Titre: Title:	Immersed Boundary Methods for the Modeling of In-Flight Ice Accretion
Auteur: Author:	Pierre Lavoie
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lavoie, P. (2021). Immersed Boundary Methods for the Modeling of In-Flight Ice Accretion [Ph.D. thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6577/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal
ET
Institut Supérieur de l'Aéronautique et de l'Espace
Université de Toulouse

Immersed Boundary Methods for the Modeling of In-Flight Ice Accretion

PIERRE LAVOIE
Département de génie mécanique

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie mécanique

Mai 2021

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

ET

Institut Supérieur de l'Aéronautique et de l'Espace

Université de Toulouse

Cette thèse intitulée :

Immersed Boundary Methods for the Modeling of In-Flight Ice Accretion

présentée par **Pierre LAVOIE**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*

a été dûment acceptée par le jury d'examen constitué de :

Jean-Yves TRÉPANIER, président

Éric LAURENDEAU, membre et directeur de recherche

Philippe VILLEDIEU, membre et codirecteur de recherche

Yannick HOARAU, membre

Alberto GUARDONE, membre externe

DEDICATION

To Anne-Marie,
for her relentless support

ACKNOWLEDGEMENTS

This thesis was conducted as a joint project between Polytechnique Montréal and the Office National d'Études et de Recherches Aérospatiales (ONERA). It was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Alexander Graham Bell Canada Graduate Scholarships (CGS D). This allowed me to entirely focus on my research, whether in Montréal or Toulouse.

I am extremely grateful to my research directors, Éric Laurendeau and Philippe Villedieu who gave me this great joint project opportunity which allowed me to grow as a scientist and as a person. They provided me with guidance and their invaluable expertise. I am also deeply indebted to my supervisors, Emmanuel Radenac and Ghislain Blanchard who provided much needed technical advice and thoroughly reviewed my work despite the short timeframes. I would also like to thank Pierre Trontin for his support with IGLOO2D and icing in general.

Special thanks to Julien and Luana for kindly providing me with a place to stay during my first few weeks in Toulouse, and also to Anthony for bearing with me during all this time sharing an office at ONERA and then an apartment. With all this, I definitely brought back some French expressions home and some knowledge about football too.

I very much appreciated my time at ONERA (Toulouse) thanks to the warm welcome, the great people, the coffee breaks, the lunch breaks and other social events. Thanks to Matthias, Thibault X., Lola, Virgile, Pierre D., Baptiste, Jeff, Anthony, Tanguy, Nicolas P., Marc, Vincent, Thibault D., Julien L. and Swann... I do not play *coinche* and *tarot* much these days.

After all these years at Polytechnique Montréal, I had great pleasure working with Simon, Frédéric, Hélène, Alexandros, Aviral and the rest of the team. It is always fun to have a chat with you (technical or not). Special thanks to Simon for all those technical discussions about icing and to Frédéric for letting me know when I was "going too far". I would also like to thank my friends from the *other lab*: Tanja, Pablo, Jean and Cristian. Because of you all, I had a great time at Polytechnique.

Finally, my warmest thanks to my family for believing in me and constantly pushing me forward, and to Anne-Marie for always being there to support and motivate me, especially during the last year confined at home due to the COVID-19 pandemic.

RÉSUMÉ

Le givrage en vol est un danger pour la sécurité d'un aéronef, car il peut affecter son aérodynamique, ses sondes et ses capteurs. La modélisation de ce phénomène se fait généralement par un appel séquentiel à des modules traitant la génération de maillage, l'aérodynamique, la trajectoire des gouttelettes, le transfert de chaleur à la paroi, la thermodynamique et la mise à jour de la géométrie. Il est important d'automatiser ce processus, car les modules sont contenus dans une boucle temporelle (multi pas) pouvant être répétée plusieurs fois pour une prévision adéquate du givre. La robustesse des outils numériques de givrage est souvent limitée par la difficulté à générer un maillage sur des formes de givre complexes et aussi par la mise à jour de la géométrie qui peut mener à des croisements de surfaces. L'objectif principal de cette thèse est d'évaluer le potentiel des méthodes de frontières immergées (IBMs) de résoudre ces problèmes en éliminant l'intervention de l'utilisateur tout en conservant la même précision que les approches basées sur des maillages conformes (BFM).

Les développements sont effectués dans le logiciel IGLOO2D. L'écoulement d'air y est modélisé par les équations d'Euler couplé à un modèle de couche limite. La méthodologie consiste à démarrer les simulations sur un maillage conforme (BF) et d'appliquer l'IBM seulement sur le givre. Un raffinement du maillage initial autour de la zone d'accrétion permet d'éviter complètement le remaillage. Parce que le givre est immergé dans le maillage, les modules volumiques (aérodynamique et gouttelettes) sont modifiés pour appliquer les conditions aux limites sur la frontière immergée (IB). Les données surfaciques sont extraites sur l'IB puisqu'elles sont requises par les modules surfaciques (thermodynamique et couche limite) qui eux, restent inchangés. De plus, une méthode *level-set* remplace l'approche de déplacement de nœuds habituellement employée pour la mise à jour de la géométrie.

Tout d'abord, une nouvelle méthode de pénalisation (une IBM) est proposée pour les équations d'Euler en se basant sur la méthode *Characteristic-Based Volume Penalization* (CBVP). L'approche provenant de la littérature pénalise les équations d'Euler en appliquant une vitesse de glissement et une paroi adiabatique tout en considérant la courbure de la paroi. Une nouvelle approche (CBVP-Hs) est proposée afin d'imposer la conservation de l'entropie et de l'enthalpie totale dans la direction normale à l'IB, en remplacement de la condition adiabatique. Les résultats démontrent que la nouvelle méthode est plus précise sur des maillages plus grossiers et se comporte mieux sur des géométries à forte courbure.

Deuxièmement, une méthode de pénalisation est développée pour la trajectoire des gouttelettes afin de combler un manque dans la littérature. Pour ce système d'équations, la condition

de paroi doit être traitée de façon à éviter une réinjection des gouttelettes dans le fluide à partir d'une paroi solide. Cette situation est réglée par l'ajout d'un masque pour les gouttelettes en plus du masque solide usuel, permettant une détection automatique des conditions aux limites dans la zone d'impact et dans la zone d'ombre des gouttelettes. Les résultats démontrent la capacité de la nouvelle méthode à reproduire le comportement d'une approche BF.

Troisièmement, les deux méthodes de pénalisation et la méthode *level-set* sont intégrées dans IGLOO2D. Des simulations de givrage multi pas sont effectuées en 2D sur des profils d'aile. Les résultats correspondent généralement à ceux obtenus par une approche BF, même si un maillage plus raffiné est parfois nécessaire pour une bonne prévision de la forme du dépôt.

Cette thèse offre une alternative intéressante à une approche BF classique tout en permettant une extension facile à des simulations 3D, une application pour laquelle les IBMs et la méthode *level-set* présentent encore plus d'avantages.

ABSTRACT

In-flight ice accretion poses a serious threat to an aircraft safety by affecting its aerodynamics, probes and sensors. The numerical modelling of this phenomenon generally involves a sequential call to different modules including mesh generation, aerodynamics, droplet trajectories, wall heat transfer, ice accretion and geometry update. The automation of this process is critical as the solvers are embedded in a time loop which is repeated several times (multi-step) to obtain an accurate ice shape prediction. The robustness of ice accretion tools is often limited by the difficulty of generating meshes on complex ice shapes and also by the geometry update which can exhibit overlaps if not treated properly. The objective of this thesis is to investigate the potential of Immersed Boundary Methods (IBMs) to solve these issues by eliminating the user intervention in the mesh update while maintaining the accuracy obtained from a Body-Fitted (BF) approach.

The developments are done in the ice accretion suite IGLOO2D, using the Euler equations to model the airflow and a boundary layer model to retrieve the wall heat transfer. The proposed methodology is to use the usual BF mesh to start the simulation and apply the IBM to deal with the ice shape only. Re-meshing is avoided entirely by properly refining the initial mesh where ice accretion is expected. As the ice shape can freely cut through the mesh, the volume solvers (aerodynamics and droplet trajectories) are modified to enforce the boundary conditions on the Immersed Boundary (IB). Surface data extraction at the IB is also performed as required by the surface solvers (boundary layer and ice accretion), which are left unchanged. In addition, the level-set method is implemented as a replacement to the Lagrangian node displacement method in order to solve the issues related to the geometry update.

First, an IBM is developed for the Euler equations. The volume penalization method (an IBM) is commonly used for viscous flows but only one application to inviscid compressible flows can be found, which uses the CBVP method. This approach penalizes the Euler equations to enforce a no-penetration velocity and an adiabatic wall while accounting for wall curvature. A new approach based on the CBVP is proposed to impose the conservation of entropy and total enthalpy in the normal direction to the wall instead of the classical adiabatic condition. Numerical tests show that the new CBVP-Hs method is more accurate than the CBVP method on coarser meshes and better at retrieving attached flows for curved geometries.

Second, a new penalization method is developed for the Eulerian droplet equations as no

application to this system of equations is available in the literature. The wall boundary condition must be treated with care to avoid droplets re-injection from a solid boundary into the fluid. This is solved by the introduction of a droplet mask function in addition to the usual solid mask, providing an automatic detection of the wall boundary conditions in the impingement and shadow zones. The results show that a BF solution can be reproduced using this new penalization method.

Third, the previously developed penalization methods are integrated in the ice accretion suite along with the level-set method. Multi-step ice shape predictions are performed on 2D rime and glaze ice cases. The results are generally in good agreement with the BF approach but the IBM sometime requires a finer mesh to obtain a good ice shape prediction, especially in the presence of detached flows.

The proposed methodology is an interesting alternative to the classical body-fitted approach and should be easy to extend for 3D ice accretion, where the use of an IBM and level-set method is expected to show greater benefits.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

ACLSVOF	Adaptive CLSVOF
BC	Boundary Condition
BF	Body-Fitted
BL	Boundary Layer
BP	Boundary Point
CBVP	Characteristic-Based Volume Penalization
CBVP-Hs	Characteristic-Based Volume Penalization-Hs
CFD	Computational Fluid Dynamics
CLSVOF	Coupled Level Set Volume-of-Fluid
CURVIBM	Curvilinear Immersed Boundary Method
FDM	Finite Difference Method
FEM	Finite Element Method
FSI	Fluid Structure Interaction
FVM	Finite Volume Method
GC	Ghost Cell
HTC	Heat Transfer Coefficient
IB	Immersed Boundary
IBM	Immersed Boundary Method
IP	Image Point
LES	Large Eddy Simulation
LSQ	Least Square
LWC	Liquid Water Content

MAC	Marker and Cell
NS	Navier-Stokes
PDE	Partial Differential Equation
PLIC	Piecewise Linear Interface Calculation
RANS	Reynolds Averaged Navier-Stokes
Re	Reynolds number
RHS	Right Hand Side
SLIC	Simple Linear Interface Calculation
SWIM	Shallow Water Icing Model
THINC	Tangent of Hyperbola Interface Capturing
VoF	Volume of Fluid
WLSQ	Weighted Least Square

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Context

In-flight ice accretion typically occurs when aircraft encounter a cloud of supercooled water droplets [1]. Depending on the type and severity of the ice buildup, it can affect the aerodynamic performance via an increase of drag and a reduction of lift [2]. It may also affect the aircraft controllability, and the readings from the probes and sensors. All of these effects can, in the worst-case scenario, lead to an aircraft crash (e.g., American Eagle Flight 4184, 1994). For these reasons, anti-icing/de-icing systems are installed on critical aircraft components.

As ice accretion is a serious threat to aircraft safety, it must be shown that the aircraft can be safely operated in icing conditions as part of the aircraft certification process [3]. To do so, the most critical icing conditions must be determined for each flight phase, i.e., the ice shape producing the worst aerodynamic performance degradation. Numerical tools for the prediction of ice accretion are important in determining these worst-case ice shapes as they allow many conditions to be tested at a relatively low cost compared to wind tunnel or flight testing. Numerical tools are also a safer mean than flight testing. Then, only the most critical icing conditions (or ice shapes) are tested in wind tunnels and in flight.

Two-dimensional icing tools have been used for many years in the certification process although they are less accurate for the simulation of complex ice shapes which typically induce 3D aerodynamic flows. Current three-dimensional icing tools have not been widely used within industrial processes due to their lack of robustness. Their use is often limited to single-layer simulations whereas multi-layer simulations are commonly used in 2D tools and are generally required for accurate ice shape predictions. One of the key difficulty is the mesh update which often requires user intervention to obtain grids of reasonable quality. To correctly predict the ice shape, the mesh must be updated several times during the ice accretion simulation, thus efficiency (e.g. minimal HPC resources) and robustness (e.g. minimal computational breakdown) are of special concern. These requirements are leading towards investments in the development of novel algorithms in 2D and 3D icing suites within the research community.

1.2 Problem Statement

In terms of numerical simulation of ice accretion, it is important to note two different fields: the study of the effects of icing (aircraft performance) and the prediction of ice shapes (ice

accretion). The former focuses on the evaluation of performance degradation induced by the presence of the ice shape (e.g., variations in lift, drag and stall characteristics). The latter is concerned with the prediction of the ice accumulation (location, thickness, shape, etc.). Other field of study can also be listed such as the prediction of water impingement for the analysis of ice protection systems (no ice) or the tracking of ice particles due to ice breakup and shedding (e.g., for the prediction of engine ice ingestion). In this thesis, however, the emphasis is on the ice shape prediction.

1.2.1 Simulation of Ice Accretion

Ice accretion is a multi-physics phenomenon and its modelling is usually divided in multiple blocks. First, knowledge of the aerodynamic field is required as the airflow dictates the droplets trajectory, the heat transfer by convection and the motion of the water film (if any) on the geometry's surface. Second, information about the water impingement rate is required as it feeds the ice accretion. Third, a thermodynamic balance determines whether the water impinging the surface freezes or not. The balance takes into account the movement of unfrozen water along the surface, known as runback water. Thus, for a specific exposure time to icing conditions, the ice thickness is recovered.

For the numerical prediction of ice shapes, the ice accretion process is typically modelled by a sequential call to different modules, implicitly using a segregated approach through a quasi-steady hypothesis. This segregation technique is commonly accepted in the icing community and is used in many ice accretion tools such as LEWICE [4], CANICE [5], FENSAP-ICE [6], PoliMIce [7], NSMB-ICE [8], etc. The main modules can be listed as follows (Figure 1.1).

1. The aerodynamics solver (Reynolds Averaged Navier-Stokes (RANS), Euler, potential) provides the velocity, pressure and density fields.
2. The heat transfer and shear stress solver provides information from the boundary layer such as the convective Heat Transfer Coefficient (HTC) which is critical to obtain a good ice shape prediction. For an inviscid airflow (e.g. Euler, potential), a dedicated module is required to obtain the information (e.g. integral boundary layer code). For a viscous airflow (e.g. RANS), the required information can be directly retrieved from the aerodynamics solution.
3. The droplet solver determines the amount of water impacting the surface which will potentially freeze.
4. The thermodynamics solver (ice accretion) performs a heat and mass balance on the surface and provides the ice thickness or the ice accretion rate.

5. The geometry evolution solver generates the new iced geometry which can be used for subsequent ice accretion step if a multi-step process is employed.

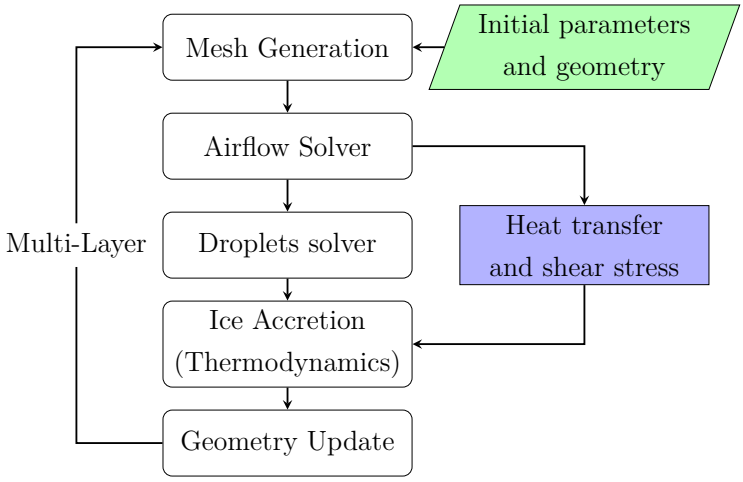
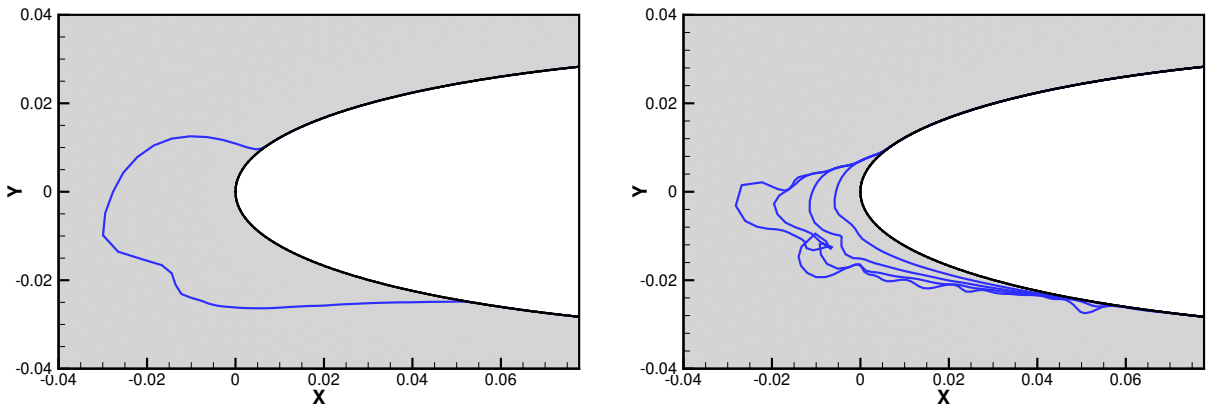


Figure 1.1 Typical multi-layer icing simulation

In a multi-step (multi-layer) approach, the ice accretion time is divided into steps (layers). For each step, the modules 1 to 5 are called sequentially. Since for each step a new geometry is generated, it implies that the mesh must be updated after each layer by, for instance, performing a full mesh regeneration.

The overall robustness of an ice accretion suite is limited by its least robust module. The critical phase is usually the interaction between the mesh generator and the geometry update.

The geometry evolution solver is robust as it never fails at generating the new iced geometry. However, the newly generated geometry can sometimes exhibit overlaps or geometry crossing as shown on the 4^{th} ice layer of Figure 1.2b. In this case, the ice accretion simulation was set for 5 layers but failed at the 4^{th} layer due to the overlap. This situation generally occurs when concave zones are formed on the geometry. The mesh generator then fails to update the mesh on this unphysical and problematic geometry. For 2D simulations, methods are available to fix these kinds of issues, involving the detection of edge crossing, node deletion and node merging. For instance, this type of correction is used in LEWICE as described in an old version of the users manual [9]. Although a fix for edge crossing can be implemented quite easily, it is not straightforward to deal with all the possible pathological cases. In 3D, the fix becomes more complex as it must be applied on 2D surface meshes and thus, an alternative approach would be preferred.



(a) Single layer (b) Multi-layer (4/5 layers)

Figure 1.2 Example of ice accretion for a rime ice case

As several icing layers (steps) are generally required to achieve accurate ice shape predictions [10], the mesh update must also be efficient and robust. The mesh update is, however, not trivial as it should be done automatically without the intervention of the user and this, on complex iced geometries. Clearly, the computational effort and complexity are increasing in 3D and mesh quality is not easily ensured.

1.2.2 Geometry Evolution

The usual method to generate the new geometry is a simple node displacement. Knowing the ice accretion rate ($\dot{m}_{ice}$) or the ice thickness (h_{ice}), the nodes describing the wall are moved using a Lagrangian approach:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + h_{ice}\mathbf{n} \tag{1.1}$$

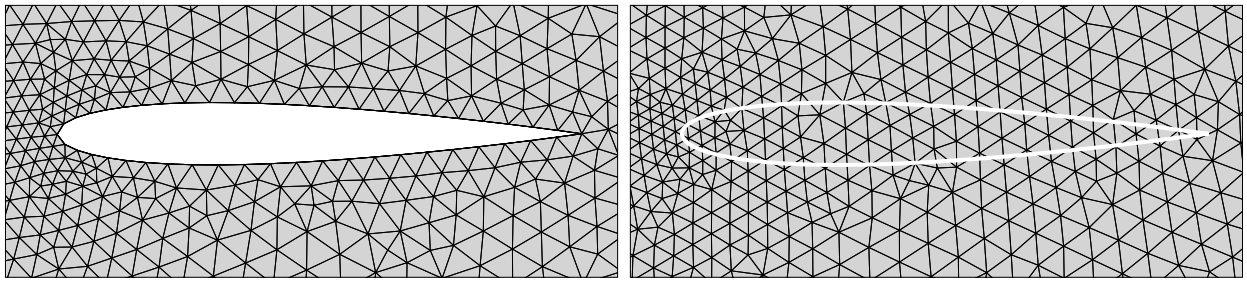
where $\mathbf{x} = (x, y)$ is the location of a node and $\mathbf{n}$ is the unit normal to the wall for this node. Without special consideration of the pathological cases (e.g., geometry overlap), this method might lead to a failure of the mesh generation, as discussed earlier.

Recently, a Level-Set method was used for the evolution of the ice shape [8]. Its main advantage compared to a more classical approach is its ability to handle the overlapping of the geometry automatically. With this method, there is no need to fix the pathological cases and it can be directly applied to 3D simulations. Thus, it seems to be a valid candidate to increase the robustness of current icing tools.

1.2.3 Grid Considerations

Classical ice accretion tools use body-fitted meshes to represent the solid body of interest (e.g., an airfoil, a wing, a complete aircraft). Body fitted structured grids are difficult to generate on complex geometries since their topology impose constraints which are not easily satisfied. Typically, the orthogonality at the wall and the wall refinement are propagated into the domain. Unstructured grids alleviate this difficulty where wall orthogonality and refinement only have a local effect. Hence, it allows a simpler and more flexible mesh generation. There is, however, a price to pay in terms of computation costs and memory footprint as information about neighbouring cells, edges and vertices must be stored. Despite their differences, structured and unstructured grids still remain challenging to use for automatic mesh generation on complex ice shapes.

A more computation-efficient approach is to use Cartesian grids which allow the use of simple and fast multi-grid algorithms and line-iterative techniques [11]. This type of grid can also be associated with a memory efficient data structure. However the use of Cartesian grids on complex geometries requires a special treatment of the boundary conditions using an Immersed Boundary Method (IBM).



(a) Body fitted mesh (b) Immersed boundary mesh

Figure 1.3 Example of body-fitted and immersed boundary unstructured meshes around a NACA0012 airfoil (coarsened for visualization)

For a body-fitted approach, time and effort is spent on the generation of a mesh that matches the boundaries (Figure 1.3a). With Immersed Boundary Methods, the boundary is left immersed in the grid and crossing through the control volumes (Figure 1.3b). A special treatment is then applied to force the boundary condition on the immersed boundary. Hence, the use of IBMs can simplify the mesh generation and be applied to any kind of geometries.

Regardless of the type of grid, when performing RANS simulations, grid refinement at the wall is required to obtain a proper resolution of the viscous boundary layer. This boundary

layer is more accurately represented using body-fitted mesh. Alternatively, a Euler flow solver coupled with an integral boundary layer code allows the use of a coarser mesh in the wall vicinity and alleviate the difficulty of meshing the boundary layer. Although RANS is typically used to assess the effect of icing, a Euler flow solver coupled with an integral boundary layer solver can adequately predict ice shapes, at least for single element airfoils [12].

1.2.4 Ice Accretion and IBM

Immersed Boundary Methods have been applied with success in many fields but have rare applications to the prediction of ice accretion. The CIRA used a discrete IBM to solve compressible inviscid flows on 3D Cartesian grids in [13] which was later extended for compressible viscous flows [14, 15]. A discrete IBM was also applied to an Eulerian droplet solver [16] with the intent of performing ice accretion simulations. However, no ice prediction results using these IBMs have been shown to this day.

Another research team from the University of Strasbourg applied IBMs to a 3D ice accretion code (NSMB-ICE) with a level-set approach [17]. The compressible viscous flow solver employs a penalty method (an IBM) and the Eulerian droplet solver uses a discrete approach similar to the one of [16]. The level-set approach was initially proposed by [8] for single step ice accretion and is used to update the iced geometry. Multi-step ice accretion results are presented up to 5 steps on a single test case in [17]. According to the authors, the implementation is currently limited to laminar flow and rime ice. Furthermore, no comparison is made against a more classical body-fitted approach nor experimental results.

1.3 Thesis Objectives

As the mesh generation and the geometry evolution appear as the critical elements impacting the robustness and automation of ice accretion software, the general objective of this thesis is to investigate the potential of Immersed Boundary Methods to solve these issues by eliminating the user intervention in the mesh update while maintaining the accuracy obtained from a body-fitted approach. This implies the development of suitable numerical methods, their implementation in existing solvers and their verification and validation to assess the accuracy and robustness of the new proposed methodology.

Specific objectives:

1. Replace the re-meshing step by a more appropriate method to reach an automatic

process (i.e., no user intervention required);

2. Develop a method for the evolution of the ice accretion applicable to complex geometries (e.g., no pathological case);
3. Assess the accuracy and cost efficiency of the new aero-icing suite in comparison with a body-fitted grid and surface tracking approach (Lagrangian).

1.3.1 Constraints and Requirements

For this thesis, the selected development platform is IGLOO2D [12], a 2D unstructured ice accretion suite developed at ONERA. However, all the developments have to be made with a future 3D implementation in mind as IBMs show greater potential for 3D ice accretion.

IGLOO2D [12] can deal with both structured and unstructured meshes. It provides potential flow and Euler flow solvers for the aerodynamics, but it can also be coupled with an external aerodynamics solver (e.g., RANS solver). For the evaluation of the droplet impingement, Lagrangian and Eulerian approaches are available. The convective heat transfer and shear stress at the wall are obtained from a boundary layer code (when not available from a viscous flow solution). The evaluation of the ice accretion (thermodynamics) is based on the approach of Messinger [18]. The geometry evolution follows a simple node displacement technique (Lagrangian approach). The modules can be classified in two categories: (1) the volume solvers (aerodynamics, droplet trajectories) which are solved to steady state on a volume mesh (2D mesh for a 2D ice accretion suite) and (2) the surface solvers (boundary layer code and thermodynamics solver) which require a surface mesh (1D for a 2D ice accretion suite).

In this context, the methods developed in this thesis must be:

- applicable for both 2D and 3D simulations on complex geometries;
- applicable for both structured and unstructured grids (for generality and compatibility with IGLOO2D);
- automatic, i.e., do not require any user intervention;
- robust, i.e., no failure of the simulations;
- accurate in comparison with previous body-fitted methods;
- applicable to multi-layer icing simulations (multi-step);
- compatible with the surface solvers (Boundary Layer (BL) code, ice accretion module).

1.3.2 Suggested Methodology

To meet the objectives and requirements, the suggested methodology is to apply an IBM on a complete ice accretion suite, namely in the aerodynamic and droplet solvers (Figure 1.1). In addition, the Lagrangian geometry evolution will be replaced by an Eulerian approach, via a Level-Set method. These two main modifications should lead to an increased robustness and a simplification of the mesh generation.

Solving a viscous flow (e.g. RANS) implies a very restrictive mesh refinement in the wall vicinity in order to resolve the boundary layer. With an IBM, re-meshing can be replaced by a mesh adaptation technique in order to retrieve the required mesh resolution between each new ice layer. In this thesis, a simplifying assumption is to use the Euler equations for the aerodynamic field coupled with an integral boundary layer code (one-way coupling). Compared to a viscous flow simulation, it alleviates the grid refinement in the wall vicinity and allows the use of a more uniform grid size. With this approach, the mesh adaptation become less costly. Furthermore, with a proper generation of the initial mesh, re-meshing and mesh adaptation can be replaced altogether. The strategy is thus to generate a single mesh and re-use it for all subsequent ice layer without any modification.

The use of the level-set for the geometry update implies that only an implicit definition of the air-ice interface is available. Thus, an additional method to retrieve an explicit definition of the interface (e.g. surface mesh) must also be implemented to ensure the compatibility with the surface solvers (boundary layer and ice accretion).

1.4 Outline

This manuscript is presented as an article-based thesis and is structured as follows. In Chapter 2, a literature review is performed covering existing 2D/3D icing software, IBMs and their application, and interface tracking and interface capturing methods. Chapter 3 discusses the strategy for the application of the Immersed Boundary Method in the ice accretion software and introduces the articles presented in this manuscript. Chapter 4 presents the first article which covers the application of a penalization method (an IBM) to the Euler flow solver in IGLOO2D. Chapter 5 covers the second article which describes a penalization method for the Eulerian droplets solver. Chapter 6 contains the third and last article which provides the results for multi-step ice accretion using the previously developed IBMs and the level-set method for the geometry evolution. Chapter 7 discusses the benefits and limitations of the proposed methodology. Chapter 8 concludes on this research project by providing a summary of the completed work and by discussing future research paths.

CHAPTER 2 LITERATURE REVIEW

This chapter presents a literature review on several aspects required for the development of Immersed Boundary Methods for ice accretion.

First, a review of existing ice accretion codes provides an overview of the methods employed and identify any implementation involving Immersed Boundary Methods. Then, different types of IBMs are reviewed covering geometric, continuous and discrete methods. This helps select an appropriate IBM for the Euler flow and droplet impingement solvers.

Another review focuses on the representation of the air-ice interface (e.g., level-set method). It includes additional topics related to the implementation of the level-set method for ice accretion, such as the propagation of the icing velocity and the evaluation of the signed distance field.

2.1 Ice Accretion Tools / Icing Software

Several 2D and 3D icing codes from the literature are listed in Table 2.1 and 2.2, providing an overview of the variety of approaches. It does not provide an exhaustive list of existing ice accretion tools, but shows a selection of common ones. The features listed refer to the ice accretion process depicted in Figure 1.1.

First generation of ice accretion tools like LEWICE [4] and CANICE [5] are typically 2D and composed of a potential flow solver (e.g., panel method) coupled with an integral boundary layer to retrieve the heat transfer coefficient at the wall. A Lagrangian approach is used to track the droplet trajectories and determine the water impingement rate at the surface. Then the thermodynamic balance is performed using a Messinger-type approach [18], determining the ice thickness. The geometry update involves the displacement of the surface panels (a Lagrangian approach). If multi-step ice accretion is performed, only the panel distribution on the surface is updated (re-panelling). This selection of methods for the ice accretion software leads to fast compute times, however they are not well suited for configurations involving separated flow and flow recirculation (e.g., multi-element airfoil [19]). Moreover, they are not easy to generalize for 3D applications and more complex configurations (e.g., full aircraft).

The so-called second generation of icing tools as introduced by FENSAP-ICE [6, 20] involves the modification of the airflow and droplets solver to use general Partial Differential Equation (PDE) formulations. The airflow is modelled by either the Euler or RANS equations. The droplet trajectories and impingement rates are recovered using an Eulerian droplet

impingement solver [21]. When using the RANS solver, the wall heat transfer is directly extracted from the flow solution using the temperature gradients, an approach referred to as *from RANS* in the tables below. Furthermore, a PDE formulation of the thermodynamic model is also used, the Shallow Water Icing Model (SWIM) [22]. The methods introduced in this second generation of icing tools allow simpler handling of 3D general configurations while correctly representing detached flows, at the expense of calculation times.

Common icing software use a combination of methods from the first and second generations as shown in the Tables 2.1–2.2. An approach which lies in between is the use of the Euler equations coupled with a boundary layer code [12]. This allows faster computation and simpler mesh generation compared to using RANS solvers.

Table 2.1 List of 2D Icing Software

	CANICE	**LEWICE**	**IGLOO2D**	**NSCODE-ICE**	**PoliMIce**
Institution	Poly. Montreal	NASA	ONERA	Poly. Montreal	Poli. Milano
Grid type	Panels	Panels	Struct./Unstruct.	Structured	–
IBM	No	No	No	No	No
Airflow	Potential Flow	Potential Flow	Euler	Navier-Stokes	Navier-Stokes
HTC	Integral BL	Integral BL	BL code	from RANS	Constant
Droplets	Lagrangian	Lagrangian	Lagrangian/Eulerian	Eulerian	Lagrangian
Thermodynamics	Messinger	Messinger	Messinger or PDE	PDE (SWIM)	Myers
Geo. Evolution	Lagrangian	Lagrangian	Lagrangian	Lagrangian/Level-Set	Lagrangian
Mesh Update	Re-panelling	Re-panelling	Re-meshing	Re-meshing	Morphing
Domain	2D	2D	2D	2D	2D/3D
References	[5]	[4]	[12]	[23]	[7, 24]

Table 2.2 List of 3D Icing Software

	FENSAP-ICE	**LEWICE3D**	**IGLOO3D**	**NSMB-ICE**	**PoliMIce**
Institution	Uni. McGill	NASA	ONERA	Uni. Strasbourg	Poli. Milano
Grid Type	Unstructured	panels	Struct./Unstruct.	Structured	–
IBM	No	Yes (FUN3D)	Yes (ELSA/CEDRE)	Penalization/Discrete-Forcing	No
Airflow	Navier-Stokes	Navier-Stokes	Navier-Stokes	Navier-Stokes	Navier-Stokes
HTC	from RANS	Integral BL	BL code or from RANS	from RANS	Constant
Droplets	Eulerian	Lagrangian	Lagrangian/Eulerian	Eulerian	Lagrangian
Thermodynamics	PDE (SWIM)	Messinger	Messinger or PDE	PDE (SWIM)	Myers
Geo. Evolution	Lagrangian	Lagrangian	Lagrangian	Lagrangian/Level-Set	Lagrangian
Mesh Update	Deformation (ALE)	Re-panelling	Re-meshing	Re-meshing	Morphing
Domain	3D	3D	3D	3D	2D/3D
References	[6]	[25]	[26]	[8]	[7, 24]

Some 3D icing software are a combination of different tools which were not developed as a fully integrated icing suite, namely LEWICE3D and IGLOO3D. For LEWICE3D, the

aerodynamic data is provided by an external solver (e.g. RANS, Euler, potential flow, etc.). The droplet trajectories and ice accretion are handled by LEWICE itself. Furthermore, this software is not fully 3D as, for a wing, it computes 2D ice accretion cuts along the span and performs geometry lofting to obtain the 3D ice shape [27]. IGLOO3D is also a combination of multiple external tools, but it is actually fully 3D capable. Ice accretion results were published in [26, 28] along with a description of the 3D icing suite. Similarly, PoliMIce is an ice accretion tool performing the thermodynamic balance and the geometry update. It is designed to be connected to external solvers as well, namely OpenFOAM was used in [7, 24]. A 3D Lagrangian particle solver is also available in PoliMIce [29] with a recent application to rotorcraft using sliding meshes [30]. Other icing software not listed in the tables include, for instance, Multi-ICE [31, 19] developed by the CIRA and GlennIce developed by the NASA [32].

Considering the use of IBMs for in-flight ice accretion, NSMB-ICE [8, 17] is the only fully integrated icing suite under development towards IBMs (to the author's knowledge). However, note that IGLOO3D and LEWICE3D can be coupled with aerodynamic solvers featuring IBMs. For the ONERA, FastS [33, 34] can be used in combination with the Cassiopee preprocessor [35] to perform simulations using a direct-forcing method. For the NASA, FUN3D [36, 37] and CART3D [38] provide IBM functionalities. Furthermore, NSCODE-ICE and NSMB-ICE are the only ice accretion codes taking advantage of the Level-Set approach. Note that the research group of [39] uses an IBM and the level-set for the tracking of ice blocks but does not perform ice accretion.

2.2 Immersed Boundary Methods (IBMs)

For body-fitted grids, efforts are invested in the generation of good quality meshes matching the boundaries of the computational domain (i.e., farfield, wall of the geometry, etc.). Conversely, for IBMs the boundary freely cuts through the control volumes and effort is instead invested in the imposition of the boundary condition (Figure 1.3). Hence, it allows simpler mesh generation and simpler mesh adaptation and, in the context of ice accretion, it can ease or even eliminate the mesh update between ice layers. Furthermore, certain types of IBMs are well suited for simulating flows with moving boundaries, which could be a useful feature for ice accretion. As one would expect, IBMs also present some drawbacks. The main challenge of this method is the imposition of the Boundary Condition (BC) which is not as straightforward as for body-fitted grids. Also, the accuracy and conservation properties near the boundary are not ensured, depending on the type of IBM employed.

References [11] and [40] present a thorough review of the different IBMs available and their

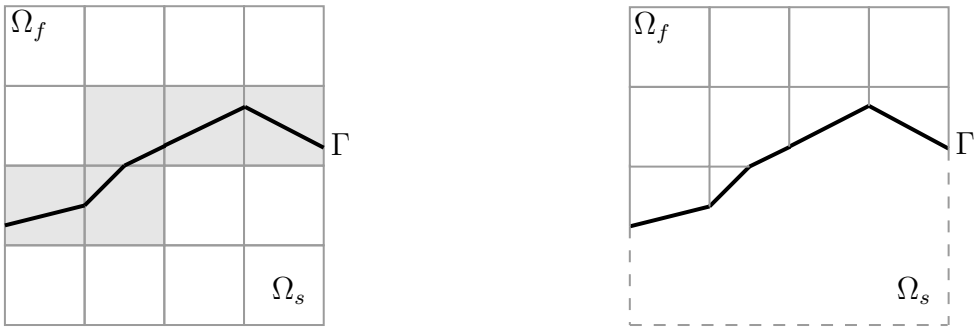
applications. They classify the methods in two different manners. In [40], the methods are divided in two main classes: diffuse interface and sharp interface methods. On the other hand, in [11] the classification is based on continuous forcing and discrete forcing. Here, the classification is inspired by the latter with the addition of one category.

It is convenient to define three classes: geometric methods, continuous forcing methods and discrete methods. This classification is based on how an existing solver or software has to be modified to account for the IB. For the geometric methods, the modifications mainly lie into the geometric preprocessing of the grid without major changes to the solvers, if not at all. The continuous methods apply a correction term to a continuous set of equations independently of the discretization. On the other hand, the discrete methods apply a special treatment directly into the discretized set of equations. Relevant methods and applications are reviewed in the following sections.

2.2.1 Geometric/Cut-Cell

For this family of methods, a body-fitted mesh is created from the intersection of the IB with the volume mesh. There is only one method in this category, the cut-cell approach. It is very different from the discrete or continuous forcing methods and is sometimes not even considered to be an IBM at all.

It was introduced by [41] to compute Euler flows around multi-element airfoils. The cells are cut by the interface to create cells of arbitrary shapes (Figure 2.1) which are body-fitted to the geometry. This implies that the solvers must be adapted to handle arbitrary cell types (arbitrary polygons) as the cutting process does not only produce triangular or quadrilateral cells (in 2D). This method is rather complex to implement due to the variety of possible shapes for the cut cells and the complexity rapidly increases in 3D. Note that this approach is mostly used on Cartesian grids and is usually restricted to specific problems such as inviscid compressible flows (e.g., CART3D [38]), but the method was also extended to viscous flows [42, 43]. For instance, [42] apply this approach to high Reynolds number RANS equations in combination with a level-set method.



(a) Cut cells identified in grey (b) Mesh after cell cutting

Figure 2.1 Representation of the Cut-Cell process (Ω_f: fluid zone, Ω_s: solid zone, Γ: interface)

The main advantage of the cut-cell method is its conservation property and accuracy (sharp boundary). In terms of implementation, the pre-processing of the grid (cell cutting) represents the major part of the work. The cells intersected by the interface must first be detected. Then, the cells are to be cut to represent the actual boundary. The cell cutting can get very cumbersome in three dimensions. It can also become costly if applied to problems with moving boundaries.

On the other hand, the solvers can remain for the most part unchanged. The main concern is about the stability and time step evaluation. Problems can arise when the cutting process generates a large difference in cell sizes (e.g., tiny cut cell vs. uncut cell). Hence, special techniques must be considered such as cell merging [44, 45, 46], cell linking [47] or cell mixing [48, 49]. As the cut cells are body conformal, the boundary condition can be applied as for body-fitted meshes.

Cut-cell and Level-Set: An interesting class of cut-cell methods use a Level-Set to implicitly track the interface. The Level-Set is useful to detect the cells intersected by the IB, hence simplifying the cell cutting task. For example, in [48] applications are presented for the Euler equations and provide a description to evaluate the intersection of the level-set with the grid faces. In [49], the level-set/cut-cell method is applied to incompressible Large Eddy Simulation (LES) with description of the mixing procedure for cut cells of small volume. In [50], the LS-STAG method is borrowing elements of the level-set, Marker and Cell (MAC) and cut-cell methods.

2.2.2 Continuous Forcing

For this category of methods, the boundary condition is enforced by the addition of forcing terms in the continuous form of the equations. Different continuous forcing approaches can be derived depending on the strategy employed to determine these forcing terms.

2.2.2.1 The Immersed Boundary Method of Peskin

Although the term IBM has a general meaning in this manuscript, the original *Immersed Boundary Method* was initially proposed in [51] and was used to solve the incompressible Navier-Stokes equation (blood) around immersed elastic boundaries (heart valves). This method is well suited for the simulation of elastic fibres or thin membrane immersed in a fluid, where the physical fluid is solved on both sides of the boundary. The approach allows the fluid to move the membrane and in turn, the elasticity of the boundary induces a force feedback on the fluid. The description of the method is first made in 2D for elastic boundaries followed by a short discussion on its application for rigid bodies.

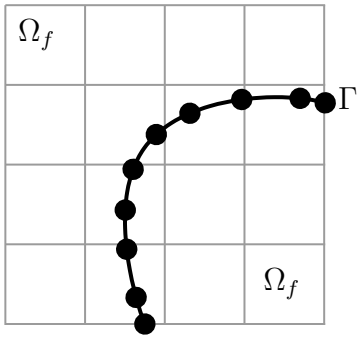


Figure 2.2 Representation of the Lagrangian surface mesh (Γ) surrounded by fluid (Ω_f) and immersed in a Cartesian grid.

This method uses a combination of Eulerian and Lagrangian variables where the flow is solved on a background grid and the boundary is tracked by the use of massless Lagrangian markers forming a surface (Figure 2.2) represented by curvilinear coordinates (s). The boundary moves with the fluid velocity and does not correspond with the background grid. Using the

notation from [52], the method is written as:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}\right)+\nabla p=\mu\Delta\mathbf{u}+\mathbf{f} \tag{2.1}$$

$$\nabla\cdot\mathbf{u}=0 \tag{2.2}$$

$$\mathbf{f}(\mathbf{x},t)=\int_{\Gamma}\mathbf{F}(s,t)\delta(\mathbf{x}-\mathbf{x}_k(s,t))ds \tag{2.3}$$

$$\frac{\partial \mathbf{x}_k(s,t)}{\partial t}=\mathbf{u}(\mathbf{x}_k(s,t),t)=\int_{\Omega}\mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{x}_k(s,t))d\mathbf{x} \tag{2.4}$$

where $\mathbf{x}$ and $\mathbf{x}_k$ represent the coordinates of the background grid and surface mesh, respectively. The incompressible Navier-Stokes equations with the forcing term $\mathbf{f}$ are given by Eqs. (2.1)–(2.2). Eq. (2.3) applies the boundary force on the fluid (background grid) and Eq. (2.4) moves the surface mesh with the fluid. The force $\mathbf{F}(s,t)$ defined on the immersed boundary is determined by a constitutive law (e.g., the Hooke's law) for elastic boundaries. As the markers do not generally coincide with the grid, the force is spread over a few grid points via a smooth approximation of the Dirac function $\delta(\mathbf{x}-\mathbf{x}_k(s,t))$. This smooth Dirac function is one of the key elements of the method since it ensures a smooth imposition of the boundary condition. However, it leads to a diffuse interface.

This method can be solved using an explicit time scheme but is likely to be unstable for stiff bodies (large forcing term). A stability analysis of the method is done in [53, 54]. Also, unconditionally stable implicit approaches are derived in [55, 56] and semi-implicit approaches are presented in [57, 58, 59, 60].

Because the forcing is spread using the smooth δ function, the method is limited to 1^{st} order accuracy in space near sharp interfaces. Formally 2^{nd} order accurate implementations of this method are derived in [52, 61, 62]. The term "formally" is used here since the method is second order accurate everywhere except in the vicinity of the interface; hence it is not strictly 2^{nd} order.

An extension of the method accounting for the mass of the IB is presented in [63, 64]. More details about the implementation of the method can be found in [64].

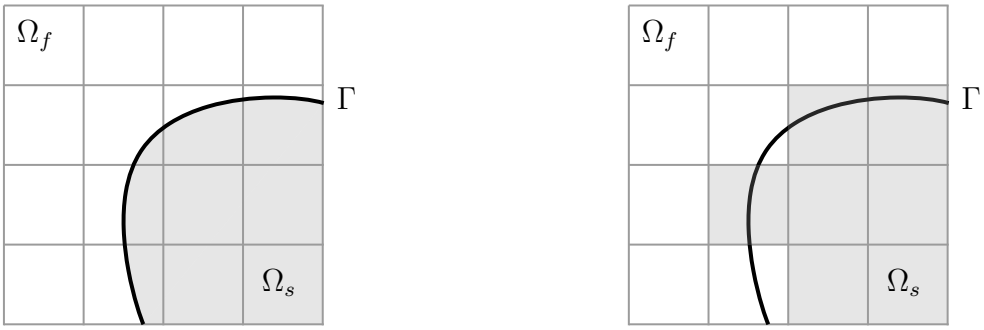
Rigid Bodies A rigid body can be simulated by modelling a very stiff boundary. For instance, Lai et al. [52] applied the idea of a spring with a restoring force to an equilibrium location denoted $\mathbf{x}_{k,e}$.

$$\mathbf{F}(s,t)=-\kappa\left(\mathbf{x}_k(s,t)-\mathbf{x}_{k,e}(s)\right) \tag{2.5}$$

Here κ must be large to represent the rigid body, leading to a stiff system of equation. Also, when the value of κ is too low, spurious oscillations can occur. Another approach is to observe the problem as a body with a permeable boundary, leading to the so-called penalization method presented in the following section (§2.2.2.2).

2.2.2.2 Penalization/Penalty

Another approach in the continuous forcing class of methods is the penalization method. This type of approach can be found under various names in the literature such as penalization method, penalty method, Brinkman penalization and fictitious domain method. It was originally presented in the work of [65, 66] to simulate the flow in a porous medium using a Darcy drag law. It was later applied to solid obstacles in [67] by simulating a porous boundary with very low permeability.



(a) Ideal representation (b) Staircase representation

Figure 2.3 Representation of the solid zone Ω_s (penalized zone in gray)

This type of approach is generally applied to the incompressible Navier-Stokes (NS) equations in the laminar regime, where the momentum equation is penalized to enforce the no-slip velocity boundary condition. The penalized incompressible NS equations read:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} + \frac{1}{\rho}\nabla P - \mathbf{g} &= \mathbf{f} \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{2.6}$$

with

$$\mathbf{f} = \lambda\chi(\mathbf{u}_{bc} - \mathbf{u}) \tag{2.7}$$

where $\mathbf{u}_{bc}$ is the velocity to be enforced at the immersed boundary and χ is the solid mask function. The forcing term or penalization term ($\mathbf{f}$) enforces the Dirichlet condition $\mathbf{u} = \mathbf{u}_{bc}$

only in the solid zone (Ω_s, Figure 2.3) while the usual NS equations are solved in the fluid (Ω_f). The mask function generally takes the form of a sharp Heaviside function (Eq. (2.8)). It can be evaluated based on a signed distance function (ϕ) with $\phi < 0$ in the solid, $\phi > 0$ in the fluid and $\phi = 0$ on the immersed boundary.

$$\chi = \begin{cases} 0 & \phi > 0 \\ 1 & \phi \leq 0 \end{cases} \tag{2.8}$$

The penalization parameter must be large ($\lambda \gg 1$) in order to represent a solid body (e.g., $\lambda = 10^{10}$ [68], $\lambda = 10^8$ [39]). Note that the accuracy of the method depends on the value of λ [67]. This creates a very large source term compared to the other terms in the equation. For this reason, the resulting stiff system is typically solved via implicit or semi-implicit methods. Alternatively, a Strang splitting is applied to the PDE in [68, 69], avoiding the time step restriction introduced by the penalization term. The approach allows more flexibility in the selection of numerical schemes.

Accuracy The classic implementation of the volume penalization method is limited to 1^{st} order accuracy in space because the boundary condition is applied on an approximation of the IB. For instance, in a cell-centred finite volume implementation on a Cartesian grid, the boundary is applied to the nearest solid cell centre to the IB, leading to a staircase representation (Figure 2.3b). Second order accuracy can be achieved by using information at the discrete level (e.g., [70, 71]). Alternatively the accuracy of the method can be improved by using an automatic mesh refinement technique with a fine mesh near the IB [72, 68], reducing the discretization errors.

The penalization method is mostly applied to incompressible viscous flows, but extension to compressible flows have been discussed in [73, 72, 68, 74, 75]. Typically, the Brinkman penalization method is used to impose Dirichlet boundary conditions. However, the method is also applicable to Neumann and Robin boundary conditions according to [76, 75, 71, 77].

Applications related to icing In [39, 78] the penalized incompressible NS equations are combined with a level-set to simulate the motion of ice particles and ice blocks, but no ice accretion is performed. In [17], the penalization method is applied to the compressible NS equation in NSMB-ICE. It is combined with a level-set approach and is used to perform ice accretion predictions.

The only known application of a penalization method to the Euler equations is available in [79] using the Characteristic-Based Volume Penalization (CBVP) method [75]. It is important to

note that no reference were found on the application of the penalty method for the droplet equations.

2.2.3 Discrete Forcing

To implement discrete forcing methods, modification are made to the discretized equations to enforce the desired boundary conditions. Some geometry pre-processing is usually necessary in order to detect interface cells, solid cells, fluid cells, ghost cells, faces crossed by the IB, etc. Their implementation is thus not as straightforward as for continuous methods. However, they allow a sharp representation of the immersed boundary and do not add extra stability constraints (i.e., an explicit solver can be used). Discrete methods are suitable for high Reynolds number (Re) flows but are more costly to apply to simulations with body motion because of the required pre-processing. This section provides some examples of discrete forcing methods.

2.2.3.1 Direct-Forcing

The direct forcing IBM was introduced in [80]. It is also referred to as *indirect forcing method* in [11]. In opposition to the penalty method, where a stiff forcing term is added to the continuous form of the equation, the direct forcing approach imposes the forcing on a time-discrete equation. For instance, [80] applied this method to the incompressible NS equations:

$$\frac{\partial \mathbf{u}}{\partial t} = RHS + \mathbf{f} \tag{2.9}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.10}$$

where $\mathbf{f}$ is the forcing term and RHS includes the convective term, viscous term and pressure gradients. The time-discrete equation gives for the forward Euler scheme:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = RHS + \mathbf{f} \tag{2.11}$$

If a velocity $\mathbf{u}_{bc}$ is to be imposed at the boundary (Γ), then the forcing term can be determined from the velocity update and gives:

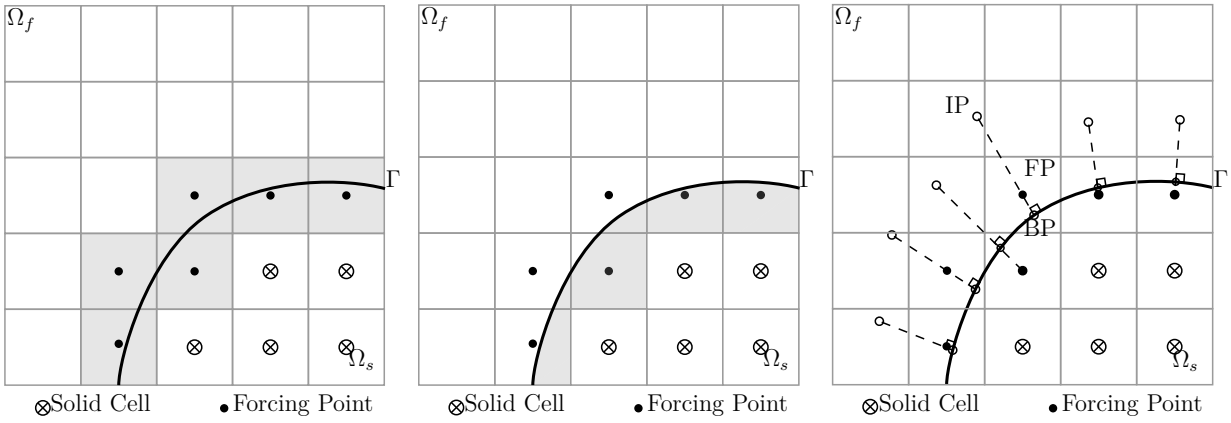
$$\mathbf{f} = \begin{cases} -RHS + \frac{1}{\Delta t}(\mathbf{u}_{target} - \mathbf{u}^n) & \text{on } \Gamma \\ 0 & \text{elsewhere} \end{cases} \tag{2.12}$$

In [81], the forcing term ($\mathbf{f}$) is applied to the equations for the computation points closest to the IB (e.g., cell centres) and the accuracy of the method is determined by the evaluation of $\mathbf{u}_{target}$. Assuming a Finite-Volume cell-centred formulation, three approaches are described in [81] and illustrated in Figure 2.4.

(a) $\mathbf{u}_{target} = \mathbf{u}_{bc}$ (Figure 2.4a). If the boundary lies on the cell centre, the forcing term leads to the exact imposition of the boundary condition. However, this situation rarely occurs, thus leading to a staircase representation of the IB. This approach is very simple but leads to a diffuse interface as the boundary condition is not directly applied on the IB.

(b) $\mathbf{u}_{target} = C\mathbf{u}_{bc}$ (Figure 2.4b) where $C \in [0, 1]$ is the volume fraction (colour function) which can be computed from a VoF method (see for instance [82]). According to [81], the weighting method allows a 1^{st} order accuracy.

(c) $\mathbf{u}_{target}$ is evaluated at the forcing point (FP) from a linear interpolation (Figure 2.4c). A line perpendicular to the IB and passing through the FP is constructed. $\mathbf{u}_{bc}$ is known at the Boundary Point (BP) and a value $\mathbf{u}_{IP}$ is interpolated at an Image Point (IP) from surrounding fluid data points. The FP can either be on the fluid or solid side of the interface. This approach leads to 2^{nd} order accuracy.



(a) No interpolation (staircase) (b) Volume fraction weighting (c) Linear interpolation

Figure 2.4 Representation of 3 interpolation techniques for the direct-forcing approach

Various strategies can be employed to determine the value of $\mathbf{u}_{target}$ with varying level of accuracy. For instance, the forcing point (FP) can be selected to be only on the fluid side or only on the solid side of the interface. However, the particularity of the direct-forcing method

lies in the imposition of the boundary condition through a forcing term in the time-discrete form of the equations.

Contrary to the penalty method, the direct forcing method does not lead to a stiff system of equations. The time-discrete equations are used to compute the forcing term, which allows the use of explicit scheme with a reasonable time step. Furthermore, in the penalty method, tuning of the permeability constant λ is required to mimic a solid body. For the direct forcing it is not the case. The method is therefore not user dependent and is more automatic.

However, the implementation of the method and the definition of the forcing term depend on the time discretization. Thus, the forcing must be adapted to the time scheme. For instance, forcing terms for the incompressible NS equations are given in [80] for third order Runge-Kutta and semi-implicit Crank-Nicholson schemes.

Some Applications In [37], an IBM is implemented for icing applications in FUN3D, based on the method of [83]. The idea is to start from a body-conformal grid. As the ice accumulates on the surface, the detection of *inside* and *outside* cells allow the detection of the immersed boundary. The BC is imposed at the wall via direct forcing. A refined structured grid is overlaid on the ice shape via an overset grid method, hence the grid is only updated locally. This paper provides one rare application of the IBM to aircraft icing (effect of icing). The combination of the overset grid method and IBM is interesting and shows that comparable accuracy can be obtained on ice shapes in terms of C_L, C_D, with the IBM results being slightly less accurate. According to [37], it can be attributed to the boundary layer spacing which is too coarse near the ice horns tips. It shows the additional difficulty of IBM when dealing with viscous flows. Note that no ice accretion is performed in this paper.

At ONERA, a mixed overset/direct-forcing method is used in FastS [34] where the pre-processing is performed by Cassiopee [35]. A fringe of points is defined in the fluid to be used for the interpolation procedure and the forcing point are selected in the solid zone only, similar to a Ghost-Cell approach (§2.2.3.2). This method was applied to compressible high Reynolds number flows.

2.2.3.2 Ghost-Cell

Instead of adding a forcing term to the continuous or to the time-discrete equations, the ghost-cell approach set solid cells (ghost cells) to the correct value in order to impose the boundary condition on the IB. The ghost cells are the computation points (e.g., cell centres) located in the solid closest to the IB. The process is very similar to a body-fitted approach except the fluid point used to set the ghost cell does not match the grid and must be interpolated.

The main particularity of this method lies in the identification of Image Points (IP) used to set the corresponding Ghost Cells (GC). Assuming the normal to the interface is known at least in the vicinity of the IB, the general process is as follows (Figure 2.5):

1. Identify the Ghost Cells as the nearest solid cell-centres to the IB.
2. Identify the Image Points. First a ray normal to the interface is cast from the ghost cell. The image point is located on this ray at a specified distance. For instance, it can be a symmetric point to the ghost cell or the same distance for all the ghost cells.
3. The variables are interpolated at the IP from the neighbouring fluid points.
4. The ghost cell is set to impose the correct boundary condition on the Boundary Point (BP) via linear interpolation between the Image Point (IP) and the Ghost Cell (GC).

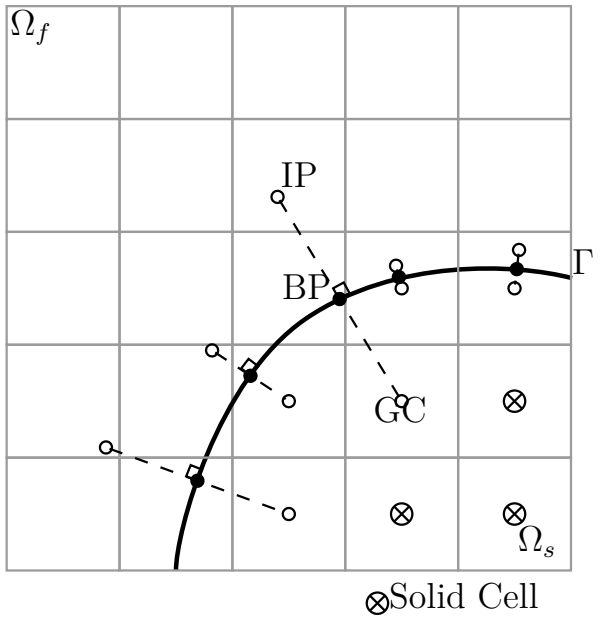


Figure 2.5 Representation of the image points (IP) and the ghost cells (GC) for the ghost-cell approach.

The accuracy of the method depends on the order of the interpolation. In [84], 1^{st} and 2^{nd} order interpolation are discussed for this type of method. Furthermore, the selection of the distance for the IP affects the smoothness and accuracy of the results. Using the same distance to the interface for all the IPs can result in a smoother solution.

With the ghost cell method, there is no problem associated with tiny cells as for the cut-cell method. There is no stiff source term incorporated in the equation as for the continuous methods. In fact, no modification is made to the equations and the usual solvers can be

reused. The method can be incorporated as an external library where identification of the IPs and GCs is performed. However, modification must be made to the boundary condition routine to interpolate the IP and set the GC. The method is simple to apply for 3D simulations. The pre-processing for this method is less expensive than the cut-cell method but more than the continuous forcing methods. As for the cut-cell method, this method can be costly for the simulation of moving interface since it requires the re-identification of the IPs and GCs.

This approach is similar to some implementations of the direct forcing method if the forcing points were to be selected only in the solid. The difference between the two methods lies in the imposition of the immersed boundary condition. For the ghost-cell approach the value is directly set in the ghost cell while for the direct-forcing method this ghost-cell (or forcing point) is forced to the correct value via a forcing term in the time-discrete equations.

2.2.3.3 Face Forcing

A discrete forcing method similar to the ghost-cell approach is presented in [13], where the fluxes are imposed on faces crossed by the IB to obtain the correct boundary condition. In this thesis it is referred to as *face forcing*. The approach is applied to Euler flows in [13], to RANS equations in [14]. In [16], a similar approach is applied to solve an Eulerian droplet field with the intent to perform ice accretion in future research. The ice accretion results have not been published yet, if any.

This IBM follows a procedure similar to the ghost-cell approach where an IP is identified and interpolated. However the IP is constructed using the normal to the interface passing through the face centre. With this method, there is no need to identify ghost cells, but the face intersected by the interface must be identified. The procedure is summarized as follows (Figure 2.6):

1. Trace the normal to the interface passing through the face centre;
2. Set an IP at a fixed distance from the IB;
3. Interpolate the flow quantities at the IP from the surrounding fluid cells;
4. Interpolate the flow quantities at the face centre to impose the correct BC at the interface (linear interpolation between the IP and the boundary);
5. The flux is obtained by multiplying by the entire face area.

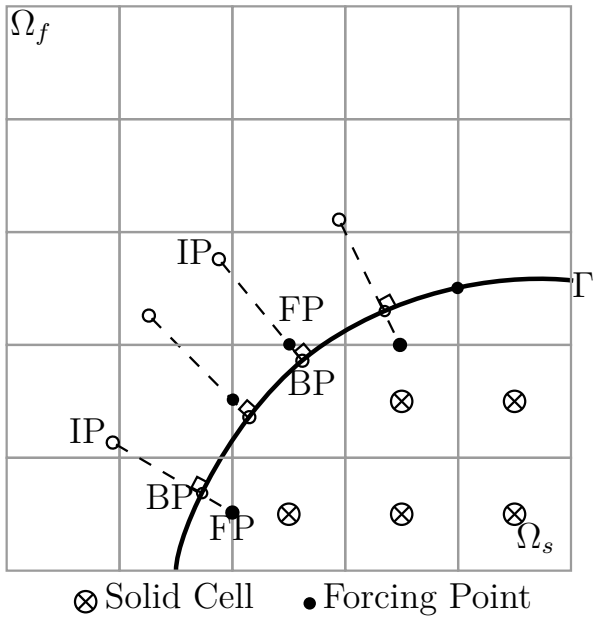
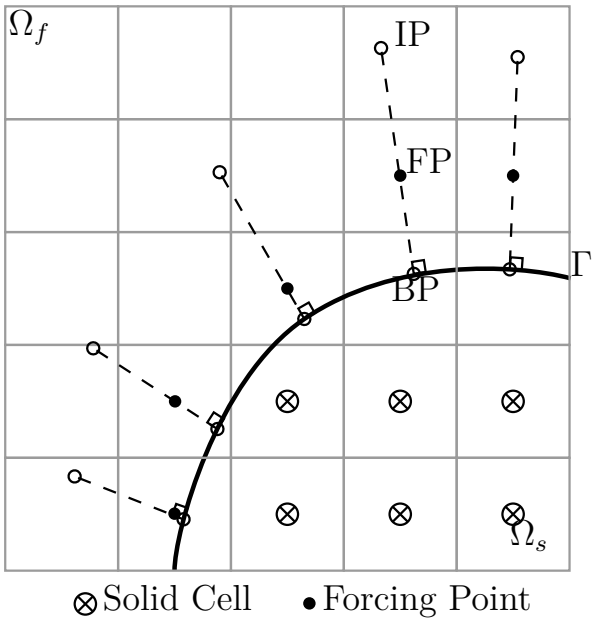


Figure 2.6 Representation of the face forcing method

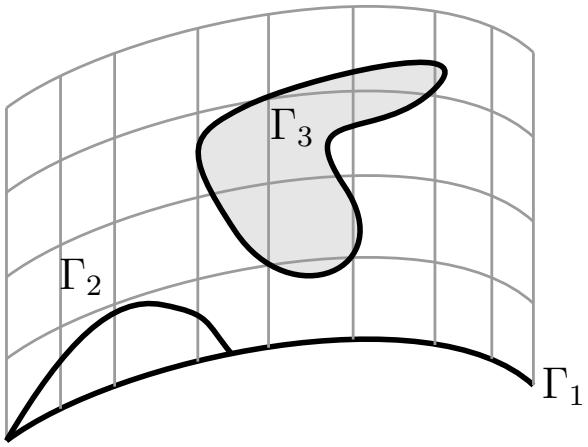
From the literature [13, 14, 16, 15], it is not clear whether the face forcing is applied only for fluid face-centre. According to [13], the IB-faces (where the forcing is applied) are faces sharing a fluid cell and a solid cell. If those cells are defined according the the location of their cell-centre, thus some IB-faces can be located on the solid side as shown in Figure 2.6.

2.2.3.4 Curvilinear Immersed Boundary Method (CURVIBM)

The CURVIBM [85] is similar to the face forcing and ghost-cell approaches. The difference lies in the forcing which is applied to the fluid cell-centre nearest to the interface. A similar approach is used in [17] to solve the Eulerian droplet impingement for ice accretion. Considering Figure 2.7a, the boundary condition to impose is known at the Boundary Point (BP). An Image Point (IP) is selected on a ray normal to the interface and passing through the Forcing Point (FP). The IP is interpolated using its neighbouring fluid values and the FP is finally set by a linear interpolation between the IP and BP in order to impose the boundary condition at the interface.



(a) Forcing on the fluid side only



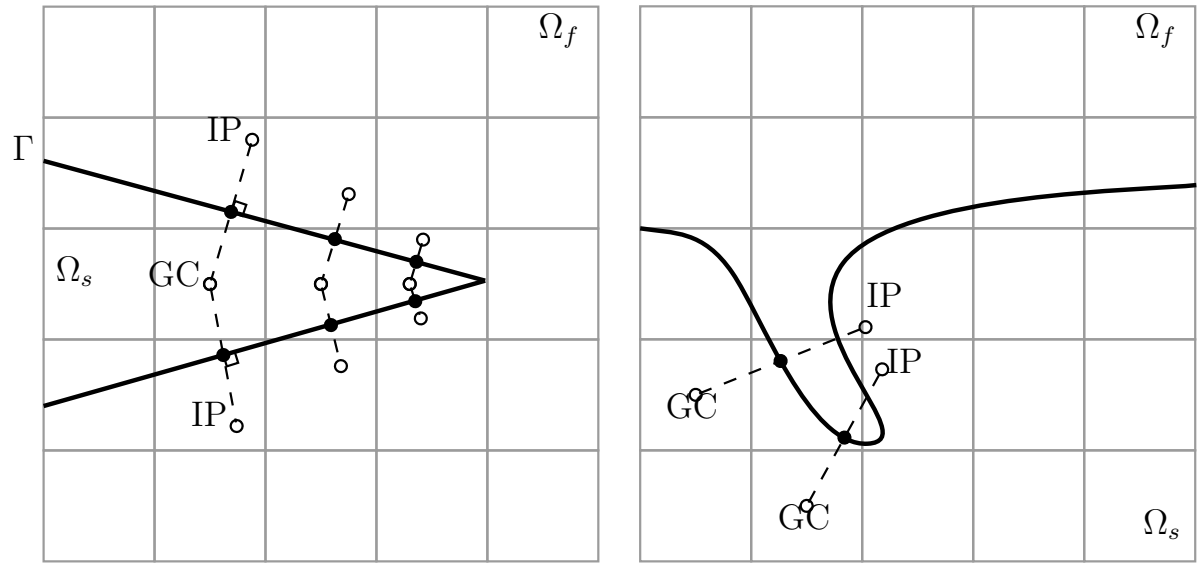
(b) Mixed body-fitted and immersed boundaries. Γ_1: body-fitted, Γ_2 and Γ_3: Immersed Boundary Method.

Figure 2.7 Representation of the CURVIBM

The CURVIBM combines the use of curvilinear body-fitted grids on simple geometries and an Immersed Boundary Method on complex ones (Figure 2.7b). This makes for a powerful tool since the mesh quality is likely to be better where a body-fitted grid can be generated easily. The method is extended to the use of overset grids in [86] where the method is called overset-CURVIB.

2.2.3.5 Degenerate cases

For the discrete methods, the definition of the Image Points and Forcing Points (or Ghost Cells) can lead to degenerate cases. For instance, the projection of the GC onto the IB can lead to more than one possibility for the Image Point [87, 88] as illustrated in Figure 2.8a. This situation may also occur for a forcing point located on the solid side of the IB (i.e., direct-forcing, face forcing). Furthermore, for thin geometries like sharp trailing edges the Ghost Cell can be undefined or located in the fluid (if 2 layers of ghost-cells are used [88]). For concave features, the IP might be located in the solid which makes it an invalid point as no interpolation is possible from fluid neighbours [89] as shown in Figure 2.8b. These configurations must be treated with care to obtain a robust method.



(a) Multiple image points for a single ghost cell (b) Symmetrical image point in the solid zone

Figure 2.8 Example of pathological cases for the ghost-cell method

According to [13], the face forcing approach has the advantage of avoiding the degenerate situations related to the identification of ghost cells and requires no special treatment for sharp trailing edges. This observation is correct if the forcing points are located on the fluid side only like for the CURVIBM (Figure 2.7a). However, it may still be problematic to identify suitable Image Points with fluid neighbours, similar to Figure 2.8b.

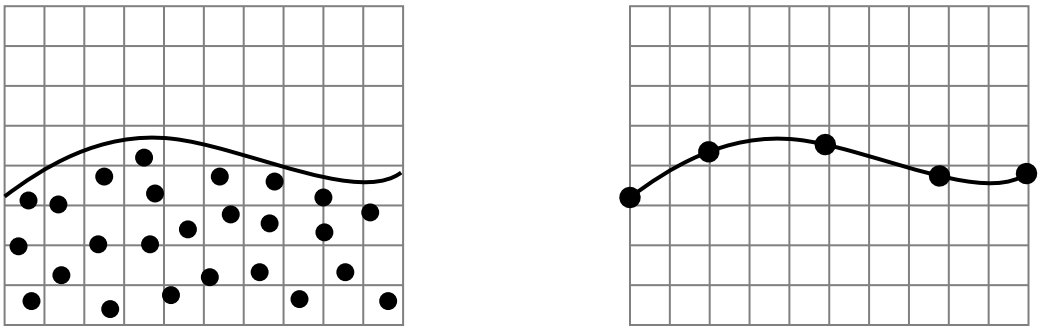
2.3 Surface Representation

There are two large families of methods for the evolution of an interface: tracking and capturing techniques. In a general point of view, the interface tracking approach is based on a Lagrangian definition while the interface capturing method is based on an Eulerian definition. These two categories of methods are discussed in the following sections.

2.3.1 Interface Tracking

Interface tracking methods are useful to solve problems with a discontinuous interface (e.g., two-phase fluid). They are characterized by a Lagrangian tracking of the interface, generally through the use of markers. In this section, three large families of methods are discussed: volume markers, surface markers, moving mesh methods. A review can be found in [90, 91] for instance.

The first strategy is to use massless markers to identify the fluid in the grid as initially proposed in [92] with the Marker and Cell (MAC) method. With this technique, the markers are distributed over the volume (Figure 2.9a) and tracked in a Lagrangian manner with the interface. Multiple types of markers can be used in a single simulation which makes it useful for simulations involving different types of materials (e.g., more than 2). However, the location of the interface is only approximated by the markers and typically, more markers are required near the interface for an accurate representation, which can become expensive [90]. Moreover, the interface must be reconstructed from the neighbouring volume markers, which is less accurate than simply using surface markers. A parallel can be made with the Volume of Fluid (VoF) method (§2.3.2.2) which tracks the fraction of material present in a cell instead of using markers to determine this fraction.



(a) Volume Markers (b) Surface Markers

Figure 2.9 Illustration of surface tracking methods using markers

With surface markers methods, connected markers are located on the interface forming a surface mesh (Figure 2.9b) and tracked in a Lagrangian manner. Better accuracy is achieved with fewer markers (compared to volume markers) as they lie directly on the interface. It allows a sharp definition of the interface and the accurate representation of thin flow structures (e.g., filaments). Since the velocity is not specifically available on the interface, interpolation from the control volumes onto the markers is required. This can reduce the accuracy of the method, depending on the interpolation method employed. Furthermore, these methods have difficulty to handle large deformations. The interface can be deformed with its motion leading to poorly defined zones (not enough markers). Re-seeding is thus advised to maintain the accuracy. Also, topological changes are not straightforward to deal with for this type of method (e.g., front merging). The main drawback is the complexity of implementation, but these methods can track the front very accurately. A significant body of research is available on such methods both in 2D and 3D (e.g., [93, 94, 95, 96]).

Volume marker and surface marker methods use a fixed mesh combined with the Lagrangian markers. Another strategy is to move/adapt the mesh so grid points lie on the interface as discussed in [90, 91]. Grid moving methods are suitable for small interface displacements but large movements might require a complete re-meshing. Similar to the surface marker methods, the implementation can be complex especially in 3D and when dealing with front merging. This method relies mostly on mesh generation and mesh adaptation techniques.

In ice accretion applications, the Lagrangian node displacement method used in the geometry update (Figure 1.1) falls into the surface tracking/surface markers category. The air-ice interface is tracked using Lagrangian markers which happen to be the nodes of the surface mesh. It is generally desired to conserve the surface mesh connectivity during the advection process. This can become difficult as reseeding or marker merging might be required to obtain a usable and accurate representation of the interface. Contrary to the methods presented in this section, the markers describing the ice shape are moved using a velocity (icing velocity) which is not re-interpolated from the volume mesh. The icing velocity usually remains constant for each marker during the accretion process.

2.3.2 Interface Capturing

Interface capturing methods use an implicit definition of the interface through an Eulerian formulation. The level-set and Volume of Fluid (VoF) are the two main methods in this category.

2.3.2.1 Level-Set

The level-set method [97] consists in the advection of a n dimensions interface by solving a $(n+1)$ dimensions hyperbolic PDE in the domain. The interface is implicitly represented by a level-set ($\phi = constant$). The level-set function is advected in the domain using [97, 98]:

$$\frac{\partial \phi}{\partial t} + \mathbf{V} \cdot \nabla \phi = 0 \tag{2.13}$$

where $\mathbf{V}$ is a velocity field responsible for the motion of the level-set. When simulating a solid interface in a fluid, it is convenient to select the level-set function (ϕ) as,

$$\begin{cases} \phi > 0 & \text{fluid} \\ \phi = 0 & \text{interface} \\ \phi < 0 & \text{solid} \end{cases} \tag{2.14}$$

where the zero level-set is the interface. Furthermore, if ϕ is selected as a signed distance function, then the normal $\mathbf{n}$ and curvature κ of the interface are directly available from [98]:

$$\mathbf{n} = \frac{\nabla\phi}{||\nabla\phi||} \tag{2.15}$$

$$\kappa = \nabla \cdot \frac{\nabla\phi}{||\nabla\phi||} \tag{2.16}$$

The level-set method automatically handles topological changes and multiple interfaces, but in opposition to the VoF, it is not mass-conservative. Compared to a Lagrangian approach, surface capturing methods are more expensive since $(n+1)$ dimensions are solved to obtain the definition of a n dimensions interface. However, the solution is usually smoother. The use of a signed distance field is very useful to obtain the implicit location of the immersed interface. It also helps in detecting the cells crossed by the interface (e.g., for IBMs) and it can be used in the imposition of the boundary condition.

As the level-set is advected into the domain, the signed distance to the interface is usually not conserved. An additional re-initialization step is therefore required to retrieve the correct solution (§2.4.2).

Furthermore, the velocity is required in the field or at least in the vicinity of the interface to advect the level-set. This information is not always available, thus another step can be added to extend the interface velocity into the field (§2.5). For more details, [98] and [99] provide a substantial review of the level-set approach.

Some applications with the Level-Set Many applications can be found for the level-set method. For instance, [100] use it with an IBM for Fluid Structure Interaction (FSI). Level-Set method can also be used for two-phase flows [101].

Recently, the level-set method has gained interest in the icing community. A Level-Set is coupled with a Stefan Problem for ice accretion in [102]. In [8], a single layer ice accretion model using a Level-Set is presented. The Level-Set is implemented to replace the typical node displacement method and avoid the issues of topology changes, like geometry intersection. The idea is extended to multi-layer (multi-step) icing simulation using the level-set in [17, 103].

Contour Extraction The level-set method provides an implicit definition of the interface using the signed distance field. However, it can be useful to retrieve an explicit interface definition (e.g., surface mesh) to solve the surface modules of the ice accretion suite (ice

accretion module and boundary layer code). As the signed distance field is available, the obvious approach is to use a contour[1] extraction technique to retrieve the explicit interface from ϕ. By detecting sign changes in ϕ, intersections between the implicit interface and the mesh can be identified to extract the contour. Well-known methods can be used for this matter such as marching cubes [104], marching tetrahedra [105] and marching diamonds [106].

The contour can also be obtained by adding a connected list of markers on the zero level-set (similar to interface tracking with surface markers) [107, 103]. This approach is similar to the Particle Level-Set [108] and Marker Level-Set [109, 110] methods which use unconnected markers to help retrieve a sharp definition of the interface. The markers can be used to correct the location of the level-set near sharp corners or in under-resolved regions of the Eulerian grid, thus improving the behaviour of the level-set. In [103], the normals from the signed distance field are used to advect the markers for ice accretion prediction. As the normals do not cross, it reduces the risk of introducing intersection in the surface mesh. After the advection process, a NURBS is fitted to the markers to retrieve a more regular contour. The difficulty comes from conserving the connectivity between the nodes defining the contour, especially in 3D.

2.3.2.2 Volume of Fluid (VoF)

With the VoF method, a variable represents the volume fraction of a fluid, gas or solid in a field. Let's use f to represent the volume fraction, defined as [111]:

$$f(\mathbf{x},t) = \begin{cases} 0 & \text{fluid} \\ [0,1] & \text{interface} \\ 1 & \text{solid} \end{cases} \tag{2.17}$$

The advection equation for the volume fraction (Eq. (2.18)) is the same as for the level-set. But contrary to the level-set, the function f is discontinuous at the interface.

$$\frac{\partial f}{\partial t} + \mathbf{V} \cdot \nabla f = 0 \tag{2.18}$$

Common methods for solving the VoF involve two steps: (1) advection of the volume fraction (Eq. (2.18)) and (2) geometrical reconstruction of the interface. The reconstruction technique defines the accuracy of the method and is also used to enforce mass conservation. For instance, the Simple Linear Interface Calculation (SLIC) [112] is a 1^{st} order accurate approach while

[1] Here the word *contour* is used to identify a contour in 2D or an iso-surface in 3D.

the Piecewise Linear Interface Calculation (PLIC) algorithm leads to 2^{nd} order accuracy (Figure 2.10). These 2D reconstruction methods are also discussed by [113, 91]. For the PLIC the interface reconstruction is made by estimating the normal computed from the negative gradient of the volume fraction ($\mathbf{n} = -\nabla f$) and by enforcing mass conservation using the volume fraction f.

(a) SLIC (b) PLIC

Figure 2.10 Reconstruction of the interface for the VoF method

The main advantage of the VoF over the level-set method is its conservation property (i.e., conservation of mass). Like the level-set, it also deals with topology changes automatically, but the reconstructed interface is not as smooth. The reconstruction scheme is important to retrieve the desired accuracy and maintain a sharp interface. Its implementation is rather complex in 3D as it involves dealing with multiple possible configurations for the front intersection with the computational cell. The implementation of the method is well known on Cartesian grids but its complexity increases for unstructured grids. Tools for a 3D implementation of the VoF are discussed in [114].

In [115], a high order scheme is used to solve the Eq. (2.18). It provides good conservation properties and avoids the geometric reconstruction of the interface (e.g., SLIC and PLIC) usually required for each advection step. The approach is also made applicable to general grids: Cartesian, structured, unstructured.

An alternative to the VoF has recently gained interest: the Tangent of Hyperbola Interface Capturing (THINC) [116, 117, 118]. It can be seen as an approximation of the VoF and uses a smooth approximation of the volume fraction (f) in order to avoid the complex geometric reconstructed of the PLIC. It is designed to be mass-conservative and offer limited smearing of the interface. It provides similar accuracy to the VoF method while being simpler to implement. Although the volume fraction is smooth when using the THINC, its gradient is still steep near the interface and thus, the geometric information of the interface is not as

accurate as for the level-set method.

2.3.2.3 Coupled Level Set Volume-of-Fluid (CLSVOF) Method

The CLSVOF method is used to overcome the weaknesses of both the level-set and VoF method [119]. The level-set provides the distance from a smooth interface, its normal and curvature. The VoF allows mass conservation through the volume fraction (f).

With this approach both the level-set and VoF are advected into the domain. The level-set location is corrected by the VoF to ensure mass conservation. A re-initialization of the level-set is performed after the correction step to retrieve the signed distance field. This process is repeated at each time step where the level-set and the VoF are both solved at each iteration. The process is summarized in Figure 2.11.

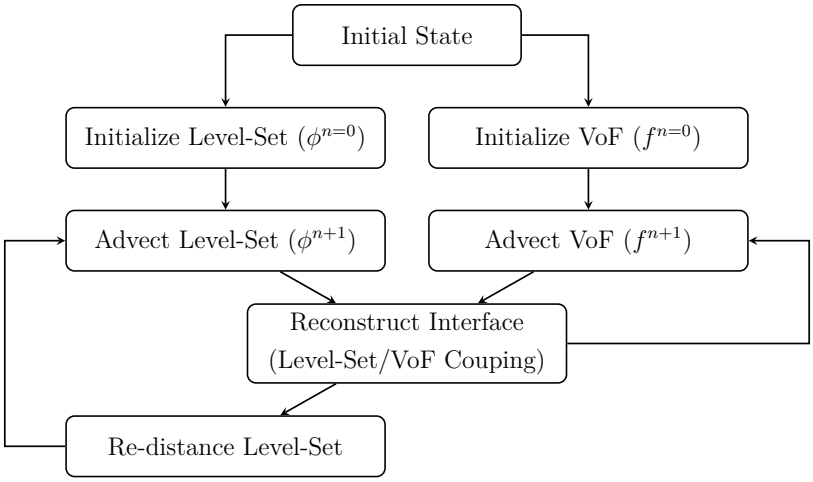


Figure 2.11 Flow chart for the CLSVOF method (based on [120])

The CLSVOF method has gained popularity and is used by several authors [121, 122, 123, 124, 125, 126]. For instance, the Adaptive CLSVOF (ACLSVOF) method is presented in [127] where the level-set and VoF are solved on 2D unstructured triangular meshes. Furthermore, an adaptive mesh refinement procedure is used to improve the accuracy of the method in the vicinity of the interface. The CLSVOF is used for 3D Cartesian grids combined with a ghost-fluid IBM in [120].

The main drawback of the CLSVOF method is the complexity of implementation and its computation cost, since both the level-set and VoF are required. On the other hand, it provides a superior solution compared to the individual level-set and VoF methods, providing both smoothness and mass conservation.

An alternative is to use the THINC (an approximation of the VoF) coupled with the level-set method [118]. It makes for a simpler implementation as no geometric reconstruction of the interface is required (e.g., no PLIC).

2.4 Evaluation of the Distance Field

The evaluation of the distance field from a geometry or a wall is required in many applications. In CFD, it can be used for instance in turbulence models or for the implementation of IBMs. Methods to evaluate the signed distance are discussed by [128] where a performance and accuracy assessment are also presented.

2.4.1 Signed Distance

Consider a domain Ω divided in a solid zone Ω_s and a fluid zone Ω_f with the fluid-solid interface denoted Γ (Figure 2.12).

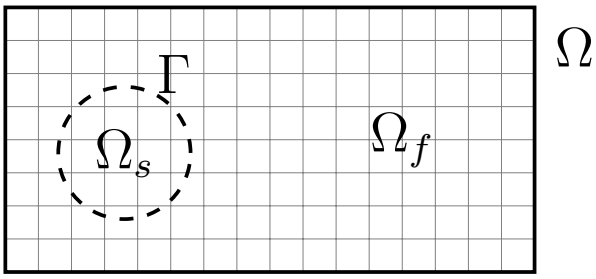


Figure 2.12 Definition of the domain for the signed distance

The signed distance is defined as:

$$\phi = \begin{cases} +d & \Omega_f \\ 0 & \Gamma \\ -d & \Omega_s \end{cases} \tag{2.19}$$

where d represent the minimum distance from a point $\mathbf{x} = (x, y)$ to the interface Γ. In this section, two approaches are reviewed: a geometric and a PDE-based approach.

2.4.1.1 Geometric Approach

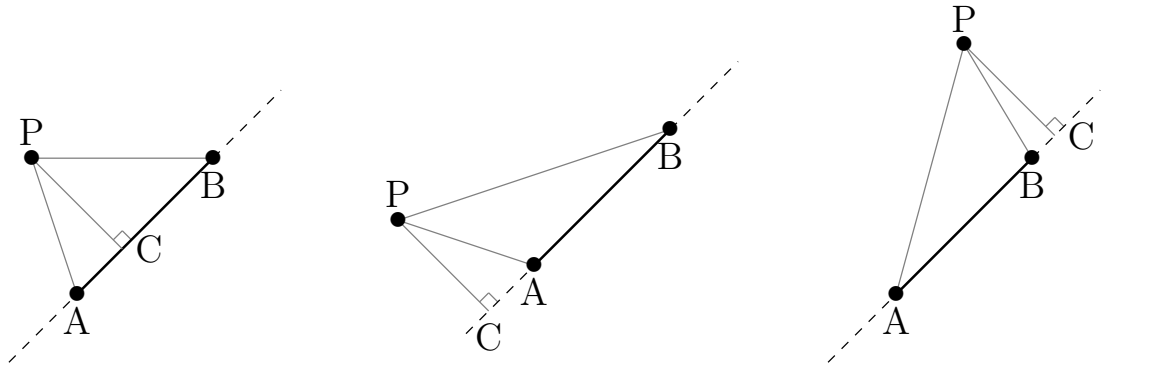
The signed distance can be computed algebraically using the coordinates of each cell centre and the location of the nearest interface element (e.g., an edge or a face). In 2D, the problem

falls back to computing the minimum distance between a point and a polygon defined by a list of edges.

The minimum distance between a point and a segment can be divided into 3 cases (Figure 2.13):

- the projected distance to the line (Figure 2.13a), or
- the distance between the point P and vertex A (Figure 2.13b), or
- the distance between the point P and vertex B (Figure 2.13c).

The cases in Figures 2.13b–2.13c occur when the projection to the line leads to an intersection outside the limits of the segment. Hence in this case, the minimum distance will be related to one of the two vertices. An algorithm to compute the minimum distance between a point and a segment is available for instance in [129].



(a) Minimum distance is $\overline{PC}$ (b) Minimum distance is $\overline{PA}$ (c) Minimum distance is $\overline{PB}$

Figure 2.13 Minimum distance between a point and a segment – representation of the 3 cases

The minimum distance between a point and a polygon can be determined by computing the distance between a point P and a list of segments (Algorithm 1). The sign of the distance can be evaluated using a Ray Casting algorithm (§2.4.3).

Algorithm 1 Minimum distance to a polygon

Require: list of edges or points defining the polygon

```
for cell in mesh do
  {sign is : inside=-1, outside=1}
  Determine the sign of the cell (via Ray Casting (algo. 2))
  for edge in polygon do
    Determine minimum distance cell-center to edge (point to segment)
    if (distance_current < distance) then
      distance = min(distance, distance_current)
    end if
  end for
  distance = sign × distance
end for
```

The method can be extended to 3D by computing the signed distance to a face instead of a segment. In this case, Algorithm 1 is modified to loop on the faces forming the surface. The faces are first split into triangles. Then, the minimum distance between a point and a triangle is computed. Similar to the 2D cases, the minimum projected distance to the triangle must lie in the triangle itself or on its limits (edge and vertices). A routine is proposed in [129] to perform this task efficiently.

This approach is accurate and can be parallelized, however the normals and the curvature are not readily available and must be computed by yet another step. For instance, the normals can be retrieved by evaluating the gradient of the distance field ($\mathbf{n} = \nabla\phi/||\nabla\phi||$). Also, an explicit definition of the interface is required to use this method which is not always the case for IBMs or when the level-set method is used.

2.4.1.2 PDE-Based Method

Another approach consists in solving the Eikonal equation (Eq. (2.20)). It can be solved using well-known techniques such as the Fast Marching Method [130, 131, 132], Fast Sweeping Method [133, 134] and other derivatives. It can also be transformed into a hyperbolic Partial Differential Equation (Eq. (2.21)) which can be solved using standard Computational Fluid Dynamics (CFD) techniques (e.g., Finite Volume Method (FVM), Finite Difference Method

(FDM)).

$$||\nabla\phi|| = 1 \tag{2.20}$$

$$\frac{\partial\phi}{\partial t} + \frac{\nabla\phi}{||\nabla\phi||} \cdot \nabla\phi = 1 \tag{2.21}$$

For instance, this type of approach is used in [135, 136] to compute wall distances. The wall boundary condition is set as a Dirichlet condition ($\phi = 0$) and a Neumann condition is applied to other boundaries ($\partial\phi/\partial\mathbf{n} = 0$). The distance front is thus propagated from the wall until a steady state is reached, providing the distance in the field.

If the distance is not computed from a wall but rather from an interface or an Immersed Boundary, the distance will have a different sign on each side: typically positive on the fluid side and negative on the solid side. The front propagation can be performed in the positive direction (Eq. (2.22)) on one side and in the negative direction on the other (Eq. (2.23)). The combination of the two leads to Eq. (2.24).

$$\frac{\partial\phi}{\partial t} + \mathbf{n} \cdot \nabla\phi = 1 \tag{2.22}$$

$$\frac{\partial\phi}{\partial t} - \mathbf{n} \cdot \nabla\phi = -1 \tag{2.23}$$

$$\frac{\partial\phi}{\partial t} + sign(\phi)\mathbf{n} \cdot \nabla\phi = sign(\phi) \tag{2.24}$$

with the normal and curvature to the interface defined as:

$$\mathbf{n} = \frac{\nabla\phi}{||\nabla\phi||} \tag{2.25}$$

$$\kappa = \nabla \cdot \frac{\nabla\phi}{||\nabla\phi||} \tag{2.26}$$

This approach is directly applicable to 2D and 3D simulations. Furthermore, it directly provides the normal and curvature to the interface. It can also evaluate the distance to an implicit definition of the interface which comes in handy for IBM.

The Eq. (2.24) is closely related to the re-initialization of the level-set described in the next section.

2.4.2 Re-initialization

As stated earlier, the level-set method sometimes requires the re-initialization of the signed distance field. The basic idea is to reset the signed distance both on the positive and negative

sides of the interface. Four approaches are considered.

1. The re-initialization can be done entirely via algebraic/geometric method (§2.4.1.1) if an explicit definition of the interface is available (e.g., a list of points).
2. The cell close to the interface can be initialized via the algebraic/geometric approach, then the remainder of the domain can be updated via a PDE approach (§2.4.1.2). This still requires an explicit definition of the interface.
3. A band of cells close to the interface can be "frozen" (not updated) and the rest of the domain solved via PDE. This approach assumes that the signed distance is valid close to the interface. It does not require the explicit definition of the interface and the location of the level 0 remains the same.
4. The field can be solved entirely via PDE (§2.4.1.2), but this approach has a tendency to move the level-set (the zero interface). This approach is described in more details below, as it is useful when dealing with an implicit definition of the interface.

For the PDE method, the re-initialization equation (Eq. (2.27)) is solved to a steady state to reset the signed distance field [137]:

$$\frac{\partial \phi}{\partial t} + \text{sign}(\phi_0)\left(\mathbf{n} \cdot \nabla \phi - 1\right) = 0 \tag{2.27}$$

$$\text{sign}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \varepsilon^2}} \tag{2.28}$$

where ϕ_0 is the initial signed distance field and $\text{sign}(\phi_0)$ is a smoothed sign function. The parameter ε is generally of the order of the cell size (Δx). According to [99], better results can be achieved by using the definition suggested by [138]:

$$sign(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + ||\nabla \phi||^2 \varepsilon^2}} \tag{2.29}$$

These sign functions are used to smooth the signed distance near the interface which presents a discontinuity (the sign changes from 1 to -1 abruptly). The re-initialization is generally applied for a few steps at each advection step of the level-set in order for ϕ to remain a signed distance field. However, this approach does not guarantee that the level zero will remain at the exact same location during the re-initialization. In [139], a method is proposed to preserve the location of the zero level-set by adding a correction term to the re-initialization equation. This approach was successfully used by [17] for ice accretion, conserving the ice mass during the re-initialization.

2.4.3 Ray Casting Algorithm

The ray casting algorithm is a useful tool to detect if a point, or a cell centre for instance, is inside or outside a solid body. This is required to determine the sign in the signed distance evaluation or to simply tag solid and fluid cells in an Immersed Boundary Method.

The idea is to cast a ray from a point P in the direction $(1, 0)$ in 2D. The number of times the ray crosses the polygon determines whether the point lies inside or outside the polygon. An even number of crossings is associated with the point being outside while an odd number of crossings means the point lies inside the polygon. The algorithm has the advantage to be valid for a general polygon and is therefore not restricted by the convexity hypothesis. The algorithm can also be extended to 3D simulations. The following algorithm (Algorithm 2) is inspired by [129] where instead of counting the number of crossings, a boolean is toggled each time a crossing occurs.

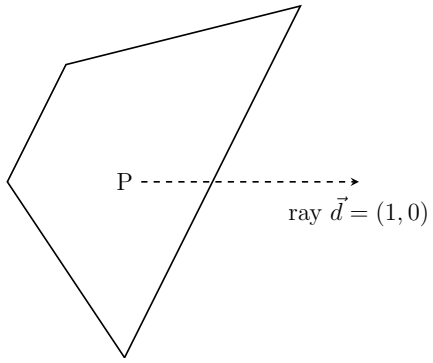


Figure 2.14 Representation of the ray casting algorithm

Algorithm 2 Ray Casting

Require: point P, Polygon
  inside = FALSE
  for edge in Polygon **do**
    if (edge.n1.y <= P.y **or** P.y < edge.n2.y) **then**
      compute x_intersect {*intersection of ray with edge*}
      if (x_intersect > P.x) **then**
        inside = **not** inside
      end if
    end if
  end for

Note that this approach is not the only possible one to detect inside and outside cells. This kind of problems is common for overset grid methods for which various techniques have been derived. For instance an X-Ray technique is described in [140] to detect inside and outside cells for closed 3D bodies where a comparison of different methods is also performed.

2.5 Extension/Propagation of Surface Data

For the level-set method (§2.3.2.1), a scalar ϕ is advected into the domain following a specified velocity field. Thus the velocity must be available in a few-cell-thick band near the interface. When $\mathbf{V}$ is only known at the interface, it must be propagated into the field (generally following a direction normal to the interface). A scalar a (e.g., the magnitude of $\mathbf{V}$) can be extended following the normal to the interface $\mathbf{n}$ using a simple advection equation [98, 99, 141, 142]:

$$\frac{\partial a}{\partial t} + sign(\phi)\mathbf{n} \cdot \nabla a = 0 \tag{2.30}$$

Notice that normal ($\mathbf{n}$) remains constant during the propagation process. It produces bands normal to the interface of constant value a.

2.6 Intermediate Conclusion

A review of ice accretion tools reveals that IBMs are not common in the icing community. In fact, only NSMB-ICE is known to be able to predict ice shapes using an IBM + Level-Set framework. A Penalization method is used for the aerodynamic field (compressible viscous flow) and a discrete forcing method is used for the Eulerian droplet impingement solver (forcing points located only on the fluid side). The capabilities seems to be currently limited to rime ice and laminar flows. Other ice accretion tools could use IBM capabilities by connecting external solvers (e.g., LEWICE3D, IGLOO3D) but they still lack an IBM for the droplets solver. This review shows that the capabilities of IBMs for multi-step glaze ice prediction is still to be investigated. Furthermore, the combination of the Euler equations with a boundary layer model has not been investigated yet for ice accretion using IBMs.

Amongst the different IBMs reviewed in this section, discrete forcing approaches such as the ghost-cell and face-forcing approaches are very promising as they are applicable to general meshes, second order accurate and require few modifications to the solvers. Furthermore, these methods have already been applied to the Euler and droplet equations on Cartesian grids. However, when dealing with complex geometries, problems arise in the determination

of the Image Points required for the accurate imposition of the boundary condition. The penalization method is a viable alternative despite its lower accuracy. It is applicable to general meshes, independent of the space and time schemes and require only the addition of source terms in the continuous form of the equations. Also, unlike discrete methods, there is almost no preprocessing required and thus, no issue related to the identification of the Image Point. The 1^{st} order implementation of the penalization method is simple to implement as it does not require information at the discrete level and it might provide sufficient accuracy for the prediction of ice shapes. The stiff nature of the penalized system of equations is not problematic as implicit solvers are primarily used in IGLOO2D.

Interface tracking methods are similar to the Lagrangian node displacement approach used in common ice accretion tools. They are accurate, but it is difficult to keep a connected list of Lagrangian markers to represent the surface mesh as the interface is displaced. Also, special treatment is required to handle geometry overlaps that may occur in concave regions. Interface capturing methods are an interesting alternative as they automatically handle this type of scenario. This is one of the desired characteristics to improve the robustness of ice accretion tools. Although the VoF is mass-conservative, the difficulty in reconstructing the interface from the volume fraction makes it less appealing. The surface normals provided by the VoF can also be noisy due to piecewise linear approximation of the interface. On the other hand, the level-set method can be written as an hyperbolic PDE which can then be solved using traditional CFD methods on general meshes. Thus, it is possible to re-use the code structure from other modules within the ice accretion suite. It is also easier to implement than the VoF as no geometry reconstruction is required. When combined with a signed distance field, the level-set method directly provides a smooth implicit interface, its normals and its curvature which is useful for the implementation of IBMs. The level-set is not mass-conservative but previous results from the literature showed satisfactory behaviour compared to a more classical node displacement approach.

CHAPTER 3 METHODOLOGY/THESIS STRUCTURE

This chapter presents the strategy for the implementation of the Immersed Boundary (IB) and Level-Set methods in an icing suite (e.g., IGLOO2D) to perform multi-step ice accretion simulations. It also introduces the three articles which constitute the next three chapters of this thesis and discusses their relevance in relation to the objectives.

As a reminder, the general objective of this thesis is to investigate the potential of Immersed Boundary Methods to solve the issues related to the mesh generation and the geometry update in ice accretion simulations. This implies the development of suitable numerical methods, their implementation in existing solvers and their verification and validation to assess the accuracy and robustness of the new proposed methodology.

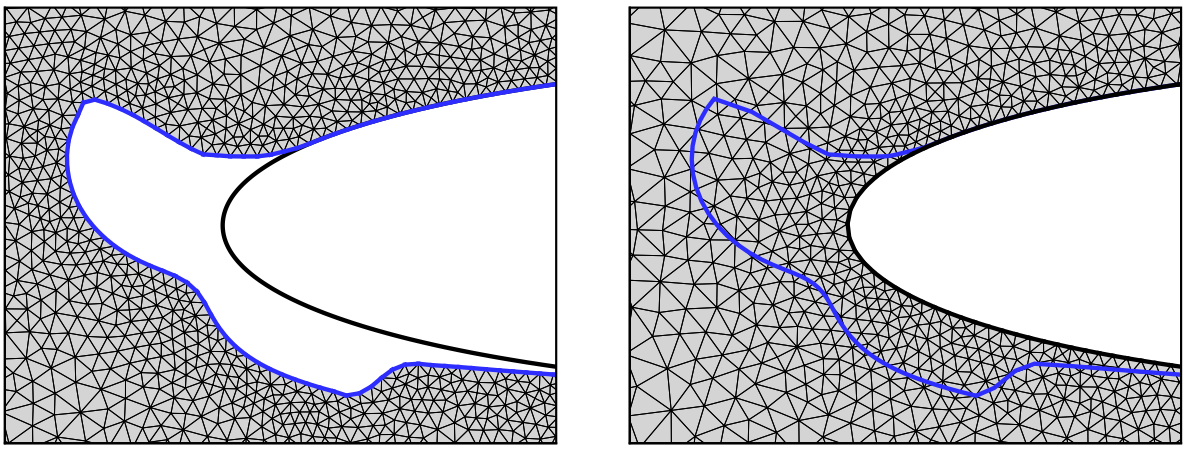
Specific objectives:

1. Replace the re-meshing step by a more appropriate method to reach an automatic process (i.e., no user intervention required);
2. Develop a method for the evolution of the ice accretion applicable to complex geometries (e.g., no pathological case);
3. Assess the accuracy and cost efficiency of the new software in comparison with a body-fitted grid and surface tracking approach (Lagrangian).

As an additional note on the first specific objective, it should be noted that current meshing technologies allow for automatic 2D and 3D re-meshing. The objective here is to use a simpler approach which would ideally lead to a lower possibility of failure on complex geometries typical of ice accretion.

3.1 Description of the Solution

In order to meet the requirements of automation and simplification of the mesh generation process, an IBM is used in the ice accretion suite. The selected strategy is inspired from the Curvilinear Immersed Boundary Method (CURVIBM) [85] which combines the use of body-fitted meshes for simple geometries and an IBM for complex ones. In the context of a multi-step icing simulations, it is desired to obtain the same result on a clean geometry (i.e., without ice) whether a body-fitted approach or an IBM is used. The strategy is thus to use a body-fitted mesh for the clean part of the geometry and use the IBM for the ice shape only.



(a) Re-meshing for the body-fitted approach (b) Initial mesh re-used with immersed boundary

Figure 3.1 Mesh representation – body-fitted vs immersed boundary approach. (Ice: thick blue line, Clean geometry: thick black line)

As a clean geometry is simpler to mesh than an ice shape, the process starts with a body-fitted mesh where the usual methods are used to generate the 1^{st} ice layer. Once the geometry is updated to account for the ice accretion, re-meshing usually occurs to match the new solid boundary as shown in Figure 3.1a. Instead of re-meshing, the iced geometry is left immersed in the mesh and the IBM is applied for the ice shape (Figure 3.1b). A mesh adaptation technique could be used to refine the mesh in the vicinity of the immersed ice shape. However, assuming that the mesh is initially refined where the ice is expected to build up, the initial mesh can be reused directly. With this approach, the zones where no ice accretion occurs remain body-fitted, retaining the associated accuracy, and the IBM is applied in the iced zone only, avoiding the re-meshing step. The idea of avoiding the re-meshing step completely is only viable because a Euler solver is used. If the RANS equations were to be solved, the mesh size would be too restrictive and would lead to a very large number of cells, especially in 3D.

The implementation of the IBM in the ice accretion suite requires modifications to existing modules and the addition of new ones as highlighted in red in Figure 3.2. The main modifications concern the volume solvers (airflow and droplets) for which an IBM must be implemented.

The boundary layer and the ice accretion (thermodynamics) are solved on a surface mesh which is body-fitted for the 1^{st} step, but can be immersed in the volume mesh for subsequent

steps. To limit the number of modifications to the ice accretion suite, it was decided to keep these surface modules untouched. These solvers still require the airflow and droplets solutions on the surface mesh, but the data is only available at the cell centres near the immersed boundary. Hence, routines are added to the ice accretion process to reconstruct the data on the interface (surface data extraction, Figure 3.2).

To deal with the issue related to the evolution of the iced geometry, the Lagrangian geometry evolution is replaced by a surface capturing approach (Level-Set), allowing the automatic management of geometry overlaps. The use of an implicit definition of the interface has many advantages, but an explicit definition is still required for the surface solvers (i.e., no surface mesh directly available). Hence, a surface mesh extraction is performed, consisting of two parts: (1) a contour extraction to obtain an explicit definition of the zero level-set and (2) a surface re-meshing (using GMSH) to obtain an appropriate refinement and node distribution. The former is performed in the surface mesh extraction module and the latter is done at the pre-processing step (Figure 3.2). It can be thought of as replacing the volume re-meshing by a surface re-meshing for each new ice layer when using the IBM.

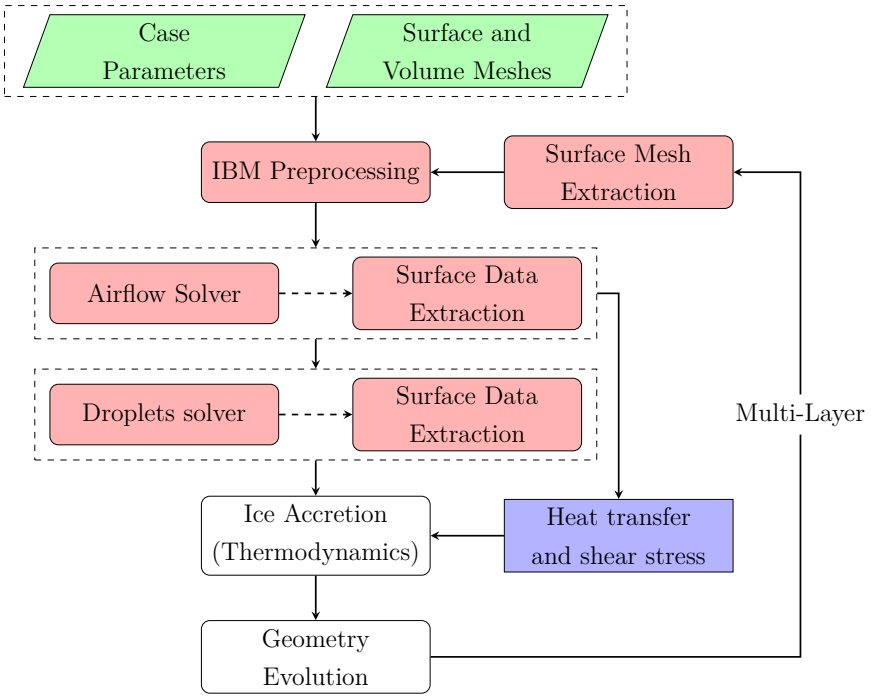


Figure 3.2 Multi-layer icing simulation. Necessary modifications for an IBM represented by red blocks

3.2 Presentation of the Articles

This section describes the content of the three articles used in this thesis and discusses their relation with the objectives and the previously defined implementation strategy.

3.2.1 First Article

The first article deals with one of the key modifications to the ice accretion software IGLOO2D: the implementation of an Immersed Boundary Method for the Euler equations suitable for ice accretion applications. This contributes to the main objective of the thesis by providing the first step towards an ice accretion suite using IBMs. The implementation of the IBM is also required in order to simplify the mesh update in the multi-step icing process, or in this case, to eliminate the re-meshing completely (specific objective 1).

Many types of IBMs are available from the literature as reviewed in §2.2 each with their own advantages and drawbacks. In this thesis, a continuous IBM (i.e., volume penalization) is selected for the Euler flow solver for its simplicity of implementation and independence on the discretization of the equations. The volume penalization method only requires the addition of source terms in the continuous form of the equation to impose the desired boundary condition. In the article, different combinations of source terms are tested, seeking a simple and accurate method to implement in comparison with a reference solution generated using a body-fitted approach. The source terms are selected to obtain a continuous solution near the interface which help in the surface data extraction by allowing the use of simple interpolation techniques.

The CBVP method of [75] is applied to the Euler equations in [79] and to the author's knowledge, it represents the only implementation of a penalization method for the Euler equations. The CBVP is implemented in IGLOO2D and shows promising results on clean geometries. However, for icing applications, the method shows some issues especially on geometries exhibiting ice horns typical of glaze ice conditions.

In order to improve the behaviour of the CBVP method on curved geometries typical of ice accretion, a new set of source terms inspired by [143] is implemented. The resulting method, called Characteristic-Based Volume Penalization-Hs (CBVP-Hs), is designed to conserve the entropy (s) and total enthalpy (H) in the normal direction to the IB. In the article, the CBVP-Hs is compared to the CBVP method, showing better mesh convergence and better conservation of entropy and total enthalpy on a cylinder, a clean NACA0012 airfoil and on an ice accreted GLC305 airfoil. The comparison is also performed against a body-fitted simulation (specific objective 3).

The implementation of this IBM also implies the addition of a surface data extraction process (Figure 3.2) which is required by the surface solvers. The data is retrieved using a Weighted Least Square (WLSQ) interpolation briefly mentioned in the article. More details can be found in Appendix A.

3.2.2 Second Article

The second article describes the development and implementation of an IBM for the Eulerian droplets solver in IGLOO2D. Following the main objective of the thesis, this article represents the 2^{nd} main modification towards a new ice accretion framework using IBMs. Furthermore, the implementation of an IBM for the droplets solver brings the multi-step icing process one step closer to avoiding the re-meshing step (specific objective 1).

In this article, a penalization method is again selected for its simplicity of implementation. Although a discrete IBM is applied to the Eulerian droplet equations in [16] and [17], no application of the penalization method is found for this system of equations. A new set of source terms is thus developed in order to adequately penalize the droplet equations, allowing impingement and avoiding re-injection from the fluid to the solid. Only Dirichlet conditions are required, leading to a simple implementation and low computation cost.

The accuracy of the method is assessed against body-fitted results on cylinders, a NACA0012 airfoil and an ice accreted GLC305 airfoil (specific objective 3). The surface data extraction process (Figure 3.2) required by the surface solvers is again performed using a Weighted Least Square (WLSQ) interpolation (Appendix A).

3.2.3 Third Article

The third article combines the penalization of the Euler (article 1) and droplet (article 2) equations, to perform multi-step ice accretion predictions using both the Lagrangian node displacement and level-set approaches (geometry evolution). In this article, the re-meshing between icing layer is totally avoided in the multi-step process (specific objective 1).

Multi-step ice accretion simulations are performed for rime ice and glaze ice cases where the body-fitted and IB approaches are compared using the Lagrangian method for the geometry evolution first. This helps isolating the effect of the IBM on the ice shapes compared to a body-fitted simulation (specific objective 3). Then, a comparison between the Lagrangian and Eulerian methods (geometry evolution) is made on a manufactured case in order to exhibit the ability of the level-set to deal with geometry overlaps (specific objective 2). The two methods for the geometry evolution are also compared on rime and glaze ice cases.

The use of the level-set method provides an implicit definition of the ice shape while the surface solvers require an explicit definition. Thus an approach to extract a surface mesh from the level-set solution is also described in this article.

CHAPTER 4 ARTICLE 1: AN IMPROVED CHARACTERISTIC BASED VOLUME PENALIZATION METHOD FOR THE EULER EQUATIONS TOWARDS ICING APPLICATIONS

P. Lavoie, E. Radenac, G. Blanchard, Éric Laurendeau, and P. Villedieu, "An improved characteristic based volume penalization method for the euler equations towards icing applications," Computers & Fluids, 2021. DOI: https://doi.org/10.1016/j.compfluid.2021.104917

4.1 Abstract

Immersed boundary methods (IBMs) are an interesting alternative to the usual body-fitted mesh approach when dealing with complex geometries as they allow simpler mesh generation. The volume penalization method (an IBM) is commonly used for incompressible and compressible viscous flows but only one application to compressible inviscid flows can be found, which uses the characteristic-based volume penalization (CBVP) method. This approach penalizes the Euler equations to enforce a no-penetration velocity and an adiabatic wall while accounting for wall curvature. A new penalization method based on the CBVP is proposed to impose the conservation of entropy and total enthalpy in the normal direction to the wall instead of the classical adiabatic condition. The two approaches are compared and numerically tested on several cases: weakly compressible flow around a circular cylinder, subsonic flow around a NACA0012 airfoil and flow around a challenging high curvature ice horn. The new method is found to be more accurate than the CBVP on coarser meshes and better at retrieving attached flows for curved geometries. The paper concludes that the proposed method is suitable for general aerospace applications and beneficial for icing simulations which can exhibit highly curved geometries.

4.2 Introduction

Immersed boundary methods (IBM) are an interesting alternative to the usual body-fitted (BF) mesh approach when dealing with complex and moving geometries. With BF meshes, the boundary conditions can be imposed exactly but the mesh generation is more restrictive as it must conform to the geometry. Alternatively when using an IBM, the geometry can arbitrarily cut through the mesh, even allowing the use of Cartesian grids which are simpler to generate and allow the use of fast and efficient algorithms [11]. On the other hand, effort must be spent on the correct imposition of the boundary conditions.

Amongst the variety of IBMs available in the literature (see for instance [11] or [40] for a review), continuous IBMs have the advantage of being independent of the discretization and numerical methods. Furthermore, continuous approaches such as the volume penalization method of [67] are appealing for their simplicity of implementation. This type of approach stems from the Brinkman penalization method [66] where a penalization term is applied as a basic source term in the momentum equation of the incompressible Navier-Stokes equations to account for the presence of a porous medium. The idea was generalized for solid bodies in [67, 76], where the no-slip wall boundary condition is applied by assuming a porous medium with very low permeability. An analysis of the method is also given in [67] along with error estimates for the penalization. The Brinkman penalization or volume penalization was applied for instance in [39] for the tracking of ice shedding trajectories in a Cartesian grid, again using an incompressible laminar viscous flow.

For compressible viscous flow, a penalization term is added to the continuity equation in [73]. The same approach is employed by [69] where a comparison between the penalization and direct-forcing method (another type of IBM) is performed. In [144], the momentum equation is penalized along with the energy equation in order to apply a fixed wall temperature, but the continuity equation is left untouched. A similar approach is followed by [72, 68].

A generalization of the Brinkman penalization method is proposed by [77] where the imposition of Dirichlet, Neumann and Robin boundary conditions is discussed for diffusion and convection-diffusion problems. Another generalization is proposed by [75], the characteristic-based volume penalization (CBVP) method, which provides a systematic way of implementing Dirichlet, Neumann and Robin conditions using source terms and hyperbolic penalization terms.

As shown by the previous literature review, the Brinkman penalization method is commonly used with incompressible or compressible viscous flows and rare applications to compressible inviscid flows can be found. The Brinkman penalization is applied to the prediction of acoustic scattering in [145] using the linearized Euler equations. The slip wall boundary condition is achieved by penalization of the normal velocity component (on the momentum equation). For aerodynamics, only one application is found in the literature where the Euler equations are penalized [79] using the method of [75].

Contrary to the penalization of the Navier-Stokes equations, the penalization of the Euler equations involves only one component of the velocity ($\mathbf{v}\cdot\mathbf{n}$) for the wall boundary condition. Without proper treatment, the continuity of the other variables at the wall is not ensured and can pollute the near-wall solution and thus the wall data extraction. This problem is not observed for Navier-Stokes equation which exhibits its own set of issues related to the

characteristics of the boundary layer (e.g. flow anisotropy, strong gradients).

In this paper, the Characteristic-Based Volume Penalization (CBVP) of [75, 79] is applied to the Euler equations in an alternative way. Our approach uses a different set of boundary conditions inspired by [143] where the conservation of entropy and total enthalpy in the normal direction are applied. This set of boundary conditions also accounts for the wall curvature and allows superior mesh convergence for curved geometries [143]. This type of boundary condition is useful for general applications and was found beneficial for the numerical simulation of ice accretion where ice shapes can exhibit features of high curvature. It is worth mentioning that the Euler equations are still relevant in the icing community where many numerical tools for industrial applications are based on inviscid-viscous coupling (e.g. LEWICE [4], IGLOO2D [12]). The Euler equations can also be used during the preliminary design of aerodynamic shapes (with or without viscous-inviscid interaction). For instance, [13] developed a ghost-cell method (a type of IBM) on Cartesian grids following this goal.

For this paper, the implementation of the penalization method is performed in the Euler flow solver of ONERA's 2D icing suite: IGLOO2D [12]. Although higher benefits are achieved when using IBMs on Cartesian grids in terms of computational effort and mesh generation, unstructured meshes are used in this paper for more flexibility in mixing body-fitted and immersed boundary approaches. It allows the simultaneous use of the two approaches within a simulation and also help in comparing the IBM to the body-fitted approach as the numerical code is against itself. Moreover, the penalization method is independent of the discretization and therefore the developments made in this paper are equally applicable to Cartesian, structured and unstructured meshes. The penalization methods are presented in 2D but their extension to 3D does not present any new difficulty as the evaluation of the curvature in 3D (a key element of the method) has already been treated in the literature [146, 79].

The paper starts with the review of two types of numerical wall boundary conditions for the Euler equations and their application to body-fitted meshes in a finite volume context. This is helpful to understand the boundary conditions that are to be imposed with the penalization method. Then in section 3, the representation of the immersed boundary is discussed. Section 4 is dedicated to the description of the penalization method. More precisely, the penalization method of [75] and its application to the Euler equations [79] is discussed. Then the development of the new penalization method is presented along with implementation details for a Finite Volume Method. In section 5, verification of the new method is made on canonical test cases and on a challenging 2D ice horn case. Some comparisons are also made against the CBVP method.

4.3 Wall Boundary Conditions for the Euler Equations

In this section, wall boundary conditions for the Euler equations are reviewed for motionless body-fitted meshes. The Euler equations are reminded in both conservative and non-conservative forms as both formulations are used later in this paper. Then two types of numerical wall boundary conditions are described for finite volume implementation.

The non-conservative form of the Euler equations is:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \rho \boldsymbol{\nabla} \cdot \mathbf{v} + \mathbf{v} \cdot \boldsymbol{\nabla} \rho = 0 \\ \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \boldsymbol{\nabla} \mathbf{v} + \boldsymbol{\nabla} P = 0 \\ \rho \frac{\partial e}{\partial t} + \rho \mathbf{v} \cdot \boldsymbol{\nabla} e + P \boldsymbol{\nabla} \cdot \mathbf{v} = 0 \end{aligned} \tag{4.1}$$

and its conservative form is written as:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) = 0 \\ \frac{\partial (\rho \mathbf{v})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v} \otimes \mathbf{v} + P \mathbf{I}) = 0 \\ \frac{\partial (\rho E)}{\partial t} + \boldsymbol{\nabla} \cdot ((\rho E + P) \mathbf{v}) = 0 \end{aligned} \tag{4.2}$$

where ρ is the density, $\mathbf{v}$ the velocity, P the pressure, e the internal energy, E the total energy, H is the total enthalpy and $\mathbf{I}$ the identity tensor.

$$e = \frac{R}{\gamma - 1} T \tag{4.3}$$

$$E = e + \frac{1}{2} ||\mathbf{v}||^2 \tag{4.4}$$

$$H = \gamma e + \frac{1}{2} ||\mathbf{v}||^2 \tag{4.5}$$

The system is closed using the ideal gas law.

$$P = \rho R T \tag{4.6}$$

The specific gas constant for air is $R = 287.0$ and the specific heat ratio is $\gamma = 1.4$.

The wall boundary conditions for the Euler equations are set in order to obtain a no-penetration velocity (or slip velocity) where $\mathbf{v} \cdot \mathbf{n} = 0$ (the boundaries are assumed motionless

for this paper). This leads to the following wall flux (in 2D):

$$\mathbf{F}_{wall} = \begin{bmatrix} \rho\mathbf{v}\cdot\mathbf{n} \\ \rho u\mathbf{v}\cdot\mathbf{n} + Pn_x \\ \rho v\mathbf{v}\cdot\mathbf{n} + Pn_y \\ \rho H\mathbf{v}\cdot\mathbf{n} \end{bmatrix} = \begin{bmatrix} 0 \\ Pn_x \\ Pn_y \\ 0 \end{bmatrix} \tag{4.7}$$

On a physics point of view, only the no-penetration velocity is a required boundary condition at the wall. However on a numerical point of view (i.e. finite volume method), adequate values for the variables ρ, u, v, P are required in the ghost cells for both the evaluation of the wall flux and the evaluation of the gradients (MUSCL reconstruction). Two approaches are reviewed here, which are both implemented in the unstructured Euler flow solver of IGLOO2D: the Symmetry Technique (ST) and the Curvature Corrected Symmetry Technique (CCST).

4.3.1 Symmetry Technique (ST)

Considering a Finite Volume cell-centered discretization using ghost cells at the wall boundary (Figure 4.1), the Symmetry Technique consists in imposing the following variables in the ghost cells (g) to obtain the appropriate wall flux:

$$\mathbf{v}_g = \mathbf{v}_d - 2(\mathbf{v}_d\cdot\mathbf{n})\mathbf{n} \tag{4.8}$$

$$\rho_g = \rho_d \tag{4.9}$$

$$P_g = P_d \tag{4.10}$$

where $\mathbf{n}$ is the wall normal. It follows the assumption that the wall is locally flat (negligible curvature). This assumption holds if the mesh in the vicinity of the wall is sufficiently refined (see Eq. (4.12) with $\Delta n = 0$).

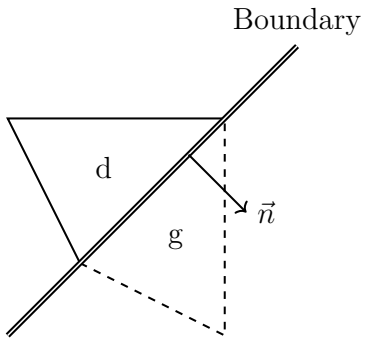


Figure 4.1 Representation of the domain (d) and ghost (g) cells at the boundary along with the wall normal ($\mathbf{n}$)

4.3.2 Curvature Corrected Symmetry Technique (CCST)

The CCST is presented in 2D in [143, 88], then extended to 3D in [146] and applied to 2D unstructured meshes in [147]. This approach imposes the normal momentum relation to account for the wall curvature in the pressure extrapolation, which can be written as:

$$\left.\frac{\partial P}{\partial \mathbf{n}}\right|_w = \rho_w \kappa ||\mathbf{v}_w||^2 \tag{4.11}$$

where κ is the signed wall curvature (positive if the center of curvature is on the domain/fluid side, negative on the ghost/solid side). Note that Eq. (4.11) is derived from the momentum equation and is valid for both steady and unsteady flows. However its application is limited to stationary and non-deformable bodies ($\frac{\partial \mathbf{n}}{\partial t} = 0$). As a consequence of Eq. (4.11) the ghost cell values are computed as:

$$P_g = P_d + \rho_w \kappa ||\mathbf{v}_w||^2 \Delta n \tag{4.12}$$

$$\rho_g = \rho_d \left(\frac{P_g}{P_d}\right)^{1/\gamma} \tag{4.13}$$

$$(\mathbf{v}\cdot\mathbf{n})_g = -(\mathbf{v}\cdot\mathbf{n})_d \tag{4.14}$$

$$(\mathbf{v}\cdot\mathbf{t})_g^2 = (\mathbf{v}\cdot\mathbf{t})_d^2 + \frac{2\gamma}{\gamma-1}\left(\frac{P_d}{\rho_d} - \frac{P_g}{\rho_g}\right) \tag{4.15}$$

where Δn is the distance between the centers of the domain (d) and ghost (g) cells along the normal direction. Eq. (4.13) and Eq. (4.15) result from the conservation of entropy and total

enthalpy respectively. The wall (w) values in Eq. (4.12) are typically taken as [143, 147]:

$$\rho_w = \rho_d \tag{4.16}$$

$$\mathbf{v}_w = \mathbf{v}_d - (\mathbf{v}_d \cdot \mathbf{n})\mathbf{n} \tag{4.17}$$

The norm of the tangential velocity is computed from Eq. (4.15) and its direction follows the unit tangent vector computed as:

$$\mathbf{t} = \frac{\mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}}{||\mathbf{v} - (\mathbf{v} \cdot \mathbf{n})\mathbf{n}||} \tag{4.18}$$

The CCST is shown to generate less numerical entropy and to exhibit faster grid convergence for steady flow computations around a cylinder [143]. However, the ST and CCST boundary conditions converge towards the same solution as the mesh is refined. Furthermore, when the curvature is zero ($\kappa = 0$), the CCST method simplifies to the ST approach.

To assess if a simulation benefited from the CCST, the correction term ($\rho_w \kappa ||\mathbf{v}_w||^2 \Delta n$) in the pressure extrapolation (Eq. (4.12)) can be evaluated. It can be interpreted as the error (ΔP, Eq. (4.19)) committed on the pressure extrapolation when using the ST instead of the CCST, where the characteristic wall cell size (Δx) is used to estimate Δn. Using the relative error Eq. (4.20), one can observe that it depends not only on curvature and cell size, but also on the local wall Mach number (M_w). Thus a scenario with high curvature and a coarse mesh would greatly benefit from the use of the CCST.

$$\Delta P = \rho_w ||\mathbf{v}_w||^2 \kappa \Delta x \tag{4.19}$$

$$\frac{\Delta P}{P_w} = \gamma M_w^2 \kappa \Delta x \tag{4.20}$$

4.4 Immersed Boundary Representation

Before describing the penalization methods for the Euler equations, it is worth discussing how the immersed boundary (IB) is represented.

4.4.1 Signed Distance

For this paper, the immersed boundary (IB) is defined by a discrete list of nodes (2D). The location of the IB is defined by the level-set $\phi = 0$, where ϕ is the signed distance field from the IB. Values of ϕ are computed using a geometric approach : evaluating the minimum projected distance to the edges forming the IB [129]. The signed distance is negative ($\phi < 0$)

in the solid and positive in the fluid ($\phi > 0$).

Using a signed distance field leads to a simple evaluation of the normals ($\mathbf{n}_\phi$) and curvature (κ) of the IB using:

$$\mathbf{n}_\phi = -\frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} \tag{4.21}$$

$$\kappa = \boldsymbol{\nabla} \cdot \mathbf{n}_\phi. \tag{4.22}$$

Note that the normal based on ϕ has a negative sign in order to point towards the solid zone ($\phi < 0$). This is useful in the definition of the penalization methods presented in the next section.

4.4.2 Data Extraction

To extract the data at the IB (e.g. density, velocity, pressure), a weighted least square interpolation at the discrete nodes defining the IB is used. The weight is based on the inverse distance with a threshold on the distance ($0.5\Delta x$) to avoid division by a very small number and provide some smoothing to the data. The interpolation stencil is determined firstly by identifying the cell containing the IB node, and secondly by saving the extended neighborhood of this cell. The penalization methods described in the following sections extend the fluid data into the solid zone (from outside the geometry to its inside). Thus the interpolation stencil in the vicinity of the solid/fluid interface is assumed to be filled with valid data to perform the interpolation.

4.5 Penalization Method

In this section, the volume penalization method [67, 76] is presented along with a discussion on its application to the Euler equations. Then the penalization method of [75] (CBVP) is described followed by the development of the improved penalization approach called CBVP-Hs because it conserves total enthalpy (H) and entropy (s).

4.5.1 Volume Penalization

The volume penalization method consists in adding source terms in the continuous form of the equation to enforce the desired boundary condition. The source terms are activated/deactivated using a mask function (χ) equal to unity in the solid and zero in the fluid. In this way, only the solid is penalized and the usual equations are retrieved in the

fluid. The penalization parameter ($\eta \ll 1$) ensures the boundary condition is enforced accurately. This type of method is mostly used for Dirichlet boundary conditions but a general extension of the method to Neumann and Robin conditions is proposed in [77]. The volume penalization is limited to 1^{st} order accuracy in space [148]. However, a 2^{nd} order adaptation of the method, the sub-mesh penalty method, is proposed by [148].

The volume penalization is widely used for the Navier-Stokes equations where the velocity $\mathbf{v}$ is penalized on the momentum equations to obtain $\mathbf{v} = 0$ in the solid (for a stationary body). Some authors also apply penalization terms to the density and energy equation for compressible flows (e.g. [73],[144]). Other implementations for adiabatic walls do not require the penalization of these equations (e.g. [72]). A simple adaptation of this approach to the Euler equations consists in penalizing only the momentum equations to obtain $\mathbf{v} \cdot \mathbf{n} = 0$ in the solid instead, similar to [145]. In brief, the goal is to obtain a slip velocity in the solid instead of the no-slip condition. In non-conservative form the penalized Euler equations read:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}) &= 0 \\ \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \boldsymbol{\nabla} \mathbf{v} + \boldsymbol{\nabla} P &= -\frac{\chi}{\eta} \rho (\mathbf{v} \cdot \mathbf{n}_\phi) \mathbf{n}_\phi \\ \rho \frac{\partial e}{\partial t} + \rho \mathbf{v} \cdot \boldsymbol{\nabla} e + P \boldsymbol{\nabla} \cdot \mathbf{v} &= 0 \end{aligned} \tag{4.23}$$

In the fluid ($\chi = 0$), the usual Euler equations are retrieved. In the solid the penalization term is activated by $\chi = 1$. As $\eta \ll 1$, the physical terms in the momentum equation are negligible in front of the penalization term. It thus comes back to solving an ordinary differential equation of the form:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\eta} (\mathbf{v} \cdot \mathbf{n}_\phi) \mathbf{n}_\phi \tag{4.24}$$

where the penalization parameter η can be seen as a characteristic timescale. The solution of this ODE is a rapidly decaying exponential (see Eq. (4.25) for the x component), which means the no-penetration velocity is imposed almost instantaneously (Eq. (4.26)).

$$(\mathbf{v} \cdot \mathbf{n}_\phi) n_{\phi,x} = A_0 e^{-(t n_{\phi,x})/\eta} \tag{4.25}$$

$$(\mathbf{v} \cdot \mathbf{n}_\phi) \mathbf{n}_\phi = 0 \tag{4.26}$$

The Euler equations are also solved in the solid but because of Eq. (4.26), only the tangential

component of the velocity remains ($\mathbf{v}_t$):

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{v}_t) &= 0 \\ \rho \frac{\partial \mathbf{v}_t}{\partial t} + \rho \mathbf{v}_t \cdot \boldsymbol{\nabla} \mathbf{v}_t + \boldsymbol{\nabla} P &= 0 \\ \rho \frac{\partial e}{\partial t} + \rho \mathbf{v}_t \cdot \boldsymbol{\nabla} e + P \boldsymbol{\nabla} \cdot \mathbf{v}_t &= 0 \end{aligned} \tag{4.27}$$

In [79], more complex penalized Euler equations are briefly suggested but no justification for their selection is provided. Numerical experiments showed that with a simpler form like Eq. (4.23), the velocity tends to zero inside the solid and an artificial boundary layer is created near the immersed boundary. This behavior was unexpected as, contrary to the Navier-Stokes equations where the fluid is at rest in the solid (no-slip wall), Eq. (4.27) shows that there should be a tangential fluid flow in the solid with the Euler equations (slip wall). Because the no-penetration velocity is imposed at the IB and because there is no diffusion term, there is a lack of communication between the fluid and the solid. One constraint only is imposed: the IB is a streamline because it is parallel to the flow. But some discontinuities in tangential velocity ($\mathbf{v}_t$), pressure, entropy and total enthalpy are allowed across the IB, which is thus a slip line. The objective is then to retrieve continuity across the streamline at the IB by the imposition of additional constraints in the normal direction. This can be achieved by enforcing Neumann boundary conditions, which can be implemented using the penalization method of [75], presented in the following section.

4.5.2 Characteristic-Based Volume Penalization (CBVP)

The Characteristic-based Volume Penalization (CBVP) method [75] provides a systematic way of implementing Dirichlet, Neumann and Robin boundary conditions by the addition of hyperbolic penalization terms. It uses a sharp Heaviside function for the mask (χ) where $\chi = 1$ in the solid (Ω_s) and $\chi = 0$ in the fluid (Ω_f). This leads to a staircase definition of the immersed boundary (IB) as shown in Figure 4.2. However, a smooth solution is recovered at the IB by the use of hyperbolic penalization terms for Neumann and Robin conditions, and by the use of dissipation terms for Dirichlet conditions.

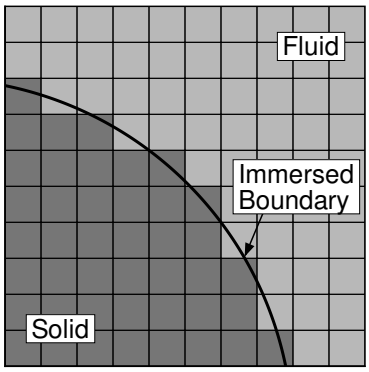


Figure 4.2 Staircase representation of the solid

In [79], a set of penalized Euler equations are presented very briefly. In non-conservative form, it writes:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\rho - \kappa\frac{\rho^2}{P}||\mathbf{v}||^2\right) \\
\rho\frac{\partial \mathbf{v}}{\partial t} + (1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v} + \boldsymbol{\nabla}P) &= -\frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi + \chi\rho\nu_\eta\boldsymbol{\nabla}^2\mathbf{v} \\
\rho\frac{\partial e}{\partial t} + (1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}e + P\boldsymbol{\nabla}\cdot\mathbf{v}) &= -\frac{\chi}{\eta_c}(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}e)
\end{aligned} \tag{4.28}$$

where $\mathbf{n}_\phi$ is the normal to the IB pointing towards the solid. Note that $1/\eta_c$ can be seen as a characteristic velocity with $\eta_c \ll 1$. Here, hyperbolic penalization terms of the form $(\mathbf{n}\cdot\boldsymbol{\nabla}u - q)/\eta_c = 0$ are added to the continuity and energy equations to apply Neumann boundary conditions. As explained in [75], for the CBVP method the physical flux terms are removed from Ω_s to prevent any interaction with the penalization terms. For Dirichlet conditions, an artificial dissipation term is added using the artificial viscosity ν_η which must be of the order $\nu_\eta \geq \Delta x^2/\eta$. According to [75], this term helps in retrieving a smooth solution at the IB for Dirichlet conditions.

In the fluid ($\chi = 0$), the usual Euler equations are retrieved. In the solid ($\chi = 1$), by setting the time derivative to zero (steady state), the following conditions are enforced:

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\rho = \kappa\frac{\rho^2}{P}||\mathbf{v}||^2 \tag{4.29}$$

$$-\frac{1}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi + \nu_\eta\boldsymbol{\nabla}^2\mathbf{v} = 0 \tag{4.30}$$

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}e = 0 \tag{4.31}$$

This set of equations is equivalent to imposing an adiabatic wall (Eq. (4.31)), a no penetration velocity (Eq. (4.30)) and the normal momentum relation which relates the pressure gradient to the wall curvature (Eq. (4.29)), except it is written in terms of the density gradient. The normal momentum relation can be written as:

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} P = \kappa \rho ||\mathbf{v}||^2 \tag{4.32}$$

where in 2D, the curvature (κ) is computed from the normals to the IB ($\mathbf{n}_\phi$) as $\kappa = \boldsymbol{\nabla} \cdot \mathbf{n}_\phi$. To obtain Eq. (4.29), one must use the ideal gas law ($P = \rho e(\gamma - 1)$) in combination with the normal momentum relation (Eq. (4.32)) and apply the adiabatic wall condition ($\mathbf{n}_\phi \cdot \boldsymbol{\nabla} e = 0$).

The dissipation term in Eq. (4.30) helps in obtaining continuity of the tangential velocity but also hinders the imposition of $(\mathbf{v} \cdot \mathbf{n})\mathbf{n} = 0$. Thus this parameter must be selected small enough for the no-penetration velocity to be enforced, but large enough to ensure continuity and stability. In this paper the dissipation parameter is taken as $\nu_\eta = \Delta x^2/\eta$ as suggested in [75]. Note that this penalization method depends on three adjustable parameters: η, η_c and ν_η instead of one for the volume penalization method.

4.5.3 CBVP-Hs

Because of the dissipation term in the CBVP method, there is a smooth transition of the velocity from the fluid to the solid. However, this transition is not based on the physics of the problem and does not ensure the conservation of total enthalpy. A new penalization of the Euler equations is proposed hereafter, where the CBVP method of [75] is used to apply the wall boundary conditions of the CCST [143, 88, 146].

The goal is to develop a method that would respect the properties of the inviscid ideal gas flow in the vicinity of the wall streamline. The streamline is, of course, defined by the no-penetration velocity (Eq. (4.36)). The continuity of tangential velocity, pressure, entropy and total enthalpy must be ensured by additional equations. The normal momentum relation (Eq. (4.32)) allows the continuity of pressure. The normal conservation of total enthalpy (Eq. (4.33)) and the normal conservation of entropy (Eq. (4.34)) are imposed to close the system:

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} H = 0 \tag{4.33}$$

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} s = 0 \tag{4.34}$$

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} P = \kappa \rho ||\mathbf{v}||^2 \tag{4.35}$$

$$(\mathbf{v} \cdot \mathbf{n}_\phi)\mathbf{n}_\phi = 0 \tag{4.36}$$

where $\mathbf{n}_\phi$ is the normal to the IB based on the signed distance field (ϕ). This set of boundary conditions allows the continuity of the flow variables including the norm of the velocity across the boundary. It is better suited for homentropic and homenthalpic flows which are the primary target of our model, although the presence of a shock wave for instance may not hinder the use of (Eq. (4.34)) as long as the shock is parallel to $\mathbf{n}_\phi$.

Because this new method is based on the CBVP but designed to conserve the entropy (s) and total enthalpy (H), it is referred to as CBVP-Hs. It is important to understand that the only physical wall boundary condition required for the Euler equations is the no-penetration velocity. The other conditions are numerical and are used to improve the accuracy of the model in terms of wall pressure extrapolation, generation of entropy and conservation of total enthalpy. The boundary conditions also improve the continuity of the solution near the IB and provide adequate support for the evaluation of the fluxes and gradients (e.g. for a 2nd order MUSCL approach). Furthermore, it allows for the use of an interpolation method for the extraction of the wall data.

In the following sections, penalization terms are first derived for the primitive variables and applied to the non-conservative form of the Euler equations. It is useful to start with the primitive variables as the penalization terms can easily be derived for them. Then the transition from the non-conservative to the conservative form naturally provides the penalized Euler equations in terms of conservative variables.

4.5.3.1 Penalization of the Primitive Variables

In this section, penalization terms are derived for the non-conservative form of the Euler equations (Eq. (4.1)). Hyperbolic penalization terms of the form $\mathbf{n} \cdot \boldsymbol{\nabla} u = q$ are sought, where u is ρ,$\mathbf{v}$ or e.

The penalization term for the density is derived from Eqs. (4.34)–(4.32). The conservation of entropy can be written as:

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} s = \mathbf{n}_\phi \cdot \boldsymbol{\nabla} \left(\frac{P}{\rho^\gamma} \right) = 0 \tag{4.37}$$

$$\Rightarrow \mathbf{n}_\phi \cdot \boldsymbol{\nabla} \rho = \frac{\rho}{\gamma P} \mathbf{n}_\phi \cdot \boldsymbol{\nabla} P \tag{4.38}$$

By substituting the normal momentum relation (Eq. (4.32)) in Eq. (4.38), a relation for the normal density gradient is obtained. It is applied to the continuity equation as a hyperbolic

penalization term (Eq. (4.39)).

$$\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi \cdot \boldsymbol{\nabla}\rho - \frac{\rho^2}{\gamma P}\kappa||\mathbf{v}||^2\right) = 0 \tag{4.39}$$

A penalization term for the internal energy can be derived from equations (4.6), (4.3) and (4.38), which gives:

$$\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi \cdot \boldsymbol{\nabla}e - \frac{1}{\gamma}\kappa||\mathbf{v}||^2\right) = 0 \tag{4.40}$$

At this point, only the penalization term on velocity is missing. This term is the application of the no-penetration velocity (Eq. (4.36)) as a Dirichlet condition and a hyperbolic penalization term denoted $\mathcal{P}_\mathbf{v}$ which must be determined to set the velocity magnitude.

$$\begin{aligned}\rho\frac{\partial \mathbf{v}}{\partial t} + (1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v} + \boldsymbol{\nabla}P) = \\ \mathcal{P}_\mathbf{v} - \frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi\end{aligned} \tag{4.41}$$

In addition, the conservation of total enthalpy must still be enforced. The term $\mathcal{P}_\mathbf{v}$ can thus be derived in two steps. First, by using the normal conservation of total enthalpy (Eq. (4.33)) one can derive a condition on kinetic energy. The conservation of total enthalpy ($\mathbf{n}_\phi\cdot\boldsymbol{\nabla}H = 0$) can be written as:

$$\frac{\gamma}{\gamma-1}\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{P}{\rho}\right) + \mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{1}{2}||\mathbf{v}||^2\right) = 0 \tag{4.42}$$

with

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{P}{\rho}\right) = \frac{\mathbf{n}_\phi}{\rho}\cdot\boldsymbol{\nabla}P - \frac{P\mathbf{n}_\phi}{\rho^2}\cdot\boldsymbol{\nabla}\rho \tag{4.43}$$

By using Eq. (4.43), the normal momentum relation (Eq. (4.32)) and the condition on density (Eq. (4.39)), the condition on kinetic energy is retrieved.

$$\Rightarrow \mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{1}{2}||\mathbf{v}||^2\right) + \kappa||\mathbf{v}||^2 = 0 \tag{4.44}$$

It can also be recast in a penalized equation for the kinetic energy in the solid:

$$\rho\frac{\partial\left(\frac{1}{2}||\mathbf{v}||^2\right)}{\partial t} = -\frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{1}{2}||\mathbf{v}||^2\right)+\rho\kappa||\mathbf{v}||^2\right) \tag{4.45}$$

Second, a relation for kinetic energy can also be computed from the momentum equation (4.41) scalar product with $\mathbf{v}$ as:

$$\begin{aligned}\mathbf{v}\cdot\rho\frac{\partial\mathbf{v}}{\partial t}+\mathbf{v}\cdot(1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v}+\boldsymbol{\nabla}P) = \\ \mathbf{v}\cdot\mathcal{P}_\mathbf{v}-\mathbf{v}\cdot\frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi\end{aligned} \tag{4.46}$$

By considering this equation in the solid ($\chi = 1$) and by keeping only terms of the same order of magnitude ($\eta \ll 1$ and $\eta \ll \eta_c$), it reads:

$$\mathbf{v}\cdot\rho\frac{\partial\mathbf{v}}{\partial t} = \mathbf{v}\cdot\mathcal{P}_\mathbf{v} \tag{4.47}$$

or

$$\rho\frac{\partial\left(\frac{1}{2}||\mathbf{v}||^2\right)}{\partial t} = \mathbf{v}\cdot\mathcal{P}_\mathbf{v} \tag{4.48}$$

This simplification follows the assumption that η and η_c are not of the same order of magnitude and thus, that $\mathbf{v}\cdot\mathbf{n}_\phi$ exponentially tends towards zero. In order to equate Eq. (4.48) and Eq. (4.45), $\mathcal{P}_\mathbf{v}$ must be selected to respect a condition on kinetic energy (Eq. (4.49)).

$$\mathbf{v}\cdot\mathcal{P}_\mathbf{v} = -\frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\left(\frac{1}{2}||\mathbf{v}||^2\right)+\rho\kappa||\mathbf{v}||^2\right) \tag{4.49}$$

A natural approach is to select $\mathcal{P}_\mathbf{v}$ orthogonal to $\mathbf{n}$ in order to decouple the hyperbolic penalization term and the Dirichlet condition (see Eq. (4.41)). However, no practical formulation was found for use with the Finite Volume Method. An alternative and simpler choice for $\mathcal{P}_\mathbf{v}$ is:

$$\mathcal{P}_\mathbf{v} = -\frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\mathbf{v}+\kappa\rho\mathbf{v}\right) \tag{4.50}$$

However note that Eq. (4.50) is not the only possible choice for $\mathcal{P}_\mathbf{v}$. In Eq. (4.41), the hyperbolic penalization term (Eq. (4.50)) and the Dirichlet condition (Eq. (4.36)) are decoupled by using $\eta \ll \eta_c$.

The non-conservative form of the penalized equations can now be updated using the hyper-

bolic terms for ρ, $\mathbf{v}$ and e:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\rho - \kappa\frac{\rho^2}{\gamma P}||\mathbf{v}||^2\right)\\
\rho\frac{\partial \mathbf{v}}{\partial t} + (1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v} + \boldsymbol{\nabla}P) &= -\frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\mathbf{v} + \kappa\rho\mathbf{v}\right)\\
&\quad - \frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi\\
\rho\frac{\partial e}{\partial t} + (1-\chi)(\rho\mathbf{v}\cdot\boldsymbol{\nabla}e + P\boldsymbol{\nabla}\cdot\mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}e - \kappa\frac{\rho}{\gamma}||\mathbf{v}||^2\right)
\end{aligned} \tag{4.51}$$

4.5.3.2 Penalization of the Conservative Variables

By transferring to the conservative form, the equations including the penalization terms become:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\rho - \kappa\frac{\rho^2}{\gamma P}||\mathbf{v}||^2\right)\\
\frac{\partial \rho\mathbf{v}}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot(\rho\mathbf{v}\otimes\mathbf{v} + P\mathbf{I}) &= -\frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi\\
&\quad - \frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}(\rho\mathbf{v}) + \kappa\rho\mathbf{v}\left(1 - \frac{\rho}{\gamma P}||\mathbf{v}||^2\right)\right)\\
\frac{\partial \rho E}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot((\rho E + P)\mathbf{v}) &= -\frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2\\
&\quad - \frac{\chi}{\eta_c}\left(\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}E + \frac{(\gamma-1)}{\gamma}\rho\kappa||\mathbf{v}||^2\right)
\end{aligned} \tag{4.52}$$

The energy equation can also be written in terms of total enthalpy by expanding the term containing the curvature and using Eq. (4.32) and Eq. (4.39):

$$\frac{(\gamma-1)}{\gamma}\rho\kappa||\mathbf{v}||^2 = \rho n_\phi\cdot\boldsymbol{\nabla}\left(\frac{P}{\rho}\right) \tag{4.53}$$

By substitution of Eq. (4.53) in the energy equation Eq. (4.52), it gives:

$$\begin{aligned}
\frac{\partial \rho E}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot((\rho E + P)\mathbf{v}) &= -\frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2\\
&\quad - \frac{\chi}{\eta_c}\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}H
\end{aligned} \tag{4.54}$$

Using this set of penalization terms ensure Eqs. (4.33)–(4.36) are respected. Furthermore, the term $\mathcal{P}_\mathbf{v}$ replace the dissipation term employed in [75] on the momentum equation. This

convection term is derived from physical arguments in order to respect the conservation of total enthalpy instead of using a numerical artifice. Ultimately, this should translate into a reduction in entropy generation, improved conservation of enthalpy and reduced flow separation as shown by [143, 88, 146] with the CCST.

4.5.3.3 Implementation Details

The penalized Euler equations of Eq. (4.52) are implemented in a cell-centered Finite Volume framework using unstructured meshes. They are written in vector form as:

$$\frac{\partial \mathbf{W}}{\partial t} + (1-\chi)\boldsymbol{\nabla}\cdot\mathbf{F}_{Euler} = -\chi\boldsymbol{\nabla}\cdot\mathbf{F}_{ibm} + \chi\mathbf{S}_{ibm} \tag{4.55}$$

where the penalization terms are conveniently split into flux ($\mathbf{F}_{ibm}$) and source ($\mathbf{S}_{ibm}$) terms.

$$\mathbf{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{F}_{Euler} = \begin{bmatrix} \rho\mathbf{v} \\ \rho u\mathbf{v} + P_x \\ \rho v\mathbf{v} + P_y \\ \rho H\mathbf{v} \end{bmatrix} \tag{4.56}$$

$$\mathbf{W}_{ibm} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho H \end{bmatrix}, \quad \mathbf{F}_{ibm} = \frac{1}{\eta_c}\begin{bmatrix} \rho\mathbf{n}_\phi \\ \rho u\mathbf{n}_\phi \\ \rho v\mathbf{n}_\phi \\ \rho H\mathbf{n}_\phi \end{bmatrix} \tag{4.57}$$

The IB source term ($\mathbf{S}_{ibm}$) is split in Dirichlet, convective and curvature contributions:

$$\mathbf{S}_{ibm} = \mathbf{S}_{ibm,D} + \mathbf{S}_{ibm,conv} + \mathbf{S}_{ibm,curv} \tag{4.58}$$

$$\mathbf{S}_{ibm,D} = \frac{1}{\eta}\begin{bmatrix} 0 \\ -\rho(\mathbf{v}\cdot\mathbf{n}_\phi)n_{\phi,x} \\ -\rho(\mathbf{v}\cdot\mathbf{n}_\phi)n_{\phi,y} \\ -\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2 \end{bmatrix} \tag{4.59}$$

$$\mathbf{S}_{ibm,conv} = \frac{1}{\eta_c}\begin{bmatrix} \rho\boldsymbol{\nabla}\cdot\mathbf{n}_\phi \\ \rho u\boldsymbol{\nabla}\cdot\mathbf{n}_\phi \\ \rho v\boldsymbol{\nabla}\cdot\mathbf{n}_\phi \\ \rho H\boldsymbol{\nabla}\cdot\mathbf{n}_\phi \end{bmatrix} \tag{4.60}$$

$$\mathbf{S}_{ibm,curv} = \frac{1}{\eta_c}\kappa\begin{bmatrix} \rho\frac{\rho}{\gamma P}||\mathbf{v}||^2 \\ \rho u\left(\frac{\rho}{\gamma P}||\mathbf{v}||^2 - 1\right) \\ \rho v\left(\frac{\rho}{\gamma P}||\mathbf{v}||^2 - 1\right) \\ \rho H\frac{\rho}{\gamma P}||\mathbf{v}||^2 \end{bmatrix} \tag{4.61}$$

where the convective source term $\mathbf{S}_{ibm,conv}$ comes from rewriting Eq. (4.52) to isolate $\mathbf{F}_{ibm}$. From the IB flux (Eq. (4.57)), the system is shown to be hyperbolic in the solid with eigenvalues $\mathbf{n}_\phi, \mathbf{n}_\phi, \mathbf{n}_\phi$ and $\gamma\mathbf{n}_\phi$.

The penalization of the Euler equations leads to a stiff system as the penalization parameters (η and η_c) are very small. One way to alleviate this problem is to solve the system implicitly. In this paper it is solved using a BICGSTAB algorithm with a block Jacobi preconditioner. The Euler fluxes are computed using a Roe scheme [149] (no entropy fix) and a MUSCL reconstruction to achieve 2^{nd} order accuracy in space [150]. The IB fluxes are evaluated with a simple 1^{st} order upwind scheme (upwinded by $\mathbf{n}_\phi$) of the form:

$$\begin{aligned}\mathbf{F}_{edge} = \frac{1}{2}\big[&(\mathbf{n}_\phi\cdot\mathbf{n}_{edge})(\mathbf{W}_{ibm,R} + \mathbf{W}_{ibm,L}) \\ &- |\mathbf{n}_\phi\cdot\mathbf{n}_{edge}|\,(\mathbf{W}_{ibm,R} - \mathbf{W}_{ibm,L})\big]\end{aligned} \tag{4.62}$$

where L and R represent the left and right state respectively and $\mathbf{n}_{edge}$ is the normal vector to the cell edge.

The mask function χ is the Heaviside function ($\mathcal{H}$) based on the signed distance field ϕ.

$$\chi = \mathcal{H}(-\phi) \tag{4.63}$$

The level-set $\phi = 0$ determines the location of the IB with $\phi < 0$ in the solid and $\phi > 0$ in the fluid. It can be seen that in the fluid ($\chi = 0$), the usual Euler equations are retrieved while in the solid ($\chi = 1$) only the penalization terms are activated.

From [75], $\eta_c \ll 1$ and $\eta \ll 1$ in order to enforce the correct condition in the solid by penalization. In the current implementation, a slightly different approach is followed. Since the Euler fluxes are deactivated in the solid, η_c is not required to be very small in order to propagate the fluid properties towards the solid. Here, the IB fluxes are not penalized but are merely a replacement for the Euler fluxes inside the solid by using $\eta_c = 1$. It comes back to solving propagation equations in the solid, which are penalized to apply the no-penetration velocity. By using $\eta_c = 1$, the equations in the solid are:

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \mathbf{n}_\phi \cdot \boldsymbol{\nabla}\rho = \kappa \frac{\rho^2}{\gamma P}||\mathbf{v}||^2 \\
&\frac{\partial \rho \mathbf{v}}{\partial t} + \mathbf{n}_\phi \cdot \boldsymbol{\nabla}(\rho\mathbf{v}) = \kappa\rho\mathbf{v}\left(\frac{\rho}{\gamma P}||\mathbf{v}||^2 - 1\right) - \frac{1}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi \\
&\frac{\partial \rho E}{\partial t} + \rho\mathbf{n}_\phi \cdot \boldsymbol{\nabla}H = -\frac{1}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2
\end{aligned} \tag{4.64}$$

where the only penalization terms are $\frac{1}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi$ and $\frac{1}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2$. A typical value for the penalization parameter is $\eta = 10^{-10}$, which accurately enforces the slip velocity ($\mathbf{v}\cdot\mathbf{n}_\phi = 0$) near the IB. As for the volume penalization method, $\mathbf{v}\cdot\mathbf{n}_\phi$ exponentially tends towards zero (almost instantaneously). This approach is suitable if the Euler equations are solved for a steady state.

In this paper $\eta_c = 1$ is used, but it is also possible to use $\eta_c < 1$. In such a case, it is important to respect a ratio η/η_c as there is an interaction between the Neumann and Dirichlet conditions in the momentum equation. The idea is to keep the imposition of $\mathbf{v}\cdot\mathbf{n} = 0$ dominant over the propagation of the information. To do so $\eta \ll \eta_c$ and a ratio of $\eta/\eta_c = 10^{-6}$ was found to be sufficient for most applications.

With the CBVP-Hs method, two adjustable parameters are used: η and η_c. Thus, it has fewer parameters to calibrate compared to the CBVP method which has three of them: η, η_c and ν_η.

4.5.3.4 Dimension of the Problem

The approach was derived and assessed in 2D in the present article, without losing generality. Since the penalization method is based on the continuous form of the equations and is independent of the discretization, the CBVP-Hs can naturally be extended to 3D. In practice, some difficulties might arise when evaluating the curvature term (κ), a key element of the method. In 2D, the curvature is purely geometric and therefore simple to estimate. For 3D applications, the curvature also depends on the direction of the wall streamlines and is

therefore linked to the flow velocity at the wall. This additional difficulty is covered for instance in [79] and [146], where the CBVP method and a 3D implementation of the ghost cell method are respectively discussed. Thus a 3D implementation of the CBVP-Hs method should be straightforward by building on previous work from the literature.

4.5.4 Notes on Moving Boundaries

Although moving boundaries are not considered in this paper, the CBVP-Hs method is still directly applicable where boundary displacement can be decoupled from the aerodynamic flow. For instance, in typical numerical tools for the prediction of in-flight icing (e.g. LEWICE [4], IGLOO2D [12]), the aerodynamic flow is computed to steady state and then the ice shape is updated according to the ice growth in a segregated step. With the current penalization method, the geometry update can be accounted for in the aerodynamics solver by re-evaluating the signed distance field (ϕ) and associated metrics ($\mathbf{n}_\phi$, κ, χ). By doing so, the penalization method automatically applies the boundary conditions on the new geometry. Note that the CBVP-Hs method is not limited to icing applications. A similar process could be applied to perform shape optimization where a steady or unsteady flow is computed on a fixed geometry. The shape update would again be accounted for by re-evaluating the signed distance field.

4.5.5 Geometry Fidelity

The volume penalization method is limited to 1^{st} order accuracy because the boundary condition is applied at the cell centers close to the boundary and not on the immersed boundary itself. Some extensions to second order accuracy are available in the literature [71, 70] by using information at the discrete level. Here the implementation is limited to the classical first order accuracy. With this type of approach, the accuracy of the method is usually improved by refining the mesh in the vicinity of the IB.

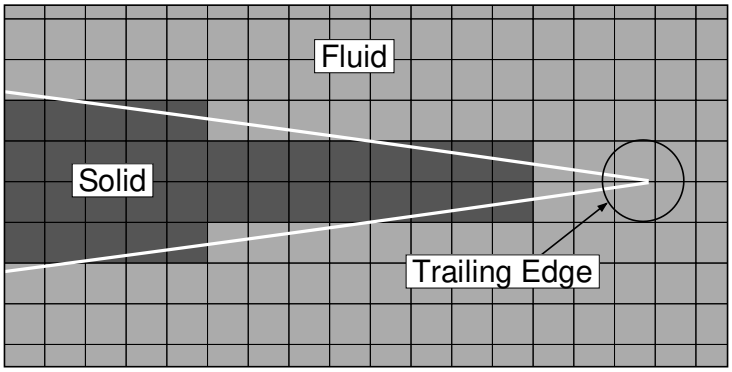


Figure 4.3 Trailing edge representation for a NACA0012 airfoil (white line) on a Cartesian grid

In the current implementation, even if the sharp Heaviside function leads to a staircase representation of the IB, the Neumann boundary conditions (convective terms) are smoothing the interface and improving the solution on coarser meshes (e.g. when extracting the pressure coefficients). However, because of the staircase representation of the IB and because the cell centers are used to determine if a cell is fluid or solid, the current implementation is not well suited to deal with sharp features (e.g. sharp trailing edge of an airfoil). As shown on Figure 4.3, the sharp trailing edge is seen as blunt by the penalization method as the fluid cells covering the trailing edge are not penalized. A refinement in the vicinity of the sharp feature can improve the solution by providing a more accurate representation of the geometry. In this paper, the sharp trailing edge is simply treated as blunt. In this way, the cell refinement can be coarser while allowing the correct overall solution to be retrieved. This approach is used for the NACA0012 airfoil presented in the next section.

For icing applications [151], the selected approach is to deal with sharp features on the clean geometry using a body-fitted mesh (e.g. sharp trailing edge on an airfoil). Only the ice will be immersed in the mesh (see Figure 4.18b). Although some sharp features may be generated by the ice growth, it is acceptable for them to be slightly smoothed out. With this approach, the accuracy of the body-fitted approach is retrieved where possible and the flexibility of the IBM is used otherwise. The ice accretion case presented in this paper follows this approach.

4.5.6 Mesh Particularities

For unstructured meshes made of triangles, some cell configurations lead to a poor propagation of the information from the fluid to the solid. This situation occurs when a fluid cell near the IB is trapped between solid cells and the numerical fluxes allow no communication with neighboring fluid cells. One example is illustrated in Figure 4.4, where cell A is fluid

and cell B is solid. Cell A has 2 solid cells and one fluid cell as direct face neighbors. In this specific configuration, cell A is emptied and the solution is propagated in the solid (e.g. cell B).

In this paper, the issue is solved by applying a correction on the mask function (χ) for these pathological cells: instead of using a sharp definition ($\chi = 0$ or 1), χ is set to the solid fraction of the cell ($\chi \in [0, 1]$). This correction blends the Euler and the penalization fluxes leading to a smoother solution as the communication with the fluid is restored.

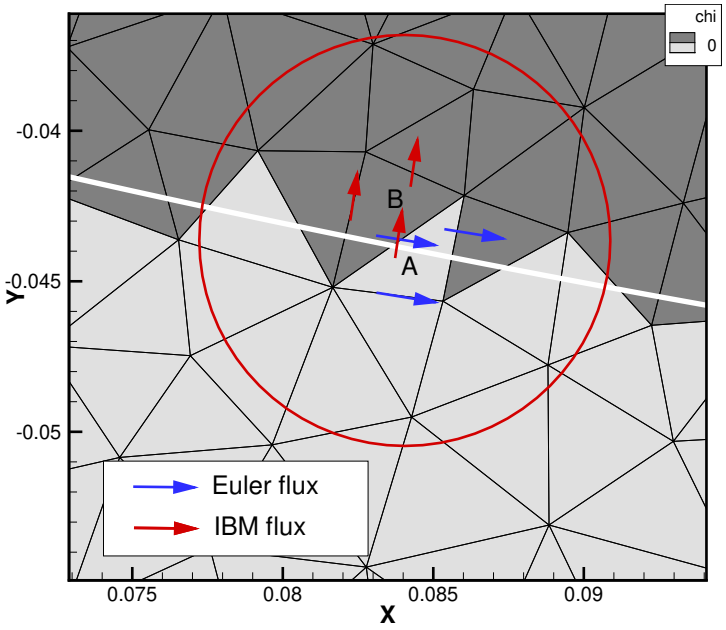


Figure 4.4 Pathological cell example with flux representation and mask function χ

From numerical experiments, the issue was observed for only a few cells (1% to 2% of the IB cells) and not for all meshes. Some unstructured meshes will present no pathological cells while for structured meshes made of quadrilateral cells, the issue was never observed. Also, note that the issue is present for both the CBVP and CBVP-Hs methods.

4.5.7 Brief Comparison with other type of IBMs

For completion, please note that other type of IBMs have been successfully applied to the Euler equations, offering their own set of advantages and drawbacks. For discrete approaches, the desired boundary conditions are imposed by enforcing them at the discrete level (e.g. [88, 146], [13]). Thus they are dependent on the selected discretization method and also on the type of mesh used. These approaches have the advantage of being accurate (2^{nd} order, e.g. [87]) and can also deal with sharp trailing edges in a simple manner (e.g. [88, 13]).

Embedded "cut-cell" methods (e.g. [41, 42]) are also suitable for solving the Euler equations with similar advantages compared to discrete methods. This paper inserts itself in the scope of penalization methods and suggests a suitable approach for the penalization of the Euler equations. Although the current CBVP-Hs method is only 1^{st} order accurate (globally), the advantage of penalization methods lies in their independence from the discretization method or the type of mesh used as they are based on the continuous form of the equations. They are also independent of the physical dimension of the problem and simple to implement as they require minimal information from the geometry. A simple evaluation of the signed distance field (ϕ) and mask function (χ) is sufficient to determine the location of the immersed boundary and activate the penalization terms in the solid.

4.6 Results

In this section, the behavior of the CBVP-Hs method is shown on different test cases. A mesh refinement study is first performed for the weakly compressible flow around a circular cylinder. Then the subsonic flow around a NACA0012 airfoil is studied. The penalization method is also tested on an ice horn which was found difficult to solve in a previous communication [151] due to its high curvature. The simulation parameters are summarized in Table 4.1.

The solution from the penalization method is verified against analytical solutions when applicable or against numerical solutions obtained by a body-fitted approach. A comparison is also performed against the CBVP method. Wall pressure coefficients, entropy and total enthalpy errors are compared and discussed.

Table 4.1 Simulation Parameters

	Cylinder	**Airfoil**	**Ice Horn**
Geometry	Cylinder	NACA0012	GLC305
Chord	D=2.0	1.0	0.9144
AoA	0.0	1.25	4.0
Mach	0.1	0.5	0.273
$\mathbf{P_{static}}$	100kPa	100kPa	101.325 kPa
$\mathbf{T_{static}}$	300.0K	300.0K	268.3K
LWC	–	–	$0.54g/m^3$
MVD	–	–	$20\mu m$
Icing Time	–	–	$1350s$

In Table 4.1, LWC stands for Liquid Water Content and represents the mass of water per

volume of air. The median volume diameter (MVD) represents the droplet size for the icing simulation. The list of icing parameters are provided but note that only the aerodynamics is treated in this paper.

For the following simulations, the convergence is determined by monitoring the L_2 and L_∞ norms of the wall pressure coefficient (Cp). The usual density, momentum and energy residuals are also monitored. In the following sections, when the convergence threshold is specified, the criterion on Cp is used. For the penalization method, the convergence check requires a pressure interpolation at the IB at each time step. It is performed by a weighted least square interpolation method.

4.6.1 Weakly Compressible Flow Around a Cylinder

The first test case is the weakly compressible flow around a circular cylinder. As the Euler flow solver in IGLOO2D is compressible only, the incompressible analytical solution is approached by performing the simulation at a low Mach number (Mach= 0.1). The analytical solution for the pressure coefficient (Cp) is, in 2D Cartesian coordinates:

$$Cp_{analytical} = \frac{2R_c^2(x^2 - y^2) - R_c^4}{(x^2 + y^2)^2} \tag{4.65}$$

where R_c is the radius of the cylinder.

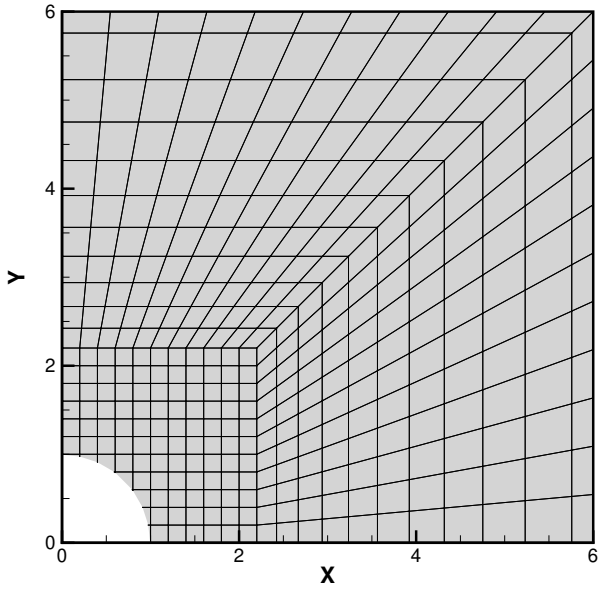


Figure 4.5 Mesh for the cylinder with cell size $D/\Delta x = 10$

A family of five meshes is generated ranging from $D/\Delta x = 10$ to $D/\Delta x = 160$, where D is the diameter of the cylinder and Δx the Cartesian cell size. The cylinder is immersed into a uniform Cartesian grid. To reduce the total number of cells the uniform Cartesian grid is limited to a width of 2 diameters. Outside this zone a structured mesh is used allowing cell growth and stretching, but still ensuring mesh symmetry. The far field is located at 50 diameters from the cylinder. Part of the mesh for $D/\Delta x = 10$ is illustrated in Figure 4.5 with a zoom on the blanked cylinder.

For the results presented below, the wall Cp residual is converged to 10^{-8}. Figures 4.6 and 4.7 illustrate the Cp distribution for both the CBVP and CBVP-Hs methods. From these figures, both methods converge towards the analytical solution. However, the mesh convergence is faster for the CBVP-Hs method for which a good global Cp distribution is achieved on a mesh as coarse as $D/\Delta x = 40$. In comparison, the CBVP method exhibits a satisfactory global solution for the finest mesh only ($D/\Delta x = 160$).

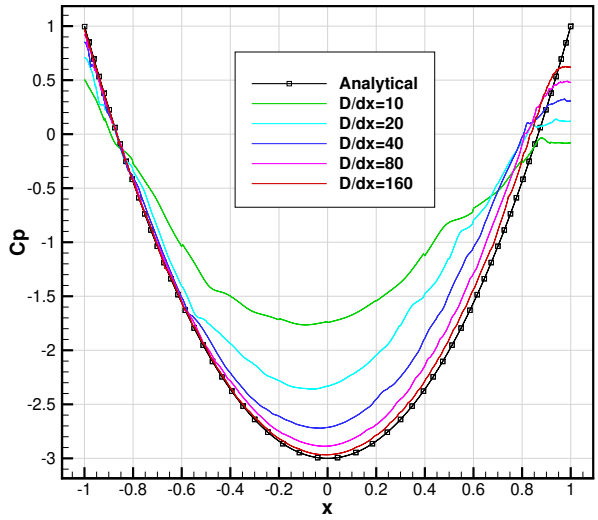


Figure 4.6 Wall Cp with mesh refinement for the cylinder (CBVP)

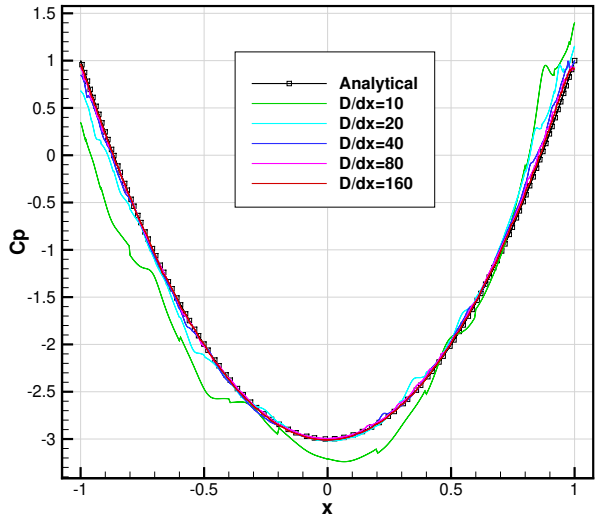


Figure 4.7 Wall Cp with mesh refinement for the cylinder (CBVP-Hs)

For a more detailed analysis, pressure coefficients are extracted at locations of interest: $(x, y) = (-1, 0)$, $(x, y) = (0, 1)$ and $(x, y) = (1, 0)$. For these locations the expected analytical solution is respectively: $Cp = 1$, $Cp = -3$ and $Cp = 1$. The data is shown on Figure 4.8 as the Cp error (ΔCp) against the analytical solution for the three locations:

$$|\Delta Cp| = |Cp - Cp_{analytical}| \tag{4.66}$$

The observation of Figure 4.8 shows that as the mesh is refined the error for the forward stagnation point ($x = -1$) is roughly the same for both methods (CBVP and CBVP-Hs). However, the error is much lower at the maximum suction peak ($x = 0$) and at the aft stagnation point ($x = 1$) for the CBVP-Hs method. For instance, on a mesh as coarse as $D/\Delta x = 20$ the suction peak is captured to an accuracy of $\Delta Cp \approx 10^{-4}$ with the CBVP-Hs compared to $\Delta Cp \approx 10^{-0.5}$ for the CBVP method. The reattachment point at $x = 1$ is also captured more accurately with $\Delta Cp \approx 10^{-2.5}$ for the CBVP-Hs against $\Delta Cp \approx 10^{-1}$ on the finest mesh ($D/\Delta x = 160$). For the forward stagnation point, the order of convergence is observed to be 1 for both methods. It is, however, difficult to conclude on the order of convergence for the other locations. The order of convergence will be discussed based on the root mean squared (RMS) in the next section.

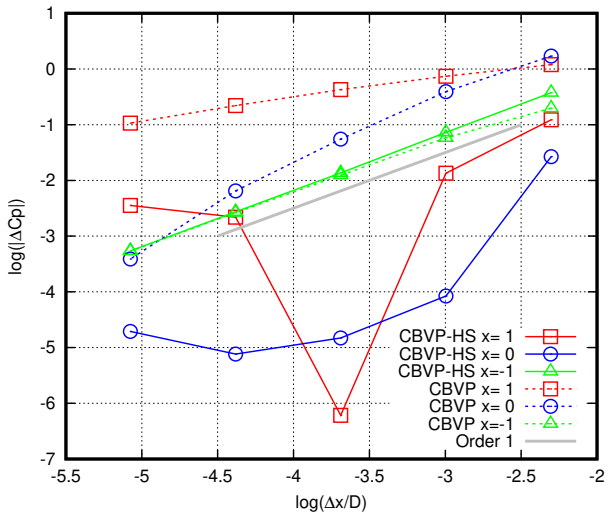


Figure 4.8 Cylinder ΔCp convergence as the mesh is refined

4.6.1.1 Entropy and Total Enthalpy

As the flow around the cylinder presents no heat source nor external force, the entropy (s) and total enthalpy (H) should be conserved in the entire computational domain. Thus, entropy and total enthalpy errors can be evaluated against the freestream conditions as:

$$\Delta s = |(s - s_\infty)| / s_\infty \tag{4.67}$$

$$\Delta H = |(H - H_\infty)| / H_\infty \tag{4.68}$$

$$s_\infty = P_\infty / \rho_\infty^\gamma \tag{4.69}$$

$$H_\infty = \frac{\gamma}{\gamma - 1}\frac{P_\infty}{\rho_\infty} + \frac{1}{2}||\mathbf{v}_\infty||^2 \tag{4.70}$$

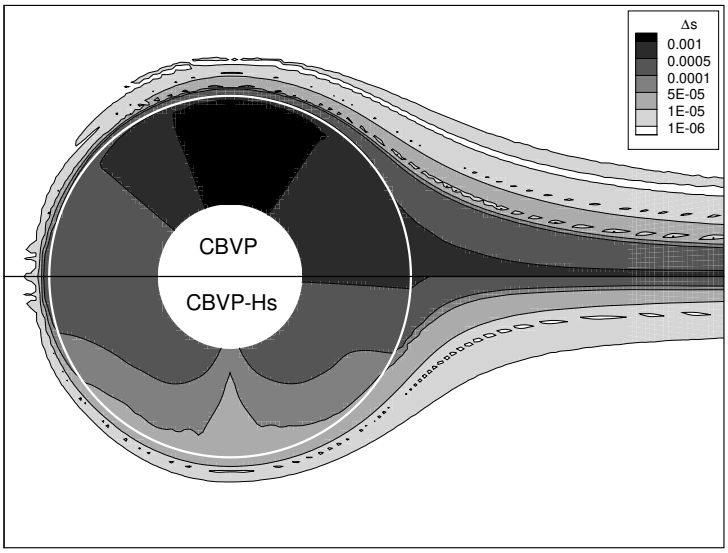


Figure 4.9 Entropy error for the cylinder on the Cartesian mesh $D/\Delta x = 80$. Immersed Boundary: white line. The center of the cylinder is blanked for clarity.

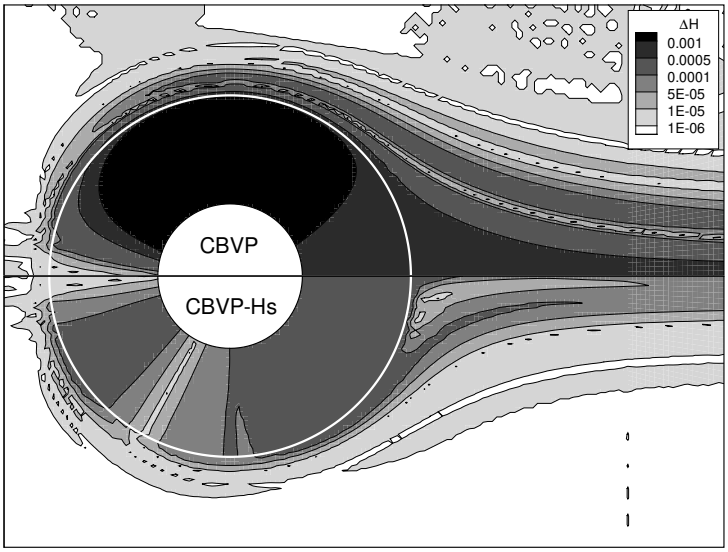


Figure 4.10 Total enthalpy error for the cylinder on the Cartesian mesh $D/\Delta x = 80$. Immersed Boundary: white line. The center of the cylinder is blanked for clarity.

The contours for the entropy and total enthalpy error are illustrated on Figures 4.9–4.10. These figures confirm that the CBVP-Hs method reduces the error in the wake and at the maximum suction points of the cylinder. However, the error is similar for both methods at the forward stagnation point. As an additional comparison point, the maximum entropy error

in the field for the CBVP is ≈1e-3 while it is ≈6e-4 for the CBVP-Hs. The same behavior is observed for the total enthalpy error whose maximum is ≈1e-3 for the CBVP and ≈3e-4 for the CBVP-Hs. As a comparison, a body-fitted simulation using an equivalent wall cell size was performed (not shown). The error for entropy and total enthalpy is respectively ≈1e-5 and ≈8e-6 using the CCST boundary condition, which shows that even for a body-fitted mesh the entropy and total enthalpy are not conserved to machine accuracy.

The error at the wall for Cp, s and H are plotted on Figure 4.11 for both methods and for the different refinement levels. The error is presented as the RMS of the local ΔCp, Δs and ΔH respectively. This gives an indication of the global wall error (RMS) and confirms that the CBVP-Hs is more accurate. In fact, the CBVP solution with $D/\Delta x = 160$ is equivalent to the CBVP-Hs with $D/\Delta x = 40$ in terms of the RMS.

From Figure 4.11, the error is observed to be lower for the CBVP-Hs method while also offering a faster convergence rate on ΔH and Δs. The order of convergence is observed to be 1 for the CBVP-Hs method on entropy and total enthalpy, while it is slightly lower for the CBVP method. The volume penalization method is limited to 1^{st} order accuracy, thus it is logical to obtain a global 1^{st} order of convergence at best although the solution is 2^{nd} order accurate in the field (2^{nd} order Roe Scheme).

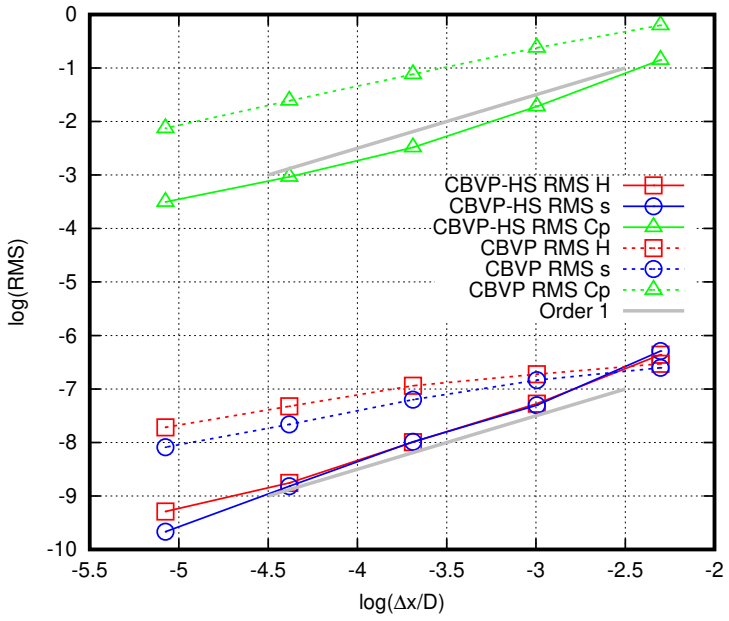


Figure 4.11 Global wall error (RMS) for the cylinder

4.6.1.2 Continuity of the Solution

The cylinder case with $D/\Delta x = 80$ is used to illustrate the continuity of the solution in the vicinity of the IB and inside the solid zone. Both the CBVP and CBVP-Hs are designed with hyperbolic penalization terms on the continuity and energy equations allowing the propagation of information from the fluid to the solid in the normal direction to the interface. However, the two methods differ on the momentum equation where the continuity of the velocity is imposed by an artificial dissipation term for the CBVP and by a hyperbolic penalization term for the CBVP-Hs. For both methods, a no-penetration velocity ($\mathbf{v}\cdot\mathbf{n}_\phi = 0$) is enforced in the solid, thus only the tangential component of the velocity remains.

As shown on Figure 4.12, the tangential velocity is continuous across the IB for both methods. For the CBVP, the continuity is enforced via numerical dissipation. This does not only produce an unphysical extension of the tangential velocity in the solid, it also pollutes the solution in the fluid near the IB. On the other hand, the CBVP-Hs extends the tangential velocity following the physics of the wall streamline. Also note that the curvature terms are only activated in the vicinity of the interface, the remainder of the solid zone only sees convection terms in the direction normal to the IB. It translates into a proper extension of the fluid properties near the IB, then into straight contour lines pointing towards the center of the cylinder.

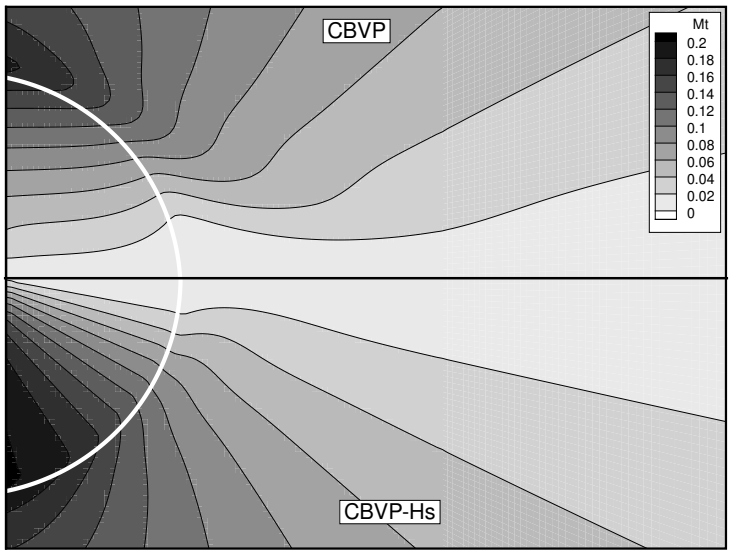


Figure 4.12 Continuity of the tangential Mach number (M_t) near the IB and inside the solid for both CBVP and CBVP-Hs ($D/\Delta x = 80$).

The continuity of the pressure (Cp) is illustrated on Figure 4.13. In terms of pressure both

the CBVP and CBVP-Hs methods behave in a similar and satisfactory manner inside the solid. The main difference is coming from the quality of the solution, which is more accurate for the CBVP-Hs on an equivalent mesh. This is evidenced for instance by the Cp near the aft stagnation point which is closer to the analytical value of $Cp = 1$.

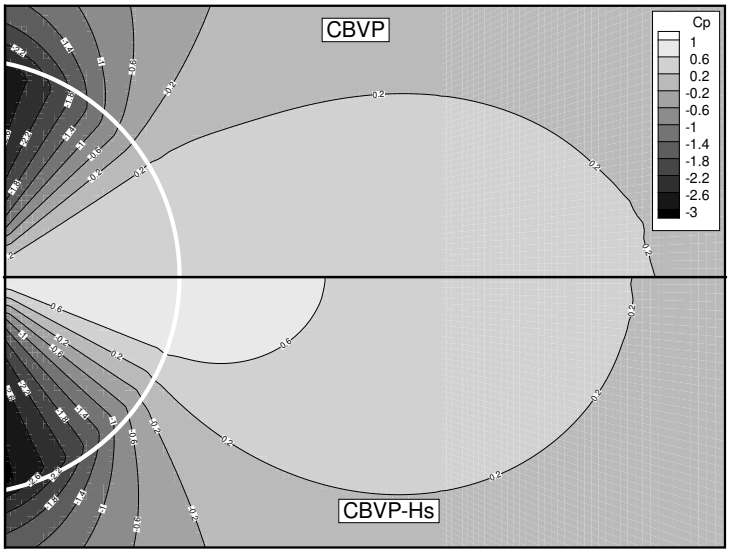


Figure 4.13 Continuity of the pressure coefficient (Cp) near the IB and inside the solid for both CBVP and CBVP-Hs ($D/\Delta x = 80$).

It is also interesting to verify the continuity of entropy and total enthalpy as it should be strongly imposed using the CBVP-Hs. As illustrated in Figures 4.9–4.10, both methods propagate the entropy and total enthalpy from the fluid to the solid. The CBVP method exhibits larger error at the suction peak and aft of the cylinder on the fluid side. This error is also propagated into the solid. On the other hand, the CBVP-Hs method shows a lower error on the fluid side (for both Δs and ΔH). This is also reflected inside the solid as expected. These two figures emphasize the difference in accuracy of the two methods in the vicinity of the IB, with a better solution achieved by the CBVP-Hs.

4.6.2 Subsonic Flow Around a NACA0012

A more representative test case for aerospace applications is the subsonic flow around an airfoil. As stated earlier, the first order accurate penalization method is not well suited for the representation of sharp features like trailing edges. Here, the issue is worked around by using a NACA0012 airfoil with a blunt trailing edge [152], allowing easier meshing of the solid (interior mesh). An unstructured mesh is used to show that the method is also

applicable to general meshes. This also allows a better comparison between the body-fitted and penalization methods.

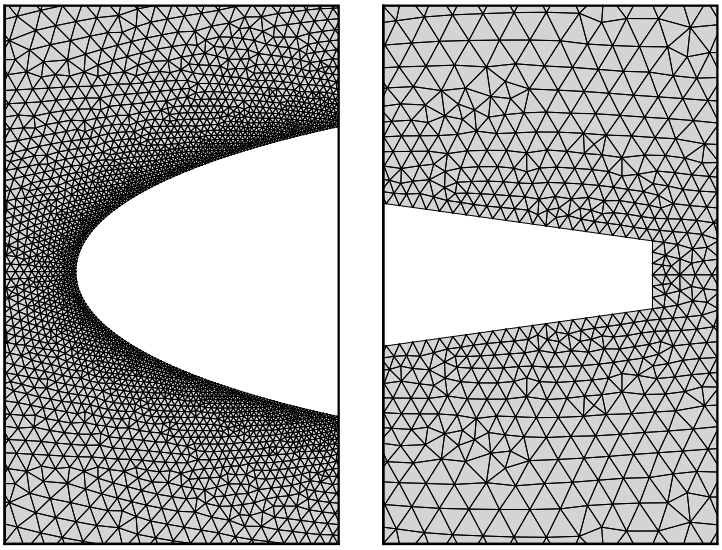


Figure 4.14 Body-fitted mesh for the NACA0012 airfoil with blunt trailing edge. Right: zoom on the leading edge. Left: zoom on the trailing edge.

The body-fitted and IB meshes are respectively shown in Figures 4.14–4.15. The meshes are refined at the leading and trailing edges and the far field is located at 50 chords from the airfoil. The same parameters are used to generate the IB and body-fitted meshes, ensuring comparable cell sizes and refinement. In terms of the chord (c), the cell sizes at the wall, leading edge, trailing edge and far field are respectively: 5e-3c, 1e-3c, 5e-4c and 4c. A linear cell size growth is allowed from the wall to the far field boundary. For the body-fitted and IB meshes, it gives a total of 36420 and 49136 cells respectively.

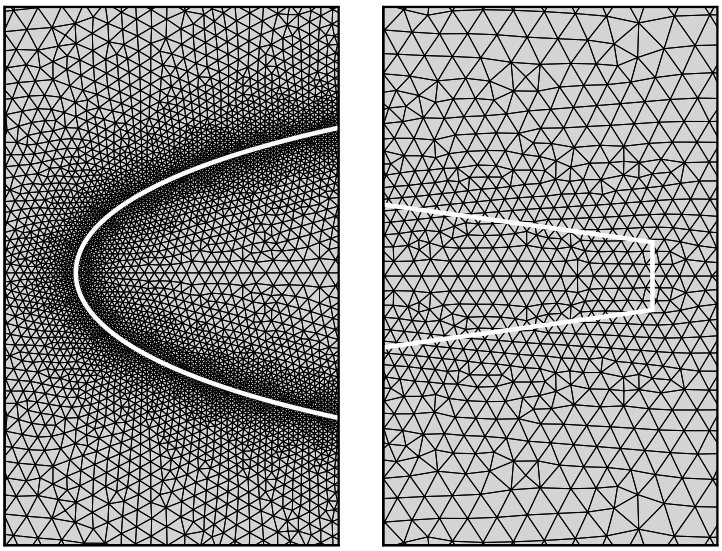
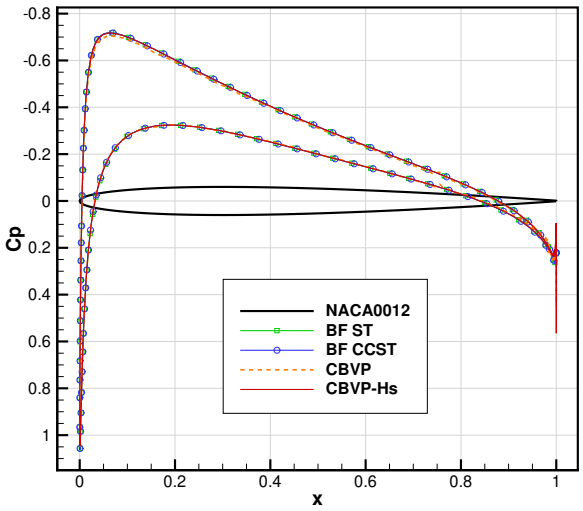
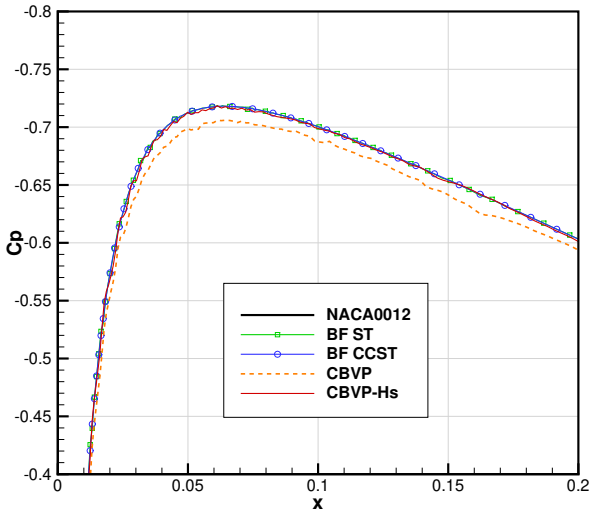


Figure 4.15 Immersed boundary mesh for the NACA0012 airfoil with blunt trailing edge. Right: zoom on the leading edge. Left: zoom on the trailing edge.

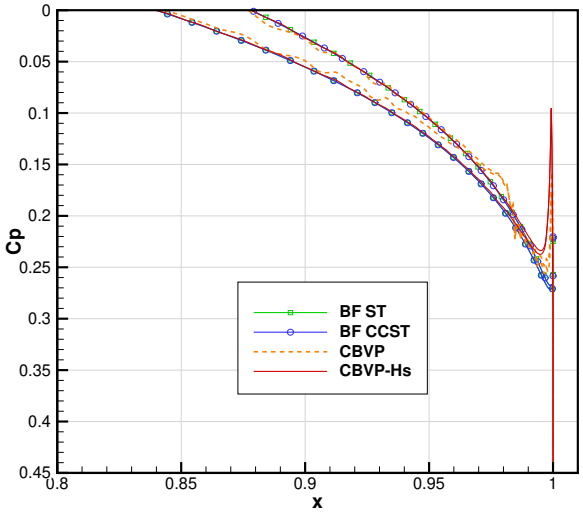
For this test case, the standard subsonic flow conditions of $Mach = 0.5$ and $AoA = 1.25\,\text{deg}$ are used [153]. To allow a fair comparison with the penalization methods, the body-fitted (BF) simulation is performed with both the ST and CCST wall boundary conditions. As a reminder, the CBVP and CBVP-Hs both account for the wall curvature, similar to the CCST.



(a) Global view



(b) Zoom near Cp_{max}



(c) Zoom at the trailing edge

Figure 4.16 Wall Cp for the NACA0012 airfoil at Mach= 0.5 and AoA= 1.25 deg. Comparison to the BF ST and BF CCST.

The results from Figure 4.16 illustrate the Cp distribution at the wall. For the body-fitted results, the ST and CCST provide a similar solution where the two Cp curves are superimposed on Figure 4.16b. On a geometry with moderate curvature (κ) and using a mesh with adequate refinement (Δx) both the ST and CCST are expected to be equivalent. This is justified by Eq. (4.20) which provides an error estimate for the pressure extrapolation $\Delta P/P_w$. When $\kappa \Delta x \to 0$, $\Delta P/P_w \to 0$ and the CCST and ST become equivalent. For this test case,

the ST shows a maximum error of $\Delta P/P_w \approx 1\%$ with $\gamma M_w^2 \kappa = 12.3$ and a wall cell size of $\Delta x =$ 1e-3. An error of $\approx 1\%$ is deemed small enough to explain the superposition of the Cp curves for the ST and CCST in Figure 4.16. The lift coefficients for the body-fitted solutions are respectively $C_L = 0.1776$ and $C_L = 0.1777$ for the ST and CCST boundary conditions.

Now considering the penalization methods, the Cp distribution for the CBVP method is close to the body-fitted solution with a slight overestimation in the vicinity of the maximum suction. Also, the pressure coefficient is not as accurate at the trailing edge for this method. On the other hand, the Cp distribution for the CBVP-Hs fits well with the body-fitted results. The lift coefficients are respectively $C_L = 0.1693$ and $C_L = 0.1768$ for the CBVP and CBVP-Hs. Again the CBVP-Hs is much closer to the body-fitted solution ($C_L \in [0.1776, 0.1777]$). This result shows that this penalization method is able to perform well on an airfoil case if the mesh is sufficiently refined, especially around sharp features. For instance, it was observed that the trailing edge must contain at least 6 solid cells in its thickness for the penalization method to work well (Figure 4.15). In this way, the interpolation and gradient stencils use only valid solid cells and avoid the ones near the body centerline. The results also confirm that the penalization method is suitable for unstructured grids.

4.6.3 Flow Around an Ice Horn

Ice shapes can be very challenging for mesh generation and for the flow solver. One particular case that was found difficult to deal with in a previous communication [151] is the ice horn generated by the conditions listed in Table 4.1. This ice shape exhibits an ice horn of high curvature which can lead to flow separation, even with a Euler flow solver. This behavior is shown in Figure 4.17, where the simulation is performed using the body-fitted mesh of Figure 4.18a and the ST wall boundary condition.

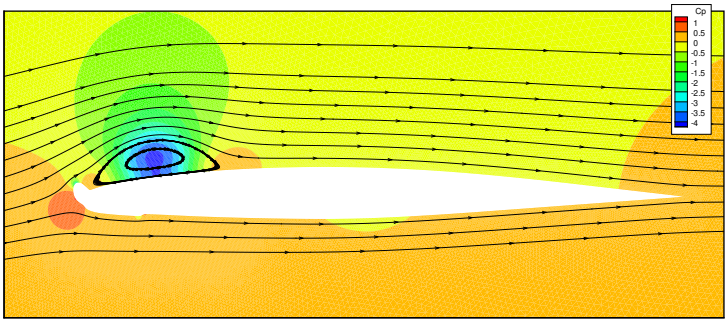
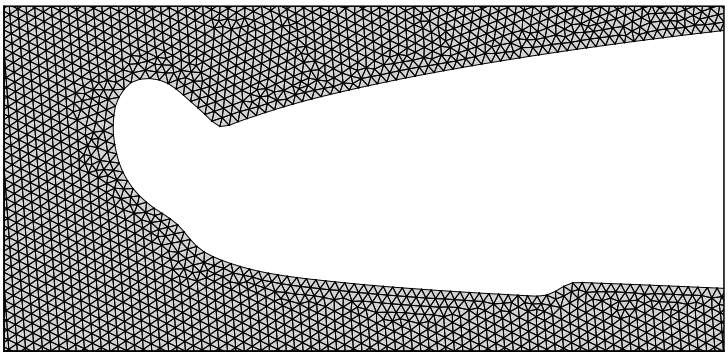


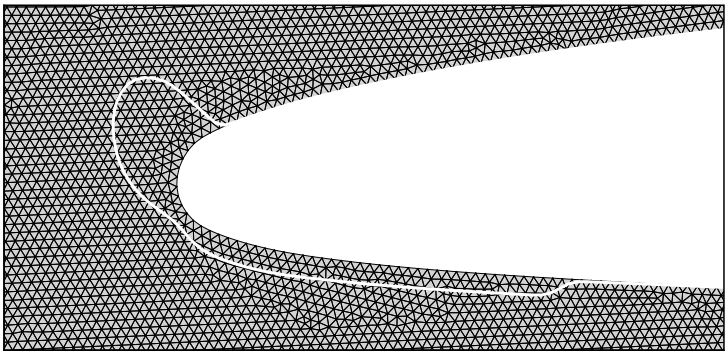
Figure 4.17 Field of Cp contours and velocity streamtraces for the ice horn case using a body-fitted mesh and the ST wall boundary condition

For this test case, the strategy is to use a body fitted mesh for the clean geometry (GLC305) and to use an IB to account for the ice shape (Figure 4.18b). In this way, the sharp trailing edge is solved accurately and the ice shape can be dealt with to first order accuracy (penalization).

Here, the Euler flow is solved on the ice accreted GLC305 where the ice shape was generated with IGLOO2D for an icing time of $675s$ (half the full ice accretion time). The body-fitted and IB meshes around the ice horn are shown in Figure 4.18. Again the same cell size is used in the vicinity of the wall for both cases to allow a fair comparison. For this test case, the wall Cp is converged to 10^{-6}.



(a) Body-Fitted



(b) Immersed Boundary

Figure 4.18 Mesh around the ice horn

The Cp distribution on the ice horn is compared in Figure 4.20. First, the body-fitted solutions using the two types of wall boundary conditions are examined: ST and CCST. The ST was previously shown to give a separated flow in Figure 4.17 and this can now be observed in Figure 4.20. This flow separation is induced by the numerical viscosity of the flow solver

and has no physical meaning. It would be preferable to obtain an attached solution to be in line with the theory of inviscid flows. Numerical experiments showed that a finer mesh allows an attached solution to be obtained with the ST (Figure 4.21).

An alternative approach is to apply the CCST to account for the wall curvature and therefore increase the accuracy of the solution on coarser mesh. On Figures 4.19–4.20, it is shown that the use of the CCST indeed leads to an attached flow solution on the current mesh. This also significantly increases the suction peak at the ice horn tip which is now at $Cp \approx -13$ in comparison to $Cp \approx -3$ for the ST approach. As the mesh is refined (Figure 4.21), the ST solution tends toward the CCST solution except the suction peak is not fully recovered yet. Assuming an error of 1% is desired on the pressure extrapolation (Eq. (4.20)), a cell size can be estimated for the ice horn. According to this approach, the cell size should be around 2e-5 for the ST and CCST to be equivalent.

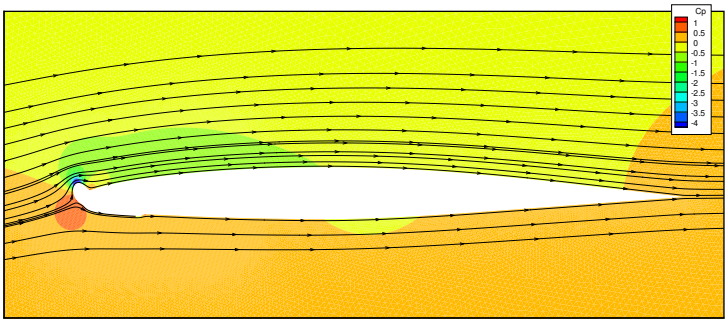
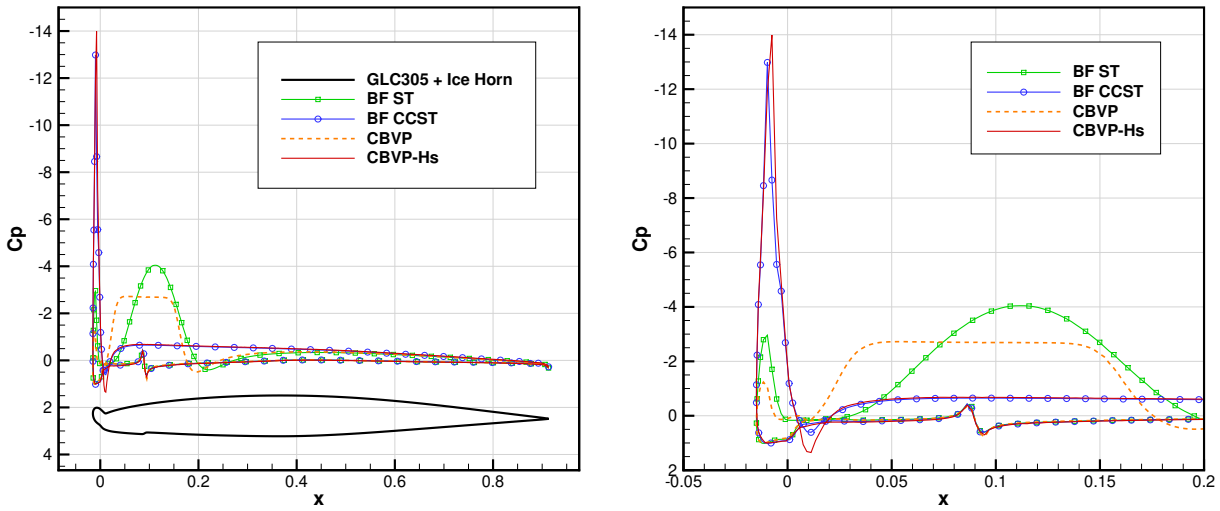


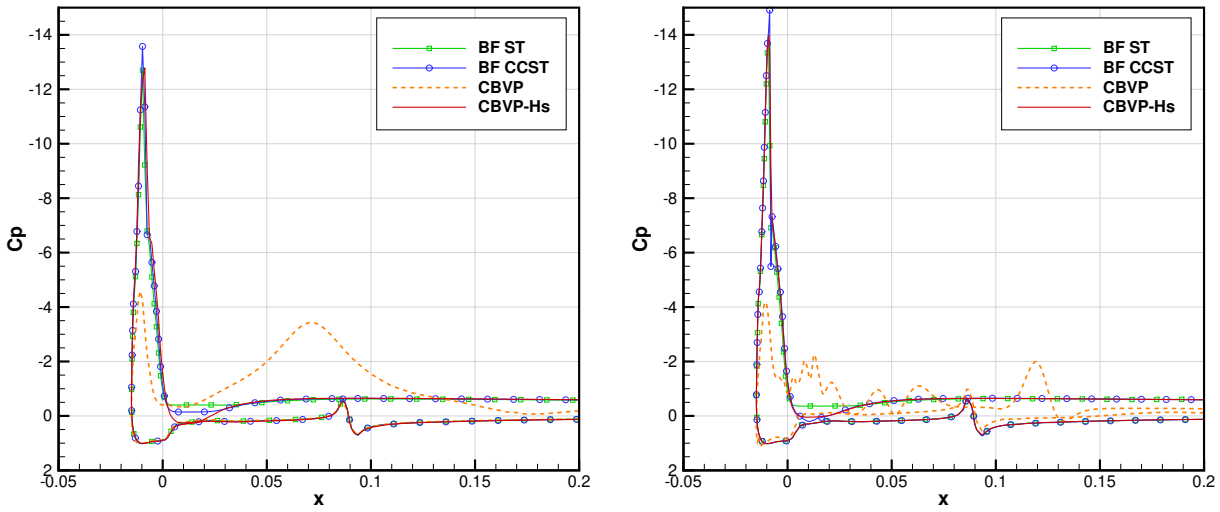
Figure 4.19 Field Cp contours and velocity streamtraces for the ice horn case using a body-fitted mesh and the CCST wall boundary condition



(a) Global view

(b) Zoom near the ice horn

Figure 4.20 Wall Cp for the ice horn (horn tip cell size 2e-3).



(a) horn tip cell size 1e-3

(b) horn tip cell size 5e-4

Figure 4.21 Wall Cp for the ice horn

The results from the penalization methods are examined next: CBVP and CBVP-Hs. Both method account for the wall curvature in the imposition of the wall boundary condition. However, as shown on Figure 4.20, the CBVP still exhibits flow separation on the ice horn case while the CBVP-Hs does not. This behavior is similar to the results obtained earlier on

the cylinder, where the CBVP-Hs was more accurate and behaved better aft of the cylinder for coarser meshes. Again, it suggests that the CBVP would give similar results if the mesh were refined. As the mesh is refined (Figure 4.21), the amplitude of the flow separation reduces at first (Figure 4.21a) and the suction peak increases. However by refining the mesh once more, the CBVP fails to converge to steady state, exhibiting unsteady flow behavior. This explains the poor CBVP solution on Figure 4.21b. In fact, it was difficult to converge the Cp further than 1e-4 for the CBVP method whereas the CBVP-Hs exhibited good convergence to steady state.

The difference between the two methods could be explained by the boundary conditions which conserve entropy and total enthalpy for the CBVP-Hs. Also the CBVP-Hs better impose the streamline conditions near the IB. The CBVP-Hs follows the boundary conditions imposed by the CCST and exhibits similar behavior on the ice horn case. From the previous observations, the CBVP-Hs is concluded to be more accurate and more suitable for geometries with high curvature, particularly on coarser meshes.

4.6.4 Transonic Flow Around a NACA0012

In order to show that the method is not only limited to subsonic flows, the transonic flow over a NACA0012 airfoil is computed. The flow parameters are selected as per [153], with $Mach = 0.8$ and $AoA = 1.25\,\text{deg}$. The comparison is performed between the Body-Fitted mesh approach using the CCST and the CBVP-Hs on meshes of equivalent cell sizes. The mesh used are twice as fine as the ones used for the subsonic test case with the same refinements zones and geometry (i.e. NACA0012 with a blunt trailing edge).

The results of Figure 4.22 show that for a mesh of comparable refinement, the CBVP-Hs is able to capture the shock location with reasonable accuracy compared to the CCST results. The lift coefficient for the CBVP-Hs is $C_L = 0.3460$ which is 1.6% off the body-fitted results ($C_L = 0.3517$). A better C_L could be obtained by properly refining the mesh in the vicinity of the shocks. However, it still shows the potential of the CBVP-Hs for the simulation of transonic and supersonic flows.

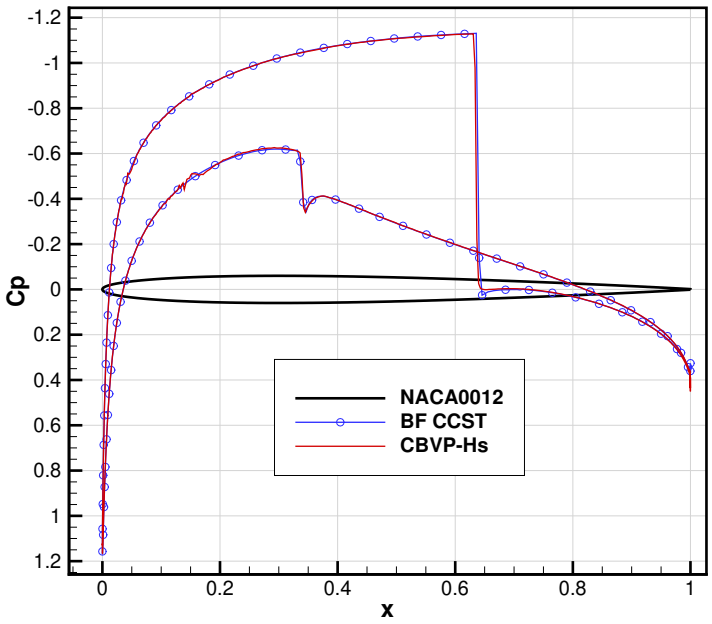


Figure 4.22 Wall Cp for the NACA0012 airfoil at Mach= 0.8 and AoA= 1.25 deg. Comparison to the BF ST and BF CCST.

4.7 Conclusion

In this paper, a penalization method which conserves entropy and total enthalpy is developed for the Euler equations.

Two types of numerical wall boundary conditions are described for body-fitted meshes: the symmetry technique (ST) and curvature-corrected symmetry technique (CCST). For the immersed boundary method, the penalization is selected for its simplicity of implementation and its mesh and discretization independence. The penalization method is usually applied to the Navier-Stokes equations and applications to the Euler equations are few. The characteristic-based volume penalization (CBVP) of [75] showed to be suitable for most cases but exhibited flow separation in some icing application with high curvature geometries. A new method is thus developed, the CBVP-Hs, which can be seen as the application of the CCST boundary conditions using the penalization method.

The body-fitted and penalization methods are tested on four test cases. A mesh convergence is performed for the weakly compressible flow around a circular cylinder. Then the subsonic and transonic flows around a NACA0012 airfoil are solved. Also, the airflow around an ice horn of high curvature is evaluated to assess the methods on a challenging icing case.

The CBVP and CBVP-Hs are found to tend towards the same solution for the cylinder case. However, the CBVP method exhibits a higher error level on the pressure coefficient, conservation of total enthalpy and conservation of entropy compared to the CBVP-Hs. The CBVP-Hs is also found to be more accurate on coarser meshes. For instance, on the cylinder the CBVP-Hs solution on the $D/\Delta x = 40$ mesh is equivalent to the CBVP on the $D/\Delta x = 160$ mesh. For the ice horn case on a coarse mesh, the CBVP exhibits flow separation while the flow remains attached for the CBVP-Hs, which is more in line with inviscid flow theory. A mesh refinement was not sufficient to obtain an attached flow with the CBVP. Also, the CBVP-Hs shows a good potential for the simulation of transonic flows as it can capture the same shock location as a Body-Fitted simulation of comparable mesh refinement.

It is concluded that the application of the penalization method to the Euler equations requires special care to obtain accurate solutions on high curvature geometries. The newly developed approach (CBVP-Hs) proved to be a suitable candidate for this purpose. Although the method was presented in 2D in this article, no major difficulty is expected for the extension of this penalization method in 3D, which will be done in the future.

CHAPTER 5 ARTICLE 2: A PENALIZATION METHOD FOR EULERIAN DROPLET IMPINGEMENT SIMULATIONS TOWARDS ICING APPLICATIONS

P. Lavoie, E. Radenac, G. Blanchard, Éric Laurendeau, and P. Villedieu, "A penalization method for eulerian droplet impingement simulations towards icing applications," AIAA Journal, 2021, (submitted).

5.1 Abstract

The numerical prediction of in-flight ice accretion generally involves geometry updates and re-meshing as the ice builds up. However, the generation of body-fitted meshes around complex ice shapes is not trivial and can be repeated several times to obtain the final ice shape. The use of an immersed boundary method can simplify the mesh generation and help in the automation of the ice accretion process. This paper studies the application of an immersed boundary method to Eulerian droplet impingement simulations. A penalization method is suggested requiring only the addition of source terms in the continuous form of the equations. The wall boundary condition must be treated with care to avoid droplets re-injection in the computational domain from a solid boundary. This is solved by the introduction of a droplet mask function in addition to the usual solid mask, providing an automatic detection of the wall boundary condition and therefore avoiding droplet re-injection. The approach is tested on canonical cylinder cases and on more realistic NACA0012 airfoil and ice horn cases. The results show that the solution from a body-fitted simulation can be reproduced using the penalization method.

5.2 Introduction

Numerical tools for the prediction of in-flight ice accretion have been developed and used for many years (e.g. [154]). Ice accretion is an unsteady multi-physics process where supercooled water droplets impinge on a cold surface [2] (e.g. aircraft wings, tail, etc.) and might freeze upon impact or run back and freeze farther downstream. Typical tools for the prediction of ice accretion segregate the simulation of the physics in different modules (e.g. LEWICE [155]) that are called sequentially in a quasi-steady approach:

1. mesh generation;

2. computation of the aerodynamics;
3. computation of the droplet trajectories and impingement rates;
4. evaluation of the convective heat transfer at the wall;
5. computation of the ice accretion (mass and heat balance);
6. geometry update (the ice shape is generated).

This process (modules 1 to 6) is usually repeated several times to increase the accuracy of the ice accretion prediction using a multi-step approach [10], requiring mesh generation for each step.

When using body-fitted (BF) meshes, the effort is spent on generating a good quality mesh that matches the geometry. This process can be difficult to automate on complex ice geometries, especially in 3D. On the other hand, when using an Immersed Boundary Method (IBM), the mesh generation can be much simpler (e.g. Cartesian grid) as the geometry is allowed to arbitrarily cut through the mesh. In this case the effort is spent on the correct imposition of the boundary condition on the immersed boundary. It is thus envisioned to replace the classical body-fitted meshes by the use of an immersed boundary method which could ease the automation of the ice accretion process.

Although there is a higher benefit in using Immersed Boundary Methods for 3D ice accretion, the developments are first performed in 2D using ONERA's IGLOO2D ice accretion suite [12]. For the aerodynamics, IGLOO2D uses a Euler flow solver combined with an integral boundary layer code. For the evaluation of the droplet trajectories, both Lagrangian and Eulerian solvers are available. An immersed boundary method (penalization) was previously developed for the Euler equations and presented in [156]. As a continuation, the objective of this paper is to apply the penalization method to the Eulerian solver for the droplet trajectories.

A variety of Immersed Boundary Methods are available from the literature, ranging from the geometric Cut-Cell approach [41, 157, 42] to discrete [80, 87, 158] and continuous methods [51, 67]. Continuous immersed boundary methods have the advantage of being independent of the discretization and numerical method. Furthermore, continuous approaches such as the penalization method of [67] are appealing for their simplicity of implementation. The penalization method was applied to incompressible Navier-Stokes equations [67, 39] and extended to compressible flows [73, 74, 72]. A variant of the penalization method, the Characteristic-Based Volume Penalization, was developed by [75] and applied to the Navier-Stokes and Euler equations [79, 156].

Applications of an immersed boundary method for the Eulerian droplet trajectory solver are few. A discrete method (a type of ghost-cell approach) is used by [16] to perform both 2D and 3D simulations on Cartesian grids. A similar approach is followed by [17] where a discrete approach is used in combination with a level-set. According to the authors' knowledge, there is no application of a penalization method for Eulerian trajectory solvers in the literature.

The application of the penalization method to the Eulerian droplet trajectory equations is not straightforward because of the nature of the wall boundary condition that changes along the wall according to the droplet trajectories. It is quite simple to deal with this situation when using ghost-cells, but it must be treated with care with the penalization method as droplets could otherwise enter the solid body and be re-injected in the field downstream. This paper suggests a way to apply the penalization method to the droplet equations which avoids droplet re-injection by using a double mask function.

The paper is structured in three main sections. First the Eulerian droplet equations and their application for Body-Fitted simulations are reviewed. Second, a penalization approach is suggested for droplet impingement and the double mask function is explained. Third, the verification of the method is made on canonical cylinder cases, on a NACA0012 clean airfoil in icing conditions and on an ice accreted GLC305 airfoil exhibiting an ice horn.

5.3 Eulerian Droplets Impingement

For ice accretion prediction, the fundamental information to be retrieved is the droplet impingement rate ($\dot{m}_{imp}$) on the body (e.g. an airfoil). This is generally computed in terms of collection efficiency (β) which can be seen as a non-dimensional impingement rate. The Eulerian approach for the evaluation of the droplet impingement consists in retrieving the volume fraction of water (α) and the droplet velocity ($\mathbf{v}_d$) in the field surrounding the body of interest (e.g. an airfoil). A system of PDE is then solved for the droplets assuming a one-way coupling with the aerodynamic field.

5.3.1 Governing Equations

The non-dimensional continuity and momentum equations for the droplets are respectively in non-conservative form [21]:

$$\begin{aligned} &\frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d) = 0 \\ &\frac{\partial \mathbf{v}_d}{\partial t} + \mathbf{v}_d \cdot \boldsymbol{\nabla} \mathbf{v}_d = \frac{C_D Re_d}{24 Stk} (\mathbf{v}_a - \mathbf{v}_d) + \left(1 - \frac{\rho_a}{\rho_d}\right) \frac{1}{Fr^2} \mathbf{g} \end{aligned} \tag{5.1}$$

In conservative form the equations become:

$$\begin{aligned} \frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d) &= 0 \\ \frac{\partial (\alpha \mathbf{v}_d)}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d \otimes \mathbf{v}_d) &= \frac{C_D Re_d}{24 Stk} \alpha (\mathbf{v}_a - \mathbf{v}_d) + \alpha \left(1 - \frac{\rho_a}{\rho_d}\right) \frac{1}{Fr^2} \mathbf{g} \end{aligned} \tag{5.2}$$

Where α is the non-dimensional volume fraction of water, $\mathbf{v}_d$ is the non-dimensional droplets velocity, $\mathbf{v}_a$ is the non-dimensional air velocity, ρ_d is the droplets density (density of water), ρ_a is the air density and C_D is the droplets drag coefficient. The droplets Reynolds number (Re_d), the Stokes number (Stk) and the Froude number (Fr) are respectively defined as:

$$Re_d = \frac{\rho_a ||\mathbf{v}_a - \mathbf{v}_d|| D_d}{\mu} \tag{5.3}$$

$$Stk = \frac{\rho_d D_d^2 U_\infty}{18 L \mu} \tag{5.4}$$

$$Fr = \frac{U_\infty}{\sqrt{Lg}} \tag{5.5}$$

where D_d is the droplet diameter, μ the dynamic viscosity of air and L a characteristic dimension (e.g. the chord length for an airfoil). The drag model of Schiller and Naumann [159] is used for the droplets which are assumed to remain spherical:

$$C_D = \begin{cases} \frac{24}{Re_d}(1 + 0.15 Re_d^{0.687}) & Re_d \leq 1000 \\ 0.4 & Re_d > 1000 \end{cases} \tag{5.6}$$

The Eulerian formulation of the equations allows simple evaluation of the collection efficiency (β) at the wall as α and $\mathbf{v}_d$ are known everywhere in the field and $\mathbf{n}_{bc}$ is known from the geometry.

$$\beta = \alpha \mathbf{v}_d \cdot \mathbf{n}_{bc} \tag{5.7}$$

The impinging water flux is then retrieved by:

$$\dot{m}_{imp} = (LWC) U_\infty \beta \tag{5.8}$$

where LWC is the Liquid Water Content and U_∞ the magnitude of the freestream velocity.

The complete equations are presented here for generality. However for the remainder of this paper the gravity term is neglected and thus only the drag force will act on the droplets.

5.3.2 Boundary Conditions

One important aspect of the Eulerian model for droplet impingement is the treatment of the boundary conditions. The system of equations is hyperbolic and therefore boundary conditions are only required for incoming characteristics [160, 21] (Figure 5.1). Additional numerical boundary schemes compatible with the physics of the problem are applied where boundary conditions are not imposed. For instance, a zero-order extrapolation is used for outgoing characteristics. Assuming the boundary normals are pointing out of the computational domain (Figure 5.1), the boundary condition for the droplet equations can be formulated as follows.

Far-field :

$$\left.\begin{array}{rl} \alpha & = \alpha_\infty \\ \mathbf{v}_d & = \mathbf{v}_\infty \end{array}\right\} \text{if} \quad \mathbf{v}_d \cdot \mathbf{n}_{bc} \leq 0 \tag{5.9}$$

Wall :

$$\left.\begin{array}{rl} \alpha & = 0 \\ \mathbf{v}_d & = 0 \end{array}\right\} \text{if} \quad \mathbf{v}_d \cdot \mathbf{n}_{bc} \leq 0 \tag{5.10}$$

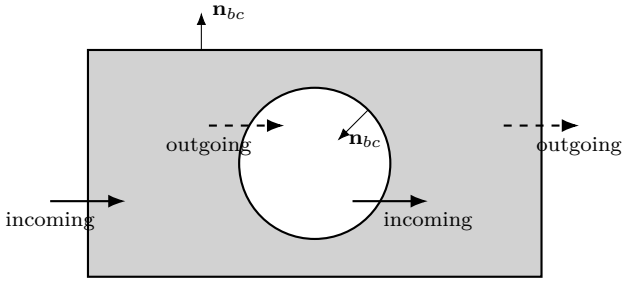


Figure 5.1 Boundary Condition Diagram for Droplets Incoming and Outgoing Information

In this paper, the focus is on the wall boundary condition as only this condition is treated with the penalization method. When the droplets enter the computational domain from the solid ($\mathbf{v}_d \cdot \mathbf{n}_{bc} \leq 0$), a boundary condition is applied on the primitive variables (Eq. (5.10)), enforcing a null flux. Otherwise when droplets impinge the body ($\mathbf{v}_d \cdot \mathbf{n}_{bc} > 0$), an extension of the primitive variable is performed. For body-fitted meshes, this is typically done by setting $\mathbf{n}_{bc} \cdot \boldsymbol{\nabla}\alpha = 0$ and $\mathbf{n}_{bc} \cdot \boldsymbol{\nabla}\mathbf{v}_d = 0$ at the boundary, but a slightly different approach is used for the penalization method in the next section.

5.4 Penalization Method

This section discusses the definition of the immersed boundary and presents the penalization method applied to the droplet equations.

5.4.1 Immersed Boundary Representation

For this paper, the immersed boundary is defined by a discrete list of nodes (2D) and its location in the mesh is defined by the level-set $\phi = 0$, where ϕ is the signed distance field from the immersed boundary. Values of ϕ are computed using a geometric approach : evaluating the minimum projected distance to the edges forming the immersed boundary [129].

The sign of ϕ is determined by a ray casting algorithm [129] where the immersed boundary is considered as a closed body. Using a signed distance field leads to a simple evaluation of the normals ($\mathbf{n}_\phi$) and curvature (κ) of the immersed boundary using:

$$\mathbf{n}_\phi = -\frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} \tag{5.11}$$

$$\kappa = \boldsymbol{\nabla} \cdot \mathbf{n}_\phi. \tag{5.12}$$

Note that the normal based on ϕ has a negative sign in order to point towards the solid zone ($\phi < 0$). This is useful in the definition of the penalization method presented next.

To extract the data at the immersed boundary (e.g. α, $\mathbf{v}_d$), a weighted least square interpolation at the discrete nodes defining the immersed boundary is used. The interpolation stencil is determined firstly by identifying the cell containing the immersed boundary node, and secondly by storing the extended neighborhood of this cell (neighbors sharing a node with the IB cell). An inverse distance weighting Eq. (5.13) is used for the interpolation with a smoothing parameter ($\epsilon = 0.5\Delta x_J$) to avoid dividing by a small value when cell centers are near the interpolation point. The weight for the interpolation point P and a cell-center J (part of the interpolation stencil) is written as:

$$w_J = \frac{1}{\sqrt{||\mathbf{r}_{PJ}||^2 + \epsilon^2}} \tag{5.13}$$

where Δx_J is the characteristic size of cell J and $||\mathbf{r}_{PJ}||$ is the distance between P and J.

In this paper, the collection efficiency (β) is evaluated at the immersed boundary (IB) in two steps. First, the primitive variables (α, $\mathbf{v}_d$) and the normals ($\mathbf{n}_\phi$) are interpolated at the IB discrete nodes using the weighted least square method. Second, the collection efficiency is

evaluated with $\beta = \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi$. As the interpolation stencil includes solid cells, the penalization method described in the following sections must be designed so the interpolation stencil in the vicinity of the solid/fluid interface is filled with valid data to perform the interpolation (e.g. valid data in the solid).

5.4.2 Application to the Droplet Equations

The Volume Penalization method [67] consists in adding source terms in the continuous form of the equation to enforce the desired boundary condition. The source terms are activated/deactivated using a mask function (χ) equal to unity in the solid and zero in the fluid. In this way, only the solid is penalized and the usual equations are retrieved in the fluid. A penalization parameter (η) ensures the boundary condition is enforced accurately. The volume penalization enforces the boundary conditions at the cell centers surrounding the immersed boundary (for a finite volume cell-centered method). Thus only an approximation of the IB is seen by the solver and this typically limits the penalization method to 1^{st} order accuracy in space. However, second order accurate approaches can be implemented by a discretization of the penalization source term [70, 71]. In this paper, the former approach is used.

For the droplet equations, no boundary condition should be applied for impinging droplets and a Dirichlet condition must be enforced in the shadow zone ($\mathbf{v}_d \cdot \mathbf{n}_\phi \leq 0$) to avoid droplet re-injection in the computational domain (see Eq. (5.10) and Figure 5.1). A typical penalization method uses a fluid/solid mask function (χ) to apply the penalization term everywhere in the solid. In this paper a sharp Heaviside function (Figure 5.2 and Eq. (5.14)) is used. However this is not sufficient as the Dirichlet condition would then also be applied in the impingement zone ($\mathbf{v}_d \cdot \mathbf{n}_\phi > 0$), where no boundary condition is required. To solve this issue, an inflow/outflow droplet mask function (χ_d) is also defined as a sharp Heaviside function (Figure 5.3 and Eq. (5.15)).

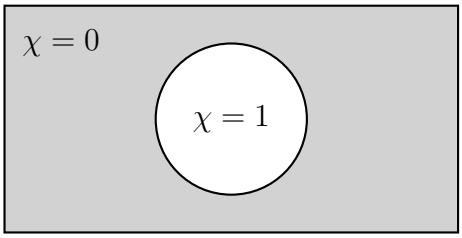
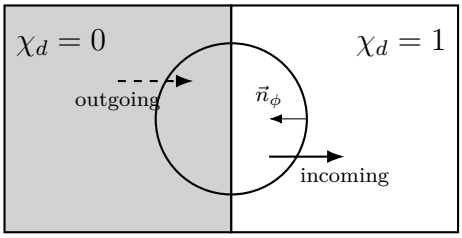


Figure 5.2 mask function (Solid:$\chi = 1$, Fluid:$\chi = 0$)

$$\chi = \begin{cases} 0 & \phi \geq 0 \\ 1 & \phi < 0 \end{cases} \tag{5.14}$$



$$\chi_d = \begin{cases} 0 & \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi \geq 0 \\ 1 & \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi < 0 \end{cases} \tag{5.15}$$

Figure 5.3 droplet mask function (shadow zone:$\chi_d = 1$, impingement zone:$\chi_d = 0$)

To apply the penalization term correctly, the two mask functions are combined ($\chi\chi_d$) which allows penalizing the equations only in the solid shadow zones ($\chi\chi_d = 1$, Figure 5.4).

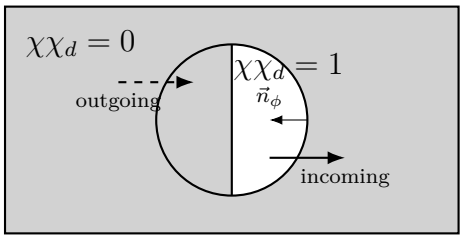


Figure 5.4 combined droplet mask function (shadow solid:$\chi\chi_d = 1$, impingement solid/fluid:$\chi\chi_d = 0$)

With the combined mask function defined, the penalized droplet equations can be obtained. The penalization terms are first applied to the non-conservative form of the equations so the boundary conditions are applied on the primitive variables. Then, the equations are transformed in conservative form for implementation. The non-conservative form can be expressed as, including the penalization terms:

$$\begin{aligned} &\frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d) = -\frac{\chi\chi_d}{\eta}\alpha \\ &\frac{\partial \mathbf{v}_d}{\partial t} + \mathbf{v}_d \cdot \boldsymbol{\nabla}\mathbf{v}_d = (1 - \xi\chi)\left(\frac{C_D Re_d}{24Stk}(\mathbf{v}_a - \mathbf{v}_d) + \left(1 - \frac{\rho_a}{\rho_d}\right)\frac{1}{Fr^2}\mathbf{g}\right) - \frac{\chi\chi_d}{\eta}\mathbf{v}_d \end{aligned} \tag{5.16}$$

Implementation-wise the conservative form of the equations is used and it can be written as:

$$\begin{aligned} \frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d) &= -\frac{\chi\chi_d}{\eta}\alpha \\ \frac{\partial (\alpha \mathbf{v}_d)}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d \otimes \mathbf{v}_d) &= (1 - \xi\chi)\left(\frac{C_D Re_d}{24Stk}\alpha(\mathbf{v}_a - \mathbf{v}_d) + \alpha\left(1 - \frac{\rho_a}{\rho_d}\right)\frac{1}{Fr^2}\mathbf{g}\right) - 2\frac{\chi\chi_d}{\eta}\alpha\mathbf{v}_d \end{aligned} \tag{5.17}$$

For droplets traveling from the solid to the fluid zone ($\chi = \chi_d = 1$), the penalization terms drive the volume fraction of water (α) and the droplet velocity ($\mathbf{v}_d$) to zero. Otherwise when either χ or χ_d are null, no penalization is applied and the physical equations are solved in the solid acting as a natural extension of the droplet characteristics from the fluid to the solid.

When solving Eq. (5.16) or Eq. (5.17) in the solid shadow zone ($\chi = \chi_d = 1$), η is selected small enough for the physical droplet equations to be negligible in front of the penalization terms ($\eta \ll 1$). Using the droplet continuity equation as an example, it comes back to solving an ordinary differential equation of the form:

$$\frac{d\alpha}{dt} = -\frac{\alpha}{\eta} \tag{5.18}$$

The solution of this ODE is a rapidly decaying exponential, meaning $\alpha = 0$ is imposed almost instantaneously:

$$\alpha = A_0 e^{-t/\eta} \tag{5.19}$$

The penalization parameter η can be seen as a characteristic timescale which must be small ($\eta \ll 1$) in order to enforce the boundary condition. In this paper, it is selected as $\eta = 10^{-8}$.

In this paper, two penalization methods are presented and are both available from Eq. (5.16) or Eq. (5.17) using the optional parameter ξ. The Volume Penalization (VP) method is obtained with $\xi = 0$ and the VP-SSO (VP Solid Source Off) method with $\xi = 1$. Option $\xi = 0$ seems to be a natural expression of the penalization method. But it requires setting a value for the gas velocity field in the solid area as the physical source terms are activated. No investigation will be made on the ideal value to be applied to this fictitious gas velocity field. If $\xi = 0$, it will be assumed that such a velocity field is available, for example the one provided by an IBM solution on the aerodynamic field. Otherwise, option $\xi = 1$ is available, where the physical source terms are deactivated in the solid zone rendering the droplet solution independent of the gas velocity field.

5.4.3 Numerical Method

In this paper the droplet equations are discretized using a cell-centered Finite Volume Method. The fluxes are evaluated with a HLL scheme [161] inspired by the work of [162]. A piecewise linear reconstruction is used to obtain a 2^{nd} order accurate scheme. The flux at

the face ($\mathbf{F}_{n,f}$) is obtained from the left (L) and right (R) states using:

$$S_L = \min\left(\mathbf{v}_{d,L}\cdot\mathbf{n},\ \mathbf{v}_{d,R}\cdot\mathbf{n}\right) \tag{5.20}$$

$$S_R = \max\left(\mathbf{v}_{d,L}\cdot\mathbf{n},\ \mathbf{v}_{d,R}\cdot\mathbf{n}\right) \tag{5.21}$$

$$\mathbf{F}_{n,f} = \begin{cases} \mathbf{F}_L & S_L \geq 0 \\ \mathbf{F}_{HLL} & S_L \leq 0 \leq S_R \\ \mathbf{F}_R & S_R \leq 0 \end{cases} \tag{5.22}$$

$$\mathbf{F}_{HLL} = \frac{S_R\mathbf{F}_L - S_L\mathbf{F}_R + S_L S_R(\mathbf{W}_R - \mathbf{W}_L)}{S_R - S_L} \tag{5.23}$$

with

$$\mathbf{W} = \begin{bmatrix} \alpha \\ \alpha u_d \\ \alpha v_d \end{bmatrix}, \quad \mathbf{F} = (\mathbf{v}_d\cdot\mathbf{n})\mathbf{W} \tag{5.24}$$

An explicit time scheme with implicit source terms is used to avoid stability issues related to the drag and penalization terms. The gravity term is deactivated for all simulations.

5.5 Results

In this section, four test cases on three different configurations are used to assess the penalization method. The simulation parameters are summarized in Table 5.1.

Table 5.1 Simulation Parameters

	Cylinder	**Airfoil**	**Ice Horn**
Geometry	Cylinder	NACA0012	GLC305
Chord	D=2.0	0.5334	0.9144
LE radius	1.0	8.46e-3	–
AoA	0.0	4.0	4.0
Mach	0.1	0.185	0.273
$\mathbf{P_{static}}$	100kPa	95.61kPa	101.325 kPa
$\mathbf{T_{static}}$	300.0K	245.2K	268.3K
LWC	1.0 g/m^3	1.3 g/m^3	0.54g/m^3
MVD	304.5 μm	20 μm	20μm
Stk	9.688	9.689	–

The parameters for the cylinder are selected to achieve a nearly incompressible flow and

maintain a similar Stokes number than encountered in icing simulations. The Stokes number is evaluated against the leading edge radius which can be computed as $R_{LE} = 1.1019(t_c)^2 c$ for a NACA 4-digit airfoil [152], where t_c is the thickness to chord ratio.

In the following sections, when comparing body-fitted and immersed boundary results, an equivalent mesh size is used at the wall to provide a fair comparison. Both VP and VP-SSO approaches will be used. Although no definite rule is given for the definition of the fictitious aerodynamic velocity field, simple choices will be made for this term in order to compare the VP and VP-SSO methods, and to evaluate what maintaining a source term in the immersed area tends to produce. For the airfoil cases, the aerodynamic velocity field is provided by the Euler IBM [156]. For the cylinder cases, the field provided by the potential solution is extended inside the solid domain.

5.5.1 No Drag nor Gravity

An interesting test case for the droplet equations occurs when the droplet drag and gravity are set to zero. In such a case, the droplet equations are independent of the aerodynamic field and an analytical solution is retrieved for the collection efficiency (β) as long as an analytical equation is available for the wall geometry.

5.5.1.1 Analytical Solution

Considering a cylinder of radius R_c centered at $(x, y) = (0, 0)$, the analytical solution at the wall is expressed as:

$$\begin{cases} \alpha = \alpha_\infty, \quad \mathbf{v}_d = \mathbf{v}_\infty & \text{if} \quad \mathbf{v}_d \cdot \mathbf{n}_\phi > 0 \\ \alpha = 0, \quad \mathbf{v}_d = 0 & \text{if} \quad \mathbf{v}_d \cdot \mathbf{n}_\phi \leq 0 \end{cases} \tag{5.25}$$

where the droplet velocity is:

$$\mathbf{v}_\infty = U_\infty(\cos(AoA), \sin(AoA)) \tag{5.26}$$

and for a cylinder the normal to the wall is:

$$\phi = \sqrt{x^2 + y^2} - R_c \tag{5.27}$$

$$\mathbf{n}_\phi = -\frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} = -\frac{(x, y)}{\sqrt{x^2 + y^2}}. \tag{5.28}$$

In turn, the collection efficiency β is evaluated as:

$$\beta = \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi \tag{5.29}$$

It means that for $AoA = 0$, the collection efficiency is defined at the wall as:

$$\beta = \alpha_\infty U_\infty \max\left(-\frac{x}{R_c}, 0\right) \tag{5.30}$$

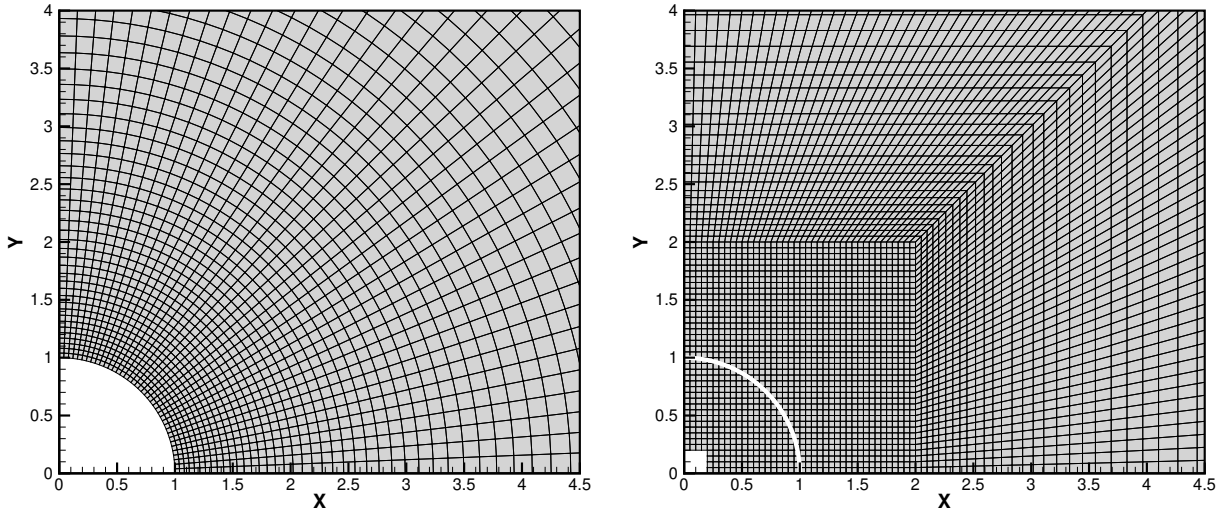
or in polar coordinates:

$$\beta = \alpha_\infty U_\infty \max(-\cos\theta, 0) \tag{5.31}$$

For this specific case, the curvilinear distance is $s = (\theta - \pi)R_c$, which is useful to present the results.

5.5.1.2 Meshes

The simulations are performed for a unit radius cylinder using a body-fitted structured mesh and a Cartesian grid for the immersed boundary method. The structured mesh has a wall cell size of 0.04 radius for a total of 16 000 cells with a far-field located at 50 radii from the cylinder. The Cartesian grid has a wall cell size of 0.05 radius for a total of 31 936 cells with a far-field located at 50 radii from the cylinder. The Cartesian grid is only generated for a square zone surrounding the immersed boundary (Figure 5.5b) and a structured mesh is used up to the farfield boundary in order to reduce the number of cells. Both meshes are illustrated in Figure 5.5 and correspond to the mesh refinement level 2 from Table 5.2.



(a) Body-fitted structured mesh ($D/\Delta x = 50$, 16K cells) (b) Cartesian mesh ($D/\Delta x = 40$, ∼32K cells)

Figure 5.5 Meshes for the unit radius cylinder

5.5.1.3 Numerical Solution

The wall collection efficiency (β) is presented in Figure 5.6 where the analytical solution is compared with the body-fitted simulation and the penalization method. It shows that the penalization method is able to reproduce the analytical solution as all the curves are overlaid.

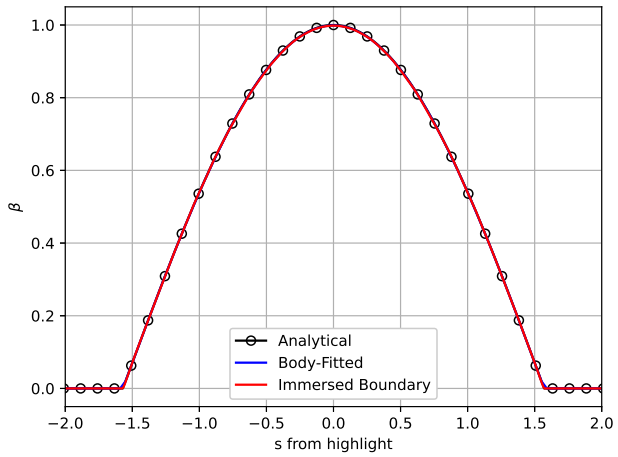
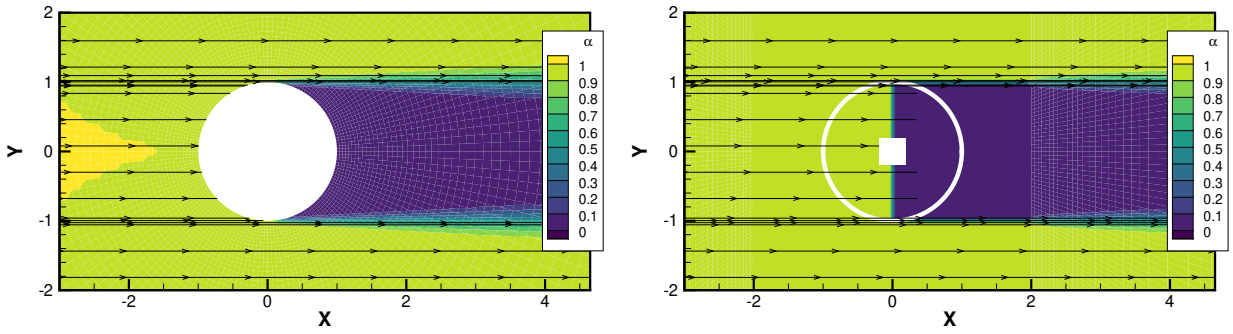


Figure 5.6 Analytical wall collection efficiency (β) compared to the Body-Fitted and Immersed boundary results

Note that for this test case the VP and VP-SSO methods are equivalent as the drag and gravity term are deactivated everywhere in the fluid and the solid zones. Thus only one curve labeled "immersed boundary" is illustrated.

The field values for the volume fraction of water (α) are illustrated in Figure 5.7 where the impingement ($\alpha > 0$) and shadow zones ($\alpha = 0$) are clearly visible. The immersed cylinder is represented by the white circle in Figure 5.7b. This figure shows that the combined mask function behaves as expected for this test case. The upstream solid portion of the cylinder is filled with droplets ($\alpha = \alpha_\infty$, impingement zone) while the downwind solid portion is empty ($\alpha = 0$, shadow zone). Some streamtraces seem to extend in the solid portion of the shadow zone but in this area $\alpha \approx 0$ and $\mathbf{v}_d \approx 0$. The streamtraces are displayed but carry no droplet mass and therefore no droplet is in fact re-injected in the computational domain.



(a) Body-Fitted mesh

(b) IB Volume Penalization

Figure 5.7 Volume fraction of water (α) field and droplet streamtraces for the cylinder with no drag nor gravity

This canonical test shows that the penalization method behaves properly. However, as the droplet field is a constant in the impingement zone, this test case is not representative of a real situation and cannot evaluate the quality of the data extraction at the immersed boundary. For a more realistic test, the drag term is reactivated in the next section.

Note that the white square in Figure 5.7b is required by the ice accretion suite as it expects a closed body (a closed wall boundary). However, it does not affect the quality of the solution in the field or in the solid. It can also be used to reduce the number of cells inside the solid and therefore the computation cost. In newer versions of IGLOO2D this dummy body is not mandatory.

5.5.2 Low Mach flow around a Cylinder

In this section, a mesh convergence study is performed on a cylinder of unit radius using a family of structured meshes for the BF solution and Cartesian grids for the IB solution. Some mesh characteristics are listed in Table 5.2 where five levels of refinement are used for both the structured and Cartesian meshes. The finest structured mesh (Level 5) is used to generate a reference solution.

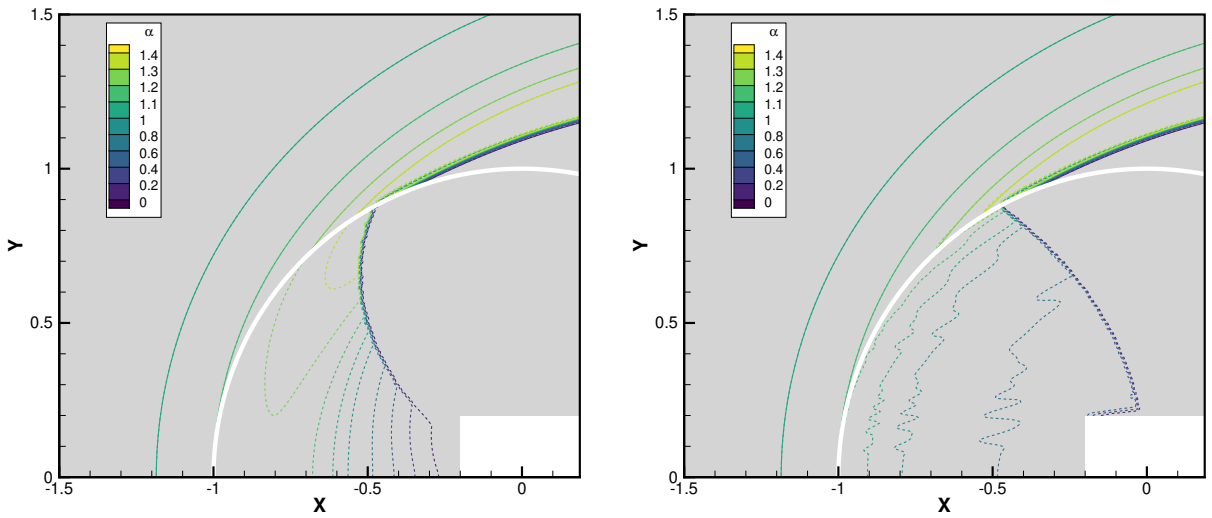
Table 5.2 Family of meshes

	Structured			Cartesian		
Refinement	Wall Δx [m]	N cell	$D/\Delta x$	Wall Δx [m]	N cell	$D/\Delta x$
Level 0	1.60e-1	1K	12.5	2.00e-1	∼2K	10
Level 1	8.00e-2	4K	25.0	1.00e-1	∼8K	20
Level 2	4.00e-2	16K	50.0	5.00e-2	∼32K	40
Level 3	2.00e-2	64K	100.0	2.50e-2	∼128K	80
Level 4	1.00e-2	256K	200.0	1.25e-2	∼512K	160
Level 5	5.00e-3	1024K	400.0	6.25e-3	∼2048K	320

To avoid possible inaccuracies from the penalized aerodynamic field, the analytical potential flow solution around the cylinder is used. Its definition can be found in the appendix. In this way, the order of convergence for the droplets can be determined independently of the flow solution.

The simulation parameters are summarized in Table 5.1. In order to avoid that the droplets behave like tracers ($Stk < 0.1$), the MVD is increased to $304.5\mu m$ (compared to $20\mu m$ for the NACA0012). This gives a Stokes number ($Stk \approx 9.7$) close to typical icing conditions, as computed for the NACA0012 case.

The field values for the volume fraction of water (α) are illustrated in Figure 5.8 where the VP and VP-SSO methods are compared to the body-fitted simulation. The finest meshes (refinement level 5) are used for the comparison, where both VP and VP-SSO methods seem to reproduce the BF solution in the fluid. However, the solution in the solid behaves differently with the VP method, exhibiting a smoother transition from the fluid to the solid zone compared to the VP-SSO. The data is interpolated at the IB with a stencil including both fluid and solid cells. Thus, although the solution on the fluid side is very similar, the differences on the solid side influence the data extraction at the immersed boundary.

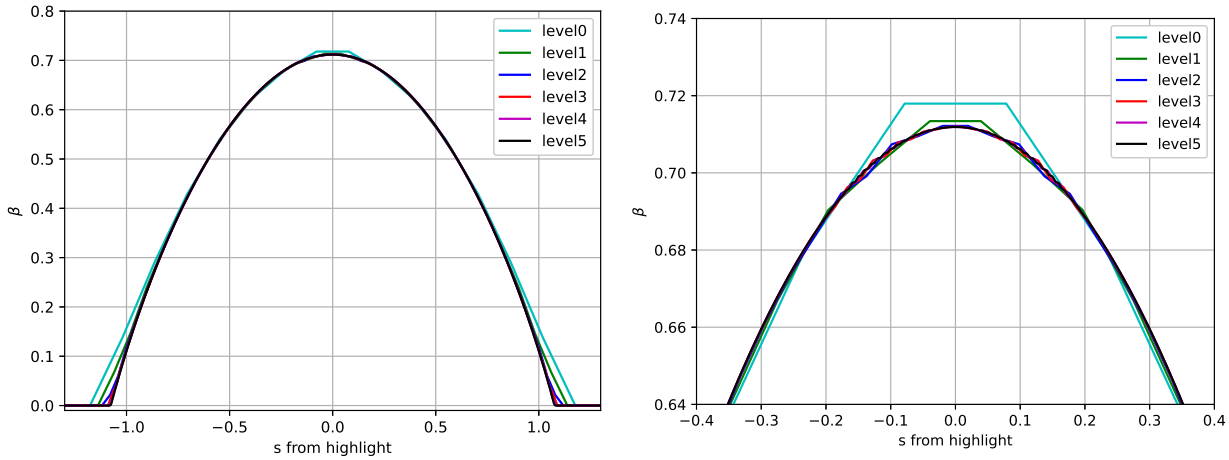


(a) Volume Penalization (VP)

(b) Volume Penalization Solid Source Off (VPSSO)

Figure 5.8 α field for the BF and IB methods. solid lines: BF, dashed lines: IB

To assess the quality of the data extraction, β is evaluated at the wall for all mesh refinement levels for the BF, VP and VP-SSO methods. On Figure 5.9, a mesh converged β distribution is observed from refinement level 4 for the BF method. It justifies the use of refinement level 5 as the reference mesh when comparing with the IBMs.

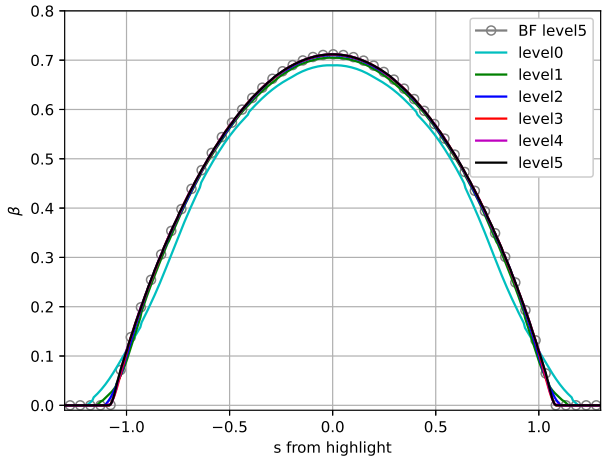


(a) Body-Fitted

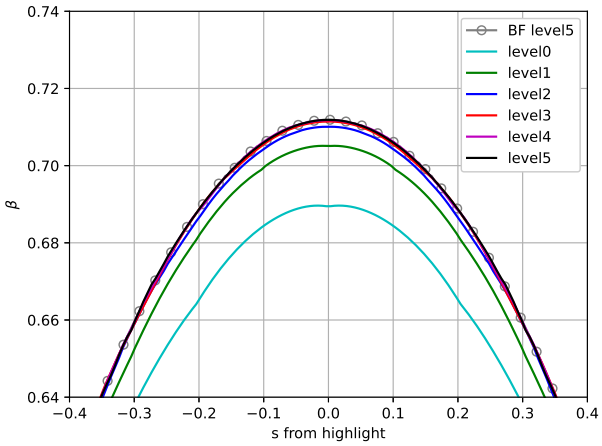
(b) Body-Fitted, LE Zoom

Figure 5.9 Collection Efficiency (β) with mesh refinement for the BF mesh.

Figure 5.10 shows the collection efficiency with mesh refinement for the VP method. The mesh convergence seems to be slower than for the BF simulation, especially in terms of the maximum β achieved. However, from mesh refinement level 4, the VP solution offers a good match with the reference solution and is also free of oscillations.

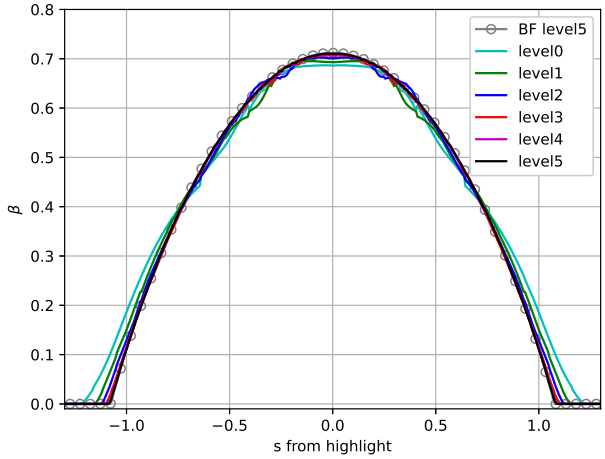


(a) Volume Penalization (VP)

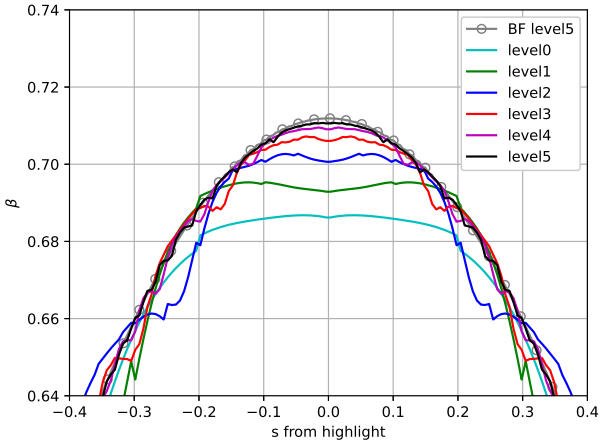


(b) Volume Penalization (VP), LE Zoom

Figure 5.10 Collection Efficiency (β) with mesh refinement for the VP method.



(a) Volume Penalization (VPSSO)



(b) Volume Penalization (VPSSO), LE Zoom

Figure 5.11 Collection Efficiency (β) with mesh refinement for the VPSSO method.

For the VP-SSO method (Figure 5.11), the β distribution also converges towards the BF reference solution. Contrary to the VP method, the VP-SSO solution exhibits oscillations.

The amplitude of these oscillations reduces with mesh refinement but they are still present for the finest mesh. Furthermore, the maximum β does not match the reference solution yet. This suggests that the VP-SSO method requires an even more refined mesh to obtain a mesh converged solution.

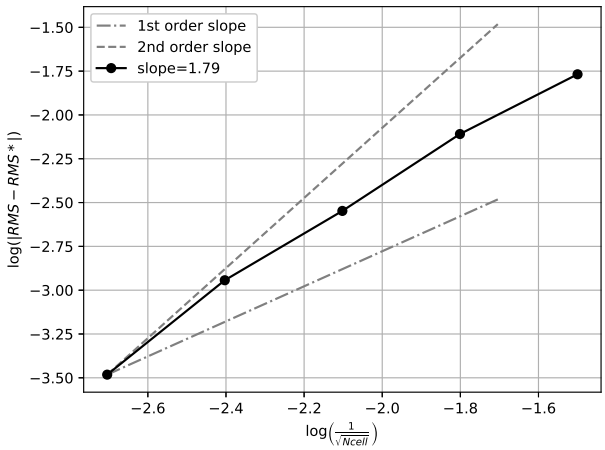
The idea behind the VP-SSO method is to avoid having to define an aerodynamic solution in the solid zone, which is only required for numerical purposes. This is achieved by deactivating the drag term in the solid. However, the physical flux is still computed in the solid which provides an extension of the droplet solution from the fluid to the solid. As illustrated in Figure 5.11b, this method fails to generate a smooth and accurate solution for β. This might be explained by the drag term being turned on and off near the immersed boundary in a staircase manner. On the other hand, the solution is much better when computing the drag term in the solid (VP method), for this particular test case. This seems logical as the drag then influences the droplet field more gradually (no staircase effect). Also, the activation of the physical source term allows a better continuity of the solution near the immersed boundary as previously shown in Figure 5.8).

The order of convergence is verified using local (Eq. (5.32)) and global (Eq. (5.33)) criteria. The local criterion compares the local collection efficiency (β_i) against the reference solution (body-fitted mesh, refinement level 5) and combines it using a root mean square (RMS). The global criterion is computed as the collection efficiency integral along the cylinder's surface (denoted β_{tot}).

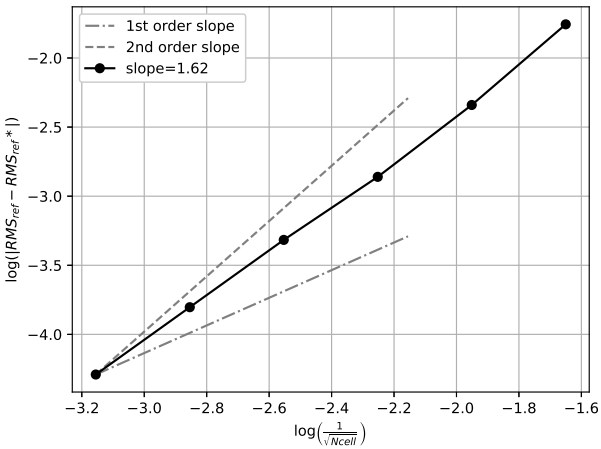
$$\mathrm{RMS}_{ref} = \sqrt{\frac{1}{N}\sum_{i}^{N}\left(\beta_i - \beta_{i,ref}\right)^2} \tag{5.32}$$

$$\beta_{tot} = \int_0^s (\beta_i) ds \tag{5.33}$$

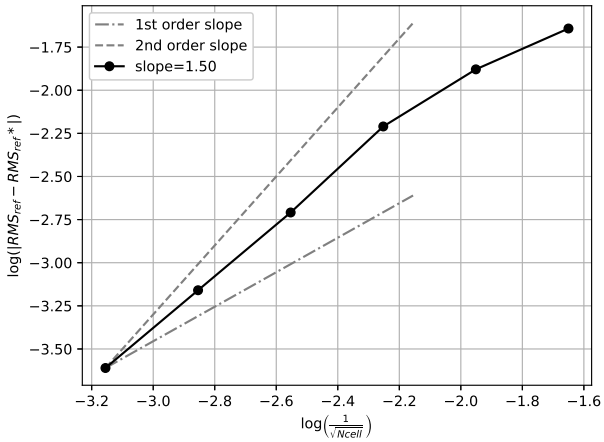
The order of convergence p is evaluated using the method described by [153] based on a criterion $\mathcal{F}$ which in this paper is either RMS_{ref} or β_{tot}. This requires a monotonic sequence of $\mathcal{F}$ on a minimum of three meshes (coarse, medium, fine). Then the order p can be solved for using a Newton method and the limiting value of $\mathcal{F}$ as $\Delta x \to 0$ (denoted $\mathcal{F}^*$) can be estimated.



(a) Body-Fitted



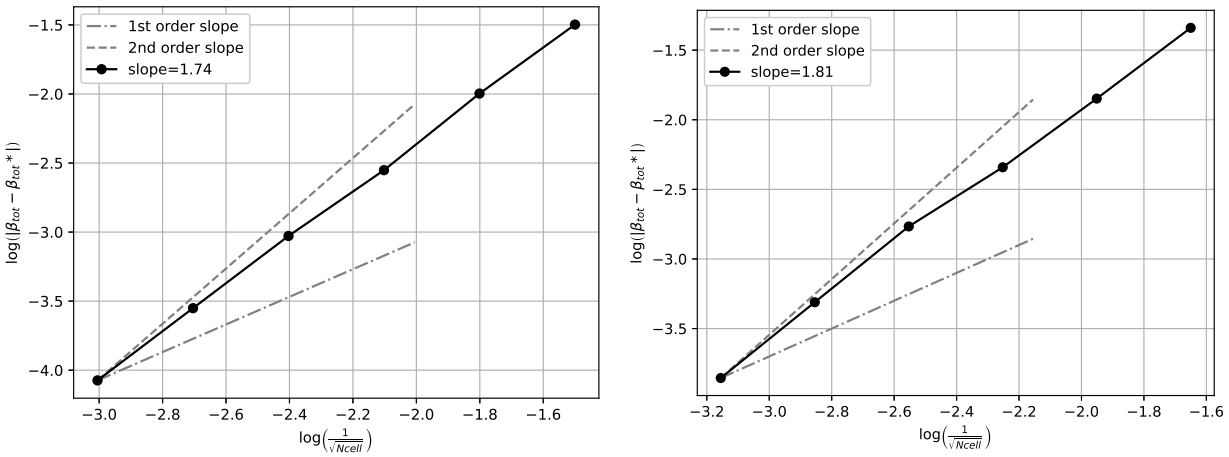
(b) Volume Penalization (VP)



(c) Volume Penalization (VP-SSO)

Figure 5.12 Order of convergence based on RMS_{ref}

The order of convergence is illustrated on Figures 5.12–5.13 along with the 1^{st} and 2^{nd} order theoretical slopes. Using the structured family of meshes (body-fitted), the order of convergence is $p > 1.7$ for both criteria which is close to 2^{nd} order accuracy. This gives a baseline on what to expect for the penalization methods. The VP method gets close to the BF order of accuracy with $p = 1.62$ when using RMS_{ref} and $p = 1.81$ when using β_{tot}. The VP-SSO method also provides an order of accuracy greater than unity ($p = 1.5$) for RMS_{ref} but does not perform as well as the VP method. The order of convergence is not shown for the VP-SSO method using the β_{tot} criterion since the sequence is not monotonic with mesh refinement. Thus, p cannot be computed with the current method.



(a) Body-Fitted

(b) Volume Penalization (VP)

Figure 5.13 Order of convergence based on the β_{tot}

Globally, the IBMs are approaching 2^{nd} order accuracy similar to the body-fitted approach. This was not expected as the volume penalization method usually limits the order of accuracy to 1^{st} order. However, it is logical to obtain more than 1^{st} order accuracy for the droplet equations as no penalization is applied in the impingement zone, where β is evaluated. The penalization terms are only applied in the shadow zone where $\beta = 0$, thus not affecting the order of convergence.

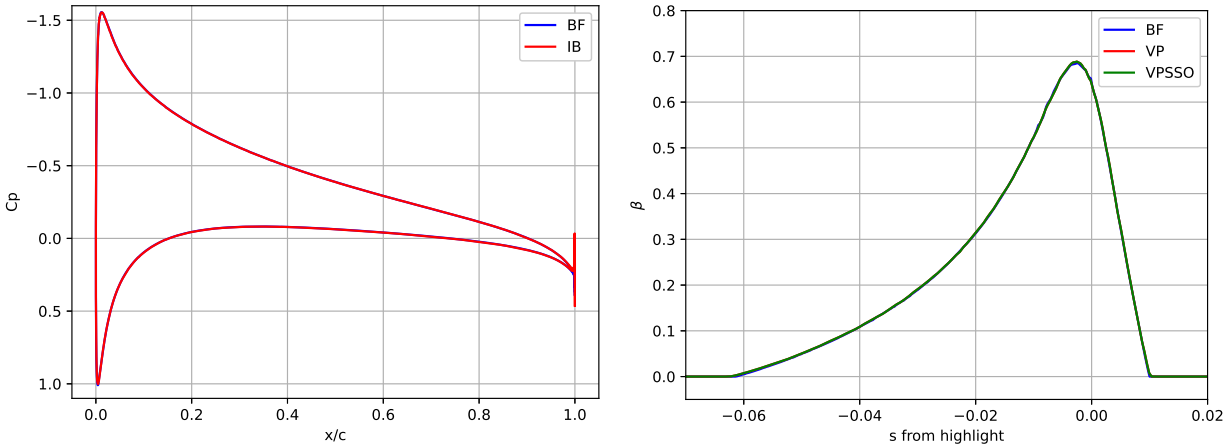
5.5.3 NACA0012 in icing conditions

In this section, a clean NACA0012 airfoil is used to assess the penalization methods for typical icing conditions. This test case uses an unstructured mesh made of triangles with a far-field located at 50 chords (c). The wall cell size is 2.5e-3c with refinements at the leading edge (size 5e-4c) and trailing edge (size 2.5e-4c). A linear growth is applied from the wall to the far-field (size: 4.0c) which gives a cell count of 77 172. For the immersed boundary mesh, the cell count is 112 172.

As the drag term is activated, the droplet trajectories are influenced by the aerodynamic solution. For this test case, it is evaluated using the penalized Euler equations [156] to deal with the immersed boundary. Using this approach, an aerodynamic solution is readily available in the solid ensuring continuity near the immersed boundary which is useful for the VP method. Note that the results presented in this section include the effect of both the penalized Euler [156] and droplet equations.

The wall pressure coefficients (Cp) are first compared against the body-fitted method to

assess the quality of the aerodynamic solution, where a good match can be observed Figure 5.14a. The comparison of the collection efficiency is presented in Figure 5.14b where a good match can also be observed between the body-fitted and the penalization methods. Different mesh refinements were also tested for the NACA0012 case but only the mesh converged solution are shown here. Figure 5.14b illustrates that with mesh refinement the BF and IB methods converge towards the same solution. Furthermore, smooth results are obtained for both the VP and VP-SSO methods. As a reminder, oscillations were observed with the VP-SSO on the cylinder case (Figure 5.11b).

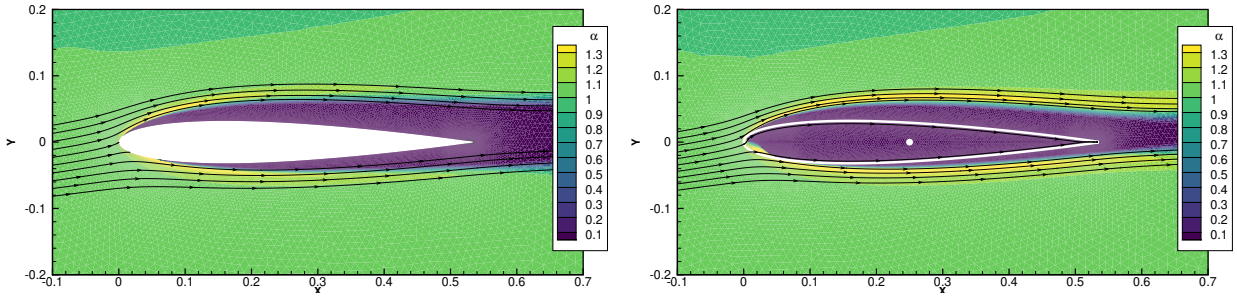


(a) Wall pressure coefficient (Cp)

(b) Wall collection efficiency (β)

Figure 5.14 Comparison of wall data for the NACA0012 airfoil

The α field is represented in Figure 5.15 for the body-fitted and immersed boundary simulations. It shows that the combined mask function behaves correctly not only for the cylinder but also for the NACA0012 case. In the impingement zone, the droplets travel inside the solid but their re-injection in the fluid is avoided by the application of the Dirichlet condition in the shadow zone. Again there is a white cylinder inside the solid zone which acts as a dummy body in Figure 5.15b. The immersed NACA0012 is represented in by the white line.

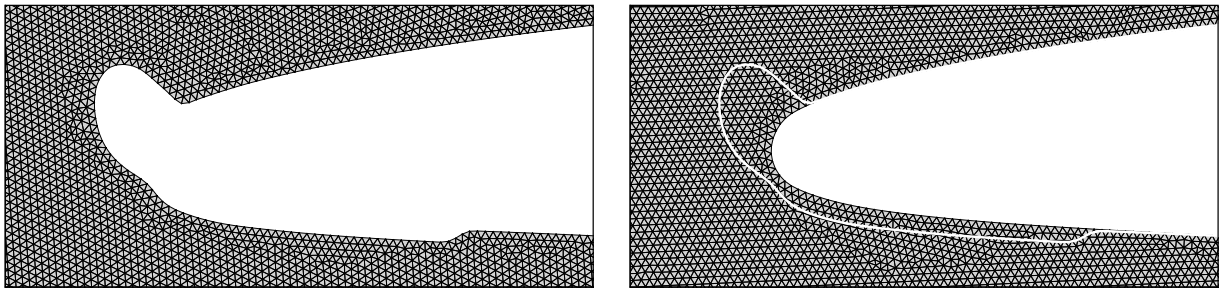


(a) Body-Fitted mesh (b) IB Volume Penalization

Figure 5.15 Volume fraction of water (α) field

5.5.4 High Curvature Ice Horn Case

In this section an ice accreted GLC305 airfoil is used to assess the penalization method on a more challenging geometry (Figure 5.16). This test case uses an unstructured mesh made of triangles with a far-field located at 20 chords (c). The wall cell size is 2.5e-3c with refinements at the leading edge (size 6.25e-4c) and trailing edge (size 2.5e-3c). A linear growth is applied from the wall to the far-field (size: 4.0c).

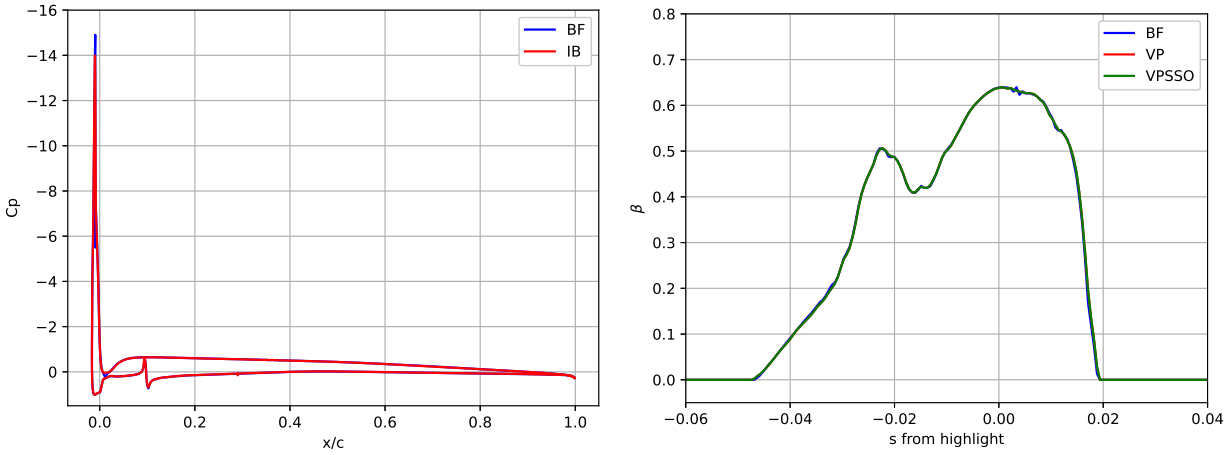


(a) Body-Fitted (b) Immersed Boundary

Figure 5.16 Mesh around the ice horn

For this test case, the aerodynamic field is again evaluated using the penalized Euler equation of [156]. The pressure distribution Cp from Figure 5.17a shows that a good match is obtained between the BF and IB methods. However the suction peak is slightly overestimated by the IB method (near $x/c = 0$). This does not seem to affect the collection efficiency (β) at the wall as shown on Figure 5.17b, where the curves are overlaid for the BF, VP and VP-SSO methods. Again the collection efficiency is free of oscillation for both the VP and VP-SSO solution. This shows that the penalization method developed for the droplet equations are

still able to reproduce the BF solution on airfoils and shows some potential for ice accreted airfoils.



(a) Wall pressure coefficient (Cp)

(b) Wall collection efficiency (β)

Figure 5.17 Comparison of wall data for the ice horn

5.6 Conclusion

This paper proposes a penalization method for the Eulerian simulation of droplet impingement. The application of a penalization method is not straightforward for the droplet equations as the required wall boundary condition changes along the immersed boundary and must be treated with care to avoid that droplets enter the solid in an impingement zone and then be re-injected in the computational domain. A technique based on a double mask function is suggested to treat the wall boundary correctly.

Amongst the suggested penalization methods (VP and VP-SSO), a better behavior is achieved using the VP method, which is shown to reproduce the body-fitted solution on all cases while also providing a smooth solution. The activation of the physical source term in the solid zone (VP method) help in retrieving a smooth solution across the IB thus increasing the quality of the data extraction. However, it requires the definition of a fictitious aerodynamic field in the solid. On the other hand, the VP-SSO method is independent of the fictitious aerodynamics in the solid zone and is able to provide similar results to the VP and BF methods in terms of accuracy and smoothness for the clean NACA0012 airfoil and the ice horn. However, it generates an oscillating β distribution for the cylinder case (with non-zero droplet drag). With mesh convergence the oscillations become less significant and the collection efficiency tends towards the BF solution. These results suggest that the activation of the physical

source term in the solid zone (VP method) is beneficial for some cases but is not always necessary.

The volume penalization (VP) is simple to implement, provide smooth results and is able to achieve second order accuracy like the body-fitted approach. Thus, it provides an interesting alternative to the ghost-cell approaches (a type of IBM) usually employed for the Eulerian simulation of droplet impingement.

CHAPTER 6 ARTICLE 3: AN IMMERSED BOUNDARY METHODOLOGY FOR MULTI-STEP ICE ACCRETION USING A LEVEL-SET

P. Lavoie, E. Radenac, G. Blanchard, Éric Laurendeau, and P. Villedieu, "An immersed boundary methodology for multi-step ice accretion using a level-set," Journal of Aircraft, 2021, (submitted).

6.1 Abstract

The numerical prediction of in-flight ice accretion involves a sequential call to different modules including mesh generation, aerodynamics, droplet trajectories, wall heat transfer, ice accretion and geometry update. Automation of this process is critical as these solvers are embedded in a time loop which is repeated several times to obtain an accurate ice shape prediction. The robustness of ice accretion tools is often limited by the difficulty of generating meshes on complex ice shapes and also by the geometry update which can exhibit overlaps if not treated properly. As a replacement to the usual body-fitted approach, this paper investigates the application of an immersed boundary method in the ice accretion framework to avoid the mesh generation step. A level-set method is also used for the geometry update to automatically handle pathological cases. The proposed methodology is tested on 2D rime and glaze ice cases from the 1^{st} AIAA Ice Prediction Workshop, showing good correspondence with the body-fitted approach. The new methodology also performs well for a 2D three-element airfoil configuration when a proper mesh refinement is used. The IBM combined with the level-set ice accretion provides a viable alternative to the body-fitted approach.

6.2 Introduction

Numerical tools for the prediction of ice accretion on aircraft are typically based on a quasi-steady assumption where modules are called sequentially and solved to steady state within a time-iterative scheme. The process is illustrated in Figure 6.1 where the modules are: (1) a mesh generation tool, (2) a solver for the aerodynamics, (3) a solver to obtain the droplet trajectories and impingement rates, (4) a solver to obtain the wall convective heat transfer (in the boundary layer), (5) a solver to perform a heat and mass balance applied to the deposited water to obtain the ice accretion rate and finally (6) a tool to update the geometry based on the ice thickness evolution. Modules (1) to (6) are embedded in a time loop for

which the total ice accretion time is divided in time steps (multi-step) generating successive layers of ice (multi-layer). When using Body-Fitted meshes, a mesh update is required with each new ice layer. It can be repeated several times in order to obtain the final ice shape prediction. This leads to additional costs related to the mesh update and additional difficulty in updating the ice shape which can exhibit unphysical surface overlaps in concave regions when using a Lagrangian approach (displacement of surface mesh nodes, a method which is usually employed in ice accretion codes).

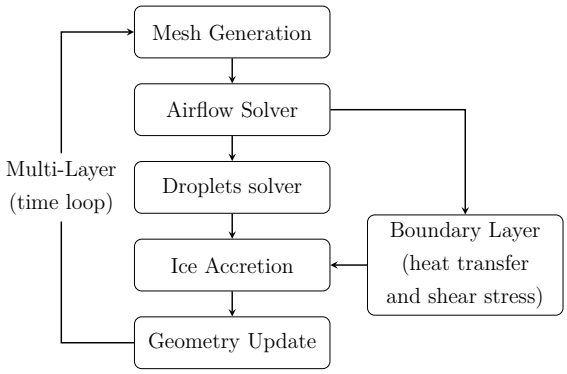


Figure 6.1 Sequential call to modules in multi-step icing simulations.

This paper is concerned by the methods required to automate icing suites using a multi-step approach, more specifically by the geometry and mesh updates. An IBM can be used to help automate ice accretion tools by avoiding the re-meshing or at least, minimizing the work required to adapt the mesh. An initial attempt is performed in [151], where a penalization method (a specific type of immersed boundary method) is applied to the aerodynamic (Euler equations) and droplet solvers (Eulerian formulation). The 1^{st} step of the multi-step process is performed using a Body-Fitted mesh while for the subsequent steps, the ice shape is immersed on the initial mesh. A geometric approach was used to evaluate the signed distance field required by the penalization method. Furthermore, a Lagrangian node displacement approach was used to update the geometry.

This paper extends the contribution of [151] with several key features. First, an improved penalization method suitable for ice horn accretion is applied to the Euler equations [156]. Second, a level-set approach [8] is implemented in the IBM multi-step ice accretion process to solve the issues related to unphysical geometry update, replacing the Lagrangian geometry update. Third, several icing cases are examined, including 2D cases from the AIAA Ice Prediction Workshop [163].

The paper first presents the ice accretion suite used as the development platform, IGLOO2D

[12]. Then, implementation details for the penalization and level-set methods are covered: preprocessing, penalization of the Euler equations, penalization of the droplets solver, extraction of the surface data and implementation of the level-set method. A section discusses the benefits of using the level-set method in the IBM ice accretion framework on a manufactured case. Then, rime and glaze ice cases from the AIAA Ice Prediction Workshop (IPW) are used for verification where the Body-Fitted and penalized solutions are compared using the multi-step process. Additional validations are performed for the ice accretion around a NACA0012 airfoil. Finally, ice accretion simulations are performed on a three-element airfoil before conclusions are drawn.

6.3 Methodology

The 2D ice accretion suite IGLOO2D [12] is used as the development environment. In IGLOO2D, different types of solvers are available for each module but only the ones used in this paper are discussed. The unstructured mesh generation is handled by GMSH [164]. The aerodynamic field is evaluated using the Euler equations and the convective heat transfer is evaluated using a simplified integral boundary layer method (SIM2D) [12]. For the droplet trajectories and impingement evaluation, the Eulerian solver is selected. The ice accretion solver is based on a Messinger-type mass and energy balance to obtain the ice thickness. Finally, the ice geometry is generated by a Lagrangian displacement of the surface nodes. The Eulerian evolution of the geometry using a level-set is also treated in this paper.

The modules can be classified either as volume or surface solvers. The aerodynamics (EULER2D) and the Eulerian droplet trajectories (TRAJE2D) are solved on 2D volume meshes. On the other hand, the simplified integral boundary layer method (SIM2D) and the ice accretion (MESSINGER2D) are solved on 1D surface grids.

For the application of the IBM, the suggested approach is to start the multi-step ice accretion process from a standard BF mesh, thus keeping the original BF solution for the 1^{st} ice layer (as well as for the clean areas of the surface for the following steps). Usually, the BF mesh is updated to match the new ice geometry for each subsequent step. With our IBM, the volume mesh update is avoided and a penalization method is applied to the volume solvers (airflow and droplets trajectory) to impose the correct boundary conditions on the immersed boundary which arbitrarily cuts through the mesh. The ice surface is, however, re-meshed to retain an adequate representation of the ice shape for the IBM. The use of the penalization method requires some modifications to the ice accretion suite : the addition of a preprocessing step, the modification of the volume solvers and the extraction of surface data, as highlighted in red in Figure 6.2. These modifications are discussed in the following sections along with

the integration of the level-set method in the multi-step process.

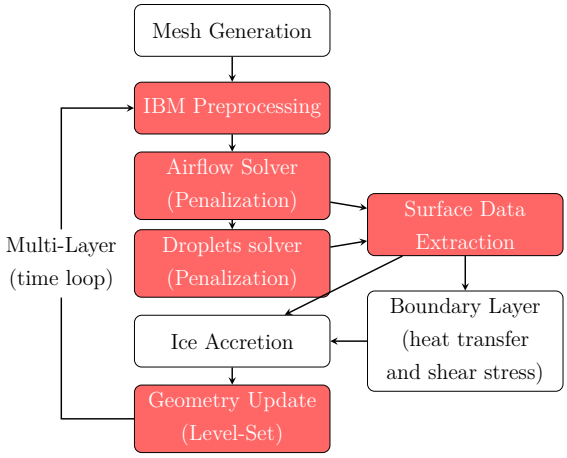


Figure 6.2 Sequential call to modules in a multi-step icing simulation using an immersed boundary method.

6.3.1 Immersed Boundary Pre-Processing

In this paper, both an explicit and implicit definition of the IB are required. The multi-step ice accretion process starts from a BF volume mesh, thus initially providing a surface mesh which represents the solid-air interface. It can be interpreted as an Immersed Boundary which correspond to the BF surface for the first step. This explicit definition must be conserved throughout the multi-step ice accretion process in order to use the surface solvers (i.e., ice accretion, boundary layer). On the other hand, the penalization methods (a type of IBM) implemented in IGLOO2D use a signed distance field (implicit definition) to obtain information about the interface at any point in the volume mesh.

The IB preprocessor evaluates the signed distance field (ϕ) by first detecting the inside (solid) and outside (fluid) cells. Knowing the list of edges defining a closed immersed boundary, a ray casting algorithm can be used for this matter [129]. Once this information is known, it can be used to determine the sign of the signed distance field where $\phi > 0$ in the fluid and $\phi < 0$ in the solid. The distance is evaluated by taking advantage of the available explicit definition of the interface. For each cell, a geometric approach determines the minimum projected distance to the list of edges (or faces in 3D) defining the IB. Then, the normals to

the IB ($\mathbf{n}_\phi$) and its curvature (κ) can be evaluated from the signed distance field (ϕ) as:

$$\mathbf{n}_\phi = -\frac{\nabla\phi}{||\nabla\phi||} \tag{6.1}$$

$$\kappa = \nabla \cdot \mathbf{n}_\phi \tag{6.2}$$

Notice that $\mathbf{n}_\phi$ is defined to point towards the solid, contrary to the usual definition, which is useful in the implementation of the penalization methods. An example of signed distance field around a clean NACA23012 airfoil is illustrated in Figure 6.3 along with the normals to the wall ($\mathbf{n}_\phi$). Here, the signed distance field is strictly positive because it is evaluated on a BF mesh where the contour $\phi = 0$ is the surface of the airfoil.

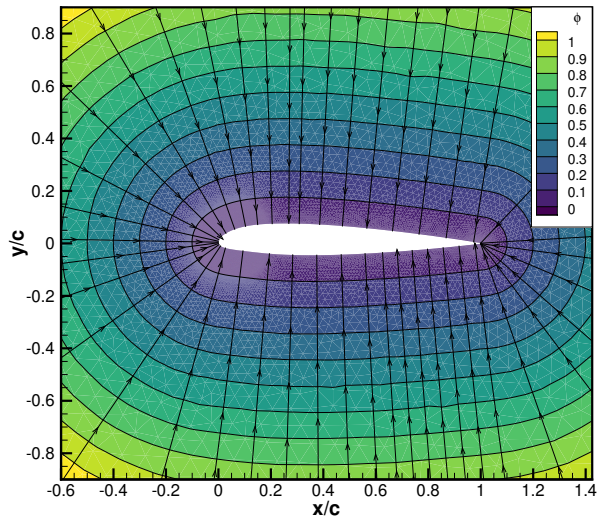


Figure 6.3 Signed distance contours (ϕ) and surface normals ($\mathbf{n}_\phi$) for a clean NACA23012 airfoil

Although the volume mesh update is avoided when using the IBM, the IB is re-meshed at the pre-processing phase for the 2^{nd} ice accretion step and further. This is possible because the immersed boundary discretization (surface mesh) is independent of the volume mesh. The surface re-meshing is done using GMSH where a B-spline is fitted through the discrete list of nodes defining the ice shape. The nodes are then redistributed according to a user-specified characteristic mesh size. This provides a surface mesh discretization which is very close to what is obtained with the BF ice accretion process.

6.3.2 Volume Penalization Method

With a volume penalization method, the boundary conditions are applied by the addition of source terms in the continuous form of the governing equations in order to enforce the desired condition at the immersed boundary. The source terms are turned on if the computational volume (a cell) is located inside the solid zone and turned off if in the fluid zone. Hence the governing equations are solved as usual in the fluid but penalized in the solid. The source terms are turned on and off using a mask function (χ) which takes the form of a sharp Heaviside function (Eq. (6.3)).

$$\chi = \begin{cases} 0 & \phi \geq 0 \text{ (fluid)} \\ 1 & \phi < 0 \text{ (solid)} \end{cases} \tag{6.3}$$

For the aerodynamics, the penalization of the Euler equations is performed using the CBVP-Hs method of [156]. This approach enforces the no-penetration velocity (slip wall, Eq. (6.4)) and uses the normal momentum relation to account for the wall curvature in the pressure extrapolation (Eq. (6.5)). The conservation of total enthalpy (Eq. (6.6)) and entropy (Eq. (6.7)) are also enforced across the immersed boundary to close the system.

$$(\mathbf{v} \cdot \mathbf{n}_\phi)\mathbf{n}_\phi = 0 \tag{6.4}$$

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} P = \kappa \rho ||\mathbf{v}||^2 \tag{6.5}$$

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} H = 0 \tag{6.6}$$

$$\mathbf{n}_\phi \cdot \boldsymbol{\nabla} s = 0 \tag{6.7}$$

This method was found to perform well for ice shapes exhibiting high curvature such as ice horns. The set of penalized Euler equations is given by Eq. (6.8) where the penalization terms enforcing Eqs. (6.4)–(6.7) are gathered on the right-hand side (RHS).

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + (1-\chi)\,\boldsymbol{\nabla} \cdot (\rho \mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi \cdot \boldsymbol{\nabla}\rho - \kappa \frac{\rho^2}{\gamma P}||\mathbf{v}||^2\right) \\
\frac{\partial (\rho \mathbf{v})}{\partial t} + (1-\chi)\,\boldsymbol{\nabla} \cdot (\rho \mathbf{v} \otimes \mathbf{v} + P\mathbf{I}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi \cdot \boldsymbol{\nabla}(\rho \mathbf{v}) + \kappa \rho \mathbf{v}\left(1 - \frac{\rho}{\gamma P}||\mathbf{v}||^2\right)\right) - \frac{\chi}{\eta}\rho(\mathbf{v} \cdot \mathbf{n}_\phi)\mathbf{n}_\phi \\
\frac{\partial (\rho E)}{\partial t} + (1-\chi)\,\boldsymbol{\nabla} \cdot ((\rho E + P)\mathbf{v}) &= -\frac{\chi}{\eta_c}\rho \mathbf{n}_\phi \cdot \boldsymbol{\nabla} H - \frac{\chi}{\eta}\rho(\mathbf{v} \cdot \mathbf{n}_\phi)^2
\end{aligned} \tag{6.8}$$

In Eq. (6.8), $\mathbf{v}$ is the air velocity, $\mathbf{I}$ is the identity tensor, E is the total energy, η and $1/\eta_c$ are penalization parameters that can be respectively interpreted as a characteristic time and

a characteristic velocity.

For the Eulerian droplet equations, the penalization method of [165] is used. When droplets impinge the body ($\mathbf{v}_d \cdot \mathbf{n}_\phi > 0$), no penalization is applied and the physical equations are solved in the solid. The droplets are thus allowed to cross the immersed boundary and enter the body. However, when the droplets enter the computational domain from the solid ($\mathbf{v}_d \cdot \mathbf{n}_\phi \leq 0$) a boundary condition is applied on the primitive variables (Eq. (6.9)), enforcing a null flux and avoiding re-injection of the droplets.

$$\left.\begin{array}{ll} \alpha & = 0 \\ \mathbf{v}_d & = 0 \end{array}\right\} if \quad \mathbf{v}_d \cdot \mathbf{n}_\phi \leq 0 \tag{6.9}$$

To translate this behaviour to the droplet equations using penalization terms, the usual mask function (χ, Eq. (6.3)) is multiplied by a droplet mask function (χ_d, Eq. (6.10)), ensuring the penalization term is only active in the solid if the droplets are reinjected in the fluid.

$$\chi_d = \begin{cases} 0 & \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi \geq 0 \text{ (impingement)} \\ 1 & \alpha \mathbf{v}_d \cdot \mathbf{n}_\phi < 0 \text{ (re-injection)} \end{cases} \tag{6.10}$$

The set of penalized droplet equations is given by Eq. (6.11), where the influence of gravity is neglected and the penalization terms are highlighted in red.

$$\begin{aligned} \frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d) &= -\frac{\chi \chi_d}{\eta} \alpha \\ \frac{\partial (\alpha \mathbf{v}_d)}{\partial t} + \boldsymbol{\nabla} \cdot (\alpha \mathbf{v}_d \otimes \mathbf{v}_d) &= \frac{C_D Re_d}{24 Stk} \alpha (\mathbf{v}_a - \mathbf{v}_d) - 2 \frac{\chi \chi_d}{\eta} \alpha \mathbf{v}_d \end{aligned} \tag{6.11}$$

In Eq. (6.11), α is the non-dimensional volume fraction of water, $\mathbf{v}_d$ is the non-dimensional droplets velocity, $\mathbf{v}_a$ is the non-dimensional air velocity and C_D is the droplets drag coefficient. The droplets Reynolds number (Re_d) and the Stokes number (Stk) are:

$$Re_d = \frac{\rho_a ||\mathbf{v}_a - \mathbf{v}_d|| D_d}{\mu} \tag{6.12}$$

$$Stk = \frac{\rho_d D_d^2 U_\infty}{18 L \mu} \tag{6.13}$$

where D_d is the droplet diameter, μ the dynamic viscosity of air and L a characteristic dimension (e.g., the chord length for an airfoil). The drag model of Schiller and Naumann

[159] is used for the droplets which are assumed to remain spherical:

$$C_D = \begin{cases} \frac{24}{Re_d}(1 + 0.15Re_d^{0.687}) & Re_d \leq 1000 \\ 0.4 & Re_d > 1000 \end{cases} \tag{6.14}$$

The penalization parameters must be small ($\eta \ll 1$, $\eta_c \ll 1$) to accurately enforce the boundary conditions, which leads to a stiff system of equations. The penalization terms are thus treated implicitly when solving the system of equations for both the aerodynamics and droplet trajectories.

6.3.3 Surface Data Extraction

Relevant surface information from the volume solvers (aerodynamics and droplet trajectory) must be communicated to the surface solvers (boundary layer, ice accretion) at each step of the multi-step loop (e.g., pressure, velocity, droplet impingement rate). However, the penalization method does not explicitly provide the data on the Immersed Boundary (IB). Instead, the variables are known in the surrounding cells and an additional extraction step is thus required to recover the surface data.

In this paper, the data is interpolated on the IB using a weighted least square approach. The nearest cell to the interpolation point is first detected. Then, all the cells sharing a node with the identified cell are flagged as neighbours and used for the interpolation. The penalization methods used in this paper are designed to fill the solid cell with valid data. The solid cells are included in the interpolation stencil, hence the need for methods ensuring a controlled continuity of the solution across the solid-fluid interface, as described in [156, 165]. The interpolation uses an inverse distance weight with a smoothing parameter to avoid dividing by zero when the interpolation point and stencil points are too close. The weight between a cell centre J (part of the stencil) and the interpolation point P is evaluated as:

$$w_J = \frac{1}{\sqrt{||\mathbf{r}_{PJ}||^2 + \epsilon^2}} \tag{6.15}$$

where $||\mathbf{r}_{PJ}||$ is the distance between P and J. The smoothing parameter is selected as $\epsilon = 0.5\Delta x_J$ with Δx_J the characteristic size of cell J,

6.3.4 Geometry Update via the Level-Set method

A Lagrangian approach can be used to update the geometry according to the normals to the wall ($\mathbf{n}$, pointing towards the fluid) and the ice thickness (h_{ice}) provided by the ice accretion

solver. A simple node update can be performed as:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + h_{ice}\mathbf{n} \tag{6.16}$$

where $\mathbf{x}_{new}$ and $\mathbf{x}_{old}$ are respectively the new and old node locations. This type of approach does not naturally handle the overlaps that can occur near concave region and requires methods for collision detection and front merging to obtain a usable surface mesh. A simple fix can be implemented in 2D as described in [9]. However, it does not directly translate to a 3D implementation which involves more complex geometric operations on a 2D surface mesh. Alternatively, the level-set method can be used to update the geometry. This was done for instance by [8] where the level-set equation (Eq. (6.17), [99]) is used with an icing velocity field ($\mathbf{V}_{ice}$) and solved on the volume mesh.

$$\frac{\partial \phi}{\partial t} + \mathbf{V}_{ice} \cdot \nabla \phi = 0 \tag{6.17}$$

This approach has the benefit of being valid for both 2D and 3D simulations. It also naturally handles the issues related to the geometry update such as geometry overlaps. Here, the level-set method reuses the signed distance field (ϕ) computed at the IB pre-processing step. The interface (IB or BF) is represented by the contour $\phi = 0$ and is advanced in time (Eq. (6.17)) to generate the ice shape, following the icing velocity vector field $\mathbf{V}_{ice}$. In this paper, the level-set is discretized using a 2^{nd} order scheme in time (Heun's method) and space (upwind with MUSCL extrapolation). The following sections describe a method to retrieve the icing velocity field and discuss the need for a re-initialization step in the advection of the level-set.

6.3.4.1 Velocity Propagation

The icing velocity magnitude ($V_{ice,surf}$) can be computed from the ice accretion time (Δt_{ice}) and the ice thickness (h_{ice}) provided on the surface mesh by the thermodynamics solver.

$$V_{ice,surf} = \frac{h_{ice}}{\Delta t_{ice}} \tag{6.18}$$

However, $V_{ice,surf}$ must be propagated in the volume mesh in order to perform the level-set advection (Eq. (6.17)). To obtain a behaviour similar to the Lagrangian node displacement approach (Eq. (6.16)), the icing velocity is propagated from the surface mesh in the normal direction, producing constant velocity bands. A PDE-based approach (Eq. (6.19)) is used to propagate the information from the surface ($V_{ice,surf}$) to the field (V_{ice}) following the normal

direction to the surface.

$$\frac{\partial V_{ice}}{\partial t} = \text{sign}(\phi)\mathbf{n}_\phi \cdot \nabla V_{ice} \tag{6.19}$$

When the surface mesh corresponds to the Body-Fitted mesh boundary, $V_{ice,surf}$ is imposed as a Dirichlet boundary condition using ghost cells and is propagated in the fluid zone to obtain (V_{ice}). When the ice shape is immersed in the mesh (IB), the surface no longer corresponds to the mesh boundaries. For instance, this occurs from the 2^{nd} ice layer onward in the multi-step icing process. In this situation, a band of cells in the vicinity of the interface is initialized by a nearest neighbour search, taking advantage of the explicit definition of the interface. These cells are then frozen (no update) so they can act as ghost cells when solving Eq. (6.19) on both sides of the IB. The update is prevented by setting the Right Hand Side (RHS) of Eq. (6.19) to zero for the frozen cells. The propagation Eq. (6.19) accounts for the sign of ϕ in order to propagate $V_{ice,surf}$ from the band of initialized cells towards the fluid ($\phi > 0$) and solid zones ($\phi < 0$).

Once the icing velocity magnitude is known in the volume mesh, the vector field is set as:

$$\mathbf{V}_{ice} = -V_{ice}\mathbf{n}_{\phi,0} \tag{6.20}$$

where $\mathbf{n}_{\phi,0}$ represents the normal to the initial contour $\phi = 0$ (before the advection process begins). In other words, the icing velocity field remains fixed during the advection of the level-set. An example of propagated icing velocity field is illustrated in Figure 6.4, showing the constant velocity bands in the normal direction to the interface.

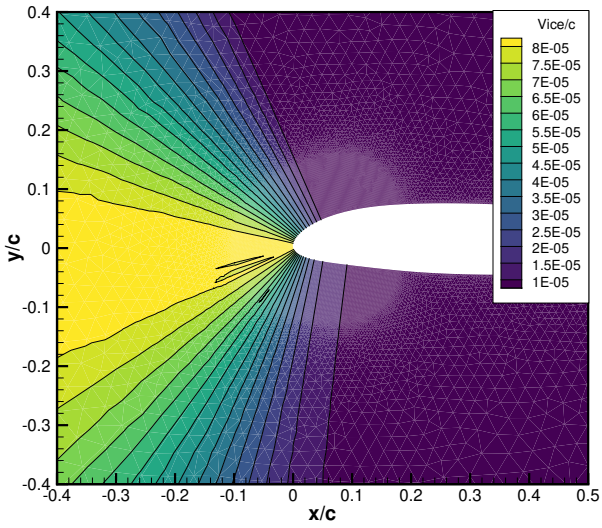


Figure 6.4 Example of a propagated icing velocity field for a clean NACA23012 airfoil, body-fitted surface. Coordinates and velocity non-dimensionalized by the chord (c).

6.3.4.2 Level-Set Advection and Re-Initialization

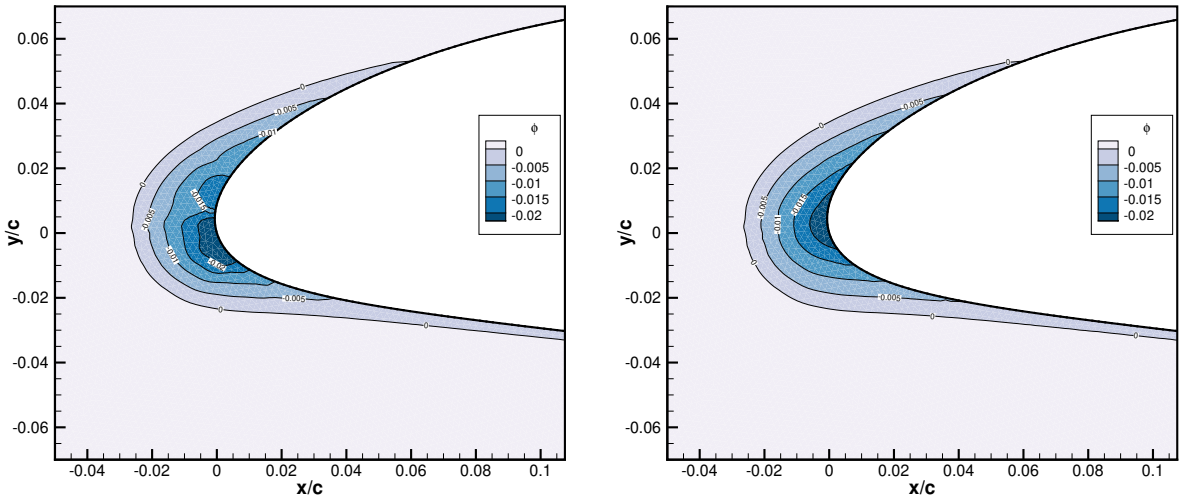
While the contour $\phi = 0$ is advected using the level-set equation (6.17), ϕ does not conserve the properties of a signed distance field [137]. A re-initialization of the level-set is thus performed (i.e., the signed distance field is re-evaluated). This could be done by reusing the geometric approach from the pre-processing step. Because the new location of the IB is only known via its implicit definition at this stage, this would imply the application of a contour extraction technique to obtain an explicit definition of the interface (new surface mesh). The signed distance field is instead updated using the re-initialization equation [99], as follows:

$$\frac{\partial \phi}{\partial t} = \mathcal{S}(\phi_0)\left(\mathbf{n}_\phi \cdot \nabla\phi + 1\right) \tag{6.21}$$

$$\mathcal{S}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \epsilon^2}} \tag{6.22}$$

This equation incorporates a smoothed sign function $\mathcal{S}(\phi_0)$ which is based on the signed distance before re-initialization (ϕ_0). According to [137], it ensures that ϕ remains unchanged at the interface during the re-initialization process. In practice, numerical experiments showed the introduction of wiggles in the contour $\phi = 0$ when using this approach, an undesirable behaviour as a surface mesh is to be constructed from this extracted interface. To ensure the interface remains exactly at the same location, the idea used for the velocity propagation is repurposed here: freezing the update of a band of cells in the vicinity of the interface. Again,

it is done by setting the RHS to zero in Eq. (6.21) for the frozen cells. This approach follows the assumptions that ϕ remains close to a signed distance field in the vicinity of the interface. In this paper, two iterations of Eq. (6.21) are performed at every time step of the level-set advection process (Eq. (6.17)). An example of level-set advection is shown on Figure 6.5, where the ϕ contours are displayed inside the ice shape only. Without re-initialization (Figure 6.5a), the signed distance field is distorted inside the solid while activating the re-initialization (Figure 6.5b) provides a more regular and sensible solution.



(a) Without re-initialization

(b) With re-initialization

Figure 6.5 Example of level-set advection from a clean NACA23012 airfoil, single step ice accretion on a BF mesh

6.3.5 Surface Mesh Extraction

Once the level-set advection is completed and the signed distance field is re-initialized, the surface mesh extraction can be performed. It consists of two parts: (1) the contour extraction providing an explicit definition of the IB from the level-set and (2) the meshing of the surface (i.e., using GMSH). The first part is performed in the level-set module (geometry evolution solver) while the second part is performed when preprocessing the IB.

A surface discretization can be obtained by performing the extraction of the contour $\phi = 0$. Note that the contour extraction is not performed before the re-initialization because of our general method which combines a Body-Fitted surface for the clean geometry and an Immersed Boundary for the ice shape. It is desired to extract a contour which matches the BF surface where there is no ice while the IB is normally extracted for the iced zones. To do

so, a check based on ϕ is performed at the BF surface during the extraction process. If the Body-Fitted surface is in the solid zone and far from the contour $\phi = 0$ (e.g., leading edge in Figure 6.5a), the signed distance field can be so distorted that the contour may be falsely detected at the BF surface. The re-initialization of the level-set solves this issue and thus helps the contour extraction.

For the contour extraction, well-known methods such as the marching cubes [104], marching tetrahedra [105] and marching squares (a 2D equivalent of the marching cubes) can be used. In this paper, 2D unstructured meshes made of triangles are used which allows for a simple contour extraction method. We consider four possible configurations for the contour intersection with a triangular cell: edge to edge, edge to vertex, vertex to vertex or vertex only (Figure 6.6).

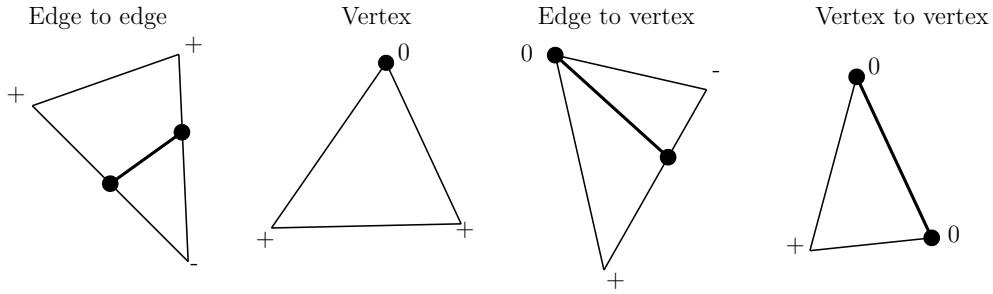


Figure 6.6 Example of contour intersection with a triangle cell: edge and vertex cases

These cases are all handled automatically by performing an edge-based interpolation assuming a single intersection point per edge. The process marches from cell to cell and adds consecutive intersection points to a linked list, forming a surface discretization. Tested edges are tagged along the way to avoid adding duplicates to the list. Once the marching process can no longer find any intersection on untested edges, the contour is completed. An edge is intersected by the contour if there is a sign change in ϕ between its two vertices. If ϕ at one vertex is below a specified threshold, the intersection is assumed to occur at the vertex and no interpolation is made. In this case, all the edges sharing the vertex are tagged as tested. This approach retains the discretization of the body-fitted surface where there is no ice accretion ($\phi \approx 0$) and perform a more classical contour extraction for the immersed boundary. It also directly provides an ordered list of points (surface mesh) for each body when dealing with a multi-element configuration. Note that in IGLOO2D, ϕ is reconstructed at the vertices from a weighted least square interpolation using the cell-centre solution. An example of the marching process illustrated in Figure 6.7 where the vertices are identified as positive, negative or zero and the extracted contour is illustrated in red.

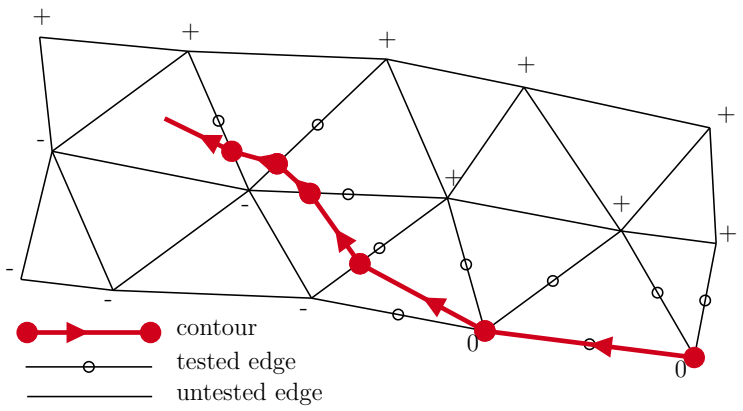
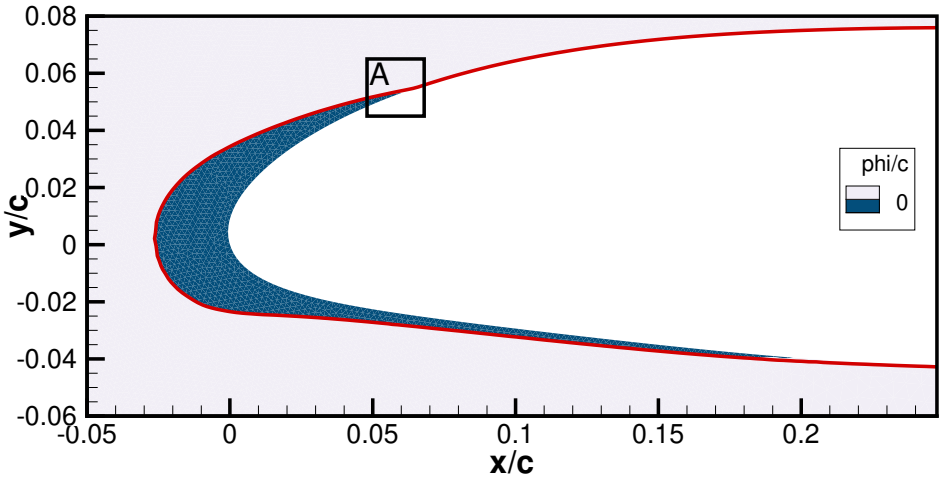


Figure 6.7 Example for the contour extraction marching process

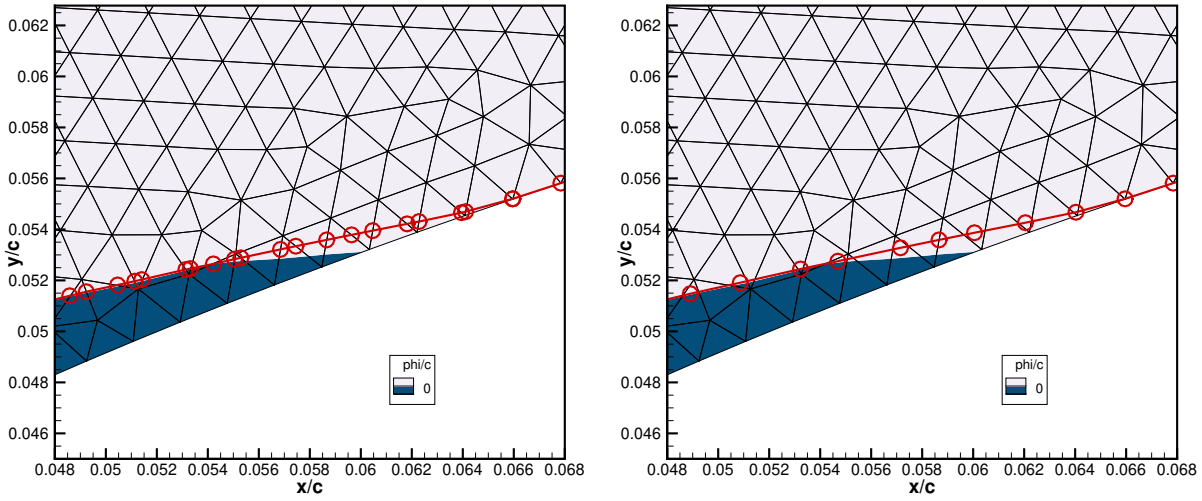
The extraction process usually produces an irregular discretization where nodes can be very close to each other when the edge intersection is detected near a vertex. To help retrieve a more uniform and smoother surface mesh, nodes are merged if they are too close and inserted if they are too far apart. The merge and insertion process is performed by an arithmetic average followed by a correction to bring the node back on the level-set. Following the idea presented in [108], the correction takes the form:

$$\mathbf{x}_{corrected} = \mathbf{x}_{merged/inserted} + (\phi - \phi_{target})\psi\mathbf{n}_{\phi} \tag{6.23}$$

where $\phi_{target} = 0$ and ψ is a relaxation parameter set to unity but that can be reduced to avoid erroneous corrections (e.g., point near the wrong contour if multiple contours are involved). An example of contour extraction is provided in Figure 6.8a near the leading edge of an iced NACA23012. The effect of node merging and insertion is illustrated in Figures 6.8b–6.8c. Note that this node correction process is not mandatory as the surface is later re-meshed using GMSH. However it was found to improve the quality of the resulting mesh.



(a) Leading edge view



(b) Close-up of window A, without node correction

(c) Close-up of window A, with node correction

Figure 6.8 Example of a level-set extraction on an iced NACA23012 airfoil. red: surface extraction; blue: ice

6.4 Ice Accretion Results

In this section, the new ice accretion framework using the IB and level-set methods is assessed. The objective is to reproduce the ice accretion results obtained with a classical BF approach while improving the robustness of the numerical tool (e.g., no failure). In order to demonstrate the benefits of using the level-set approach, ice accretion over a manufactured

ice shape is first performed using the level-set method and compared to the Lagrangian node displacement method. Then, rime ice *case 241* and glaze ice *case 242* from the 1^{st} AIAA Ice Prediction Workshop (IPW, [163]) are tested. These cases are respectively *run ED1977* and *run ED1978* from [166], with slightly corrected icing conditions. Additional ice accretion cases from [12] (cases 001, 003 and 004) are also tested to further demonstrate the behaviour of the IBM. Finally, the new framework is tested on the multi-element McDonnell-Douglas LB606b Airfoil (MDA) [19] to illustrate the flexibility of the method on complex high-lift systems. The simulation parameters are summarized in Table 6.1.

Table 6.1 Simulation Parameters

	Rime 241	**Glaze 242**	**Case 001**	**Case 003**	**Case 004**	**Multi-Element**
Geometry	NACA23012	NACA23012	NACA0012	NACA0012	NACA0012	MDA
Chord [m]	0.4572	0.4572	0.5334	0.5334	0.5334	0.9144
AoA [deg]	2.0	2.0	4.0	4.0	4.0	8.0
Mach	0.325	0.315	0.325	0.317	0.317	0.27
$\mathbf{P_{static}}$ [kPa]	92.528	92.941	101.325	101.325	101.325	101.325
$\mathbf{T_{static}}$ [K]	250.15	266.05	250.7	262.3	262.3	268.2
LWC $[g/m^3]$	0.42	0.81	0.55	1.0	0.6	0.6
MVD [μm]	30.0	15.0	20.0	20.0	15.0	20.0
Icing Time [s]	300	300	420	231	384	360
Roughness (ks) [mm]	0.4572	0.4572	0.5334	0.5334	0.5334	0.9144

In this section, two methods are available for the representation of the ice shape (IBM or BF) and two for the geometry update (node displacement or Level-Set). This makes four possible combinations. When simply referring to the Immersed Boundary Method, the use of the level-set method is implied. Similarly, when referring to the Body-Fitted method, the use of the Lagrangian node displacement approach is implied (the standard approach in the icing community). In addition, the calculations are carried out with IGLOO2D. The default options described in [12] were used for the Body-Fitted approach, in particular for the MESSINGER2D solver and the boundary-layer solver SIM2D. Regarding the meshes, unstructured grids generated by GMSH were systematically used. The wall mesh size is in the range of 1e-3 to 5e-3 chords (with a refinement in the range of 5e-4 chords for blunt trailing edges). These mesh sizes are fairly representative of default mesh sizes used in IGLOO2D. They generally allow obtaining a good trade-off between solution accuracy and computational time.

For all the calculations, the wall mesh size is kept constant near the leading edge and extended over 0.75 chords as shown in figure Figure 6.9. This is required when using the IBM in order to avoid the re-meshing during the multi-step process while maintaining an equivalent wall

cell size compared to the BF method (with re-meshing).

A multi-step approach is adopted, using 2 to 10 steps. For the IBM approach, the calculations are performed by changing the calculation strategy for the volume solvers EULER2D and TRAJE2D (penalization) and for the ice shape transportation (level-set), all other parameters remaining the same.

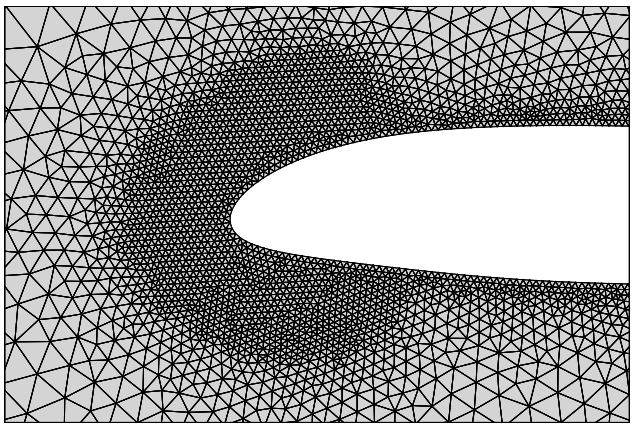


Figure 6.9 Example mesh around a NACA23012 with an extended refinement zone near the leading edge

6.4.1 Manufactured Ice Shape

In order to clearly show the behaviour of the level-set method against the usual Lagrangian node displacement approach, a manufactured ice shape is used with a fixed ice accretion rate (thickness and time). The ice accretion time is set to $400s$ and the ice thickness is enforced to $0.02m$ for every surface node with coordinate $x < 0.05m$.

This manufactured geometry was first presented in [167] and is generated from a NACA0012 airfoil with added artificial ice near the leading edge. The three-horn configuration was selected to obtain multiple flow recirculation zones and create a difficult situation for the ice growth solver because of the presence of highly concave and convex features.

On Figure 6.10, the ice accreted three-horn geometry is illustrated with the enforced ice accretion thickness. The level-set solution is represented in blue, indicating the zone with $\phi < 0$. The ice shape generated by the Lagrangian node displacement is shown as a solid black line, where the geometry overlaps can be seen near concave regions of the geometry. The contour $\phi = 0$ is extracted by our edge marching method and represented by the red line with markers. As observed in Figure 6.10, the level-set method automatically handles the geometry overlaps and the extracted contour provides an explicit surface mesh discretization

that can be used in the multi-step ice accretion process.

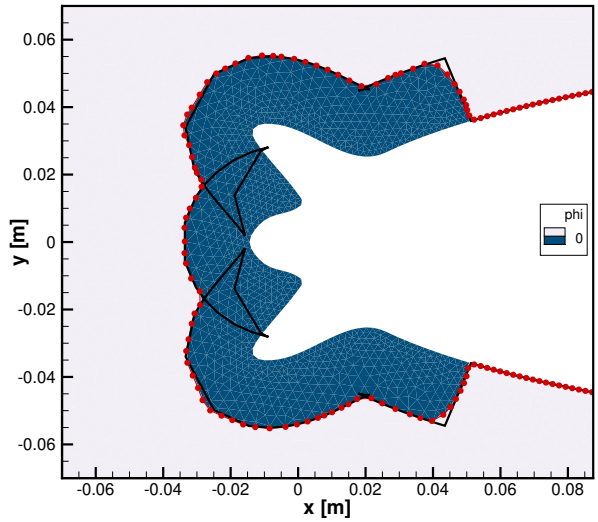
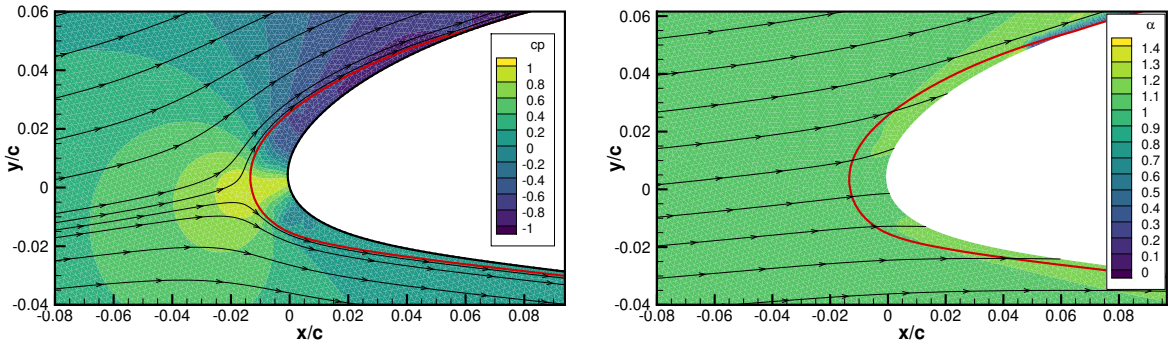


Figure 6.10 Comparison between Lagrangian node displacement and Level-Set approach with contour extraction

6.4.2 Rime Ice 241

For the results presented in this paper, only the ice shape is used as an Immersed Boundary and the clean geometry is still treated using a Body-Fitted approach. As an illustration, the aerodynamic and droplet fields for the rime ice case are shown in Figure 6.11 where the immersed boundary (the ice shape) is represented by the red line and the solid body is white.



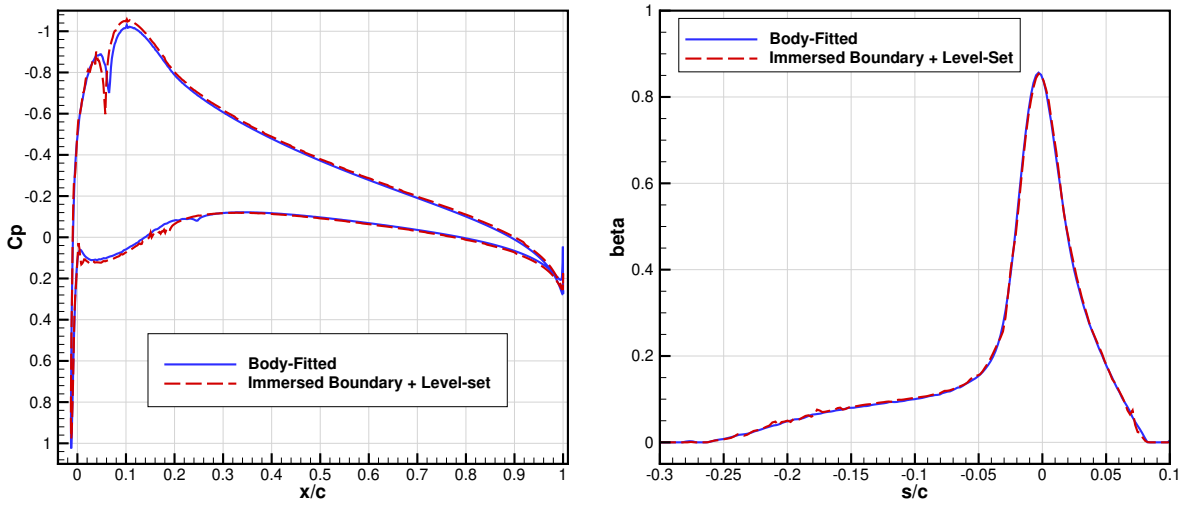
(a) Pressure coefficient

(b) Volume fraction of water (α)

Figure 6.11 Rime ice case 241: solution from the volume solvers around the 1^{st} ice layer of a two-layer simulation.

In this section, two-step and 10-step ice accretion simulations are performed for the rime ice case 241. The results are compared between the Body-Fitted and IB methods. Here the geometry update is done using a Lagrangian node displacement for the BF approach and using the level-set for the IBM. The wall mesh size is about 2e-3 chords with a refinement to 5e-4 chords at the trailing edge.

In Figure 6.12, the pressure coefficients (Cp) and collection efficiency (β) are compared for the BF and IB methods on the 1^{st} ice layer of a two-step simulation. Ideally, the IB method should reproduce the results obtained with a BF approach. Figure 6.12a illustrates a slight mismatch in Cp near the point of maximum suction. Nonetheless, the collection efficiency is very close between the two methods (Figure 6.12b). As rime ice accretion is mostly governed by the collection efficiency, it generates very similar ice shapes for the BF and IB methods despite the difference in pressure coefficients (Figure 6.13a). The ice shapes are also in good agreement with the experimental results. The experimental ice shape is the so-called MCCS (Maximum-Combined-Cross-Section) [168] derived by the experimentalists from the ice scans (it is more or less the envelope of the ice shape).



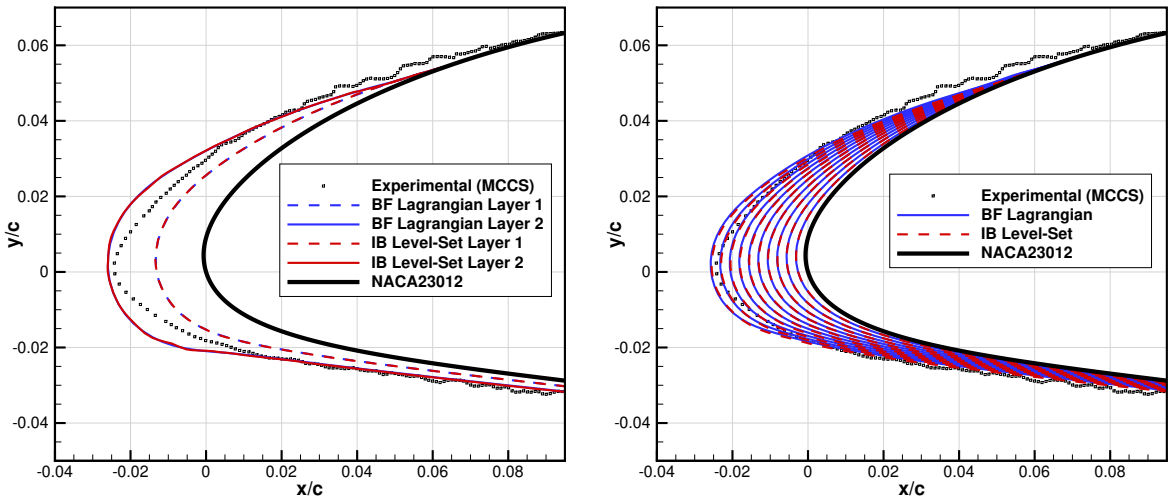
(a) Pressure coefficient (Cp)

(b) Collection efficiency (β)

Figure 6.12 Rime ice case 241 – comparison of wall surface data on the 1^{st} ice layer of a two-layer simulation for the BF and IB methods.

Since the simulation starts from a BF mesh, the 1^{st} step is not affected by the IBM and thus, the 1^{st} ice layer should be the same for both methods. However, a difference might be introduced by the geometry evolution solver which can use either a level-set with contour extraction or the Lagrangian node displacement approach. On Figure 6.13a, a two-step

ice accretion prediction is made where the 1^{st} ice layer is illustrated with a dashed line. As the generation of the 1^{st} ice layer is not influenced by the IBM, the Lagrangian and Eulerian (level-set) geometry updates can be compared, showing negligible difference. Thus, discrepancies observed in Figure 6.12 for the surface data can be attributed to the IB method and not to the level-set approach.



(a) 2 layers

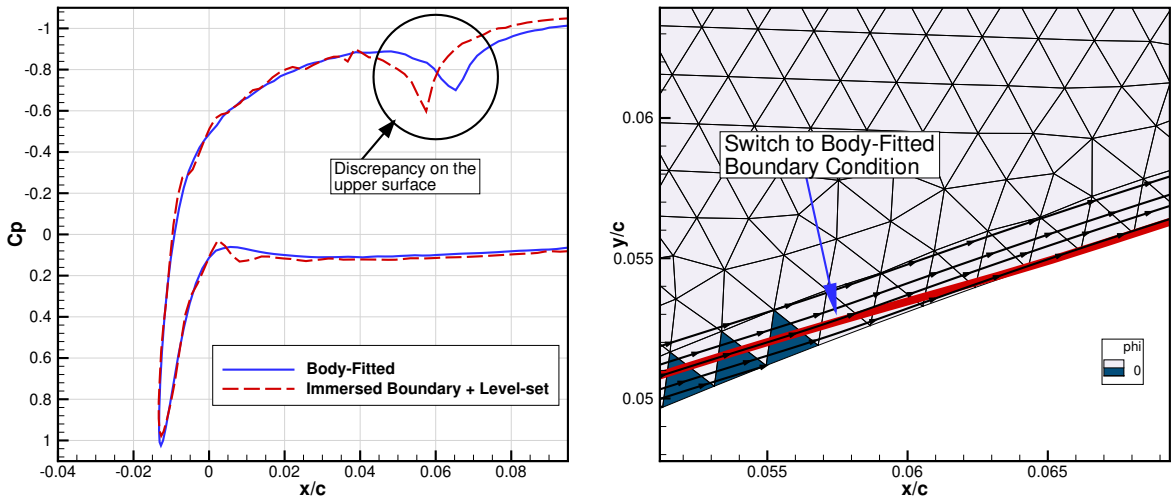
(b) 10 layers

Figure 6.13 Rime ice case 241 – multi-step ice shape predictions

A close-up of the leading edge Cp distribution is given in Figure 6.14a to highlight the discrepancy observed near $x/c = 0.06$. The difference between the two solutions can be explained by the interaction between the body-fitted wall and the IBM. Figure 6.14b provides a view of the airfoil's upper surface near the location of interest. For the rime ice case 241, the ice (illustrated in red) is getting thinner as we approach the impingement limits and eventually it merges with the body-fitted surface. The issue comes from the volume penalization method which, in our case, use a 1^{st} order implementation of the boundary condition. The cells are penalized if $\phi < 0$ at the cell centre and the usual physical equations are solved otherwise. The boundary conditions are applied at the cell centre, regardless of the location of the immersed boundary within the cell. Thus, when the ice shape is thin enough (Figure 6.14b), the cells are no longer penalized. In turn, a slip velocity is applied relative to the body-fitted wall instead of a velocity tangential to the ice shape. This premature switch to a BF wall boundary condition explains the difference in Cp near $x/c = 0.06$ in Figure 6.14a. The issue could be solved by implementing a 2^{nd} order discretization of the penalization terms or by performing a local mesh refinement to obtain a better representation of the ice shape near

the impingement limits. However, the current implementation is still able to provide a good prediction of the ice shape in comparison with the BF results (Figure 6.13).

For a 10-layer ice shape prediction (Figure 6.13b), the solution is still in good agreement with the experimental data for both methods. Increasing the number of steps reduces the thickness of each ice layer and this might affect the behaviour of the penalization method for the same reason described earlier (1^{st} order discretization of the penalization terms). For instance, the penalization method might effectively see the same geometry for 2 consecutive ice layers even though the ice shape has actually moved. This typically occurs if the ice layer is too thin relative to the mesh cells. For the 10-step simulation presented here, the mesh cell size is about the same as the thickness of a single layer, providing good results.



(a) Pressure coefficient (Cp) – leading edge close-up

(b) Boundary condition switch between IB and BF. Red line: ice shape. Blue fill: penalized cells

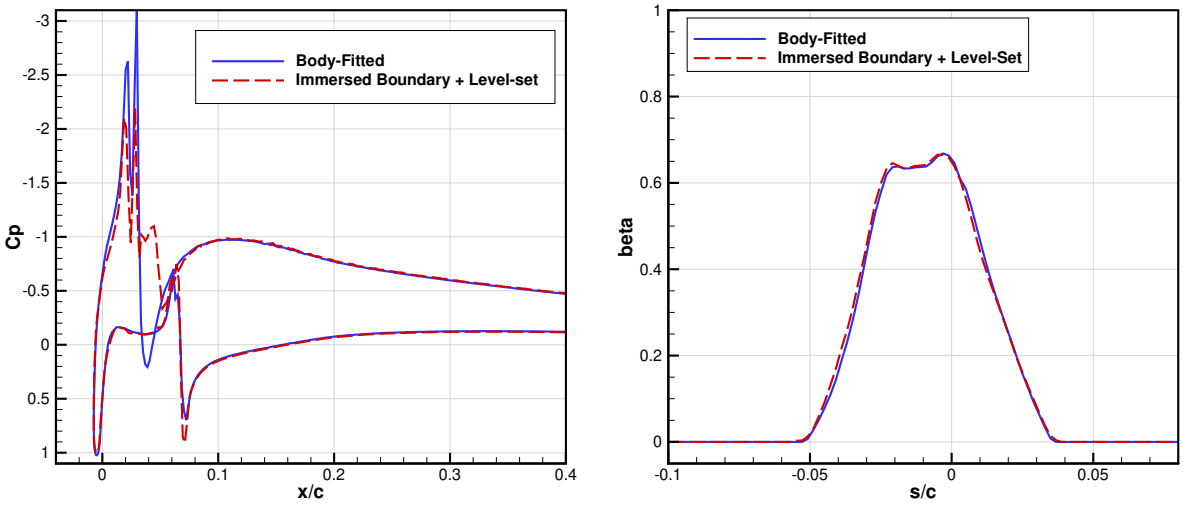
Figure 6.14 Rime ice case 241 – illustration of the Cp discrepancy near $x/c = 0.06$, two-step ice accretion

6.4.3 Glaze Ice 242

For the glaze ice case, the mesh characteristics are the same as for the rime ice case 241. The wall mesh size is about 2e-3 chords with a refinement to 5e-4 chords at the trailing edge.

For this case, there is again a slight mismatch on the Cp distribution (Figure 6.15a), but a very similar collection efficiency for both methods (Figure 6.15b). As glaze ice accretion is sensitive to the heat transfer coefficient which is in turn driven by the aerodynamics, the

mismatch in Cp might explain the slight difference observed on the ice shape Figure 6.16a. The effect of using the Lagrangian node displacement vs. the level-set approach can again be estimated by analyzing the 1^{st} ice layer on Figure 6.16a, where a negligible difference is observed. It suggests that the difference in Cp and β is due to the IBM.

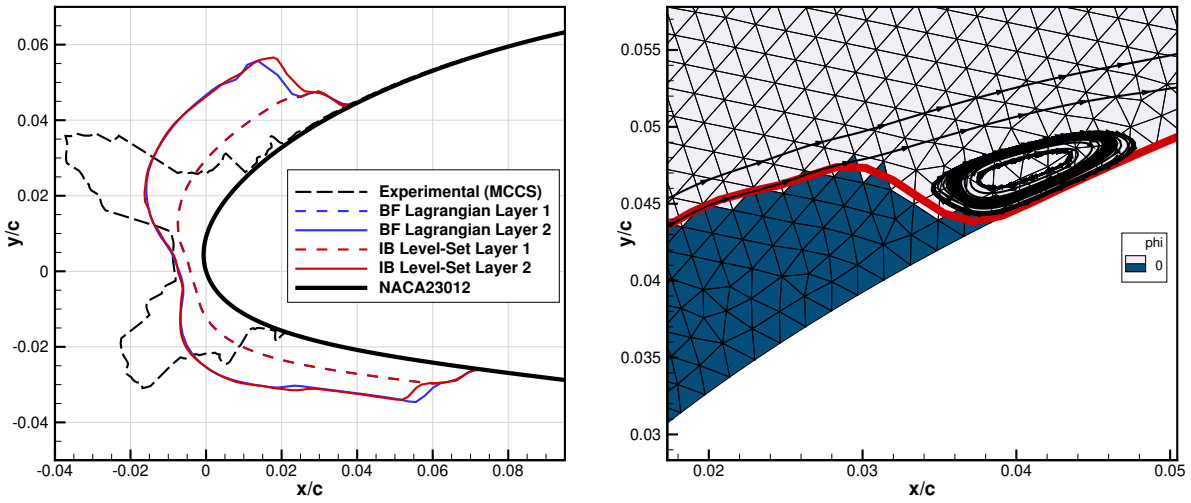


(a) Pressure coefficient (Cp)

(b) Collection efficiency (β)

Figure 6.15 Glaze ice case 242 – comparison of wall surface data on the 1^{st} ice layer for the BF and IB simulations.

By observing Figure 6.15a, the Cp distribution corresponds well between the BF and IB methods for $x/c < 0.02$ and $x/c > 0.06$. The zone where the discrepancy occurs is located near the ice accretion limits where the ice shape stops sharply. For the IBM, this results in a detached flow with a recirculation zone (Figure 6.16b) while it is not the case for the BF method, explaining the difference. Here, the comparison is made between the two methods with equivalent mesh size. However, this result suggests that the penalization method require a finer mesh near curved features to be equivalent to the BF approach.

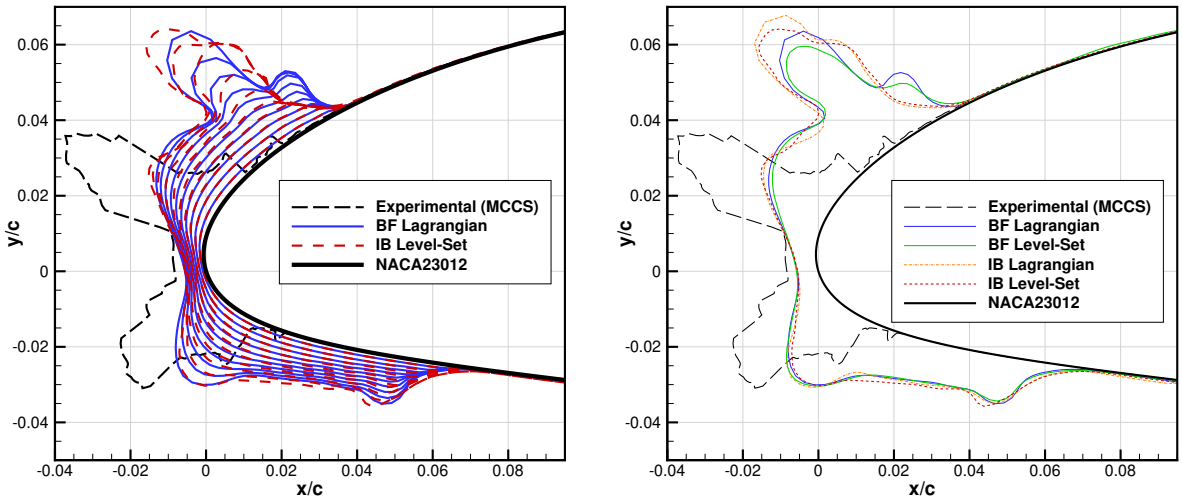


(a) Ice shape comparison (b) Upper surface detached flow (IB Level-Set)

Figure 6.16 Glaze ice case 242 – two-step ice accretion

Although the ice shape prediction for the 2^{nd} layer is similar for both methods, it does not reproduce the experimental measurements (Figure 6.16a). Note that icing experiments carry large uncertainties as well as spanwise variations [169, 170]. However, the divergence from the experimental ice shape seems too large to be attributed only to these uncertainties. Some tests were performed by refining the mesh and manually increasing the wall roughness (by a factor 2), without significant improvement. Here, the use of a droplet size distribution might help in obtaining a prediction towards the experimental ice shape. This option was, however, not tested as it is not yet available for use with our penalization method.

By increasing the number of ice layers to 10 (Figure 6.17a), the ice shape prediction is still far from the experimental results. Moreover, an ice horn is created but not in the same location. When comparing the BF and IB methods, the ice shape is similar for the most part, but with a larger difference near the ice horn (where the effect of the aerodynamics becomes more dominant on the ice accretion). The difference in ice shape is due to the combined effect of the penalization and level-set methods compared to the BF and Lagrangian approach (standard approach). In Figure 6.17b, all four combination of methods are shown for the 10^{th} ice layer only. The figure illustrates that the use of the level-set method has only a limited impact while the IB methods have a larger effect on the difference in ice shape. This is similar to the observation made for the two-step ice accretion simulation, where the penalization method requires a finer mesh near curved features to be equivalent to the BF solution. Using an equivalent cell size, the IBM however provides a good approximation.



(a) Comparison over 10 layers

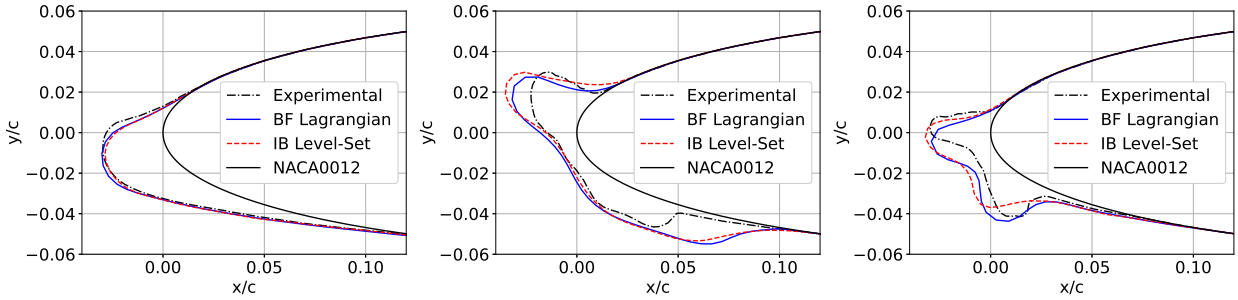
(b) Comparison between all the combinations, 10^{th} layer

Figure 6.17 Glaze ice case 242 – 10-step ice accretion

6.4.4 Additional Cases on a NACA0012

In this section, 10-step ice accretion calculations are performed on cases 001, 003 and 004 from [12] to further assess the behaviour of the IBM. The calculations are performed on a coarser mesh (wall mesh size of 5e-3), but it is still representative of typical ice accretion simulations with IGLOO2D.

When comparing the ice shape prediction obtained from the IB and the BF methods, a good match is observed for the rime ice case 001 (Figure 6.18a), but a larger difference is seen for the glaze ice cases 003 and 004 (Figures 6.18b–6.18c). This is in line with the observation made in the previous sections. Glaze ice shape are more sensitive to the airflow solution and a perfect correspondence is not obtained for the wall data between the two methods (e.g., Cp distribution, Figure 6.15a). The solution can be improved by refining the mesh. Nonetheless, using the IB and the level-set methods still provides a good estimation of the ice shapes when compared to the experimental data (Figure 6.18), even on coarser meshes.



(a) Case 001 (b) Case 003 (c) Case 004

Figure 6.18 Additional cases on a NACA0012 airfoil, 10-step ice accretion

6.4.5 Three-Element Airfoil (MDA)

In this section, ice accretion is performed on the three-element McDonnell-Douglas Airfoil (MDA, Figure 6.19) using the icing conditions provided in [19]. This test case is selected to show the flexibility of the immersed boundary and level-set methods.

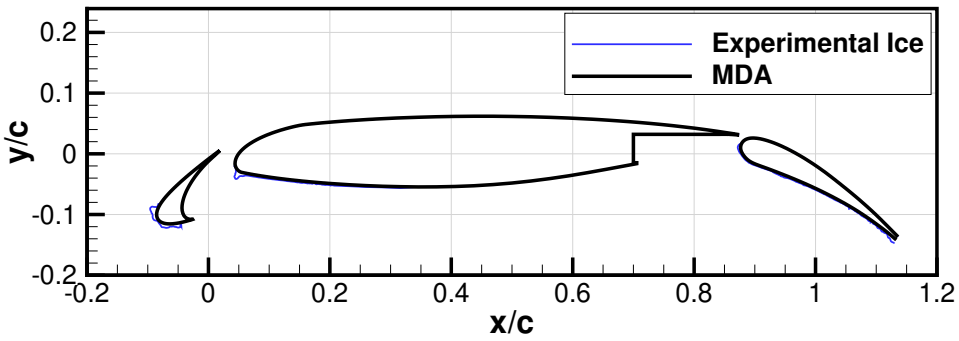


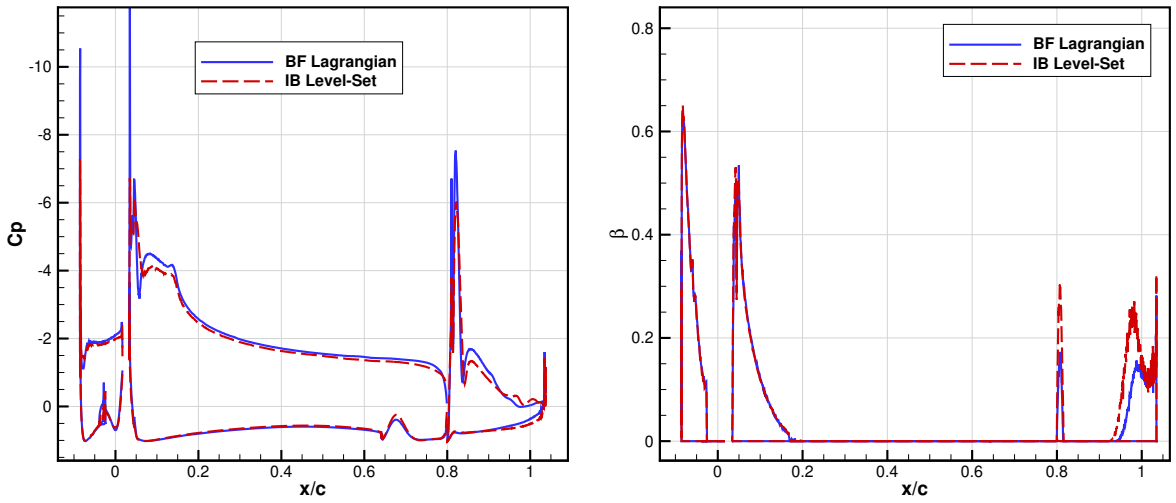
Figure 6.19 Global view of the McDonnell-Douglas LB606b airfoil and its experimental ice shape

A two-step ice accretion simulation is performed with both the BF and IB methods. Here the objective is to see if the IB method can reproduce the BF solution on this more challenging configuration. To do so, a mesh refinement was performed to obtain similar pressure coefficients and collection efficiency on the 1^{st} ice layer (2^{nd} time step), as shown in Figure 6.20. The wall cell sizes of the resulting mesh are summarized in Table 6.2. A finer mesh is required on this test case due to the flow separation downwind of the flap. Because a Euler flow solver is used (inviscid), this flow detachment is very sensitive to the mesh size. The current mesh set-up allowed the IB and BF to behave in a similar way (e.g., similar onset of

the flow detachment).

Table 6.2 Wall mesh characteristics for the three-element airfoil in terms of the chord ($\mathbf{\Delta x/c}$)

	Slat	**Main**	**Flap**
Wall	1e-3	1e-3	2.5e-4
Leading Edge	1e-3	2e-3	5e-4
Trailing Edge	1e-3	1e-3	5e-4



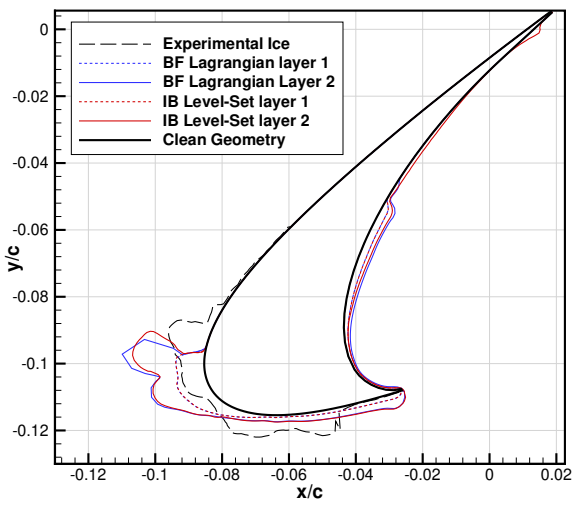
(a) Pressure Coefficients (Cp)

(b) Collection efficiency (β)

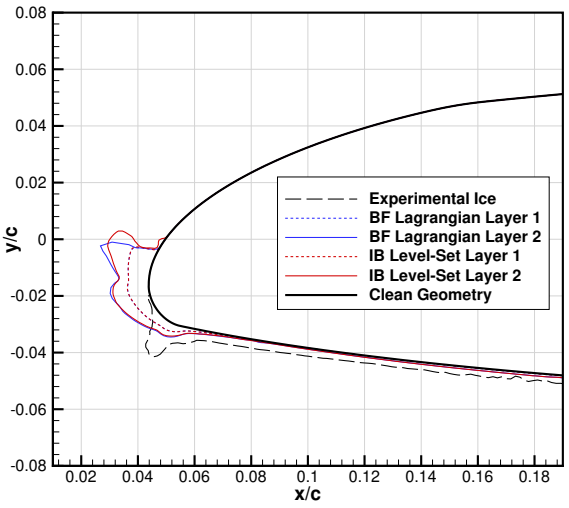
Figure 6.20 Wall data on the 1^{st} ice layer (2^{nd} step) for the McDonnell-Douglas multi-element airfoil (MDA LB606b)

Although a better ice shape prediction can be achieved using a RANS solver and a droplet size distribution, it was shown in [19] that a fair estimation of the ice shape can be achieved using a Euler flow solver and a single size of droplets for this specific test case. The ice accretion results obtained with IGLOO2D for both methods are shown for the flap, slat and main element in Figure 6.21. The predicted ice shapes are not so far from the experiment for the flap and slat but are quite different from the expected solution for the main element. Perhaps a simulation involving more ice layers would improve the results. For instance, six steps are used in [19]. The use of a polydisperse droplet distribution more representative of the cloud composition would also help (which is not available in this version of the IBM code). In this paper we are concerned about reproducing the ice shapes from the BF method with the IBM and in this regard, the ice shapes (Figure 6.21) are in fact similar for both

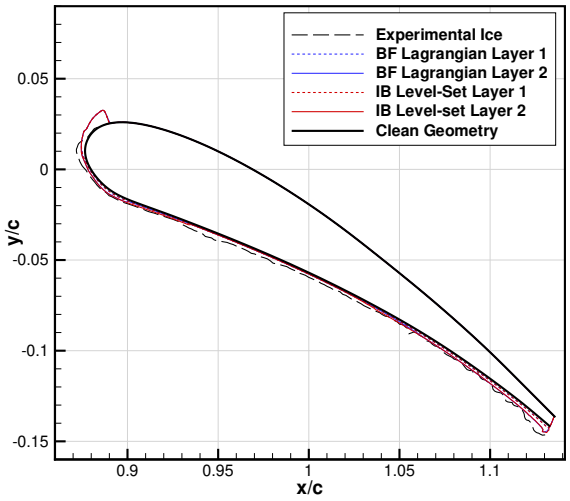
methods. The comparison still exhibits the usual discrepancy due to the accuracy of the IBM for the aerodynamics.



(a) Slat



(b) Main Element



(c) Flap

Figure 6.21 Two-step ice accretion on the McDonnell-Douglas multi-element airfoil (MDA LB606b)

6.5 Conclusion

This paper investigates the application of an Immersed Boundary Method (IBM) towards simulation of ice accretion within IGLOO2D. A penalization method is applied to the aerody-

namics and droplet trajectories. The surface data is extracted using a weighted least square approach in order to use the boundary layer and ice accretion modules. The geometry (the ice shape) is updated using either a Lagrangian or an Eulerian (level-set) approach. A contour extraction process is also described for 2D meshes made of triangles in order to retrieve the explicit definition of the ice-air interface.

Using a manufactured test case, the level-set is shown to automatically handle geometry folding during the ice shape update while the Lagrangian approach fails at providing a useable surface discretization unless a correction is added. For ice shape predictions, a Body-Fitted mesh is used for the clean geometry and only the ice shape is treated as an IB. Following this approach, rime and glaze ice cases from the Ice Prediction Workshop are performed using up to 10 ice layers. The IBM predicted an ice shape equivalent to the body-fitted approach on the rime ice case. For the glaze ice case, the predicted ice shape is close to the body-fitted solution but exhibits a larger difference where ice accretion is most dependent on the aerodynamics (e.g., near ice horns). The difference is mostly attributed to the accuracy of the IBM and not to the use of the level-set. Additional rime and glaze ice cases on a NACA0012 showed that the current approach (IBM + level-set) provides a fair estimation of the ice shape when compared to both the BF method and the experimental results, even on coarser meshes. Moreover, a 2-step ice shape prediction on the McDonnell-Douglas multi-element airfoil showed that with proper mesh refinement, the IBM combined with the level-set method can reproduce the BF solution on a more challenging configuration.

Although some improvements can be made in terms of efficiency and accuracy, this paper shows the potential of the proposed methodology for automatic multi-step ice shape predictions. Also, the extension the 3D ice accretion is, in theory, straightforward except for the contour extraction process which will require some adaptation to deal with a 2D surface mesh.

CHAPTER 7 GENERAL DISCUSSION

In the previous chapters, a methodology was described for the application of IBMs in an ice accretion suite (IGLOO2D). The three articles presented in this thesis illustrated the potential of the methodology for the prediction of ice shapes while entirely avoiding the regeneration of the volume mesh.

7.1 Summary and Benefits of the Methodology

The strategy is to represent the clean geometry with a BF mesh while only the ice shape remains immersed in the initial mesh. Thus the accuracy of the solution is maintained for the clean geometry (BF), including sharp features like sharp airfoil trailing edges. The mesh is initially refined in the zone where ice accretion is expected in order to maintain the wall cell size near the IB during the multi-step process.

An IBM is applied to the volume solvers (Euler and droplet equations) to enforce the boundary condition on the IB. A volume penalization method is selected for its flexibility and simplicity of implementation. This type of approach is independent of the spatial and temporal schemes. Thus, once the proper penalization terms are determined for the continuous form of the equations, the method can be easily applied to general meshes, two and three dimensions and various discretization methods (e.g., FVM, FDM, Finite Element Method (FEM), Spectral methods). This is useful as the methods developed in this thesis may be applied to other ice accretion suites in the future (e.g. 3D implementation).

Novel penalization methods are developed for the Euler and droplet equations with a special focus on the continuity of the solution in the vicinity of the Immersed Boundary (IB) while respecting the physics of the problem (e.g., conservation of entropy and total enthalpy). This simplifies the extraction of surface data as the continuity allows for an interpolation method with a stencil including both fluid and solid cells. This approach also provides relatively smooth surface data for use in the surface modules.

The volume penalization method enforces the boundary condition on the IB, only to 1^{st} order accuracy. This low order approach simplifies the implementation as no information is required at the discrete level. It also avoids the issues related to the identification of Image Points which are often required for 2^{nd} order boundary conditions. Nonetheless, the methodology is able to reproduce ice shapes obtained by BF simulations (with re-meshing) for rime ice cases. For glaze ice cases, which are more sensitive to the airflow solution, a good estimation of the

ice shape is still obtained, given a proper mesh refinement around high curvature features.

Despite the 1^{st} order imposition of boundary condition, 2^{nd} order accuracy can still be achieved for the droplet equations in the impingement zone, where no penalization terms are applied. This is a significant advantage as accurate results are obtained for the collection efficiency (impingement zone) while maintaining the simplicity of implementation from the 1^{st} order approach. A penalization method for the droplet equations was not available in the literature and the proposed approach is remarkably simple to implement as it requires only the addition of source terms with a special definition of the mask function to correctly impose the boundary conditions.

The level-set method is used to update the ice shape in the multi-step ice accretion process. Compared to the common node displacement approach (Lagrangian), the level-set method automatically handles the geometry overlaps that may be created in concave zones during the ice growth. It can also be naturally extended to three dimensions. Because it only provides an implicit definition of the interface, a contour extraction of the zero level-set is performed to retrieve the explicit definition (connected list of edges or nodes). The extracted contour is then re-meshed by fitting a B-Spline (using GMSH) in order to retrieve a smoother and more uniform surface mesh. The extracted contour actually provides a better discretization of the ice shape near high curvature regions (like ice horns) compared to the Lagrangian approach. The level-set method is able to reproduce the ice shape from a Lagrangian approach when using proper mesh refinement in the volume and proper numerical schemes (here 2^{nd} order accurate in space and time). The approach was shown to be flexible enough to handle multi-element airfoil configurations, extracting one surface mesh per body.

7.2 Limitations and Recommendations

As previously discussed, the proposed methodology (IBM + level-set) offers many benefits, including a good estimation of ice shapes for both rime and glaze ice cases. However, it still presents some limitations which are discussed in this section along with recommendations.

7.2.1 Accuracy

In this work, the implementation of the volume penalization method is limited to 1^{st} order accuracy which has some advantages as discussed previously. One of the drawbacks is that the aerodynamic solution near ice horns, ice accretion limits and sharp features like trailing edges is not well reproduced without additional mesh refinement. The sharp trailing edge issue is treated with a BF mesh in this thesis, but to make the method more general a

2^{nd} order implementation should be investigated.

Additionally, the current penalization method (airflow) is not accurate when the interface displacement is smaller than the cell size. This may occur when the number of steps is increased in the multi-step process and the ice shape becomes thinner. It suggests a minimum time step ($\Delta t_{ice} \geq \Delta x / V_{ice}$) in the multi-step process based on the characteristic mesh size (Δx) and icing velocity (V_{ice}). It means that the mesh should be refined as the number of ice layers is increased to obtain an accurate ice shape prediction. A 2^{nd} order implementation could also improve the behaviour of the penalization method for this situation, reducing the impact of cell size on the solution.

A second order accurate penalization requires information at the discrete level similar to the discrete forcing methods and thus the determination of interpolation points (e.g., Image Point) near concave regions or poorly refined zones will have to be treated with care. For instance, the method could fall back to 1^{st} order accuracy in these pathological zones to improve robustness. If a 2^{nd} order penalization method is to be implemented, discrete forcing methods should also be considered because applications are already documented in the literature for the Euler and droplet equations (e.g. face forcing, ghost-cell).

Instead of investing efforts in the implementation of a 2^{nd} order penalization method, an alternative approach would be to consider a volume fraction weighting of the boundary condition (see for instance Figure 2.4b in §2.2.3.1). It leads to a simpler implementation and avoids the definition of Image Points or Ghost-Cells, thus avoiding degenerate cases. However, further investigation is required to evaluate the improvements induced by such a method, if any.

7.2.2 Meshing

The idea of refining the mesh a priori allows to entirely avoid re-meshing during the multi-step simulations. However, the refinement is extended up to a specific distance from the BF wall regardless of the location of the IB and it leads to unnecessary cells in the solid far from the interface and to an over-refinement on the fluid side for the first few ice layers. Depending on the wall cell size and geometry, we quantified the increase in cell count to be in the range of 16% to 26% compared to a classical BF mesh without the refinement zone [151]. As the multi-step process may be performed over a large number of ice layers, it becomes important to reduce the additional CPU cost associated with the increased cell count. A simple approach, to reduce the number of unnecessary cells present in the solid is to deactivate them entirely. This could be easily done using a criterion based on the signed distance field. However, this does not reduce the number of cells present on the fluid side of

the refinement zone.

Moreover, the mesh size is constant in the refinement zone but the IBM requires different refinement level for optimal results (e.g., near high curvature , sharp features and ice accretion limits). This is currently treated by reducing the cell size for the entire refinement zone, which is not optimal.

Furthermore, one must know in advance where the ice accretion will occur for this method to work. The assumption made in this thesis is that a mesh refinement near the leading edge is sufficient for most 2D airfoil ice accretion simulations, but this might not always work for complex configurations. For instance, where should the refinement be applied for a full 3D aircraft configuration? The mesh could be refined everywhere along the wall up to a specified height, but this would significantly increase the number of cells. A more automatic approach should be used in the future for general applications.

One advantage of IBMs is that it is easier to perform mesh adaptation as the cells do not need to match the boundary. Thus, a mesh adaptation [72] (or automatic mesh refinement) could be performed in the vicinity of the immersed boundary (level $\phi = 0$) for each new ice layer. The process could also automatically refine the mesh based on curvature and ice thickness relative to the BF surface, solving most of the issues discussed in this section. This approach should be less expensive than a complete re-meshing of the volume mesh and should reduce the discretization error of the penalization method even if the 1^{st} order implementation is maintained. This would also reduce the number of cells in the refinement zone, reducing computation costs for the airflow and droplet solvers.

7.2.3 Robustness

The common strategy for multi-step ice accretion simulations is to use BF meshes combined with a Lagrangian node displacement method for the geometry update while the volume mesh is regenerated for each new ice layer. The robustness of this approach is affected by the Lagrangian geometry update which may present geometry overlaps, leading to a failure of the re-meshing or the failure of the volume solvers if the mesh was successfully generated but present invalid cells (e.g. negative volumes). Mesh re-generation may also fail for complex ice shapes, depending on the type of mesh.

The suggested methodology, combining IBMs and the level-set method, is robust as it is able to perform multi-step ice accretion simulations on rime ice, glaze ice and multi-element cases without failure. It also does not fail due to geometry overlaps, which is an improvement over the usual Lagrangian approach. As the calculations are performed on the initial mesh

(no re-meshing), the current methodology is thus more robust than a BF approach as the re-meshing may fail in some situations.

However, assuming a BF mesh can be successfully generated, the IBM methodology is still less robust than a BF approach as it is more sensitive to the mesh. For instance, the airflow solver may fail when solving the implicit system if the mesh is not sufficiently refined around sharp or high curvature features. It is up to the user to decide the required mesh refinement for a specific icing case. This can lead to a trial and error process as in a real case scenario the ice shape is not known a priori. With a BF approach, the re-meshing help in maintaining proper mesh refinement. Also, BF simulations do not usually fail due to poor mesh refinement, the mesh must actually present an issue (e.g., negative cell volumes) in order for the calculations to fail. In this regard, the BF approach is more robust than the IB method, again assuming a BF mesh can successfully be generated. The application of an automatic mesh adaptation procedure in the vicinity of the IB can again help in solving this issue by ensuring proper mesh refinement.

7.2.4 Explicit Definition of the Immersed Boundary

The contour extraction for each new ice layer adds complexity in the methodology and the implementation process. A simple approach was derived in 2D involving the detection of intersections between edges and the zero level-set. The approach also included node merging and node insertion to improve the regularity of the extracted contour. In 3D, the contour extraction process must be adapted to handle a 2D surface (probably made of triangles), where the merging and insertion will become more complex to implement. Although methods are available from the literature to perform 3D contour extraction, it is not trivial to obtain a robust method providing a good quality surface mesh. Instead of implementing an in-house tool, one possibility is to use software like Paraview [171] and GMSH [164] in order to perform the contour extraction and re-meshing tasks respectively. These steps can also be performed in 3D using Cassiopee [35], a tool developed at the ONERA.

An alternative approach, which should be investigated, is to entirely avoid the use of the explicit surface discretization. This could be done by solving the surface modules (ice accretion and boundary layer) on the volume mesh in a restricted band surrounding the interface (e.g., [172]). In this way, the surface mesh extraction (contour extraction) could be removed from the ice accretion process and only the implicit definition based on ϕ would be required. This method involves the modification of the surface module to account for an additional dimension (volume). In addition, it would be easier to apply this approach using PDEs in the surface modules, which is not currently the case, thus implying further modifications. For

instance, PDE-based models for the thermodynamics [173, 22] and boundary layer [174, 175] could be used. The accuracy and compatibility of such a method in combination with the current methodology (mixing immersed and body-fitted boundaries) is still to be investigated.

CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

This thesis investigates the application of Immersed Boundary Methods in ice accretion suites to determine if they are a viable replacement for classical body-fitted approach while helping in the automation of the multi-step ice accretion process.

The development platform (IGLOO2D) uses a Euler flow solver coupled with a boundary layer code, alleviating the mesh restrictions associated with RANS simulations. This strategy, however, shows its limitations on geometries exhibiting flow recirculation and flow separation because of the one-way coupling between the Euler and boundary layer codes. Because coarser meshes can be used, the strategy is to refine the initial mesh where ice accretion is expected and to avoid the re-meshing of the volume. With the proposed methodology, an IBM is applied to the volume solvers (airflow and droplets) while the surface solvers are left unmodified (ice accretion and boundary layer). A surface data extraction is also added to the ice accretion process to provide data to the surface solvers, data which is not directly available when using an IBM. To improve the robustness of the geometry evolution, the usual node displacement approach (Lagrangian) is replaced by a level-set method (Eulerian).

The first step towards the application of this methodology is to implement an IBM for the Euler flow solver. A volume penalization method was selected for its simplicity of implementation and independence of the discretization as it is based on the continuous form of the equations. Only one application was found for the Euler equations, using the Characteristic-Based Volume Penalization (CBVP) method. This approach from the literature was implemented and tested against a newly developed method, the CBVP-Hs method. This new approach is based on the CBVP method but with a new set of penalization terms designed to enforce the conservation of entropy and total enthalpy in the direction normal to the wall. The new penalization terms also provide a smooth solution in the vicinity of the IB to ease the interpolation process (surface data extraction). The two methods are compared on a circular cylinder, a NACA0012 and ice accreted GLC305 exhibiting an ice horn. The new method produces a lower error on entropy and total enthalpy while also providing more accurate Cp distributions than the CBVP method when using coarser meshes. The CBVP-Hs method also performs better for high curvature geometries such as ice horns by providing an attached flow solution which is more in line with the inviscid flow theory. The CBVP-Hs method is also shown to reproduce the BF solution in terms of wall pressure coefficient when using a proper mesh refinement.

As a second step, a volume penalization method is developed for the droplet impingement

solver. To the authors knowledge, this has never been done before. For this system of equations, a different set of wall boundary condition is imposed depending on the sign of the droplet velocities, determining an impingement zone and a shadow zone. The selection of the penalization terms for the droplet equations is not straightforward as they must prevent droplet re-injection from a solid wall to the fluid zone. A combined mask function is suggested to automatically detect the switch between the two types of boundary conditions and penalizing the equation only in the solid shadow zone. The physical equations are solved as usual in the impingement zone (both fluid and solid) allowing a smooth solution across the IB and improving the quality of the surface data extraction. The proposed penalization method is shown to provide smooth and accurate collection efficiency against the BF approach on a cylinder, a NACA0012 and an ice accretion GLC305 (ice horn). The method is also shown to be 2^{nd} order accurate in the impingement zone as no penalization is applied there.

As a third step, the previously developed penalization methods are combined with the level-set approach in the ice accretion software. A brief comparison of the Eulerian and Lagrangian approaches for the geometry evolution is made on a manufactured case. Contrary to the Lagrangian approach, the level-set method is shown to automatically handle the geometry intersections occurring in the concave zones, thus improving the robustness of the multi-step process. Then, multi-step ice accretion simulations are performed for airfoils and a multi-element configuration. The IBM combined with the level-set approach is compared to the BF method combined with the Lagrangian node displacement method. For rime ice cases, equivalent ice shapes are predicted using the IBM and BF methods. For glaze ice simulations, the IBM provides an ice shape close to the BF solution but with a larger difference where the aerodynamic solution has the largest influence (e.g., high curvature features like ice horn). The IBM combined with the level-set method is shown to be flexible enough to perform multi-step ice accretion simulations on a 3-element airfoil (high lift system) and still provide a good estimation of the ice shape compared to the BF approach. In general, the penalized Euler equations may require a finer mesh around ice horns, trailing edges and ice accretion limits to match the results from a BF approach. However, a fair prediction of the ice shape can still be obtained on coarser meshes.

An IBM was successfully applied to IGLOO2D to perform automatic multi-step ice accretion predictions. By using the IBM, the volume mesh update is totally avoided and only an initial mesh is required with a refined zone where ice accretion is expected. The accuracy is maintained for the clean part of the geometry by using a BF mesh. The IBM is used only for the ice shape and provides solutions in good agreement with the body-fitted results. The use of the level-set method for the geometry update automatically handles the pathological cases and therefore increases the robustness of the multi-step process. Although the proposed

methodology may require finer meshes for robustness and accuracy, it is still able to predict ice shapes in good agreement with the BF methodology and experimental results.

In the future, some improvements may be investigated such as the use of adaptive mesh refinement to improve the accuracy of the penalization method while also reducing the mesh size. Volume fraction weighting of the penalization terms and the implementation of a 2^{nd} order accurate IBM (penalization or not) are also considered in order to improve the representation of sharp features and also allow the use of coarser meshes near the IB. The proposed methodology shows great potential for 3D applications where it would exhibit greater benefits in the automation of the multi-step ice accretion.

REFERENCES

[1] R. Kind, M. Potapczuk, A. Feo, C. Golia, and A. Shah, "Experimental and computational simulation of in-flight icing phenomena," *Progress in Aerospace Sciences*, vol. 34, no. 5, pp. 257 – 345, 1998.

[2] R. W. Gent, N. P. Dart, and J. T. Cansdale, "Aircraft icing," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1776, pp. 2873–2911, 2000.

[3] FAA, *Electronic Code of Federal Regulations, Title 14*. U.S. Government Publishing Office, 2021, ch. Part 25 – Airworthiness Standards: Transport Category Airplanes, §25.1419 Ice Protection. [Online]. Available: https://www.ecfr.gov/cgi-bin/ECFR

[4] W. B. Wright, "User manual for the NASA Glenn Ice accretion code LEWICE version 2.2.2," NASA, CR 2002-211793, 2002. [Online]. Available: https://ntrs.nasa.gov/citations/20020080990

[5] F. Saeed, S. Gouttebroze, and I. Paraschivoiu, "Modified Canice for improved prediction of airfoil ice accretion," in *8th Aerodynamic Symposium, 48th CASI Conference*, 2001.

[6] H. Beaugendre, "A PDE-based 3D approach to in-flight ice accretion," Ph.D. dissertation, McGill University, June 2003. [Online]. Available: http://digitool.library.mcgill.ca/webclient/StreamGate?folder_id=0&dvs=1486315257461~532

[7] G. Gori, M. Zocca, M. Garabelli, A. Guardone, and G. Quaranta, "PoliMIce: A simulation framework for three-dimensional ice accretion," *Applied Mathematics and Computation*, vol. 267, pp. 96 – 107, 2015, the Fourth European Seminar on Computing (ESCO 2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300315007055

[8] D. Pena, Y. Hoarau, and E. Laurendeau, "A single step ice accretion model using level-set method," *Journal of Fluids and Structures*, vol. 65, pp. 278–294, 2016.

[9] G. A. Ruff and B. M. Berkowitz, "Users manual for the NASA Lewis ice accretion prediction code (LEWICE)," NASA, CR 185129, 1990. [Online]. Available: https://ntrs.nasa.gov/citations/19900011627

[10] K. Hasanzadeh, E. Laurendeau, and I. Paraschivoiu, "Quasi-steady convergence of multistep Navier–Stokes icing simulations," *Journal of Aircraft*, vol. 50, no. 4, pp. 1261–1274, 2013.

[11] R. Mittal and G. Iaccarino, "Immersed boundary methods," *Annual Review of Fluid Mechanics*, vol. 37, pp. 239–261, 2005.

[12] P. Trontin, G. Blanchard, A. Kontogiannis, and P. Villedieu, "Description and assessment of the new onera 2D icing suite IGLOO2D," in *9th AIAA Atmospheric and Space Environments Conference*, 2017.

[13] F. Capizzano, "A compressible flow simulation system based on cartesian grids with anisotropic refinements," in *Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics, Jan. 2007, pp. –.

[14] ——, "Turbulent wall model for immersed boundary methods," *AIAA Journal*, vol. 49, no. 11, pp. 2367–2381, 2011.

[15] ——, "Coupling a wall diffusion model with an immersed boundary technique," *AIAA Journal*, vol. 54, no. 2, pp. 728–734, Nov. 2015.

[16] F. Capizzano and E. Iuliano, "A Eulerian method for water droplet impingement by means of an immersed boundary technique," *Journal of Fluids Engineering*, vol. 136, no. 4, pp. 040 906–040 906–8, Feb. 2014.

[17] A. Al-kebsi, R. Mose, and Y. Hoarau, "Multi-Step Ice Accretion Simulation Using the Level-Set Method," in *SAE International Conference on Icing of Aircraft, Engines, and Structure*, Minneapolis, United States, Jun. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02562783

[18] B. L. Messinger, "Equilibrium temperature of an unheated icing surface as a function of air speed," *Journal of the Aeronautical Sciences*, vol. 20, no. 1, pp. 29–42, 1953.

[19] F. Petrosino, G. Mingione, A. Carozza, T. Gilardoni, and G. D'Agostini, "Ice accretion model on multi-element airfoil," *Journal of Aircraft*, vol. 48, no. 6, pp. 1913–1920, Nov. 2011. [Online]. Available: https://doi.org/10.2514/1.C031346

[20] H. Beaugendre, F. Morency, and W. Habashi, "Development of a second generation in-flight icing simulation code," *Journal of Fluids Engineering*, vol. 128, 03 2006.

[21] Y. Bourgault, W. G. Habashi, J. Dompierre, and G. S. Baruzzi, "A finite element method study of Eulerian droplets impingement models," *International Journal for Numerical Methods in Fluids*, vol. 29, no. 4, pp. 429–449, 1999.

[22] Y. Bourgault, H. Beaugendre, and W. G. Habashi, "Development of a shallow-water icing model in fensap-ice," *Journal of Aircraft*, vol. 37, no. 4, pp. 640–646, 2000.

[23] S. Bourgault-Côté, K. Hasanzadeh, P. Lavoie, and E. Laurendeau, "Multi-layer icing methodologies for conservative ice growth," in *7th European Conference for Aeronautics And Aerospace Sciences (EUCASS)*, 2017.

[24] G. Gori, G. Parma, M. Zocca, and A. Guardone, "Local solution to the unsteady stefan problem for in-flight ice accretion modeling," *Journal of Aircraft*, vol. 55, no. 1, pp. 251–262, 2018. [Online]. Available: https://doi.org/10.2514/1.C034412

[25] C. S. Bidwell and M. G. Potapczuk, "Users manual for the NASA lewis three-dimensional ice accretion code (LEWICE 3D)," NASA, Tech. Rep., 1994, nASA/TM–105974. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19940017117.pdf

[26] E. Radenac, "Validation of a 3D ice accretion tool on swept wings of the SUNSET2 program," in *AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, Jun. 2016, pp. –. [Online]. Available: https://doi.org/10.2514/6.2016-3735

[27] C. S. Bidwell, "Icing analysis of a swept NACA0012 wing using LEWICE3D version 3.48," in *6th AIAA Atmospheric and Space Environments Conference*, 2014.

[28] E. Radenac, H. Gaible, H. Bezard, and P. Reulet, "Igloo3d computations of the ice accretion on swept-wings of the sunset2 database," in *SAE Technical Paper*. SAE International, 06 2019. [Online]. Available: https://doi.org/10.4271/2019-01-1935

[29] T. Bellosta, G. Parma, and A. Guardone, "A robust 3d particle tracking solver for in-flight ice accretion using arbitrary precision arithmetic," in *COUPLED VIII: proceedings of the VIII International Conference on Computational Methods for Coupled Problems in Science and Engineering*, 2019. [Online]. Available: http://hdl.handle.net/2117/190171

[30] M. Morelli, T. Bellosta, and A. Guardone, "Lagrangian particle tracking in sliding mesh applicable for rotorcraft icing applications," in *45th European Rotorcraft Forum*

(ERF 2019), 2019. [Online]. Available: https://re.public.polimi.it/handle/11311/1111961?mode=full.753

[31] G. Mingione and V. Brandi, "Ice accretion prediction on multielement airfoils," *Journal of Aircraft*, vol. 35, no. 2, pp. 240–246, 1998.

[32] W. B. Wright, M. G. Potapczuk, and L. H. Levinson, "Comparison of LEWICE and GlennICE in the SLD regime," in *46th Aerospace Sciences Meeting and Exhibit*. AIAA Paper 2008-0439, 2008, NASA/TM—2008-21517.

[33] S. Péron, T. Renaud, I. Mary, C. Benoit, and M. Terracol, "An immersed boundary method for preliminary design aerodynamic studies of complex configurations," in *AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, Jun. 2017, pp. –. [Online]. Available: https://doi.org/10.2514/6.2017-3623

[34] S. Péron, C. Benoit, T. Renaud, and I. Mary, "An immersed boundary method on cartesian adaptive grids for the simulation of compressible flows around arbitrary geometries," *Engineering with Computers*, 2020.

[35] C. Benoit, S. Péron, and S. Landier, "Cassiopee: A cfd pre- and post-processing tool," *Aerospace Science and Technology*, vol. 45, pp. 272–283, 2015.

[36] R. T. Biedron, J.-R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, K. E. Jacobson, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, C. L. Rumsey, J. L. Thomas, K. B. Thompson, A. C. Walden, L. Wang, , and W. A. Wood, "FUN3D Manual: 13.7," NASA, TM 20205010139, 2020.

[37] C. Sarofeen, R. Noack, and R. Kreeger, "A non-cut cell immersed boundary method for use in icing simulations," in *Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics, Jan. 2012, pp. –, also NASA/TM–2013-217881. [Online]. Available: http://dx.doi.org/10.2514/6.2012-1204

[38] M. Aftosmis, M. Berger, and G. Adomavicius, "A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries," in *38th Aerospace Sciences Meeting and Exhibit*, 2000.

[39] H. Beaugendre, F. Morency, F. Gallizio, and S. Laurens, "Computation of ice shedding trajectories using cartesian grids, penalization, and level sets," *Model. Simul. Eng.*, vol. 2011, pp. 3:1–3:15, jan 2011.

[40] F. Sotiropoulos and X. Yang, "Immersed boundary methods for simulating fluid–structure interaction," *Progress in Aerospace Sciences*, vol. 65, pp. 1 – 21, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0376042113000870

[41] D. K. Clarke, H. A. Hassan, and M. D. Salas, "Euler calculations for multielement airfoils using cartesian grids," *AIAA Journal*, vol. 24, no. 3, pp. 353–358, Mar. 1986. [Online]. Available: https://doi.org/10.2514/3.9273

[42] M. Berger and M. Aftosmis, "Progress towards a cartesian cut-cell method for viscous compressible flow," in *50th AIAA Aerospace Sciences Meeting*, 2012, aIAA 2012-1301.

[43] M. Harada, Y. Tamaki, Y. Takahashi, and T. Imamura, "Simple and robust cut-cell method for high-reynolds-number-flow simulation on cartesian grids," *AIAA Journal*, vol. 55, no. 8, pp. 2833–2841, 2017.

[44] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy, "An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries," *Journal of Computational Physics*, vol. 156, no. 2, pp. 209 – 240, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999199963568

[45] H. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna, "A sharp interface cartesian grid method for simulating flows with complex moving boundaries," *Journal of Computational Physics*, vol. 174, no. 1, pp. 345 – 380, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999101969165

[46] J. Glimm, X. Li, Y. Liu, Z. Xu, and N. Zhao, "Conservative front tracking with improved accuracy," *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1926–1947, 2003. [Online]. Available: https://doi.org/10.1137/S0036142901388627

[47] M. Kirkpatrick, S. Armfield, and J. Kent, "A representation of curved boundaries for the solution of the Navier-Stokes equations on a staggered three-dimensional cartesian grid," *Journal of Computational Physics*, vol. 184, no. 1, pp. 1 – 36, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199910200013X

[48] X. Hu, B. Khoo, N. Adams, and F. Huang, "A conservative interface method for compressible flows," *Journal of Computational Physics*, vol. 219, no. 2, pp. 553 – 578, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106001926

[49] M. Meyer, A. Devesa, S. Hickel, X. Hu, and N. Adams, "A conservative immersed interface method for large-eddy simulation of incompressible flows," *Journal of Computational Physics*, vol. 229, no. 18, pp. 6300 – 6317, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999110002275

[50] Y. Cheny and O. Botella, "The ls-stag method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties," *Journal of Computational Physics*, vol. 229, no. 4, pp. 1043 – 1076, 2010.

[51] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, no. 2, pp. 252–271, 1972.

[52] M.-C. Lai and C. S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity," *Journal of Computational Physics*, vol. 160, no. 2, pp. 705 – 719, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999100964830

[53] J. M. Stockie and B. T. R. Wetton, "Stability analysis for the immersed fiber problem," *SIAM Journal on Applied Mathematics*, vol. 55, no. 6, pp. 1577–1591, 1995. [Online]. Available: http://dx.doi.org/10.1137/S0036139994267018

[54] J. M. Stockie and B. R. Wetton, "Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes," *Journal of Computational Physics*, vol. 154, no. 1, pp. 41 – 64, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999199962976

[55] C. Tu and C. S. Peskin, "Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 6, pp. 1361–1376, 1992. [Online]. Available: http://dx.doi.org/10.1137/0913077

[56] Y. Mori and C. S. Peskin, "Implicit second-order immersed boundary methods with boundary mass," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 25–28, pp. 2049 – 2067, 2008, immersed Boundary Method and Its Extensions. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782507002952

[57] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of Computational Physics*, vol. 25, no. 3, pp. 220 – 252, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999177901000

[58] T. Y. Hou and Z. Shi, "Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations," *Journal of Computational Physics*, vol. 227, no. 21, pp. 9138 – 9169, 2008, special Issue Celebrating Tony Leonard's 70th Birthday. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199910800137X

[59] ——, "An efficient semi-implicit immersed boundary method for the Navier-Stokes equations," *Journal of Computational Physics*, vol. 227, no. 20, pp. 8968 – 8991, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108003732

[60] H. D. Ceniceros, J. E. Fisher, and A. M. Roma, "Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method," *Journal of Computational Physics*, vol. 228, no. 19, pp. 7137 – 7158, 2009.

[61] B. E. Griffith and C. S. Peskin, "On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems," *Journal of Computational Physics*, vol. 208, no. 1, pp. 75 – 105, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999105000835

[62] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, "An adaptive, formally second order accurate version of the immersed boundary method," *Journal of Computational Physics*, vol. 223, no. 1, pp. 10 – 49, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106004207

[63] L. Zhu and C. S. Peskin, "Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method," *Journal of Computational Physics*, vol. 179, no. 2, pp. 452 – 468, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199910297066X

[64] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, p. 479–517, 2002.

[65] E. Arquis and J. P. Caltagirone, "Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide - milieu poreux: application à la convection naturelle," in *C. R. Acad. Sci. Paris*, ser. série II, 1984, vol. 299, no. 1, pp. 1–4.

[66] J. P. Caltagirone and E. Arquis, "Recirculations en milieu poreux," in *C. R. Acad. Sci. Paris*, ser. série II, 1986, vol. 302, no. 14.

[67] P. Angot, C.-H. Bruneau, and P. Fabrie, "A penalization method to take into account obstacles in incompressible viscous flows," *Numerische Mathematik*, vol. 81, no. 4, pp. 497–520, 1999.

[68] L. Nouveau, "Adaptive residual based schemes for solving the penalized Navier Stokes equations with moving bodies: application to ice shedding trajectories," Theses, Université de Bordeaux, Dec. 2016. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01500093

[69] A. Piquet, O. Roussel, and A. Hadjadj, "A comparative study of Brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows," *Computers & Fluids*, vol. 136, pp. 272 – 284, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793016301888

[70] A. Sarthou, S. Vincent, and J. Caltagirone, "A second-order curvilinear to cartesian transformation of immersed interfaces and boundaries. application to fictitious domains and multiphase flows," *Computers & Fluids*, vol. 46, no. 1, pp. 422 – 428, 2011, 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics (ICFD 2010). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793010003130

[71] A. Etcheverlepo, "Développement de méthodes de domaines fictifs au second ordre," Ph.D. dissertation, Université Bordeaux I, 2013. [Online]. Available: http://ori-oai.u-bordeaux1.fr/pdf/2013/ETCHEVERLEPO_ADRIEN_2013.pdf

[72] R. Abgrall, H. Beaugendre, and C. Dobrzynski, "An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques," *Journal of Computational Physics*, vol. 257, pp. 83 – 101, 2014.

[73] Q. Liu and O. V. Vasilyev, "A Brinkman penalization method for compressible flows in complex geometries," *Journal of Computational Physics*, vol. 227, no. 2, pp. 946 – 966, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999107003403

[74] E. Feireisl, J. Neustupa, and J. Stebel, "Convergence of a Brinkman-type penalization for compressible fluid flows," *Journal of Differential Equations*, vol. 250, no. 1, pp. 596 – 606, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022039610003682

[75] E. Brown-Dymkoski, N. Kasimov, and O. V. Vasilyev, "A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows," *Journal of Computational Physics*, vol. 262, pp. 344 – 357, 2014.

[76] K. Khadra, P. Angot, S. Parneix, and J.-P. Caltagirone, "Fictitious domain approach for numerical modelling of Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 34, no. 8, pp. 651–684, 2000. [Online]. Available: http://dx.doi.org/10.1002/1097-0363(20001230)34:8<651::AID-FLD61>3.0.CO;2-D

[77] I. Ramière, P. Angot, and M. Belliard, "A fictitious domain approach with spread interface for elliptic problems with general boundary conditions," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 4, pp. 766 – 781, 2007.

[78] H. Beaugendre and F. Morency, "Innovative model for flow governed solid motion based on penalization and aerodynamic forces and moments," INRIA Bordeaux, équipe CARDAMOM ; IMB ; INRIA, Research Report RR-8718, Apr 2015. [Online]. Available: https://hal.inria.fr/hal-01144855

[79] E. Brown-Dymkoski, N. Kasimov, and O. V. Vasilyev, "A characteristic-based volume penalization method for arbitrary mach flows around solid obstacles," in *Direct and Large-Eddy Simulation IX*, J. Fröhlich, H. Kuerten, B. J. Geurts, and V. Armenio, Eds. Cham: Springer International Publishing, 2015, pp. 109–115.

[80] J. Mohd-Yusof, "Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries," *Annual Research Briefs, Center for Turbulence Research*, pp. 317–328, 1997. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19990063249.pdf

[81] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof, "Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations," *Journal of Computational Physics*, vol. 161, no. 1, pp. 35 – 60, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999100964842

[82] J. Ravoux, A. Nadim, and H. Haj-Hariri, "An embedding method for bluff body flows: Interactions of two side-by-side cylinder wakes," *Theoretical and Computational Fluid Dynamics*, vol. 16, no. 6, pp. 433–466, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00162-003-0090-4

[83] Y. Cho, J. Chopra, and P. Morris, "Immersed boundary method for compressible high-reynolds number viscous flow around moving bodies," in *Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics, Jan. 2007, pp. –.

[84] Y.-H. Tseng and J. H. Ferziger, "A ghost-cell immersed boundary method for flow in complex geometry," *Journal of Computational Physics*, vol. 192, no. 2, pp. 593 – 623, 2003.

[85] L. Ge and F. Sotiropoulos, "A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries," *Journal of Computational Physics*, vol. 225, no. 2, pp. 1782 – 1809, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999107000873

[86] I. Borazjani, L. Ge, T. Le, and F. Sotiropoulos, "A parallel overset-curvilinear-immersed boundary framework for simulating complex 3D incompressible flows," *Computers & Fluids*, vol. 77, pp. 76 – 96, 2013.

[87] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," *Journal of Computational Physics*, vol. 227, no. 10, pp. 4825 – 4852, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108000235

[88] A. Dadone and B. Grossman, "Ghost-cell method for inviscid two-dimensional flows on cartesian grids," *AIAA Journal*, vol. 42, no. 12, pp. 2499–2507, 2004.

[89] S. Péron, C. Benoit, V. Gleize, I. Mary, and M. Terracol, "A mixed overset grid/immersed boundary approach for CFD simulations of complex geometries," in *54th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum*, Jan. 2016, aIAA 2016-2055.

[90] J. M. Hyman, "Numerical methods for tracking interfaces," *Physica D: Nonlinear Phenomena*, vol. 12, no. 1, pp. 396 – 407, 1984. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727898490544X

[91] R. Scardovelli and S. Zaleski, "Direct numerical simulation of free-surface and interfacial flow," *Annual Review of Fluid Mechanics*, vol. 31, no. 1, pp. 567–603, 1999. [Online]. Available: https://doi.org/10.1146/annurev.fluid.31.1.567

[92] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *The Physics of Fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[93] J. Glimm, J. Grove, B. Lindquist, O. A. McBryan, and G. Tryggvason, "The bifurcation of tracked scalar waves," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 1, pp. 61–79, 1988.

[94] S. O. Unverdi and G. Tryggvason, "A front-tracking method for viscous, incompressible, multi-fluid flows," *Journal of Computational Physics*, vol. 100, no. 1, pp. 25 – 37, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/002199919290307K

[95] J. Glimm, J. W. Grove, X. L. Li, K.-m. Shyue, Y. Zeng, and Q. Zhang, "Three-dimensional front tracking," *SIAM Journal on Scientific Computing*, vol. 19, no. 3, pp. 703–727, 1998.

[96] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, "A front-tracking method for the computations of multiphase flow," *Journal of Computational Physics*, vol. 169, no. 2, pp. 708 – 759, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999101967269

[97] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999188900022

[98] J. A. Sethian and P. Smereka, "Level set methods for fluid interfaces," *Annual Review of Fluid Mechanics*, vol. 35, no. 1, pp. 341–372, 2003. [Online]. Available: https://doi.org/10.1146/annurev.fluid.35.101101.161105

[99] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, 1st ed., ser. Applied Mathematical Sciences. Springer-Verlag New York, 2003, vol. 153.

[100] G.-H. Cottet and E. Maitre, "A level set method for fluid-structure interactions with immersed surfaces," *Mathematical Models and Methods in Applied Sciences*, vol. 16, no. 03, pp. 415–438, 2006. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218202506001212

[101] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, "An adaptive level set approach for incompressible two-phase flows," *Journal of Computational Physics*, vol. 148, no. 1, pp. 81 – 124, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199919896106X

[102] A. Criscione, D. Kintea, Ž. Tuković, S. Jakirlić, I. V. Roisman, and C. Tropea, "On computational investigation of the supercooled Stefan problem," in *ICLASS 2012, 12th Triennial International Conference on Liquid Atomization and Spray Systems*, 2012. [Online]. Available: http://ilasseurope.org/ICLASS/iclass2012_Heidelberg/Contributions/Paper-pdfs/Contribution1366_b.pdf

[103] S. Bourgault-Côté, J. Docampo-Sánchez, and E. Laurendeau, "Multilayer airfoil ice accretion simulations using a level-set method with B-Spline representation," *AIAA Journal*, vol. 57, no. 8, pp. 3299–3308, 2019.

[104] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, p. 163–169, aug 1987.

[105] G. Treece, R. Prager, and A. Gee, "Regularised marching tetrahedra: improved iso-surface extraction," *Computers & Graphics*, vol. 23, no. 4, pp. 583–598, 1999.

[106] J. C. Anderson, J. C. Bennett, and K. I. Joy, "Marching diamonds for unstructured meshes," in *VIS 05. IEEE Visualization, 2005.*, 2005, pp. 423–429.

[107] P. Trontin, "Développement d'une approche de type LES pour la simulation d'écoulements diphasiques avec interface : application à l'atomisation primaire," Ph.D. dissertation, 2009, thèse de doctorat dirigée par Caltagirone, Jean-Paul et Estivalèzes, Jean-Luc Dynamique des fluides Toulouse, ISAE 2009. [Online]. Available: http://www.theses.fr/2009ESAE0021

[108] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing," *Journal of Computational Physics*, vol. 183, no. 1, pp. 83–116, 2002.

[109] V. Mihalef, D. Metaxas, and M. Sussman, "Textured liquids based on the marker level set," *Computer Graphics Forum*, vol. 26, no. 3, pp. 457–466, 2007.

[110] R. Nourgaliev, S. Kadioglu, and V. Mousseau, "Marker redistancing/level set method for high-fidelity implicit interface tracking," *SIAM Journal on Scientific Computing*, vol. 32, no. 1, pp. 320–348, 2010.

[111] P. Rauschenberger, A. Criscione, K. Eisenschmidt, D. Kintea, S. Jakirlić, Ž. Tuković, I. Roisman, B. Weigand, and C. Tropea, "Comparative assessment of volume-of-fluid and level-set methods by relevance to dendritic ice growth in

supercooled water," *Computers & Fluids*, vol. 79, pp. 44 – 52, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793013000996

[112] W. F. Noh and P. Woodward, "SLIC (Simple Line Interface Calculation)," in *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede*, A. I. van de Vooren and P. J. Zandbergen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 330–340.

[113] W. J. Rider and D. B. Kothe, "Reconstructing volume tracking," *Journal of Computational Physics*, vol. 141, no. 2, pp. 112 – 152, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199919895906X

[114] J. López and J. Hernández, "Analytical and geometrical tools for 3D volume of fluid methods in general grids," *Journal of Computational Physics*, vol. 227, no. 12, pp. 5939 – 5948, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108001629

[115] O. Ubbink and R. Issa, "A method for capturing sharp fluid interfaces on arbitrary meshes," *Journal of Computational Physics*, vol. 153, no. 1, pp. 26–50, 1999.

[116] F. Xiao, Y. Honma, and T. Kono, "A simple algebraic interface capturing scheme using hyperbolic tangent function," *International Journal for Numerical Methods in Fluids*, vol. 48, no. 9, pp. 1023–1040, 2005. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.975

[117] F. Xiao, S. Ii, and C. Chen, "Revisit to the thinc scheme: A simple algebraic vof algorithm," *Journal of Computational Physics*, vol. 230, no. 19, pp. 7086–7092, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999111003615

[118] L. Qian, Y. Wei, and F. Xiao, "Coupled thinc and level set method: A conservative interface capturing scheme with high-order surface representations," *Journal of Computational Physics*, vol. 373, pp. 284–303, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118304534

[119] I. Marter, "Méthode d'interface immergée pour la simulation directe de l'atomisation primaire," Theses, Institut supérieur de l'Aéronautique et de l'Espace (ISAE) ; Université de toulouse, Dec. 2017. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01767721

[120] Z. Wang, J. Yang, B. Koo, and F. Stern, "A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves," *International Journal of Multiphase Flow*, vol. 35, no. 3, pp. 227 – 246, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301932208001833

[121] G. Son and N. Hur, "A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 42, no. 6, pp. 523–542, 2002. [Online]. Available: http://dx.doi.org/10.1080/10407790260444804

[122] G. Son, "Efficient implementation of a coupled level-set and volume-of-fluid method for three-dimensional incompressible two-phase flows," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 43, no. 6, pp. 549–565, 2003. [Online]. Available: http://dx.doi.org/10.1080/713836317

[123] M. Sussman and E. G. Puckett, "A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows," *Journal of Computational Physics*, vol. 162, no. 2, pp. 301 – 337, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999100965379

[124] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," *Journal of Computational Physics*, vol. 187, no. 1, pp. 110 – 136, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999103000871

[125] M. Sussman, K. Smith, M. Hussaini, M. Ohta, and R. Zhi-Wei, "A sharp interface method for incompressible two-phase flows," *Journal of Computational Physics*, vol. 221, no. 2, pp. 469 – 505, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106002981

[126] T. Ménard, S. Tanguy, and A. Berlemont, "Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet," *International Journal of Multiphase Flow*, vol. 33, no. 5, pp. 510 – 524, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301932206001832

[127] X. Yang, A. J. James, J. Lowengrub, X. Zheng, and V. Cristini, "An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids," *Journal of Computational Physics*, vol. 217, no. 2, pp. 364 – 394, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106000143

[128] M. W. Jones, J. A. Baerentzen, and M. Sramek, "3D distance fields: a survey of techniques and applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 581–599, July 2006.

[129] P. J. Schneider and D. H. Eberly, "Chapter 13 - computational geometry topics," in *Geometric Tools for Computer Graphics*, ser. The Morgan Kaufmann Series in Computer Graphics, D. H. SCHNEIDER, PHILIP J.and EBERLY, Ed. San Francisco: Morgan Kaufmann, 2003, pp. 673 – 825. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781558605947500163

[130] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996. [Online]. Available: http://www.pnas.org/content/93/4/1591.abstract

[131] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proceedings of the National Academy of Sciences*, vol. 95, no. 15, pp. 8431–8435, 1998. [Online]. Available: https://www.pnas.org/content/95/15/8431

[132] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.

[133] H. Zhao, "A fast sweeping method for Eikonal equations," *Mathematics of Computation*, vol. 74, no. 250, pp. 603–627, 2005.

[134] J. Qian, Y. Zhang, and H. Zhao, "Fast sweeping methods for Eikonal equations on triangular meshes," *SIAM Journal on Numerical Analysis*, vol. 45, no. 1, pp. 83–107, 2007.

[135] P. Tucker, "Differential equation-based wall distance computation for DES and RANS," *Journal of Computational Physics*, vol. 190, no. 1, pp. 229 – 248, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999103002729

[136] H. Xia and P. G. Tucker, "Finite volume distance field and its application to medial axis transforms," *International Journal for Numerical Methods in Engineering*, vol. 82, no. 1, pp. 114–134, 2010. [Online]. Available: http://dx.doi.org/10.1002/nme.2762

[137] M. Sussman, P. Smereka, and S. Osher, "A level set approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, no. 1, pp. 146 – 159, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999184711557

[138] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, "A PDE-based fast local level set method," *Journal of Computational Physics*, vol. 155, no. 2, pp. 410 – 438, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999199963453

[139] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM Journal on Scientific Computing*, vol. 20, no. 4, pp. 1165–1191, 1999.

[140] R. Meakin, *Object X-rays for cutting holes in composite overset structured grids*, 2001. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2001-2537

[141] T. D. Aslam, "A partial differential equation approach to multidimensional extrapolation," *Journal of Computational Physics*, vol. 193, no. 1, pp. 349 – 355, 2004.

[142] Z. Li and K. Ito, *The Immersed Interface Method*. Society for Industrial and Applied Mathematics, 2006. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9780898717464

[143] A. Dadone and B. Grossman, "Surface boundary conditions for the numerical solution of the Euler equations," *AIAA Journal*, vol. 32, no. 2, pp. 285–293, feb 1994.

[144] O. Boiron, G. Chiavassa, and R. Donat, "A High-Resolution Penalization Method for large Mach number Flows in the presence of Obstacles," *Computers and Fluids*, vol. 38, no. 3, pp. 703–714, 2009.

[145] Y. Bae and Y. J. Moon, "On the use of Brinkman penalization method for computation of acoustic scattering from complex boundaries," *Computers & Fluids*, vol. 55, pp. 48 – 56, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045793011003215

[146] A. Dadone and B. Grossman, "Ghost-cell method for analysis of inviscid three-dimensional flows on cartesian-grids," *Computers & Fluids*, vol. 36, no. 10, pp. 1513 – 1528, 2007, special Issue Dedicated to Professor Michele Napolitano on the Occasion of his 60th Birthday.

[147] Z. J. Wang and Y. Sun, "Curvature-based wall boundary condition for the Euler equations on unstructured grids," *AIAA Journal*, vol. 41, no. 1, pp. 27–33, 2003.

[148] A. Sarthou, S. Vincent, J. P. Caltagirone, and P. Angot, "Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with

complex objects," *International Journal for Numerical Methods in Fluids*, vol. 56, no. 8, pp. 1093–1099, 2008.

[149] P. Roe, "Approximate Riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, no. 2, pp. 357 – 372, 1981.

[150] T. Barth and D. Jespersen, "The design and application of upwind schemes on unstructured meshes," in *27th Aerospace Sciences Meeting*, 1989.

[151] P. Lavoie, G. Blanchard, E. Radenac, E. Laurendeau, and P. Villedieu, "A Penalization Method for 2D Ice Accretion Simulations," in *SAE Technical Paper*, no. 2019-01-1939. SAE International, jun 2019.

[152] C. L. Ladson, C. W. B. Jr., A. S. Hill, and D. W. Sproles, "Computer program to obtain ordinates for NACA airfoils," NASA, Tech. Rep., 1996, nASA–TM–4741. [Online]. Available: https://ntrs.nasa.gov/citations/19970008124

[153] J. C. Vassberg and A. Jameson, "In pursuit of grid convergence for two-dimensional Euler solutions," *Journal of Aircraft*, vol. 47, no. 4, pp. 1152–1166, Jul. 2010.

[154] W. B. Wright, R. W. Gent, and D. Guffond, "DRA/NASA/ONERA collaboration on icing research: Part II–Prediction of airfoil ice accretion," NASA, CR 202349, 1997. [Online]. Available: https://ntrs.nasa.gov/citations/19970023937

[155] W. B. Wright, "User's manual for LEWICE version 3.2," NASA, CR 2008-214255, 2008. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080048307.pdf

[156] P. Lavoie, E. Radenac, G. Blanchard, Éric Laurendeau, and P. Villedieu, "An improved characteristic based volume penalization method for the euler equations towards icing applications," *Computers & Fluids*, p. 104917, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793021000839

[157] M. BERGER and R. LEVEQUE, "An adaptive cartesian mesh algorithm for the Euler equations in arbitrary geometries," in *Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics, Jun. 1989, pp. –. [Online]. Available: https://doi.org/10.2514/6.1989-1930

[158] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *Journal of*

Computational Physics, vol. 152, no. 2, pp. 457 – 492, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999199962368

[159] L. Schiller and Z. Naumann, "A drag coefficient correlation," *Z. Ver. Deutsch. Ing.*, vol. 77, pp. 318–323, 1935.

[160] F. Durst, D. Miloievic, and B. Schönung, "Eulerian and Lagrangian predictions of particulate two-phase flows: a numerical study," *Applied Mathematical Modelling*, vol. 8, no. 2, pp. 101 – 115, 1984.

[161] A. Harten, P. D. Lax, and B. v. Leer, "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws," *SIAM Review*, vol. 25, no. 1, pp. 35–61, 1983.

[162] E. Norde, "Eulerian method for ice crystal icing in turbofan engines," Ph.D. dissertation, University of Twente, May 2017.

[163] A. Broeren, "1st AIAA ice prediction workshop," Website, Jan. 2021. [Online]. Available: http://folk.ntnu.no/richahan/IPW/

[164] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579

[165] P. Lavoie, E. Radenac, G. Blanchard, E. Laurendeau, and P. Villedieu, "A penalization method for Eulerian droplet impingement simulations towards icing applications," in *AIAA SciTech 2021 Forum*. American Institute of Aeronautics and Astronautics, 2021, AIAA-2021-1442.

[166] S. Lee, A. P. Broeren, R. E. Kreeger, M. G. Potapczuk, and L. Utt, "Implementation and validation of 3-D ice accretion measurement methodology," in *6th AIAA Atmospheric and Space Environments Conference*, 2014.

[167] P. Lavoie, P. Dorian, H. Yannick, and E. Laurendeau, "Comparison of thermodynamic models for ice accretion on airfoils," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 28, no. 5, pp. 1004–1030, 2018.

[168] A. P. Broeren, M. G. Potapczuk, S. Lee, A. M. Malone, B. P. Paul, and B. Woodard, "Ice-accretion test results for three large-scale swept-wing models in the NASA icing research tunnel," in *8th AIAA Atmospheric and Space Environments Conference*, 2016.

[169] G. Ruff and D. Anderson, "Quantification of ice accretions for icing scaling evaluations," in *Aerospace Sciences Meetings*. American Institute of Aeronautics and Astronautics, Jan. 1998, pp. –, nASA TM–2003-212308. [Online]. Available: https://doi.org/10.2514/6.1998-195

[170] W. B. Wright, "Validation methods and results for a two-dimensional ice accretion code," *Journal of Aircraft*, vol. 36, no. 5, pp. 827–835, sep 1999.

[171] J. Ahrens, B. Geveci, and C. Law, *ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook*. Elsevier, 2005.

[172] S. J. Ruuth and B. Merriman, "A simple embedding method for solving partial differential equations on surfaces," *Journal of Computational Physics*, vol. 227, no. 3, pp. 1943–1961, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002199910700441X

[173] R. Chauvin, P. Villedieu, P. Trontin, and L. Bennani, "A robust coupling algorithm applied to thermal ice protection system unsteady modeling," in *AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics, Jun. 2014, pp. –.

[174] C. Bayeux, E. Radenac, and P. Villedieu, "Theory and validation of a 2d finite-volume integral boundary-layer method for icing applications," *AIAA Journal*, vol. 57, no. 3, pp. 1092–1112, 2019.

[175] E. Radenac, C. Bayeux, and P. Villedieu, "Use of a two-dimensional finite volume integral boundary-layer method for ice-accretion calculations," *AIAA Journal*, vol. 58, no. 4, pp. 1592–1606, 2020.

APPENDIX A WEIGHTED LEAST SQUARE INTERPOLATION

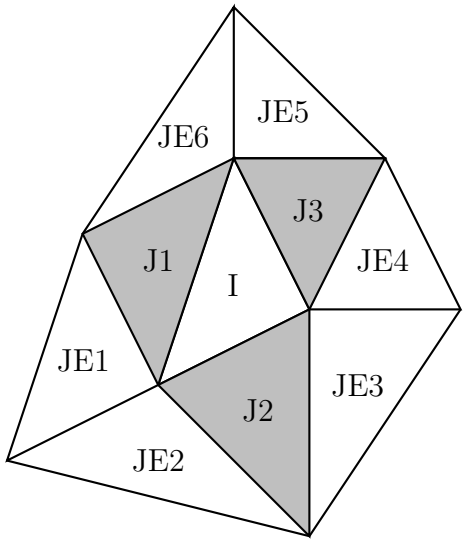


Figure A.1 Cell I, its restricted neighborhood J and extended neighborhood JE

Least Square Interpolation

A Least Square (LSQ) approach can be used for data interpolation at an arbitrary point in a grid. Assuming the value ϕ_P is sought at a point P located within cell I, the data point can be interpolated by constructing a linear approximation between each neighbor cell J to the cell I:

$$\phi_J = \phi_P + \nabla\phi_P \cdot \vec{r}_{PJ} \tag{A.1}$$

where $\vec{r}_{PJ}$ is the distance vector from point P pointing towards J. There are three unknowns: ϕ_P, $\frac{\partial\phi}{\partial x}_P$ and $\frac{\partial\phi}{\partial y}_P$. Considering the restricted neighborhood of Figure A.1, the system to solve becomes (in 2D):

$$\begin{bmatrix} (x_I - x_P) & (y_I - y_P) & 1 \\ (x_{J1} - x_P) & (y_{J1} - y_P) & 1 \\ (x_{J2} - x_P) & (y_{J2} - y_P) & 1 \\ (x_{J3} - x_P) & (y_{J3} - y_P) & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial\phi}{\partial x}_P \\ \frac{\partial\phi}{\partial y}_P \\ \phi_P \end{bmatrix} = \begin{bmatrix} \phi_I \\ \phi_{J1} \\ \phi_{J2} \\ \phi_{J3} \end{bmatrix} \tag{A.2}$$

This approach can be extended to an arbitrary number of neighbors and can be solved using for instance a QR factorization (typically using the modified Gram-Smith orthogonal decomposition).

The selected neighborhood will affect the interpolation quality. For instance, if the interpolation point is close to a cell edge, the stencil is slightly shifted. An alternative approach is to build an interpolation support based on distance and symmetry criteria. In this way the interpolation point is always centered with respect to the neighborhood. The identification of the cells forming the interpolation support is however more costly than simply reusing the already known cell neighbors.

Weighted Least Square Interpolation

A WLSQ method can be used to improve the accuracy of the interpolation on irregular meshes. The contribution of each neighbouring cells is assigned a weight (w), leading to the following system (in 2D):

$$\begin{bmatrix} w_I(x_I - x_P) & w_I(y_I - y_P) & w_I \\ w_{J1}(x_{J1} - x_P) & w_{J1}(y_{J1} - y_P) & w_{J1} \\ w_{J2}(x_{J2} - x_P) & w_{J2}(y_{J2} - y_P) & w_{J2} \\ w_{J3}(x_{J3} - x_P) & w_{J3}(y_{J3} - y_P) & w_{J3} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi}{\partial x}_P \\ \frac{\partial \phi}{\partial y}_P \\ \phi_P \end{bmatrix} = \begin{bmatrix} w_I \phi_I \\ w_{J1} \phi_{J1} \\ w_{J2} \phi_{J2} \\ w_{J3} \phi_{J3} \end{bmatrix} \tag{A.3}$$

A typical choice for the weight w_J is:

$$w_J = \frac{1}{||\vec{r}_{PJ}||^t} \tag{A.4}$$

with $t = 0, 1$ or 2. The exponent t is typically set to unity.

When the interpolation point P is too close to one of the cell-centers (J) forming the stencil, the distance ($||\vec{r}_{PJ}||$) tends towards zero. In theory, this leads to a very large weight and thus no interpolation is performed. The unknown at point P is simply the value at cell-center J: $\phi_P = \phi_J$. In practice, it leads to a division by very small number. The weight is thus modified as:

$$w_J = \frac{1}{(||\vec{r}_{PJ}||^2 + \epsilon^2)^{t/2}} \tag{A.5}$$

where ϵ is a small number.

When extracting the surface data at the IB, smoothing is introduced by setting $\epsilon = 0.5\Delta x_J$. The characteristic cell size is $\Delta x_J = \sqrt{4A_J/3}$ for a triangular cell and $\Delta x_J = \sqrt{A_J}$ for a quadrilateral cell. Here, A_J represent the area of the cell J.