and a choice of another function to ensure the correction is only at the interface, without disturbing the distance away from the interface, [98] concluded the following formulation :

$$\lambda = -\frac{\int_{\Omega_{i,j,k}} \delta(\phi_0)(\phi^{n+1} - \phi_0)/\Delta t}{\int_{\Omega_{i,j,k}} \delta^2(\phi_0)|\nabla \phi_0|} \tag{5.13}$$

The integration in Eq.5.13 in the numerator or denominator is an integration over the cell $\Omega_{i,j,k}$ and is evaluated from a 9-point stencil in 2d Eq. 5.14 or from a 27-point stencil in 3d as given in Eq. 5.15

$$\int_{\Omega_{i,j}} f = \frac{\Delta x^2}{24}\left(16 f_{i,j} + \sum_{m,n=-1;(m,n)\neq(0,0)}^{1} f_{i+m,j+n}\right) \tag{5.14}$$

$$\int_{\Omega_{i,j,k}} f = \frac{\Delta x^3}{78}\left(52 f_{i,j,k} + \sum_{m,n,q=-1;(m,n,q)\neq(0,0,0)}^{1} f_{i+m,j+n,k+q}\right) \tag{5.15}$$

Eq. 5.9 is first solved as a prediction step then Eq. 5.12 corrects ϕ. The additional source term impacts the LS function ϕ only at the interface where $\delta(\phi) > 0$. So one can avoid the computational cost of the prediction step by calculating the correction term only at points where $\delta(\phi) > 0$.

Method of Hartmann et al. 2010 [27]

Another method was proposed by [27] by constraining the reinitialization equation. The main idea is to add a forcing source term to equation 5.9. By defining a set of points on the interface, a local target distance is then computed via linear interpolation. The modified equation takes the form in equation 5.16

$$\frac{\partial \phi}{\partial t} + sign(\phi_0)\frac{\nabla \phi}{|\nabla \phi|} \cdot \nabla \phi = sign(\phi) + \beta \delta_F \mathcal{F} \tag{5.16}$$

where β is a weighting factor set to $\beta = 0.5$ in [27]. The idea behind the High-order Constrained Reinitialization (HCR) proposed in [27], is to use a correction either from one side of the interface and thus optimally setting $\beta = 1$, or for smoother and better results to use a correction on all cells in the interface to reduce the weight factor to $\beta = 0.5$.

Eq. 5.9 is solved first, as a prediction step, then the forcing term can be determined. The predicted LS function ϕ is then corrected at all the cells in order to restore the zero level-set. The Dirac delta ensures that this second step affects ϕ only at the interface.

In Eq. 5.16 the forcing term $\mathcal{F}$ drives ϕ towards the target value. The HCR1 proposed by [27], minimizes the deviations of the interface as given in Eq 5.17 via a least square method.

$$\mathcal{F} = \frac{1}{\nabla x}\left(\frac{1}{M}\sum_{\alpha=1}^{M} r_\alpha \phi_\alpha - \phi\right) \tag{5.17}$$

where M is the number of neighbors across the interface, i.e., those with different sign, and r_α is given by : $r_\alpha = \phi_0/\phi_{\alpha 0}$.

Another formulation proposed by [27] (HCR2) is derived by preserving the location of the interface throughout the reinitialization via a determined problem. In this formulation the interface location is restrained at a single point instead of minimizing the interface deviation at all locations identifiable by the linear interpolation between cell groups across the interface. The forcing term $\mathcal{F}$ is calculated via Eq. 5.18.

$$\mathcal{F} = \frac{1}{\nabla x}\left(\tilde{r}\sum_{\alpha=1}^{M} \phi_\alpha - \phi\right) \tag{5.18}$$

where $\tilde{r}$ is given by : $\tilde{r} = \phi_0/\sum_{\alpha=1}^{M}\phi_{\alpha 0}$.

5.3 Advection of a Field In the Normal Direction

Given a velocity field or any information available at the interface, there exists a system of partial equations to propagate it in the direction normal to the interface. In our case an icing velocity is calculated at the interface by the accumulation of freezing mass 5.6, and needs to be propagated throughout the whole domain. This velocity is the velocity used to advect the LS function given in Eq. 5.7. The Hamilton-Jacobi equation system given in Eq. 5.19 is used to extrapolate the variable $\mathbf{v}_{ice,wall}$ perpendicularly to the interface.

$$\frac{\partial \mathbf{w}}{\partial t} + sign(\phi)(\nabla\phi)\cdot\nabla\mathbf{w} = 0 \tag{5.19}$$

where the initial value is $\mathbf{w}_0 = \mathbf{v}_{ice,wall}$; in other words it is equal to the icing velocity at the interface and zero everywhere else. At steady state after convergence, $\mathbf{w}$ gives the required propagated or advected velocity field normal to the wall. It should be stated that, when using an explicit scheme with the restricted time step for the conditional stability, the propagation follows a strict marching behaviour. At each step the information is propagated one step further from the zero level-set and in the normal direction. This is quite important since we don't need this velocity throughout the whole domain but only on the region where icing is to occur, a small percentage of the chord length, instead of the whole numerical domain, which could extend to thirty times the chord.

5.4 Considerations On The Level-Set Equations Discretization

Many Discretization schemes were proposed in the literature, for both the temporal and spatial derivatives. Most researchers approach the problem using the Finite Difference Method, and only few used the Finite Volume Method. The level-set equations (5.7,5.9,5.19) can all be written in the more general form seen in Eq. 5.20.

$$\frac{\partial \psi}{\partial t} + \overrightarrow{\mathbf{U}} \cdot \nabla \psi = c \tag{5.20}$$

These variables are given in table 5.1.

General variable	ψ	$\overrightarrow{\mathbf{U}}$	c
Level-Set Advection Eq.	ϕ	$\mathbf{v}_{ice}$	0
Reinitialization Eq.	ϕ	$sign(\phi_0)\frac{\nabla\phi}{\|\nabla\phi\|}$	$sign(\phi_0)$
Propagation in the Normal Direction	$\mathbf{w}$	$sign(\phi)(\nabla\phi)$	0

Table 5.1 – The Level-Set equations variables written in the general format given by Eq. 5.20.

The first step in developing a general conservative Finite Volume discretization is to rewrite the last equation in the conservative form as given in Eq. 5.21.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\overrightarrow{\mathbf{U}}\psi) - \psi \nabla \cdot \overrightarrow{\mathbf{U}} = c \tag{5.21}$$

A cell in three-Dimensional structured grid is given by its three indices (i, j, k), $\Omega_{i,j,k}$, has a volume V and a boundary Γ given as six surfaces or faces (n, s, e, w, f, b) . By integrating the Eq. 5.21 over the cell volume we obtain

$$\int_\Omega \frac{\partial \psi}{\partial t} + \int_\Omega \nabla \cdot (\overrightarrow{\mathbf{U}}\psi) = \int_\Omega c + \int_\Omega \psi \nabla \cdot \overrightarrow{\mathbf{U}}$$

Using Ostrogradsky's theorem and averaging over the cell, and over the faces we obtain :

$$\frac{\partial \psi}{\partial t} V + \sum_{f:n,s,e,w,f,b} (\overrightarrow{\mathbf{U}}\psi)_f \cdot \mathbf{S_f} = cV + \psi \sum_{f:n,s,e,w,f,b} \overrightarrow{\mathbf{U}}_f \cdot \mathbf{S_f}$$

5.4.1 Implicit Time Schemes applied to LS equations

Using implicit time schemes restricts the spatial scheme since the use of a biger stencil is very costly and could become prohibitive. Pena et al. 2016 [81] used an implicit time scheme coupled with upwind, central, and higher order total variation demenishing scheme via deffered correction as spatial schemes, in all of which the implicit stencil is made of a single neighbor at each side. The right hand side term $\overrightarrow{\mathbf{U}}_f \cdot \mathbf{S_f}$ however, is treated explicitly. This explicit treatment broke the unconditional stability of the implicit scheme. Results obtained in [81] showed minor mass loss due to numerical diffusion, that becomes significant for non smooth interfaces. Multi-layer icing simulation would accumulate such numerical diffusion obtained at each layer. Consequently, the iced shape would be drastically smoothed.

Another variation on the implicit treatment of the right hand side was proposed by Mikula et al. [64, 65]. They proposed a semi-implicit scheme based on an inflow-implicit/outflow-explicit treatment. The flux is first calculated at each face $\overrightarrow{\mathbf{U}}_f \cdot \mathbf{S_f}$, then based on its direction an implicit or explicit representation of the variable reconstructed at the face and at the cell center, $\psi)_f$ and ψ consecutively, are set implicitly or explicitly. The main advantage of semi-implicit scheme over the full implicit scheme is the reduction of the artificial diffusion on the numerical solution, thus higher accuracy is presumed to be obtained.

However, we found the unconditional stability of the method gets deteriorated since the method is not purely implicit. The main problem is with the consistency of these almost implicit schemes. Numerical diffusion is impossible to eliminate whatever the time step used is. Actually, an explicit scheme gives better results than an implicit scheme for the same time step, $CFL \leq 1$.

5.4.2 Explicit Time Schemes applied to LS equations

We will first deal with the spatial schemes available with explicit time methods. Typically, the fifth-order Weighted Essentially Non Oscillatory (WENO5) scheme is used to discretize the LS equations generalized in Eq. 5.20 [77, 114, 100, 103]. The studies of Osher [77] have shown that Essentially Non Oscillatory (ENO) and/or WENO schemes are adequate for the discretization of the LS equations. Highly robust, capable of dealing with singularities likely to develop in the solutions of these equations, and on the other hand, thanks to their high order precision, one can hope to maintain a fairly good description of the fine structures of the interface.

A comparative study made by Solomenko [114] contains an extensive comparison on the use of many high order spatial schemes, such as the WENO5 of [50], and the High-Order Upstream Central (HOUC) of [75]. They showed that the results obtained with the WENO5 scheme were more satisfactory than results of the HOUC scheme.

For the temporal scheme many explicit variations exist in the literature. The Euler explicit scheme is the least expensive scheme but is often insufficient for complex simulations. The Adams-Bashforth method of second order is the most commonly used. Frequently used as well, the Runge-Kutta schemes offer very good compromise between numerical stability and precision.

5.5 Frequency and Importance of Reinitialization

When advecting the LS function the signed distance gets deteriorated and diffused, especially close to the interface. As such, it has been a practice to reinitialize it at a given frequency, to restore the signed distance. Solomenko et al. [114] showed that excessive reinitialization of the level-set given in Eq. 5.9 increases the numerical error. He showed that when using high temporal and spatial discretization schemes the reinitialization given by Eq. 5.9 worsens the results, even when it restores a signed distance. However, when coupled with the fix of Sussman and Fatemi [98] as given in Eq. 5.12 to conserve the interface from local deviations, better results are obtained. He compared the two methods of local interface conservation, and showed that the method of Sussman and Fatemi [98] outperforms the method of Hartmann et al. [27] by one order of magnitude, in light of the mass loss generated by interface deviation. The table 5.2 summarizes their findings.

Method of Reinitialization	Mass Loss
No Reinitialization	$-1.8 \cdot 10^{-4}$
Reinitialization with no source term	$-4.1 \cdot 10^{-3}$
HCR1 de Hartmann	$2.1 \cdot 10^{-4}$
HCR2 de Hartmann	$2.4 \cdot 10^{-4}$
Sussman and Fatemi	$-2.8 \cdot 10^{-5}$

Table 5.2 – Summery of results obtained in [114], calculated for a translating sphere test, and using WENO5 and RK2 schemes.

Chapitre 6

OTHER ASPECTS IN ICE MODELLING

Many other aspect relating to aircraft icing have been studied in the literature. Some of the main aspects are summarized in this chapter.

6.1 Helicopter Icing

Despite the fact that icing is less dangerous to helicopter than other aspects, it is still an important factor. With the increasing use of helicopters, the potential to undergo inconvenient conditions could make ice greatly hazardous. A crucial problem arises when setting a certification procedure for helicopters ; the danger and cost, as well as the peculiar conditions to be applied, of such tests motivated the development of numerical analysis models. For helicopters, concern with the rotor-blade airfoil icing is crucial. Many authors studied the rotor-blade degrading aerodynamics and the rotor torque in icing condition [57, 16, 30].

Cao and Chen [19] summarized helicopter icing simulation methods and challenges in helicopter icing simulation. They concluded the difficulties in numerical simulation of helicopter icing in three main points :

1. The water behaves unpredictably due to the flapping and lagging of the rotor blades.
2. The variations of the flow from one point on the blade to another ; the rotor flow incurs complexities in estimating the flow field accurately. The flow can even reach transonic zones at the tip, due to its high velocity.
3. The icing being highly sensitive to local surface roughness, which varies significantly from one point on the blade to another, makes it very difficult to accurately estimate the heat transfer.

6.2 Ice Crystal Icing

A new concern appeared when it was noticed that ingestion of ice crystals in turbofan engines caused power loss or even engine damage [73]. They developed an eulerian method to trace the trajectories of ice crystals. They assume that the dispersed ice phase is diluted in a continuous air phase. The method is one was coupled as in water droplet trajectories used in conventional icing models. The form of the ice crystal is treated by an adaptive drag coefficient, depending on some shape factors fitted to experimental data [34]. The ice crystal are subjected to convective and evaporative heat transfer with the air dictated by empirical formulas. They concluded that the model accurately predicted the ice shapes.

6.3 Reduced-Order Model and POD

[51] suggested the use of reduced order model for aircraft icing simulation. [32] used POD for multi-parameter analysis of ice accretion. They used POD and multidimensional interpolation. Their approach predicted the ice shapes with fair accuracy. They then suggested to optimize the procedure of selecting snapshots in areas with nonlinearities. [111, 112] used a local POD to optimally select snapshots for non-linear problems by using machine learning algorithms.

6.4 Wind tunnel effects on ice accretion

[115] studied the effects of wind tunnel on the ice accretion. They predict that the scaling of subscale models ad the presence of walls confining the flow have detrimental effects on the simulation of ice accretion. They carried out experiments in free air flow and inside wind tunnels at varying blockages. They showed a dependence of the blockage, and errors upto ten percent on the estimation of lift force on the iced airfoil. And concluded that all experimental results should quantify their dependence on wall blockage.

6.5 Wall roughness effects on icing Models

[24] investigated the effects of wall roughness on ice models by proposing a new wall function approach. [61] employed a model to characterize ice roughness evolution on an airfoil surface. They investigated the roughness evolution experimentally as well.

Chapitre 7

EMBEDDED GRID TECHNIQUE

In this chapter we will introduce Embedded Grid Techniques. Methods such as Immersed Boundary Method (IBM) and Level-Set (LS) are usually used when the surface description of the solid body is irregular. Such is the case for fluid-structure interaction problems. The central point in these techniques is the application of the boundary condition at the wall. The wall is given either explicitly or implicitly. In explicit representation, the wall is usually given as a list of points in 2D or as a list of triangles in 3D, e.g., STL format. In implicit representation on the other hand, a mathematical function such as the LS function is used. This function is defined in the studied domain, and the solid-liquid interface is its zero contour.

7.1 Immersed Boundary Method

The immersed boundary method allows the study of complex geometries using simple Cartesian meshes. By introducing a forcing term into the discretized or the continuous equation, the required boundary condition is applied at the immersed elements at the boundaries. It is therefore a means of managing boundary conditions for surfaces not aligned with the mesh. It allows management of mobile geometries in a fixed mesh and saves the use of grid regeneration or moving grids.

Excellent summaries of the different existing IBM methods can be found in Mittal and Iaccarino [68] or Merlin [62], which are briefly reproduced in part in this section.

Imposing or forcing the boundary conditions on immersed boundaries is the key to the development of an IB method. It is also what distinguishes one method from another. Consider a differential equation describing a conservation law

$$\frac{\partial C}{\partial t} + \nabla \cdot \mathbf{u}C = f \qquad\qquad \text{in}\,\Omega_{\mathrm{f}} \tag{7.1}$$

with

$$C = C_\Gamma \qquad\qquad \text{at the boundary}\,\Gamma \tag{7.2}$$

When discretized using Finite Volume method on a structured grid, the discretized equation takes the form :

$$A_p(i,j,k)C(i,j,k) + \sum_{nb} A_{nb}(i,j,k)C_{nb}(i,j,k) = rhs(i,j,k) \tag{7.3}$$

where nb runs through the six sides, e, w, n, s, f, b.

7.1.1 Implicit Forcing

Imposing the required boundary condition on the continuous form 7.1 is called implicit forcing, continuous since it is applied to the continuous form, and implicit since the body needs an implicit definition. Since the IB body is given by an explicit list of points, Lagrangian points, a Forcing term F_k is transferred from the forcing point to neighboring fluid points. The effect of the immersed boundary on the fluid around it is applied by a forcing term applied to the momentum equation of the conservation law given by :

$$f_m(\mathbf{x},t) = \sum_k \mathbf{F}_k(t)\delta(|\mathbf{x} - \mathbf{X}_k|) \tag{7.4}$$

where δ is the Dirac Delta function and $\mathbf{X}_k$ the coordinates of the Lagrangian point k. The Dirac function is defined in the whole domain where it is set to zero everywhere except close to the Lagrangian points, where these points set the Dirac in the neighboring fluid points using a smooth continuous representation.

The forcing term $\mathbf{F}_k(t)$ is taken by assuming the body as elastic but highly rigid, as being attached to an equilibrium position [58, 11], or as a dampened oscillator [36, 46]. It can be given by :

$$\mathbf{F}_k(t) = -\kappa\left(\mathbf{X}_k - \mathbf{X}_k^e(t)\right) \tag{7.5}$$

where κ is a stiffness parameter, $\mathbf{X}_k^e$ the equilibrium position of the k^{th} Lagrangian point.

Applying a Dirichlet boundary condition, such as the no-slip BC required for fluid momentum equation, is straightforward. However, imposing a Neumann BC, or a mixed Robin BC to the momentum equation is inconceivable, at least to our knowledge.

7.1.2 Explicit Forcing

Using the discretized form of any PDE as the one given in Eq. 7.3, any type of BC can be applied directly. Direct Explicit Forcing is achieved by changing the coefficients in Eq. 7.3, as well as the right-hand-side term. The forcing can be achieved at the cell center by imposing the state vector required to achieve the BC at the wall. The forcing can as well be achieved at the cell edges to keep the conservative form of the discretized equation. The forcing can be achieved at the layer of cells adjacent to the interface ; there exists two layers, one on the fluid side and the other on the solid side of the IB. By forcing on the layer on the fluid side we decouple the cells from their neighbors by changing their constants, Eq. 7.3, namely A_p, A_{nb}, rhs. Forcing the BC on the solid side is achieved by considering them as ghost cells. Firstly, all the points on the solid side are decoupled completely to impose the solid body state, e.g., its velocity $\mathbf{u}_s$. Then, the state vector at the layer adjacent to the IB, is interpolated from the fluid side, applying the required BC.

7.2 Level-Set Approach to Boundary Condition Forcing

When the solid body is defined implicitly through a Level-Set function ϕ_s , the BC can be applied in similar manner as in IBM. In this implicit representation or LS disposition the solid is given by a characteristic function χ_s, which is set to a smoothed Heaviside function H_s. The smoothed Heaviside function in turn uses the signed distance LS function $H_s(-\phi_s)$. Setting the LS function as a signed distance enhances the numerical accuracy [78, 70], since the resulting characteristic function can be smoothed. A penalty term is added as a source term to the PDE to respect the no-slip boundary condition at the wall [4, 1] for NS equations. This term resembles the continuous Boundary Condition forcing used in IBM. The only difference is that the Heaviside function is constructed from the LS signed distance function. In this method the solid bodies are treated as impermeable media in the so called Brinckman-Navier-Stokes equations. The penalty term takes the form :

$$f_u = \frac{1}{\eta}\chi_s\left(\mathbf{u}_f - \mathbf{u}_s\right) \tag{7.6}$$

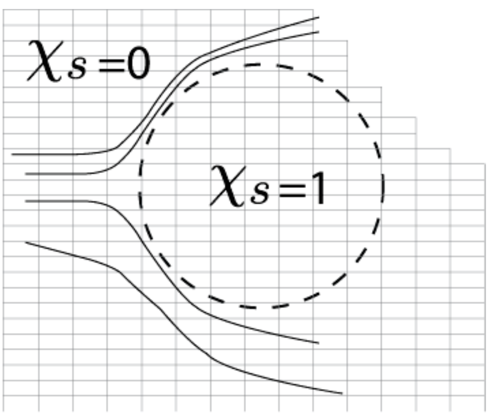


Figure 7.1 – Solid body defined via χ_s treated as impermeable, blocking the flow.

where $\mathbf{u}_s$ as stated before is the solid body velocity required to be imposed for the no-slip boundary condition at the wall, $\mathbf{u}_f$ the state vector, and $\frac{1}{\eta}$ a penalization parameter.

Deuxième partie

Numerical Methods

Chapitre 8

NSMB Solver

The NSMB code [104] solves the compressible Navier-Stokes equations using a multi-block, Finite Volume formulation, parallelized in MPI (Message Parsing Interface). NSMB solver has many features : many numerical discretization schemes (spatial and temporal), a wide variety of turbulence models including LES and DES, equilibrium and out-of-equilibrium air chemistry for hypersonic flow calculations as well as immersed boundary methods and chimera grid techniques. The NSMB manual details the code's functionalities [104]. In this chapter, we briefly present Navier-Stokes compressible equations, as well as the main models and numerical schemes used in our icing simulations.

8.1 History

In 1989, the Euler multi-block EULMB structured solver was developed at the Swiss Federal Institute of Lausanne (EPFL). The innovative choices of EULMB's design were as following :

1. The use of dynamic memory allocation at runtime
2. Designed for parallel computing by solving blocks in parallel on many processors using distributed memory.
3. written with the MEMCOM database system containing the data structure for simulation including mesh size, boundary condition information and the fluid solutions. The MEMCOM database provides quick access to data both for reading and writing.

In 1991, the viscous terms were added to EULMB and the first version of NSMB was born. From 1992 to the end of 2003, NSMB was developed as part of an NSMB consortium comprising several universities (EPFL, Lausanne, Switzerland, SERAM, Paris, France, IMFT, Toulouse, France, KTH, Stockholm, Sweden), CERFACS (Toulouse, France), and several industrial partners like EADS-France (Airbus France and EADS Space Technologies), SAAB Military Aircraft and CFS Engineering. In 2003, Airbus-France and CERFACS decided to leave the consortium in favor of the ONERA elsA code. Since 2004, the development of NSMB has been pursued by EPF-Lausanne, ETH-Zurich, ICUBE Strasbourg, IMFT, KTH Stockholm, CFS Engineering and the University of Dresden. Originally written in Fortran 77, the code has since been converted to Fortran 90.

8.2 Governing Equations

The compressible Navier Stokes Equations are the main pillar of the Fluid Solver. They describe the conservation of mass, momentum and energy of the flow field. In three dimensions and Cartesian coordinates (x, y, z), Navier-Stokes' unsteady compressible equations can be written in the following conservative form [104] :

$$\frac{\partial}{\partial t}(W) + \frac{\partial}{\partial x}(f - f_v) + \frac{\partial}{\partial y}(g - g_v) + \frac{\partial}{\partial z}(h - h_v) = 0 \tag{8.1}$$

where t is time. The state vector W is defined as :

$$W = (\rho, \rho u, \rho v, \rho w, \rho E) \tag{8.2}$$

f, g, h denote the convective fluxes and are given by :

$$f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho E + p) \end{pmatrix}, g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(\rho E + p) \end{pmatrix}, h = \begin{pmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(\rho E + p) \end{pmatrix} \tag{8.3}$$

where ρ is the density, u, v and w the Cartesian velocity components, p the pressure and E the total energy.

f_v, g_v, h_v denote the viscous fluxes and are given by :

$$f_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ (\tau U)_x - q_x \end{pmatrix}, g_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ (\tau U)_y - q_y \end{pmatrix}, h_v = \begin{pmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ (\tau U)_z - q_z \end{pmatrix} \tag{8.4}$$

where τ is the shear stress tensor defined as :

$$\begin{aligned}
\tau_{xx} &= \tfrac{2}{3}\mu\left(2\tfrac{\partial u}{\partial x} - \tfrac{\partial v}{\partial y} - \tfrac{\partial w}{\partial z}\right) \\
\tau_{yy} &= \tfrac{2}{3}\mu\left(-\tfrac{\partial u}{\partial x} + 2\tfrac{\partial v}{\partial y} - \tfrac{\partial w}{\partial z}\right) \\
\tau_{zz} &= \tfrac{2}{3}\mu\left(-\tfrac{\partial u}{\partial x} - \tfrac{\partial v}{\partial y} + 2\tfrac{\partial w}{\partial z}\right) \\
\tau_{xy} &= \tau_{yx} = \mu\left(\tfrac{\partial v}{\partial x} + \tfrac{\partial u}{\partial y}\right) \\
\tau_{yz} &= \tau_{zy} = \mu\left(\tfrac{\partial w}{\partial y} + \tfrac{\partial v}{\partial z}\right) \\
\tau_{xz} &= \tau_{zx} = \mu\left(\tfrac{\partial u}{\partial z} + \tfrac{\partial w}{\partial x}\right)
\end{aligned} \tag{8.5}$$

where μ is the viscosity.

U in Eq. 8.4 denotes the viscous dissipation and is calculated from :

$$\begin{aligned}(\tau U)_x &= \tau_{xx}u + \tau_{xy}v + \tau_{xz}w \\ (\tau U)_y &= \tau_{yx}u + \tau_{yy}v + \tau_{yz}w \\ (\tau U)_z &= \tau_{zx}u + \tau_{zy}v + \tau_{zz}w\end{aligned} \tag{8.6}$$

q_x, q_y, q_z in Eq. 8.4 denote the conductive heat flux components and are calculated using Fourier's law for thermal conduction,

$$(q_x, q_y, q_z) = \left(-k\frac{\partial T}{\partial x}, -k\frac{\partial T}{\partial y}, -k\frac{\partial T}{\partial z}\right) \tag{8.7}$$

where T is the temperature and k the thermal conductivity. For perfect gas liquids the viscosity, μ, can be calculated from the Sutherland's law which states the following for air at standard atmospheric pressure :

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2} \frac{(T_\infty + S_1)}{(T + S_1)} \tag{8.8}$$

where μ_∞ is the viscosity at the reference temperature of T_∞, S_1 which for air is a constant set to 110.3. The Prandtl number is assumed constant and is set for air to $Pr = 0.72$, the coefficient of heat conductivity can then be found via :

$$k = \frac{\mu c_p}{Pr} \tag{8.9}$$

c_p denoting the specific heat at constant pressure, and c_v at constant volume, are constant for a caloric perfect gas, and both can be calculated from $c_v = R/(\gamma - 1)$ and $c_p = \gamma c_v$ respectively, where $\gamma = 1.4$, and R is the gas constant, which for air is given as $287(J/kgK)$. To close the system of equations Eq. 8.1, the pressure p must be coupled to the state vector W. Depending on the model used to describe the thermodynamic properties of the gas, a coupling is defined. Such coupling is given for a caloric perfect gas through the following relation :

$$p = \rho e(\gamma - 1) = \rho c_v T(\gamma - 1) = \rho RT \tag{8.10}$$

where e denotes the internal energy which is related to the total energy by :

$$e = E - \frac{1}{2}\left(u^2 + v^2 + w^2\right) \tag{8.11}$$

8.3 Numerical Method

In this section we introduce the numerical method utilized to descritize the compressible Navier-Stokes equations 8.1.

8.3.1 Finite Volume Method

For any conservative law represented by a Partial Differential Equation PDE the spatial domain is sub-divided into finite volumes Ω. The following step is to take the volume integral over each cell. For the Navier-Stokes equations given in Eq. 8.1 this integration gives :

$$\int_{\Omega} \frac{\partial W}{\partial t} dV + \int_{\Omega} divH \; dV = 0 \tag{8.12}$$

where $H = (f - f_v, g - g_v, h - h_v)$ denotes the flux tensor.

Then by using the divergence theorem which relates a tensor field to its flux we obtain :

$$\int_{\Omega} \frac{\partial W}{\partial t} dV + \oint_{\Gamma} H \cdot n dS = 0 \tag{8.13}$$

where n denotes the unit normal vector pointing in the outward direction of the volume, or cell Ω at its boundary Γ.

Using the following assumptions we obtain Eq. 8.14

1. At each cell center the state vector W is approximated by its average over the cell.
2. For a constructed cubic grid each cell i, j, k has six neighbors.

$$\frac{d}{dt} \left(V_{i,j,k} W_{i,j,k} \right) + Q_{i,j,k} = 0 \tag{8.14}$$

where $V_{i,j,k}$ denotes the cell volume, and $Q_{i,j,k}$ the net flux, which in turn is given by :

$$Q_{i,j,k} = h_{i+1/2,j,k} - h_{i-1/2,j,k} + h_{i,j+1/2,k} - h_{i,j-1/2,k} + h_{i,j,k+1/2} - h_{i,j,k-1/2} \tag{8.15}$$

where $h_{i-1/2,j,k}$ is defined as the flux oriented in the I-direction through the cell side w (west) ; the same definition applies for all the other sides, see Fig. 8.1. The minus sign before the terms $h_{i-1/2,j,k}$, $h_{i,j-1/2,k}$ and $h_{i,j,k-1/2}$ is due to the fact that the surface normals at these sides of the cell are pointing in the inward direction.

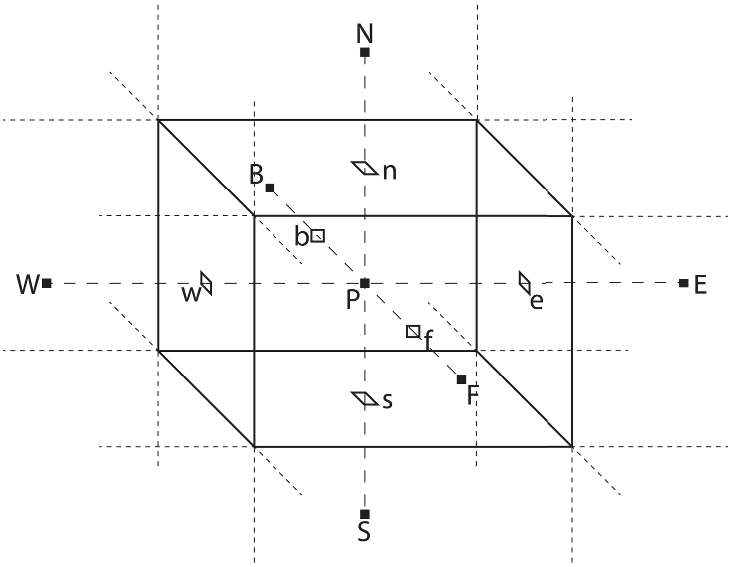


FIGURE 8.1 – Notation of a Finite Volume cell.

The flux at a cell side $h_{i-1/2,j,k} = h_w$ is approximated via :

$$h_{i-1/2,j,k} = \int_w H \cdot ndS = H_w \cdot \int_w ndS \tag{8.16}$$

8.3.2 Temporal and Spatial Discretization Scheme

Spatial Scheme

The main scheme used for the compressible air flow solver is the central scheme. The inviscid flux is treated using the central scheme with matrix dissipation. According to [48] the inviscid flux tensor H^I at the side w is reconstructed through :

$$H^I_w = H^I\left(\frac{1}{2}\left[W_{i,j,k} + W_{i-1,j,k}\right]\right) \tag{8.17}$$

The central scheme being non stable, is augmented by the addition of artificial dissipation. Both adaptive scalar and matricial dissipation schemes are developed within NSMB [13]. A second order artificial viscosity is used near discontinuities, and a forth order dissipation term is used to suppress odd-even oscillations.

After addition of the dissipative terms, Eq. 8.14 takes the following form :

$$\frac{d}{dt}\left(V_{i,j,k}W_{i,j,k}\right) + Q_{i,j,k} - D_{i,j,k} = 0 \tag{8.18}$$

where D is the dissipation operator, and is split as :

$$D_{i,j,k} = d_{i+1/2,j,k} - d_{i-1/2,j,k} + d_{i,j+1/2,k} - d_{i,j-1/2,k} + d_{i,j,k+1/2} - d_{i,j,k-1/2} \tag{8.19}$$

where each dissipative flux is evaluated at the same location as the convective fluxes. The dissipative participation at the west side of cell i, j, k is given by the following form :

$$d_{i-1/2} = r_{i-1/2} \left[\epsilon^{(2)}_{i-1/2} \left(W_i - W_{i-1} \right) - \epsilon^{(4)}_{i-1/2} \left(W_{i+1} - 3W_i + 3W_{i-1} - W_{i-2} \right) \right] \tag{8.20}$$

where r is a scaling factor relating the dissipative flux to the magnitude of the convective flux through this cell side. The terms $\epsilon^{(2)}$ and $\epsilon^{(4)}$ are used to control the dissipative fluxes locally, and are defined as :

$$\epsilon^{(2)}_{i-1/2} = k^{(2)} \nu_{i-1/2} \epsilon^{(4)}_{i-1/2} = max \left(0, k^{(4)} - \epsilon^{(2)}_{i-1/2} \right) \tag{8.21}$$

The parameters $k^{(2)}$ and $k^{(4)}$ are constants used to control the dissipation, having typical values of $k^{(2)} = 1.0$ and $k^{(4)} = 0.05$. The sensor ν used as a switch is given by :

$$\nu_{i-1/2} = max \left(\mu_{i-1} \, , \, \mu_i \right) \tag{8.22}$$

where μ relates the second order dissipation to the normalized second order difference of the pressure gradient, and is written as :

$$\mu_{i-1/2} = \left| \frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}} \right| \tag{8.23}$$

Expressions for other dissipative flux participations ($d_{i+1/2}, d_{j-1/2}$...etc) are obtained analogous to previous equations.

The viscous flux tensor H^V given in Eq. 8.4 uses the gradients calculated at the surface center, by using the Green's theorem and constructing an additional shifted control volume [83, 13]. The method is presented in Appendix 14.3.5

Temporal Scheme

The temporal scheme used in the airflow solver in Level-Set framework is currently limited to explicit time integration methods. The discretization obtained in Eq. 8.14 represents a set of ordinary differential equations. If the volume of each cell is assumed constant Eq. 8.14 can be written as :

$$\frac{d}{dt} \left(W_{i,j,k} \right) + \frac{1}{V_{i,j,k}} Q_{i,j,k} = 0 \tag{8.24}$$

In Icing simulation the steady state solution of the airflow is sought. Consequently, an explicit multistage Runge Kutta time stepping scheme is chosen. This class of methods is of high order accuracy in time, but provides stability and damping properties as well.

The form of this multistage scheme for Eq. 8.24 yields :

$$\begin{array}{rcl} W_{i,j,k}^{n+1/k} & = & W_{i,j,k}^{n/k} - \beta\alpha_1 \frac{\Delta t_{i,j,k}}{V_{i,j,k}} \left[Q\left(W_{i,j,k}^{n+0/k}\right) - D\left(W_{i,j,k}^{n+0/k}\right)\right] \\ W_{i,j,k}^{n+2/k} & = & W_{i,j,k}^{n/k} - \beta\alpha_2 \frac{\Delta t_{i,j,k}}{V_{i,j,k}} \left[Q\left(W_{i,j,k}^{n+1/k}\right) - D\left(W_{i,j,k}^{n+1/k}\right)\right] \\ \vdots & & \vdots \\ W_{i,j,k}^{n+1} & = & W_{i,j,k}^{n/k} - \beta\alpha_k \frac{\Delta t_{i,j,k}}{V_{i,j,k}} \left[Q\left(W_{i,j,k}^{n+(k-1)/k}\right) - D\left(W_{i,j,k}^{n+(k-1)/k}\right)\right] \end{array} \quad (8.25)$$

where α_k are Runge Kutta scheme coefficients, and β is the CFL number. for a four stage scheme where k=4 the values of α_k are given as :

$$\alpha_1 = 1/4,\ \alpha_2 = 1/6,\ \alpha_3 = 1/2,\ \alpha_4 = 1 \quad (8.26)$$

8.3.3 Boundary Conditions

In NSMB Boundary Conditions are usually imposed using ghost cells, which are relevant when using body fitted meshes. The values of the state vector W in these cells are evaluated using the prescribed boundary conditions. For Free-stream or Far-field boundary conditions the physical values at the ghost cells are known ; however, for solid walls these values are extrapolated from neighboring cells. For block connectivity it is assumed that the coordinates are continuous across the interface, which means that the values at the ghost cells of a block are directly communicated from its neighboring block. A maximum of two ghost cell layers are used to limit communication times, and all discretization schemes are adapted to a stencil with two neighboring cells.

In the Level-Set framework, the grid is not body-fitted. Consequently, Wall boundary conditions, Dirichlet or Neumann type can't be directly applied at wall points utilizing ghost cells(since the wall is given implicitly by the LS function). This subject is explained in the following chapter.

8.4 Chimera Method

Chimera methods were first developed in NSMB to study the movement of a sphere in a vertical tube [28]. First, superpositions are determined between super imposed blocks. Superpositioned cells are cells on a higher level block "A" inclused in a parallelepiped formed by the cells on a lower level block "B". Secondly, the tetrahedral, or triangle in 2D, including the superpositioned cell is determined.

The transfer of the state vector W is done by interpolation. The numerical precision depends on the precision of the interpolation. Three interpolation methods were developed in NSMB : tetravolumic, distance inverse, and trilinear interpolation.

The tetravolumique interpolation is based on a weighed average, averaging the state vector from the three enclosing cells, the weight being equal to the volume of the tetrahedral between the superpositioned cell and two neighboring cells. the inverse distance interpolation is also based on a weighed average, where the weight is equal to the inverse of the distance up to a prescribed power, usually 2. The advantages of this interpolation is the ability to use any preferred number of points to average from, with a minimum of 8 in 3D and 4 points in 2D. The final method, the trilinear interpolation, is composed of three separate linear interpolations following the three principal directions. Its highly precise but difficult to implement and highly costly on the memory.

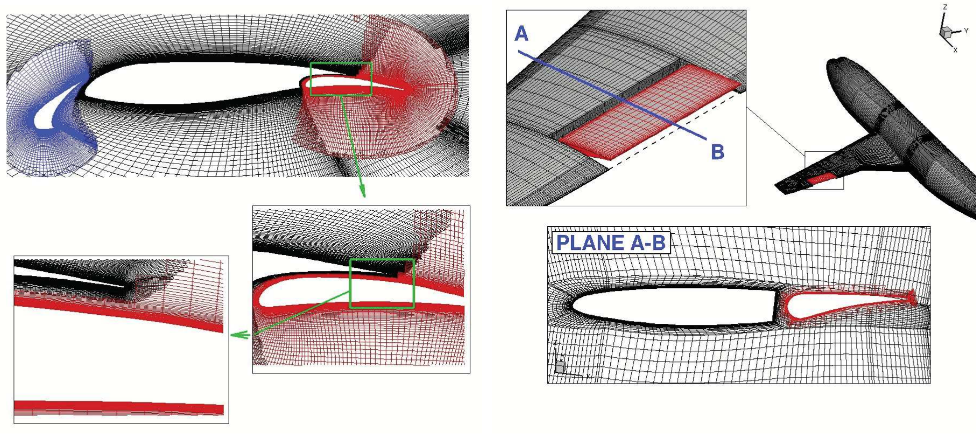


FIGURE 8.2 – Chimera Grids Used within NSMB.

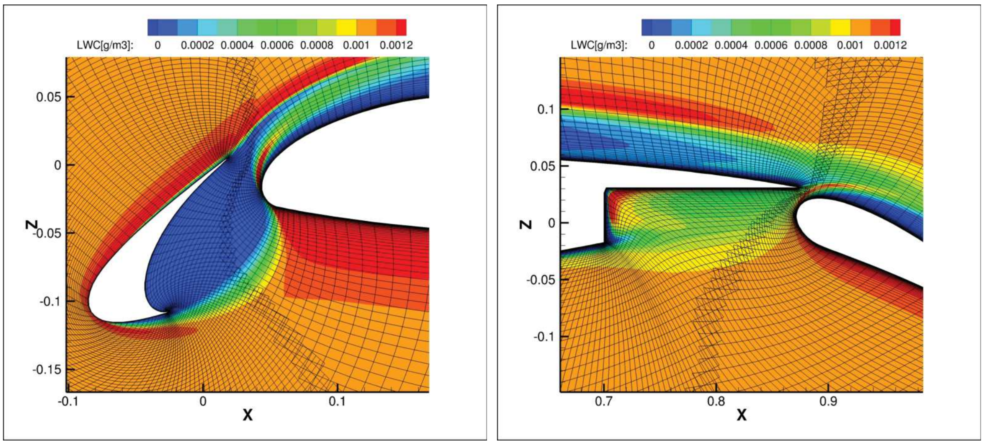


FIGURE 8.3 – Liquid water content simulated on the high-lift multi-element airfoil using chimera grids.

8.5 Turbulence Model : Spalart-Allmaras

In the field of numerical simulation in aerodynamics, Navier-Stokes equations are usually coupled with the one-equation turbulence closure model, The Spalart-Allmaras model. Through averaging the Navier-Stokes equations, and simplifying the turbulent terms the turbulent shear stress is defined as a function of a turbulent viscosity. Spalart-Allmaras equation models the transport of the turbulent viscosity. Through many empirical parametrization, dimensional analysis, Galilean invariance and selective dependence on the molecular viscosity, the equation takes the form 8.27 :

$$\underbrace{\frac{D\tilde{\nu}}{Dt}}_{convection} = \underbrace{c_{b1}\tilde{S}\tilde{\nu}}_{production} + \underbrace{\frac{1}{\sigma}[\nabla\cdot((\nu+\tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2]}_{diffusion} - \underbrace{c_{w1}f_w(r)(\frac{\tilde{\nu}}{d})^2}_{dissipation} \tag{8.27}$$

where d is the distance to the wall, denoted usually as y_w, and coincidently, in the Level-Set framework, is equal to ϕ, the signed distance function. The eddy viscosity is defined as

$$\mu_t = \rho\tilde{\nu}f_{v1} \equiv \rho\nu_t \tag{8.28}$$

Where f_{v1} is a damping function used to fit $\tilde{\nu}$ in the log layer, buffer layer and viscous sublayer and is defined as :

$$f_{v1} = \frac{\lambda^3}{\lambda^3 + {c_{v1}}^3} \tag{8.29}$$

$$\lambda \equiv \frac{\tilde{\nu}}{\nu} \tag{8.30}$$

The function $\tilde{S}$ in the production term is modified as to maintain its log-layer behavior up to the wall :

$$\tilde{S} = S^{1/2} + \frac{\tilde{\nu}}{(\kappa d)^2}f_{v2} \tag{8.31}$$

where κ is the Von Karman constant whose value is $\kappa = 0.41$. The function f_{v2} is given by :

$$f_{v2} = 1 - \frac{\lambda}{1 + \lambda f_{v1}} \tag{8.32}$$

The destruction/dissipation term should diminish in the outer region of the boundary layer. Spalart-Allmaras proposed the following formula :

$$f_w(r) = g[\frac{1 + c^6{}_{w3}}{g^6 + c^6{}_{w3}}]^{1/6} \tag{8.33}$$

where r is defined as :

$$r = \frac{\tilde{\nu}}{(\kappa d)^2\tilde{S}} \tag{8.34}$$

The value of r and $f_w(r)$ in the log layer is set to one, no effect on the destruction term, and then they decrease rapidly in the outer region. the function g is a limiter used to prevent large values of f_w and is given by :

$$g = r + c_{w2}(r^6 - r) \tag{8.35}$$

The different constants of the Spalart-Allmaras model are :

$$\begin{array}{llll} c_{b1} = 0.1355, & c_{b2} = 0.622, & c_{w2} = 0.3, & c_{v1} = 7.1 \\ \sigma = \frac{2}{3}, & c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{(1+c_{b2})}{\sigma}, & c_{w3} = 2 & \end{array} \tag{8.36}$$

Chapitre 9

Level-Set Icing Framework

In the previous chapter we explained how the airflow solver can be obtained for a body fitted mesh. In earlier chapters we also discussed how the droplets can be transported using the Eulerian formulation, as well as having an adaptive boundary condition that depends on the state variable. In this chapter we discuss how to treat these steps in a Level-Set framework, where the grid is not necessarily body-fitted, and the body being given by an implicit function. We follow this by discussing other aspects relating to consequent steps and developing the discretized form of the LS equation system Eq. 5.20.

9.1 Penalized Compressible Flow

In Level-set framework the solid body is given implicitly by its characteristic function. The boundary condition is applied on the continuous form of the Navier-Stokes equations as explained in section 7.1. Solid bodies are treated as porous media, with impermeable porosity. A no slip boundary condition on the velocity field, and a Dirichlet, Neumann, or mixed boundary condition on the temperature are applied by the addition of a penalization term to decouple the solid part. An impermeable solid is one whose porosity tends to zero, thus a very high penalization parameter $1/\eta >> 1$ is applied. The solid body is given by a characteristic function χ_{s_i} given in Eq. 5.4. The penalized compressible NS equations take the form :

$$\frac{\partial}{\partial t}(W) + \frac{\partial}{\partial x}(f - f_v) + \frac{\partial}{\partial y}(g - g_v) + \frac{\partial}{\partial z}(h - h_v) + f_p = 0 \tag{9.1}$$

where f_p is the forcing penalty applied to each conservation equation and is given by :

$$f_p = \begin{pmatrix} 0 \\ \frac{1}{\eta}\sum_i^{NS} \chi_i \rho \left(\mathbf{u}_a - \mathbf{u}_i\right) \\ \frac{1}{\eta}\sum_i^{NS} \left[\theta_i \chi_i \rho \left(\epsilon(T_a) - \epsilon(T_i)\right) + \chi_i \rho \left(\mathbf{u}_a - \mathbf{u}_i\right)\right] \end{pmatrix} \tag{9.2}$$

For an explicit time scheme, we first find the solution W^{n+1} from the Runge Kutta multistage scheme Eq. 8.25. Then we correct the solution by adding the forcing term obtaining (example given here for the second component of the state vector) :

$$\rho u_x = (\rho u_x)^{rk} + \frac{1}{\eta}\sum_i^{NS} \chi_i \rho\left(\mathbf{u}_x - \mathbf{u}_{sx}\right) \tag{9.3}$$

by rearranging the sought variable, ρu_x, we obtain :

$$\rho u_x = \frac{1}{1 - \frac{1}{\eta}\sum_i \chi_i}\left[(\rho u_x)^{rk} - \frac{1}{\eta}\sum_i^{NS} \chi_i \rho \mathbf{u}_{s_x}\right] \tag{9.4}$$

The penalization method proves useful in many flow situations. In icing however, it was found lacking, as will be discussed in the following chapters. To summarize the problem, penalization diffuses the interface to neighboring cells through the smoothed Dirac Delta or Heaviside function, which in turn imposes a miscalculation at the very first cells layer outside the wall. Icing is very sensitive to this discrepancy. The state vector close to the wall must be accurately predicted. To alleviate this problem, two methods were proposed. The first method is by imposing the state vector close to the wall via linear interpolation, the same as going to be done for the droplets in the following section. The second more realistic method, is by using the power-law velocity profile instead of linear interpolation. The seventh power-law usually used for pipe flows was the only choice applicable without going into higher complications, such as three-dimensional integral turbulent boundary layer modeling. Other more accurate velocity profiles exist in the literature [99, 7]. However, they usually depend on the wall shear stress as well as the boundary layer thickness. These variables are difficult to estimate using the proposed embedded grid technique.

These two methods are explained as following :

Consider Fig. 9.1. The required velocity to be constructed from the velocity profile will be imposed on the cells marked with x and are henceforth called immersed boundary IB points. The procedure is summarized in the following points :

(1) First we detect this layer of points outside the interface by detecting the change of $sign(\phi)$ with neighbors. An array containing their indexes is saved IBp_{ind}.

(2) Each IB point is mirrored against the solid surface resulting in an image point (im) further from the interface at a distance D from the IB point along the normal direction. The normal direction is available from the signed-distance LS function $\mathbf{n} = \nabla\phi$. The distance D is chosen/calculated from the cell size to ensure that the image point (im) is close to a fluid cell and outside the IB point.

(3) The coordinates of these image points are calculated via Eq. 9.5 and are saved in an array $\overrightarrow{X}_{im}$.

$$\overrightarrow{X}_{im} = \overrightarrow{X}_{IP} + \nabla\phi_{IP} * D \tag{9.5}$$

(4) The distance to the wall at these image points, which is the same as the LS function

at these image points, is calculated via Eq. 9.6 and saved in a third array ϕ_{im}.

$$\phi_{im} = \phi_{IP} + D \tag{9.6}$$

(5) An algorithm searches for the fluid cell closest to the image point. The indices of the closest point are saved in an array CP_{ind}.

(6) This closest point is used to interpolate the state vector $\mathbf{Q}$ to the image point. The gradient of the state vector is calculated at our list of closest points $\nabla\mathbf{Q}_{cp}$ and is then used to interpolate the state vector $\mathbf{Q}_{cp}$ to the image point as given in Eq. 9.7.

$$\mathbf{Q}_{im} = \mathbf{Q}_{cp} - \nabla\mathbf{Q}_{cp} \cdot (D\nabla\phi) \tag{9.7}$$

(7) The required velocity is applied through linear interpolation or power-law velocity profile.

(8) If and only if the Level-Set is moved by advection, resulting in a new interface, the arrays IBp_{ind}, $\overrightarrow{X}_{im}$, ϕ_{im}, and CP_{ind} are re-evaluated.

Using linear interpolation, we interpolate between the state vector at the image point

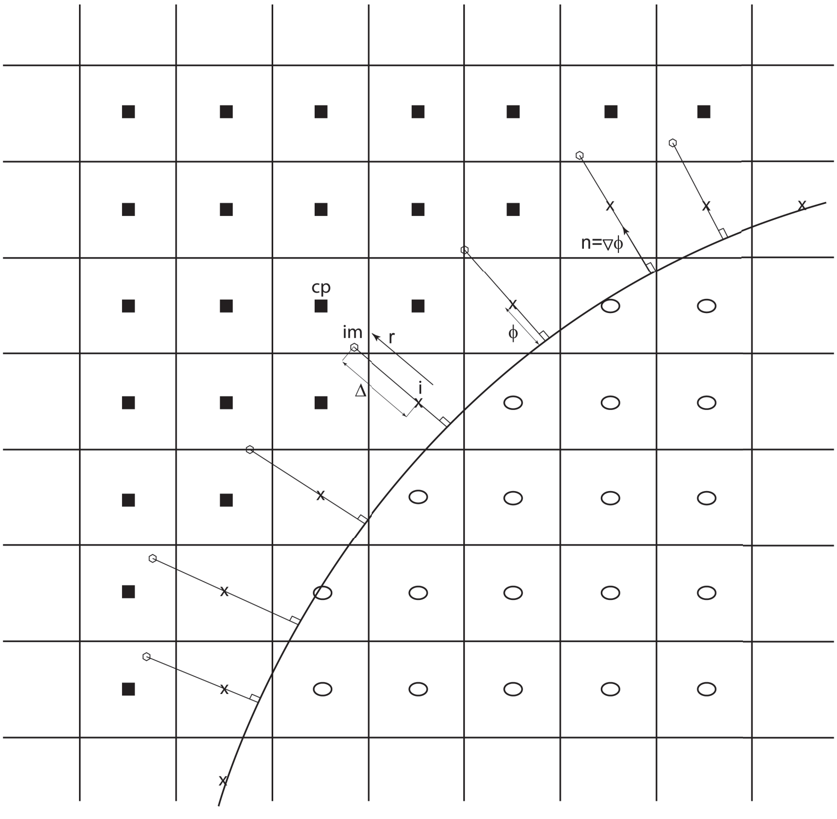


Figure 9.1 – Schematics of IBM-LS framework. ■ : flow domain, x : IB cells layer, ○ : solid domain.

and a no slip BC at the wall to obtain :

$$\begin{array}{lll} \mathbf{u}(\phi=0)=0 & \Longrightarrow \mathbf{u}_{IB} & =\frac{\phi_{IB}}{\phi_{im}}\mathbf{u}_{im} \\ \frac{\partial P}{\partial n}=0 & \Longrightarrow P_{IB} & =P_{im} \end{array} \tag{9.8}$$

Using the power-law interpolation, we follow the formulation provided by [26] and [96]. The velocity is first decomposed into its tangential and normal components. It is important to note here that the transversal velocity component that could appear inside a three-dimensional turbulent boundary layer, for swept wings as example, is not taken into account. So, the tangential component of the velocity at the image point is used to reconstruct the tangential velocity at the IB point using the seventh power law. On the other hand, the normal component is interpolated linearly. The procedure is as follows :

First the normal component of the velocity is calculated at the image point. The normal at the image point is the same as at the IB point :

$$\begin{array}{l} u_n=\mathbf{u}\cdot\overrightarrow{\mathbf{n}}=\mathbf{u}\cdot\nabla\phi \\ \mathbf{u}_n=u_n\overrightarrow{\mathbf{n}} \end{array} \tag{9.9}$$

The tangential component is then calculated

$$\mathbf{u}_t=\mathbf{u}-\mathbf{u}_n \tag{9.10}$$

Then both components are interpolated using the appropriate law :

$$\begin{array}{l} \mathbf{u}_{tIB}=\left(\frac{\phi_{IB}}{\phi_{im}}\right)^{1/7}\mathbf{u}_{tim} \\ \mathbf{u}_{nIB}=\left(\frac{\phi_{IB}}{\phi_{im}}\right)\mathbf{u}_{nim} \end{array} \tag{9.11}$$

The velocity is then the sum of both components :

$$\mathbf{u}_{IB}=\mathbf{u}_{tIB}+\mathbf{u}_{nIB} \tag{9.12}$$

9.1.1 Spalart-Allmaras Turbulence Model : IBM-LS BC

The solid body being defined by a characteristic function, was treated via penalization, or direct enforcing/interpolation of the state vector close to the wall. The zero boundary condition of $\tilde{\nu}$ at the wall requires similar treatment. In the same manner of equation 9.4, the turbulent viscosity is first solved, Eq. 8.27, using an explicit scheme, then is corrected by adding the penalization term, which gives :

$$\tilde{\nu}=\frac{1}{1-\frac{1}{\eta}\sum_i\chi_i}\left[(\tilde{\nu})^{exp}\right] \tag{9.13}$$

However, such formulation enforces very low values of the turbulent viscosity, close to zero. Since the grids used in embedded grid techniques do not respect a very small distance close to the wall, such underestimation is detrimental. Consequently, an interpolation method is utilized to ensure the value at the first cell is not equal to zero. For simplicity a linear profile is assumed, and the resulting interpolation gives :

$$\tilde{\nu}_{IB} = \left(\frac{\phi_{IB}}{\phi_{im}}\right)\tilde{\nu}_{im} \tag{9.14}$$

9.2 Eulerian Droplet Flow

The droplets flow is simulated in an Eulerian approach as presented in subsection 2.2.2 in Eq. 2.5. In this section we discuss the discretization methods available in our solver, followed by discussing the droplet adaptive boundary condition and its application in the LS framework.

9.2.1 Droplet Flow Discretization

Droplet flow fields given in Eq. 2.5 are discretized using a finite volume approach like the one used in the airflow solver. The equations can be rewritten in the following form :

$$\frac{\partial W_d}{\partial t} + \nabla \cdot (W_d \mathbf{u}_d) = S_{W_d} \tag{9.15}$$

where W_d is the droplet state vector, $\mathbf{u}_d$ the dropelt velocity, and S_{W_d} the right hand side of the droplet equations, all given by :

$$u_d = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}, W_d = \begin{pmatrix} \alpha \\ \alpha u_x \\ \alpha u_y \\ \alpha u_z \end{pmatrix}, S_{W_d} = \begin{pmatrix} 0 \\ S_{u_x} \\ S_{u_y} \\ S_{u_z} \end{pmatrix} \tag{9.16}$$

where the source terms are defined as in Eq. 2.5. By taking the volume integral over each cell in the spatial domain we obtain, for cell (i, j, k) denoted p of volume Ω :

$$\int_\Omega \frac{\partial W_d}{\partial t} d\Omega + \int_\Omega \nabla \cdot (W_d \mathbf{u}_d) d\Omega = \int_\Omega S_{W_d} d\Omega \tag{9.17}$$

Then by using the divergence theorem, assuming the volume constant, and assuming an average value at each cell center we obtain :

$$\begin{gathered} \tfrac{\partial W_d}{\partial t}_{i,j,k}\Omega_{i,j,k} + \textstyle\int_\Gamma (W_d \mathbf{u}_d) \cdot \mathbf{n} = S_{W_{d i,j,k}}\Omega_{i,j,k} \\ \tfrac{\partial W_d}{\partial t}_{p}\Omega_p + \textstyle\int_\Gamma (W_d \mathbf{u}_d) \cdot \mathbf{n} = S_{W_{d p}}\Omega_p \end{gathered} \tag{9.18}$$

the surface integral in the last equation can be written as the sum of values averaged on all faces composing the cell sides. As in the previous section the cells mesh is cubic structured and each cell p has six neighbors in 3D and 4 neighbors in 2D $(E, W, N, S,\ F, B)$, capital letters stand for the neighboring cells. The sides of cell P in contact with its neighbor are denoted by small letters $(e, w, n, s,\ f, b)$. The internal product of the velocity vector by the surface normal vector represents the outward flux and is therefore denoted F. The previous equation is rewritten as :

$$\frac{\partial W_d}{\partial t}_p + \frac{1}{\Omega_p}\sum_{n,s,e,w,f,b} W_d F = S_{W_{dp}} \tag{9.19}$$

To accelerate the convergence towards the steady state solution required in this step, an implicit first order scheme was satisfactory. The equation being nonlinear is linearized and the source term is split into an implicit term and an explicit term. The equation is written in implicit form as :

$$W_{dP}^{n+1} + \frac{\Delta t}{\Omega_p}\sum_{n,s,e,w,f,b} W_{dP}^{n+1} F + \Delta t S_{imp} = \Delta t S_{exp} + W_{dP}^{n} \tag{9.20}$$

The source term given in Eq. 2.5 is decomposed in two parts, an explicit linear part, and an implicit linearized part :

$$\begin{aligned} S_{exp} &= \alpha \tfrac{C_D Re_d}{24K}\mathbf{u}_a + \alpha\left(1 - \tfrac{\rho_a}{\rho_w}\right)\tfrac{1}{Fr^2}\mathbf{g}) \\ S_{imp} &= \alpha \tfrac{C_D Re_d}{24K}\mathbf{u}_d = \tfrac{C_D Re_d}{24K} W_d \end{aligned} \tag{9.21}$$

The spatial scheme used in NSMB is generally an Upwind difference scheme UDS, although variations of central blended scheme and total variation diminishing scheme via deffered correction are also developed. For an upwind scheme, first the flow is evaluated at the cell sides by averaging the velocity vector between the cell P being studied and its neighbor :

$$F_e = F_{i+1/2,j,k} = \frac{1}{2}\left(\mathbf{u}_{dp} + \mathbf{u}_{dE}\right)\cdot \mathbf{S}_e \tag{9.22}$$

where $\mathbf{S}_e$ is the surface normal vector at the east side of the cell P. Using analogous expressions we evaluate the fluxes at the other sides.

The choice of the averaged value of the state vector at the following time step $W_{dP}^{n+1}|_{e,w,...}$ is dependent on the direction of the flux in an upwind scheme :

$$\begin{aligned} W_{dP}^{n+1}|_e &= W_{dP}^{n+1} \quad \text{if } F_e > zero \\ W_{dP}^{n+1}|_e &= W_{dE}^{n+1} \quad \text{if } F_e < zero \end{aligned} \tag{9.23}$$

We rewrite the droplet equation in the matrix form as follows :

$$A_P W_{dP} + \sum_{nb=E,W,N,S,F,B} A_{nb} W_{dnb} = R_P \tag{9.24}$$

where A are coefficient matrices with the same size of the grid. They are generated as follows :

First we calculate the mass flow at each side using the old state vector W_{dP}^{n}. Then the upstream direction is checked and the flux is added to the appropriate coefficient. The implicit source term is then added to the coefficient. The right hand side R_P term is composed of the explicit source term S_{exp} and the state vector at the previous time step, as given in Eq. 9.20.

Under this configuration, A_P and A_{nb} coefficient matrices are fed to the Sparskit library which is integrated within NSMB. It offers a vast choice of linear solvers : GMRES, BiCGSTAB, CGSTAB,..., and many other choices on the preconditioner : ILUT, ILU0,...etc.

To increase convergence stability an under-relaxation can also by applied. A relaxation coefficient $0 < \omega < 1$ is used when updating the state vector.

$$W_{dP}^{n+1} = (1-\omega)W_{dP}^{*} + \omega W_{dP}^{n} \tag{9.25}$$

where W_{dP}^{*} is the state vector obtained from the linear solver, the solution of Eq. 9.24.

Since the droplet equations are solved implicitly, and only the steady state solution is sought, a local time stepping scheme is utilized. The local time step for cell P is :

$$\Delta t_P = \Omega_P \frac{CFL_{imposed}}{F_{max}} \tag{9.26}$$

where $F_{max} = max\,(F_x, F_y, F_z)$, and the directional fluxes are averaged from the sides to the cell center via :

$$F_x = \frac{1}{2}(F_e + F_w) \tag{9.27}$$

The $CFL_{imposed}$ is chosen by the user, and since the implicit scheme guarantees unconditional stability, its value can be higher than one. The value of the local time step Δt_P is limited between a lower limit, $\Delta t_P > 10^{-6}$, and a higher limit chosen by the user, depending on the geometry and the coarseness of the grid in use ; the limits have nothing to do with the CFL, and are used to ensure positive definite physical values.

9.2.2 Eulerian Droplet Boundary Condition

The droplet state vector is firstly initialized. The nondimensionalized water volume fraction α is set equal to one. The dimensionless velocity vector $\mathbf{u}_d$ is set equal to the nondimensionalized air velocity vector at the farfield "freestream". At the edges of the domain, two boundary condition types are applicable : inlet and outlet. For inlet boundary condition a standard Dirichlet condition is applied, and at outlet a Neumann boundary condition is applied :

$$\begin{aligned} W_d|_i &= \begin{pmatrix} \alpha \\ \alpha \mathbf{u}_d \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha \frac{\mathbf{u}_a}{U_\infty} \end{pmatrix} \\ \frac{\partial W_d}{\partial n}|_o &= \begin{pmatrix} \frac{\partial \alpha}{\partial n} \\ \frac{\partial \alpha \mathbf{u}_d}{\partial n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned} \tag{9.28}$$

At the inlet the required free stream value is applied at the ghost cell. In the outlet region the state vector is extrapolated from physical cells to the ghost cells to restrain a zero gradient at the interface.

Since we seek a steady state solution of the Eulerian system the droplets are assumed to impinge on the solid body and disappear inside, a wet zone is formed in the region where the droplet impinge, and a dry zone is shaded further downstream, Fig. 9.2. This type of boundary condition is dependant on the velocity vector as explained in Eq. 2.7. In a body fitted grid, such boundary condition is applied directly using the ghost cells method.

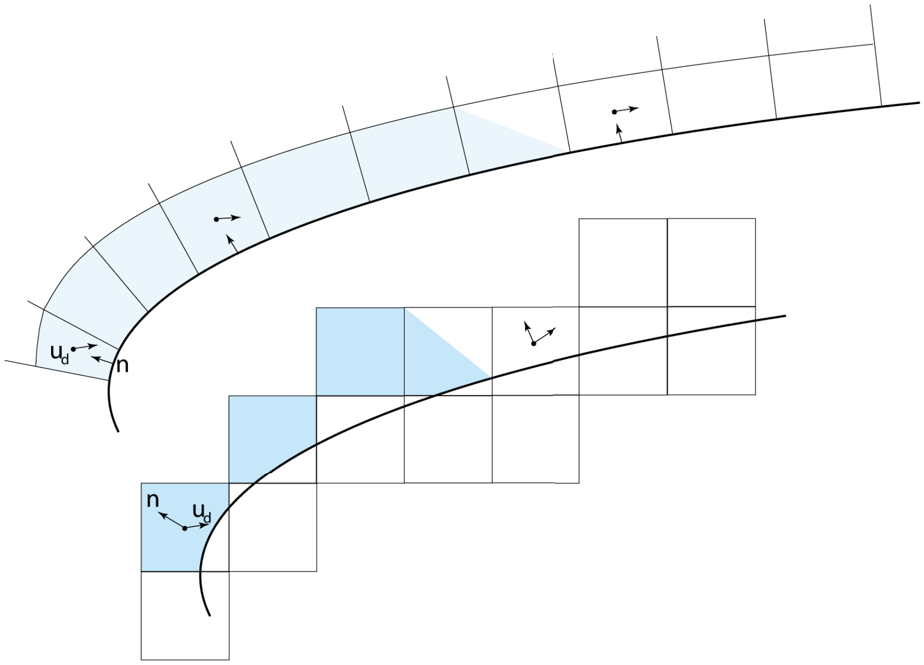


FIGURE 9.2 – Eulerian droplet field adaptive boundary condition.

9.2.3 BC Through Penalization ? "Implicit Forcing"

In a Level-Set framework the solid body is not defined by the grid, and thus no wall points exist where one can apply the boundary condition as with a body fitted grid. The airflow no slip boundary condition was applied through penalization, a source term added to the continuous differential equation. The droplets however have a unique adaptive boundary condition, at some points, wet zones, the boundary condition is of Neumann type, and at other points, dry zones, it is of Dirichlet type. The gridded domain in the LS framework is the whole domain, solid and liquid parts are both meshed. The conservation equations given in Eq. 2.5 conserve quantities between inlets and outlets. The wet zone of the solid interface should act as an outlet. In other words, conservation should be lost at the interface in the wet zone. However no information should be propagated from this zone to the flow, a zero gradient is required. We could not find any logical penalty term that could achieve this behaviour : *loss of mass at the solid side without affecting the physical values at the fluid side, coupled with no introduction of the lost mass downstream.*

Different proposals were tested but all failed. The most pronounced proposal was to introduce a real solid inside (body fitted) to act as the absorbant as seen in Fig. 9.3. The droplet flow is not penalized ; the droplets are free to pass through the LS zero interface and continue their passage in real solid defined by the negative LS zones. The droplets get absorbed on the additional wall defined inside the real wall. On wet zones of this fictious wall, an outlet BC is applied, and at dry zones a Dirichlet BC is applied. Results on simple configurations are shown in Fig. 9.4. The main problem with this method is manual work required to generate the body fitted mesh for complex configurations. Another problem is the discontinuity region between the wet and dry zones, where a fictious solid hump appears and perturbes the flow.

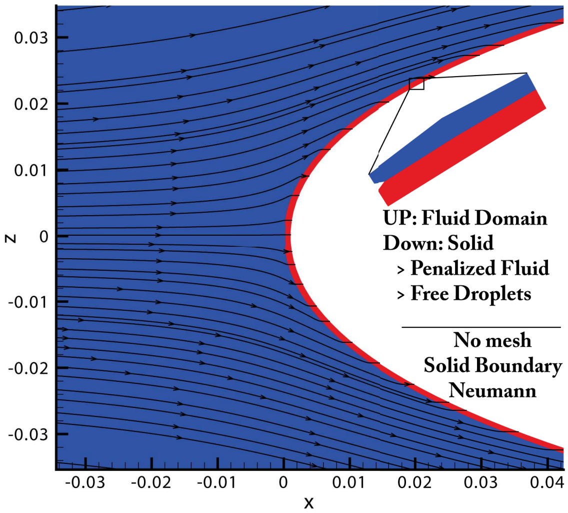


FIGURE 9.3 – Droplet Flow Penalization by introducing a real solid acting as outlet.

9.2.4 BC Through Explicit Forcing : IBM-LS

To enforce the droplets adaptive boundary condition, the interface given by the LS function can be transformed into an explicit list of points, through which direct forcing of the required boundary condition is achievable on the discretized form. An explicit forcing of the boundary condition using a IBM representation of the LS implicit function can easily achieve this adaptive BC. Consequently, we use the first layer of cells adjacent to the interface to directly enforce the required boundary condition.

Following the same procedure as in section 9.2.4, we construct the droplet state vector. The required BC is imposed by checking the droplet velocity component normal to the wall. Consequently, we impose Neumann condition for the wet zones, and Dirichlet condition for the dry zones

$$\begin{aligned} \mathbf{u}_d \cdot \mathbf{n} < 0 \quad &\Longrightarrow \quad \mathbf{Q}_{IB} = \mathbf{Q}_{im} \\ \mathbf{u}_d \cdot \mathbf{n} > 0 \quad &\Longrightarrow \quad \mathbf{Q}_{IB} = \mathbf{Q}_{im}\frac{\phi_{IB}}{\phi_{im}} \end{aligned} \tag{9.29}$$

This method was also developed using two image points to better apply the Neumann BC on the wall. Another variation of the method was also developed, where we reconstruct the flux at the wall instead of at the cell center as proposed by [18]. However, we do not follow the method depicted in [18], where he extrapolates from the image point to the wall without enforcing a zero gradient. We found that the first variation presented here was more stable than these two.

9.2.5 *General Droplets Algorithm*

After the coefficient matrices A and the right hand side R, of Eq. 9.24, are constructed from the fluxes and the source terms, the inlet and outlet boundary condition are applied on the ghost cells. Following that, the values at the wall points Q_{IB} are evaluated as explained. Then the coefficient A and the right hand side R are directly modified at each

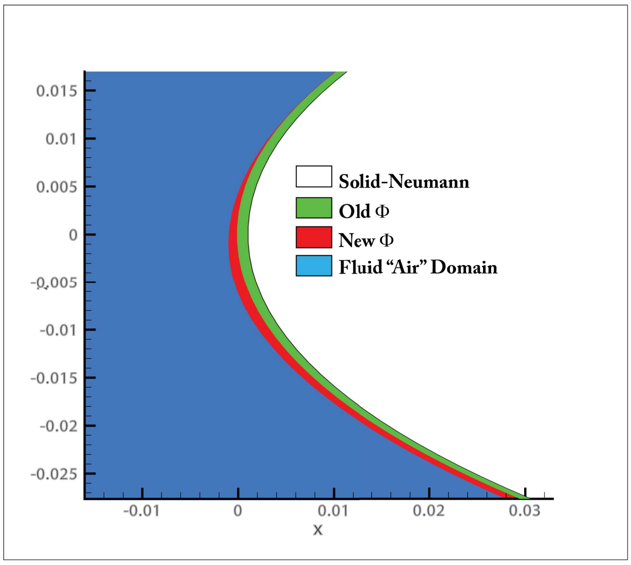


Figure 9.4 – Droplet Flow Penalization by introducing a real solid acting as outlet : resulting ice shape.

wall point :

$$\begin{aligned} &A_P = 1 \\ &R_P = Q_P \\ &A_N = A_S = A_E = A_W = A_F = A_B = Zero \end{aligned} \tag{9.30}$$

Another version compatible with the PSBLAS library[PSBLAS] is also developed. In this version instead of building a matrix for the coefficient at the point and 6 other matrices for the coefficient of neibouring cells, we build three arrays ; the coefficient matrix is written in sparse form. A procedure for local numbering, then global numbering of points is first run within the MPI parallel environment. Then the nonzero values, along with their local and global coordinates are saved in different arrays. The according right hand side and previous state vector are saved in two other arrays. The PSBLAS linear solver receives these arrays ; using a selected solver, a preconditioner, and convergence criteria, the system is solved within the MPI parallel environment.

9.2.6 Droplet Collection Efficiency

The collection efficiency is directly calculated using the formula in Eq. 9.31.

$$\beta = \alpha \mathbf{u}.\mathbf{n} \tag{9.31}$$

In the IBM-LS configuration this variable can be calculated at either the first layer of cells outside which are accessible through the array WP_{ind}, or at the cells whose Dirac Delta $\delta(\phi)$ is higher than zero and whose LS function ϕ is positive. The collection efficiency is to be used later on to evaluate the ice accretion mass rate.

If the first method is used, where we calculate the collection efficiency at first layer of cells outside the wall, these cells are in a staircase formation. Such formation would generate discontinuities in the ice accretion calculated later on, which would result in further discrepencies in the Level-Set formulation. This problem can be seen in Fig. 9.5.

In the second method this problem is alleviated by the use of the smoothed Dirac Delta $\delta(\phi)$. The conventional approach is to multiply the resulting collection efficiency β by $\delta(\phi)$, ensuring an adaptive model. However, β would be underestimated, since $\delta(\phi)$ is always lower than one. Add to that there is no physical value of β inside the solid, for which $\delta(\phi)$ is still higher than zero. Hence, we check for points whose $\delta(\phi)$ is higher than zero and whose ϕ is positive, and evaluate the collection efficiency β at these cells. Results obtained with such method are shown in Fig. 9.6.

9.3 Icing Thermodynamics in Level-Set Framework

The thermodynamic model developed in NSMB is based on the work of [14], [9] called the Shallow-Water for Ice Modelling, SWIM. The SWIM module is based on conservation equations of mass and energy. The interested reader can refer to [81] where the implementation of the model is fully explained.

In a Level-Set framework where there is no skin mesh, such model is not easily applicable. Consequently, only dry rime ice configurations are being studied so far, where the impinging mass is assumed to solidify on impact. Thus, resultant mass rate of ice accretion $\dot{m}_{ice}$ is evaluated through Eq. 9.32.

$$\dot{m}_{ice} = \dot{m}_{imp} = LWC \cdot V_\infty \cdot \beta \tag{9.32}$$

The variable LWC represents the liquid water content and V_∞ the free stream or far fields velocity.

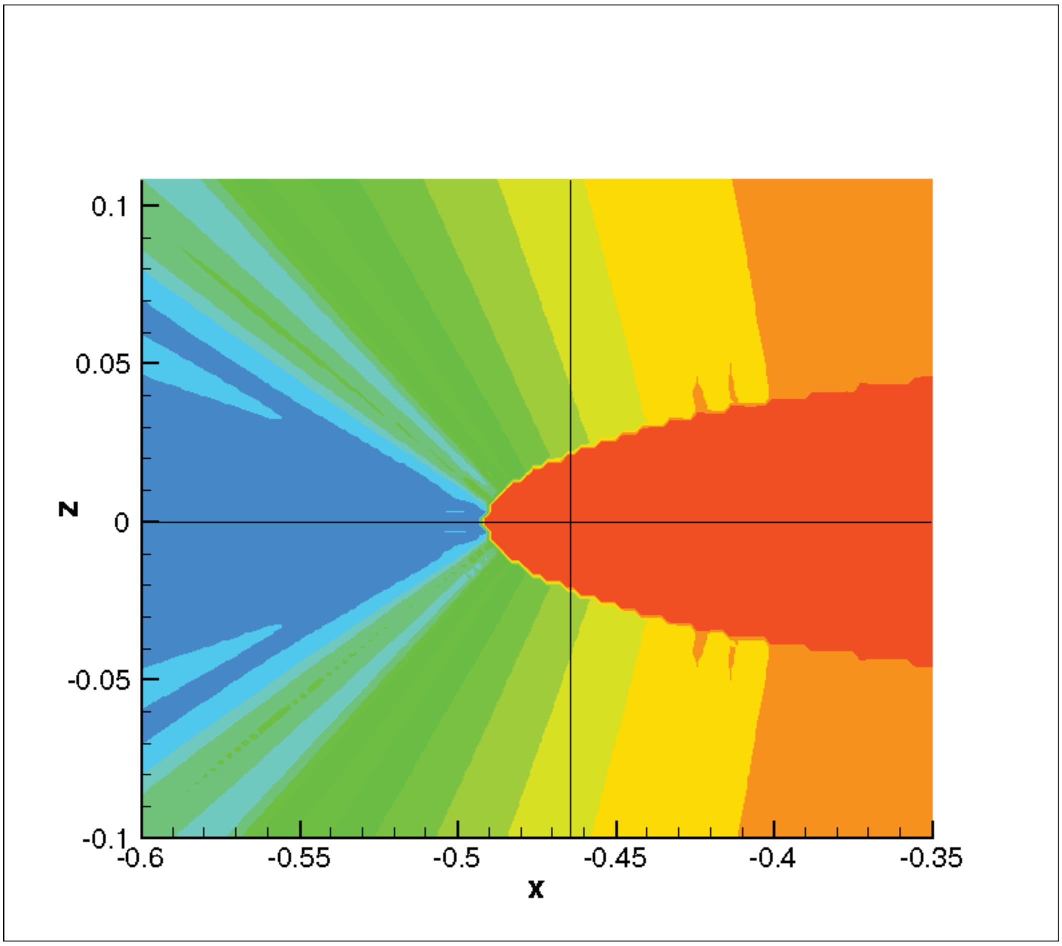


FIGURE 9.5 – Discontinuities in the resulting icing velocity caused by staircase evaluated β.

9.4 Level-Set Equations

As introduced in section 5.4, the Level-Set equations written in the general form in Eq. 5.20, after being integrated and after the cell averaging, were rewritten as :

$$\frac{\partial \psi}{\partial t} V + \sum_{f:n,s,e,w,f,b} (\overrightarrow{\mathbf{U}} \psi)_f \cdot \mathbf{S_f} = cV + \psi \sum_{f:n,s,e,w,f,b} \overrightarrow{\mathbf{U}}_f \cdot \mathbf{S_f} \tag{9.33}$$

9.4.1 Spacial Discretization Scheme

The value of the flux at the cell side $\overrightarrow{\mathbf{U}}_f \cdot \mathbf{S_f}$ is evaluated first. For the Level-Set advection equation, the icing velocity is averaged to evaluate the flux at the edge :

$$F_e = \overrightarrow{\mathbf{U}}_e \cdot \mathbf{S_e} = (\mathbf{v}_{iceP} + \mathbf{v}_{iceE}) \cdot \mathbf{S_e} \tag{9.34}$$

where $\mathbf{v}_{ice}$ calculated at the wall via Eq.5.6 is first propagated to the whole domain before being used here. For the Level-Set reinitialization equation, no reconstruction at the edge is required by averaging, $\overrightarrow{\mathbf{U}} = sign(\phi_0)\nabla\phi/|\nabla\phi|$. The value of the sign function is not to be averaged. A clear mistake. Each cell on both sides of the edge has its own sign

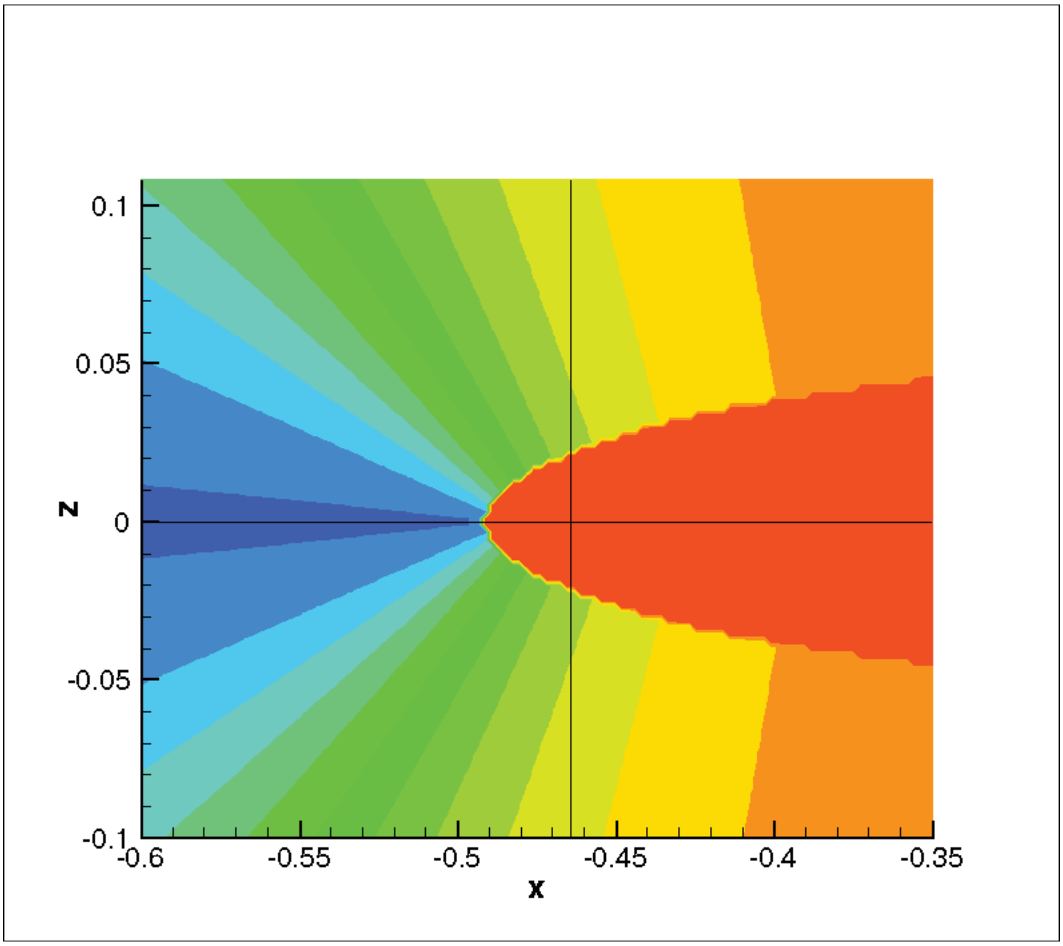


FIGURE 9.6 – Continuous resulting icing velocity resulted from continuous evaluated β.

function and thus its own flux evaluation. The gradient is calculated directly at the cell face, by constructing an additional shifted control volume as shown in appendix 14.3.5 :

$$F_e = \overrightarrow{\mathbf{U}}_e \cdot \mathbf{S_e} = sign(\phi_0)_P \frac{\nabla \phi_f}{|\nabla \phi|_f} \tag{9.35}$$

Similarly for the ice propagation equation, no reconstruction is required by averaging, the flux takes a similar form to that of the reinitialization equation :

$$F_e = \overrightarrow{\mathbf{U}}_e \cdot \mathbf{S_e} = sign(\phi)_P \nabla \phi_f \tag{9.36}$$

In order to improve the interface description and reduce mass losses , we implemented the WENO5 scheme for spatial discretization. The basic idea of the WENO5 scheme is to use smoothing and weighting techniques. Unlike its non-conservative form, the conservative WENO5 scheme is not applied to first-order derivatives like in finite difference scheme, but directly to the values of the variables to be reconstructed at the edge (ψ in our case).

For the stencil shown in Fig. 9.7, for cell (i, j, k), through the ENO reconstruction we obtain :

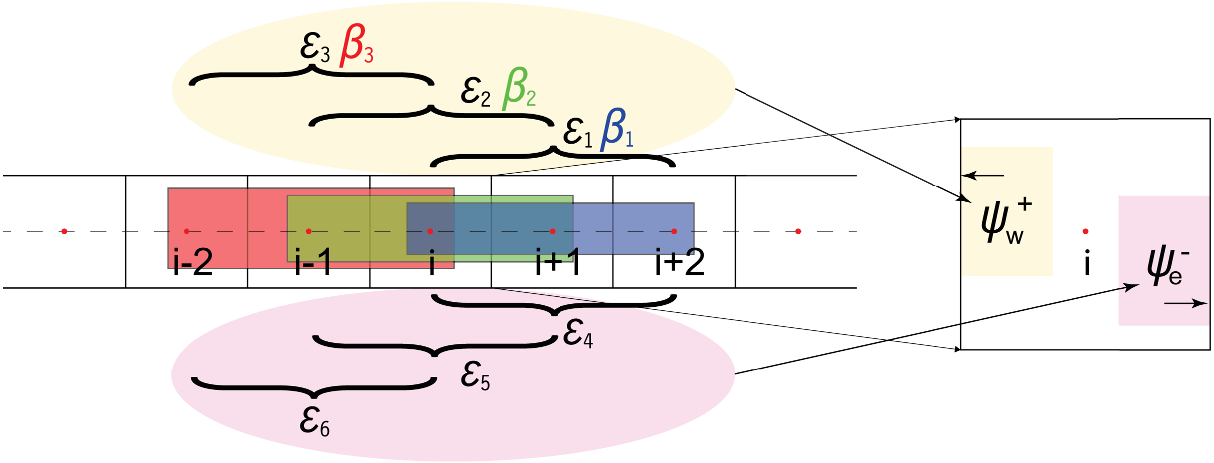


FIGURE 9.7 – WENO5 stencil used to reconstruct ψ_w^+ and ψ_e^-.

$$\begin{aligned}
\epsilon_1 &= \tfrac{11}{6}\psi_i - \tfrac{7}{6}\psi_{i+1} + \tfrac{1}{3}\psi_{i+2} \\
\epsilon_2 &= \tfrac{1}{3}\psi_{i-1} + \tfrac{5}{6}\psi_i - \tfrac{1}{6}\psi_{i+1} \\
\epsilon_3 &= -\tfrac{1}{6}\psi_{i-2} + \tfrac{5}{6}\psi_{i-1} + \tfrac{1}{3}\psi_i \\
\epsilon_4 &= \tfrac{1}{3}\psi_i + \tfrac{5}{6}\psi_{i+1} - \tfrac{1}{6}\psi_{i+2} \\
\epsilon_5 &= -\tfrac{1}{6}\psi_{i-1} + \tfrac{5}{6}\psi_i + \tfrac{1}{3}\psi_{i+1} \\
\epsilon_6 &= \tfrac{1}{3}\psi_{i-2} - \tfrac{7}{6}\psi_{i-1} + \tfrac{11}{6}\psi_i
\end{aligned} \tag{9.37}$$

Then we evaluate the smoothness indicators β :

$$\begin{aligned}\beta_1 &= \tfrac{3}{12}V_1^2 + \tfrac{1}{4}V_2^2\\ \beta_2 &= \tfrac{3}{12}V_3^2 + \tfrac{1}{4}V_4^2\\ \beta_3 &= \tfrac{3}{12}V_5^2 + \tfrac{1}{4}V_6^2\end{aligned} \tag{9.38}$$

where the variables V_{1-6} are given by :

$$\begin{aligned}V_1 &= \psi_i - 2\psi_{i+1} + \psi_{i+2}\\ V_2 &= 3\psi_i - 4\psi_{i+1} + \psi_{i+2}\\ V_3 &= \psi_{i-1} - 2\psi_i + \psi_{i+1}\\ V_4 &= \psi_{i-1} - \psi_{i+1}\\ V_5 &= \psi_{i-2} - 2\psi_{i-1} + \psi_i\\ V_6 &= \psi_{i-2} - 4\psi_{i-1} + 3\psi_i\end{aligned} \tag{9.39}$$

Using the smoothness indicators, we define the nonlinear weights :

$$\omega_k = \frac{\alpha_k}{\sum_{l=1}^{3} \alpha_l} \tag{9.40}$$

with α_k defined as :

$$\alpha_k = \frac{\gamma_k}{(\eta + \beta_k)^2} \tag{9.41}$$

where η is a small positive number added to avoid devision by zero, typically having a value of $\eta = 10^{-6}$.

Depending on which side of the cell (i, j, k) we define two sets of values for the constant γ_k, and the corresponding weighted essentially non oscillatory reconstruction of the state vector ψ.

For the east side :

$$\gamma_k = \begin{bmatrix} \frac{3}{10} & \frac{3}{5} & \frac{1}{10} \end{bmatrix} \tag{9.42}$$

$$\psi_e^- = \omega_1 \epsilon_4 + \omega_2 \epsilon_5 + \omega_3 \epsilon_6 \tag{9.43}$$

and for the west side :

$$\gamma_k = \begin{bmatrix} \frac{1}{10} & \frac{3}{5} & \frac{3}{10} \end{bmatrix} \tag{9.44}$$

$$\psi_w^+ = \omega_1 \epsilon_1 + \omega_2 \epsilon_2 + \omega_3 \epsilon_3 \tag{9.45}$$

For each face two reconstructions are obtained ; for the east face of cell (i, j, k) we reconstructed ψ_e^- from the stencil shown above, and ψ_e^+ from the stencil one step to the right, right being in eastern direction. It should be noted that the same procedure is applied to the three main Level-Set equations : Level-Set advection, Reinitialization, and Propagation equations. Using the flux calculated in equations 9.34,9.35 and 9.36, we use an upwind scheme to select which reconstruction to keep :

$$\begin{cases} \psi_e & = \psi_e^- = if F_e > 0 \\ \psi_e & = \psi_e^+ = if F_e < 0 \end{cases} \tag{9.46}$$

Going back to equation 9.33, only the right hand side is still to be treated. The state vector ψ is evaluated at the cell center and multiplied by the summation of the fluxes on all sides, the fluxes are the same as those evaluated for the left hand side through equations 9.34, 9.35 and 9.36. After adding all the explicit terms together and dividing by the volume we obtain :

$$\frac{\partial \psi}{\partial t} = \mathbf{R}\left(\psi\right) \tag{9.47}$$

9.4.2 Temporal Discretization Scheme

The Level-Set equations, being spatially discretized with the WENO 5 scheme, are treated explicitly. Thus, a criterion on the time step is necessary to ensure the stability of the calculations. For this type of equation, we can obtain stability by using the CFL ("Courant-Friedrichs-Lewy") condition.

The (TVD) Runge-Kutta method is shown to outperform other explicit methods [114]. The 3rd order accurate version of the TVD Runge-Kutta discretization method is one of the most popular methods :

$$\begin{cases} \psi^1 & = \psi^n + \Delta t\, \mathbf{R}\left(\psi^n\right) \\ \psi^2 & = \frac{3}{4}\psi^n + \frac{1}{4}\psi^1 + \frac{1}{4}\Delta t\, \mathbf{R}\left(\psi^1\right) \\ \psi^{n+1} & = \frac{1}{3}\psi^n + \frac{2}{3}\psi^2 + \frac{2}{3}\Delta t\, \mathbf{R}\left(\psi^2\right) \end{cases} \tag{9.48}$$

9.4.3 Level-Set Reinitialization Local conservation

Before dealing with the boundary condition, we remind here that the reinitialization equation needs a fix, additional term to ensure local preservation of the interface. We used the fix of Sussman and Fatemi 1999 [98]. After solving the multistage 3rd order RK and obtaining the prediction of the redistanced LS function, this value is first communicated in the MPI parallel environment, then the correction proposed by [98] is applied. The integral in Eq. 5.14 or 5.15 is evaluated, in case of 2d or 3d geometry respectively, from which the function λ is also evaluated, Eq. 5.13. Then Eq. 5.12 is used to correct the state vector :

$$\phi^{n+1} = \phi^{n+1} - \Delta t\, \lambda \delta \left(\phi^0\right) \left|\nabla \phi^0\right| \tag{9.49}$$

9.4.4 Level-Set Equations Boundary Conditions

The Level-Set equations do not require any special boundary condition treatment. The only type is an interpolation boundary condition. At walls, inlets, or outlets the LS function ϕ is interpolated from interior cells to the ghost cells. These are applied to not impose any information on ϕ. The only source of change in ϕ or $\mathbf{v}_{ice}$ exists on the interface and propagates from there to the flow domain. At block connectivity treated by the MPI parallel environment, the values at the ghost cells are available from neighboring blocks. The only problem is the size of the stencil. In NSMB communication is restricted to two ghost cells. The WENO5 stencil needs three cells to reconstruct the state vector at the face of the block. Consequently two propositions were made : restrict the upwind scheme to use the present stencil, use a lower TVD scheme of order 3.

By restricting the upwind scheme at the cell side that coincides at the block connectivity, we only use the available stencil (the stencil with three cells inside the block and two outside), For the east side of a cell to the east of the block its state ψ_e is directly equal to ψ_e^-, since the other reconstruction ψ_e^+ requires three cells to the east.

A lower order scheme 3rd TVD scheme was also added and compared with the method of accepting the available stencil. The state vector for the unavailable stencil is reconstructed through :

$$\begin{aligned} \psi_e^+ &= \psi_P &&+\theta/2\,(\psi_E &&- \psi_P) \\ \psi_{i+1/2}^+ &= \psi_i &&+\theta/2\,(\psi_{i+1} &&- \psi_i\ ) \end{aligned} \tag{9.50}$$

where θ is the flux limiter function. In NSMB many types of limiter are available, such as Van Leer limiter, minmod1, minmod2 and the Superbee limiter. Here we chose the van Leer symmetric limiter given by :

$$\theta_i = \frac{r_i + |r_i|}{1 + |r_i|} \tag{9.51}$$

where r represents the ratio between the successive gradients of the state ψ and is given by :

$$r_i = \frac{\psi_i - \psi_{i-1}}{\psi_{i+1} - \psi_i} \tag{9.52}$$

9.4.5 Ice Velocity Propagation Restriction

The ice velocity $\mathbf{v}_{ice}$ calculated at the IB points via Eq. 5.6 could generate the same problem as the collection efficiency discussed in subsection 9.2.6. Consequently, it is evaluated at a larger band. Since points further from the interface are irrelevant to ice accumulation at the wall, or at the IB points, the propagation equation is allowed to modify the velocity at all points except the IB points. In other words, the icing velocity at the IB points is kept constant while being propagated elsewhere. Different formulations were tested where only a percentage of the initial velocity is kept, to allow for smoother profiles. Written in symbolic form, at each propagation step :

$$\mathbf{w}|_{IB} = \alpha \mathbf{w}_0|_{IB} + (1-\alpha)\mathbf{w}*|_{IB} \tag{9.53}$$

where $\mathbf{w}_0|_{IB}$ is the initial value obtained from Eq. 5.6, $\mathbf{w}*|_{IB}$ the solution obtained after each propagation step and $0 < \alpha \leq 1$ is the factor used to smooth the solution and should be close to one.

9.5 Modules Interaction : Multi-Step Icing Algorithm

We present here the flow diagram of the different modules, summarizing the important aspects.

1. First the LS function ϕ is initialized manually by a list of points, or a mathematical function describing the solid, i.e., circle,square,sphere, NACA0012,...etc. The distance from each grid point to the solid is then generated by localizing the closest point.
2. ϕ and $\nabla\phi$ being generated we calculate the sign, Heaviside H and Dirac Delta δ functions.
3. Then, the IBM-LS information are generated, namely, the IB points indexes IBp_{ind}, for each IB point its image point location $\overrightarrow{X}_{im}$, the distance from the wall to the image point ϕ_{im}, and the indexes of the closest fluid point CP_{ind}
4. Following that, the air-flow is solved ; the turbulent URANS system is solved. The air-flow state vector and the turbulent viscosity are interpolated at the image points, then are used to impose the required profile at the IB points. The interior region where $\phi < 0$ is penalized.
5. The droplet solver follows, wherein the air velocity is the input, and the collection efficiency is the output. The droplet state vector is interpolated at the image points, then used to impose the required boundary condition at the wall, wet or dry. The droplet solver being fully implicit requires that interpolated state not be imposed directly, but that the coefficient matrices be decoupled. For a droplet size

distribution, the solver is run many times for each size, then the collection efficiency is averaged. The collection efficiency β is evaluated at a band of cells larger than a single cell, to avoid staircase formation, which results in discontinuities.

6. Currently, only rime ice is available. Thus, from the collection efficiency, the impinging liquid mass rate is assumed to turn all into ice. The ice accretion velocity is then evaluated at the same band where β was evaluated, and is fed to the following module.
7. The icing velocity is propagated in the normal direction. The ice velocity at the IB points is restricted/kept constant. The velocity is then utilized to advect the Level-Set function ϕ. ϕ is restricted to outward movements, no positive valued cell becomes negative. This is important, to avoid numerical problems met when the iced surface becomes very irregular after multiple ice shots, or when two fronts meet and coalesce. The reinitialization equation is solved to restore the signed distance function, at a predefined frequency, that is to be studied in the following chapter. The reinitialization equation is fixed by the local conservation additional term proposed by Sussman and Fatemi, called SF fix.
8. Finally, the Level-Set is reinitialized at the end of the advection step. At the end of the reinitialization, the new ϕ and $\nabla\phi$ are available.
9. For a following ice shot the steps 2 to 8 are repeated.

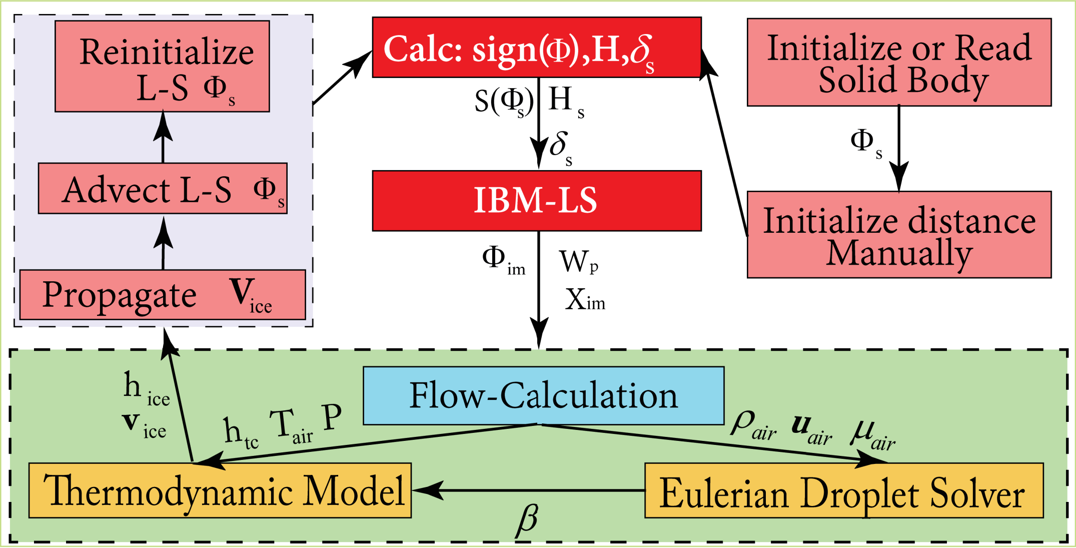


FIGURE 9.8 – Modules Interaction : multi-step icing algorithm in IBM-LS embedded grid disposition

Troisième partie

Results

Chapitre 10

LEVEL-SET EQUATIONS VALIDATIONS

For quantitative comparisons, we approximate the area or mass of the solid part by integrating the smoothed Heaviside function. This mass is given by :

$$M(t) = \frac{1}{\Omega} \int_0^{\Delta x} \int_0^{\Delta y} \int_0^{\Delta z} H\left(\phi\right) dxdydz \tag{10.1}$$

For simplicity the integral over the volume is considered equal to the average value in the cell center, which gives :

$$M(t) = \sum H\left(\phi\right) \tag{10.2}$$

The relative mass loss or change is then calculated via :

$$\epsilon(t) = 1 \, - \, \frac{M(t)}{M0} \tag{10.3}$$

To validate this choice we run a simple test case of a moving circle. The relative mass loss is then compared to the loss in area, where the area is estimated graphically. The resulting Level-Set zero contour is checked visually to estimate the change in the radius, four points on the peripheral are used and averaged to get an average radius r_{av}. The relative area change is then estimated via :

$$\epsilon_a(t) = 1 \, - \, \left(\frac{r_{av}}{r_0}\right)^2 \tag{10.4}$$

A circle of radius 0.4 length units in a two dimensional domain of 30 x 30, is subjected to a velocity field of $0.1, 0.1$. The grid is equally spaced in the initial region. The circle is advected 2 time units ensuring the circle staying in the region where the grid is equally spaced. Another test advects the circle 3 time units forcing the circle to move into regions where the grid is elongated, and non-equally spaced.

In Fig. 10.1 We see three snapshots of the moving circle. In black the initial circle centered at $(0, 0)$, in blue the circle after 1 time unit, and in magenta the circle after 2 time units that should be centered at $(0.2, 0.2)$. The small circles represent the zero contour whereas the big circles represent another contour at 0.4. In Fig. 10.2 we see the relative mass change during the advection which oscillates around zero and has a maximum change of $\epsilon_{max} = -1.29 \times 10^{-3}\%$. The final resulted relative mass change was much lower at $\epsilon_{end} = -8.5 \times 10^{-6}\%$. The area change at the end was checked through Eq. 10.4 and was found to be around $\epsilon_{a,end} = -1.9 \times 10^{-4}\%$. Qualitatively, the approach is plausible, and could be used to compare different discretization schemes.

However, if the circle is advected further until its zero interface falls in a region where the mesh is not equidistant anymore, the evaluation of the Heaviside function becomes sensitive. For a circle moving 3 time units, we see the resulting relative mass change in Fig. 10.3. The deviation of the mass is not related to a real deformation of the interface but due to way the mass is calculated. This could be avoided by investigating the area instead of the mass as evaluated in Eq. 10.1. By multiplying the said mass by the cell area, when the interface passes to larger cells the total area will not change. For the current scenarios studied in this chapter, we focus on discussing the effects of the discretization methods. We advect the interface in a region of equidistant cells and hence, the mass evaluation given by Eq. 10.1 is sufficient.

The oscillations of the relative mass loss are a result of the local positioning of the interface with respect to the cell it passes through. This is due to the primitive method of calculating the mass through the averaged Heaviside function Eq. 10.2.

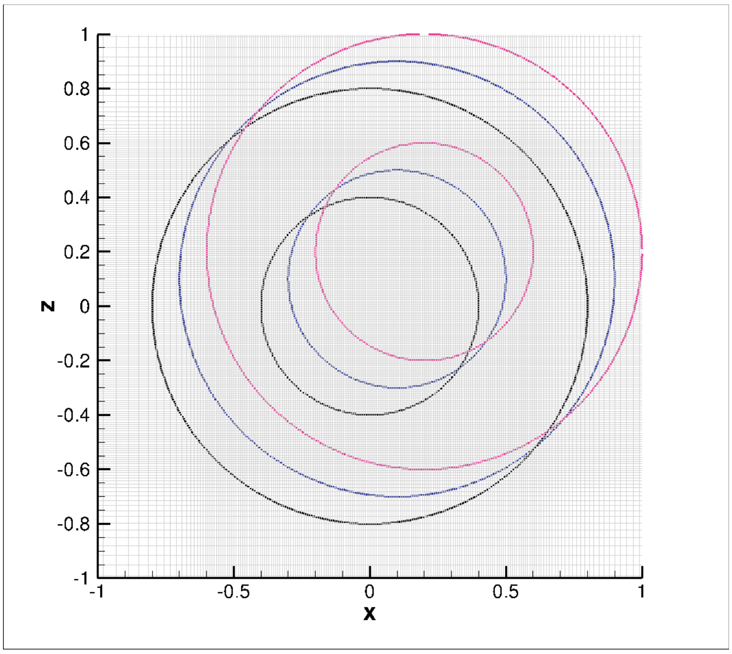


FIGURE 10.1 – Contours (0 and 0.4) of a moving circle, initially, midpath, and after 2 time units.

10.1 Level-Set Advection

In this section we present different results obtained for the LS advection Eq. 5.7. The discretization schemes discussed before are to be tested and compared in terms of relative

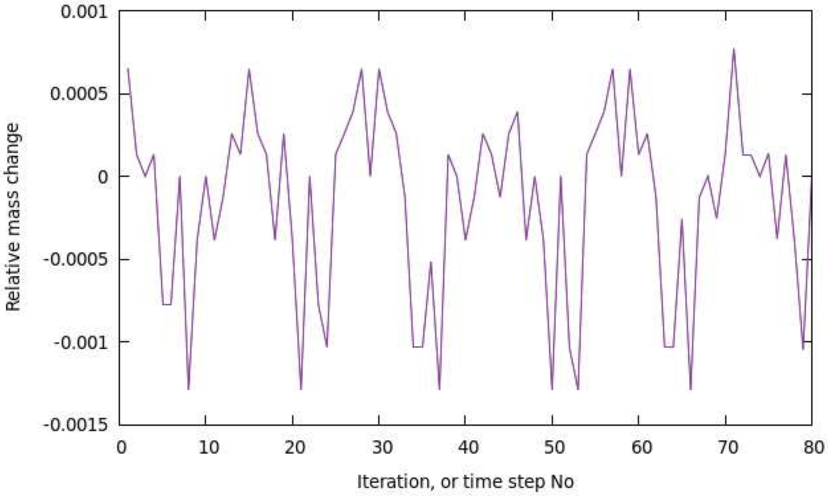


FIGURE 10.2 – Relative mass change for the moving circle.

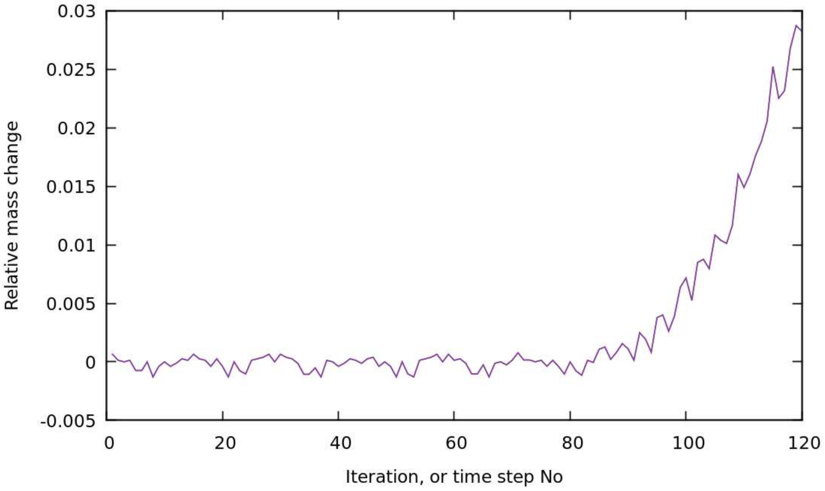


FIGURE 10.3 – Relative mass change for the moving circle, passing through nonequally distanced cells.

mass change. The first three cases in 2D seen in Fig. 10.4, are a circle of radius 0.4, a square of side 0.2, and a NACA 0012 of chord length 1. These three geometries are well initialized manually in the whole domain. The grid is of size 0.008×0.008, and contains 318×288 cells, noting that this grid is the same to be used later for icing simulation. The velocity vector used is $(0.1, 0.1)$, and the advection is done for 2 time units.

In tables 10.1,10.2 and 10.3 we summarize the results obtained for the three geometries : circle, square, and NACA 0012 respectively.

The implicit scheme always results in an increasing or decreasing mass change, or in other words an accumulative degradation of the interface. The Inflow-Implicit-Outflow-Explicit scheme results in underestimation of the advection, a body gets advected only a portion of the expected path. The explicit scheme of 5th order WENO5 is the most stable scheme. The time scheme coupled with the WENO5 scheme is also essential, where the

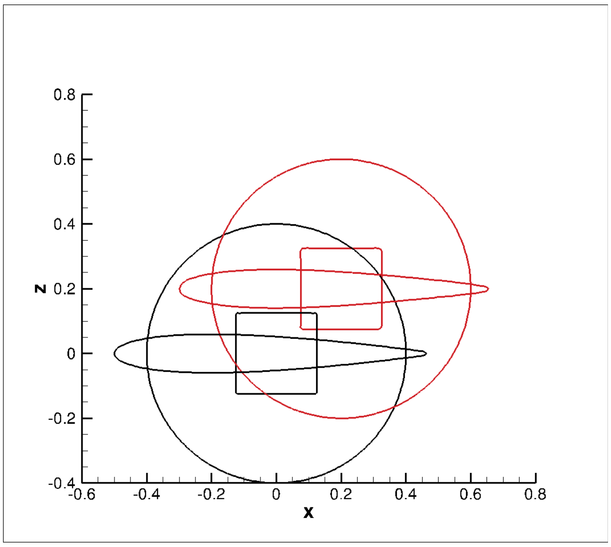


FIGURE 10.4 – Zero Contour of moving circle, square, and NACA0012, initially, and after 2 time units.

RK3 scheme is the only choice that shows consistent behavior. On the other hand, when coupled with the 1st order explicit scheme "Euler" accumulative degradation is resulted.

First, when advecting the circle we studied the maximum relative mass loss. The best behavior was obtained using the WENO5 RK3 schemes. A $CFL > 1$ for the explicit scheme results in high degradations as is to be expected. Using an implicit time scheme, accumulative degradation is apparent. Using a pure central scheme results in less degradation than what's obtained with upwind or blended schemes. The Inflow-Implicit-Outflow-Explicit scheme is restricted to $CFL \leq 1$ time steps. For a higher CFL the scheme becomes unstable, behaving like an explicit scheme, which defeats the purpose. Worse, even with a low CFL, the results weren't satisfactory, as stated in the previous paragraph.

It should be noted here why the error measured is in terms of the resulting maximum relative mass change "or loss". As stated before, the mass loss oscillates due to the local positioning of the interface with respect to the cell through which it pass. Is this aspect crucial when studying the mass of ice formation ? Should the different discretization methods or any numerical aspects be compared with this "$\epsilon(t)_{max}$" as the deciding factor ?

For the second case, the square of side 0.2, we study the average of absolute relative mass loss. The best behavior is again obtained using the WENO5 RK3 schemes. The reduction in the error when reducing the CFL is numerically superficial ; it is a result of the averaging. When investigating the final error, both results were similar. The implicit scheme always shows an accumulative degradation, indicating loss of consistency. Increasing the time step, increases the degradations, while the method stays unconditionally stable. And finally, the Inflow-Implicit-Outflow-Explicit scheme is not satisfactory.

For the last case, NACA0012 with chord length 1, we study the average of absolute

Scheme			Time (s)	% $\epsilon(t)_{max} \times 10^3$	Comments
Exp. Weno5	RK3	cfl=0.5	55	-1.29	Oscillates
		cfl=1	31	-1.29	Oscillates
		cfl=2	20	-522	Loss of form
	Euler	cfl=0.5	52	-4.44	Monotonic decrease
Imp. cfl=1	central	pure	32	6.2	Mono.
		blend 0.3	32	10.6	Mono.
		blend 0.6	32	13.5	Mono.
	upwind	pure	30	16.4	Mono.
InImp, OutExp	Central	cfl=1	15	-1.57	Underestimated advection
	Upwind	cfl=1	14	-6.4	Und. adv.

TABLE 10.1 – Resulting maximum relative mass loss when advecting the circle case.

Scheme			Time (s)	% $\epsilon(t)_{av,abs} \times 10^3$	Comments
Exp. Weno5	RK3	cfl=0.5	49	7	Oscil.
		cfl=1	28	12	Oscil.
	Euler	cfl=0.5	48	14	Mono.
Imp.	central	cfl=1	30	32	Mono.
		cfl=2	19	66	Mono.
		cfl=4	16	139	Mono.
	upwind	cfl=1	28	82	Mono.
InImp, OutExp	Central	cfl=1	29	8	Underestimated advection
	Upwind	cfl=1	32	28	Und. adv.

TABLE 10.2 – Resulting average of absolute, relative mass loss when advecting the square case.

relative mass loss. The best behavior is again obtained using the WENO5 RK3 schemes. Reducing the CFL does not affect the results. The implicit scheme shows the same inconsistency as for the other two cases.

An additional three-dimensional case is also investigated. A sphere of radius 10h (h being the grid spacing), is advected with a velocity of (0.1,0.1,0.1) from an initial position (0,0,0) for 5 time units. The mesh is divided in 16 blocks, and the case is run in parallel. The central part of the grid consists of equidistant cells (h=0.04), which then increase in size in all directions up to h=0.34. The sphere getting advected in this scenario remains in the zone with same sized cells. It should be recalled here that the WENO5 scheme falls to a TVD 3rd order scheme at the block connectivity.

For this case only the explicit WENO5 RK3 scheme succeeded in keeping the spherical form. Table 10.4, summarizes the results obtained using different CFL criteria. For a high CFL that lower than 1, detachments of the sphere appear when passing through block

Scheme			Time (s)	% $\epsilon(t)_{av,abs} \times 10^3$	Comments
Exp. Weno5	RK3	cfl=0.5	81	11	Oscil.
		cfl=1	61	11	Oscil.
	Euler	cfl=0.5	80	17	Mono.
Imp.	central	cfl=1	61	73	Mono.
		cfl=2	51	165	Mono.
		cfl=4	48	370	Mono.
	upwind	cfl=1	60	220	Mono.

Table 10.3 – Average of absolute, relative mass loss when advecting the NACA0012 case.

connectivities due to the TVD scheme. At lower CFL these detachments are absent and the sphere is advected adequately. Both the max and average error replicate the same findings : the lower the CFL the lesser the degradations. In Fig. 10.5 we visualize the sphere, to the left the sphere is severely detached when passing through blocks, at center the sphere is adequately shifted, and at its initial position at the right.

Scheme	CFL	$\%\epsilon(t)_{max} \times 10^3$	% $\epsilon(t)_{av,abs} \times 10^3$	Comments
Exp. Weno5 RK3	0.85	-70.1	38.9	Detachments
	0.4	-41.4	19.9	
	0.2	-13.1	5.3	
	0.04	13.3	5.1	

Table 10.4 – Relative mass loss when advecting a sphere of radius 10h (grid size h=0.04).

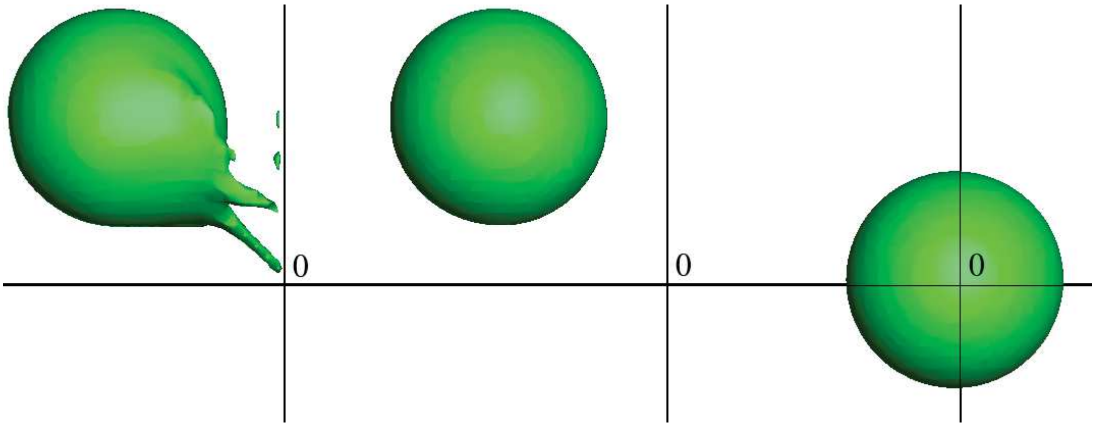


Figure 10.5 – Moving sphere advected with $CFL = 0.85$, $CFL = 0.2$, and its initial position.

To further investigate the necessity of the very low CFL we decided to add a case where the sphere returns to its initial position. The sphere gets advected in the same manner as before and then gets advected back. The results obtained are depicted in table 10.5. It should be stated that for the high $CFL = 0.85$ although the final relative error might seem low, the geometry is highly detached as seen in Fig. 10.6. From this figure, it is clear that the only deteriorations are on the part crossing the blocks (TVD 3rd).

We deduce from this same table 10.5, that the max, average or final error, all give the same indication. This indication could be right or could be wrong. Other tests should be applied when suspected, like a visual check of the profile.

Scheme	CFL	$\%\epsilon(t)_{max} \times 10^3$	$\% \ \epsilon(t)_{av,abs} \times 10^3$	$\% \ \epsilon(t)_{final} \times 10^3$
Exp. Weno5 RK3	0.85	-72.3	40.6	-16.8
	0.2	19.3	6	14.9
	0.04	19	5.9	11.2

Table 10.5 – Relative mass loss when advecting (forth and back) a sphere of radius 10h (grid size h=0.04).

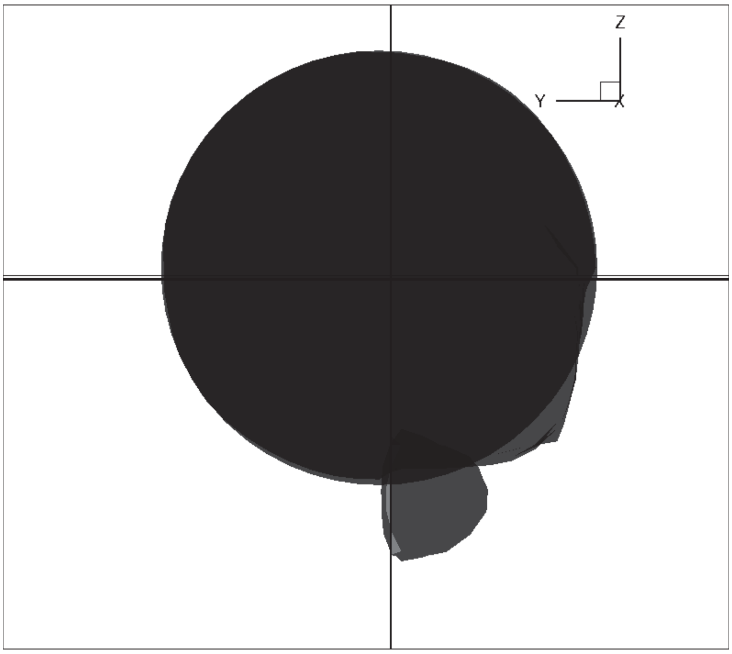


Figure 10.6 – Deterioration on the sphere advected forth and back with $CFL = 0.85$.

Another interesting point is the effect of passing through cells of different sizes. The returning sphere is advected 10 time steps, reaching a region in the mesh where cells get elongated ($h = 0.4 \rightarrow 0.6$), then returns back to its initial position. In table 10.6 we report the error obtained. The maximum and average errors are greater than those reported in table 10.5 for the same CFL, the same behavior as in Fig. 10.3. However, if we check the final error it is close to that reported in 10.5. Figure 10.7 shows the time evolution of the error, which is peaking at the region of elongated cells ; and which comes back to the same level when the sphere comes back to the region of equidistant cells.

Scheme	CFL	$\%\epsilon(t)_{max} \times 10^3$	$\% \ \epsilon(t)_{av,abs} \times 10^3$	$\% \ \epsilon(t)_{final} \times 10^3$
Exp. Weno5 RK3	0.04	137	20.3	8.4

Table 10.6 – Relative mass loss when the sphere passes cells of non-equal sizes.

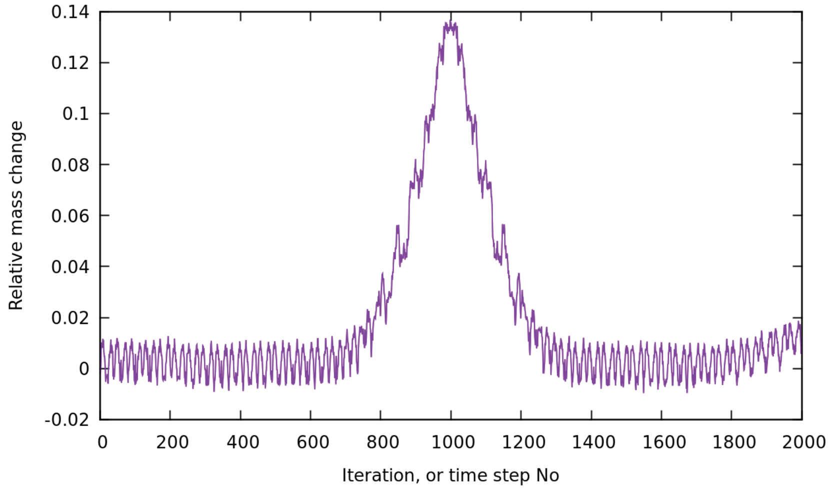


FIGURE 10.7 – Relative mass change for the returning sphere, passing through non-equally distanced cells.

10.2 Level-Set Reinitialization

10.2.1 Reinitialization Improvement on Level-Set Advection Conservation

The reinitialization is required when the Level-Set function is no more a signed distance, or when this feature is deteriorated in zones of interest. In ice simulation the zone of interest is the zone where the characteristic function χ_s, the Heaviside function $H(\phi)$, and the Dirac Delta function $\delta(\phi)$ are smoothed. It is also important to have a good signed distance along the zone where the interface passes. Actually the sign distance gets deteriorated when the interface is advected by an external velocity that is not uniform in the direction normal to the interface. In icing this is not the case. As mentioned in section 5.3 the velocity calculated at the interface is advected in the normal direction. Consequently, in icing, where the velocity in the direction normal to the interface is uniform, the reinitialization is only supplementary.

First, only the explicit scheme WENO5, RK3 results are discussed. This is because the implicit and explicit Euler scheme results were totally erroneous. The CFL is fixed at 0.9. We reinitialize a well initialized LS function, for the three 2-D cases discussed in the last section (circle, square, NACA0012), a total of 200 iterations. We are investigating the effect of the local conservation (interface preservation) Eq. 5.12 on the reinitialization procedure. We present in table 10.7 the results obtained. From a first look, the effect of the SF fix (Sussman and Fatemi 1999) is underestimated. The reason is when the results are visually checked the changes in topology with no local conservation are crucial. The NACA0012 case is reported in Fig. 10.8. In black, we see the initial profile that fits perfectly with the reinitialized profile using the SF fix. The red profile is the result of the reinitialization with no local conservation fix. The degradation is always within the cell containing the profile. The degradation is crucial since it introduces nonsmoothness, backsteps, and recirculation regions. Another interesting aspect is the oscillation of the error reported in table 10.7. In Fig. 10.9 we plot the error obtained while reinitializing the NACA0012 case. It is clear that with local conservation the method is stable and the numerical errors are negligible. On the other hand, without the local conservation, phases of topology changes occur. The final error is a very compromised factor to be used for decision making.

Case	Local Conservation	$\% \ \epsilon(t)_{200}$	Comments
Circle	NO	8.41×10^{-8}	Mono.
	SF	4.99×10^{-8}	Converging
Square	NO	2.12×10^{-3}	Oscil.
	SF	6.52×10^{-3}	Mono.
NACA0012	NO	1.36×10^{-3}	Oscil.
	SF	3.02×10^{-5}	Converging

TABLE 10.7 – Final relative mass loss $N_{it} = 200$, when reinitializing the well initialized LS function, WENO5 RK3, cfl=0.9. "SF" is the local fix given in Eq. 5.12

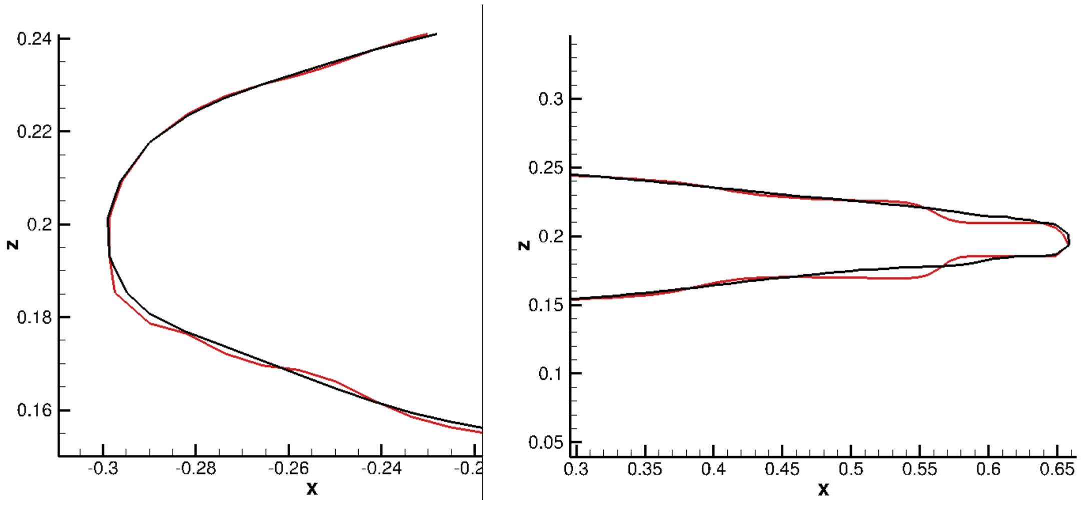


FIGURE 10.8 – Reinitialized profile, with local conservation, and without local conservation, on the leading and trailing edge of NACA0012. The initial profile coincides exactly with the profile obtained with local conservation.

Following that, we study the effect or the necessity of reinitialization during advection ; does reinitializing the LS while being advected reduce numerical errors ? We investigate the frequency of reinitialization and the number of reinitialization iterations. In tables 10.8,10.9 and 10.10 we compare results obtained using with and without the reinitialization, with and without the local conservation fix, and different frequencies and number of iterations.

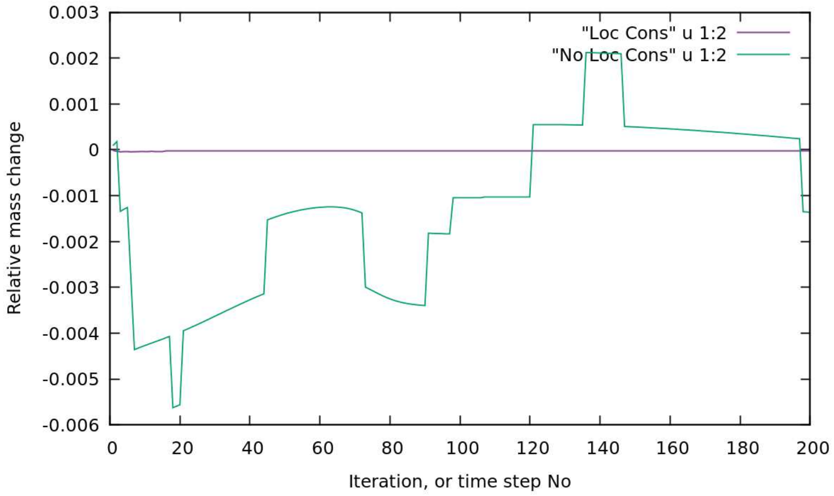


FIGURE 10.9 – Relative mass change, reinitializing the NACA0012 with and without the local conservation fix.

It could be concluded that reinitialization was only useful for the NACA0012 case. For the circle case, no improvements were obtained. For the square, having sharp edges, reinitialization while advecting is inadvisable. This conclusion can be generalized as : for geometries with very sharp edges reinitialization is inadvisable, for very smooth geometries reinitialization is irrelevant, and for other geometries reinitialization is recommended.

On the necessity of the local conservation, although all results show that the SF fix increases the numerical errors, visual checks prove otherwise. Reinitialization without local conservation considerations changes the topology. Contradictory to the numerical results, local conservation is imperative. This problem can be detected from the numerical values with a more thorough inspection.

Excessive reinitialization at each advection step produces a negative impact. This is seen in the case of running ten reinitialization steps each advection step. An important point here is that the topological degradation incurred due to reinitialization can be clearly detected with this negative scenario. For the three geometries, with excessive reinitialization, the degradation resulted with the SF fix is remarkably lower than without it. The best results were obtained for the case of ten reinit. steps every tenth advection step.

Lastly, we check the same effects on the three-dimensional geometry, a moving sphere. The frequency of reinitialization is set to (10/10) as was find optimal. The results obtained for the advecting sphere (forth and back) are presented in table 10.11. The CFL was restricted to avoid degradations resulted by block connectivity. Reinitialization with the local conservation SF fix gives the best results. Same conclusions as before.

Reinitialization	Loc. Cons.	Reinit. frequency	% $\epsilon(t)_{av,abs} \times 10^4$
No Reinit.	-	-	3.94
With Reinit.	NO	1 Re / 1 adv	3.94
		1 Re / 10 adv	3.94
		10 Re / 1 adv	4.12
		10 Re / 10 adv	3.99
	SF	1 Re / 1 adv	3.94
		1 Re / 10 adv	3.94
		10 Re / 1 adv	4.01
		10 Re / 10 adv	3.94

Table 10.8 – Effect of reinitialization frequency on the average mass loss, for the moving circle case.

Reinitialization	Loc. Cons.	Reinit. frequency	% $\epsilon(t)_{av,abs} \times 10^4$
No Reinit.	-	-	121
With Reinit.	NO	1 Re / 1 adv	192
		1 Re / 10 adv	139
		10 Re / 1 adv	744
		10 Re / 10 adv	129
	SF	1 Re / 1 adv	126
		1 Re / 10 adv	134
		10 Re / 1 adv	176
		10 Re / 10 adv	125

Table 10.9 – Effect of reinitialization frequency on the average mass loss, for the moving square case.

10.3 Ice Velocity Propagation

The only interesting part to be discussed here is the effect of the staircase formation, the effect of restriction, and the effect of passing through block connectivities, where the WENO5 falls to TVD3.

The effects of the staircase formation were discussed and treated in subsection 9.2.6. To summarize, the collection efficiency and the ice velocity are evaluated at a band of at least 2 cells normal to the interface, to avoid discontinuities in the resulting propagated ice velocity.

The effects of restriction can be investigated by looking at the resulting ice shape. We present in Fig. 10.10 and 10.11 the ice shape formed on a NACA0012 profile, using a single shot, and 2 ice shots respectively. The conditions of the flow are irrelevant for this investigation. In the first figure the effects are not important, and thus no conclusion can be made. However, for a higher number of ice shots, where the interface becomes irregular/non-smooth, these effects are more pronounced. By checking the form obtained in Fig. 10.11, it is clear that the restriction on the velocity is important, since without it, the resulted ice form is atypical.

Reinitialization	Loc. Cons.	Reinit. frequency	% $\epsilon(t)_{av,abs} \times 10^4$
No Reinit.	-	-	107
With Reinit.	NO	1 Re / 1 adv	88.3
		1 Re / 10 adv	105
		10 Re / 1 adv	324
		10 Re / 10 adv	95.2
	SF	1 Re / 1 adv	107
		1 Re / 10 adv	104
		10 Re / 1 adv	142
		10 Re / 10 adv	98.9

TABLE 10.10 – Effect of reinitialization frequency on the average mass loss, for the moving NACA0012 case.

Scheme	Reinit.	% $\epsilon(t)_{av,abs} \times 10^3$	% $\epsilon(t)_{final} \times 10^3$
Exp. Weno5 RK3, $CFL = 0.04$	No reinit.	5.9	11.2
	Reinit with SF	5.6	8.2
	Re without SF	7.1	19.3

TABLE 10.11 – Effect of reinitialization on the average mass loss, for the returning sphere.

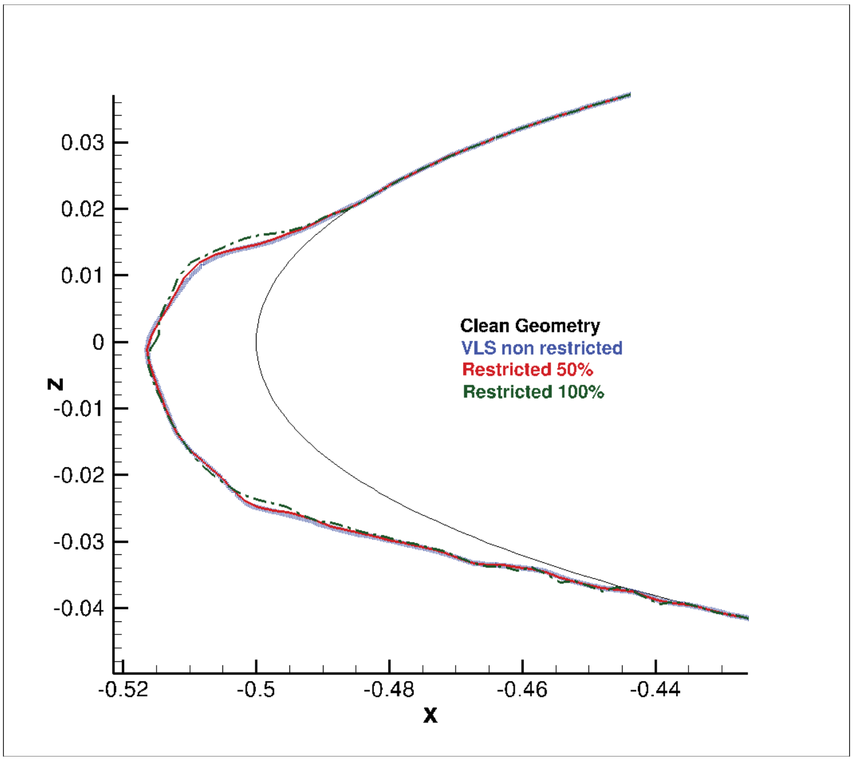


FIGURE 10.10 – Resulting Ice shape when the ice velocity is restricted or not. A single ice shot.

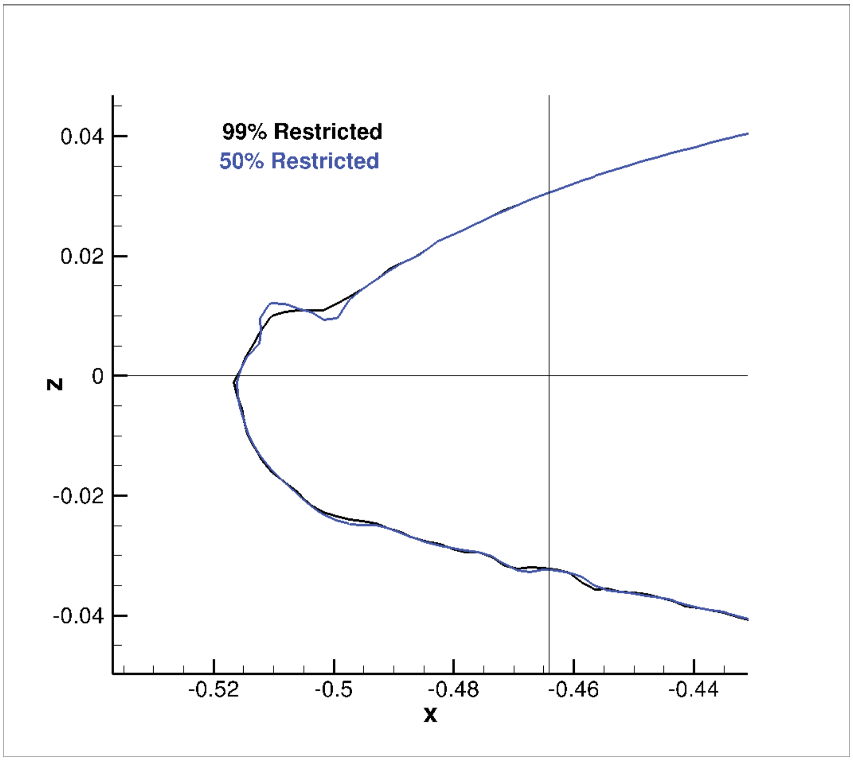


Figure 10.11 – Resulting Ice shape when the ice velocity is restricted or not. Two ice shots.

Finally, we present in Fig. 10.12 and 10.13 the resulting propagated icing velocity using a multi-block grid. The block connectivity does not produce any foreseeable problems. The only investigation for this part is by comparing these results with results obtained from body-fitted grids. These results are kept for later chapters, where icing is studied under defined conditions.

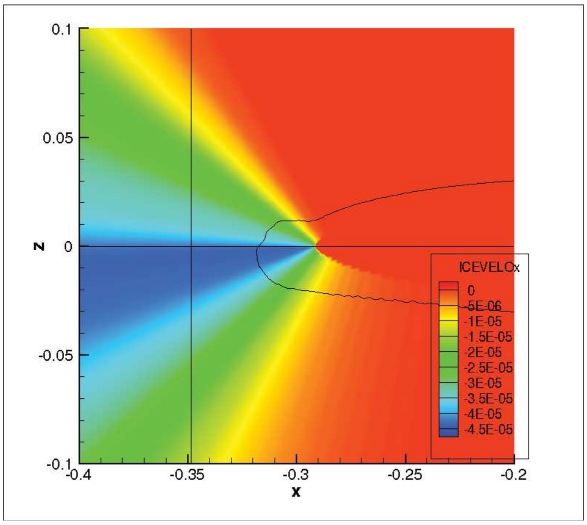


Figure 10.12 – Ice velocity : x component

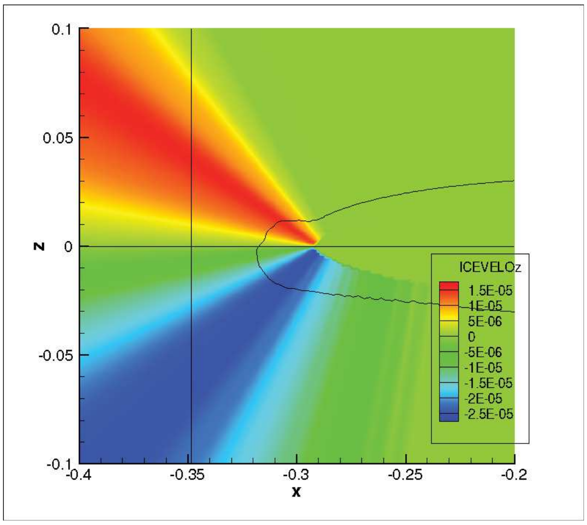


Figure 10.13 – Ice velocity : y component

10.4 Level-Set on Chimera Grids

In this section we discuss the applicability of the Level-Set equations with a chimera grid. Consider the superposed/chimera grid in Fig. 10.14. A circle of radius 0.3 centered at the origin $(0, 0)$, is advected with a velocity of $(0.1, 0.1)$, falling entirely on the higher fine grid level, two time units, passing to the lower grid level, and then is advected back to its original position. The result is shown in Fig. 10.15. The resulting area change is shown in Fig. 10.16. Since the interface passes through non-equisized cells, the mass change is corrected to account for the change of area ; equation 10.2 is replaced with equation 10.5.

$$A(t) = \sum_i \left[H\left(\phi\right)_i \times \Omega_i\right] \tag{10.5}$$

where Ω_i is the area or volume of the cell i.

The area change through the first part is negligible. Once the circle passes to the coarse level, the area change increases monotonically. This is mainly due to the the integration of H. Once the circle starts moving back the error decreases. When the circle is fully back in the fine level, the change in the error becomes negligible again. The final error however, is slightly higher than the error at the initial phase. This could be incurred by the errors resulted from the interpolation used passing the chimera levels. We conclude that the advection of the LS function is safe, considering the abrupt change in mesh size.

It is also of great interest to test the reinitialization over a chimera grid. The reinitialization, being highly sensitive to the order of the discretization scheme, is tested on the same chimera grid used in the last example. A circle of radius 0.4, whose zero level passes from one chimera level to another, is first initialized on a band of distance 0.2. Outside the initialized band, ϕ is set equal to 0.2 outside and 0 inside. After reinitialization the distance is propagated outside and inside as depicted in Fig. 10.17. The figure shows the distance being generated along a vertical line passing through the center of the circle. The method seems to work rather accurately.

A deeper look however, shows more interesting complications. By zooming on the zero contour of ϕ at the passage between the chimera levels, we observe topological degradations, see Fig. 10.18. These degradations are not the result of chimera interpolations, rather by the choice of the value of η used to smooth δ, and Δx used to smoothly calculate the *sign* function. If these are set to be equal to the smaller grid size, neither the reinitialization is done accurately, nor is the local conservation applied correctly. This problem is easily solved by allowing each block, or even each cell to impose its own size when smoothing δ and the *sign* functions, see Fig. 10.19

Again, if the other contour lines are visualized, we notice other complications. Looking at Fig. 10.20 the contour lines being advected/reinitialized from the lower coarse level to the higher fine level are slightly deformed, mainly due to the chimera integration. The contour lines advected from the fine level to the coarse level are accurate on the other hand.

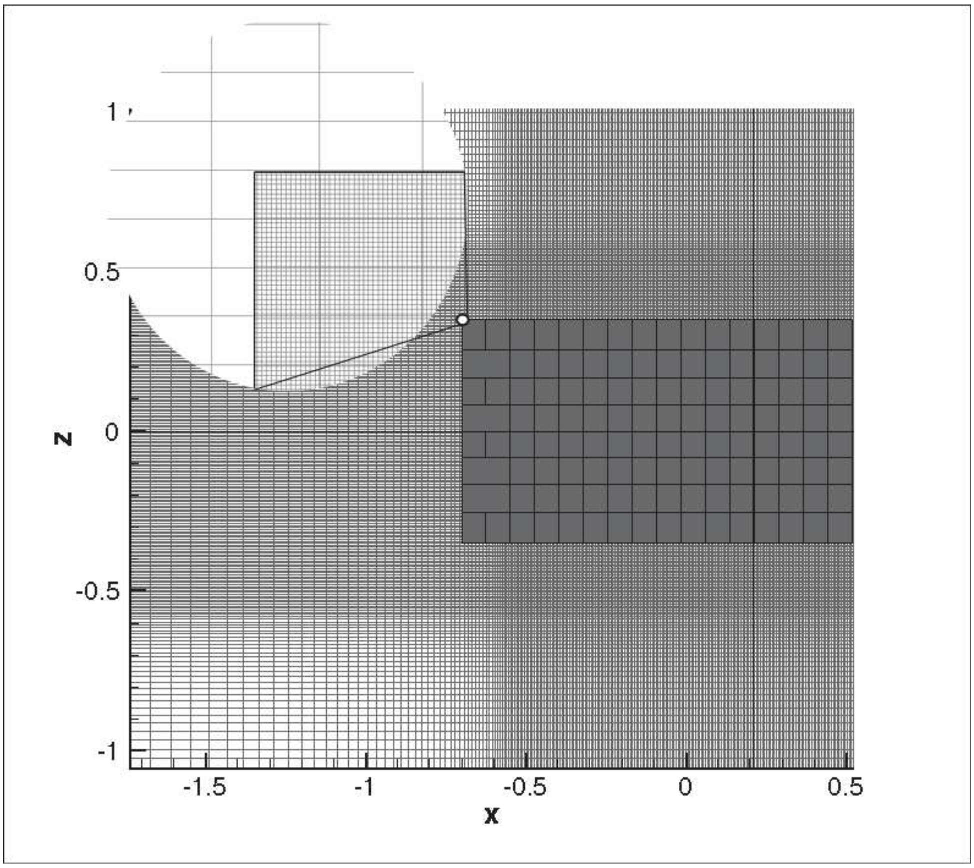


FIGURE 10.14 – Chimera grid : 128 blocks, 2 levels

Consequently, a smooth transition of grid sizing between chimera levels could alleviate this problem.

10.5 Conclusion

In this section we summarize the results obtained throughout this chapter.

The WENO5 spatial scheme coupled with RK3 temporal scheme was the only consistent scheme, where no monotonic increase in errors is present. We concluded that the mass or area change as a measure of error is not dependable, whether it be the maximum, the average or the final error. Sometimes a visual check is required. This could be alleviated if the curvature is checked as well at each cell through which the zero level passes, to evaluate topological changes. We also found that excessive reinitialization damages the LS function, and the best results were obtained when the reinitialization is performed 1/1 or 10/10, N_{reinit}/N_{adv}. The local conservation proved essential in removing any topological degradations. The ice advection or propagation in the normal direction was also investigated. Good results were obtained when the velocity at the IB points is highly restricted/kept constant. The propagation through block connectivities, where the WENO5 scheme falls to a TVD3 scheme produced acceptable if not perfect results. Advecting the

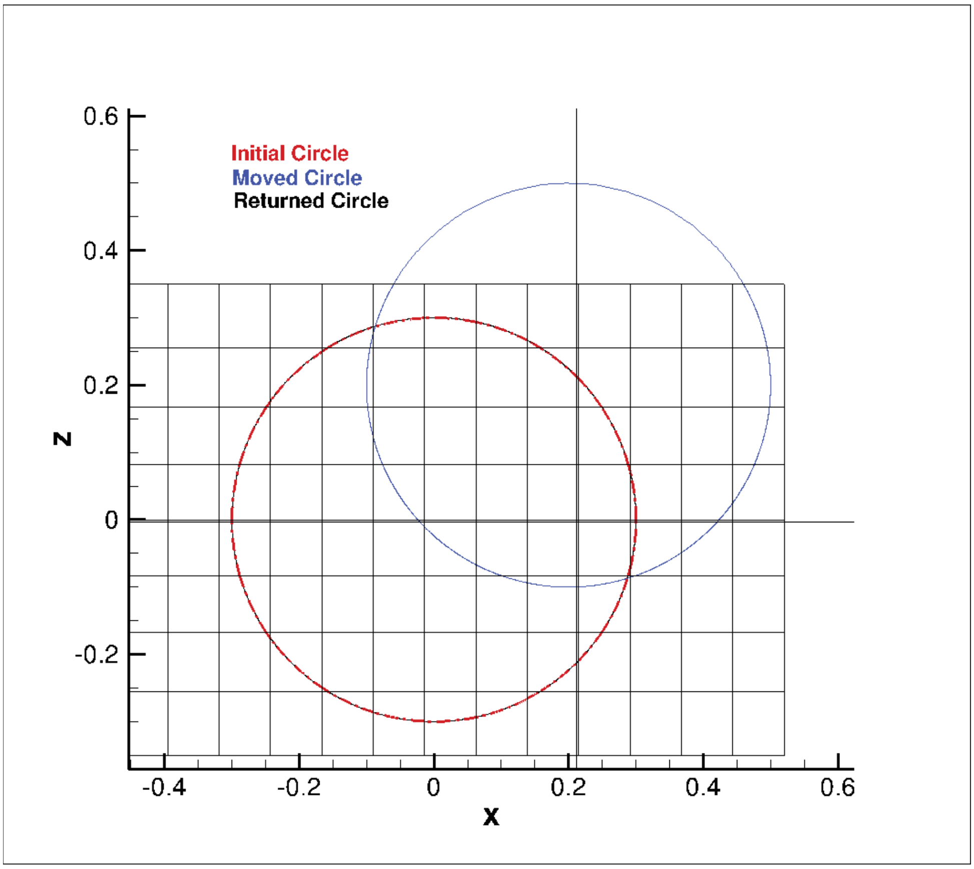


FIGURE 10.15 – Circle centered at $(0, 0)$ advected to point $(0.2, 0.2)$ then advected back.

LS function through chimera grids was also checked and was found acceptable. Reinitialization through chimera grids was also found acceptable with certain considerations. Smooth transition of cell sizes between different chimera grid levels is advised. The smoothed LS functions, δ, H, and $sign$, should depend on the local grid size.

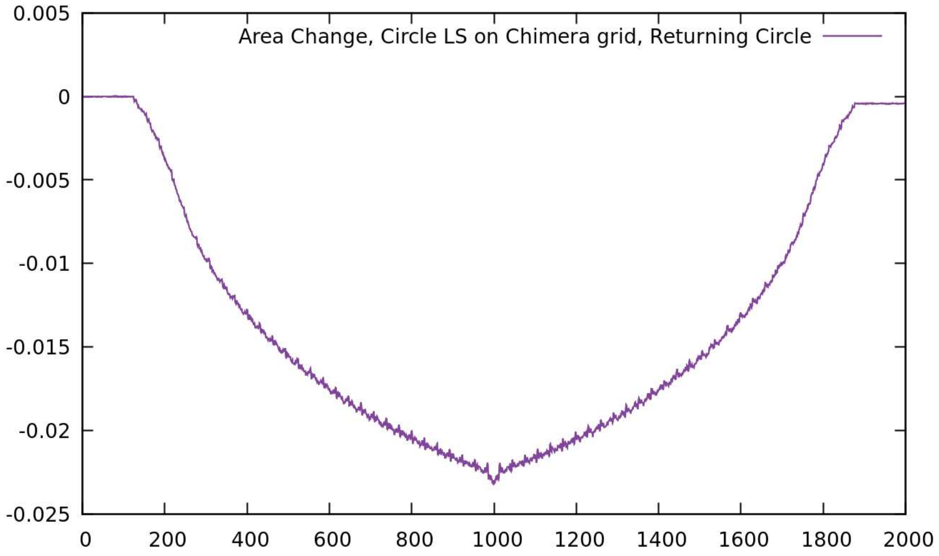


Figure 10.16 – Area change for the moving circle from fine grid level to coarse grid level, then back.

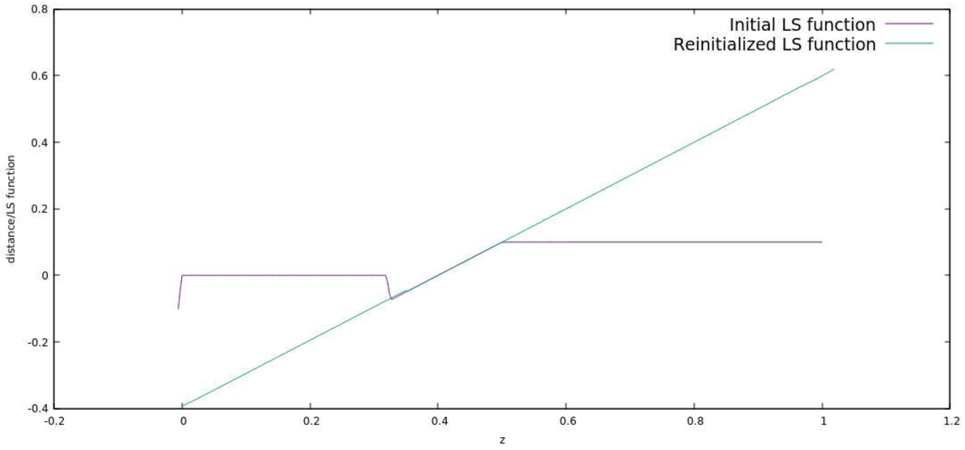


Figure 10.17 – Reinitialization over a chimera grid.

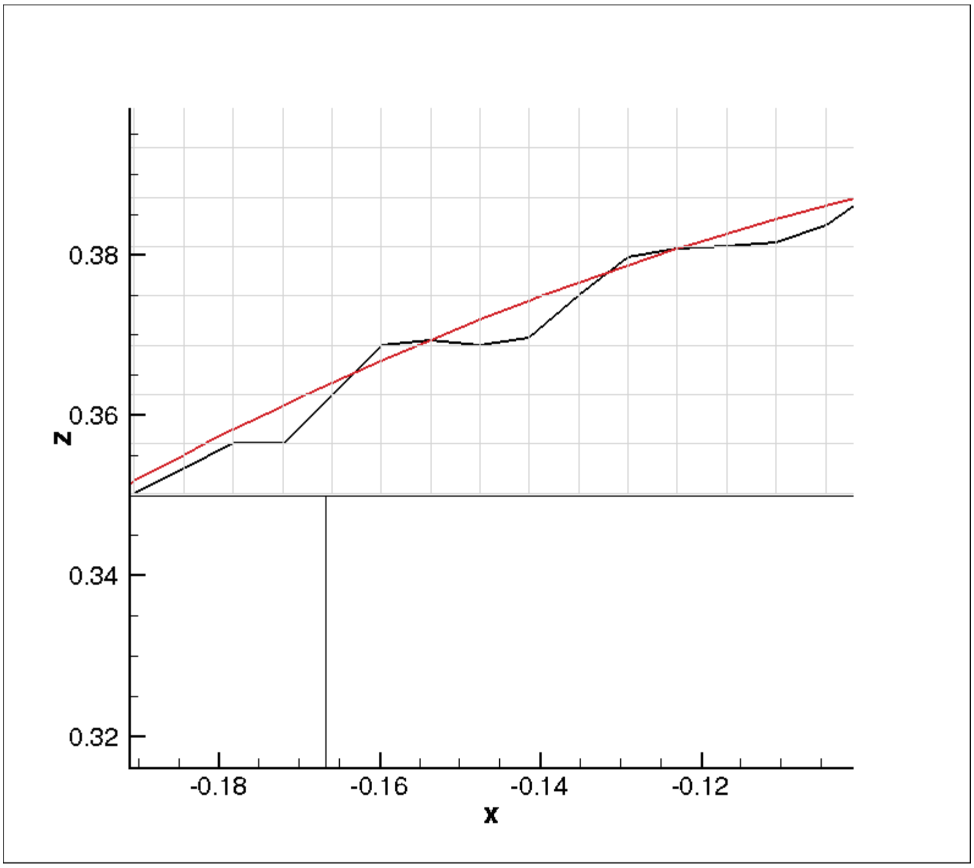


FIGURE 10.18 – Topological degradations on the level-set due to ill-chosen η and Δx.

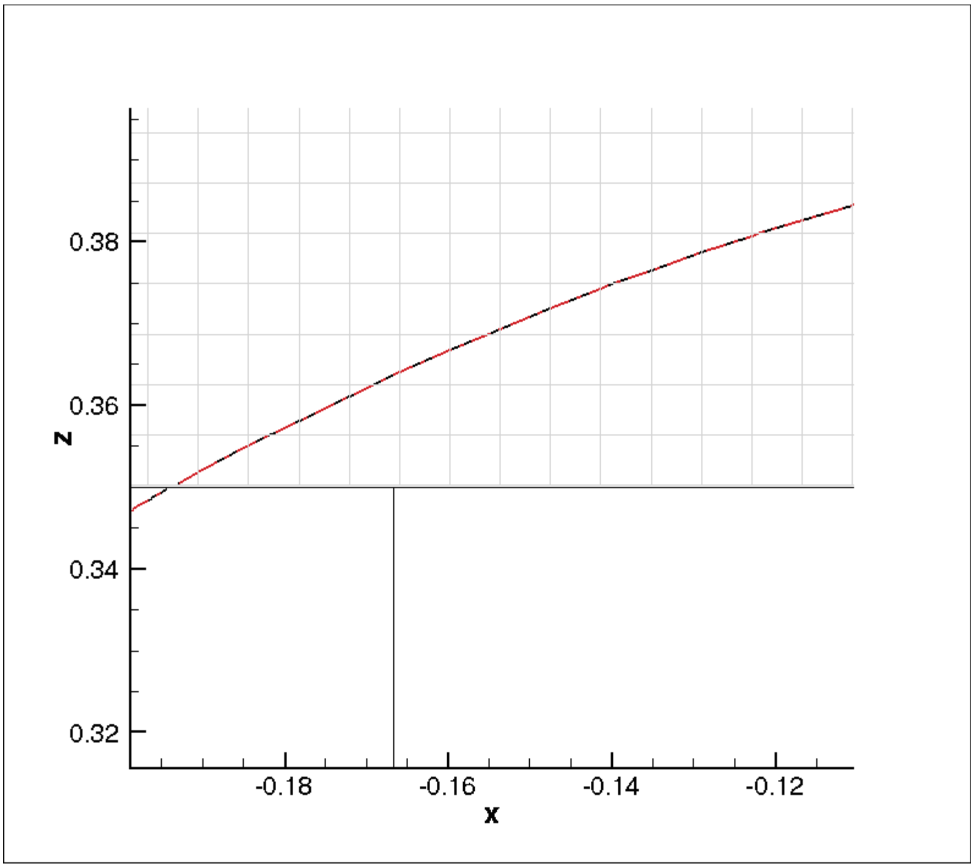


Figure 10.19 – Topological degradations reduced by allowing dynamic η and Δx.

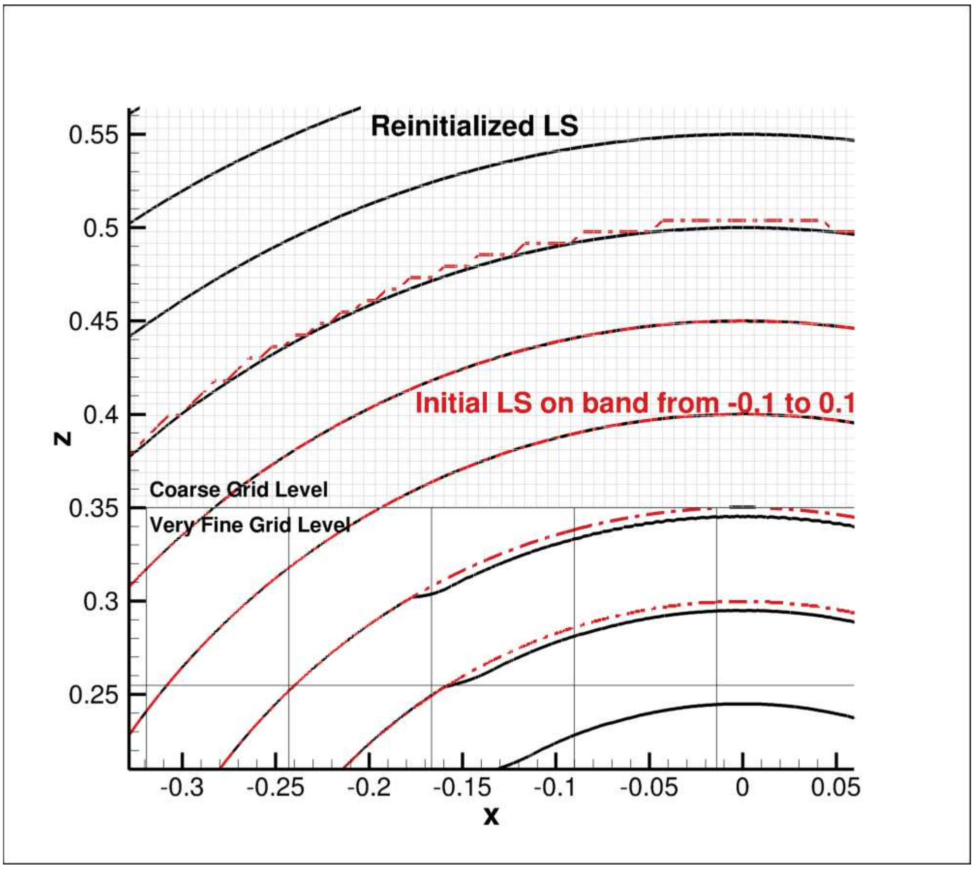


Figure 10.20 – Chimera interpolation effects on the redistancing.

Chapitre 11

IBM-LS Framework : Air-Flow + Droplet Capture

In section 9.2.4 we developed a coupled IBM-LS method to evaluate the droplet flow field. The use of an explicit representation of the interface was needed because of the adaptive boundary condition. In this chapter we propose a simple test case - a Laplacian - to validate the method first. Following that, we estimate the degradations resulting from such approximation. Once this is all done, we move to cases of droplet impingement.

11.1 Laplacian Test Case

In this section we propose a test case to validate the IBM-LS boundary condition discrete forcing technique. We solve a PDE whose analytic solution is known - A Laplacian equation. The PDE is given in Eq. 11.1, whose analytic solution is clearly $U = x^2 + y^2$. The domain is a 2d square of size 1. We use the same method used for the droplet field explained in section 9.2.4. We utilize finite volume method, central scheme for the second order derivative, and matrix inversion method to solve the resulting linear system. The coefficient matrices are given in Eq. 11.2. We use a Cartesian grid with equisized cells. Thus the FVM discretization resembles finite difference method. Since we use matrix inversion method, the boundary conditions are applied by changing the coefficient matrices. The required field value is imposed at the right hand side and corresponding coefficient is set to zero in the matrix.

$$\begin{aligned} \Delta U &= 4 \quad in \quad \Omega \qquad and \qquad U = x^2 + y^2 \quad in \quad \partial\Omega \\ U_i &= 0 \quad in \quad \Omega \setminus \partial\Omega \end{aligned} \tag{11.1}$$

$$A_P = 4, \qquad A_{nb} = -1 \qquad and \qquad RHS = -4\Delta x^2 \tag{11.2}$$

where U_i is the initial condition/guess. We calculate the L2 and H1 norms to estimate the numerical errors, as given in Eq. 11.3 and 11.4. We use these norms as indicators of the discretization order.

$$\|\epsilon\|_{L_2} = \sqrt{\sum_{i,j} \left(U(i,j) - U_{an}(i,j)\right)^2} \tag{11.3}$$

$$\|\epsilon\|_{H_1} = \sqrt{\sum_{i,j} \left(U(i,j) - U_{an}(i,j)\right)^2 + \|\nabla U(i,j) - \nabla U_{an}(i,j)\|^2} \tag{11.4}$$

We use three grids whose spacing is (h=0.0156, h=0.0078, h=0.0039). The resulting norms are reported in table 11.1. The numerical values shown follow an unexpected behavior, where the error is higher for a finer mesh. However, these errors are infinitesimal, and result from machine precision and from the sudden stop of the iterations before convergence ; we run a total of 25000 iterations in each case. Actually, the results of the coarse grid where already converged, while the two finer grids hadn't had converged yet when stopped after 25000 iterations, and the convergence was very slow.

h	$\|\epsilon\|_{L_2} \times 10^{14}$	$\|\epsilon\|_{H_1} \times 10^{14}$
0.015625	1.58	46.45
0.007815	3.36	197.55
0.003905	13.23	817.77

Table 11.1 – $\|\epsilon\|_{L_2}$ and $\|\epsilon\|_{H_1}$ norms resulted from solving the Laplacian equation.

11.1.1 Laplacian Coupled with a Level-Set

To test the IBM-LS disposition when applying the boundary condition, we add a level-set inside the domain. A circle of radius 0.2 is added to the center of the domain. The cells inside the circle are decoupled from the Laplacian solver. The first outward grid layer is used to enforce the boundary condition, in the same fashion as applied for the droplets in section 9.2.4. The only type of BC applied here is a Dirichlet BC. Although U is known at this layer of IB points, we try to imitate the method used for the droplets. The cells are mirrored against the LS wall, closest points are used to reconstruct the field at the mirrored points, and then by knowing the required value expected at the interface we interpolate between the mirror point and the wall to estimate the field at the IB point. Table 11.2 summarizes the errors obtained. The L2 norm is evaluated at all the cells outside the LS ; the H1 norm is evaluated in the same manner except for the IB points where the derivative is absent. The errors are higher by many orders of magnitude, -14 without the IBM-LS, and -3 with the IBM-LS. The log-scale L2 norm has a slope of nearly one indicating that the method is of order one. This is due to the limitation imposed by the interpolation used for the BC forcing. The Laplacian is discretized using a 2nd order scheme, the interpolation from the fluid to the image point is of 2nd order accuracy, but the imposition of the BC at the IB point uses a linear interpolation from the image point and the wall. This first order interpolation is the deciding factor on the order of the whole method. However, the most important conclusion here is that refining the mesh reduces the total error.

h	$\|\epsilon\|_{L_2} \times 10^3$	$\|\epsilon\|_{H_1} \times 10^3$
0.015625	9.09	73.65
0.007815	6.13	313.94
0.003905	2.54	83.4

Table 11.2 – $\|\epsilon\|_{L_2}$ and $\|\epsilon\|_{H_1}$ norms resulted from solving the Laplacian equation, where a part of the domain is blanked by a Level-Set circle. From the log graph we obtain a slope of 0.92.

11.1.2 Laplacian Coupled with a Moving Level-Set

An interesting investigation can be done here. If the Level-Set was being advected by an external velocity field, what behavior would the IBM-LS boundary forcing have on the Laplacian case ? What is the importance of local conservation, SF fix, discussed in previous chapters ? Consequently, the circle is advected from an initial position around (0.25,0.25) to a final position at (0.75,0.75). U is initialized to zero after each advection step, and is iterated until convergence, after which, ϕ is advected a single step. The evolution of the L2 and H1 norms is of great interest, since it indicates the effect of the moving ϕ on a physical problem that worked fine when ϕ was stationary. It can also show the necessity and/or impact of reinitialization, and the ensuing topological changes. The resulting error evolution is depicted in Fig. 11.1. Only with the locally conserved reinitialization do the errors not accumulate.

The contours of U at the final step are also shown in Fig. 11.2. The known solution of U predicts the isocontours to be perfect circles around the origin, $U_{an} = x^2 + y^2$. Only the results obtained with locally conserved reinitialization (in black) satisfies the predicted results.

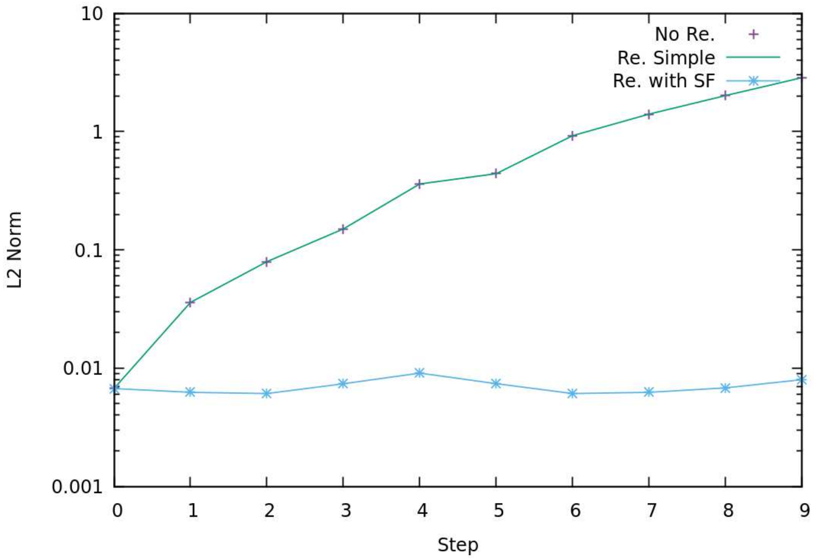


Figure 11.1 – Evolution of the L2 norm for the Laplacian case with the moving IBM-LS surface.

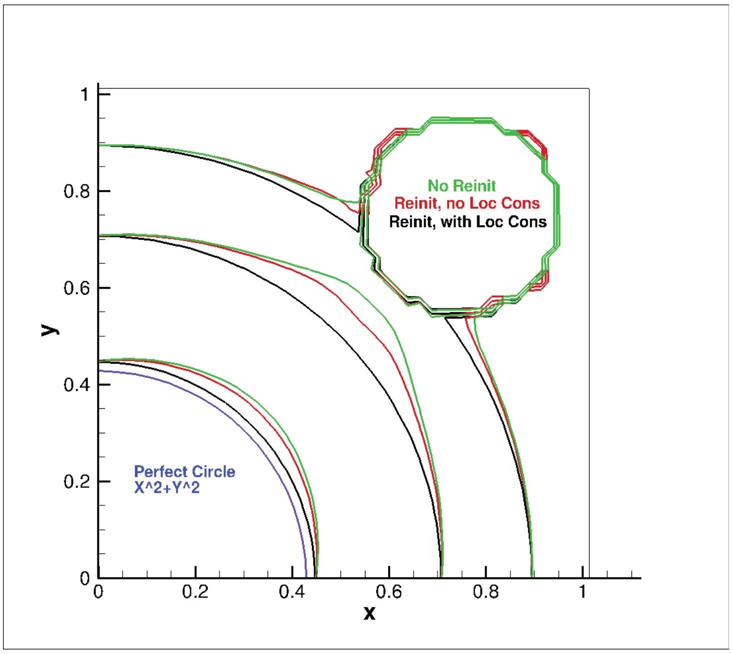


FIGURE 11.2 – Isocontours of U at the final step, with and without reinitialization, and local conservation.

11.2 Air Flow : Penalization vs. Velocity Profile

In this section we investigate the defect resulted from the penalization method on the velocity close to the wall. The penalization method is first validated for a laminar case with low Re. A square of side 1, $(50h)$, is meshed, once using multi-block body-fitted mesh, and another where the empty square is replaced with an embedded grid, basically the same grid. A flow of $Re = 30$, $\alpha = 0°$ is simulated. The x-component velocity profile along a horizontal line is depicted in Fig. 11.3. The result is very satisfactory, and the velocity profile is rather acceptable given the grid is not very fine.

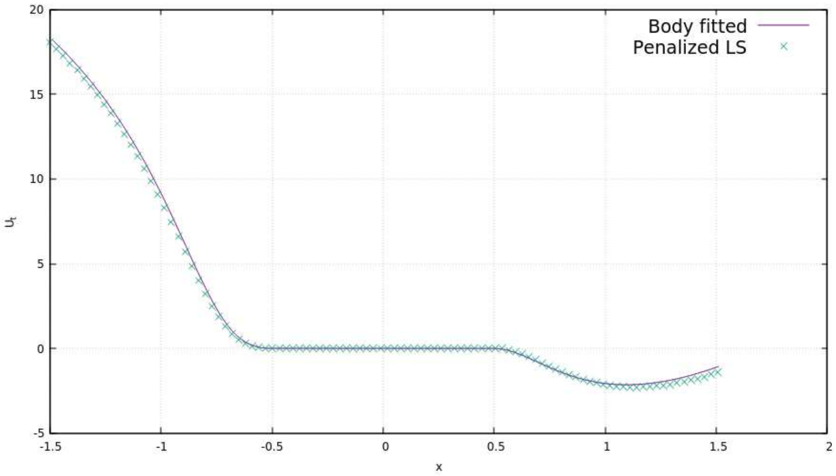


FIGURE 11.3 – u_x along a horizontal line, passing through center of square, $Re = 30$.

For icing situations the flow is mainly turbulent, thus we restart the case with a high Reynolds number. The velocity profile is shown in Fig. 11.4. The velocity profile looks rather good, however, the first cell at the wall has a velocity of zero for the penalized case. It should be stated here that for both grids no turbulent wall laws are available in NSMB. However, the fact still is that penalization highly underestimates the velocity close to the wall. If the penalization is replaced with the seventh power-law velocity profile, the results are shown in Fig. 11.5. The velocity at the first cell is acceptable.

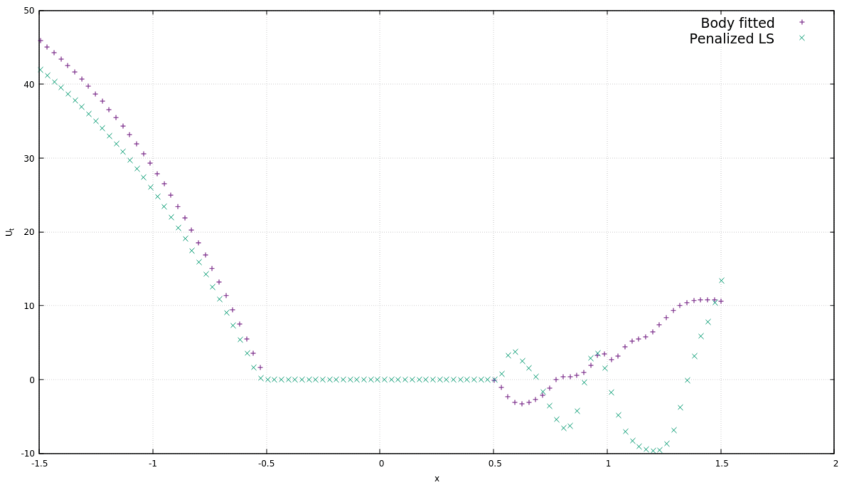


FIGURE 11.4 – u_x along a horizontal line, passing through center of square $Re = 5 \times 10^6$.

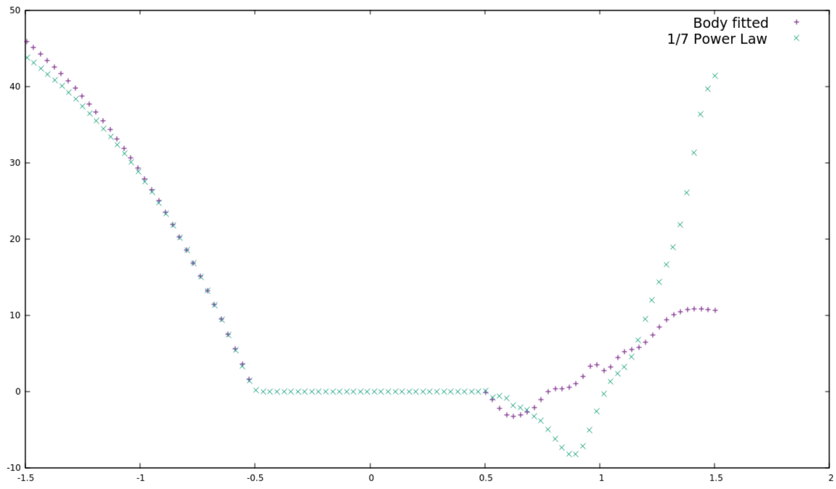


FIGURE 11.5 – u_x along a horizontal line, passing through center of square $Re = 5 \times 10^6$.

Another interesting test case is a NACA0012 of chord $c = 0.537$, for which the body-fitted grid solution is available. The embedded grid mesh is Cartesian of size 0.002. On the other hand, the body-fitted mesh is a turbulent grid, orthogonal to the wall. The computational domain extends to $50 \times c$. The flow is simulated at an angle of attack α of $4°$, a Mach number of $M = 0.3168$, a Reynolds number of $Re = 9.02 \times 10^6$, a static temperature of $T = 250.37$. The Navier-Stokes solver is coupled with the Spalart-Allmaras solver, with its ONERA roughness extension for the body-fitted grid. The Spalart-Allmaras was used for the embedded grid as explained in previous chapters. The resulting velocity profile at the leading edge and on top of the wing, are depicted in Fig. 11.6 and 11.7.

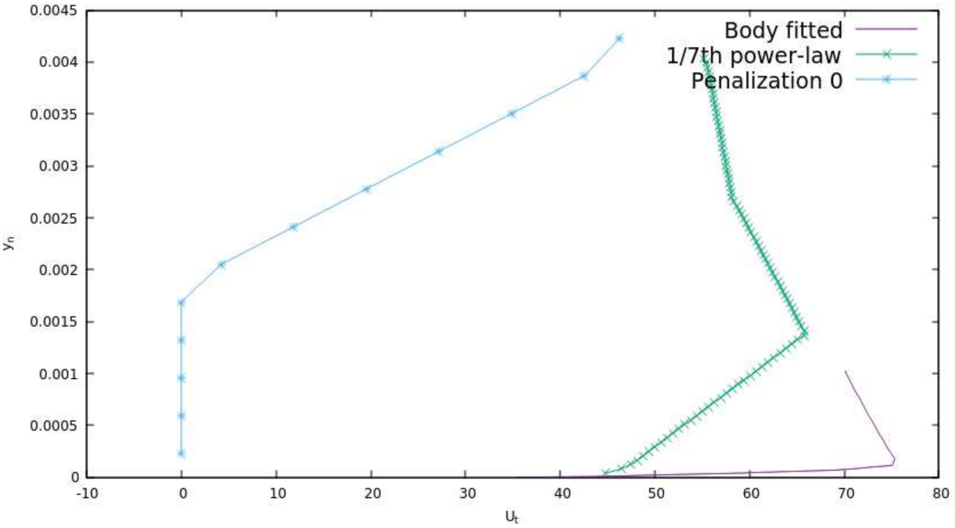


Figure 11.6 – Tangential velocity profile perpendicular to the wall at the tip of NACA0012.

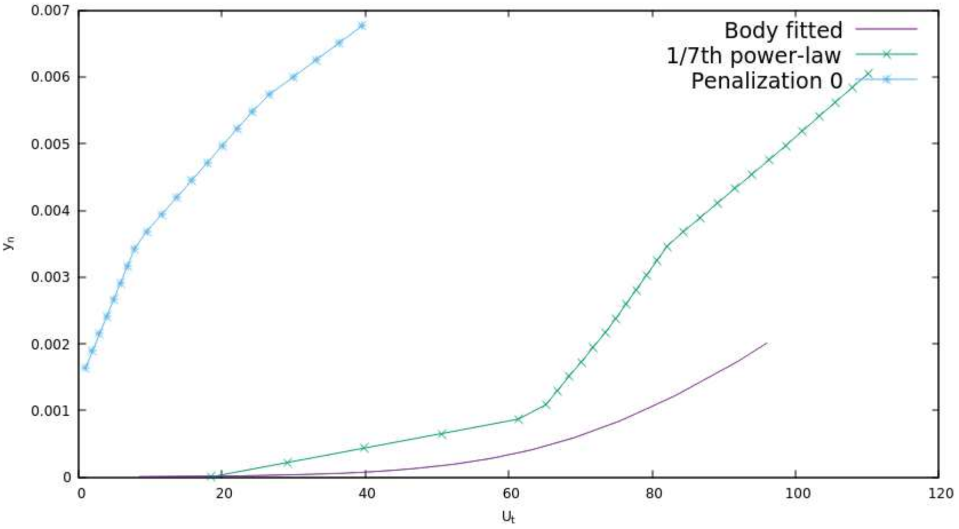


Figure 11.7 – Tangential velocity profile perpendicular to the wall at the top of NACA0012.

Although the velocity profile resulted from the seventh power law is not purely accurate, it far exceeds the ability of the zeroth penalization. The penalization method is discarded. The boundary layer is overestimated by the seventh power lay. This is indeed expected, since the gradient never reaches zero, which should be the case at the end of the log layer, or the turbulent boundary layer. It should be noted here that these results are obtained with a grid size of 0.002, the points in both figures are interpolated by the visualization software. This is why we see linear segments, each line is a single cell. Further refinement of the grid consumes huge resources since we use pure structured grids. Better results are expected with finer grids close to the wall. The applicability of chimera grids, to avoid massive cell numbers, is invaluable.

11.3 Air Flow : Turbulence

As stated before, the Spalart-Allmaras model is utilized in the IBM-LS configuration. The velocity close to the wall being imposed via the seventh power-law with no regard to which part of the boundary layer the cell falls under, is a risky choice. The easiest nonetheless. Giving no regards to turbulence wall-law, and directly interpolating linearly the value of turbulent viscosity should impose high limitations on the flow. This is the case, as can be seen in Fig. 11.8. It is clear from this graph that the eddy viscosity, or the turbulent viscosity is underestimated in the IBM-LS embedded grid case. The turbulent viscosity develops near the trailing edge and is late to diffuse the vortices. The Spalart-Allmaras model alleviates the problem of the vortex shedding that would be present if no Reynolds averaging is applied to the laminar NS equations. However, because of the poor refinement of mesh close to the wall, because of the power-law velocity profile and because of the linear interpolation of the turbulent viscosity, downstream vortices are not eliminated, Fig. 11.9. Again, utilization of chimera grids to refine close to the wall could provide massive improvements.

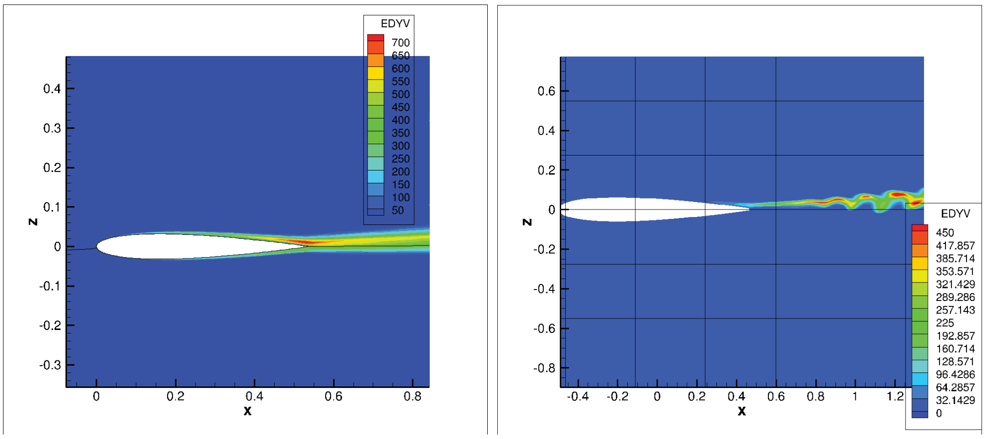


FIGURE 11.8 – $\tilde{\nu}$ Body-fitted grid vs. embedded grid

11.4 Droplet Capture

In this section we investigate the droplet capture through the boundary condition enforcement through the IBM-LS framework. The same case of NACA0012 in the previous section is tested under icing. The droplet field is simulated with a mono-dispersed droplet

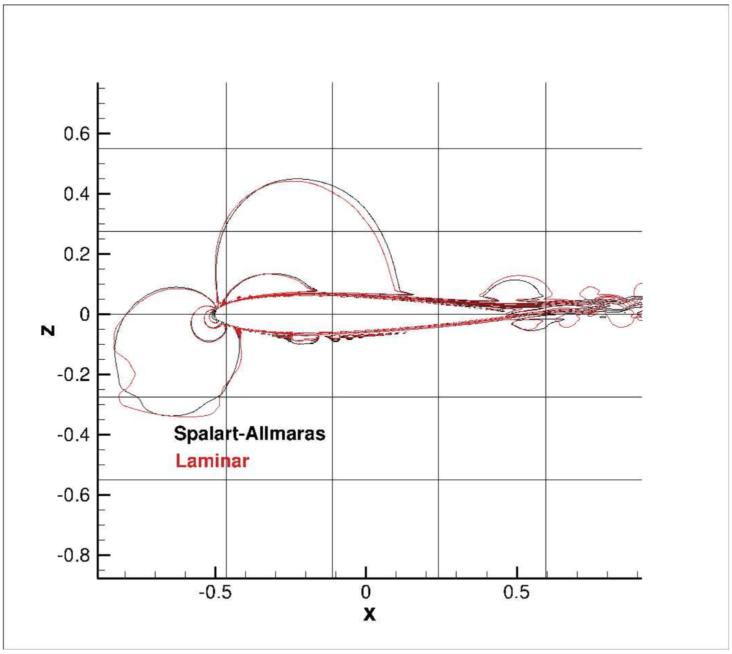


FIGURE 11.9 – Velocity contours, comparison between laminar ($Re > 10^6$) and RANS solution.

diameter of size $d = 20\mu m$ and a liquid water content of $LWC = 0.55 g/m^3$. The total exposure time is $t = 420s$, irrelevant at this point. The resulting droplet field with the IBM-LS embedded grid is compared with the one resulted from the turbulent body-fitted grid in Fig. 11.10, 11.11, and 11.12. The result shows good agreement qualitatively, with slight deviations, however. The resulting collection efficiency β is depicted in Fig. 11.13. Again, the results are qualitatively acceptable. The maximum value is overestimated, and the wet zone is slightly shifted to the top. The shifted wet zone is partially due to the velocity profile, and slightly due to miscalculation of the arc length. For body-fitted grids, the arc length is easily calculated. Contrarily for embedded grids, the arc length is evaluated by summing the length of linear segments from one IB point to the next. In other words, in body-fitted grids orthogonal to the wall, the segment tangent to the wall is a good estimation of the arc length, whereas with embedded grids the linear segments form a staircase. This can be alleviated, if needed, by more complex interpolation of the wall points using the LS information.

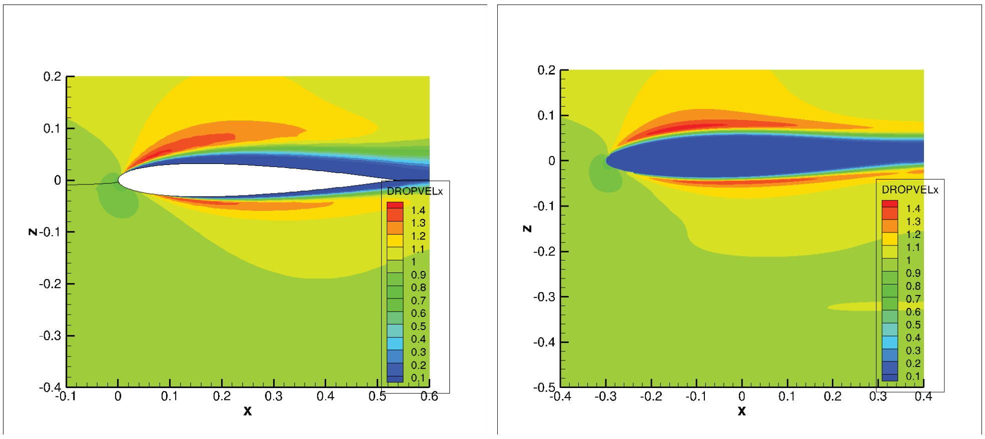


FIGURE 11.10 – Droplet u_x, body-fitted grid vs. embedded grid

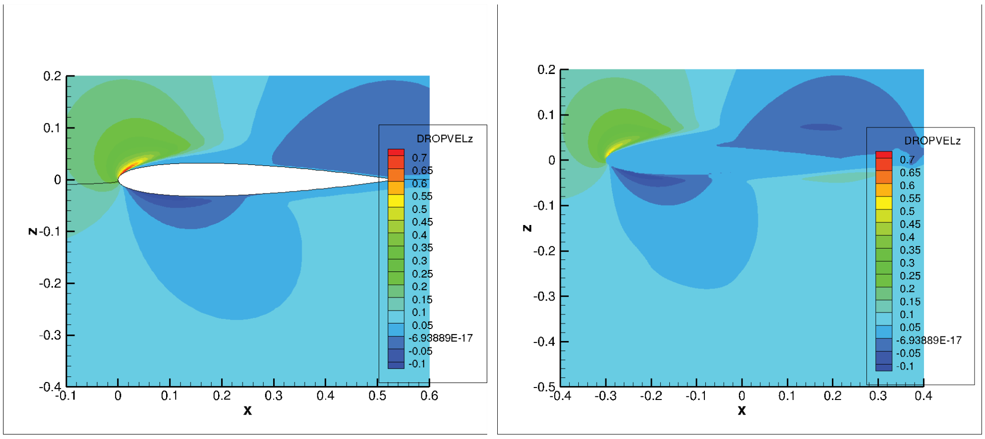


FIGURE 11.11 – Droplet u_y, body-fitted grid vs. embedded grid

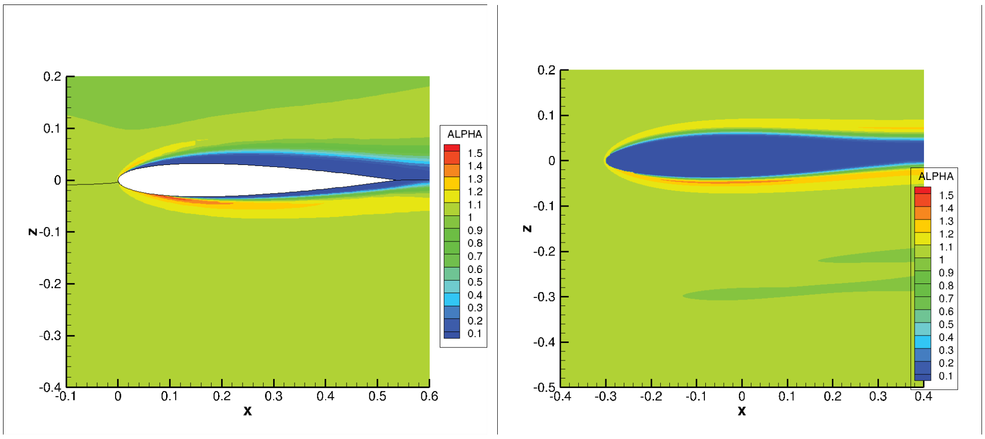


FIGURE 11.12 – Droplet volume fraction α, body-fitted grid vs. embedded grid

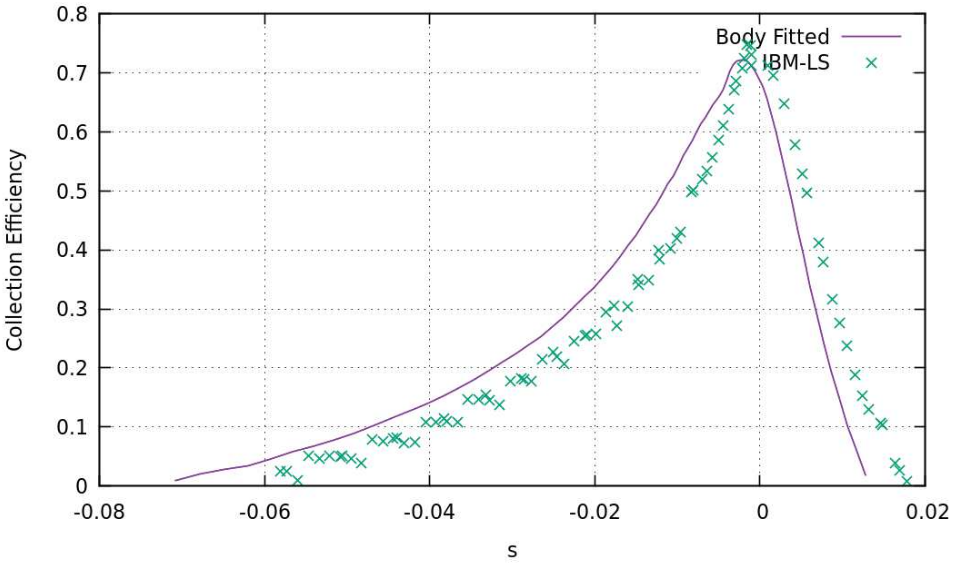


FIGURE 11.13 – Comparison of collection efficiency β

11.5 Conclusion

In this chapter we validated the IBM-LS framework, for the air-flow velocity enforcing at the wall, and for the droplet capture adaptive boundary condition. First a Laplacian test case, whose analytic solution is known a priori, was investigated. When coupled with an LS the method showed converging results. When the level-set was moving, the importance of reinitialization and local conservation for reinitialization were found essential. The one seventh power-law velocity profile was found more accurate than pure penalization for high Reynolds numbers. Which resulted in acceptable air velocity close to the wall, to which ice accretion is highly sensitive. Although the velocity profile was not exactly what anticipated, it was suggested that finer grids might alleviate the problem, chimera grids being highly appropriate. Turbulence modeling through the Spalart-Allmaras model proved difficult with coarse grids, taking into consideration that no turbulence wall modeling is utilized. The droplet capture was also investigated and was rather acceptable, with small deviations. In other words, the methodology is indeed capable of producing satisfying results, given the huge contrast in the near wall mesh quality between the body-fitted and the embedded grid. The problem currently is to couple the existing LS framework with an ongoing implementation of dynamical chimera patch superposition, which could ensure a high grid quality close to the wall. The LS framework is confirmed to work as expected, producing satisfying results. Yet, the said dynamical chimera patching is not ready.

Chapitre 12

MULTI-STEP ICE ACCRETION

In this chapter we present results obtained for multi-step ice accretion. The only test case provided is for rime ice configuration. The test case has been presented in the previous chapter, and effects of different aspects has been provided. In this chapter we discuss the resulting ice shape, with different number of shots.

12.1 Naca0012 Rime Ice

The NACA0012 of chord $c = 0.537$, is subjected to icing, the body-fitted grid solution being available. Different variables of the flow are : $\alpha = 4°$, $M = 0.3168$, $Re = 9.02 \times 10^6$, $T = 250.37$, mean droplet diameter $d = 20\mu m$, $LWC = 0.55g/m^3$ and a total exposure time of $t = 420s$. First the resulting ice shape for a single shot is compared with result from the body-fitted grid.

We first compare the results obtained for the embedded grid with the Spalart-Allmaras and without it. Fig. 12.1, shows the resulting ice shape, using a single shot. No significant difference is discernible, except for slight variations. Thus work needs to be done, for it is impossible for the NS equations to produce accurate results with no averaging or filtering of any sort on the turbulent variations. In other words, it is unexpected and unreliable that the turbulence diffusion have insignificant effects.

Following that we compare the results of both body-fitted and embedded grid ice accretion. In Fig. 12.2 we present the resulting ice shapes. The slight differences between both results is justified by the fact that the mesh quality is highly in contrast between both grids. Add to that the effect of the velocity profile interpolation used with the embedded grid. The responsible variable being the ice velocity, is shown in Fig. 12.3 and 12.4. The ice velocity is slightly greater in the embedded grid, which is expected since the collection efficiency was also greater.

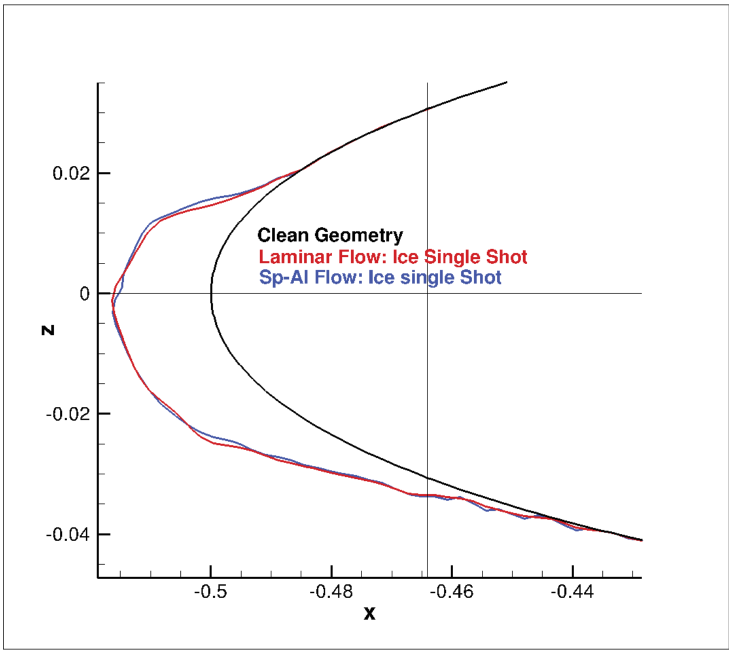


FIGURE 12.1 – Comparison of Spalart-Allmaras vs. no turbulence on the ice shape.

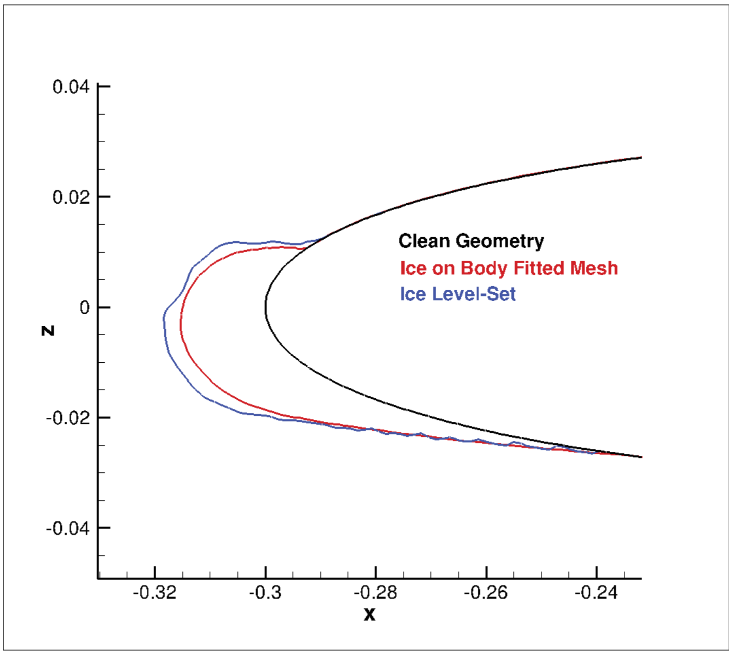


FIGURE 12.2 – Comparison of Ice shape obtained from body-fitted grid, and embedded grid, with a single ice shot.

Lastly we resolve the last configurations using different number of layers. The total exposure time is divided by the number of shots. For two shots each has an exposure time of 210s, and for five shots 84s. In Fig. 12.5, we plot the icing shape obtained using different number of shots.

Single step ice accretion models tend to result in an excess of ice formation [102]. Results obtained in [102] predicted that multi-step ice modeling reduces the excess of

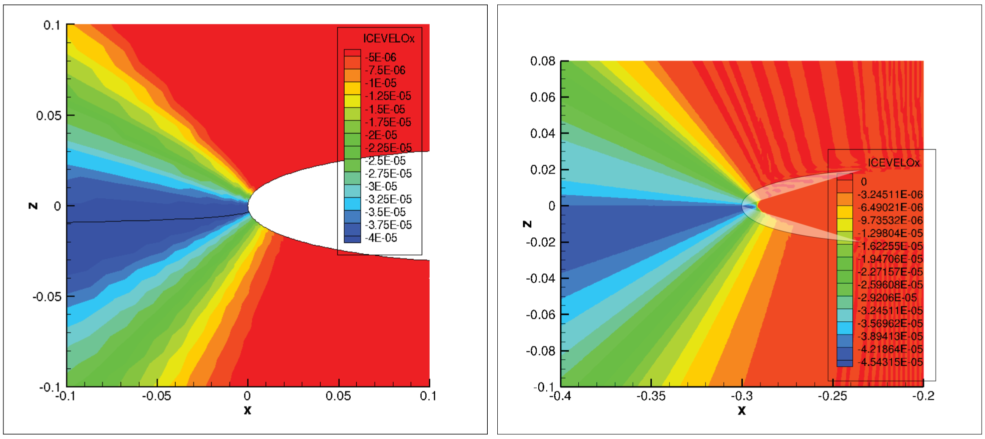


FIGURE 12.3 – Comparison of v_{ice_x} from body-fitted grid vs. embedded grid

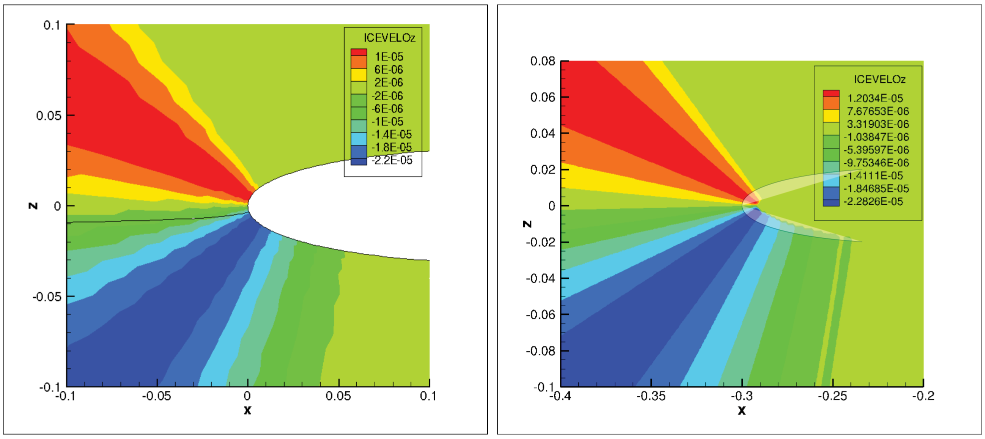


FIGURE 12.4 – Comparison of v_{ice_y} from body-fitted grid vs. embedded grid

ice away from the leading edge, away from the stagnation point, which is in agreement with our results. The maximum thickness of ice not changing when changing the number of shots is expected by the fact that it depends solely on the flow conditions. $\delta_{ice} = v_ice \times t_{exposure} = LWC \times V_\infty \times \beta \times t_{exposure}/\rho_{ice}$. The high irregularities appearing for a higher number of shots are a result of the coarse grid close to the wall. It has been our intention to alleviate this problem by utilizing chimera grids to improve the results, but this is still in progress.

Lastly we show in Fig. 12.6 the ice accretion in five shots. It is interesting to note that the single shot dictates the size of the total wet zone. By advancing the leading edge, the upper part becomes shaded, which justifies the reduction in ice size when simulating with more than one shot.

12.2 Conclusion

We finalize this chapter by summarizing the important points. Only rime ice configurations are available/conceivable. The test case of NACA0012 showed that the ice form

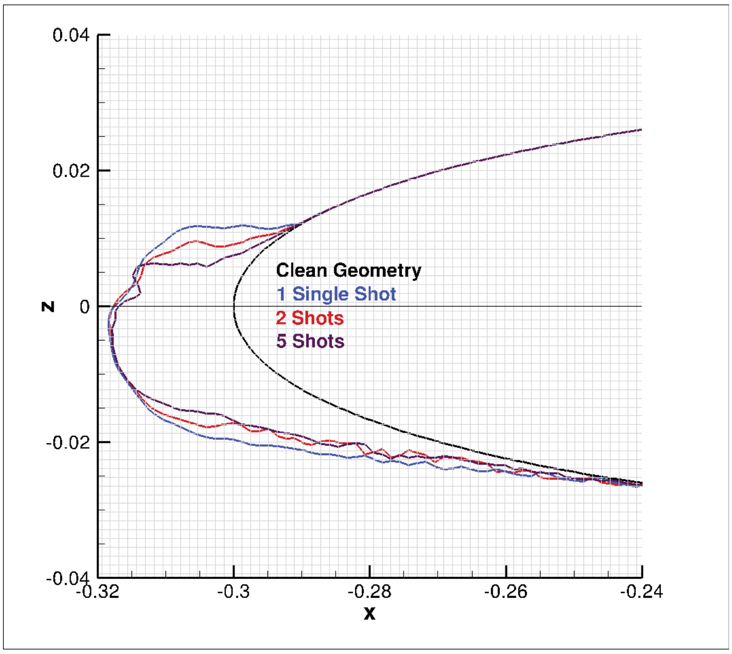


FIGURE 12.5 – Multi-Step ice formation using different number of ice shots.

was insensitive to the averaging of NS equations when coupled with the Spalart-Allmaras turbulence model. Refinement of the mesh, could alleviate this unexpected behavior. The resulting ice shape, using the IBM-LS grid, was slightly in accord with the form obtained using body-fitted turbulent grid. A small overestimation of the ice thickness resulted from the overestimated collection efficiency. In turn, the icing velocity was shown to be slightly overestimated as well, being related to both. The effect of increasing the number of shots is in accordance with results obtained in the literature. Where, the higher the number of shots are, the less the excess of ice away from the leading edge is. The resulting ice shape being highly irregular dictates the use of finer grids close to the wall, motivating the use of chimera grids.

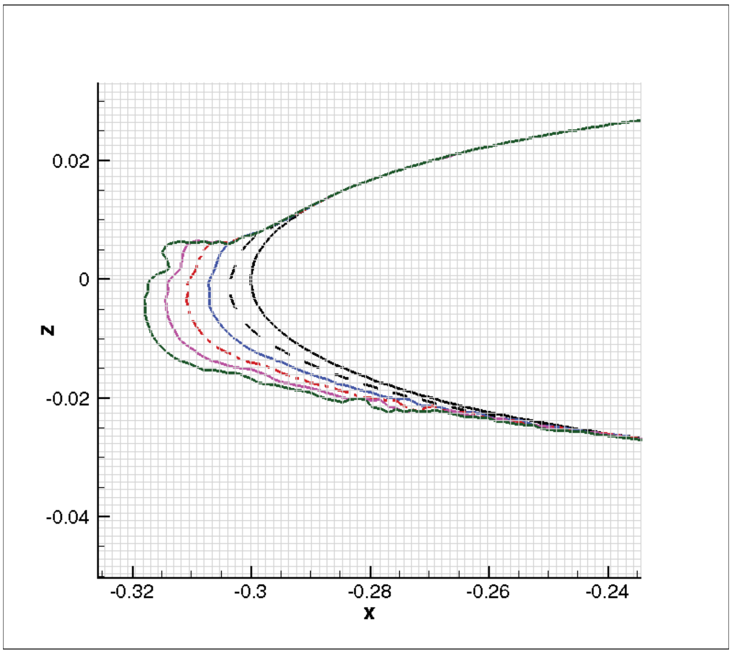


Figure 12.6 – Multi-Step ice accretion, step by step, 5shots.

Chapitre 13

CONCLUSION and PERSPECTIVES

This thesis work focused in implementing an embedded grid platform to run multi-step, multi-layer or multi-shot ice accretion simulation. The real aim was to avoid the need of remeshing that proves problematic for certain situations. The platform is developed in multi-block and parallelized withing NSMB. The main modules are : IBM-LS module, air-flow solver and droplet capture solver.

The IBM-LS module initializes the LS related functions at the start, generates the required information for the IBM-LS, and is called at the end to advect the icing velocity, to advect the geometry representing ice accretion, and to reinitialize the distanced LS function. The air-flow solver than utilizes the IBM-LS information to enforce a velocity profile close to the wall to replicate the no slip boundary condition. And lastly, the droplet capture module transports the droplets and utilizes the IBM-LS information as well to achieve the capture procedure in the wet zones.

An extensive analysis on the importance of different aspects of the Level-Set equations was presented in discussed. The WENO5 coupled to RK3 schemes showed good performance. The reinitialization requires the local conservation fix proposed by Sussman and Fatemi to reduces topological degradations. The Level-Set equations worked greatly over chimera superposed grids, both advection, and reinitialization. Then, we validated the IBM-LS framework using simple test cases. The air-velocity profile imposed via IBM-LS was also compared to results obtained from body-fitted grids, and showed better results than with penalization. It was concluded that better formulations of the velocity profile are needed, and that refinement of the grid close to the wall could improve the results. The droplet capture was also investigated, and results in terms of collection efficiency were acceptable, with slight deviations. This method proved very good despite the said deviations, since the quality of the two grids used for the comparison are unequal.

The algorithm was then put to the test ; ice accretion was compared between two grids, one embedded, the other turbulent body-fitted grid. The resulting ice shape is rather acceptable, with very slight excess resulting from the slight excess in the collection efficiency, mainly due to the grid being used. The ice shape using different number of shots was presented at last.

The LS or IBM-LS framework presented in this work produced exceptional results given the coarseness of the grids and the simplifications used. Certain improvements are highly recommended at this stage. First is to test these modules using a chimera grid, that provides good mesh quality close to the wall. A colleague is in the process of making all the tables or arrays in NSMB dynamically re-allocatable using pointers instead of long vectors. This would allow for dynamics imposition of chimera grid blocks or patches at regions of interest, Cartesian or curvilinear. The second important improvement is to develop a more thorough approach for interpolation of the velocity close to the wall. The refinement should be coupled to the type of velocity profile used, to ensure that the cells close to the wall fall in the intended sublayer of the turbulent boundary layer. The third important action is adding glaze ice modeling. Under the current IBM-LS framework, where we try mostly to avoid remeshing the flow domain, the best option is by using reconstructed points at the wall to form triangular or linear segments. Following that, by using unstructured mesh method, the skin mesh is to be used to solve SWIM equations and evaluate the icing thermodynamics. The interface can be reconstructed by using the information related to the LS function ϕ at each cell of the domain. From each cell we can project a point on the interface via the following equation :

$$\mathbf{X}_{wall} = \mathbf{X} - \phi\nabla\phi \tag{13.1}$$

Using only ten percent of the cells the surface depicted in Fig. 13.1 was reconstructed. The following task would be to interpolate the state at each point from neighboring fluid points.

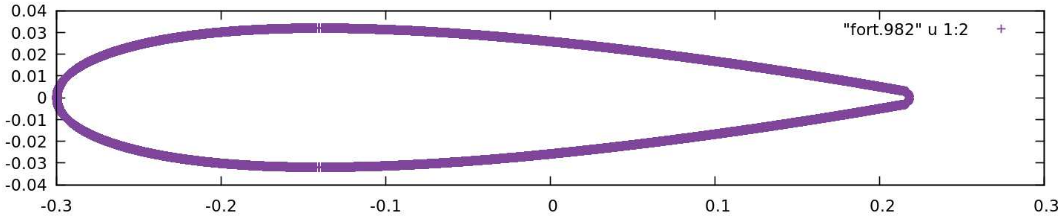


Figure 13.1 – Reconstructing the skin mesh.

Chapitre 14

Additional Complementary Work

This chapter is dedicated to the additions and modifications made after the submission of the thesis and presented at the oral defense. Three major modifications were considered, new progress was made in each and new perspectives are sought. The first major modification concerns the one seventh power law used to enforce the BC by reconstructing the tangential velocity at the wall. Ideally, a wall modeling of the turbulence is more suitable, the code was modified to incorporate a wall model for the airflow BC. The second major modification, far from being a modification is a validation of the methodology on a chimera grid, along with other test cases from the literature. The wall model, the droplet capture model and all the pertaining models are shown to work perfectly for a chimera grid. Additional test cases show better results than those initially presented in the manuscript. The third major modification, still needing some development, concerns the glaze-ice situations : a model for the heat transfer using the SWIM model, applied to a reconstructed surface grid is presented.

These modifications are further developed in the following sections.

14.1 Wall Model

In section 9.2.4, we discussed the treatment of the air-flow boundary condition. Penalization alone was found incapable of producing an acceptable turbulent boundary layer velocity profile, and was replaced with reconstruction methods. The simplest method was to assume linear interpolation from the points in the flow to the required close-to-wall point which were called IB points. The second method was to follow the power-law. The one seventh power-law was indeed a better choice and provided great results. The one seventh power-law generally approximates the logarithmic distribution that we would expect near the wall. Thus the grid points should fall in the logarithmic layer, not further, nor closer. Another limiting factor is that the flow is assumed to be always attached with no flow separation [26].

Since all icing situations are at high-Reynolds number regime, the use of linear interpolation is highly lacking. Moreover, the IBM-LS grid used in our NSMB solver is strictly structured, with no cell-splitting mesh refinement possible. Consequently, there is no guarantee on the cell sizing in the near to the wall region ; more specifically, the wall distance can not be enforced at all the cells, even with chimera blocks refinement. Everything can be done in an ideal situation, but when the ice formation takes horrendous formations no structured grid can be cast fitting orthogonally to the wall. Further more arguments can be made to justify the use of turbulent wall modeling, especially for icing where the evolution of the flow field with the accumulated ice is highly dependent on the turbulence treatment close to the wall.

We opted to use the numerical wall function proposed by [33], motivated by its application within an IBM code by [39].

The wall model is of the form :

$$\frac{u_t}{u_\tau} = \frac{1}{\kappa} ln\left(1 + \kappa y^+\right) + c\left(1 - e^{-y^+/d^+} - \frac{y^+}{d^+} e^{-by^+}\right) \tag{14.1}$$

where u_t represents the tangential component of the velocity to be imposed at the IB point.

$$b = \frac{1}{2}\left(\frac{d^+\kappa}{c} + \frac{1}{d^+}\right)$$

$$c = \frac{1}{\kappa} ln\left(\frac{E}{\kappa}\right)$$

$$y^+ = \frac{u_\tau y}{\nu}$$

$$u_\tau = \sqrt{\frac{\tau}{\rho}}$$

where $\kappa = 0.4187$, $E = 9.793$ and $d^+ = 11$ [33].

The main idea behind the method is to estimate the value of the shear velocity at the image point **im** within the fluid domain where the state vector is obtainable through interpolation as demonstrated in section 9.2.4, refer to Fig. 14.1. The shear velocity u_τ can be obtained from Eq.14.1, using an iterative Newton-Raphson method. By redefining the variables :

$$\kappa^* = \kappa\frac{\delta}{\nu}$$

,

$$b^* = b\frac{\delta}{\nu}$$

we rewrite the equation as :

$$h\left(u_\tau\right) = \frac{u_t}{u_\tau} = \frac{1}{\kappa} ln\left(1 + \kappa^* u_\tau\right) + c\left(1 - e^{-\frac{\delta}{\nu d^+} u_\tau} - \frac{\delta}{\nu d^+} u_\tau e^{-b^* u_\tau}\right) \tag{14.2}$$

where the derivative of the function h becomes :

$$h' = \frac{dh}{du_\tau} = -u_t/u_\tau^2 - \frac{\kappa^*}{\kappa}\left[\frac{1}{1+\kappa^* u_\tau}\right] - c\frac{\delta}{\nu d^+}\left[e^{-\frac{\delta}{\nu d^+}u_\tau} - e^{-b^* u_\tau} + b^* u_\tau e^{-b^* u_\tau}\right] \tag{14.3}$$

We initiate the shear velocity for the iterative method by assuming a linear velocity profile.

$$u_{\tau,0} = \sqrt{\frac{\nu u_t}{\delta}}$$

The Newton-Raphson method then iterates the shear velocity until convergence, which is usually achieved within 10 iterations.

$$u_{\tau,i} = u_{\tau,i-1} - h/h'$$

Having calculated the value of the shear velocity at the im point, which is fundamentally assumed to be equal to that of the IB point, the tangential velocity at the IB point is calculated from the wall function through Eq.14.1. The normal component of the velocity is interpolated linearly. The state vector is reconstructed and reprojected in the Cartesian coordinates.

14.1.1 Wall Model : Spalart Allmaras Turbulence Model

In section 9.1.1, we proposed a linear profile for the calculation of the eddy viscosity. However, an improvement can be made on this aspect, as follows :

The one-equation Spalart-Allmaras model transporting the eddy viscosity requires the calculation of the turbulent viscosity at the IB point μ_t given by a mixture length formulation :

$$\mu_t = \rho\kappa^2 y^2 \left(1 - exp\left(\frac{y^+}{26}\right)\right)^2 \frac{U_\tau}{\kappa y} \tag{14.4}$$

$\tilde{\nu}$ is related to μ_t through :

$$\mu_t = \rho\tilde{\nu} f_{v1} \quad , \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \quad , \quad \chi = \frac{\rho\tilde{\nu}}{\mu} \tag{14.5}$$

which gives a forth order equation on χ as :

$$\chi^4 - \mu_r\chi^3 = \mu_r C_{v1}^3 \quad , \quad \mu_r = \mu_t/\mu \tag{14.6}$$

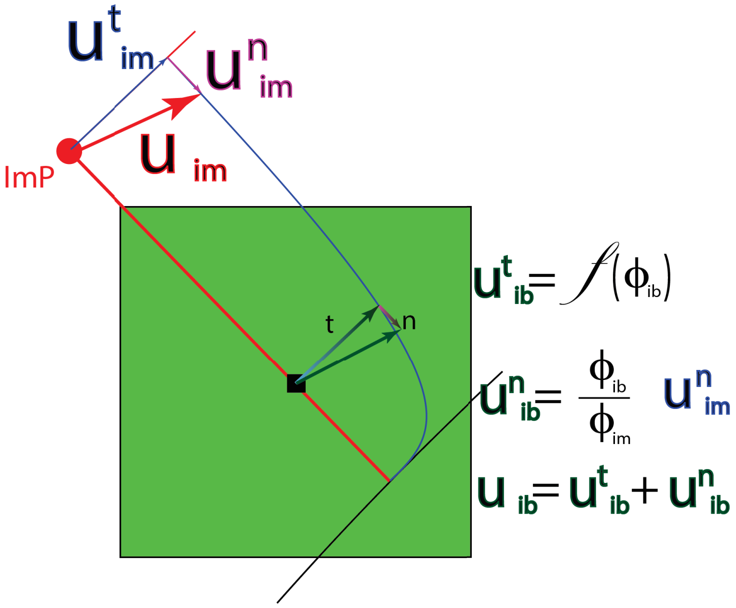

u^t_{im}
u^n_{im}
ImP
u_{im}
t
n
$u^t_{ib} = f(\phi_{ib})$
$u^n_{ib} = \frac{\phi_{ib}}{\phi_{im}} u^n_{im}$
$u_{ib} = u^t_{ib} + u^n_{ib}$

FIGURE 14.1 – Wall model estimation at im point, enforcing at IB point.

An analytical solution results in four different roots, only one of which holds a real positive physical value, which after simplification takes the following form :

$$\chi = \frac{2^{\frac{1}{6}}3^{\frac{1}{2}}\left[2^{\frac{5}{6}}3^{\frac{1}{2}}\mu_r C_3^{\frac{1}{6}}C_4^{\frac{1}{4}} + C_4^{\frac{3}{4}} + 2\left(3^{\frac{3}{2}}2^{\frac{3}{2}}\mu_r^3 C_3^{\frac{1}{2}} - C_4^{\frac{1}{2}}\left(-3\mu_r^2 2^{\frac{2}{3}}C_3^{\frac{1}{3}} - 48C_1 + 2^{\frac{4}{3}}C_3^{\frac{2}{3}}\right)\right)^{\frac{1}{2}}\right]}{24C_3^{\frac{1}{6}}C_4^{\frac{1}{4}}} \tag{14.7}$$

$$C_1 = C_{v1}^3\mu_r \quad , \quad C_2 = C_{v1}^3\mu_r^3 \tag{14.8}$$

and

$$C_3 = -27C_2 + 3^{\frac{3}{2}}\left(256C_1^3 + 27C_2^2\right)^{\frac{1}{2}} \quad , \quad C_4 = 2^{\frac{10}{3}}C_3^{\frac{2}{3}} - 192C_1 + 6\mu_r^2 2^{\frac{2}{3}}C_3^{\frac{1}{3}} \tag{14.9}$$

It should be stated that many cleaning and grouping of the terms is possible but we leave here the result obtained using maple. The reason behind this is : this form is highly unstable with different orders of magnitudes resulting in very bad calculations of the resulting positive root, we wouldn't want to instigate the thought that further simplification is imprecise and the reason of the wrong calculation. Consequently, we preferred to use an iterative procedure to find the positive root, which converges within 5 to 10 iterations. The starting value should be as high as possible to avoid converging to the negative root. The use of tables, and correlations was also tested but was found not precise and more costly than 10 iterations.

Implementing the wall model to other turbulence models is out of the scope of our work. However, the same methodology can be used, and further implementation of different turbulence models can be followed through.

14.1.2 NACA0012 Wall Model : 1st test case

The first investigated test case is the same one used in section 11.2. A NACA0012 of chord $c = 0.537$, for which the body-fitted grid solution is available at an angle of attack α of 4°, a Mach number of $M = 0.3168$, a Reynolds number of $Re = 9.02 \times 10^6$, a static temperature of $T = 250.37$ is studied. The tangential velocity profile perpendicular to the wall at the top of NACA0012 is compared for the the different methods used to enforce the BC.

In Fig. 14.2 we observe the improvement on the profile by using the wall function on a coarse grid. In Fig. 14.3 we also observe a slight improvement when using a finer grid. The one-seventh power law is improved as expected. The IB points are forced closer to the wall which enforces them to lay closer to the logarithmic layer. Yet, the wall function still results in better profile overall.

The resulting eddy viscosity is also shown in Fig. 14.4. Although the results seem quite incomparable, we observe that the wall function dissipates the eddies downstream as expected, whereas the power law is incapable of reproducing the diffusion effect of the RANS behavior. The difference between the wall function and body fitted resulting eddy viscosity is due to the lack of the roughness extension with the wall model.

Another important aspect is the improvement on the collection efficiency required from the wall modeled air flow. We present in Fig. 14.5 the resulting collection efficiency with a mean valued diameter of droplets of $MVD = 20\mu m$. The improvement is quite clear ; the stagnation point, where icing is as its maximum is better correlating with the body fitted results, the overall distribution is also improved. A slight deviation at the top of the wing is the result of the lack of a roughness extension, with which the body fitted is augmented.

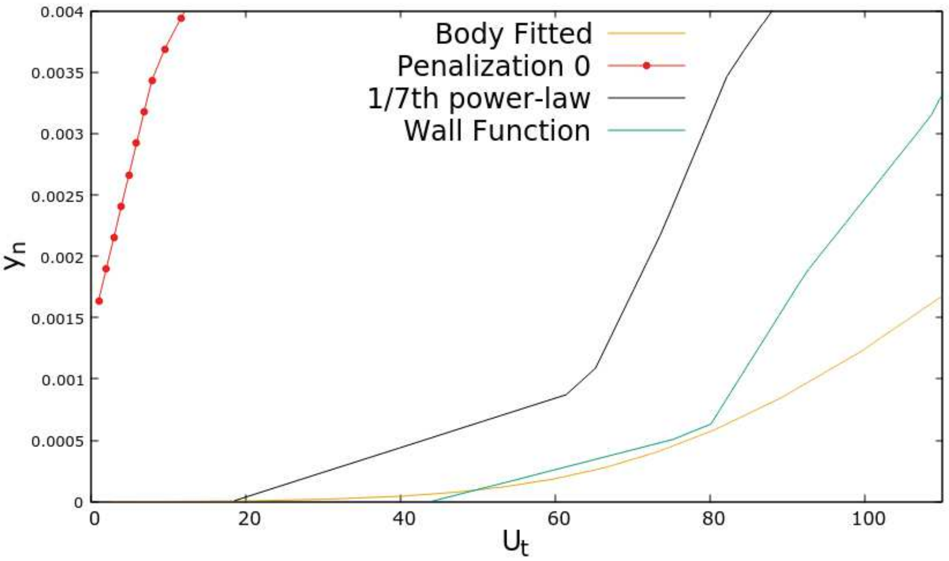


FIGURE 14.2 – Tangential velocity profile perpendicular to the wall at the top of NACA0012.

14.2 Chimera + Additional test case

14.2.1 Chimera

After slight modifications and debugging, the test case used in the previous section is studied using a chimera grid overlapping on the surface of the wing and the ice shape. It is very important to verify that the chimera interpolation won't deteriorate the results where we enforce the wall model, the droplet calculation, and the ice/Level-Set evolution.

The resulting x-component velocity field is shown in Fig. 14.6. The results are identical with slight differences, notably on the boundary layer thickness, but the difference is due to the mesh refinement rather than the overlapping blocks.

14.2.2 NACA0012 Wall Model + Pure Rime : Additional test case

The only multi-step ice accretion test case presented in the manuscript showed a slight difference on the maximum ice thickness. The body-fitted grid resulted in an ice layer 16.6% less than that using the IBM-LS grid. Whereas the resulted collection efficiency was only of 0.5% difference. The reason behind this is not a failure on the LS advection equations as will be detailed here.

Under pure rime ice assumption, the maximum ice thickness could be calculated as follows :

$$h_{ice} = \mathbf{v}_{ice} * t_{exposure}$$

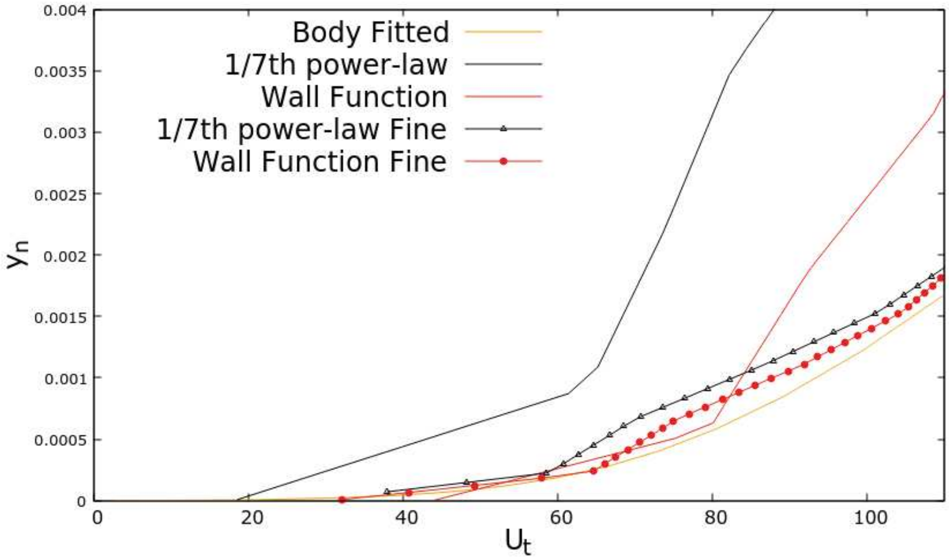
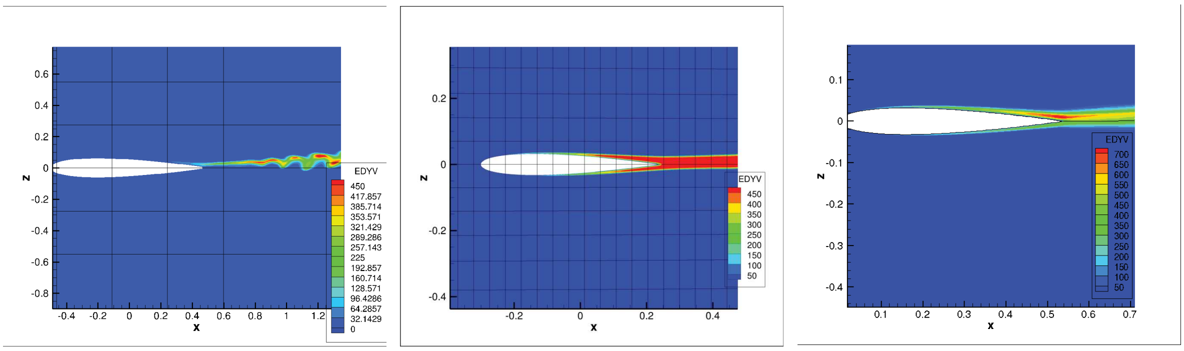


FIGURE 14.3 – Tangential velocity profile perpendicular to the wall at the top of NACA0012.



Power law

Wall function

Body fitted

FIGURE 14.4 – Eddy viscosity around NACA0012.

$$\mathbf{v}_{ice} = \frac{\dot{m}_{ice}}{\rho_{ice}}$$

$$\dot{m}_{ice} = LWC\, V_{\infty}\beta$$

For the case under study :

h_{ice} should be around 0.0183 to 0.0185 m

β	0.72 - 0.73
lwc	0.55 g/m^3
V_{∞}	100.5 m/s
ρ	917 kg/m^3
$t_{exposure}$	420 s

However, from Fig. 12.2 we find the following :

For the IBM-LS under Rime ice we obtained : $h_{ice} = 0.0184$ m

While for Body Fitted we obtained : $h_{ice} = 0.0158$ m

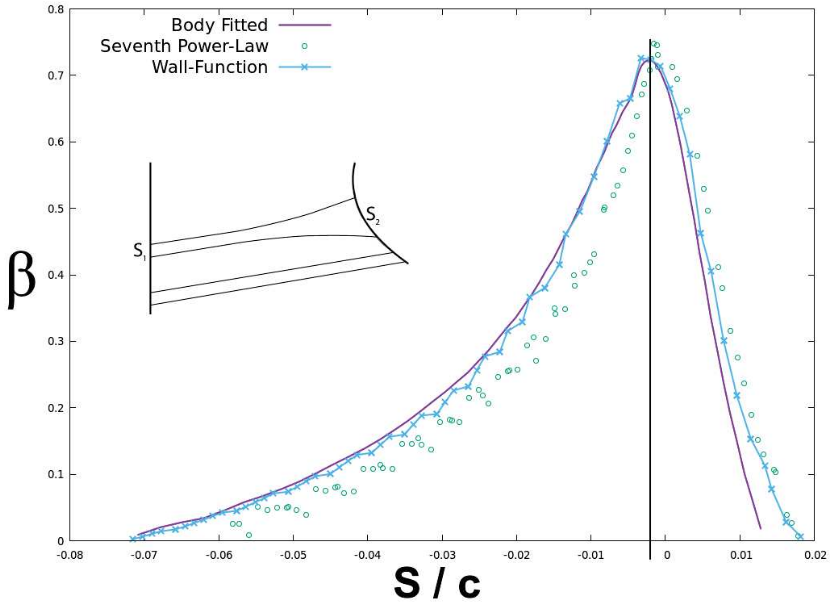
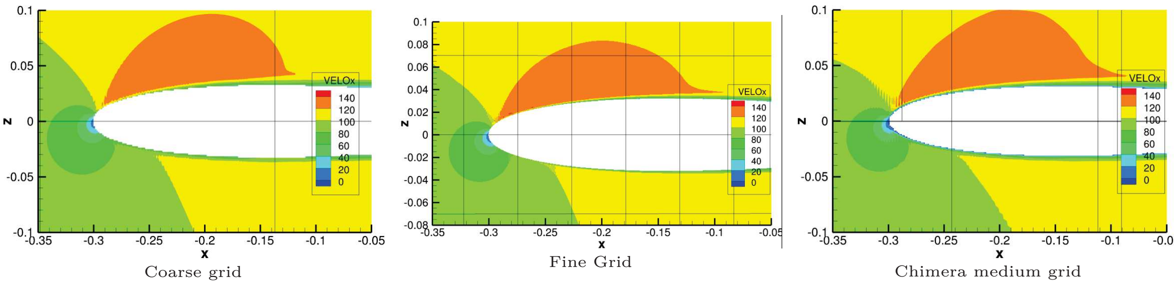


FIGURE 14.5 – Collection efficiency NACA0012. compare with FIG. 11.13



Coarse grid

Fine Grid

Chimera medium grid

FIGURE 14.6 – X-component of the velocity around NACA0012

Which means, the IBM-LS results fall within the expected physical values, whereas the body-fitted results did not. The main reason was that the body-fitted was augmented by the SWIM solver which checks the heat transfer at the surface, non pure rime-ice. The temperature for this case wasn't low enough to ensure a pure rime-ice situation.

To confirm this result we opted to study another test case, one for which the temperature is very low. The case presented here studied by[47] and [25] is a NACA0012 with the conditions shown in the following table.

V_∞	$67\ m/s$
P_∞	$101325\ Pa$
T_∞	$244.8\ K$
α	4
LWC	$1\ g/m^3$
MVD	$20\ \mu m$
$t_{exposure}$	$360\ s$

The only result available in both articles, [47] and [25], is the final complex ice shape. The resulting ice shape using the wall model approach introduced in the previous section is shown in FIG.14.7. If the power law was rather used here the results wouldn't differ as much, as seen in the previous section where only the stagnation point was shifted, and the collection efficiency was slightly increased by 2.5%. It is notable that in this test case the ice thickness is exactly the same as found by the two authors as expected. It is due to the very low temperature that ensures pure rime ice formation. Moreover, the wetted surface is also identical. The only difference here is an excess ice on the top. This problem however, is due to the use of a single ice shot simulation.

In Fig. 14.8 we present the resulting ice shape when using different number of shots. It was stated and explained in the manuscript that using more shots generates a shaded region on the top which reduces the accumulation of ice the further we move from the stagnation point. We also observe a rough ice shape on the bottom side as the number of shots is increased. The reason behind this shape is the size of the mesh. Currently dynamic refinement through dynamic chimera blocks super-positioning is still under development. It is rather quite reassuring and encouraging that such heavy scenarios is still conceivable and that the air-flow solver coupled with the wall model is producing satisfactory results for such rough geometry. The power law would fail in treating such topologies. The grid used is shown in Fig. 14.9

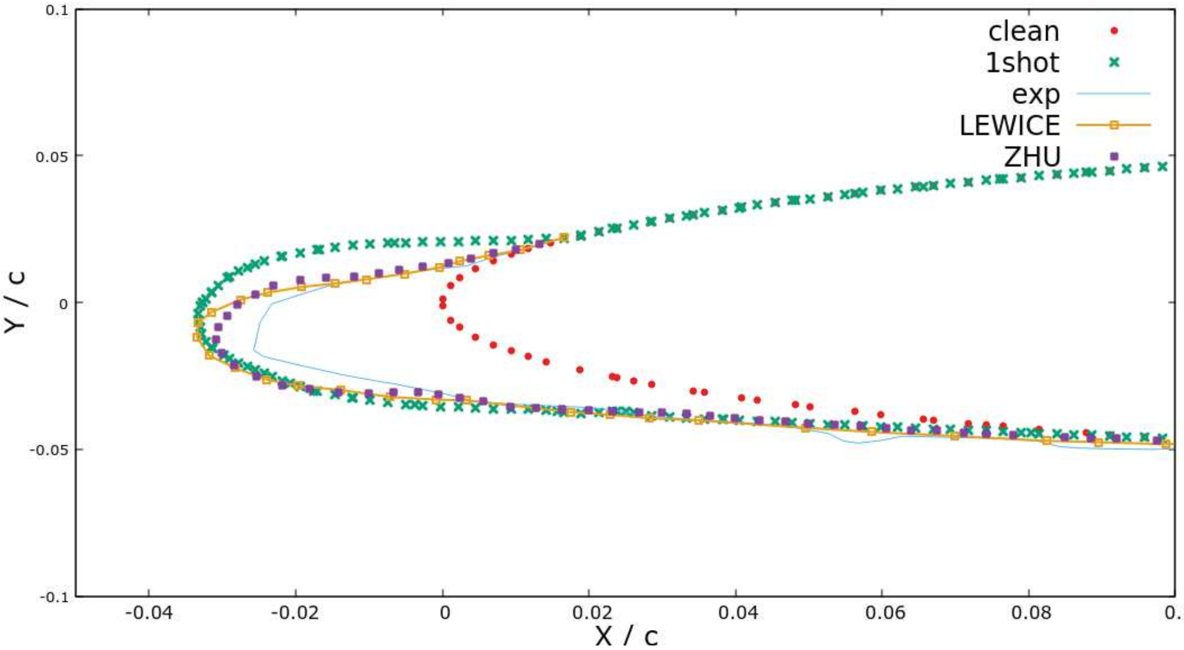


Figure 14.7 – 1 ice shot 360s NACA0012

The resulting ice shape using five shots is compared to the results from LEWICE [47] in FIG. 14.10. We observe that both results are quite similar on many aspects.

In conclusion, the use of the wall model is quite satisfactory if not essentially required. The assumption of rime ice must be taken with great caution. The overall method is very

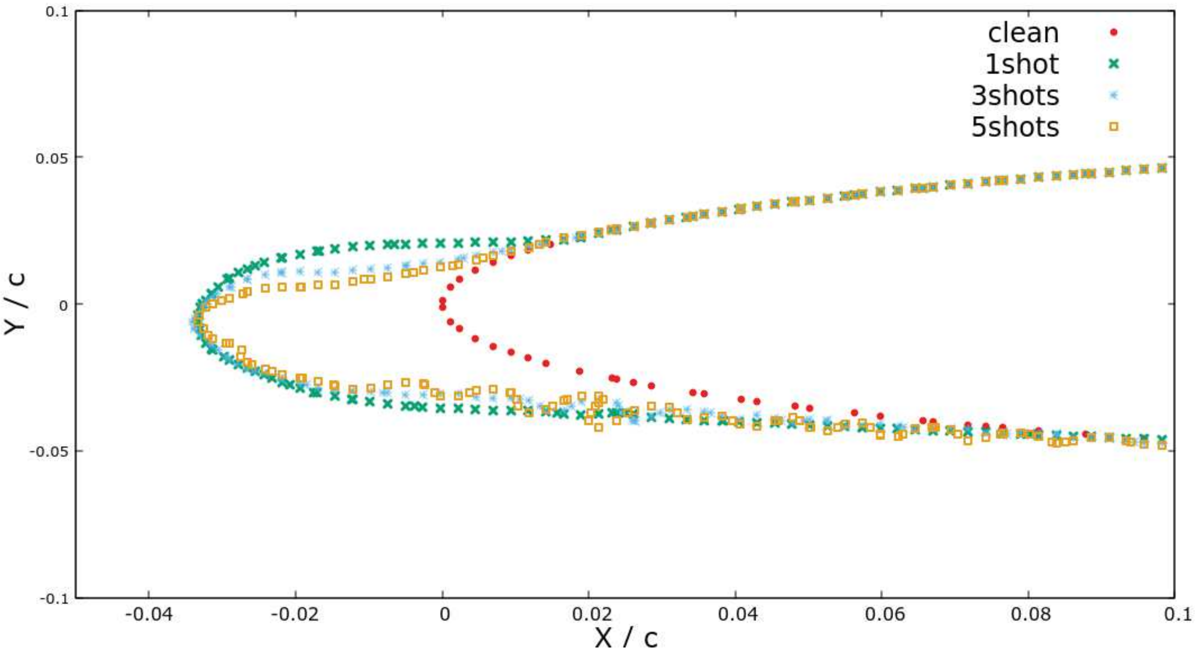


FIGURE 14.8 – Multiple ice shots 360s NACA0012

effective in studying very complex ice shapes. The only drawback is that very complex ice shapes are usually due to glaze ice formations, which are not yet realizable. We present in the following section the first steps towards a SWIM model on a reconstructed surface grid.

14.3 2D Glaze solver

In this section we present the outlay of a two-dimensional glaze solver. The main idea can be further developed for three-dimensional geometries.

The main steps are as follows :

— We start by reconstructing the interface.

— We follow that by rearranging the interface points to define adjacencies. In three dimensions we would need to define the connectivity tables as in unstructured grid or FEM solvers.

— The MPI communications are redefined as connectivities are different from the main flow blocks connectivities. In 2D, blocks through which the interfaces passes share a single cell, or more, depending on the scheme to be used and the size of the stencil. In 3D, these blocks would share edges, and connectivities and communications would have to take this into account. Add to it the required stencil size.

— The SWIM PDE equations complemented by its four compatibility relations coupled with the the try and error variable reduction procedure are solved on the resulting skin mesh. Refer to section 3.5

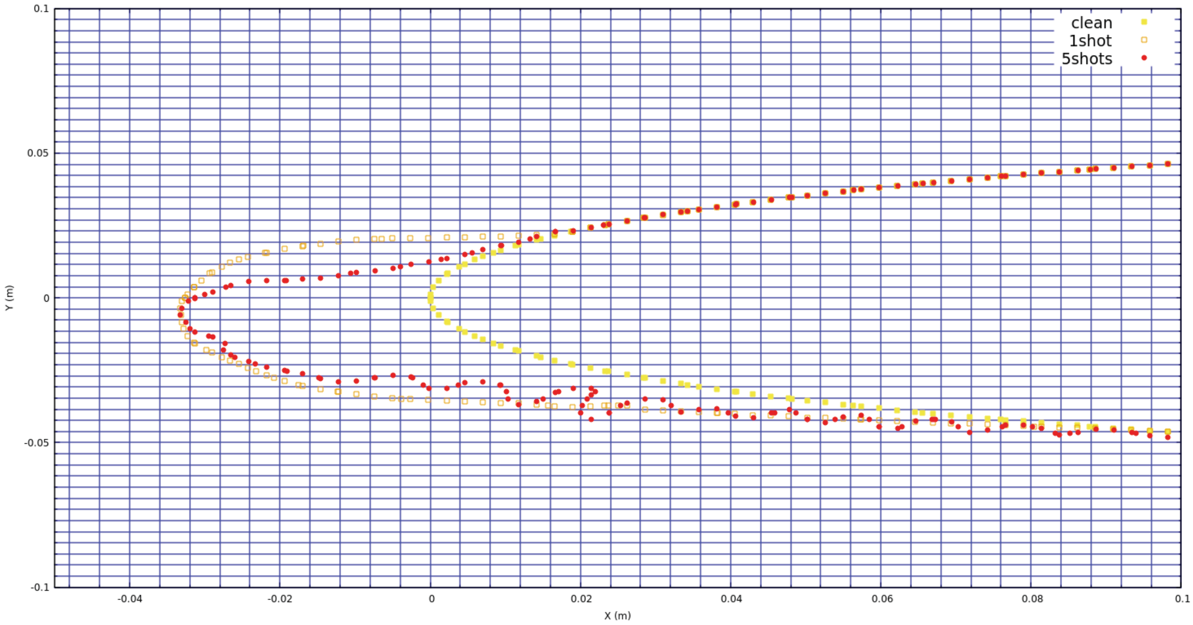


Figure 14.9 – Close up view of the grid NACA0012

14.3.1 Interface reconstruction

The first step is to reconstruct the interface by projection. For each cell on the background grid there is a projection on the interface, Fig. 14.11. However, it suffices to project a smaller number of points. We project each IB point onto the interface creating a list of points for which all physical variables are available through projection/interpolation. The coordinates of the surface points are available using :

$$\mathbf{X}_{SP} = \mathbf{X_{IB}} - \phi\nabla\phi \tag{14.10}$$

The coordinates of these surface points are saved in an array **SP**, Fig. 14.12.

14.3.2 Adjacencies/Rearrangement/Connectivities

The next step treats the definition of connectivity between the different surface points saved in the array **SP**. In a three-dimensional code the method would resemble unstructured grids connectivity tables for each grid point. However for simplicity, in two-dimensions the points could be rearranged to form a structured like grid. Instead of doing a full rearrangement of the coordinates, we opt to only save the indexes in an arranged list. In such way the connection to the adjacent IB points is not lost.

We start by defining the first point as the one with the lowest angle from the block's left bottom corner. Then we compare distances to find the closest points from then forward. The arranged list indexes are saved in an array **Ali**, Fig. 14.13.

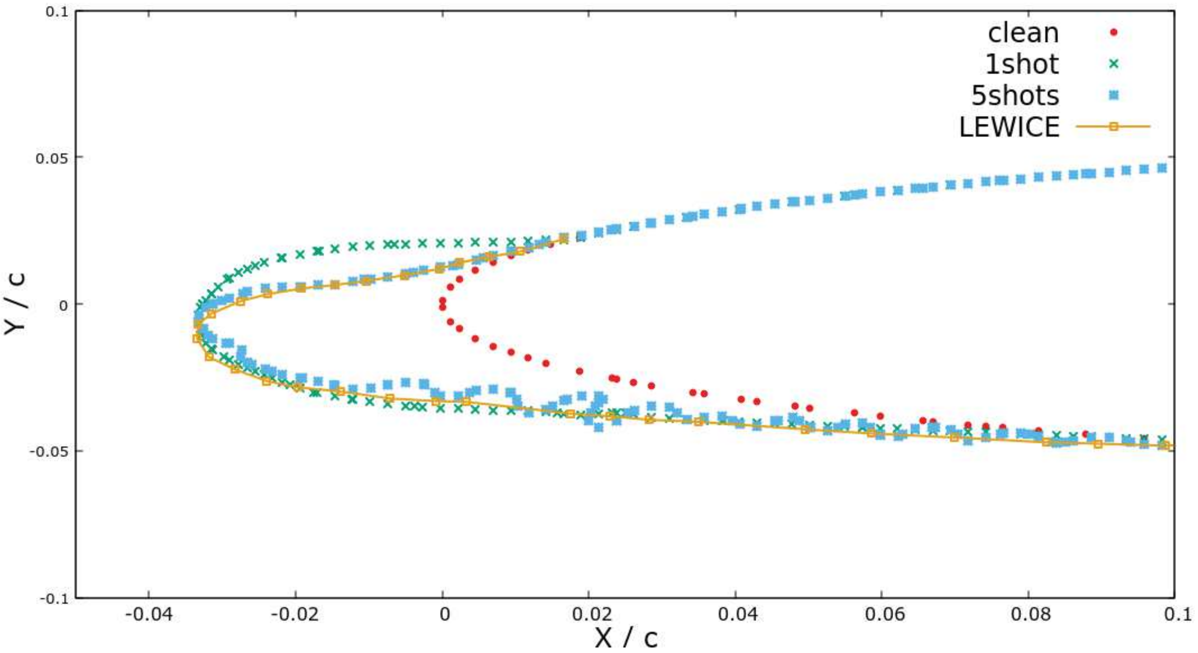


Figure 14.10 – 5 Ice shots compared to LEWICE NACA0012

14.3.3 MPI Reconnection

The MPI communication is initially defined in NSMB by dispatching and dividing the jobs based on the constructed two-dimensional multi block grid, Fig. 14.14. For the reconstructed skin grid the connectivity is not available yet. We communicate the left and right states to all neighbors, neighbors in the sense of the old NSMB MPI. Each block checks minimum distances of its own left and right points to communicated points, and decides if it is connected to sender block or not. If the distance is within range of the local cell size, connection is defined and the rest of the communicated variables are used for the discretization scheme treated afterwards, Fig. 14.15

14.3.4 MPI Reconnection

The SWIM PDE equations presented in detail in section 3.5 are written in the following form :

$$\frac{\partial U}{\partial t} + \nabla . F(U) = S \tag{14.11}$$

with

$$U = [h_f, h_f T], \quad F(U) = \left[\frac{h_f^2}{2\mu_w}\tau_{wall}, \frac{h_f^2 T}{2\mu_w}\tau_{wall}\right], \quad S = \left[\frac{S_1}{\rho_w}, \frac{S_2}{\rho_w C_{p,w}}\right] \tag{14.12}$$

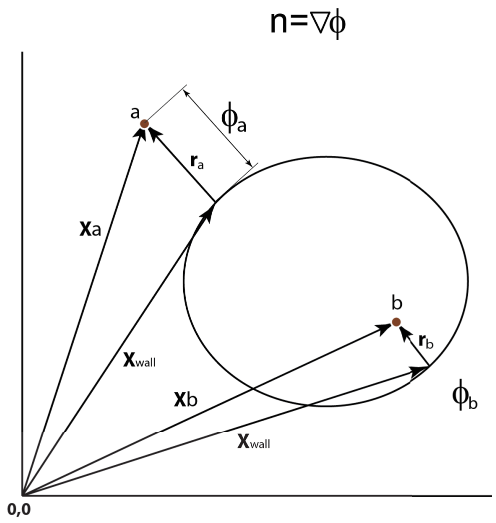


FIGURE 14.11 – Interface reconstruction

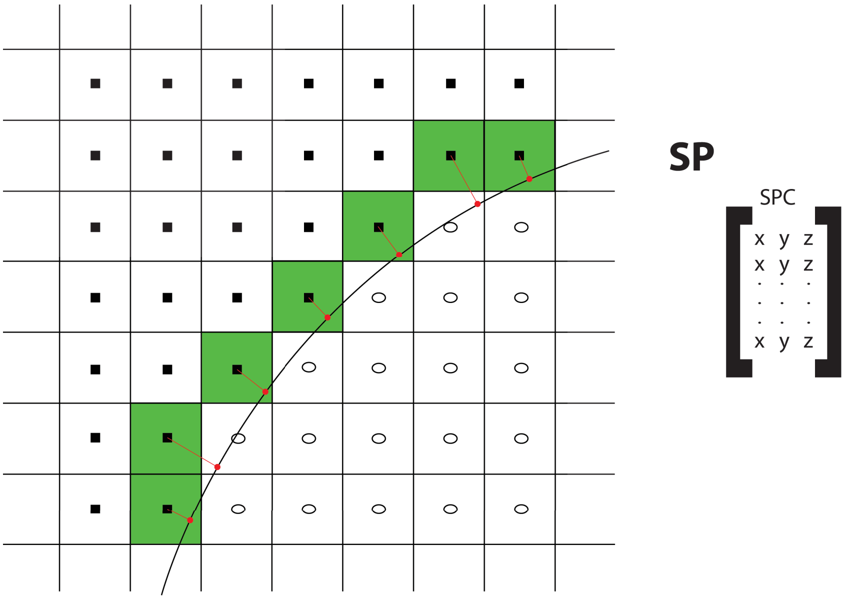


FIGURE 14.12 – Surface points projection

The convective part is discretized using finite volume first order Roe scheme :

$$\Omega_i \frac{\partial U}{\partial t} + \sum_{j \neq i} \int_{\partial \Omega_i} \Phi^{Roe}(U_i, U_j, \mathbf{n}) ds = \Omega S_i \tag{14.13}$$

Ω_i being the volume of node i, which in our case is half the distance between the east and west neighbors, $\Omega_i = 1/2\left(\mathbf{X}_{i+1} - \mathbf{X}_{i-1}\right)$ and the Roe flux at the interface Φ^{Roe} is given by :

$$\Phi^{Roe} = \frac{1}{2}\left(\mathbf{F}(U_i) + \mathbf{F}(U_j)\right).\mathbf{n} - \frac{1}{2}|\mathbf{J}(U_{i+1/2}).\mathbf{n}|(U_j - U_i) \tag{14.14}$$

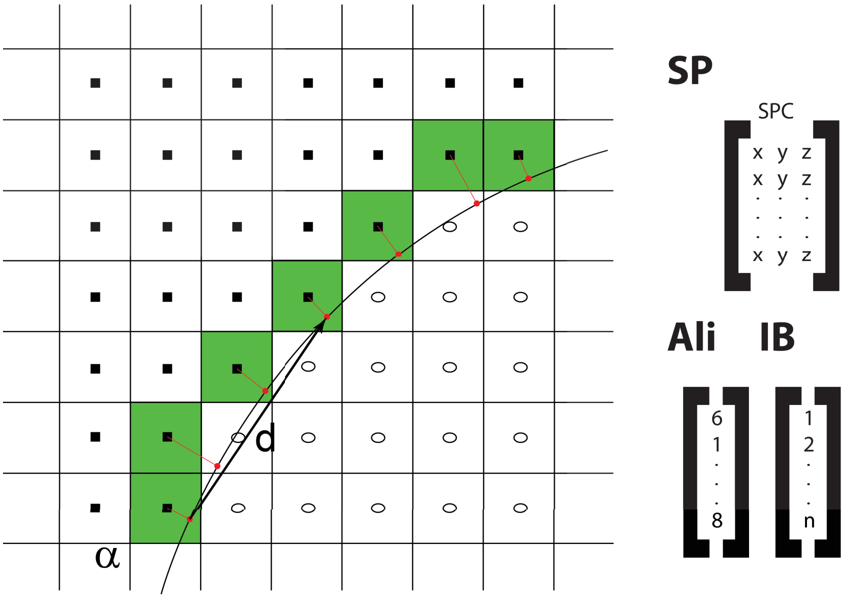


FIGURE 14.13 – Arranged list of indexes Ali

The Jacobien of the system $\mathbf{J}(U_{i+1/2})$ is given by :

$$\mathbf{J}(U_{i+1/2}).\mathbf{n} = \begin{bmatrix} \frac{h_f}{\mu_w}(\tau_{wall} \cdot \mathbf{n}) & 0 \\ \frac{h_f T}{2\mu_w}(\tau_{wall} \cdot \mathbf{n}) & \frac{h_f}{2\mu_w}(\tau_{wall}.\mathbf{n}) \end{bmatrix} \tag{14.15}$$

The shear stress at the wall τ_{wall} is estimated from the wall function. The direction of $\tau_{wall} \cdot \mathbf{n}_{i+1/2}$ is decided from the tangential velocity averaged at the face $i + 1/2$.

Finally, at each node we solve the following system :

$$\Omega \left(\rho_w \frac{\partial h_i}{\partial t} - S_{1_i} \right) + \sum_{nb} R_{1_{nbf}} = 0 \tag{14.16}$$

$$\Omega \left(\rho_w \frac{\partial C_w h_i T_i}{\partial t} - S_{2_i} \right) + \sum_{nb} R_{2_{nbf}} = 0 \tag{14.17}$$

where $R_{x_{nbf}}$, calculated at east and west sides of node i with respect to its neighboring cell, is given by :

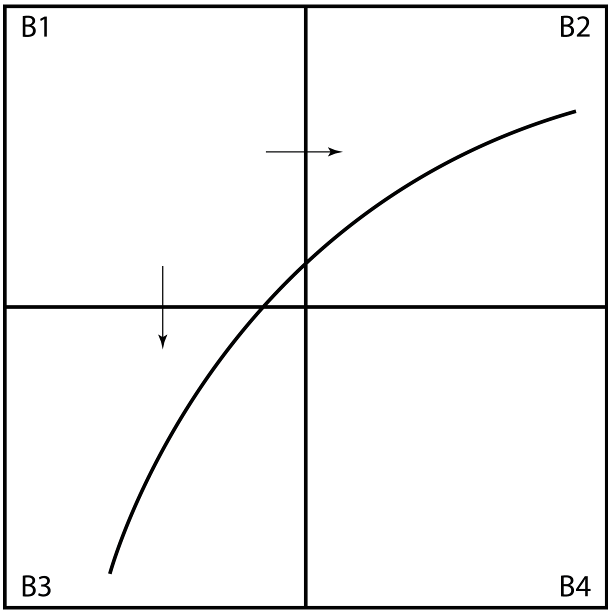


FIGURE 14.14 – NSMB MPI, Block B1 is connected on the right to B2 and on bottom to B3.

$$
\begin{aligned}
R_{1_{i+1/2}} &= \frac{1}{2}\rho_w\left(h_i^2 + h_{i+1}^2\right)\frac{\tau_{wall}}{2\mu_w}\cdot\mathbf{n}_{i+1/2} \\
&\quad - \rho_w\left|\frac{\tau_{wall}}{2\mu_w}\cdot\mathbf{n}_{i+1/2}\right| h_{i+1/2}\left(h_{i+1} - h_i\right) \\
R_{2_{i+1/2}} &= \frac{C_w}{2}\rho_w\left(h_i^2 T_i + h_{i+1}^2 T_j\right)\frac{\tau_{wall}}{2\mu_w}\cdot\mathbf{n}_{i+1/2} \\
&\quad - \rho_w\left|\frac{\tau_{wall}}{2\mu_w}\cdot\mathbf{n}_{i+1/2}\right|\left[\frac{C_w\left(h_i T_i + h_{i+1}T_{i+1}\right)}{2} + h_{i+1/2}\left(h_{i+1}T_{i+1} - h_i T_i\right)\right]
\end{aligned}
\tag{14.18}
$$

where $h_{i+1/2}$ is found by averaging the variables from both cells sharing the interface, cell i and cell $i+1$. The value of the inner product $\tau_{wall}\cdot\mathbf{n}_{i+1/2}$ is directly set to the same value found from the wall function, and the sign is imposed from the direction of the tangential velocity.

The temporal derivative is discretized using a first order Euler scheme. The time step is controlled from the stability condition given by a CFL lower than one.

The source terms S_1 and S_2 still depend on further variables, m_ice and T. So in total three variables must be calculated with only 2 equations. The compatibility relations and a system of try and error is run at each cell at every iteration as explained in section 3.5.

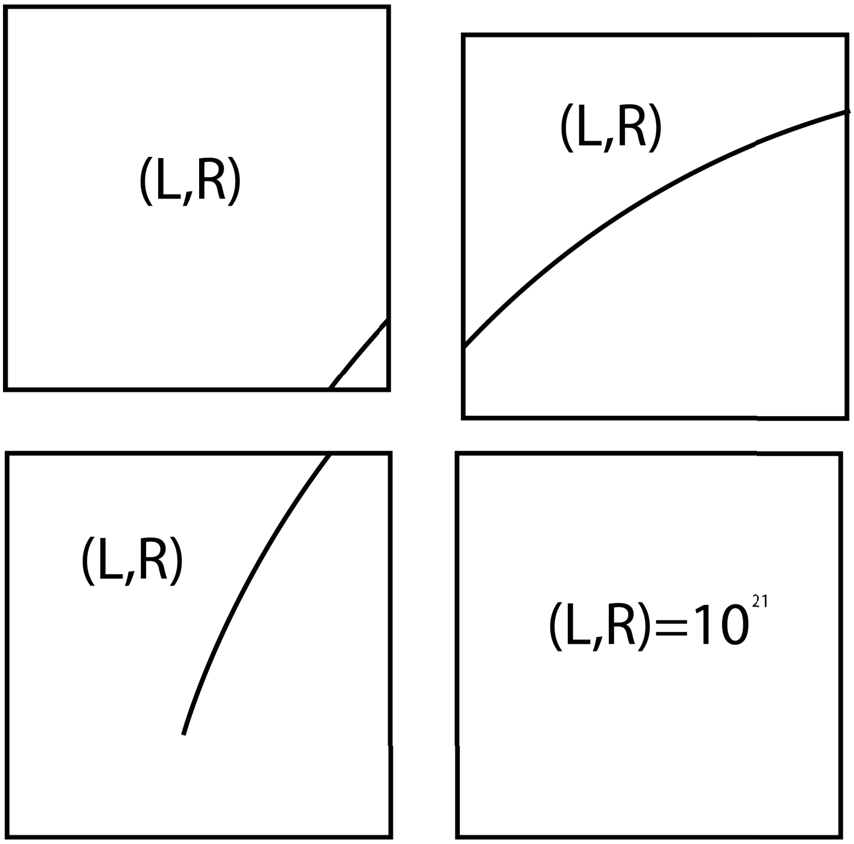


Figure 14.15 – (L,R) states communcations between adjacent blocks

14.3.5 Heat transfer coefficient

The main difficulty concurrent codes face is the determination of the heat transfer coefficient, the main crucial factor on the heat transfer and consequently the ice formation on the surface. A maximized collection efficiency would result in zero ice if the heat transfer is very low. The prominent term deciding the heat transfer, the coefficient h_{tc} is still a non-available term for the system to be solved. Two different methods exist in the literature to solve this task. Both of which are developed within the body-fitted ice module in NSMB solver. The first method is an empirical formula based on boundary layer similarity through the use of the dimensionless Stanton number. The convective heat transfer coefficient is given by :

$$h_{tc} = \rho_a C p_a U_e St \tag{14.19}$$

where ρ_a is the air density, Cp_a is the air specific heat capacity at constant pressure, U_e is the air velocity at the edge of the boundary layer, and finally St is the dimensionless Stanton number.

The Roughness Reynolds Number is defined as [23] :

$$Re_k = \frac{U_e k_s}{\nu} \tag{14.20}$$

where k_s is defined as the sand-grain roughness equivalent, and is estimated empirically by [93] :

$$k_s = 0.0008C\left(0.047T_c - 11.27\right)\left[0.5714 + 0.2457LWC + 1.2571LWC^2\right] \tag{14.21}$$

where C denotes the chord and T_c the temperature at infinity.

The Stanton number depends on the flow regime :

$$St = \begin{cases} \frac{1}{2}C_f Pr^{-\frac{2}{3}} & Re_k < 600 \\ \frac{1}{2}C_f / \left(0.9 + \sqrt{\frac{1}{2}C_f}\, St_k\right) & Re_k > 600 \end{cases} \tag{14.22}$$

where St_k is the turbulent Stanton number given by :

$$St_k = 1.92 Re_{kt}^{-0.45} Pr^{-0.8} \tag{14.23}$$

Followed by the turbulent roughness Reynolds number Re_{kt} given by Eq. 14.20 when replacing the edge velocity by the turbulent layer edge velocity, both given by :

$$\begin{cases} U_e = \sqrt{|1 - C_p|}U_\infty \\ U_t = U_e\sqrt{\frac{1}{2}C_f} \end{cases} \tag{14.24}$$

The same NACA0012 test case presented in the manuscript was tested with this 2D glaze solver using the empirical heat transfer coefficient. The poor resulting heat transfer coefficient and the ice mass rate are shown in Fig. 14.16. The empirical formulation underestimates h_{tc} at the stagnation point, dropping to zero. This results in poor rime ice shape with horns as would be expected from a higher temperature glaze formation.

Other empirical forms exist in the literature, notably ones where h_{tc} depends on the boundary layer thickness and integrals along the wing starting from the stagnation point. Such models still need some development for the required terms and would be a fortunate future work.

The second method used to calculate the convective heat transfer coefficient h_{tc} is from the CFD air solver. Two successive executions of the air flow solver are required. In the first we evaluate the recovery/adiabatic temperature. And in the second step, we impose an arbitrary wall temperature, from which we can calculate the heat transfer.

$$h_{tc} = \frac{q_{wall}}{T_{rec} - T_{wall}} \tag{14.25}$$

where the heat transfer is calculated at each SP or IB point via :

$$q = k_{cond}\frac{T_{imposed} - T_{i,j,k}}{\phi_{i,j,k}} \tag{14.26}$$

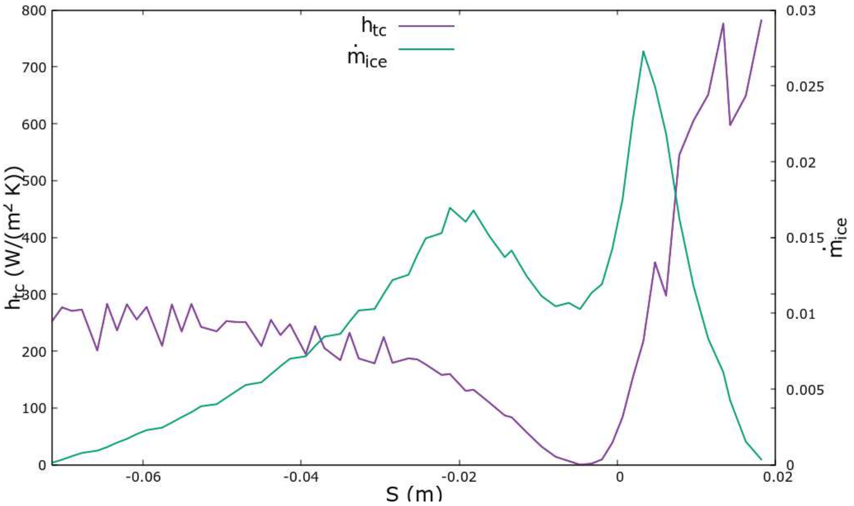


FIGURE 14.16 – The 1st test case, incorrect rime formation due to poor h_{tc} calculation

and the turbulent conductive heat transfer coefficient is given by :

$$k_{cond} = -\left(k_{air} + \mu_t \frac{C_p}{Pr_t}\right) \tag{14.27}$$

Although this method was the best method used with the body-fitted ice module with NSMB solver, it proved highly difficult to implement currently within our IBM-LS framework. More specifically, it is not the idea itself that proves difficult, but the oscillations in pressure and temperature close to the interface, which are induced by the wall model coupled in our compressible NSMB solver, Fig. 14.17.

In conclusion the 2D glaze solver developed in this section is working perfectly as can be observed from Fig. 14.16. The resulting ice formation was very satisfactory given the supplied h_{tc}. However, estimating the correct convective coefficient is still the main difficulty. It should be either tackled empirically with more sophisticated formulas, or by using CFD, by treating the wall model to take into account compressibility effects. Compressibility effects are however thought not to be the main source of errors here. There are many studies in the literature treating the pressure oscillations proposing different approaches to tackle these problems.

We conclude this report at this stage with clear future perspectives. The wall model should be treated to take into account compressibility effects. The pressure and temperature oscillations should also be treated in a more rigorous manner. The dynamic chimera block super-positioning is highly required to ensure good mesh quality for complex ice formations. The glaze model introduced should be extended into three dimensions, after the calculation of the heat transfer coefficient is improved.

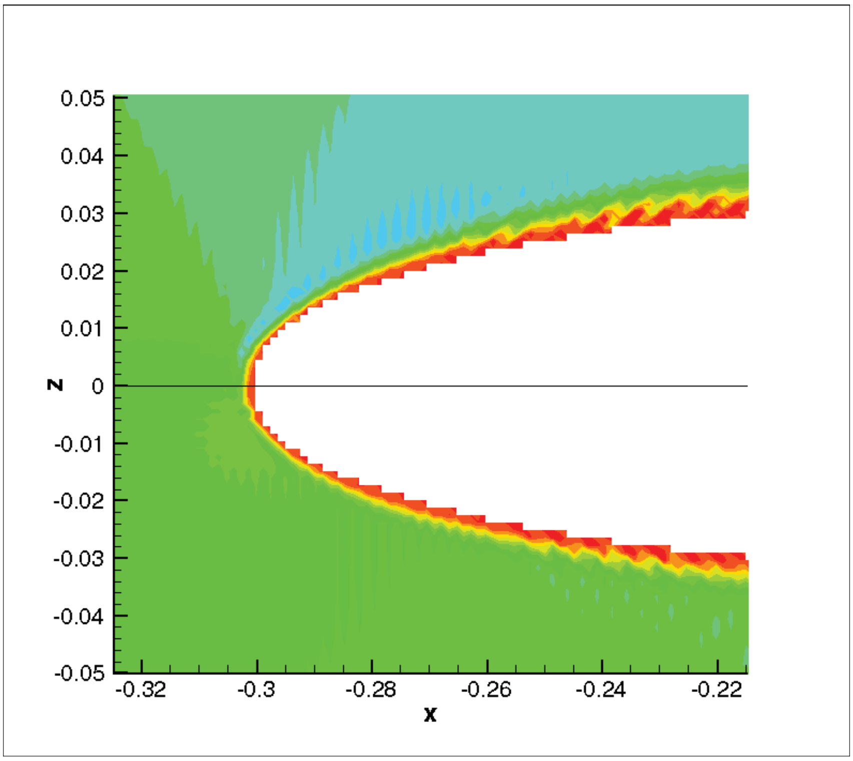


FIGURE 14.17 – Pressure oscillations due to temperature oscillations

Bibliographie

[1] R. Abgrall, H. Beaugendre et C. Dobrzynski, « An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques », *in* : *Journal of Computational Physics* 257 (2014), p. 83–101.

[2] *Accidents Rate per Year*, `http://www.baaa-acro.com/statistics/crashs-rate-per-year`, [Online ; accessed 03-January-2018], 2018.

[3] N. Aliaga et al., « A third-generation in-flight icing code : FENSAP-ICE-unsteady », *in* : *SAE Technical Papers* (2007).

[4] P. Angot, C.-H. Bruneau et P. Fabrie, « A penalization method to take into account obstacles in viscous flows », *in* : *Numerische Mathematik* 81 (fév. 1999), p. 497–520.

[5] B. Aupoix et P. R. Spalart, « Extensions of the Spalart–Allmaras turbulence model to account for wall roughness », *in* : *International Journal of Heat and Fluid Flow* 24.*4* (2003), p. 454–462.

[6] C. Bai et A. D. Gosman, « Development of methodology for spray impingement simulation. », *in* : *SAE Paper 950283* (1995), p. 69–87.

[7] C. BAYEUX, « Méthode intégrale pour la couche limite tridimensionnelle Applications au givrage », thèse de doct., UNIVERSITÉ DE TOULOUSE, 2017.

[8] H. Beaugendre, F. Morency et W. G. Habashi, « Development of a second generation in-flight icing simulation code », *in* : *Journal of fluids engineering* 128.*2* (2006), p. 378–387, url : `http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1430310` (visité le 26/03/2014).

[9] H. Beaugendre, F Morency et W.G. Habashi, « FENSAP-ICE's three-dimensional in-flight ice accretion module : ICE3D », *in* : *Journal of Aircraft* 40.*2* (2003), p. 239–247, url : `http://arc.aiaa.org/doi/pdf/10.2514/2.3113` (visité le 26/03/2014).

[10] H. Beaugendre et al., « Computation of ice shedding trajectories using cartesian grids, penalization, and level sets », *in* : *Modelling and Simulation in Engineering* 2011 (2011), p. 3.

[11] R. P. Beyer et R. J. Leveque, « Analysis of a One-Dimensional Model for the Immersed Boundary Method », *in* : *SIAM Journal on Numerical Analysis* 29.*2* (1992), p. 332–364.

[12] C. S. Bidwell et M. G. Potapczuk, « Users manual for the NASA Lewis three-dimensional ice accretion code (LEWICE 3D) », *in* : *Technical Memorandum, 105974* (1993).

[13] J. Blazek, *Computational fluid dynamics : principles and applications*, Butterworth-Heinemann, 2015.

[14] Y. Bourgault, H. Beaugendre et W. Habashi, « Development of a shallow-water icing model in FENSAP-ICE. », *in* : *Journal of Aircraft* 37.*4* (2000), p. 640–646.

[15] Y. Bourgault et al., « A finite element method study of Eulerian droplets impingement models », *in* : *International Journal for Numerical Methods in Fluids* 29.*4* (1999), p. 429–449.

[16] R. Britton, T. Bond et R Flemming, « An overview of a model rotor icing test in the NASA Lewis Icing Research Tunnel », *in* : *32nd Aerospace Sciences Meeting and Exhibit*, 1994, doi : 10.2514/6.1994-716.

[17] A. P. Broeren et al., « Effect of High-Fidelity Ice-Accretion Simulations on Full-Scale Airfoil Performance », *in* : *Journal of aircraft* 47.*1* (2010), p. 240–254.

[18] Francesco C. et Emiliano I., « A Eulerian Method for Water Droplet Impingement by Means of an Immersed Boundary Technique. », *in* : *J. Fluids Eng* (2014).

[19] Y. Cao et K. Chen, « Helicopter icing », *in* : *Aeronautical Journal* 114.*1152* (2010), p. 83–90.

[20] Y. Cao et S. Hou, « Extension to the Myers Model for Calculation of Three-Dimensional Glaze Icing », *in* : *Journal of Aircraft* 53.*1* (2016), p. 106–116.

[21] Y. Cao, J. Huang et J. Yin, « Numerical simulation of three-dimensional ice accretion on an aircraft wing », *in* : *International Journal of Heat and Mass Transfer* 92 (2016), p. 34–54.

[22] Y. Cao et al., « Numerical simulation of ice accretions on an aircraft wing », en, *in* : *Aerospace Science and Technology* 23.*1* (déc. 2012), p. 296–304, issn : 12709638, doi : 10.1016/j.ast.2011.08.004, url : http://linkinghub.elsevier.com/retrieve/pii/S127096381100126X (visité le 26/03/2014).

[23] Y. Cao et al., « Numerical simulation of ice accretions on an aircraft wing », *in* : *Aerospace Science and Technology* 23.*1* (2012), p. 296–304.

[24] F. Chedevergne et B. Aupoix, « Accounting for wall roughness effects in turbulence models : a wall function approach », *in* : *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)* (2017).

[25] ZHIGUO SUN CHENGXIANG ZHU BIN FU et CHUNLING ZHU, « 3D ICE ACCRETION SIMULATION FOR COMPLEX CONFIGURATION BASING ON IMPROVED MESSINGER MODEL », *in* : *International Journal of Modern Physics : Conference Series* 19 (2012), p. 341–350.

[26] J.-I. Choi et al., « An immersed boundary method for complex incompressible flows », *in* : *Journal of Computational Physics* 224.*2* (2007), p. 757–784, issn : 0021-9991, doi : `https://doi.org/10.1016/j.jcp.2006.10.032`, url : `http://www.sciencedirect.com/science/article/pii/S0021999106005481`.

[27] H. Daniel, M. Matthias et S. Wolfgang, « The constrained reinitialization equation for level set methods », *in* : *Journal of Computational Physics* 229.*5* (2010), p. 1514–1535, issn : 0021-9991, doi : `https://doi.org/10.1016/j.jcp.2009.10.042`, url : `http://www.sciencedirect.com/science/article/pii/S0021999109006032`.

[28] T. Deloze, « Couplage fluide-solide appliqué à l'étude de mouvement d'une sphère libre dans un tube vertical », thèse de doct., University of Strasbourg, 2012.

[29] W. Dong et al., « Calculation and Analysis of Runback Water Flow on Anti-Icing Airfoil Surface », *in* : *Journal of Aircraft* 53.*6* (2016).

[30] R Flemming, T. Bond et R. Britton, « Results of a sub-scale model rotor icing test », *in* : 1991.

[31] *Flying still the safest form of travel*, `http://news.bbc.co.uk/2/hi/uk_news/736582.stm`, [Online ; accessed 03-January-2018], 2018.

[32] M. Fossati et W. Habashi, « Multiparameter Analysis of Aero-Icing Problems Using Proper Orthogonal Decomposition and Multidimensional Interpolation », *in* : 51 (avr. 2013), p. 946–960.

[33] Kalitzin G. et Iaccarino G., « Turbulence modeling in an immersed-boundary RANS method », *in* : *CTR Annual Briefs* (2002), p. 415–426.

[34] G. H. Ganser, « A rational approach to drag prediction of spherical and nonspherical particles », *in* : *Powder Technology* 77.*2* (1993), p. 143–152.

[35] R. Gent, D. Guffond et J. Reinmann, « Dra/nasa/onera collaboration on icing research : Part i-prediction of water droplet trajectories. », *in* : *Technical report, DRA/MS/TR93085/1.* (1993).

[36] D. Goldstein, R. Handler et L. Sirovich, « Modeling a No-Slip Flow Boundary with an External Force Field », *in* : *Journal of Computational Physics* 105.*2* (1993), p. 354–366.

[37] A. Gosset, « Prediction of rivulet transition in anti-icing applications. », *in* : *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)* (2017).

[38] D. Guffond et T. Hedde, « Prediction of ice accretion : Comparison between 2D and 3D codes. », *in* : *La Recherche Aérospatiale 2* (1994), p. 103–115.

[39] Beaugendre H. et Morency F., « Penalization of the Spalart–Allmaras turbulence model without and with a wall function : Methodology for a vortex in cell scheme », *in* : *Computers and Fluids* 170 (2018), p. 313–323.

[40] Y. Han et J. Palacios, « Surface Roughness and Heat Transfer Improved Predictions for Aircraft Ice-Accretion Modeling. », *in* : *AIAA Journal* (2016).

[41] O. Harireche et al., « Explicit finite volume modeling of aircraft anti-icing and de-icing », *in* : *Journal of Aircraft* 45.*6* (2008).

[42] K. Hasanzadeh, E. Laurendeau et I. Paraschivoiu, « Quasi-Steady Convergence of Multistep Navier–Stokes Icing Simulations », *in* : *Journal of Aircraft* 50.*4* (2013), p. 1261–1274.

[43] K. Hasanzadeh et al., « Wing aerodynamic performance analysis and stall prediction using canice2d-ns icing code », *in* : *In 20th Annual Conference of the CFD Society of Canada.* (2012).

[44] J.M. Hospers, « Eulerian method for super-cooled large-droplet ice-accretion on aircraft wings. », thèse de doct., University of Twente, 2013.

[45] *How Many Planes Are in the Air at Any Moment*, https://www.travelandleisure.com/airlines-airports/number-of-planes-in-air, [Online ; accessed 03-January-2018], 2018.

[46] G. Iaccarino, G. Kalitzin et G. J. Elkins, « Numerical and experimental investigation of the turbulent flow in a ribbed serpentine passage », *in* : *Technical report, DTIC Document* (2003).

[47] Hsun H. Chen Jaiwon Shin et Tuncer Cebeci, « A Turbulence Model for Iced Airfoils and Its Validation », *in* : *NASA Technical Memorandum - AIAA* (1992).

[48] A. Jameson, « Steady state Solutions of the Euler Equations for Transonic Flow by a Multigrid Method », *in* : *Advances in Scientific Comp.* (1982), p. 37–70.

[49] O. W. Jayaratne et B. J. Mason, « The coalescence and bouncing of water drops at an air/water interface », *in* : *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences* 280.*1383* (1964), p. 545–565, doi : 10.1098/rspa.1964.0161.

[50] G.-S. Jiang et C.-W. Shu, « Efficient Implementation of Weighted ENO Schemes », *in* : *Journal of Computational Physics* 126.*1* (1996), p. 202–228, issn : 0021-9991, doi : https://doi.org/10.1006/jcph.1996.0130, url : http://www.sciencedirect.com/science/article/pii/S0021999196901308.

[51] S. K. et al. Jung, « An efficient CFD-based method for aircraft icing simulation using a reduced order model », *in* : *Journal of Mechanical Science and Technology* 25.*3* (2011), p. 703–711.

[52] S. Jung et R. Myong, « Numerical Modeling for Eulerian Droplet Impingement in Supercooled Large Droplet Conditions », *in* : *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (jan. 2013).

[53] S.K. Jung, R. S. Myong et T.-H. Cho, « Development of Eulerian droplets impingement model using HLLC Riemann solver and POD-based reduced order method », *in* : *AIAA Paper* 3907 (2011), p. 2011.

[54] W. M. Kays, M. E. Crawford et B. Weigand, *Convective heat and mass transfer*, McGraw-Hill, 2012.

[55] A. Al-Kebsi et al., « Development of a level-set based multi-step icing simulation code », *in* : 2017.

[56] J. Kim et al., « Ice Accretion Modeling Using an Eulerian Approach for Droplet Impingement », *in* : *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (jan. 2013).

[57] K. D. KORKAN, L. DADONE et R. J. SHAW, « Performance degradation of helicopter rotor in forward flight due toice », *in* : *Journal of Aircraft* 22.*8* (1985), p. 713–718, doi : 10.2514/3.45191, url : https://doi.org/10.2514/3.45191.

[58] M.-C. Lai et Peskin C., « An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity », *in* : *Journal of Computational Physics* 160.*2* (2000), p. 705–719.

[59] T. Mao, D. C. S. Kuhn et H. Tran, « Spread and rebound of liquid droplets upon impact on flat surfaces », *in* : *AIChE Journal* 43.*9* (1997), p. 2169–2179.

[60] S. Mark, S. Peter et O. Stanley, « A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow », *in* : *Journal of Computational Physics* 114.*1* (1994), p. 146–159.

[61] S. T. McClain et al., « Ice Accretion Roughness Measurements and Modeling », *in* : *7TH EUROPEAN CONFERENCE FOR AERONAUTICS AND AEROSPACE SCIENCES (EUCASS)* (2017).

[62] C. Merlin, « Simulation numérique de la combustion turbulente : Méthode de frontères immergées pour les écoulements compressibles, application à la combustion en aval d'une cavité », thèse de doct., INSA de Rouen, 2011.

[63] B. L. Messinger, « Equilibrium temperature of an unheated icing surface as a function of air speed », *in* : *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)* 20.*1* (1953).

[64] K. Mikula et M. Ohlberger, « A New Level Set Method for Motion in Normal Direction Based on a Semi-Implicit Forward-Backward Diffusion Approach », *in* : *SIAM Journal on Scientific Computing* 32.*3* (2010), p. 1527–1544, doi : `10.1137/09075946X`.

[65] K. Mikula, M. Ohlberger et J. Urbán, « Inflow-implicit/outflow-explicit finite volume methods for solving advection equations », *in* : *Applied Numerical Mathematics* 85 (2014), p. 16–37, issn : 0168-9274, doi : `https://doi.org/10.1016/j.apnum.2014.06.002`, url : `http://www.sciencedirect.com/science/article/pii/S0168927414001032`.

[66] G. Mingione et V. Brandi, « Ice Accretion Prediction on Multielement Airfoils », *in* : *Journal of Aircraft* 35 (1998), p. 240.

[67] G. Mingione, V. Brandi et Esposito B., « Ice Accretion Prediction on Multi-Element Airfoils. », *in* : *AIAA paper* (1997).

[68] R. Mittal et G. Iaccarino, « IMMERSED BOUNDARY METHODS », *in* : *Annual Review of Fluid Mechanics* 37.*1* (2005), p. 239–261.

[69] E. Montreuil et al., « ECLIPPS : 1. Three-Dimensional CFD Prediction of the Ice Accretion », *in* : *1st AIAA Atmospheric and Space Environments Conference* (2009).

[70] W. Mulder, S. Osher et J. A. Sethian, « Computing interface motion in compressible gas dynamics », *in* : 1992.

[71] C. Mundo, M. Sommerfeld et C. Tropea, « On the modeling of liquid sprays impinging on surfaces. », *in* : *Atomization and Sprays* 8 (1998), p. 625–652.

[72] T. G. Myers, « Extension to the Messinger model for aircraft icing », *in* : *AIAA journal* 39.*2* (2001), p. 211–218, url : `http://arc.aiaa.org/doi/pdf/10.2514/2.1312` (visité le 25/06/2014).

[73] Ellen N., *Eulerian method for ice crystal icing in turbofan engines*, 2017.

[74] J. Naber et R.D. Reitz, « Modeling Engine Spray/Wall Impingement », *in* : *SAE Paper 880107* (1988).

[75] R.R. Nourgaliev et T.G. Theofanous, « High-fidelity interface tracking in compressible flows : Unlimited anchored adaptive level set », *in* : *Journal of Computational Physics* 224.*2* (2007), p. 836–866, issn : 0021-9991, doi : https://doi.org/10.1016/j.jcp.2006.10.031, url : http://www.sciencedirect.com/science/article/pii/S0021999106005511.

[76] *Number of flights performed by the global airline industry*, https://www.statista.com/statistics/564769/airline-industry-number-of-flights/, [Online ; accessed 03-January-2018], 2018.

[77] Stanley O. et James A. S., « Fronts propagating with curvature-dependent speed : Algorithms based on Hamilton-Jacobi formulations », *in* : *Journal of Computational Physics* 79.*1* (1988), p. 12–49, issn : 0021-9991, doi : https://doi.org/10.1016/0021-9991(88)90002-2, url : http://www.sciencedirect.com/science/article/pii/0021999188900022.

[78] S. Osher et R. Fedkiw, *Level set methods and dynamic implicit surfaces*, t. 153, Springer Science & Business Media, 2006.

[79] S. Ozgen et M. Canibek, « Ice accretion simulation on multi-element aifoils using extended MessingerModel. », *in* : *Heat and Mass Transfer* 45 (2009), p. 305–322.

[80] M. Papadakis et al., « Large and small droplet impingement data on airfoils and two simulated ice shapes », *in* : (2007), url : http://ntrs.nasa.gov/search.jsp?R=20070034950 (visité le 26/03/2014).

[81] D. Pena, E. Laurendeau et Y. Hoarau, « A single step ice accretion model using Level-Set method », *in* : *Journal of Fluids and Structures* 65 (2016), p. 278–294.

[82] D. Peng et al., « A PDE-Based Fast Local Level Set Method », *in* : *Journal of Computational Physics* 155 (1999), p. 410–438.

[83] R. Peyret et T. D. Taylor, *Computational methods for fluid flow*, Springer Science & Business Media, 2012.

[84] *Planes in Real Time*, https://www.flightradar24.com/48.81,6.86/2, [Online ; accessed 03-January-2018], 2018.

[85] X. PRESTEAU, « MODELISATION MICROPHYSIQUE TRIDIMENSIONNELLE DES DEPOTS DE GIVRE : APPLICATION AU GIVRE EN QUEUE DE HOMARD SUR DES CYLINDRES EN FLECHE. », thèse de doct., UNIVERSITE BLAISE PASCAL, 2009.

[86] A.l. Reehorst, « National Transportation Safety Board Aircraft Accident Investigation Supported. », *in* : (2005).

[87] F. Saeed, S. Gouttebroze et I. Paraschivoiu, « Modified canice for improved prediction of airfoil ice accretion », *in* : *In 8th Aerodynamic Symposium of 48thCASI Conference, Toronto, Canada, Apr.* (2001).

[88] W. Samenfink et al., « Droplet interaction with shear-driven liquid films : analysis of deposi- tion and secondary droplet characteristics. », *in* : *International Journal of Heat and Fluid Flow* 20 (1999), p. 462–469.

[89] W. Sang, Y. Shi et C. Xi, « Numerical simulation of icing effect and ice accretion on three-dimensional configurations », en, *in* : *Science China Technological Sciences* 56.*9* (sept. 2013), p. 2278–2288, issn : 1674-7321, 1869-1900, doi : `10.1007/s11431-013-5277-3`, url : `http://link.springer.com/10.1007/s11431-013-5277-3` (visité le 25/06/2014).

[90] L. Schiller et Z. Naumann, « A drag coefficient correlation », *in* : *Vdi Zeitung* 77.*318* (1935), p. 51.

[91] R. Schmehl et al., « Cfd analysis ofspray propagation and evaporation including wall film formation and spray/film. », *in* : *International Journal of Heat and Fluid Flow* 20 (1999), p. 520–529.

[92] J.N. Scott et al., « Navier-stokes solution to the flowfield over ice accretion shapes », *in* : *Journal of Aircraft* 25.*8* (1988), p. 710–716.

[93] J. Shin et T. H. Bond, *Experimental and computational ice shapes and resulting drag increase for a NACA 0012 airfoil*, National Aeronautics et Space Administration, 1992.

[94] C. Son, S. Oh et K. Yee, « Quantitative analysis of a two-dimensional ice accretion on airfoils », *in* : *Journal of Mechanical Science and Technology* 26.*4* (avr. 2012), p. 1059–1071.

[95] P. R. Spalart et S. R. Allmaras, « A One-Equation Turbulence Model for Aerodynamic Flows. », *in* : *AIAA journal* 94 (1992).

[96] M. Specklin et Y. Delauré, « A sharp immersed boundary method based on penalization and its application to moving boundaries and turbulent rotating flows », *in* : *European Journal of Mechanics - B/Fluids* 70 (2018), p. 130–147, issn : 0997-7546, doi : `https://doi.org/10.1016/j.euromechflu.2018.03.003`, url : `http://www.sciencedirect.com/science/article/pii/S0997754617302364`.

[97] D. W. Stanton et C. Rutland, « Modeling fuel film formation and wall interaction in diesel engines. », *in* : *SAE Paper 960628* (1996).

[98] M. Sussman et E. Fatemi, « An Efficient, Interface-Preserving Level Set Redistancing Algorithm and Its Application to Interfacial Incompressible Fluid Flow », *in* : *SIAM Journal on Scientific Computing* 20.*4* (1999), p. 1165–1191, doi : 10.1137/S1064827596298245.

[99] T. W. Swafford, « Analytical approximation of two-dimensional separated turbulent boundary-layer velocity profiles », *in* : *AIAA Journal* 21.*6* (1983), p. 923–926.

[100] S. Tanguy, « Développement d'une méthode de suivi d'interface. Applications aux écoulements diphasiques », thèse de doct., Université de Rouen, 2004.

[101] M. F. Trujillo et al., « Impingement and Atomization of a Liquid Spray on a Wall. », *in* : *International Journal of Engine Research* 1.*1* (2000), p. 87–104.

[102] P. Verdin, J. P. F. Charpin et Thompson C. P., « Multistep Results in ICECREMO2 », *in* : *Journal of Aircraft* 46.*5* (2009), p. 1607–1613.

[103] K. Voronetska, « Simulation numérique directe des écoulements à phases dispersées », thèse de doct., Bordeaux 1, 2012.

[104] et al Vos, « NSMB Handbook », *in* : *Technical report, Technical Report 4.5* (2000).

[105] William W., *Users manual for the improved NASA Lewis ice accretion code LEWICE 1.6*, 1995.

[106] D.M. Wang et A.P. Watkins, « Numerical modelling of diesel wall spray phenomena », *in* : *International Journal of Heat and Fluid Flow* 14 (1993), p. 301–312.

[107] S. Wang et E. Loth, « Droplet Impact Efficiency on Aerodynamic Surfaces with a Globally Eulerian/Locally Lagrangian Method », *in* : *Journal of Aircraft* 54.*1* (2017), p. 104–113.

[108] W. B. Wright, « Users Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.0. », *in* : *NASA CR-209409* (1999).

[109] W.B. Wright, « LEWICE 2.2 Capabilities and thermal validation. », *in* : *AIAA paper 0383* (2002).

[110] W.B. Wright, R. Gent et D. Guffond, « DRA/NASA/ONERA Collaboration on Icing Research : Part II–Prediction of Airfoil Ice Accretion », *in* : 1997.

[111] Z. Zhan, W. Habashi et M. Fossati, « Local Reduced-Order Modeling and Iterative Sampling for Parametric Analyses of Aero-Icing Problems », *in* : 53 (avr. 2015), p. 1–12.

[112] Z. Zhan, W. Habashi et M. Fossati, « Real-Time Regional Jet Comprehensive Aeroicing Analysis via Reduced-Order Modeling », *in* : 54 (juil. 2016), p. 1–16.

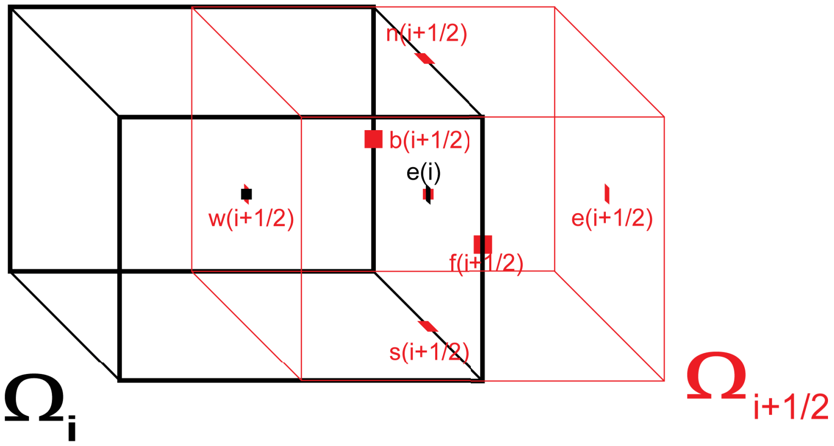
[113] C. Zhu et al., « 3D ICE ACCRETION SIMULATION FOR COMPLEX CONFIGURATION BASING ON IMPROVED MESSINGER MODEL », en, *in* : *International Journal of Modern Physics : Conference Series* 19 (jan. 2012), p. 341–350, issn : 2010-1945, 2010-1945, doi : `10.1142/S2010194512008938`, url : `http://www.worldscientific.com/doi/abs/10.1142/S2010194512008938` (visité le 26/03/2014).

[114] S. Zlatko et al., « Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows : A comparative study », *in* : *International Journal of Multiphase Flow* 95 (2017), p. 235–256, issn : 0301-9322, doi : `https://doi.org/10.1016/j.ijmultiphaseflow.2017.06.004`, url : `http://www.sciencedirect.com/science/article/pii/S0301932216304669`.

[115] M. Zocca, G. Gori et A. Guardone, « Wind tunnel effects on ice accretion over aircraft wings », *in* : (2015).

Shifted Cell : Evaluation of Gradient at the Faces

In this part we introduce the shifted cell method used to evaluate the gradient of any variable at the cell faces. Consider the cell (i,j,k) shown in the figure below, the gradient is to be evaluated at three of its sides, east $(i+1/2)$, north $(j+1/2)$, and front $(k+1/2)$. We develop the equations for the east side, where the other two follow in the same manner.



Shifted cell used to calculate the gradient at the edges.

For any variable, say ϕ, using finite volume averaging we obtain :

$$\nabla\phi_{i+1/2} = \frac{1}{\Omega_{i+1/2}} \iiint \nabla\phi_{i+1/2} d\Omega \tag{28}$$

The integral, denoted here (I) in the last equation, can be exchanged by the closed surface integral, divergence theorem :

$$I = \iint \phi_F \cdot \mathbf{dS} = \sum_{F=ewnsfb} \phi_F \cdot \mathbf{S}_F \tag{29}$$

The values of the variable ϕ_F at the faces of the shifted volume are interpolated from values at the cell center as follows :

$$\begin{aligned}
\phi_e(i+1/2) &= \phi_{(i+1,j,k)} \\
\phi_w(i+1/2) &= \phi_{(i,j,k)} \\
\phi_n(i+1/2) &= \tfrac{1}{4}\left(\phi_{(i,j,k)} + \phi_{(i,j+1,k)} + \phi_{(i+1,j,k)} + \phi_{(i+1,j+1,k)}\right) \\
\phi_s(i+1/2) &= \tfrac{1}{4}\left(\phi_{(i,j,k)} + \phi_{(i,j-1,k)} + \phi_{(i+1,j,k)} + \phi_{(i+1,j-1,k)}\right) \\
\phi_f(i+1/2) &= \tfrac{1}{4}\left(\phi_{(i,j,k)} + \phi_{(i,j,k+1)} + \phi_{(i+1,j,k)} + \phi_{(i+1,j,k+1)}\right) \\
\phi_b(i+1/2) &= \tfrac{1}{4}\left(\phi_{(i,j,k)} + \phi_{(i,j,k-1)} + \phi_{(i+1,j,k)} + \phi_{(i+1,j,k-1)}\right)
\end{aligned} \tag{30}$$

The surfaces in NSMB are defined at the cell side as vectors, as is usually the case in most fluid solvers. The surfaces required $\mathbf{S}_F$ to be evaluated for the shifted cell are given as follows :

$$\begin{aligned}
S_e &= \tfrac{1}{2}\left(S_{(i+1/2,j,k)} + S_{(i+1+1/2,j,k)}\right) \\
S_w &= \tfrac{1}{2}\left(S_{(i-1/2,j,k)} + S_{(i+1/2,j,k)}\right) \\
S_n &= \tfrac{1}{2}\left(S_{(i,j+1/2,k)} + S_{(i+1,j+1/2,k)}\right) \\
S_s &= \tfrac{1}{2}\left(S_{(i,j-1/2,k)} + S_{(i+1,j-1/2,k)}\right) \\
S_f &= \tfrac{1}{2}\left(S_{(i,j,k+1/2)} + S_{(i+1,j,k+1/2)}\right) \\
S_b &= \tfrac{1}{2}\left(S_{(i,j,k-1/2)} + S_{(i+1,j,k-1/2)}\right)
\end{aligned} \tag{31}$$

where $S_{(i+1/2,j,k)}$ is the surface vector east of cell (i,j,k), and likewise for other surfaces. Finally the volume of the shifted cell is averaged as follows :

$$\Omega_{i+1/2} = \frac{1}{2}\left(\Omega_i + \Omega_{i+1}\right) \tag{32}$$