An Immersed Boundary Methodology for Multi-Step Ice Accretion using a Level-Set

Pierre Lavoie, Emmanuel Radenac, Ghislain Blanchard, Eric Laurendeau, Philippe Villedieu

An Immersed Boundary Methodology for Multi-Step Ice Accretion using a Level-Set

Pierre Lavoie *
Polytechnique Montreal, Montreal, Quebec H3T 1J4, Canada

Emmanuel Radenac† and Ghislain Blanchard ‡
ONERA / DMPE, Université de Toulouse, F-31055 Toulouse, France

Eric Laurendeau§
Polytechnique Montreal, Montreal, Quebec H3T 1J4, Canada

Philippe Villedieu ¶
ONERA / DMPE, Université de Toulouse, F-31055 Toulouse, France

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

I. Introduction

Numerical tools for the prediction of ice accretion on aircraft are typically based on a quasi-steady assumption where modules are called sequentially and solved to steady state within a time-iterative scheme. The process is illustrated in Fig. 1 where the modules are: (1) a mesh generation tool, (2) a solver for the aerodynamics, (3) a solver to obtain the droplet trajectories and impingement rates, (4) a solver to obtain the wall convective heat transfer (in the boundary layer), (5) a solver to perform a heat and mass balance applied to the deposited water to obtain the ice accretion rate and finally (6) a tool to update the geometry based on the ice thickness evolution. Modules (1) to (6) are embedded in a time loop for which the total ice accretion time is divided in time steps (multi-step) generating successive layers of ice (multi-layer). When using Body-Fitted meshes, a mesh update is required with each new ice layer. It can be repeated several times in order to obtain the final ice shape prediction. This leads to additional costs related to the mesh update and additional difficulty in updating the ice shape which can exhibit unphysical surface overlaps in concave regions when using a Lagrangian approach (displacement of surface mesh nodes, a method which is usually employed in ice accretion codes).

*PhD Candidate, Department of Mechanical Engineering, C.P. 6079, succ. Centre-ville, pierre.lavoie@polymtl.ca
†Research Engineer, Aerodynamics and Energetic Modeling Department, BP 4025 2 avenue Ed. Belin
‡Research Engineer, Aerodynamics and Energetic Modeling Department, BP 4025 2 avenue Ed. Belin
§Professor, Department of Mechanical Engineering, C.P. 6079, succ. Centre-ville, Senior AIAA Member
¶Research Director, Aerodynamics and Energetic Modeling Department, BP 4025 2 avenue Ed. Belin

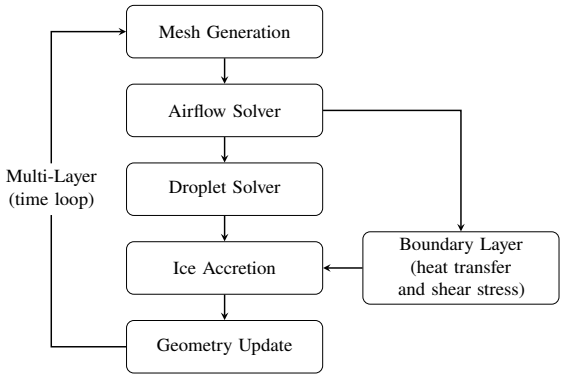


Fig. 1 Sequential call to modules in multi-step icing simulations.

This paper is concerned by the methods required to automate icing suites using a multi-step approach, more specifically by the geometry and mesh updates. An Immersed Boundary Method (IBM) can be used to help automate ice accretion tools by avoiding the re-meshing or at least, minimizing the work required to adapt the mesh.

Immersed Boundary Methods have been applied with success in many fields but have rare applications to the prediction of ice accretion. The CIRA used a discrete IBM to solve compressible inviscid flows on 3D Cartesian grids in [1] which was later extended for compressible viscous flows [2, 3]. A discrete IBM was also applied to an Eulerian droplet solver [4] with the intent of performing ice accretion simulations. However, no ice prediction results using these IBMs have been published yet.

Another research team from the University of Strasbourg applied IBMs to a 3D ice accretion code (NSMB-ICE) with a level-set approach [5]. The compressible viscous flow solver employs a penalty method (an IBM) and the Eulerian droplet solver uses a discrete approach similar to the one of [4]. The level-set approach was initially proposed by [6] for single step ice accretion and is used to update the iced geometry. Multi-step ice accretion results are presented up to 5 steps on a single test case in [5]. According to the authors, the implementation is currently limited to laminar flow and rime ice. Furthermore, no comparison is made against a more classical body-fitted approach nor experimental results. The idea of using a level-set for the prediction of ice accretion is also reused in [7] where multi-step simulations are performed in 2D combined with the use of NURBS. An explicit tracking of the air-ice interface is also performed in order to enforce mass conservation.

Our initial attempt at applying IBMs to an ice accretion suite is presented in [8], where a penalization method (a specific type of immersed boundary method) is applied to the aerodynamic (Euler equations) and droplet solvers (Eulerian formulation). The 1^{st} step of the multi-step process is performed using a Body-Fitted mesh while for the subsequent steps, the ice shape is immersed on the initial mesh. A geometric approach was used to evaluate the signed distance field required by the penalization method. Furthermore, a Lagrangian node displacement approach was used to update the geometry.

This paper extends the contribution of [8] with several key features. First, an improved penalization method suitable for ice horn accretion is applied to the Euler equations [9]. Second, a level-set approach [6] is implemented in the IBM multi-step ice accretion process to solve the issues related to unphysical geometry update, replacing the Lagrangian geometry update. Third, several icing cases are examined, including 2D cases from the AIAA Ice Prediction Workshop [10].

The paper first presents the ice accretion suite used as the development platform, IGLOO2D [11]. Then, implementation details for the penalization and level-set methods are covered: preprocessing, penalization of the Euler equations, penalization of the droplet solver, extraction of the surface data and implementation of the level-set method. A section discusses the benefits of using the level-set method in the IBM ice accretion framework on a manufactured case. Then, rime and glaze ice cases from the AIAA Ice Prediction Workshop (IPW) are used for verification where the Body-Fitted and penalized solutions are compared using the multi-step process. Additional validations are performed for the ice accretion on a NACA0012 airfoil. Finally, ice accretion simulations are performed on a three-element airfoil before conclusions are drawn.

II. Methodology

The 2D ice accretion suite IGLOO2D [11] is used as the development environment. In IGLOO2D, different types of solvers are available for each module but only the ones used in this paper are discussed. The unstructured mesh generation is handled by GMSH [12]. The aerodynamic field is evaluated using the Euler equations and the convective heat transfer is evaluated using a simplified integral boundary layer method (SIM2D) [11]. For the droplet trajectories and impingement evaluation, the Eulerian solver is selected. The ice accretion solver is based on a Messinger-type mass and energy balance to obtain the ice thickness. Finally, the ice geometry is generated by a Lagrangian displacement of the surface nodes. The Eulerian evolution of the geometry using a level-set is also treated in this paper.

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

For the application of the IBM, the suggested approach is to start the multi-step ice accretion process from a standard Body-Fitted (BF) mesh, thus keeping the original BF solution for the 1^{st} ice layer (as well as for the clean areas of the surface for the following steps). Usually, the BF mesh is updated to match the new ice geometry for each subsequent step. With our IBM, the volume mesh update is avoided and a penalization method is applied to the volume solvers (airflow and droplets trajectory) to impose the correct boundary conditions on the immersed boundary which arbitrarily cuts through the mesh. The ice surface is, however, re-meshed to retain an adequate representation of the ice shape for the IBM. The use of the penalization method requires some modifications to the ice accretion suite : the addition of a preprocessing step, the modification of the volume solvers and the extraction of surface data, as highlighted in red in Fig. 2. These modifications are discussed in the following sections along with the integration of the level-set method in the multi-step process.

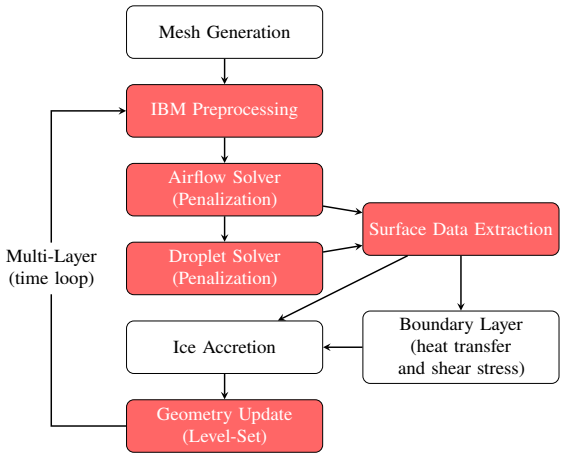


Fig. 2 Sequential call to modules in a multi-step icing simulation using an immersed boundary method.

A. Immersed Boundary Pre-Processing

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

The IB preprocessor evaluates the signed distance field (ϕ) by first detecting the inside (solid) and outside (fluid) cells. Knowing the list of edges defining a closed immersed boundary, a ray casting algorithm can be used for this matter [13]. Once this information is known, it can be used to determine the sign of the signed distance field where $\phi > 0$ in the fluid and $\phi < 0$ in the solid. The distance is evaluated by taking advantage of the available explicit definition of the interface. For each cell, a geometric approach determines the minimum projected distance to the list of edges (or faces in 3D) defining the IB. Then, the normals to the IB ($\mathbf{n}_\phi$) and its curvature (κ) can be evaluated from the signed distance

field (ϕ) as:

$$\mathbf{n}_\phi = -\frac{\nabla\phi}{||\nabla\phi||} \tag{1}$$

$$\kappa = \nabla \cdot \mathbf{n}_\phi \tag{2}$$

Notice that $\mathbf{n}_\phi$ is defined to point towards the solid, contrary to the usual definition, which is useful in the implementation of the penalization methods. An example of signed distance field around a clean NACA23012 airfoil is illustrated in Fig. 3 along with the normals to the wall ($\mathbf{n}_\phi$). Here, the signed distance field is strictly positive because it is evaluated on a BF mesh where the contour $\phi = 0$ is the surface of the airfoil.

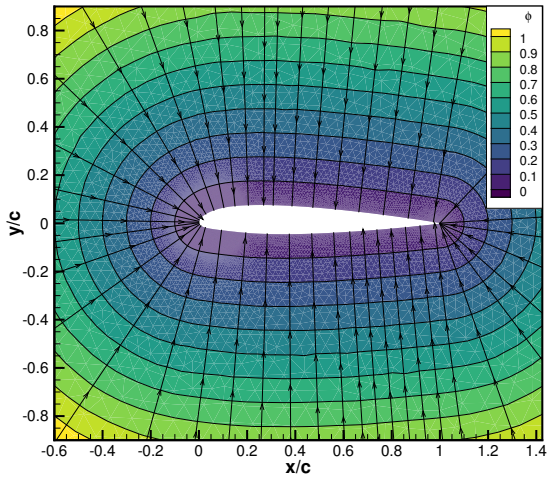


Fig. 3 Signed distance contours (ϕ) and surface normals (n_ϕ) for a clean NACA23012 airfoil

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

B. Volume Penalization Method

With a volume penalization method, the boundary conditions are applied by the addition of source terms in the continuous form of the governing equations in order to enforce the desired condition at the immersed boundary. The source terms are turned on if the computational volume (a cell) is located inside the solid zone and turned off if in the fluid zone. Hence the governing equations are solved as usual in the fluid but penalized in the solid. The source terms are turned on and off using a mask function (χ) which takes the form of a sharp Heaviside function (Eq. (3)).

$$\chi = \begin{cases} 0 & \phi \geq 0 \text{ (fluid)} \\ 1 & \phi < 0 \text{ (solid)} \end{cases} \tag{3}$$

For the aerodynamics, the penalization of the Euler equations is performed using the CBVP-Hs method of [9]. This approach enforces the no-penetration velocity (slip wall, Eq. (4)) and uses the normal momentum relation to account for the wall curvature in the pressure extrapolation (Eq. (5)). The conservation of total enthalpy (Eq. (6)) and entropy

(Eq. (7)) are also enforced across the immersed boundary to close the system.

$$(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi = 0 \tag{4}$$

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}P = \kappa\rho||\mathbf{v}||^2 \tag{5}$$

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}H = 0 \tag{6}$$

$$\mathbf{n}_\phi\cdot\boldsymbol{\nabla}s = 0 \tag{7}$$

This method was found to perform well for ice shapes exhibiting high curvature such as ice horns. The set of penalized Euler equations is given by Eq. (8) where the penalization terms enforcing Eqs. (4)–(7) are gathered on the right-hand side (RHS).

$$\begin{aligned}
\frac{\partial\rho}{\partial t} + (1-\chi)\,\boldsymbol{\nabla}\cdot(\rho\mathbf{v}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}\rho - \kappa\frac{\rho^2}{\gamma P}||\mathbf{v}||^2\right)\\
\frac{\partial(\rho\mathbf{v})}{\partial t} + (1-\chi)\,\boldsymbol{\nabla}\cdot(\rho\mathbf{v}\otimes\mathbf{v} + P\mathbf{I}) &= -\frac{\chi}{\eta_c}\left(\mathbf{n}_\phi\cdot\boldsymbol{\nabla}(\rho\mathbf{v}) + \kappa\rho\mathbf{v}\left(1-\frac{\rho}{\gamma P}||\mathbf{v}||^2\right)\right) - \frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)\mathbf{n}_\phi\\
\frac{\partial(\rho E)}{\partial t} + (1-\chi)\,\boldsymbol{\nabla}\cdot((\rho E+P)\mathbf{v}) &= -\frac{\chi}{\eta_c}\rho\mathbf{n}_\phi\cdot\boldsymbol{\nabla}H - \frac{\chi}{\eta}\rho(\mathbf{v}\cdot\mathbf{n}_\phi)^2
\end{aligned} \tag{8}$$

In Eq. (8), $\mathbf{v}$ is the air velocity, $\mathbf{I}$ is the identity tensor, E is the total energy, η and $1/\eta_c$ are penalization parameters that can be respectively interpreted as a characteristic time and a characteristic velocity.

For the Eulerian droplet equations, the penalization method of [14] is used. When droplets impinge the body ($\mathbf{v}_d\cdot\mathbf{n}_\phi > 0$), no penalization is applied and the physical equations are solved in the solid. The droplets are thus allowed to cross the immersed boundary and enter the body. However, when the droplets enter the computational domain from the solid ($\mathbf{v}_d\cdot\mathbf{n}_\phi \leq 0$) a boundary condition is applied on the primitive variables (Eq. (9)), enforcing a null flux and avoiding re-injection of the droplets.

$$\left.\begin{array}{ll}\alpha & = 0\\ \mathbf{v}_d & = 0\end{array}\right\} if \quad \mathbf{v}_d\cdot\mathbf{n}_\phi \leq 0 \tag{9}$$

To translate this behavior to the droplet equations using penalization terms, the usual mask function (χ, Eq. (3)) is multiplied by a droplet mask function (χ_d, Eq. (10)), ensuring the penalization term is only active in the solid if the droplets are reinjected in the fluid.

$$\chi_d = \begin{cases} 0 & \alpha\mathbf{v}_d\cdot\mathbf{n}_\phi \geq 0 \text{ (impingement)}\\ 1 & \alpha\mathbf{v}_d\cdot\mathbf{n}_\phi < 0 \text{ (re-injection)}\end{cases} \tag{10}$$

The set of penalized droplet equations is given by Eq. (11), where the influence of gravity is neglected and the penalization terms are highlighted in red.

$$\begin{aligned}
\frac{\partial\alpha}{\partial t} + \boldsymbol{\nabla}\cdot(\alpha\mathbf{v}_d) &= -\frac{\chi\chi_d}{\eta}\alpha\\
\frac{\partial(\alpha\mathbf{v}_d)}{\partial t} + \boldsymbol{\nabla}\cdot(\alpha\mathbf{v}_d\otimes\mathbf{v}_d) &= \frac{C_D Re_d}{24Stk}\alpha(\mathbf{v}_a-\mathbf{v}_d) - 2\frac{\chi\chi_d}{\eta}\alpha\mathbf{v}_d
\end{aligned} \tag{11}$$

In Eq. (11), α is the non-dimensional volume fraction of water, $\mathbf{v}_d$ is the non-dimensional droplets velocity, $\mathbf{v}_a$ is the non-dimensional air velocity and C_D is the droplets drag coefficient. The droplets Reynolds number (Re_d) and the Stokes number (Stk) are:

$$Re_d = \frac{\rho_a||\mathbf{v}_a-\mathbf{v}_d||D_d}{\mu} \tag{12}$$

$$Stk = \frac{\rho_d D_d^2 U_\infty}{18L\mu} \tag{13}$$

where D_d is the droplet diameter, μ the dynamic viscosity of air and L a characteristic dimension (e.g., the chord length for an airfoil). The drag model of Schiller and Naumann [15] is used for the droplets which are assumed to remain spherical:

$$C_D = \begin{cases} \frac{24}{Re_d}(1+0.15Re_d^{0.687}) & Re_d \leq 1000\\ 0.4 & Re_d > 1000\end{cases} \tag{14}$$

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

C. Surface Data Extraction

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

In this paper, the data is interpolated on the IB using a weighted least square approach. The nearest cell to the interpolation point is first detected. Then, all the cells sharing a node with the identified cell are flagged as neighbors and used for the interpolation. The penalization methods used in this paper are designed to fill the solid cell with valid data. The solid cells are included in the interpolation stencil, hence the need for methods ensuring a controlled continuity of the solution across the solid-fluid interface, as described in [9, 14]. The interpolation uses an inverse distance weight with a smoothing parameter to avoid dividing by zero when the interpolation point and stencil points are too close. The weight between a cell center J (part of the stencil) and the interpolation point P is evaluated as:

$$w_J = \frac{1}{\sqrt{||\mathbf{r}_{PJ}||^2 + \epsilon^2}} \tag{15}$$

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

D. Geometry Update via the Level-Set method

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

$$\mathbf{x}_{new} = \mathbf{x}_{old} + h_{ice}\mathbf{n} \tag{16}$$

where $\mathbf{x}_{new}$ and $\mathbf{x}_{old}$ are respectively the new and old node locations. This type of approach does not naturally handle the overlaps that can occur near concave region and requires methods for collision detection and front merging to obtain a usable surface mesh. A simple fix can be implemented in 2D as described in [16]. However, it does not directly translate to a 3D implementation which involves more complex geometric operations for a 2D surface mesh. Alternatively, the level-set method can be used to update the geometry. This was done for instance by [6] where the level-set equation (Eq. (17), [17]) is used with an icing velocity field ($\mathbf{V}_{ice}$) and solved on the volume mesh.

$$\frac{\partial \phi}{\partial t} + \mathbf{V}_{ice} \cdot \nabla \phi = 0 \tag{17}$$

This approach has the benefit of being valid for both 2D and 3D simulations. It also naturally handles the issues related to the geometry update such as geometry overlaps. Here, the level-set method reuses the signed distance field (ϕ) computed at the IB pre-processing step. The interface (IB or BF) is represented by the contour $\phi = 0$ and is advanced in time (Eq. (17)) to generate the ice shape, following the icing velocity vector field $\mathbf{V}_{ice}$. In this paper, the level-set is discretized using a 2^{nd} order scheme in time (Heun's method) and space (upwind with MUSCL extrapolation). The following sections describe a method to retrieve the icing velocity field and discuss the need for a re-initialization step in the advection of the level-set.

1. Velocity Propagation

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

$$V_{ice,surf} = \frac{h_{ice}}{\Delta t_{ice}} \tag{18}$$

However, $V_{ice,surf}$ must be propagated in the volume mesh in order to perform the level-set advection (Eq. (17)). To obtain a behavior similar to the Lagrangian node displacement approach (Eq. (16)), the icing velocity is propagated from the surface mesh in the normal direction, producing constant velocity bands. A PDE-based approach (Eq. (19)) is used to propagate the information from the surface ($V_{ice,surf}$) to the field (V_{ice}) following the normal direction to the surface.

$$\frac{\partial V_{ice}}{\partial t} = \text{sign}(\phi)\mathbf{n}_\phi \cdot \nabla V_{ice} \tag{19}$$

When the surface mesh corresponds to the Body-Fitted mesh boundary, $V_{ice,surf}$ is imposed as a Dirichlet boundary condition using ghost cells and is propagated in the fluid zone to obtain (V_{ice}). When the ice shape is immersed in the mesh (IB), the surface no longer corresponds to the mesh boundaries. For instance, this occurs from the 2^{nd} ice layer onward in the multi-step icing process. In this situation, a band of cells in the vicinity of the interface is initialized by a nearest neighbor search, taking advantage of the explicit definition of the interface. These cells are then frozen (no update) so they can act as ghost cells when solving Eq. (19) on both sides of the IB. The update is prevented by setting the Right Hand Side (RHS) of Eq. (19) to zero for the frozen cells. The propagation Eq. (19) accounts for the sign of ϕ in order to propagate $V_{ice,surf}$ from the band of initialized cells towards the fluid ($\phi > 0$) and solid zones ($\phi < 0$).

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

$$\mathbf{V}_{ice} = -V_{ice}\mathbf{n}_{\phi,0} \tag{20}$$

where $\mathbf{n}_{\phi,0}$ represents the normal to the initial contour $\phi = 0$ (before the advection process begins). In other words, the icing velocity field remains fixed during the advection of the level-set. An example of propagated icing velocity field is illustrated in Fig. 4, showing the constant velocity bands in the normal direction to the interface.

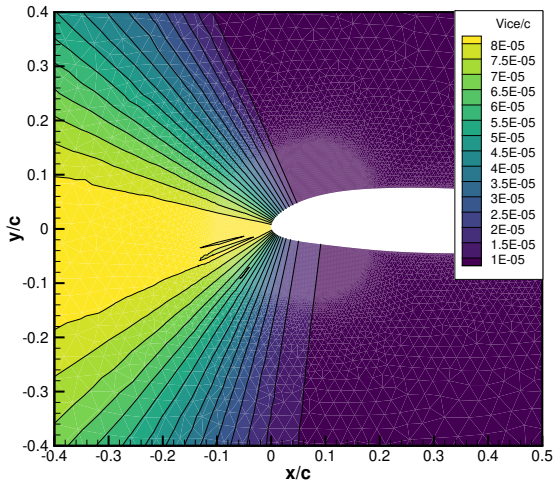


Fig. 4 Example of a propagated icing velocity field for a clean NACA23012 airfoil, body-fitted surface. Coordinates and velocity non-dimensionalized by the chord (c).

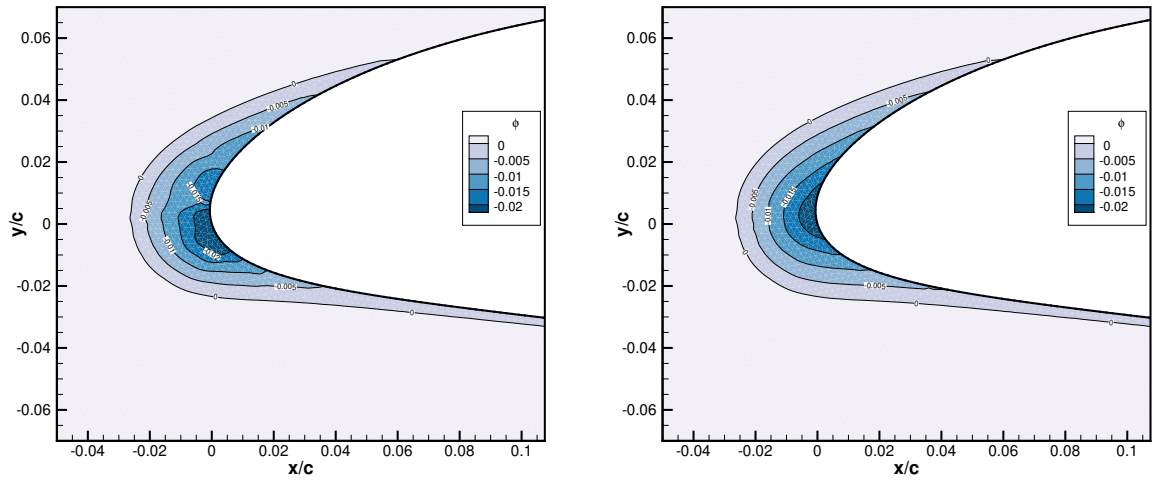
2. Level-Set Advection and Re-Initialization

While the contour $\phi = 0$ is advected using the level-set equation (17), ϕ does not conserve the properties of a signed distance field [18]. A re-initialization of the level-set is thus performed (i.e., the signed distance field is re-evaluated). This could be done by reusing the geometric approach from the pre-processing step. Because the new location of the IB is only known via its implicit definition at this stage, this would imply the application of a contour extraction technique to obtain an explicit definition of the interface (new surface mesh). The signed distance field is instead updated using the re-initialization equation [17], as follows:

$$\frac{\partial \phi}{\partial t} = \mathcal{S}(\phi_0)\left(\mathbf{n}_\phi \cdot \nabla\phi + 1\right) \tag{21}$$

$$\mathcal{S}(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \epsilon^2}} \tag{22}$$

This equation incorporates a smoothed sign function $\mathcal{S}(\phi_0)$ which is based on the signed distance before re-initialization (ϕ_0). According to [18], it ensures that ϕ remains unchanged at the interface during the re-initialization process. In practice, numerical experiments showed the introduction of wiggles in the contour $\phi = 0$ when using this approach, an undesirable behavior as a surface mesh is to be constructed from this extracted interface. To ensure the interface remains exactly at the same location, the idea used for the velocity propagation is repurposed here: freezing the update of a band of cells in the vicinity of the interface. Again, it is done by setting the RHS to zero in Eq. (21) for the frozen cells. This approach follows the assumptions that ϕ remains close to a signed distance field in the vicinity of the interface. In this paper, two iterations of Eq. (21) are performed at every time step of the level-set advection process (Eq. (17)). An example of level-set advection is shown on Fig. 5, where the ϕ contours are displayed inside the ice shape only. Without re-initialization (Fig. 5a), the signed distance field is distorted inside the solid while activating the re-initialization (Fig. 5b) provides a more regular and sensible solution.



(a) Without re-initialization **(b) With re-initialization**

Fig. 5 Example of level-set advection from a clean NACA23012 airfoil, single step ice accretion on a BF mesh

E. Surface Mesh Extraction

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

A surface discretization can be obtained by performing the extraction of the contour $\phi = 0$. The contour extraction is performed after the re-initialization to avoid possible issues in using a non re-initialized and distorted signed distance field (Fig. 5a). In this paper, 2D unstructured meshes made of triangles are used which allows for a simple contour extraction method. An edge-based interpolation is performed assuming a single intersection point per edge. The process marches from cell to cell and adds consecutive intersection points to a linked list, forming a surface discretization. Tested edges are tagged along the way to avoid adding duplicates to the list. Once the marching process can no longer find any intersection on untested edges, the contour is completed. An edge is intersected by the contour if there is a sign change in ϕ between its two vertices. If ϕ at one vertex is below a specified threshold, the intersection is assumed to occur at the vertex and no interpolation is made. In this case, all the edges sharing the vertex are tagged as tested. This approach retains the discretization of the body-fitted surface where there is no ice accretion ($\phi \approx 0$) and perform a more classical contour extraction for the immersed boundary. It also directly provides an ordered list of points (surface mesh) for each body when dealing with a multi-element configuration. Note that in IGLOO2D, ϕ is reconstructed at the vertices from a weighted least square interpolation using the cell-center solution. An example of the marching process illustrated in Fig. 6 where the vertices are identified as positive, negative or zero and the extracted contour is illustrated in red.

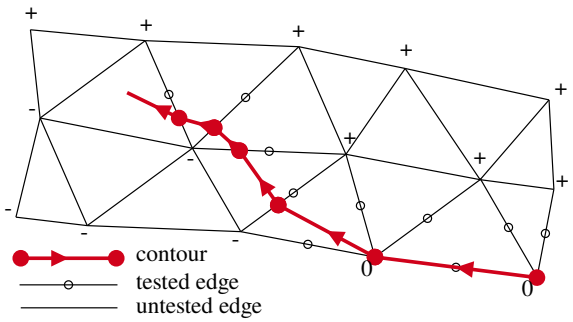
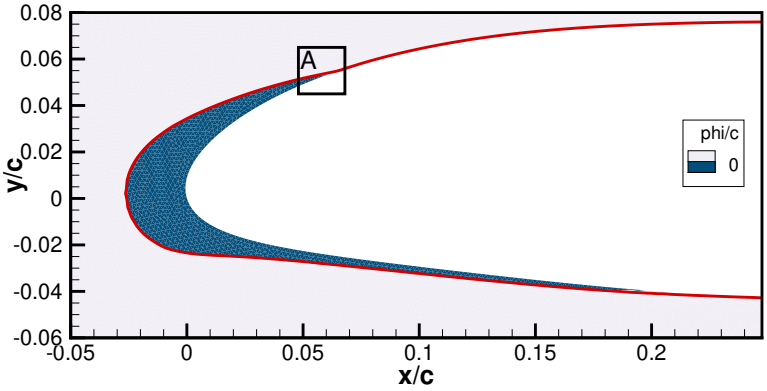
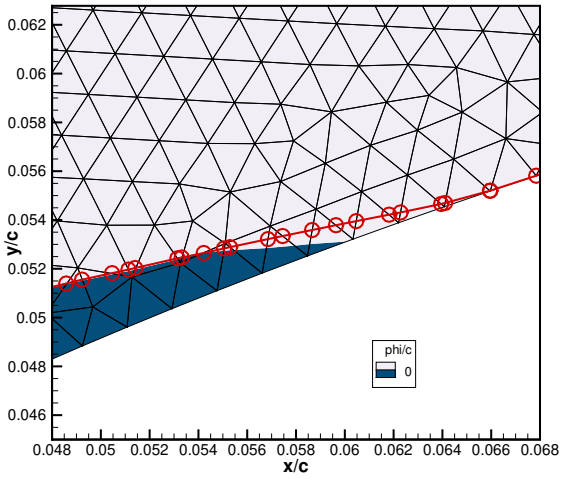


Fig. 6 Example for the contour extraction marching process

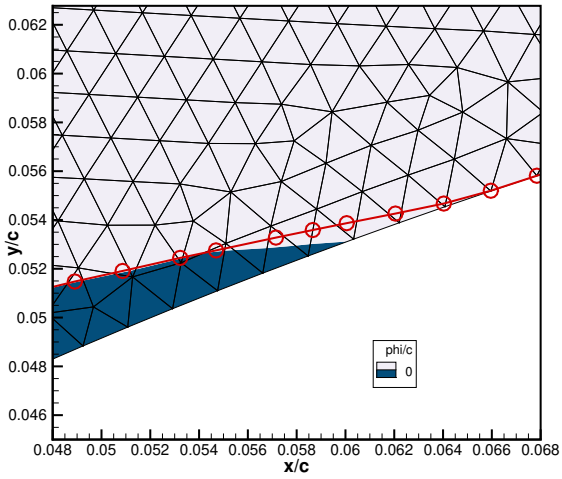
The extraction process usually produces an irregular discretization where nodes can be very close to each other when the edge intersection is detected near a vertex. To help retrieve a more uniform and smoother surface mesh, nodes are merged if they are too close and inserted if they are too far apart. An example of contour extraction is provided in Fig. 7a near the leading edge of an iced NACA23012. The effect of node merging and insertion is illustrated in Figs. 7b–7c. Note that this node correction process is not mandatory as the surface is later re-meshed using GMSH. However it was found to improve the quality of the resulting mesh.



(a) Leading edge view



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



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

Fig. 7 Example of a level-set extraction on an iced NACA23012 airfoil. red: surface extraction; blue: ice

III. Ice Accretion Results

In this section, the new ice accretion framework using the IB and level-set methods is assessed. The objective is to reproduce the ice accretion results obtained with a classical BF approach while improving the robustness of the numerical tool (e.g., no failure). In order to demonstrate the benefits of using the level-set approach, ice accretion over a manufactured ice shape is first performed using the level-set method and compared to the Lagrangian node displacement method. Then, rime ice *case 241* and glaze ice *case 242* from the 1^{st} AIAA Ice Prediction Workshop (IPW, [10]) are tested. These cases are respectively *run ED1977* and *run ED1978* from [19], with slightly corrected icing conditions. Additional ice accretion cases from [11] (cases 001, 003 and 004) are also tested to further demonstrate the behavior of the IBM. Finally, the new framework is tested on the multi-element McDonnell-Douglas LB606b Airfoil (MDA) [20] to illustrate the flexibility of the method on complex high-lift systems. The simulation parameters are summarized in Table 1.

Table 1 Simulation Parameters

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

In this section, two methods are available for the representation of the ice shape (IBM or BF) and two for the geometry update (node displacement or Level-Set). This makes four possible combinations. When simply referring to the Immersed Boundary Method, the use of the level-set method is implied. Similarly, when referring to the Body-Fitted method, the use of the Lagrangian node displacement approach is implied (the standard approach in the icing community). In addition, the calculations are carried out with IGLOO2D. The default options described in [11] were used for the Body-Fitted approach, in particular for the MESSINGER2D solver and the boundary-layer solver SIM2D. Regarding the meshes, unstructured grids generated by GMSH were systematically used. The wall mesh size is in the range of 1e-3 to 5e-3 chords (with a refinement in the range of 5e-4 chords for blunt trailing edges). These mesh sizes are fairly representative of default mesh sizes used in IGLOO2D. They generally allow obtaining a good trade-off between solution accuracy and computational time.

For all the calculations, the wall mesh size is kept constant near the leading edge and extended over 0.75 chords as shown in figure Fig. 8. This is required when using the IBM in order to avoid the re-meshing during the multi-step process while maintaining an equivalent wall cell size compared to the BF method (with re-meshing).

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

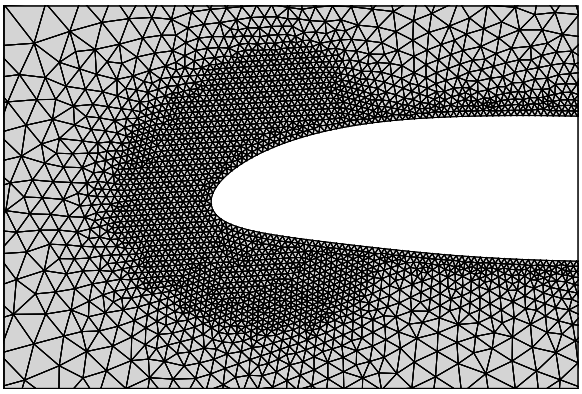


Fig. 8 Example mesh around a NACA23012 with an extended refinement zone near the leading edge

A. Manufactured Ice Shape

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

This manufactured geometry was first presented in [21] and is generated from a NACA0012 airfoil with added artificial ice near the leading edge. The three-horn configuration was selected to obtain multiple flow recirculation zones and create a difficult situation for the ice growth solver because of the presence of highly concave and convex features.

On Fig. 9, the ice accreted three-horn geometry is illustrated with the enforced ice accretion thickness. The level-set solution is represented in blue, indicating the zone with $\phi < 0$. The ice shape generated by the Lagrangian node displacement is shown as a solid black line, where the geometry overlaps can be seen near concave regions of the geometry. The contour $\phi = 0$ is extracted by our edge marching method and represented by the red line with markers. As observed in Fig. 9, the level-set method automatically handles the geometry overlaps and the extracted contour provides an explicit surface mesh discretization that can be used in the multi-step ice accretion process.

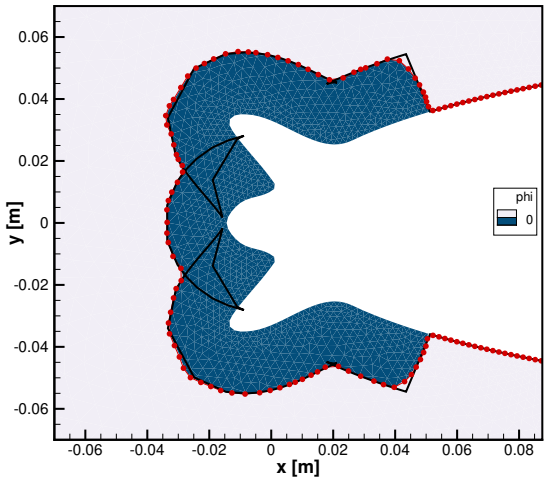
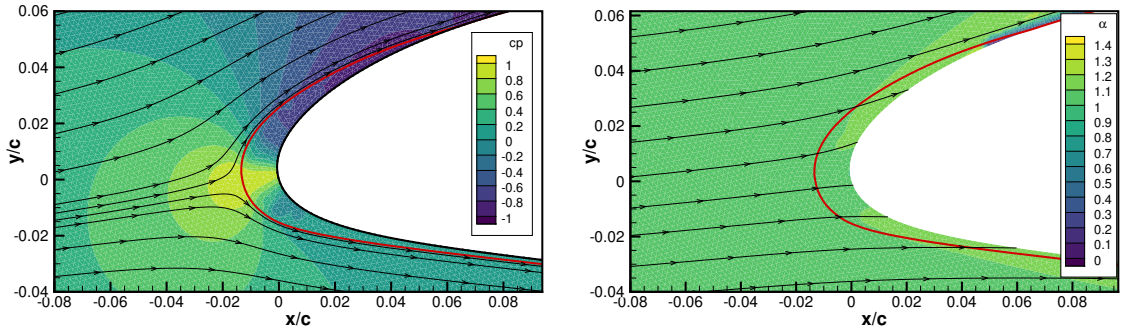


Fig. 9 Comparison between Lagrangian node displacement and Level-Set approach with contour extraction

B. Rime Ice 241

For the results presented in this paper, only the ice shape is used as an Immersed Boundary and the clean geometry is still treated using a Body-Fitted approach. As an illustration, the aerodynamic and droplet fields for the rime ice case are shown in Fig. 10 where the immersed boundary (the ice shape) is represented by the red line and the solid body is white.

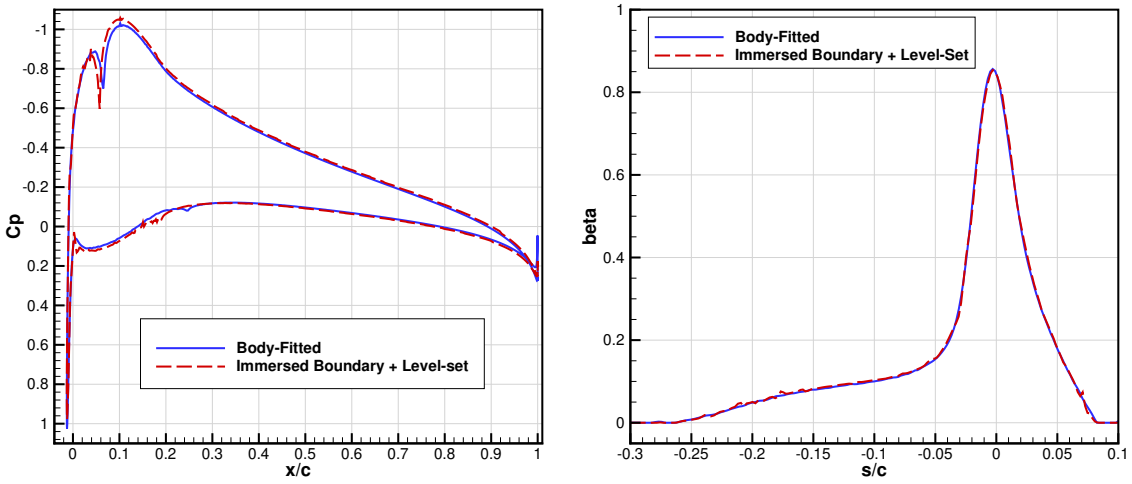


(a) Pressure coefficient **(b) Volume fraction of water (α)**

Fig. 10 Rime ice case 241: solution from the volume solvers around the 1^{st} ice layer of a two-layer simulation.

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

In Fig. 11, the pressure coefficients (Cp) and collection efficiency (β) are compared for the BF and IB methods on the 1^{st} ice layer of a two-step simulation. Ideally, the IB method should reproduce the results obtained with a BF approach. Fig. 11a illustrates a slight mismatch in Cp near the point of maximum suction. Nonetheless, the collection efficiency is very close between the two methods (Fig. 11b). As rime ice accretion is mostly governed by the collection efficiency, it generates very similar ice shapes for the BF and IB methods despite the difference in pressure coefficients (Fig. 12a). The ice shapes are also in good agreement with the experimental results. The experimental ice shape is the so-called MCCS (Maximum-Combined-Cross-Section) [22] derived by the experimentalists from the ice scans (it is more or less the envelope of the ice shape).



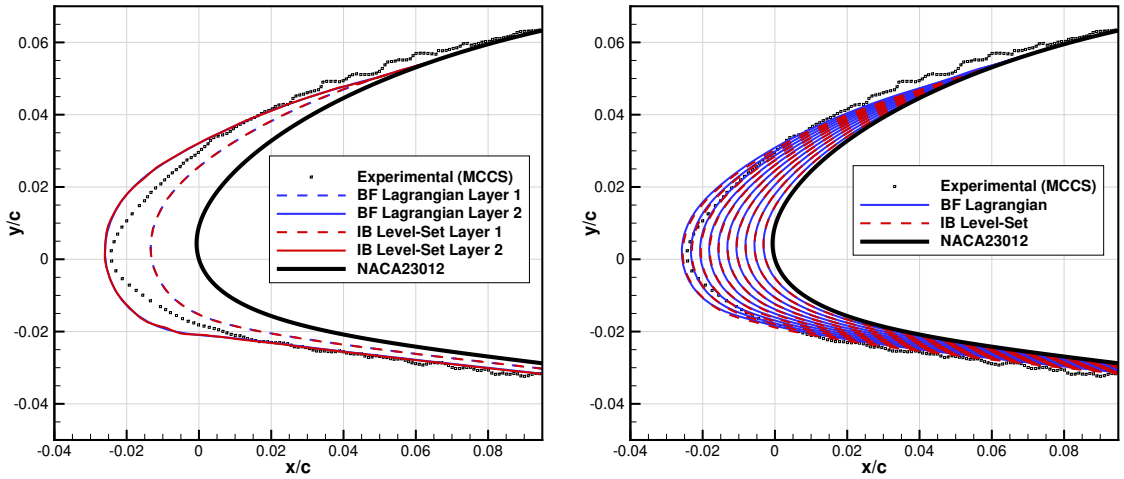
(a) Pressure coefficient (Cp) **(b) Collection efficiency (β)**

Fig. 11 Rime ice case 241 – comparison of wall surface data on the 1^{st} ice layer of a two-layer simulation for the BF and IB methods.

The discrepancy observed near $x/c = 0.06$ in Fig. 11a can be explained by the interaction between the body-fitted wall and the IB. As the penalization terms are applied only if $\phi < 0$ at the cell centers, the boundary condition is no longer applied relative to the ice shape but relative to the wall when the ice shape becomes thin near the ice accretion limits. The issue could be solved by implementing a 2^{nd} order discretization of the penalization terms or by performing

a local mesh refinement to obtain a better representation of the ice shape near the impingement limits. However, the current implementation is still able to provide a good prediction of the ice shape in comparison with the BF results (Fig. 12).

Since the simulation starts from a BF mesh, the 1^{st} step is not affected by the IBM and thus, the 1^{st} ice layer should be the same for both methods. However, a difference might be introduced by the geometry evolution solver which can use either a level-set with contour extraction or the Lagrangian node displacement approach. On Fig. 12a, a two-step ice accretion prediction is made where the 1^{st} ice layer is illustrated with a dashed line. As the generation of the 1^{st} ice layer is not influenced by the IBM, the Lagrangian and Eulerian (level-set) geometry updates can be compared, showing negligible difference. Thus, discrepancies observed in Fig. 11 for the surface data can be attributed to the IB method and not to the level-set approach.



(a) 2 layers **(b) 10 layers**

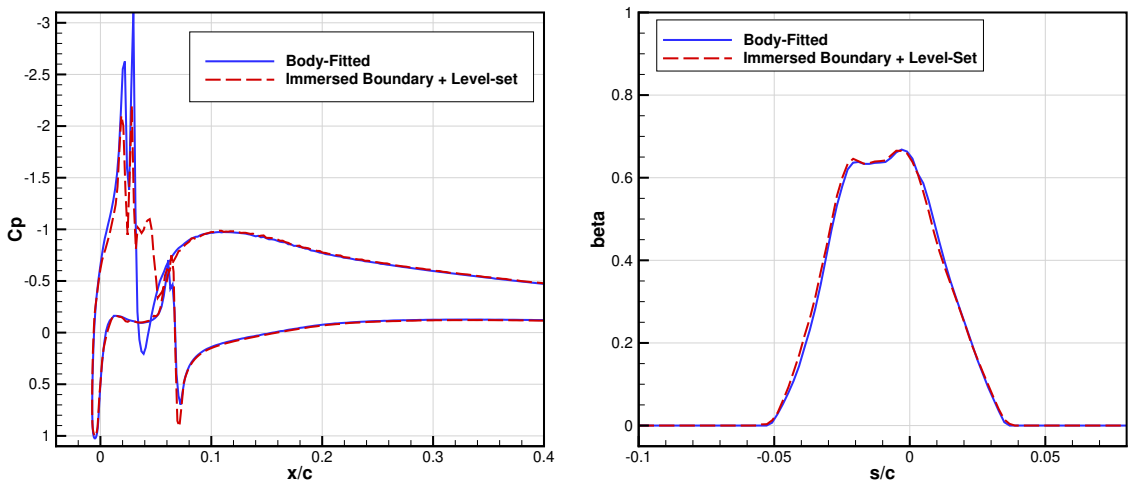
Fig. 12 Rime ice case 241 – multi-step ice shape predictions

For a 10-layer ice shape prediction (Fig. 12b), the solution is still in good agreement with the experimental data for both methods. Increasing the number of steps reduces the thickness of each ice layer and this might affect the behavior of the penalization method for the same reason described earlier (1^{st} order discretization of the penalization terms). For instance, the penalization method might effectively see the same geometry for 2 consecutive ice layers even though the ice shape has actually moved. This typically occurs if the ice layer is too thin relative to the mesh cells. For the 10-step simulation presented here, the mesh cell size is about the same as the thickness of a single layer, providing good results.

C. Glaze Ice 242

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

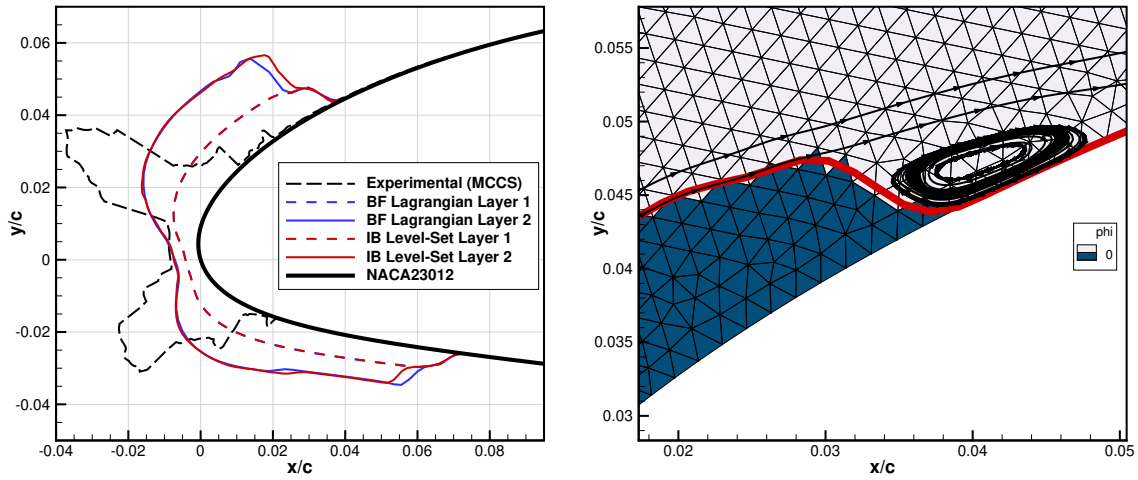
For this case, there is again a slight mismatch on the Cp distribution (Fig. 13a), but a very similar collection efficiency for both methods (Fig. 13b). As glaze ice accretion is sensitive to the heat transfer coefficient which is in turn driven by the aerodynamics, the mismatch in Cp might explain the slight difference observed on the ice shape Fig. 14a. The effect of using the Lagrangian node displacement vs. the level-set approach can again be estimated by analyzing the 1^{st} ice layer on Fig. 14a, where a negligible difference is observed. It suggests that the difference in Cp and β is due to the IBM.



(a) Pressure coefficient (Cp) **(b) Collection efficiency (β)**

Fig. 13 Glaze ice case 242 – comparison of wall surface data on the 1^{st} ice layer for the BF and IB simulations.

By observing Fig. 13a, the Cp distribution corresponds well between the BF and IB methods for $x/c < 0.02$ and $x/c > 0.06$. The zone where the discrepancy occurs is located near the ice accretion limits where the ice shape stops sharply. For the IBM, this results in a detached flow with a recirculation zone (Fig. 14b) while it is not the case for the BF method, explaining the difference. Here, the comparison is made between the two methods with equivalent mesh size. However, this result suggests that the penalization method require a finer mesh near curved features to be equivalent to the BF approach.



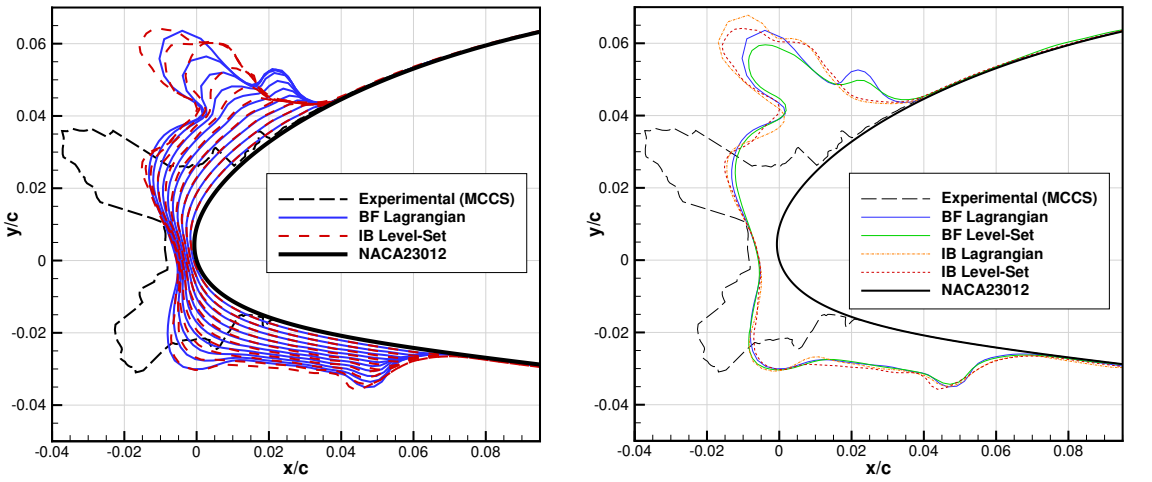
(a) Ice shape comparison **(b) Upper surface detached flow (IB Level-Set)**

Fig. 14 Glaze ice case 242 – two-step ice accretion

Although the ice shape prediction for the 2^{nd} layer is similar for both methods, it does not reproduce the experimental measurements (Fig. 14a). Note that icing experiments carry large uncertainties as well as spanwise variations [23, 24]. However, the divergence from the experimental ice shape seems too large to be attributed only to these uncertainties. Some tests were performed by refining the mesh and manually increasing the wall roughness (by a factor 2), without significant improvement. Here, the use of a droplet size distribution might help in obtaining a prediction towards the experimental ice shape. This option was, however, not tested as it is not yet available for use with our penalization

method.

By increasing the number of ice layers to 10 (Fig. 15a), the ice shape prediction is still far from the experimental results. Moreover, an ice horn is created but not in the same location. When comparing the BF and IB methods, the ice shape is similar for the most part, but with a larger difference near the ice horn (where the effect of the aerodynamics becomes more dominant on the ice accretion). The difference in ice shape is due to the combined effect of the penalization and level-set methods compared to the BF and Lagrangian approach (standard approach). In Fig. 15b, all four combination of methods are shown for the 10^{th} ice layer only. The figure illustrates that the use of the level-set method has only a limited impact while the IB methods have a larger effect on the difference in ice shape. This is similar to the observation made for the two-step ice accretion simulation, where the penalization method requires a finer mesh near curved features to be equivalent to the BF solution. Using an equivalent cell size, the IBM however provides a good approximation.



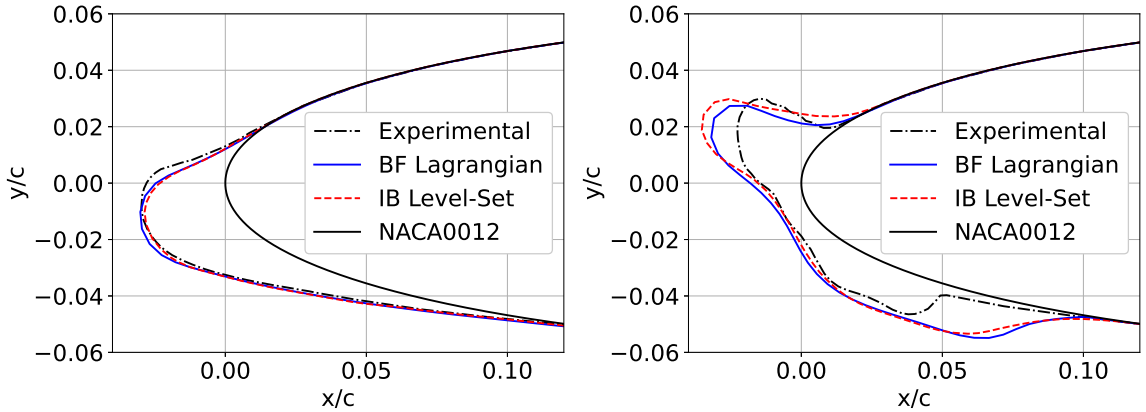
(a) Comparison over 10 layers **(b) Comparison between all the combinations, 10^{th} layer**

Fig. 15 Glaze ice case 242 – 10-step ice accretion

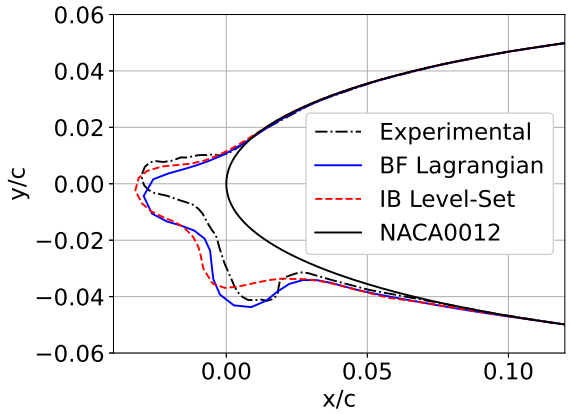
D. Additional Cases on a NACA0012

In this section, 10-step ice accretion calculations are performed on cases 001, 003 and 004 from [11] to further assess the behavior of the IBM. The calculations are performed on a coarser mesh (wall mesh size of 5e-3), but it is still representative of typical ice accretion simulations with IGLOO2D.

When comparing the ice shape prediction obtained from the IB and the BF methods, a good match is observed for the rime ice case 001 (Fig. 16a), but a larger difference is seen for the glaze ice cases 003 and 004 (Figs. 16b–16c). This is in line with the observation made in the previous sections. Glaze ice shape are more sensitive to the airflow solution and a perfect correspondence is not obtained for the wall data between the two methods (e.g., Cp distribution, Fig. 13a). The solution can be improved by refining the mesh. Nonetheless, using the IB and the level-set methods still provides a good estimation of the ice shapes when compared to the experimental data (Fig. 16), even on coarser meshes.



(a) Case 001 **(b) Case 003**



(c) Case 004

Fig. 16 Additional cases on a NACA0012 airfoil, 10-step ice accretion

E. Three-Element Airfoil (MDA)

In this section, ice accretion is performed on the three-element McDonnell-Douglas Airfoil (MDA, Fig. 17) using the icing conditions provided in [20]. This test case is selected to show the flexibility of the immersed boundary and level-set methods.

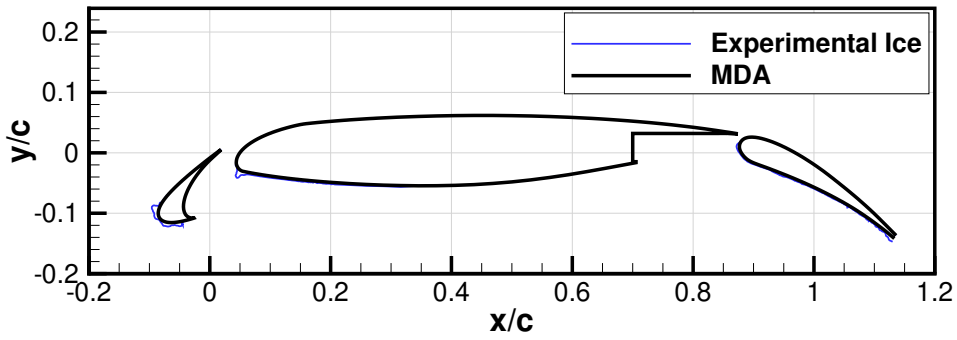


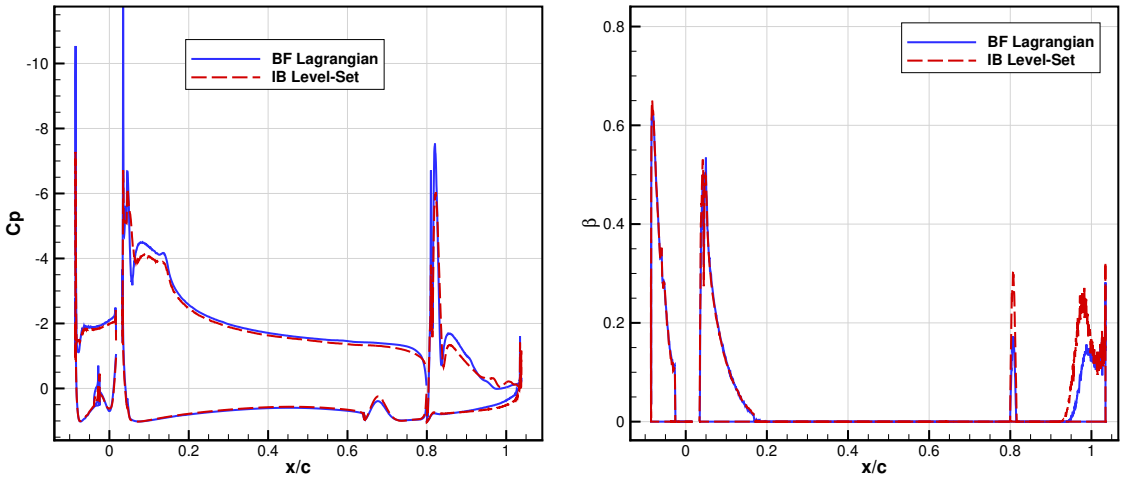
Fig. 17 Global view of the McDonnell-Douglas LB606b airfoil and its experimental ice shape

A two-step ice accretion simulation is performed with both the BF and IB methods. Here the objective is to see if

the IB method can reproduce the BF solution on this more challenging configuration. To do so, a mesh refinement was performed to obtain similar pressure coefficients and collection efficiency on the 1^{st} ice layer (2^{nd} time step), as shown in Fig. 18. The wall cell sizes of the resulting mesh are summarized in Table 2. A finer mesh is required on this test case due to the flow separation downwind of the flap. Because a Euler flow solver is used (inviscid), this flow detachment is very sensitive to the mesh size. The current mesh set-up allowed the IB and BF to behave in a similar way (e.g., similar onset of the flow detachment).

Table 2 Wall mesh characteristics for the three-element airfoil in terms of the chord ($\Delta x/c$)

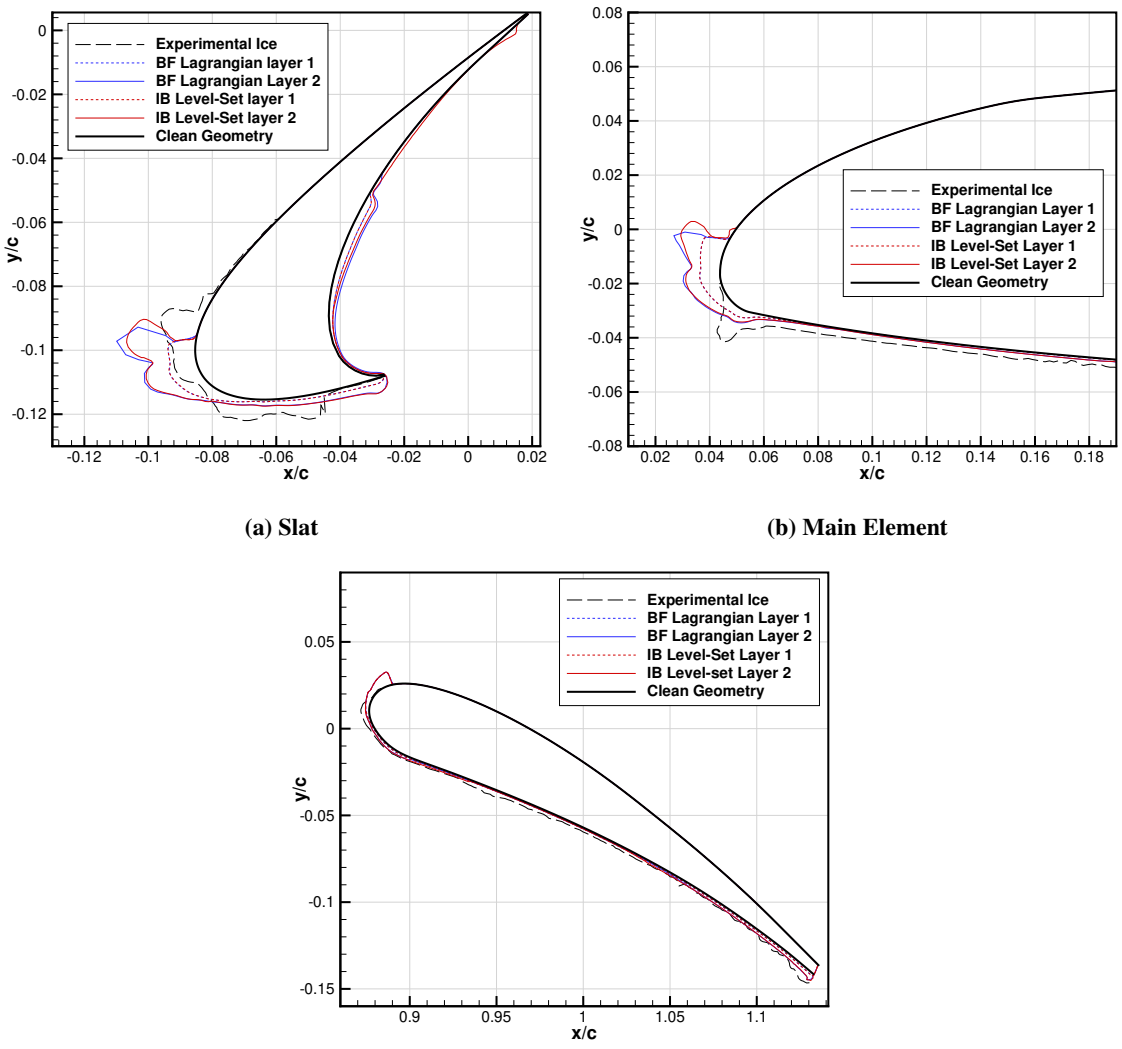
	Slat	**Main**	**Flap**
Wall	1e-3	1e-3	2.5e-4
Leading Edge	1e-3	2e-3	5e-4
Trailing Edge	1e-3	1e-3	5e-4



(a) Pressure Coefficients (Cp) **(b) Collection efficiency (β)**

Fig. 18 Wall data on the 1^{st} ice layer (2^{nd} step) for the McDonnell-Douglas multi-element airfoil (MDA LB606b)

Although a better ice shape prediction can be achieved using a RANS solver and a droplet size distribution, it was shown in [20] that a fair estimation of the ice shape can be achieved using a Euler flow solver and a single size of droplets. The ice accretion results obtained with IGLOO2D for both methods are shown for the flap, slat and main element in Fig. 19. The predicted ice shapes are not so far from the experiment for the flap and slat but are quite different from the expected solution for the main element. Perhaps a simulation involving more ice layers would improve the results. For instance, six steps are used in [20]. The use of a polydisperse droplet distribution more representative of the cloud composition would also help (which is not available in this version of the IBM code). In this paper we are concerned about reproducing the ice shapes from the BF method with the IBM and in this regard, the ice shapes (Fig. 19) are in fact similar for both methods. The comparison still exhibits the usual discrepancy due to the accuracy of the IBM for the aerodynamics.



(a) Slat **(b) Main Element**

(c) Flap

Fig. 19 Two-step ice accretion on the McDonnell-Douglas multi-element airfoil (MDA LB606b)

IV. Conclusion

This paper investigates the application of an Immersed Boundary Method (IBM) towards simulation of ice accretion within IGLOO2D. A penalization method is applied to the aerodynamics and droplet trajectories. The surface data is extracted using a weighted least square approach in order to use the boundary layer and ice accretion modules. The geometry (the ice shape) is updated using either a Lagrangian or an Eulerian (level-set) approach. A contour extraction process is also described for 2D meshes made of triangles in order to retrieve the explicit definition of the ice-air interface.

Using a manufactured test case, the level-set is shown to automatically handle geometry folding during the ice shape update while the Lagrangian approach fails at providing a useable surface discretization unless a correction is added. For ice shape predictions, a Body-Fitted mesh is used for the clean geometry and only the ice shape is treated as an IB. Following this approach, rime and glaze ice cases from the Ice Prediction Workshop are performed using up to 10 ice layers. The IBM predicted an ice shape equivalent to the body-fitted approach on the rime ice case. For the glaze ice case, the predicted ice shape is close to the body-fitted solution but exhibits a larger difference where ice accretion is most dependent on the aerodynamics (e.g., near ice horns). The difference is mostly attributed to the accuracy of the IBM and not to the use of the level-set. Additional rime and glaze ice cases on a NACA0012 showed that the current

approach (IBM + level-set) provides a fair estimation of the ice shape when compared to both the BF method and the experimental results, even on coarser meshes. Moreover, a 2-step ice shape prediction on the McDonnell-Douglas multi-element airfoil showed that with proper mesh refinement, the IBM combined with the level-set method can reproduce the BF solution on a more challenging configuration.

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

Acknowledgments

This work is financed by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chair program and the ONERA. The authors would like to thank Francesco Petrosino for kindly providing the MDA geometry and the experimental ice accretion data.

References

[1] Capizzano, F., "A Compressible Flow Simulation System Based on Cartesian Grids with Anisotropic Refinements," *Aerospace Sciences Meetings*, American Institute of Aeronautics and Astronautics, 2007, pp. –. https://doi.org/10.2514/6.2007-1450.

[2] Capizzano, F., "Turbulent Wall Model for Immersed Boundary Methods," *AIAA Journal*, Vol. 49, No. 11, 2011, pp. 2367–2381. https://doi.org/http://dx.doi.org/10.2514/1.J050466.

[3] Capizzano, F., "Coupling a Wall Diffusion Model with an Immersed Boundary Technique," *AIAA Journal*, Vol. 54, No. 2, 2015, pp. 728–734. https://doi.org/10.2514/1.J054197.

[4] Capizzano, F., and Iuliano, E., "A Eulerian Method for Water Droplet Impingement by Means of an Immersed Boundary Technique," *Journal of Fluids Engineering*, Vol. 136, No. 4, 2014, pp. 040906–040906–8. https://doi.org/10.1115/1.4025867.

[5] Al-kebsi, A., Mose, R., and Hoarau, Y., "Multi-Step Ice Accretion Simulation Using the Level-Set Method," *SAE International Conference on Icing of Aircraft, Engines, and Structure*, Minneapolis, United States, 2019. https://doi.org/10.4271/2019-01-1955, URL https://hal.archives-ouvertes.fr/hal-02562783.

[6] Pena, D., Hoarau, Y., and Laurendeau, E., "A single step ice accretion model using Level-Set method," *Journal of Fluids and Structures*, Vol. 65, 2016, pp. 278–294. https://doi.org/https://doi.org/10.1016/j.jfluidstructs.2016.06.001.

[7] Bourgault-Côté, S., Docampo-Sánchez, J., and Laurendeau, E., "Multilayer Airfoil Ice Accretion Simulations Using a Level-Set Method with B-Spline Representation," *AIAA Journal*, Vol. 57, No. 8, 2019, pp. 3299–3308. https://doi.org/10.2514/1.J057905.

[8] Lavoie, P., Blanchard, G., Radenac, E., Laurendeau, E., and Villedieu, P., "A Penalization Method for 2D Ice Accretion Simulations," *SAE Technical Paper*, SAE International, 2019. https://doi.org/10.4271/2019-01-1939.

[9] Lavoie, P., Radenac, E., Blanchard, G., Éric Laurendeau, and Villedieu, P., "An Improved Characteristic Based Volume Penalization Method for the Euler Equations Towards Icing Applications," *Computers & Fluids*, 2021, p. 104917. https://doi.org/https://doi.org/10.1016/j.compfluid.2021.104917.

[10] Broeren, A., "1st AIAA Ice Prediction Workshop," Website, Jan. 2021. URL http://folk.ntnu.no/richahan/IPW/.

[11] Trontin, P., Blanchard, G., Kontogiannis, A., and Villedieu, P., "Description and assessment of the new ONERA 2D icing suite IGLOO2D," *9th AIAA Atmospheric and Space Environments Conference*, 2017. https://doi.org/10.2514/6.2017-3417.

[12] Geuzaine, C., and Remacle, J.-F., "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, 2009, pp. 1309–1331. https://doi.org/10.1002/nme.2579.

[13] Schneider, P. J., and Eberly, D. H., "Chapter 13 - computational geometry topics," *Geometric Tools for Computer Graphics*, edited by D. H. SCHNEIDER, PHILIP J.and EBERLY, The Morgan Kaufmann Series in Computer Graphics, Morgan Kaufmann, San Francisco, 2003, pp. 673 – 825. https://doi.org/https://doi.org/10.1016/B978-155860594-7/50016-3.

[14] Lavoie, P., Radenac, E., Blanchard, G., Laurendeau, E., and Villedieu, P., "A Penalization Method for Eulerian Droplet Impingement Simulations towards Icing Applications," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, 2021. https://doi.org/https://doi.org/10.2514/6.2021-1442, AIAA-2021-1442.

[15] Schiller, L., and Naumann, Z., "A drag coefficient correlation," *Zeitschrift des Vereins Deutscher Ingenieure*, Vol. 77, 1935, pp. 318–323.

[16] Ruff, G. A., and Berkowitz, B. M., "Users Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)," CR 185129, NASA, 1990. URL https://ntrs.nasa.gov/citations/19900011627.

[17] Osher, S., and Fedkiw, R., *Level Set Methods and Dynamic Implicit Surfaces*, 1st ed., Applied Mathematical Sciences, Vol. 153, Springer-Verlag New York, 2003. https://doi.org/10.1007/b98879.

[18] Sussman, M., Smereka, P., and Osher, S., "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," *Journal of Computational Physics*, Vol. 114, No. 1, 1994, pp. 146 – 159. https://doi.org/http://dx.doi.org/10.1006/jcph.1994.1155.

[19] Lee, S., Broeren, A. P., Kreeger, R. E., Potapczuk, M. G., and Utt, L., "Implementation and Validation of 3-D Ice Accretion Measurement Methodology," *6th AIAA Atmospheric and Space Environments Conference*, 2014. https://doi.org/10.2514/6.2014-2613.

[20] Petrosino, F., Mingione, G., Carozza, A., Gilardoni, T., and D'Agostini, G., "Ice Accretion Model on Multi-Element Airfoil," *Journal of Aircraft*, Vol. 48, No. 6, 2011, pp. 1913–1920. https://doi.org/10.2514/1.C031346.

[21] Lavoie, P., Dorian, P., Yannick, H., and Laurendeau, E., "Comparison of thermodynamic models for ice accretion on airfoils," *International Journal of Numerical Methods for Heat & Fluid Flow*, Vol. 28, No. 5, 2018, pp. 1004–1030. https://doi.org/10.1108/HFF-08-2016-0297.

[22] Broeren, A. P., Potapczuk, M. G., Lee, S., Malone, A. M., Paul, B. P., and Woodard, B., "Ice-Accretion Test Results for Three Large-Scale Swept-Wing Models in the NASA Icing Research Tunnel," *8th AIAA Atmospheric and Space Environments Conference*, 2016. https://doi.org/10.2514/6.2016-3733.

[23] Ruff, G., and Anderson, D., "Quantification of ice accretions for icing scaling evaluations," *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998. https://doi.org/10.2514/6.1998-195.

[24] Wright, W. B., "Validation Methods and Results for a Two-Dimensional Ice Accretion Code," *Journal of Aircraft*, Vol. 36, No. 5, 1999, pp. 827–835. https://doi.org/10.2514/2.2516.