Simulation of In-Flight Rime Icing Using a Non-Body-Fitted Mesh

by

Paul Geiger

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement of the degree of
Master of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada 2020

Author’s Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis will be made electronically available to the public.

Acknowledgements

I would like to thank professor Fue-Sang Lien, my supervisor. Throughout my research, scholarship applications, paper submissions, thesis, and seminar, you were always there to support me. I know that I never would have made it this far without your help and guidance.

Thank you to the Queen Elizabeth II Graduate Scholarship in Science and Technology program, funded jointly by the Province of Ontario and the University of Waterloo, for my scholarship. This scholarship alleviated financial burden, helping me to push forward and focus on my research.

Thank you as well to the University of Waterloo as a whole. The administrative staff has been a great help in guiding me through this process, and the environment created at the university has been one of hard work, passion, dedication, and innovation. I found my experiences here to be truly aspiring and I hope I take them with me as much as possible going forward.

Abstract

In-flight rime icing occurs when supercooled droplets strike the surface of an aircraft and instantly freeze. This often occurs in the high moisture, cold environments of clouds and is of major concern to aircraft manufacturers and pilots due to safety issues, as the icing degrades the aerodynamic performance of the plane. Rime icing models are important in their own right and as a step towards more complex glaze ice/mixed ice models.

Most current commercially available software use boundary-fitted approaches to the air and droplet phases along with a node-displacement method for the boundary. This can cause issues as the node-displacement approach to the boundary can cause tangles in the boundary which are unphysical, and the boundary-fitted methodologies require remeshing after each ice accretion step which can degrade mesh quality and cause crashes.

The goal of this thesis is to create a non-boundary-fitted approach towards rime icing simulations. Level set methodologies, which are unsusceptible to tangles, are used to handle the evolution of the boundaries. Static meshes which are Cartesian in the vicinity of the boundary are used, and high Reynolds wall functions are incorporated into the methodology so that the meshes can be made coarser and solutions can be reached faster. This proposed model is implemented via modifications of the OpenFOAM® 5.0 open-source environment. Where possible, results are compared to experimental results and to the commercially available FENSAP-ICE software by ANSYS, Inc.

Immersed boundary and cut-cell methodologies are explored for the droplets near the boundary. It was found that the developed cut-cell methodology is about twice as accurate as the immersed boundary method developed and that this is one of the dominant discretization errors in many tests.

It was shown that as droplet size goes down, particularly below 10 μm, differences in-between the FENSAP-ICE and the proposed model start increasing, with the total collection efficiency going from 0.2% different in-between the models for droplets of size 35.52 μm up to as much as 300% for 4.96 μm, and it is suggested that modifications to the model and discretization be made before tests for sub-10 μm mean-value droplet diameter (MVD) are run, although these tend to be rarely encountered in nature as most droplets tend to range from 10-50 μm. For simulation using a MVD of 16 μm and a 7-bin Langmuir-D distribution, it was found that the model differed in total collection predictions from FENSAP-ICE by 0.8%, and by 2.9% using just the MVD, as the smallest droplets contribute very little to the net collection.

Using a pseudo-steady-state, multistep icing procedure, it was found that over a NACA23012 airfoil at a 2° angle of attack, the total accreted ice for a 1-step ice accretion differed by 0.21% from FENSAP-ICE results, and for a 4-step test, it differed by 0.11% with similar resultant ice shapes when a developed model for the ending time-step which increases the global conservation was used, and by 17% for a 1-step test when it was not. Some additional numerical roughness was noted, and this was dealt with via a developed artificial diffusion model for the evolution velocity for a NACA0012 airfoil at 4° under 2 different conditions. Using this model, the roughness is reduced and for an 8-step test, the total accreted ice differs from FENSAP-ICE results by 0.17% and 2.1% depending on the conditions with similar ice shapes. FENSAP-ICE was not able to handle 16-step tests with either condition, while the proposed model was, showing the improved robustness of the proposed model.

Some important next steps were presented, including extensions required for glaze ice and for 3D simulations.

Table of Contents

1. Introduction

1.1. In-Flight Icing

In aviation, icing of the wings tends to reduce lift, increase drag, and lower the stall angle of attack of an aircraft, creating dangerous scenarios for those on board [1]. This problem, along with icing of other parts of the aircraft such as the propellers, are commonly tackled by a combination of anti-icing measures governed by ice protection certification and preflight planning, with ground de-icing procedures being available for minor icing [1]. Through simulation, it is hoped that effective and fast design decisions can be made in the future for the creation of anti-icing measures, creating safer flights for everyone.

Icing can be categorized by the temperature of the supercooled droplets. Icing is known to theoretically occur for static air temperatures as low as -40°C, and practically to occur from about +2°C to -20° in high moisture environments such as clouds. Freezing drizzle or freezing rain can also create icing, but here the average droplet size is relatively very high, up to 100x as high as seen in typical icing scenarios, and referred to as supercooled large droplets (SLD). SLD may require some special physics to model some of the droplet behaviours associated with the scale of these sizes, such as shape deformation, effectively, and thus will be avoided here-in.

In the coldest of conditions and in dilute icing scenarios, rime ice forms [2]. Rime icing tends to form conformal or wedge-shaped accretions around the object being iced [3]. In the rime icing regime, the icing occurs from the instantaneous and complete phase change of the entirety of the droplets as they strike the aircraft [3].

Warmer droplets, often at or just below the freezing point where the sensible heat required to raise the temperature up to the freezing point is not enough to overcome the latent heat the droplets must give off as freezing occurs, are categorized as glaze, or clear, ice [2]. Glaze ice is characterized by complex physics due to the thermodynamics involved in the partial freezing, and the motion of the thin water film that forms on top of the surface being iced. While some researchers have found that icing will normally present as either rime or glaze ice, a mixture of these forms, known as mixed icing, has been known to occur in which bits of rime icing present embedded in glaze ice, creating a high degree of surface roughness [2].

Rime icing will be the focus here-in as the physics being modeled is simpler, allowing for a core to be created that can support the more complex physics of a glaze/mixed ice solver being built on top of it.

1.2. Literature Review

In 1994 American Eagle Flight 4184 crashed, resulting in the deaths of all 68 people on board [4]. In last few years, following this tragedy and others like it as well as much work from academia in understanding the causes of these horrific events regulations around the world regarding in-flight icing started to change. In 2015, North American and European regulators responded with new requirements for aircraft certification under SLD conditions, with computational fluid dynamics being specifically referenced in European regulations under section *1.2.2.5 Hazard assessment* in regards to hazard identification in new designs [5] [6]. These SLD conditions have been known to cause severe roll upset and ice accretion on relatively far back portions of the wing which may previously have been unprotected by anti-icing mechanisms [2]. These issues can be fatal if not detected and responded to early via pilot action and anti-icing mechanisms, underlying the severity of these situations and the importance of the current response.

The unique challenges associated with SLD have led to a renewed interest in ice accretion modelling. As some researchers look at how to more accurately model SLD, glaze ice, and other issues, others look at the foundations of the existing models to work towards a new next generation commercial solver. While building a such a next generation solver, it is possible to implement some advancements proposed and researched over the last few years. Methods such as implicit boundary representation using level set methods have been proposed for use in recent years by researchers such as Bourgault-Côté, S., et al. (2017) [7]. This could improve robustness issues seen in other codes such as FENSAP-ICE. As will be discussed in *Section 1.3 - Thesis Goals*, these issues arise in the boundary evolution and remeshing strategies used, which are rooted in the explicit, boundary-fitted approach taken to the walls.

Existing solvers available for commercial purposes include FENSAP-ICE by ANSYS, Inc., LEWICE by NASA, and ONERA icing suites. These specific models implemented differ drastically in their basic approach, for example the older LEWICE and ONERA codes used a combination of a Lagrangian particle tracking approach for the droplets and inviscid panel codes with boundary layer corrections for handling the air [8], while the more modern FENSAP-ICE is, uses a Eulerian approach for the droplets, and a compressible Navier-Stokes-based approach for the air. These approaches have been since updated for ONERA with IGLOO2D being a tool released in 2017 to handle concerns such as SLD, and while IGLOO2D is strictly a 2D code, both ONERA and LEWICE have created codes with 3D capabilities [9] [10] [11]. Both IGLOO2D and modern LEWICE codes are still however boundary-fitted, particle-tracking, incompressible, inviscid codes with boundary layer corrections [9] [11]. Part of the reason LEWICE and ONERA use inviscid panel codes is computational limitations in the time period they were originally released in. For reference: LEWICE 1.0 was first released from NASA for use in U.S. industry and academia in May 1990 [12] although it existed within NASA prior to this, with an earlier version being discussed by Shaw, R.J. as early as 1984 [13], ONERA released its first 3D formulation in 1995 [14], and FENSAP-ICE released its first 3D formulation in 2003 [15].

The more modern FENSAP-ICE code is an FEM-based solver and uses a compressible, viscous air model and a one-way coupled, Eulerian droplet model [16]. The code can perform 3D, glaze ice, and SLD simulations [16]. When using FENSAP-ICE, it is possible evolve the solution in time using a pseudo-steady-state or an unsteady formulation [16].

A one-way coupling from the air to the droplets is employed via a drag model. This lowers coupling of the system by allowing the air and then transport equations to be solved separately with the air solution impacting the droplets but not the other way. This allows for the solution to be obtained faster/easier than for two-way or four-way coupled systems, which means systems which include droplet-air and droplet-droplet interactions. This methodology is generally agreed to be appropriate for very dilute systems, up to droplet volume fractions of about 1E-6 (liquid water contents of up to 1 g/m^3), beyond which two-way or even four-way coupling may need to be considered [3].

The pseudo-steady-state evolution methodology favoured by many solvers such as ONERA and LEWICE will be focused on here-in due to its simplicity relative to newer unsteady formulations. This methodology breaks the boundary evolution into *n* “pseudo-time-steps”. For any given boundary evolution step, the curve of droplets impinging on the curve coming from the steady-state result of the air and droplet fields over the current boundary is assumed to be a good representation of the real solution. This is to say that it is assumed that the time that it takes for the air and droplets to reach an approximate steady-state is small compared to the time-scale of a single boundary evolution step. This implies that the only truly unsteady process in the model is the boundary evolution. Progress to alternative unsteady formulations has been made in recent years, with models such as FENSAP-ICE Unsteady by Aliaga, C.N. et al (2007) [17], but many researchers still use the pseudo-steady-state formulation.

Currently, the methodologies of the largest in-use commercial codes: ONERA, LEWICE, and FENSAP-ICE all still use a boundary-fitted approach in their discretizations. Especially for older codes like LEWICE and ONERA, this may be due to history as switching to a non-body-fitted approach would require remaking the core model and therefore a complete overhaul of the software. Unfortunately, this approach can lead to robustness issues, and this thesis will look towards revisiting this approach to improve robustness of future models.

1.3. Thesis Goals

The goal of this thesis is to simulate the growth of ice with reasonable accuracy compared to the commercial software FENSAP-ICE by ANSYS, Inc. while achieving increased robustness. This thesis will look towards level set methodology to implicitly describe the boundary in its evolution and work with non-boundary-fitted approaches to the discretization to explore the possibility that next generation solvers may go in this direction.

The proposed model created here-in will work on reworking the core 2D, rime ice, non-SLD model used by, for-example, FENSAP-ICE in order to build in a new direction. The proposed model will work with a viscous model like FENSAP-ICE but will keep the model incompressible like ONERA and LEWICE codes to simplify the air-side formulations and calculations. On the droplet-side, a one-way coupled, Eulerian model will be used like the one employed by FENSAP-ICE. The model will be compared to experimental results and FENSAP-ICE results since, of the 3 popular commercially available software suites looked at, those being LEWICE by NASA, IGLOO2D by ONERA, and FENSAP-ICE by ANSYS, Inc., FENSAP-ICE is the most similar commercially available software to the proposed model.

There are issues in robustness of the current commercially available models due to issues in the boundary-fitted approach. In these approaches, a Lagrangian node-displacement approach is often used for the boundary. As seen in Fig. 1 (a) and (b), this interpretation of the boundary is susceptible to unphysical geometry clashes in the boundary which can cause crashes. After a successful boundary evolution, the mesh must be made boundary-fitted again and so a remeshing technique must be applied. Along with mesh quality degradation, the remeshing required by boundary-fitted techniques can also cause crashes as seen in Fig. 1 (c).

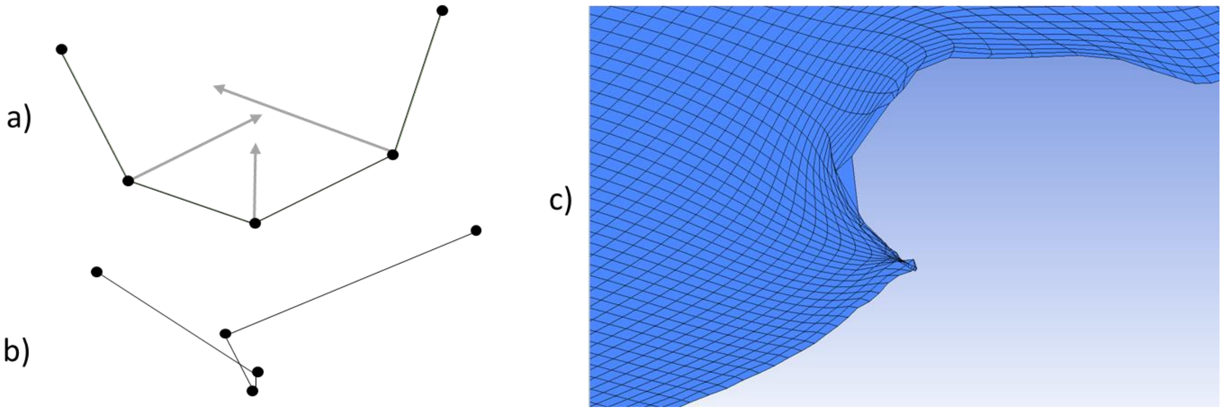


Fig. 1: a) A Lagrangian representation of a boundary before evolution. b) The boundary from (a) after evolution displaying a tangle. c) A mesh in FENSAP-ICE displaying mesh degradation and remeshing errors near the boundary after evolution.

Unlike FENSAP-ICE, the proposed model will attempt to use a non-boundary fitted approach with an implicitly described boundary to avoid issues seen in FENSAP-ICE calculations and move toward a more robust core model upon which 3D, glaze ice, unsteady, and SLD models can be built for specific applications. To keep the turbulence modelling relatively simple, stable, and efficient, a standard k-ε approach will be used and coupled with high-Reynolds wall functions. An FVM solver will be employed over FENSAP-ICE's FEM solver approach as mesh degradation which arises in FENSAP-ICE due to the boundary-fitted approach will not be an issue. The OpenFOAM® 5.0 open-source environment will be used as a basis for the model to reduce coding and validation time as several required pieces such as the linear algebra solver are already present and validated.

The level set methodology will be used to handle the geometry evolution as it is unsusceptible to geometry clashes. A static mesh will be used so that no complex remeshing algorithm is required and so that no remeshing errors can be present. Immersed boundary and cut-cell techniques will be employed around the boundary to consider the effects of the wall in nearby cells. An orthogonal mesh will be used, with the region around the boundary being a regular Cartesian mesh for simplicity of code design and high mesh quality. The commercial code FENSAP-ICE will be used along with experimental data for comparison and validation.

1.4. Overview

There are 3 major aspects of the calculations. The air phase calculations and the droplet phase calculations are broken apart via a 1-way coupling assumption allowing these 2 components to be handled in series. Finally, after the collection efficiency is extracted from the droplet calculations, a mass flux curve along the boundary is calculated and from this, the boundary is evolved. The boundary evolution is decoupled from the airflow and droplets, through the pseudo-steady-state time-stepping. This approximation says that the air and droplet fields are well approximated by their steady-state result at any given period of time in the evolution of the ice on the boundary and so the air and droplets can be calculated independently of the boundary. A basic representation of the overall algorithm is given in Fig. 2.

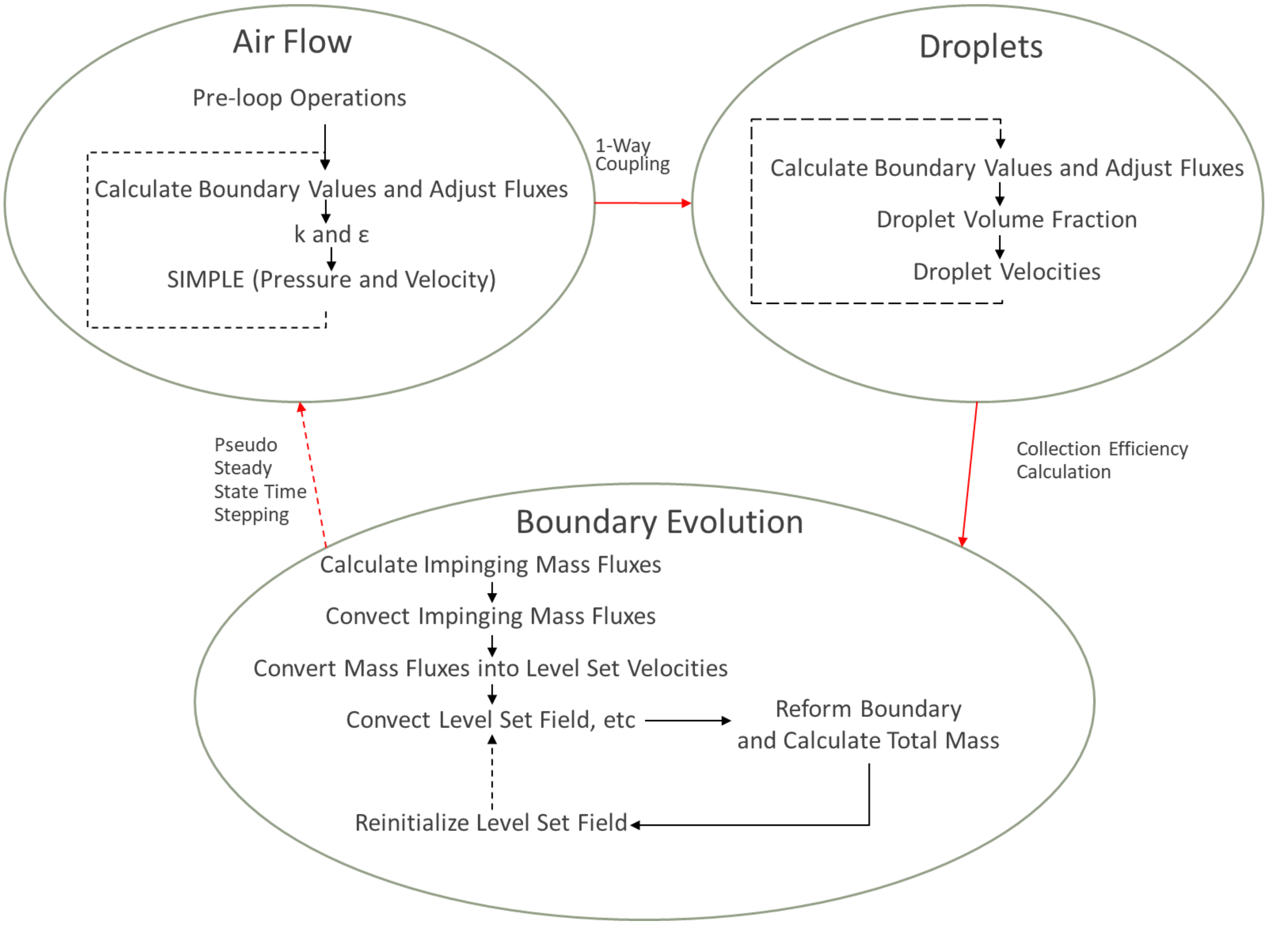

Air Flow
Pre-loop Operations
Calculate Boundary Values and Adjust Fluxes
k and ε
SIMPLE (Pressure and Velocity)
1-Way Coupling
Droplets
Calculate Boundary Values and Adjust Fluxes
Droplet Volume Fraction
Droplet Velocities
Pseudo Steady State Time Stepping
Collection Efficiency Calculation
Boundary Evolution
Calculate Impinging Mass Fluxes
Convect Impinging Mass Fluxes
Convert Mass Fluxes into Level Set Velocities
Convect Level Set Field, etc
Reform Boundary and Calculate Total Mass
Reinitialize Level Set Field

Fig. 2: Graphical representation of overall algorithmic architecture.

On the air-side, a steady-state, Newtonian, incompressible, constant air properties, RANS approach is used with a Eulerian reference frame, and a standard k-ε equation is used to model the effects of the Reynold's stresses. The governing equations are as follows:

Mass Balance:

$$\nabla \cdot \vec{U} = 0 \tag{1}$$

Momentum Balance:

$$\nabla \cdot \vec{U}\vec{U} - \vec{U}\nabla \cdot \vec{U} - \nabla \cdot \left[\left(\mu + \rho C_\mu \frac{k^2}{\epsilon}\right)\nabla \vec{U}\right] + \nabla\left[\frac{\mathrm{P}}{\rho}\right] = 0 \tag{2}$$

Transport of Turbulent Kinetic Energy:

$$\nabla \cdot \rho \vec{U} k - \frac{1}{3}\rho k \nabla \cdot \vec{U} - \nabla \cdot \rho D_k \nabla k - G_k + \rho\epsilon = 0 \tag{3}$$

Transport of Turbulent Dissipation Rate:

$$\nabla \cdot \rho \vec{U} \epsilon - \rho\epsilon\nabla \cdot \vec{U} - \nabla \cdot \rho D_\epsilon \nabla \epsilon - \frac{C_1 \epsilon}{k}(G_k + C_3 G_b) + C_2 \rho \frac{\epsilon\epsilon}{k} = 0 \tag{4}$$

Where $\vec{U}$ is the air velocity, ρ is the air density, P is the pressure, k is the turbulent kinetic energy, and ε is the turbulent dissipation rate. For more information please see *Section 2.6 - Off-Wall Air Model.*

On the droplets-side, a steady-state, one-way coupled approach is used with a Eulerian reference frame. The governing equations are as follows:

Mass Balance:

$$\nabla \cdot \alpha \overrightarrow{U_d} = 0 \tag{5}$$

Momentum Balance:

$$\nabla \cdot \alpha \overrightarrow{U_d}\overrightarrow{U_d} - \overrightarrow{U_d}\nabla \cdot \alpha \overrightarrow{U_d} + D_{Ud}\alpha\left(\overrightarrow{U_d} - \vec{U}\right) = 0 \tag{6}$$

Where α is the droplet volume fraction, $\overrightarrow{U_d}$ is the droplet velocity, $\vec{U}$ is the air velocity, and D_{Ud} is the droplet drag term. For more information please see *Section 2.8 - Droplet Model.*

To handle the presence of the boundary for air and droplet calculations given an object embedded within the mesh, the fields are broken up into 3 sections via logical masks, the dead cells which are inside the boundary, boundary cells which are surrounded by at least 1 live and 1 dead cell and have a piece of the approximate boundary inside of them, and fluid/live/far cells which are every other cell. Each of these cells are handled differently with dead cells having all fields explicitly set to 0, boundary cells being handled by immersed boundary methodology or cut-cell methodology, and the fluid cells being handled by FVM-discretized transport equations.

For algorithmic simplicity, the boundary cells are handled explicitly. It is thought that handling the boundaries of certain fields, in-particular pressure, implicitly instead of explicitly would increase the computational efficiency of the simulation. Terms such as the velocity, k, and ε would however be difficult to handle implicitly due to the non-linear interpolation scheme used to incorporate the log-law from the surrounding velocities, and the droplet quantities, concentration, and droplet velocity, would be difficult to make fully implicit because of the multiple boundary conditions used, as explained in *Section 2.7 - Near-Wall Air Model* and *2.8 - Droplet Model.*

Before the airflow loop starts some pre-loop operations are performed to better the efficiency of the algorithm. If the level set field has not been created, it is created based on a set of ordered points from an input file. The normal direction field is then calculated from the gradient of the level set field. Masks are created differentiating dead, boundary, and fluid cells. The boundary is reconstructed and cut-cell corrections to the boundary cell faces and volumes as well as corrections to the cell centres and level set field of boundary cells are calculated.

The airflow loop starts by calculating values for pressure, velocity, k, and ε for each boundary cell determined by an immersed boundary methodology described in *Section 2.7 - Near-Wall Air Model.* Face airflow calculations are modified to correctly linearly interpolate between the modified boundary cell centres and the surrounding cells. The fluid cell values for k and ε are determined via the standard k-ε two-equation model. Finally, the SIMPLE method is used to evolve the velocity and pressure fields towards a steady-state. This process is looped until the residual of k and ε go below 1E-6, the residual of the velocity goes below 5E-5, and the residual of the pressure goes below 1E-8. The relaxation coefficients used for velocity, k, and ε are 0.7, and for pressure 0.3 is used.

If IBM is being used the droplet loop starts off by explicitly calculating values for the droplet concentration and velocity of boundary cells, this. If the cut-cell methodology is used the droplet concentration and droplet velocity flux at the boundary is obtained explicitly to correct the equations here. The face fluxes of the droplet velocities are then updated to account for the change in boundary cell centres. Finally, the droplet volume fraction and then velocity equations are solved via conservation equations for the mass and momentum of the droplets respectfully. This process is looped until calculated residuals are both below 1E-6, with relaxation coefficients of 0.5 being used for both droplet concentration and velocity. After completion of this loop collection efficiency along the boundary is calculated and passed to the boundary evolution module of the algorithm.

The collection efficiency taken in by the boundary evolution module is converted into a curve of the impinging mass fluxes of droplets over the boundary. These mass fluxes are convected off the boundary in the normal direction both ways while keeping their on-boundary values constant until a steady-state is reached and then are converted into velocity values. The level set field (and pressure, velocity, etc. for improved efficiency overall) is then convected at a rate equal to the calculated velocity at each point on the mesh until an exit condition is reached. Two methods will be explored as exit conditions for the level set convection loop, one based on the simple amount of time lapsed and a total time limit, and one based instead on the total mass grown and the total initial mass impingement on the boundary times the inputted time. This second option requires the total mass of ice per unit span at any given time-step be calculable, which was done by reforming the boundary at every time-step, extracting out the cut-cell volumes and from this determining the amount of ice.

The final step of the boundary evolution is to reinitialize. Errors in the level set field can appear during evolution and the properties of the level set field are not strictly kept, in particular, that the magnitude of the gradient is 1 and that the absolute value of the field represents the minimum distance to the boundary. This last issue is particularly problematic as this property is used directly in the wall functions and IBM formulations. Reinitialization fixes this by directly addressing one or more of these issues to reset the level set field as a whole back to its intended state, usually starting from the 0-contour of the field (the boundary) and moving outwards. This is not required at every time-step and can introduce additional errors into the boundary evolution process due to imperfect discretization of the methodologies, and thus is generally only performed as a final step in the process but can be performed mid-loop if required.

2. Methodology

2.1. Level Set Field Description and Toolbox

The Level Set Methodology, first developed by Osher, S. and Sethian, J.A. and released in their 1988 journal paper "*Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*" [18], is a technique for implicit surface representation which allows for a Eulerian analysis of boundary motion and/or evolution in 2D or 3D [19].

In the original 1988 formulation of the Level Set Methodology, Osher, S. and Sethian, J.A. initialized the key field, the level set field, with the form $1 \pm d^2$ where d is the minimum distance from the boundary to the interface and the sign of ± is determined by whether it was inside or outside the interface [18]. The level set field's definition is not fixed to this function however and must only be Lipschitz continuous with a definable contour representing the boundary [19]. Often the function chosen to represent the level set field has some physical meaning and/or other features which may be useful for the rest of simulation. Since the original paper of Osher, S. and Sethian, J.A., it has become more standard to use ±d during initialization for most applications, this representation being referred to by some as the "conventional level set field", and even Osher, S. along with Fedkiw, R. in 2003 said that this is a "better choice for initializing [the level set field]" [20].

The conventional level set field, represented by the symbol ϕ, is an implicit representation of any arbitrary object or objects via the minimum distance from each cell centre on a mesh to the surface of the object, with a sign determined by whether the cell is inside the boundary (negative) or outside of the boundary (positive). This representation is associated with the following set of useful properties used here-in [20]:

1) $|\phi_{x,y}| = d_{x,y}$ *where $d_{x,y}$ is the normal/minimum distance from the point to the boundary*

2) $sign\phi_{x,y} = \frac{\phi_{x,y}}{|\phi_{x,y}|} = \begin{pmatrix} +1 & if\ point(x,y)\ is\ outside\ boundary \\ -1 & if\ point\ (x,y)\ is\ inside\ boundary \end{pmatrix}$

3) $\overrightarrow{n_{x,y}} = \frac{\nabla\phi_{x,y}}{||\nabla\phi_{x,y}||}$ *where $\overrightarrow{n_{x,y}}$ is the outward-facing normal direction to the boundary at point (x,y)*

4) $||\nabla\phi_{x,y}|| = 1$

5) *The 0 contour of the level set field represents the boundary of the object*

2.2. Level Set Field Creation

In the creation of the level set field, the object being created is assumed to be a single, connected figure in 2D. An ordered inputted set of points representing the outside curve is interpreted using the direct initialization method [20]. In this method, each cell centre is assigned the minimum distance from line segments in-between connected point pairs. A sign is associated with this distance, giving the signed-distance function which is key to the conventional level set method. Points interior to a closed curve are determined by the property that unless it passes exactly parallel to a curve, a line drawn from that point in an arbitrary direction will cross an odd number of connections, and the opposite is true for external points [20]. The convention that internal points are negative and external points are positive is adopted, which affects, for example, the direction of the normal vector obtained from the gradient of the signed-distance function.

Assuming that the line from the cell centres to the reference point is not colinear with any line segments and that the reference point is placed outside of the object, this process is handled via the following equation for each cell "c":

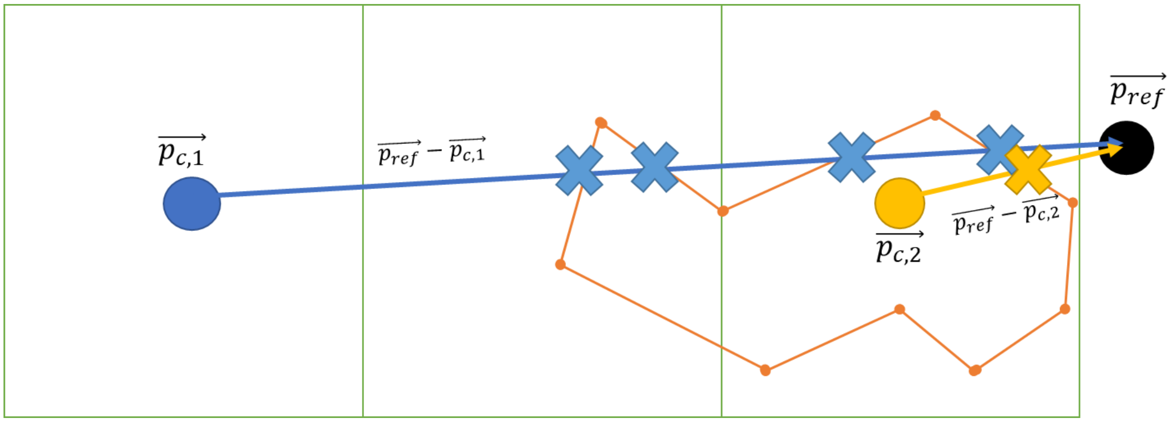
$$parallel\colon (\overrightarrow{p_c} - \overrightarrow{p_{ref}}) \times (\overrightarrow{p_j} - \overrightarrow{p_i}) = 0 \tag{7}$$

$$inBounds\colon 0 \leq \frac{(\overrightarrow{p_c} - \overrightarrow{p_i}) \times (\overrightarrow{p_j} - \overrightarrow{p_i})}{(\overrightarrow{p_c} - \overrightarrow{p_{ref}}) \times (\overrightarrow{p_j} - \overrightarrow{p_i})} \leq 1 \text{ AND } 0 \leq \frac{(\overrightarrow{p_c} - \overrightarrow{p_i}) \times (\overrightarrow{p_c} - \overrightarrow{p_{ref}})}{(\overrightarrow{p_c} - \overrightarrow{p_{ref}}) \times (\overrightarrow{p_j} - \overrightarrow{p_i})} \leq 1 \tag{8}$$

$$sign\phi_c = 1 - 2\, mod \left(\sum_{point\ pairs\ (i,j)} \begin{pmatrix} 1 & (\text{NOT } parallel) \text{ AND } inBounds \\ 0 & otherwise \end{pmatrix}, 2 \right) \tag{9}$$

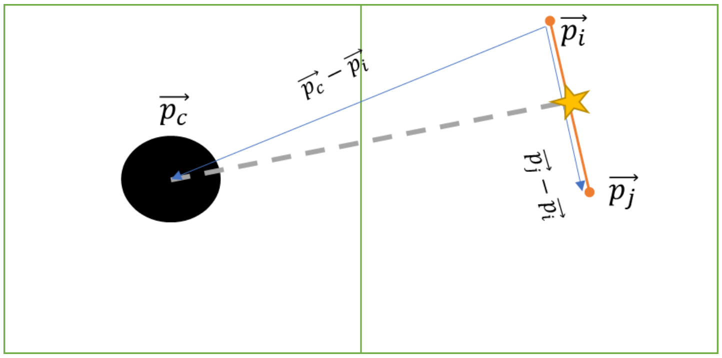
$$\phi_c = sign\phi_c * \text{min over point pairs(i, j)} \left\| \overrightarrow{p_c} - \left(\overrightarrow{p_i} + (\overrightarrow{p_j} - \overrightarrow{p_i})\, max\left(0, min\left(1, \frac{(\overrightarrow{p_j} - \overrightarrow{p_i}) \cdot (\overrightarrow{p_c} - \overrightarrow{p_i})}{\|\overrightarrow{p_j} - \overrightarrow{p_i}\|}\right)\right)\right)\right\| \tag{10}$$

Where $\overrightarrow{p_i}$ and $\overrightarrow{p_j}$ are the positions of a set of 2 neighbouring points in the figure, $\overrightarrow{p_c}$ is the position of a cell centre, and $\overrightarrow{p_{ref}}$ is an arbitrary point outside the object. These equations are accompanied by Fig. 3 which presents the concept behind the calculation of the sign of the level set field and Fig. 4 for calculation of the minimum distance from a cell centre to a line segment.



$\overrightarrow{p_{ref}}$
$\overrightarrow{p_{c,1}}$
$\overrightarrow{p_{ref}} - \overrightarrow{p_{c,1}}$
$\overrightarrow{p_{ref}} - \overrightarrow{p_{c,2}}$
$\overrightarrow{p_{c,2}}$

Fig. 3: A graphical representation for the algorithm used to find the sign of the level set field, C,1 in blue where the value would be +1 since it intersects with the shape in orange an even number of times, and C,2 in yellow where the value would be -1 since it intersects with the shape an odd number of times.



$\overrightarrow{p_i}$
$\overrightarrow{p_c} - \overrightarrow{p_i}$
$\overrightarrow{p_c}$
$\overrightarrow{p_j} - \overrightarrow{p_i}$
$\overrightarrow{p_j}$

Fig. 4: A graphical representation for the algorithm used to find the minimum distance, the length of the grey line, from a cell centre C, represented by a black circle, to a line segment in orange with terminal points i and j.

2.3. Boundary Reconstruction

In order to have the methodology able to handle ice accretion, it needs to be flexible enough to recreate the boundary after icing sub-steps. Accurate and fast boundary reconstruction allows for the cut-cell method to be appropriately put into practice, calculation of the amount of ice, and integrations over the surface for properties such as the total collection efficiency, among other useful options. To handle the reconstruction the 0 contour of the level set field must be extracted out.

From Osher, S. and Fedkiw, R. (2003), points along the boundary can be determined from a linear interpolation of the signed distance function towards 0 from points in the field in which the sign changes [20]. From a normalized linear interpolation of the normal direction vector field (the normalized gradient of the signed-distance function) to this point, a line is formed. This line is cut at the boundaries of the cell and the intersection of the line with the boundaries is marked in memory as a representation of the line. As shown in Fig. 5 a, it is possible that 2 lines are found in the same (the purple lines), in this case the terminal points of the line segments are averaged and the resulting points are re-snapped to the cell edges, resulting in the blue line. This step assumes that the boundary represented is smooth such that no horns or pinches are present. This may be problematic towards the tail of an airfoil, so the tails of airfoils will be cut. This will also cause issues in glaze icing in cases where horns and/or pinches may be present, and this step will need to be reformulated for this case if extended to glaze icing.

The terminal points of the boundary lines are then averaged again on the faces to force continuity of the boundary as shown in Fig. 5 b. The cell-centres are recalculated along with the signed-distance at the boundary containing cells.

The boundary is forced to be continuous, arguably at the cost of accuracy as the normal direction may not match the wall-normal direction, but the advantage is that in the cut-cell calculations there will be no ambiguity nor will there be a unique solution for the fluxes on either side of the faces.

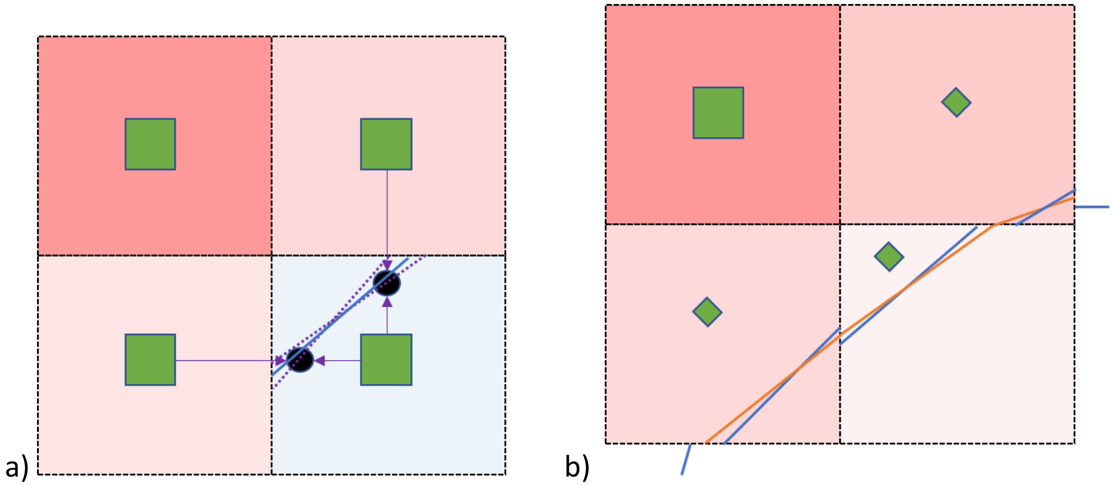


Fig. 5: Example of boundary reconstruction.

a) Phase 1. Sign of the signed-distance distance function is given by the cell colour and distance is represented by opacity. Green squares represent cell centres. Purple arrows represent linear interpolation to find points along the boundary. Black circles represent points interpolated to along the boundary. Purple lines represent test line segments coming from the black circles and the interpolated tangent directions. Blue lines represent the final line segments within Phase 1 coming from averaging of the terminal points of the purple line segments.

b) Phase 2. Sign of the signed-distance distance function is given by the cell colour and distance is represented by opacity. Green squares represent cell centres. Green diamonds represent modified cell-centres. Blue lines represent the line segments coming from Phase 1 of boundary reconstruction. The orange lines represent the final continuous boundary coming from averaging the terminal points of the blue line segments.

2.4. Cartesian Cut-Cell Method

The cartesian cut-cell method directly uses the modified cell descriptions coming from the boundary reconstruction. The reconstruction results in a set of points on the faces representing where the cut is made. In-between 2 face-lying points in a cell, a single straight line is assumed to exist.

Based on this description, the face fluxes are reduced by multiplication by a coefficient $\beta_{faceCut}$ which ranges between 0 and 1 and which is equal to the fraction of the face which is fluid, with the remaining portion being covered by the solid. This relationship can be seen in the example given in Fig. 6.

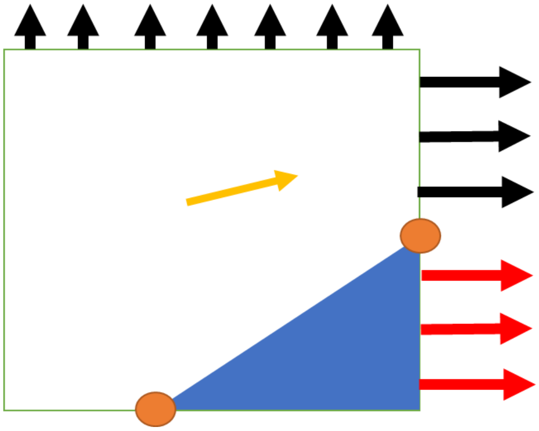


Fig. 6: An example of a cut-cell with the fluid velocity represented by a yellow arrow at the centre, fluxes out of the cell on the faces represented as black arrows, and fluxes which are blocked by the solid boundary being represented by red arrows. Solid portions are in blue, fluid portions in white, and the markers used on the cut faces are represented by orange circles.

The value of $\beta_{faceCut}$ is determined from the following set of equations for each cut-point containing face, which assumes that only 2 face-lying cut-points exist in the cell, as shown in Fig. 8 and Fig. 9:

$$\vec{d_t} = \begin{pmatrix} \dfrac{\vec{p_f} - \vec{p_1}}{w_c} & \text{where } \dfrac{\vec{p_f} - \vec{p_1}}{w_c} \leq 0.5 \\ \dfrac{\vec{p_f} - \vec{p_2}}{w_c} & \text{where } \dfrac{\vec{p_f} - \vec{p_2}}{w_c} \leq 0.5 \end{pmatrix} \tag{11}$$

$$\beta_{faceCut} = 0.5 + \frac{\vec{n} \cdot \vec{d_t}}{\left|\vec{n} \cdot \vec{d_t}\right| + SMALLVAL} \left\|\vec{d_t}\right\| \tag{12}$$

Where $\vec{p_1}$ and $\vec{p_2}$ are the 2 point ends of the line segment spanning across the cut-face, $\vec{n}$ is the representative normal direction of the cut-face with the cell centre normal direction being used here-in, $\vec{p_f}$ is the position of the original face centre, w_c is the cell side length and cell span, and SMALLVAL is a small number used to avoid divide by 0 possibilities with 1E-20 being used here-in. This formula can be visualized in Fig. 7 where the length of the black arrows is $\left\|\vec{d_t} w_c\right\|$ and there direction is $\frac{\vec{n} \cdot \vec{d_t}}{\left|\vec{n} \cdot \vec{d_t}\right|}$.

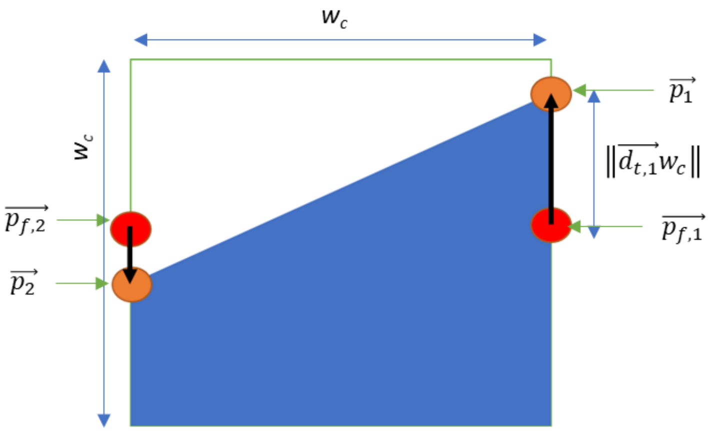

w_c
w_c
$\overrightarrow{p_1}$
$\|\overrightarrow{d_{t,1}} w_c\|$
$\overrightarrow{p_{f,1}}$
$\overrightarrow{p_{f,2}}$
$\overrightarrow{p_2}$

Fig. 7: A representation of the $\beta_{faceCut}$ algorithm. Solid portions are in blue, fluid portions in white, markers used on the cut-faces are represented by orange circles, and the red circles are the face-centres.

The flux, φ_f over a face with an original area $A_{f,orig}$, is calculated for a vector linearly interpolated from the cell centres to the face centres, F_{face} and the face normal as follows:

$$\frac{\varphi_f}{A_{f,orig}} = (\overrightarrow{F_f} \cdot \overrightarrow{n_f})\beta_{faceCut} \tag{13}$$

An additional flux term at the cut-face arises. The area of this face, $A_{cutFace}$, comes from the position of the face cut-off points, p_1 and p_2, which are along the same cell of span and side length w_c as follows:

$$A_{cutFace} = w_c \, \|\overrightarrow{p_1} - \overrightarrow{p_2}\| \tag{14}$$

The source/sink from this new cut-face, φ_{cut}, depends on the boundary condition being implemented in the cell. For Dirichlet boundary condition on a generic vector $\overrightarrow{F_{BC}}$, in this case the specified droplet velocity at the wall, the new face source term is calculated as follows:

$$\varphi_{cut} = \left(\overrightarrow{F_{BC}} \cdot \vec{n}_{cutFace}\right) A_{cutFace} \tag{15}$$

For zero-gradient Neumann boundary conditions, the cut-cell face source/sink term can be calculated from the boundary cell vector $\overrightarrow{F_b}$, in this case the droplet velocity of the cell in question, as follows:

$$\varphi_{cut} = \left(\overrightarrow{F_b} \cdot \vec{n}_{cutFace}\right) A_{cutFace} \tag{16}$$

The volume of the boundary cell must be corrected by reducing them by a multiplying factor β_{volCut}. Using the fact that the cases studied will be 2D, and assuming a smooth representation of the surface, the value β_{volCut} is calculated by using relationships for the area of triangles and squares (with pentagon shapes being the area of the full square minus the area of the triangle representing the boundary filled portion) and applying this to finding the area of the fluid portion of the boundary cell, then dividing this area by the area of the whole square. Examples of each representative shape are provided in Fig. 8.

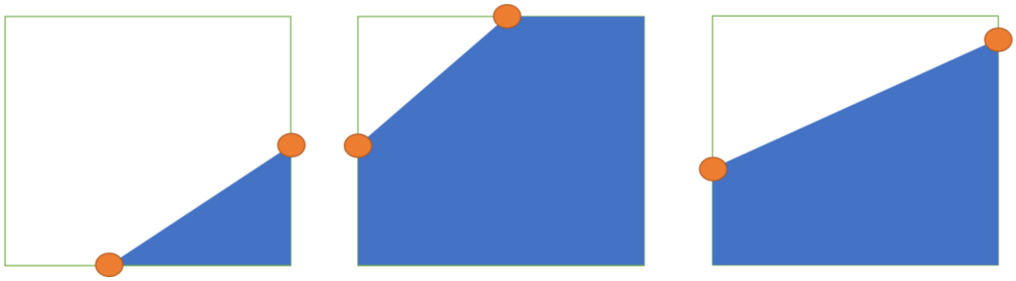


Fig. 8: The three basic shapes used to construct the cut-cells. Solid portions are in blue, fluid portions in white, and the markers used on the cut faces are represented by orange circles.

This formulation notably ignores the following 6 possible shapes shown in Fig. 9. While not an exhaustive list as one can create an infinite number of these not handled cases, these shapes are recognizable for not having 2 face-lying points on different faces, an assumption of the formulation. The 2 and 4 point unacceptable cases are the most common and should reduce in occurrence as the grid is refined if the shape being represented does not have any sharp points, but this is notably not true for a sharp-tipped airfoil, making it necessary to sufficiently blunt the sharp tail of any airfoil.

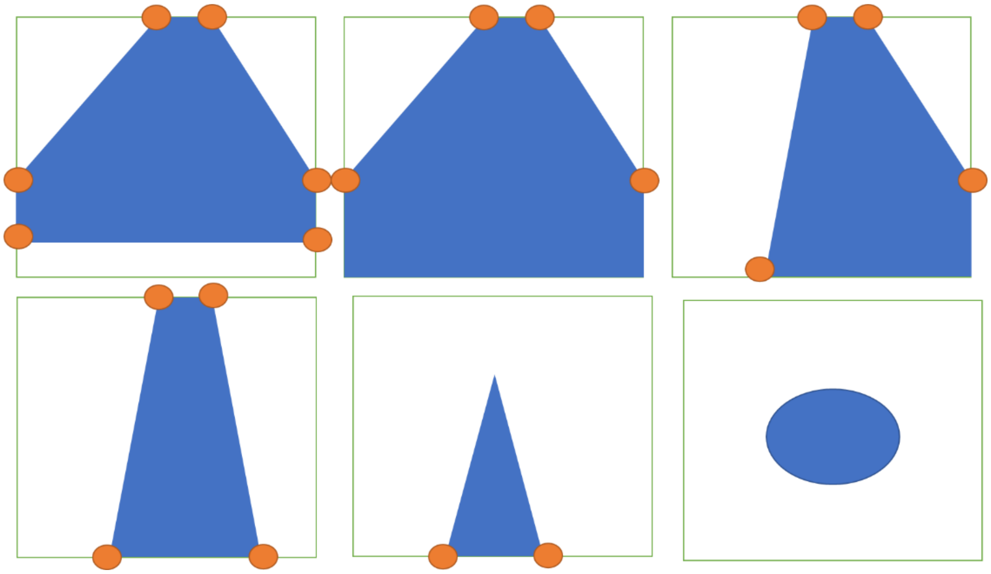


Fig. 9: Examples of possible 2D shapes not handled in this cut-cell formulation. Solid portions are in blue, fluid portions in white, and the markers used on the cut faces are represented by orange circles.

The formulas for finding the cut-cell volume is provided below based on the number of edges that the fluid region has. Heron's Formula for the area of a triangle is used in the triangle and pentagon formulations, and the quadrilateral is assumed to be a trapezoid based on acceptable shapes shown in Fig. 9 and thus the area of a trapezoid is used.

$$V_{cut} = \begin{pmatrix} w_c\sqrt{p(p-a)(p-b)(p-c)} & triangle\ (3\ edges) \\ \dfrac{H(A+B)}{2} & quadrilateral\ (4\ edges) \\ V_{cell} - w_c\sqrt{p'(p'-a')(p'-b')(p'-c')} & pentagon\ (5\ edges) \end{pmatrix} \tag{17}$$

$$\beta_{volCut} = \frac{V_{cut}}{V_{cell}} \tag{18}$$

Where V_{cell} is the original cell volume, V_{cut} is the reduced or cut-cell volume, w_c is the cell span and the cell side length; a, b and c are the side lengths of a triangular fluid region i.e., the reduced face lengths $\left(\beta_{faceCut}w_c\right)$ of the face with cut-points lying on them and the cut-face length ($\|\overrightarrow{p_1} - \overrightarrow{p_2}\|$ for cut-point ends $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$), and p is the half perimeter of this $\left(\frac{a+b+c}{2}\right)$; a', b', c', and p' are the same things but for a triangular solid region with a', b', and c' now using$\left(\left(1-\beta_{faceCut}\right)w_c\right)$ and p' being the half perimeter of the solid region$\left(\frac{a'+b'+c'}{2}\right)$; H is the height of the trapezoid (w_c), and A and B are the bases, i.e. the reduced face lengths $\left(\beta_{faceCut}\frac{A_{f,orig}}{w_c}\right)$ of the faces with cut-points lying on them. The full set of cases in-order of triangle, quadrilateral and pentagon is shown with annotation for the terms used for each side in Fig. 10.

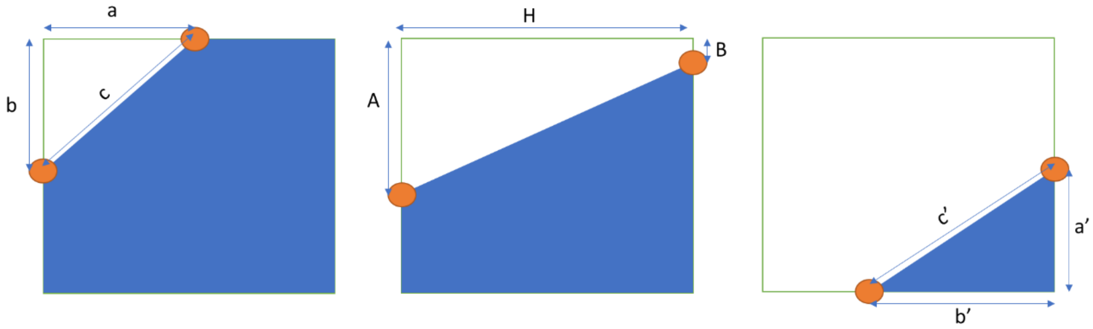


Fig. 10: Examples of triangular, quadrilateral, and pentagon type figures and the respective notation used to find fluid volumes of the cut-cells.

2.5. Immersed Boundary Method

An explicit, direct-forcing immersed boundary method is used. The boundary points are placed in the fluid domain in cells that contain boundary segments, and the values interpolated to are strictly set within the solver, and the values set in the boundary cells are updated after each step but not in-between. A point normal to the boundary point, referred to as the interpolated point, is linearly interpolated directly from the 2 cells surrounding the boundary cell which are in the normal direction (which have a higher signed-distance) and which have the smallest tangent projection of the distance vector in-between that cell centre and the boundary cell centre. If only 1 fluid cell surrounds the cell, then that cell is used as is as the interpolated point. Linear interpolation is then performed to get to the boundary cell, by using the interpolated point and a point on the boundary at a distance away equal to the signed distance of the boundary cell.

Unlike in the Cartesian cut-cell method, the face area and cell volume modifications do not affect this methodology as the values at the boundary are explicitly specified and boundary-cell to boundary cells fluxes are neglected as the boundary value is independent of the flux terms at the boundaries.

The value at the boundary point is equal to the prescribed boundary value if a Dirichlet boundary condition is used. If a Neumann boundary condition is required, then the boundary point value is extrapolated from one or more of the interpolated points. As described by Huangrui, M. (2019), for the zero-gradient Neumann boundary conditions used in this study, only the one interpolated point is required to achieve second-order accuracy, and the boundary value is set equal to that value [21]. This is presentable for the boundary value of a generic field, F_b, as follows [21]:

$$F_{b,Dirichlet} = F_{BC} + (F_i - F_{BC})\frac{||\overrightarrow{p_b} - \overrightarrow{p_{BC}}||}{||\overrightarrow{p_i} - \overrightarrow{p_{BC}}||} = F_{BC} + (F_i - F_{BC})\frac{\phi_b}{\phi_i} \tag{19}$$

$$F_{b,zeroGradient} = F_i \tag{20}$$

This notably applies to the pressure field and, when used, the droplet volume fraction and droplet velocity, but the turbulence quantities and air phase model are handled in special ways to incorporate a two-layer wall model.

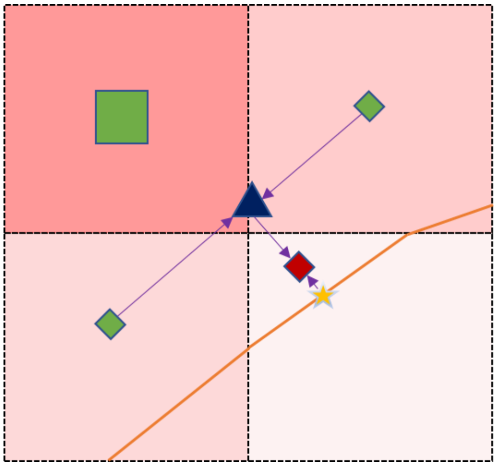


Fig. 11: Example of the used immersed boundary method interpolation scheme. The boundary point is given as a red diamond. The boundary point is given as a yellow star. The interpolated point is given as a blue triangle. Lines of interpolation are given as purple arrows.

2.6. Off-Wall Air Model

An incompressible, isothermal Reynolds-Averaged Navier-Stokes methodology is used to model the flow in fluid cells off of the boundary. The SIMPLE algorithm implemented in OpenFOAM® 5.0 is used to iterate the solution to a steady-state and find the pressure/velocity off the boundary.

For modelling turbulence off the boundary, the standard k-ε model implemented by OpenFOAM® 5.0 is used. This model was chosen for its advantages in robustness and speed compared to many competitors [22].

This model is not perfect, however many of its strengths and weaknesses have become known, giving a certain amount of predictability compared to less common turbulence models [22]. The standard k-ε model is notably known to not handle transitioning flows which are not fully developed, which may impact the flow near stagnation points [22]. The k-ε model is also known to poorly predict the amount and onset of flow separation under adverse pressure gradients when standard wall functions are used [23]. For this study, these effects are thought to play a relatively small factor in the droplet impingement, except for the cylinder case which displays large separation.

Based on the work of Ferziger, J.H. and Peric, M. (2002) [24], implemented in OpenFOAM®, and modified to incorporate the immersed boundary, the following SIMPLE-based algorithm was used:

Velocity predictor step:

$$\begin{pmatrix} \vec{\boldsymbol{U}}^{*} = 0 & \textit{dead cells} \\ \vec{\boldsymbol{U}}^{*} = \vec{U}_b & \textit{boundary cells} \\ \nabla \cdot \vec{U}\vec{\boldsymbol{U}}^{*} - \vec{\boldsymbol{U}}^{*}\nabla \cdot \vec{U} - \nabla \cdot \mu_{eff}\nabla\vec{\boldsymbol{U}}^{*} = -\nabla \mathrm{P} & \textit{fluid cells} \end{pmatrix} \tag{21}$$

Pressure:

$$\begin{pmatrix} \mathbf{P} = 0 & \textit{dead cells} \\ \mathbf{P} = P_b & \textit{boundary cells} \\ \nabla \cdot \frac{1}{a_p^{U^*}}\nabla\mathbf{P} - \nabla \cdot \frac{H\left(\vec{U}^{*}\right)}{a_p^{U^*}} = 0 & \textit{fluid cells} \end{pmatrix} \tag{22}$$

Velocity corrector step:

$$\vec{\boldsymbol{U}}^{\boldsymbol{k+1}} = \begin{pmatrix} \vec{U}^{*} & \textit{dead and boundary cells} \\ \frac{H\left(\vec{U}^{*}\right)}{a_p^{U^*}} - (1 - \delta_{dOB})\frac{\nabla P}{a_p^{U^*}} & \textit{fluid cells} \end{pmatrix} \tag{23}$$

The turbulence quantities k and ε can then be determined from the following set of equations, developed by Launder and Spalding [25], implemented in OpenFOAM®, and modified to incorporate the immersed boundary:

$$\left(\begin{array}{cc} \mathbf{k} = 0 & \mathit{dead\ cells} \\ \mathbf{k} = k_b & \mathit{boundary\ cells} \\ \nabla \cdot \rho \vec{U} \boldsymbol{k} - \frac{1}{3} \rho \boldsymbol{k} \nabla \cdot \vec{U} - \nabla \cdot \rho D_k \nabla \boldsymbol{k} - G_k + \rho \epsilon = 0 & \mathit{fluid\ cells} \end{array}\right) \tag{24}$$

$$\left(\begin{array}{cc} \boldsymbol{\epsilon} = 0 & \mathit{dead\ cells} \\ \boldsymbol{\epsilon} = \epsilon_b & \mathit{boundary\ cells} \\ \nabla \cdot \rho \vec{U} \boldsymbol{\epsilon} - \rho \boldsymbol{\epsilon} \nabla \cdot \vec{U} - \nabla \cdot \rho D_\epsilon \nabla \boldsymbol{\epsilon} - \frac{C_1 \boldsymbol{\epsilon}}{k} (G_k + C_3 G_b) + C_2 \rho \frac{\epsilon \boldsymbol{\epsilon}}{k} = 0 & \mathit{fluid\ cells} \end{array}\right) \tag{25}$$

The turbulence corrections to the effective viscosity are then made from the following [22] [26]:

$$\mu_t = \left(\begin{array}{cc} 0 & \mathit{dead\ cells} \\ \nu \kappa y^+ \left(1 - \mathrm{e}^{-\frac{y^+}{16}}\right)^2 & \mathit{boundary\ cells} \\ \rho C_\mu \frac{k^2}{\epsilon} & \mathit{fluid\ cells} \end{array}\right) \tag{26}$$

$$\mu_{eff} = \mu + \mu_t \tag{27}$$

2.7. Near-Wall Air Model

The pressure at the boundary points is extrapolated from the surrounding points from a second-order zero-gradient direct-forcing IBM technique discussed earlier, this methodology is recommended by many other researchers, such as Brehm, C. et al. (2018) [27]. Using this methodology in 2D, for two surrounding fluid cells(*1* and *2*) with weights based on the distance from their fluid cell centres and an interpolated point, $\vec{p}_i$*:* The tangent vector is obtained directly from a 90° rotation of the normal direction obtained from the gradient of the level set field.

$$d_{t,i} = \left|(\vec{p}_F - \vec{p}_i) \cdot \vec{t_i}\right| \approx \left|(\vec{p}_o - \vec{p}_i) \cdot \vec{t_f}\right| \tag{28}$$

$$P_b = P_i \approx \alpha_1 P_1 + \alpha_2 P_2 = \frac{d_{t,2} P_1 + d_{t,1} P_2}{d_{t,1} + d_{t,2}} \tag{29}$$

Two common approaches to turbulent boundary condition handling using RANS models based on the spacing of the nearest points to the wall: viscous sublayer wall models and log-law region models. Viscous sublayer wall models are applied generally to y^+ values around 1 or even smaller while the standard log-law region models are applied above 30 and below 500 [28]. The reduced memory and CPU requirements using log-law region models can be very important, especially for the computation of complex 3D flow fields [29]. The goal of this research is to test the accuracy capabilities with meshes which are sufficiently coarse near the wall so as to use the log-law sublayer wall models, allowing for computational savings, however even with this goal, due to the fact that the mesh is in no way conformal to the geometry, some boundary cells may end up well below this goal and may even be close enough to be within the viscous sublayer. Due to this, a two-layer approach is used very near to the boundary.

The dimensionless velocity, u^+, and wall distance, y^+, are defined as follows [28]:

$$u^{+} = \frac{U}{U_{\tau}} \tag{30}$$

$$y^{+} = \frac{U_{\tau} y}{\nu} \tag{31}$$

$$U_{\tau} = \sqrt{\frac{\tau_w}{\rho}} \tag{32}$$

Where y is the wall distance equal to the value of the level set field, ϕ, since the cells are outside the boundary, ν is the kinematic viscosity of the air, τ_w is the shear stress at the wall, and U_τ is the friction velocity. Using these definitions, the viscous and log-law layers of the boundary are described using the following [28]:

$$u^{+}_{viscous} = y^{+} \tag{33}$$

$$u^{+}_{log-law} = \frac{1}{\kappa}\ln(Ey^{+}) \tag{34}$$

The viscous region is assumed to begin at the wall ($y^{+} = 0$) and end where the two models intersect ($y^{+} = y^{+}_{limit}$). The values for κ and E were set to 0.41 and 9.8 respectfully based on typical empirical values given by, for example, Sreenivasan, K.R. (1989) [30], giving a y^{+}_{limit} of 11.53. This velocity profile is compared in Fig. 12 to DNS data Eitel-Amor, G. et al. (2014) for a flat plate with a Re_θ of 4060 [31]. In Fig. 12 it can be seen that the biggest deviations in-between the boundary model and DNS results occur in the ignored buffer layer in the region of approximately $5 \leq y^{+} \leq 30$ and for $y^{+} > 500$.

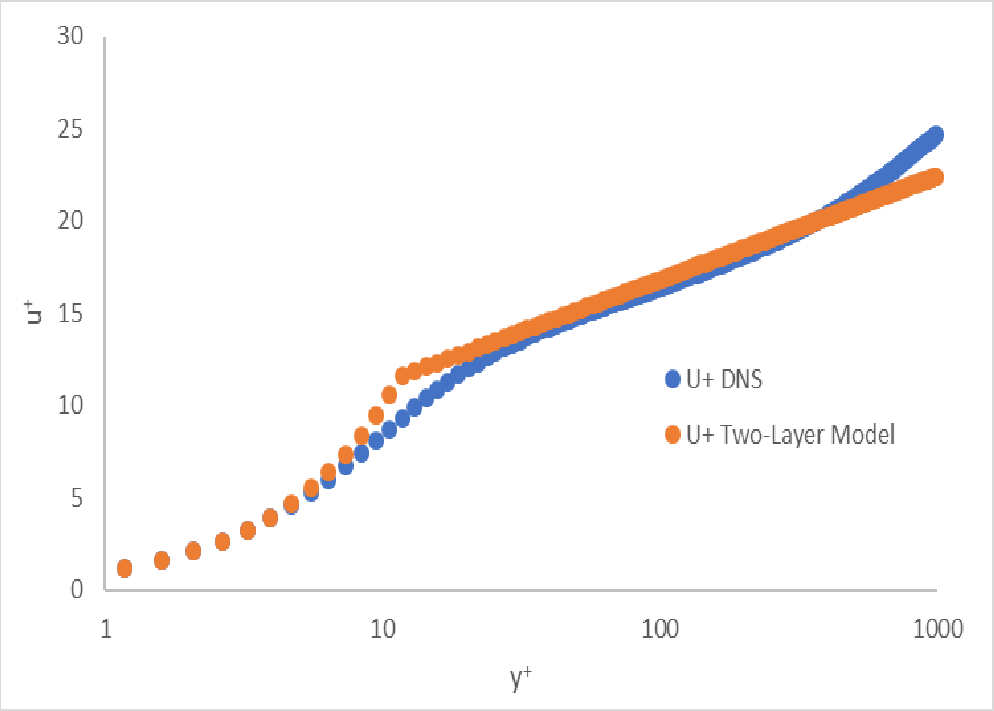


Fig. 12: Comparison of the two-layer boundary model used to DNS results from Eitel-Amor, G. et al. (2014) for a flat plate with a Reθ of 4060 [31].

These wall functions used are known to have their own issues when strong pressure gradients lead to boundary layer separation, and so for cases where separation is a major concern, non-equilibrium wall functions or local mesh refinement and resolving of the viscous sublayer using other turbulence models like k-ω SST may be required [22].

The value of the dimensionless wall distance, velocity, and wall shear stress normal to this point are all determined from an iterative solver using Newton's Method as done in OpenFOAM® [32]. One iteration of Newton's Method and the resulting friction velocity, $U_{\tau,F}$, and wall shear stress, $\tau_{w,F}$, is as follows:

$$y_F^{+^{k+1}} = \begin{pmatrix} \sqrt{\dfrac{\phi_F}{\nu}\left(\overrightarrow{U_F} \cdot \overrightarrow{t_F}\right)} & \text{if } y_F^{+^k} \leq y_{limit}^{+} \\ \dfrac{\kappa \dfrac{\phi_F}{\nu}\left(\overrightarrow{U_F} \cdot \vec{t}\right) + y_F^{+^k}}{1 + \ln\left(E y_F^{+^k}\right)} & \text{if } y_F^{+^k} > y_{limit}^{+} \end{pmatrix} \tag{35}$$

$$U_{\tau,F} = \begin{pmatrix} \dfrac{\left(\overrightarrow{U_F} \cdot \overrightarrow{t_F}\right)}{y_F^{+}} & \text{if } y_F^{+} \leq y_{limit}^{+} \\ \dfrac{\left(\overrightarrow{U_F} \cdot \overrightarrow{t_F}\right)}{\dfrac{1}{\kappa}\ln\left(E y_F^{+}\right)} & \text{if } y_F^{+} > y_{limit}^{+} \end{pmatrix} \tag{36}$$

$$\tau_{w,F} = \rho U_{\tau,F}^2 \tag{37}$$

A second-order accurate formulation of the wall functions over the implicitly described surface is achieved by interpolating over the wall shear stresses obtained from the surrounding fluid cell velocities and level set field. The surface is assumed locally flat in the calculations of the interpolating coefficients for the wall shear stress. For a locally flat surface, the coefficients for the interpolation are based on the tangential projection in-between the fluid cell centres and the boundary cell centre described earlier, $d_{t,F}$, and the formulation becomes:

$$\tau_w \approx a_1 \tau_{w,1} + a_2 \tau_{w,2} = \frac{d_{t,2}\tau_{w,1} + d_{t,1}\tau_{w,2}}{d_{t,1} + d_{t,2}} \tag{38}$$

$$U_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{39}$$

Since the wall shear stress is interpolated, and the surface is assumed flat, the dimensionless wall distance, y^+, can then be obtained using separate normal and tangential interpolations from the far point values $F = 1$ and 2:

$$y_{b,F}^{+} = \frac{U_{\tau,F}}{\nu} y_b = \frac{U_{\tau,F} y_F}{\nu} \frac{y_b}{y_F} = y_F^{+} \frac{\phi_b}{\phi_F} \tag{40}$$

$$y_b^{+} \approx a_1 y_{b,1}^{+} + a_2 y_{b,2}^{+} = \frac{d_{t,2} y_1^{+} \dfrac{\phi_b}{\phi_1} + d_{t,1} y_2^{+} \dfrac{\phi_b}{\phi_2}}{d_{t,1} + d_{t,2}} \tag{41}$$

Like the y^+ interpolations, the boundary cell velocity calculation requires interpolations in the normal direction and the tangential direction. The normal direction interpolation is done directly through the profile provided by the wall function and for each far point the tangential interpolation is done via linear interpolation as before:

$$u_b^+ = \begin{pmatrix} y_b^+ & \text{where } y_b^+ \leq y_{limit}^+ \\ \frac{1}{\kappa}\ln(Ey_b^+) & \text{where } y_b^+ > y_{limit}^+ \end{pmatrix} \tag{42}$$

$$u_F^+ = \begin{pmatrix} y_F^+ & \text{where } y_F^+ \leq y_{limit}^+ \\ \frac{1}{\kappa}\ln(Ey_F^+) & \text{where } y_F^+ > y_{limit}^+ \end{pmatrix} \tag{43}$$

$$\vec{U}_{b,F} = \vec{t}\,\frac{u_b^+}{u_F^+}\left(\overrightarrow{U_F} \cdot \overrightarrow{t_F}\right) \tag{44}$$

$$\vec{U}_b \approx a_1\vec{U}_{b,1} + a_2\vec{U}_{b,2} = \vec{t}\left(\frac{d_{t,2}\frac{u_b^+}{u_1^+}\left(\overrightarrow{U_1} \cdot \overrightarrow{t_1}\right) + d_{t,1}\frac{u_b^+}{u_2^+}\left(\overrightarrow{U_2} \cdot \overrightarrow{t_2}\right)}{d_{t,1} + d_{t,2}}\right) \tag{45}$$

Near-wall turbulence quantities, *k* and *ε*, are updated based on the two-layer wall functions implemented in ANSYS FLUENT directly from the interpolated wall shear stress, τ_W, and boundary y^+ as follows [22]:

$$k_b = \begin{pmatrix} \frac{\tau_W}{\rho C_\mu^{0.5}}\left(\frac{y_b^+}{y_{limit}^+}\right)^2 & \text{where } y_b^+ \leq y_{limit}^+ \\ \frac{\tau_W}{\rho C_\mu^{0.5}} & \text{where } y_b^+ > y_{limit}^+ \end{pmatrix} \tag{46}$$

$$\epsilon_b = \begin{pmatrix} 2\frac{\nu k_b}{\phi_b^2} & \text{where } y_b^+ \leq y_{limit}^+ \\ \frac{C_\mu^{0.75}}{\kappa}\frac{k_b^{1.5}}{\phi_b} & \text{where } y_b^+ > y_{limit}^+ \end{pmatrix} \tag{47}$$

Where C_μ is set to the default value in the standard k-ε model implemented in OpenFOAM® 5.0, 0.09.

To test the validity of the air phase model and the wall functions used, a fully turbulent flat-plate test case was devised. This case is well studied, and several semi-empirical results have been devised for the friction coefficient, C_f, profile along the plate. The Prandtl 1/5 model for skin friction at a given position along the flat plat, x_L, given a fluid kinematic viscosity of ν, and a freestream velocity of U_∞, as follows [33]:

$$Re_L = \frac{U_\infty x_L}{\nu} \tag{48}$$

$$C_{f,model} = \frac{0.059}{Re_L^{1/5}} \tag{49}$$

Based on this, for a Reynolds number, Re_L, of 5E6, 7.5E6, and 1E7 the coefficient of friction is estimated to be 0.002698, 0.002488, and 0.002349 respectfully.

The flat plate test case used was based on the 2D Zero Pressure Gradient Flat Plate Verification Case from the NASA Langley Research Center's Turbulence Modeling Resource [34]. The mesh used is somewhat special compared to the meshes for other tests in a couple of ways. Firstly, the square cell condition does not need to be strictly enforced here as only a single direction matters in the IBM-scheme for the air phase, the direction perpendicular to the flat plate. In this direction, the closest cell to the boundary and 3 cells above it is kept at a constant cell width in the perpendicular direction. Below the plate, 2 dead cell layers are placed to avoid any issues in the algorithm when identifying the boundary layer. Above the constant layer, the cells are kept at a constant expansion ratio. In-front of the plate the dead cell layer is not placed, and a horizontal symmetry plane boundary condition is placed at the bottom. In the direction along the plate, a constant expansion ratio is placed going toward and then away from the point at which the plate starts. The expansion ratio towards and along the plate, away and along the plate, and away and perpendicular to the plate are all set to the same values, and at the leading edge of the plate, the cell length along the plate direction is set to the cell length perpendicular to this direction to reduce the number of free variables to fix when doing a mesh refinement study down to 2: the expansion ratio and the minimum cell length. A freestream condition is placed 1 m above the plate, a fixed velocity/k/ε, zero-gradient pressure inlet condition is placed 0.5 m in front of the plate; and a zero-gradient velocity/k/ε, fixed pressure outlet condition is placed 4 m along the plate. This setup can be seen graphically alongside an example mesh for a nearest wall cell length of 1.5E-3 m and an expansion rate of 3% in Fig. 13.

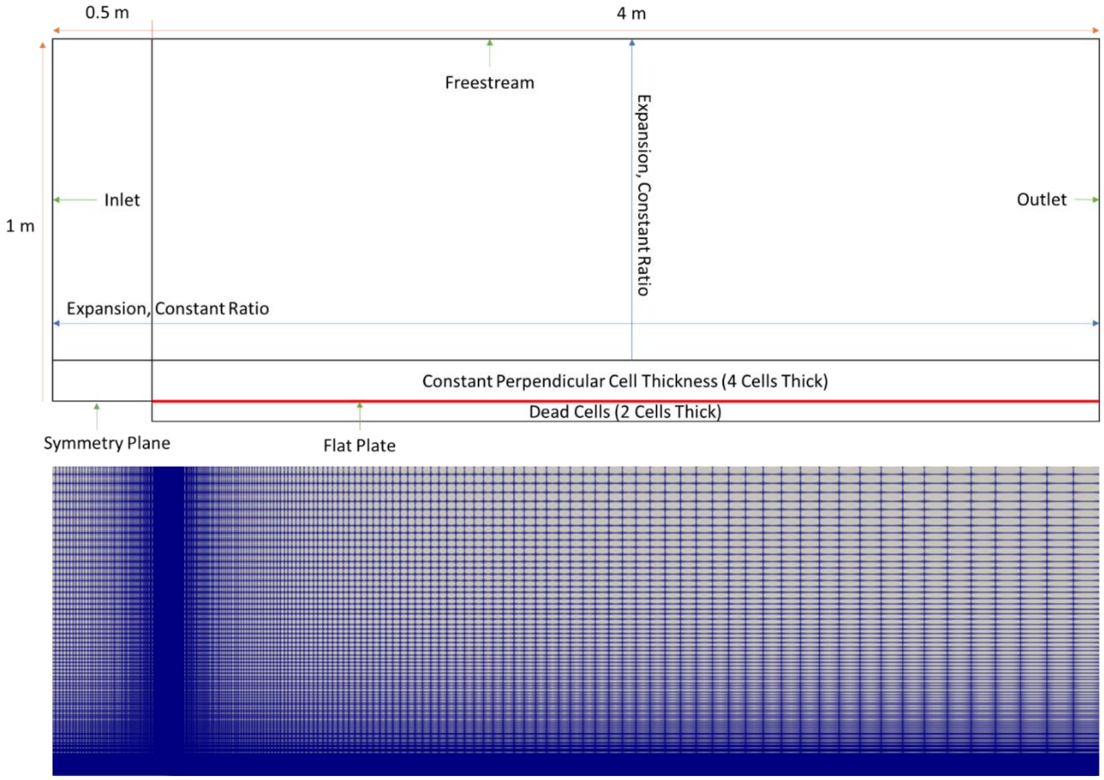

0.5 m
4 m
Freestream
Expansion, Constant Ratio
1 m
Inlet
Outlet
Expansion, Constant Ratio
Constant Perpendicular Cell Thickness (4 Cells Thick)
Dead Cells (2 Cells Thick)
Symmetry Plane
Flat Plate

Fig. 13: Setup for flat plate test case and the resulting mesh for a nearest wall spacing of 1.5E-3 m and an expansion rate of 3%.

The inlet velocity is set to 69.4 m/s, the density is set to 1.344 kg/m^3, and the dynamic viscosity is set to 1.866 Pa·s, k is set to 1E-3 m^2/s^2 and ε is set to 0.8 m^2/s^3. These settings correspond to Reynolds numbers of 5E6, 7.5E6, and 1E7 respectfully. The mesh setups, resulting friction coefficients, and comparison to the 1/5 Prandtl model are given in Table 1.

Table 1: Flat plate mesh setups, resulting friction coefficients, and comparisons to 1/5 Prandtl model results.

Min Cell length(m)	y^+_{min} [1 m]	Expansion Rate	Number of Cells Along	Number of Cells Perpendicular	C_f [1 m]	C_f [1.5 m]	C_f [2 m]	$\left\lvert\frac{C_f}{C_{f,model}}-1\right\rvert$ [1 m]	$\left\lvert\frac{C_f}{C_{f,model}}-1\right\rvert$ [1.5 m]	$\left\lvert\frac{C_f}{C_{f,model}}-1\right\rvert$ [2 m]
1.5E-3	137	3%	230	103	0.002684	0.002478	0.002379	0.51%	0.42%	1.28%
7.5E-4	68.3	5%	187	86	0.002679	0.002496	0.002373	0.70%	0.30%	1.03%
7.5E-4	68.3	3%	275	126	0.002677	0.002480	0.002369	0.77%	0.32%	0.86%
7.5E-4	68.3	1%	607	267	0.002680	0.002487	0.002367	0.69%	0.02%	0.76%
7.5E-4	68.3	0.5%	960	408	0.002679	0.002489	0.002367	0.71%	0.05%	0.77%
3.75E-4	34.0	3%	321	149	0.002650	0.002478	0.002379	1.77%	0.42%	1.28%
1.875E-4	17.3	3%	368	172	0.002738	0.002589	0.002479	1.49%	4.07%	5.55%
1.1E-5	0.828	3%	559	268	0.001950	0.001883	0.001834	27.72%	24.32%	21.93%

As seen in Table 1, tests showed that if the far point, which is equal to 3 times y^+_{min} in this case, is kept above 90 and below 400, the resultant coefficient of friction is estimated within a 5% error at a Reynolds number of 5E6. Results seen in Table 1 show that even as low as $y^+ = 17$, the resultant coefficient of friction, is within 5% accuracy, but in the buffer layer, approximately from $5 \leq y^+ \leq 30$ [30], the boundary layer velocity profile is up to 25% error based on DNS results from Eitel-Amor, G. et al. (2014) for a flat plate with a Re_θ of 4060 [31] seen in Fig. 12, with the maximum error occurring approximately at the intersection of the viscous and log-law layers. This indicates that placing points in this region should be avoided as much as possible. Serious concerns in the skin friction and boundary profile start to appear when the tests are refined so that the far y^+ is down past the log-law region, for example, a y^+ of 0.83, or a far y^+ of 2.5 as shown in Table 1 and Fig. 14. If the far point appears in the viscous sublayer, then the k-ε model is known to cause issues, as was seen, unless complex non-linear damping functions are used [23]. To avoid this, models such as the k-ω SST model can be employed [23].

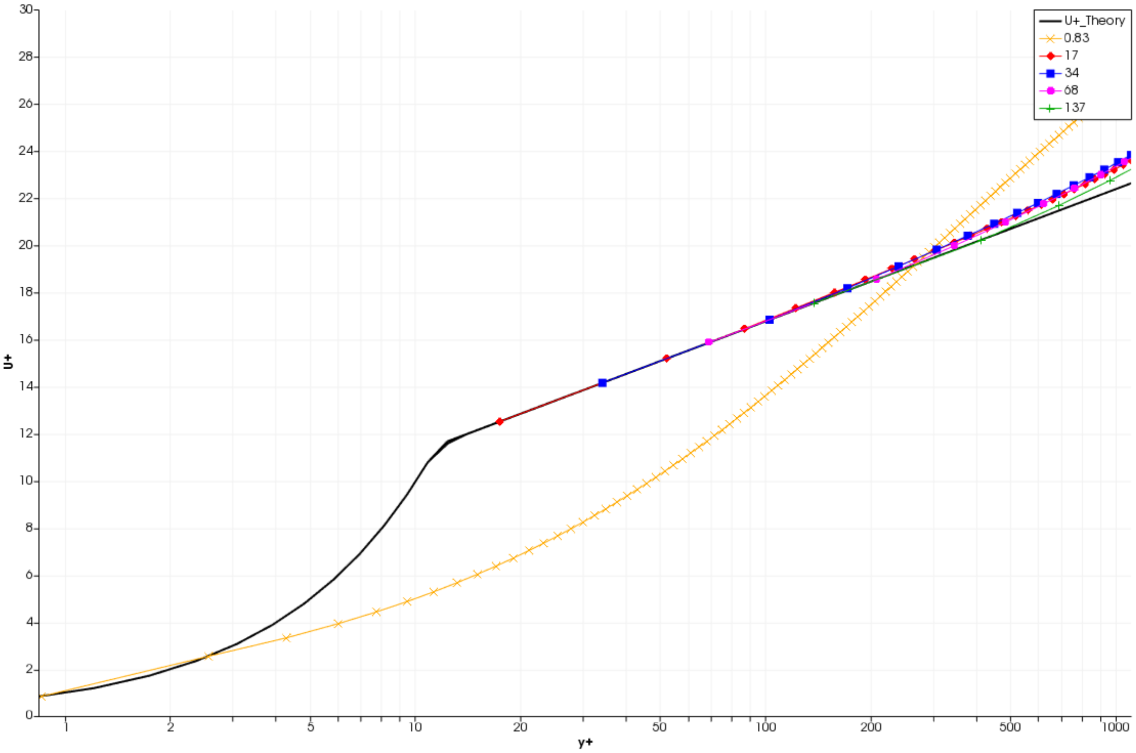


Fig. 14: Numerically obtained and theoretical curves for dimensionless wall distance (y^+) and velocity (U^+) for a distance of 1 m (Re_L = 5E6) along the flat plate case, labelled by the respective y^+ at the closest cell to the wall.

With angles included in cases with variable thickness such as airfoils and cylinders, it becomes more difficult to estimate the relationship between the far y^+ and the close y^+ since 2 points are now used. Additionally, there is in these cases a distribution of wall distances (from 0 to sqrt(2) times the half-cell width) and cell-wall orientations which lead to a distribution of near-wall and far point y^+'s as well as freestream effects which further complicate the scenario for the boundary layer relative to the flat plate scenario.

To test the boundary layer model's performance over a curved surface, data from Shin and Bond Case 1 before ice accretion was used. This case is a NACA0012 airfoil with a 0.5334 m chord at a 4° angle of attack, with a setup described in Table 19 in the Appendix. Two positions were tested on the airfoil: the top point of the rotated airfoil at a position 0.1890 chords in the x-direction (at a 4° angle to the chord) from the tip, and the bottom point at a position 0.5256 chords in the x-direction from the tip.

An inner region cell spacing of 8E-4 m was found to correspond to an average y^+ of approximately 45, with 4E-4 m being half of this at about 22.5 and 1.6E-3 m being twice this at about 90, this said the top and bottom values are not at these values, with some tests being farther than others, due to the distribution of near-wall spacings and orientations discussed earlier. The probe locations and velocity fields obtained under these conditions on these grids can be seen in Fig. 15.

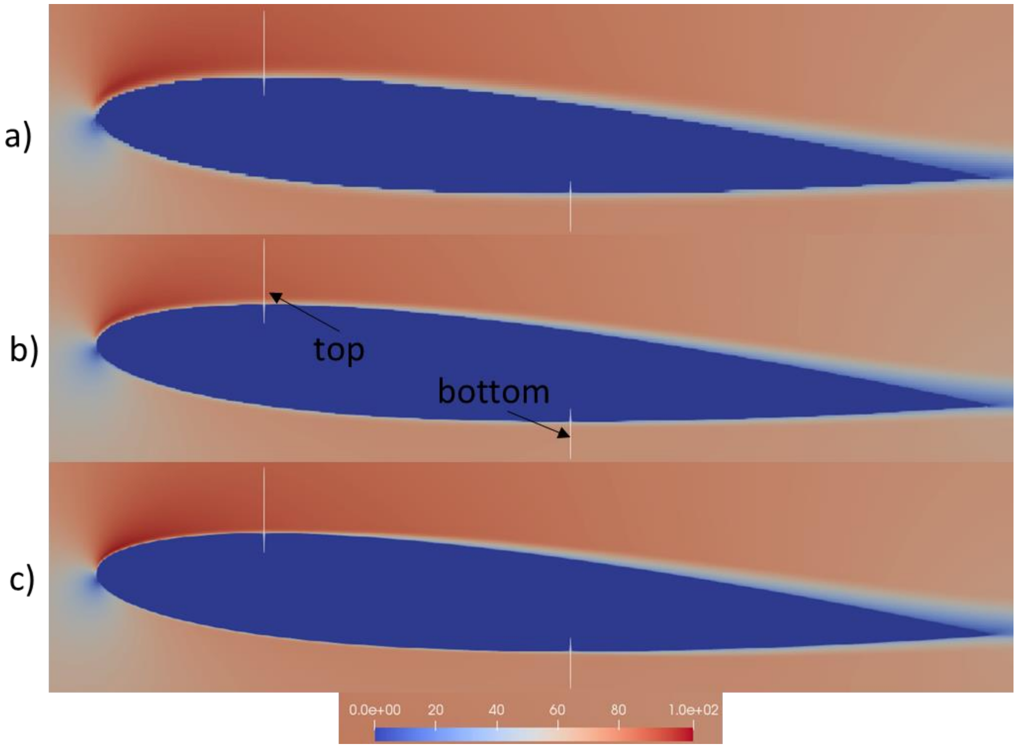


Fig. 15: Initial velocity magnitude fields for Shin and Bond Case 1 using a 5% expansion rate, 10-chord farfield distance, and an inner region cell spacing of a) 1.6E-3 m, b) 8E-4 m, c) 4E-4 m with the top and bottom of the airfoil given the 4^{o} orientation outlined using white lines. Velocity magnitude is given in m/s.,

As seen in Fig. 16, the deviation of the boundary layer velocity profile from flat plate theory is under 10% up to a y^+ of 500 for the average y^+ of 45 and 90 cases, similar to that seen in Fig. 14 for a flat plate. For the case where the average y^+ is about 22.5, which is in the relatively poorly modelled buffer region, the deviation is still under 20%, although care has to be taken in these cases as the boundary layer profile in the buffer is not modelled accurately.

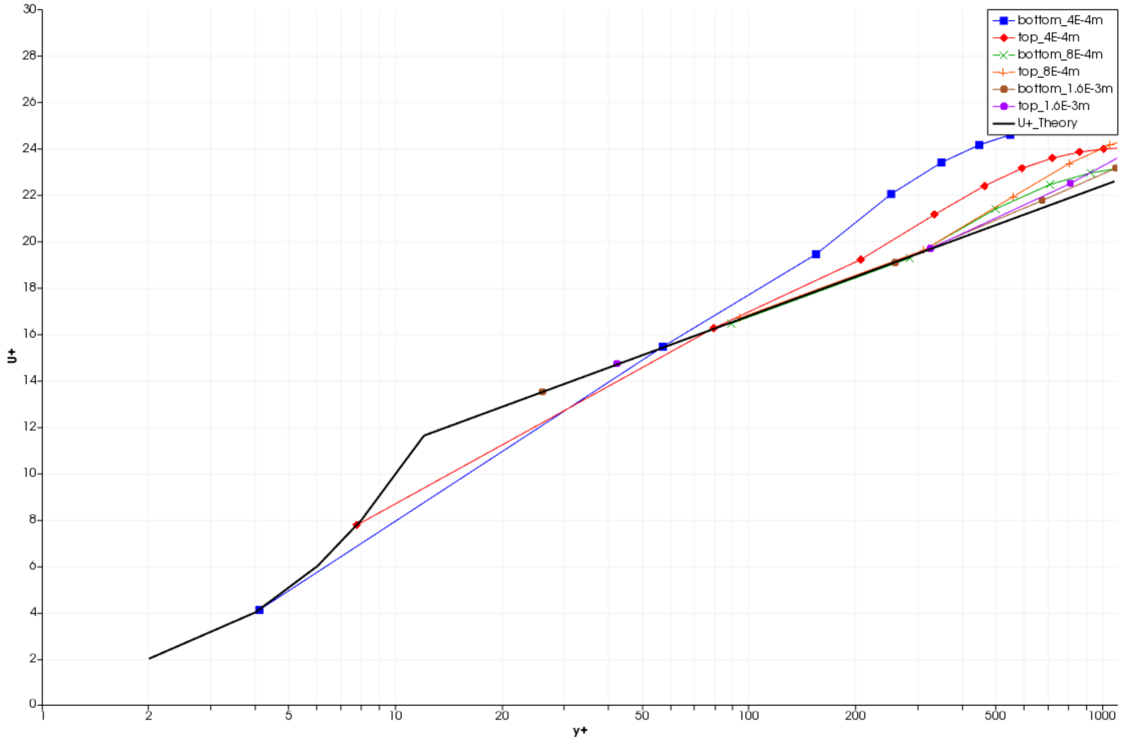


Fig. 16: Numerically obtained and theoretical curves for dimensionless wall distance (y^+) and velocity (U^+) at the top point and bottom point evaluated at the angle of attack of 5⁰ along the Shin and Bond Case 1 NACA0012 airfoil, labelled by the respective y^+ at the closest cell to the wall.

One region of particular concern for the collection efficiency over an airfoil is the region near to the stagnation point. While difficult to fully quantify this behaviour and its impact on the collection efficiency short of analyzing the collection efficiency as will be done in later sections, Fig. 17 shows that while many features of the FENSAP-ICE velocity field near the stagnation point seem to agree with the OpenFOAM®-based results, the circled region in the figure is noteably different. This region over the airfoil shows the maximum velocity over the airfoil as being notably farther from the airfoil compared to FENSAP-ICE, especially on coarser grids. This may be related to the forced log-law profile used for the boundary cells, and while it tends to decrease with mesh refinement, this may be a source of error for tests as refining the mesh to remove this issue may mean refining down to a region where the boundary model will have errors.

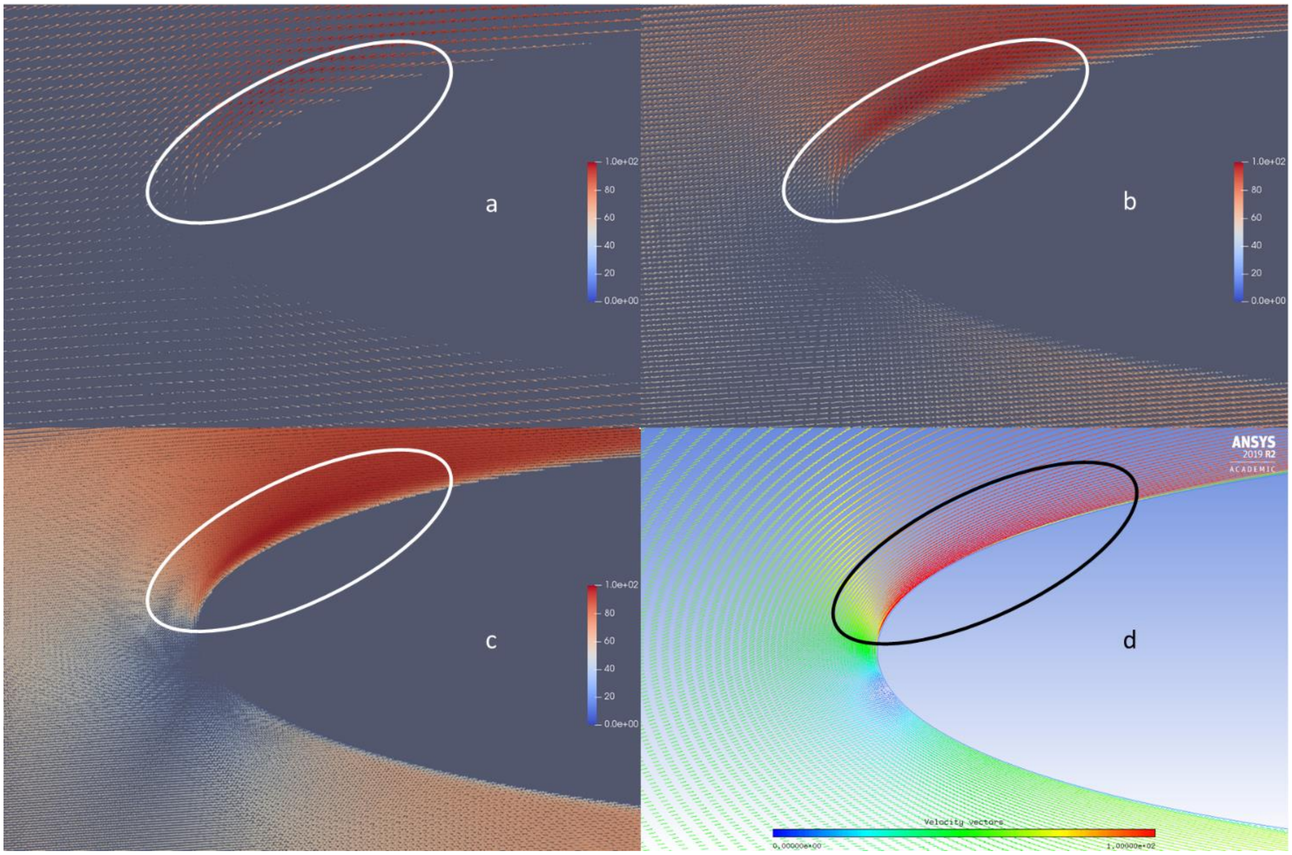


Fig. 17: Scaled vector plot of air velocity near the stagnation point for Shin and Bond Case 1 using a 5% expansion rate, 10-chord farfield distance, and an inner region cell spacing of a) 1.6E-3 m, b) 8E-4 m, c) 4E-4 m, alongside a FENSAP-ICE velocity vector plot which followed a mesh refinement study, with an average y^+ of 45. Velocity magnitude is presented in m/s. The area over the airfoil is circled due to interest in behavior upon refinement and in difference in-between the proposed model and FENSAP-ICE results.

2.8. Droplet Model

Unlike Lagrangian particle-tracking models traditionally used in the field like LEWICE and ONERA codes, the Eulerian continuum approach to modelling droplets requires no averaging over the surface to obtain impinging fluxes [8]. This model, which was first applied to the field of droplet impingement by Bourgault, Y. et al (1999), makes the following assumptions [35]:

- Droplets are perfect spheres, with no deformation or breaking being accounted for, this assumption may be questioned, and require modifications for supercooled liquid droplets, as done in the drag force model employed by Lavoie, P. (2017) [36].
- Droplet collisions, coalescence, and splashing are neglected to reduce model complexity, stability and efficiency.
- Mass transfer between the droplets and the surrounding air is assumed to be small and is neglected.

- The only force assumed to be acting on the droplets is assumed to be drag, with the minor effects of gravity, and all unsteady forces such as virtual mass and Basset forces being neglected.

This means that only the effects of convection and drag from the air phase are considered. The model in differential form can be represented by the following equations [35]:

$$\nabla \cdot \alpha \overrightarrow{U_d} = 0 \tag{50}$$

$$\nabla \cdot \alpha \overrightarrow{U_d U_d} - \overrightarrow{U_d} \nabla \cdot \alpha \overrightarrow{U_d} + D_{Ud} \alpha \left(\overrightarrow{U_d} - \vec{U} \right) = 0 \tag{51}$$

The convection of the droplet velocities can be simplified, turning the equations on-conservative form, based on recommendations by Bourgault, Y. et al. (1999) [8]. This form of the equation, shown below, is questioned and avoided by some, for example, Tong, X.L. and Luke E.A. (2010) [35], and is not strictly conservative. It can be noted that the whole equation is no longer directly dependent on the concentration of the cell or the surrounding cells.

$$\nabla \cdot \overrightarrow{U_d U_d} - \overrightarrow{U_d} \nabla \cdot \overrightarrow{U_d} + D_{Ud} (\overrightarrow{U_d} - \vec{U}) = 0 \tag{52}$$

The non-conservative form of the droplet velocity equation when the droplet concentration drops to below 1% of the inlet value in-order to improve stability and numerical efficiency of the model, which as noted by researchers such as Bourgault, Y. et al. (1999) [8], the conservative model can struggle with. One alternative is to apply an artificial viscosity in regions of colliding droplet streamlines, for example in low concentration regions in eddies in separated regions on the back end where low concentration eddies may form, as done by Tong, X.L. and Luke E.A. (2010) [35]. Tong, X.L. and Luke E.A (2010) identify that in these regions, the Eulerian methodology incorrectly predicts impulses resulting from colliding droplet streamlines [35]. This issue causes instabilities and is non-physical because droplet streamlines should be completely non-interacting in a one-way coupled model [35]. The model resulting from this switch to the non-conservative form avoids the need for artificial viscosity, at least on simple geometries, and balances accuracy, conservation, computational efficiency, and stability.

The boundary conditions used are described by the following set of equations, handled by the previously described IBM and cartesian cut-cell methodologies in this work:

$$\begin{pmatrix} \dfrac{\partial \alpha}{\partial \vec{n}} = 0 \\ \dfrac{\partial \overrightarrow{U_d}}{\partial \vec{n}} = 0 \end{pmatrix} \text{if}\, \overrightarrow{U_d} \cdot \vec{n} < 0 \tag{53}$$

$$\begin{pmatrix} \alpha = 0 \\ \overrightarrow{U_d} = 0 \end{pmatrix} \text{ if } \overrightarrow{U_d} \cdot \vec{n} \geq 0 \tag{54}$$

Using IBM, these boundary conditions are handled via interpolation in the tangential, and in the Dirichlet case, the normal direction. For cut-cell methodology, these conditions are both directly incorporated into the equations via the term $\varphi_{d,cut}$, representing the sink from the cell from the flux onto the cut face in the boundary cell in question. The resulting set of equations can be described fully from the following set of integrated equations, assuming 2 valid surrounding points for the IBM formulation:

$$\left(\begin{array}{cc} \boldsymbol{\alpha} = 0 & dead\ cells \\ \boldsymbol{\alpha} = \frac{d_{t,2}\alpha_1 + d_{t,1}\alpha_2}{d_{t,1} + d_{t,2}} & boundary\ cells, IBM, \overrightarrow{U_d} \cdot \vec{n} < 0 \\ \boldsymbol{\alpha} = \left(\frac{\phi_c}{\left(\frac{1}{d_{t,1} + d_{t,2}}\right)\left(d_{t,2}\phi_1 + d_{t,1}\phi_2\right)} \right)\left(\frac{d_{t,2}\alpha_1 + d_{t,1}\alpha_2}{d_{t,1} + d_{t,2}}\right) & boundary\ cells, IBM, \overrightarrow{U_d} \cdot \vec{n} \geq 0 \\ \sum_{faces\ f} \boldsymbol{\alpha_s}\varphi_f + \boldsymbol{\alpha_c}\varphi_{cut} = 0 & boundary\ cells, cut - cell \\ \sum_{faces\ f} \boldsymbol{\alpha_s}\varphi_f = 0 & fluid\ cells \end{array}\right) \quad (55)$$

$$\left(\begin{array}{cc} \overrightarrow{\boldsymbol{U_d}} = 0 & dead\ cells \\ \overrightarrow{\boldsymbol{U_d}} \approx \frac{d_{t,2}\overrightarrow{U_{d,1}} + d_{t,1}\overrightarrow{U_{d,2}}}{d_{t,1} + d_{t,2}} & boundary\ cells, IBM, \overrightarrow{U_d} \cdot \vec{n} < 0 \\ \overrightarrow{\boldsymbol{U_d}} \approx \left(\frac{\phi_c}{\left(\frac{1}{d_{t,1} + d_{t,2}}\right)\left(d_{t,2}\phi_1 + d_{t,1}\phi_2\right)} \right)\left(\frac{d_{t,2}\overrightarrow{U_{d,1}} + d_{t,1}\overrightarrow{U_{d,2}}}{d_{t,1} + d_{t,2}}\right) & boundary\ cells, IBM, \overrightarrow{U_d} \cdot \vec{n} \geq 0 \\ \sum_{faces\ f} (\overrightarrow{\boldsymbol{U_{d,s}}} - \overrightarrow{\boldsymbol{U_{d,c}}}\beta_{cut})\varphi_{d,f} + \overrightarrow{\boldsymbol{U_{d,c}}}(1 - \beta_{cut})\varphi_{d,cut} + D_{Ud}(\overrightarrow{\boldsymbol{U_{d,c}}} - \overrightarrow{U_c})V_{cut} = 0 & boundary\ cells, cut - cell\ , non - conservative \\ \sum_{faces\ f} \left((\alpha\overrightarrow{\boldsymbol{U_d}})_s - \alpha_f\overrightarrow{\boldsymbol{U_{d,c}}}\beta_{cut}\right)\varphi_{d,f} + \alpha_c\overrightarrow{\boldsymbol{U_{d,c}}}(1 - \beta_{cut})\varphi_{d,cut} + \alpha_c D_{Ud}(\overrightarrow{\boldsymbol{U_{d,c}}} - \overrightarrow{U_c})V_{cut} = 0 & boundary\ cells, cut - cell, conservative \\ \sum_{faces\ f} (\overrightarrow{\boldsymbol{U_{d,s}}} - \overrightarrow{\boldsymbol{U_{d,c}}})\varphi_{d,f} - D_{Ud}(\overrightarrow{\boldsymbol{U_{d,c}}} - \overrightarrow{U_c})V = 0 & fluid\ cells, non - conservative \\ \sum_{faces\ f} \left((\alpha\overrightarrow{\boldsymbol{U_d}})_s - \alpha_f\overrightarrow{\boldsymbol{U_{d,c}}}\right)\varphi_{d,f} + \alpha_c D_{Ud}(\overrightarrow{\boldsymbol{U_{d,c}}} - \overrightarrow{U_c})V = 0 & fluid\ cells, conservative \end{array}\right) \quad (56)$$

Where $\left(\alpha\overrightarrow{U_d}\right)_s$ and $\overrightarrow{U_{d,s}}$ are combinations of the neighbouring a current cell by the weighted according to the specific schemes used, α_c and $\overrightarrow{U_{d,c}}$ have a weight of 1 associated with the current cell and 0 with neighbouring cells, α_f is an explicit term linearly interpolated to the face centre, $\varphi_{d,f}$ is the droplet velocity source/sink coming from the flux explicitly interpolated to the boundary, $\varphi_{d,cut}$ is the sink coming from the flux associated with the special cut face for the cut-cell formulation, V is the (original) volume of the cell, V_{cut} is the volume of the modified cell to account for the dead portion of a cut-cell in the cut-cell formulation, β_{cut} is V_{cut} over V, $\overrightarrow{U_c}$ is the air velocity of the current cell, ϕ_c is the signed distance at the current cell centre (the level set field value), ϕ_1 and ϕ_2 are signed distances of the surrounding cell centres, and $d_{t,1}$ and $d_{t,2}$ are the length of the tangential projection of the distances from the surrounding cells onto the current boundary cell centre. The drag term, D_{Ud}, is calculated from the following set of equations based on drag over a sphere from Clift, R. and Gauvin, W.H. (1970) [37]:

$$Re_d = \frac{d_p \left\| \overrightarrow{U_d} - \vec{U} \right\|}{\nu} \tag{57}$$

$$f = \frac{Re_d}{24} C_d = 1 + 0.15 Re_d^{0.687} + \frac{0.0175}{1 + 42500 Re_d^{-1.16}} \tag{58}$$

$$\tau_p = \frac{\rho_d d_p^2}{18\mu} \tag{59}$$

$$D_{Ud} = \frac{f}{\tau_p} \tag{60}$$

For a droplet with a mean value diameter of d_p, a density ρ_d moving through a fluid with a freestream velocity of U_∞, a kinematic viscosity ν, and a dynamic viscosity μ.

The droplet velocity field was found to not change very much when using the proposed model over the non-conservative model unless the mesh is very coarse, at least for the Ma0.4 NACA0012 Case described by Table 22 with a 5-chord outer region and an expansion rate of 5% using cut-cell methodology. The result is that the total collection efficiency decreases by 0.1% when using the non-conservative model compared to the proposed model for a 6E-4 m On a mesh with a 1.2E-3 m inner region spacing this decrease changes to 0.5% and for 2.4E-3 the solution is 0.6% higher than the solution predicted with the proposed model.

From a theoretical standpoint, the conservative form of the equations add one more constraint to the system and can lead to a less efficient solution of the system [8], but with the current setup, the difference is not very large compared to the time taken for the air solver. In the FVM solution of conservative forms of the droplet momentum equation, regardless of the coarseness of the mesh, the droplet momentum conservation is preserved, and errors take only the form of for example diffusivity [38]. Large differences in-between the two forms are expected were found to, only appear only near shocks and discontinuities [8], namely the region in-between the shadow and non-shadow regions, as seen in Fig. 18 over the top of the airfoil. In certain cases this issue may impact the solution, but this is not the case for the tests run here-in.

The differences that were seen in-between the two models being small adds validity to the original assumptions made by Bourgault, Y. et al. (1999) in the development of the more standard non-conservative model in the current case [8], however on more coarse meshes this lack of strict conservation may play a small role in the error that the conservative model will help. Due to this and the relative lack of impact on net efficiency of the method from using the proposed model led to the decision to use the conservative form of the equations.

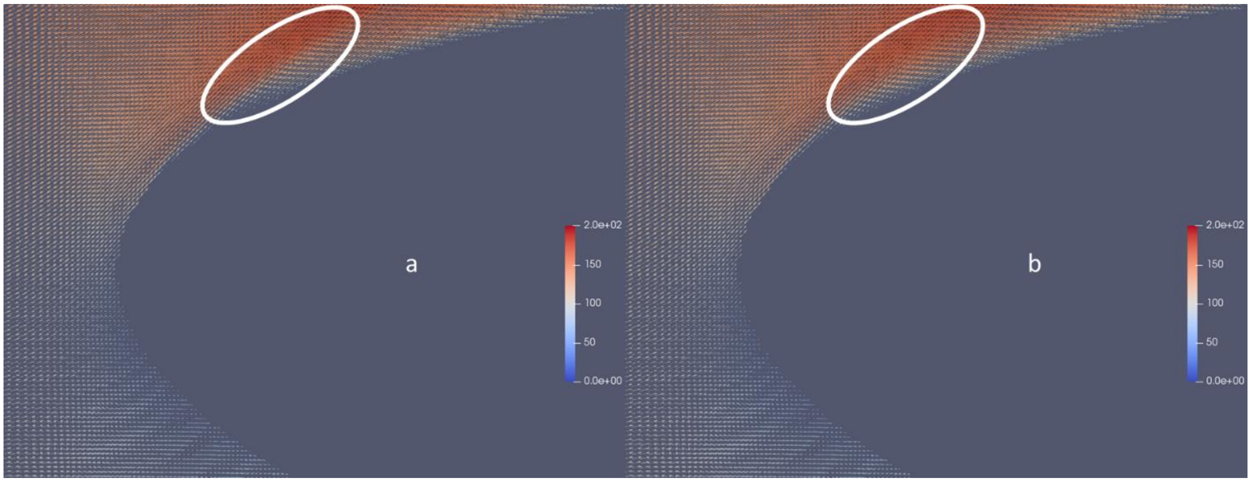


Fig. 18: Scaled vector plot of droplet velocity in m/s for Ma 0.4 NACA0012 using cut-cell methodology with a 5-chord outer region with an expansion rate of 5%, and an inner region spacing of 6E-4 m, with the airfoil being viewed at a 5° angle and zoomed on the front-end. a) The proposed droplet momentum equation is used. b) The non-conservative form of the droplet momentum equation is used. The beginning of the shadow region of the droplets, where the largest differences in the two models are seen, are highlighted with white circles.

2.9. Numerical Schemes

In time, equations were evolved using the first-order backwards (implicit) Euler method.

Gradients were calculated using a *Gauss linear* scheme, meaning that a second-order accuracy is achieved for example for the normal direction calculation. The air and droplet velocity convection discretizations are handled using a *cellLimited linearUpwindV* scheme, a higher-than-first-order scheme. The divergence of droplet velocity times concentration is discretized using a *limitedLinear01* scheme, which is a higher-than-first-order TVD scheme which helps to ensure boundedness around the physical bounds of the volume fraction of 0 and 1. The convection of k and ε were handled using the first-order accurate upwind scheme for stability and boundedness. Diffusion and interpolations were handled via second-order *linear* methodologies. The *orthogonal* methodology is used for snGrad.

The numerical schemes implemented in OpenFOAM® 5.0 were not modified to account for the modified cell centres near the boundary, generating a first-order error in this region. In future studies, it is recommended that these schemes be modified so that the weights reflect the modifications to the cell centres.

2.10. Farfield Turbulence Setup

Since the k-ε/ k-ω SST models used in the OpenFOAM® and FENSAP-ICE models tested here-in introduce 2 new transport equations for the turbulence modelling, 2 conditions must be specified at each boundary in-order to prevent under-specification. At the outlets, a zero-gradient condition is applied to the turbulence quantities in question, and at boundaries of the object the flow is going over, wall models handle these quantities, however at the inlet, 2 quantities must be specified, leading to 2 degrees of freedom in how these are handled for each case. While in theory for external cases, these quantities should be set such that the turbulence is as close to zero as possible, this can lead to some numerical issues, and in the original paper outlining the k-ω SST model, the following recommendations were made to get any 2 of the following farfield quantities: *k*, *ω*, *ε*, viscosity ratio $\left(\frac{\mu_t}{\mu}\right)$, and/or turbulence intensity (*I*) based on two user-definable coefficients, α_k and α_ω as well as freestream fluid properties and velocity [39]:

$$k_{farfield} = \alpha_k \frac{U_\infty^2}{Re_L} = \alpha_k \frac{U_\infty \nu}{L} = \frac{3}{2}\left(U_\infty I_{farfield}\right)^2 \tag{61}$$

$$\omega_{farfield} = \alpha_\omega \frac{U_\infty}{L} = \frac{3}{2\nu}\left(U_\infty I_{farfield}\right)^2 \left(\frac{\mu_t}{\mu}\right)_{farfield}^{-1} \tag{62}$$

$$\left(\frac{\mu_t}{\mu}\right)_{farfield} = \frac{k_{farfield}}{\nu \omega_{farfield}} = \frac{\alpha_k}{\alpha_\omega} \tag{63}$$

$$I_{farfield} = \frac{u'_{farfield}}{U_\infty} = \frac{\sqrt{\frac{2}{3} k_{farfield}}}{U_\infty} = \sqrt{\frac{2}{3}\frac{\alpha_k}{L}\frac{\nu}{U_\infty}} \tag{64}$$

$$\epsilon_{farfield} = \frac{C_\mu}{\nu} k_{farfield}^2 \left(\frac{\mu_t}{\mu}\right)_{farfield}^{-1} = C_\mu \alpha_k \alpha_\omega \nu \left(\frac{U_\infty}{L}\right)^2 = \frac{9C_\mu}{4\nu}\left(U_\infty I_{farfield}\right)^4 \left(\frac{\mu_t}{\mu}\right)_{farfield}^{-1} \tag{65}$$

$$10^{-5} < \alpha_k < 0.1 \tag{66}$$

$$1 < \alpha_\omega < 10 \tag{67}$$

Here *L* is a reference length and is recommended to be the approximate the length of the computational domain [39]. Here-in the chord length/diameter of the object in question are used for a fixed reference length, even while the effects of changing mesh size are being tested. This means that the true recommended farfield turbulence intensity may be smaller if the true total computational domain was used as the reference length using the same chosen coefficients, α_k and α_ω, with a factor of 10 difference which is common reducing the recommended value by about a factor of 3. The coefficient α_k is chosen as 0.05 and α_ω is chosen to be 5 which gives an eddy viscosity ratio of 0.01 and a turbulence intensity which is approximately in the middle of the acceptable range based on the modified reference length, which may be a bit on the high side of the acceptable range if the actual computational domain length was used as a reference length. The results were found to not be too sensitive to this. If the farfield turbulence intensity is reduced by a factor of 100 for the Shin and Bond 2 Case described in Table 20, the total collection efficiency was found to be reduced by 0.34% when using FENSAP-ICE.

2.11. Boundary Evolution

2.11.1. Square Test Setup

To test the algorithms used for boundary evolution, a 1x1 m square was inscribed inside a 2x2 m mesh with 100x100 square cells. This shape, seen inside the mesh in Fig. 19, is grown outwards at an equal rate of 0.025 m/s in each direction over 10 s with 1E-3 s implicit Euler method time-steps.

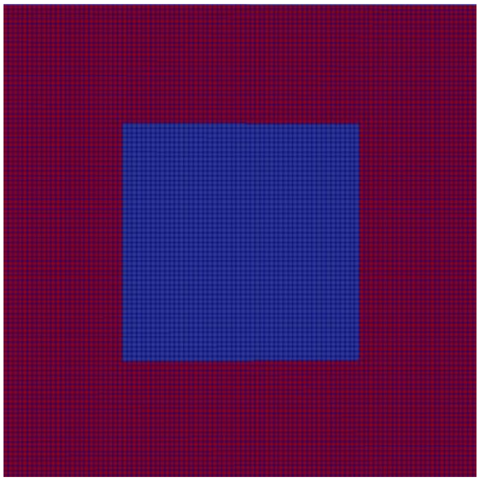


Fig. 19: Initial square and mesh used in square test case setup. Cells inside of the square are shown in blue and cells outside are shown in red.

2.11.2. Boundary Velocity

The boundary evolution module starts with the collection efficiency curve along the boundary coming from the droplet module. The flux of impinging droplets along the wall is calculated from the following [16]:

$$m'_{imp,b} = C_{E,b} U_\infty \alpha_\infty \rho_d \tag{68}$$

Where $m'_{imp,b}$ and $C_{E,b}$ are the impinging mass flux and collection efficiency over the portion of the boundary b respectfully, U_∞ and α_∞ are the freestream droplet velocity and droplet volume fraction respectfully, and ρ_d is the droplet water density, approximated as a constant 1000 kg/m^3 here-in.

In order to evolve the global level set field, the local boundary values of the impinging mass fluxes or the resulting velocity field must be convected outwards, with the impinging mass fluxes being used here as it is more efficient to evolve a scalar then a vector in OpenFOAM®. As the fluxes reach the farfield boundaries, a boundary condition must be applied, and here-in zero-gradient boundary conditions are used. The direction that the level set field will be assumed to evolve in is important here as the impinging mass fluxes should be evolved along this direction and its reverse from the boundary. The boundary is here-in assumed to evolve in the normal direction, an assumption which is not unquestionable, especially for small numbers of icing steps, but one which was made by many other researchers, such as Cao, Y. et al. (2008) [40]. The resulting two-way convection equation used to describe the evolution of the impinging mass fluxes off of the boundary is as follows:

$$\left(\begin{array}{cc} \boldsymbol{m'_{imp}} = m'_{imp,b} & boundary\ cells \\ \nabla \cdot \left(sign\phi\,\vec{n}\,\boldsymbol{m'_{imp}}\right) - \boldsymbol{m'_{imp}}\nabla \cdot sign\phi\,\vec{n} = 0 & fluid\ cells\ and\ dead\ cells \end{array}\right) \tag{69}$$

To discretize this into FVM formulation the signed-normal field $(sign\phi\,\vec{n})$ must be interpolated to the faces. This is done via linear interpolation of the normal, and the sign field is set to the sign of the value of the level set field linearly interpolated to the faces. The resulting solution scheme becomes the following for fluid and dead cells in the inner region with square cells, with a similar discretization being done for cells outside of this region, but the linear interpolation weights for the signed-normal are not 0.5 for the cell and 0.5 for its neighbour as in the inner region case:

$$\sum_{faces\ f} \left(\boldsymbol{m'_{imp,s}} - \boldsymbol{m'_{imp,c}}\right)\overrightarrow{A_f}\left(\frac{\phi_c + \phi_n}{|\phi_c + \phi_n| + SMALLVAL}\frac{\overrightarrow{n_c} + \overrightarrow{n_n}}{2} \cdot \overrightarrow{n_f}\right) = 0 \tag{70}$$

Where $m'_{imp,c}$ is the impinging flux term at the cell centre in question, $m'_{imp,s}$ is the combination of impinging flux term from the cell centre and neighbouring cell of the face f based on the discretization of the divergence scheme, $\overrightarrow{A_f}$ is the area of the face *f*, $\frac{\phi_c+\phi_n}{|\phi_c+\phi_n|}$ is the sign at the face *f* based on linear interpolation, $\overrightarrow{n_f}$ is the normal of the face *f* itself while $\frac{\overrightarrow{n_c}+\overrightarrow{n_n}}{2}$ is the interpolation of the normal direction to the boundary to the face *f*, and SMALLVAL is a small value to avoid divide-by-zero errors with 1E-20 being used here-in. This equation was solved using a zero-gradient condition on all farfield boundaries.

This methodology requires some relaxation to achieve steady-state, with 0.5 or higher being found to be usable, but 0.3 was used to be safe. The PBiCGStab algorithm implemented in OpenFOAM® 5.0 was used as this was also found to improve stability.

Using the assumption that the level set field evolution velocity $\overrightarrow{v_\phi}$ at each point will be in the direction normal to the boundary, $\vec{n}$, this quantity is then calculated for each cell from the impinging flux field as follows:

$$\overrightarrow{v_\phi} = \frac{m'_{imp}}{\rho_{ice}}\vec{n} \tag{71}$$

Where ρ_{ice} is the density of this ice. A constant value of 917 kg/m^3 is used here-in for all ice densities. Based on experiments from Vargans, M. et al. (2007) the real value of the local ice density may vary slightly, but 917 kg/m^3 is generally a good representation for rime icing, and it was noted that dramatic changes in this value only occur when air pockets get trapped in the ice [41].

In Fig. 20 it can be seen that the algorithm described is capable of convecting the velocity/impinging fluxes for the square test case. In the evolution of the square however, the true solution will be used to eliminate any discretization error from this step in the evolution. The velocity was evolved directly in this case instead of the impinging fluxes since the velocities were not being calculated from a collection curve but were instead directly user-defined.

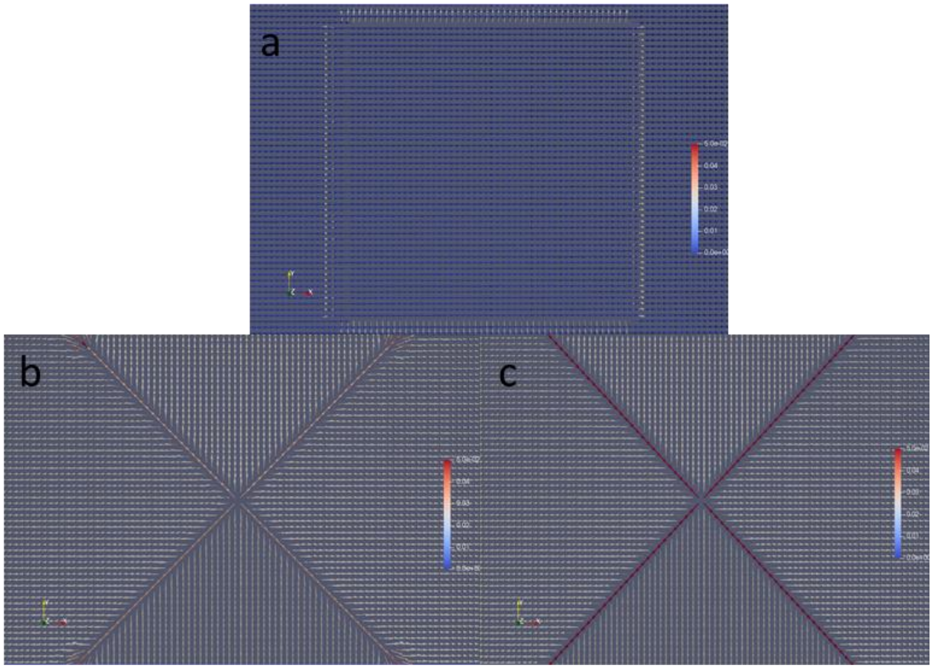


Fig. 20: Square test velocity convection test. Level set evolution velocity from 0 to 0.5 m/s is shown. a) Velocity is described only at boundary cells. b) After velocity extension. c) True solution.

2.11.3. Boundary Evolution

The only unsteady process in the algorithm which is taking place over the space of "real-time" in the simulation is the evolution of the level set field. This field is evolved by the unsteady convection equation shown below:

$$\frac{d\boldsymbol{\phi}}{dt} = \nabla\cdot\overrightarrow{v_\phi}\boldsymbol{\phi} - \boldsymbol{\phi}\nabla\cdot\overrightarrow{v_\phi} \tag{72}$$

Here ϕ is the level set field as normal, but for boundary cells modifications for any changes to the centre are not accounted for.

This equation is discretized using the first-order accurate, implicit, backwards Euler scheme for time. For cells in the inner region the discretization will look as follows, with a small difference to the weights of the velocity interpolation to the faces of outer region cells as the weighting may not be 0.5 on both the cell and neighbour:

$$\frac{\boldsymbol{\phi}_c^{\boldsymbol{i}} - \phi_c^{i-1}}{\Delta t} = \sum_{faces\, f} (\boldsymbol{\phi}_s^{\boldsymbol{i}} - \boldsymbol{\phi}_{\boldsymbol{c}}^{\boldsymbol{i}})\overrightarrow{A_f}\left(\frac{\overrightarrow{v_{\phi,c}} + \overrightarrow{v_{\phi,n}}}{2} \cdot \overrightarrow{n_f}\right) \tag{73}$$

Where ϕ_c^i is the level set field value of the cell in question at the next time-step, ϕ_c^{i-1} is the level set field value of the cell in question at the previous time-step, ϕ_s^i is a combination of the level set field value of the cell in question and its neighbours at the next time-step with weights coming from the divergence scheme, $\overrightarrow{A_f}$ and $\overrightarrow{n_f}$ are the area and normal direction of the face f, $\frac{\overrightarrow{v_{\phi,c}}+\overrightarrow{v_{\phi,n}}}{2}$ is the velocity field at the face f based on linear interpolation for 2 equal-sized squares. This equation was solved by using a zero-gradient condition on all farfield cells. While a gradient of 1 condition would technically be more accurate, these boundaries are expected to be far enough away from the boundary that this boundary condition does not impact the solution near the boundary where the accuracy matters.

The question of when to stop the evolution of the level set field is not as obvious as it may at first seem. For a 10-minute growth, the growth could terminate after 10 minutes which will be tested, however, this is expected to generate an issue for concave and convex surfaces as shown below in Fig. 21. A concave surface should generate an unexpectedly low ice mass and a convex surface should generate an unexpectedly high ice mass based on the initial estimate from the impinging mass flux over the surface [7]. Other minor sources if conservation loss also exists, namely discretization errors in convection and reinitialization can contribute to this as will be shown shortly.

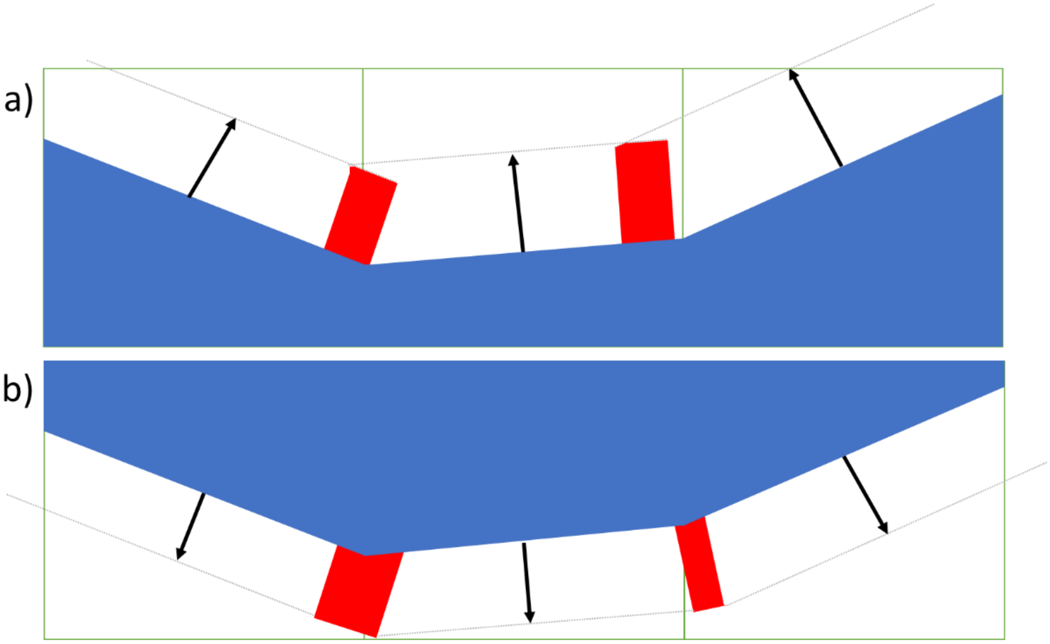


Fig. 21: (a) Concave and (b) convex boundary evolution with approximate regions of conservation error shown in red. The original solid region is shown in blue, with the new surface being shown in a grey dotted line. The black arrows describe the evolution velocity multiplied by the time for evolution.

One way of viewing the issue shown in Fig. 21 (b) is to imagine a 1 m radius cylinder of depth 1 m, a convex shape. This cylinder is growing outwards at 1 m/min everywhere for 2 min. In the first minute it grows from a π m^3 volume to a 4π m^3 volume as its radius grows to 2 m, or it experiences a 3π m^3 growth. In the second minute, it grows to 9π m^3 as its radius grows linearly to 3 m, in this minute a growth of 5π m^3 is experienced, almost twice that of the last minute. The velocities are based on the initially impinging mass of droplets, and therefore the volume grown from one-time step to the next should not change, as this initially impinging mass does not change, and the final mass should reflect the mass of the droplets initially estimated to hit the surface. This is only true if it is assumed that the rime ice tends to accrete linearly over the time period, something that will be shown in later tests to be mostly true.

For complex shapes this may be difficult to enforce locally especially as reinitialization and discretization errors are considered, however, to combat this globally and enforce conservation of the global ice mass, the following end goal is recommended and will be used unless otherwise specified:

$$m^0 = \iiint \rho_{solid} dV^0 \approx \sum\nolimits_{boundary\ cells} \rho_{solid}(V_{cell} - V_{cut}) + \sum\nolimits_{dead\ cells} \rho_{solid}\, V_{cell} \tag{74}$$

$$\Delta m^i = \iiint \rho_{solid} dV^i - m^0 \approx \sum\nolimits_{boundary\ cells} \rho_{solid}(V_{cell} - V_{cut}) + \sum\nolimits_{dead\ cells} \rho_{solid}\, V_{cell} - m^0 \tag{75}$$

$$\Delta m_t = \Delta t_t \oiint m'_{imp} ds \approx \Delta t_t \sum\nolimits_{boundary\ cells} m'_{imp} A_{cut} \tag{76}$$

$$EvolutionStopCondition\colon \Delta m^i \geq \Delta m_t \tag{77}$$

Where ρ_{solid} is the density of the solid region which is here-in taken to be equal to the constant icing density of 917 kg/m^3, Δt_t is the total amount of time for the growth period, m'_{imp} is the mass impinging flux curve of the droplets over the surface *s*, m^0 and V^0 are the initial mass and volume of the solid respectfully, V_{cell} is the cell volume, V_{cut} is the cut-cell volume (the reduced cell volume representing the volume of the fluid), A_{cut} is the cut-cell area, Δm^i and V^i are the ice mass that has grown and total solid volume at the current iteration in time, Δm_t is the total mass of impinging droplets, and *EvolutionStopCondition* is the condition at which the boundary evolution stops.

Any deviations over/under the stop condition reflect a lack of conservation. In order to minimize this error, it is important to put an appropriate limiter on the time-step itself as the system approaches the stop conditions. The following is recommended as an upper limit for the time-step when studying a convex shape under accretion:

$$\Delta t_{max} = \max\left(\Delta t_{min},\ \Delta t_t\left(1 - \frac{\Delta m^i}{\Delta m_t}\right)\right) \tag{78}$$

$$\Delta t = \min(\Delta t_{max}, \Delta t_{prescribed}) \tag{79}$$

Where Δt is the time-step for the next iteration, Δt_{min} is a user-defined minimum time-step, with 1E-4 s being used here-in, Δt_{max} being the time-step limiter, $\Delta t_{prescribed}$ is the desired time-step, and Δt_t / Δm^i / Δm_t are the same as defined previously for the stop condition. This description was developed based on the maximum time limit for a locally flat surface, and so would under-estimate the time required for a concave leading to the requirement of a user-defined Δt_{min}. While this still over-estimates the total amount of mass growth for a convex surface, and a final non-conservative reinitialization is performed after the growth finishes, in-practice using this method the total mass growth was always within 0.05% of the initially predicted growth from the impingement curve was in all tests seen, and usually closer to about 0.01%, even for a $\Delta t_{prescribed}$ being 1% or more of the Δt_t.

So, while this model, referred to here-in as the “conservative model” for the time-step is not truly conservative, it has improved global conservation of ice mass than the simpler model where-in a 10-minute test would be run for 10 minutes, which is referred to here-in as the “non-conservative model”. This topic will be further explored in the NACA23012 Case.

Each other property (pressure, velocity, k, ε, droplet concentration, and droplet velocity) is also convected alongside the level set field via the same equation which is shown for an arbitrary property F as follows:

$$\frac{d\boldsymbol{F}}{dt} = \nabla \cdot \overrightarrow{v_\phi}\boldsymbol{F} - \boldsymbol{F}\nabla \cdot \overrightarrow{v_\phi} \tag{80}$$

This evolution was found to reduce the required number of iterations by up to 90% in some cases compared to restarting by setting every value of the field to the freestream value.

2.11.4. Reinitialization

Reinitialization ensures properties of the level set field after the convection of the level set field causes steepening and/or flattening of the gradient of the level set field, which invalidates the used properties of the field around the interface [42] [20]. There are different types of reinitialization to be discussed with different orders of speed, ease of implementation, and accuracy.

The fast marching method which is based on ordering the set of all level set field from closest to the boundary to the farthest and then basing all level set values sequentially on closer elements that have already been solved for, thus attempting to implement property 1 of the level set field toolbox directly. This methodology tends to demonstrate the best time scaling of the 3 discussed, scaling with O($n \log n$) for n cells due to the cost for the ordering of the field [43]. Using this methodology, it is difficult to achieve higher than first-order accuracy and requires special treatment of the initial band of points around the interface or use of other methods in this region [20].

The direct reinitialization methodology, like the direct initialization scheme used to create the level set field, is based-on directly satisfying property 1 of the level set field by reconstructing the level set field using a set of segments, calculating the minimum distance from each cell centre to each segment, and then taking the minimum of this set for each cell centre and multiplying it by the sign of the level set field that that cell had before reinitialization, with this method being fundamentally easily scaleable to general 3D and unstructured grids, as done by Ngo, L.C. and Choi, H.G. (2017) [44]. This method does require boundary reconstruction for higher than first-order accurate reinitializations, but since a linear reconstruction of the boundary is already done for calculating the total amount of ice, the cut-cell formulation, etc. and since distance from each cell centre to each line segment is an algorithm that was already implemented for level set field construction, this algorithm is relatively easy to implement and its accuracy scales with the accuracy of the boundary reconstruction. The downside is that it is limited by any assumptions made in the boundary reconstruction (no sharp points, 2D, constant ice density) and that it is relatively slow compared to the other methodologies without modification/limiting the algorithm to certain regions, with a time-scaling of O(n b) for n cells and b line segments comprising the boundary.

The final method looked at, often call the Hamilton-Jacobi method, attempts to directly satisfy property 4 of the level set field which is that the magnitude of the gradient of the level set field should be 1 everywhere. It is commonly used in the following unsteady form and evolved in pseudo-time until a steady-state is reached [20].

$$\frac{\partial \boldsymbol{\phi}}{\partial t} = sign\phi(1 - |\nabla \boldsymbol{\phi}|) \tag{81}$$

This equation is difficult to use since it is not in an easily FVM-discretizable form and requires pseudo-time-steps. Instead, the following steady-state FVM-discretizable equation was derived and used here-in:

$$\begin{gathered} \nabla \cdot \left(sign\phi_{original}\,\vec{n}\,\boldsymbol{\phi}\right) - \boldsymbol{\phi}\nabla \cdot sign\phi_{original}\,\vec{n} = sign\phi_{original} \\ \nabla \cdot \left(\frac{\phi^0}{|\phi^0|}\frac{\nabla\phi}{\|\nabla\phi\|}\boldsymbol{\phi}\right) - \boldsymbol{\phi}\nabla \cdot \left(\frac{\phi^0}{|\phi^0|}\frac{\nabla\phi}{\|\nabla\phi\|}\right) = \frac{\phi^0}{|\phi^0|} \end{gathered} \tag{82}$$

Where $sign\phi_{original} = \frac{\phi^0}{|\phi^0|}$ is the sign of the level set field before reinitialization as this should not change since ideally the reinitialization algorithm only changes things outwards from the original boundary. This equation is the same as the normal direction convection equation used for the impinging fluxes previously but with a source term of 1, although it is complicated by the fact that the normal may now change from iteration to iteration, adding non-linearity to the problem. The same discretization, solution, and relaxation schemes were used for this as for the impinging fluxes however with no problem. The resulting discretized equation is as follows for inner region cells, with similar a discretization for other cells except for that different coefficients than 0.5 being used for the linear interpolation of the signed-normal $\left(sign\phi_{original}\,\vec{n} = \frac{\phi^0}{|\phi^0|}\frac{\nabla\phi}{\|\nabla\phi\|}\right)$ lead to changes in this term:

$$\sum_{faces\,f}(\boldsymbol{\phi}_s - \boldsymbol{\phi}_c)\overrightarrow{A_f}\left(\frac{1}{2}\frac{\phi_c^0 + \phi_n^0}{|\phi_c^0 + \phi_n^0| + SMALLVAL}\left[\left(\frac{\nabla\phi}{\|\nabla\phi\| + SMALLVAL}\right)_c + \left(\frac{\nabla\phi}{\|\nabla\phi\| + SMALVAL}\right)_n\right]\cdot\overrightarrow{n_f}\right) = \frac{\phi_c^0}{|\phi_c^0| + SMALLVAL} \tag{83}$$

Where ϕ_c is the level set field value of the cell in question, ϕ_s is a combination of the level set field value of the cell in question and its neighbours with weights coming from the divergence scheme, $\overrightarrow{A_f}$ and $\overrightarrow{n_f}$ are the area and normal direction of the face f, $\frac{\phi_c^0+\phi_n^0}{|\phi_c^0+\phi_n^0|}$ is the sign of the level set field at the face *f* based on linear interpolation, $\frac{\phi_c^0}{|\phi_c^0|}$ is the sign of the current cell in question, $\left(\frac{\nabla\phi}{\|\nabla\phi\|}\right)_c$ and is $\left(\frac{\nabla\phi}{\|\nabla\phi\|}\right)_n$ the normal direction at the cell in question and its neighbour at the face *f* respectfully and SMALLVAL is a small value used to avoid divide-by-zero errors with 1E-20 being used here-in. This equation was solved by using a zero-gradient condition on all farfield cells. Again, due to the large expected distance from the farfield to the boundary, the error in the near boundary region from using this boundary condition instead of a gradient of 1 condition which it should be is not expected to be very large.

This reinitialization equation was found to be faster in the tests it was used in then direct reinitialization, with similar accuracy. This was the method used in all tests here-in.

In Fig. 22 the evolution of the level set field can be seen. Part (a) shows the original level set field from -0.1 m to +0.1 m while (b) shows the field after evolution, (c) shows (b) after reinitialization, and (d) shows the true level set field of the desired final object. The true velocity field is used, but due to discretization errors, the solution of just convecting the level set field, shown in part (b) of the figure, has some small errors around the corners of the figure, the top-left of which is highlighted with a black circle. In this region, not only is there a difference totaling about a 0.33% loss in area compared to the true solution after the boundary is reconstructed, but also the level set field no longer retains its properties. Reinitialization fixes this last part and it can be seen that comparing the highlighted regions of part (b) and part (c) to part (d) of the figure, the level set field more resembles the true solution after reinitialization than before, but in reality, there is an additional estimated 0.02% mass loss. This is common as reinitialization is not perfectly conservative for the interior area [20].

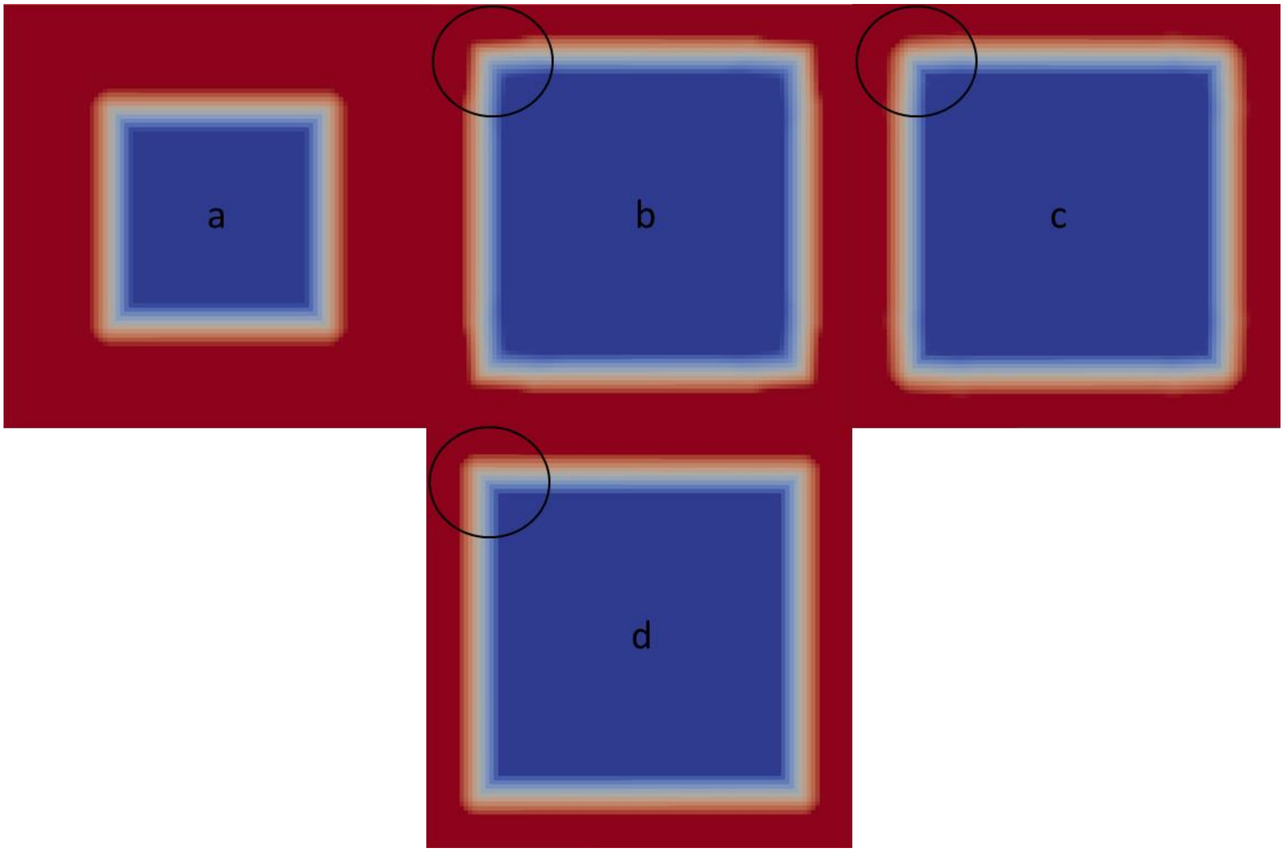


Fig. 22: Square level set field convection test. Level set field from -0.1 m to +0.1 m is shown. a) Before convection. b) After convection, no reinitialization. c) After convection with reinitialization via the reinitialization equation. d) True solution. The top-left corner is highlighted in (b), (c), and (d) with a black circle to show the impact of reinitialization.

2.1. Post Processing

2.1.1. Collection Efficiency and Total Collection Efficiency

The local collection efficiency, *C_E*, is defined when using a Eulerian methodology from the following [45]:

$$C_E = \max\left(0, \frac{-(\overrightarrow{U_d} \cdot \vec{n})}{U_\infty} \frac{\alpha}{\alpha_\infty}\right) \tag{84}$$

Where U_∞ and α_∞ are the freestream velocity and droplet volume fraction respectfully, and $\vec{n}$ is the normal direction.

Since the boundary condition wherever the collection efficiency is non-zero is that the rate of change of the droplet volume fraction and velocity fields are zero in the direction normal to the surface, the collection efficiency at a position normal to a boundary cell centre along the corresponding surface can be obtained to a second-order accuracy by using directly the values for the droplet volume fraction and velocity of that boundary cell.

Further, in 2D, the collection efficiency can be integrated over the surface contour, s, of a cell with a cut face area of $A_{cutFace}$, a reference length of L_{Ref} (such as the chord of an airfoil), and a cell span of w_c to give the total collection per reference length, C_T, as such:

$$C_T = \oint \frac{C_E}{A_{ref}} ds \approx \sum \frac{A_{cutFace}}{w_c L_{Ref}} C_E \tag{85}$$

When drawing the collection efficiency curves, the y-axis is simply the collection efficiency extracted out from the boundary cells as described, but for the x-axis, a boundary point corresponding to these efficiencies must be extracted. These points must be shifted and rotated to align the x-axis with the chord of the airfoil for airfoil tests, and the tip with (0,0). These operations are done via the following set of equations:

$$\vec{p}_{curve} = \vec{p}_b - \phi\vec{n} - \overrightarrow{\text{offset}} \tag{86}$$

$$\vec{p}'_{curve} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \vec{p}_{curve} \tag{87}$$

Where $\vec{p}_{curve}$ refers to an unrotated position on the curve, $\vec{p}'_{curve}$ refers to the rotated position on the curve, $\vec{p}_b$ refers to the centre of a boundary cell, $\vec{n}\phi$ is the level set field value multiplied by the normal direction and refers to the nearest distance from the boundary cell centre to the wall in the direction pointing away from the wall evaluated at the boundary cell centre, the $\overrightarrow{\text{offset}}$ refers to a constant vector which offsets each point on the curve to match that the tip is at (0, 0), and θ refers to the angle of attack for an airfoil case or 0^{o} for the cylinder test case

For the x-axis of airfoil test cases, the direction perpendicular to the chord (the y coordinate of $\vec{p}'_{curve}$) over the chord length is used. For the cylinder case, the arc length over the diameter is used. The arc length of the cylinder per diameter, $\frac{L_{arc}}{D}$, at a position along the curve in the y-direction, y_{curve}, is calculated as follows:

$$\frac{L_{arc}}{D} = \frac{1}{2} \frac{y_{curve}}{|y_{curve}|} \sin^{-1}\left(\frac{2|y_{curve}|}{D}\right) \tag{88}$$

2.1.2. Total Solid Mass per Unit Span

The total solid mass is calculated from the following, assuming a constant ice density

$$m_{ice} = \iiint \rho_{solid} dV - \iiint \rho_{solid} dV^0 \approx \sum\nolimits_{boundary\ cells} \rho_{solid}(V_{cell} - V_{cut}) + \sum\nolimits_{dead\ cells} \rho_{solid} V_{cell} - \iiint \rho_{solid} dV^0 \tag{89}$$

Where V_{cell} is the cell, volume V_{cut} is the modified volume of the boundary cells which represents an approximation of the true fluid volume, ρ_{solid} is the density of the solid (for a constant ice density, that density can be used here), m_{ice} is the mass of ice per unit span of the wing, $\iiint \rho_{solid} dV^0$ is the original cell mass, calculated from the original shape using the same algorithm

2.1.3. Ice Shape and Net Ice Accretion

After calculating, offsetting, and rotating a set of points along the curve so that experiments, FENSAP-ICE, and the OpenFOAM®-model are all centred at the tip and rotated so that the chord goes along the x-axis, these points are divided by a reference length (usually the chord) and the x and y directions are plotted against each other, resulting in the ice shape.

For net ice accretion plots, this is done the same way, but the axes are flipped, and from the direction along the chord at each point perpendicular to the chord, the original ice shape is subtracted away. It is assumed that the area of accreted ice per unit span at a particular location in the direction perpendicular to the curve, $\vec{p}'_{curve,y}$, *NetIceAccretion*, is a validly represented as a function of $\vec{p}'_{curve,y}$, so for each location in $\vec{p}'_{curve,y}$ only 1 surface may be present. This notably may not be true for certain airfoils and if there is roughness on the surface. In these cases, the result of these calculations will not be the true amount of ice accreted at that location and a point will be generated for all points at $\vec{p}'_{curve,y}$. The net ice accretion is calculated per chord length as follows:

$$\frac{NetIceAccretion}{c}\left(\frac{\vec{p}'_{curve,y}}{c}\right) = -\left(\frac{\vec{p}'_{curve,x}}{c}\left(\frac{\vec{p}'_{curve,y}}{c}\right) - x_{airfoil}\left(\frac{\vec{p}'_{curve,y}}{c}\right)\right) \qquad (90)$$

The original airfoil shape along the chord at each point perpendicular to the chord, centred at the tip, and per unit chord, represented by $x_{airfoil}$are approximated via the piecewise functions.

For the NACA0012 airfoil, which was the most common airfoil studied, the value of $x_{airfoil}$ is based on the rotated and shifted y-direction values of the curve scaled by the chord length: $y = \left|\frac{\vec{p}'_{curve,y}}{c}\right|$. The final region was difficult to approximate accurately with a simple polynomial so instead, the following was used: $y_* = \cos^{-1}\left(\frac{y}{max(y)}\right) \approx \cos^{-1}\left(\frac{y}{0.0600}\right)$. The reason this case was broken up into the 6 cases seen in Table 2 instead of fewer, was to meet the accuracy requirements for very short icing tests where inaccuracies in this model may be on the order of the amount of accreted ice.

Table 2: NACA0012 piecewise front-end profile for net ice-accretion profile calculations.

Region	Model	R^2
y≤0.01	$x_{airfoil} = 1492.0996487y^4 + 144.65210051y^3 + 31.531291231y^2 - 2.0390630880 * 10^{-5}y$	0.99999999998
0.01<y|≤0.02	$x_{airfoil} = 3803.4966240y^4 + 45.650581300y^3 + 33.191817919y^2 - 0.01.2815431194y + 3.7804182936 * 10^{-5}$	1.0000000000
0.02<y≤0.03	$x_{airfoil} = 10048.873558y^4 - 476.86587867y^3 + 49.767998128y^2 - 0.24882678495y + 1.3084892929 * 10^{-3}$	0.99999999995
0.03<y≤0.035	$x_{airfoil} = 23635.466064y^4 - 2083.7106372y^3 + 121.25748733y^2 - 1.6668856614y + 0.011889433111$	0.99999999991
0.035<y≤0.04	$x_{airfoil} = 48118.262329y^4 - 5548.9005522y^3 + 305.35712491y^2 - 6.0181437341y + 0.050491943693$	1.0000000000
0.04<y≤0.045	$x_{airfoil} = 117709.34912y^4 - 16821.907045y^3 + 990.66663908y^2 - 24.548043629y + 0.238512013681$	1.0000000000
0.045<y≤0.05	$x_{airfoil} = 385291.48389y^4 - 65662.736464y^3 + 4335.5492270y^2 - 126.41433705y + 1.4024748697$	0.99999999983
0.05<y≤0.055	$x_{airfoil} = 2420171.9927y^4 - 479969.60714y^3 + 35979.428626y^2 - 1200.9525608y + 15.090045343$	0.99999999552
y>0.055	$x_{airfoil} = 0.80603829026y_*^6 - 1.0603640527y_*^5 + 0.53781717271y_*^4 - 0.12203339115y_*^3 + 0.11089801497y_*^2 - 0.36520073906y_* + 0.29983931048$	0.99999994390

The NACA23012 airfoil was only used in only 1 net accretion result and thus was not required to be accurately curve-fit over the entire region and instead was fit only to the region from $-0.03 \le y = \frac{\vec{p}'_{curve,y}}{c} \le 0.06$.

Table 3: NACA23012 piecewise front-end profile for net ice-accretion profile calculations.

Region	Model	R^2
-0.03≤y≤-0.015	$x_{airfoil} = 2876298030.9y^6 + 381556528.13y^5 + 20288932.062y^4 + 552790.00408y^3 + 8344.3467558y^2 + 64.968935073y + 0.21062884980$	0.99999999450
-0.015<y≤0	$x_{airfoil} = -11320592.000y^6 - 2828717.6094y^5 - 17530.091309y^4 - 1234.9613266y^3 + 36.230243117y^2 - 0.30697782792y$	0.99999999793
0<y≤0.03	$x_{airfoil} = 432183.75781y^6 - 68108.921631y^5 + 8062.1980515y^4 - 502.52797604y^3 + 37.650770792y^2 - 0.30494894973y$	0.99999999950
0.03<y≤0.06	$x_{airfoil} = 12722842.547y^6 - 3028734.0016y^5 + 305554.97032y^4 - 16403.326283y^3 + 513.07303302y^2 - 7.8288109860y + 0.049168380924$	1.0000000000

2.1.4. Richardson Extrapolation

In analyzing approximate remaining errors due to discretization, Richardson extrapolation is utilized. For a variable f on 2 grids A and B, usually the most refined grids tested, with representative spacings Δx_A and Δx_B, the mesh independent result, f_*, is determined by the following [46]:

$$f_* = f_A + \frac{f_A - C_{T,8}}{\left(\frac{\Delta x_B}{\Delta x_A}\right)^n - 1} \tag{91}$$

Where n is the scaling of the model. An approximation of n=1 is used in places, which overestimates any approximate errors compared to higher scalings if higher scalings are more appropriate. This is done to give a high-end estimation of the error where the scaling is difficult to determine.

2.2.General Mesh Generation and Boundary Conditions

Unless specified otherwise, meshes used for testing and validation of the OpenFOAM®-based model are generated with 2 regions, referred to here-in as the inner region and the outer region. The inner region is a rectangle which consists of square Cartesian cells, and the outside region is a rectangle encompassing the inner region with a constant expansion rate out from the inner region. To simplify the design and testing of the meshes, all meshes keep the expansion rate out from the inner region in each direction at a single expansion ratio. The inner region extends out from the closest point on the boundary on the left minus 4 cell lengths to the closest point on the boundary on the right plus 4 cell lengths and the same bottom-to-top. The outer region is a specified number of chord lengths from the boundary edge.

On the left-side of the boundary of the outer region, an inlet boundary condition is placed in which all quantities being transported but the pressure (velocity, k, ε, droplet volume fraction, and droplet velocity) are set to fixed inlet values and the pressure is set to a zero-gradient condition. On the right-side, an outlet boundary condition is placed where all quantities being transported but pressure are set to zero-gradient conditions, and pressure is set to a fixed value. On the top and bottom freestream conditions are used in which if the fluid is entering from the boundary an inlet condition is used and if it is entering into the boundary an outlet condition is used. The object in the centre is rotated until the flow is aligned such that the airflow at the inlet on the left-most face of the outer region is aligned perfectly with the inlet face.

This setup and a sample case with, a 1.14E-3 m minimum cell length and a 0.54x0.076 m inner region, as well a 5% expansion rate, and a distance from the boundary to the far edge of 5.3 m in Fig. 23.

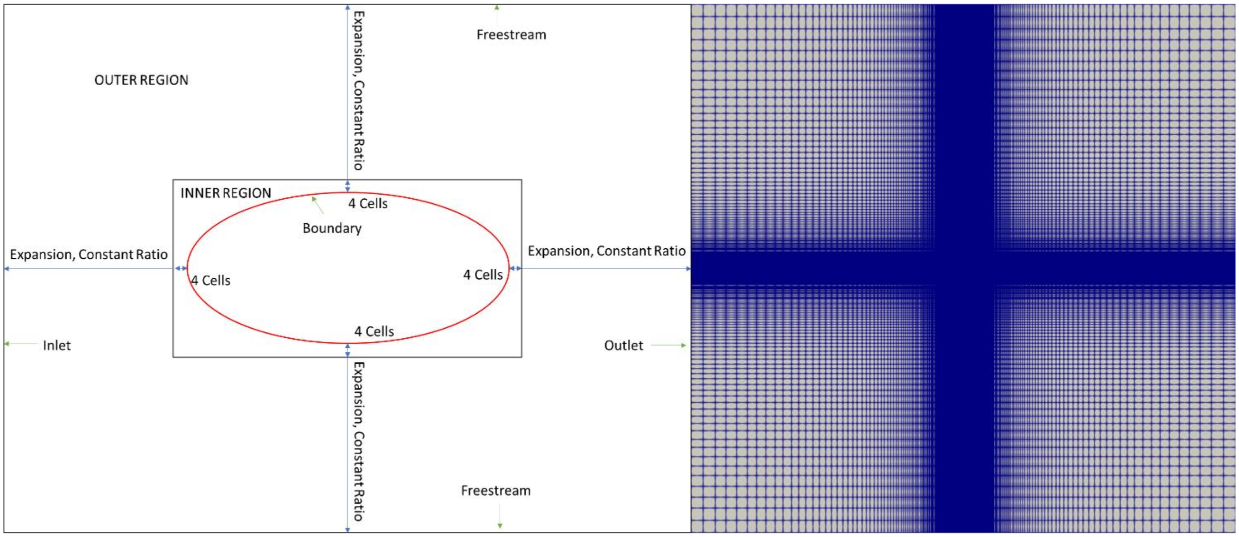

OUTER REGION
Expansion, Constant Ratio
Freestream
INNER REGION
4 Cells
Boundary
Expansion, Constant Ratio
4 Cells
4 Cells
Expansion, Constant Ratio
4 Cells
Inlet
Outlet
Expansion, Constant Ratio
Freestream

Fig. 23: General OpenFOAM® case setup for airfoil and an example mesh with a 5.3 m outer region, a 5% expansion rate, an 8E-4 m minimum cell length, and a 0.54x0.076 m inner region.

For FENSAP-ICE, an O-grid structured mesh is created. For cylinders, the domain is split into 2 equal zones along the y-axis. For airfoils the domain is broken up into 4 zones: the first zone goes over the front end of the airfoil to 0.3 chords down the airfoil and half of the outer ring; the second and third zones go from the front end ends to the tips of the cut back end to the back minus 2⁰ cut out of the centre, the final region from the cut back end to the 2⁰ cut out of the outer ring. A constant expansion rate is placed from the airfoil to the outer regions. The number of divisions along zone 2 and 3 are set equal to half the number of divisions along zone 1 each, and a bias factor of 10 is placed along the divisions at the front-end with the tip being the most refined. The backend must be defined independently and is set such that the outer ring is approximately evenly spaced when going from zone 2 to 4 to 3.

The airfoil is set to an adiabatic no-slip wall, the outer region of zone 1 is set to a subsonic inlet, and the outlet boundaries are set to a subsonic outlet with radial equilibrium removed. Fairfield turbulence is set through turbulence intensity and eddy/laminar viscosity ratio.

This setup and a sample mesh with a 1.83 m chord NACA23012 airfoil, with a 1 chord outer radius, a 5% expansion rate and 50 cells perpendicular to the airfoil, 1 division on zone 4, 25 on zone 2 and 3, and 50 on zone 1 are given in Fig. 24.

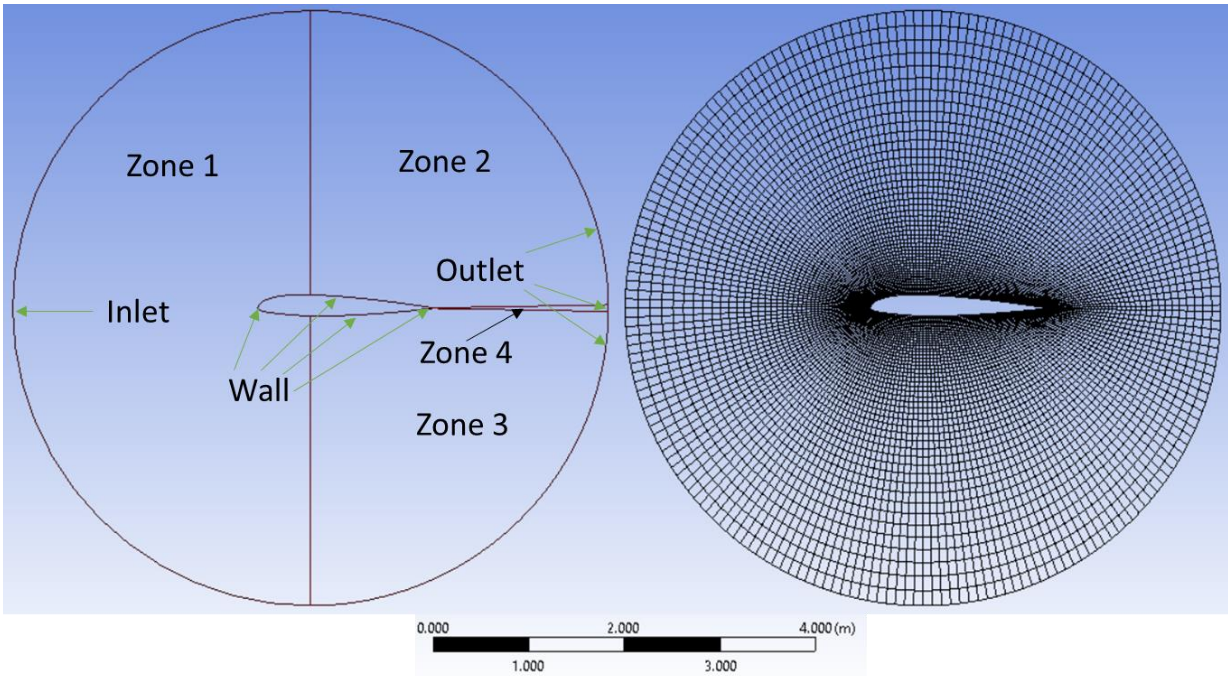


Fig. 24: General FENSAP-Ice airfoil case setup and example mesh for a 1.83 m chord NACA23012 airfoil, with a 1 chord outer radius, a 5% expansion rate and 50 cells perpendicular to the airfoil, 1 division on zone 4, and 200 cells along the rest of the airfoil.

2.3.FENSAP-ICE

The main source for comparable data in this study is the commercial solver owned by ANSYS, Inc. The methods used by FENSAP-ICE assume a body-fitted mesh and thus are used to compare to the non-conformal methodology proposed here-in.

Unlike the solver built for this study in OpenFOAM®, FENSAP-ICE uses a finite element method solver, and not a finite volume method.

The air is modeled using a compressible, non-isothermal solver. For this study, all tests will be using adiabatic conditions applied to the walls for FENSAP-ICE. Since the k-ε model is not offered by FENSAP-ICE for the air, the k-ω SST model will be used instead.

The droplets are modeled using a Eulerian methodology [16].

Many aspects of the boundary growth model are not included in the user manual, such as how conservation is maintained, how exactly “concavity fixing” works and the direction of ice growth which is assumed for each time-step.

The average residual target for the air solver was set to 1E-10, local time-stepping was used with a CFL number of 200, streamline upwind methodology was used with 90% second-order methodology being used and a 1E-7 artificial viscosity. For the droplet model, the CFL for local time-stepping was changed to 20, for convergence a 1E-8 residual target was used along with a maximum change in the total beta of 1E-10. For ice accretion, a concavity fix of 80º was used along with automatic time-stepping.

3. Cut-Cell and Immersed Boundary Method Comparison

There were 2 methodologies developed for the near-wall droplet phase discretization: a cut-cell methodology and an immersed boundary methodology. To determine which of these should be used for later tests as well as the validity of these models, these 2 methods were compared against each other and FENSAP-ICE as well as experimental results for the collection efficiency. It was determined that while the two methodologies converge to the same answer, the cut-cell methodology is approximately twice as accurate in terms of the total collection efficiency as the IBM method and that when using the IBM this discretization error can be a dominant error in the simulation.

3.1. Cylinder Case

3.1.1. OpenFOAM® Model Case Setup

The purpose of this case was to isolate and show the scaling of the 2 wall models, and thus less attention is put on identifying other errors in the system. This test may have discretization errors from the expansion ratio or from the air side of things that will not be addressed and thus it is expected that the results from the OpenFOAM®-model and the FENSAP-ICE case will have a larger difference than in other cases.

A 0.1016 m cylinder was studied with a freestream 40 diameters away from the edges. The freestream velocity is set at 80 m/s, the eddy/laminar viscosity ratio at 0.01, the turbulence intensity at 2.6E-4, the droplet volume fraction at 1E-6, and the droplet diameter at 16 μm. The Mach number of the case is estimated at 0.24 and the Reynold's number based on the diameter at 5E5. A full breakdown of the case can be seen in Table 15.

In the OpenFOAM®-based model, the freestream was modeled by a zero-gradient pressure, fixed value for other quantities inlet condition on the left side; a fixed value pressure, zero-gradient for other quantities outlet condition; and a freestream condition on the top and bottom. The density is set at 1.10 kg/m^3 and the dynamic viscosity was set to 1.78E-5 Pa·s.

For the proposed solver, a structured mesh was created as in the general case, but the mesh was cut in half and a horizontal symmetry plane was placed on the bottom. From the base case shown in Fig. 25, which contains 371x181 cells, 0.1036x0.0518 m inner region and a 5% expansion rate, the velocity field shown in Fig. 26 is obtained from the air solver. To isolate the effects of the droplet phase solver discretization, and to better understand the effects of the boundary models, this air result is then linearly interpolated onto grids of successive refinement using the *mapFields* utility built into OpenFOAM® 5.0. The base grid is referred to as "2", having an average inner region cell side length of 2.8E-4 m. The least refined case, "1" has a side length of 5.6E-4 m. The most refined case, "16", has a side length of 3.5E-5 m.

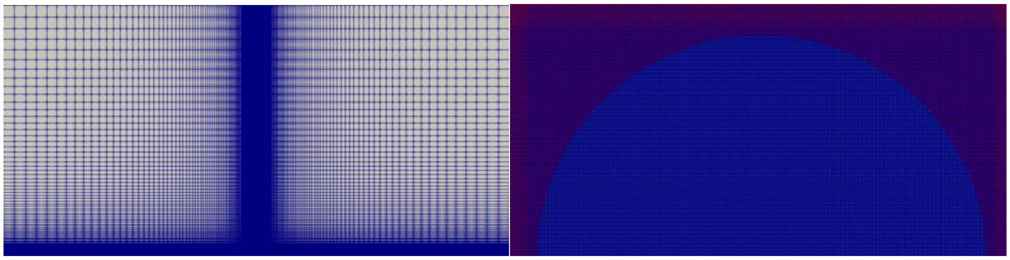


Fig. 25: Left: A zoom out of the mesh used as a base for the cylinder test case, focusing on the outer region. Right: a zoom in on the inner region, coloured with external cells in red and internal cells in blue.

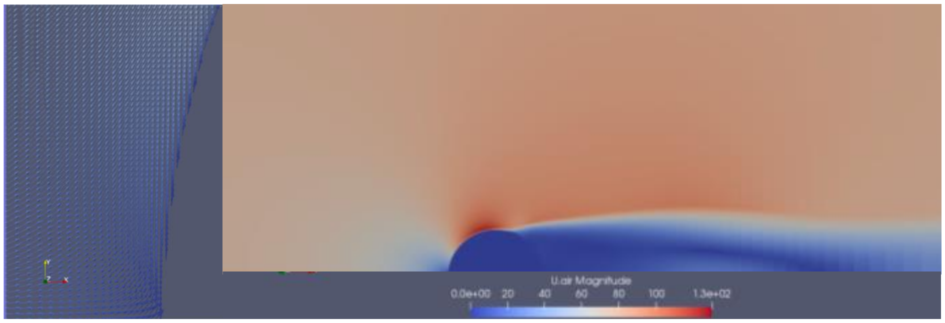


Fig. 26: Vector plot near stagnation point and velocity magnitude field over the cylinder, in m/s.

3.1.2. FENSAP-ICE Setup

For FENSAP-ICE, the previously described structured O-Grid mesh setup is used with an 8.75E-6 m near-wall spacing, a 2.5% constant expansion, and 200 divisions along the cylinder was used following a grid refinement study. The left-half of the outside of the O-Grid is a subsonic inlet condition, and the right-half is a subsonic outlet with a constant pressure of 89867 Pa. The static air temperature is set at 285 K. The final mesh can be seen in Fig. 27.

Unlike in other tests, the minimum wall spacing was brought down until convergence, well into the viscous sublayer. This was done because the goal of the FENSAP-ICE tests was not to give a comparison to results from a similar method to show the validity of the model, but instead to give a converged result as a baseline for comparison and give some idea of whether there were errors in the setup.

A mesh independence study was performed, seen in Table 21. The velocity magnitude field around the cylinder is shown in Fig. 28, which can be compared to the results from the proposed model shown in Fig. 26. At a 40-diameter farfield distance, the total collection was found to be 0.58% different than at a 160-diameter distance. The scaling with respect to the number of divisions was not clear based on the tests run, however even using a first-order Richardson extrapolation, the error for the total collection efficiency for 200 divisions along the cylinder was found to be 0.24%. Based on a Richardson extrapolation from 2.5% and 5% expansions given the approximately second-order scaling seen in the expansion effects, at an expansion rate of 2.5% the error in the total collection efficiency is approximately 0.23%. Based on a first-order Richardson extrapolation, a minimum node spacing of 8.75E-6 m has an error of 0.55% and upwards of this follows an approximately first-order trend with the wall spacing. This wall spacing corresponds to a minimum y^+ of the nearest node to the wall of about 2 which indicates that refining the boundary layer down to this region may be important for this case, something that would cause issues for the proposed model.

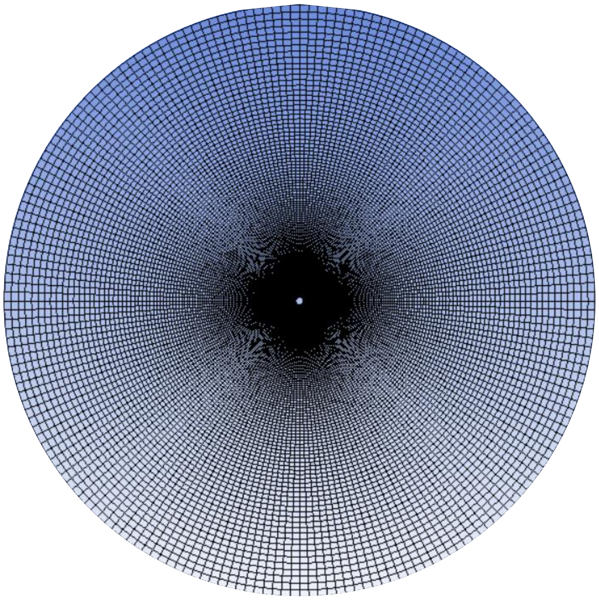


Fig. 27: Final FENSAP-ICE structured O-grid for use in the Cylinder Case.

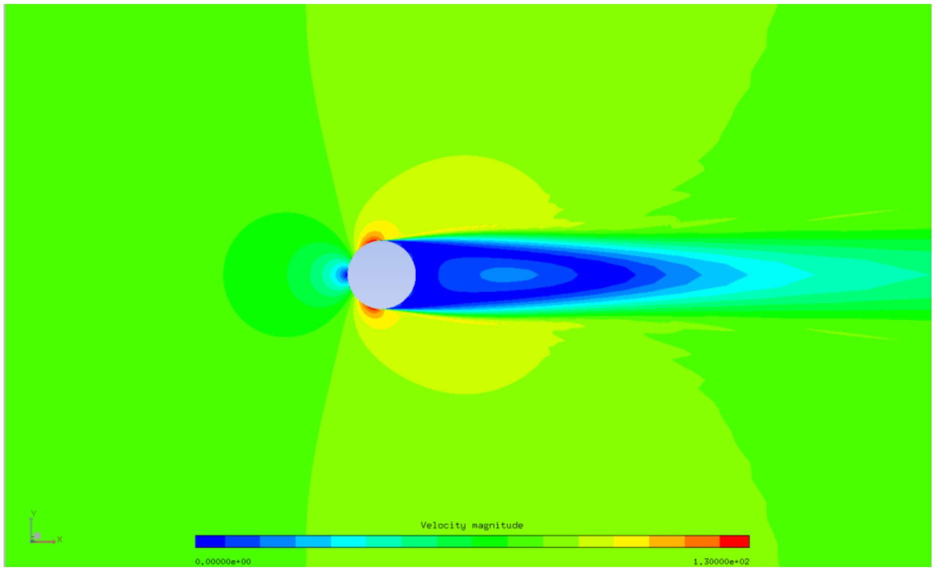


Fig. 28: Velocity magnitude field outputted from FENSAP-ICE on the final mesh, in m/s.

3.1.3. Comparison of Models for Given Inner Region Mesh Sizes

Using the 2 most refined grids, “16” with an inner region spacing of 3.5E-5 m, and “8” with an inner region spacing twice that at 7E-5 m a final solution for the total collection efficiency was calculated using Richardson extrapolation. The order of scaling is shown in Fig. 29 to be in good agreement with the experimental results for a value of 1 for the IBM results. Especially on coarser grids, the cut-cell method appears to display a complex error order scaling.

As seen in Fig. 30, IBM tended to over-estimate collection efficiency on coarser grids, and while the cut-cell methodology may not be smoothly varying in the collection efficiency for coarser grids, the results on coarse grids due appear to closely follow the trend of the finer grid results compared to the immersed boundary result. On the finest grid tested, the cut-cell methodology was found to be about 1.8X more accurate than the immersed boundary method indicating as well that the cut-cell methodology proposed does have advantages in terms of accuracies on the same mesh compared to the immersed boundary method proposed. The cut-cell method collection efficiency was not plotted for the most refined grid ("16") in Fig. 30, because it is almost indistinguishable from the immersed boundary result at that level of refinement.

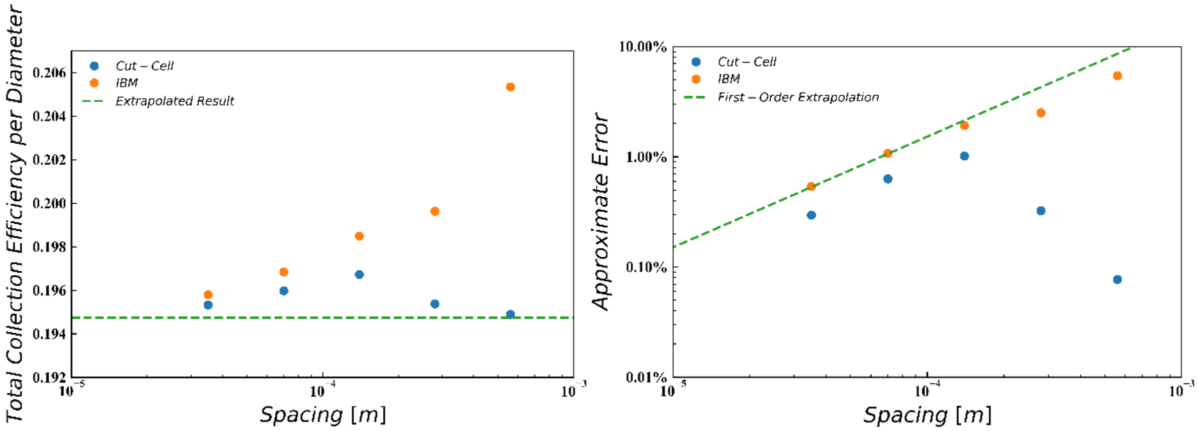


Fig. 29: Total collection efficiency per diameter and resulting error relative to the Richardson extrapolation-derived data from the most refined 2 IBM cases, assuming a first-order scaling of the IBM result, plotted against inner region spacing.

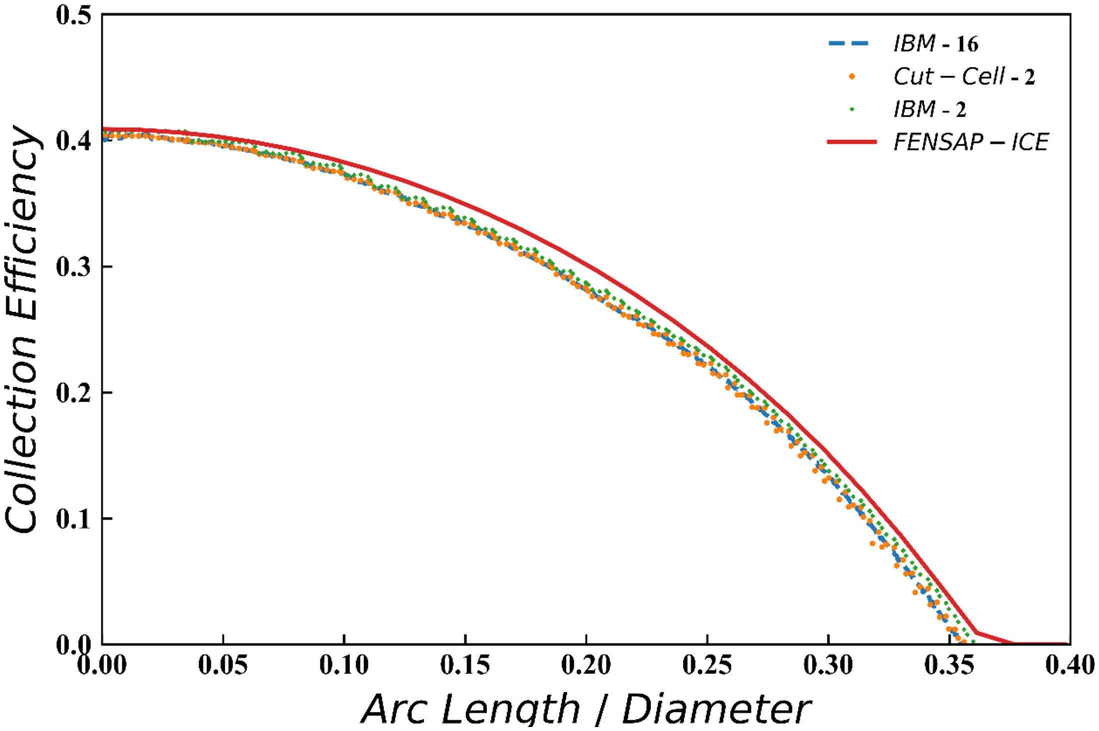


Fig. 30: Collection efficiency vs arc length in the cylinder test case.

The difference in-between the results and the FENSAP-ICE results may be caused by the different turbulence models since large separation is seen at the back and the standard k-ε model with standard wall functions have been known to fail at predicting separation and the FENSAP-ICE results requiring refinement of the mesh down to a y^+ of about 2 to achieve mesh independence with respect to the total collection efficiency here-in [22] [23]. Compressibility effects may also play a small role. Overall, the results were found to have a total collection efficiency which is 5.4% lower than the FENSAP-ICE results on the most refined grid tested, with an inner region spacing of 3.5E-5 m.

As seen in Fig. 31 (a), an error can be seen in the circled region near the boundary when the switch from the front-end boundary condition to the back-end boundary condition occurs, i.e. the point collection efficiency reaches 0 or the end of the collection region. The droplet concentration suddenly rises in this region seems to buckle against the change. This issue gets smaller as the grid is refined as seen in Fig. 31 (b). This is thought to be related to the lack of conservation using IBM, and the sudden switch in the boundary droplet velocity model, which as seen in the circled region in Fig. 32 causes a sharp drop in the velocity of the surrounding cells compared to the cut-cell methodology. This causes the droplets in-front of this to collect, resulting in a higher volume fraction in that region. This likely did not cause any issues in the cases studied here-in as the collection efficiency data is only collected in front of this phenomenon, however, it is thought that this may have a notable impact on tests over complex geometry such as airfoil flaps, where collection efficiency downstream of another body is important to consider.

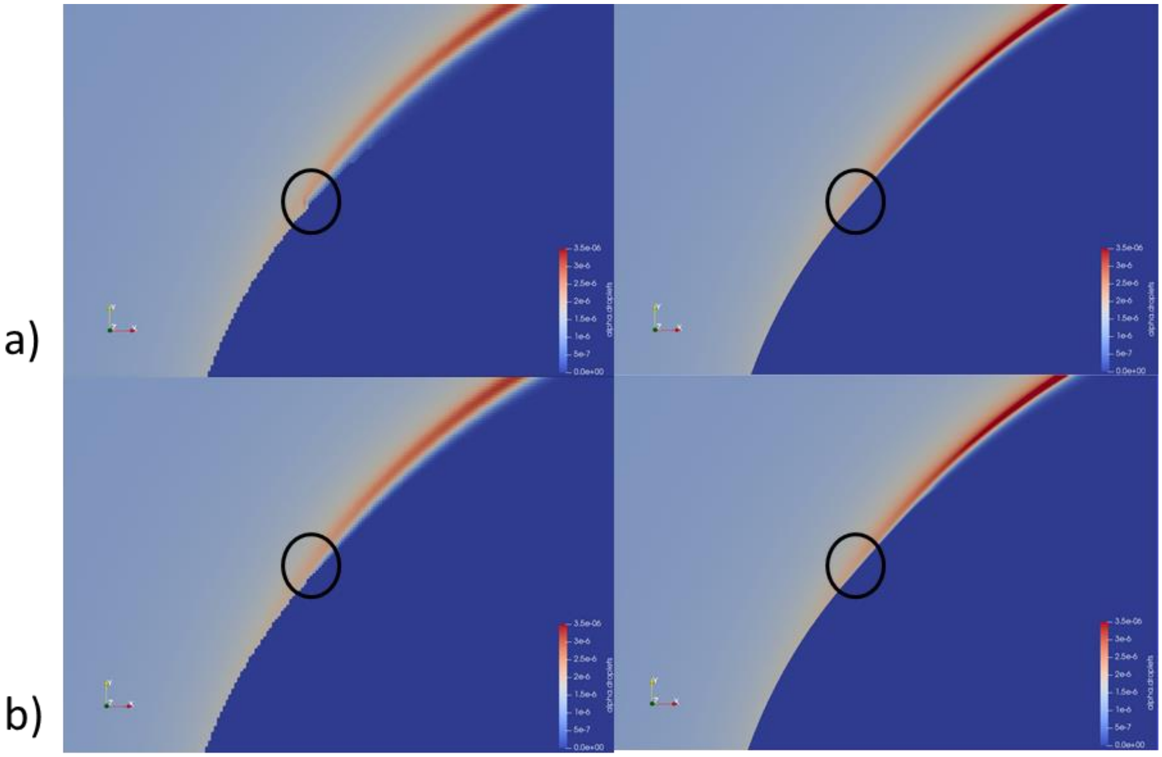


Fig. 31: Close-up of the droplet volume fraction field on grids 2 and 16, while using IBM (a) and then using cut-cell methodology (b). The end of the collection region is highlighted with black circles.

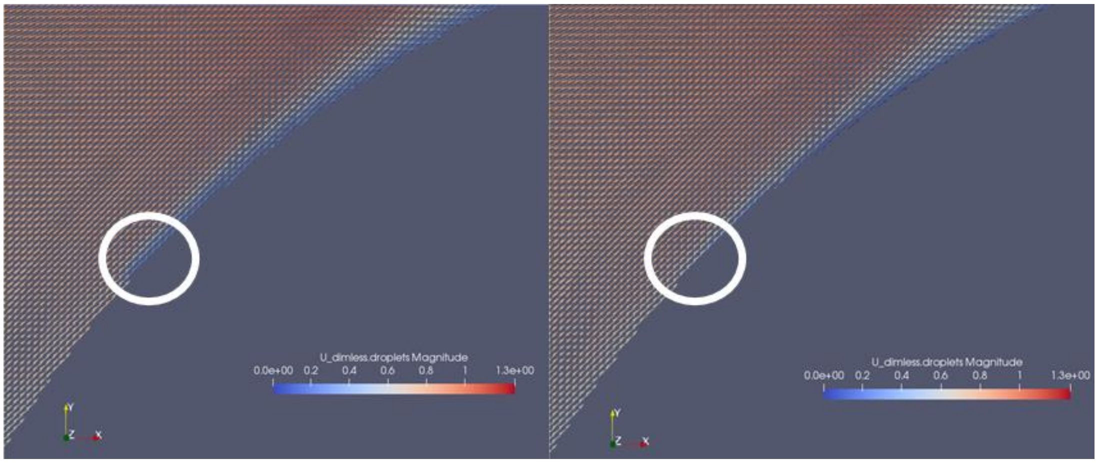


Fig. 32: Close-up of the droplet velocity vectors divided by the freestream velocity on grid 2, showing where the switch from front to back end boundary model occurs while using IBM and cut-cell methodology respectfully. The end of the collection region is highlighted with white circles.

3.2. Ma 0.4 NACA0012 Case

3.2.1. OpenFOAM® Model Case Setup

A NACA0012 airfoil with a 1 m chord was tested with a Mach number of 0.4 and a Reynold's number of 8.8E6 based on the chord length. The freestream velocity was set to 139 m/s, the viscosity ratio to 0.01, the turbulence intensity to 6.2E-5, the droplet volume fraction at 1E-6, and the droplet diameter at 16 μm. The full case setup can be seen in Table 18.

The results were found to be relatively insensitive to freestream distance, as seen in Table 22, and so the freestream was kept 5 chords away from the edges of the airfoil. The airfoil was placed at a 5⁰ angle to the alignment of cells in the inner region, such that the freestream flow was aligned with the cells.

The generalized mesh setup discussed previously was used here. The expansion rate in the outer region was kept at 5%. At the left side of the outer region, the freestream was modeled by a zero-gradient pressure, fixed value for other quantities as inlet condition; on the right side a fixed value pressure, zero-gradient for other quantities outlet condition; and on the top and bottom a freestream condition on was used. The density is set at 1.18 kg/m^3 and the dynamic viscosity was set to 1.86E-5 Pa·s. The test was arranged such that at the inlet boundary the freestream is aligned with the inlet face and the airfoil is at a 5⁰ angle inside of the boundary.

For an average closest cell y^+ of 45 over the airfoil, the inner region was set with a cell side length of 6.0E-4 m. For an average y^+ target of 90 this side length was doubled, etc. The back 1% of the airfoil was cut-off.

Changing the expansion rate from 5% to 2.5% the total collection efficiency was found to increase by 0.37%. Based on a first-order Richardson extrapolation, this means that the error from using an expansion rate of 5% is approximately 0.76%.

The back 1% of the airfoil was cut-off in all calculations to meet the demands of the OpenFOAM®-based cut-cell methodology created.

3.2.2. FENSAP-ICE Setup

The O-Grid FENSAP-ICE mesh setup explained previously was used. The temperature was set to 300 K and the pressure to 1 atm. On the final grid, 400 divisions were placed along the airfoil and an extra 2 on the backend. A 10% expansion rate was used, with a 9.35E-5 m minimum cell size, and a 5-chord farfield distance. The final mesh is seen in Fig. 33.

A mesh independence study was performed, seen in Table 22. The difference in-between an 80-chord farfield distance and a 5-chord farfield distance was found to be 0.45% on the meshes tested. The difference caused by changing the expansion rate from 10% to 2.5% on the meshes tested was found to be 0.009%. Based on a first-order Richardson extrapolation this means that the effects of this variable on the total collection efficiency are about 0.092% for a 10% expansion rate and this is likely an over-estimation given that the data fits better to a second-order curve which would mean even less error at approximately 0.089% error. Increasing the number of divisions along the airfoil from 402 to 804 was found to change the total collection efficiency by 0.39%, which based on a first-order Richardson extrapolation indicates the 402 division test has approximately 0.49% error from this variable. Increasing the minimum node spacing from 9.35E-5 m, corresponding to roughly to an average y^+ for the wall adjacent nodes of 45, to 1.87E-4 m, corresponding to an average y^+ of 90 differed by 1.4% which indicates that the choice of using a minimum cell spacing of 9.35E-5 m over refining down lower and even into the viscous sublayer may significantly impact the results, but this was kept for comparability to the 45 minimum y^+ tests used in the OpenFOAM® model as refining down to the viscous sublayer means that the k-ω SST and k-ε model results may start increasing in their deviation from one another.

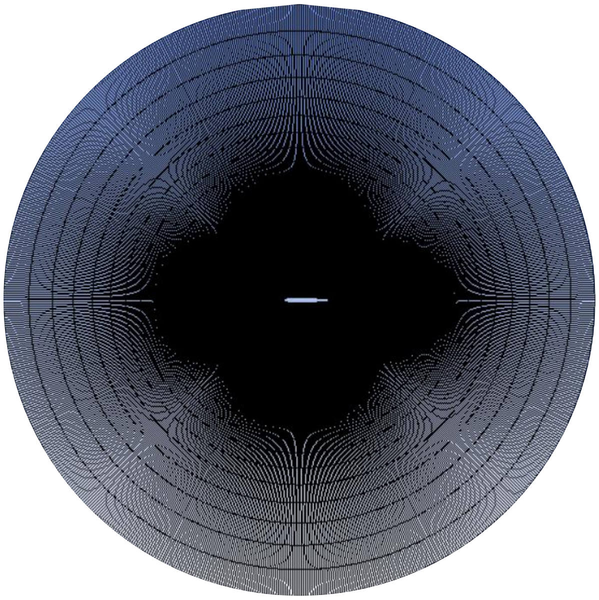


Fig. 33: Final FENSAP-ICE structured O-grid for use in the Ma 0.4 NACA0012 Case.

3.2.3. Comparison of Models for Given Inner Region Mesh Sizes

In Fig. 34 the resulting liquid water content field, a key component of the collection efficiency curves along the airfoil, is shown when using cut-cell methodologies with an inner region mesh spacing of 1.2E-3 m in (a), and with an inner region mesh spacing of 6.0E-4 m in (c). Similarly, (b) and (d) give the liquid water content field when immersed boundary method is used for inner region mesh spacings of 1.2E-3 m and 6.0E-4 m respectfully. The resulting FENSAP-ICE liquid water content field is given in (e) for reference. Terminal points on the collection region of the airfoil are highlighted in black. These portions are where some of the largest differences can be seen between the methodologies and the meshes. Especially on the courser meshes and on the lower section of the airfoil, the droplet concentration field appears notably more diffuse when using IBM than when using a cut-cell approach or the FENSAP-ICE model.

Comparing the results using a 6.0E-4 m inner region spacing to FENSAP-ICE results and results obtained using a Lagrangian droplet phase solver by Wirogo, S. and Srirambhatla, S. (2003) [45], in Fig. 35 the proposed model appears to capture the collection efficiency at the stagnation point and the general trend, adding validity to the model used. The model however notably over-estimates the collection efficiency on both sides. This may be caused by several factors including the differences in the air solver, compressibility, the wall models, the turbulence model, droplet momentum conservation, and while the different boundary models agree to within about 4% for the total collection efficiency, the results may not be fully mesh independent. The cut-cell methodology on the grid with a 6E-4 m was found to be a 4.2% error compared to the FENSAP-ICE result in-terms of the total collection efficiency, and the IBM result was found to be 7.9%. It should be noted that experimental results from Hu, L. et al. (2017) [47] also did not fit perfectly with either set of data, giving a hint as to the scale of model inaccuracies even in FENSAP-ICE.

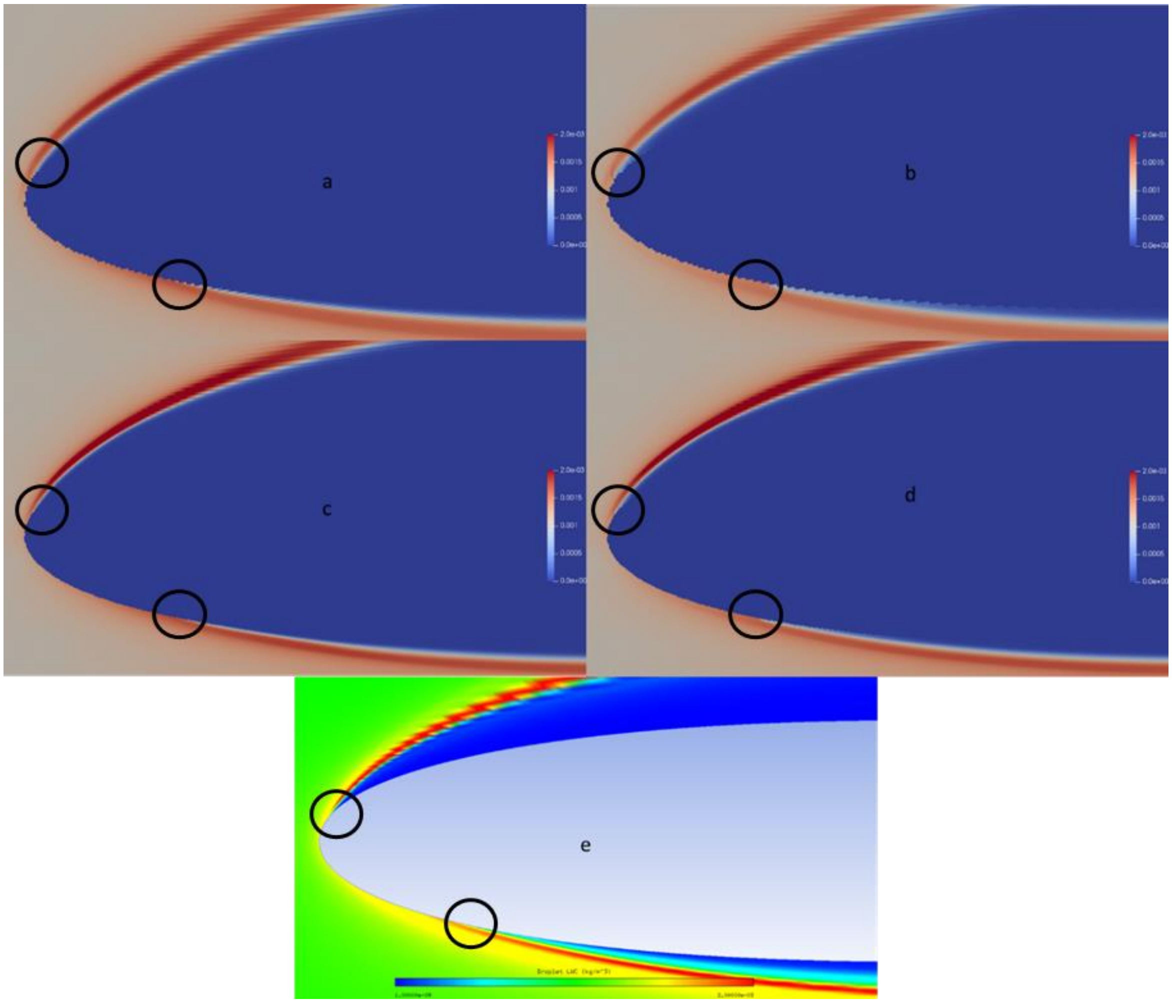


Fig. 34: Liquid water content in kg/m^3 over the front half of the Mach 0.4 airfoil case, with cut-cell methodology (a) and immersed boundary methodology (b) for an inner region spacing of 1.2E-3 m, with cut-cell methodology (c) and immersed boundary methodology (d) for an inner region spacing of 6E-4 m, and using FENSAP-ICE (e). Ends of the collection region on the airfoil are highlighted with black circles.

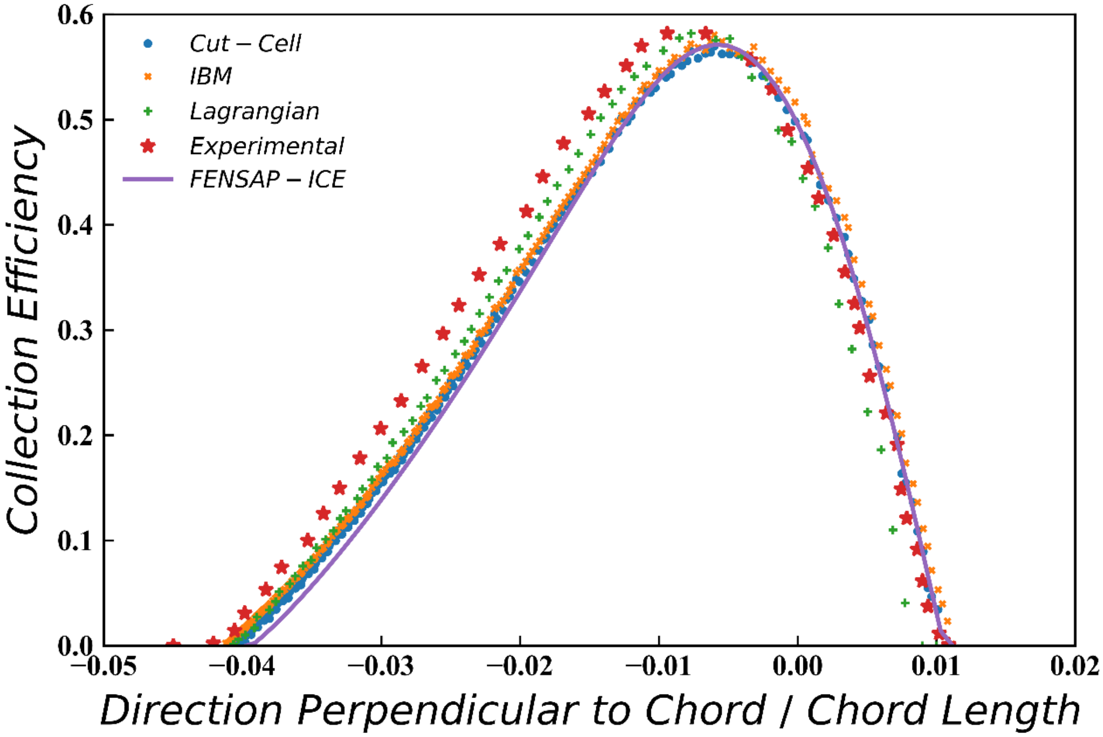


Fig. 35: Comparison of IBM and cut-cell methodology on most refined grid tested with an inner region wall spacing of 6E-4 m to FENSAP-ICE results, as well as results using a Lagrangian droplet phase model from Wirogo, S. and Srirambhatla, S. (2003) [45] and experimental results from Hu, L. et al. (2017) [47] for the Mach 0.4 airfoil case.

3.3. Ma 0.2 NACA0012 Case

3.3.1. OpenFOAM® Model Case Setup

The setup was much the same as the Ma 0.4 NACA0012 Case but the velocity was modified to 69.5 m/s such that the Mach number was changed to 0.2. This caused the Reynolds number based on the chord length to lower to 4.4E6 and the incoming turbulence intensity was changed to 8.7E-5. The full case setup can be seen in Table 17.

The distance from the airfoil edge to the farfield was increased to 10 chords. A 5% expansion rate was kept in the outer region. The airfoil was placed at a 5º angle to the alignment of cells in the inner region, such that the freestream flow was aligned with the cells. The 0.996x 0.144 m inner region was set with a cell side length of 1.14E-3 m, for a y^+ of 45. For an average y^+ target of 90 this side length was doubled, etc. The back 1% of the airfoil was cut-off.

Changing the expansion rate to 2.5% was found to increase the total collection efficiency by 0.29%. Based on a first-order Richardson extrapolation, this means that the error from using an expansion rate of 5% is approximately 0.67%.

3.3.2. FENSAP-ICE Setup

The O-Grid FENSAP-ICE mesh setup explained previously was used. The temperature was set to 300 K and the pressure to 1 atm. On the final grid, 800 divisions were placed along the airfoil and an extra 4 on the backend. A 5% expansion rate was used, with a 1.85E-4 m minimum cell size, and a 10-chord farfield distance. The final mesh is seen in Fig. 36.

A mesh independence study was performed, seen in Table 22. The difference in-between an 80-chord farfield distance and a 10-chord farfield distance was found to be 0.32% on the meshes tested. The difference caused by changing the expansion rate from 10% to 5% on the meshes tested was found to be 0.04%. Based on a first-order Richardson extrapolation this means that the effects of this variable on the total collection efficiency are about 0.03% for a 5% expansion rate. Increasing the number of divisions along the airfoil from 402 to 804 was found to change the total collection efficiency by 0.28%, which based on a first-order Richardson extrapolation indicates the 804 division test has approximately 0.28% error from this variable. Increasing the minimum node spacing from 1.87E-5 m, corresponding to roughly to an average y^+ for the wall adjacent nodes of 45, to 3.74E-4 m, corresponding to an average y^+ of 90 differed by 2.1% which indicates that the choice of using a minimum cell spacing of 1.87E-5 m over refining down lower and even into viscous sublayer may significantly impact the results, but this was kept for comparability to the 45 minimum y^+ tests used in the OpenFOAM® model as refining down to the viscous sublayer means that the k-ω SST and k-ε model results may start increasing in their deviation from one another.

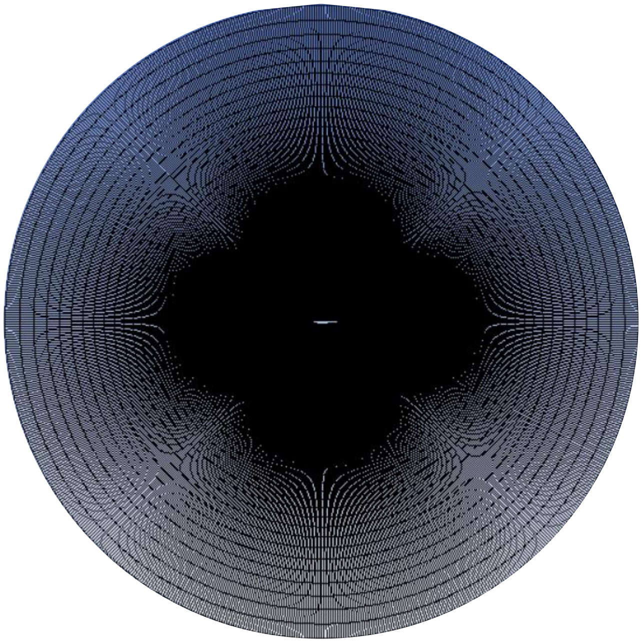


Fig. 36: Final FENSAP-ICE structured O-grid for use in the Ma 0.2 NACA0012 Case.

3.3.3. Comparison of Models for Given Inner Region Mesh Sizes

In Fig. 37 the resulting liquid water content field is shown for an inner region mesh spacing of 1.14E-3 m, when using cut-cell methodologies in (a), and when immersed boundary method (b). The resulting FENSAP-ICE liquid water content field is given in (c) for reference. Terminal points on the collection region of the airfoil are highlighted in black. In front of both terminal points, the immersed boundary method prediction is notably more diffuse than the FENSAP-ICE or cut-cell prediction.

In Fig. 38 it can be seen that compared to the cut-cell methodology, the immersed boundary method overestimates the collection efficiency over the airfoil, an issue which gets worse as the grid is coarsened. The results of on the most refined mesh tested on, with an inner region spacing of 1.14E-3 m using IBM and cut-cell methodologies are about 11% different from each other in-terms of total collection efficiency, and IBM predicts a 8.0% higher result from FENSAP-ICE, while the cut-cell method predicts a 2.7% higher result from FENSAP-ICE.

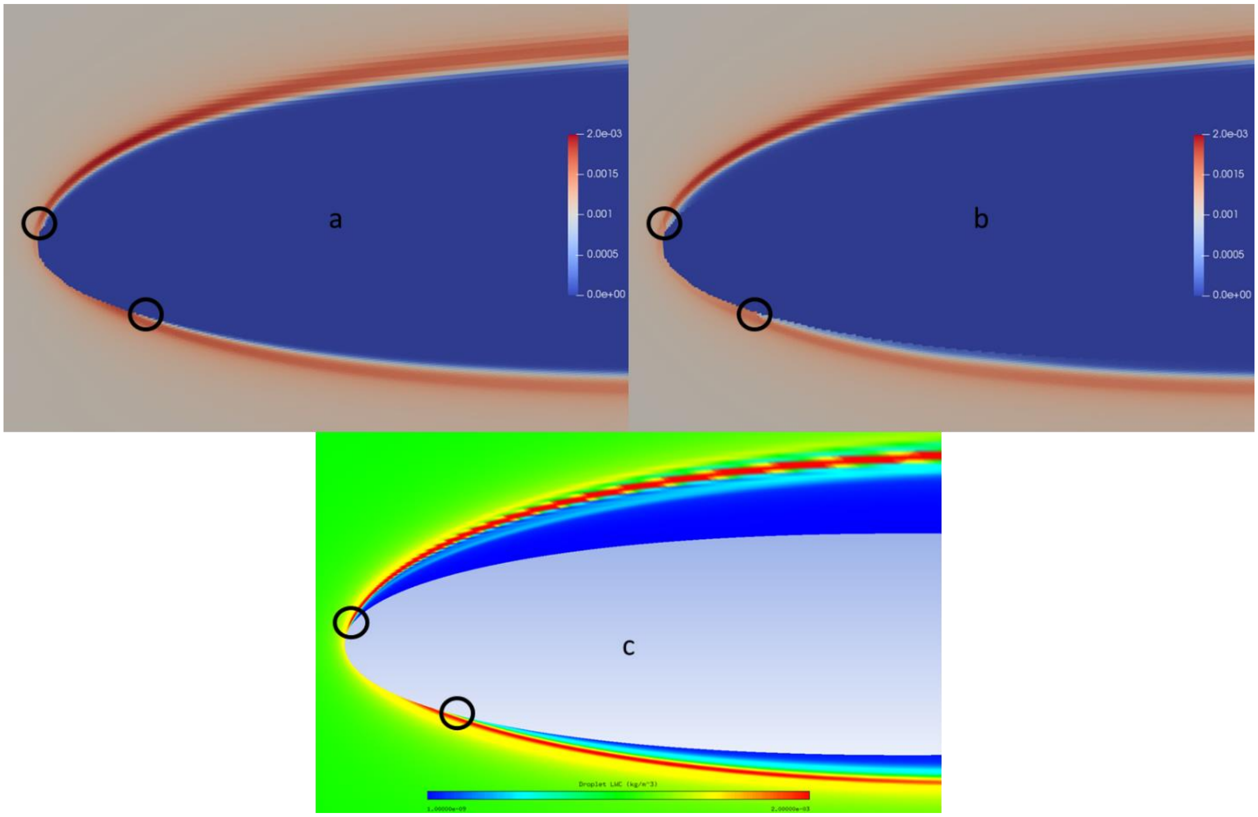


Fig. 37: Liquid water content in kg/m^3 over the front half of the airfoil for the Mach 0.2 airfoil case, with cut-cell methodology (a) immersed boundary methodology (b) and for an inner region spacing of 1.14E-3m, as well as liquid water content from FENSAP-ICE (c) results. Ends of the collection region on the airfoil are highlighted with black circles.

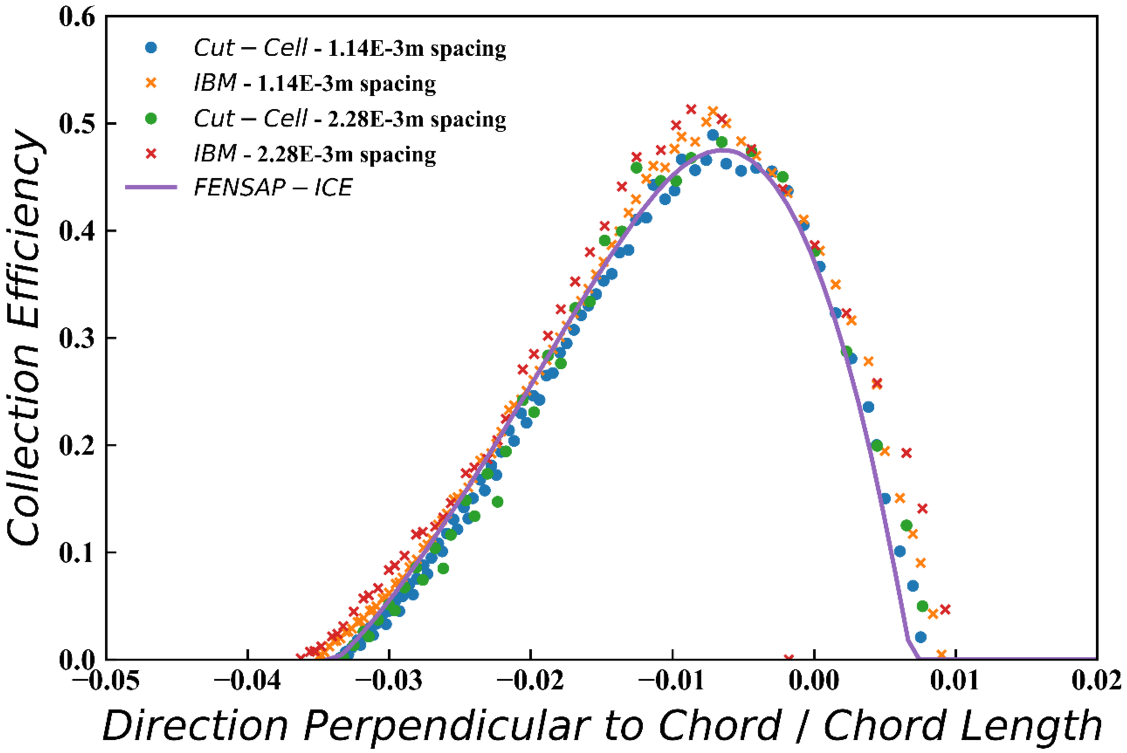


Fig. 38: Comparison of IBM and cut-cell methodology on most refined 2 grids with inner region spacings of 1.14E-3 m and 2.28E-3 m to FENSAP-ICE results for the Mach 0.2 airfoil case.

4. Droplet Size Effects

4.1. Changing Droplet Sizes

As the droplets are made smaller and smaller, they have an average inertia which scales down with the cube of the droplet diameter, and thus the effects of drag tend to become larger. Since, unlike the droplet phase, the air will never go into the walls of the object and will instead always go around it, as the drag effects are increased, the droplet tends to avoid the boundary and the total collection efficiency is expected to decrease with decreasing droplet size.

To study these effects, the Ma 0.2 NACA0012 Case previously discussed when comparing the Cut-Cell and Immersed Boundary Methods was used. All tests were run on the most refined mesh of these tests with an inner region spacing of 1.14E-4 m, a 10-chord farfield distance, a 5% expansion rate, and with the cut-cell methodology for the droplet phase.

In Table 4 a trend can be seen that as the droplets get smaller, especially as they go below 10 μm, FENSAP-ICE and the proposed model disagree, percentage-wise, more and more. The main collection regions outlined with solid circles in Fig. 39 show that the droplet concentration field behave qualitatively similarly for the proposed model and for FENSAP-ICE, even for small droplets. In the dashed region for the 4.96 μm sized droplet case however, Fig. 39 (a) shows a clear difference from Fig. 39 (d). The proposed model being predicts a small number of droplets hitting the surface over nearly the entire bottom of the object while the FENSAP-ICE case 0 collection outside of the region directly around the stagnation point. This discrepancy is thought to be caused by discretization errors on the droplet side because droplets hug so close to the wall in this case. It was found to nearly double when the upwind model is used for the droplet momentum convection instead of *cellLimited linearUpwindV*, but even using the less diffusive *cellLimited linearUpwindV* model, this is responsible for an estimated 58% of the total collection efficiency being from this bottom portion. This portion was filtered out from the results shown in Fig. 40 to make the results all comparable on the chart as is as some of the points are on the back end and some on the front end with this chart being unable to distinguish between the two.

Even neglecting this, FENSAP-ICE and the OpenFOAM®-based model do not agree quantatively for smaller droplets, showing in Fig. 40 that the maximum collection efficiency is over twice as high when calculated using the OpenFOAM®-based model then the FENSAP-ICE model. This is thought to be due to a combination of discretization errors and differences in the air models playing larger roles in these cases relative to the larger droplet cases. These issues imply that this model and/or the discretization may require further refinement and research before being applied to small droplets, for example going below 10 μm in this test case may be unadvisable.

Table 4: Comparison of OpenFOAM® and FENSAP-ICE models prediction for total collection efficiency at various droplet sizes.

Droplet Size (μm)	**FENSAP-ICE Total Collection Efficiency per Chord, $C_{T,FENSAP}$**	**OpenFOAM®-Based Total Collection Efficiency per Chord, $C_{T,OpenFOAM}$**	$C_{T,OpenFOAM} - C_{T,FENSAP}$	$\left\lvert \frac{C_{T,OpenFOAM}}{C_{T,FENSAP}} - 1 \right\rvert$
4.96	0.0001856	0.0007177	5.32E-04	300 %
8.32	0.002500	0.002755	2.56E-04	10 %
11.36	0.007129	0.007216	8.66E-05	1%
16	0.01501	0.01457	-4.33E-04	3%
21.92	0.02489	0.02466	-2.28E-04	0.9%
27.84	0.03409	0.03393	-1.64E-04	0.5%
35.52	0.04484	0.04476	-8.19E-05	0.2%

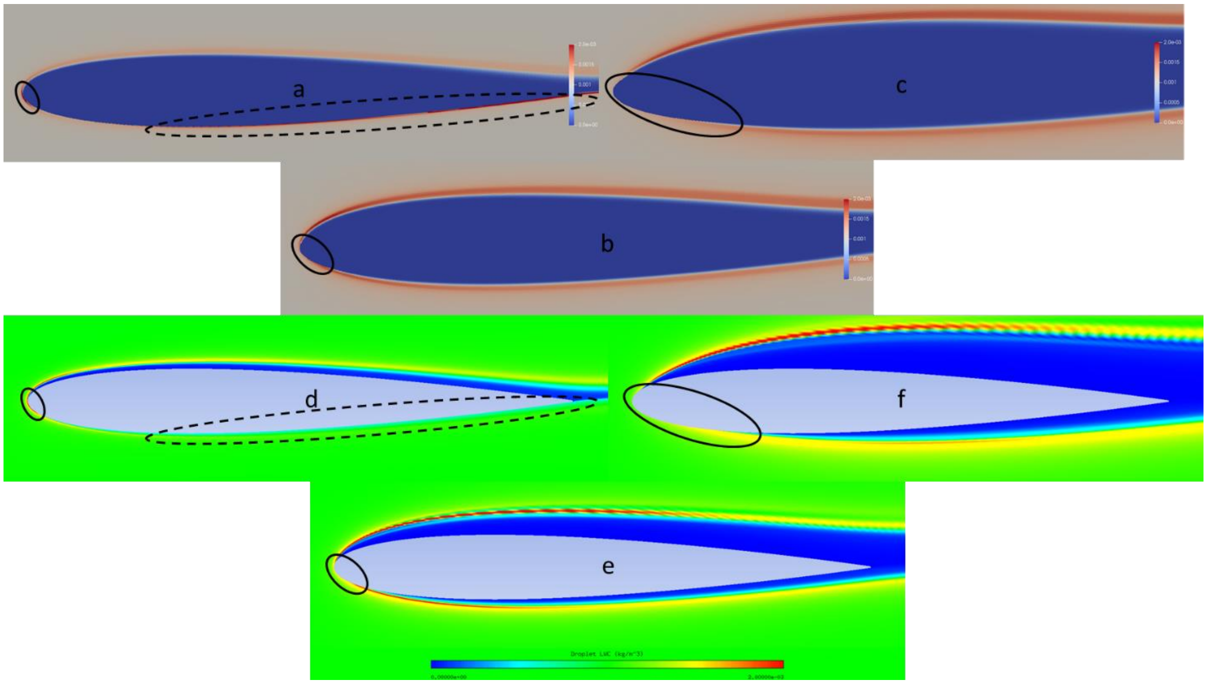


Fig. 39: Liquid water content in kg/m^3 for 4.96 µm droplets (a/d), 16 µm droplets (b/e), and 35.52 µm droplets (c/f) using the developed OpenFOAM®-based model (a/b/c) and FENSAP-ICE (d/e/f). Solid black circles are used to highlight the collection regions and dashed black circles are used in (a/d) to highlight the back-end bottom portion for which the proposed model has notable disagreements with the FENSAP-ICE results.

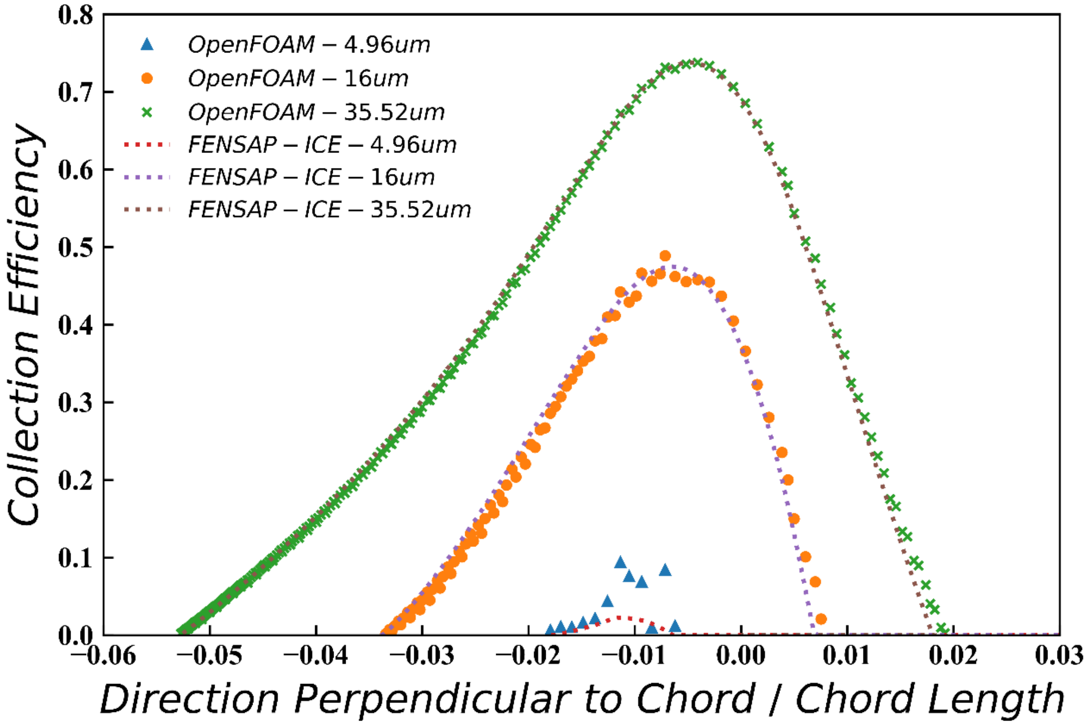


Fig. 40: Comparison of collection using 4.96 µm droplets, 16 µm droplets, and 35.52 µm droplets using the developed OpenFOAM®-based model and FENSAP-ICE.

4.2. Capturing Effects of Multidroplet Distribution

Up until now, it was assumed that the droplet size distribution (DSD) is very narrow and thus a single droplet size at the mean-value diameter (MVD) is a good representation of the droplets. For realistic conditions in both nature and under experimental conditions, this may contribute to a large amount of model error compared to other neglected aspects of the droplet model for large droplets, being critical to supercooled large droplet simulations of modern interest, as identified by Hospers, J. (2013) [48].

In the Eulerian form of the constitutive droplet equations used here-in and in for example ANSYS' FENSAP-ICE, a single droplet size is assumed leaving no room for a DSD [16]. Since droplets of different sizes are assumed in the model to not interact with each other in any way, one workaround for this issue is to run the droplet loop multiple times for a set of droplet sizes representing the distribution and then put the results together at the end, as presented by Papadakis, M. et al. (2007) [49], LEWICE 2.2 [11], and others via a linear combination of the collection efficiency as follows:

$$C_E(s) = \sum_{droplet\ size\ i} w_i C_{E,i}(s) \tag{92}$$

Where $C_E\,(s)$ and $C_{E,i}\,(s)$ are the combined and individual collection efficiencies at a point s on the surface, and w_i is the fraction, by weight, of the droplet size i in the collection of droplets in the freestream.

One common distribution used in icing analysis known as the Langmuir-D Distribution, described in "*The Collected Works of Irving Langmuir*" (1960) [50]. This distribution is noted for being a favourite in icing analysis by Jeck, R. (2006) with the U.S. Federal Aviation Administration who suggested that this distribution is used so commonly as it is in at least one standard reference book, that book being "Engineering Summary of Airframe Icing Technical Data" by Bowden, D.T. et al (1963) from within the same organization [51] [52]. This distribution is presented in Table 5 for an MVD of 16 μm and a 7 bin predefined setup for this droplet size and distribution from FENSAP-ICE, alongside previous results for each droplet size for the Ma 0.2 NACA0012 airfoil case seen in Table 17 on a mesh with an inner region spacing of 1.14E-4 m, a 10-chord farfield distance, a 5% expansion rate, and with the cut-cell methodology for the droplet phase.

As can be seen in Table 5 and graphically in Fig. 41, the strong relative deviations previously noted in-between the two models' collection efficiency predictions for small droplets are not expected to dominate the overall collection curve as they tend to have low weightings and small collection efficiencies in the first place. Larger droplets that are more accurately predicted tend to have a larger effect on the collection, and due to this, the overall difference in total collection efficiency was found to be even smaller than in the MVD case at a 0.8% difference instead of a 2.9% difference. Overall, due largely to the influence of these larger droplets, the collection efficiency curve was found to push farther down the airfoil than the MVD case, as seen in Fig. 42.

Table 5: Langmuir-D distribution for an MVD of 16 µm using 7 bins, and a comparison of total collection efficiencies multiplied by mass fractions.

Droplet Size (µm)	Mass Fraction	FENSAP-ICE Total Collection Efficiency per Chord * Mass Fraction	OpenFOAM®-Based Total Collection Efficiency per Chord * Mass Fraction
4.96	0.05	9.28E-6	3.59E-5
8.32	0.1	2.50E-4	2.76E-4
11.36	0.2	1.43E-3	1.44E-3
16	0.3	4.50E-3	4.37E-3
21.92	0.2	4.98E-3	4.93E-3
27.84	0.1	3.41E-3	3.39E-3
35.52	0.05	2.24E-3	2.24E-3

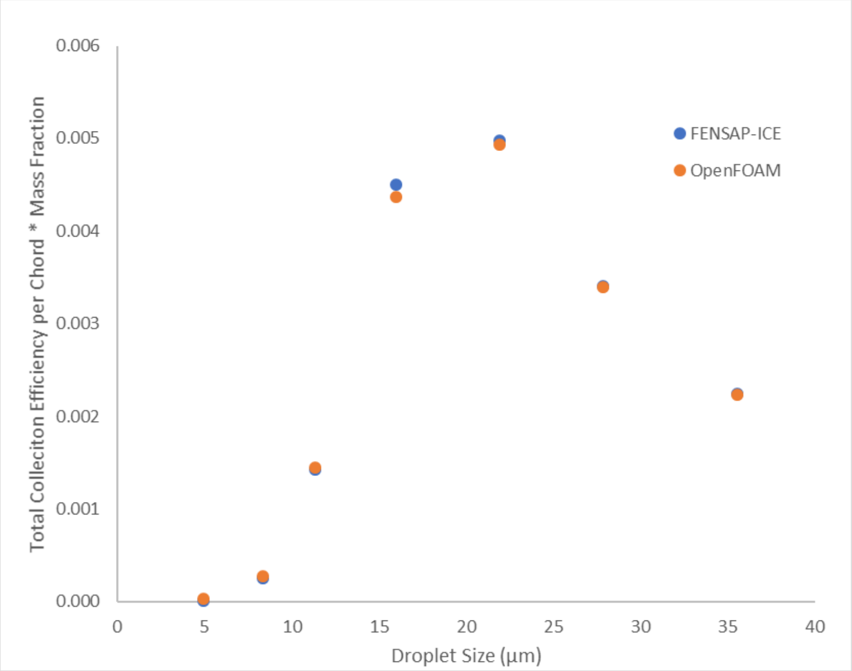


Fig. 41: Comparison of total collection efficiencies multiplied by mass fractions using a 7-bin Langmuir-D distribution setup.

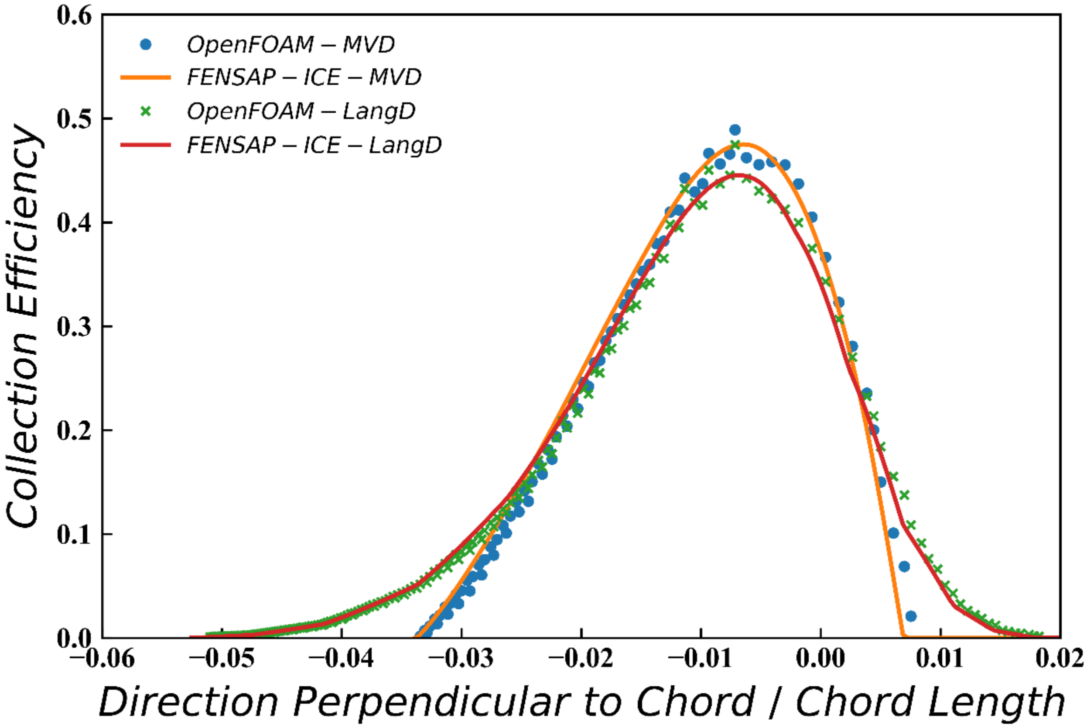


Fig. 42: Comparison of initial collection efficiency for a Langmuir-D distribution and an MVD-only case.

While water droplets would not be expected freeze under the conditions of this case, the temperature and pressure serve to describe the thermodynamic properties of the system, but not the freezing fraction in a pure rime ice calculation as setup, and thus a simulation can still be performed. A 10 s test was run to see the impact of the droplet diameter on a single-step ice accretion test. Based on a comparison of Fig. 43 and Fig. 42, the resulting ice-shape does seem to deviate slightly in different ways from the original collection efficiency curve due to differences in the assumptions of the boundary growth model and methodologies. Multi-step ice accretion which tackles the effects of these assumptions, as well as approximating changes in the droplet collection efficiency curve over time as ice is accreted will be analyzed in later sections.

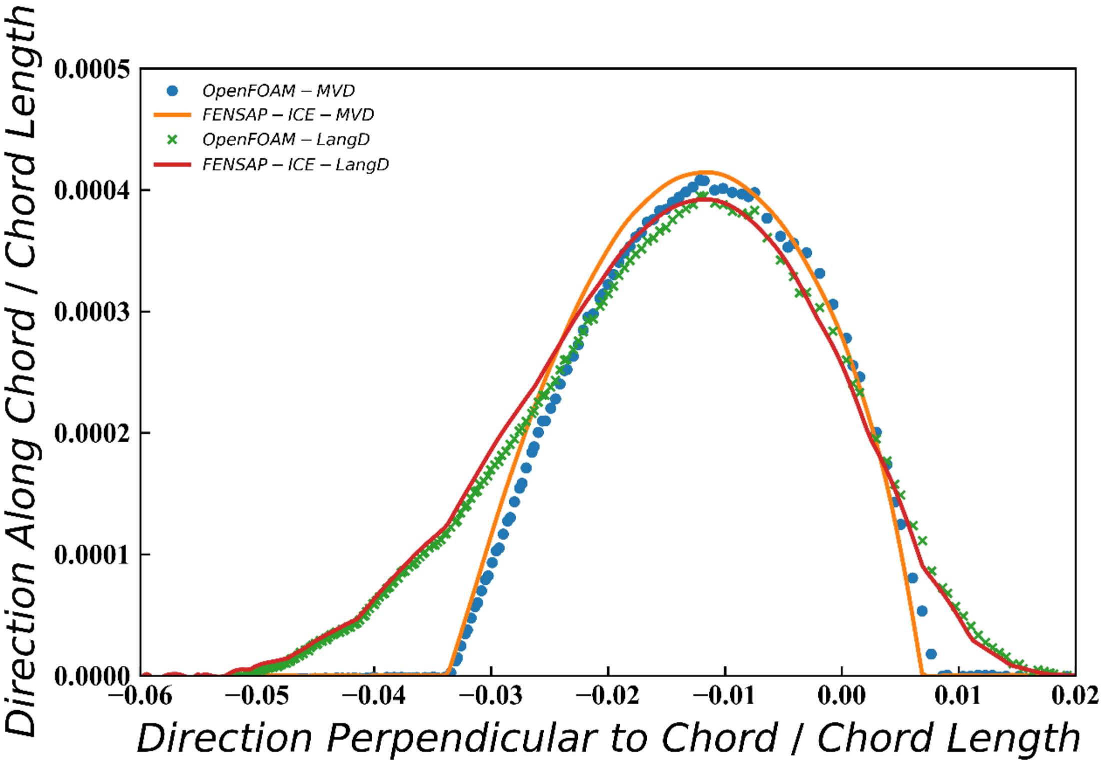


Fig. 43: Comparison of net ice accretion resulting from a single-step ice accretion for a Langmuir-D distribution and an MVD-only case.

5. Multi-Step Ice Accretion Tests

It has been seen that single-step ice accretion, that being basing ice growth off of the initial collection efficiency curve, creates issues due to different interpretations of how the boundary should grow, different methodologies for boundary growth, and changing flow fields and collection curves as the boundary grows. Multi-step ice accretion is one solution to these issues.

In this methodology, the boundary evolution is broken up evenly into *n* steps. After each of these steps, the air and then droplet phase steady-state calculations are redone with the accreted boundary, allowing for a recalculation of the collection efficiency curve. The boundary is then evolved in the new normal direction based on the new collection efficiency curve.

This methodology introduces a new variable to tune alongside those in single-step/initial collection tests: how many steps are required. As the number of steps goes up, differences caused by assumptions about the boundary growth tend to go down, and the solutions tend to approach a single solution. This will take more time to complete the simulation, and as will be discussed further, issues with numerical roughness along the boundary can also be hindering for many solvers at large numbers of time-steps for certain cases.

5.1. NACA23012 Case

5.1.1. OpenFOAM® Model Case Setup

This test is based on experimental tests from Broeren, A.P. et al. (2011) over a 1.83 m chord NACA23012 airfoil at a 2° angle of attack, listed by these researchers as IRT run EG1162 [53]. The back 1% of the airfoil was cut-off in all calculations to meet the demands of the OpenFOAM®-based cut-cell methodology created. The cut-cell methodology was used in all droplet calculations as it was found to be more accurate than the IBM methodology.

At a pressure of 1 bar and a temperature of 248 K, an air density of 1.41 kg/m^3 and a viscosity of 1.59E-6 Pa·s was used. The freestream velocity of 77.2 m/s means that the Reynolds number over the chord is 1.25E7 and the Mach number is 0.24.

The general mesh setup explained previously was used. The expansion rate in the outer region was kept at 5%, the farfield was placed 10 chords away, and the inner region was meshed with spacings of 8E-4 m for an average y^+ of approximately 45. The airfoil was placed at a 2° angle inside the inner region. One reason this test was performed was to see the ease of meshing. While a new airfoil may be used here, the NACA23012 instead of NACA0012, the same basic mesh structure was used, and the same set of parameters were modified during mesh creation, making the mesh creation process very fast.

Little impact in the collection efficiency curve resulted from decreasing the outer region expansion rate from 5% to 2.5%, as seen in Table 6. This corresponds to going from 144 cells from the centre of farfield to the nearest point in the region separating inner and outer regions to 258 cells. The total collection efficiency was found to increase by 0.02%. This implies that in this case, the discretization error in the expansion direction of the outer region on this mesh on this mesh is likely small compared to other errors. Based on a first-order Richardson extrapolation, the expansion effects are responsible for a total of approximately 0.03% error on the 5% mesh.

Table 6: Initial total collection efficiency per chord for NACA23012 for different expansion rates the OpenFOAM®-based model.

Expansion Rates	Number of Cells in Direction from Inner to Outer Boundary	Initial Total Collection Efficiency per Chord
5%	144	0.021222
2.5%	258	0.021216

The impact of the boundary evolution time-step was also found to be small, as seen in Table 7. For the 10 min test, a non-conservative ending time-step model was used so that the effects of the time-step could be compared based on the final ice mass. Changing the time-step from 1 s to 0.1 s for a min single-step test was found to change the final ice mass by 0.10%, which means that based on a first-order Richardson extrapolation that the temporal discretization error impacts the total ice mass by approximately 0.11% for a 1 s time-step. It is thought that using the conservative time-step model will further minimize the impact of this as the final ice mass will be independent of the time-step and only the resultant ice shape will be impacted.

Table 7: Net amount of accreted ice per span for NACA23012 for different time-steps using the OpenFOAM®-based model with the non-conservative ending time-step model for a single-step.

Time-step (s)	Net Ice Accreted per Span (kg/m)
1	0.4508
0.1	0.4261

5.1.2. FENSAP-ICE Setup

The O-Grid FENSAP-ICE mesh setup explained previously was used. The temperature was set to 248 K and the pressure to 1 bar. On the final grid, 800 divisions were placed along the airfoil and an extra 4 on the backend. A 2.5% expansion rate was used, with a 1.8E-4 m minimum cell size, and a 10-chord farfield distance. The final mesh is seen in Fig. 44.

A mesh independence study was performed, seen in Table 24. The difference in-between an 80-chord farfield distance and a 10-chord farfield distance was found to be 0.26% on the meshes tested. The difference caused by changing the expansion rate from 5% to 2.5% on the meshes tested was found to be 0.07%, and based on a first-order Richardson extrapolation this means that for a 2.5% expansion rate, there is approximately a 0.05% remaining error due to the expansion rate. Increasing the minimum node spacing from 1.8E-4 m, corresponding to roughly to an average y^+ for the wall adjacent nodes of 45, to 2.9E-4 m, corresponding to an average y^+ of 70 resulted in a change in the total collection efficiency of 0.54%. 1.8E-4 m was kept for comparability to the 45 minimum y^+ tests used in the OpenFOAM® model as refining down to the viscous sublayer means that the k-ω SST and k-ε model results may start increasing in their deviation from one another. Increasing the number of divisions along the airfoil from 804 to 1608 was found to change the total collection efficiency by 0.35%. Based on a first-order Richardson extrapolation, this means that the 804 division test has approximately 0.69% error from the discretization along the airfoil. After a 4-step ice accretion for 10 min, the difference in total accreted ice mass was found to be 0.69% in-between a mesh with 804 divisions and 1608 divisions, which implies a 1.4% discretization error in this based on a first-order Richardson extrapolation.

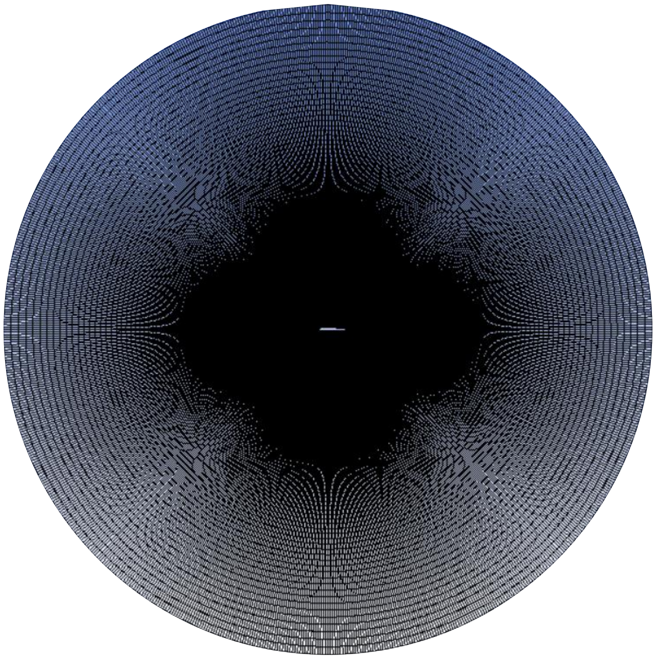


Fig. 44: Final FENSAP-ICE structured O-grid for use in the NACA23012 Case.

5.1.3. Inner Region Mesh Size

The effects of the inner region mesh spacing on the initial collection efficiency for the OpenFOAM®-based case were tested, as seen in Table 8. The mesh spacings of 3.2E-3 m, 1.6E-3 m, 8E-4 m, and 4E-4 m that were tested, correspond to approximate average y^+'s of 180, 90, 45, and 22.5. The outer region expansion rate was kept at 5% and the outer region distance was kept at 10 chords. The resulting initial collection efficiency curves are shown in Fig. 45. The resulting initial total collection efficiencies are given in On the y^+ of ~45 grids, the total collection efficiency is 0.21% different than the FENSAP-ICE result.

On the coarsest grid, the OpenFOAM®-based prediction of the total collection efficiency is 4.5% different than the FENSAP-ICE result, and on the finest grid, the total collection efficiency is 1.4% different than the FENSAP-ICE result. The finest grid test is associated with an average y^+ of ~23 over the airfoil. This means that the cells nearest to the wall on average are dipping into the buffer region of the wall model, which may indicate some additional issues are present in the boundary layer for these tests even though the discretization error is smaller. On the y^+ of ~45 grids, the total collection efficiency is 0.21% different than the FENSAP-ICE result.

Table 8: NACA23012 Case initial total collection efficiency vs inner region spacing.

Inner Region Spacing (m)	Initial Total Collection Efficiency per Chord
3.2E-3	0.02023
1.6E-3	0.02074
8E-4	0.02122
4E-4	0.02146
FENSAP-ICE	0.02118

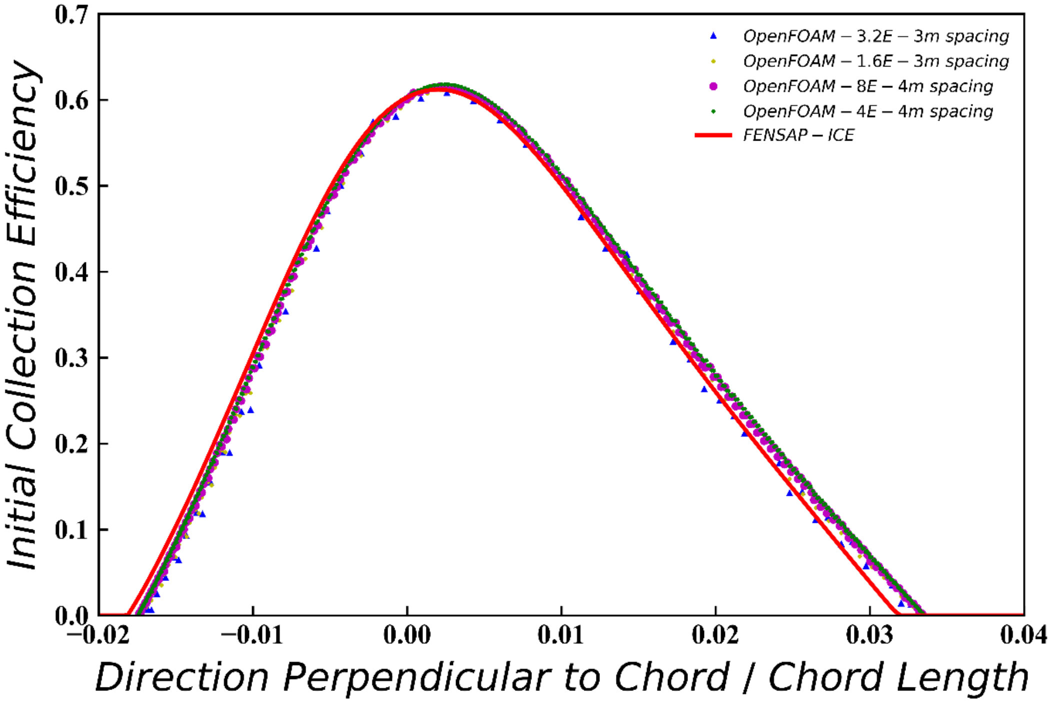


Fig. 45: NACA23012 Case initial collection efficiency curve for various inner region spacings.

5.1.4. Ending Time-step Model

To analyze the effectiveness of the “conservative” ending time-step model proposed in which the 10 min ice accretion ends when the total amount of ice accreted equals the amount determined from the initial total collection and the time for accretion, it is compared for a single-step accretion to the “non-conservative” model in which the 10-minute ice accretion ends after 10 minutes of growth in the normal direction at the initially determined rates.

The initial total collection efficiency using the OpenFOAM®-based model was found to be 0.21% higher than the FENSAP-ICE result when using an 8E-4 m inner region spacing. The conservative model predicts a 0.21% higher total ice mass per span from the FENSAP-ICE result, corresponding to the deviation of the total collection efficiency. The non-conservative model predicts a 17% higher ice mass than FENSAP-ICE. The resulting ice shape using the non-conservative model predicts a notably higher ice accretion near the stagnation point compared to FENSAP-ICE and the conservative model, as seen in Fig. 46. As explained previously, the “conservative” model does not guarantee perfect conservation as it is set up, as a final reinitialization is performed after the evolution, and for the final time-step an assumption is made about the time-step which is to be used which is expected to overestimate the required time-step for a convex surface, but in this case and in all cases tested, the lack of conservation is on the order of ~0.05% or less, which is an improvement over the 10-50% seen in this case and others when using the non-conservative model. The conservation of the conservative model also relies on accurate estimates of the total collection efficiency and the total ice mass, which require accurate surface integral methods and boundary reconstruction.

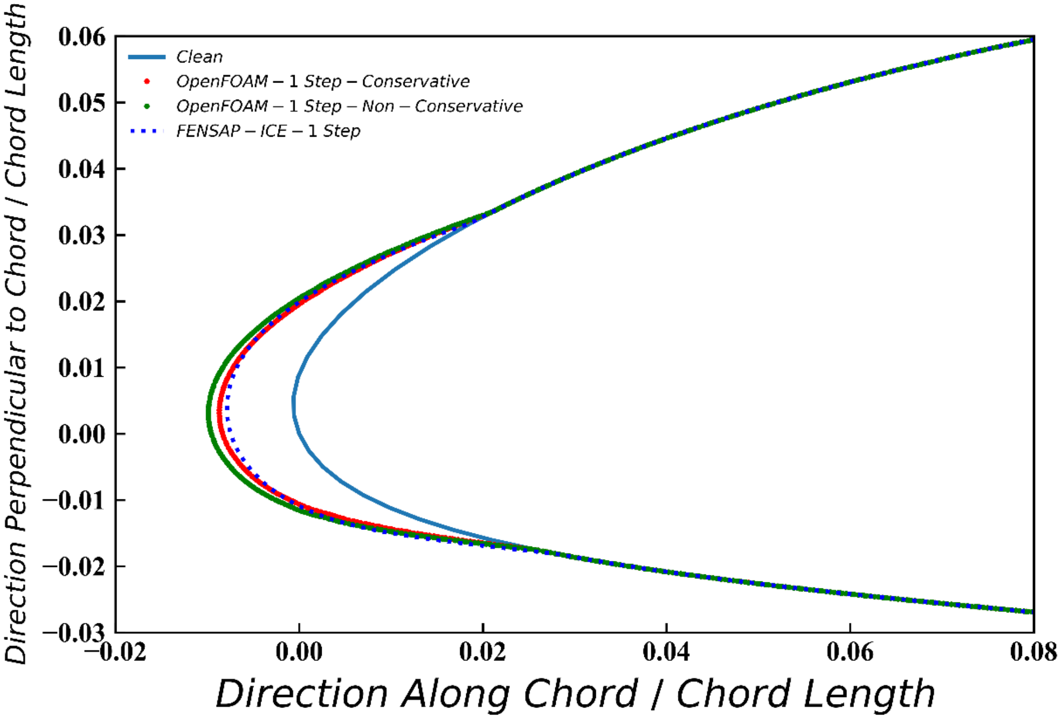


Fig. 46: NACA23012 Case accreted ice shapes for a single-step ice accretion with conservative and non-conservative ending time-step model and comparison to FENSAP-ICE.

5.1.5. Number of Steps

In multi-step ice accretion, the choice of the number of steps is one key parameter towards determining the resulting ice shape. In Fig. 47, 1 and 4 step tests are compared for both FENSAP-ICE and the OpenFOAM®-based model. The difference in-between the total ice mass from the OpenFOAM®-based model and FENSAP-ICE is 0.21% for a 1-step test, and 0.11% for a 4-step test. The 4-step test accretes 4.3% less ice mass than the 1 step test according to the OpenFOAM®-based model and 4.0% less according to FENSAP-ICE, as seen in Table 9.

Table 9: Net amount of accreted ice per span for NACA23012 Case initial total collection efficiency vs inner region spacing.

Number of Steps	Net Ice Accreted per Span – OpenFOAM®-Based Model (kg/m)	Net Ice Accreted per Span – FENSAP-ICE (kg/m)
1	0.9894	0.9873
4	0.9470	0.9480

For both 1-step and 4-step tests the resulting ice shapes from the 2 simulation environments are closer together than to experimental results by Broeren, A.P. et al. (2011) [53]. It was noted by Wright, W.B. and Rutowski, A. in a 1999 validation report for NASA's LEWICE 2.0 software for ice accretion that a number of experimental errors can occur including repeatability and tracing errors, with the former referring to experimental setup issues, and the latter referring to human error in drawing the resulting ice shape [54]. The DSD is assumed to be represented by the single

MSD may also play a large role in this error from the modeling side. As was discussed earlier, if a wide DSD resulted from the sprayer used experimentally, the resulting ice shape may change.

While it may be difficult to see in the ice shape in Fig. 47 and Fig. 48, analyzing the net ice accretion in Fig. 49, it can be seen that as the number of steps goes from 1 to 4, the results from the proposed model and from FENSAP-ICE seem to approach each other more closely and the experimental results near the front. Some additional roughness on the iced surface can be seen for the 4 step tests using the proposed model in the OpenFOAM®-based model compared to the FENSAP-ICE results, an issue which will be addressed later in the Shin and Bond tests.

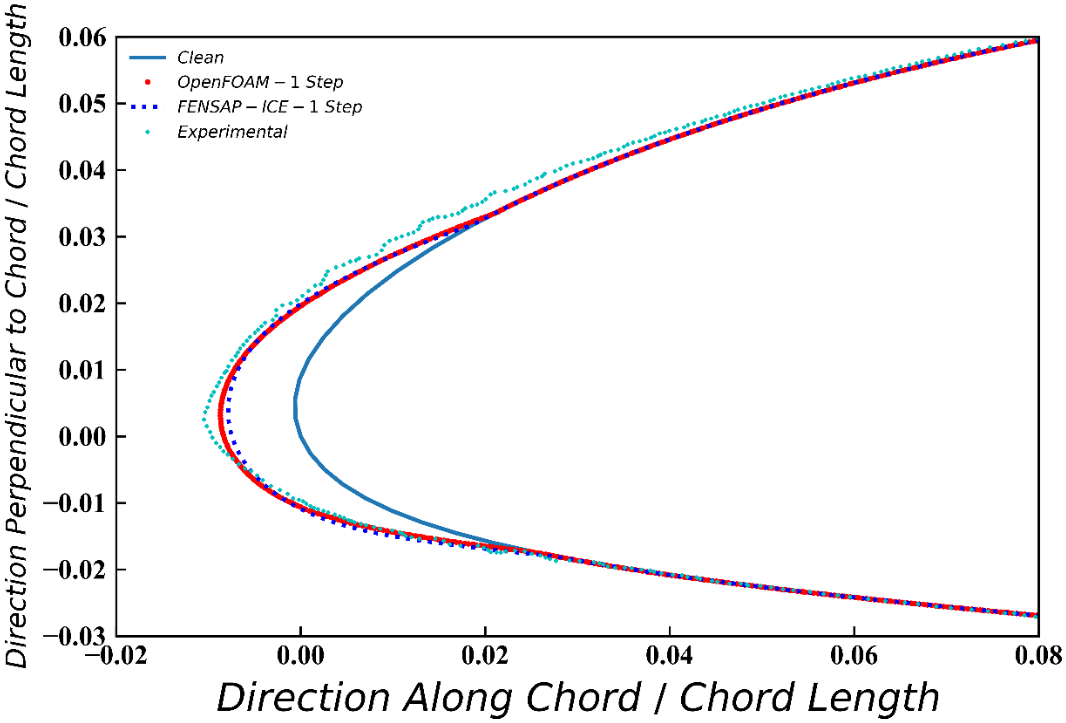


Fig. 47: NACA23012 Case accreted ice shapes for 1 step. Experimental results from Broeren, A.P. et al. (2011) are included [53].

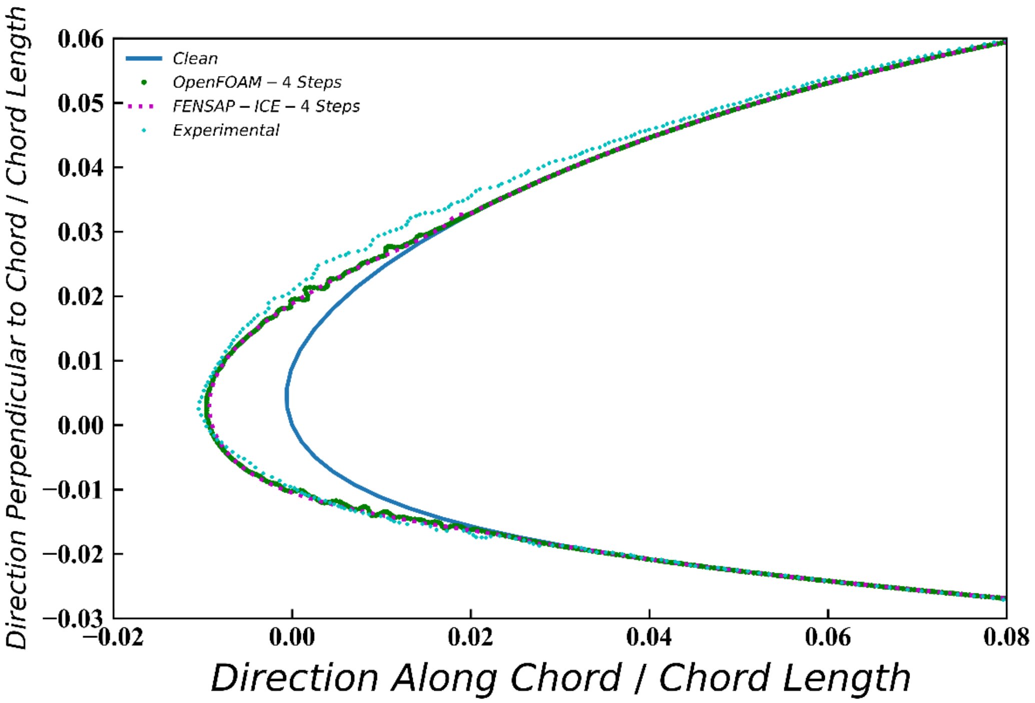


Fig. 48: NACA23012 Case accreted ice shapes for 4 steps. Experimental results from Broeren, A.P. et al. (2011) are included [53].

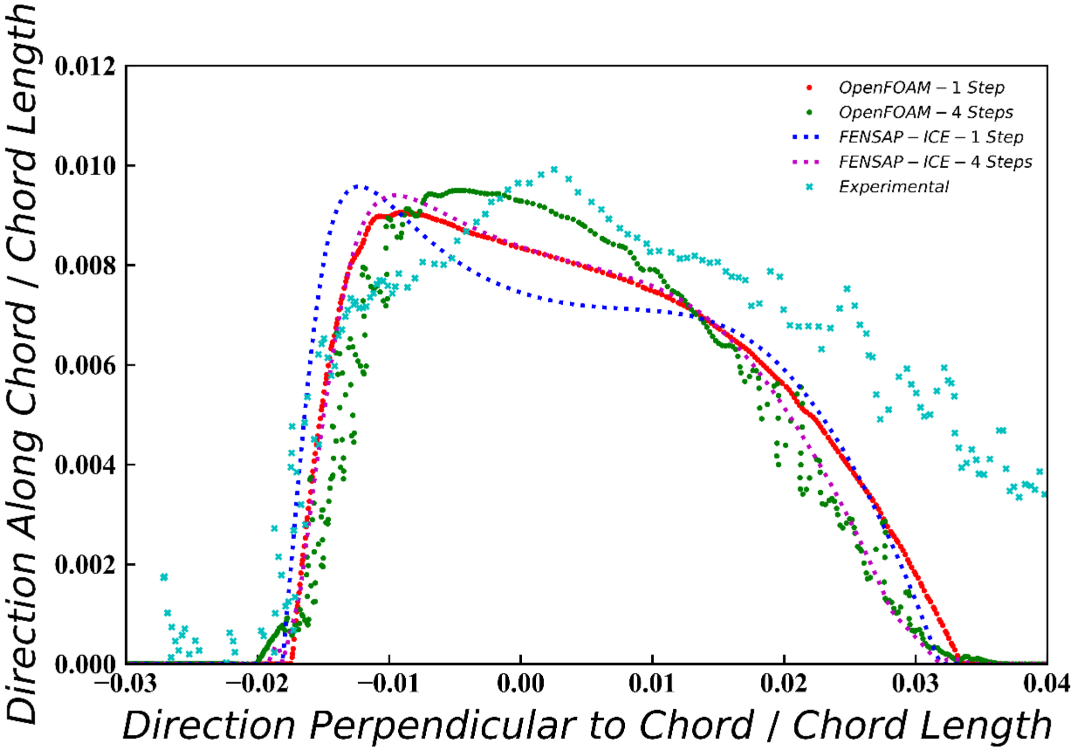


Fig. 49: NACA23012 Case net ice accretion for various time-steps. Experimental results from Broeren, A.P. et al. (2011) are included [53].

5.2. Shin and Bond Cases

5.2.1. OpenFOAM® Model Case Setup

There will be 2 cases explored here-in, both based off experimental runs by Shin, J. and Bond, T.H. (1992) for ice accretion on a NACA0012 airfoil with a 0.5334 m chord at a 4° angle of attack [55]. The back 1% of the airfoil was cut-off in all calculations to meet the demands of the OpenFOAM®-based cut-cell methodology created. The cut-cell methodology was used in all droplet calculations as it was found to be more accurate than the IBM methodology.

In both cases, 20 μm droplets will be used, and a very narrow droplet distribution will be assumed, meaning that only the MVD droplet will be simulated. The temperatures are constant at 247 K and 1 bar of pressure will be used in both cases, meaning the density and dynamic viscosities of both cases will be the same for the incompressible OpenFOAM®-based model at 1.41 kg/m^3 and 1.59E-6 Pa·s respectfully. This means that the Reynolds number over the chord for Case 1 is 3.2E6 and the Mach number is 0.21. For Case 2 the increased freestream velocity means that the Reynolds number over the chord changes to 4.9E6 and the Mach number increases to 0.33.

The difference in the cases is in the freestream velocities and liquid water contents. The case referred to here-in as “Shin and Bond Case 1” has a freestream speed of 67.1 m/s and a freestream liquid water content of 1 g/m^3 while for “Shin and Bond Case 2” the freestream is velocity is 103 m/s and the freestream liquid water content is 0.55 kg/m^3.

The general mesh setup explained previously was used. In both cases, the expansion rate in the outer region was kept at 5%, the farfield was placed 10 chords away, and the inner region was meshed with spacings to meet an average y^+ of approximately 45. For Case 1 this means that the inner region spacing is 8E-4 m and for Case 2 this means 5.4E-4 m. The airfoils were placed at a 4° angle inside the inner region.

Little impact in the collection efficiency curve resulted from decreasing the outer region expansion rate from 5% to 2.5%, as seen in Table 10. For Case 1 this would correspond to going from 119 cells from the centre of farfield to the nearest point in the region separating inner and outer regions to 207 cells, and for Case 2 this would correspond to going from 127 to 223 cells. The total collection efficiency was found to increase by 0.12% for Case 1 and 0.19% for Case 2. This implies that in these cases, the discretization error in the expansion direction of the outer region on these meshes is likely small compared to other errors. Based on a first-order Richardson extrapolation, the expansion effects are responsible for a total of approximately 0.28% error on the 5% mesh for Case 1 and 0.45% for Case 2, and these are likely over-estimations as the real scaling is likely closer to second-order in this region based off of the numerical schemes used.

Table 10: Initial total collection efficiency per chord for Shin and Bond Case 1 and 2 for different expansion rates the OpenFOAM®-based model.

Expansion Rates	Number of Cells in Direction from Inner to Outer Boundary	Initial Total Collection Efficiency per Chord
Case 1		
5%	119	0.03498
2.5%	207	0.03503
Case 2		
5%	127	0.04208
2.5%	223	0.04217

Using the non-conservative ending time-step model so that the effects be compared based on the final ice mass, the effects of the time-step discretization were analyzed, as seen in Table 11. Changing the time-step from 1 s to 0.1 s for the course of the tests (6 min for Case 1 and 7 min for Case 2), single-step test was found to change final ice mass by 0.81% for Case 1 and 0.90% for Case 2, which means that based on a first-order Richardson extrapolation of the time-step, the temporal discretization error impacts the total ice mass by approximately 0.90% for Case 1 and 0.98% for Case 2 for a 1 s time-step. It is thought that using the conservative time-step model will further minimize the impact of this as the final ice mass will be independent of the time-step and only the resultant ice shape will be impacted.

Table 11: Net amount of accreted ice per span for NACA23012 for different time-steps using the OpenFOAM®-based model with the non-conservative ending time-step model for a single-step.

Time-step (s)	Net Ice Accreted per Span (kg/m)
Case 1	
1	0.6394
0.1	0.6446
Case 2	
1	0.7477
0.1	0.7543

5.2.2. FENSAP-ICE Setup

The O-Grid FENSAP-ICE mesh setup explained previously was used for both cases. The temperature was set to 247 K and the pressure to 1 bar. For the final grid of Case 1, 800 divisions were placed along the airfoil and an extra 4 on the backend, a 5% expansion rate was used, with a 2E-4 m minimum cell size, and a 10-chord farfield distance. For the final grid of Case 2, 800 divisions were placed along the airfoil and an extra 4 on the backend, a 5% expansion rate was used, with a 1.3E-4 m minimum cell size, and a 10-chord farfield distance. The final meshes can be seen in Fig. 50.

Mesh independence studies were performed, seen in Table 25 for Case 1 and Table 26 for Case 2. The difference in-between an 80-chord farfield distance and a 10-chord farfield distance was found to be 0.67% for Case 1 and 0.59% for Case 2 on the meshes tested. The difference caused by changing the expansion rate from 5% to 2.5% on the meshes tested was found to be

0.08% for Case 1 and 0.05% for Case 1, and based on a first-order Richardson extrapolation this means that for a 5% expansion rate, there is approximately a 0.19% error due to the expansion rate for Case 1 and 0.11% for Case 2. Increasing the number of divisions along the airfoil from 402 to 804 was found to change the total collection efficiency by 0.14% for Case 1 and 0.44% for Case 2. Based on a first-order Richardson extrapolation, this means that the 804 division test has approximately these same errors remaining. After a 4-step ice accretion for 6 min for Case 1 / 7 min for Case 2, the total mass of accreted ice per span was found to change by 1.2% from 402 divisions to 804 divisions for Case 1 and 1.4% for Case 2. For Case 1 increasing the minimum node spacing from 2E-4 m, corresponding to roughly to an average y^+ for the wall adjacent nodes of 45, to 2.6E-4 m, corresponding to an average y^+ of 60 resulted in a change in the total collection efficiency of 0.8%. For Case 2 increasing the minimum node spacing from 1.3E-4 m, corresponding to roughly to an average y^+ for the wall adjacent nodes of 45, to 1.73E-4 m, corresponding to an average y^+ of 60 resulted in a change in the total collection efficiency of 2.2%. Minimum wall spacings corresponding to approximate y^+'s of 45 for comparability to tests used in the OpenFOAM® model as refining down to the viscous sublayer means that the k-ω SST and k-ε model results may start increasing in their deviation from one another.

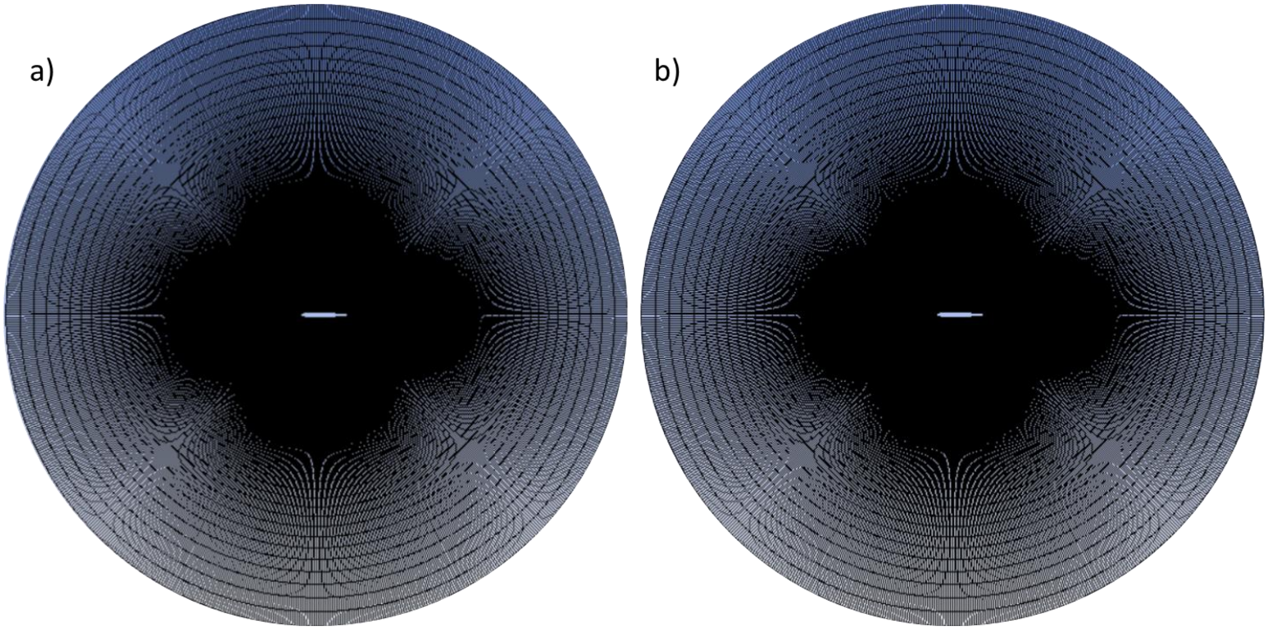


Fig. 50: Final FENSAP-ICE structured O-grids for use in Shin and Bond Case 1 (a) and 2 (b).

5.2.3. Inner Region Mesh Size and Airfoil Orientation

The effects of the inner region mesh spacing on the initial collection efficiency for the OpenFOAM®-based model were tested for both cases, as seen in Table 12. Mesh spacings of 1.6E-3 m, 8E-4 m, and 4E-4 m were tested for Case 1 and mesh spacing of 1.08E-3 m, 5.4E-4 m, and 2.7E-4 m were tested for Case 2, corresponding to average y^+'s of 90, 45, and 22.5. The outer region expansion rate was kept at 5% and the outer region distance was kept at 10 chords. The resulting initial collection efficiency curves are shown in Fig. 51 and Fig. 52 for Case 1 and Case 2 respectfully. The resulting initial total collection efficiencies are given in Table 12.

On the coarsest grid, the OpenFOAM®-based prediction of the total collection efficiency is 4.0% different than the FENSAP-ICE result for Case 1 and 5.6% different, and on the finest grid, the total collection efficiency is 1.2% different than the FENSAP-ICE result for Case 1 and 3.5% different for Case 2. The finest grid test is associated with an average y^+ of ~23 over the airfoil. This means that the cells nearest to the wall on average are dipping into the buffer region of the wall model, which may indicate some additional issues are present in the boundary layer for these tests even though the discretization error is smaller. On the 45 y^+ grids, the total collection efficiencies are 0.82% different than the FENSAP-ICE result for Case 1 and 2.2% different for Case 2.

While in most cases tested the setup used is such that the freestream flow is pointing straight from the inlet on the left to the outlet on the right boundary, with the airfoil in the inner region being rotated based on the angle of attack, one alternative description is tested here where the flow in the freestream is instead angled based on the angle of attack, and the chord of the airfoil is aligned with the cells, i.e. pointing from the inlet to the outlet, referred to here-in as the "chord-aligned" orientation. The rotation of the airfoil changes the number of small cut-cells and the distribution of nearest cell wall spacings, which may provide complex changes to the results based on the orientation.

It was found for Shin and Bond Case 1 that the predicted total collection efficiency differed by 1.2% from FENSAP-ICE results for the chord-aligned orientation, while the typically used orientation has a 0.82% difference, however, based on Fig. 51, the decision over which orientation to use may be questionable, as when comparing meshes with the same inner region spacing, the initial collection efficiency determined using the chord-aligned airfoil is more in-line with the FENSAP-ICE results with the notable exception of the region of maximum collection efficiency. These differences in-between the orientations are expected to be small for the small angles of attack tested here-in.

Table 12: Shin and Bond Case 1 and 2 initial total collection efficiency vs inner region spacing.

Inner Region Spacing (m)	Initial Total Collection Efficiency per Chord
Case 1	
1.6E-3	0.03386
8E-4	0.03498
4E-4	0.03568
Chord-Aligned – 8E-4	0.03570
FENSAP-ICE	0.03527
Case 2	
1.08E-3	0.04349
5.4E-4	0.04208
2.7E-4	0.04264
FENSAP-ICE	0.04119

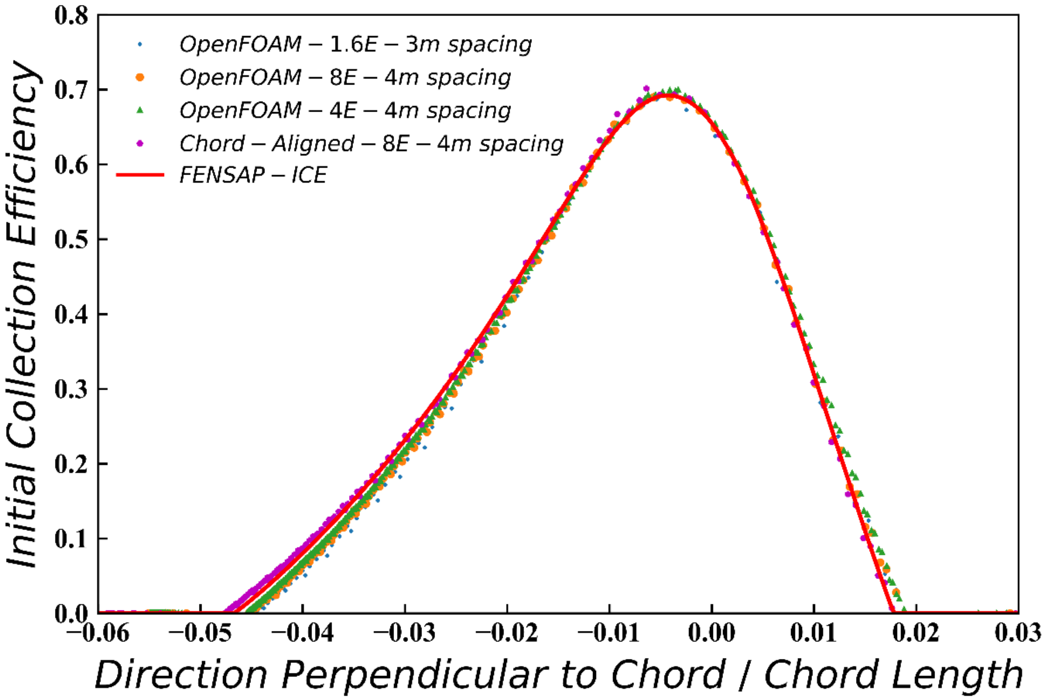


Fig. 51: Shin and Bond Case 1 initial collection efficiency curve for various inner region spacings and 2 orientations.

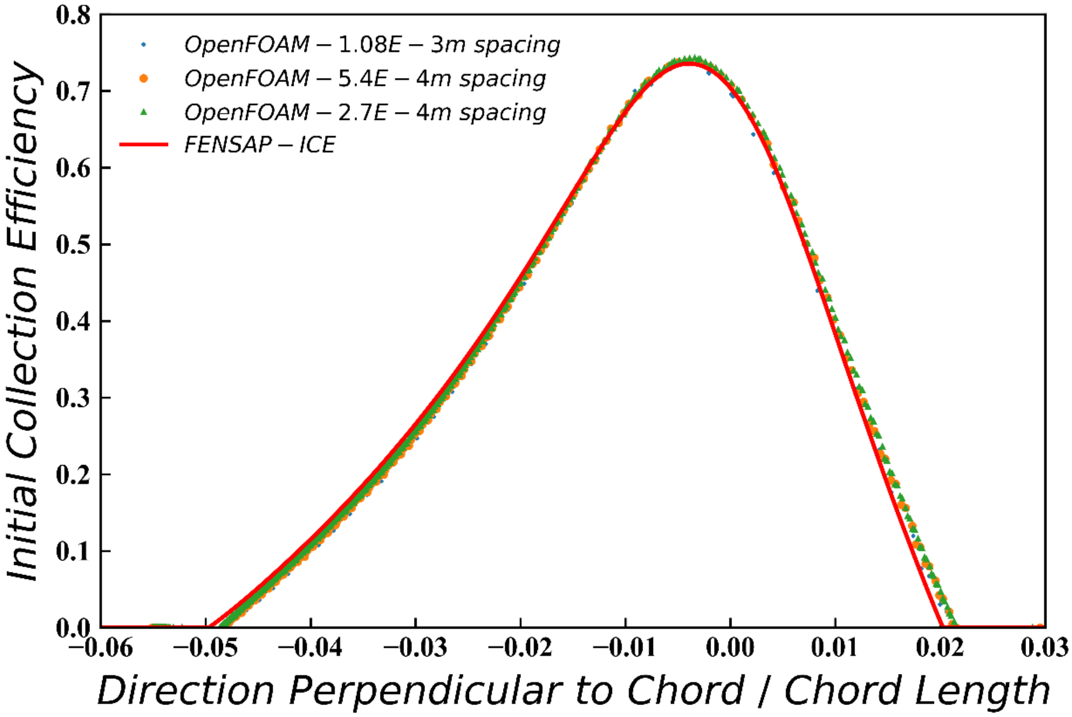


Fig. 52: Shin and Bond Case 2 initial collection efficiency curve for various inner region spacings.

5.2.4. Numerical Roughness and Artificial Diffusion of Evolution Velocity Field

Small errors in boundary reconstruction for cut-cell, the discretization of air, discretization of droplets, and discretization of boundary convection/reinitialization add together to small local errors in the boundary. If not dealt with by smoothing the boundary, it was found that in many cases these errors can grow into bumps along the boundary and can become quite severe, growing into spikes and growths as seen in Fig. 53-Fig. 55 and Fig. 56-Fig. 58, especially as the number of pseudo-time-steps is increased. These can be seen in results from FENSAP-ICE under certain conditions, in particular, if a high number of pseudo-times are used as in Fig. 59 and have been encountered and been issues since the beginnings of the field, being talked about for example as being issues in early LEWICE codes by Shaw, R.J. in 1984 where it was discussed that artificial boundary smoothing may be appropriate [13]. This said, it was found that the proposed model was even more susceptible to these issues than FENSAP-ICE. While experimentally, some roughness can be seen on the surface for rime icing, for a smooth Eulerian methodology which attempts to model some average ice shape, the result is often rougher than expected, and sharp spikes are of particular concern as these can lead to issues in the cut-cell methodology, total ice mass calculations (and therefore the ending time-step and post-processing accuracy), accuracy in normal direction calculation, poorly predicted local boundary separation due to the k-ε model and boundary layer model used, etc. If handled and a smooth surface results from the evolution, some researchers recommend putting in modifications into the boundary layer to handle the roughness via modeling, but in this case, an additional model must be included for the amount of roughness [16] [36].

Some recommendations have been made for systems with similar issues, for example modifying and/or increasing the rate of reinitialization has been recommended and smoothing techniques can be applied here [20]. One alternative solution which is explored here-in is the use of artificial diffusion of the impinging mass fluxes/boundary velocity field in-order to dampen out errors in the growth of the boundary directly, and thus create a smooth boundary before any roughness occurs. This methodology was developed to give more direct control over accuracy vs smoothing. This damping is done via solving the following diffusion equation for some amount of pseudo-time $\Delta t_{ad:}$

$$\frac{\partial m'_{imp}}{\partial t} = \nabla \cdot \kappa_m \nabla m'_{imp} \tag{93}$$

The diffusion coefficient, κ_m, and the total time for diffusion, Δt_{ad}, are kept at constant user-defined values that need tuning. Even if this methodology stops spikes along the boundary that can limit the accuracy of some of the discretizations, especially the cut-cell methodology as it is derived here, and can lead to large separation zones that are unfavourable to the k-ε model, it is important to note that the addition of unphysical diffusion may still limit the accuracy of the resulting ice shape. The advantage of this methodology over for example excessive reinitialization is that by tuning κ_m and Δt_{ad}, one can directly balance accuracy and robustness. Another advantage that was found was that having a smoother shape means fewer separation zones form, which, as a result, means the Navier-Stokes solver has to do less to find the new flow field after boundary evolution, and thus the efficiency was found to improve.

The overall amount of ice accreted over multiple steps was found to change by 1-3% when the artificial diffusion technique was applied, as seen in Table 13. The resulting ice shapes were found to drastically improve in smoothness and therefore repeatability relative to FENSAP-ICE code, as seen in Fig. 53-Fig. 55 and Fig. 56-Fig. 58.

Table 13: Comparison of Shin and Bond Case 1 and 2 net accreted ice mass per span with and without 1E-7 m^2/s artificial diffusion of boundary evolution for 10 s.

Number of Steps	Net Ice Accreted per Span – OpenFOAM®-Based Model, no Artificial Diffusion (kg/m)	Net Ice Accreted per Span – OpenFOAM®-Based Model, Artificial Diffusion (kg/m)	Net Ice Accreted per Span – FENSAP-ICE (kg/m)	Difference Caused by Artificial Diffusion (%)
Case 1				
4	0.4361	0.4261	0.4364	2.3
8	0.4169	0.4234	0.4241	1.6
Case 2				
4	0.5138	0.5126	0.5024	2.0
8	0.5110	0.4991	0.4886	2.3

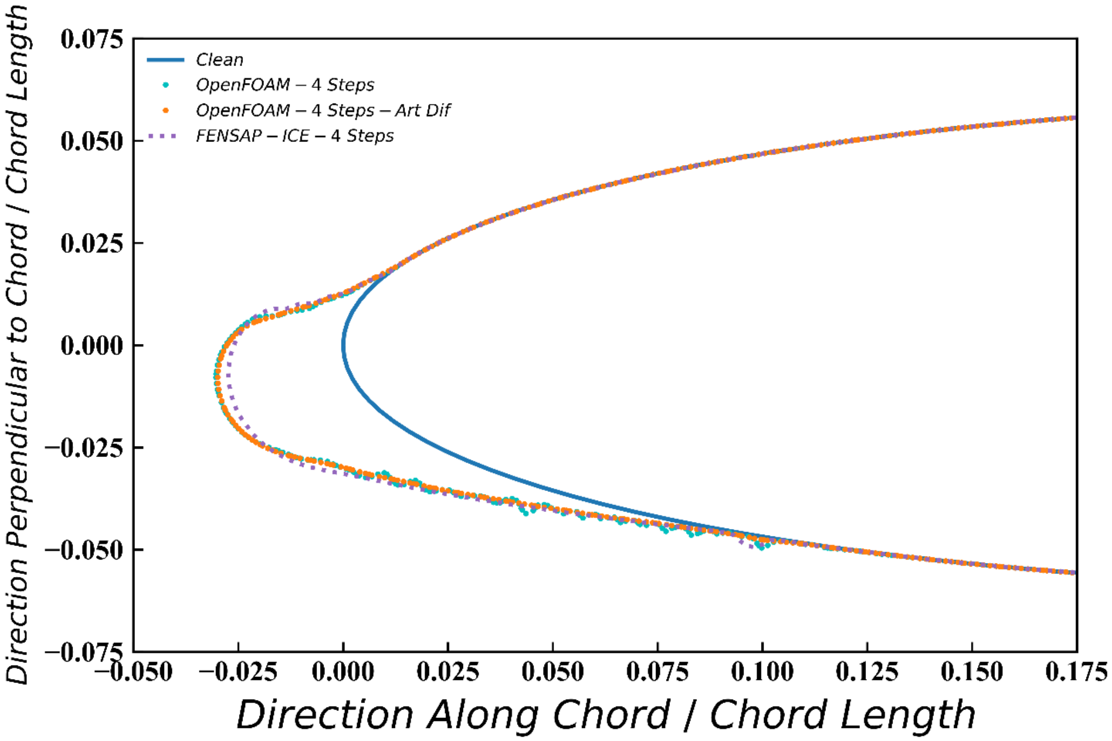


Fig. 53: Comparison of resulting ice shapes from 4-step ice accretion with and without a 1E-7 m^2/s artificial diffusion of the boundary evolution for 10 s for Shin and Bond Case 1.

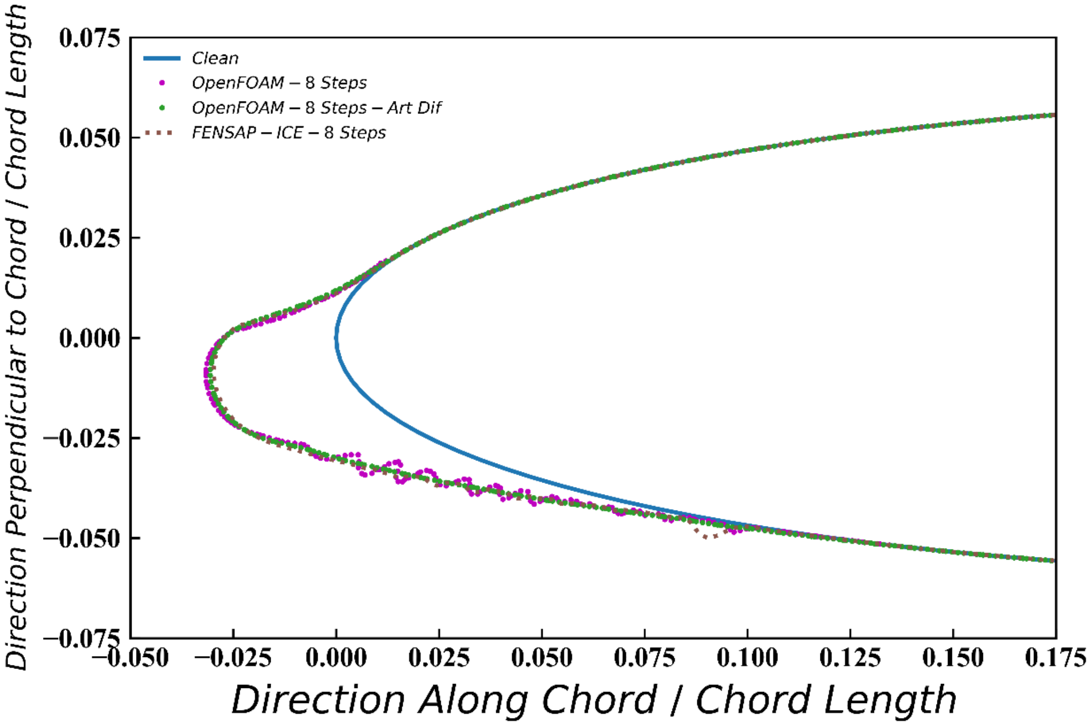


Fig. 54: Comparison of resulting ice shapes from 8-step ice accretion with and without a 1E-7 m^2/s, 10 s artificial diffusion of the boundary evolution for Shin and Bond Case 1.

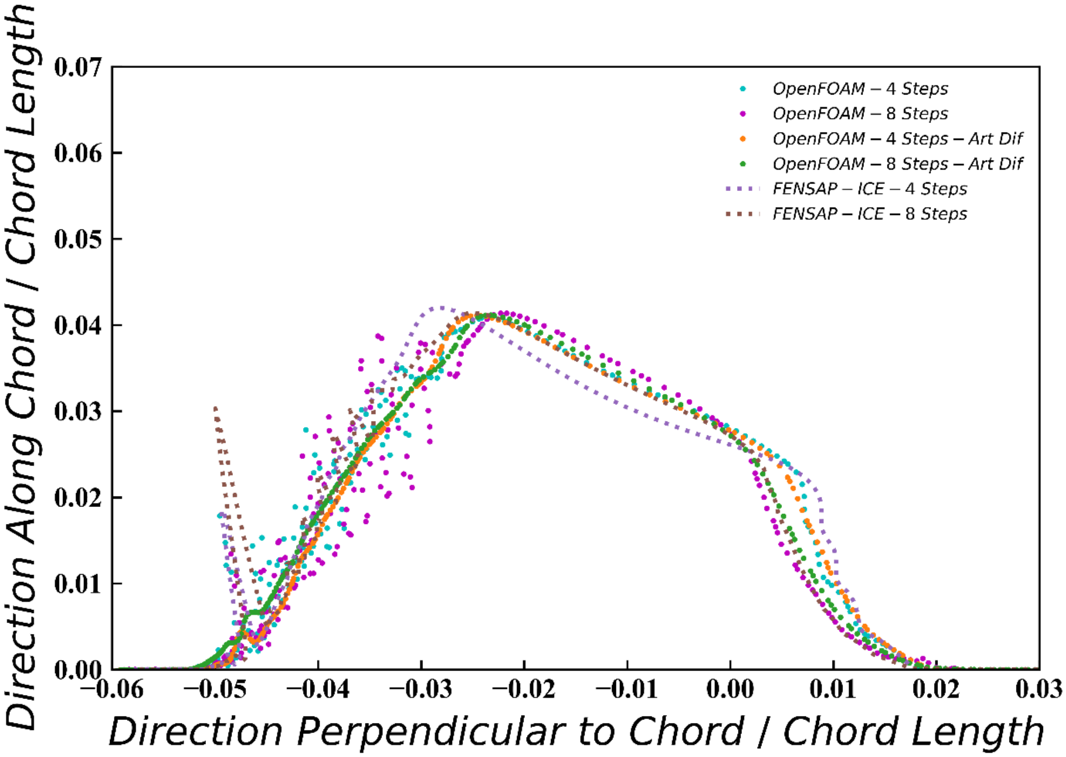


Fig. 55: Comparison of net ice accretion with and without a 1E-7 m^2/s, 10 s artificial diffusion of the boundary evolution for Shin and Bond Case 1.

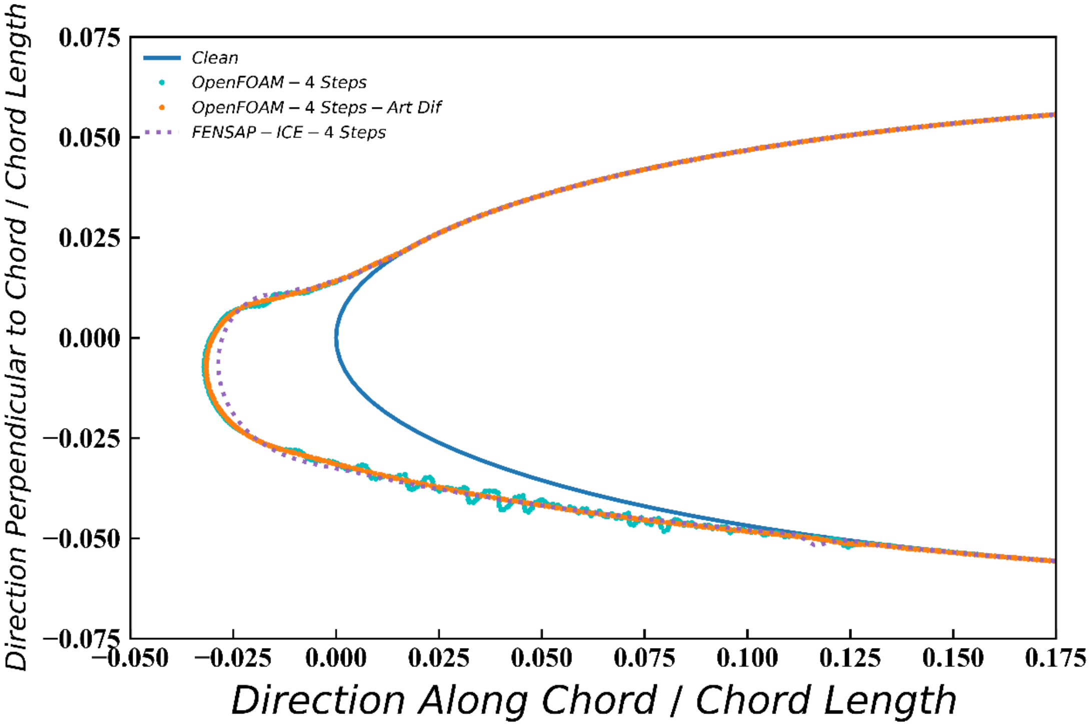


Fig. 56: Comparison of resulting ice shapes from 4-step ice accretion with and without a 1E-7 m^2/s artificial diffusion of the boundary evolution for 10 s for Shin and Bond Case 2.

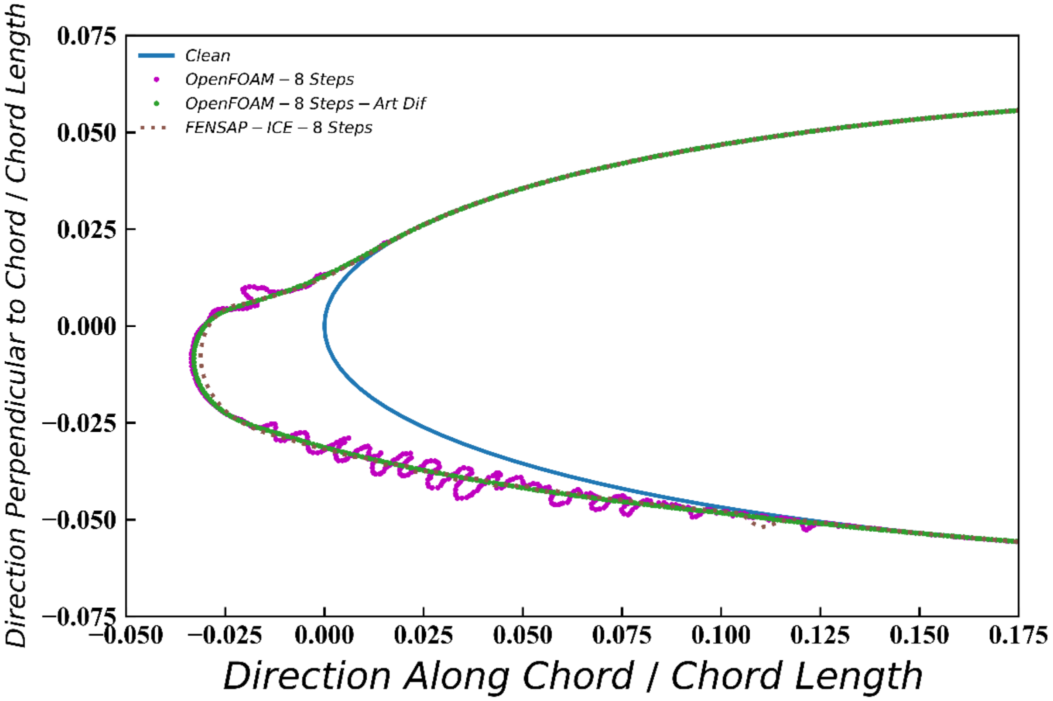


Fig. 57: Comparison of resulting ice shapes from 8-step ice accretion with and without a 1E-7 m^2/s artificial diffusion of the boundary evolution for 10 s for Shin and Bond Case 2.

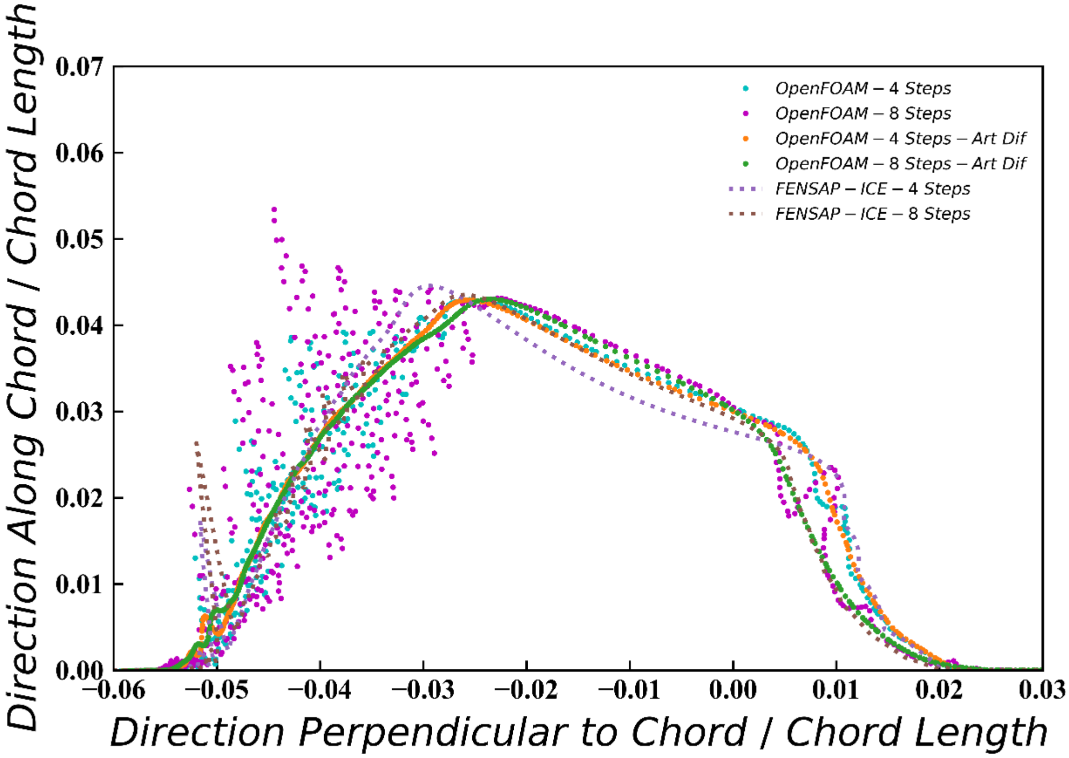


Fig. 58: Comparison of net ice accretion with and without a 1E-7 m^2/s, 10 s artificial diffusion of the boundary evolution for Shin and Bond Case 2.

5.2.5. Number of Steps

Using the artificial diffusion technique described previously to modify the impinging mass fluxes field with a 1E-7 m^2/s coefficient for 10 s using 1 s backwards Euler time-steps, Shin and Bond Case 1 and 2 are analyzed for 4-step, 8-step, and 16-step tests are compared to FENSAP-ICE results and experimental results from Shin, J. and Bond, T.H. (1992) [55]. Expansion rates in the 10-chord outer region were set to 5%, and the inner region spacing was set to 8E-4 m for Case 1 and 5.4E-4 m for Case 2.

As seen in Table 14, for 4 steps the predicted ice mass is 2.4% lower than the FENSAP-ICE result for Case 1 and 2.0% higher for Case 2. For 8 steps the predicted ice mass is 0.17% lower than the FENSAP-ICE prediction for Case 1 and 2.2% higher in Case 2. This and comparability of ice shapes resulting from the OpenFOAM®-based model and FENSAP-ICE seen in Fig. 60-Fig. 62 and Fig. 63-Fig. 65 lend credence to the validity of the model proposed. Overall the resulting ice shapes from the proposed model are again closer from FENSAP-ICE results than to experimental results as seen in Fig. 60-Fig. 62 and Fig. 63-Fig. 65. The resulting ice shapes from FENSAP-ICE notably appear closer to each other more from 4 to 8 steps.

For 16 steps no comparison could be made as the FENSAP-ICE simulation failed for both Case 1 and Case 2 but did not for the OpenFOAM®-based model. The meshes at the failed time-step from FENSAP-ICE, seen in Fig. 59, issues with roughness along the boundary and remeshing issues hindered the tests. This means that at least for these cases, the proposed model is robust enough to handle scenarios FENSAP-ICE cannot handle easily.

Table 14: Net amount of accreted ice per span for Shin and Bond Case 1 and 2 for different numbers of pseudo-time-steps.

Number of Steps	Net Ice Accreted per Span – OpenFOAM®-Based Model (kg/m)	Net Ice Accreted per Span – FENSAP-ICE (kg/m)
Case 1		
1	0.4508	0.4544
4	0.4261	0.4364
8	0.4234	0.4241
16	0.4192	-
Case 2		
1	0.5341	0.5223
4	0.5126	0.5024
8	0.4991	0.4886
16	0.4947	-

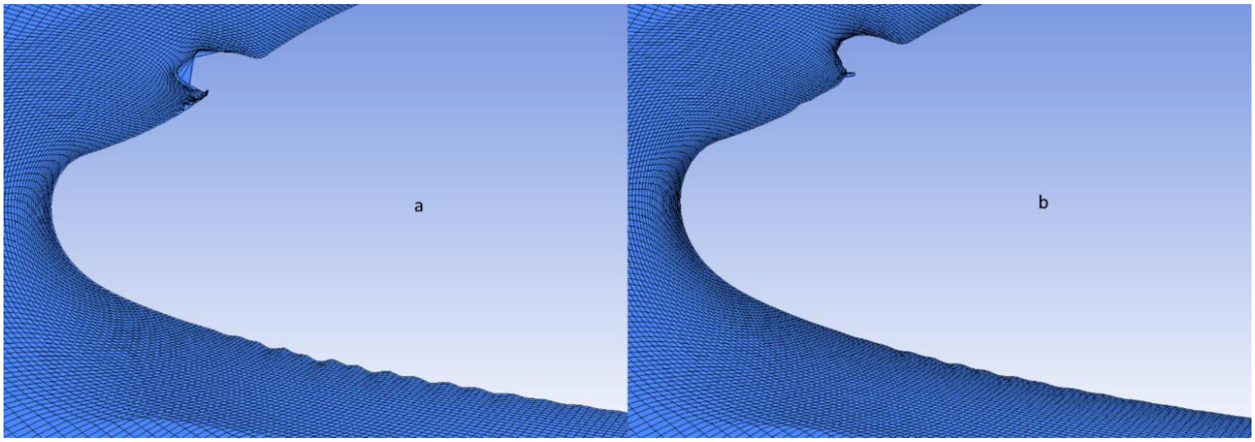


Fig. 59: Failed meshes for Shin and Bond Cases using FENSAP-ICE zoomed-in on front-end where failures occurred. a) Case 1 after 13 pseudo-time-steps. b) Case 2 after 12 pseudo-time-steps.

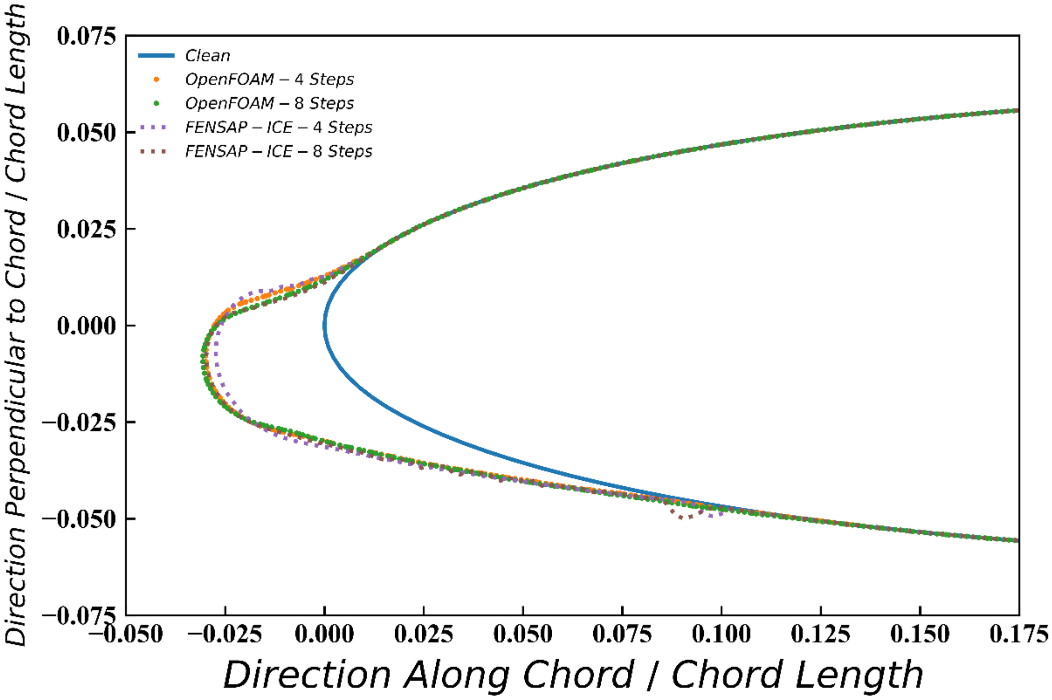


Fig. 60: Shin and Bond Case 1 ice shapes comparison for the effect of the number of steps for the proposed OpenFOAM®-based model vs FENSAP-ICE.

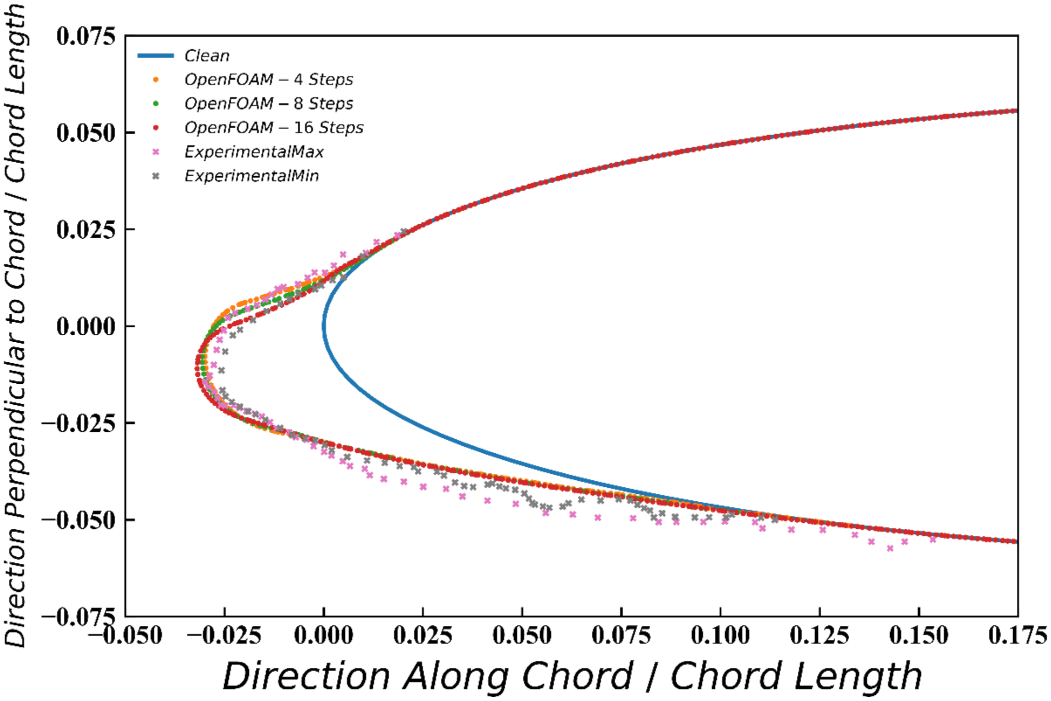


Fig. 61: Shin and Bond Case 1 ice shapes comparison for the effect of the number of steps for the proposed OpenFOAM®-based model and comparison to experimental results from Shin, J. and Bond, T.H. (1992) [55].

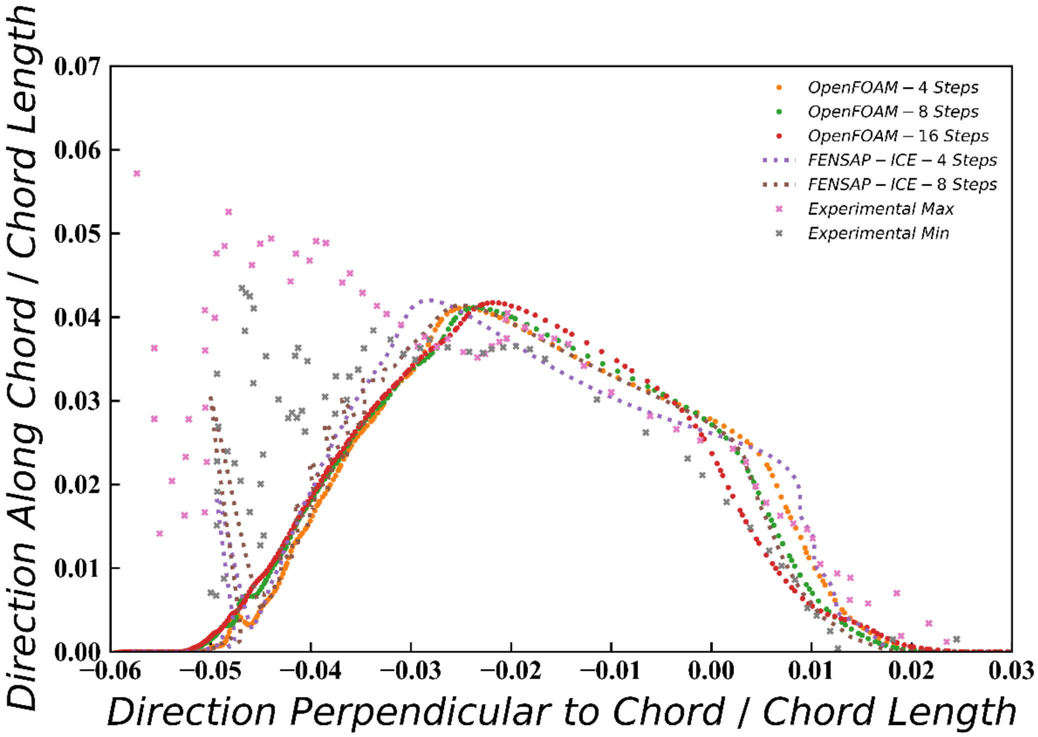


Fig. 62: Shin and Bond Case 1 net ice accretion for various time-steps. Experimental results from Shin, J. and Bond, T.H. (1992) are included [55].

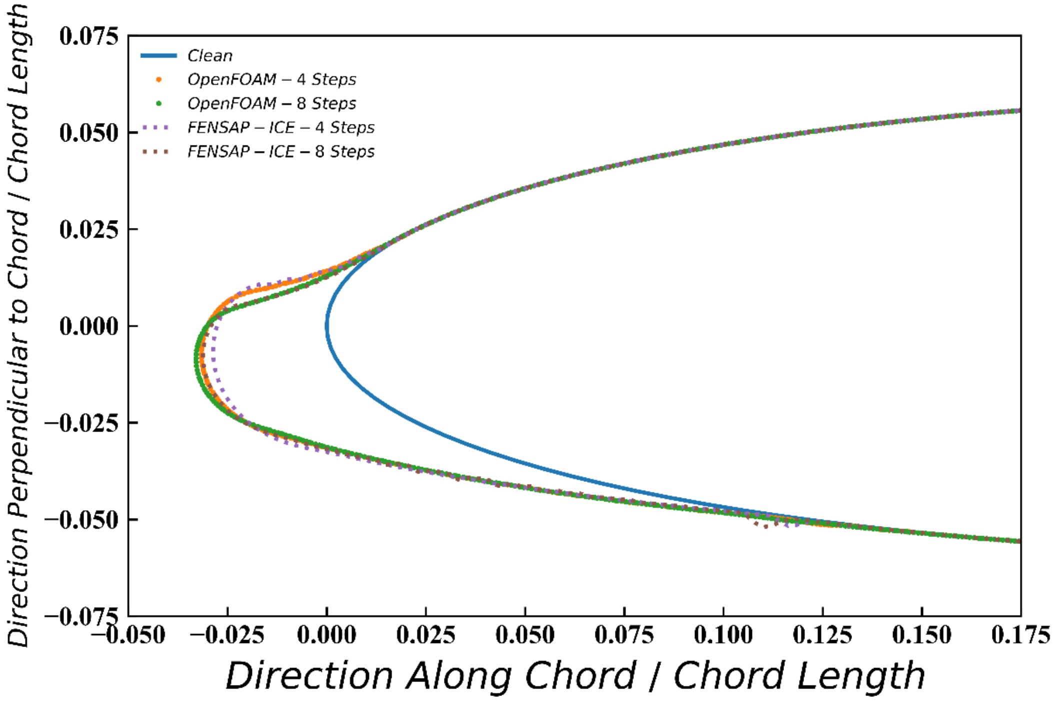


Fig. 63: Shin and Bond Case 2 ice shapes comparison for the effect of the number of steps for the proposed OpenFOAM®-based model vs FENSAP-ICE.

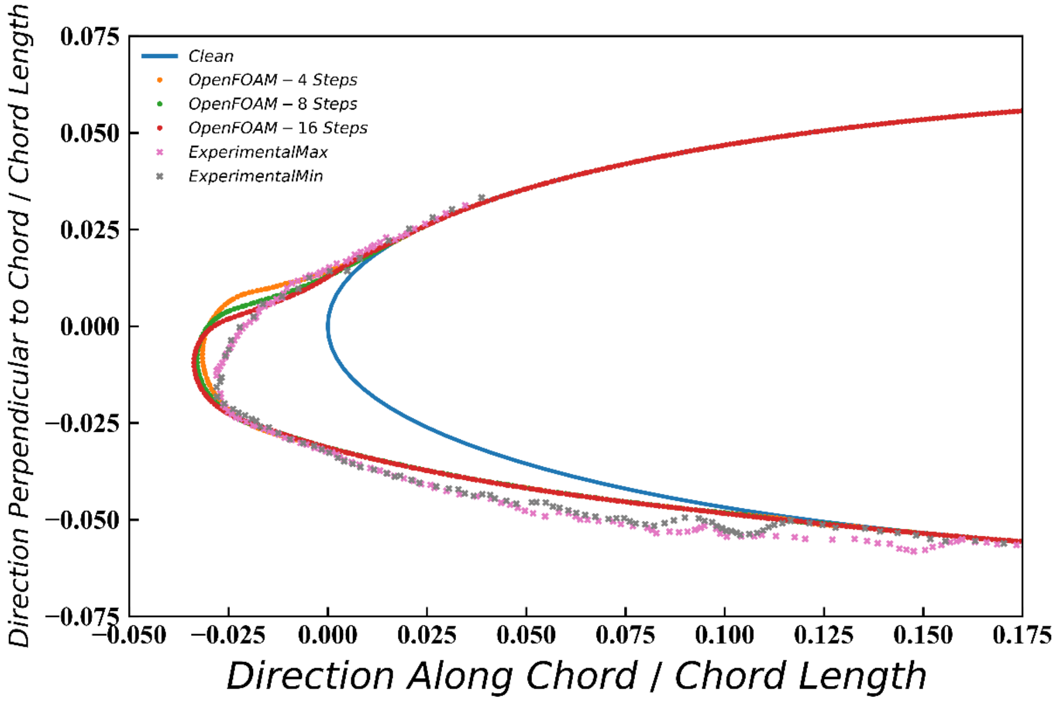


Fig. 64: Shin and Bond Case 2 ice shapes comparison for the effect of the number of steps for the proposed OpenFOAM®-based model and comparison to experimental results from Shin, J. and Bond, T.H. (1992) [55].

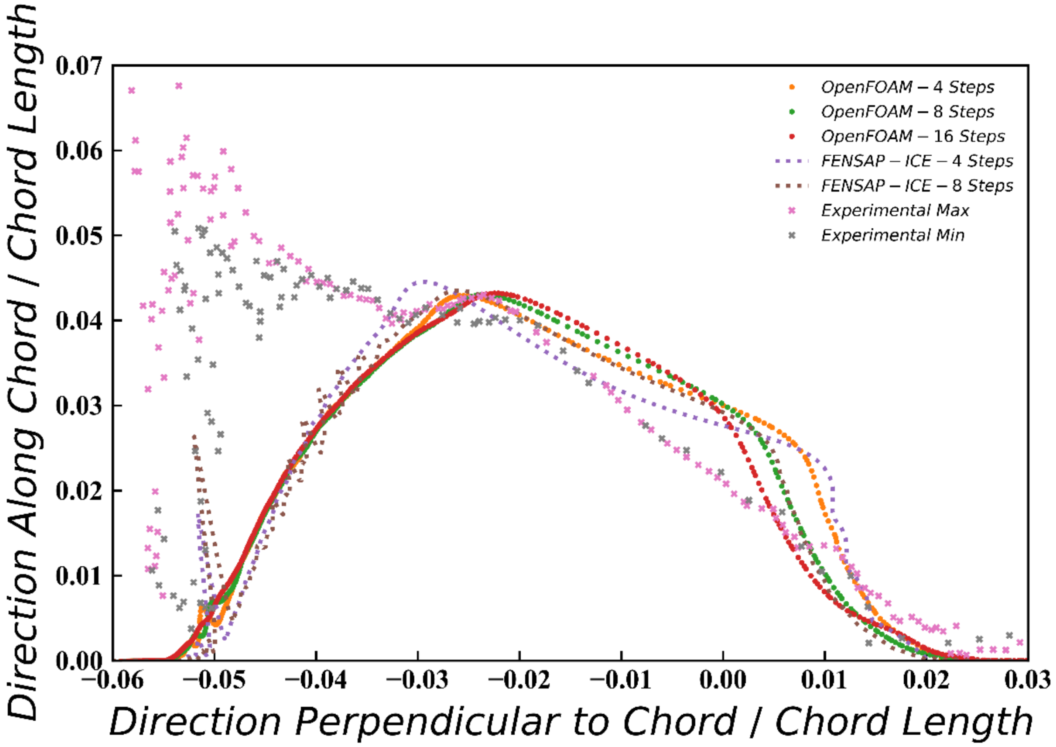


Fig. 65: Shin and Bond Case 2 net ice accretion for various time-steps. Experimental results from Shin, J. and Bond, T.H. (1992) are included [55].

6. Conclusions

6.1. Novel Contributions

The following novel contributions have been made:

- Incorporated high Reynolds wall functions into a non-body-fitted mesh framework to simulate rime icing with a relatively coarse, static mesh (*Section 2.7 - Near-Wall Air Model*).

- Created an immersed boundary and a cut-cell methodology for the near-wall droplet modeling on a non-body-fitted mesh. Identified this as being a major contributor to overall discretization error in the scheme and the cut-cell methodology as being roughly twice as accurate overall as the immersed boundary methodology (*Section 2.8 - Droplet Model* for model and *Section 3 - Cut-Cell and Immersed Boundary Method Comparison* for validation/comparison).

- Developed a novel methodology for creating smooth boundary evolution via artificial diffusion of the level set evolution field which is highly controllable/tunable (*Section 5.2.4 - Numerical Roughness and Artificial Diffusion of Evolution Velocity Field*).

- Developed a stop condition for the evolution of the level set field which drastically improves global conservation of the ice mass, regardless of what happens during the evolution process (*Section 2.11.3 - Boundary Evolution* for model and *Section 5.1.4 - Ending Time-step Model* for validation).

- Created simulations with this methodology inside of the open-source code OpenFOAM® 5.0 and achieved total collection efficiencies and net ice accretions typically within 5% of FENSAP-ICE results and with similar resultant ice shapes (*Section 5.1.5 - Number of Steps* and *Section 5.2.5 - Number of Steps*).

- The proposed model was able to handle multiple scenarios that caused FENSAP-ICE to crash (*Section 5.2.5 - Number of Steps*).

6.2. Sources of Error and Recommendations

While numerous assumptions may lead to some small errors in the system, the following sources of error in the validation and next steps for the model and simulation environment have been identified.

6.2.1. Experimental Errors

In comparing to experimental data, it is important to note that these results are not perfect. From the LEWICE 2.0 validation manual, the major issues with experimental data in the field of icing are repeatability issues and tracing errors [54].

Repeatability errors refer to issues such as having a slight scratch on the sprayer, or not accurately measuring the droplet size distribution which limits the repeatability of the experiment, both physically and numerically.

Tracing errors are slightly more unique to the field and involve errors in the drawing of the resultant ice shapes by experimenters. These can include issues such as not centring the axis of the drawing the 3 and poor drawings. Here-in drawings were then transcribed electronically from experimental sources via drawing over the drawings, adding another layer of tracing error.

6.2.2. Discretization Errors

It was identified that the discretization error may be lowered by updating all discretization schemes used to have a special case for the near boundary cells. This may take some time as several schemes must be updated, including *cellLimited linearUpwindV* and *limitedLinear01* divergence schemes for boundary and boundary-adjacent cells, as well as the *Gauss linear* gradient scheme for boundary-adjacent cells, and the linear scheme for diffusion in boundary-adjacent cells. This is important towards lowering the discretization errors as this generates a first-order error near the boundary.

A cut-cell boundary layer model may be recommended to remove errors over the front of the airfoil seen in the velocity field (see Fig. 17), and if this is done, the diffusion and gradient discretizations will need to be updated in the boundary cells as well.

Better handling of some of the specific special cases described in Fig. 9 in the cut-cell regime to handle spikes, particularly at the back-end of the airfoils.

6.2.3. Model Extensions

Several extensions to the current model can be made. A few to consider are:

- To better account for separation, if resolving the boundary layer down to the viscous sublayer is possible, the model may be updated to work with, for example, k-ω SST. This may require a complex Cartesian remeshing algorithm to be feasible.

- Variable ice density models.

- Subtle droplet effects such as droplet deformation, 2-way coupling, splashing, coalescence, breakup, collisions.

- Supercooled Large Droplets: While typical icing encounters occur with droplets with diameters of 10-50 μm, supercooled large droplets (SLD) can have diameters of up to 1000 μm, or about 100x larger than the smallest of typical icing encounters [3]. These large diameters mean that the can go even farther aft on the wing, into the often unprotected regions of the aircraft [3], which can lead to dangerous conditions, and thus are of active interest. These size scales may provide challenges to the current model due to some of the neglected droplet effects such as deformation.

- Surface Roughness: Can be added to the boundary layer model. This may be especially important for mixed icing scenarios, as these tend to produce very rough iced surfaces [2].

- Glaze Ice: Glaze ice presents with not only new physics to be modelled such as the thermodynamics of the problem, but with new issues that require addressing such as the handling of the runback water model on a body-fitted mesh. The ice horns that can form under glaze ice scenarios are expected to lead to issues with the current formulation as the strong separation generated behind them means that the k-ε approach may no longer be recommendable and a higher resolution of the boundary may be recommended in certain regions, the spiked tips means that the current cut-cell formulation will struggle, that the artificial diffusion technique for the evolution velocity may be questionable, and that the reinitialization techniques which naturally smooth the boundary may also be questionable or require higher-order discretization.

- Compressibility Effects: Using a compressible solver may be especially important for glaze ice simulations as the heat generated near the stagnation point may change the icing behaviours.

- 3D: One of the major issues with moving the simulation environment into 3D is speed, due to the use of Cartesian cells and the static mesh, this may not be efficient in 3D tests. Special remeshing algorithms may help in the transition to 3D by improving efficiency.

- Unsteady Formulation: In the current formulation the accuracy of the model is directly dependent on the choice of pseudo-time-step. Creating a fully unsteady formulation could remove this. Such a modification may require major changes to the implementation of the wall functions to include the effect of the boundary motion.

7. References

[1] European General Aviation Safety Team, "In Flight Icing - For General Aviation Pilots," 2015.

[2] Transport Canada, "Commercial and Business Aviation Advisory Circular (CBAAC) No. 0130R: Guidance Material," Government of Canada, 2018.

[3] M. Wadel, "A Pilot's Guide to Inflight Icing - Module I - Before You Fly," NASA, 27 September 2016. [Online]. Available: https://aircrafticing.grc.nasa.gov/1_1_2_2.html. [Accessed 1 April 2020].

[4] R. Pruvis and F. Smith, "Improving aircraft safety in icing conditions," in *UK Success Stories in Industrial Mathematics*, Springer, 2016, pp. 145-151.

[5] European Aviation Safety Agency, "Notice of Proposed Amendment 2015-07 - Use of comparative analysis when showing compliance with SLD icing specifications," 2015.

[6] National Research Council Canada, "The National Research Council of Canada's Altitude Icing Wind Tunnel allows manufacturers to certify aircraft components and probes," Government of Canada, 14 June 2017. [Online]. Available: https://nrc.canada.ca/en/stories/national-research-council-canadas-altitude-icing-wind-tunnel-allows-manufacturers-certify-aircraft. [Accessed 1 April 2020].

[7] S. Bourgault-Côté, K. Hasanzadeh, P. Lavoie and E. Laurendeau, "Multi-Layer Icing Methodologies for Conservative Ice Growth," in *7th European Conference for Aeronautics and Space Sciences*, Politecnico di Milano, Campus Bovisa La Masa Via Lambruschini, 2017.

[8] Y. Bourgault, W. Habashi, J. Dompierre and G. Baruzzi, "A Finite Element Method Study of Eulerian Droplets Impingment Models," *International Journal For Numerical Methods in Fluids,* vol. 29, pp. 429 - 499, 1999.

[9] P. Trontin, G. Blanchard, A. Kontogiannis and P. VIlledieu, "Description and assessment of the new ONERA 2D icing suite IGLOO2D," in *9th AIAA Atmospheric and Space Environments Conference*, Denver, Colorado, 2017.

[10] T. Hedde and D. Guffond, "ONERA three-dimensional icing mdoel," *AIAA Journal,* vol. 33, no. 6, pp. 1038 - 1046, 1996.

[11] W. Wright, "User Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.2.2," NASA, 2002.

[12] R. Flemming, R. Britton and T. Bond, "Role of Wind Tunnels and Computer Codes in the Certification and Qualification of Rotorcraft for Flight in Forecast Icing," NASA Technical Memorandum - 106747, 1994.

[13] R. Shaw, "Progress Toward the Development of an Aircraft Icing Analysis Capability," NASA Technical Memorandum - 83562 - AIAA PAPER 84-0105, Reno, Nevada, 1984.

[14] T. Hedde and D. Guffond, "ONERA three-dimensional icing model," *AIAA Journal,* vol. 33, no. 6, pp. 1038-1045, 1995.

[15] H. Beaugendre and M. H. W. François, "FENSAP-ICE's three-dimensional in-flight ice accretion module: ICE3D," *Journal of Aircraft,* vol. 40, no. 2, pp. 239-247, 2003.

[16] ANSYS, Inc., ANSYS FENSAP-ICE User Manual, 2019.

[17] C. Aliaga, M. Aubé, G. Baruzzi, W. Habashi and S. Nadarajah, "A Third-generation In-flight Icing Code: FENSAP-ICE-Unsteady," *SAE Transactions,* pp. 697 - 703, 2007.

[18] S. Osher and J. Sethian, "Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations," *Journal of Computational Physics,* vol. 79, pp. 12 - 49, 1988.

[19] S. Osher and R. Fedkiw, "Level Set Methods: An Overview and Some Recent Results," *Journal of Computational Physics,* vol. 169, no. 2, pp. 463 - 502, 2001.

[20] S. Osher and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.

[21] M. Huangrui, "Mesoscale modeling and direct simulation of explosively dispersed granular materials," University of Waterloo, 2019, PhD Thesis.

[22] ANSYS, Inc., "Chapter 4. Turbulence," in *ANSYS FLUENT 12.0 Theory Guide*, 2009, pp. 4-1 - 4-88.

[23] ANSYS, Inc., "Chapter 2. Turbulence and Wall Function Theory," in *ANSYS CFX-Solver Theory Guide*, 2009, pp. 53 - 97.

[24] J. H. Ferziger and M. Peric, "7. Solution of the Navier-Stokes Equation," in *Computational Methods for Fluid Dynamics, Springer, 3rd Ed.*, Springer, 2002, pp. 157 - 217.

[25] B. Launder and D. Spalding, "The Numerical Computation of Turbulent Fows," *Computer Methods in Applied Mechanics and Engineering,* vol. 3, no. 2, pp. 269 - 289, 1974.

[26] W. Cabot and P. Moin, "Approximate Wall Boundary Conditions in the Large-Eddy Simulation of High Reynolds Number Flow," *Flow, Turbulence and Combustion,* vol. 63, pp. 269 - 291, 1999.

[27] C. Brehm, O. Browne and N. and Ashton, "Towards a Viscous Wall Model for Immersed Boundary Methods," in *2018 AIAA Aerospace Sciences Meeting*, Kissimmee, Florida, 2018.

[28] H. Versteeg and W. Malalasekera, "3.4.2 Flat plate boundary layer and pipe flow," in *An Introduction to Computational Fluid Dynamics, The Finite Volume Method, Second Edition*, Pearson Education Limited, 2007, pp. 57 - 59.

[29] D. Sondak, "Wall functions for the k - [epsilon] turbulence model in generalized nonorthogonal curvilinear coordinates," Iowa State University, Iowa, 1992, PhD Thesis.

[30] K. Sreenivasan, "The Turbulent Boundary Layer," in *Frontiers in Experimental Fluid Mechanics*, vol. 46, Berlin, Heidelberg, Springer, 1989, pp. 159 - 209.

[31] G. Eitel-Amor, R. Orlu and P. I. Schlatter, "Simulation and validation of a spatially evolving turbulent boundary layer up to Reθ= 8300," *International Journal of Heat and Fluid Flow,* vol. 47, pp. 57 - 69, 2014.

[32] F. Liu, "Chapter 2 Wall functions in OpenFOAM," in *A Thorough Description Of How Wall Functions Are Implemented In OpenFOAM*, Chalmers University of Techology, 2017, pp. 25 - 26.

[33] M. Khademi, A. Hezave, D. Mowla and M. Taheri, "A Simple Model for Turbulent Boundary Layer Momentum Transfer on a Flat Plate," *Chemical Engineering & Technology,* vol. 6, pp. 867 - 877, 2010.

[34] C. Rumsey, "VERIF/2DZP: 2D Zero Pressure Gradient Flat Plate Verification Case," NASA Langlet Research Centre TRurbulence Modeling Resource, 28 August 2018. [Online]. Available: https://turbmodels.larc.nasa.gov/flatplate.html. [Accessed 9 February 2020].

[35] X. Tong and E. Luke, "Robust and Accurate Eulerian Multiphase Simulation ofIcing Collection Efficiency Using Singularity Diffusion Model," *Engineering Applications of Computational Fluid Mechanics,* vol. 4, no. 4, pp. 483 - 495, 2010.

[36] P. Lavoie, "Modeling of Thin Water Films on Swept Wings in Icing Condition," Université De Montréal, Montreal, 2017, Master Thesis.

[37] R. Clift and W. Gauvin, "The motion of particles in turbulent gas streams," *Proc. Chemeca,* vol. 14, no. 1, 1970.

[38] R. Johnson, "41: Solution Methods for the Incompressible Navier–Stokes Equations," in *Handbook of Fluid Dynamics - Second Edition*, CRC Press, 2016, pp. 1 - 21.

[39] F. Mentor, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal,* vol. 34, no. 8, pp. 1598 - 1605, 1994.

[40] Y. Cao, Q. Zang and J. Sheridan, "Numerical simulation of rime ice accretions on an aerofoil using an Eulerian method," *The Aeronautical Journal,* vol. 120, no. 1229, pp. 243 - 249, 2008.

[41] M. Vargas, H. Boughton, J. Sims, B. Bleeze and V. Gaines, "Local and Total Density Measurements in Ice Shapes," *Journal of Aircraft,* vol. 44, no. 3, pp. 780 - 789, 2007.

[42] C. Min, "On reinitializing level set functions," *Journal of Computational Physics,* vol. 229, no. 8, pp. 2764 - 2772, 2010.

[43] C. Winkelmann, "Interior Penalty Finite Element Approximation of Navier-Stokes Equations and Application to Free Surface Flows," École Polytechnique Fédérale de Lausanne, 2007, PhD Thesis.

[44] L. Ngo and H. Choi, "Efficient direct re-initialization approach of a level set method for unstructured meshes," *Computers and Fluids,* vol. 154, pp. 167 - 183, 2017.

[45] S. Wirogo and S. Srirambhatla, "An Eulerian Method to Calculate the Collection Efficiency on Two and Three Dimensional Bodies," in *41st Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2003.

[46] T. Phillips and C. Roy, "Richardson Extrapolation-based Discretization Uncertainty Estimation for Computational Fluid Dynamics," *ASME Journal of Fluids Engineering,* vol. 136, no. 12, 2014.

[47] L. Hu, X. Zhu, C. Hu, J. Chen and D. Z., "Calculation of the Water Droplets Local Collection Efficiency on the Wind Turbines Blade," *Journal of Energy Resources Technology,* vol. 139, no. 5, 2017.

[48] J. Hopers, "Eulerian method for super-cooled large-droplet ice-accretion on aircraft wings," University of Twente, 2013, PhD Thesis.

[49] M. Papadakis, S. Wong, A. Rachman, K. V. G. Hung and C. Bidwell, "Large and Small Droplet Impingement Data on Airfoils and Two Simulated Ice Shapes," NASA Technical Report - 2007-213959, E-15275 , 2007.

[50] I. Langmuir and S. Guy, The Collected Works of Irving Langmuir, New York: Pergamon Press, 1960.

[51] R. Jeck, "Cloud Sampling Instruments for Icing Flight Tests: (3) Cloud Droplet Sizers," U.S. Department of Transportation - Federal Aviation Administration - Technical Report - AR-TN06/31, 2006.

[52] D. Bowden, A. Gensemer and C. Skeen, "Engineering Summary of Airframe Icing Technical Data," U.S. Department of Transportation - Federal Aviation Administration - Technical Report - ADS-4, 1963.

[53] A. Broeren, H. Addy Jr., M. Bragg, G. Busch and E. Montreuil, "Aerodynamic Simulation of Ice Accretion on Airfoils," NASA Technical Paper - 2011-216929, 2011.

[54] W. Wright and A. Rutowski, "Validation Results for LEWICE 2.0," NASA, 1999.

[55] J. Shin and T. Bond, "Results of an icing test on a NACA 0012 airfoil in the NASA Lewis Icing Research Tunnel," in *30th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 1992.

8. Appendix

8.1. Case Setups

Table 15: Air, Droplet, and Geometry Properties for Cylinder Case

Property	**Value**
Freestream Velocity	80 m/s
Freestream Pressure	898670 Pa
Temperature	285 K
Air Density	1.10 kg/m3
Dynamic Viscosity	1.78E-5 Pa·s
Mach Number	0.24
Reynolds Number	5.0E5
Diameter of Cylinder	0.1016 m
Liquid Water Content	1 g/m3
Freestream Droplet Volume Fraction	1E-6
Droplet Diameter	16 µm

Table 16: Air, Droplet, and Geometry Properties for the Ma 0.4 NACA0012 Case

Property	**Value**
Freestream Velocity	139 m/s
Freestream Pressure	1 atm
Temperature	300 K
Air Density	1.18 kg/m3
Dynamic Viscosity	1.86E-6 Pa·s
Mach Number	0.4
Reynolds Number	8.8E6
Chord Length	1 m
Angle of Attack	5^{o}
Liquid Water Content	1 g/m3
Freestream Droplet Volume Fraction	1E-6
Droplet Diameter	16 µm

Table 17: Air, Droplet, and Geometry Properties for the Ma 0.2 NACA0012 Case

Property	Value
Freestream Velocity	69.5 m/s
Freestream Pressure	1 atm
Temperature	300 K
Air Density	1.18 kg/m3
Dynamic Viscosity	1.86E-6 Pa·s
Mach Number	0.2
Reynolds Number	4.4E6
Chord Length	1 m
Angle of Attack	5^o
Liquid Water Content	1 g/m3
Freestream Droplet Volume Fraction	1E-6
Droplet Diameter	16 μm

Table 18: Air, Droplet, and Geometry Properties for the NACA23012 Case

Property	Value
Freestream Velocity	77.2 m/s
Freestream Pressure	1 bar
Temperature	248 K
Air Density	1.41 kg/m^3
Dynamic Viscosity	1.59E-6 Pa·s
Mach Number	0.24
Reynolds Number	1.25E7
Chord Length	1.83 m
Angle of Attack	2^o
Liquid Water Content	0.55 g/m^3
Freestream Droplet Volume Fraction	5.5E-7
Droplet Diameter	30 μm
Icing Time	10 min

Table 19: Air, Droplet, and Geometry Properties for the Shin and Bond 1 Case

Property	Value
Freestream Velocity	67.1 m/s
Freestream Pressure	1 bar
Temperature	247 K
Air Density	1.41 kg/m^3
Dynamic Viscosity	1.59E-6 Pa·s
Mach Number	0.21
Reynolds Number	3.2E6
Chord Length	0.5334 m
Angle of Attack	4^o
Liquid Water Content	1 g/m^3
Freestream Droplet Volume Fraction	1E-6
Droplet Diameter	20 µm
Icing Time	6 min

Table 20: Air, Droplet, and Geometry Properties for the Shin and Bond 2 Case

Property	Value
Freestream Velocity	103 m/s
Freestream Pressure	1 bar
Temperature	247 K
Air Density	1.41 kg/m^3
Dynamic Viscosity	1.59E-6 Pa·s
Mach Number	0.33
Reynolds Number	4.9E6
Chord Length	0.5334 m
Angle of Attack	4^o
Liquid Water Content	0.55 g/m^3
Freestream Droplet Volume Fraction	0.55E-6
Droplet Diameter	20 µm
Icing Time	7 min

8.2. FENSAP-ICE Mesh Independence Studies

Table 21: Mesh independence study for the Cylinder Case using FENSAP-ICE.

Number of Nodes Along	Farfield Distance in Diameters	Expansion Ratio	Min Node Spacing (m)	Number of Nodes Perpendicular	Total Collection Efficiency per Diameter
200	10	1.1	1.4E-4	69	0.2019
200	20	1.1	1.4E-4	76	0.1962
200	40	1.1	1.4E-4	84	0.1940
200	80	1.1	1.4E-4	91	0.1930
200	160	1.1	1.4E-4	98	0.1927
50	40	1.1	1.4E-4	84	0.1921
100	40	1.1	1.4E-4	84	0.1945
400	40	1.1	1.4E-4	84	0.1937
200	40	1.05	1.4E-4	149	0.1947
200	40	1.025	1.4E-4	267	0.1957
200	40	1.1	7E-5	91	0.1976
200	40	1.1	3.5e-5	98	0.2015
200	40	1.1	1.75E-5	105	0.2042
200	40	1.1	8.75E-6	113	0.2053
200	40	1.025	8.75E-6	379	0.2066

Table 22: Mesh independence study for the Ma 0.4 NACA0012 Case using FENSAP-ICE.

Number of Cells Along	Farfield Distance in Chords	Expansion Ratio	Min Cell Spacing (m)	Number of Cells Perpendicular	Total Collection Efficiency per Chord
201	2.5	1.05	1.4E-4	140	0.02240
201	5	1.05	1.4E-4	154	0.02244
201	10	1.05	1.4E-4	168	0.02250
201	80	1.05	1.4E-4	211	0.02255
201	5	1.1	1.4E-4	86	0.02242
201	5	1.025	1.4E-4	276	0.02244
402	5	1.1	1.4E-4	86	0.02264
804	5	1.1	1.4E-4	86	0.02270
1608	5	1.1	1.4E-4	86	0.02273
402	5	1.1	9.35E-5	91	0.02232
402	5	1.1	1.87E-4	83	0.02264
402	5	1.1	3.74E-4	76	0.02260

Table 23: Mesh independence study for the Ma 0.2 NACA0012 Case using FENSAP-ICE.

Number of Cells Along	Farfield Distance in Chords	Expansion Ratio	Min Cell Spacing (m)	Number of Cells Perpendicular	Total Collection Efficiency per Chord
5	202	1.05	1.87E-4	148	0.01482
10	202	1.05	1.87E-4	162	0.01486
20	202	1.05	1.87E-4	176	0.01500
80	202	1.05	1.87E-4	205	0.01490
10	402	1.1	1.87E-4	91	0.01496
10	402	1.05	1.87E-4	162	0.01497
10	804	1.05	1.87E-4	162	0.01501
10	804	1.05	3.74E-4	135	0.01470

Table 24: Mesh independence study for the NACA23012 Case using FENSAP-ICE.

Number of Node Along	Farfield Distance in Chord Lengths	Expansion Ratio	Min Node Spacing (m)	Number of Nodes Perpendicular	Initial Total Collection Efficiency per Chord	Accreted Ice Mass per Span After a 4-Step Ice Accretion Test (kg/m)
402	10	1.05	1.8E-4	175	0.02111	-
402	80	1.05	1.8E-4	187	0.02106	-
402	10	1.025	1.8E-4	318	0.02113	-
402	10	1.05	2.9E-4	166	0.02123	-
804	10	1.025	1.8E-4	318	0.02118	0.9480
1608	10	1.025	1.8E-4	318	0.02120	0.9525

Table 25: Mesh independence study for the Shin and Bond Case 1 using FENSAP-ICE.

Number of Node Along	Farfield Distance in Chord Lengths	Expansion Ratio	Min Node Spacing (m)	Number of Nodes Perpendicular	Initial Total Collection Efficiency per Chord	Accreted Ice Mass per Span After a 4-Step Ice Accretion Test (kg/m)
402	10	1.05	2.6E-4	142	0.03548	0.4379
402	80	1.05	2.6E-4	185	0.03572	-
402	10	1.025	2.6E-4	253	0.03551	-
804	10	1.05	2.6E-4	142	0.03556	0.4432
804	10	1.05	2E-4	147	0.03527	0.4364

Table 26: Mesh independence study for the Shin and Bond Case 2 using FENSAP-ICE.

Number of Node Along	Farfield Distance in Chord Lengths	Expansion Ratio	Min Node Spacing (m)	Number of Nodes Perpendicular	Initial Total Collection Efficiency per Chord	Accreted Ice Mass per Span After a 4-Step Ice Accretion Test (kg/m)
804	10	1.05	1.74E-4	151	0.04191	-
804	80	1.05	1.74E-4	193	0.04216	-
402	10	1.05	1.3E-4	156	0.04107	0.4998
402	10	1.025	1.3E-4	281	0.04109	-
804	10	1.05	1.3E-4	156	0.04125	0.5069