Titre: Title:	Immersed Boundary Methods for the Three-Dimensional Modeling of In-Flight Ice Accretion
Auteur: Author:	Pablo Elices Paz
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Elices Paz, P. (2025). Immersed Boundary Methods for the Three-Dimensional Modeling of In-Flight Ice Accretion [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/68458/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/68458/
Directeurs de recherche: Advisors:	Éric Laurendeau, & Emmanuel Radenac
Programme: Program:	Génie mécanique

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 Three-Dimensional Modeling of In-Flight Ice Accretion

PABLO ELICES PAZ
Département de génie mécanique

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

Septembre 2025

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 Three-Dimensional Modeling of In-Flight Ice Accretion

présentée par **Pablo ELICES PAZ**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Roberto PAOLI, président
Éric LAURENDEAU, membre et directeur de recherche
Emmanuel RADENAC, membre et codirecteur de recherche
Héloïse BEAUGENDRE, membre externe
Yannick HOARAU, membre externe

ACKNOWLEDGEMENTS

This PhD has been an extraordinary journey, made possible by the many people who have supported me along the way, across two countries, two institutions, and more than a few time zones.

I am deeply grateful to my supervisors: Emmanuel Radenac, Éric Laurendeau, Stéphanie Péron, Ghislain Blanchard, and Philippe Villedieu. A special thanks goes to my thesis directors, Emmanuel and Éric. Emmanuel, thank you for your unwavering involvement in this project and for guiding me in what it truly means to do research. Your insight and mentorship have shaped not just this thesis, but the way I approach scientific work. Éric, thank you for your technical expertise, and also for helping me understand the broader context of this research. I've learned a great deal about aviation through our discussions, and it has been both inspiring and eye-opening.

To all the researchers, engineers, and fellow students at ONERA and Polytechnique Montréal, thank you for the many scientific exchanges, your generosity with your time and ideas, and for making the workdays more enjoyable.

Pursuing this PhD far from home, between Toulouse and Montreal, would not have been possible without the constant support of my friends. To those back in Spain, who always managed to make me feel close despite the distance, and to the new friends I've met in France and Canada, thank you for your presence and encouragement throughout these years.

And last but never least, I want to thank my family. To my mother and my sister, thank you for getting me here, for your love and strength, and for the many hours of phone calls that kept me grounded and motivated. And to Mario, thank you for your constant support and encouragement.

None of this would have been possible without you.

RÉSUMÉ

Le givrage en vol est un danger bien connu pour la sécurité et les performances des aéronefs. Il se produit généralement dans des conditions météorologiques froides, lorsqu'un avion rencontre un nuage surfondu. L'un des principaux effets de ce phénomène est la dégradation des performances aérodynamiques. Malgré le développement de systèmes de protection contre le givrage, les avions commerciaux doivent pouvoir démontrer leur capacité à voler dans de telles conditions. Par conséquent, l'intérêt pour le développement de simulations de givrage en vol s'est accru ces dernières années, avec un focus particulier sur les simulations 3D.

Les simulations numériques du givrage reposent généralement sur l'hypothèse que le temps d'exposition de la surface aux conditions de givrage est significativement plus long que le temps caractéristique de l'écoulement aérodynamique. Cette différence d'échelle temporelle permet une stratégie de modélisation modulaire, dans laquelle le processus global est décomposé en une succession d'appels de modules. Tout d'abord, un maillage volumique est généré à partir de la surface considérée afin de résoudre l'écoulement aérodynamique stationnaire et les trajectoires des gouttes surfondues. Ensuite, un bilan thermodynamique est fait à la surface afin de déterminer l'épaisseur locale de glace. Enfin, la géométrie est mise à jour pour intégrer la nouvelle couche de glace formée. Dans une simulation *multi-step*, le temps d'exposition est subdivisé en plusieurs incréments. Cette approche répète de manière itérative la succession des modules, permettant ainsi à la simulation de prendre en compte l'évolution de la géométrie et son influence sur l'écoulement et l'impact des gouttes, améliorant ainsi la précision de la solution. Cependant, des défis importants apparaissent lors de l'automatisation de ce processus, principalement en raison de la complexité associée à la génération de maillages de haute qualité à chaque itération.

L'objectif principal de ce travail est d'intégrer les méthodes de frontières immergées (*Immersed Boundary Method*, IBM) dans le cadre de simulation du givrage afin d'éliminer la nécessité de générer un maillage volumique à chaque étape, tout en conservant le niveau de précision obtenu avec une approche conforme (*Body-Fitted*, BF). Ceci permet d'utiliser un seul maillage volumique tout au long des étapes d'une simulation *multi-steps*, facilitant ainsi l'automatisation du processus.

Tout d'abord, une méthode IBM est développée pour les simulations 3D d'écoulement d'air, où l'écoulement non-visqueux est modélisé en résolvant les équations d'Euler. Cependant, comme le bilan thermodynamique à la surface dépend des échanges diffusifs, la solution d'Euler doit être couplée à un solveur de couche limite. Une méthode IBM basée sur un

forçage discret à partir d'un maillage conforme à la géométrie initiale lisse a été développée afin de prendre en compte la modification de la géométrie à chaque pas d'accrétion.

Ensuite, une IBM pour les équations eulériennes 3D des gouttes est nécessaire. En raison de la nature des équations et des conditions aux limites, une méthode de pénalisation a été choisie. Cette méthode a été validée dans des simulations 3D, démontrant la préservation de la précision de la solution.

Enfin, les IBMs développées sont intégrées dans IGLOO3D, la suite de simulation 3D de givrage de l'ONERA, pour réaliser des simulations *multi-steps*. Cela nécessite des modifications mineures dans le reste de la chaine de simulation. Différents cas de givrage sont étudiés, incluant des profils 2D et des ailes 3D en flèche, ainsi que des conditions de givre blanc (*rime*) et givre transparent (*glaze*). Les résultats montrent une bonne concordance avec l'approche BF. Comparée aux données expérimentales, la méthodologie donne globalement de bonnes performances. Cependant, des défis subsistent dans les cas où un décollement massif de l'écoulement est attendu, ouvrant la voie à de futures recherches.

ABSTRACT

In-flight ice accretion is a well-known hazard to flight safety and performance. It typically occurs under cold weather conditions, when an aircraft encounters a supercooled cloud. One of the main results of the presence of this phenomenon is the degradation of the aerodynamic performance, increasing the drag and reducing the lift of the aircraft, but it also impairs the controllability of the aircraft as well as the data readout among others. Despite the development of aircraft ice protection systems, commercial aircraft must be able to prove that they are capable of flying under icing conditions. As a result, there has been an increased interest in developing simulations of in-flight ice accretion. In recent years, work has focused on the development of three-dimensional simulations.

Numerical simulations of ice accretion commonly rely on the assumption that the surface exposure time to icing conditions is significantly longer than the characteristic time of the aerodynamic flow. This time scale difference enables a modular modeling strategy, in which the overall process is decomposed into a sequential call of modules. First, a volume mesh is generated from the surface under consideration to solve the steady-state aerodynamic flow and the trajectories of the supercooled water droplets. Next, the thermodynamic exchanges at the surface are evaluated to determine the local ice thickness. Finally, the geometry is updated to include the newly formed ice layer. In a multi-step simulation, the exposure time is subdivided into multiple steps. This approach iteratively repeats the sequential call of modules, enabling the simulation to capture the evolving geometry and its impact on the flow and droplet impingement, thereby enhancing solution accuracy. However, significant challenges arise in automating this process, primarily due to the complexity involved in generating high-quality meshes at each iteration.

The main objective of this work is to integrate Immersed Boundary Methods (IBMs) into the ice accretion framework to eliminate the need for volume mesh generation at each step, while preserving the level of accuracy achieved with a Body-Fitted (BF) approach. This allows a single volume mesh to be used throughout all steps of a multi-step simulation, thereby enhancing the automation of the process.

First, an IBM is developed for three-dimensional airflow simulations, where the inviscid flow is modeled by solving the Euler equations. However, since the thermodynamic exchanges at the surface depend on the diffusive transfer quantities, the Euler solution must be coupled with a boundary layer solver. An IBM method based on a discrete forcing starting from a body-conformal mesh around the smooth initial geometry has been developed to account for the modification of the geometry at each step of the accretion process.

Second, an IBM for the three-dimensional Eulerian droplet equations is needed. Due to the nature of the governing equations and boundary conditions, a penalization method was chosen. This method has been validated in three-dimensional simulations, demonstrating preservation of solution accuracy.

Finally, the previously developed IBMs are integrated into IGLOO3D, the three-dimensional ice accretion suite of ONERA, to perform multi-step simulations. This requires minor modifications to the rest of the simulation workflow. Different ice accretion cases are considered, including two-dimensional airfoils and three-dimensional swept wings, as well as rime and glaze icing conditions. The results show good agreement with the BF approach. When compared to experimental data, the methodology performs well overall; however, challenges arise in cases where significant flow separation is expected, paving the way for future research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

AGCS	Adaptive Ghost-Cell-Surrounding
AMR	Adaptive Mesh Refinement
BF	Body-Fitted
CCST	Curvature-Corrected Symmetry Technique
CFD	Computational Fluid Dynamics
FAST	Flexible Aerodynamic Solver Technology
FVM	Finite Volume Method
GCIBM	Ghost-Cell Immersed Boundary Method
IB	Immersed Boundary
IBM	Immersed Boundary Method
IPS	Ice Protection Systems
IPW	Ice Prediction Workshop
IRT	Icing Research Tunel
LWC	Liquid Water Content
MVD	Median Volume Diameter
RANS	Reynolds-Averaged Navier–Stokes
SLD	Supercooled Large Droplets
ST	Symmetry Technique
SWIM	Shallow Water Icing Model

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Context

In-flight ice accretion is a well-known hazard to flight safety and performance. It typically occurs under cold weather conditions, when an aircraft encounters supercooled clouds. A droplet is called supercooled when it is in liquid phase but its temperature is below the freezing point. This metastable state may persist from a few seconds to several days. However, minor exterior perturbations, such as the impact with an aircraft, can initiate the phase change, causing ice accretion. One of the main consequences of this phenomenon is the degradation of the aerodynamic performance, increasing the drag and reducing the lift, but it also impairs the controllability of the aircraft as well as the data readout among others.

While icing was not initially considered a major issue in the early days of aviation, its importance gradually increased over time. Starting in the 1940s, research on icing became more relevant, particularly from theoretical and experimental perspectives. Icing wind tunnels became necessary, not only to assess the degradation of aerodynamic performance, but also to evaluate the effectiveness of de-icing and anti-icing systems. The National Aeronautics and Space Administration (NASA) was the first institution to develop an icing wind tunnel, the Icing Research Tunel (IRT) [93]. Shortly thereafter, the National Research Council (NRC) of Canada built the Low-Speed Icing Tunnel to support similar experimental studies [123]. Figure 1.1 shows an example of an icing experiment performed in the IRT. Since then, several other research centers have developed their own icing wind tunnels, such as those of Rail Tec Arsenal (RTA) in Austria [66] or ONERA (Office National d'Études et de Recherches Aérospatiales) in France [21].

The development of computational capabilities from the 1980s onward enabled the creation of the first ice accretion codes. Due to the inherent difficulty in accurately predicting ice shapes, collaboration between research centers has existed from the very beginning. In the 1990s, NASA, DRA (Defence Research Agency), and ONERA led the first comparison studies of simulation results [189], with the aim of improving and validating aircraft icing simulation codes. In recent years, this spirit of collaboration has evolved into the organization of ice prediction workshops. To date, the first two editions of these workshops have been held [84, 86]. The objective remains the same: to improve and validate icing codes. To that end, several test cases are defined, experiments are conducted in icing wind tunnels, and the numerical results from different participants are compared and analyzed. These workshops help identify the main challenges in ice accretion simulations, as well as define guidelines for future work aimed at improving the codes.

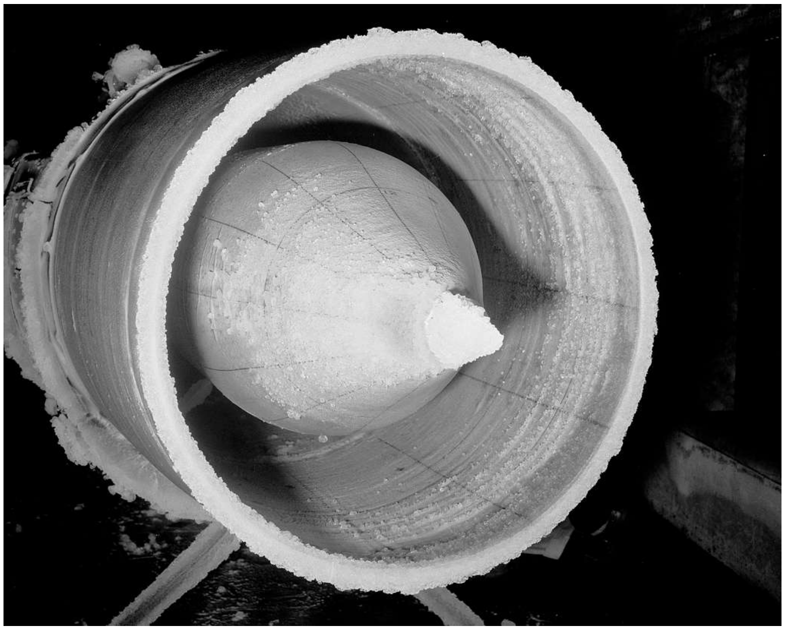


Figure 1.1 Icing of inlet of B-58 airplane in the IRT. Image from [119].

1.1.1 Ice Morphologies

Depending on the icing conditions, different types of ice can form. At very low temperatures, supercooled droplets freeze almost immediately upon impact, trapping air and forming a porous, opaque accretion known as rime ice. As the temperature increases, the solidification process is delayed, allowing for the development of a liquid water film on the surface. In regions with enhanced convective heat transfer, this film may flow downstream, a phenomenon known as runback, before eventually freezing. This process leads to the formation of a denser, more transparent accretion referred to as glaze ice, often characterized by horn-like shapes. The term mixed ice is sometimes used to describe cases where the surface simultaneously exhibits features of both rime and glaze ice.

Additionally, for swept wings in glaze or mixed icing conditions, ice accretion can be divided into three subcategories known as complete scallops, incomplete scallops, and no scallops [53] (see Fig. 1.2). Scallops refer to spanwise-periodic patterns of ice ridges that form along the leading edge of the wing [29]. They often resemble overlapping lobster tails, with a series of rounded ridges separated by troughs. From a numerical perspective, these complex three-dimensional features are particularly challenging to capture accurately and typically require high-fidelity ice accretion models [126].

Figure 1.2 Scallop ice patterns observed on the CRM65 midspan model in the IRT for the 2^{nd} IPW. Image from [126].

1.1.2 Icing Certification

With the increasing number of flights, flying in icing conditions has become increasingly common since the 1920s, leading to the formulation of specific certification requirements to ensure safety. Historically, the foundational regulatory framework for icing certification has been defined in Appendix C, which characterizes typical in-flight icing conditions. Initially developed by the Federal Aviation Administration (FAA) in December 1964 and later adopted by the European Union Aviation Safety Agency (EASA) under CS-25 [63], this appendix defines the certification envelope for typical icing conditions (e.g., by restricting droplet diameters to values below 50 μm) and has served as the primary reference for Ice Protection Systems (IPS) design and validation for several decades.

However, a number of incidents during the 1990s and early 2000s revealed the limitations of Appendix C, particularly in its inability to account for more complex and hazardous icing environments, such as those involving droplet sizes beyond the Appendix C range, including Supercooled Large Droplets (SLD) and ice crystals. Notable examples include the crash of American Eagle Flight 4184 in 1994 [120], and the accident involving Air France Flight 447 in 2009 [32]. These events prompted significant research efforts and joint regulatory initiatives between the FAA and EASA, ultimately leading to the introduction of Appendix O and Appendix P, which extend the icing certification envelope to SLD and ice crystals conditions respectively.

Despite the efforts of certification authorities to ensure flight safety in icing conditions, accidents due to ice accretion continue to occur, as evidenced by the accident involving the domestic Brazilian passenger flight 2283 operated by Voepass in December 2024. The aircraft involved, an ATR 72-500, crashed after entering flat spin. The preliminary report of the accident [2], published by CENIPA, the Brazilian Aeronautical Accidents Investigation and Prevention Center, confirmed that pilots faced difficulties with icing buildup and de-icing attempts.

The development of IPSs has not eliminated the requirement for commercial aircraft to demonstrate their ability to operate under icing conditions. This process involves three different types of studies related to ice accretion: flight tests, wind tunnel experiments, and numerical simulations. Flight tests are the most costly option and are generally not conducted as an initial step due to the inherent risks involved. Moreover, since icing conditions are difficult to predict, they are typically carried out using artificial ice shapes, obtained either from wind tunnel experiments or numerical simulations. Wind tunnel experiments pose significant challenges when scaling the problem. Specifically, establishing a similarity analysis on icing-relevant dimensionless numbers is generally difficult or insufficient, typically leading a mitigation based on the modification of the wing shape [29]. Additionally, icing experiments are known to exhibit considerable variance, meaning that two identical tests conducted in the same icing wind tunnel can yield different ice shapes. Due to these challenges, there is a growing need to conduct in-flight ice accretion simulations.

1.1.3 Ice Accretion Simulations

Ice accretion is a highly multi-physics phenomenon. However, the process is typically addressed by dividing it into three main blocks. First, the aerodynamic flow around the wing is computed. Then, based on this flow field, the trajectories and impingement of the supercooled droplets are evaluated. Finally, a thermodynamic mass and energy balance is performed to determine the ice thickness on the exposed surfaces.

Given the large difference in time scales, with surface exposure to icing conditions typically lasting hundreds of seconds and the characteristic time of the airflow and droplets being around hundredths of a second, the phenomenon can be considered quasi-steady. Furthermore, due to the low water concentration, with typical volume fractions around $1 \cdot 10^{-6}$, two-way coupling between the airflow and the droplet trajectory solver is not required, i.e., the effect of droplets on the aerodynamic solution is neglected. As a result, each block can be solved sequentially, as illustrated in Fig. 1.3, where two additional modules are employed:

mesh generation and geometry update. The former generates the volume mesh used by both the airflow and droplet solvers, while the latter displaces the ice shape from the ice thickness.

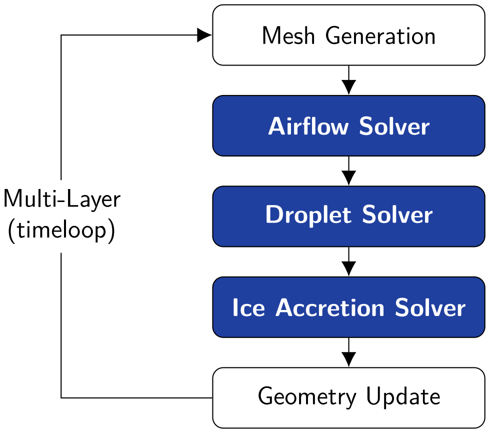
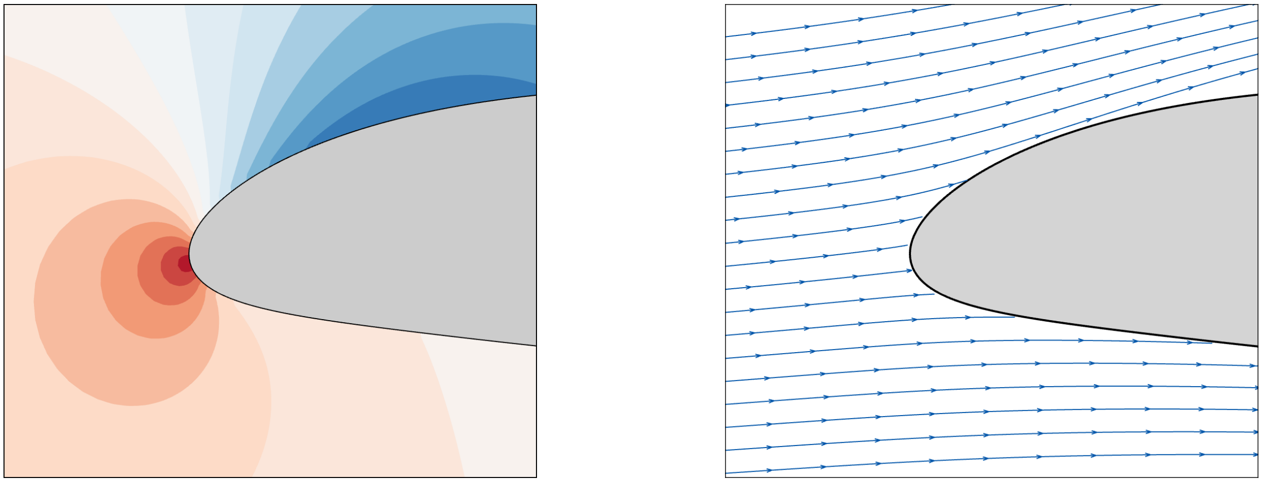


Figure 1.3 Sequential call to modules in ice accretion simulations.

In a one-shot simulation, also known as one-step or single-step simulation, the main loop is only performed once. While this is the most cost-effective method, it is also the least accurate for ice accretion simulations. One alternative, which enhances the accuracy of the simulation without a substantial increase in computational cost, is the use of a predictor-corrector approach [141]. A more commonly used alternative, known as the multi-step method, involves dividing the exposure time into N discrete steps. This allows for the consideration of variations in both the surrounding flow and the droplet impingement as the ice accretes [176]. As the number of steps increases, so does the computational cost of the simulation, and the feasibility of automation is hindered. To better illustrate the procedure, Fig. 1.4 shows the multi-step approach used for a two-dimensional ice accretion simulation.

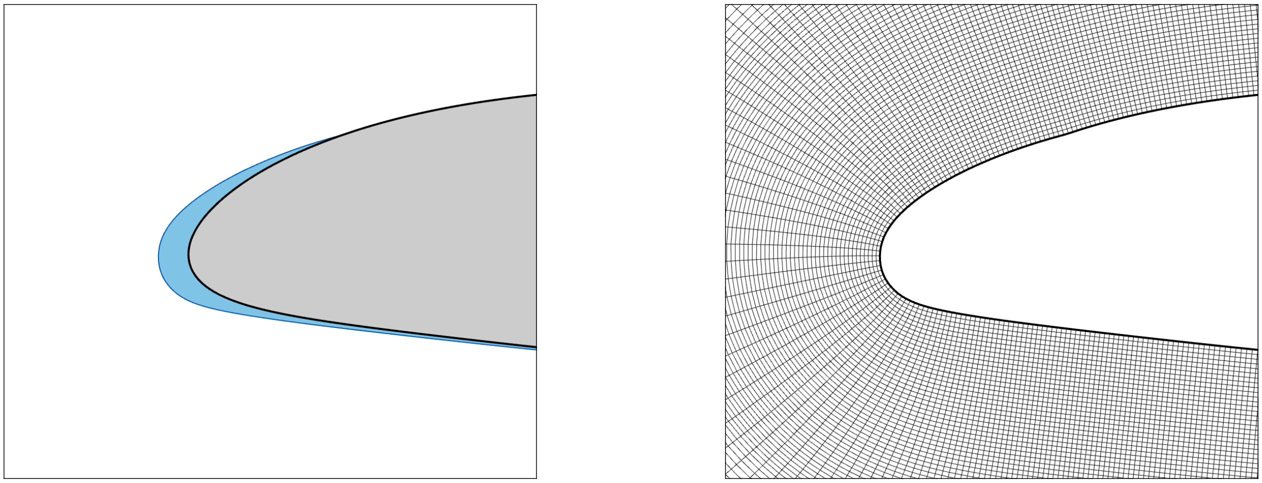
1.2 Problem Statement

In two-dimensional ice accretion simulations, the multi-step approach is widely adopted. A good practice when performing ice accretion simulations is to ensure convergence by progressively increasing the number of steps, thereby verifying that the solution remains unchanged when reducing the time step used to update the ice shape. Some studies even employ over hundred steps to achieve this [69]. However, the high number of steps gives rise to two main challenges: the computational cost and the level of automation of the entire simulation. The latter is primarily due to mesh generation, which typically requires manual intervention.



(a) Airflow solution

(b) Droplet solution



(c) Ice accretion and geometry update

(d) New mesh generation

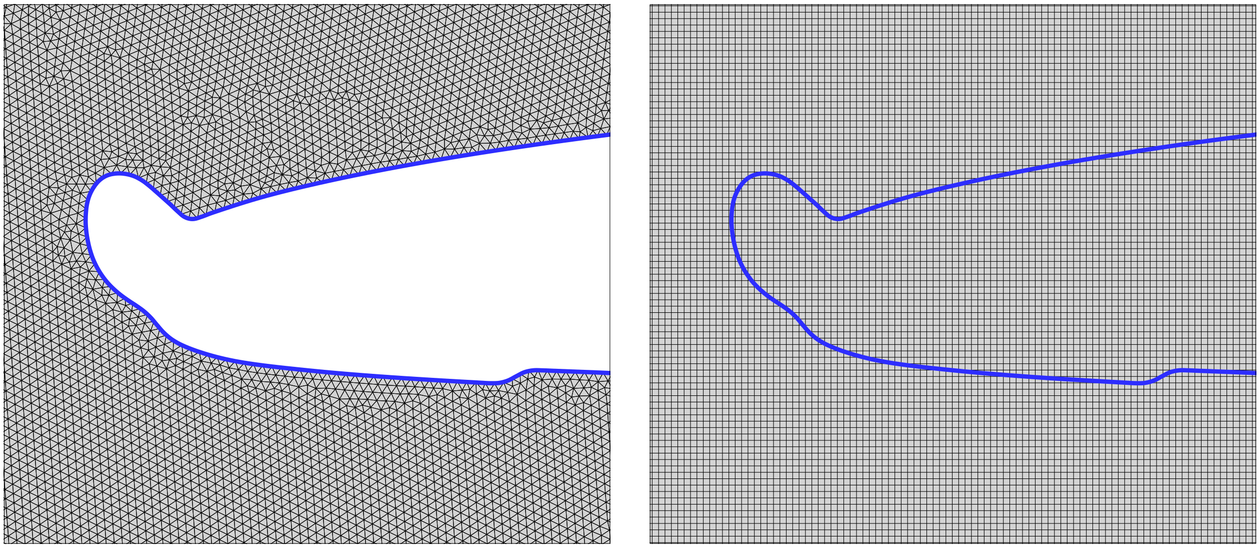
Figure 1.4 Illustration of the multi-step approach. At each iteration, a new ice layer is computed, and a volume mesh is generated for use in the next iteration.

In three-dimensional simulations, these difficulties increase significantly. New equations must be solved and the total number of cells grow exponentially, making the computational cost much higher compared to two-dimensional simulations. Furthermore, volume mesh generation around complex iced surfaces becomes a highly demanding task. The complexity of these simulations was evident during the 1^{st} Ice Prediction Workshop (IPW) [86], where fewer than 25 percent of participants chose to undertake multi-step three-dimensional simulations. However, multi-step simulations are essential for accurately predicting three-dimensional icing features such as scallops. These scallops create a masking effect on the water collection that cannot be properly resolved without accounting for the evolution of the ice shape.

1.3 Preceding Research

This thesis builds upon the doctoral work of Pierre Lavoie [87], carried out jointly by Polytechnique Montréal and ONERA, which focused on the use of Immersed Boundary Methods (IBMs) for two-dimensional ice accretion simulations.

Immersed Boundary Methods are numerical techniques typically used to simulate fluid flows around complex or moving geometries without the need to conform the computational mesh to the body's shape. By embedding the solid object within a fixed grid, IBMs simplify the meshing process. Figure 1.5 compares a classical Body-Fitted (BF) mesh against an Immersed Boundary (IB) mesh around a GLC305 iced airfoil. For multi-step ice accretions, the need of generating a new mesh at each iteration is then avoided, improving the automation of the ice accretion chain.


(a) Body-fitted mesh

(b) IB Cartesian mesh

Figure 1.5 Comparison between BF and IB meshes around a iced GLC305 airfoil.

Whether using conventional BF meshes or IBMs, the level of fidelity of the aerodynamic solution in the simulations significantly impacts their computational cost. In particular, the airflow simulation is often one of the most computationally demanding when high-fidelity methods are employed. For instance, Reynolds-Averaged Navier–Stokes (RANS) models, which account for the viscous boundary layer by refining the mesh around the wall, has become the standard practice in ice accretion simulations. However, this model significantly increases the computational cost of the simulation compared to inviscid approaches, such as the Euler equations or even panel methods. As an alternative, inviscid flow models weakly coupled with an integral boundary layer are still advantageous in ice accretion simulations. Notably, IGLOO2D [176], the two-dimensional ice accretion suite developed by ONERA, and LEWICE [152], NASA's ice accretion code, are both based on inviscid flow assumptions and are widely recognized for their effectiveness in this context.

It is in this context that Lavoie's thesis was carried out, focusing on the development of two-dimensional IBMs to solve the inviscid flow and compute droplet trajectories in IGLOO2D. Several conclusions can be drawn from his work:

- The use of IBM enabled fully automated multi-step ice accretion simulations. Particular care must be taken during the geometry update phase to prevent overlaps in the newly formed ice layer.
- The IBM implemented in the airflow solver was limited to first-order accuracy. As a result, the solver may struggle to capture small variations in the ice shape, which restricts the maximum number of accretion steps that can be employed.
- In contrast, the IBM developed for the droplet solver achieved second-order accuracy. Furthermore, its extension to three-dimensional simulations is expected to be relatively straightforward.

Another important prior contribution is the work conducted at ONERA on integral boundary layer solvers. In particular, the doctoral thesis of Charlotte Bayeux [13] aimed to develop a three-dimensional integral method that, when coupled with an inviscid flow simulation, could yield results comparable to those obtained with RANS models, while significantly improving the efficiency and reducing the computational cost of the aerodynamic simulation. While the model proved successful in two-dimensional cases, its extension to three-dimensions still presents some robustness challenges.

1.4 Objectives of the Present Study

The primary objective of this thesis by compendium is *to explore the application of Immersed Boundary Methods in 3D ice accretion simulations under Appendix C icing conditions, with a dual focus on substantially streamlining the automation process while maintaining the high accuracy standards achievable through Body-Fitted approaches.* This global objective is achieved by specific objectives in terms of solver development and validation, solution integration as well as applicability of the presented methodology.

Therefore, to accomplish the main objective, the following sub-objectives are defined:

Development of Numerical Methods and Validation:

O.1 Design a 3D Immersed Boundary Method for Aero-Icing Computations The first task of the present work is to develop an IBM to solve the 3D Euler equations. Firstly, second-order accuracy in benchmark test cases must be demonstrated. Secondly, flow over complex geometries should be analyzed, including two-dimensional and three-dimensional ice shapes. Additionally, the solution must be validated with results from BF simulations or solutions from other codes.

O.2 Verify and Validate the Proposed 3D Immersed Boundary Method for the Calculations of Droplet Trajectories The penalization method for the Eulerian droplet impingement equations, presented in IGLOO2D, must be validated in three-dimensional simulations. To this end, the method will be applied to a set of benchmark problems, including both two-dimensional and three-dimensional cases, with well-established reference solutions.

O.3 Propose a Simplified Integral Method for Modeling Boundary Layers on 3D Surfaces The simplified integral method, which is currently restricted to laminar flows, should be extended to turbulent flows. Knowledge of boundary layer theory, as well as numerical methods for its modeling, is required. The proposed code should support laminar-turbulent transition, and allow the simulation of three-dimensional swept wings. Furthermore, verification and validation cases should be presented, and a comparison of the solution against other codes should be analyzed.

Solver Integration:

O.4 Integrate the Immersed Boundary Methods into a 3D Computational Workflow The present work should be integrated into the three-dimensional ice accretion suite of ONERA, IGLOO3D. Firstly, connection between the different modules must be verified. Therefore, converter scripts need to be developed. Secondly, modifications to the existing codes may be required, as the boundary conditions of the volume mesh do not match the ice-fluid interface.

Applications:

O.5 Evaluate the Proposed Methodology in Ice Accretion Simulations Scenarios The present methodology should be applied to typical ice accretion simulations. Two- and three-dimensional representative icing cases should be chosen to assess the effectiveness of the IB multi-step simulations.

1.5 Scientific Dissemination

This thesis has resulted in various publications and presentations, summarized as follows. In particular, it includes two first-authored submitted publications in *Computers & Fluids* and *Journal of Aircraft*, fulfilling the requirements of both Polytechnique Montreal and ISAE.

1.5.1 Journal Publications

Submitted Manuscripts

- **Elices Paz, P.**, Radenac, E., Péron, S., Blanchard, G., Laurendeau, E., & Villedieu, P. (2025). *Second-Order Ghost-Cell Immersed Boundary Methodology for Inviscid 3D Flows in Complex Geometries.* Computers & Fluids. (Under review. Submitted on 14 February 2025)

- **Elices Paz, P.**, Radenac, E., Blanchard, G., Péron, S., Laurendeau, E., & Villedieu, P. (2025). *Immersed Boundary Methodology for 3D Multi-Step Ice Accretion Simulations.* Journal of Aircraft. (Under review. Submitted on 19 June 2025)

1.5.2 Conference Publications

- **Elices Paz, P.**, Radenac, E., Péron, S., Blanchard, G., Laurendeau, E., & Villedieu, P. (2023). *3D Immersed Boundary Methods for the Calculations of Droplet Trajectories towards Icing Application.* SAE Technical Paper 2023-01-1458, 2023, https://doi.org/10.4271/2023-01-1458
- **Elices Paz, P.**, Radenac, E., Blanchard, G., Péron, S., Laurendeau, E., & Villedieu, P. (2024, July 29). *Immersed Boundary Methodology for 3D Multi-Step Ice Accretion Simulations.* AIAA Aviation Forum and Ascend 2024, Las Vegas, Nevada. https://doi.org/10.2514/6.2024-4161

1.6 Thesis Outline

This manuscript is presented as an article-based thesis and is organized as follows.

Chapter 2 provides a comprehensive literature review, covering existing two- and three-dimensional ice accretion codes, as well as the various IBMs available in the literature, with particular attention to their application in ice accretion simulations.

Chapter 3 outlines the overall strategy for integrating the IBMs into the three-dimensional ice accretion framework and introduces the articles that form the core of this thesis.

Chapter 4 presents the first article, which focuses on the definition and implementation of a ghost-cell IBM for inviscid three-dimensional flows.

Chapter 5 provides a detailed presentation of the three-dimensional simplified integral boundary layer method.

Chapter 6 contains the second article, analyzing the results of three-dimensional multi-step ice accretion simulations using the presented IB ice accretion framework.

Chapter 7 analyzes the strengths and limitations of the proposed methodology.

Finally, Chapter 8 summarizes the main contributions of this work, and outlines potential directions for future research.

CHAPTER 2 LITERATURE REVIEW

Simulating in-flight ice accretion involves several multiphysics phenomena, such as fluid dynamics, multiphase flows, heat transfer and thermodynamic balance on the iced surface. As a result, a wide range of modeling strategies have been developed over the years, leading to significant variation across existing ice accretion codes, depending on the specific choices made by each developer. The first part of this chapter is therefore dedicated to examining the different models currently employed to perform ice accretion simulations under icing conditions. Particular focus is placed on the first two Ice Prediction Workshops (IPWs) [84, 86], which serve as key references for understanding the modeling approaches currently employed by the community.

The second part of the chapter shifts focus to Immersed Boundary Methods. We review the main types of IBMs, including cut-cell, continuous, and discrete approaches, outlining their principles and key differences.

Finally, the chapter examines the application of IBMs in ice accretion simulations. This includes a comprehensive review of existing codes that have integrated IBMs, emphasizing their advantages and limitations.

2.1 Ice Accretion Models

To assess the current state-of-the-art in ice accretion codes, the models employed by all participants of the 1^{st} IPW [86] held in 2022 are taken as a representative sample. The workshop submission included results from many research and industry institutions, as shown in Table 2.1.

ID	Participant	ID	Participant	ID	Participant
1	Boeing	8	CIRA	15	Polytechnique Montreal
2	-	9	Oxford	16	FAA
3	NASA	10	AIT	17	Honeywell
4	ONERA	11	NTNU	18	AeroTex GmbH
5	Georgia Tech/NASA	12	NRC	19	Siemens - Lockheed
6	Politechnico Di Milano	13	ANSYS-Bombardier	20	Universität Braunschweig
7	Textron	14	Embraer	21	ATS

Table 2.1 Participants of the 1^{st} IPW [86].

The first notable observation when analyzing the different ice accretion models is that they all adopted the previously introduced quasi-steady assumption, which allows the ice accretion process to be modeled through a sequential call of individual modules. Figure 2.1 illustrates the detailed workflow of the multi-step approach, which was consistently followed by all participants.

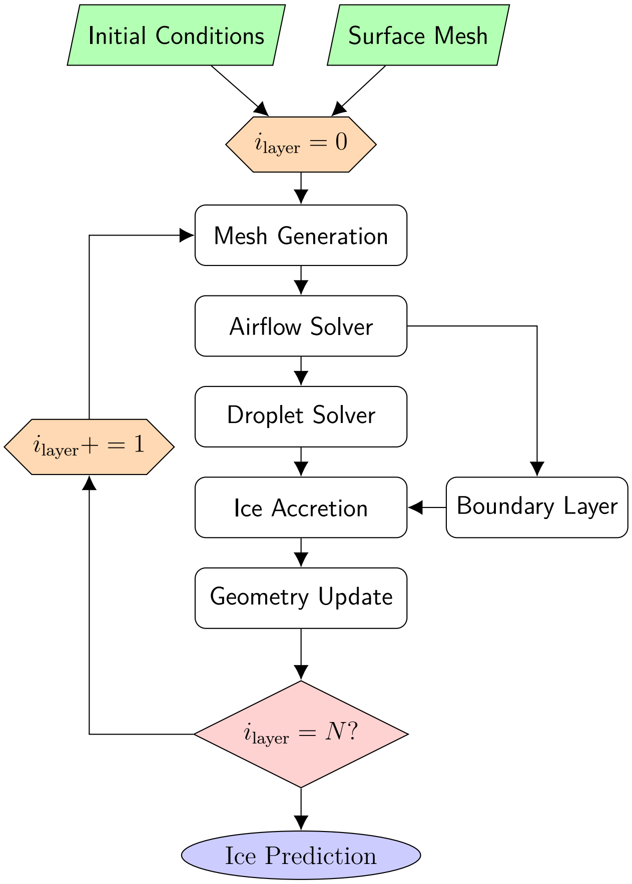


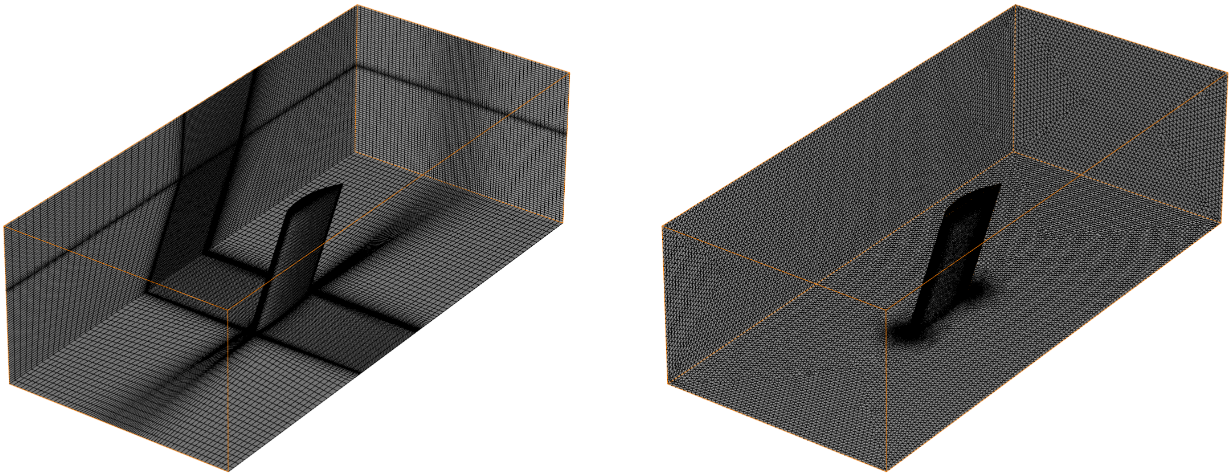
Figure 2.1 Typical multi-step ice accretion simulations workflow.

The number of layers employed by each participant varied significantly. In two-dimensional cases, 33% of the participants performed one-shot simulations, while in the three-dimensional simulations, this percentage increased to 75%. These values highlight the challenges associated with conducting multi-step simulations in three-dimensional cases. Moreover, one of the conclusions drawn from the workshop was the notable sensitivity of the predicted ice shapes to the number of layers, influencing, for instance, the total ice mass [126]. Furthermore, convergence studies of the predicted ice shape with increasing number of layers (see, e.g., [69, 176]) were notably absent, primarily due to the difficulty and computational cost associated with such studies.

An analysis of the different models used in each module of the workflow is presented below.

2.1.1 Mesh Generation

The simulation begins with the volume mesh generation, built from the geometry under study. Both structured and unstructured meshes can be used. To generate these meshes, a wide range of techniques has been developed over the years. Structured grids are often created by solving partial differential equations [167]. In contrast, unstructured meshes are typically built using different algorithms [104], such as the Delaunay triangulation [137] or the advancing-front method [132]. As an illustration, Fig. 2.2 shows both type of meshes provided by the 1^{st} IPW committee for Case 361, which involves a 30º swept NACA 0012 wing. Although structured solvers typically provide higher performance, the flexibility of unstructured meshes in representing complex geometries has made them the preferred choice for ice accretion simulations, particularly in three-dimensional cases, as evidenced in previous IPWs [84, 86].


(a) Structured mesh

(b) Unstructured mesh

Figure 2.2 Provided meshes by the 1^{st} IPW for Case 361.

Despite the efforts made over the past decades to automate this process, the generation of high-quality meshes from complex geometries remains far from fully automatic. Whether structured or unstructured grids are desired, the algorithms typically require several input parameters, often making human intervention necessary to verify the resulting mesh and, in most cases, to modify parameters or adjust the mesh to improve the result.

When performing multi-step simulations, this task must be repeated after the computation of each ice layer. As an alternative, mesh deformation techniques [172], commonly used in moving boundary problems, can be employed [55]. These methods are generally less computationally expensive than regenerating the volume mesh at each step. However, despite their development, they often fail to preserve mesh quality when applied to complex ice shapes. For the 1^{st} IPW, both mesh deformation [113] and mesh regeneration [141] have been employed.

2.1.2 Aerodynamic Flow Model

Solving the aerodynamic flow serves three main purposes in the context of ice accretion simulations: it governs the particle trajectories, provides the wall heat transfer field, and defines the direction of water runback on the surface.

The first generation of ice accretion codes relied on 2D inviscid solvers, typically panel methods, to compute the pressure distribution around the airfoil. Most of these codes were developed before the 2000s. Examples include LEWICE [188], developed by NASA; ONICE [70], developed by ONERA; and CANICE [153], developed at Polytechnique Montréal. These were weakly coupled with integral boundary layer solvers to estimate the convective heat transfer. This approach has been largely replaced by RANS models. Concerning the choice made by the participants of the 1^{st} IPW, most of the codes relies on RANS models. The most commonly used turbulence models were either Spalart–Allmaras [165] or k–ω (Menter [105] or variants thereof), and surface roughness effects were generally accounted for.

While RANS simulations improve the fidelity of the results, they require extremely fine grid resolution near the wall, which significantly increases the computational cost. Nonetheless, the steady growth in computational resources has made such simulations feasible. However, this approach also complicates mesh generation, particularly for 3D geometries, which partially explains the low number of participants performing multi-step simulations in 3D cases.

Inviscid solvers address both of these issues (computational cost and mesh generation) but rely on integral boundary layer solvers. Although strong coupling between the inviscid and boundary-layer equations can theoretically reproduce RANS-level accuracy [41, 85], the icing community typically employs weak coupling schemes, as the region of interest is usually confined to the vicinity of the leading edge.

Among the 1^{st} IPW participants, IGLOO2D [142], and LEWICE are examples of codes that rely on inviscid flow solvers. Some other participants also performed inviscid simulations, primarily as a means of comparison with their RANS results. As part of the effort to evaluate the impact of using different aerodynamic models, the IPW committee included the wall pressure distribution over the clean geometry as one of the comparison metrics. The results showed a generally good agreement among participants.

In contrast, the wall heat transfer coefficient h_{tc} was one of the variables that exhibited the most significant differences across participants. This is mainly because, even when RANS simulations are conducted, the airflow solver typically does not provide this variable as an output. As a result, it is often necessary to perform two separate RANS simulations or rely on approximation methods to estimate it [141]. A more detailed analysis is presented in Chapter 5.

2.1.3 Droplet Trajectory Model

Accurately predicting droplet impingement on exposed surfaces is a critical aspect of ice accretion simulations, as it determines where and how much water is available to freeze. The amount and size of droplets suspended in the airflow are typically characterized by two key parameters: the Liquid Water Content (LWC) and the Median Volume Diameter (MVD). The LWC quantifies the mass of liquid water per unit volume of air and depends strongly on the atmospheric conditions. For instance, LWC values can range from as low as 0.001 $\mathrm{g/m^3}$ in cirrus clouds to over 3 $\mathrm{g/m^3}$ in cumulonimbus [151]. The MVD represents the diameter that splits the droplet volume distribution in two equal halves and is used to characterize the droplet size spectrum.

The typically low values of LWC found in icing conditions imply very small liquid volume fractions in the air, commonly denoted by α. This parameter represents the ratio between the volume of droplets and the total volume of air. In most atmospheric icing scenarios, α is on the order of $3 \cdot 10^{-6}$ or lower. This allows the use of a one-way coupling assumption [59]: the droplets, representing the dispersed phase, are transported by the airflow, but they do not influence the flow field nor interact with each other. This assumption greatly simplifies the droplet trajectory computation and the entire ice accretion simulation.

Additionally, the motion of the water droplets is assumed to be only affected by aerodynamic forces and gravity [192]. Under typical icing conditions, the aerodynamic force is dominant, leading some models to neglect gravitational effects altogether. This aerodynamic force is commonly expressed through the drag coefficient, $\mathrm{C_D}$, of a droplet, which is generally assumed to be spherical. Figure 2.3 illustrates the evolution of $\mathrm{C_D}$ as a function of the particle (droplet) Reynolds number $\mathrm{Re_p}$, showing several models proposed in the literature as well as experimental data. For instance, the Schiller-Naumann model [121] is used in both the two- [176] and three-dimensional [141] solvers developed at ONERA.

The main output of droplet trajectory simulations is the collection efficiency, denoted by β.

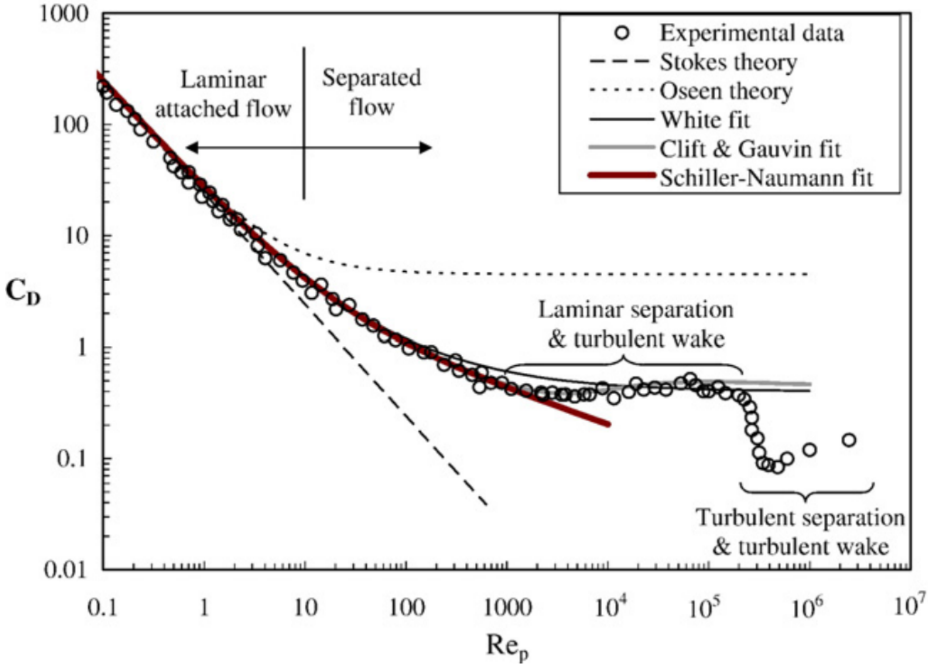


Figure 2.3 Evolution of the flow regime and the drag coefficient of a rigid sphere particle, C_D as a function of the Reynolds number of the particle Re_p. Image from [97].

This dimensionless quantity represents the ratio between the water droplet mass flux on the surface to that of the freestream. As an illustrative example, Fig. 2.4 shows the collection efficiency distribution on a swept wing.

From a numerical perspective, two main approaches are employed to compute droplet trajectories: the Lagrangian and the Eulerian approaches (see, e.g., [161] for a thorough review).

In the Lagrangian method, droplets are treated as individual entities, and their trajectories are computed based on Newton's second law. This approach is particularly efficient when the airflow is solved using potential methods, as no volume mesh is required. However, in other cases, the computational cost is typically one of the main drawbacks of the method, primarily for two reasons. First, since droplet positions rarely coincide with the flow grid, costly numerical procedures are generally required to determine the exact location of each droplet within the mesh and to perform the necessary interpolations. Second, parallelizing the computation is generally challenging [160]. Additionally, the seeding region must be sufficiently large, and the seeding density high enough, to ensure adequate resolution and statistical convergence over the entire impingement area, which often involves a large number of droplets.

In the Eulerian approach, the dispersed phase is treated as a continuous field. Under this

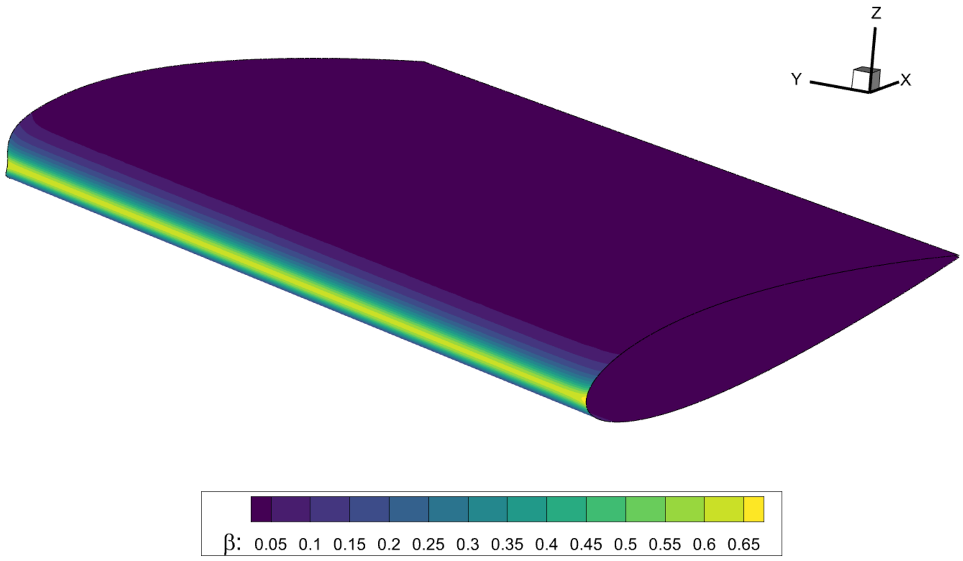


Figure 2.4 Collection efficiency distribution on a 30º swept wing. Data from a computation with IGLOO3D [142].

assumption, their transport is typically described by solving a hyperbolic partial differential equation for the volume fraction and droplet velocity, in a manner similar to how the Euler equations are computed [1]. A major advantage of this method is its natural compatibility with the volume mesh used by the airflow solver, which simplifies implementation and enables straightforward parallelization. Nevertheless, certain physical phenomena related to SLD, such as droplet rebound, splashing, and breakup, are more challenging to represent within an Eulerian framework compared to a Lagrangian one [71].

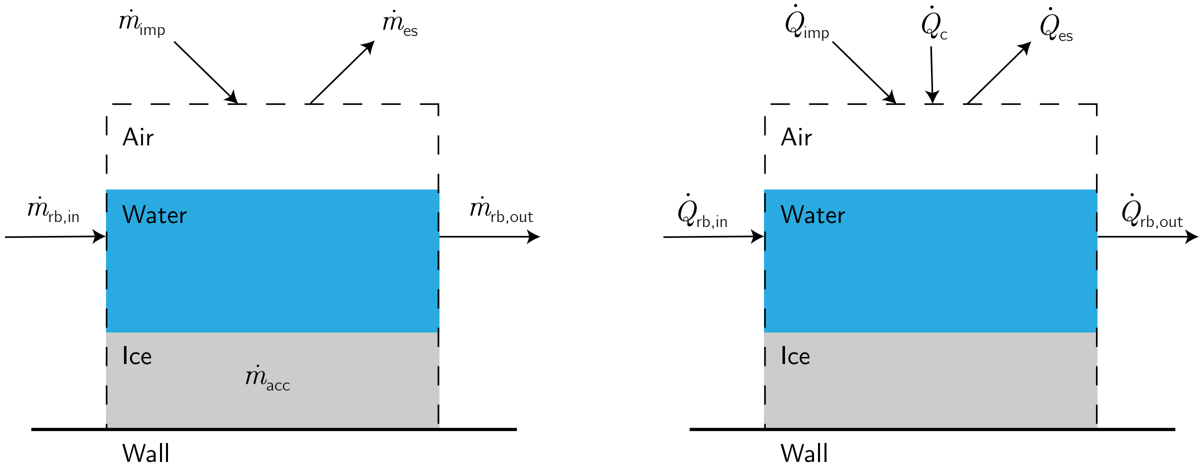
A notable shift is observed when comparing the droplet approaches reported in the 1^{st} and 2^{nd} IPW datasets. The 1^{st} IPW showed an even split between Eulerian and Lagrangian approaches (13 each), reflecting no clear preference within the community at the time. In contrast, the 2^{nd} IPW shows a strong predominance of Eulerian formulations, with 13 participants adopting this approach and only 4 using Lagrangian methods. This pronounced shift may be partly explained by the increased focus on three-dimensional cases in the 2^{nd} IPW, where the computational cost of Lagrangian tracking can become significantly higher.

---

[1] The main differences between the droplet transport equations and the Euler equations lie in two aspects: the absence of a pressure term, and the presence of a source term in the momentum equation. As a result, the conservation equations are not strictly hyperbolic, and a mathematical relaxation procedure is often required [161].

2.1.4 Ice Accretion Model

The Messinger model [106] is one of the most widely used approaches in ice accretion codes (see, e.g., [88] for an almost comprehensive review of thermodynamics models for 2D ice accretion simulation). It consists of mass and energy balances at the icing surface, accounting for droplet impingement, liquid-water runback, phase changes (evaporation, sublimation, and solidification), and convective heat exchange with the surrounding airflow, as illustrated in Fig. 2.5.



(a) Mass balance (b) Energy balance

Figure 2.5 Mass and energy balance in Messinger approach.

The model typically assumes an adiabatic wall, which is generally valid as long as heated surfaces, such as those commonly used in IPS, are not considered. In addition, under Appendix C icing conditions, where neither SLDs nor ice crystals are present, the entire impinging droplet mass is treated as deposited mass. In a general presentation of the model, no water accumulation is considered.

The mass balance considers incoming water through both droplet impingement $\dot{m}_{\mathrm{imp}}$ and incoming runback $\dot{m}_{\mathrm{rb,in}}$. It is balanced by outgoing runback $\dot{m}_{\mathrm{rb,out}}$, mass loss due to evaporation or sublimation $\dot{m}_{\mathrm{es}}$, and the accumulation of ice through solidification $\dot{m}_{\mathrm{acc}}$. All mass rates are expressed in kg/(m^{2}s):

$$\dot{m}_{\mathrm{imp}} + \dot{m}_{\mathrm{rb,in}} = \dot{m}_{\mathrm{rb,out}} + \dot{m}_{\mathrm{es}} + \dot{m}_{\mathrm{acc}} \tag{2.1}$$

Note that, in the original Messinger model, any unfrozen water inside a face must be transferred to the next face as runback water, meaning that liquid water cannot accumulate inside a face.

The energy balance accounts for the incoming runback flow $\dot{Q}_{\text{rb,in}}$, the outgoing runback $\dot{Q}_{\text{rb,out}}$, the cooling effect due to droplet impingement $\dot{Q}_{\text{imp}}$, latent heat losses due to evaporation or sublimation $\dot{Q}_{\text{es}}$, and convective heat exchange with the air flow $\dot{Q}_{\text{c}}$. The combination of these terms determines the variation of the total energy of the water system (ice and liquid water) ΔU:

$$\Delta U = \dot{Q}_{\text{imp}} + \dot{Q}_{\text{rb,in}} - \dot{Q}_{\text{rb,out}} - \dot{Q}_{\text{evs}} + \dot{Q}_{\text{c}} \tag{2.2}$$

Once the convective heat transfer and droplet impingement rates are known (obtained from airflow and droplet trajectory simulations), the system of Eqs. 2.1 and 2.2 contains three unknowns: the wall temperature T_{w}, the surface ice accretion rate $\dot{m}_{\text{acc}}$, and the outgoing water runback flux $\dot{m}_{\text{rb,out}}$. To close the system, an additional thermodynamic relation must be introduced, which depends on the state of the surface:

- Rime-ice regime: All incoming water freezes upon impact, resulting in no outgoing runback, i.e., $\dot{m}_{\text{rb,out}} = 0$.
- Glaze-ice regime: A part of the incoming water freezes, while the rest forms a liquid film that runs back along the surface. In this case, $\dot{m}_{\text{rb,out}} > 0$, and the wall temperature reaches the melting point, $T_{\text{w}} = T_{\text{m}}$.
- Liquid-surface regime: No ice forms on the surface, meaning that $\dot{m}_{\text{acc}} = 0$, and all incoming water remains in liquid form.
- Dry surface: No water is present on the surface, so no ice accretion occurs, and consequently, $\dot{m}_{\text{acc}} = 0$.

The model was initially presented in two-dimensional geometries with a single stagnation point, making the direction of the runback straightforward: from the stagnation point to the trailing edge. However, in complex two-dimensional geometries with multiple stagnation points, or in three-dimensional cases, the direction of the runback must be defined. In these cases, a commonly used option is the direction of skin friction. Additionally, in three-dimensional surfaces, the alignment between the mesh and the runback direction is generally lost; therefore, there is no longer a direct relation between the variation of the structured wall index i and the direction of the runback flow. As a result, an iterative method is often employed to obtain a converged solution of the thermodynamic balance, which is referred to as the iterative Messinger model [195].

Several alternatives to the original Messinger model have been proposed to improve the modeling of the freezing process. Myers et al. [117] extended the Messinger model by incorporating heat conduction through the ice layer, accounting for the temperature difference between the airfoil surface and the ice surface.

Another significant development is the Shallow Water Icing Model (SWIM) presented by Bourgault et al. [26], which is based on lubrication theory. This model treats the freezing process in an unsteady manner, extending the range of applicable conditions at the expense of significantly higher computational cost compared to approaches based on Messinger's energy balance.

Regarding the models used by the participants of the 1^{st} IPW, the majority employed a Messinger-based model, while only one participant used the SWIM model.

2.1.5 Ice Density Model

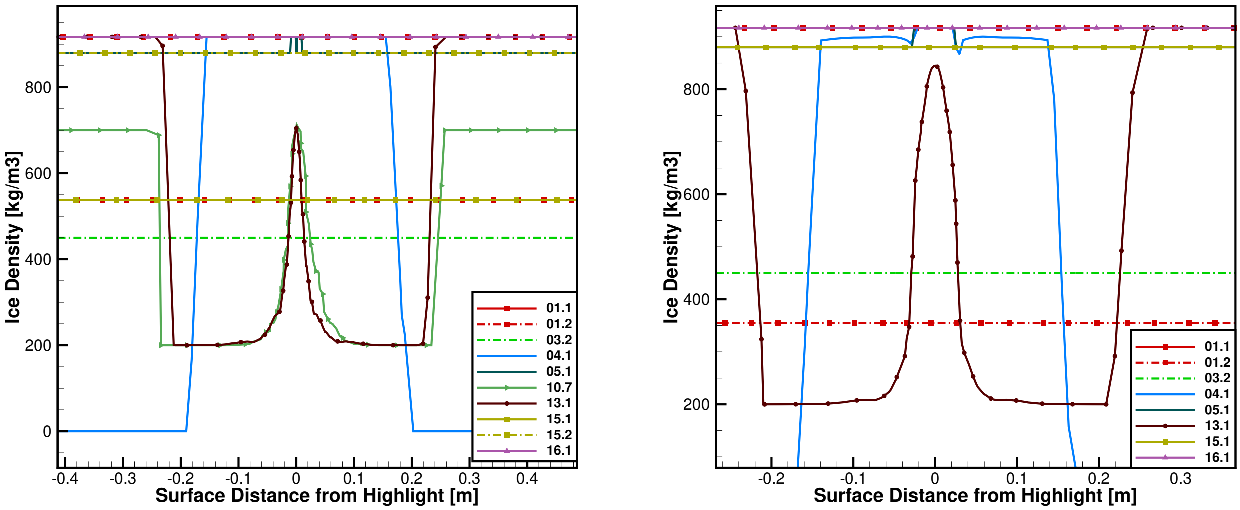
In numerical simulations, the density model directly plays a key role, as it affects the predicted ice thickness h_{ice}, and consequently, the resulting ice shape:

$$h_{\text{ice}}(\Delta t) = \frac{\dot{m}_{\text{acc}} \Delta t}{\rho_{\text{ice}}} \tag{2.3}$$

where Δt is the ice accretion time step and ρ_{ice} is the ice density, which can vary significantly depending on the ice accretion conditions [180].

For unswept wings, typically modeled as two-dimensional, a value of 917 kg/m^3 is commonly used under glaze icing conditions. Under rime icing conditions, however, the ice tends to be more porous, and lower values are often employed. For instance, IGLOO3D [141] uses the density model of Makkonen and Stallabras [102], and constrains the density values between 750 and 917 kg/m^3. For swept wings, the bulk density is affected not only by the porosity of the ice but also by the scallop structure, whose voids are challenging to capture. Consequently, lower values are frequently used [143]. For example, in the 2^{nd} IPW, typical values range from 350 to 450 kg/m^3 [84, 181].

Regarding the 1^{st} IPW results, one of the conclusions drawn is the variability in the ice density values used [86]. For instance, Fig. 2.6 shows the ice density values adopted by participants for cases 361 and 362, which examine ice accretion on a 30º swept wing under rime and glaze icing conditions, respectively. The figure illustrates the wide range of densities selected by different models, reflecting the diversity in assumptions related to ice microstructure and porosity.



(a) Case 361: rime icing conditions

(b) Case 362: glaze icing conditions

Figure 2.6 Cases 361 and 362 from the 1^{st} IPW [86]. Comparison between ice density values on a 30º swept wing.

2.1.6 Surface Deformation Model

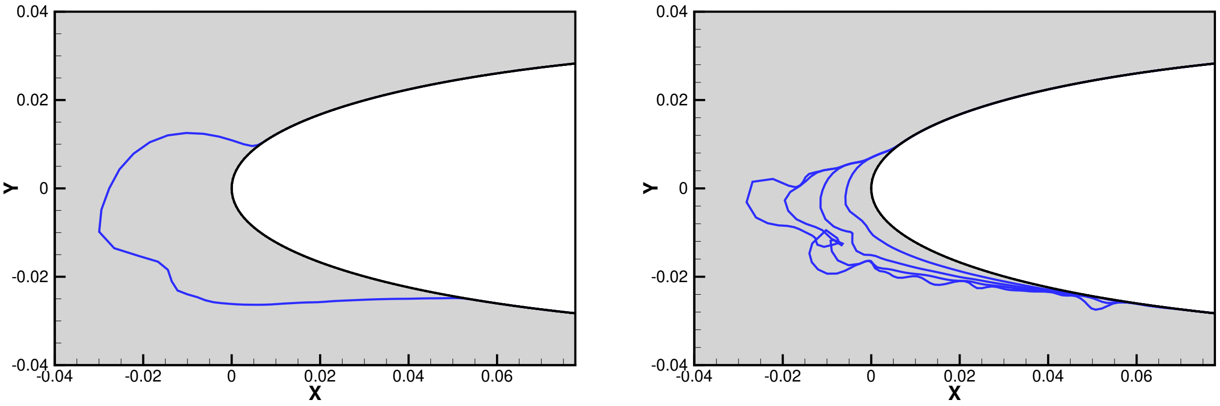
The final step of the ice accretion loop consists in updating the geometry based on the computed ice thickness. The most widely adopted approach so far has been to apply a Lagrangian displacement of each node of the discretized surface along the normal direction $\vec{n}$, i.e.,

$$\vec{x}_{\text{new}} = \vec{x}_{\text{old}} + h_{\text{ice}}\vec{n} \tag{2.4}$$

where $\vec{x}$ denotes the position of a surface node.

A limitation of this method may arise in multi-step simulations, particularly when complex ice shapes are formed: it may lead to self-intersections (commonly referred to as overlaps), which may significantly affect the robustness of the simulation [27]. For instance, Fig. 2.7 shows the predicted ice shape for a two-dimensional rime case obtained from one- and five-step simulations. In the case of a one-step simulation, the Lagrangian displacement updates the surface nodes effectively. However, in a five-step simulation, the final iteration cannot be completed, as the overlaps generated during the fourth step prevent the generation of the next volume mesh.

To improve the robustness of ice accretion simulations, several research groups have explored alternatives to the Lagrangian displacement method [124]. One of the most widely adopted strategies consists in solving the level-set equation. Originally introduced by Osher and



(a) Single layer

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

Figure 2.7 Example of self-intersection in a 2D multi-step ice simulation. Image from [87].

Sethian [125], the level-set equation is a simplified form of the Hamilton-Jacobi equation used to track evolving implicit interfaces. It provides an Eulerian description of the evolution of the signed distance, represented as a scalar variable ϕ, which is advected according to a propagation velocity field $\vec{V}$. The level-set equation reads:

$$\frac{\partial \phi}{\partial t} + \vec{V} \cdot \vec{\nabla} \phi = 0 \tag{2.5}$$

and is typically solved using upwind schemes without the need for artificial dissipation [27].

For multi-step ice accretion simulations, Eq. 2.5 is solved until the defined ice accretion time step Δt is reached. At this point, the updated geometry is implicitly defined by the level-set field. Specifically, the new interface corresponds to the iso-surface where $\phi = 0$. To proceed with subsequent iterations, a surface mesh must then be generated. Extracting this iso-surface is a nontrivial task, particularly when a high-quality surface mesh is required. Common numerical techniques used for this purpose include the marching cubes algorithm [96] and marching tetrahedra [175]. If needed, a B-spline geometry representation can be employed to improve the surface mesh [28].

Alternatively, Radenac et al.[141] propose a hybrid approach, where a Lagrangian displacement of the surface mesh is performed based on the level-set field. In this method, the ice growth direction $\vec{n}_\phi$ is defined in the volume mesh as:

$$\vec{n}_\phi = \frac{\vec{\nabla} \phi}{||\vec{\nabla} \phi||}. \tag{2.6}$$

When N_L sub-iterations are considered, the displacement of each node at iteration k is indeed given by the following equation:

$$\Delta\vec{x}^k = \frac{h_\mathrm{ice}}{N_\mathrm{L}}\vec{n}_\phi(\vec{x}^k) \tag{2.7}$$

where $\Delta\vec{x}^k$ is the displacement of each node, and $\vec{n}_\phi(\vec{x}^k)$ is the interpolated normal vector at the current position $\vec{x}^k$.

Level-set methods have gained attention in recent years for ice accretion simulations, leading in the development of various numerical approaches [7, 28, 56, 57, 90, 141]. However, this interest has not yet been reflected in the choices of participants in the first two IPW, where alternatives to Lagrangian displacement remain rare and have so far been limited to contributions from Politecnico di Milano [54, 113].

Another approach to updating the iced geometry relies on an advancing front method coupled with a stochastic model, also known as a morphogenetic or nondeterministic model. Originally proposed by Szilder and Lozowski [169], this morphogenetic approach simulates ice accretion by incorporating stochasticity and accumulating ice in discrete building blocks, analogous to pixels, on a Cartesian grid. Stochasticity is introduced in two key aspects of the icing simulation process: (1) droplet impingement and (2) the random motion and freezing of fluid elements on the surface [168]. Unlike traditional methods, the ice density is not prescribed as an input but emerges as an output of the simulation. A comprehensive analysis of the morphogenetic model can be found in the PhD thesis of Bourgault-Côté [27]. Morphogenetic simulations in previous IPWs have so far been limited to work led by Szilder's team at the National Research Council Canada [67] and research efforts at Polytechnique Montréal [24].

2.2 Immersed Boundary Methods

The Immersed Boundary Methods (IBMs) refers to any approach that solves the governing equations on a computational mesh that does not conform to the boundaries describing the geometry. The boundary condition is then recovered implicitly by locally forcing the solution in the vicinity of the obstacle. Immersed Boundary Methods thus have the advantage of reducing the cost of the mesh generation step, particularly to deal with complex geometries in CFD. Their implementation on automatically generated adaptive Cartesian grids eliminates the bottleneck of the mesh generation step. They can also be applied to a body-conforming mesh that represents a simplified geometry, in order to take into account for local details.

The origins of IBMs trace back to the pioneering work of Charles Peskin in the early 1970s [133]. Motivated by the need to simulate blood flow through the human heart, Peskin introduced a method to represent flexible, moving boundaries within a fixed Cartesian grid by adding force terms in the incompressible Navier–Stokes equations to account for the presence of boundaries. Since then, the popularity of IBMs in Computational Fluid Dynamics (CFD) has increased significantly, especially over the last two decades. To illustrate this trend, Fig. 2.8 shows the number of annual publications that include the term "immersed boundary" or "immersed boundaries" in the title, abstract, or keywords, based on a search conducted using a similar methodology to that of [182]. The data, updated to include recent years, confirms the sustained growth in interest in IBMs. It is worth noting that this analysis underestimates the actual number of studies involving IBMs, as many works adopt alternative terminology (such as embedded boundaries, immersed interfaces, immersed body, penalty method or ghost-cell method among others), or apply IBMs-related concepts without explicitly referring to them.

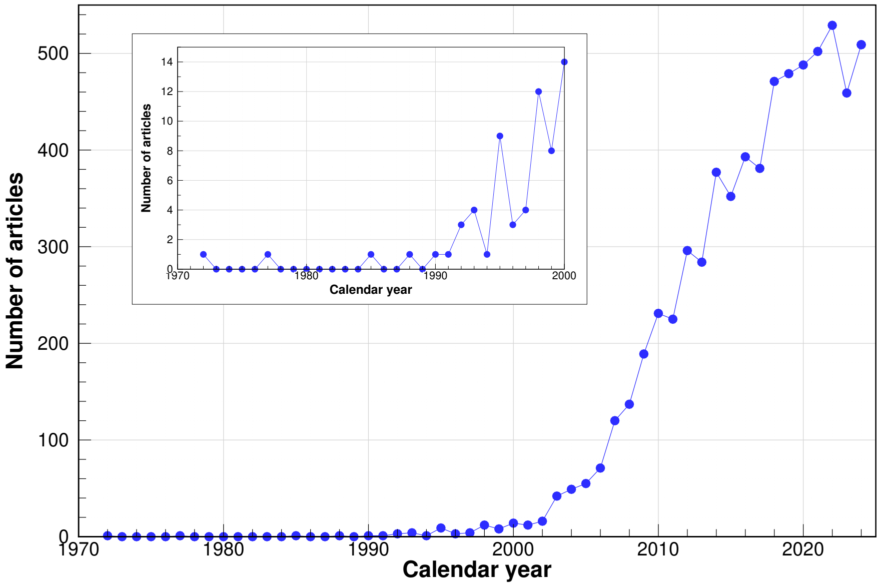
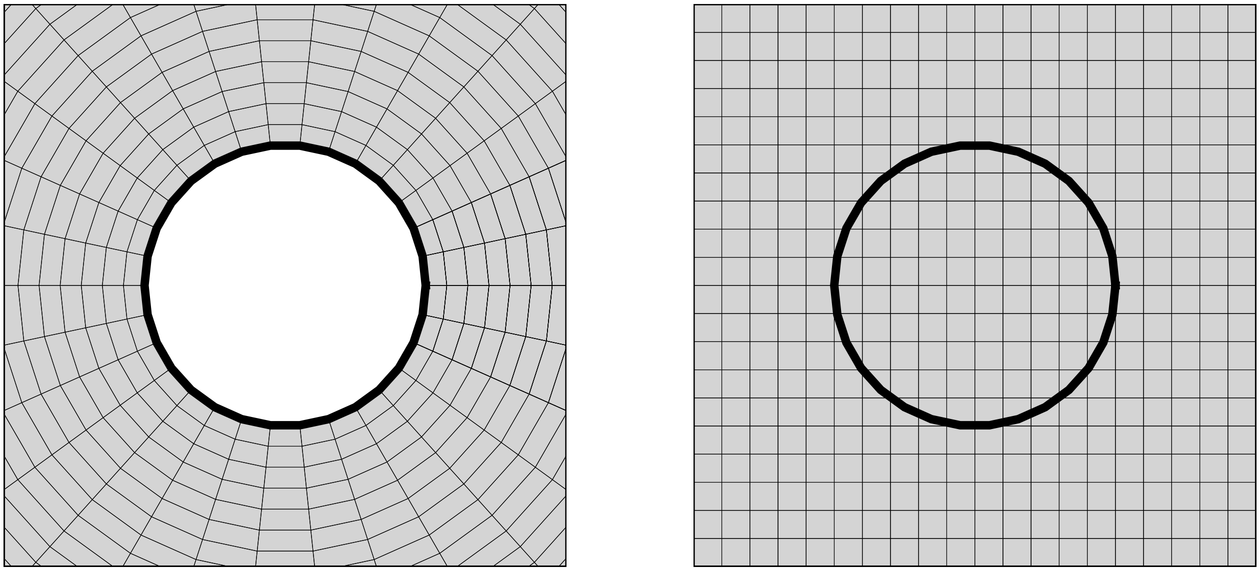


Figure 2.8 Number of publications per year containing the terms "immersed boundary" or "immersed boundaries" in the title, abstract or keywords. Data from Scopus (accessed May 16, 2025).

Immersed Boundary Methods have been primarily employed in the simulation of fluid dynamics. Applications span a wide range of flow regimes, from inviscid flows [31, 47] to turbulent flows [73, 170]. They have also been used in the simulation of multiphase flows [127, 184]. In addition, IBMs have been extensively applied to fluid–structure interaction problems [72, 174, 191]. Their use has also been extended to other areas, including solid mechanics [144] and electromechanical systems [10], among others.

2.2.1 Immersed Boundary Meshes

The main advantage of IBMs is that they allow the use of general meshes that are independent of the geometry under study. For instance, Fig. 2.9 compares a traditional BF mesh with an IB mesh for the simulation of the flow around a two-dimensional cylinder. While IBMs are compatible with arbitrary mesh types, Cartesian meshes are particularly advantageous, as they enable the application of high-performance structured solvers, which significantly improves the overall efficiency of the simulation. The key benefit of using IBMs is clear: the same mesh can be employed regardless of whether the target geometry is a cylinder, an airfoil, or any other complex shape.



(a) Body-Fitted

(b) Immersed Boundary

Figure 2.9 Different type of meshes to simulate flow around a 2D cylinder.

2.2.1.1 Mesh Adaptation on Cartesian Grids

While Cartesian meshes offer advantages such as simplicity, ease of generation, and high-performance numerical methods, they present a significant challenge when it comes to performing local refinement in a targeted zone. In contrast, Body-Fitted (BF) structured meshes naturally incorporate information about the wall geometry, explicitly defining both the normal and tangential directions relative to the surface. This allows for anisotropic mesh refinement, particularly in the direction normal to the wall, where gradients of key flow variables tend to be the largest. As a result, very fine cells can be concentrated near the surface, while coarser cells can be used farther away, where the influence of the solid boundary is minimal and flow gradients are expected to be small. This limitation of using Cartesian meshes without an adaptation technique becomes even more critical in high Reynolds number flows, where resolving the boundary layer accurately requires extremely fine resolution in the direction normal to the surface, typically requiring the use of wall models [76, 116, 150].

The challenge of achieving local refinement in Cartesian grids has been the focus of numerous studies. Among these, two techniques stand out:

- Adaptive Mesh Refinement (AMR): Originally proposed by Berger and Oliger [20], AMR dynamically increases the mesh resolution in regions of interest based on a predefined criterion, often represented by a sensor field. In its basic formulation, each tagged cell in the Cartesian mesh is subdivided into N isotropic cells, with $N = 4$ in two dimensions and $N = 8$ in three dimensions. To ensure a smooth transition between refinement levels, adjacent cells may also be subdivided in accordance with prescribed mesh regularity constraints. The most basic sensor is simply the location of the fluid–solid interface (see Fig. 2.10a). The resulting mesh is typically unstructured and the number of neighbors per cell is no longer constant, which also complicates data management and increases the complexity of numerical algorithms. A key drawback of AMR is that it typically requires significant modifications to the solver, as the structured nature of the Cartesian mesh is lost. However, AMR benefits for a high level of flexibility. This is a widely adopted refinement strategy in the context of Immersed Boundary Methods [34, 50].

- Octree-Based Mesh Adaptation: This approach was initially introduced by MacNeice et al. in PARAMESH [98]. Starting from one root cell representing the whole computational domain, an octree is built by recursively dividing the root cell into N children cells until the desired resolution is reached. Again, $N = 4$ and $N = 8$ for two- and three-dimensional meshes, respectively. Examples of the application of octree-based refinement technique to IBMs are presented in [118, 139]. An adaptation of this method has been built inside Cassiopee [138], accounting for curved bodies. This approach uses the overset grid methods [166], also referred to as Chimera methods [18] to build an octree structure based on overlapping uniform Cartesian meshes. As an illustration, Fig. 2.10b shows the resulting octree-based mesh using Cassiopee. As the resulting mesh is structured and isotropic, the solver can benefits from the high-performance achieved when performing CFD computation on uniform Cartesian blocks; however, the solver must be capable of handling overset grids.

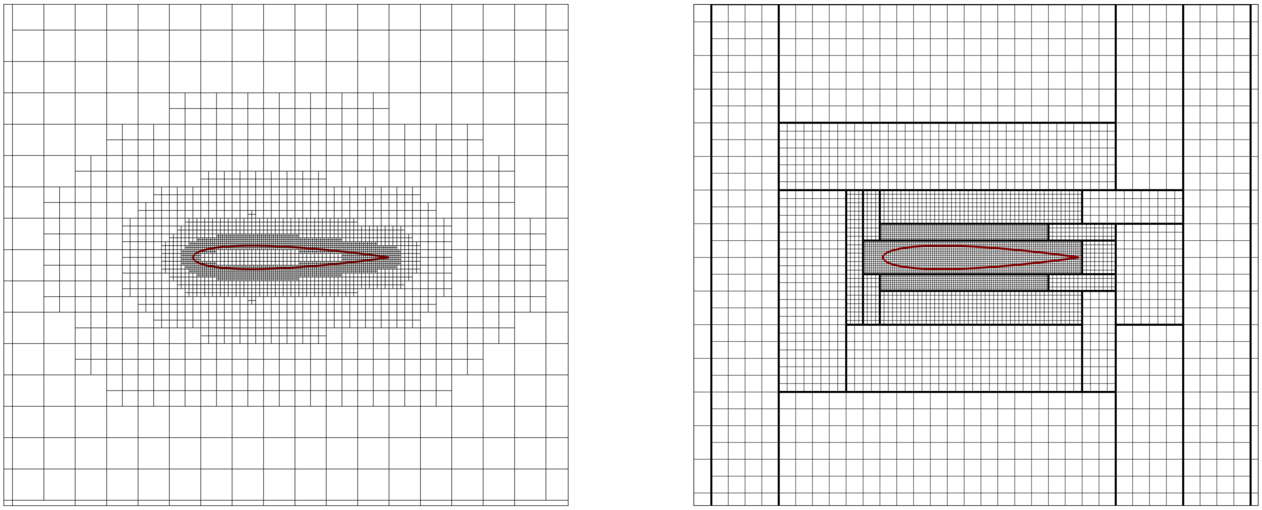

(a) Adaptive Mesh Refinement (b) Octree

Figure 2.10 Representative mesh adaptation strategies commonly applied in IB simulations.

Recent internal studies conducted at ONERA have compared the performance of both mesh refinement techniques in practical applications for the same mesh resolution. In particular, simulations using AMR within the unstructured solver CODA[2] were compared to those using octree-based refinement with the structured solver FAST. In both cases, mesh adaptation was performed using Cassiopee. The findings indicate that, although AMR typically results in a significantly lower number of cells (often less than half of those required by the octree), the performance of the overall simulation is comparable to the octree-based approach. This is primarily due to the high computational efficiency achievable when solving fluid dynamics problems on uniform Cartesian blocks. As a result, the increased number of cells required by octree-based refinement is often offset by faster solver execution, leading to comparable overall performance. These results highlight that the choice between the two strategies is not straightforward.

2.2.1.2 Local Immersed Boundary on Body-Conforming Meshes

For some applications, it is more advantageous to apply the IBMs locally, i.e., only to a portion of the geometry, an approach sometimes referred to as hybrid IBM in the literature [196]. Typical examples involve control devices. For instance, Mochel et al.[110] employ an IBM exclusively for the control device of a simplified space launcher afterbody, thereby avoiding the difficulties associated with generating structured body-fitted meshes around the control

---

[2] CODA is an unstructured CFD software jointly developed by ONERA, DLR and Airbus.

device. Similarly, Zhu et al.[196] apply an IBM to the moving trailing-edge flap of a NACA 0012 airfoil to study its unsteady behavior on a fixed grid. As an illustration, Fig. 2.11 presents a zoomed view of the structured mesh around the trailing edge.

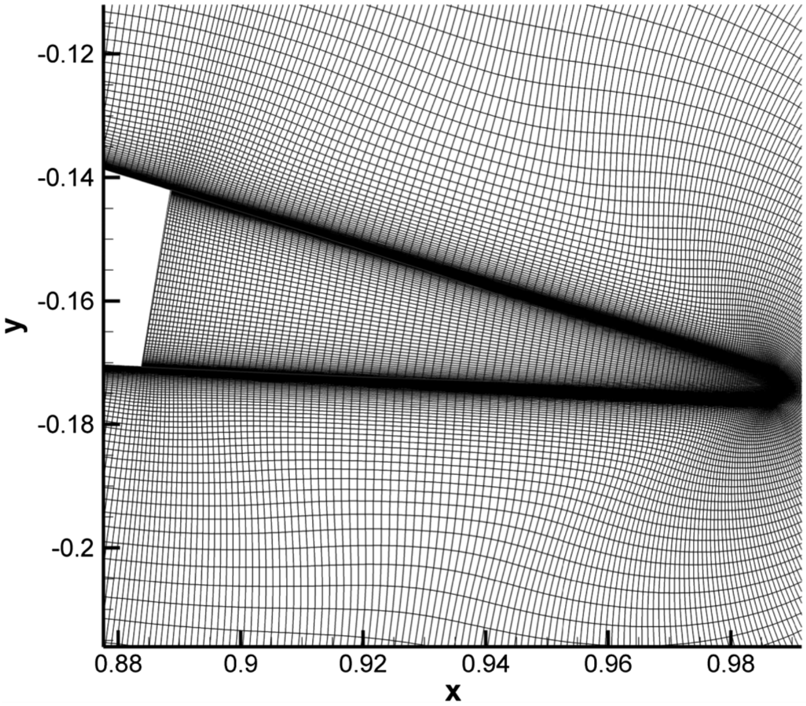


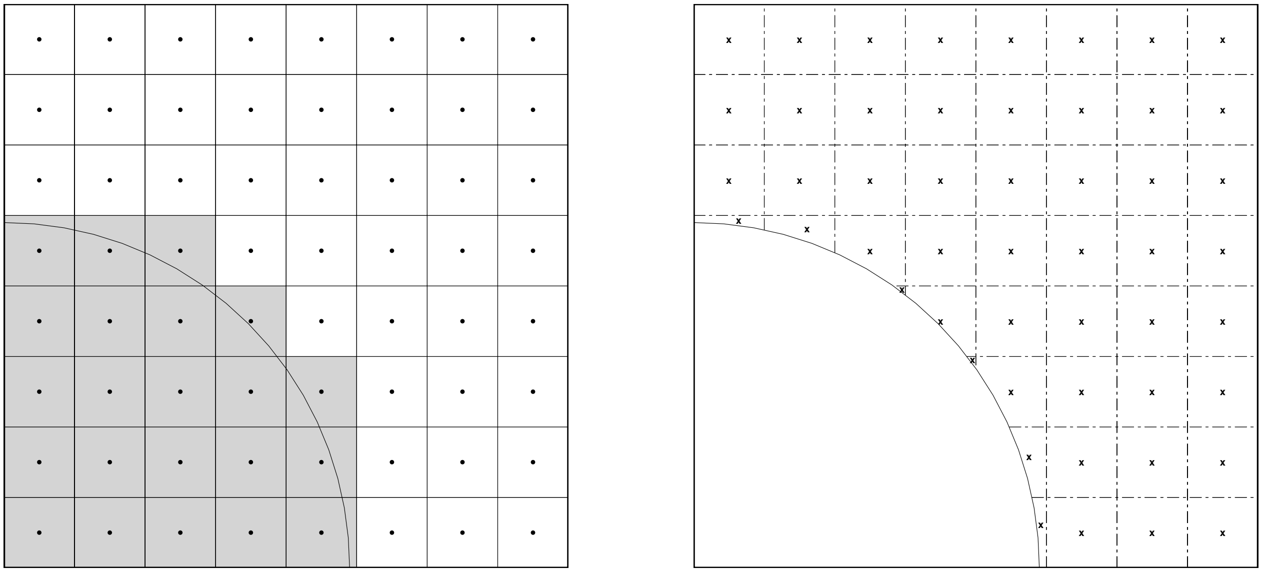
Figure 2.11 Mesh configuration used in the hybrid IBM approach. The volume mesh corresponds to a BF mesh from the clean airfoil, while IBMs are restricted to the trailing-edge flap. Image from [196].

2.2.2 Immersed Boundary Types

There is no universally accepted classification of IBMs in the literature [109, 164], as various authors propose different categorizations depending on the context, the governing equations, or the specific application. Following the classification proposed by Lavoie [87], IBMs can be grouped into cut-cell, continuous forcing, and discrete forcing methods. In the following, each of these categories is briefly described.

2.2.2.1 Cut-Cell Method

The first category of IBMs features a single approach: the cut-cell method. Originally introduced by Clarke et al. [43] in 1986 for the simulation of two-dimensional inviscid flows, it has since been extended to a broad range of aerodynamic applications involving rigid bodies, viscous [114] and turbulent [107] flows, as well as multiphase flows with moving boundaries [190]. In this method, cells cut by the immersed boundary are truncated so that they conform to the body shape of the boundary surface, as illustrated in Fig. 2.12, thereby imposing the boundary conditions as in a BF approach. Because this method involves exclusively direct modification of the computational mesh, some authors do not classify it as an IBM.


(a) Tagging of solid cells.

(b) Resulting cut-cell mesh.

Figure 2.12 Cut-cell Method. Domain identification (left) and cut mesh (right). Solid cells are shown in gray, and fluid cells in white.

The method therefore focuses on a pre-processing stage. Typically applied to Cartesian meshes, it begins by identifying the cells intersected by the IB, followed by the trimming of the mesh.

The main advantage of this approach lies in its strict global and local conservation of mass, momentum, and energy, as well as in its good accuracy near sharp-interface boundaries, which are, in general, common issues in IBMs. However, two main challenges arise, which become more severe in 3D simulations. The first issue is related to the morphology of the cut cells. For instance, in the case shown in Fig. 2.12, the intersection of the IB with the Cartesian grid results in various polygonal shapes near the geometry, such as triangles, trapezoids, or pentagons. Consequently, the solver must be able to handle general polyhedral elements, typically losing the structured nature of the solver. On the other hand, cut cells may be arbitrarily small, which has several implications. Since the Courant number is inversely proportional to the cell dimension, very small cells may require significantly reduced time steps to preserve numerical stability when explicit time integration schemes are used. In addition, the strong variation in cell sizes can lead to stiffness in the resulting linear systems. To avoid this problem, numerical mesh techniques can be employed such as cell merging [75, 178] or cell linking [81].

2.2.2.2 Continuous Forcing Methods

The second category of IBMs is the class of continuous forcing methods. In this approach, a distributed force term is introduced directly into the continuous form of the equations to represent the interaction between the fluid and the immersed boundary [109].

To illustrate this, the mass and momentum equation of the incompressible Navier-Stokes equations are considered:

$$\nabla \cdot \vec{u} = 0 \tag{2.8a}$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{u} + \vec{f} \tag{2.8b}$$

where $\vec{u}$ is the velocity field, ρ the fluid density, p the pressure, ν the kinematic viscosity and $\vec{f}$ represents the additional forcing term introduced to model the presence of the immersed boundary. Several expressions for $\vec{f}$ are possible, depending on the specific IBM employed.

In the following, a detailed analysis of two continuous forcing methods is presented. The first corresponds to Peskin's original immersed boundary method, whose applicability is essentially restricted to elastic boundaries, whereas the second is the penalization method, intended for rigid boundaries.

Peskin's Immersed Boundary Method

The original IBM of Peskin [133] was designed to model the elastic boundary of the heart. In this method, the IB is represented in a Lagrangian manner, by a set of nodes. Each node of the boundary, $\vec{X}_k$ is subjected to external (hydrodynamic) and internal (boundary deformation) loads, represented by a total force $\vec{F}_k$. The influence of each node to the surrounding cells of the volume mesh $\vec{x}_{ij}$ is made through a smooth distribution function $d(|\vec{x}_{ij} - \vec{X}_k|)$, as shown in Fig. 2.13. In this manner, the force applied to the IB $\vec{F}_k$ is spread to the surrounding fluid cells via $\vec{f}$:

$$\vec{f}(\vec{x}_{ij}, t) = \sum_k \vec{F}_k(t) d(|\vec{x}_{ij} - \vec{X}_k|) \tag{2.9}$$

The choice of the distribution function d is a key aspect and it varies between different methods [133, 154].

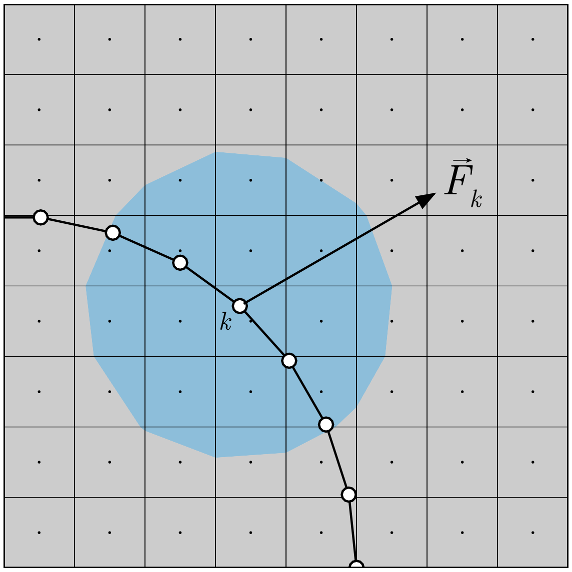


Figure 2.13 Transfer of forcing $\vec{F}_k(t)$ from the IB node k to surrounding fluid cells. Shaded region represent the extent of the force distribution.

Although there exist adaptations of this method for rigid boundaries, its application to flows involving rigid bodies presents significant challenges. This is primarily because the constitutive laws formulated for elastic boundaries are generally ill-posed in the rigid limit [109], generally leading to stability issues.

Penalization Immersed Boundary Method

For flows involving rigid boundaries, the volume penalization method constitutes a commonly adopted alternative continuous forcing approach. It was initially introduced by Arquis and Caltagirone [12] to simulate flow through a porous medium using Darcy's law. Later, it was proven that the method can be applied to the Navier-Stokes equations with no-slip boundary conditions [11], modeling the solid interface as a porous medium with very low permeability.

Here, a source term, typically referred to as the penalization term, is introduced and is activated only in the solid domain, therefore, requiring a mask function χ to distinguish between the fluid Ω_f and the solid Ω_s region:

$$\chi(\vec{x},t) = \begin{cases} 1, & \text{if } \vec{x} \in \Omega_s, \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

In the case of the incompressible Navier–Stokes equations (Eq. 2.8), the penalization term takes the following form:

$$\vec{f} = \frac{\chi}{\eta}(\vec{u}_{\mathrm{bc}} - \vec{u}) \tag{2.11}$$

where $\vec{u}_{\mathrm{bc}}$ is the boundary condition for velocity in the solid interface and η is the penalization parameter, which has units of time. By defining η small enough, the rest of the terms are negligible in front of the penalization terms, and the boundary conditions is recovered inside the solid:

$$\vec{u}_{\mathrm{bc}} - \vec{u} = 0 \tag{2.12}$$

Note that the physical interpretation of volume penalization method is that the solid is assumed to be a porous medium with sufficiently small permeability in the solid. In this way, reducing the value of the penalization parameter effectively decreases the permeability inside the solid region. This leads to a stronger enforcement of the boundary conditions, as the fluid velocity is more strongly damped within the solid. As a result, the accuracy of the overall solution is improved. However, small η leads to very stiff source terms, compromising the stability of the simulation. In other words, the values of η are desired to be as small as possible without compromising the stability of the solution.

One of the key advantages of the penalization method lies in its straightforward implementation, as it typically requires only the definition of the mask function to distinguish between fluid and solid cells. Another important benefit is that the penalization method has a well-established theoretical foundation. Rigorous mathematical analyses have demonstrated the convergence of the method as the penalization parameter tends to zero. In particular, it is possible to derive error estimates that quantify the influence of the penalization parameter on the accuracy of the solution [155]. In addition, the method is compatible with a wide range of numerical discretizations and can be directly integrated into existing solvers with minimal modifications. Its versatility, combined with low preprocessing requirements, makes it a robust and efficient choice for simulating flows involving rigid or moving solid boundaries.

Despite their advantages, penalization IBMs present several limitations. First, sharp interfaces are smeared over a region comparable to the local mesh width, which reduces the accuracy of the solution near the boundary, i.e., boundary conditions are applied on an approximation of the IB. As a result, these methods are typically limited to first-order accuracy in space. However, high-order methods are becoming increasingly widespread. For instance, Kou et al. [82] proposed a high-order volume penalization approach within a Finite Element framework for solving compressible viscous flows. In their method, the distinction between solid and fluid regions is made at each solution point, corresponding to the Gaussian quadrature points within each element. This allows the penalization term to be applied in a spatially localized and continuous manner across elements, enabling a more accurate and flexible representation of the IB.

Another drawback of this type of IBM is the reliance on adjustable parameters. In the presented case, only a single penalization parameter, η, is required. However, more complex boundary conditions generally involve additional parameters. For instance, the volume penalization approach of Brown-Dymkoski et al. [30], developed for compressible viscous flows, requires three penalization parameters.

2.2.2.3 Discrete Forcing Method

One of the earliest formulations of the direct discrete forcing method within the IB framework can be attributed to the work of Mohd-Yusof [111], who in 1997 introduced a simple yet effective strategy to impose boundary conditions on Cartesian grids. Unlike Peskin's original IBM [133], the approach of Mohd-Yusof explicitly modified the discretized momentum equations near the solid interface to enforce the desired velocity field. This direct imposition of the boundary condition laid the foundation for a family of sharp-interface IBMs that would later be developed and refined for a wide range of applications.

Originally developed in the context of spectral methods, the direct discrete forcing approach was soon adapted to other numerical frameworks. In particular, it was rapidly extended to Finite Difference [64] and later to Finite Volume [80] methods.

In this approach, the discrete forcing term is applied to the time-discrete equations. For the Navier–Stokes equations, and using a forward Euler time-integration scheme, the momentum equation at an immersed boundary point can be written as [64]:

$$\frac{\vec{u}^{n+1} - \vec{u}^{n}}{\Delta t} = RHS^{n+1/2} + \vec{f}^{\,n+1/2} \tag{2.13}$$

where Δt is the time step, RHS represents the discretized form of the convective, viscous and pressure gradient terms, and the superscript n denotes the current time level. The forced term $\vec{f}^{\,n+1/2}$ is evaluated in an intermediate time step to impose the desired boundary condition $\vec{u}^{n+1} = \vec{u}_{\mathrm{bc}}$ on the immersed boundary interface Ω_{IB}:

$$\vec{f}^{\,n+1/2} = \begin{cases} \frac{1}{\Delta t}(\vec{u}_{\mathrm{bc}} - \vec{u}^{n}) - RHS^{n+1/2}, & \text{on } \Omega_{\mathrm{IB}}, \\ 0, & \text{elsewhere.} \end{cases} \tag{2.14}$$

Although this method is presented by introducing a forcing term $\vec{f}$, in practice, it is not explicitly computed. Instead, the velocity at the cell is directly overwritten with the desired boundary value. Majumdar et al. [100] interpreted this approach as a generalization of the *ghost-cell* approach, traditionally used in body-fitted simulations to fulfill the numerical stencil of the cells around the wall. However, it was Tseng and Ferziger [177] who first coined the term Ghost-Cell Immersed Boundary Method (GCIBM) for this category of IBMs. Figure 2.14 illustrates this method using a two-dimensional cylinder immersed in a Cartesian mesh. The first fluid cells around the geometry (gray cells) require fulfillment of their computational stencil. This is achieved by enforcing values in the ghost cells (red cells), thereby accounting for the presence of the IB.

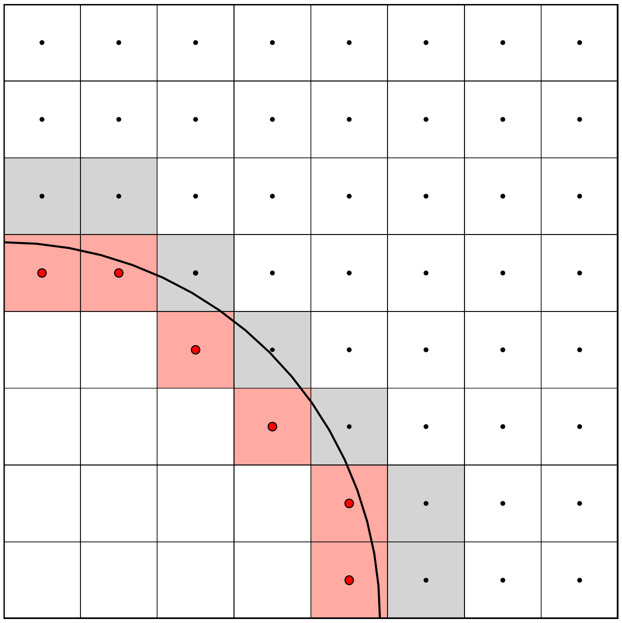
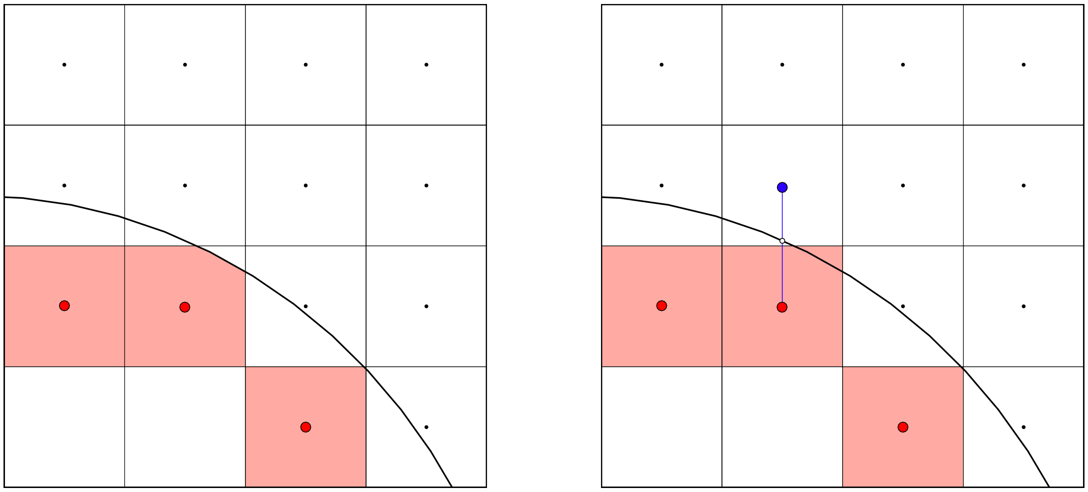


Figure 2.14 Illustration of the GCIBM applied to a two-dimensional cylinder immersed in a Cartesian grid.

Up to this point, the value assigned to ghost cells has typically been taken directly as the boundary condition $\vec{u}_{\mathrm{bc}}$. Although this represents the simplest approach to enforcing boundary conditions, it introduces significant numerical errors and consequently limits the solution accuracy to first order in the vicinity of the IB. Furthermore, the staircase representation of the IB on Cartesian grids often induces spurious oscillations near the solid interface.

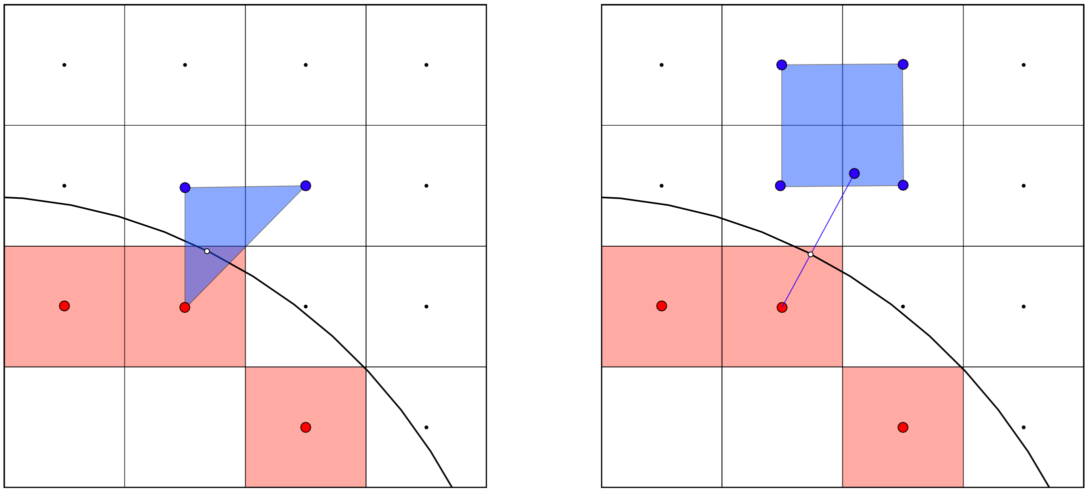
A common strategy to achieve a smoother solution consists in evaluating the volume fraction ψ_{s} of each cell occupied by the solid relative to the total cell volume ψ (see Fig. 2.15a). The ratio ψ_{s}/ψ is then employed to scale the forcing term, effectively smoothing the imposed boundary conditions.

An alternative and widely adopted approach in GCIBM involves determining the ghost-cell values by interpolation from neighboring fluid cells. Various interpolation techniques have been proposed in the literature [64, 73]. The most straightforward consists of a linear one-dimensional interpolation between the ghost cell and a nearby fluid cell, with the boundary condition $\vec{u}_{\text{bc}}$ known at the intersection point between the immersed boundary and the line connecting the two cells (see Fig. 2.15b). More sophisticated interpolations incorporate multiple fluid cells to enhance accuracy (see Fig. 2.15c) [73]. Another frequently adopted method employs the symmetric point of the ghost cell with respect to the immersed boundary, often referred to as the *image point* or *target point* (see Fig. 2.15d). Since this point generally does not coincide with a computational grid node, additional interpolation steps are required to obtain its value [48, 49, 51].



(a) Volume fraction weighting interpolation

(b) Linear one-dimensional interpolation



(c) Linear multi-dimensional interpolation

(d) Image point and multi-dimensional interpolation

Figure 2.15 Sketch of the interpolation procedures in GCIBM.

Up to this point, only one row of ghost cells has been considered to complete the computational stencil of fluid cells adjacent to the IB. However, the required number of ghost-cell rows depends on several factors, such as the numerical scheme employed and the specific boundary conditions imposed. For instance, Dadone et al. [49] use a five-cell stencil in each spatial direction to solve the Euler equations, which requires the addition of two rows of ghost cells within the solid domain. In simulations involving more complex flows or numerical methods, the number of ghost-cell rows may increase [45].

The advantages of these methods lie in their ability to accurately represent the geometry, and when appropriate interpolation algorithms are employed, they do not reduce the numerical scheme's order of accuracy. Furthermore, the solver itself requires minimal modification, as only the manner of imposing boundary conditions is altered by adding an interpolation step. However, these methods also present several drawbacks. To begin with, the preprocessing step necessary to identify ghost cells and the associated interpolation data can be non-negligible in terms of computational cost. Additionally, in certain cases, such as highly complex geometries or regions with sharp features, it can be challenging to position the required number of ghost cells [49, 118].

Despite its widespread use, the GCIBM is not the only type of discrete forcing method. For instance, Capizzano [33, 34] developed a face-based discrete forcing method. In this approach, instead of enforcing values in certain cells of the grid, the fluxes are directly imposed on the faces of fluid cells that intersect the IB. The fluxes at these so-called *IB-faces* are reconstructed using a target (or image) point, in a manner similar to ghost-cell methods (see Fig. 2.16). One major advantage of enforcing the solution at the *IB-faces* is that it circumvents the limitations associated with insufficient space for inserting ghost cells near complex immersed boundaries. However, unlike ghost-cell methods, where the flux computation for fluid cells remains unchanged, face-based approaches require a dedicated treatment of fluxes at the *IB-faces*.

2.3 Current Use of Immersed Boundary Methods for Ice Accretion

In this section, we present a concise review of the different IBMs that have been used specifically for ice accretion simulations. Table 2.2 summarizes the characteristics of the IBMs used by all research centers identified in this review as having applied such methods to ice accretion simulations. Other icing-related applications, such as the assessment of aerodynamic performance degradation due to ice [156, 187] or the prediction of ice shedding trajectories [16], are not considered here, as they fall outside the scope of this review.

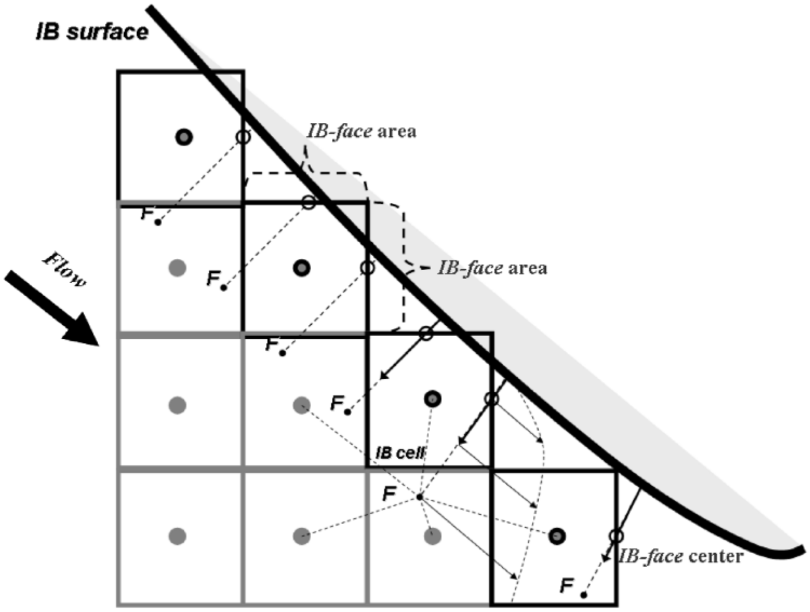


Figure 2.16 Illustration of the face-based discrete IBM. Image from [34].

Institution	**CIRA**	**ONERA**	**U. Strasbourg**	**Poly. Montreal**
Code	SIMBA	IGLOO2D	NSMB-ICE	CHAMPS
Airflow	RANS/RANS-LES	Euler	Laminar NS	Laminar NS
Airflow IBM	Discrete	Continuous	Continuous	Discrete
Droplets	Eulerian	Eulerian	Eulerian	Eulerian
Droplets IBM	Discrete	Continuous	Discrete	Continuous
Domain	2D/3D	2D	2D	2D

Table 2.2 Comparison of IBMs employed by research centers in ice accretion simulations.

2.3.1 CIRA: SIMBA

The Italian Aerospace Research Center (Centro Italiano Ricerche Aerospaziali – CIRA) is one of the earliest institutions to have applied IBMs to ice accretion simulations [38]. CIRA developed the in-house numerical tool system SIMBA (Simulation system based on an IMmersed Boundary Approach), to perform two- and three-dimensional aerodynamic-based simulations. SIMBA incorporates several specialized modules to perform ice accretion computations [37]:

- Mesh generation module:
  The volume meshes are generated from a coarse Cartesian grid around the wing. Then, a non-isotropic AMR procedure is applied to achieve the desired mesh resolution near the wall [35].

- Airflow module:
  A Finite Volume Method (FVM) is used to solve inviscid/viscous flows [33, 34]. For IB ice accretion simulations, RANS equations are typically solved applying the $\kappa - \omega$ TNT turbulence model, and IB conditions are imposed using proper turbulent wall models [34].
- Droplet trajectory module:
  The Eulerian droplet trajectory equations are solved to obtain the water impingement on exposed surfaces [38].
- Ice accretion module:
  The Messinger mass and energy balance is used to obtain the ice thickness [52].
- Geometry update module:
  The geometry update module performs a Lagrangian displacement of the surface [52].

The SIMBA code developed by CIRA represents one of the most advanced and comprehensive IBM-based frameworks for ice accretion simulations currently available. One of its major strengths lies in its ability to perform fully three-dimensional IB simulations, which is still uncommon among other research institutions. SIMBA is also the only code in this review that presented results using IBMs in both the first and second Ice Prediction Workshops [36, 37, 84, 86]. Another major advantage is the high level of integration within SIMBA: all stages of the simulation seem to be handled in a single environment specifically designed for IBMs. This avoids the need for external preprocessing or coupling tools, enabling a seamless and consistent workflow.

2.3.2 ONERA: IGLOO2D

As a result of the doctoral work of P. Lavoie [87], the IGLOO2D code was extended to incorporate an IB methodology for two-dimensional ice accretion simulations. In particular, two penalization IBMs were developed to perform the aerodynamic and droplet trajectory simulations. These approaches rely on two distinct mask functions. The first one, denoted by χ, is the standard mask function used to distinguish between solid ($\chi = 1$) and fluid ($\chi = 0$) cells. The second, χ_{d}, is employed exclusively by the droplet trajectory solver to differentiate between the shadow zone (outgoing characteristics, $\chi_{\mathrm{d}} = 1$) and the impingement zone (incoming characteristics, $\chi_{\mathrm{d}} = 0$) [91]. Figure 2.17 illustrates both mask functions for a cylinder.

According to [90], the employed modules used for two-dimensional ice accretion simulations

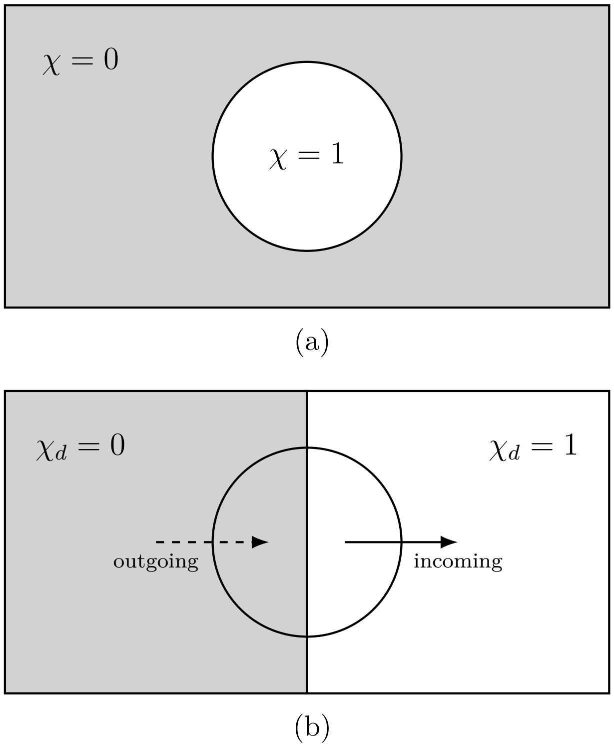


Figure 2.17 Mask function (a) and droplet mask function (b) for a cylinder. Image adapted from [91].

are the following:

- Mesh generation module:
  The proposed approach implemented in IGLOO2D employs the BF unstructured mesh of the clean airfoil for the entire multi-step simulation. This way, the IBMs are only applied in the ice zone, while the rest of the solid boundary is treated as BF, enhancing the accuracy. Specifically, unstructured meshes are generated using Gmsh [68].

- Airflow module:
  IGLOO2D computes the inviscid airflow using a weak coupling with an integral boundary layer model [176]. Regarding the IBM, the Euler equations are solved via a penalization approach [89]. This IB methodology distinguishes itself by replacing the standard slip boundary condition with the more sophisticated Curvature-Corrected Symmetry Technique (CCST) [46], which incorporate curvature effects in the pressure gradient and ensure conservation of entropy and total enthalpy in the wall-normal direction.
  The developed IBM yields lower error levels compared to other penalization approaches

for the Euler equations, specially on coarse meshes. However, its limitation to first-order accuracy represents a significant drawback.

- Droplet trajectory module:
  Similarly, a penalization methodology is implemented in IGLOO2D to solve the Eulerian droplet trajectory equations. Two methods were developed: the Volume Penalization (VP) and the Volume Penalization Solid Source Off (VP-SSO). While the VP method offers a more natural formulation, it requires solving the airflow field within the solid region. In contrast, the VP-SSO computes droplet trajectories independently of the air velocity inside the solid. Despite their differences, both methods converge to the BF solution. Due to the nature of the boundary conditions in the impingement zone, the penalization methods achieves second-order accuracy in space [91].

- Ice accretion module:
  The Messinger balance is used to compute the ice thickness at the surface [176].

- Geometry update module:
  The level-set method is used to update the geometry avoiding surface mesh overlaps, and a post-processing step is added to extract the surface mesh from the level-set field using an edge-based interpolation method [90].

The results obtained in two-dimensional simulations are satisfactory, including both rime and glaze ice cases, as well as studies on multi-element airfoils, demonstrating the robustness of the method [90]. Among the advantages, the proposed IBMs allow straightforward extension to three-dimensional solvers and avoid complex pre-processing steps. However, the method presents some limitations, notably the first-order accuracy of the aerodynamic solver and the necessity of extracting and generating a surface mesh from the level-set, which can be challenging in three-dimensional simulations.

2.3.3 University of Strasbourg: NSMB-ICE

The University of Strasbourg has also been one of the first research centers to include IBMs in the two-dimensional Navier-Stokes Multi-Block ice accretion solver, NSMB-ICE [131]. The development of these methods has been primarily carried out as part of the doctoral thesis of Al-Kebsi [6]. However, in this review, we focus on [7] to analyze their IB ice accretion suite.

- Mesh generation module:
  Two types of meshes are typically used in NSMB-ICE: typical Cartesian multi-block grids and structured Chimera grids.

- Airflow module:
  A penalization method was developed to solve the laminar Navier–Stokes equations [7], based on the approach of Abgrall et al. [4]. However, later work [6] reported difficulties in accurately modeling turbulent flows with the implemented penalization method. As a result, a discrete forcing method combined with a wall model [15, 76] was also developed to simulate turbulent flows more effectively.

- Droplet trajectory module:
  The droplet transport module developed in NSMB uses an Eulerian approach. Concerning the IBM implemented, a forcing discrete method is used to define the value of the fluid cells around the solid.

- Ice accretion module:
  The SWIM model is used to obtain the ice thickness at the discretized surface [17, 26].

- Geometry update module:
  The evolution of the ice-air interface is computed using a level-set approach [131].

The simulations presented in [6] include two-dimensional rime and glaze ice accretion cases, showcasing the code's ability to model different icing conditions. NSMB-ICE was also among the first to combine level-set methods with immersed boundary techniques for ice accretion simulations, establishing a foundation that has influenced the development of other codes in this field.

2.3.4 Polytechnique Montréal: CHAMPS

The ice accretion software developed at Polytechnique Montréal, CHApel MultiPhysics Simulations (CHAMPS) compute different multi-physical calculations [130]. Concerning ice accretion simulations, Blanchet et al. [24] introduced the first IBM-based simulations of CHAMPS at the 2^{nd} IPW. This approach is notable for combining IBMs with a stochastic surface evolution model [27, 129] in 2D multi-step ice accretion simulations. The main features of each module are listed below:

- Mesh generation module:
  Cartesian multi-block grids are employed.

- Airflow module:
  The RANS equations are solved using a FVM. The IBM employed, consist in a discrete forcing method, following the ghost-cell methodology presented in [6].

- Droplet trajectory module:
  The penalization method presented by Lavoie et al. [91] for 2D simulations and extended to 3D cases in [61] is used to solve the Eulerian droplet trajectory equations.
- Ice accretion module:
  The Messinger balance is used to compute the ice thickness at the surface [176].
- Geometry update module:
  The stochastic approach is used to track the evolution of the geometry [27, 129]. However, in order to be able to run multi-step simulations, surface meshes are needed at the end of each iteration of the ice accretion loop. To do that, Blanchet et al [24] use the 3D alpha wrapping library from CGAL [9].

CHAMPS, despite being at an early stage in the integration of IBM, demonstrates a promising and innovative approach by effectively coupling IBMs with a stochastic modeling framework. While current limitations include its restriction to laminar flows and the lack of mesh refinement techniques [24], these challenges also highlight clear pathways for future development, such as extending the model to turbulent regimes and incorporating AMR.

2.4 Intermediate Conclusions

This section reviewed the state of the art in three distinct areas: ice accretion simulations, IBMs, and the integration of IBMs into icing frameworks.

Ice accretion simulations greatly benefit from a modular segmentation of the entire process. This not only simplifies the overall simulation workflow but also allows existing codes to integrate different models for each individual module. To improve the accuracy of ice shape predictions, multi-step simulations are required in order to account for the time evolution of the geometry. However, despite the development of improved numerical methods in recent decades, two major challenges often hinder the automation and robustness of the overall simulation: geometry update and the subsequent volume mesh generation. These issues motivate the exploration of alternative strategies, such as the use of IBMs.

Immersed Boundary Methods are a suitable option when dealing with complex geometries or moving boundaries. While several approaches were reviewed in this chapter, two representative methods stood out: the penalization method and the ghost-cell method. The former falls under the continuous forcing category and had been applied to a wide range of aerodynamic models. Its implementation in existing codes is generally straightforward, with preprocessing typically limited to distinguishing between fluid and solid cells. However, in applications based on the FVM, the spatial accuracy is often limited to first-order convergence due to the sharp representation of the fluid–solid interface. In contrast, the ghost-cell method is a discrete forcing IBM. Several applications can be found in the literature involving different governing equations and aerodynamic models. Unlike the penalization approach, preprocessing for ghost-cell methods tends to be more challenging, particularly when dealing with complex geometries. Nevertheless, it remains a promising alternative, as maintaining the convergence order of the solver is more feasible, being primarily tied to the interpolation scheme used for the image point.

In the context of ice accretion, there had been growing interest in applying IBMs to avoid mesh regeneration at each step of a multi-step simulation. Several research centers, including CIRA, ONERA, the University of Strasbourg, and Polytechnique Montréal, had incorporated IBMs into their icing codes. A comparison of these frameworks revealed that each group adopted a different implementation strategy, primarily distinguished by the type of mesh used (e.g., Cartesian multi-blocks, Cartesian grids with anisotropic refinement, or BF meshes derived from the clean geometry) and by the IBM selected (i.e., continuous or discrete forcing methods). Among these, CIRA is the only research center currently performing full three-dimensional ice accretion simulations.

CHAPTER 3 THESIS STRUCTURE

The present doctoral work is written in the form of a thesis by compendium. It is composed of two peer-reviewed journal articles, which are presented in Chapters 4 and 6. Chapter 5 presents additional unpublished work that complements the main contributions of the thesis. It is written in the format of a journal article in order to maintain stylistic consistency throughout the document.

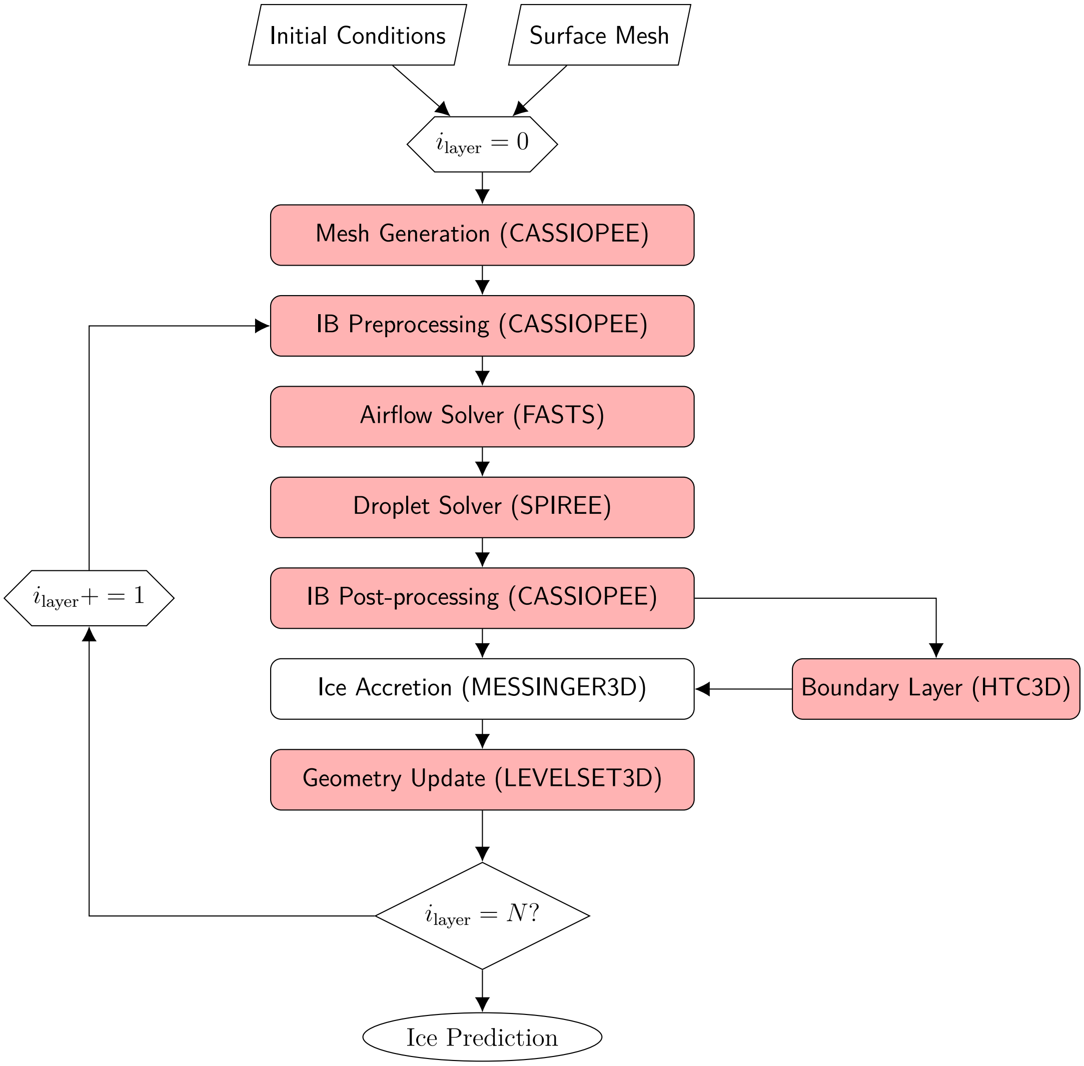
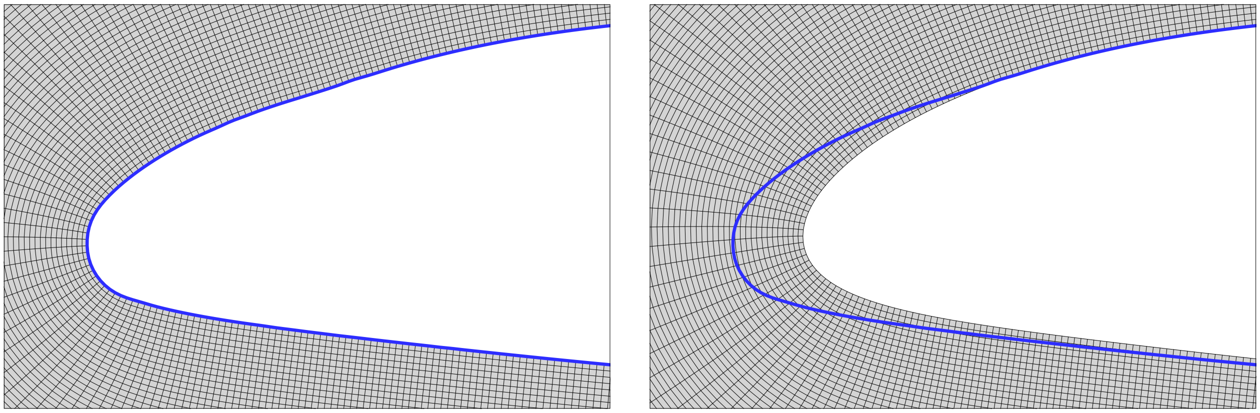


Figure 3.1 Sequential call to modules in multi-step icing simulations using IBMs in IGLOO3D. Modified or new modules are highlighted in red.

As a reminder, the general objective of this research project is to investigate the application of Immersed Boundary Methods (IBMs) in three-dimensional multi-step ice accretion simulations, enhancing the automation of the entire simulation by avoiding the generation of a new volume mesh at each ice layer. For this purpose, the three-dimensional ice accretion suite of ONERA, IGLOO3D, is chosen as the development platform. Figure 3.1 illustrates the sequential call of modules in a multi-step ice accretion simulation carried out in this thesis, indicating which code is used for each module. In this framework, the two volume solvers rely on an IB approach, which requires the addition of dedicated pre- and post-processing IB modules. In order to adapt IGLOO3D to IBMs, most of its modules have been modified; these are highlighted in red in Fig. 3.1. The only module that has remained unchanged is the ice accretion module MESSINGER3D.

For mesh generation, this thesis continues with the strategy adopted in the IB version of IGLOO2D [90]. Specifically, it starts from a BF mesh of the initial geometry and applies the IBMs exclusively in the regions where ice has grown. This approach is referred to as local IBM or hybrid IBM in the literature (see Sec. 2.2.1). However, unlike IGLOO2D, a requirement of the simulation chain presented here is that the meshes must be structured, as required by the airflow solver. This poses additional challenges in mesh generation, particularly around the wing tip of three-dimensional wings. These difficulties become even more pronounced when comparing the IB with BF meshes, since generating structured meshes around complex iced shapes is precisely what this thesis aims to avoid. In the proposed approach, O-type meshes are employed. The procedure is briefly explained below. The structured mesh on the wall serves as the first layer of the volume mesh. This layer is then extruded in the normal direction to obtain a new layer. When complex geometries are involved, the normal field can be smoothed. The extrusion process is repeated layer by layer until the desired distance to



(a) Body-fitted

(b) Immersed boundary

Figure 3.2 Examples of volume meshes used in multi-step ice accretion simulations with either BF or the IB approach proposed in this thesis. The ice shape is shown by a thick blue line.

the farfield boundary is reached. In three-dimensional meshes, the initial structured surface mesh is divided into two blocks: one for the wingtip and the other for the rest of the wing. This alleviates the previously discussed challenge associated with meshing three-dimensional wings.

For the results presented in this thesis, all meshes have been generated using Cassiopee (CFD Advanced Set of Services In an Open Python Environment) [19]. Cassiopee is an open-source library developed at ONERA, primarily for performing pre- and post-processing tasks in CFD simulations. As illustrated in Fig. 3.1, Cassiopee is also used for the pre- and post-processing of the IBMs.

A complementary study related to surface mesh smoothing techniques in multi-step ice accretion simulations has been carried out as part of this thesis. This work was presented at the 2024 AIAA Aviation Forum [60] and is detailed in Appendix A of this document.

The remaining modifications and developments applied to the other modules represent the fundamental contributions of this doctoral work. These core components are briefly introduced below, providing a general overview that links them together.

3.1 First Article

The first article of this thesis, presented in Chapter 4, aims to fulfill Objective **O.1**: the development of an IBM for solving the three-dimensional Euler equations.

The preceding thesis by P. Lavoie [87] investigated the application of IBMs to ice accretion simulations and highlighted some inherent limitations of the penalization method applied to the Euler equations, in particular its restriction to first-order accuracy. Additionally, ONERA has been investigating the use of other IBMs in ice accretion simulations for several years, as part of its private research project TRICEPS.

In fact, ONERA has been developing a suite of high-performance computing CFD solvers focused on IBMs, known as Flexible Aerodynamic Solver Technology (FAST) [3]. This numerical tool includes two main solvers: FastS, a Navier–Stokes solver, and FastLBM, a Lattice Boltzmann solver. Both solvers employ structured grids to maximize computational efficiency. FastS, in particular, is a FVM solver capable of simulating aerodynamic flows at various levels of fidelity, supporting models ranging from inviscid flow to large eddy simulation (LES) [139]. An IBM approach based on a discrete forcing has been developed in Cassiopee/FAST framework.

As part of the TRICEPS project, comparative studies were carried out analyzing results obtained with the ghost-cell method of FastS and the penalization method developed in

IGLOO2D. These studies included a range of test cases, including subsonic flows around three geometries: a cylinder, a NACA 0012 airfoil, and a GLC305 iced airfoil, the latter representing the most challenging case. The investigations examined mesh convergence, the influence of the employed mesh type (either through a local IBM as implemented in IGLOO2D or by using adaptive Cartesian mesh refinement in overset grids), and the impact of placing ghost cells either within the fluid or solid regions. One of the main conclusions drawn was the difficulty of obtaining steady solutions without spurious recirculation zones when simulating complex geometries such as the iced GLC305 airfoil. Specifically, both FastS and the penalization method in IGLOO2D encountered challenges in reaching steady-state solutions, and neither method demonstrated consistent convergence upon mesh refinement for this case.

Addressing these challenges was therefore identified as one of the main objectives of this thesis. Following a thorough review of the literature, the decision was made to develop within Cassiopee/FAST framework an improved GCIBM for three-dimensional inviscid simulations, tailored to handle complex geometries effectively.

As a first step, more accurate boundary conditions were implemented, following the work of Dadone and Grossman [46] and the improvements demonstrated by P. Lavoie [89]. Subsequently, the effect of the ghost-cell positioning was studied in detail. Three configurations were investigated: *GC-in-solid*, where all ghost cells are placed in the solid; *GC-in-fluid*, where they are placed in the fluid; and *GC-surrounding*, where the first row of ghost cells is in the fluid and the second in the solid. This intermediate work formed the core of the publication presented at the 2023 International Conference on Icing of Aircraft, Engines, and Structures (SAE) [61] and serves as the starting point of the first article.

The first article presents a GCIBM well-suited for complex geometries, such as those typically encountered in ice accretion simulations. The proposed method, named the Adaptive Ghost-Cell-Surrounding (AGCS) method, introduces an innovative strategy for ghost-cell placement based on the local curvature of the geometry. The method is compared against both body-fitted simulations and more standard ghost-cell approaches for inviscid flows, such as *GC-in-solid*. Various cases are analyzed, including subsonic flows around a 2D cylinder and a 3D sphere, demonstrating that the method retains the second-order accuracy of the numerical scheme. Additionally, the accuracy of the results is evaluated on a manufactured 2D iced airfoil and a 30º swept NACA 0012 iced wing obtained from an ice accretion simulation.

3.2 Simplified Integral Boundary Layer Solver

Using an inviscid model in ice accretion simulations requires coupling with a boundary layer solver to compute two key inputs for the ice accretion module: the convective heat transfer coefficient and the recovery temperature. An accurate resolution of these quantities requires solving the full boundary layer equations, which is a complex task that could justify a dedicated research project. To address this challenge in a practical and efficient manner, this work proposes the development of a three-dimensional simplified integral boundary layer solver. This development specifically addresses Objective **O.3** of the thesis.

The starting point for this development is the work presented in [141], where, as part of a broader study, a simplified method for solving the laminar boundary layer was introduced. Building upon that foundation, the objective of this chapter is to extend the existing model to also account for turbulent boundary layers. The proposed method is based on the simplified integral approach developed in IGLOO2D and is here generalized to three dimensions.

Although Chapter 5 is not based on a journal article, it follows the same structure (introduction, methodology, results, and conclusion) to ensure consistency and clarity throughout the thesis. The proposed method provides a trade-off between physical accuracy and robustness, making it suitable for estimating convective heat transfer in the context of this work.

The method is subsequently integrated into the ice accretion framework presented in the second journal article, where it enables multi-step simulations and assesses its predictive capabilities under realistic icing conditions.

3.3 Second Article

The second article aims to perform three-dimensional ice accretion simulations using IBMs. To achieve this, the remaining objectives outlined at the beginning of the thesis must be addressed.

First, to address Objective **O.2**, the validation of the penalization method used for droplet trajectory simulations, introduced by P. Lavoie [91], is carried out in three-dimensional simulations. This validation was presented in the SAE publication [61], demonstrating that the method retains the second-order spatial convergence of the numerical scheme employed.

Following the validation of the IBMs, their integration into IGLOO3D is carried out to fulfill Objective **O.4**. Although most modules rely on CGNS-formatted input and output files, format converters are necessary to meet the specific formatting requirements of each individual module. Additionally, minor modifications were made to the geometry update module, LEVELSET3D. This module, originally developed to perform BF simulations, is based on a hybrid Lagrangian-Eulerian approach based on the level-set field. Consequently, adaptations were necessary to enable its use in simulations involving IBMs.

The publication of the second journal article, presented in Chapter 6, builds upon the earlier work introduced in the 2024 AIAA Aviation conference paper [60], where initial ice accretion simulations using IBMs were performed. The journal article offers a more comprehensive and detailed study of the methodology and its applications, thus addressing the final specific objective of the thesis, Objective **O.5**. Since the IB and BF approaches are expected to produce consistent results, both methods are compared for each case studied through two-step simulations. Specifically, three cases from the 1^{st} IPW are analyzed in detail: the two-dimensional cases 241 and 242, corresponding to rime and glaze icing conditions, respectively, and case 361, which involves the study of a 30º sweep angle NACA 0012 wing under rime icing conditions.

CHAPTER 4 ARTICLE 1: SECOND-ORDER GHOST-CELL IMMERSED BOUNDARY METHODOLOGY FOR INVISCID 3D FLOWS IN COMPLEX GEOMETRIES

P. Elices Paz, E. Radenac, S. Péron, G. Blanchard, É. Laurendeau, and P. Villedieu, "Second-Order Ghost-Cell Immersed Boundary Methodology for Inviscid 3D Flows in Complex Geometries," (submitted to *Computers & Fluids* on 14 February 2025)

4.1 Abstract

Inviscid flow simulations are crucial in various aerospace and engineering applications, particularly when computational cost is a primary concern. However, even for such flows, the generation of meshes may become challenging when complex geometries are involved, especially in 3D cases. As a result, there is special interest in methods that can bypass this issue. In this context, the Adaptive Ghost-Cell-Surrounding (AGCS) method is introduced for simulating compressible, inviscid flows around complex, rigid geometries using a second-order finite volume solver. The method is grounded in the Immersed Boundary Method (IBM) framework, specifically using the ghost-cell method to impose boundary conditions on non-conforming meshes. AGCS employs a local curvature-based criterion to determine ghost cell placement, significantly enhancing accuracy, notably on coarse meshes. Verification tests demonstrate that the method achieves second-order spatial accuracy in numerical simulations of 2D cylindrical and 3D spherical geometries. Additionally, the AGCS method has been applied to 2D and 3D wings with complex ice accretion shapes, further demonstrating its robustness and precision. The results highlight the advantages of the proposed approach, including improved solution convergence and reduced entropy generation compared to conventional ghost-cell methods.

4.2 Introduction

Several benchmarking activities in the field of aerodynamics as applied to aircraft analysis and design, such as the American Institute of Aeronautics and Astronautics (AIAA) series of workshops (Drag, High-Lift, Aeroelastic, etc.) have focused on improving upon the industrial state of the art Reynolds-Averaged Navier–Stokes (RANS) models. These activities either try to achieve mesh convergence on high-lift take-off/landing phases, increase order of accuracy, develop effective viscous mesh adaptation or further increase the fidelity of the model towards Large Eddy Simulations. However, recent advances in artificial intelligence call for mutlifidelity approaches, as the large amount of datasets needed for training the algorithms becomes very expensive when examining highly non-linear phenomena, as found on aircraft geometries. For instance, Renganathan et al. [147] employ an inviscid solver for training machine learning models, while Catalini et al. [39] couple it with a viscous boundary layer to generate datasets. Inviscid flow models are also found in aerodynamic optimization of aircraft configurations [103], and ice accretion calculations [90, 176], among other areas. A common characteristic of these applications is the need to conduct numerous simulations with variations in geometry. Geometry modeling and the associated mesh generation for field methods used in aerospace poses significant challenges. First, body-fitted meshes require closed CAD surfaces and, when morphing [172], must retain non-negative cell volumes. The 2021 and 2023 AIAA Ice Prediction Workshops, in which RANS remains the baseline approach [84, 86], have shown that mesh deformation continues to be a critical limitation to ice shape simulations on simple cases while industrial cases pose even higher constraints. For inviscid flow simulations, the boundary layer is not resolved, eliminating the need for highly refined meshes in the wall-normal direction. Although this alleviates some of the key challenges associated with mesh deformation, the task remains difficult to automate, specially when 3D complex surfaces are involved.

This paper presents a new Immersed Boundary Method (IBM) for the solution of the Euler equations to address issues related to mesh generation. The inviscid model can eventually be coupled to boundary-layer models, which is left out-of-scope in the present discussion without losing generality. The concept of IBMs was first introduced by Charles Peskin in the 1970s [133] to simulate blood flow around heart valves. Since then, numerous IBMs have been developed, which can be categorized into two distinct groups [109]. The *continuous forcing approach* adds source terms to the continuous equations prior to discretization, resulting in an easy method to implement. However, the application of this method usually introduces challenges, primarily because of the forcing terms added, leading to a negative impact on both the numerical stability and the accuracy of the solution near the body interface. Different methods have been developed for both inviscid [89] and viscous flows [31]. Although this approach is typically limited to first order accuracy, second order adaptations are also found in the literature [157]. In contrast, the *discrete forcing approach* selects specific cells, commonly referred to as ghost cells [49, 139], or immersed faces [33, 34], where the solution is reconstructed to replicate the presence of the interface. These methods mitigate issues related to accuracy and stability; however, their implementation involves a complex preprocessing step.

This paper introduces an efficient and robust ghost-cell method, referred to as Adaptive

Ghost-Cell-Surrounding (AGCS), for simulating compressible, inviscid flows around complex, rigid geometries. The proposed method builds on the work of Dadone and Grossman [49, 51], Nakahashi [118], and Péron et al. [139]. A detailed review of the ghost-cell methodology is provided in Sec. 4.5. The method uses a second-order accurate finite volume solver designed for structured grids. Unlike other ghost-cell methods, the proposed approach places the ghost cells according to a local curvature-based criterion. In regions where the local curvature radius is significantly smaller than the cell size, ghost cells are positioned within the fluid domain, acting as a pseudo wall function and significantly improving solution accuracy, especially on coarse meshes.

This method has primarily been designed for icing applications. In fact, simplified versions of it have previously been presented by the authors for calculating droplet trajectories [61] and conducting multi-step ice accretion simulations [60]. However, in this paper, it is presented in a more general form, as its application could also be useful in other fields dealing with complex geometries.

This paper is organized in the following manner. Section 4.3 provides a description of the governing equations. Section 4.4 concerns the numerical implementation of the Euler equations. Additionally, a review of the commonly used boundary conditions is provided. Section 4.5 discusses the simplest ghost-cell method, highlighting its main issues and the common approaches used to address them. Section 4.6 describes the new AGCS method. Section 4.7 presents four numerical simulations. The first two cases focus on verification tests, where the method is applied on Cartesian meshes. Specifically, the simulations involve the inviscid flow around a 2D cylinder and a 3D sphere. Following this, the method is validated on 2D and 3D ice shapes by comparing its results with those from body-fitted simulations.

4.3 Governing Equations

The dynamics of a compressible fluid for which the effects of body forces, viscous stresses and heat flux are neglected is governed by the time-dependent Euler equations. They are a system of non-linear hyperbolic conservation laws. The conservative variables vector, $\vec{W}$, can be derived from the physical variables: density (ρ), pressure (p) and velocity ($\vec{v} = (u, v, w)$):

$$\vec{W} = [\rho, \rho u, \rho v, \rho w, \rho E]^T \tag{4.1}$$

Here E is the total energy per unit mass:

$$E = e + \frac{||\vec{v}||^2}{2} = e + \frac{u^2 + v^2 + w^2}{2}, \tag{4.2}$$

and e is the internal energy.

Consider a 3D control volume Ω as the computational domain and $\partial\Omega$ its boundary, the governing equations, in the integral form, can be expressed as:

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{W}\mathrm{d}\Omega + \oint_{\partial\Omega}\vec{F}_{inv}\mathrm{d}S = 0 \tag{4.3}$$

where $\vec{F}_{inv}$ is the vector of the inviscid fluxes

$$\vec{F}_{inv} = \begin{bmatrix} \rho(\vec{u}\cdot\vec{n}) \\ \rho u(\vec{u}\cdot\vec{n}) + n_x p \\ \rho v(\vec{u}\cdot\vec{n}) + n_y p \\ \rho w(\vec{u}\cdot\vec{n}) + n_z p \\ (\rho E + p)(\vec{u}\cdot\vec{n}) \end{bmatrix}, \tag{4.4}$$

and $\vec{n} = (n_x, n_y, n_z)$ is the normal vector to the surface $\partial\Omega$, pointing outward from the solid.

To complete the system, two additional equations are required, which will establish thermodynamic relationships between the state variables. For a calorically perfect gas, the equations are:

$$p = \rho r T, \tag{4.5}$$

$$e = \frac{r}{\gamma - 1}T, \tag{4.6}$$

where r denotes the specific gas constant, T the temperature and $\gamma = c_P/c_V$ is the ratio of specific heat coefficients.

For inviscid flows, the only physical condition is that the fluid slips along the surface, i.e., the velocity vector must remain tangent to the surface, with no flow passing through the surface, that is,

$$(\vec{v}\cdot\vec{n})_w = 0, \tag{4.7}$$

where the subscript w refers to the wall.

4.4 Numerical Method

The ONERA HPC FastS solver [3] is used to solve the compressible Euler equations using a cell-centered finite-volume method on structured meshes. The temporal integration is solved implicitly using a second-order backward differentiation formula. The Roe scheme [149] is used to compute the inviscid fluxes and a MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) [179] extrapolation with a minmod slope limiter is employed to achieve second-order accuracy in space. The left and right values of the conservative variables at the interface $i+1/2$ are obtained according to the following expressions:

$$\begin{cases} W_L = W_i + \frac{1}{2}\Phi\left(r_L\right)\left(W_{i+1} - W_i\right) \\ W_R = W_{i+1} - \frac{1}{2}\Phi\left(r_R\right)\left(W_{i+2} - W_{i+1}\right) \end{cases} \tag{4.8}$$

where Φ is the slope limiter and $r_L = \frac{W_i - W_{i-1}}{W_{i+1} - W_i}$ and $r_R = \frac{W_{i+1} - W_i}{W_{i+2} - W_{i+1}}$ are the slopes at each side of the interface. This results in a five-cell computational stencil in each direction.

4.4.1 Wall Boundary Conditions

Wall boundary conditions play a fundamental role in simulating inviscid flows. From a numerical perspective, when imposing the slip boundary conditions (Eq. 4.7), additional information is required. In body-fitted simulations, where the cell interfaces align with the wall, the non-penetration condition ensures that the inviscid fluxes (Eq. 4.4) depend exclusively on the pressure at the wall, i.e.,

$$\vec{F}_{inv} = \begin{bmatrix} 0 \\ n_x p_w \\ n_y p_w \\ n_z p_w \\ 0 \end{bmatrix}. \tag{4.9}$$

Various methods can be used to determine the wall pressure using a cell-centered approach. For instance, Dadone and Grossman [46] developed two methods: the Symmetry Technique (ST), only valid for planar walls; and the Curvature-Corrected Symmetry Technique (CCST), which takes into account the curvature of the solid surface. The latter has proven to be an excellent option for simulating inviscid flows around high-curvature geometries, significantly enhancing solution accuracy.

Both boundary conditions involve introducing two additional rows of ghost cells within the solid (denoted as -1 and -2 in Fig. 4.1), also known as dummy cells, to fulfill the computational stencil of the fluid cells near the boundary. These are constructed by reflecting the internal cells 1 and 2 symmetrically across the body surface.

For the CCST [46], the variable states $\vec{W}$ at the ghost cells are set to make the wall act as a streamline, ensuring the non-penetration condition (Eq. 4.7), the wall normal momentum relation [148],

$$\left(\frac{\partial p}{\partial n}\right)_w = -\rho_w \left|\left|\vec{v}_w\right|\right|^2 \kappa_w, \tag{4.10}$$

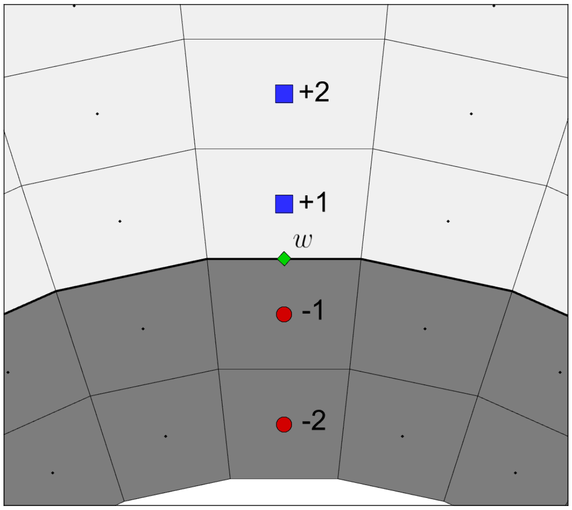


Figure 4.1 Ghost cells used in a body fitted simulation to impose the symmetry technique wall boundary conditions.

and the conservation of the entropy ($(\vec{n} \cdot \nabla s)_w = 0$) and total enthalpy ($(\vec{n} \cdot \nabla H)_w = 0$) along the wall normal. In Eq. 4.10, κ_w refers to the signed curvature of the surface at the wall point w. It is considered positive when the center of curvature lies within the solid and negative when it is located within the fluid.

Subsequently, modifications were introduced to simplify this original formulation [49, 51]. First, the wall density and velocity values of Eq. 4.10 were replaced by the fluid cell values, i.e., $\rho_w = \rho_i$ and $||\vec{v}_w||^2 = (v_i^\tau)^2$. In addition, the non penetration condition was also simplified, and antisymmetric values were used. This leads to the following set of boundary conditions:

$$v_{-i}^n = -v_{+i}^n \tag{4.11}$$

$$p_{-i} = p_{+i} - \rho_{+i}(v_{+i}^\tau)^2 \kappa_w \Delta n_i \tag{4.12}$$

$$\rho_{-i} = \rho_{+i} \left(\frac{p_{-i}}{p_{+i}} \right)^{1/\gamma} \tag{4.13}$$

$$(v_{-i}^\tau)^2 = (v_{+i}^\tau)^2 + \frac{2\gamma}{\gamma - 1} \left(\frac{p_{+i}}{\rho_{+i}} - \frac{p_{-i}}{\rho_{-i}} \right) \tag{4.14}$$

where $v^n = \vec{v} \cdot \vec{n}$ and $v^\tau = \vec{v} \cdot \vec{\tau}$ are the normal and the tangential component of the velocity to the body, Δn_i is the distance between points $+i$ and $-i$, and the subscript i indicates the cell index relative to the wall (as illustrated in Fig. 4.1). Note that the tangential unit vector

$\vec{\tau}$ is defined as

$$\vec{\tau} = \frac{\vec{v} - (\vec{v} \cdot \vec{n})\vec{n}}{||\vec{v} - (\vec{v} \cdot \vec{n})\vec{n}||}. \tag{4.15}$$

When the curvature is neglected, the ST boundary conditions are recovered:

$$p_{-i} = p_{+i} \tag{4.16}$$

$$\rho_{-i} = \rho_{+i} \tag{4.17}$$

$$v^n_{-i} = -v^n_{+i} \tag{4.18}$$

$$v^\tau_{-i} = v^\tau_{+i} \tag{4.19}$$

As noted [89], the error committed on the pressure extrapolation, ϵ_p, when using the ST instead of the CCST can be expressed as:

$$\epsilon_p = \rho_w \, ||\vec{v}_w||^2 \, \kappa_w \Delta x \tag{4.20}$$

where Δx is the characteristic wall cell size. It becomes apparent that the CCST method is particularly useful in highly curved accelerated regions.

Employing CCST boundary conditions therefore offers significant advantages for the analysis of inviscid flows around complex geometries. This approach leads to reduced numerical entropy and promotes faster grid convergence in both body-fitted [46] and immersed boundary methods for 2D [49] and 3D [51] simulations. Furthermore, its suitability in icing applications has been confirmed in previous research [60, 90].

4.4.2 Numerical Assessment of Boundary Curvature

In two dimensions, the definition of the curvature is unambiguous and can be analytically determined when the mathematical expression of the curve is known. For a curve $y = f(x)$, the curvature κ is obtained from the following expression:

$$\kappa = \frac{|f''|}{(1 + f'^2)^{3/2}}. \tag{4.21}$$

where f' and f'' are the first and second derivatives of $f(x)$ with respect to x respectively. Note that this value depends exclusively on geometrical aspects. In three dimensions, the needed value does not correspond to the curvature of the three-dimensional surface, but to the curvature of the intersection of the *osculating plane*, formed from the streamline direction and the surface normal, with the body interface [51, 194]. In this way, this task can be seen as a bi-dimensional problem. However, the curvature becomes dependent on the direction of the flow velocity vector around the wall.

For discrete curves, the literature presents various approaches to define the local curvature. If the curve is connected by straight segments, the first and second derivatives of $f(x)$ can be approximated by central differences [95, 194]. The curvature can also be calculated using the radius of the circumcircle formed by three successive points of the discrete curve [185]. Alternatively, using a signed distance field ϕ allows a simple evaluation of the normal $\vec{n}_\phi$ and the curvature [31] fields:

$$\vec{n}_\phi = \vec{\nabla}\phi, \tag{4.22}$$

$$\kappa = \vec{\tau} \cdot (\vec{\tau} \cdot \vec{\nabla})\vec{n}_\phi, \tag{4.23}$$

where the signed distance field is defined negative in the solid ($\phi < 0$) and positive in the fluid ($\phi > 0$); and the normal field to the wall, $\vec{n}_\phi$, points towards the fluid zone. For two-dimensional geometries, the flow dependency is lost, and a simplified expression is obtained [89],

$$\kappa = \vec{\nabla} \cdot \vec{n}_\phi. \tag{4.24}$$

This results in a convenient approach when dealing with an immersed boundary, as it is usually represented by the level-set $\phi = 0$.

4.5 The Ghost-Cell Immersed Boundary Method

The Ghost-Cell Immersed Boundary Method expands upon the concept of ghost cells presented above, by enabling the definition of boundary conditions in meshes that do not conform to the solid geometry. In this case, the grid cells are misaligned with respect to the normal of the solid interface, as illustrated in. Fig. 4.2. The general process can be divided into four steps:

1. Identification of the *ghost cells* (point GC in Fig. 4.2). For inviscid flows, ghost cells are commonly located in the solid cells around the geometry [49, 51, 118]. In the remainder of this paper, this approach is referred to as the *GC-in-solid* method. The number of ghost cells needed in each direction depends on the size of the solver stencil.

2. Projection of the ghost-cell centers on the solid interface to determine the *wall point* (point w in Fig. 4.2). This can typically be made by an orthogonal projection or following a normal direction $\vec{n}_\phi$.

3. Identification of the *image points* as the symmetric point of the ghost-cell center to

the wall (point IP in Fig. 4.2). They will be used to set the flow state values $\vec{W}$ of the corresponding ghost cell. Since the image points do not necessarily coincide with the grid points, their values must be obtained from the neighboring points (i.e., D_1, D_2, D_3, D_4 as shown in Fig. 4.2). The interpolation algorithm and its stencil may vary across the different existing ghost-cell methods.

4. The values at the ghost cells are determined using the data from image points when applying the boundary conditions. In this context, the CCST is usually employed [49, 51, 95, 185].

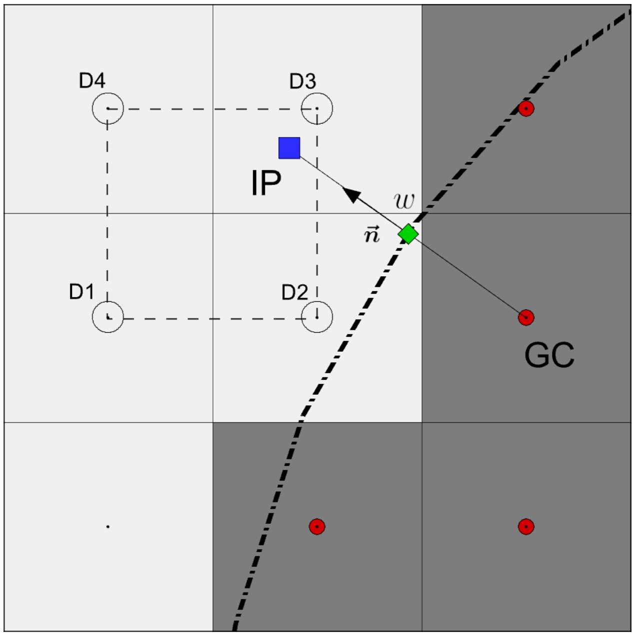


Figure 4.2 2D sketch illustrating the Ghost-Cell Immersed Boundary Method. Immersed boundary is represented in a dashed dot thick line and solid cells in dark gray.

The simulation can then be conducted without significant modifications to the solver, as no changes to the equations are necessary. The only required adjustment occurs just before imposing the boundary conditions (step 4) where the interpolation is performed. All other aspects remain unchanged.

Nevertheless, the literature highlights two major difficulties with this immersed boundary methodology [33, 49, 118, 139]. The first, referred to as the *multi-value problem*, presents the main challenge of the method, while the second involves the placement of image points in complex geometries.

4.5.1 Multi-Value Cells

In thin surfaces, such as trailing edges or elongated iced horns, there is often insufficient space to insert the desired number of ghost cells within the solid. The issue becomes particularly critical when no ghost cells can be placed inside the solid, leading to simulations where the surface becomes non-existent for the computed cells.

For instance, consider the region surrounding a complex, high-curvature geometry, as shown in Fig 4.3. In this sketch, the ghost cell highlighted (point P) is surrounded by three computed cells, with centers designated as Q, R, and S. In this scenario, the ghost cell should have three values, depending on which interface is being evaluated. This is the so-called *multi-value problem.* In addition, given the non-compact stencil used in this article, cell P is not the only one with multiple values in the figure presented. When linear extrapolation is employed to reconstruct the state variables at the interface, additional cells are utilized to interpolate the values on both sides of the interface. For example, at the interface between Q and P, cell S is used to obtain the value on the southern side of the interface. This implies that cells S and Q must have a double value, as they can be computed cells or ghost cells depending on the interface of the cell P which is being computed.

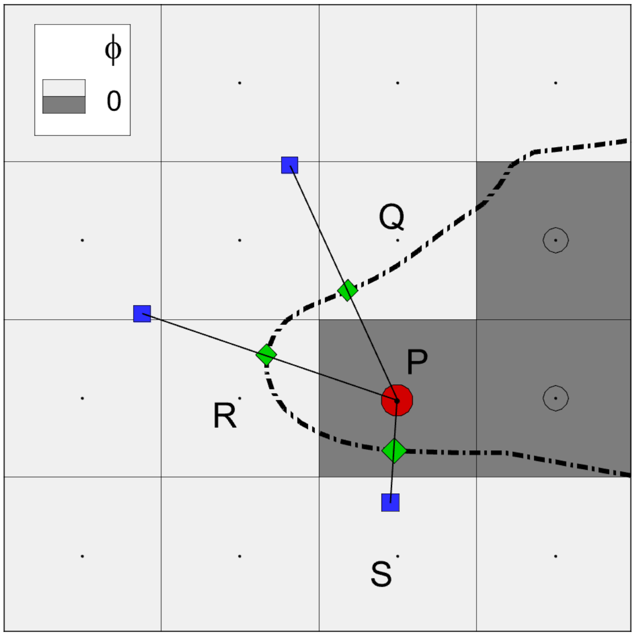


Figure 4.3 Illustration of multi-value cells. The image (blue square) and wall (green diamond) points are shown for a specific ghost cell (red circle). Solid cells ($\phi < 0$) are displayed in dark gray.

The literature presents various approaches to address this issue. For instance, Dadone and

Grossman [49] suggest storing the multiple values of the affected cells and applying each value according to the specific interface being solved. Although this method effectively addresses the problem, it requires special treatment in structured solvers, rendering its implementation less straightforward.

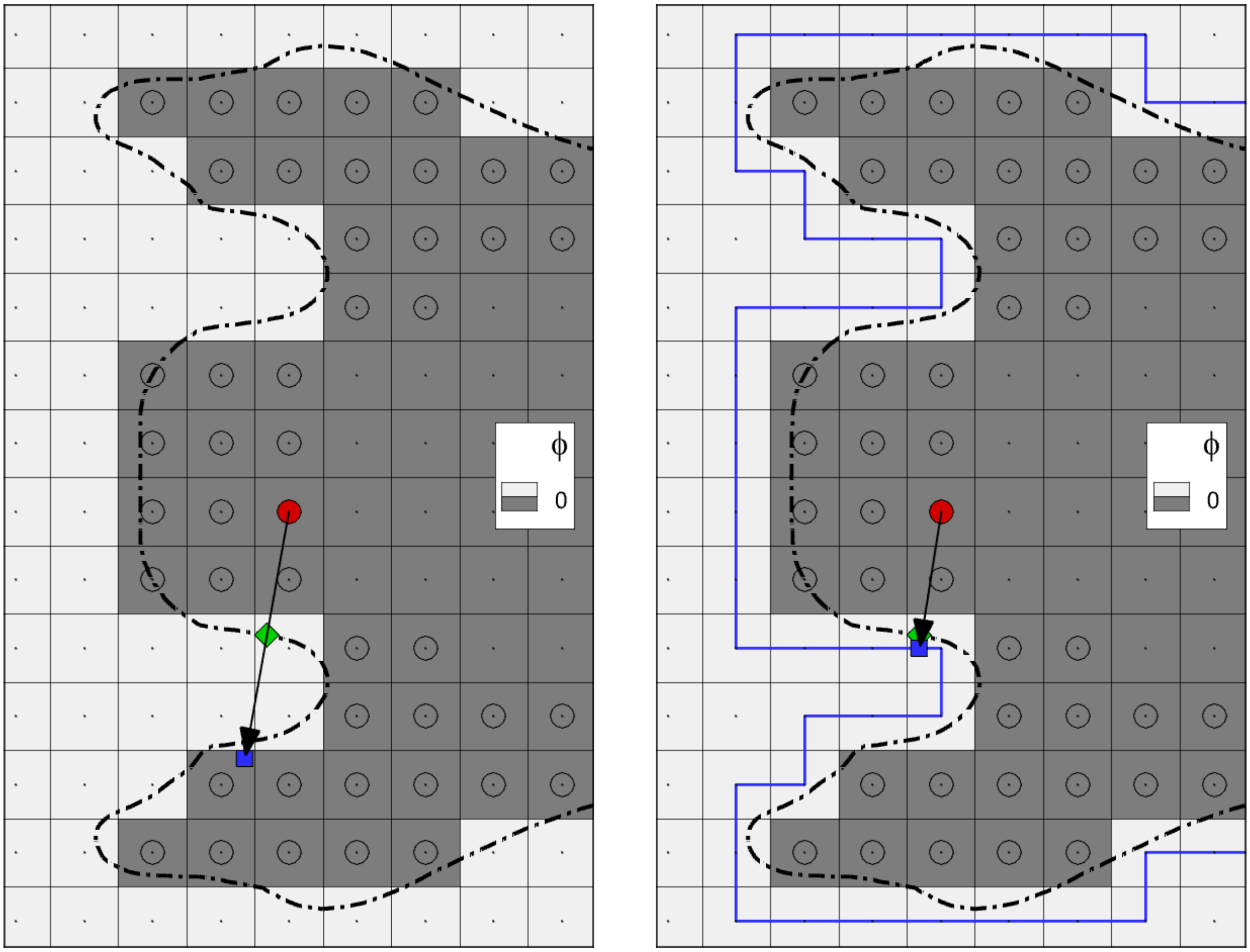
Nakahashi [118] presents a simpler alternative approach to the issue. First, the MUSCL extrapolation at interfaces is not applied when ghost cells are involved, significantly reducing the risk of creating multi-value cells but also reducing to a 1^{st}-order accurate flow at the wall. In addition, ghost cells can be placed within the fluid regions of the domain. In the example discussed above, cells P, Q and R would be classified as ghost cells, resulting in the absence of multi-value cells.

4.5.2 Issue with Image Point Location

Another challenge may arise when defining the image points as the symmetric points of the ghost cell centers. First, when the ghost cell is positioned in close proximity to the wall, its symmetric point will also remain near it, and consequently, the ghost cell may fall within the interpolation stencil of the image point, leading to an implicit rather than explicit definition of itself. In contrast, when the ghost cell is positioned farther from the wall, its symmetric point may be located at a significantly greater distance. This effect is especially relevant in inviscid flow simulations, which are the focus of the present study, as larger computational cells are used compared to viscous flow computation. Consequently, in concave regions, the symmetric point may fall within the solid domain (Fig. 4.4a). This issue is more commonly found in 3D simulations.

Nakahashi [118] proposes shifting the image point farther from the wall when the ghost cell is positioned close to it, typically by half the cell size, thereby avoiding the inclusion of the ghost cell within the interpolation stencil of the image point. However, no maximum distance is defined, potentially leading to the image point falling within the solid. Péron et al. [139] propose an alternative approach, which involves constructing a front surface where the image points must lie. This *front*, illustrated in blue in Fig. 4.4b, is defined by the first computed cells at the fringe of the ghost cells, which helps to address both issues.

In summary, a Ghost-Cell Immersed Boundary Method is defined by four primary parameters: the positions of the ghost cells (GC), wall points (WP), and image points (IP), and the boundary conditions. In addition to these primary parameters, secondary parameters, typically derived from the primary ones, are essential in determining the overall accuracy of the method. These secondary parameters include the total number of triplets (GC, WP, and IP) and how multi-value cells are treated.



(a) Image point as the symmetric point of the ghost cell.

(b) Image point as the projection of the ghost cell on the *front* surface.

Figure 4.4 Illustration of image point definitions in ghost-cell methods around a complex geometry. Ghost cells are represented in circles. The image point (blue square) and wall (green diamond) points are shown for a specific ghost cell (red circle). Solid cells ($\phi < 0$) are displayed in dark gray.

4.6 The Adaptive Ghost-Cell-Surrounding Method

This section introduces a new ghost-cell method, the Adaptive Ghost-Cell-Surrounding (AGCS), which aims to address the challenges discussed above and improve the solution for high-curvature geometries. Based on an extended stencil, and therefore requiring two rows of ghost cells, the key principle of the methods relies in locally allowing the possibility of placing the first row of ghost cells in the fluid and the second row in the solid region, i.e., placing the ghost cells surrounding the wall surface. This approach can be viewed as an extension of Nakahashi's method [118], which uses only a single row of ghost cells and allows their placement in the fluid region where the body thickness is smaller than the cell size. Our method expands upon this idea, significantly reducing the likelihood of encountering the multi-value problem while preserving the discretization order near the wall.

This approach significantly enhances solution accuracy, as will be demonstrated later. The CCST method is employed to enforce the immersed boundary (IB) conditions like in Dadone and Grossman's work [49, 51]. We will show in Section 4.7.3.1 that part of the improvement brought by the AGCS method does not come exclusively from addressing the multi-value problem. Rather, it arises from the CCST reconstruction of the solution in ghost cells placed in the fluid, which acts as a pseudo-wall function, enabling more accurate gradient predictions near the wall streamline in regions of high dimensionless curvature. A detailed explanation of the method follows.

4.6.1 IB Preprocesing

The steps undertaken during the preprocessing phase of the method are detailed below. This phase primarily focuses on setting up the necessary triplets, i.e, the ghost cells, and their corresponding image and wall points.

1. First, a line-of-sight algorithm [19] is used to distinguish solid and fluid cells. Figure 4.5a displays the nature of each cell.

2. Then, the signed distance field from the IB, ϕ, is computed. Note that the location of the IB is defined by the level-set $\phi = 0$. Then normal field $\vec{n}_\phi$ is computed following Eq. 4.22. The signed distance and the normal direction, which has been materialized with streamtraces, are displayed in Fig. 4.5b. Similarly, the derivatives of the normal field are obtained in order to calculate the curvature (Eq. 4.23). In 3D simulations, the curvature depends on the velocity direction, so its value must be updated during the simulation.

3. The next step is to identify the ghost cells. Typically, two rows of ghost cells are positioned within the solid; however, in the present method, some fluid cells around the interface of the solid may be considered as ghost cells. These exceptions are outlined in detail below:

    - In high-curvature regions, and where the geometry is locally convex, the solution is improved by adding ghost cells in the fluid zone. Cells are considered locally convex if they meet the local dimensionless curvature criterion $\kappa\Delta x > 1/10$. Note that the criterion is mainly geometry-base and that the arbitrary value of 1/10 will be justified later, in Section 4.7.1.2. In Fig. 4.5c, these additional cells are those located in the white fluid zone, which indicate that they meet the criterion for dimensionless curvature.

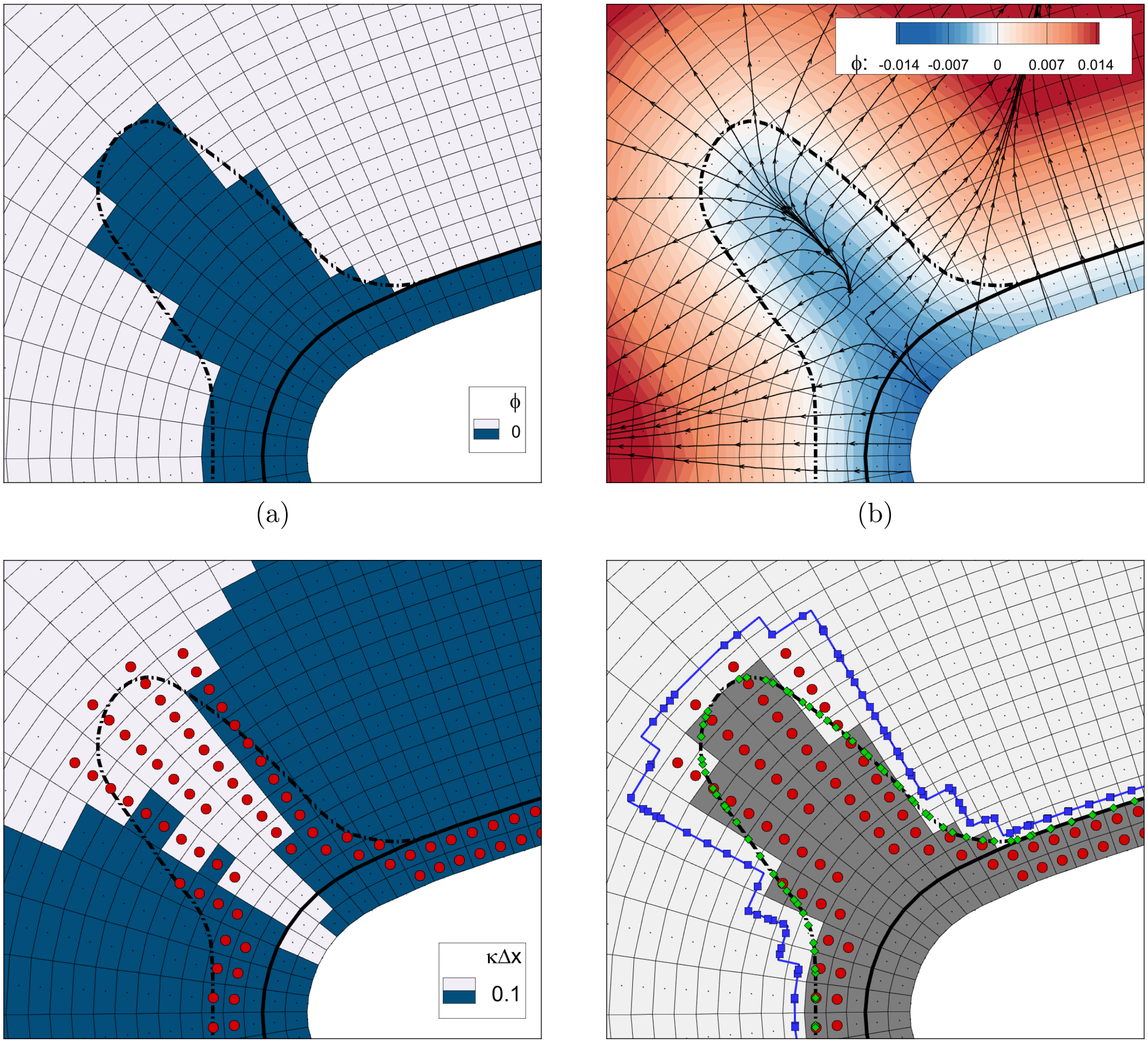
- If the thickness of a body narrows and there is insufficient space to place four ghost cells, some ghost cells must account for both sides of the body. Following the approach proposed by Nakahashi [118], ghost cells are added into the flow region to mitigate the multi-value problem. In Fig. 4.5c, these additional cells are those located in the dark fluid zone, which indicate that, despite they don't meet the criterion for dimensionless curvature, they are added to alleviate the multi-value problem.

As will be demonstrated in Section 4.7.3, these cells significantly enhance the solution without increasing the complexity of the algorithm.

4. Then, the *front* surface is determined at the fringe of the ghost cells. In some regions the wall may coincide with this front, potentially leading to numerical inconsistencies. To mitigate this issue, a minimum distance, typically set as a fraction of the cell size, is required. Numerical experiments suggest that the method is not highly sensitive to the distance of the image points to the wall, as will be shown in Section 4.7.3.1. The image points are obtained by projecting, following the normal field $\vec{n}_\phi$, the centers of the ghost cells onto the *front*, while the wall points are the projections onto the wall. Figure 4.5d displays the IB triplets as well as the *front* surface, which is represented by a blue solid line.

5. Finally the interpolation data for image points are computed (donor cell indices and weights). In the present approach, the interpolation is achieved by a 2^{nd}-order Lagrange interpolation. The interpolation stencil consists of two cell centers in each direction surrounding the image point, i.e., 4 points are used for 2D simulations, while 8 points are required for 3D simulations.

At the conclusion of the preprocessing stage, all cells of the grid are classified as computed cells, ghost cells, or non-computed cells, where solution is not obtained. It is important to note that non-computed cells are those for which obtaining a solution is unnecessary, as they do not influence the remainder of the simulation domain.



(a) (b)

(c) (d)

Figure 4.5 Close-up view near the top horn of the manufactured iced geometry (dashed-dotted line) around the GLC305 airfoil (solid line), illustrating the IB preprocessing steps. Ghost cells are indicated by red circles, wall points by green diamonds, and image points by blue squares.

4.6.2 IB Conditions

The CCST method is used to enforce the immersed boundary conditions. Unlike other ghost-cell methods [49, 51], in this approach, the image point of each ghost cell is not located symmetrically with respect to the immersed boundary, as illustrated in Fig. 4.6.

Consequently, the non-penetration condition (Eq. 4.11) must account for the different signed

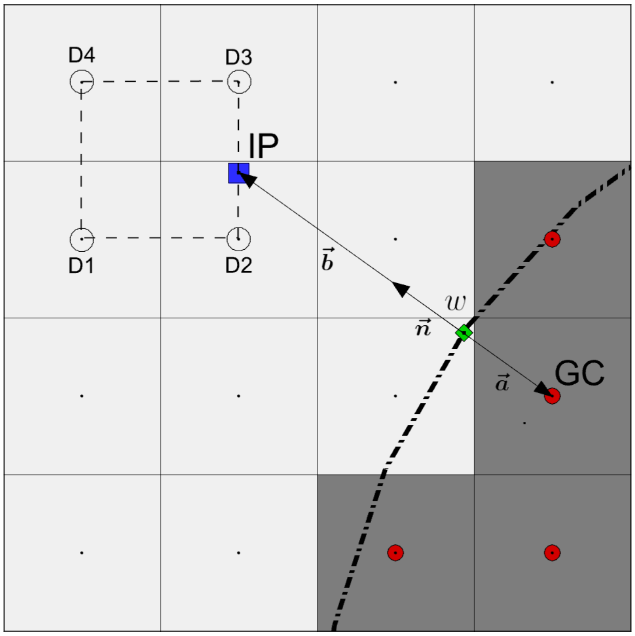


Figure 4.6 Ghost cell used in an IB simulation to impose the boundary conditions. Immersed boundary is represented in a dashed dot thick line and solid cells in dark gray.

distances between the image point and the ghost cell relative to the wall, that is,

$$v^n_{\text{GC}} = \frac{\alpha}{\beta} v^n_{\text{IP}}, \tag{4.25}$$

where $\alpha = \vec{a} \cdot \vec{n}$ and $\beta = \vec{b} \cdot \vec{n}$ are the signed distance between the wall point to the ghost cell and to the image point respectively, with $\vec{a}$ and $\vec{b}$ displayed in Fig. 4.6.

The application of the normal momentum relation (Eq. 4.12) and normal entropy conservation (Eq. 4.13) is straightforward:

$$p_{\text{GC}} = p_{\text{IP}} - \rho_{\text{IP}} (v^\tau_{\text{IP}})^2 \kappa_w (\beta - \alpha), \tag{4.26}$$

$$\rho_{\text{GC}} = \rho_{\text{IP}} \left(\frac{p_{\text{GC}}}{p_{\text{IP}}} \right)^{1/\gamma}. \tag{4.27}$$

However, the total enthalpy conservation (Eq. 4.14) requires additional considerations. Since the normal velocity component does not necessarily take the same value for the ghost cell and the image point (Eq. 4.25), these variables must be taken into account in the total enthalpy conservation equation:

$$(v^{\tau}_{\mathrm{GC}})^2 = (v^{\tau}_{\mathrm{IP}})^2 + (v^{n}_{\mathrm{IP}})^2 + \frac{2\gamma}{\gamma - 1}\left(\frac{p_{\mathrm{IP}}}{\rho_{\mathrm{IP}}} - \frac{p_{\mathrm{GC}}}{\rho_{\mathrm{GC}}}\right) - \left(\frac{\alpha}{\beta}v^{n}_{\mathrm{IP}}\right)^2 . \tag{4.28}$$

Note that under certain conditions the term $-\left(\frac{\alpha}{\beta}v^{n}_{\mathrm{IP}}\right)^2$ may become predominant in Eq. 4.28. This dominance may result in a non-physical solution for the tangential velocity at the ghost cell ($(v^{\tau}_{\mathrm{GC}})^2 < 0$), and instead, a symmetrical value is used: $v^{\tau}_{\mathrm{GC}} = v^{\tau}_{\mathrm{IP}}$. However, such circumstances are extremely rare and are typically confined to points where the velocity at the image point aligns with the surface normal. For instance, in the case of the flow around the cylinder discussed in Section 4.7.1.1, this phenomenon occurs in only four ghost cells across all studied meshes: two in the leading-edge region and two in the trailing-edge region.

4.6.3 IB Post-Processing

In contrast to body-fitted approaches, extracting the flow fields directly at wall boundaries is not straightforward when using IBMs. Different methods can be used to obtain the solution at the nodes of the discrete surface. In this approach, the same interpolation technique used to derive the value of the image point is employed, i.e., a 2$^{\mathrm{nd}}$-order Lagrange interpolation.

To perform the interpolation, all cells in the interpolation stencil for each node of the discrete surface must be classified as either computed or ghost cells. However, since the present method may place some ghost cells in the fluid domain, the two rows of the ghost cells and the non-computed cells may locally be shifted away from the solid region. There may thus be instances where non-computed cells are included in the interpolation stencil. An example is illustrated in Fig. 4.7, where the discrete surface is represented by a dashed dot thick line. In this figure, cell A is not initially required to be a ghost cell, as no fluid cell includes it in its computational stencil. However, during the post-processing step, all wall nodes within the dashed square require its information for interpolation. For this reason, the flow state variables of cell A are determined by treating it as a ghost cell, meaning its values are derived from its associated image and wall points.

4.7 Numerical Results

The current method demonstrates a degree of flexibility that makes it suitable for various applications; however, it has been primarily designed for conducting multi-step icing simulations. These simulations present several noteworthy aspects. First, the initial geometry, without any ice accretion, is well-known and has been extensively studied, allowing for easy

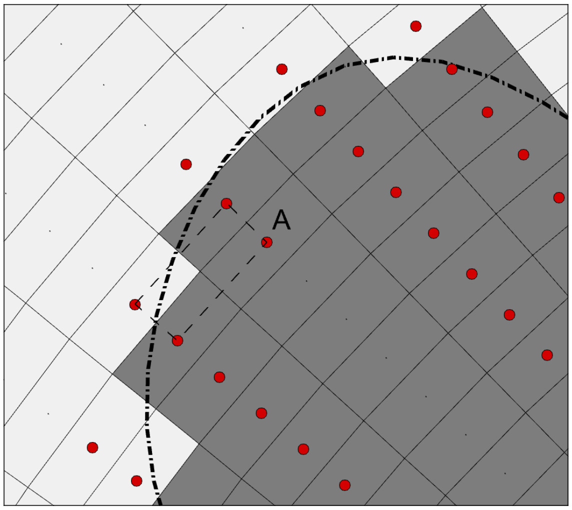


Figure 4.7 Classification of a non-computed cell (A) as a ghost cell for post-processing interpolation.

access to pre-existing volume meshes of the wing. In addition, ice typically accumulates on a limited region of the wing, such as the leading edge in single-element airfoil simulations, leaving much of the geometry unchanged. Finally, the aerodynamic solver is often integrated with other simulations, such as the droplet trajectory module, requiring the meshes to be easily transferable between different codes. While many IBMs are employed with Cartesian meshes, it is advantageous here to use body-fitted meshes for the clean initial geometry and apply the IBM only in the localized region around the ice accretion, as has been successfully applied in previous studies [61, 89].

In this section, the performance of the Adaptive Ghost-Cell-Surrounding method is presented across various test cases. First, two verification cases are analyzed, where the new approach is studied in detail on Cartesian meshes: the flow around a two-dimensional cylinder and a sphere. These cases are used to confirm that second-order accuracy in space is achieved. Next, two real-world applications of icing simulations are examined to validate the method. The first case involves a manufactured iced geometry, based on a GLC305 airfoil, which features two horns near the leading edge. This case shows the interest of the AGCS method. The final case study focuses on an iced shape around a three-dimensional swept wing.

In the following simulations, convergence is assessed using the L_2 and L_∞ norms of the density, momentum, and energy residuals.

4.7.1 Verification Case 1: Flow Around a Cylinder

The effectiveness of the present method on 2D Cartesian grids has been thoroughly evaluated by examining the subsonic, compressible, and inviscid flow around a cylinder. The analysis assumes a freestream Mach number of $M_\infty = 0.38$, leading to a peak Mach number of about 0.92 on the cylinder's surface. This case has previously been studied in other immersed boundary methods for 2D simulations [49, 185].

A set of five meshes is created (see Table 4.1), with $D/\Delta x$ values of 10, 20, 40, 80 and 160, where D represents the cylinder's diameter and Δx denotes the Cartesian cell size of the inner region. For each mesh, the refined central region extends horizontally and vertically $2.5D$, ensuring improved resolution in critical areas. Beyond this region, the grid cells increase progressively in size, resulting in a coarser mesh that effectively minimizes the total number of cells required for the simulation. The far-field boundary is 20 diameters from the cylinder. Figure 4.8 presents a partial view of the mesh with cell size $D/\Delta x = 20$. Additionally, to minimize errors associated with geometric discretization, the cylinder surface is represented using 2,048 points. For the results presented below, the residuals are converged to 10^{-8} using a CFL number of 2.

$D/\Delta x$	Total number of cells
10	3,200
20	6,400
40	25,600
80	102,400
160	409,600

Table 4.1 2D cylinder case: mesh parameters.

4.7.1.1 Mesh Convergence

To evaluate the accuracy of the IBM, the method's order of convergence, p, is determined using the finest meshes with grid resolutions of 40, 80, and 160. Due to the cylinder's constant curvature along its surface, $\kappa = 2/D$, no ghost cells are located within the fluid region for the meshes considered. This occurs because the local curvature values in these grids do not meet the established criterion ($\tilde{\kappa} \geq 1/10$).

The evaluation of the order of convergence focuses on analyzing three key variables across the selected grids. According to other ghost-cell methods [49, 51], the root mean square (RMS) of the dimensionless velocity normal to the surface, $\langle \tilde{v}^n \rangle$, is used. In addition, the

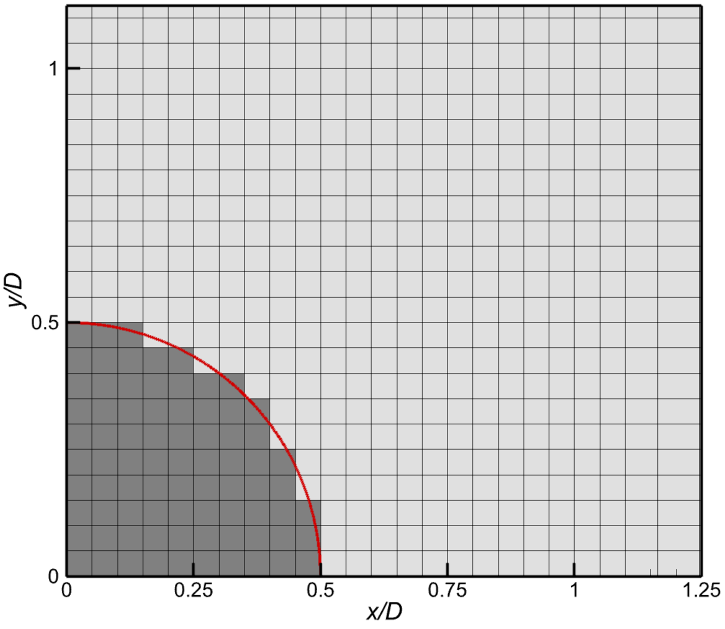


Figure 4.8 2D cylinder case: partial view of the mesh with cell size $D/\Delta x = 20$.

Mach number at the leading edge M_{le} and at the trailing edge M_{te} are also employed. It is important to note that the three variables of interest tend toward zero, allowing their values to be interpreted as the corresponding errors. These three parameters are plotted in Fig. 4.9. The results indicate that the asymptotic region of convergence has been attained.

The observed order of convergence p, derived from the errors of two meshes [1], is expressed as follows:

$$p = \frac{\ln(E_{h_2}/E_{h_1})}{\ln(r_{21})}, \tag{4.29}$$

where E_{h_i} represents the error associated with mesh i, and $r_{21} = \Delta x_2/\Delta x_1$ denotes the refinement factor. In this case, for two consecutive meshes $r_{21} = 2$. The convergence orders observed from the last three, p_{40-160}, and the last two, p_{80-160}, meshes are reported in Table 4.2. Despite some oscillations in the $\langle \tilde{v}^n \rangle$ curve, the observed order of convergence closely aligns with the theoretical value of $p = 2$.

	M_{le}	M_{te}	$\langle \tilde{v}^n \rangle$
Order p_{40-160}	1.98	1.96	1.83
Order p_{80-160}	1.99	1.98	2.28

Table 4.2 2D cylinder case: observed order of convergence using the present IBM.

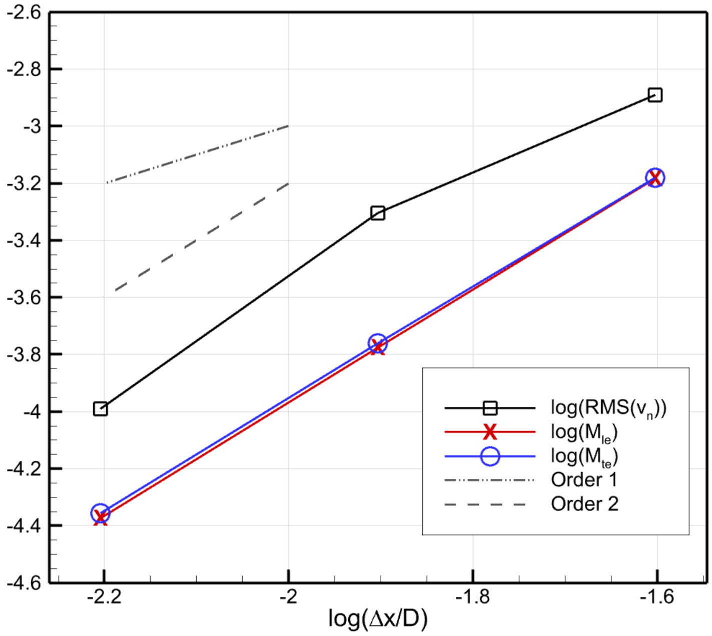


Figure 4.9 2D cylinder case: mesh convergence using the present IBM.

4.7.1.2 Local dimensionless curvature criterion

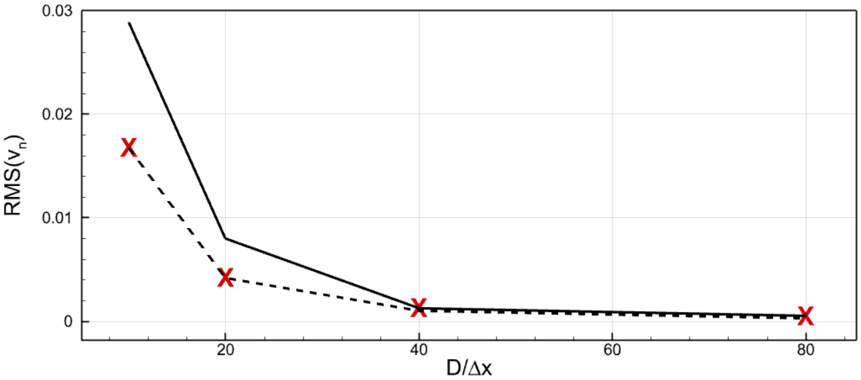
The objective of this subsection is to justify the election of $\tilde{\kappa} \geq 1/10$ as the curvature-based criterion of the AGCS method. Although the cylinder is a relatively simple geometry, it allows to analyze the impact of the value used in the local dimensionless curvature criterion. Due to its constant curvature, there exists a cell size at which the transition occurs from the *GC-surrounding* method, where the first row of ghost cells is positioned on the fluid side, to the *GC-in-solid* method, where all ghost cells are placed within the solid. By adopting the threshold value of the dimensionless curvature $\tilde{\kappa} = 1/10$, the smallest cell size where ghost cells are still placed on the fluid side corresponds to $D/\Delta x = 20$.

To conduct this research, the RMS of two variables are chosen to carry out this study: the dimensionless velocity normal to the wall and the normalized surface total pressure error,

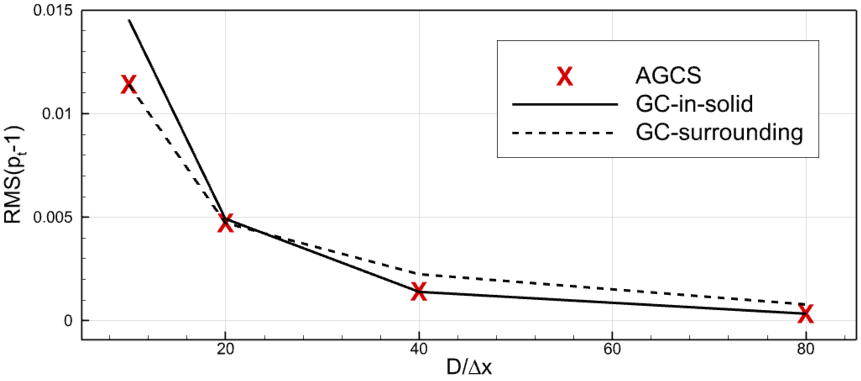
$$\tilde{p}_t - 1 = \frac{p_t - p_{t,\infty}}{p_{t,\infty}}. \tag{4.30}$$

The convergence of each error is shown in Fig. 4.10.

Concerning the normal velocity error, *GC-surrounding* provides better results for all meshes, as it involves a larger number of ghost cells, and therefore, reducing the discretization error of the boundary. For the studied meshes, the total number of ghost cells of both methods



(a) Surface normal velocity error



(b) Total pressure error

Figure 4.10 2D cylinder case: mesh convergence using different methods. Solid lines refer to *GC-in-solid*, dashed lines to *GC-surrounding*, and "X" marks to AGCS.

are presented in Table 4.3. In this context, Fig. 4.10a further validates the previous discussion, illustrating the trends observed in the normal velocity error and confirming that *GC-surrounding* performs better than the *GC-in-solid* method, especially for the coarser meshes.

Mesh, $D/\Delta x$	10	20	40	80	160
GC-in-solid	48	108	220	444	900
GC-surrounding	68	136	264	536	1068

Table 4.3 2D cylinder case: number of ghost cell for the studied meshes for the *GC-in-solid* and *GC-surrounding* methods.

Regarding the total pressure error, Fig. 4.10b highlights the importance of transitioning to the *GC-in-solid* method as the mesh resolution increases. For meshes with cell sizes

$D/\Delta x \leq 20$, the *GC-surrounding* method exhibits lower total pressure error. In contrast, for finer meshes ($D/\Delta x > 20$), the *GC-in-solid* method presents better performance. Based on the results of this case, the criterion for considering locally convex cells has been established, $\tilde{\kappa} \geq 1/10$. However, it is important to note that this value is not an absolute optimum, as the optimal threshold depends on various aspects of the simulation, including the geometry, the slope limiters employed, and the free-stream conditions, among other factors. Furthermore, additional simulations with slight variations in the transition value for complex geometries have demonstrated that the results are not particularly sensitive to these changes. Moreover, it has also been observed that total enthalpy conservation improves when using the *GC-in-solid* approach.

In Section 4.7.3.1, a high-curvature 2D geometry is studied, demonstrating the advantage of positioning ghost cells in high-curvature regions within the fluid.

4.7.2 Verification Case 2: Flow Around a Sphere

To further verify the performance of the present IBM, we analyze the subsonic, compressible, and inviscid flow around a sphere. This case examines a freestream Mach number of $M_\infty = 0.52$, producing a peak Mach number of approximately 0.89 at the sphere's surface, at $x = 0$ [51].

For this case, three grid resolutions with $D/\Delta x$ values of 16, 24, and 32 are used (see Table 4.4), where D represents the diameter of the sphere and Δx is the size of the Cartesian cells in the refined region around the sphere. The computational domain extends $10D$ in each direction, ensuring negligible influence of the outer boundary on the flow near the sphere. The central refined region spans $1.5D$ in each direction around the sphere, providing higher resolution close to the surface, while grid cell sizes progressively increase in the outer regions to reduce computational cost.

$D/\Delta x$	Total number of cells
16	512,000
24	1,728,000
32	4,096,000

Table 4.4 3D sphere case: mesh parameters.

The sphere's surface is represented by 14,406 quadrilaterals to ensure high accuracy in geometric discretization. Residuals are converged to a value of 10^{-8} for all simulations, using a CFL number of 2.

The convergence rate of the present IBM is evaluated again by examining $\langle \tilde{v}^n \rangle$, M_{le} and M_{te}. These variables, once more, tend toward zero, allowing their values to serve as error measures. Figure 4.11 presents the convergence behavior for these three parameters.

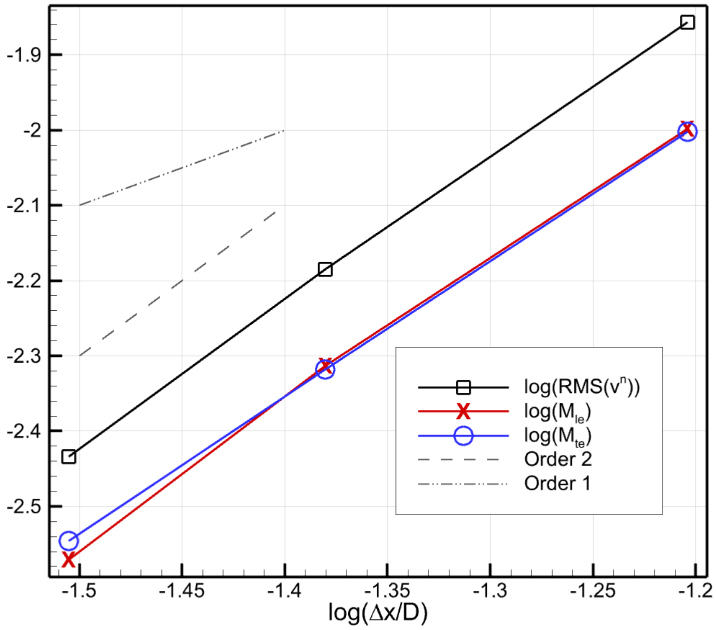


Figure 4.11 3D sphere case: mesh convergence using the present IBM.

The convergence orders p are calculated again based on the errors obtained from two pairs of meshes and are presented in Table 4.5. The results show that $p \approx 2$ is achieved for each variable, aligning with the second-order accuracy expected of the method.

	M_{le}	M_{te}	$\langle \tilde{v}^n \rangle$
Order p_{16-32}	1.90	1.81	1.92
Order p_{24-32}	2.06	1.87	1.99

Table 4.5 3D sphere case: observed orders of convergence using the present IBM.

4.7.3 Application Case 1: Flow Around an Iced Airfoil.

In this section, the flow around an artificially defined iced GLC305 airfoil is examined. The geometry replicates characteristic double-horn shapes observed in experimental studies, such as Case 242 from the 1^{st} Ice Prediction Workshop (IPW) [86]. The analysis considers subsonic flow at a freestream Mach number $M_\infty = 0.25$ and zero angle of attack, $\alpha = 0$. Please, note that the shape created is intentionally slightly non-symmetrical to represent a more realistic case. Specifically, the small difference between the upper and lower horns results in a higher intensity of the suction peak in the lower horn. For all simulations performed in this case, the residuals converge to 10^{-6}, using a CFL number of 2.

Two types of mesh are analyzed: body-conforming and non-conforming. The body-conforming meshes, also referred as Body-Fitted (BF) meshes, represent the ideal case for an IBM, as the cells align with the geometry, avoiding the multi-value problem (Fig. 4.12 top). These meshes not only provide a reference solution for comparison, but also allow for a detailed examination of the effect of identifying certain fluid cells as ghost cells in regions of high dimensionless curvature near the wall. The non-conforming meshes, also referred as Immersed Boundary (IB) meshes, demonstrate a practical application in ice accretion simulations (Fig. 4.12 bottom). As previously mentioned, a body-fitted mesh of the clean airfoil, without ice, is utilized. In addition, the mesh size is kept constant in the normal direction to the wall for a distance of approximately 10% of the characteristic length, in an attempt to maintain consistent discretization error.

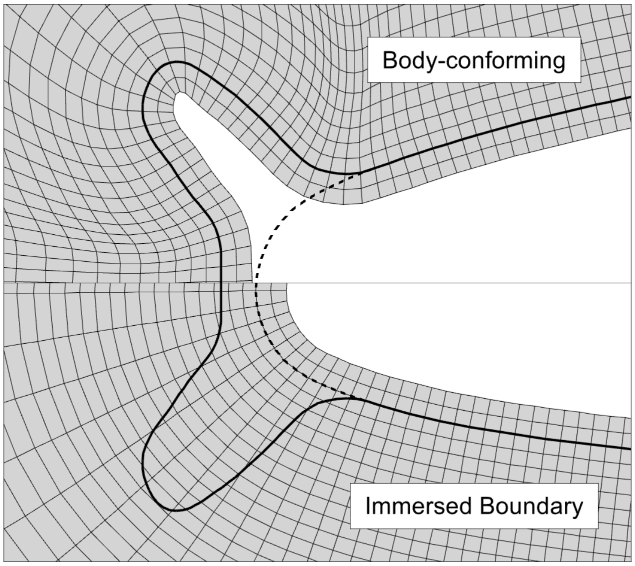


Figure 4.12 2D iced airfoil case: partial view of the coarse meshes used ($\Delta x_{\mathrm{le}}/c = 2 \cdot 10^{-3}$). The ice shape is represented by a thick solid line, while the clean GLC305 airfoil by a thick dashed line.

For each type of mesh, a family of meshes has been obtained, progressively doubling the number of cells in each direction. The parameters are presented in Table 4.6, where $\Delta x_{\mathrm{le}}/c$ represents the non-dimensional mesh size at leading edge (c is the chord length), and N the total number of cells. Figure 4.12 provides a comparison of both mesh types for a given size.

Mesh name	$\Delta x_{\mathrm{le}}/c$	N_{BF}	N_{IB}
Coarse	$2 \cdot 10^{-3}$	40,888	43,560
Medium	$1 \cdot 10^{-3}$	163,552	174,240
Fine	$5 \cdot 10^{-4}$	654,208	696,960
Very fine	$2.5 \cdot 10^{-4}$	2,616,832	-

Table 4.6 2D iced airfoil case: mesh parameters. N_{BF} and N_{IB} represent the total number of cells for the body-fitted and immersed boundary meshes, respectively.

4.7.3.1 Solution on Body-Conforming Meshes

The application of a classic wall boundary condition for a body-conformal mesh is obtained using two layers of ghost cells within FastS solver. In this case, the *GC-in-solid* approach replicates the classic boundary condition. However, applying the AGCS method on these meshes allows for isolating the impact of placing certain ghost cells within the fluid domain, as most of the other parameters remain constant. Notably, the number of triplets remains unchanged regardless of whether the ghost cells are placed in the solid or fluid regions. Additionally, due to the mesh conformity to the geometry, no cell encounters the multi-valued problem.

First, a reference solution is obtained. Next, the solution obtained using the AGCS method is presented. Finally, a detailed comparison is made between the AGCS solution and that of the *GC-in-solid* method.

Reference Solution

The very fine body-conforming mesh is used to obtain a reference solution. Specifically, the distribution of the pressure coefficient on the iced profile has been selected as the variable for analysis. However, achieving a converged solution in such complex geometries presents significant challenges, even for inviscid flows and body-conforming grids [89]. For the very fine grid, the mesh size is sufficiently small to consider all the cells around the wall locally flat. Consequently, the dimensionless curvature criterion is not met, and all ghost cells are positioned within the solid domain. In other words, the AGCS and *GC-in-solid* methods become equivalent, yielding identical solutions. In addition, the slope limiters could be deactivated while still achieving a solution with residuals converging to 10^{-6} on the fine meshes. Compared to the fine mesh, the differences in the two minimum pressure peaks are minimal, as shown in Fig. 4.13 allowing the very fine solution to serve as the reference, which will be used henceforth to assess the level of accuracy of the compared methods.

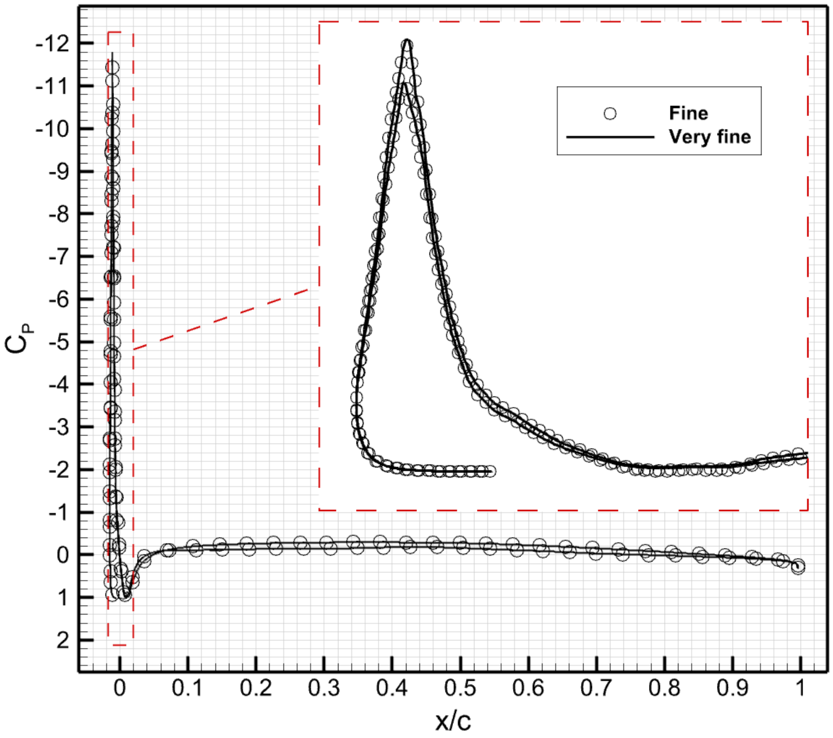


Figure 4.13 2D iced airfoil case: Cp convergence on body-conforming meshes. 2^{nd} order solution without slope limiters.

AGCS Solution

When using the coarsest meshes, slope limiters are essential to ensure both flow attachment and solution convergence. In this study, the minmod slope limiter has been selected, and it will be applied consistently in all subsequent analyses and results presented, unless explicitly mentioned.

Figure 4.14 illustrates C_P distribution around the leading edge of the geometry. Notably, regardless of the mesh resolution employed, the flow remains attached to the curved surface, indicating the robustness of the computational approach in preserving the physical characteristics of inviscid flow.

Comparison of AGCS and GC-in-solid

Next, the AGCS and *GC-in-solid* methods are compared. The study aims to exclusively analyze the influence of the position of the ghost cells. By applying the same boundary conditions (CCST), there are indeed only two parameters that change: the location of the ghost cells and their corresponding image points. Therefore, three simulations are conducted which vary the location of the ghost cells and their image points, as presented in Table 4.7

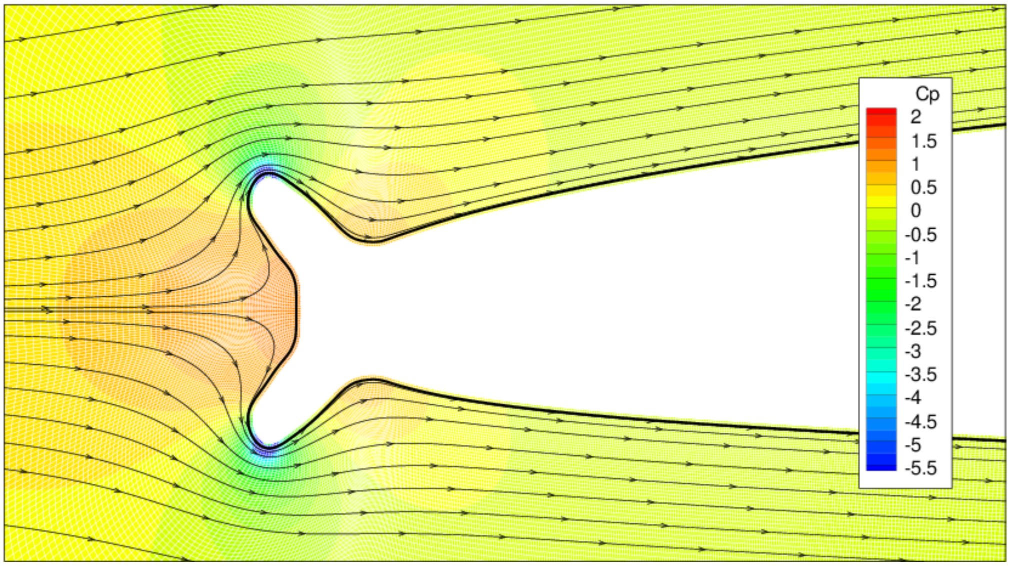


Figure 4.14 2D iced airfoil case: partial view of the flow field C_P using the fine body-conforming mesh ($\Delta x_{\mathrm{le}}/c = 5 \cdot 10^{-4}$). 2^{nd} order AGCS solution with slope limiters.

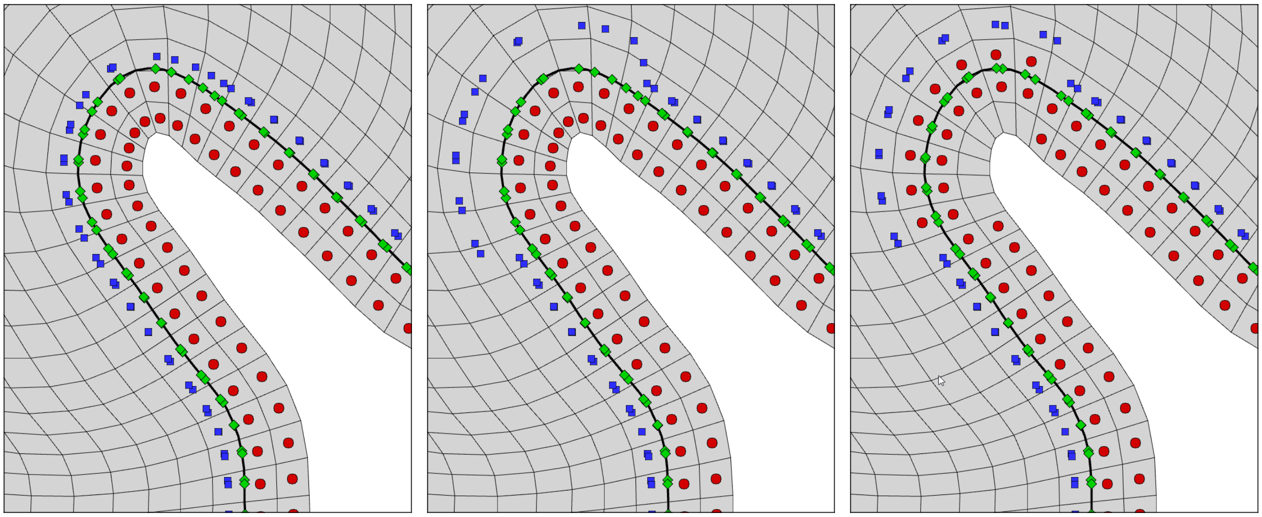
and Fig. 4.15. Note that an intermediate method is used for the comparison, and it is referred to as *GC-in-solid-AIP*, where AIP stands for Adaptive Image Points. This method facilitates a direct assessment of the influence of image point placement when compared to the *GC-in-solid* approach. Similarly, when compared to the AGCS method, the focus shifts to evaluating the impact of ghost cell positioning on the results.

	GC-in-solid	*GC-in-solid-AIP*	*AGCS*
GC	solid	solid	based on $\tilde{\kappa}$
IP	fringe of solid cells	based on $\tilde{\kappa}$	based on $\tilde{\kappa}$

Table 4.7 2D iced airfoil case: position of the ghost cell and image points for each performed simulation.

For the AGCS method, the number of ghost cells in the fluid zone is 17, 30, and 16 for the coarse, medium, and fine meshes, respectively.

The pressure coefficient distribution of the three simulations are displayed in Figure 4.16. Two key conclusions can be drawn from the results. First, the solutions of *GC-in-solid* and *GC-in-solid-AIP* show only minimal differences, confirming that the image point location does not significantly affect accuracy. This is primarily because, beside *GC-in-solid-AIP* uses image points farther away from the wall to fulfill the ghost cells, the CCST boundary conditions account for this distance, leading to comparable values. In contrast, a more pronounced impact is observed when varying the position of the ghost cells, with the improvements



(a) *GC-in-solid* (b) *GC-in-solid-AIP* (c) *AGCS*

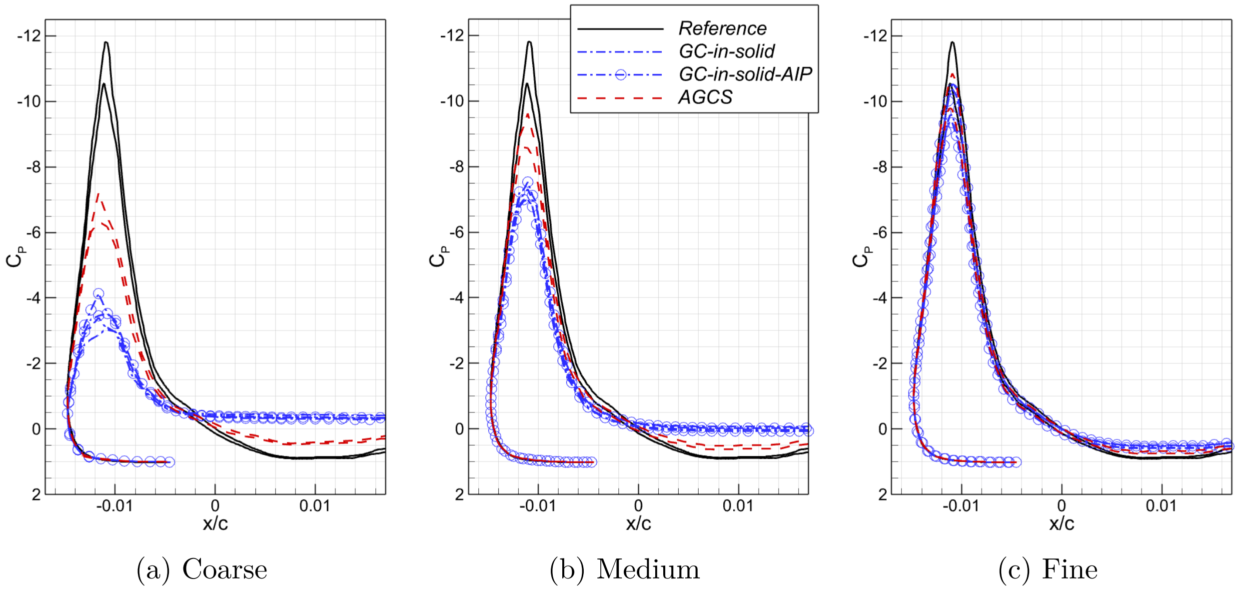
Figure 4.15 2D iced airfoil case: triplets in body-conforming meshes. Ghost cells are represented by red circles, wall points by green diamonds, and image points by blue squares.

introduced by the AGCS method being particularly significant. This enhancement is evident not only at the suction peak but also in the region downstream of the horns ($x/c > 0$). Notably, for a given mesh resolution, the AGCS solution achieves an error comparable to that obtained with the *GC-in-solid* method on the next finer mesh. As the mesh size is reduced, the discrepancy between solutions decreases, as a smaller percentage of cells satisfy the dimensionless curvature criterion.

In addition to the position of the ghost cells and image points, an indirect influence between the simulations may arise from the spatial discretization and the activation or non-activation of the slope limiters. To make sure that the improvement comes exclusively from the position of some ghost cells in the fluid, it has been decided to deactivate the MUSCL reconstruction, assuming piecewise constant values within the cells and achieving a first order solution. The same conclusions can be drawn in this case, as shown in Fig. 4.17. Once again, the AGCS method demonstrates a significant improvement, and the differences between *GC-in-solid* and *GC-in-solid-AIP* are further reduced.

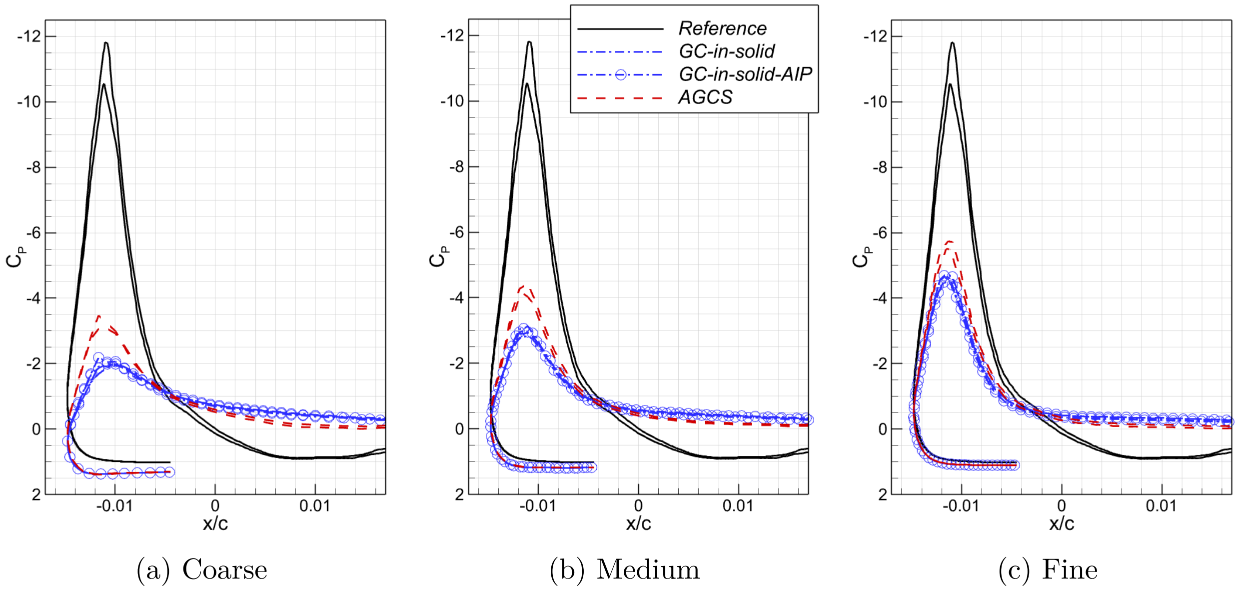
It can be concluded that reconstructing the values of selected fluid cells using CCST boundary conditions enhances the accuracy compared to directly solving the Euler equations numerically on high-curvature regions when using coarse meshes.

Since the CCST method is based on the normal momentum relation (Eq. 4.10), an additional analysis is conducted to assess whether the discretized Euler equations solution of the AGCS



(a) Coarse (b) Medium (c) Fine

Figure 4.16 2D iced airfoil case: C_P convergence around the leading edge on body-conforming meshes. 2^{nd} order solution.



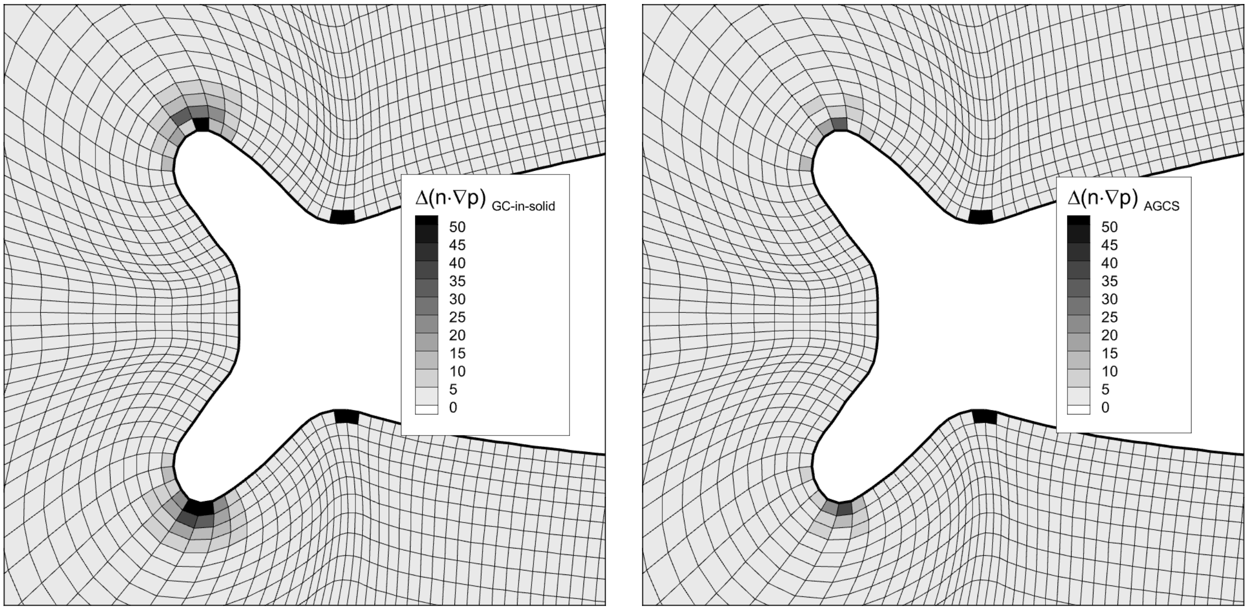
(a) Coarse (b) Medium (c) Fine

Figure 4.17 2D iced airfoil case: C_P convergence around the leading edge on body-conforming meshes. 1^{st} order solution.

method results in reduced errors in pressure gradient estimations compared to the *GC-in-solid* method. To this end, the error in the dimensionless normal pressure gradient,

$$\Delta(\vec{n} \cdot \vec{\nabla}\tilde{p}) = |(\vec{n} \cdot \vec{\nabla}\tilde{p}) - (\vec{n} \cdot \vec{\nabla}\tilde{p})_{veryFine}|, \tag{4.31}$$

is analyzed, where the pressure is expressed in terms of its freestream value, $\tilde{p} = p/p_\infty$. Note that the very fine solution is not available at the centers of the coarse grid cells. Therefore, interpolation has been employed to facilitate the comparison of the data. Figure 4.18 displays the error field on the coarse body-conforming mesh around the leading edge. The MUSCL extrapolation is used with the minmod slope limiter again. The results indicate a significant reduction in error at the horn tips when employing the proposed method (see Fig. 4.15c). In this region, the maximum value obtained by *GC-in-solid* is $\Delta(\vec{n} \cdot \vec{\nabla}\tilde{p}) \approx 45$, while AGCS provides a much lower value $\Delta(\vec{n} \cdot \vec{\nabla}\tilde{p}) \approx 22$.



(a) *GC-in-solid*

(b) *AGCS*

Figure 4.18 2D iced airfoil case: dimensionless pressure gradient error field on the coarse body-conforming mesh ($\Delta x_{\mathrm{le}}/c = 2 \cdot 10^{-3}$). 2^{nd} order solution.

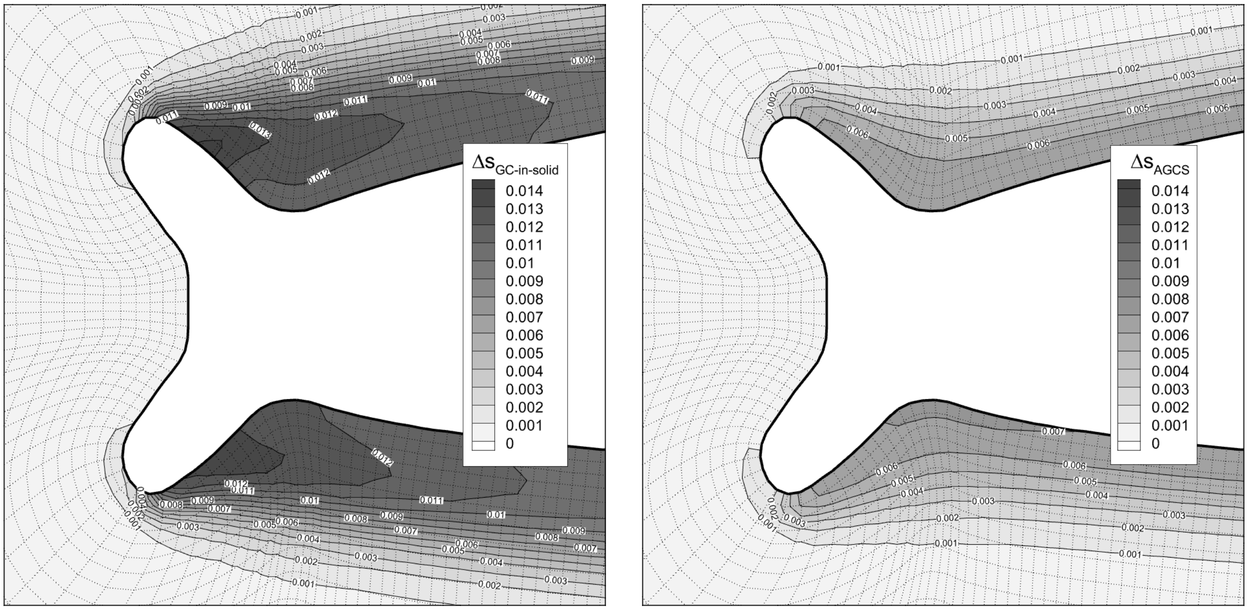
The pressure gradient is therefore more accurately represented by Eq. 4.10 than by the solution of the Euler equation on coarse meshes. The CCST approach can thus be viewed as a pseudo wall function, which obtains more accurate values of the pressure gradient in high-curvature cells when the discretization error is large.

Following the pressure error analysis, the error in entropy, s, should also be assessed, as entropy should be conserved throughout the computational domain due to the lack of heat sources and external forces in the presented inviscid flow. The entropy error is calculated relative to the freestream conditions

$$\Delta s = \frac{s - s_\infty}{s_\infty}, \tag{4.32}$$

where $p_\infty = 100\text{kPa}$ and $T_\infty = 273.15\text{K}$ are used for the reference state.

Figure 4.19 illustrates the contours for the entropy error, again for both methods on the coarse body-fitted mesh. Similar to the pressure gradient error, the *GC-in-solid* method exhibits larger entropy errors downstream of the horns, with a maximum error of $\Delta s_{\text{GC-in-solid}} = 0.014$. In contrast, the AGCS method significantly reduces the entropy error in these critical regions, achieving a maximum error of $\Delta s_{\text{AGCS}} = 0.007$, and providing a smoother and more consistent error distribution throughout the computational domain.



(a) GC-in-solid

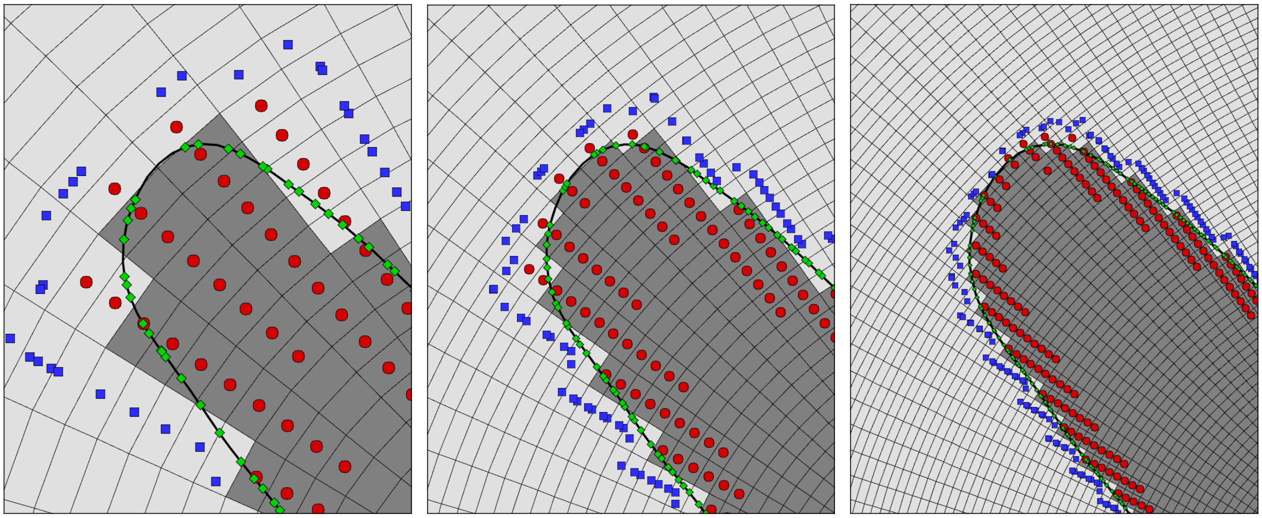
(b) AGCS

Figure 4.19 2D iced airfoil case: entropy error around the leading edge on the coarse body-conforming mesh ($\Delta x_{\text{le}}/c = 2 \cdot 10^{-3}$). 2^{nd} order solution.

Based on the previous observations, it can be concluded that, in body-conforming meshes, the AGCS method is more accurate and better suited for geometries with high curvature, particularly when applied to coarser meshes. Next, the solution using immersed boundary meshes will be analyzed in detail.

4.7.3.2 Solution on Immersed Boundary Meshes

A more relevant test case for icing application involves the treatment of the iced shape as an immersed boundary. A crucial parameter in this area is the location of the triplets. Figure 4.20 illustrates it positions for the studied meshes. In coarse meshes (Fig. 4.20a) accurately representing geometry is challenging, leading to a high likelihood of multi-valued cells. Introducing additional cells within the fluid zone alleviates this issue. When the mesh is sufficiently refined (Fig. 4.20b), multi-valued cells are avoided, and the only criterion for placing ghost cells in the fluid is the local curvature. Further refinement (Fig. 4.20c) causes cells to be treated as locally flat, resulting in most of the ghost cells being placed in the solid.

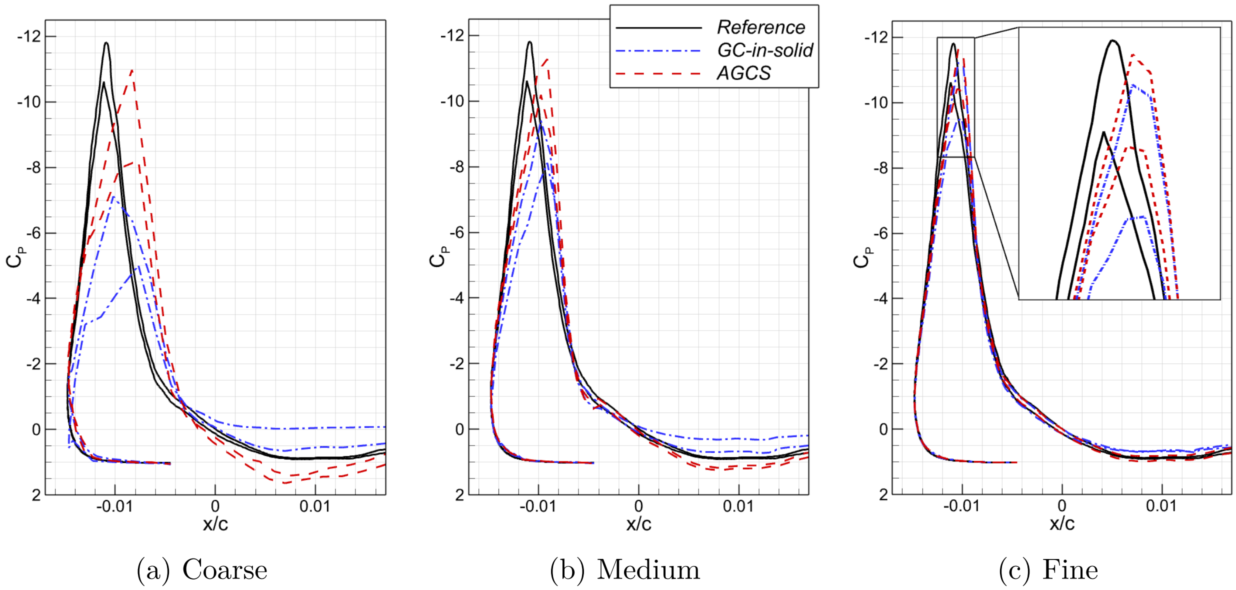


(a) Coarse (b) Medium (c) Fine

Figure 4.20 2D iced airfoil case: triplets in immersed boundary meshes. Ghost cells are represented by red circles, wall points by green diamonds, and image points by blue squares.

The C_P distributions of the three meshes using the AGCS method are presented with red dashed lines in Fig. 4.21. Again, solutions are obtained assuring that the residuals converge to 10^{-6}. The minimum C_P values are accurately captured even on the coarsest meshes, although a slight offset is present at the location where this value is obtained. As the mesh is refined, the solution converges towards the reference solution. In the finest mesh, the errors in the pressure peaks at the upper and lower horns are below 1%.

Comparing the present method with the *GC-in-solid* IBM is challenging, mainly due to the lack of a mechanism in this study to handle multi-value cells. For the coarse mesh, the improvement observed with the present method stems from two key factors. First, there is a reduction in the number of cells classified as multi-value. For instance, on the coarsest mesh, *GC-in-solid* encounters the multi-value problem on 14 ghost cells, whereas AGCS only involves one. Additionally, as previously demonstrated, positioning ghost cells within the fluid zone in regions of high local curvature enhances the solution. For the other meshes, the *GC-in-solid* method does not encounter any multi-value cell, so the improvement with the present method is entirely attributable to a more accurate representation of the pressure gradient normal to the wall around the horns.



(a) Coarse (b) Medium (c) Fine

Figure 4.21 2D iced airfoil case: C_P converge around the leading edge using immersed boundary meshes. 2^{nd} order solution.

For the fine mesh, the pressure coefficient C_P field around the horns is shown in Fig. 4.22 and can be compared to the results of the body-conforming mesh simulation (see Fig. 4.14). It is noteworthy that, despite the use of coarse meshes and the complexity of the geometry studied, all solutions obtained exhibit attached flow.

4.7.4 Application Case 2: Flow Around an Iced Wing.

The last application case deals with a NACA0012 30-degree swept wing, with a chord length of 0.914 m. This case was part of the baselines of the 1^{st} Ice Prediction Workshop [86], where it is referred to as case 361. The ice shape was obtained through a simulation similar to that described in [61], at a free-stream Mach number of $M_\infty = 0.32$ and zero degree angle of attack. This new geometry, obtained by simulating icing conditions, is symmetrical with respect to the plane $z = 0$ and it is shown in Fig. 4.23, where the dark region represents the ice accretion and the lighter region corresponds to the wing surface.

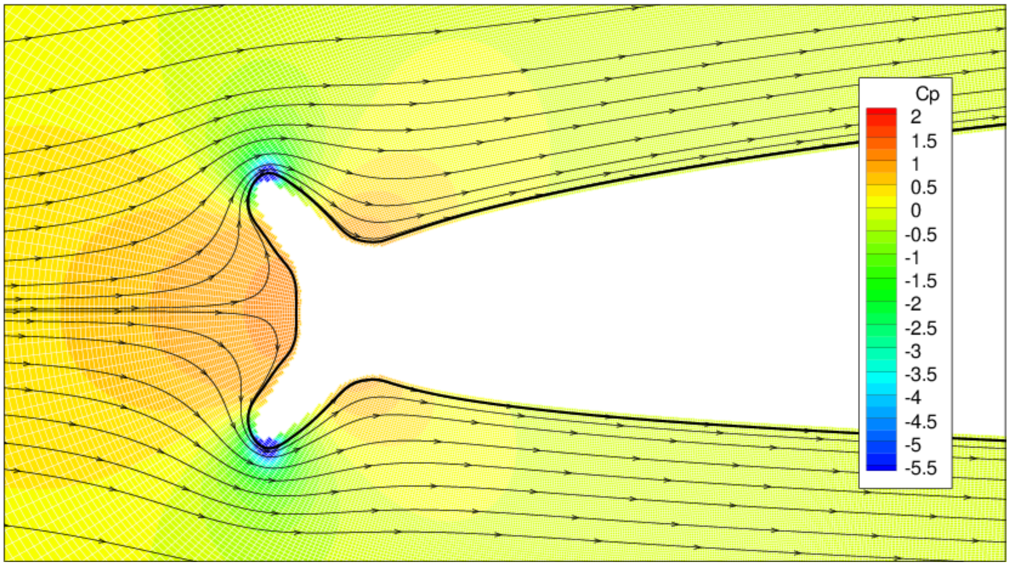


Figure 4.22 2D iced airfoil case: partial view of the flow field C_P using the fine immersed boundary mesh ($\Delta x_{\mathrm{le}}/c = 5 \cdot 10^{-4}$). 2^{nd} order AGCS solution.

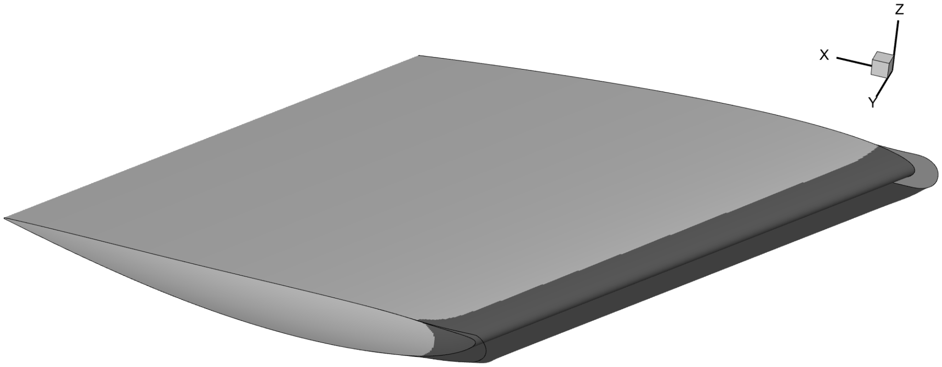


Figure 4.23 3D iced wing case: 3D view of the ice shape.

The IB simulation is performed from the volume mesh of the clean NACA0012 wing with a total number of cells of 1,839,712. The mesh size around the leading edge is $\Delta x = 3 \cdot 10^{-3}$, and it is maintained over a distance approximately equal to 10% of the chord length to the wall. The farfield extends around 4 wing-root chords in all directions. A symmetry boundary condition is enforced at the wing-root $y = 0$ plane. The three-dimensional ice shape consists of 25,664 faces and will be used as an input to perform the immersed boundary simulation. The body-conforming mesh, used for comparison, is built from the ice shape, keeping the mesh resolution of $\Delta x = 3 \cdot 10^{-3}$ at the wall.

The AGCS preprocessing is conducted and the total number of ghost cells obtained is 43,077. Figure 4.24 shows a three-dimensional view of the triplet arrangement around the ice shape. The close-up view highlights the triplets located between two sections of the wing, focusing on the leading edge. Ghost cells are marked with red circles, wall points with green diamonds and image points with blue squares. Note that the studied iced shape is relatively smooth, typical of low-temperature ice accretion simulations, so the local dimensional curvature criterion is not met. Therefore, no ghost cells are positioned within the fluid region.

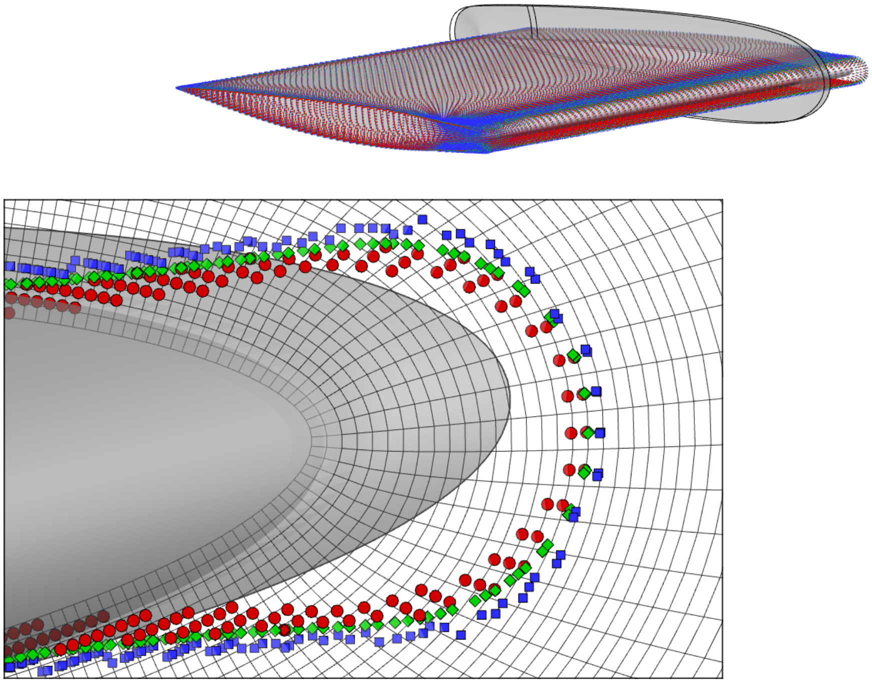


Figure 4.24 3D iced wing case: triplets in the immersed boundary mesh.

Local views of the solution pressure coefficient field at the wall is shown in Fig. 4.25. Figure 4.26 compares the Cp distribution obtained with the presented AGCSmethod (shown in red circles) and the body-fitted solution (shown in black lines), considered as references, at three spanwise locations, y/b =0.25, 0.5 and 0.75, where b is the wing span. The results show good agreement between the presented method and the reference solution. The AGCS method reproduces the Cp distribution with remarkable precision, demonstrating its ability to capture the complex flow features over a swept wing. These results highlight the effectiveness of the AGCS method and further validate its potential as a reliable tool to perform inviscid simulations in aerodynamics.

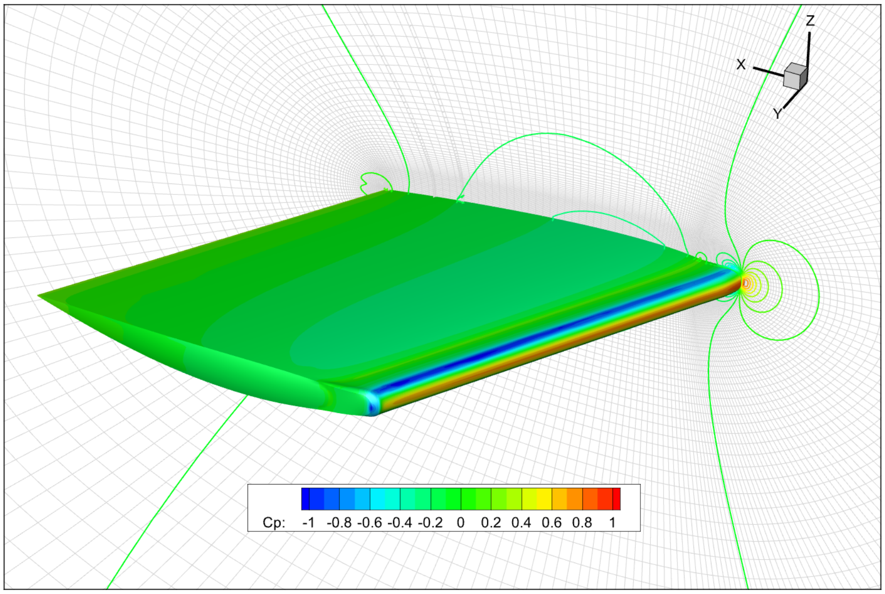


Figure 4.25 3D iced wing case. The color contours represent the distribution of pressure coefficient Cp over the iced wing surface, with the Cp isolines displayed on the $y = 0$ plane.

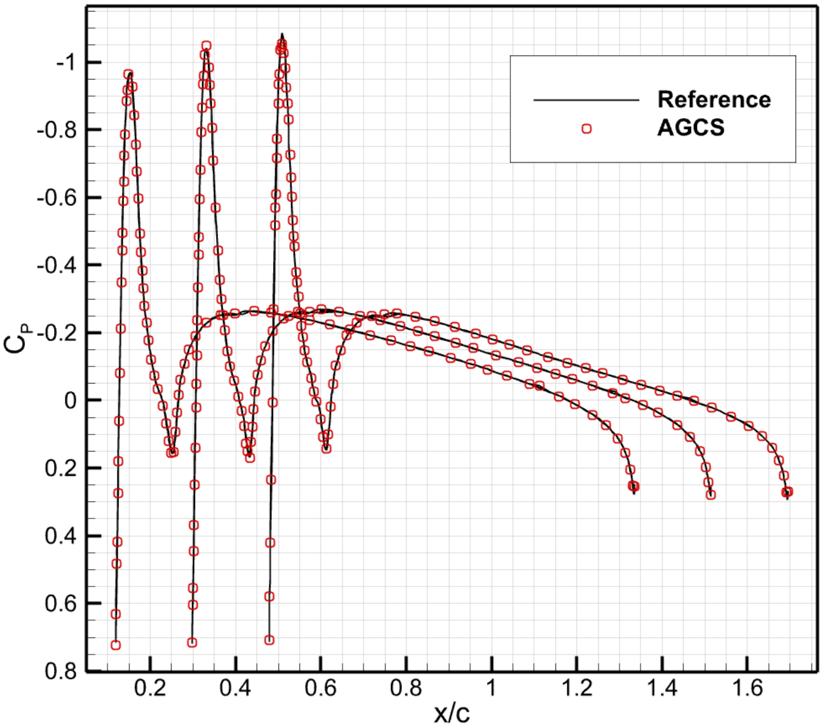


Figure 4.26 3D iced wing case: pressure coefficient Cp curves at the three spanwise locations, y/b =0.25, 0.5 and 0.75.

4.8 Conclusion

The paper presents a novel ghost-cell method, the Adaptive Ghost-Cell-Surrounding (AGCS) method, for simulating compressible, inviscid flows around complex, rigid geometries. The key principle of the method lies in locally allowing the first row of ghost cells to be placed in the fluid and the second row in the solid region, based on a local dimensionless curvature criterion. Additionally, the approach employs the Curvature-Corrected Symmetry Technique (CCST) boundary conditions, which account for wall curvature. In regions of high dimensionless curvature, where ghost cells are placed in the fluid, the CCST method acts as a pseudo-wall function, improving gradient predictions near the wall streamline. This improvement could also make the method advantageous for applications involving body-fitted meshes.

The investigation of compressible inviscid flow around a 2D cylinder and a 3D sphere demonstrates second-order accuracy on Cartesian meshes. Furthermore, the method was applied to iced shapes. The flow around a manufactured double-horn was extensively analyzed, comparing the effect of marking fluid cells as ghost cells in both body-conforming and immersed boundary meshes. Results show that the present method converges faster and generates less entropy compared to a classical IBM where the ghost cells are placed in the solid. Finally, subsonic inviscid flow over an iced 30-degree swept wing was simulated. The computed isobars on the wing surfaces and surface pressure distributions at various spanwise locations exhibit excellent agreement with body-fitted mesh results. These comparisons validate the ability of the proposed ghost-cell method to accurately model three-dimensional flow around complex ice shapes in practical applications.

Overall, the proposed method meets the original motivations as a robust and flexible simulation tool for predicting inviscid flows around complex geometries. Future work will focus on adding coupling procedures for viscous-inviscid interactions.

CHAPTER 5 THREE-DIMENSIONAL SIMPLIFIED INTEGRAL BOUNDARY-LAYER METHOD

Computing the boundary layer is necessary to obtain the diffusive transfer quantities on the wall, specifically the convective heat transfer coefficient h_{tc} and the recovery temperature T_{r}. These variables are used in the ice accretion solver to compute the wall convective heat flux $\dot{Q}_{\mathrm{c}}$ and the evaporated mass flow-rate $\dot{m}_{\mathrm{evs}}$. Since the airflow solver generally does not provide h_{tc} and T_{r} directly, different strategies for estimating these quantities can be found in the literature [140, 141].

The first approach consists in performing two Navier–Stokes simulations with different isothermal wall boundary conditions, $T_{\mathrm{w},1}$ and $T_{\mathrm{w},2}$. Based on the wall heat fluxes $\dot{Q}_{\mathrm{w},1}$ and $\dot{Q}_{\mathrm{w},2}$, which are provided by the airflow solver, the computation of h_{tc} and T_{r} is performed by solving the following system:

$$\dot{Q}_{\mathrm{w},1} = h_{\mathrm{tc}} \left(T_{\mathrm{w},1} - T_{\mathrm{r}}\right) \tag{5.1a}$$

$$\dot{Q}_{\mathrm{w},2} = h_{\mathrm{tc}} \left(T_{\mathrm{w},2} - T_{\mathrm{r}}\right) \tag{5.1b}$$

where both the heat transfer coefficient and the recovery temperature are assumed to be independent of the wall thermal condition. This approach is, however, demanding in terms of computational cost, as two Navier–Stokes simulations are required per ice accretion step.

An alternative method requires only one Navier-Stokes simulation, significantly reducing the computational cost. In this approach, the recovery temperature is determined from the recovery coefficient r_{rec}:

$$T_{\mathrm{r}} = T_{\mathrm{e}} \left(1 + r_{\mathrm{rec}} \frac{\gamma - 1}{2} M_{\mathrm{e}}^2\right), \tag{5.2}$$

where $\gamma = 1.4$ is the heat capacity ratio of air. T_{e} and M_{e} are the temperature and the Mach number at the edge of the boundary layer and can be computed from the isentropic relations [141]. The recovery coefficient is obtained from the Prandtl number of air, $\mathrm{Pr} = 0.7$. For a laminar boundary layer, $r_{\mathrm{rec}} = \mathrm{Pr}^{1/2}$. while in turbulent regime $r_{\mathrm{rec}} = \mathrm{Pr}^{1/3}$.

Another option is the use of integral boundary layer models. This approach is mandatory when performing inviscid airflow simulations [176], but it has also been used in viscid simulations [141].

Solving the inviscid flow in ice accretion simulations, as chosen in the present work, provides significant advantages, as it reduces computational cost and the complexity of mesh generation, as fine mesh cells near the wall are not required in contrast to Navier-Stokes simulations. Among the different ice accretion codes, LEWICE [152] and IGLOO2D [176], are well-known examples that rely on simplified integral boundary layer methods. Additionally, ONERA has developed an integral boundary layer method (BLIM2D and BLIM3D) as part of Charlotte Bayeux's thesis [13]. It has been applied to two- and three-dimensional ice accretion simulations [14, 23], improving the solution compared to the 2D simplified integral method and extending its applicability to the simulation of Ice Protection Systems (IPS). However, its application to certain 3D configurations still faces robustness issues.

Therefore, this chapter focuses on the extension of the simplified integral boundary layer method to three-dimensional simulations as a simple alternative to the use of BLIM3D. Section 5.1 provides an introduction to the modeling of the boundary layer, presenting the classical models commonly employed in integral solvers. Section 5.2 describes the proposed 3D simplified integral boundary layer method, detailing its formulation and implementation. Finally, Section 5.3 presents validation test cases, where the method is evaluated in two cases: the steady flows over a flat plate without pressure gradient and over a rough cylinder.

5.1 Introduction to the Modeling of the Boundary Layer

The solution of the complete viscous fluid flow around a body poses significant mathematical challenges, particularly for complex geometries. A major advancement in this field was made by Prandtl in 1904 [135], who demonstrated that for sufficiently high Reynolds numbers, viscous effects dominate only in a thin region near the wall, while the outer flow behaves as an ideal (inviscid) fluid. This insight fundamentally decoupled flow analysis into two regimes, as shown in Fig. 5.1. Within this near-wall region, known as the boundary layer, viscous forces remain comparable to inertial forces. The velocity at the wall is zero due to the no-slip condition, while at the boundary with the outer flow, it matches the velocity of the ideal fluid. Strictly speaking, the velocity within the boundary layer asymptotically approaches that of the ideal fluid only at an infinite distance from the wall, leading to some ambiguity in the definition of boundary layer thickness. A common convention is to define this thickness as the distance from the wall at which the longitudinal velocity reaches 99% of the outer flow velocity. An analogous concept applies to the thermal boundary layer, where temperature gradients become significant near the wall, as presented by Prandtl in 1910 [136]. In this region, the wall temperature is typically prescribed, and the temperature gradually approaches that of the outer flow with increasing distance from the wall.

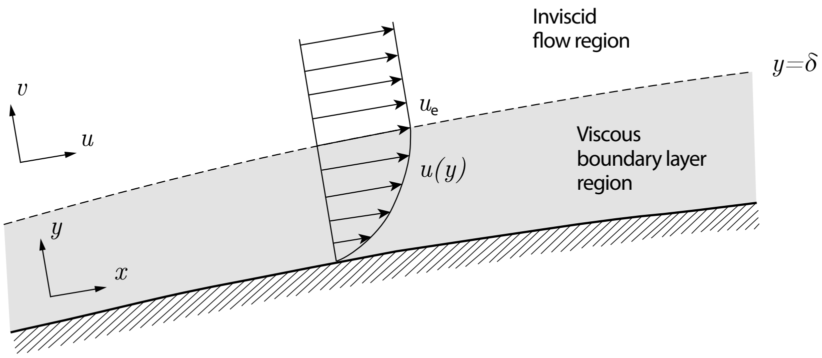


Figure 5.1 Two-dimensional momentum boundary layer.

5.1.1 Prandtl Equations

The Prandtl equations can be easily derived from the two-dimensional, incompressible, steady Navier-Stokes equations through an order-of-magnitude analysis. By assuming that the boundary-layer thickness is small relative to all other characteristic flow dimensions, the following conditions hold within the boundary layer [158]:

$$u \gg v \tag{5.3}$$

$$\frac{\partial u}{\partial y} \gg \frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \tag{5.4}$$

where, u and v are the x- and y-components of the fluid velocity in the curvilinear coordinate system, used for convenience (see Fig. 5.1). In these coordinates, x is the distance measured along the body surface from the forward stagnation point, and y is the normal distance from the body. The coordinates (x, y) are not Cartesian unless the body surface is flat, but they behave as such for all practical purposes if y is small compared to the radius of curvature R of the surface, which is assumed to be of the order of the body's characteristic length.

One key result of performing the order-of-magnitude analysis on the y-momentum equation is that the pressure remains constant in the normal direction to the wall. As a result, its value is the same as the pressure outside the boundary layer, typically given by the inviscid flow solution p_e. This leads to the following form of the Prandtl equations:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{5.5}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p_\mathrm{e}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial y}, \tag{5.6}$$

where ρ is the density and τ_{xy} represents the shear stress in the xy-plane. For laminar flows, it is defined as $\tau_{xy} = \mu \frac{\partial u}{\partial y}$, while for turbulent flows, it is given by $\tau_{xy} = \mu \frac{\partial u}{\partial y} - \rho \langle u'v' \rangle$. The term $\langle u'v' \rangle$ represents the Reynolds stress, which quantifies the turbulent momentum flux due to the correlation between the velocity fluctuations in the x and y directions.

Similarly, to characterize the temperature distribution, T, within the boundary layer, the thermal energy equation can be simplified under the boundary layer assumptions (see, e.g., [158]). This yields:

$$\rho c_{\mathrm{p}} \left(u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \right) = -\frac{\partial \phi}{\partial y} + \Phi_{\nu} \tag{5.7}$$

where c_{p} is the isobaric specific heat capacity, ϕ is the heat flux, and Φ_{ν} is the viscous dissipation function. Analogously to the shear stress, the heat flux is defined generally as

$$\phi = -k \frac{\partial T}{\partial y} + \rho c_p \langle T'v' \rangle \tag{5.8}$$

with k representing the thermal conductivity. The second term, $\rho c_p \langle T'v' \rangle$, accounts for turbulent heat transport and therefore is absent in laminar flow regimes.

Equations 5.5, 5.6, and 5.7 define the Prandtl boundary-layer equations, which are valid for steady and incompressible flows. Here, as ρ and μ are assumed constant, the dynamic and thermal equations can be decoupled, thereby simplifying their solution. Adopting a more general formulation broadens the range of problems that can be addressed; however, it sacrifices the ability to solve each equation independently.

In certain cases, analytical simplifications can be introduced, enabling simple solutions of the laminar boundary-layer equations. For instance, in specific flow configurations, self-similar solutions can be derived. Notable examples include the Blasius solution (1908) [25], which describes the steady flow over a flat plate at constant velocity and the Falkner-Skan solutions (1931) [65] for flows with pressure gradients. Similarly, Eckert (1942) [58] extended this approach to thermal boundary layers, deriving a solution for heat transfer over a flat plate. A more detail explanation can be found in [77]. These similarity solutions offer valuable insights and serve as benchmarks for validating more complex numerical methods, such as integral boundary layer solvers, which are employed in cases where analytical solutions are not possible.

5.1.2 Integral Methods for the Dynamic Boundary Layer

Integral boundary layer methods, initially developed by von Kármán [83] and Pohlhausen [134], are a practical and efficient option due to their simplicity and ease of implementation, when compared to differential methods [78, 79]. Integral methods are based on the idea that it is not necessary to know the detailed velocity profile within the boundary layer to calculate key parameters such as the skin friction coefficient. Instead, they solve the integral form of the governing equations, derived by integrating the Prandtl equations in the direction normal to the wall across the entire boundary layer thickness.

The most commonly used integral equation is the von Kármán equation, obtained by integrating Eq.5.6 in the y direction (see, e.g., [77]). This equation allows the definition of unambiguous boundary-layer thickness: the displacement thickness δ_1 and the momentum thickness δ_2, typically referred to as θ:

$$\delta_1 = \int_0^\infty \left(1 - \frac{u}{u_\mathrm{e}}\right) \mathrm{d}y \tag{5.9}$$

$$\theta = \int_0^\infty \frac{u}{u_\mathrm{e}} \left(1 - \frac{u}{u_\mathrm{e}}\right) \mathrm{d}y \tag{5.10}$$

The displacement thickness δ_1 quantifies the effective displacement of the outer streamlines due to the presence of the boundary layer, while the momentum thickness θ measures the momentum decrement caused by the boundary layer. For an incompressible flow, with no flow through the wall, the von Kármán equation reads:

$$\frac{C_\mathrm{f}}{2} = \frac{d\theta}{dx} + \theta \left(2 + \frac{\delta_1}{\theta}\right) \frac{1}{u_\mathrm{e}} \frac{du_\mathrm{e}}{dx} \tag{5.11}$$

where $C_\mathrm{f}/2 = \tau_w/(\rho_\mathrm{e} u_\mathrm{e}^2)$ is the local skin-friction coefficient, $\tau_\mathrm{w} = \mu(\frac{\partial u}{\partial y})_\mathrm{w}$ is the wall shear stress, and the pressure gradient term has been expressed in terms of the outer velocity, $-\frac{1}{\rho}\frac{dp_\mathrm{e}}{dx} = u_\mathrm{e}\frac{du_\mathrm{e}}{dx}$. The ratio $\frac{\delta_1}{\theta}$ is usually referred to as the shape factor H.

Other integral equations, such as the kinetic energy integral equation, are found in the literature [13, 145].

Whether using the von Kármán equation or multiple integral equations, the system always contains more unknowns than equations. Consequently, all integral methods rely on one or more additional relations, known as closure relations, to solve the system of equations. These closure relations can be classified into three categories depending on whether they are based on self-similar solutions, empirical correlations, or assumed velocity profiles. Moreover, their formulation varies depending on the nature of the flow, whether laminar or turbulent.

The following sections provide a concise overview of two integral methods for solving the viscous boundary layer in laminar and turbulent regimes. The first is the well-known Thwaites integral method [173], designed for laminar boundary layers, while the second is a simplified integral approach proposed by Kays and Crawford [77] for the turbulent boundary layer.

These classical methods are widely used in ice accretion codes, such as IGLOO2D [176] and LEWICE [152]. A comprehensive review of other models employing one or two integral equations with various closure relations is available in Bayeux's thesis [13].

5.1.2.1 Thwaites Method for the Laminar Boundary Layer

Thwaites method [173] is an integral approach for solving the laminar boundary layer by simplifying the von Kármán equation through empirical assumptions. The method introduces the dimensionless parameter

$$T = \frac{\theta}{\delta_4} \tag{5.12}$$

where $\delta_4 = \mu u_\mathrm{e}/\tau_\mathrm{w}$ represents the shear thickness of the boundary layer. By substituting this definition into the von Kármán equation (Eq. 5.11), the momentum integral equation can be rewritten as

$$\frac{u_\mathrm{e}}{\nu}\frac{d\left(\theta^2\right)}{dx} = 2\left[T - (H+2)\,\lambda\right], \tag{5.13}$$

where λ is defined as

$$\lambda = \frac{\theta^2}{\nu}\frac{du_\mathrm{e}}{dx}. \tag{5.14}$$

Thwaites method is based on the assumption that both T and the shape factor H depend exclusively on λ. To further simplify the equation, Thwaites introduced a linear approximation for the right-hand side of Eq. 5.13:

$$\frac{u_\mathrm{e}}{\nu}\frac{d\left(\theta^2\right)}{dx} = a - b\lambda \tag{5.15}$$

where the coefficients proposed by Thwaites are $a = 0.44$ and $b = 5.68$. This formulation allows Eq. 5.15 to be integrated analytically, yielding an expression for the momentum thickness evolution within the boundary layer:

$$\theta(s) = \frac{0.664}{u_\mathrm{e}(s)^{2.84}}\sqrt{\int_0^s \nu_\mathrm{e}\left(s'\right) u_\mathrm{e}\left(s'\right)^{4.68}\,\mathrm{d}s'}. \tag{5.16}$$

Once $\theta(x)$ is computed for a given external velocity u_e, $\lambda(x)$ can be determined from Eq. 5.14. The remaining boundary layer properties can then be derived from λ, if $T(\lambda)$ and $H(\lambda)$ are known. Thwaites [173] numerically defines these functions by matching their values to the Blasius solution for a flat plate, presenting the results in tabular form. Alternatively, Cebeci and Cousteix [41] provide analytical expressions for these functions, derived from various exact solutions.

5.1.2.2 Simplified Integral Method for the Turbulent Boundary Layer

The simplified integral method proposed by Kays and Crawford [77] solves the von Kármán equation by assuming a predefined velocity profile within the turbulent boundary layer and considering a constant free-stream velocity. While this assumption limits its applicability to turbulent flow over a flat plate, it has been shown to provide reliable results even for accelerating or mildly decelerating flows.

The method is based on a power-law representation of the velocity profile within the turbulent boundary layer, given by:

$$u^+ = 8.75 y^{1/7}, \tag{5.17}$$

where the dimensionless velocity and wall-normal coordinate are defined as:

$$u^+ = \frac{u}{u_\tau}, \quad y^+ = \frac{y u_\tau}{\nu}, \quad \text{with} \quad u_\tau = \sqrt{\frac{\tau_\mathrm{w}}{\rho}}. \tag{5.18}$$

This profile, which has been validated for $y^+ < 1500$ and zero pressure gradient, allows the velocity distribution to be expressed as a function of the wall-normal coordinate y and the wall shear stress τ_w:

$$\frac{u}{\sqrt{\tau_\mathrm{w}/\rho}} = 8.75 \left(\frac{y\sqrt{\tau_\mathrm{w}/\rho}}{\nu} \right)^{1/7} \tag{5.19}$$

At the edge of the boundary layer ($y = \delta$), the velocity u equals the external flow velocity u_e. Substituting this condition into Eq. 5.19 yields an expression for the wall shear stress τ_w:

$$\tau_\mathrm{w} = 0.0225 \rho u_\mathrm{e}^2 \left(\frac{\delta u_\mathrm{e}}{\nu} \right)^{-1/4} \tag{5.20}$$

From this, the local skin-friction coefficient can be derived as:

$$\frac{C_\mathrm{f}}{2} = \frac{\tau_\mathrm{w}}{\rho u_\mathrm{e}^2} = 0.0125 \mathrm{Re}_\theta^{-1/4}, \tag{5.21}$$

where $\mathrm{Re}_\theta = \frac{u_\mathrm{e}\theta}{\nu}$ is the Reynolds number based on the local momentum thickness θ. To obtain this expression of the local skin-friction coefficient, the relation $\theta/\delta = 0.097$ has been used [77], derived by substituting Eq. 5.17 into Eq. 5.10.

Once τ_w is known, the velocity profile $u(y)$ is fully defined within the boundary layer. The displacement thickness δ_1 and the momentum thickness θ can then be computed as functions of the boundary layer thickness δ by evaluating the integrals in Eqs. 5.9 and 5.10:

$$\delta_1 = 0.125\delta \quad \text{and} \quad \theta = 0.097\delta \tag{5.22}$$

These results can be substituted in the von Kármán equation (Eq. 5.11), yielding an ordinary differential equation with θ as the only unknown:

$$\frac{d\theta}{dx} + 3.29\frac{\theta}{u_\mathrm{e}}\frac{du_\mathrm{e}}{dx} = 0.0125\left(\frac{u_\mathrm{e}\theta}{\nu}\right)^{-1/4} \tag{5.23}$$

This equation can be further rearranged into a form that is more convenient for integration:

$$d\left(\theta^{2.5}u_\mathrm{e}^{4.11}\right) = 0.0156u_\mathrm{e}^{3.86}\nu^{0.25}dx, \tag{5.24}$$

which yields the following expression for the momentum thickness:

$$\theta = \theta_\mathrm{tr} + \left(\frac{0.016}{u_\mathrm{e}^{4.11}}\int_{s_\mathrm{tr}}^{s}\nu_\mathrm{e}^{0.25}u_\mathrm{e}^{3.86}\,ds'\right)^{0.8} \tag{5.25}$$

where s_tr is the laminar-turbulent transition point, and θ_tr is the corresponding momentum thickness. However, many applications assume that the boundary layer is fully turbulent, i.e., integrating from $s = 0$. This assumption introduces minimal errors when the region of interest is sufficiently far from the stagnation point.

The remaining boundary layer properties, such as $C_\mathrm{f}/2$ or δ, can then be derived from $\theta(x)$.

5.1.3 Integral Methods for the Thermal Boundary Layer

Integral methods for the thermal boundary layer follow a similar approach to those used for the dynamic boundary layer. They rely on integrating the energy equation (Eq. 5.7) in the wall-normal direction and introducing closure relations to solve the resulting system of equations. In this context, the variable of primary interest is typically the Stanton number St, which characterizes the convective heat transfer at the wall:

$$St = \frac{h_\mathrm{tc}}{\rho_\mathrm{e}c_\mathrm{p}u_\mathrm{e}}, \tag{5.26}$$

The simplest method to calculate the Stanton number is the Reynolds analogy. This analogy is derived from the study of incompressible flow over a flat plate with no pressure gradient. When the Prandtl number is equal to unity, the non-dimensional momentum equation becomes equivalent to the non-dimensional thermal equation, with the same boundary conditions. This implies that the Stanton number and the friction coefficient become equivalent:

$$St = \frac{C_f}{2}. \tag{5.27}$$

Additionally, the resolution of the thermal boundary layer can be slightly modified to account for other Prandtl numbers, leading to the following expression:

$$St = \frac{C_\mathrm{f}}{2} Pr^{-2/3}. \tag{5.28}$$

While this analogy is applicable to both laminar and turbulent boundary layers, its assumptions are highly restrictive. As a result, more accurate models are often employed, especially in ice accretion applications. For laminar flows, the method proposed by Smith and Spalding [162] is widely used. For turbulent flows, most methods account for surface roughness, making models for smooth thermal boundary layers less common in practice. In this section, we focus on the Smith and Spalding method for laminar flows, while the turbulent thermal boundary layer is addressed in Sec. 5.1.4, where a variation of the Reynolds analogy accounting for rough walls is presented.

5.1.3.1 Smith and Spalding Method

In ice accretion codes, the most commonly used approach to determine the heat transfer coefficient in the laminar region is the method proposed by Smith and Spalding [162]. This method gives a solution for the conduction thickness $\Delta_\mathrm{cond} = \frac{k}{h_\mathrm{tc}}$ based on similarity solutions for Falkner-Skan flows. This allows to obtain a heat transfer solution that only depends on the external velocity, meaning that the explicit solution of the dynamic boundary layer is not required. For the air ($\mathrm{Pr} = 0.7$), the Stanton number is given by:

$$St = \frac{0.2926\nu_\mathrm{e}}{u_\mathrm{e}\,\mathrm{Pr}} \sqrt{u_\mathrm{e}^{2.87} / \int_0^s \nu_\mathrm{e} u_\mathrm{e}^{1.87} \mathrm{d}s'} \tag{5.29}$$

This expression, however, exhibits a singularity at the stagnation point. As a consequence, stagnation point corrections are required to ensure a robust solution.

5.1.4 Roughness effects in boundary layers

Wall roughness primarily affects the near-wall region of the turbulent boundary layer, which leads to significant changes in wall-related quantities, such as the skin friction and the heat transfer coefficients, which depend on the gradients of velocity and temperature at the wall. Additionally, the wall roughness also affects the laminar-turbulent transition, promoting the development of a turbulent boundary layer.

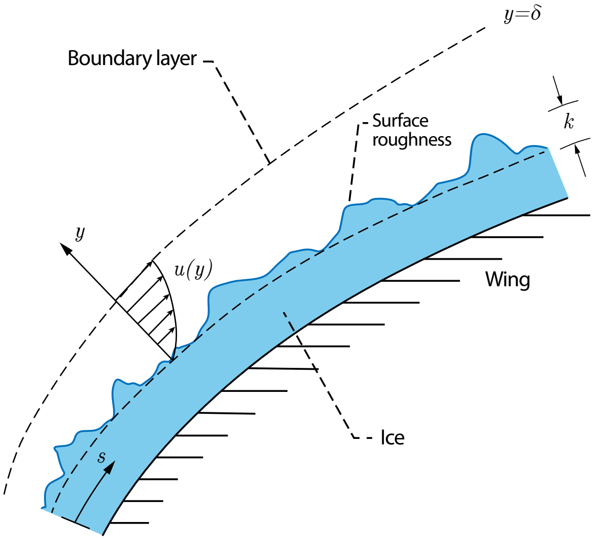


Figure 5.2 Two-dimensional momentum boundary layer over a rough iced surface.

In ice accretion modeling, it is essential to consider the effects of surface roughness for two primary reasons. First, ice accretion naturally leads to the formation of rough surfaces. Second, the heat transfer coefficient, which is strongly affected by surface roughness, plays a critical role in accurately predicting the ice shape.

Despite its importance, the surface roughness induced by ice remains poorly understood. Experimental data on ice surface roughness are scarce, and most computational models rely on the equivalent sand-grain roughness as a global parameter. In this approach, the roughness height k is treated as a constant, typically denoted as k_s. However, this global parameter fails to capture the inherently non-homogeneous nature of ice-shape roughness, which has led to the development of local roughness models. Yet, such models are seldom employed in ice accretion simulations, as demonstrated by the 1^{st} IPW[86], where only two out of 24 participants incorporated a local roughness model.

The determination of the equivalent sand-grain is, however, not straightforward [74]. Some ice accretion codes employ different definition of k_s and may vary across the studied case. For instance, a common practice in ONERA's codes [142, 176] is to define the equivalent sand-grain roughness height as:

$$k_\mathrm{s} = c/1000, \tag{5.30}$$

where c is the chord of the 2D airfoil. For 3D wings, although using c as a reference length is less straightforward, the numerical experiment have shown satisfactory results when applied to constant-chord swept wings [142].

5.1.4.1 Laminar-turbulent transition on rough walls

Surface roughness affects the laminar-turbulent transition by promoting earlier transition due to increased disturbances in the boundary layer. Even small roughness elements can generate localized flow instabilities, leading to earlier breakdown of laminar flow. In icing applications [101, 141, 152, 176], the laminar-turbulent transition criterion is usually based on the roughness Reynolds number [183]:

$$Re_{\mathrm{k}}(x_{\mathrm{tr}}) = \frac{u_{\mathrm{k}} k}{\nu_{\mathrm{e}}} = 600. \tag{5.31}$$

The flow is considered laminar at lower values of the roughness Reynolds number and turbulent at higher values. In Eq. 5.31, u_{k} represents velocity at the height of the roughness elements k . While k is usually replaced by the equivalent sand-grain height k_{s}, different velocities are used to define the roughness Reynolds number. IGLOO2D [176] uses the velocity at the outer edge of the boundary layer, $u_{\mathrm{k}} = u_{\mathrm{e}}$, whereas LEWICE3D [152] and Makkonen [101] propose modeling the velocity profile using a fourth-order polynomial, allowing u_{k} to be computed as $u_{\mathrm{k}} = u(k_{\mathrm{s}})$. Despite the difference in the definition of the roughness Reynolds number, both codes have shown similar transition points.

5.1.4.2 Skin Friction Coefficient on Rough Walls

When the boundary layer has been found turbulent, i.e. $Re_{\mathrm{k}} > 600$, the wall behaves as a *fully rough* surface [77]. A key feature of such surfaces is that the skin friction coefficient becomes independent of the Reynolds number, indicating that viscous effects are no longer significant. Instead, the wall shear stress is primarily governed by pressure drag acting on individual roughness elements. Following a similar approach to that described in Section 5.1.2.2, Kays and Crawford [77] proposed the following expression of the skin friction coefficient on fully rough surfaces:

$$\frac{C_{\mathrm{f,\,rough}}}{2} = \frac{0.168}{\ln^2\left(864\theta/k_s\right)} \tag{5.32}$$

This expression is commonly used in ice accretion models based on integral boundary layer solvers [101, 176]; however, a slightly more accurate result can be obtained by avoiding certain simplifications, which yields:

$$\frac{C_{\text{f, rough}}}{2} = \frac{0.168}{\ln^2\left(864\theta/k_s + 2.568\right)} \tag{5.33}$$

This alternative expression is also employed in several icing simulation codes [141, 152].

Whether using Eq. 5.32 or 5.33 to estimate the skin-friction coefficient, the determination of the turbulent momentum boundary layer thickness over a rough surface is nontrivial. As a result, it is a common practice to employ the value computed for smooth-wall conditions [101, 152, 176]. However, the validity of this assumption remains debatable.

5.1.4.3 Heat Transfer Coefficient on Rough Walls

A widely used approach to estimate the heat transfer coefficient in turbulent boundary layers is based on a modified Reynolds analogy that accounts for surface roughness. The formulation, proposed by Kays and Crawford [77], is given by:

$$\mathrm{St} = \frac{C_{\text{f, rough}}}{\mathrm{Pr_t} + \sqrt{C_{\text{f, rough}}}/\mathrm{St_k}} \tag{5.34}$$

where $\mathrm{Pr_t} = 0.9$ is the turbulent Prandtl number of air and $\mathrm{St_k}$ is the roughness Stanton number, given by:

$$\mathrm{St_k} = C\left(k_\mathrm{s}^+\right)^{-\alpha}\mathrm{Pr}^{-\beta} \tag{5.35}$$

where k_s^+ is the equivalent sand-grain roughness height k_s made non-dimensional in wall coordinates (see Eq 5.18). Furthermore, C, α and β are constants that depend on the type of roughness and are typically determined from experimental data. Commonly used values in ice accretion models [101, 141, 176] are:

$$C = 1.92 \quad \alpha = 0.45 \quad \beta = 0.8 \tag{5.36}$$

5.2 3D Simplified Integral Boundary Layer Solver

The 3D simplified integral boundary layer method described here builds upon the physical 2D models outlined in the previous section. The method incorporates laminar-turbulent transition based on the local roughness Reynolds number. In the laminar regime, the heat transfer coefficient is calculated using the Smith and Spalding formula. In the turbulent regime, the modified Reynolds analogy is used to estimate heat transfer on rough surfaces. For this, the momentum thickness is evaluated across the entire surface, using the Thwaites' method in the laminar region and the simplified approach by Kays and Crawford in the turbulent region. The overall procedure is summarized in Algorithm 1, which presents the main steps of the 3D simplified integral method. This approach is implemented in the HTC3D module of the IGLOO3D framework.

Algorithm 1 3D Simplified Integral Method

```
Read data from aerodynamic solution
Compute recovery temperature
Eq. 5.2
Identify the local flow regime
Eq. 5.31
Propagation of flow regime
Compute streamline integrals
Eqs. 5.16, 5.25 and 5.29
Compute the laminar momentum thickness
Eq. 5.16
Propagate the laminar momentum thickness from the laminar-turbulent front
Compute the turbulent momentum thickness
Eq. 5.25
Compute the heat transfer coefficient according to the local flow regime
Eq. 5.29 or Eq. 5.33
```

While the original models are formulated for two-dimensional boundary layers, the present approach extends their applicability to three-dimensional geometries by applying the method along surface streamlines, under the assumption that the boundary layer remains locally two-dimensional in this direction. This approximation circumvents the complexity associated with modeling cross-flow inside the boundary layer (see Figure 5.3). It is important to note that cross-flow outside the boundary layer is accounted for through the external velocity field; only the cross-flow inside the boundary layer is neglected in this model. For slightly swept wings, this procedure is generally adequate; however, it may lose accuracy in the presence of strong pressure gradients normal to the streamline direction, as typically encountered in turbomachinery or on helicopter blades, which are beyond the scope of the present study [99]. In addition, the current approach assumes a uniform wall temperature, which limits its applicability to cases without heated surfaces.

5.2.1 Numerical Method

From a numerical perspective, the main challenge stems from the fact that the face centers of the surface do not align with the streamline direction. This contrasts with two-dimensional simulations, where all nodes lie on a single plane. As a result, computing quantities along

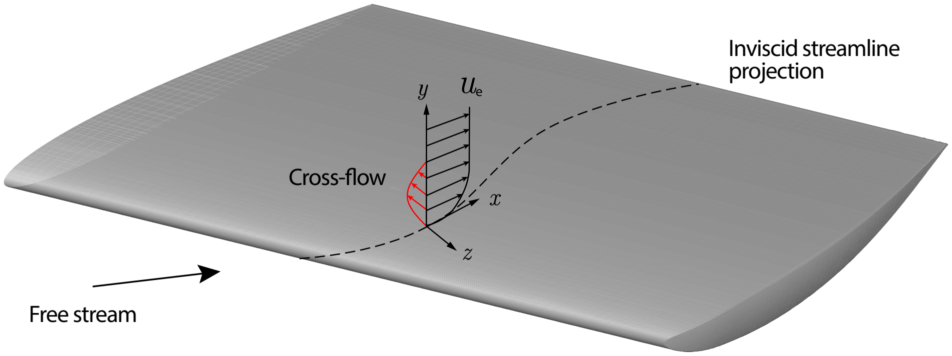


Figure 5.3 Schematic of a three-dimensional boundary layer on a generic swept wing.

inviscid wall streamlines becomes more complex, particularly in the evaluation of curvilinear integrals and in the propagation of the momentum thickness from the laminar–turbulent transition front. The procedures adopted to perform these calculations are described below.

5.2.1.1 Computation of Integrals Along Streamlines

The numerical method presented in [141], which is used to compute integrals along streamlines, is outlined below. In a general form, the integrals of Eqs. 5.16, 5.25 and 5.29 can be written, as:

$$\psi = \int_0^s \phi(s')\,\mathrm{d}s' = \int_0^s \mathrm{d}\psi' \tag{5.37}$$

where s is the local value of the curvilinear abscissa along the streamline and ϕ is the function to be integrated. Note that for $\phi(s) = 1$, the value of the curvilinear abscissa is obtained ($\psi = \int_0^s \mathrm{d}s'$).

Moving along the streamline, the time step is given by

$$\mathrm{d}t = \kappa \mathrm{d}\psi \tag{5.38}$$

where κ is a unit scalar such that its units are compatible with the pseudo time step. Equation 5.38 can be expressed as $\mathrm{d}t = \kappa\phi\mathrm{d}s$, allowing the following definition of a local propagation velocity:

$$\vec{V}_\psi = \frac{\mathrm{d}s}{\mathrm{d}t}\vec{e}_\mathrm{s} = \frac{1}{\kappa\phi(s)}\vec{e}_\mathrm{s} \tag{5.39}$$

where $\vec{e}_\mathrm{s} = \frac{\vec{u}_\mathrm{e}}{||\vec{u}_\mathrm{e}||}$, consistently with the fact that integrals are computed along streamlines.

Moreover, defining $\kappa = 1$, Eq. 5.38 implies:

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = 1 = \frac{\partial \psi}{\partial t} + \vec{V}_\psi \cdot \vec{\nabla}\psi \tag{5.40}$$

which can be expressed in a conservative form:

$$\frac{\partial \psi}{\partial t} + \vec{\nabla} \cdot \left(\psi \vec{V}_\psi\right) = 1 + \psi \vec{\nabla} \cdot \vec{V}_\psi \tag{5.41}$$

obtaining a hyperbolic PDE that can be solved with a FVM. The methodology presented in [141] is followed, which employs a basic upwind flux scheme based on the orientation of the local velocity $\vec{V}_\psi$, and uses the Gauss-Seidel algorithm to achieve the steady-state solution.

Specific treatment is made in the vicinity of the stagnation point. Three conditions are tested to check if a face i is a stagnation point:

- For all the computed edges of face i, the flux is outgoing from face i.
- For at least one edge of face i, the flux is outgoing on both sides of the edge.
- For one edge of face i a boundary condition is detected.

Figure 5.4 illustrates the stagnation point detection on a discretized 30º swept NACA 0012 wing, where two faces meet the required conditions. It should be noted that the exact location of the stagnation point is unknown. As a result, all integral calculations start from one of the center of the faces identified as stagnation points. This introduces an error of order $\mathcal{O}(\Delta x)$ in the local value of the curvilinear abscissa, which may also affect the integral calculations.

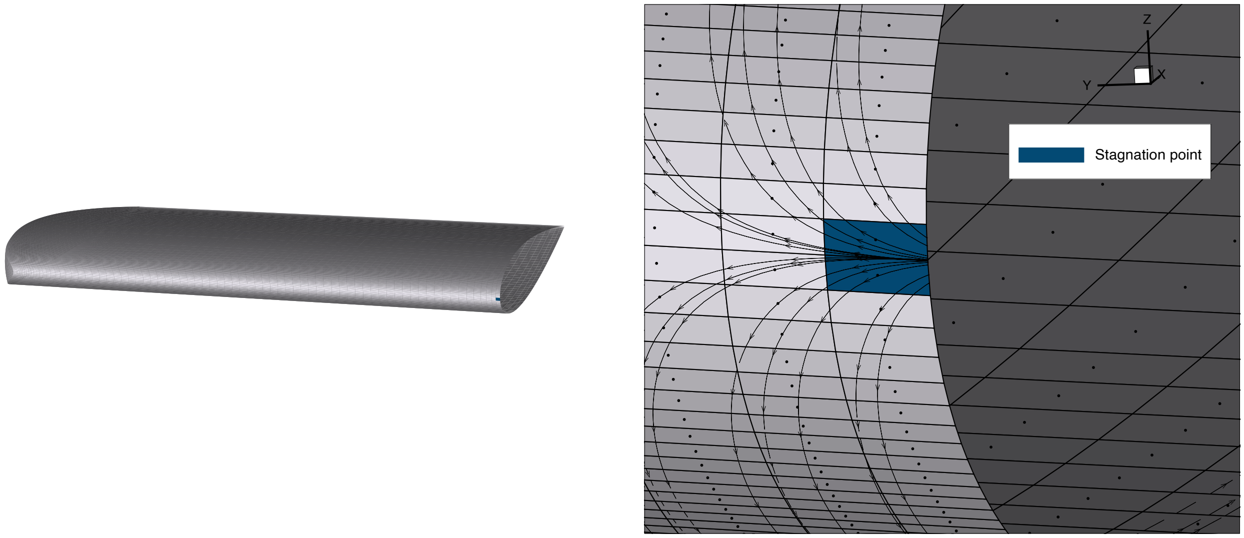


Figure 5.4 Illustration of the stagnation point detection and the velocity streamtraces for a swept wing.

5.2.1.2 Stagnation Point Correction

Integral boundary layer methods often require stagnation point corrections. Whether in 2D solvers, where the surface is represented by nodes, or in 3D solvers, where the surface is represented by faces, the node or face corresponding to the stagnation point defines the starting point of the integration. As a result, singularities arise at these locations when numerically solving Eq. 5.29, and an alternative approach is required to estimate the heat transfer coefficient.

For instance, in the model of Makkonen [101] the Stanton number is assumed to be

$$\mathrm{St} = \mathrm{Re}^{-1/2}\mathrm{Pr}^{-1}, \tag{5.42}$$

an expression that has also been adopted in LEWICE [152]. Alternatively, the present method assumes a linear velocity profile in the vicinity of the stagnation point:

$$u_{\mathrm{e}}(s) = U_0 s, \tag{5.43}$$

where U_0 is a constant determined from the neighborhood of the face. This assumption allows for an analytical solution of Eq. 5.29:

$$h_{\mathrm{tc}} = \rho_{\mathrm{e}} c_{\mathrm{p}} \frac{0.2926}{\mathrm{Pr}} \sqrt{2.87 U_0 \nu_{\mathrm{e}}}, \tag{5.44}$$

thus avoiding the singularity at the stagnation point.

Similarly, the momentum boundary layer thickness can be obtained by substituting Eq. 5.43 into Eq. 5.16:

$$\theta = \sqrt{\frac{0.077\nu_{\mathrm{e}}}{U_0}} \tag{5.45}$$

5.2.1.3 Propagation Along the Streamline Direction

To compute the momentum boundary layer thickness in the turbulent zone, each turbulent face must know the value of the momentum thickness in the laminar-turbulent transition θ_{tr}. To achieve this, a procedure similar to that used for the computation of integrals along the streamline is applied. In this case, θ_{tr} remains constant in the streamline direction, and therefore:

$$\frac{\mathrm{d}\theta_{\mathrm{tr}}}{\mathrm{d}t} = 0 = \frac{\partial \theta_{\mathrm{tr}}}{\partial t} + \vec{V}_s \cdot \vec{\nabla}\theta_{\mathrm{tr}} \tag{5.46}$$

where the propagation velocity is $\vec{V}_s = \vec{e}_s$. This equation can also be expressed in conservative form as:

$$\frac{\partial \theta_{\mathrm{tr}}}{\partial t} + \vec{\nabla} \cdot \left(\theta_{\mathrm{tr}} \vec{V}_s \right) = \theta_{\mathrm{tr}} \vec{\nabla} \cdot \vec{V}_s \tag{5.47}$$

which is again computed using the FVM of [141].

Figure 5.5 shows a zoomed-in view of the leading edge near the wing tip of a 30º swept NACA 0012 wing. In the figure, θ_{tr} is obtained at a turbulent face (black circle) by propagating the momentum thickness values from the laminar cells adjacent to the laminar-turbulent front (red circles) in the streamwise direction (streamtrace).

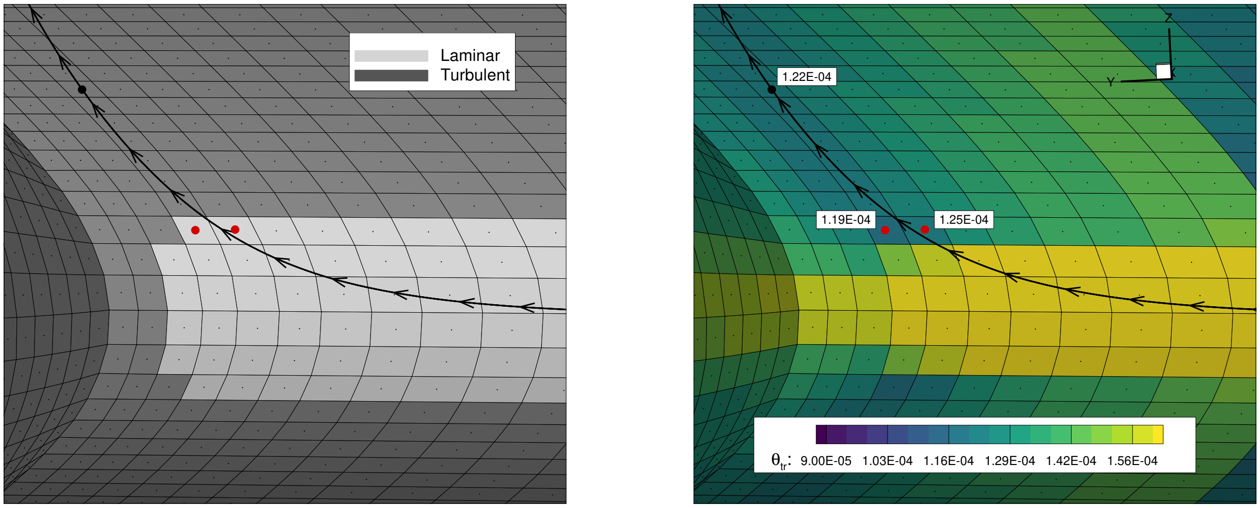
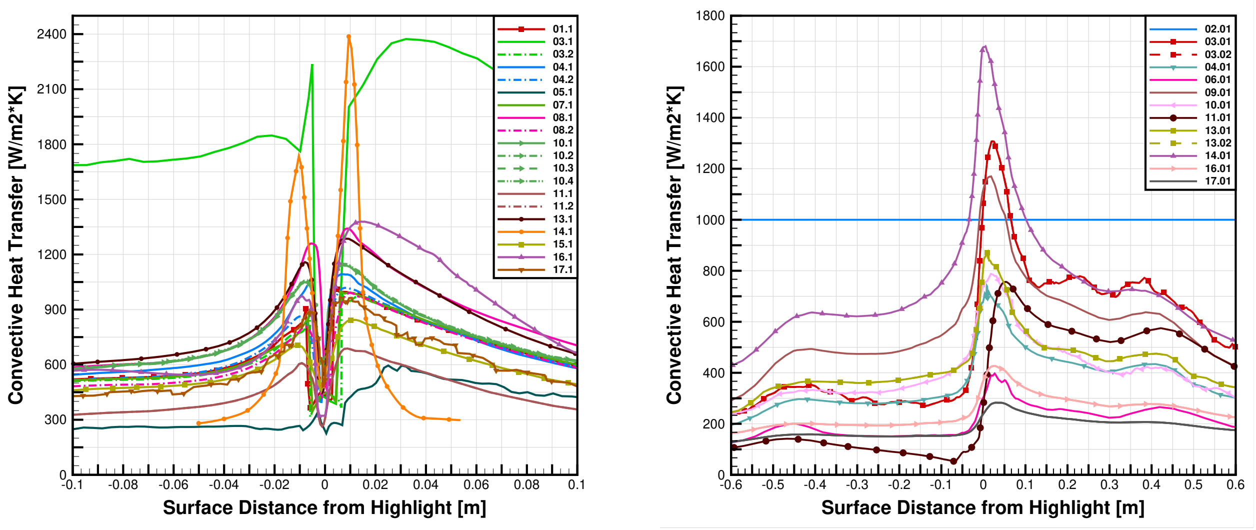


Figure 5.5 Numerical propagation of the momentum thickness of the boundary layer at the laminar-turbulent front.

A similar approach is employed to propagate the flow regime. The laminar–turbulent transition criterion used (Eq. 5.31) allows transitions from turbulent to laminar flow along a streamline, that would be unphysical. To prevent this, the flow regime is instead propagated downstream from the stagnation point, ensuring that each face has access to upstream information. Consequently, if any upstream cell along the streamline has been identified as turbulent, downstream laminar faces are reclassified as turbulent to maintain physical consistency. From a numerical perspective, this is achieved by propagating a flag variable that defines the flow regime at each face.

5.3 Validation Test Cases

The validation of the simplified integral method is focused on assessing its ability to predict the heat transfer coefficient. However, obtaining reliable reference values for this quantity is particularly challenging, even in two-dimensional configurations. In recent ice prediction workshops [84, 86], large discrepancies have been reported among the results of different ice accretion codes for the same test case (see Fig. 5.6). In some instances, the predicted heat transfer coefficients differed by more than a factor of five between participants. These discrepancies have underscored the difficulty of validating heat transfer predictions, which is partly due to the fact that this quantity is rarely measured directly in experiments.



(a) Case 242 from 1st IPW[86]

(b) Case 1.1 from 2nd IPW[84]

Figure 5.6 Convective heat transfer predictions for two cases from ice prediction workshops.

As a result of these difficulties, most boundary layer codes rely on highly simplified cases for the validation of heat transfer predictions. In this context, two validation cases are considered in this section. The first focuses on flow over a flat plate with zero pressure gradient, a classical test that enables well-defined comparisons in both laminar and turbulent regimes. The second involves flow around a rough cylinder, a case commonly used in previous integral boundary layer validations [101, 152]. This configuration introduces more complex flow features and the influence of surface roughness, providing an additional benchmark for evaluating the method's performance under more realistic, challenging conditions. Additionally, Chapter 6 presents ice accretion simulations for some baseline cases of 1st IPW, where the predicted heat transfer is further validated by comparing the results with those of the participants. As a possible future validation case, the flow over a swept cylinder could be considered [40].

5.3.1 Flat Plate Without Pressure Gradient

The boundary layer flow over a flat plate without pressure gradient is one of the most classical and extensively studied cases in the literature. This case is the simplest and represents a fundamental step for validating the simplified integral method developed, as its solution is well established. A flat plate of length L is considered, placed at zero angle of attack in a viscid, incompressible flow. The steady flow is parallel to the x axis with free stream velocity U_∞. In this case, the outer inviscid velocity is constant, that is $dp_\mathrm{e}/dx = 0$. Two different studies will be conducted to analyze the accuracy of the presented method in laminar and turbulent regimes.

The sensitivity of the solution to mesh orientation is also investigated by considering two types of surface meshes. The first configuration is a structured mesh of size $N \times 1$, in which all face centers are aligned with the free-stream velocity direction. This type of meshes is referred to as *aligned mesh* and is shown in Fig. 5.7.

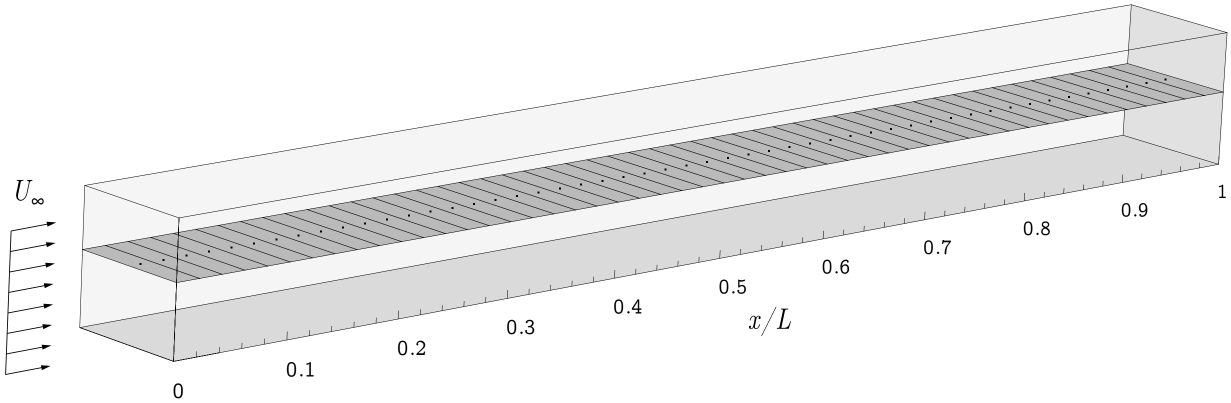


Figure 5.7 Two-dimensional aligned mesh for the flat plate ($N = 50$).

A more challenging surface mesh, referred to as the *rotated mesh*, is also considered. In this case, a square flat plate of dimensions $L/\sqrt{2} \times L/\sqrt{2}$ is discretized using $N \times N$ faces. The mesh is then rotated 45° about the z-axis, as shown in Fig. 5.8. This configuration ensures that the diagonal of the square matches the plate length L, enabling a direct comparison between both mesh orientations.

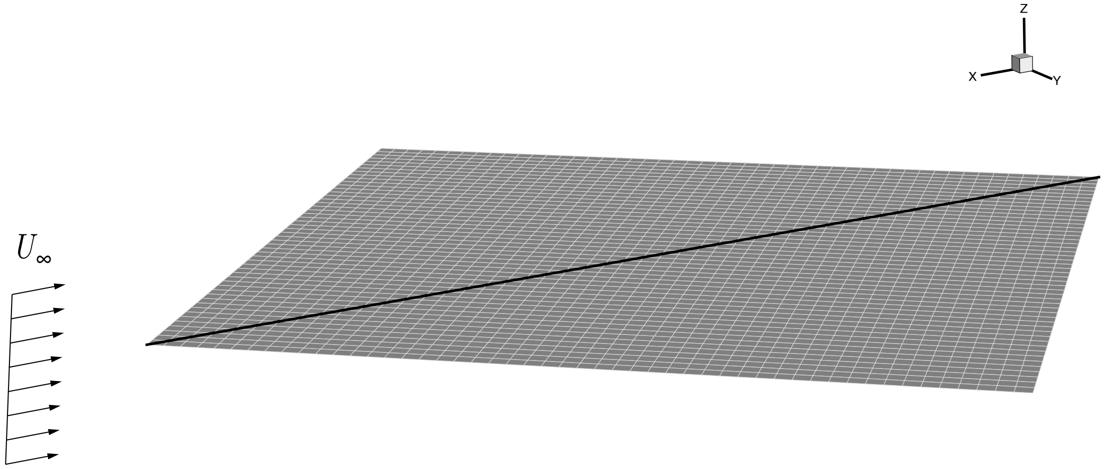


Figure 5.8 Two-dimensional rotated mesh for the flat plate ($N = 50$).

5.3.1.1 Laminar Boundary Layer on a Flat Plate

The solution of the laminar boundary layer on a flat plate is well established in the literature, with the Blasius solution serving as one of the most widely used references. Based on similarity solutions to the Prandtl equations, the evolution of the dynamic boundary layer is given by (see, e.g. [13]):

$$H = 2.591; \quad \frac{\delta_1}{s} = \frac{1.721}{\sqrt{\mathrm{Re}_s}}; \quad \frac{\theta}{s} = \frac{0.664}{\sqrt{\mathrm{Re}_s}}; \quad \frac{C_\mathrm{f}}{2} = \frac{0.332}{\sqrt{\mathrm{Re}_x}} \tag{5.48}$$

Simulation parameters are set at a Reynolds number of $6.4 \cdot 10^4$ for a 1m flat plate length at $p_\infty = 101325\mathrm{Pa}$ and $T_\infty = 303.15\mathrm{K}$. To maintain laminar flow within the boundary layer, the equivalent sand-grain height is selected to ensure the laminar-turbulent transition criterion is not met. Four numerical simulations are performed with increasing number of faces $N = 50, 100, 200, 400$. Figure 5.9 presents the numerical solution obtained with the aligned meshes, compared to the Blasius solution. The agreement between both solutions indicates good convergence. To further assess the robustness of the method, Fig. 5.10 shows a comparison between the solutions obtained with aligned and rotated meshes, both using $N = 100$. The results indicate that the method is not highly sensitive to mesh orientation.

For the laminar thermal boundary layer, the Reynolds analogy (see Eq. 5.28) allows to obtain

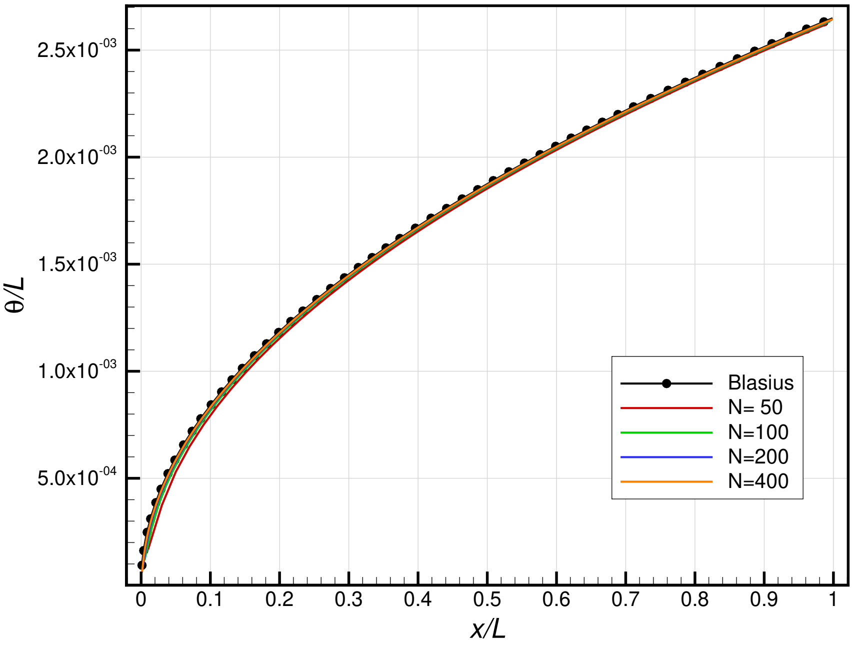


Figure 5.9 Laminar boundary layer on a flat plate. Convergence of the momentum boundary layer thickness with increasing number of faces N using aligned meshes.

an analytical solution, following the same strategy as that used by Blasius for the dynamic boundary layer. The reference heat transfer coefficient can thus be expressed as:

$$h_{\mathrm{tc}} = \rho_{\mathrm{e}} c_{\mathrm{p}} u_{\mathrm{e}} \mathrm{Pr}^{-2/3} \frac{0.332}{\sqrt{\mathrm{Re}_x}} \tag{5.49}$$

Figure 5.11 compares the numerical results with the Reynolds analogy, showing good agreement overall. Minor discrepancies are observed near the stagnation point. Although these differences are small, it is worthwhile to understand their origin. They can be mainly attributed to two factors. First, as discussed in Section 5.2.1.1, there is an error of order $\mathcal{O}(\Delta x)$ in the computation of the curvilinear abscissa. For this case, since the stagnation point is located at $x = 0$, the associated error amounts to $\Delta x/2$, which explains the downstream shift observed in the numerical results. Second, in the present case, the external velocity is constant across all faces, $u_e = U_\infty$, including at the stagnation point. However, the correction presented in Section 5.2.1.2 assumes a linear velocity profile in the vicinity of the stagnation point, i.e., $u_e = U_0 s$, which yields in slightly different values at this face (see Eq. 5.29). Note that in practical applications, the face corresponding to the stagnation point is expected to have zero velocity, meaning that this error will not be present.

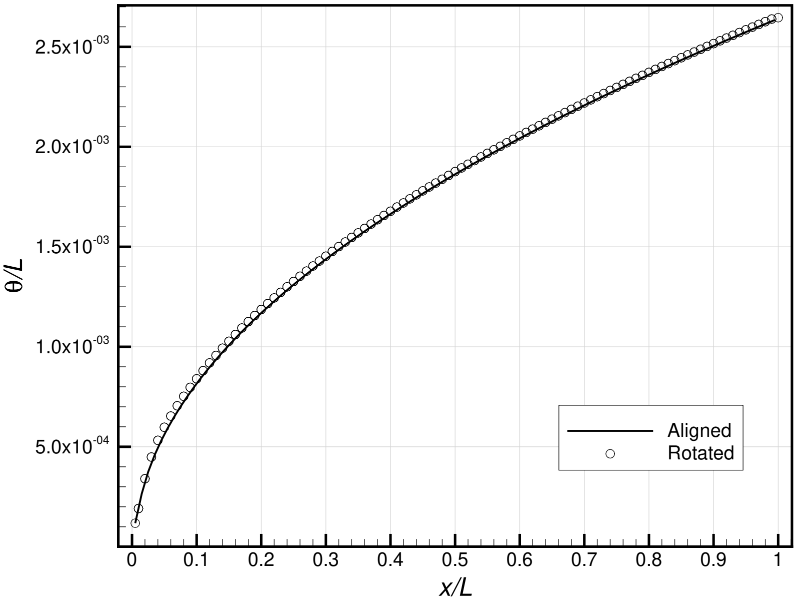


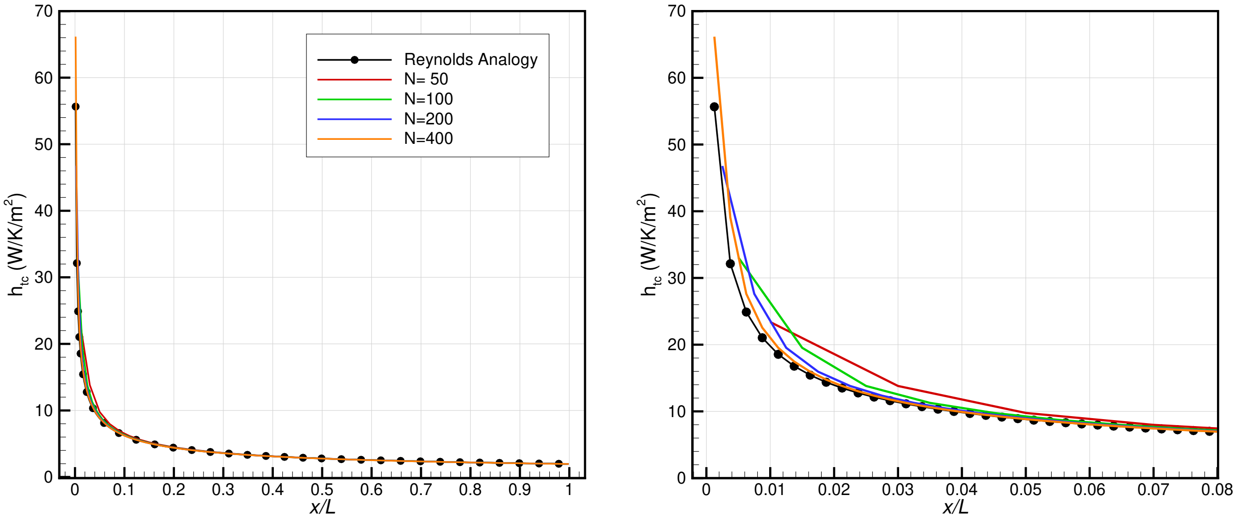
Figure 5.10 Laminar boundary layer on a flat plate. Comparison of the momentum boundary layer thickness obtained with aligned and rotated meshes, both using $N = 100$.

Besides these discrepancies, the numerical solution captures the overall heat transfer distribution with high accuracy. The agreement with the Reynolds analogy confirms the validity of the present approach in laminar flow conditions over a flat plate. Additional results using a rotated mesh are presented in Fig. 5.12, further demonstrating the method's consistency with respect to mesh orientation.

5.3.1.2 Turbulent Boundary Layer on a Flat Plate

In contrast to the laminar case, validating the turbulent boundary layer over a flat plate is more challenging, particularly due to the absence of an analytical solution. Several experiments can be found in the literature that provide measurements of the momentum boundary layer thickness over smooth walls. However, in the case of heat transfer over rough surfaces, experimental data are more scarce. For this validation test, we focus exclusively on validating the boundary layer thickness over a smooth wall. To this end, the experiment conducted by Weighardt and Tillmann [186] is used, as it has also been employed for validation in previous boundary layer solvers [13].

The experiment was conducted at atmospheric pressure, on a flat plate of 5 meters in length, with a constant freestream velocity of 33 m/s and a kinematic viscosity of $1.51 \cdot 10^{-5}$ m^2/s. The input data for this test case are listed in Table 5.1.



(a) Global view

(b) Zoom around $x = 0$

Figure 5.11 Laminar boundary layer on a flat plate. Convergence of the heat transfer coefficient with increasing number of faces N using aligned meshes.

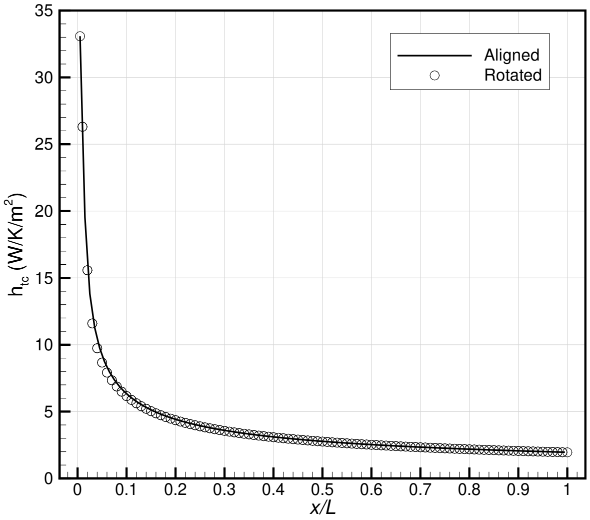


Figure 5.12 Laminar boundary layer on a flat plate. Comparison of the heat transfer coefficient obtained with aligned and rotated meshes, both using $N = 100$.

Again, four simulations are performed with increasing the number of faces using aligned meshes. Figure 5.13 presents the numerical results compared to the values obtained in the experiments conducted by Weighardt and Tillmann. The results are highly satisfactory, even for the coarsest mesh studied, showing a good match with the experimental data. This

M_∞	P_∞ (Pa)	T_∞ (K)	L (m)
0.096	101325	294	5

Table 5.1 Turbulent boundary layer on a flat plate. Simulation parameters.

suggests that the method accurately captures the boundary layer growth, even with relatively low mesh resolution. Further results obtained with the rotated mesh are presented in Fig. 5.14, showing a good agreement with the aligned mesh.

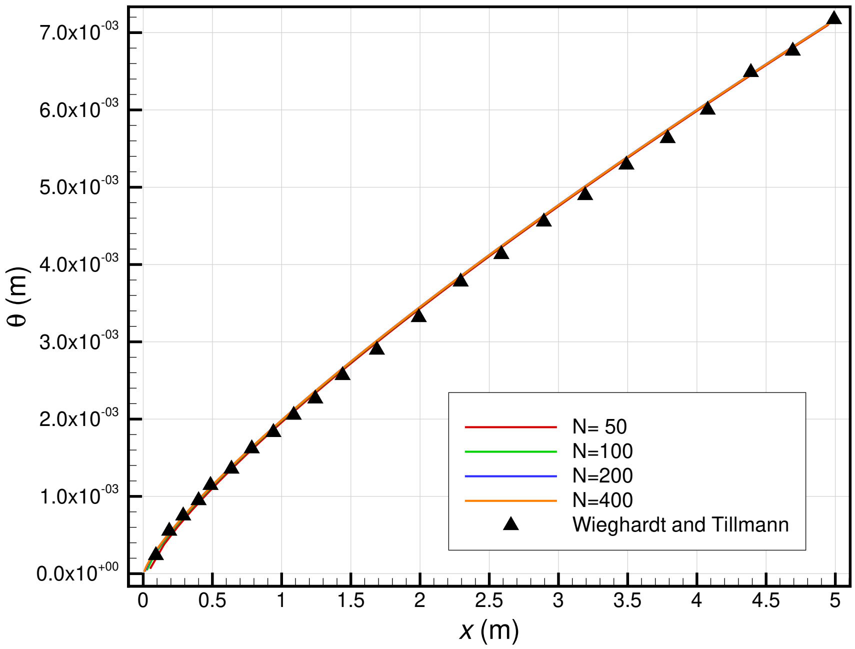


Figure 5.13 Turbulent boundary layer on a flat plate. Convergence of the momentum boundary layer thickness with increasing number of faces N using aligned meshes.

5.3.2 Boundary Layer on a Rough Cylinder

The study of boundary layer behavior on rough surfaces has been a significant topic in fluid dynamics research for several decades. In particular, from the 1970s, efforts to understand how surface roughness influences boundary layer intensified. While most research focused on the dynamic boundary layer, investigating the thermal boundary layer proved to be more challenging, and only a few studies addressed this aspect. A key contribution to this field was Achenbach's work [5], which explored the effect of surface roughness on heat transfer on a cylinder. His experiments provided valuable data that helped establish a clearer understanding of the impact of roughness on heat transfer, which has since been used to validate

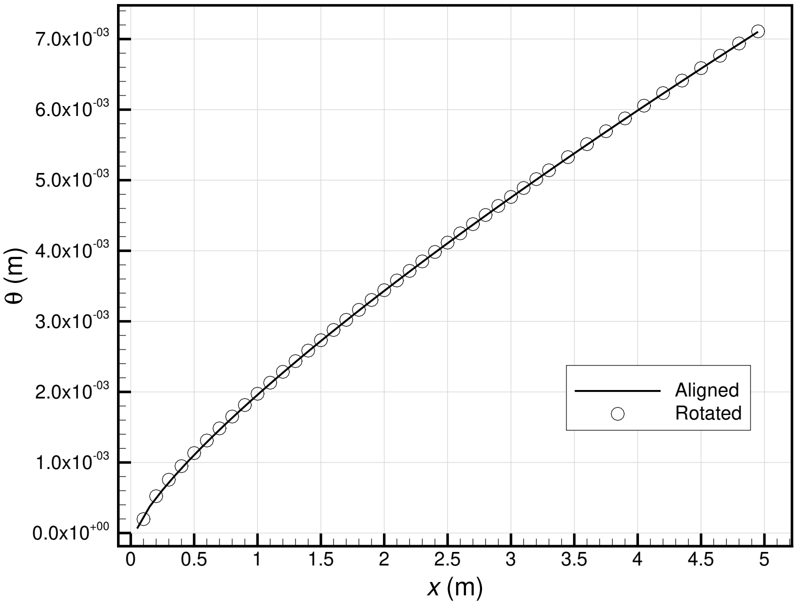


Figure 5.14 Turbulent boundary layer on a flat plate. Comparison of the momentum boundary layer thickness obtained with aligned and rotated meshes, both using $N = 100$.

boundary layer solvers in the context of icing simulations. Examples of such codes include the method developed by Makkonen [101] and LEWICE [152].

The Achenbach experiments were conducted in a high-pressure wind tunnel operating with air at up to 40 bar and constant temperature, which enabled measurements at high Reynolds numbers. Specifically, three Reynolds numbers were tested: $\mathrm{Re} = 4.8 \cdot 10^4$, $2.8 \cdot 10^5$, and $8.8 \cdot 10^5$. The measurements were performed on a 15-cm diameter rough cylinder. The roughness element height was $k = 0.9$ mm, and the equivalent sand grain roughness was $k_s = 1.35$ mm. Results were expressed in terms of the Nusselt number scaled with the square root of the Reynolds number, $\mathrm{Nu}/\sqrt{\mathrm{Re}}$. Note that the Nusselt number can be easily obtained from the Stanton number:

$$\mathrm{Nu}(x) = \mathrm{St}(x)\mathrm{Re}\mathrm{Pr} \tag{5.50}$$

In order to perform the numerical simulation, the solution provided by the potential flow theory introduces significant errors. Instead, slight variations in the pressure and velocity distribution are considered. The following expressions come from empirical fit, as proposed by Makkonen [101]:

$$C_\mathrm{p} = \frac{(p_\mathrm{e} - p_\infty)}{0.5\rho_\mathrm{e}U_\infty^2} = 1 - 2.5\sin^2(1.233\phi(x)) \tag{5.51}$$

$$u_\mathrm{e} = 1.58\sin(1.233\phi(x))U_\infty \tag{5.52}$$

where $\phi(x)$ denotes the angular position of the cylinder, measured from the front stagnation point.

A structured mesh is generated from a quarter-cylinder geometry, discretized into 100 faces. Three simulations were carried out, corresponding to the different Reynolds numbers under study. Figure 5.15 presents the results obtained with the current approach (denoted HTC3D), compared to the experimental measurements of Achenbach and the numerical results from Makkonen and LEWICE. Overall, the simplified integral method implemented in HTC3D yields satisfactory agreement with reference data. At the lowest Reynolds number, both HTC3D and LEWICE predict a transition around $\phi = 50$, whereas no transition is observed in Achenbach's measurements or Makkonen's model. For the other Reynolds numbers, the HTC3D results are in good agreement with those predicted by Makkonen's approach.

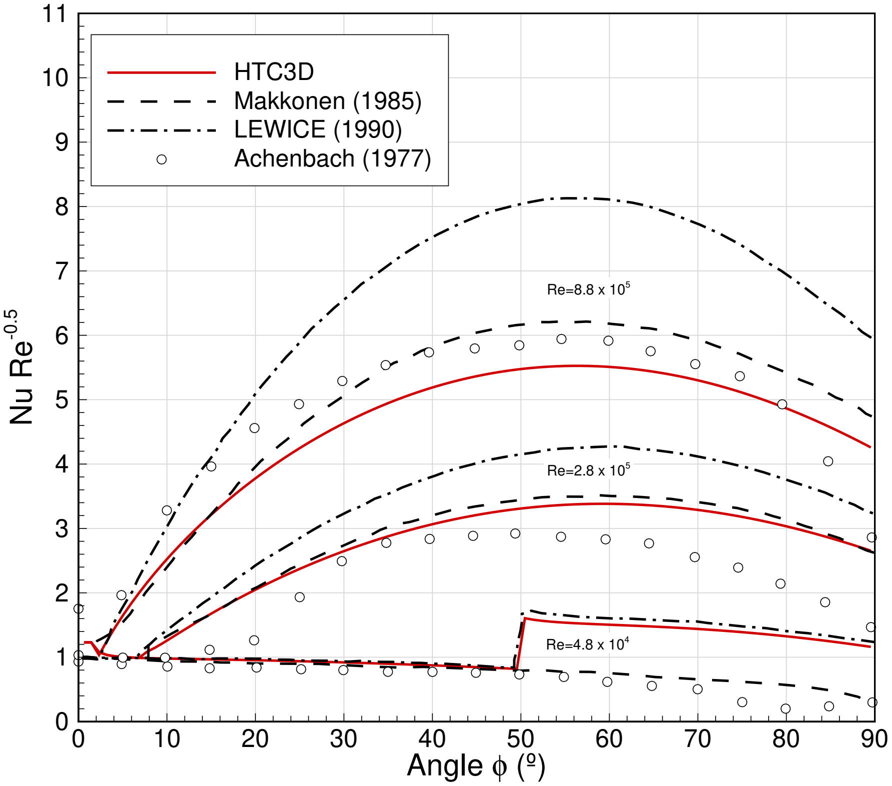


Figure 5.15 Boundary layer on a rough cylinder ($D = 15$ cm, $k_s = 1.35$ mm). Comparison of numerical and experimental local heat transfer $\mathrm{Nu}/\sqrt{\mathrm{Re}}$.

5.4 Conclusions

This chapter has presented a three-dimensional simplified integral boundary layer solver developed specifically for ice accretion simulations. The method enables the computation of diffusive transfer quantities, such as the heat transfer coefficient, from inviscid airflow computations. The approach extends simplified two-dimensional boundary layer methods to three-dimensional surfaces using a FVM, under the assumption that the flow remains two-dimensional in the streamwise direction.

The formulation distinguishes between laminar and turbulent boundary layers and incorporates the effects of surface roughness. A dedicated numerical treatment has also been introduced to handle the stagnation region appropriately.

Validation cases have shown that the solver captures the expected trends in heat transfer distributions when compared to reference solutions. While the method does not resolve the full boundary layer profile, it yields accurate surface estimates that are crucial for icing simulations. Further validation is presented in Chapter 6, where the predicted heat transfer coefficients are compared with results from other codes submitted to recent ice prediction workshops.

CHAPTER 6 ARTICLE 2: IMMERSED BOUNDARY METHODOLOGY FOR 3D MULTI-STEP ICE ACCRETION SIMULATIONS

P. Elices Paz, E. Radenac, G. Blanchard, S. Péron, É. Laurendeau, and P. Villedieu, "Immersed Boundary Methodology for 3D Multi-Step Ice Accretion Simulations," (submitted to *Journal of Aircraft* on 19 June 2025)

6.1 Abstract

In-flight ice accretion poses a significant risk to aircraft safety and performance. Despite advancements in ice protection systems, aircraft must demonstrate the ability to operate safely under icing conditions, highlighting the importance of reliable ice accretion simulations. Traditional multi-step simulations divide the ice accretion process into discrete stages, improving the accuracy of ice predictions. However, this approach increases computational costs and reduces automation, as the evolving ice shapes necessitate the generation of new grids. This paper investigates the application of Immersed Boundary Methods (IBMs) to eliminate the need for volume remeshing, thereby enhancing the automation of multi-step simulations. The proposed framework, integrated into the ONERA 3D icing suite IGLOO3D, uses a ghost-cell method for modeling the airflow and a penalization approach for simulating droplet impingement, building upon recent work by the authors. By relying on inviscid flow simulations, the method significantly reduces computational costs; however, boundary layer calculations are required. To address this, a 3D simplified integral boundary layer solver based on the resolution of partial differential equations on a surface is presented. Results from three cases of the 1st Ice Prediction Workshop are presented and analyzed, demonstrating the potential of IBMs to enhance the efficiency and practicality of 3D ice accretion simulations.

6.2 Introduction

In-flight ice accretion is a well-known hazard to flight safety and performance. It typically occurs under cold weather conditions, when an aircraft encounters a supercooled cloud. One of the main results of the presence of this phenomenon is the degradation of the aerodynamic performance, increasing the drag and reducing the lift, but it also impairs the controllability of the aircraft as well as the data readout among others. Despite the development of aircraft ice protection systems, commercial aircraft must be able to prove that they are capable of flying under icing conditions. Thus, there is a need for ice-accretion modeling. In particular, the development of 3D numerical tools is of growing interest, as evidenced by the regular workshops held on this subject [84, 86].

Numerical simulations of ice accretion typically commence with the premise that the surface exposure time to icing conditions, t_{exp}, significantly exceeds the characteristic time of the aerodynamic flow. This facilitates the division of the entire phenomenon into distinct modules (see Fig. 6.1).

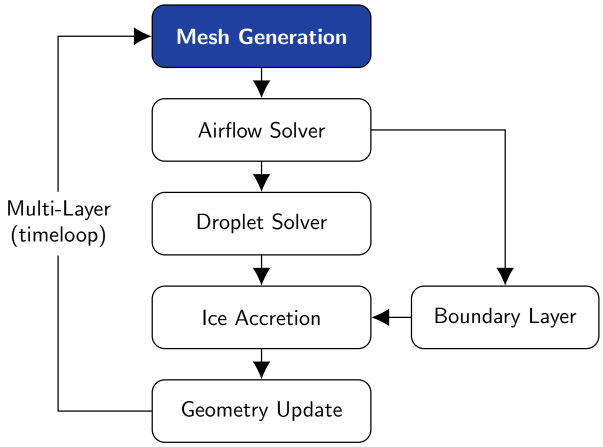


Figure 6.1 Sequential call to modules in multi-step icing simulations.

First, the steady-state aerodynamic flow is solved. Depending on the desired level of fidelity, this can be done using potential methods, solving the Euler equations, or employing more precise models like the Reynolds-Averaged Navier–Stokes (RANS) equations. It should be noted that when performing inviscid simulations, the boundary layer must be solved to feed the ice accretion simulation. It is a common practice in icing codes to weakly couple these solvers, i.e., the effect of the boundary layer on the inviscid flow is neglected. Concerning the results of the 1^{st} Ice Prediction Workshop (IPW), all participants solved the RANS equations, while only a few participants also studied the use of potential methods (LEWICE [188] and Multi-Ice [37, 108]) or inviscid flow (IGLOO2D [142, 176]) for 2D simulations. The use of higher-fidelity techniques, such as Large Eddy Simulation (LES), is rare in the literature and is restricted to the simulation of some specific physcial phenomena like droplet splashing but not of the whole ice accretion phenomenon [193]. Next, the trajectories of supercooled water droplets are typically computed using either an Eulerian or a Lagrangian approach. Regarding the outcomes of the 1^{st} IPW, half of the participants use one approach while the remaining half employ the other. Finally, the ice thickness is typically obtained by solving the Messinger model [106] or an alternative approach like the extended Messinger model [117] or the Shallow Water Icing Model (SWIM) [26]. Once the ice thickness is known, the exposed surface can be deformed through either a Lagrangian displacement of the surface mesh nodes [188] or through a level-set resolution [28, 90, 131].

In a one-shot simulation, also known as one-step or single-step simulation, the main loop is only performed once. This is the most cost-effective yet the least accurate method for conducting an ice accretion simulation. One alternative, which enhances the accuracy of the simulation without a substantial increase in computational cost, is the use of a predictor-corrector approach [141]. A more commonly used alternative, known as the multi-step method, involves dividing the exposure time into N discrete steps. This allows for the consideration of variations in both the surrounding flow and the droplet impingement as the ice accretes [176]. As the number of steps increases, so does the computational cost of the simulation, and the feasibility of automation is hindered. This challenge is predominantly attributed to the intricate task of generating meshes at the beginning of each iteration. Hence, conducting such simulations, particularly in 3D scenarios, proves challenging. For instance, in the 1^{st} IPW, only 25% of the participants successfully performed multi-step 3D simulations.

To solve these problems, an efficient alternative is the use of Immersed Boundary Methods (IBMs) that allows eliminating or, at least, significantly reducing the remeshing phase, which facilitates the automation of the process. The concept of IBM was developed by Peskin [133] in 1972 to simulate cardiac mechanisms associated with blood flow. Since then, the use of IBMs has become an interesting alternative to the usual Body-Fitted (BF) approach when dealing with complex and/or moving geometries.

Currently, there is a large variety of IBM types [109]. On the one hand, the *continuous forcing* approach is characterized by adding a forcing term to the continuous equations. This forcing term is only activated in the solid domain, although the flow equations are solved in the whole computational domain. It is an easy approach to implement; nevertheless, the method generally has a low order of accuracy. On the other hand, the *discrete forcing* approach introduces the forcing after the equations are discretized. For this purpose, some geometry preprocessing is often required, which can become challenging in most cases. Among them, there is the ghost-cell method, in which the solution is forced into ghost cells in order to impose the boundary conditions on the immersed boundary (IB).

The use of IBMs for ice accretion simulations, though still relatively limited, is gaining traction in the field. Several research centers and universities, including CIRA [33, 37, 38], ONERA [89, 90, 91], Polytechnique Montreal [24] and Strasbourg University [7] have developed IBM-based frameworks for ice accretion simulations. A summary of these methods and their key characteristics is presented in Table 6.1.

In a previous publication [61], the authors introduced a new framework for airflow and

Institution	**Software**	**Airflow**	**Airflow IBM**	**Droplets IBM**	**Domain**
CIRA	SIMBA	RANS and RANS-LES	Discrete	Discrete	2D/3D
ONERA	IGLOO2D	Euler	Continuous	Continuous	2D
ONERA	IGLOO3D	Euler	Discrete	Continuous	2D/3D
Poly. Montreal	CHAMPS	Laminar NS	Discrete	Continuous	2D
Uni. Strasbourg	NSMB-ICE	Laminar NS	Continuous	Discrete	2D

Table 6.1 Comparison of current ice accretion simulation methods using IBMs.

droplet trajectory calculations in three-dimensional simulations, relying exclusively on immersed boundary methods. This approach combines two distinct techniques: a ghost-cell method for solving the Euler equations [62] and a penalization method [91] for the droplet solver. This paper examines the application of these IBMs to three-dimensional in-flight ice accretion simulations within the IGLOO3D icing suite [141], using a multi-step approach. Section 6.3 describes the models used for the simulations and details the modifications made to IGLOO3D. Section 6.4 presents and analyzes ice accretion results for three cases from the 1^{st} IPW. Finally, Section 6.5 discusses some limitations of the proposed methodology.

6.3 Methodology

The development environment is the 3D ice accretion suite, IGLOO3D [141]. Designed with a modular architecture, this numerical tool enables different solvers to interact with each other via CGNS file inputs and outputs [112], facilitating all the necessary steps for ice accretion simulations. It includes several modules, such as those for estimating the heat transfer coefficient and surface mesh displacement, with the ice accretion module being the main component.

Regarding the IBMs, two different solvers are employed for the aerodynamics and the droplet trajectories. The former is computed using the ONERA HPC FastS solver [3]. In particular, FastS applies the Adaptive Ghost-Cell Surrounding method [62] to solve the Euler equations. The latter is computed using CEDRE [146], the ONERA's multi-physics numerical tool, which includes unstructured solvers for several physical problems (airflow, droplet trajectories, conductive heat transfer, radiative heat transfer, etc.). Among these solvers, the Eulerian dispersed multiphase flow solver SPIREE [115] is used to solve the penalized droplet trajectory equations presented in [91].

The remaining modules are used from IGLOO3D [141]. The simplified integral method, developed in HTC3D, is used for computing the heat transfer coefficient at the wall, derived from the inviscid flow and grounded on assumptions associated with 2D flow [176].

Serving as the cornerstone of IGLOO3D, the ice accretion module, MESSINGER3D [141], determines the ice thickness by solving iteratively a Messinger-type mass and energy balance equation. Finally, LEVELSET3D [141] updates the iced surface from the level-set field. Figure 6.2 shows the new sequential call to modules, where the modules related to the IBM are highlighted in red.

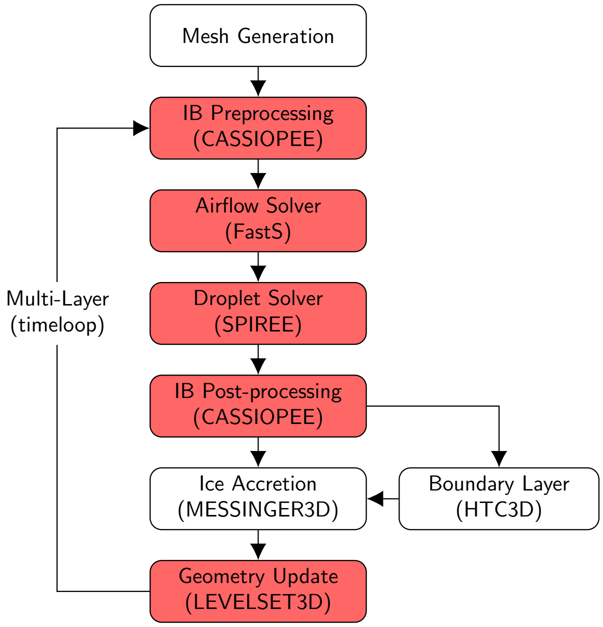


Figure 6.2 Sequential call to modules in multi-step icing simulations using IBMs.

In terms of mesh generation, it is important to highlight that this paper adopts the strategy shown in [90]. In this approach, the mesh for the entire multi-step simulation is derived from a BF mesh generated from the clean geometry of the wing. This method entails employing IBMs exclusively in the icing zone while treating the remainder of the wall as BF. Therefore, the same volume mesh is employed for all the iterations of the multi-step simulation. In addition, this mesh must be structured, as mandated by the aerodynamic solver's requirements. Concerning the iced surface mesh, it is updated at each iteration and refined in order to keep the same mesh size when it is deformed.

In the following sections, we discuss the modified modules as well as the advancements in the HTC3D module, which has undergone further development since its initial presentation in [141].

6.3.1 IB Preprocessing

During a multi-step simulation, volume solvers must implicitly define the geometry. This is usually achieved by establishing a variable that denotes the distance to the iced surface. With the surface explicitly known, the distance is computed by an orthogonal projection of each cell center of the volume mesh onto the surface. Additionally, a line-of-slight algorithm [19] is used to obtain the signed distance ϕ, which enables the distinction between fluid ($\phi > 0$) and solid ($\phi < 0$) cells. Finally, the normal vector to the IB $\vec{n}_\phi$ is required, which can be obtained from the signed distance field:

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

In the presented methodology, these geometric operations are performed using Cassiopee (CFD Advanced Set of Services In an Open Python EnvironmEnt) [19].

6.3.2 Aerodynamic Solver

This section briefly presents the immersed boundary methodology used to solve the inviscid flow.

6.3.2.1 Governing Equations

The conservative continuity, momentum, and energy Euler equations are respectively:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v}) = 0 \tag{6.2a}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \vec{\nabla} \cdot (\rho \vec{v} \otimes \vec{v} + pI) = 0 \tag{6.2b}$$

$$\frac{\partial (\rho E)}{\partial t} + \vec{\nabla} \cdot ((\rho \mathrm{E} + \mathrm{p}) \vec{v}) = 0 \tag{6.2c}$$

where ρ is the density, $\vec{v}$ is the velocity, p is the pressure, I is the identity tensor and E is the total energy per unit mass. To complete the system, two additional equations are required, which will establish thermodynamic relationships between the state variables. For a calorically perfect gas, the equations are:

$$p = \rho r T, \tag{6.3}$$

$$E = \frac{r}{\gamma - 1} T + \frac{||\vec{v}||^2}{2}, \tag{6.4}$$

where r denotes the specific gas constant, T the temperature and $\gamma = c_\mathrm{P}/c_\mathrm{V}$ is the ratio of specific heat coefficients.

The system of Eqs. 6.2 is solved in FastS [3], using a cell-centered finite-volume method on structured meshes. In the presented simulations, the inviscid fluxes are computed using a Roe scheme [149] and a second-order MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) extrapolation [179], without slope limiter. This is accomplished in FastS by using a stencil of 5 cells in each spatial direction. For temporal discretization, an implicit second-order backward differentiation formula is applied to ensure stable resolution to steady-state. The CFL number used in all the simulations is set to 2.

6.3.2.2 The Adaptive Ghost-Cell-Surrounding Method

In body-fitted simulations, ghost cells are added at the boundary to fulfill the calculation stencils of fluid cells near the boundary. The Ghost-Cell Immersed Boundary Method expands upon this concept by enabling the definition of boundary conditions in meshes that do not conform to the solid geometry (see Fig. 6.3).

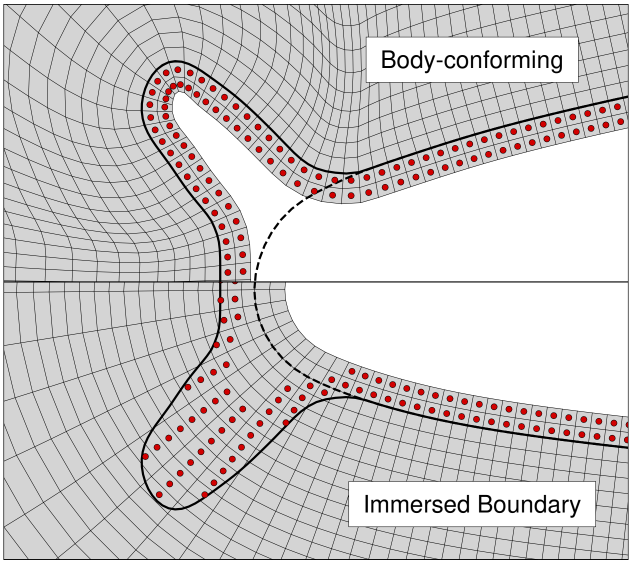


Figure 6.3 Comparison of the locations of ghost cells, shown as red circles, in body-conforming (top) and immersed boundary (bottom) meshes. The boundary of the manufactured ice shape is represented by the thick black solid line.

In contrast to traditional methods, where the ghost cells are positioned either within the solid or fluid, the Adaptive Ghost-Cell-Surrounding (AGCS) method [62] places the ghost cells according to a local curvature-based criterion. Specifically, ghost cells are placed in the fluid domain if $\kappa\Delta x > 10$, where κ is the local curvature, positive when the center of curvature lies within the solid. If this criterion is not met, the ghost cells are instead positioned within the solid. Additionally, the Curvature-Corrected Symmetry Technique [46] is employed to impose slip boundary conditions on the ghost cell, accounting for the wall curvature in the normal momentum equation while ensuring the conservation of total enthalpy and entropy in the direction normal to the wall. This approach is particularly effective for complex geometries, such as those typically found on icing applications.

To impose the boundary condition, each ghost cell has an image point, located within the computational domain, and a wall point, which corresponds to the projection of the ghost cell center onto the surface. Figure 6.4 illustrates the distribution of triplets (ghost cells, image points, and wall points) used to enforce the boundary conditions for the 2D manufactured ice shape of [62].

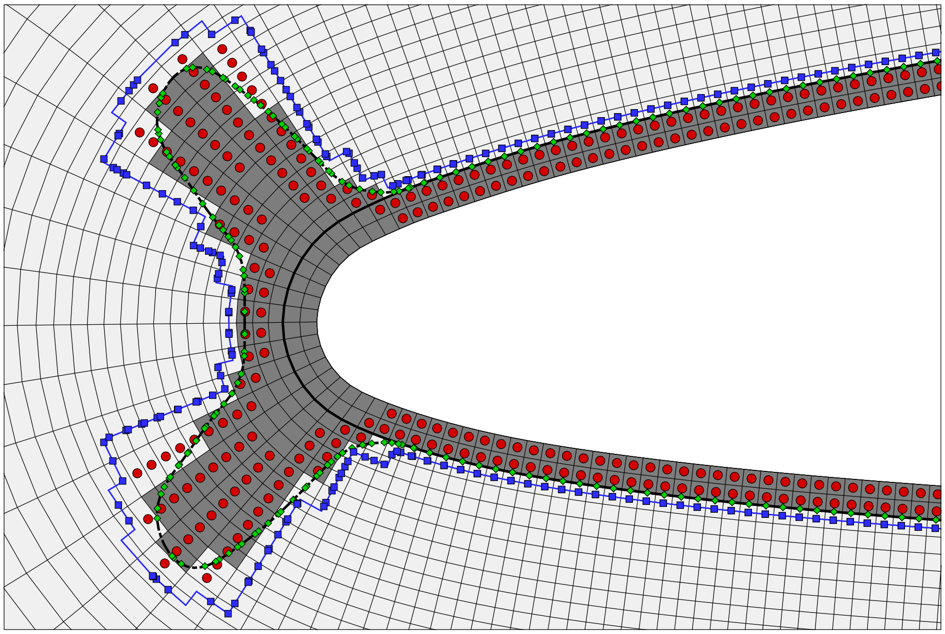


Figure 6.4 Triplets distribution for the 2D manufactured ice shaped. Ghost cells are represented by red circles, wall points by green diamonds, and image points by blue squares. Solid cells are displayed in dark gray.

6.3.3 Droplet Solver

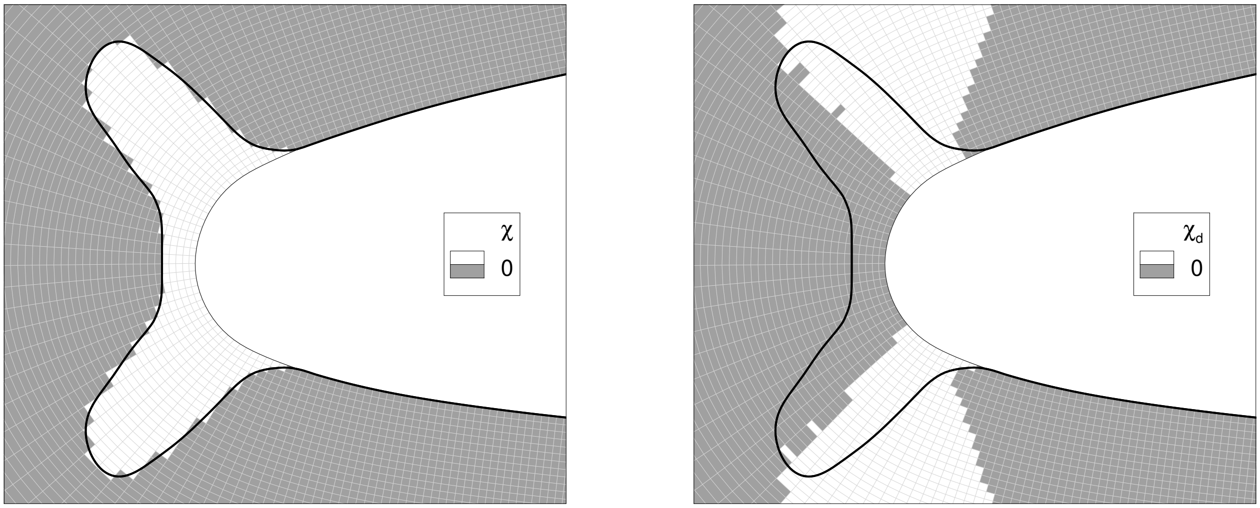
Lavoie et al. [91] introduced the penalization method to solve the droplet trajectory equations, initially applying it to 2D cases. This approach was later employed in 3D simulations in [61]. The penalization method modifies the governing equations of the droplet trajectory by adding penalty terms to enforce boundary conditions. At the impingement zone, where the non-dimensional droplet velocity $\vec{v}_\mathrm{d}$ and the normal vector to the surface $\vec{n_\phi}$ have opposing directions, i.e. $\vec{v}_\mathrm{d} \cdot \vec{n_\phi} \leq 0$, droplets can enter the solid interface. In contrast, when droplets would re-enter the computational domain from the solid, i.e. $\vec{v}_\mathrm{d} \cdot \vec{n_\phi} > 0$, a zero-flux condition is imposed to prevent re-injection. This is achieved by setting $\alpha = 0$ and $\vec{v}_\mathrm{d} = 0$, where α is the non-dimensional volume fraction of water.

In the VP-SSO (Volume Penalization Solid Source Off) [91] method, two mask functions are employed to activate or deactivate the penalization terms in order to enforce the boundary conditions. First, the conventional mask function χ is used to distinguish between the solid and fluid regions (see Fig. 6.5a):

$$\chi = \begin{cases} 0 & \phi \geq 0 \text{ (fluid)} \\ 1 & \phi < 0 \text{ (solid)} \end{cases} \tag{6.5}$$

Additionally, a second mask function, referred to as droplet mask function χ_d, differentiates between the impingement zone and the shadow zone (see Fig. 6.5b):

$$\chi_\mathrm{d} = \begin{cases} 0 & \alpha\vec{v}_\mathrm{d} \cdot \vec{n}_\phi \leq 0 \text{ (impingement zone)} \\ 1 & \alpha\vec{v}_\mathrm{d} \cdot \vec{n}_\phi > 0 \text{ (shadow zone)} \end{cases} \tag{6.6}$$



(a)

(b)

Figure 6.5 Mask function (a) and droplet mask function (b) for the 2D manufactured case.

To ensure the penalization terms are applied exclusively within the solid region, the two mask functions are combined as displayed in Fig. 6.6.

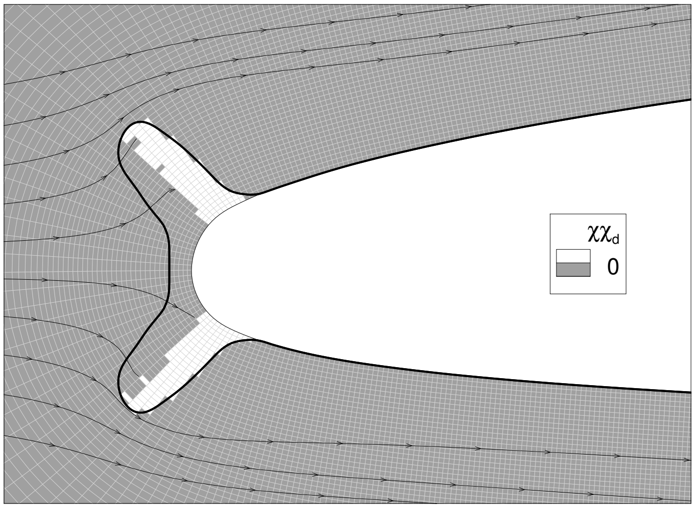


Figure 6.6 Combined mask function and droplet streamtraces for the 2D manufactured case.

When neglecting the influence of gravity, these equations are as follows:

$$\frac{\partial \alpha}{\partial t} + \vec{\nabla} \cdot (\alpha \vec{v}_\mathrm{d}) = -\frac{\chi\chi_\mathrm{d}}{\eta}\alpha \tag{6.7a}$$

$$\frac{\partial (\alpha \vec{v}_\mathrm{d})}{\partial t} + \vec{\nabla} \cdot (\alpha \vec{v}_\mathrm{d} \otimes \vec{v}_\mathrm{d}) = (1-\chi)\,\frac{\mathrm{C_D}\,\mathrm{Re_d}}{24\,\mathrm{Stk}}\alpha\,(\vec{v} - \vec{v}_\mathrm{d})\,-2\frac{\chi\chi_\mathrm{d}}{\eta}\alpha\vec{v}_\mathrm{d} \tag{6.7b}$$

where $\mathrm{C_D}$ is the droplets drag coefficient, $\mathrm{Re_d}$ is the droplets Reynolds number, Stk is the Stokes number, and $\eta \ll 1$ is chosen small enough such that the penalization terms become predominant when applied. Here, η is selected as 10^{-7}. In the impingement zone of the solid region, the physical source terms are deactivated, making the droplet solution independent of the gas velocity field.

The system of Eqs. 6.7 are solved using SPIREE [115], where the Schiller and Naumann model for drag computation is employed. The Eulerian droplet-trajectory solver employs the finite-volume method on unstructured grids. In the simulations discussed in this paper, a first-order Euler implicit scheme was employed for time discretization and a second-order spatial discretization was used, employing a Godunov-like scheme [92, 115] and MUSCL extrapolation.

6.3.4 IB Post-processing

Essential surface data obtained from the volume solvers needs to be transmitted to the surface solvers during each iteration of the multi-step loop. In contrast to body-fitted approaches, extracting the flow fields directly at wall boundaries is more complex when using IBMs. Various methods can be employed to obtain the solution at the cell centers of the discrete surface. In this framework, a second-order Lagrange interpolation is performed using Cassiopee [19].

6.3.5 Boundary Layer

The principal aim of the boundary layer module, HTC3D, is to determine the value of the convective heat transfer coefficient, which is a key parameter in ice accretion simulations. In the previous version of HTC3D, this variable was obtained from the results of a RANS simulation. The current approach introduces a simplified method to derive this value directly from the inviscid flow solution. This section first outlines the model, highlighting the key assumptions and simplifications made, followed by a brief explanation of the numerical methods used, including how the stagnation point is treated. Additionally, the proposed methodology has been validated against Achenbach's measurements [5], which provide heat transfer data for a rough cylinder (see Appendix). This case has been commonly used to validate integral boundary layer methods, such as those of Makkonen [101] and LEWICE [152].

In the subsequent section, the typical boundary layer notation is applied, wherein the aerodynamic variables are referenced to the outer edge of the boundary layer ($\cdot_{\mathrm{e}}$). The IB post-processing module supplies these variables in the present icing suite.

6.3.5.1 Simplified Integral Method

HTC3D employs a simplified integral method, based on the work of Trontin et al. [176] for 2D simulations. In the present 3D approach, the effect of transverse flow inside the boundary layer is neglected, assuming a two-dimensional behavior in the streamwise direction.

The method distinguishes between two different regions: a laminar region, confined to a small zone in the vicinity of the stagnation point, and a turbulent region, where the roughness of the surface is taken into account. The laminar-turbulent transition used is based on the roughness Reynolds numbers [183], Re_{k}. Here it is defined as:

$$\mathrm{Re}_{\mathrm{k}} = \frac{k_{\mathrm{s}} u_{\mathrm{e}}}{\nu_{\mathrm{e}}} > 600 \tag{6.8}$$

where k_{s} is the equivalent sand-grain roughness, while u_{e} and ν_{e} are the velocity and kinematic viscosity of the airflow at the edge of the boundary layer, i.e. the Euler solution values at the wall in our case. As in other ONERA's codes, the value of k_{s} is set to one-thousandth of the wing chord length.

In the laminar region, the computation of the heat transfer coefficient relies on the Smith and Spalding formula [162], expressed as:

$$h_{\mathrm{tc}} = \rho_{\mathrm{e}} c_{\mathrm{p}} u_{\mathrm{e}} \frac{0.2926 \nu_{\mathrm{e}}}{u_{\mathrm{e}} \, \mathrm{Pr}} \sqrt{u_{\mathrm{e}}^{2.87} / \int_0^s \nu_{\mathrm{e}}\left(s'\right) u_{\mathrm{e}}\left(s'\right)^{1.87} \mathrm{d}s'} \tag{6.9}$$

where c_{p} is the air specific heat capacity at constant pressure, $\mathrm{Pr} = 0.7$ is the air Prandtl number and s is the curvilinear abscissa, being 0 at the stagnation point.

In the turbulent region, the expression is formulated as follows [152, 176]:

$$h_{\mathrm{tc}} = \rho_{\mathrm{e}} c_{\mathrm{p}} u_{\mathrm{e}} \frac{C_{\mathrm{f}}/2}{\mathrm{Pr}_{\mathrm{t}} + \sqrt{C_{\mathrm{f}}/2} / \left(1.92 k_{\mathrm{s}}^{+\,-0.45} \, \mathrm{Pr}^{-0.8}\right)} \tag{6.10}$$

where C_{f} is the skin friction coefficient, $k_{\mathrm{s}}^{+} = \frac{u_{\mathrm{e}} \sqrt{C_{\mathrm{f}}/2} k_{\mathrm{s}}}{\nu_{\mathrm{e}}}$ is the non-dimensional sand-grain roughness height, and $\mathrm{Pr}_{\mathrm{t}} = 0.9$ is the turbulent Prandtl number. In HTC3D, the skin friction coefficient is computed as [77, 152]:

$$\frac{C_{\mathrm{f}}}{2} = \frac{0.168}{\ln\left(864 \theta_{\mathrm{T}} / k_{\mathrm{s}} + 2.568\right)^2} \tag{6.11}$$

where θ_{T} is the momentum thickness of the turbulent boundary layer; therefore, the dynamic boundary layer must be solved. In the laminar zone, the Thwaites method [173], originally developed for 2D flows, is used:

$$\theta_{\mathrm{L}} = \frac{0.664}{u_{\mathrm{e}}^{2.84}} \sqrt{\int_0^s \nu_{\mathrm{e}}\left(s'\right) u_{\mathrm{e}}\left(s'\right)^{4.68} \, \mathrm{d}s'} \tag{6.12}$$

In the turbulent zone, the approach presented by Kays and Crawford [77] is used. This method is based on a power-law velocity profile within the turbulent region, allowing to obtain the momentum thickness as follows:

$$\theta_{\mathrm{T}} = \theta_{\mathrm{tr}} + \left(\frac{0.016}{u_{\mathrm{e}}^{4.11}} \int_{s_{\mathrm{tr}}}^{s} \nu_{\mathrm{e}}\left(s'\right)^{1/4} u_{\mathrm{e}}\left(s'\right)^{3.86} \, ds'\right)^{4/5} \tag{6.13}$$

where θ_{tr} is the value of the momentum thickness of the boundary layer at the laminar-turbulent front following the streamline direction and s_{tr} is the local value of the curvilinear abscissa along the streamline from the laminar-transition front.

6.3.5.2 Numerical Method

From a numerical perspective, the main difficulty of the simplified method lies in the computation of integrals along the parietal streamline. These integrals are found in Eqs. 6.9, 6.12 and 6.13 and, in a general form, they can be written as:

$$\psi = \int_0^s \phi(s')\,\mathrm{d}s' = \int_0^s \mathrm{d}\psi' \tag{6.14}$$

where s is the local value of the curvilinear abscissa along the streamline and ϕ is the function to be integrated. To compute these integrals, the finite-volume method presented in [141] is used. In this approach, a pseudo time step $\mathrm{d}t$ is defined:

$$\mathrm{d}t = \kappa \mathrm{d}\psi \tag{6.15}$$

where κ is a unit scalar such that its units are compatible with the pseudo time step. The propagation velocity can be then defined locally from the velocity field, $\vec{u}_\mathrm{e}$, consistently with the fact that integrals are computed along streamlines:

$$\vec{V}_\psi = \frac{\mathrm{d}s}{\mathrm{d}t}\vec{e}_\mathrm{s} = \frac{1}{\kappa\phi(s)}\frac{\vec{u}_\mathrm{e}}{||\vec{u}_\mathrm{e}||} \tag{6.16}$$

This formulation enables Eq.6.15 to be rewritten as a hyperbolic PDE, which can be expressed in both non-conservative and conservative forms:

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = \frac{1}{\kappa} = \frac{\partial\psi}{\partial t} + \vec{V}_\psi \cdot \vec{\nabla}\psi \tag{6.17a}$$

$$\frac{\partial\psi}{\partial t} + \vec{\nabla}\cdot\left(\psi\vec{V}_\psi\right) = \frac{1}{\kappa} + \psi\vec{\nabla}\cdot\vec{V}_\psi \tag{6.17b}$$

Additionally, the value of the momentum thickness of the boundary layer at the laminar-turbulent front has to be propagated in the turbulent zone following the streamwise direction. To accomplish this, a procedure analogous to the one used for computing integrals along the streamline is employed. In this case, θ_tr remains constant in the streamline direction, and therefore:

$$\frac{\mathrm{d}\theta_\mathrm{tr}}{\mathrm{d}t} = 0 = \frac{\partial\theta_\mathrm{tr}}{\partial t} + \vec{V}_\mathrm{s}\cdot\vec{\nabla}\theta_\mathrm{tr} \tag{6.18a}$$

$$\frac{\partial\theta_\mathrm{tr}}{\partial t} + \vec{\nabla}\cdot\left(\theta_\mathrm{tr}\vec{V}_\mathrm{s}\right) = \theta_\mathrm{tr}\vec{\nabla}\cdot\vec{V}_\mathrm{s} \tag{6.18b}$$

where the propagation velocity is the normalized velocity vector $\vec{V}_\mathrm{s} = \vec{e}_\mathrm{s}$. Figure 6.7 shows a zoomed-in view of the leading edge near the wing tip of a 30º swept NACA 0012 wing. In the figure, θ_{tr} is obtained at a turbulent face (black circle) by propagating the momentum thickness values from the laminar cells adjacent to the laminar-turbulent front (red circles) in the streamwise direction (streamtrace).

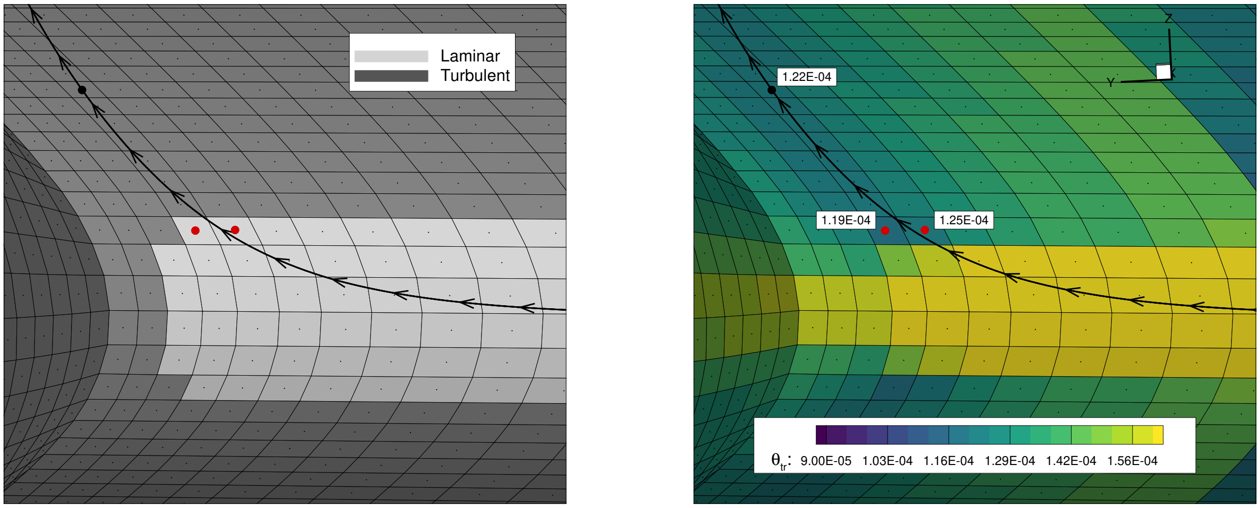


Figure 6.7 Numerical propagation of the momentum thickness of the boundary layer at the laminar-turbulent front.

6.3.5.3 Stagnation Point Correction

The simplified integral boundary layer method does not provide a solution at the faces corresponding to stagnation points, where the streamlines begin. Specifically, in these faces $\psi = 0$ and an alternative is required to provide values of the momentum boundary layer thickness and the heat transfer coefficient. To address this issue, the proposed correction entails an analytical solutions of the integrals found in Eqs. 6.9 and 6.12 by assuming a linear velocity around the stagnation point [65]

$$u_{\text{e}}(s) = U_0 s, \tag{6.19}$$

where U_0 is a constant that can be assessed from the neighborhood of the face.

6.3.6 Geometry Update

When performing body-fitted simulations, the typical method for generating a new geometry involves a straightforward grid deformation technique based on a Lagrangian node displacement. Given the assumption of perpendicular ice accretion to the body, once the ice thickness over the surface is determined, the new geometry can be obtained:

$$\vec{x}_{\text{new}} = \vec{x}_{\text{old}} + h_{\text{ice}}\vec{n} \tag{6.20}$$

where $\vec{x}$ represents the node locations, h_{ice} is the ice thickness, provided by the ice accretion solver, and $\vec{n}$ is the normal direction to the wall, pointing toward the fluid.

The use of this Lagrangian approach is recognized to entail significant challenges. Specifically, issues such as overlapping tend to arise readily, particularly in concave regions, thereby compromising the integrity of the surface mesh and disrupting the main loop of a multi-step simulation. One of the most commonly used alternatives, especially when using immersed boundary methods, is the level-set method [90, 141].

6.3.6.1 Level-Set Method

The level-set method available in IGLOO3D was first presented in [141] for BF simulations. This approach is based on a Lagrangian displacement driven by an Eulerian field. When doing N_{L} sub-iterations, the displacement of each node at iteration k, $\Delta\vec{x}^k$, is indeed given by the following equation:

$$\Delta\vec{x}^k = \frac{h_{\text{ice}}(\vec{x}^0)}{N_{\text{L}}}\vec{n}_\phi(\vec{x}^k), \quad \forall k \in [1, N_L] \tag{6.21}$$

where $h_{\text{ice}}(\vec{x}^0)$ is the ice thickness of the node obtained by the ice accretion solver, $\vec{n}_\phi(\vec{x}^k)$ is the interpolated normal vector at the current position $\vec{x}^k$ (inferred from the level-set field). As a result, the value of $\vec{n}_\phi(\vec{x}^k)$ is updated during each iteration as node $\vec{x}^k$ is shifted.

The number of sub-iterations is a parameter that reduces the likelihood of overlaps in the final surface mesh. When $N_{\text{L}} = 1$, a classical Lagrangian deformation is performed, which, under certain conditions, may lead to overlaps. By increasing the number of sub-iterations, the advancement direction of the geometry is updated at each step, decreasing the probability of overlap formation, as the direction is determined based on the normal field (see Fig. 6.8, where the signed distance field and normal directions refers to the iced shape). In the limiting case of an infinite number of sub-iterations, overlaps cannot occur. For all simulations presented in this paper, $N_{\text{L}} = 50$ has proven adequate.

Although it was initially developed for body-fitted simulations, its application in IBM simulations is straightforward.

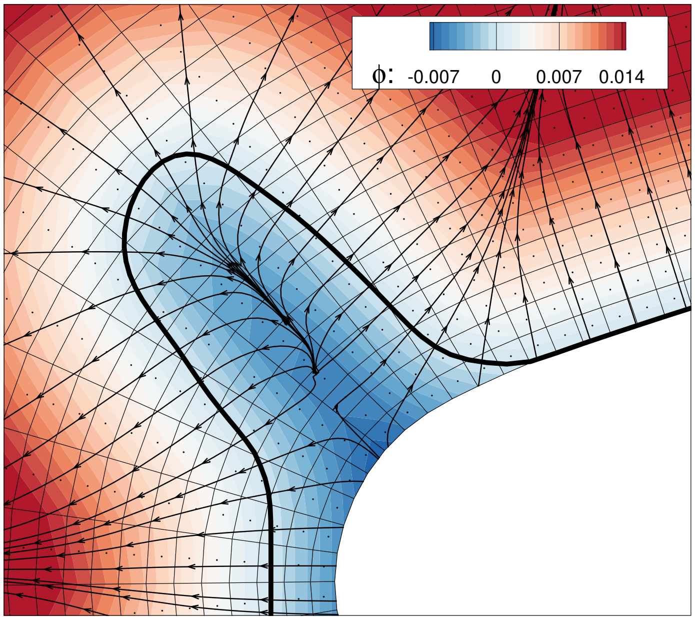


Figure 6.8 Signed distance field and normal directions, represented as streamlines, around the 2D manufactured iced geometry.

6.3.6.2 Surface Mesh Refinement

When employing IBMs for ice accretion simulations, the level-set method has traditionally been the predominant approach for updating the geometry. In typical level-set methods, an Eulerian advection equation is solved to displace the iso-0 level [90], and the final geometry is implicitly determined at the end of the simulation. Consequently, numerical post-processing tools are required to extract the geometry from the volume mesh. In contrast, the proposed approach directly displaces all surface nodes, eliminating the need for such post-processing. However, while the number of surface nodes remains constant before and after displacement, this can lead to unevenly distributed nodes, resulting in widely spaced cells. To maintain the initial mesh resolution, surface refinement is often necessary.

To address this issue, surface refinement is performed using an interpolating cubic spline [19, 159]. This method ensures that the surface solvers maintain a consistent and high-quality mesh resolution throughout the simulation. By doing so, potential numerical issues are mitigated, particularly in critical regions such as around the stagnation line.

6.4 Results

This section presents the results of applying the proposed IB approach to multi-step 3D icing simulations. Several cases from the 1^{st} IPW were selected, allowing comparison with both experimental data and other ice simulation codes. The study focuses on three baseline cases, which include both 2D and 3D geometries as well as rime and glaze icing conditions. Under rime icing conditions, characterized by low temperatures, ice formation is primarily governed by collection efficiency, as water freezes immediately upon impingement, resulting in relatively streamlined ice shapes. In contrast, under glaze icing conditions, higher temperatures allow for water runback, where impinging droplets remain liquid and flow further downstream before freezing. For all cases, the icing conditions follow Appendix C [63], assuming supercooled water droplets small enough to deposit on exposed surfaces. As a result, effects like splashing are disregarded.

The analysis begins with the 2D rime and glaze icing regimes on a straight-wing NACA 23012 at a 2° angle of attack. Then, the new framework is evaluated on a 3D NACA 0012 wing with a 30° sweep angle and a zero-degree angle of attack under rime conditions. Table 6.2 summarizes the simulation parameters for each case.

Case	**241**	**242**	**361**
Tunnel run	ED1977 [94]	ED1978 [94]	AF2146 [22]
Airfoil	NACA 23012	NACA 23012	NACA 0012
Sweep Angle [$^{\circ}$]	-	-	30
Chord [m]	0.4572	0.4572	0.91438
Angle of Attack [$^{\circ}$]	2.0	2.0	0
Mach [-]	0.325	0.315	0.32
$\mathbf{P_{static}}$ [Pa]	92528	92941	92321
$\mathbf{T_{static}}$ [K]	250.15	266.05	257
LWC [Kg/m^3]	$0.42 \cdot 10^{-3}$	$0.81 \cdot 10^{-3}$	$0.5 \cdot 10^{-3}$
MVD [m]	$30 \cdot 10^{-6}$	$15 \cdot 10^{-6}$	$34.7 \cdot 10^{-6}$
Icing Time [s]	300	300	1200
Roughness (ks) [m]	$0.4572 \cdot 10^{-3}$	$0.4572 \cdot 10^{-3}$	$0.91438 \cdot 10^{-3}$

LWC = Liquid Water Content
MVD = Median Volume Diameter

Table 6.2 Simulation Parameters

Note that for all simulations, the workshop organizing committee has recommended using a 7-bin distribution to represent the droplet-size distribution in the wind tunnel [86]. However, for simplicity, in this paper all simulations are performed using a mono-dispersed droplet size, unless stated otherwise.

6.4.1 Case 241

A series of multi-step ice accretion simulations was performed to verify and validate the proposed IB approach under 2D rime icing conditions. The numerical experiments included simulations with 2, 8, 16, 32, and 64 steps, demonstrating that the framework imposes no restriction on the number of steps. Comparisons were made both against BF meshes and the experimental data [94].

When performing IB simulations, the volume mesh remains unchanged throughout all steps. The initial geometry of the NACA 23012 airfoil, discretized using 552 faces, serves as the basis for generating the volume mesh. The mesh size around the leading edge is $\Delta x/c = 1.5 \cdot 10^{-3}$, and this resolution is maintained within a region extending approximately 10% of the chord length c from the wall, as shown in Fig. 6.9. The farfield boundary is placed at a distance of roughly 20 chord lengths, resulting in a computational domain comprising 69k cells. For the BF meshes of the intermediate layers, the number of cells is increased while preserving the same farfield distance and mesh resolution around the wall.

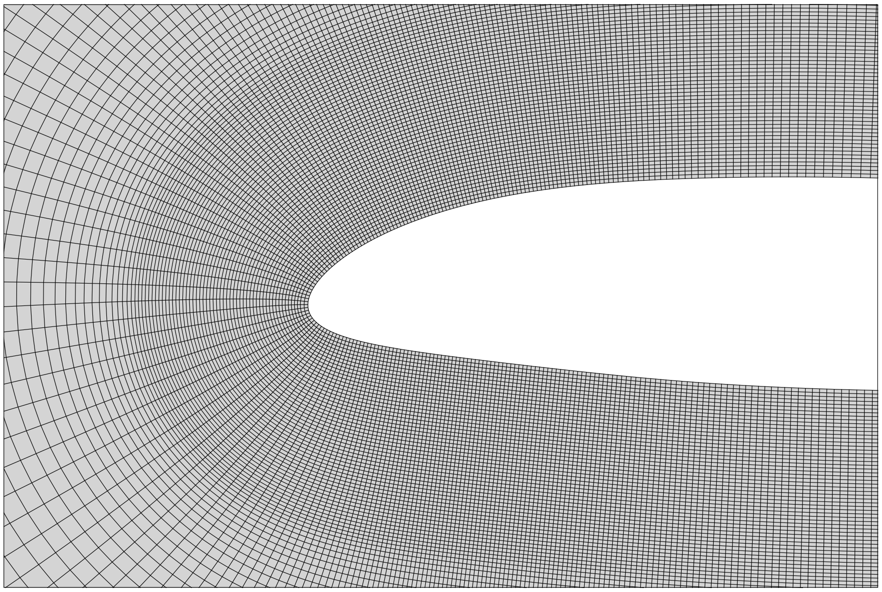
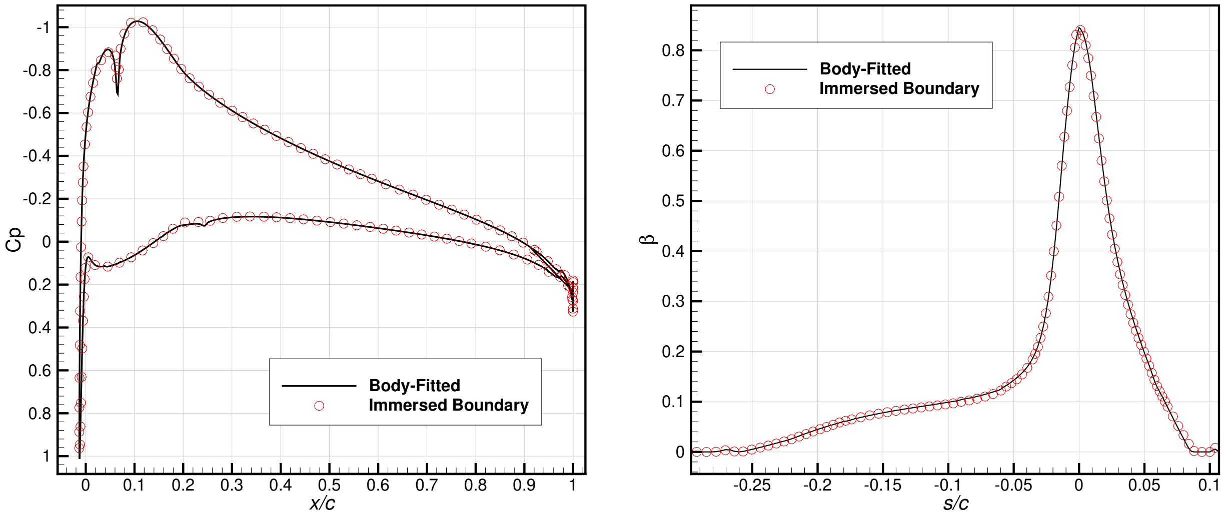


Figure 6.9 Structured mesh around a NACA 23012 airfoil.

The presented methodology is first verified by comparing the ice shapes with those obtained using the conventional BF approach. To this end, 2-step simulations have been analyzed in detail, where the first layer obtained is always identical in both cases, as the two methodologies start from the body-conforming mesh of the ice-free airfoil. Figure 6.10 presents a comparison of the pressure coefficient, C_p, and the collection efficiency, β, for the BF and IB methods applied to the first ice layer of the 2-step simulation. In Fig. 6.10b, s/c refers to the dimensionless surface wrap distance. The positive values correspond to the suction side of the airfoil. The C_p and β curves for both methods exhibit a close match, with only minor differences observed, indicating a strong correlation between the BF and IB approaches. The close agreement between the curves translates into nearly identical ice shapes for both methods, as shown in Fig. 6.11a. As more steps are employed, one would typically expect the differences between the IB and BF methods to become more pronounced. However, the 8-step simulation continues to yield results that closely match the BF solution, as illustrated in Fig. 6.11b, demonstrating the robustness of the proposed IB approach.



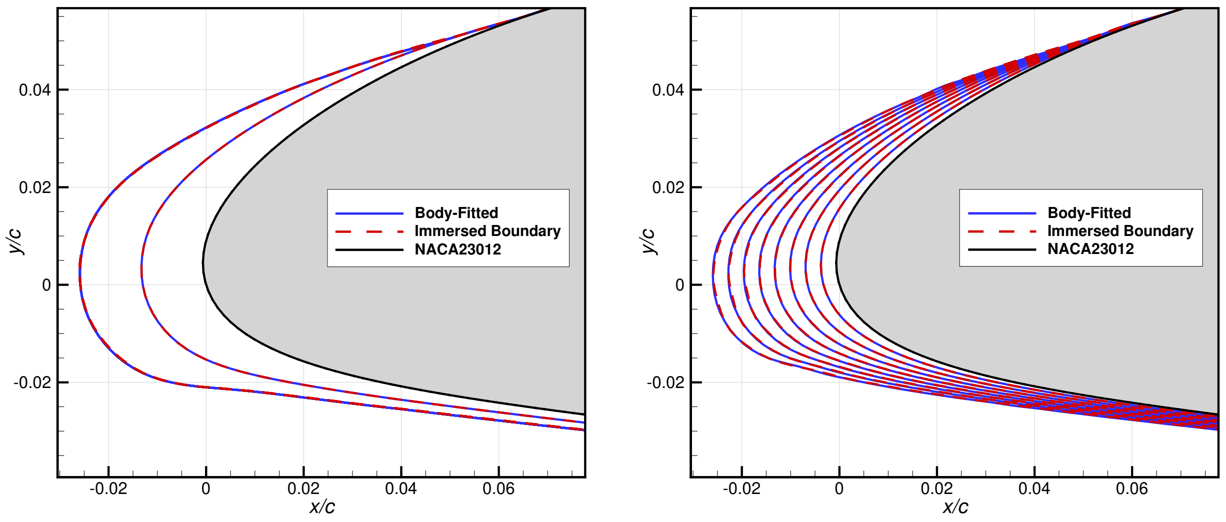
(a) Pressure coefficient

(b) Collection efficiency

Figure 6.10 Case 241, 2-step simulation. Comparison of wall surface solution on the first ice layer for the BF and IB methods.

To validate the current methodology, the different ice shapes obtained by increasing the number of layers are compared to the experimental results, as shown in Fig. 6.12. The experimental run was performed on a 3D straight wing at the NASA Icing Research Tunnel (IRT) [94], showing slight differences in the spanwise direction. Here, the result presented shows the envelope along the entire span on the xy-plane. The ice shapes simulated by the present approach converge asymptotically toward the results obtained with 64 steps. A low number of steps results in a higher total ice mass. As the number of steps increases, the predicted ice shape more closely aligns with the experimental measurements, capturing the overall ice accretion patterns with greater accuracy.

Additionally, a 32-step simulation was performed considering the 7-bin Langmuir-D droplet size distribution [128]. The resulting ice predictions, shown in Fig. 6.13, demonstrate an



(a) 2 layers

(b) 8 layers

Figure 6.11 Case 241. Comparison of BF and IB ice shapes predictions for different multi-steps simulations.

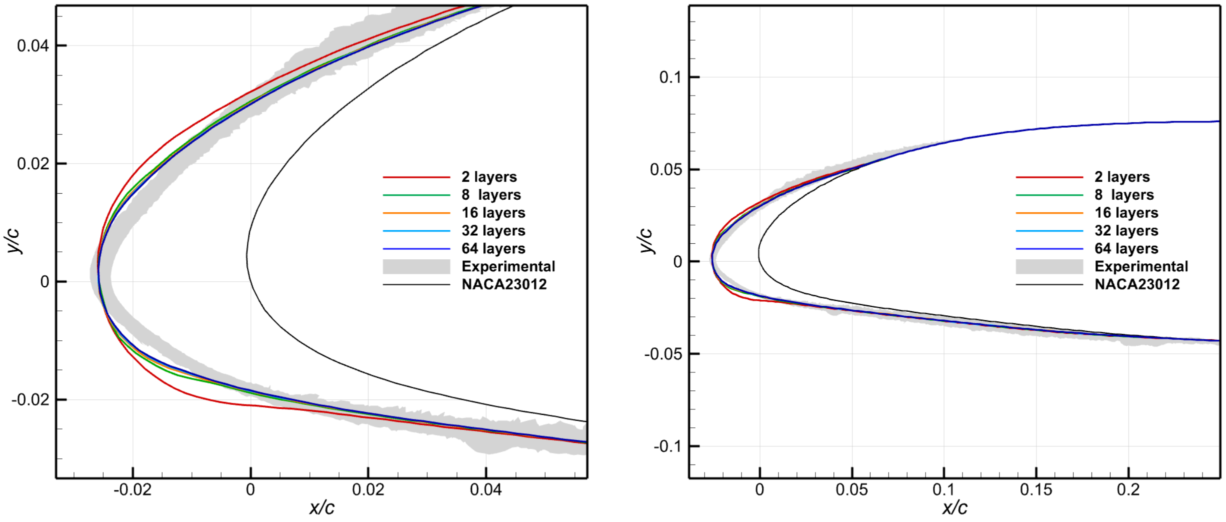


Figure 6.12 Case 241, IB simulations. Ice shape comparison with increasing number of layers.

improvement, particularly around the stagnation point and at the impingement limits. By accounting for droplet size variation, the predicted ice shape better matches the experimental data.

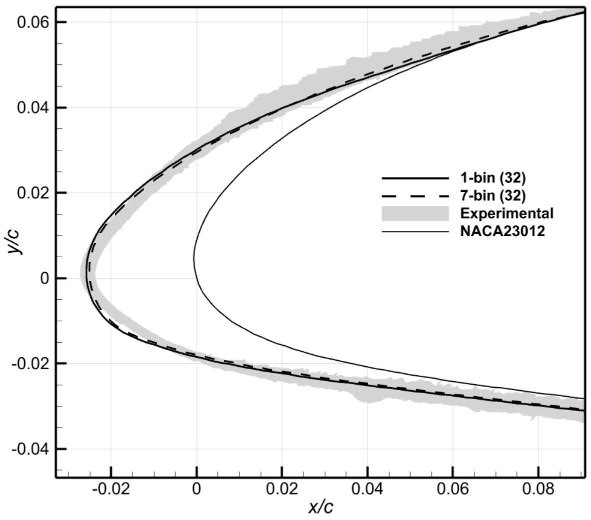


Figure 6.13 Case 241, 32-step IB simulation. Comparison of ice shapes predictions using mono-dispersed and Langmuir-D droplet sizes distribution.

6.4.2 Case 242

Glaze icing conditions are also assessed for the NACA 23012 airfoil. These conditions, distinguished by elevated temperatures, facilitate water runback behavior. This phenomenon entails the deposited water remaining partially in a liquid state rather than instantly freezing, allowing it to flow downstream before undergoing freezing. Predicting the outcomes of these simulations can be notably challenging, as evidenced by the results from the 1^{st} IPW [86].

A key factor contributing to the huge discrepancies in the ice shapes obtained by the 1^{st} IPW participants is the prediction of the convective heat transfer coefficient, h_{tc}, which plays a crucial role in glaze icing conditions. Given its importance, the analysis begins with an examination of the h_{tc} distribution over the clean airfoil. In Fig. 6.14, the results of the current approach are compared with those from IGLOO2D, presented in [90]. Similar to the present approach, IGLOO2D solves the inviscid flow and uses a simplified integral method to resolve the two-dimensional boundary layer. Additionally, to highlight the significant disparity observed in the 1^{st} IPW, the curves from some participants have been included. All participant curves were not added, in order to facilitate data readability in the figure. The present method shows good agreement with the IGLOO2D data, serving also as a validation test for HTC3D.

Different multi-step simulations are performed and compared against both the BF method and experimental results. As this case analyze the same geometry as Case 241, the same

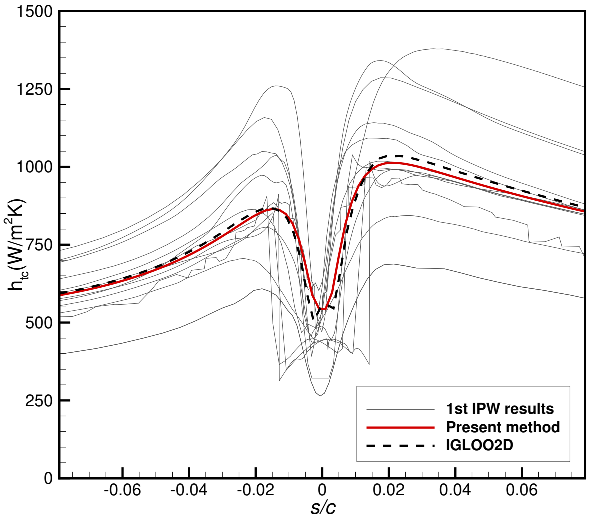
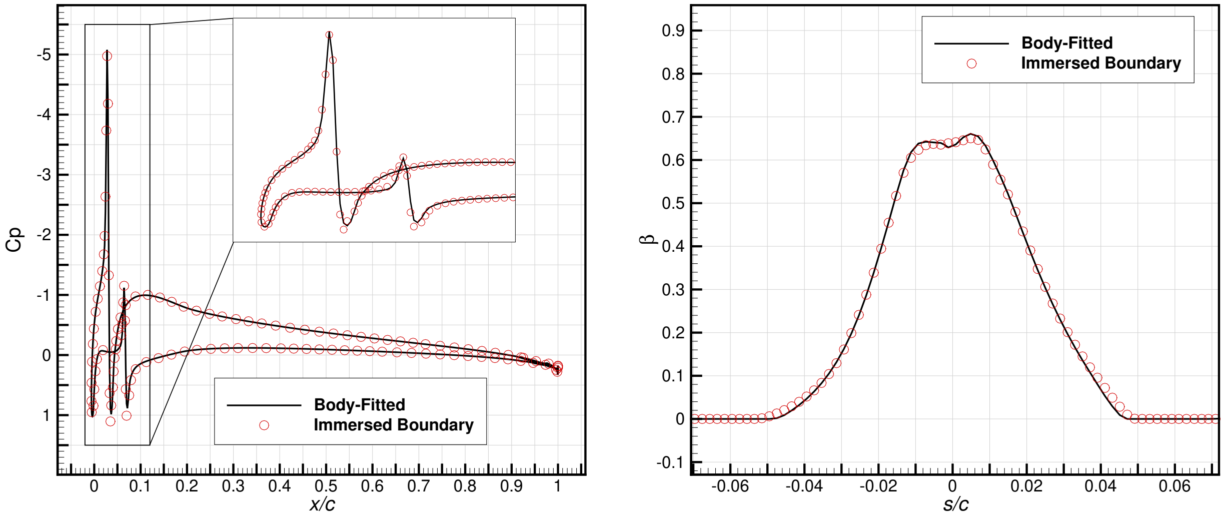


Figure 6.14 Case 242. Comparison of the heat transfer coefficient distribution produced by the present method, IGLOO2D and the participants of the IBM [86].

volume mesh is employed (see Fig. 6.9). Figure 6.15 compares the pressure coefficient, C_p, and the collection efficiency, β, for both the BF and IB approaches applied to the first ice layer in a 2-step simulation. The C_p distribution is accurately captured using the IB method. Despite the complexity of the intermediate layer, this outcome was expected, as the AGCS method has demonstrated a strong ability to simulate inviscid flows in complex geometries [62]. Regarding the collection efficiency distribution, Fig. 6.15b demonstrates good agreement between the two curves, with only slight differences observed at the limits of the impingement zone and around the stagnation point. Consequently, the final ice shape displayed in Fig. 6.16 is very similar in both approaches, with only a minor difference observed at the boundaries of ice shapes at the upper and lower regions of the wing, which compensates each other to predict approximately the same total ice mass.

Again, the experimental runs of the NASA IRT [94] are used for comparison. Contrary to the rime case, the ice shape encountered bigger variations in the spanwise direction. In order to better understand the discrepancies, five repetitions of the experiment were carried out [86]. After analyzing all the experimental ice shapes, the presence of two horns were confirmed, as well as an invariant stagnation region ice thickness. The location of the horns and their sizes had significant variations. Figure 6.17a compares the predicted ice shape to experimental data. Here, all the experimental ice shapes are plotted using a grayscale, with the grayscale representing the likelihood of ice formation. The two ice horns observed in the wind tunnel experiments are not captured in our simulation, primarily due to the low number of ice



(a) Pressure coefficient

(b) Collection efficiency

Figure 6.15 Case 242, 2-step simulation. Comparison of wall surface solution on the first ice layer for the BF and IB methods.

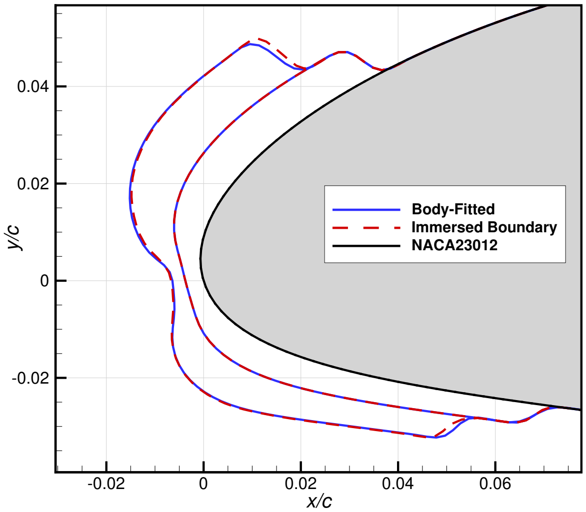
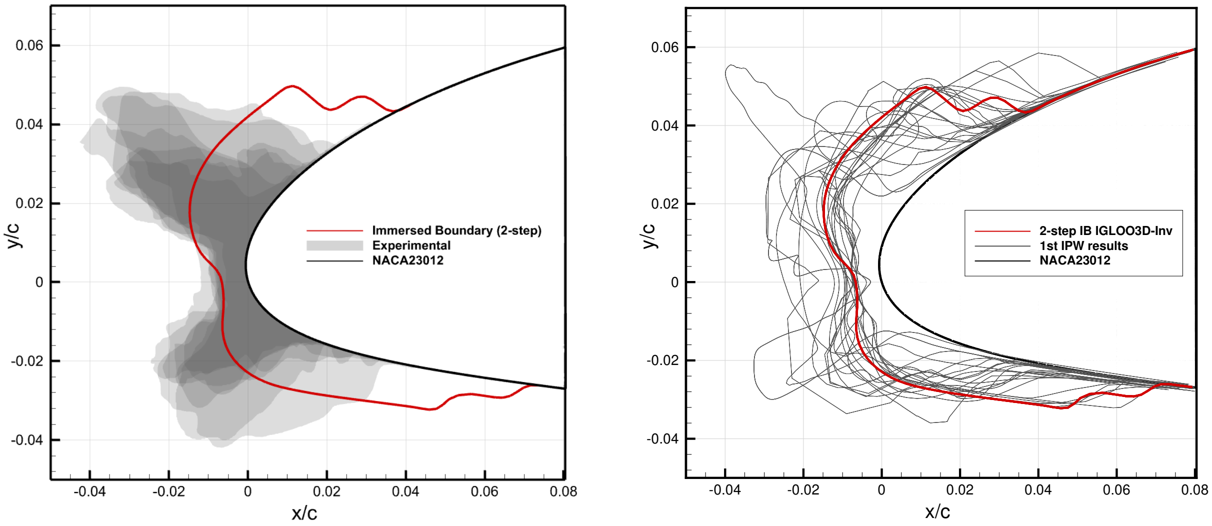


Figure 6.16 Case 242, 2-step simulation. Comparison of BF and IB ice shapes predictions.

accretion steps. However, the simulation accurately reproduces the ice thickness around the stagnation point, and the predicted total ice mass appeared consistent with experimental values. It is worth noting that this case has proven challenging, even for codes employing higher-fidelity numerical methods for airflow and droplet trajectory computations. Figure 6.17b compares the numerical ice shapes from the IPW participants, highlighting the difficulty of accurately predicting the ice horns.



(a) (b)

Figure 6.17 Case 242, 2-step IB simulation. Ice shape comparison against experimental data (a) and the results of all participants of the first IPW (b).

Improved results are expected as the number of steps employed increases. Figure 6.18 presents the final ice shape obtained from a 5-step simulation, including all the intermediate layers. Notably, the staircase-like pattern observed in the 2-step simulation has been eliminated, resulting in a more continuous ice accretion. Simulations with a higher number of steps, however, fail to accurately predict the ice shape. This is primarily due to the poor aerodynamic solution obtained in the region behind the horns, where flow separation is expected. A detailed review of this limitation is provided in Section 6.5.

6.4.3 Case 361

The third test case involves rime ice accretion on a NACA 0012 airfoil with a 30º sweep angle. The volume mesh of the clean wing is again used for the entire IB simulation, comprising around 2.4M cells and it is shown in Fig. 6.19. The mesh size around the leading edge is approximately $\Delta x/c = 3 \cdot 10^{-3}$, maintained over a distance of 10% of the chord length from the wall. The farfield extends around 20 chords in all directions, with a symmetry boundary condition enforced at the wing-root $y = 0$ plane. The three-dimensional wing is discretized with 25,664 faces. Again, the BF meshes of the intermediate layers, used for comparison purposes, maintain the same mesh parameters.

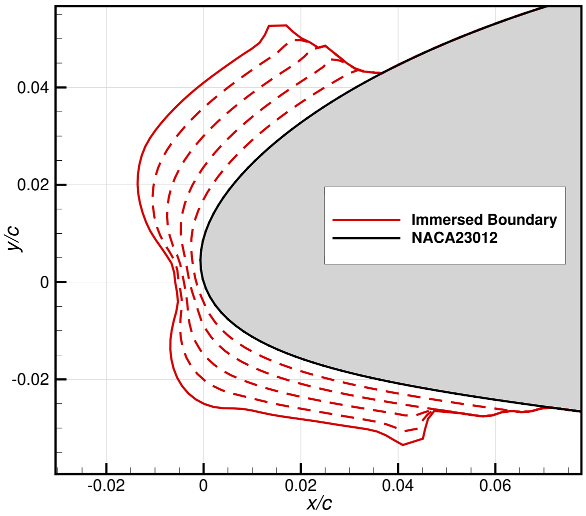


Figure 6.18 Case 242, 5-step IB simulation. Ice shape prediction and intermediate layers.

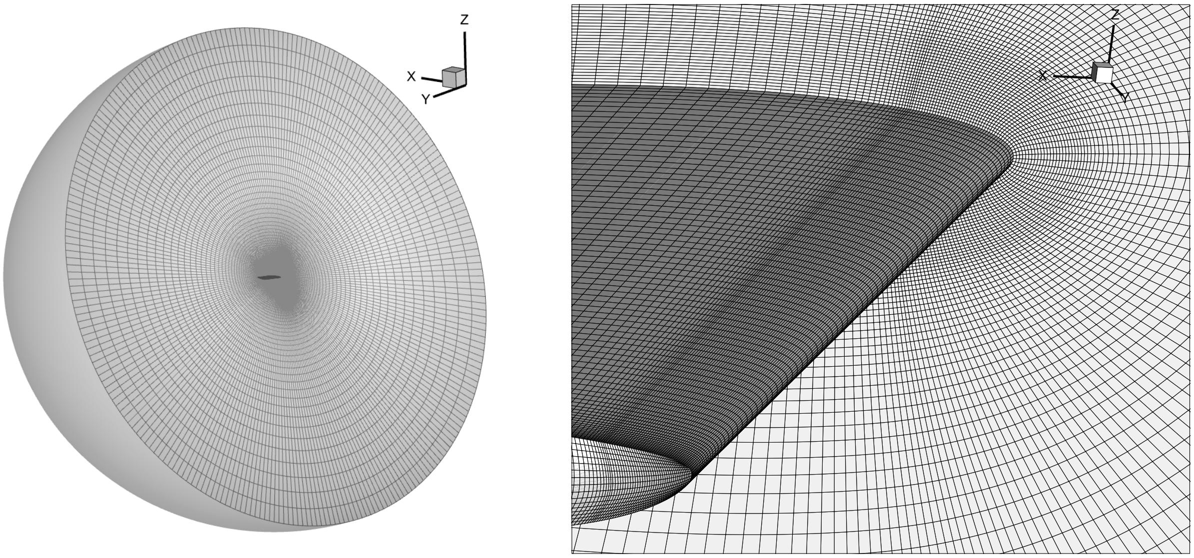


Figure 6.19 Structured mesh around a 30º sweep-angle NACA 0012 airfoil

Experimental ice shape tracings were measured at two spanwise locations by Bidwell [22] at the NASA IRT. For the numerical ice shape post-processing, ten cross-sections were extracted between the two experimental spanwise stations. Figure 6.20 illustrates the top view of the wing, showing the experimental measurement stations in red and the numerical cross-sections in black. In this section, the ice shapes are presented using the coordinate system (x', y', z'), which is derived from a 30º rotation about the y-axis from the reference coordinate system (x, y, z), as shown in Fig. 6.20.

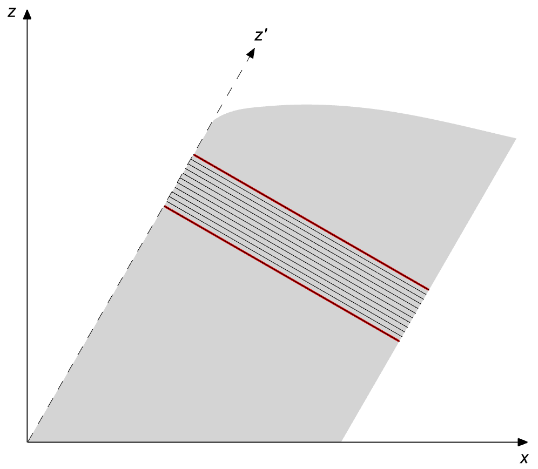


Figure 6.20 Case 361. Top view of the 30º sweep-angle NACA 0012 wing, showing the experimental measurement stations (red lines) and the numerical cross-sections (black lines).

For the clean surface geometry, the result of the simplified integral boundary layer method described in this paper are compared to the results presented by ONERA at the 1^{st} IPW[142]. These results include the heat transfer coefficient distributions using both IGLOO2D and IGLOO3D ice accretion suites. The approach used in IGLOO3D differs from both the present method and IGLOO2D. Specifically, rather than weakly coupling the inviscid flow and boundary layer solvers, h_{tc} is directly obtained from a RANS simulation of the airflow. Overall, the three codes provide the same order of magnitude, as shown in Fig. 6.21. Additionally, the good agreement between the present method and IGLOO2D, both based on simplified integral methods, further validates the presented 3D boundary layer solver. However, around the stagnation point, both IGLOO2D and the current method predict low values for the heat transfer coefficient, h_{tc}, which allows for the formation of runback and, consequently, leads to a decrease in ice thickness in this region.

This behavior has been previously observed in LEWICE3D [22], a 3D code that incorporates droplet trajectory, heat transfer, and ice shape calculations. LEWICE3D can use potential flow solutions as input, employing an integral boundary layer method to calculate the heat transfer coefficient. For swept wings, LEWICE3D also predicts a low heat transfer at the stagnation line, where the flow is considered laminar. To improve ice shape predictions, a correction to the heat transfer coefficient was introduced in this region. The proposed correction involves interpolating h_{tc} at the stagnation line between the peak values on either side. While HTC3D models the stagnation line as a turbulent region, the heat transfer remains insufficient. Rather than using a method based on interpolation as in LEWICE3D,

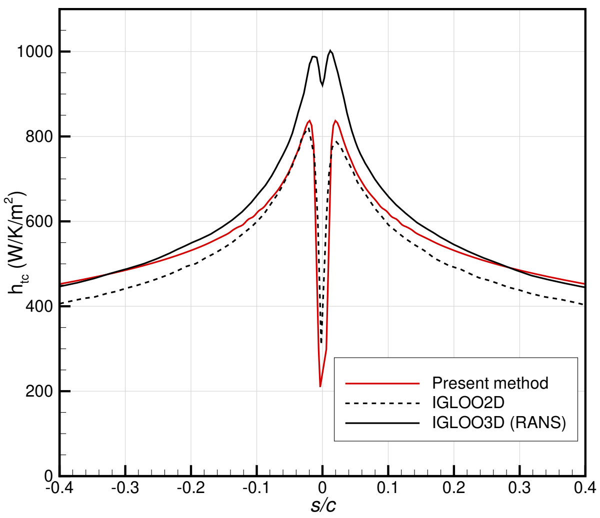


Figure 6.21 Case 361. Comparison of the heat transfer coefficient distribution produced by the present method, IGLOO2D and IGLOO3D.

a simple approach based on thresholding the h_{tc} value was used, the ambition of the article being focused more on IBMs than on fine heat transfer modeling. Then, a minimum value of 400 W/(m^{2}K) is imposed around the stagnation line to enhance the ice shape prediction. The effect of the enhancement is presented in Fig. 6.22, where 2-step IB simulations have been performed with and without the proposed correction, illustrating its impact and the necessity of its application.

To compare the BF and IBM solution, 2-step ice accretion simulations are performed, where the heat transfer correction has been employed. The first ice layer, obtained in the same manner for both approaches, is shown in Fig. 6.23. In Fig. 6.24, the pressure coefficient and collection efficiency are compared for the first ice layer at two different cross-sections. For simplicity, the collection efficiency is presented only for the lower cut. Regarding the pressure distribution, good agreement is observed between both methods; however, the IB solution shows higher values at the suction peak. Slight differences are also observed in the collection efficiency, particularly at the limit of the impingement zone, around $x'/c = 0.1$ in Fig. 6.23. These discrepancies are expected to decrease as the mesh is refined.

The ice shapes produced by both the BF and IB approaches are shown in Fig. 6.25, alongside the experimental values. The ice thickness at the stagnation line remains constant in the spanwise direction and closely matches the experimental data.

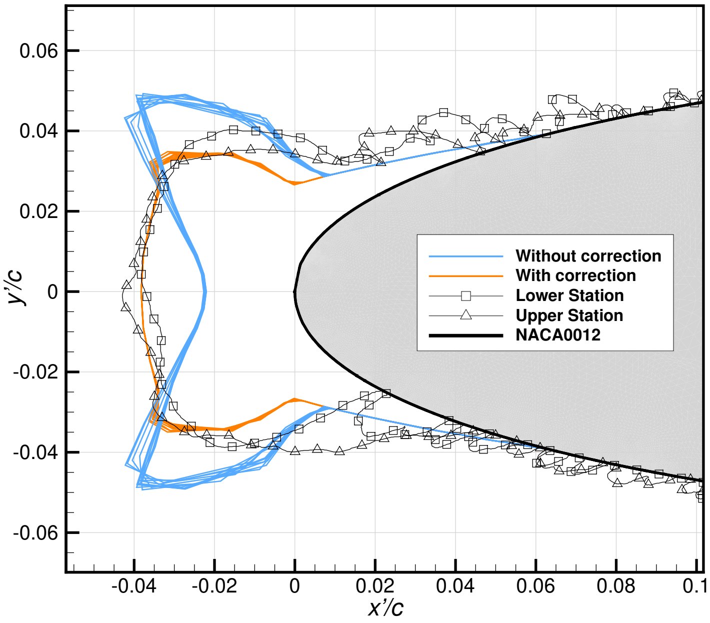


Figure 6.22 Case 361, 2-step IB simulation. Comparison of ice shapes predictions with and without the stagnation line correction.

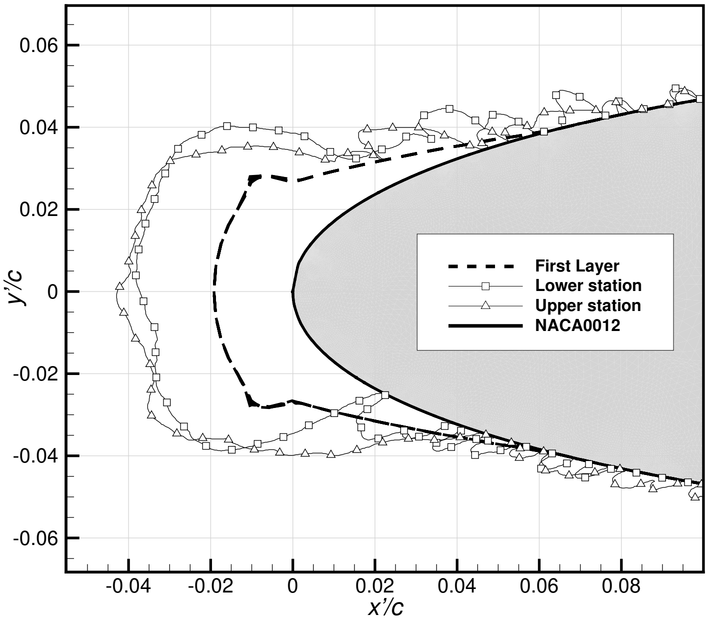
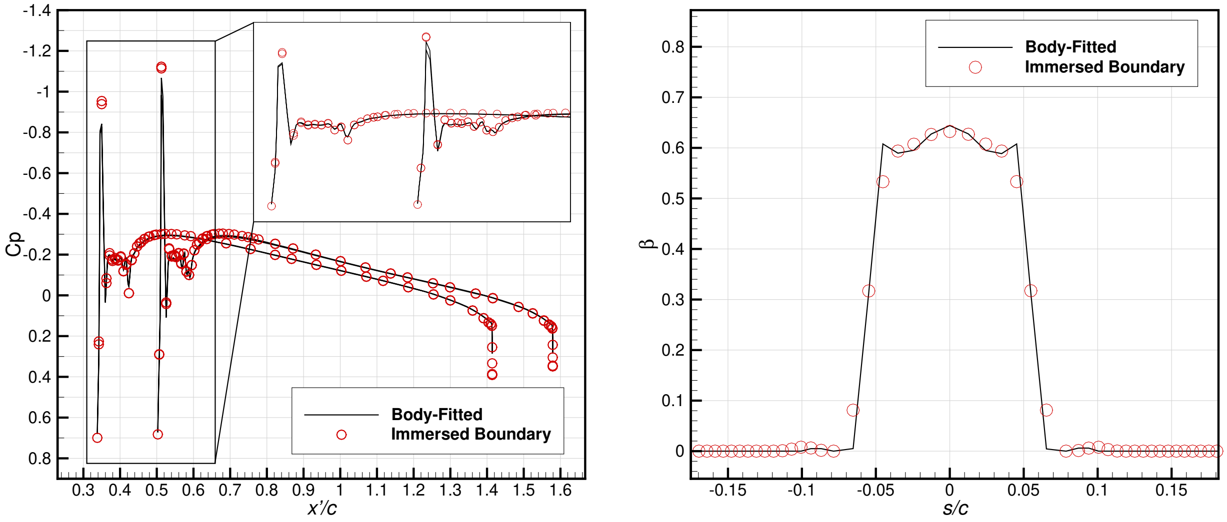


Figure 6.23 Case 361, 2-step simulation. Intermediate ice layer.



(a) Pressure coefficient

(b) Collection efficiency

Figure 6.24 Case 361, 2-step simulation. Comparison of wall surface solution on the first ice layer for the BF and IB methods.

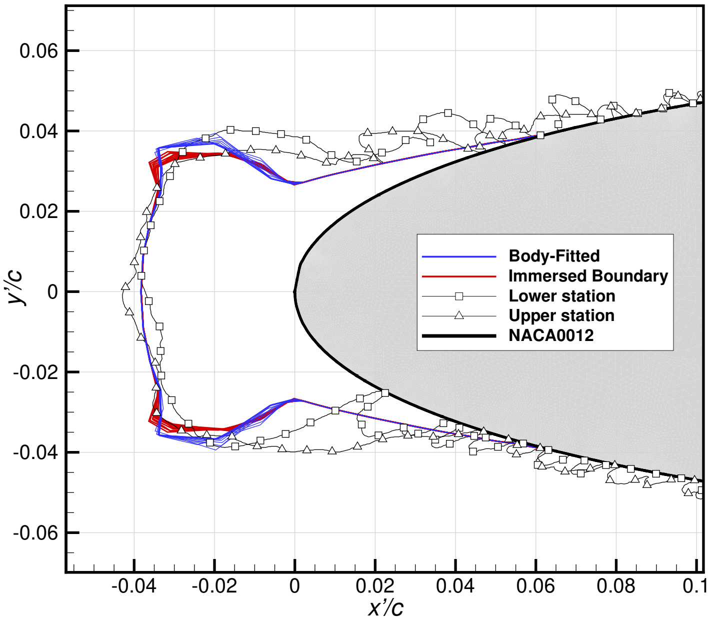
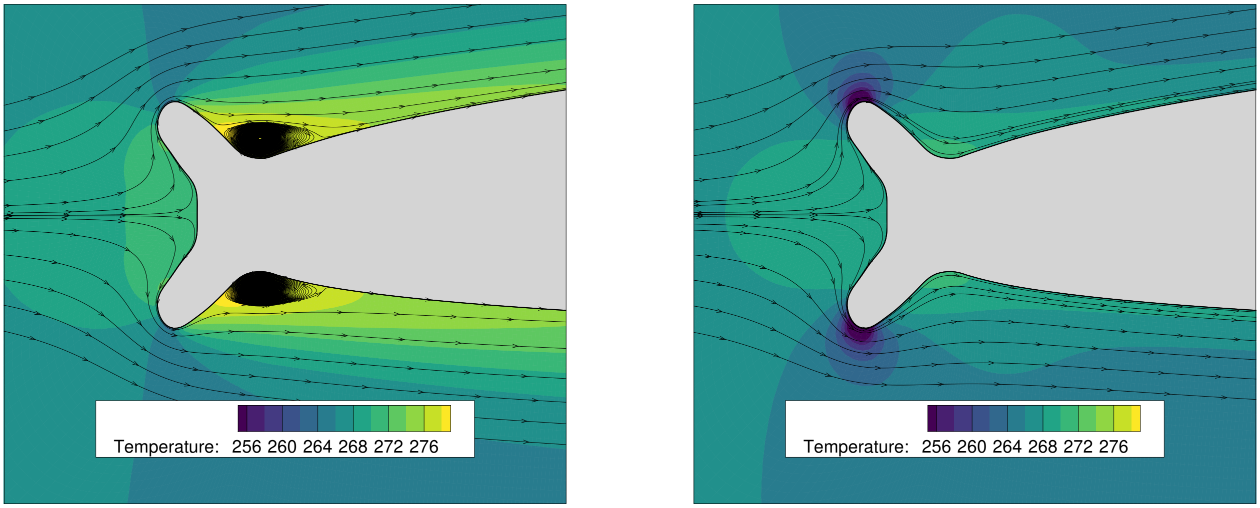


Figure 6.25 Case 361, 2-step simulation. Comparison of BF and IB ice shapes predictions.

6.5 Limitations of the Present Methodology

Inviscid flow simulations provide an idealized representation of fluid dynamics by neglecting viscous effects. However, when applied to complex geometries or multi-element airfoils, the accuracy and physical relevance of these solutions become a subject of discussion. This section focuses on better understanding the solution accuracy in complex geometries, where flow separation is commonly observed in real flows. Figure 6.26 illustrates the temperature field and streamlines around the leading edge of a 2D manufactured ice shape, comparing results obtained using a first-order scheme (Fig. 6.26a) and a second-order MUSCL extrapolation (Fig. 6.26b).



(a) 1^{st}-order

(b) 2^{nd}-order

Figure 6.26 Temperature field and streamlines around the leading edge of the 2D manufactured ice shape. Convective fluxes are computed (a) without and (b) with MUSCL reconstruction.

The first-order solution introduces significant artificial viscosity, numerically induced by sharp gradients, and leads to the formation of a spurious recirculation zone downstream of the horns. This behavior is not representative of the true Euler solution but produces a streamline pattern that more closely resembles experimental observations where physical viscosity is present. In contrast, the second-order MUSCL reconstruction minimizes numerical dissipation which in turn better resolves the expected inviscid solution of the Euler equations. However, this leads to the emergence of steep pressure and temperature gradients at the top of the ice horns, which, although not physically realistic, are consistent with the formulation of inviscid flows.

The horn ice shapes are typical ice shapes obtained in so-called glaze-ice or mixed-ice conditions. Under these conditions, and when performing multi-step simulations, the presented methodology may fail if some water runback is expected at the top of horns. Specifically, if a complex shape is obtained as an intermediate layer, the Euler solution predicts extremely low pressure and temperature downstream the ice horns. As a result, this region acts as an intensified runback blockage point, causing any incoming liquid water to freeze instantly and leading to the formation of what we define as an inviscid-driven ice horn.

It is important to note that inviscid flow solvers coupled with boundary layer methods (one-way coupling) remain a standard approach in modern ice accretion codes (e.g., LEWICE, IGLOO2D) for both rime and glaze ice conditions. The present methodology extends this well-established framework to three-dimensional cases, using immersed boundary methods. However, in cases involving extremely complex geometries, such as case 242 from 1^{st} IPW, this approach may exhibit limitations.

For instance, consider case 242, analyzed in Section 6.4.2. In the 2-step simulation, the intermediate ice layer, presented in Fig. 6.16, exhibits a distinct discontinuity at both the upper and lower regions. This corresponds to two suction peaks, with $C_p \approx -5$ and $C_p \approx -1.3$, as it can be observed in Fig. 6.15a, where the temperature drops to values of 235K and 260K, respectively. As a result, any liquid water reaching these points would freeze immediately. However, in the 2-step simulation, this does not occur, as no liquid water reaches these regions. In the 5-step simulation (see Fig. 6.18), blockage points at both the upper and lower limits of the ice shape become evident, and the formation of inviscid-driven ice horns begins to appear. To better observe this phenomenon, a 6-step simulation was performed. The final ice shape and the intermediate layers are presented in Fig. 6.27. Here, the inviscid-driven ice horns become more pronounced.

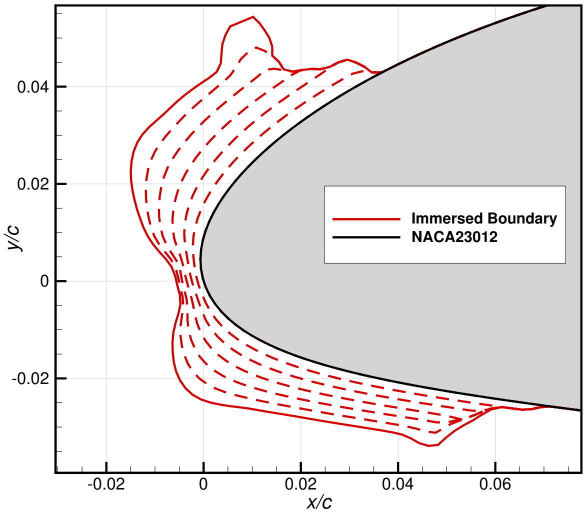


Figure 6.27 Case 242, 6-step IB simulation. Ice shape prediction and intermediate layers.

Therefore, to improve ice shape prediction under these challenging icing conditions, a mechanism to better capture the physics around the ice horns may be necessary to improve ice shape predictions. An interactive viscous/inviscid procedure is expected to address the limitations presented. While some researchers have successfully developed such techniques [42, 44], their implementation remains a challenging task, particularly due to their lack of robustness in 3D simulations.

6.6 Conclusions

In this work, we introduced an innovative framework for three-dimensional in-flight ice accretion simulations based on Immersed Boundary Methods (IBMs) within IGLOO3D, the ONERA icing suite. When performing multi-step simulations, the presented methodology effectively resolves the automation challenges commonly encountered in body-fitted simulations, notably by eliminating the re-generation of the volume mesh at each intermediate ice layer. The proposed framework integrates discrete and continuous forcing IBMs. In this setup, the aerodynamic solver employs a ghost-cell method to accurately resolve the Euler flow, while the droplet trajectory calculations rely on a penalization method, ensuring robust and reliable solutions for both aerodynamic flow and droplet dynamics under icing conditions. Additionally, developments in the boundary layer module, HTC3D, were presented. This work builds upon a simplified integral boundary layer method, enabling the modeling of heat transfer in inviscid flows.

The effectiveness of the proposed framework was demonstrated through a series of test cases from the 1^{st} IPW, encompassing both 2D and 3D configurations under rime and glaze icing conditions. In these simulations, special focus was placed on modeling heat transfer in clean surfaces. The results obtained from HTC3D were compared with those from similar methodologies and participants of the 1^{st} IPW, validating the proposed approach in various icing conditions. The ice shape results showed close agreement with those obtained from traditional body-fitted approaches, validating the accuracy and reliability of the IBM-based framework. This agreement underscores the potential of IBMs as a viable alternative to body-fitted solvers, offering comparable accuracy while significantly enhancing automation.

However, the study also identified areas for improvement. One of the main challenges is the inability to properly capture flow separation, which can significantly affect the predicted ice shapes in multi-step simulations. This highlights the need for further refinements to improve accuracy under complex icing conditions. Future work will focus on addressing this limitation, potentially by considering 3D viscous-inviscid interaction. These developments will contribute to more accurate ice accretion predictions in challenging icing scenarios.

Appendix

This appendix presents a validation case for the simplified integral boundary layer solver. It is based on Achenbach's [5] heat transfer measurements for a rough cylinder with a diameter of 0.15 m, tested at various Reynolds numbers. The roughness element height, k, is 0.9 mm, and the equivalent sand grain roughness height, k_{s}, is 1.35 mm. A detailed description of the simulation parameters can be found in [101].

The results are compared with those of Makkonen [101] and LEWICE [152], as shown in Fig. 6.28. HTC3D yields results similar to those of Makkonen, except for the simulation conducted at a Reynolds number of $\mathrm{Re} = 4.8 \cdot 10^4$, where both HTC and LEWICE predict the laminar-turbulent transition at approximately $\theta = 50^{\mathrm{o}}$.

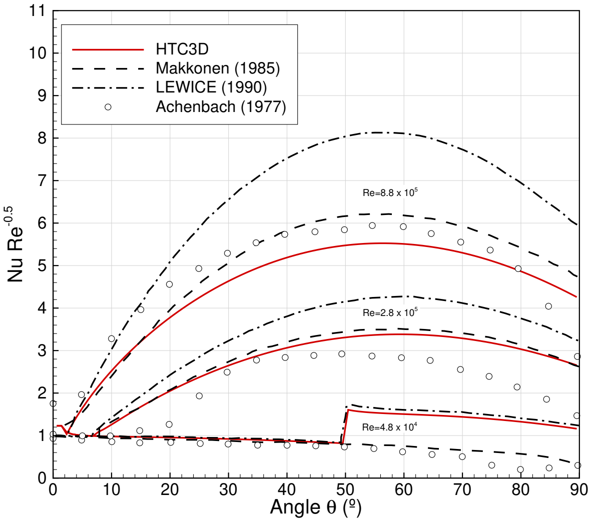


Figure 6.28 Comparison of calculated and experimental convective heat transfer on a rough cylinder.

CHAPTER 7 GENERAL DISCUSSION

The preceding chapters have presented a methodology for integrating Immersed Boundary Methods into ONERA's three-dimensional ice accretion code, IGLOO3D. The three studies constitute the core contributions of this dissertation, demonstrating that this approach accurately predicts ice accretion shapes while fully eliminating the need for volume mesh regeneration during simulations. In this manner, Chapters 4, 5, and 6 collectively address the five objectives of this research project **O.1–O.5**, as explained in Chapter 3. The proposed methodology offers a robust and efficient foundation for advancing automation in three-dimensional, multi-step ice accretion simulations.

7.1 Overview of the Proposed Methodology

As discussed in Section 2.3 (Current Use of Immersed Boundary Methods for Ice Accretion), several efforts have been made in the last decade to integrate Immersed Boundary Methods into ice accretion codes, each implementation with different mesh handling strategies, types of IBMs, and approaches to update the geometry. The present thesis contributes to this growing body of work by introducing a new IBM-based framework. This new contribution shares the same goals as those previously reviewed: to increase automation throughout the simulation process while delivering results comparable to those achieved with a BF approach. A summary of its main characteristics is provided in Table 7.1 and explained in more detail below, following the same structure used to describe the other solvers in Section 2.3.

Institution	**ONERA**
Code	IGLOO3D
Airflow	Euler
Airflow IBM	Discrete
Droplets	Eulerian
Droplets IBM	Continuous
Domain	2D/3D

Table 7.1 Overview of the key characteristics of the presented IBM-based ice accretion framework.

- Mesh generation:

  The proposed approach employs the BF mesh of the clean wing throughout the entire multi-step simulation, following the strategy introduced in the IB version of IGLOO2D.

However, unlike the two-dimensional code, this approach requires the use of structured grids. Here, O-type structured meshes are used. The mesh size is typically preserved over a distance corresponding to 10% of the chord length from the wall in order to maintain a consistent wall cell size near the IB throughout multi-step simulations.

- Airflow module:

  In this thesis, IGLOO3D computes the inviscid airflow using a weak coupling with an integral boundary layer model. Regarding the IBM, a new ghost-cell method, the AGCS method, has been developed for three-dimensional simulations in FAST and has been shown to be well suited for complex geometries. The developed IBM achieves second-order spatial accuracy and improves solution quality compared to conventional ghost-cell methods, especially on coarse meshes. Regarding the integral boundary layer computations, a simplified three-dimensional integral method has been developed to compute diffusive transfer quantities.

- Droplet trajectory module:

  Following the developments made in IGLOO2D [90, 91], a volume penalization method is used to solve the three-dimensional Eulerian droplet trajectory equations in SPIREE. Currently, the only available method is Volume Penalization Solid Source Off (VP-SSO), which shows good agreement with BF solutions.

- Ice accretion module:

  The Messinger balance is used to compute the ice thickness at the surface, as in the BF version of IGLOO3D [141].

- Geometry update module:

  A hybrid approach, based on the Lagrangian displacement of nodes following the Eulerian level-set field, is used to obtain the new ice layer at each iteration. The methodology presented in [141] for BF simulations has been adapted to IB meshes.

7.2 Main Advantages of the Proposed Methodology

The IBM-based framework introduced in this work offers several notable advantages over existing methods.

First, the developed IBM for solving the Euler equations has proven to be a well-suited approach for simulations involving complex geometries, such as ice-horn shapes encountered in glaze icing conditions. Accurately capturing wall curvature effects is essential in these simulations. Consequently, the use of CCST boundary conditions is required to enhance the accuracy of the solution. The AGCS method further improves the results, especially on coarse meshes, by using the CCST boundary conditions as a pseudo-wall function in wall regions with high curvature. The methodology reliably achieves converged solutions and successfully eliminates recirculation zones that were sometimes reported in previous studies at ONERA preceding this thesis.

Additionally, the simplified integral method used to obtain diffusive transfer quantities offers several advantages, despite its inherently two-dimensional nature. While it is not ideally suited for strongly three-dimensional configurations, it remains highly robust and easy to apply. Its simplicity and stability make it particularly attractive for preliminary studies. Furthermore, its computational cost is expected to be substantially lower than that of higher-fidelity alternatives. It is worth noting that such simplified methods are still employed in practice; for example, NASA uses a similar approach in the LEWICE3D code, where the boundary layer is solved on selected spanwise sections.

Moreover, both implemented IBMs preserve the second-order accuracy of their respective solvers. This represents a significant improvement compared to the IB version developed in IGLOO2D, presented in the doctoral thesis of Pierre Lavoie [87], where the penalization method developed in the aerodynamic solver was limited to first-order accuracy. As a result, IGLOO2D may struggle to capture small variations in the ice shape when IBM are used, restricting the total number of ice layers. In contrast, the present methodology does not impose such limitation, as evidenced by the convergence studies conducted in this thesis, where simulations with up to 64 layers were performed. This value could be further increased, as the approach implies no inherent restriction on the number of layers.

The high number of steps employed also demonstrates the robustness of the method. Typically, ice accretion codes face two major challenges related to robustness and automation: the volume mesh generation and the displacement of the surface mesh nodes from the ice thickness field. While the use of IBMs addresses the first challenge by eliminating the need for volume mesh regeneration, the geometry update process can still pose difficulties. However, the hybrid approach employed here has shown to be an effective solution, even for complex geometries, as it enables geometry updates while preventing self-intersections. Additionally, the new iced surface is explicitly known through the geometry update computation, avoiding the need for surface mesh extraction at each new ice layer.

Additionally, although computational performance is not the main concern of this thesis, the proposed framework exhibits high computational efficiency. Regarding the airflow solver, the present approach computes the inviscid flow weakly coupled with an integral boundary layer solver. As a consequence, the need of anisotropic refined meshes close to the wall is avoided and the computational cost of the simulations is reduced, compared to typical Navier-Stokes simulations. Additionally, FAST benefits from a hybrid parallelization MPI/OpenMP. In this thesis, MPI has not been used in the airflow solver for the sake of simplicity; however, no particular difficulty is expected if MPI is used. On the other hand, the droplet solver SPIREE only uses MPI parallelization, which has been employed in the presented simulations to reduce the computational cost. As an illustration, in the three-dimensional ice accretion simulations presented in Chapter 6, run on structured meshes with approximately 2.5 million cells, each iteration of the multi-step simulation requires slightly over one hour of wall-clock time when executed on a single node (44 cores) equipped with Skylake Intel® Xeon® Gold 6130 CPUs, available on ONERA's supercomputer, SPIRO. To provide further insight into the computational cost of the simulation, Fig. 7.1 shows the relative cost of each module.

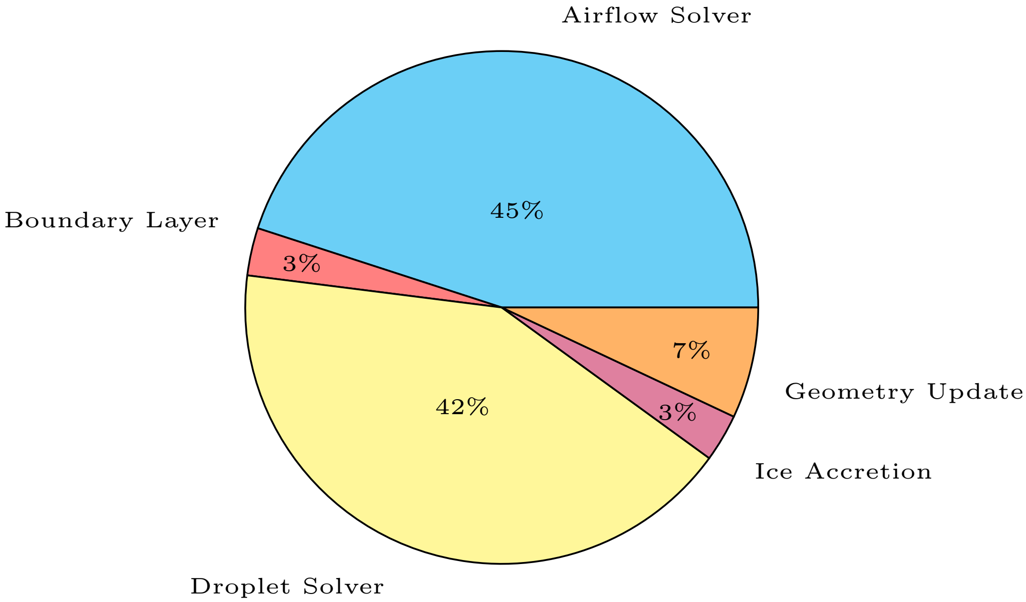


Figure 7.1 Relative computational cost of each module in the proposed IB-based framework.

7.3 Observed Limitations and Recommendations

As previously discussed, the proposed IB-based ice accretion framework still presents certain limitations. These are outlined below, along with potential strategies to address them.

Regarding the airflow solution, one of the main limitations lies in its inability to accurately capture the aerodynamic behavior in regions where severe flow separation are expected, such as downstream of complex ice horns typically formed under glaze icing conditions. The inviscid formulation provides an idealized representation of the flow, in which streamlines remain attached to the surface. As a result, the Euler solution tends to predict unrealistically low pressure and temperature values around the ice horns. These regions then act as artificially intensified runback blockage points, which complicates the convergence of multi-step simulations, particularly when a large number of steps is used. In order to address this problem, different solution may be considered. For instance, a strong coupling between the inviscid solution and the boundary layer computations should alleviate the encountered low pressure and temperature peaks. This alternative should improve the solution without considerably increasing the computational cost of the simulation. Another option is to shift to RANS simulations employing turbulent wall models. In this manner, the solution should be improved without requiring highly refined mesh around the wall. Immersed Boundary Method for RANS models are already developed in FAST, therefore, its integration into IGLOO3D is not expected to introduce any implementation difficulties. However, the computational cost of this approach is expected to be notably higher.

Additionally, in three-dimensional simulations, the simplified integral boundary layer method employed in this thesis tends to significantly underpredict the heat transfer coefficient near the stagnation line when compared to more accurate approaches. To compensate for this limitation, a corrective procedure is applied by thresholding the heat transfer coefficient. Although similar strategies are commonly adopted in ice accretion codes when using simplified integral solvers, they require defining an appropriate threshold value, a task that is inherently nontrivial and case-dependent. As an alternative, the full boundary layer equations could be solved, as demonstrated, for instance, in the work of Charlotte Bayeux [13]. However, implementing such an approach involves considerable complexity and may exhibit robustness issues, specially for three-dimensional simulations.

The current approach imposes a constraint to preserve the mesh resolution near the wall during accretion: the mesh height in the direction normal to the wall remains unchanged over a distance of approximately 10% of the characteristic length (the chord). This strategy does not allow for the accurate modeling of larger ice shapes and increases the total number of mesh cells, especially in 3D.

Alternative strategies could be considered, based on a local hierarchical mesh adaptation around the ice shape at each ice accretion iteration. Such approaches are already implemented for performance degradation assessment due to icing in the context of local immersed boundaries on unstructured meshes using the CODA solver [122]. In our case, which relies on a structured solver, local refinement could be introduced via a local patch mesh in the vicinity of the ice shape, where the IBM is applied, using a Chimera-type approach that overlaps with the BF mesh. A Cartesian approach with multiple refinement levels applied to the entire configuration could also be considered. However, these alternatives require modifications to the droplet trajectory solver SPIREE, in order to handle overlapping meshes or AMR. An alternative would be to directly solve the droplet trajectory equations within the aerodynamic solver.

CHAPTER 8 CONCLUSION

This thesis investigates the application of Immersed Boundary Methods (IBMs) in ice accretion frameworks under Appendix C icing conditions, with the objective of evaluating their feasibility as an alternative to traditional Body-Fitted (BF) approaches, while enhancing the automation of multi-step ice accretion simulations.

The three-dimensional ice accretion suite developed at ONERA, IGLOO3D, has been adapted to integrate IBMs for the computation of aerodynamic and droplet trajectory simulations. With this approach, the volume mesh remains unchanged throughout the simulation. An outstanding feature of the proposed methodology is the use of an inviscid model for the airflow solver. Additionally, the Euler equations are weakly coupled with a boundary layer solver to provide the diffusive transport quantities required by the thermodynamic model to compute ice thickness. Furthermore, an Eulerian approach has been adopted for the droplet trajectory solver.

As a first step, an IBM method was developed to solve the Euler equations. To preserve the accuracy of the airflow solver, a ghost-cell method was selected. Solving inviscid flows over complex geometries is a challenging task, often characterized by unexpected flow recirculation and convergence difficulties. The developed Adaptive Ghost-Cell-Surrounding (AGCS) method addresses these issues by accounting for wall curvature in the boundary conditions and placing some ghost cells within the fluid domain near the solid–fluid interface in high-curvature regions. In these zones, the Curvature-Corrected Symmetry Technique (CCST) boundary conditions acts as a pseudo-wall function, enhancing the solution for both body-conforming and Immersed Boundary (IB) meshes. This new approach has been thoroughly evaluated in this thesis. First, its order of accuracy was assessed by solving the flow around a two-dimensional cylinder and a three-dimensional sphere on Cartesian meshes. Then, the method has been tested on a two-dimensional manufactured iced geometry. The study involved body-conforming and IB meshes. In the latter, the mesh is built from the clean airfoil, meaning that IBM is only applied where ice has been accreted. Compared to traditional ghost-cell methods, which place ghost cells exclusively in the solid domain, the proposed method converges faster and generates less numerical entropy. Its performance was also validated on an iced 30-degree swept wing, showing good agreement with the body-fitted approach.

As a second step, a simplified integral boundary layer solver has been specifically developed for ice accretion simulations. This choice is primarily motivated by its high robustness and by the manageable development complexity, making it feasible to implement within the scope of this thesis. The solver enables the computation of both the heat transfer coefficient and the recovery temperature from the inviscid flow solution, accounting for the distinction between laminar and rough turbulent boundary layers. It builds upon existing two-dimensional simplified boundary layer theory, extending it to three-dimensional surfaces under the assumption that the flow remains two-dimensional within the boundary layer in the streamwise direction. The approach is significantly original, as the use of FVM in three-dimensional simplified integral boundary layer solvers is rare in the literature. The developed code is validated through several test cases, including the flat plate and a two-dimensional rough cylinder, both commonly used in the validation of integral boundary layer models. In addition, selected two- and three-dimensional ice accretion cases from the 1^{st} Ice Prediction Workshop (IPW) are used to evaluate the solver's performance.

As a third step, the penalization IBM for the Eulerian droplet equations, originally developed in a previous thesis, was validated through three-dimensional simulations. The validation was carried out by comparing the IBM results with those obtained using a BF approach. To this end, several cases were analyzed, including both two- and three-dimensional ice accretion cases from the IPW.

As a fourth step, the remaining components of the ice accretion suite IGLOO3D were adapted to support IBM. This included modifications to the geometry update module, as well as the integration of the necessary inter-module connections to enable automated ice accretion simulations.

As the final step, the proposed methodology was applied to perform multi-step ice accretion simulations. For this purpose, various ice accretion cases from the 1^{st} IPW were selected. First, the performance of the methodology was evaluated using case 241, a two-dimensional NACA 23012 airfoil under rime icing conditions. A convergence study with an increasing number of steps was conducted, demonstrating that the proposed methodology does not impose limitations on the number of steps used. Additionally, the predicted ice shapes were compared with experimental results obtained at NASA's Icing Research Tunel (IRT), showing excellent agreement between numerical and experimental shapes. The same airfoil was also analyzed under glaze icing conditions in case 242, one of the most challenging cases of the 1^{st} IPW, where most participants struggled to replicate the experimental ice shapes. Although the predicted total ice mass appeared consistent with experimental values, the present method had difficulty accurately predicting the formation of ice horns. Finally, the IB version of IGLOO3D was tested on a 30-degree swept wing. The numerical results showed good agreement with both BF simulations and experimental ice shapes.

Future research directions extend beyond the scope of this thesis and offer numerous opportunities for improving and expanding the proposed methodology. Several promising avenues are identified:

- Improve the aerodynamic modeling. Refining the airflow solution in scenarios involving significant flow separation could be beneficial. This may be achieved through a stronger coupling between the inviscid flow solver and the boundary layer solver, or by incorporating higher-fidelity models such as RANS simulations. In cases where inviscid models continue to be used, improvements in the boundary layer solver will be advantageous.
- Incorporate adaptive mesh refinement techniques. Locally refining the mesh around the ice shape could improve solution accuracy while maintaining a manageable computational cost for the overall simulation. If the selected mesh refinement approach is not directly compatible with both the aerodynamic solver and the droplet trajectory solver, necessary adaptations to these solvers must be implemented to ensure proper integration and functionality.
- Evaluate the parallel performance of the current workflow. Leveraging parallelism could further reduce the time required to perform 3D ice accretion simulations, making the workflow more practical for industrial applications. Although the present methodology benefits from parallel execution, it has not yet been fully tested in this context. Conducting scalability studies could provide valuable insights into potential performance improvements and help identify any computational bottlenecks.
- Evaluate the presented methodology in complex geometries and multi-element configurations. Future work should assess the accuracy of the proposed IB-based framework in more challenging scenarios, broadening the applicability of the approach. For instance, the presented methodology could be evaluated in the High-Lift Common Research Model (CRM) or in multi-body interactions, such as wing-fuselage junctions.
- Investigate the application of the presented methodology to rotorcrafts. The low computational cost of the current workflow could be advantageous for performing ice accretion simulations in such applications. However, modifications to the volume solvers may be required to account for the unsteady behavior associated with the periodic motion of the rotor blades.

REFERENCES

[1] (2009). *Standard for verification and validation in computational fluid dynamics and heat transfer: an American national standard.* ASME V&V. The American Society of Mechanical Engineers, New York, NY, reaffirmed 2016 edition.

[2] (2024). Preliminary report of the PS-VPB aircraft accident. Technical report, Center for Investigation and Prevention of Aeronautical Accidents (CENIPA), `https://dedalo.sti.fab.mil.br/en/85259`. Accident occurred on August 9, 2024, in Vinhedo, São Paulo, Brazil, involving an ATR 72-212A (500) aircraft, registration PS-VPB.

[3] (Accessed 7 June 2025). `https://github.com/onera/Fast`.

[4] Abgrall, R., Beaugendre, H., and Dobrzynski, C. (2014). An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257:83–101, DOI: `10.1016/j.jcp.2013.08.052`.

[5] Achenbach, E. (1977). The effect of surface roughness on the heat transfer from a circular cylinder to the cross flow of air. *International Journal of Heat and Mass Transfer*, 20(4):359–369, DOI: `10.1016/0017-9310(77)90157-0`.

[6] Al-Kebsi, A. (2020). *Multi-step icing modelling in the NSMB solver.* Ph.D. thesis, Université de Strasbourg, `https://theses.hal.science/tel-03467924`.

[7] Al-Kebsi, A., Mose, R., and Hoarau, Y. (2019). Multi-step ice accretion simulation using the level-set method. pages 2019–01–1955. DOI: `10.4271/2019-01-1955`.

[8] Alexa, M. (2001). Local control for mesh morphing. In *Proceedings International Conference on Shape Modeling and Applications*, page 209–215, Genova, Italy. IEEE Computer Society, DOI: `10.1109/SMA.2001.923392`.

[9] Alliez, P., Cohen-Steiner, D., Hemmer, M., Portaneri, C., and Rouxel-Labbé, M. (2024). *3D Alpha Wrapping.* CGAL Editorial Board, 5 edition, `https://doc.cgal.org/5.6.1/Manual/packages.html#PkgAlphaWrap3`. *CGAL User and Reference Manual.*

[10] Andreykiv, A. and Rixen, D. J. (2009). Numerical modelling of electromechanical coupling using fictitious domain and level set methods. *International Journal for Numerical Methods in Engineering*, 80(4):478–506, DOI: `10.1002/nme.2636`.

[11] Angot, P., Bruneau, C.-H., and Fabrie, P. (1999). A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, DOI: `10.1007/s002110050401`.

[12] Arquis, E., Caltagirone, J.-P., et al. (1984). Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: applicationa la convection naturelle. *CR Acad. Sci. Paris II*, 299(1).

[13] Bayeux, C. (2017). *Méthode intégrale pour la couche limite tridimensionnelle - Applications au givrage.* Thèse de doctorat, Toulouse, ISAE, `https://theses.fr/2017ESAE0047`.

[14] Bayeux, C., Radenac, E., and Villedieu, P. (2019). Theory and validation of a 2D finite-volume integral boundary-layer method for icing applications. *AIAA Journal*, 57(3):1092–1112, DOI: `10.2514/1.J057461`.

[15] Beaugendre, H. and Morency, F. (2018). Penalization of the Spalart–Allmaras turbulence model without and with a wall function: Methodology for a vortex in cell scheme. *Computers & Fluids*, 170:313–323, DOI: `10.1016/j.compfluid.2018.05.012`.

[16] Beaugendre, H., Morency, F., Gallizio, F., and Laurens, S. (2011). Computation of ice shedding trajectories using Cartesian grids, penalization, and level sets. *Modelling and Simulation in Engineering*, 2011(1):274947, DOI: `10.1155/2011/274947`.

[17] Beaugendre, H., Morency, F., and Habashi, W. G. (2003). FENSAP-ICE's three-dimensional in-flight ice accretion module: ICE3D. *Journal of Aircraft*, 40(2):239–247, DOI: `10.2514/2.3113`.

[18] Benek, J., Steger, J., and Dougherty, F. (1983). A flexible grid embedding technique with application to the Euler equations. In *6th Computational Fluid Dynamics Conference Danvers*, Danvers, MA, U.S.A. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.1983-1944`.

[19] Benoit, C., Péron, S., and Landier, S. (2015). Cassiopee: A CFD pre- and post-processing tool. *Aerospace Science and Technology*, 45:272–283, DOI: `10.1016/j.ast.2015.05.023`.

[20] Berger, M. J. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, DOI: `10.1016/0021-9991(84)90073-1`.

[21] Berthoumieu, P., Déjean, B., Bodoc, V., and Alary, T. (2022). ONERA research icing wind tunnel. In *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3611`.

[22] Bidwell, C. S. (2014). Icing analysis of a swept NACA 0012 wing using LEWICE3D version 3.48. In *6th AIAA Atmospheric and Space Environments Conference*, Atlanta, GA. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2014-2200`.

[23] Blanchard, G., Bempedelis, N., Bayeux, C., Radenac, E., and Villedieu, P. (2017). 3D ice accretion modeling using an integral boundary layer method. In *7th European Conference for Aeronautics and Space Sciences*. DOI: `10.13009/EUCASS2017-400`.

[24] Blanchet, M., Zayni, M. K., and Laurendeau, E. (2024). Contribution to IPW2 by studying single and multi-layer icing in 2D, 2.5D and 3D. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-3680`.

[25] Blasius, H. (1908). Grenzschichten in flüssigkeiten mit kleiner reibung (Boundary layers in fluids with small friction). *Zeitschrift für Mathematik und Physik*, 56:1–37, `https://ntrs.nasa.gov/citations/20050028493`.

[26] Bourgault, Y., Beaugendre, H., and Habashi, W. G. (2000). Development of a shallow-water icing model in FENSAP-ICE. *Journal of Aircraft*, 37(4):640–646, DOI: `10.2514/2.2646`.

[27] Bourgault-Côté, S. (2019). *Ice interface evolution modelling algorithms for aircraft icing*. PhD thesis, Polytechnique Montréal, `https://publications.polymtl.ca/3805/`.

[28] Bourgault-Cote, S., Docampo-Sánchez, J., and Laurendeau, E. (2018). Multi-layer ice accretion simulations using a level-set method with B-spline representation. In *2018 AIAA Aerospace Sciences Meeting*, Kissimmee, Florida. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2018-1835`.

[29] Broeren, A. P., Potapczuk, M. G., Lee, S., Malone, A. M., Paul, B. P., and Woodard, B. (2016). 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*, Washington, D.C. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2016-3733`.

[30] Brown-Dymkoski, E., Kasimov, N., and Vasilyev, O. V. (2014). A characteristic based volume penalization method for general evolution problems applied to

compressible viscous flows. *Journal of Computational Physics*, 262:344–357, DOI: `10.1016/j.jcp.2013.12.060`.

[31] Brown-Dymkoski, E., Kasimov, N., and Vasilyev, O. V. (2015). A characteristic-based volume penalization method for arbitrary Mach flows around solid obstacles. In Fröhlich, J., Kuerten, H., Geurts, B. J., and Armenio, V., editors, *Direct and Large-Eddy Simulation IX*, volume 20, pages 109–115. Springer International Publishing, Cham, DOI: `10.1007/978-3-319-14448-1_15`.

[32] Bureau d'Enquêtes et d'Analyses pour la sécurité de l'aviation civile (2012). Final report on the accident to Airbus A330-203, registered F-GZCP, Air France AF 447, Rio de Janeiro - Paris, 1st June 2009. `https://www.faa.gov/sites/faa.gov/files/AirFrance447_BEA.pdf`.

[33] Capizzano, F. (2007). A compressible flow simulation system based on Cartesian grids with anisotropic refinements. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2007-1450`.

[34] Capizzano, F. (2011). Turbulent wall model for immersed boundary methods. *AIAA Journal*, 49(11):2367–2381, DOI: `10.2514/1.J050466`.

[35] Capizzano, F. (2018). Automatic generation of locally refined Cartesian meshes: Data management and algorithms. *International Journal for Numerical Methods in Engineering*, 113(5):789–813, DOI: `10.1002/nme.5636`.

[36] Capizzano, F. (2025). Results from the second ice prediction workshop by using the CIRA Cartesian method. *Journal of Aircraft*, page 1–13, DOI: `10.2514/1.C038256`.

[37] Capizzano, F., Catalano, P., Carozza, A., Cinquegrana, D., and Petrosino, F. (2022). CIRA contribution to the first AIAA ice prediction workshop. In *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3400`.

[38] Capizzano, F. and Iuliano, E. (2014). A Eulerian method for water droplet impingement by means of an immersed boundary technique. *Journal of Fluids Engineering*, 136(4):040906, DOI: `10.1115/1.4025867`.

[39] Catalani, G., Agarwal, S., Bertrand, X., Tost, F., Bauerheim, M., and Morlier, J. (2024). Neural fields for rapid aircraft aerodynamics simulations. *Scientific Reports*, 14(1):25496, DOI: `10.1038/s41598-024-76983-w`.

[40] Cebeci, T. (1974). Calculation of three-dimensional boundary layers I. Swept infinite cylinders and small cross flow. *AIAA Journal*, 12(6):779–786, DOI: `10.2514/3.49350`.

[41] Cebeci, T. and Cousteix, J. (2005). *Modeling and computation of boundary-layer flows.* Springer Berlin Heidelberg, Berlin, Heidelberg, DOI: `10.1007/3-540-27624-6`.

[42] Cebeci, T., Sedlock, D., Chang, K. C., and Clark, R. W. (1989). Analysis of wings with flow separation. *Journal of Aircraft*, 26(3):214–220, DOI: `10.2514/3.45748`.

[43] Clarke, D. K., Salas, M. D., and Hassan, H. A. (1986). Euler calculations for multielement airfoils using Cartesian grids. *AIAA Journal*, 24(3):353–358, DOI: `10.2514/3.9273`.

[44] Coenen, E., Veldman, A., and Patrianakos, G. (2001). Quasi-simultaneous viscous-inviscid interaction for three-dimensional turbulent wing flow.

[45] Constant, B., Péron, S., Beaugendre, H., and Benoit, C. (2021). An improved immersed boundary method for turbulent flow simulations on Cartesian grids. *Journal of Computational Physics*, 435:110240, DOI: `10.1016/j.jcp.2021.110240`.

[46] Dadone, A. and Grossman, B. (1994). Surface boundary conditions for the numerical solution of the Euler equations. *AIAA Journal*, 32(2):285–293, DOI: `10.2514/3.11983`.

[47] Dadone, A. and Grossman, B. (2002a). An immersed body methodology for inviscid flows on Cartesian grids. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, Reno,NV,U.S.A. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2002-1059`.

[48] Dadone, A. and Grossman, B. (2002b). An immersed body methodology for inviscid flows on Cartesian grids. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, Reno,NV,U.S.A. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2002-1059`.

[49] Dadone, A. and Grossman, B. (2004). Ghost-cell method for inviscid two-dimensional flows on Cartesian grids. *AIAA Journal*, 42(12):2499–2507, DOI: `10.2514/1.697`.

[50] Dadone, A. and Grossman, B. (2006). Ghost-cell method with far-field coarsening and mesh adaptation for Cartesian grids. *Computers & Fluids*, 35(7):676–687, DOI: `10.1016/j.compfluid.2006.01.013`.

[51] Dadone, A. and Grossman, B. (2007). Ghost-cell method for analysis of inviscid three-dimensional flows on Cartesian-grids. *Computers & Fluids*, 36(10):1513–1528, DOI: `10.1016/j.compfluid.2007.03.013`.

[52] De Rosa, D., Capizzano, F., and Cinquegrana, D. (2023). Multi-step ice accretion by immersed boundaries. Vienna, Austria. SAE Technical Paper 2023-01-1484, DOI: `10.4271/2023-01-1484`.

[53] Diebold, J. M., Broeren, A. P., and Bragg, M. (2013). Aerodynamic classification of swept-wing ice accretion. In *5th AIAA Atmospheric and Space Environments Conference*, San Diego, CA. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2013-2825`.

[54] Donizetti, A., Bellosta, T., Rausa, A., and Guardone, A. (2024). Assessment of the PoliMIce toolkit from the 2nd AIAA ice prediction workshop. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-3681`.

[55] Donizetti, A., Bellosta, T., Rausa, A., Re, B., and Guardone, A. (2023). *Numerical Simulation of In-Flight Icing by a Multi-Step Level-Set Method*, page 1–32. Springer International Publishing, Cham, DOI: `10.1007/978-3-030-64725-4_30-1`.

[56] Donizetti, A., Bellosta, T., Rausa, A., Re, B., and Guardone, A. (2025). Three-dimensional front-tracking technique for multistep simulations of aircraft icing. *Journal of Aircraft*, 62(2):328–340, DOI: `10.2514/1.C037933`.

[57] Donizetti, A., Re, B., and Guardone, A. (2021). A level-set based mesh adaptation technique for mass conservative ice accretion in unsteady simulations. In *9th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*. CIMNE, DOI: `10.23967/coupled.2021.004`.

[58] Eckert, E. R. G. (1942). Die berechnung des warmeuberganges in der laminaren grenzschicht umstromter korper. *V. D. I. Forschungsheft*, 416:1–24.

[59] Elghobashi, S. (1994). On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4):309–329, DOI: `10.1007/BF00936835`.

[60] Elices Paz, P., Radenac, E., Blanchard, G., Péron, S., Laurendeau, E., and Villedieu, P. (2024). Immersed boundary methodology for 3D multi-step ice accretion simulations. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-4161`.

[61] Elices Paz, P., Radenac, E., Péron, S., Blanchard, G., Laurendeau, E., and Villedieu, P. (2023). 3D immersed boundary methods for the calculations of droplet trajectories

towards icing application. Vienna, Austria. SAE Technical Paper 2023-01–1458, DOI: `10.4271/2023-01-1458`.

[62] Elices Paz, P., Radenac, E., Péron, S., Blanchard, G., Laurendeau, E., and Villedieu, P. (2025). Second-order ghost-cell immersed boundary methodology for inviscid 3d flows in complex geometries. *Computers & Fluids*. (Under review).

[63] European Aviation Safety Agency (EASA) (2015). Certification specifications and acceptable means of compliance for large aeroplanes CS-25. Technical report, European Aviation Safety Agency (EASA), `https://www.easa.europa.eu/en/downloads/66815/en`. Amendment 17.

[64] Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, DOI: `10.1006/jcph.2000.6484`.

[65] Falkner, V. M. and Skan, S. W. (1931). Some approximate solutions of the boundary layer equations. *Philosophical Magazine*, 12(80):865–896, DOI: `10.1080/14786443109461870`.

[66] Ferschitz, H., Wannemacher, M., Bucek, O., Knöbel, F., and Breitfuß, W. (2017). Development of SLD Capabilities in the RTA Icing Wind Tunnel. *SAE International Journal of Aerospace*, 10(1):12–21, DOI: `10.4271/2017-01-9001`.

[67] Forsyth, P. and Szilder, K. (2022). Application of the morphogenetic approach to 1st AIAA ice prediction workshop test cases. In *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3609`.

[68] Geuzaine, C. and Remacle, J. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, DOI: `10.1002/nme.2579`.

[69] Hasanzadeh, K., Laurendeau, E., and Paraschivoiu, I. (2013). Quasi-steady convergence of multistep Navier–Stokes icing simulations. *Journal of Aircraft*, 50(4):1261–1274, DOI: `10.2514/1.C032197`.

[70] Hedde, T. and Guffond, D. (1995). ONERA three-dimensional icing model. *AIAA Journal*, 33(6):1038–1045, DOI: `10.2514/3.12795`.

[71] Honsek, R., Habashi, W. G., and Aubé, M. S. (2008). Eulerian modeling of in-flight icing due to supercooled large droplets. *Journal of Aircraft*, 45(4):1290–1296, DOI: `10.2514/1.34541`.

[72] Huang, W.-X., Shin, S. J., and Sung, H. J. (2007). Simulation of flexible filaments in a uniform flow by the immersed boundary method. *Journal of Computational Physics*, 226(2):2206–2228, DOI: `10.1016/j.jcp.2007.07.002`.

[73] Iaccarino, G. and Verzicco, R. (2003). Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews*, 56(3):331–347, DOI: `10.1115/1.1563627`.

[74] Ignatowicz, K., Beaugendre, H., and Morency, F. (2023). *Numerical Simulation of In-Flight Iced Surface Roughness*, page 1–48. Springer International Publishing, Cham, DOI: `10.1007/978-3-030-64725-4_29-1`.

[75] Ingram, D., Causon, D., and Mingham, C. (2003). Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3–6):561–572, DOI: `10.1016/S0378-4754(02)00107-6`.

[76] Kalitzin, G. and Iaccarino, G. (2002). Turbulence modeling in an immersed-boundary RANS method. *Center for Turbulence Research Annual Research Briefs*, pages 415–426.

[77] Kays, W. M. and Crawford, M. E. (1980). *Convective Heat and Mass Transfer*. McGraw-Hill.

[78] Keller, H. B. and Cebeci, T. (1971). *Accurate numerical methods for boundary layer flows I. Two dimensional laminar flows*, volume 8, page 92–100. Springer Berlin Heidelberg, Berlin, Heidelberg, DOI: `10.1007/3-540-05407-3_15`.

[79] Keller, H. B. and Cebeci, T. (1972). Accurate numerical methods for boundary-layer flows. II: Two dimensional turbulent flows. *AIAA Journal*, 10(9):1193–1199, DOI: `10.2514/3.50349`.

[80] Kim, J., Kim, D., and Choi, H. (2001). An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132–150, DOI: `10.1006/jcph.2001.6778`.

[81] Kirkpatrick, M., Armfield, S., and Kent, J. (2003). A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid. *Journal of Computational Physics*, 184(1):1–36, DOI: `10.1016/S0021-9991(02)00013-X`.

[82] Kou, J., Joshi, S., Hurtado-de Mendoza, A., Puri, K., Hirsch, C., and Ferrer, E. (2022). Immersed boundary method for high-order flux reconstruction based on volume penalization. *Journal of Computational Physics*, 448:110721, DOI: `10.1016/j.jcp.2021.110721`.

[83] Kármán, T. V. (1921). Über laminare und turbulente reibung. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 1(4):233–252, DOI: `10.1002/zamm.19210010401`.

[84] Laurendeau, E., Blanchet, M., Zayni, M. K., Hann, R., Radenac, E., Mussa, I., and Pueyo, A. (2024). Summary from the 2nd AIAA Ice Prediction Workshop. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-3604`.

[85] Laurendeau, E. and Boudreau, J. (2003). Drag prediction using the Euler/Navier-Stokes code FANSC. pages 2003–01–3022. DOI: `10.4271/2003-01-3022`.

[86] Laurendeau, E., Bourgault-Cote, S., Ozcer, I. A., Hann, R., Radenac, E., and Pueyo, A. (2022). Summary from the 1st AIAA Ice Prediction Workshop. In *AIAA AVIATION 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3398`.

[87] Lavoie, P. (2021). *Immersed boundary nethods for the modeling of in-flight ice accretion.* PhD thesis, Polytechnique Montréal and ISAE, `https://publications.polymtl.ca/6577/`.

[88] Lavoie, P., Pena, D., Hoarau, Y., and Laurendeau, E. (2018). Comparison of thermodynamic models for ice accretion on airfoils. *International Journal of Numerical Methods for Heat & Fluid Flow*, 28(5):1004–1030, DOI: `10.1108/HFF-08-2016-0297`.

[89] Lavoie, P., Radenac, E., Blanchard, G., Laurendeau, E., and Villedieu, P. (2021). An improved characteristic based volume penalization method for the Euler equations towards icing applications. *Computers & Fluids*, 222:104917, DOI: `10.1016/j.compfluid.2021.104917`.

[90] Lavoie, P., Radenac, E., Blanchard, G., Laurendeau, E., and Villedieu, P. (2022a). Immersed boundary methodology for multistep ice accretion using a level set. *Journal of Aircraft*, 59(4):912–926, DOI: `10.2514/1.C036492`.

[91] Lavoie, P., Radenac, E., Blanchard, G., Laurendeau, E., and Villedieu, P. (2022b). Penalization method for Eulerian droplet impingement simulations toward icing applications. *AIAA Journal*, 60(2):641–653, DOI: `10.2514/1.J060511`.

[92] Le Touze, C., Murrone, A., and Guillard, H. (2015). Multislope MUSCL method for general unstructured meshes. *Journal of Computational Physics*, 284:389–418, DOI: `10.1016/j.jcp.2014.12.032`.

[93] Leary, W. M. (2002). *"We Freeze to Please": A History of NASA's Icing Research Tunnel and the Quest for Flight Safety.* `https://ntrs.nasa.gov/citations/20020066162`.

[94] Lee, S., Broeren, A. P., Kreeger, R. E., Potapczuk, M. G., and Utt, L. (2014). Implementation and validation of 3-D ice accretion measurement methodology. In *6th AIAA Atmospheric and Space Environments Conference*, Atlanta, GA. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2014-2613`.

[95] Liu, C. and Hu, C. (2017). An immersed boundary solver for inviscid compressible flows. *International Journal for Numerical Methods in Fluids*, 85(11):619–640, DOI: `10.1002/fld.4399`.

[96] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, page 163–169. ACM, DOI: `10.1145/37401.37422`.

[97] Loth, E. (2008). Drag of non-spherical solid particles of regular and irregular shape. *Powder Technology*, 182(3):342–353, DOI: `10.1016/j.powtec.2007.06.001`.

[98] MacNeice, P., Olson, K. M., Mobarry, C., De Fainchtein, R., and Packer, C. (2000). PARAMESH: A parallel adaptive mesh refinement community toolkit. *Computer Physics Communications*, 126(3):330–354, DOI: `10.1016/S0010-4655(99)00501-9`.

[99] Mager, A. (1952). Generalization of boundary-layer momentum-integral equations to three-dimensional flows including those of rotating system. Technical Report 1067, NACA, `https://ntrs.nasa.gov/citations/19930090964`.

[100] Majumdar, S., Iaccarino, G., and Durbin, P. (2001). RANS solvers with adaptive structured boundary non-conforming grids. *Annual Research Briefs*.

[101] Makkonen, L. (1985). Heat transfer and icing of a rough cylinder. *Cold Regions Science and Technology*, 10(2):105–116, DOI: `10.1016/0165-232X(85)90022-9`.

[102] Makkonen, L. J. and Stallabrass, J. R. (1984). *Ice accretion on cylinders and wires.* DOI: `10.4224/40002646`.

[103] Martins, J. R. (2022). Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, 239:105391, DOI: `10.1016/j.compfluid.2022.105391`.

[104] Mavriplis, D. (1995). *Unstructured Mesh Generation and Adaptivity.* ICASE report. NASA Langley Research Center. Institute for Computer Applications in Science and Engineering [ICASE], `https://ntrs.nasa.gov/citations/19950020607`.

[105] Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, DOI: `10.2514/3.12149`.

[106] Messinger, B. L. (1953). Equilibrium temperature of an unheated icing surface as a function of air speed. *Journal of the Aeronautical Sciences*, 20(1):29–42, DOI: `10.2514/8.2520`.

[107] Meyer, M., Devesa, A., Hickel, S., Hu, X., and Adams, N. (2010). A conservative immersed interface method for large-eddy simulation of incompressible flows. *Journal of Computational Physics*, 229(18):6300–6317, DOI: `https://doi.org/10.1016/j.jcp.2010.04.040`.

[108] Mingione, G. and Brandi, V. (1998). Ice accretion prediction on multielement airfoils. *Journal of Aircraft*, 35(2):240–246, DOI: `10.2514/2.2290`.

[109] Mittal, R. and Iaccarino, G. (2005). Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, DOI: `10.1146/annurev.fluid.37.061903.175743`.

[110] Mochel, L., Weiss, P., and Deck, S. (2014). Zonal immersed boundary conditions: Application to a high-Reynolds-number afterbody flow. *AIAA Journal*, 52(12):2782–2794, DOI: `10.2514/1.J052970`.

[111] Mohd-Yusof, J. (1997). Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. In *Center for Turbulence Research Annual Research Briefs*, pages 317–327.

[112] Montreuil, E., Chazottes, A., Guffond, D., Murrone, A., Caminade, F., and Catris, S. (2009). ECLIPPS: 1. three-dimensional CFD prediction of the ice accretion. In *1st AIAA Atmospheric and Space Environments Conference*, San Antonio, Texas. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2009-3969`.

[113] Morelli, M., Bellosta, T., Donizetti, A., and Guardone, A. (2022). Assessment of the PoliMice toolkit from the 1st AIAA Ice Prediction Workshop. In *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3307`.

[114] Muralidharan, B. and Menon, S. (2016). A high-order adaptive Cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies. *Journal of Computational Physics*, 321:342–368, DOI: `https://doi.org/10.1016/j.jcp.2016.05.050`.

[115] Murrone, A. and Villedieu, P. (2011). Numerical Modeling of Dispersed Two-Phase Flows. *AerospaceLab J.*, 2.

[116] Musker, A. J. (1979). Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer. *AIAA Journal*, 17(6):655–657, DOI: `10.2514/3.61193`.

[117] Myers, T. G. (2001). Extension to the Messinger model for aircraft icing. *AIAA Journal*, 39(2):211–218, DOI: `10.2514/2.1312`.

[118] Nakahashi, K. (2011). Immersed boundary method for compressible Euler equations in the Building-Cube method. In *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2011-3386`.

[119] National Archives and Records Administration (1972). Mexico City: Streets and Transportation - Mexico City, Mexico. `https://catalog.archives.gov/id/354817201`. National Archives Identifier: 354817201.

[120] National Transportation Safety Board (NTSB) (1996). Aircraft accident report: inflight icing encounter and loss of control simmons airlines, d.b.a. American Eagle flight 4184 Avions de Transport Regional (ATR) Model 72-212, N401AM Roselawn, Indiana October 31, 1994. Tech. Rep. NTSB/AAR-96/01, NTSB.

[121] Naumann, Z. and Schiller, L. (1935). A drag coefficient correlation. *Zeitschrift des Vereins Deutscher Ingenieure*, 77:318–323.

[122] Nkenfack Soppi, R., Basile, F., Péron, S., Lagha, M., and Hammani, I. (2024). Local immersed boundary method on body-conformal grids for the capture of geometrical features in aerodynamic simulations. In *9th European Congress on Computational Methods in Applied Sciences and Engineering*. CIMNE, DOI: `10.23967/eccomas.2024.122`.

[123] Oleskiw, M. M. (2014). A review of 65 years of aircraft in-flight icing research at NRC. *Canadian Aeronautics and Space Journal*, `https://nrc-publications.canada.ca/eng/view/object/?id=6d705def-10da-4d98-baab-8fed31d78035`.

[124] Osher, S., Fedkiw, R., and Piechor, K. (2004). Level set methods and dynamic implicit surfaces. *Applied Mechanics Reviews*, 57(3):B15–B15, DOI: `10.1115/1.1760520`.

[125] Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, DOI: `10.1016/0021-9991(88)90002-2`.

[126] Ozcer, I. A. and Moula, G. (2024). Numerical investigations into swept-wing ice accretion rate variations with time. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-3684`.

[127] O'Brien, A. and Bussmann, M. (2018). A volume-of-fluid ghost-cell immersed boundary method for multiphase flows with contact line dynamics. *Computers & Fluids*, 165:43–53, DOI: `10.1016/j.compfluid.2018.01.006`.

[128] Papadakis, M., Wong, S.-C., Rachman, A., Hung, K. E., Vu, G. T., and Bidwell, C. S. (2007). Large and small droplet impingement data on airfoils and two simulated ice shapes. `https://ntrs.nasa.gov/citations/20070034950`.

[129] Papillon Laroche, H., Radenac, E., and Laurendeau, E. (2023). Stochastic ice accretion model using an unstructured advancing front technique. *International Journal of Multiphase Flow*, 163:104420, DOI: `10.1016/j.ijmultiphaseflow.2023.104420`.

[130] Parenteau, M., Bourgault-Cote, S., Plante, F., Kayraklioglu, E., and Laurendeau, E. (2021). Development of parallel CFD applications with the Chapel programming language. In *AIAA SciTech 2021 Forum*, VIRTUAL EVENT. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2021-0749`.

[131] Pena, D., Hoarau, Y., and Laurendeau, E. (2016). A single step ice accretion model using level-set method. *Journal of Fluids and Structures*, 65:278–294, DOI: `10.1016/j.jfluidstructs.2016.06.001`.

[132] Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. (1987). Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72(2):449–466, DOI: `10.1016/0021-9991(87)90093-3`.

[133] Peskin, C. S. (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, DOI: `10.1016/0021-9991(72)90065-4`.

[134] Pohlhausen, E. (1921). Zur näherungsweisen integration der differentialgleichung der iaminaren grenzschicht. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 1(4):252–290, DOI: `10.1002/zamm.19210010402`.

[135] Prandtl, L. (1904). Über flüssigkeitsbewegungen bei sehr kleiner reibung (On the motion of a fluid with very small viscosity). In *Verhandlungen des III. Internationalen Mathematiker-Kongresses, Heidelberg*, pages 484–491. `https://ntrs.nasa.gov/citations/19930090813`.

[136] Prandtl, L. (1910). Eine beziehung zwischen wärmeaustausch und strömungswiderstand der flüssigkeiten (On the relation between heat exchange and stream resistance of fluid flow). *Physikalische Zeitschrift*, 11:1072–1078.

[137] Preparata, F. P. and Shamos, M. I. (2008). *Computational geometry: an introduction*. Texts and monographs in computer science. Springer, New York, 6. print edition.

[138] Péron, S. and Benoit, C. (2013). Automatic off-body overset adaptive Cartesian mesh method based on an octree approach. *Journal of Computational Physics*, 232(1):153–173, DOI: `10.1016/j.jcp.2012.07.029`.

[139] Péron, S., Benoit, C., Renaud, T., and Mary, I. (2021). An immersed boundary method on Cartesian adaptive grids for the simulation of compressible flows around arbitrary geometries. *Engineering with Computers*, 37(3):2419–2437, DOI: `10.1007/s00366-020-00950-y`.

[140] Radenac, E. (2016). Validation of a 3D ice accretion tool on swept wings of the SUNSET2 program. In *8th AIAA Atmospheric and Space Environments Conference*, Washington, D.C. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2016-3735`.

[141] Radenac, E., Blanchard, G., Renaud, T., and Duchayne, Q. (2023). Workflow for predictor–corrector simulations of in-flight ice accretion, with applications on swept wings. *Engineering with Computers*, DOI: `10.1007/s00366-023-01910-y`.

[142] Radenac, E. and Duchayne, Q. (2022). IGLOO3D simulations of the 1st AIAA Ice-Prediction-Workshop database. In *AIAA Aviation 2022 Forum*, Chicago, IL & Virtual. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2022-3310`.

[143] Radenac, E., Gaible, H., Bezard, H., and Reulet, P. (2019). IGLOO3D computations of the ice accretion on swept-wings of the SUNSET2 database. pages 2019–01–1935. DOI: `10.4271/2019-01-1935`.

[144] Rangarajan, R., Lew, A., and Buscaglia, G. C. (2009). A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1513–1534, DOI: `https://doi.org/10.1016/j.cma.2009.01.018`.

[145] Rashis, B. (1953). An analytical investigation of the effect of the rate of increase of turbulent kinetic energy in the stream direction on the development of turbulent boundary layers in adverse pressure gradients. `https://ntrs.nasa.gov/citations/19930083698`.

[146] Refloch, A., Courbet, B., Murrone, A., Villedieu, P., Laurent, C., Gilbank, P., Troyes, J., Tessé, L., Chaineray, G., Dargaud, J., Quémerais, E., and Vuillot, F. (2011). Cedre software. *AerospaceLab Journal*.

[147] Renganathan, S. A., Maulik, R., and Rao, V. (2020). Machine learning for nonintrusive model order reduction of the parametric inviscid transonic flow past an airfoil. *Physics of Fluids*, 32(4):047110, DOI: `10.1063/1.5144661`.

[148] Rizzi, A. (1978). Numerical implementation of solid boundary conditions for the Euler equations. *Z.A.M.M.*, 58(7):T301–T304.

[149] Roe, P. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, DOI: `10.1016/0021-9991(81)90128-5`.

[150] Roman, F., Armenio, V., and Fröhlich, J. (2009). A simple wall-layer model for large eddy simulation with immersed boundary method. *Physics of Fluids*, 21(10):101701, DOI: `10.1063/1.3245294`.

[151] Rosenfeld, D. and Lensky, I. M. (1998). Satellite–based insights into precipitation formation processes in continental and maritime convective clouds. *Bulletin of the American Meteorological Society*, 79(11):2457–2476, DOI: `10.1175/1520-0477(1998)079<2457:SBIIPF>2.0.CO;2`.

[152] Ruff, G. A. and Berkowitz, B. M. (1990). Users manual for the NASA Lewis ice accretion prediction code (LEWICE). `https://ntrs.nasa.gov/citations/19900011627`.

[153] Saeed, F., Gouttebroze, S., and Paraschivoiu, I. (2001). Modified canice for improved prediction of airfoil ice accretion. In *48th Annual CASI Conference*, pages 283–289. `https://publications.polymtl.ca/27072/`.

[154] Saiki, E. and Biringen, S. (1996). Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450–465, DOI: `10.1006/jcph.1996.0036`.

[155] Sakurai, T., Yoshimatsu, K., Okamoto, N., and Schneider, K. (2019). Volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry. *Journal of Computational Physics*, 390:452–469, DOI: `10.1016/j.jcp.2019.04.008`.

[156] Sarofeen, C., Noack, R., and Kreeger, R. (2012). A non-cut cell immersed boundary method for use in icing simulations. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Nashville, Tennessee. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2012-1204`.

[157] Sarthou, A., Vincent, S., Caltagirone, J. P., and Angot, P. (2008). Eulerian–Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects. *International Journal for Numerical Methods in Fluids*, 56(8):1093–1099, DOI: `10.1002/fld.1661`.

[158] Schlichting, H. and Gersten, K. (2017). *Boundary-Layer Theory*. Springer Berlin Heidelberg, Berlin, Heidelberg, DOI: `10.1007/978-3-662-52919-5`.

[159] Schoenberg, I. J. (1988). *Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions*, page 3–57. Birkhäuser Boston, Boston, MA, DOI: `10.1007/978-1-4899-0433-1_1`.

[160] Serkan, O., Erhan, T., and Murat, C. (2011). Parallel computing applied to three-dimensional droplet trajectory simulation in Lagrangian approach. pages 2011–38–0106. DOI: `10.4271/2011-38-0106`.

[161] Sirianni, G., Bellosta, T., Re, B., and Guardone, A. (2024). *Numerical simulation of dispersed phase droplets impingement by a hybrid Eulerian-Lagrangian method*, page 33–59. Springer International Publishing, Cham, DOI: `10.1007/978-3-031-33845-8_28`.

[162] Smith, A. G. and Spalding, D. B. (1958). Heat transfer in a laminar boundary layer with constant fluid properties and constant wall temperature. *The Journal of the Royal Aeronautical Society*, 62(565):60–64, DOI: `10.1017/S0368393100067948`.

[163] Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., and Seidel, H.-P. (2004). Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, page 175–184, Nice France. ACM, DOI: `10.1145/1057432.1057456`.

[164] Sotiropoulos, F. and Yang, X. (2014). Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, DOI: `10.1016/j.paerosci.2013.09.003`.

[165] Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, Reno, NV, U.S.A. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.1992-439`.

[166] Starius, G. (1977a). Composite mesh difference methods for elliptic boundary value problems. *Numerische Mathematik*, 28(2):243–258, DOI: `10.1007/BF01394455`.

[167] Starius, G. (1977b). Constructing orthogonal curvilinear meshes by solving initial value problems. *Numerische Mathematik*, 28(1):25–48, DOI: `10.1007/BF01403855`.

[168] Szilder, K. and Lozowski, E. (2015). Progress towards a 3D numerical simulation of ice accretion on a swept wing using the morphogenetic approach. pages 2015–01–2162. DOI: `10.4271/2015-01-2162`.

[169] Szilder, K. and Lozowski, E. P. (2004). Novel two-dimensional modeling approach for aircraft icing. *Journal of Aircraft*, 41(4):854–861, DOI: `10.2514/1.470`.

[170] Tamaki, Y., Harada, M., and Imamura, T. (2017). Near-wall modification of Spalart–Allmaras turbulence model for immersed boundary method. *AIAA Journal*, 55(9):3027–3039, DOI: `10.2514/1.J055824`.

[171] Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95*, page 351–358. ACM Press, DOI: `10.1145/218380.218473`.

[172] Thomas, P. D. and Lombard, C. K. (1979). Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17(10):1030–1037, DOI: `10.2514/3.61273`.

[173] Thwaites, B. (1949). Approximate calculation of the laminar boundary layer. *Aeronautical Quarterly*, 1(3):245–280, DOI: `10.1017/S0001925900000184`.

[174] Tian, F.-B., Dai, H., Luo, H., Doyle, J. F., and Rousseau, B. (2014). Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems. *Journal of Computational Physics*, 258:451–469, DOI: `10.1016/j.jcp.2013.10.047`.

[175] Treece, G., Prager, R., and Gee, A. (1999). Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics*, 23(4):583–598, DOI: `10.1016/S0097-8493(99)00076-X`.

[176] Trontin, P., Blanchard, G., Kontogiannis, A., and Villedieu, P. (2017). Description and assessment of the new ONERA 2D icing suite IGLOO2D. In *9th AIAA Atmospheric and Space Environments Conference*, Denver, Colorado. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2017-3417`.

[177] Tseng, Y.-H. and Ferziger, J. H. (2003). A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, DOI: `10.1016/j.jcp.2003.07.024`.

[178] Udaykumar, H., Mittal, R., Rampunggoon, P., and Khanna, A. (2001). A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174(1):345–380, DOI: `10.1006/jcph.2001.6916`.

[179] Van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, DOI: `10.1016/0021-9991(79)90145-1`.

[180] Vargas, M., Broughton, H., Sims, J. J., Bleeze, B., and Gaines, V. (2007). Local and total density measurements in ice shapes. *Journal of Aircraft*, 44(3):780–789, DOI: `10.2514/1.23326`.

[181] Veilleux, A. E., Duchayne, Q., Radenac, E., Rouzaud, O., and Elices Paz, P. (2024). IGLOO3D simulations of the 2nd AIAA Ice-Prediction-Workshop database. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-3606`.

[182] Verzicco, R. (2023). Immersed boundary methods: Historical perspective and future outlook. *Annual Review of Fluid Mechanics*, 55(1):129–155, DOI: `10.1146/annurev-fluid-120720-022129`.

[183] Von Doenhoff and Horton (1958). A low-speed experimental investigation of the effect of a sandpaper type of roughness on boundary-layer transition. Technical Report TR 1349, National Advisory Committee for Aeronautics, `https://ntrs.nasa.gov/citations/19930092337`.

[184] Wang, L., Currao, G. M., Han, F., Neely, A. J., Young, J., and Tian, F.-B. (2017). An immersed boundary method for fluid–structure interaction with compressible multiphase flows. *Journal of Computational Physics*, 346:131–151, DOI: `10.1016/j.jcp.2017.06.008`.

[185] Wang, Z. J. and Sun, Y. (2003). Curvature-Based Wall Boundary Condition for the Euler Equations on Unstructured Grids. *AIAA Journal*, 41(1):27–33, DOI: `10.2514/2.1931`.

[186] Wieghardt, K. and Tillmann, W. (1951). On the turbulent friction layer for rising pressure. Technical Report TM 1314, National Advisory Committee for Aeronautics, `https://ntrs.nasa.gov/citations/19930093916`.

[187] Wong, M. L., Ghate, A. S., Stich, G.-D., Kenway, G. K., and Kiris, C. C. (2023). Numerical study on the aerodynamics of an iced airfoil with scale-resolving simulations.

In *AIAA SciTech 2023 Forum*, National Harbor, MD & Online. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2023-0252`.

[188] Wright, W. (2008). User's manual for LEWICE version 3.2. `https://ntrs.nasa.gov/citations/20080048307`.

[189] Wright, W. B., Gent, R. W., and Guffond, D. (1997). DRA/NASA/ONERA Collaboration on Icing Research: Prediction of Airfoil Ice Accretion - Part 2. `https://ntrs.nasa.gov/citations/19970023937`.

[190] Xie, Z. and Stoesser, T. (2020). A three-dimensional Cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies. *Journal of Computational Physics*, 416:109536, DOI: `https://doi.org/10.1016/j.jcp.2020.109536`.

[191] Yang, J. and Stern, F. (2012). A simple and efficient direct forcing immersed boundary framework for fluid–structure interactions. *Journal of Computational Physics*, 231(15):5029–5061, DOI: `10.1016/j.jcp.2012.04.012`.

[192] Yi, X. and Wang, Q. (2024). *Numerical simulation of droplet impingement by Lagrangian methods*, page 3–31. Springer International Publishing, Cham, DOI: `10.1007/978-3-031-33845-8_26`.

[193] Zabaleta, F., Jain, S. S., Bornhoft, B. J., Bose, S., and Moin, P. (2024). Large-eddy simulation of supercooled large droplets impingement using a lagrangian particle approach. In *AIAA Aviation Forum and Ascend 2024*, Las Vegas, Nevada. American Institute of Aeronautics and Astronautics, DOI: `10.2514/6.2024-4162`.

[194] Zhang, Y. and Zhou, C. H. (2014). An immersed boundary method for simulation of inviscid compressible flows. *International Journal for Numerical Methods in Fluids*, 74(11):775–793, DOI: `10.1002/fld.3872`.

[195] Zhu, C., Fu, B., Sun, Z., and Zhu, C. (2012). 3D ice accretion simulation for complex configuration basing on improved Messinger model. *International Journal of Modern Physics: Conference Series*, 19:341–350, DOI: `10.1142/S2010194512008938`.

[196] Zhu, W. J., Behrens, T., Shen, W. Z., and Sørensen, J. N. (2013). Hybrid immersed boundary method for airfoils with a trailing-edge flap. *AIAA Journal*, 51(1):30–41, DOI: `10.2514/1.J051446`.

APPENDIX A SURFACE MESH SMOOTHING TECHNIQUES FOR ICE ACCRETION SIMULATIONS

A common issue may arise if small perturbations are introduced in the geometry when performing a multi-layer simulation [28, 55]. This leads to a fluctuating collection efficiency, worsening the situation by generating a geometry with larger perturbations. Such oscillations can pose challenges as they induce significant fluctuations in the local curvature of the geometry, thereby compromising the stability of the aerodynamic solution at the wall.

This issue was encountered during the development of this thesis due to numerical oscillations, which were later resolved. Nonetheless, surface mesh smoothing techniques were investigated as a means to mitigate such behavior. In what follows, multi-step ice accretion simulations for case 241 from the 1^{st} IPW are considered. Figure A.1 illustrates the problem in an 8-step simulation where no smoothing is applied. The resulting oscillations exhibit a chaotic component. While they are barely noticeable in simulations with a small number of steps, they become increasingly pronounced as the number of steps is significantly increased.

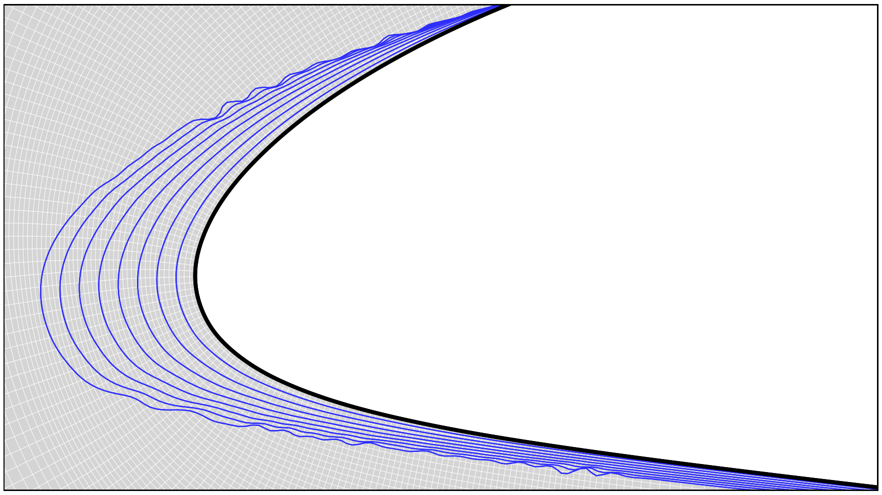


Figure A.1 Geometry evolution in an 8-step ice accretion simulation without surface smoothing, showing growing numerical oscillations.

To prevent the emergence of large oscillations, the surface mesh can be smoothed. In icing applications, it is crucial to carefully select the smoothing method, to minimize shrinkage as much as possible. Here, three types of surface smoothing are studied. These three options, already implemented in Cassiopee, are based on local approximations of the Laplacian. These methods use the differential coordinates, also called Laplacian coordinates, $\Delta\vec{x}$, to perform the mesh transformation. The Laplacian coordinates can be interpreted as the difference between the original mesh $\vec{x}$ and a smoothed version of this mesh $\vec{x}'$ [8, 163]. The discrete Laplacian of a discrete surface is usually defined by weighted averages over the neighborhoods. For a node i:

$$\Delta\vec{x}_i = \sum_{j\in i^\star} w_{i,j}(\vec{x}_j - \vec{x}_i) \tag{A.1}$$

where the weight $w_{i,j}$, are positive numbers that sum to one, and $i^\star$ represents the total number of neighboring cells. In general, these weights can be defined as a function $\varphi(\vec{x}_j, \vec{x}_i)$:

$$w_{i,j} = \frac{\varphi(\vec{x}_j, \vec{x}_i)}{\sum_{h\in i^\star} \varphi(\vec{x}_i, \vec{x}_h)} \tag{A.2}$$

In practice, the smoothed nodes are consequently derived using the following expression:

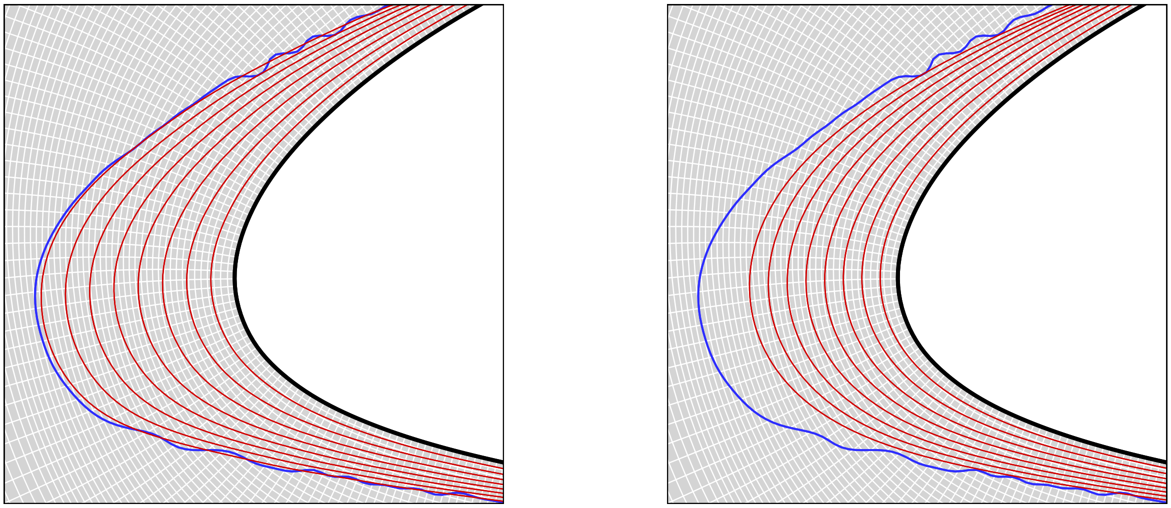
$$\vec{x}_i' = \vec{x}_i + \lambda\Delta\vec{x}_i \tag{A.3}$$

where λ is the scale factor ($0 < \lambda < 1$) [171]. Notably, if $\lambda \geq 1$, the algorithm enhances high frequencies rather than attenuating them. The previous equation can also be expressed in matrix form:

$$\vec{X}' = (I - \lambda K)\vec{X} \tag{A.4}$$

where the square matrix $K = I - W$ is defined, with I representing the identity matrix and W denoting the matrix of weights. Note that $w_{i,j} = 0$ if the node j is not a neighbor of the node i. This method, known as *Gaussian filtering*, is performed N_s times. The first two smoothing algorithms studied fall into this category.

When $\varphi(\vec{x}_j, \vec{x}_i) = 1$ is defined, an *isotropic smoothing* is performed. This type of smoothing, controlled by the parameters λ and N_s, effectively mitigates oscillations but at the cost of significant shrinkage. An inappropriate choice of the number of iterations can result in a substantial underestimation of the simulated ice amount. When applying this type of smoothing to case 241 using 10 smoothing iterations, the resulting ice shape is depicted in Fig. A.2a, where all simulation layers employing this smoothing technique are illustrated in red, juxtaposed with the final solution without smoothing. Despite noticeable shrinkage, the elimination of oscillations is evident. However, if 100 iterations are performed (Fig. A.2b), increasing the pseudo final time of the diffusion process, the obtained ice shape changes significantly, exhibiting substantial shrinkage. The choice of the parameters λ and N_s is essential in this method, frequently disrupting process automation and necessitating operator intervention.



(a) $N_s = 10$ (b) $N_s = 100$

Figure A.2 Geometry evolution in an 8-step ice accretion simulation with isotropic smoothing using $\lambda = 0.333$ (red) and without surface smoothing (blue).

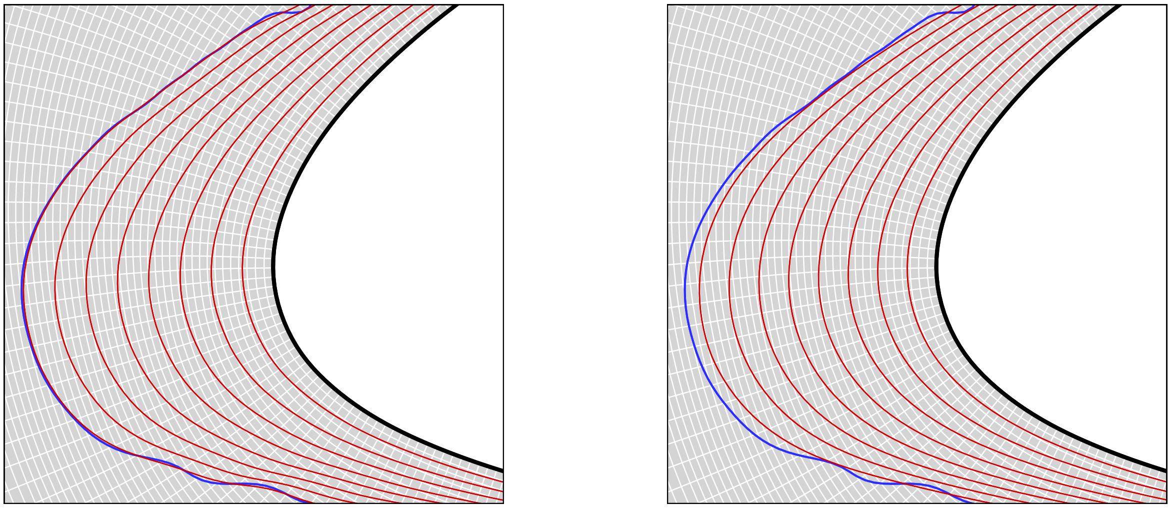
A better solution is obtained by using $\varphi(\vec{x}_j, \vec{x}_i) = \dfrac{1}{|\vec{x}_j - \vec{x}_i|}$. However, this method, known as *scale smoothing*, is less stable than the previous one and may disrupt the simulation. Once again, the examination of the influence of this smoothing technique across varying values of N_s is conducted for the icing conditions of Fig. A.1. The results are shown in Fig. A.3. In this case, the impact of the number of iterations is comparatively minor, resulting in a less noticeable shrinkage phenomenon.

One alternative to alleviate the shrinkage effect involves treating the discrete surface as signals, which allows for the application of discrete Fourier analysis. As demonstrated in [171], when a first-order neighborhood structure is employed, the matrix K possesses real eigenvalues $0 \leq k_1 \leq \ldots \leq k_n \leq 2$ and corresponding linearly independent real unit length right eigenvectors $u_1, u_2, \ldots, u_n$. These eigenvectors can be interpreted as the *natural vibration* modes of the surface, with the eigenvalues representing the associated *natural frequencies*. In this context, the term $(I - \lambda K)$ in Eq. A.4 can be considered as a *transfer function*, which can be expressed generally as $f(K)$, leading to:

$$\vec{x}' = f(K)\vec{x} = \sum_{i=1}^{n} \xi_i f(k_i) u_i \tag{A.5}$$

because $Ku_i = k_i u_i$.

A low-pass filter can be constructed to mitigate the shrinkage effect $(f(k_i)^N \approx 1$ for low



(a) $N_s = 10$ (b) $N_s = 100$

Figure A.3 Geometry evolution in an 8-step ice accretion simulation with scale smoothing using $\lambda = 0.333$ (red) and without surface smoothing (blue).

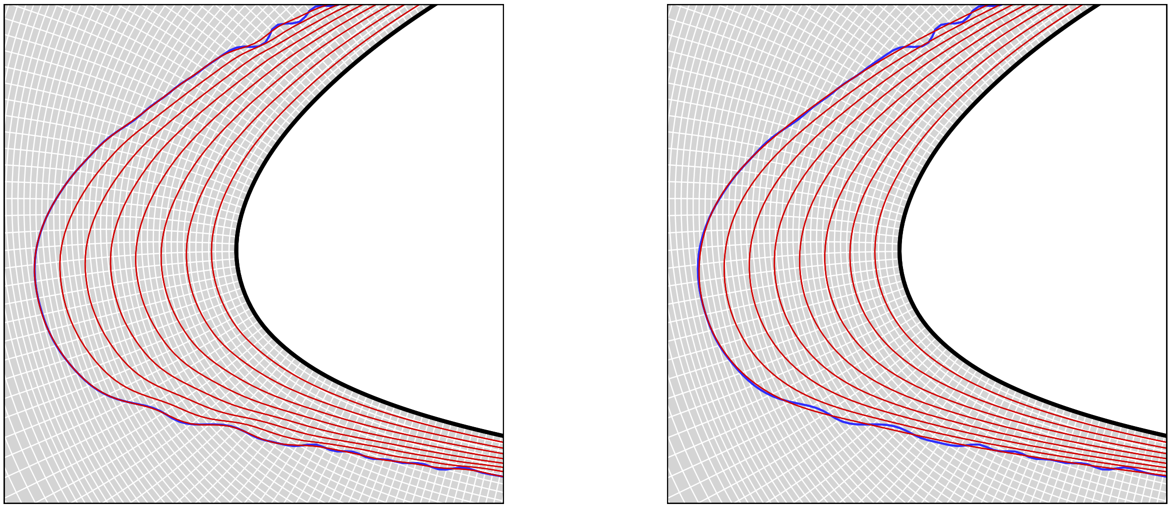
frequencies) and suppress high-frequency oscillations ($f(k_i)^N \approx 0$ for high frequencies). The transfer function utilized, as proposed in [171], is:

$$f(k) = (1 - \lambda k)(1 - \mu k) \tag{A.6}$$

where the typical choice of scale factors, $\lambda = 1/3$ and $\mu = -1/3$, are used. When λ and μ are defined, the regulation of the filter is achieved through the number of iterations N_s, which will define how fast the transfer function descends.

The influence of N_s on controlling high frequencies is evident in Fig. A.4. When $N_s = 10$ (Fig. A.4a) is employed, high-frequency oscillations (upper part) are smoothed while low-frequency oscillations (lower part) remain unaltered. Increasing the number of smoothed iterations to $N_s = 1000$ (Fig. A.4b) alters the filter configuration, expanding the stop-band region, and, therefore, smoothing the bottom part as well. Nonetheless, even with a high number of iterations, both simulations maintain the total amount of simulated ice unchanged.

Certainly, Taubin's method is well-suited for multi-layer icing simulations, facilitating process automation and allowing precise control over the smoothing extent via the filter cutoff frequency.



(a) $N_s = 10$ (b) $N_s = 1000$

Figure A.4 Geometry evolution in an 8-step ice accretion simulation with Taubin smoothing using $\lambda = 0.333$ (red) and without surface smoothing (blue).