Multi-Step Level-Set Ice Accretion Simulation with the NSMB solver

A. AL-KEBSI[a], D. PENA[a], E. LAURENDEAU[b], R. MOSE[a,c],Y. HOARAU[a]

a. Université de Strasbourg, Laboratoire ICube, 2 Rue Boussingault 67100 Strasbourg, France
aalkebsi@unistra.fr, dorian.pena@gmail.com, mose@unistra.fr, hoarau@unistra.fr
b. Polytechnique Montréal, C.P. 6079, Centre-Ville, Montréal (QC), H3C 3A7, Canada
eric.laurendeau@polymtl.ca
c. École Nationale du Génie de l'Eau et de l'Environnement de Strasbourg (ENGEES)

Résumé : (16 gras)

Ice accretion on aircraft structures reduce flight safety. Experimental set-ups to design de-icing systems are costly and complex. Various ice simulation codes existing today are mostly a single shot, where the ice layer is assumed to form without affecting external flow. A more realistic view is where the deformed ice shape affects the external flow. Few multi-step icing codes exist but require manual remeshing techniques. In this study we propose a method that would allow for a full transient solution where deformation does not require remeshing. Namely, the Level-Set method is developed within the icing model to trace the deformed geometry, and to serve as an implicit reference to the solid boundary.

Abstract :

The Level-Set (LS) framework is introduced to model the ice accretion on aircraft structures. Widely used two-dimensional and 3D icing models fail to preserve and reply to current needs. The level-set disposition permitted a fully multiple layer ice accretion model : multi-step ice accretion simulation. Wherein, the solid boundary is treated by a penalization method or an immersed boundary method (IBM)-LS. In this fashion, the solid body is defined through its level-set function. Consequently, the air flow is computed by the penalized RANS-equations with wall laws. And accordingly, droplet distribution and impingement rate are computed by an IBM-LS Eulerian approach, in which droplets are allowed to impinge on a layer of cells defined by the LS function using an IBM formulation.

Mots clefs : Multi-Step Icing Simulation, Level-Set, IBM

Introduction (16 gras)

Icing effects can reduce the flight safety under certain weather conditions. According to the US National Transport Safety Board, icing is one of the major causes of flight accidents ([1]). Supercooled water droplets present in clouds impinge on the surface of aircraft structures. They either solidify totally on impact or partially then create a thin liquid film runback depending on the flow temperature and speed

hence, creating dry rime ice or glaze wet ice respectively. Low speeds and low temperatures cause dry rime ice formations, while higher flow speeds and temperatures lead to glaze wet ice formations. Typical forms of rime and glaze icing are shown in Fig. 1. Consequently, these modified geometries induce large aerodynamic degradations. Designing an adequate de-icing mechanisms requires full knowledge of the icing phenomenon itself. Icing experimental study cannot exceed the scope of a handful of simple cases due to complexity and cost. Consequently the use of computational fluid dynamics is strongly justified.

Ice accretion simulation codes used today by aerospace industry assume the icing process to be broken up into four steps : single phase air flows around the wing transporting water suspended droplets ; droplets impinge into the surface generating a liquid or dry film exchanging energy with the surface accreted to shape the final form during a certain exposure time. Air flow is usually based on 2D inviscid panel methods coupled with an empirical boundary layer method for heat transfer calculations [2]. Following that, the droplets trajectories are traced using a Lagrangian formulation. Then, the so-called Messinger model is used to evaluate the energy exchange in the liquid film [3]. Having calculated the ice thickness, the new geometry is obtained. However, this process is usually assumed to occur on a single step (single shot) considering that the time scale of the icing process is very long compared with that of the air flow. However this one-way coupling process can be repeated for portions of the required exposure time [4] but still with decoupled time scales.

The current tendency to improve the model is to use Reynolds-Averaged Navier-Stokes (RANS) equations for the air flow such as in NASA's LEWICE3D [5], FENSAP-ICE of McGill University [6], CANICE2D-NS developed at École Polytechnique Montréal [4], and ONERA's ONICE3D [7]. To get the most of such a model based on the RANS solvers, the droplet trajectories can be modelled using an Eulerian formulation as introduced by ([8]). This is the case of several icing codes ([9, 10, 11, 7, 12, 13, 14]).

To tackle full 3D configurations, traditional 1D Messinger model is being gradually replaced by a PDE formulation in [15] and [16]. Many other sub-models for the runback water film exist but are out of the scope of our current study. One of the most important factors affecting he ice accretion is the heat transfer. In FENSAP-ICE and ONICE3D it is determined with the RANS flow solver instead of the classical empirical boundary layer method.

To generate the final geometry however, mesh deformation methods have been employed for multi-step icing calculations as in [16] and [7]. Such techniques are usually easy to implement and were ergo favoured. However, ice formation produces stretched cells and badly intersected cells which require specific treatments. These problems are intensified for 3D configurations for which a time-consuming manual remeshing is usually needed. Therefore, inefficiency of these mesh deformation algorithms requires the introduction of a different approach. Such approach must maintain a good quality of the grid especially in the boundary layer region.

The current work is presented as part of an effort to develop a new approach in ice accretion simulation. This new approach should overcome some of the limitations, and most importantly, should inspire and propose Quasi-non-Steady ice accretion abilities. One promising candidate is the Level-Set function which was first introduced by [17] and [18]. In a Level-Set disposition the interface is defined implicitly via a passive scalar function. This function is set to zero at the interface, positive outside and negative inside. The Level-Set have been used by [19] to evaluate the ice shedding trajectories. In this study however, the Level zero represent the evolution of the ice/air interface. Also, The negative zone would represents the accreted ice and one can solve a heat transfer equation in this subdomain. Likewise, the

positive side represents the external flow zone. By respecting a very fine mesh around the interface a high quality flow is obtained. The mesh in question does not need be a body fitted mesh and distance normal to the wall can be arbitrary. Such embedded-grid method maintains the same Eulerian formulation of PDEs, overcoming meshing issues. T embedded-grids can be performed using structured grids like in [20, 21, 22]. Wall boundary conditions can be treated by a penalization method, in which they are treated as impermeable media in the so called Brinckman-Navier-Stokes equations. It can be also treated in an IBM framework, wherein field values based on the boundary conditions are imposed on a layer of cells.

In previous papers [23, 24], the four modules simulating the ice accretion were developed and validated in the NSMB (Navier-Stokes-Multi-Block) flow solver. These four modules are still being optimised. The first module is the compressible Navier-Stokes solver for the air-flow. The penalization method is used at this module to reproduce the wall boundary. The droplet module is solved using total variation diminishing schemes via deferred correction and a local time stepping scheme. The wall boundary is reproduced using an IBM-LS formulation. The third module consists of a PDE system to solve the energy balance evaluating the mass of ice on the surface. The last module uses the Level-Set framework to advance the iced surface. The implementation of wall laws to model the turbulent boundary layer is still under progress.

Numerical method and governing equations

To get the most out of such model, it is developed in the NSMB solver[25]. Both compressible and incompressible RANS solvers exist within NSMB. The mesh is structured, multi-block, parallelized in MPI with grid motion techniques available. Chimera grids and Immersed boundary methods are available as well.

Methodology

A single icing step is assumed to be broken up into 4 steps :

- Single phase turbulent compressible air flows.
- Water droplets are transported then impinge into the surface.
- The liquid water film on the surface reaches a thermodynamic balance.
- The geometry is deformed by the accumulation of freezing water.

In a multi-step icing this process is assumed to repeat a number of intervals. The time scale of ice formation is much higher than the time scale of air flow. Consequently this type of one way coupling is sufficient.

Penalized compressible flow

In a Level-Set disposition the solid body is defined implicitly by the zero level. It is a known practice to set the level-set function to the signed distance function [18, 26] for its desirable properties. It was demonstrated that initializing the Level-Set function to a signed distance function increases the numerical accuracy compared with a simple Heaviside function. A penalty term in the NS equations enforces the wall boundary condition at the desired location [20, 27]. The solid body is considered a porous medium with very small permeability. The penalized compressible flow equations take the following form :

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \\ \frac{\rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \underline{\pi} + \frac{1}{\eta} \chi_s \left(\rho \mathbf{u} - \rho \mathbf{u}_s\right) \\ \\ \frac{\rho e}{\partial t} + \nabla \cdot \left((\rho e + p)\, \mathbf{u}\right) = \nabla \cdot (\underline{\pi} \mathbf{u} + \mathbf{q}) + \frac{1}{\eta} \theta_s \chi_s \rho \left(\epsilon(T) - \epsilon(T_s)\right) + \frac{1}{\eta} \chi_s \left(\rho \mathbf{u} - \rho \mathbf{u}_s\right) \cdot \mathbf{u} \end{cases} \tag{1}$$

where ρ is the density, $\mathbf{u}$ the velocity, e the specific total energy, p the pressure, q the heat flux, and $\underline{\pi}$ the stress tensor. The new terms at the right hand side represent the penalty terms where $\frac{1}{\eta}$ is a penalization parameter set to high values, and χ_s is the characteristic function of the solid. This characteristic function is a Heaviside function computed from the signed distance level-set function, and is smeared out at the interface.

Modelling of the droplet Field

As stated earlier a Eulerian approach was developed for the droplet transport. It solves a droplet velocity field and a volume fraction distribution in the complete domain. Droplets are considered as a distribution of spheres divided into a number of groups (bins) each with a mean valued diameter (LANGMUIR "D" dimensionless distribution). Under high Reynolds numbers the droplets flatten to disks. Many other assumptions are mentioned and justified in [24]. The governing equations for the conservation of mass and momentum are as follows : all field variables are non-dimensionalized

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

where α is the water volume fraction, $\mathbf{u}$ the velocity field of droplets, $\mathbf{u_a}$ the velocity of air, ρ the density of water, ρ_a the density of air, $\mathbf{g}$ the gravity vector, $K = \rho d U_\infty / 18 L \mu$ is an inertia parameter, L the reference length, and d the median diameter of the droplets. Re_d is the droplets Reynolds number and is defined based on the slip velocity between the air phase and the droplets.

$$Re_d = \frac{\rho d |\mathbf{u}_a - \mathbf{u}|}{\mu_a} \tag{3}$$

The drag coefficient of the droplets C_D is given empirically as given by [28]. Many other empirical correlation are proposed for flattened droplets. In these cases the drag coefficient is a combination and is given as, $C_D = C_{D_{sphere}} + \epsilon(C_{D_{disk}} - C_{D_{sphere}})$, where $\epsilon = 1 - (1 - 0.007 We^{0.5})^{-6}$ is a function of the Webber Number. The Webber Number measures the relative importance of the fluid's inertia against the surface tension and is given by $We = \rho |\mathbf{u_a} - \mathbf{u}|^2 MVD / \sigma_w$. Consequently, at low Re the We is low and so is the parameter ϵ and the drag is that of a sphere. The opposite is true for higher Re where the drag is that of a disk.

Droplets are allowed to impinge on the surface which resembles an outlet boundary condition, but if droplets do not approach the surface, in the shadowed region near the trailing edge it should be treated as a wall for the velocity field and the concentration. Consequently, a condition is checked to evolve the boundary condition based on the field variables. This condition is the velocity field times the surface

normal vector $\mathbf{u} \cdot \mathbf{n}$. If positive "wet zone" the boundary should allow droplets to pass and thus is Neumann type, and if negative "shadowed dry zone" no additional droplets are introduced resembling a Dirichlet boundary condition. The ability of the configuration to capture incoming droplets or the droplet impingement rate in an Eulerian representation is given by :

$$\beta = \alpha \mathbf{u}.\mathbf{n} \tag{4}$$

For a distribution of droplet diameter, the distribution is split into a number of groups each with a mean diameter and each with a weight ω_i. The transport equation is solved for each bin size separately and in the end the overall impingement rate is evaluated by the weighted average and is given by :

$$\beta = \sum_i \omega_i \beta_i \tag{5}$$

The droplet transport equation is discretized in space using finite volume method. A first order scheme, blended upwind and central differencing scheme, and a 3rd order total variation diminishing scheme are all available. A non orthogonality correction is also available to increase the performance when using poor grids. The time scheme is implicit and is found to be stable for very high courant numbers. Consequently, a local time stepping scheme is used. A wide range of linear solvers exist along with the full compatibility with PSBLAS library. The geometry and mesh are multiblock and parallelized using MPI, and chimera overset grids techniques are available.

All of the above is valid for body fitted grids. However, in an embedded-grid technique the boundary condition should be treated differently. The use of penalization method might seem a simple choice. However, keeping in mind the evolving impingement boundary condition, the penalization algorithm gets complicated. The droplets should be allowed to impinge on the surface when $\mathbf{u} \cdot \mathbf{n} > \mathbf{0}$ and thus this part of the surface should not be penalized. Whereas on the trailing edge where $\mathbf{u} \cdot \mathbf{n} < \mathbf{0}$ the momentum equation should be penalized. Penalizing the flow slows it down but mass is always conserved without the existence of a real outlet boundary where mass is free to leave. As a consequence, the droplets would get remitted into the outside flow which is not realistic. This problem can be overcome by introducing a real solid wall inside the solid geometry. In other words, one should start with a body fitted mesh with a real solid interface. Then, deformed using the level-set advection equation shown in the following sections. The inside zone will be penalized for the air flow, and free for the droplets which will flow in and impinge into the real initial solid interface as shown in Fig. 2.

However this method fails quickly for complex deformed geometries. This motivated the use of an IBM formulation to capture the droplets at the zero level interface. [29] proposed an IBM impingement technique, and showed its efficiency. We propose coupling the IBM technique with the level-set formulation. In the wet zone $\mathbf{u} \cdot \mathbf{n} > \mathbf{0}$ we impose the boundary condition on a layer of cells. One can choose the first inner layer of cells and thus a Ghost Cell IBM formulation. One can also use the first outer layer. This part is further discussed in the following sections (IBM-LS Technique).

Thermodynamic modelling of the liquid film

To evaluate the mass and energy balance of the liquid film, one need to solve the mass and energy conservation equations. The flow solver evaluates the wall shear stress τ_{wall} and the heat transfer coefficient h_c. The droplet transport module returns the impingement rate β and the droplet velocity field $\mathbf{u}$. The

PDEs used are based on the Messinger model [3], and the Shallow water equation. The mass and energy conservation equations are solved for each cell located on the wall. The model proposed by [16] is the following :

$$\begin{cases} \frac{\partial \rho_w h}{\partial t} + \nabla.(\rho_w \bar{\mathbf{u}}_f h) = \dot{m}_{imp} - \dot{m}_{evap} - \dot{m}_{ice} \\ \\ \frac{\partial \rho_w C_w h T}{\partial t} + \nabla.(\rho_w C_w \bar{\mathbf{u}}_f h T) = \dot{q}_{imp} - \dot{q}_{evp} - \dot{q}_{ice} - \dot{q}_{cnv} \end{cases} \tag{6}$$

where h is the water film height, ρ_w the water density, C_w the specific heat of water, T the surface temperature in Celsius, $\dot{m}_{evp}$ the mass rate of evaporation, $\dot{m}_{imp}$ the impinging mass rate of water droplets and $\dot{m}_{ice}$ the resultant mass rate of ice accretion. $\dot{q}_{imp}$ is the heat rate of impacting droplets, $\dot{q}_{evp}$ the evaporating heat rate and $\dot{q}_{ice}$ the heat rate of freezing water. The main contribution in the energy balance comes from the heat convection term $\dot{q}_{cnv}$.

The mass rate of impinging droplets is given by $\dot{m}_{imp} = LWC \cdot V_\infty \cdot \beta$, whereas the evaporating mas rate is given by a parametric correlation developed by [30]. The energy rates are given by :

$$\begin{cases} \dot{q}_{imp} = \dot{m}_{imp} \left(C_w T_\infty + \frac{|\mathbf{u}|^2}{2} \right) \\ \\ \dot{q}_{evp} = \dot{m}_{evp} \left(\frac{L_e + L_s}{2} \right) \\ \\ \dot{q}_{ice} = \dot{m}_{ice} \left(C_i T - L_f \right) \\ \\ \dot{q}_{cnv} = h_c (T - T_{rec}) \end{cases} \tag{7}$$

where T_{rec} is the adiabatic wall temperature, C_i the specific heat of ice, L_e the latent heat of evaporation, L_s the latent heat of sublimation and L_f the latent heat of fusion.

$\bar{\mathbf{u}}_f$ is defined as the mean velocity of the water film. And is thus given by [6] :

$$\bar{\mathbf{u}}_f(\mathbf{x}) = \frac{h}{2\mu_w} \tau_{wall}(\mathbf{x}), \tag{8}$$

where τ_{wall} is the wall shear stress, $\mathbf{x}$ the wall surfaces coordinates and μ_w the viscosity of water.

Moreover the following compatibility relations must be verified [6] :

$$\begin{cases} h \geq 0, \\ \dot{m}_{ice} \geq 0, \\ h \cdot T \geq 0, \\ \dot{m}_{ice} \cdot T \leq 0. \end{cases} \tag{9}$$

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

Eq. 6 is discretized via a finite volume method with a Roe scheme. At each node i a semi-discrete system is obtained :

$$\begin{cases} \Omega_i \left(\rho_w \frac{\partial h_i}{\partial t} - S_{h,i}\right) + \sum_{j \neq i} \phi_h^{Roe}(i,j) = 0 \\ \Omega_i \left(\rho_w C_w \frac{\partial h_i T_i}{\partial t} - S_{hT,i}\right) + \sum_{j \neq i} \phi_{hT}^{Roe}(i,j) = 0 \end{cases} \tag{10}$$

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

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

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

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

For each case, several terms disappear, and since the type of the surface is unknown a process of trial and error is used. An educated guess is chosen based on initial values, then the compatibility relations are verified. If one of the compatibility relations is not satisfied we switch to another surface type as shown in Fig. 3. The system is solved using explicit Runge-Kutta scheme and is parallelized with MPI.

Level-Set function

The Level-Set approach was developed in [24] to track the ice air interface evolution. The interface is represented implicitly by the zero level of the Level-Set function Φ. The Level-Set equation is set as the signed distance function [18]. A signed distance function is given by : $|\Phi(\overrightarrow{x})| = d(\overrightarrow{x})$. It is initialized in the computational domain as follows :

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

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

The evolution of the interface is achieved by solving the so-called Level-Set advection equation :

$$\frac{\partial \Phi}{\partial t} + \mathbf{v}\nabla.\Phi = 0 \tag{11}$$

where $\mathbf{v}$ is the Level-Set velocity field. This velocity is evaluate by assuming that water freezes normal to the wall $\mathbf{v}_{wall}$ with the following relation :

$$\mathbf{v}_{wall} = \frac{\dot{m}_{ice}}{\rho_{ice}}.\mathbf{n} \tag{12}$$

To compute the Level-Set velocity field $\mathbf{v}$, the wall velocity is propagated perpendicular to the wall.

In [24] we used a first order scheme in time. Although such scheme is expected to be dissipative it was found to be sufficient in the context of a single step icing model. However for a multi-step icing formulation where the mesh is not sufficiently refined in the Level-Set path we seek a higher order scheme. We observed a small mass loss in sharp deformed parts of the geometry due to dissipation. We first tried the method proposed by [31], where the time scheme is not purely implicit but an inflow-implicit/ outflow-explicit. However, no improvements on the dissipation were observed, since staying in the implicit context introduces a dissipative viscosity. The scheme is still under development to check for the capability of higher order spatial schemes on reducing the dissipation due to the attracting features of an implicit representation.

At the same time a fully explicit scheme is being developed and optimised. A third order Runge-Kutta scheme (RK3) is utilized and is coupled with a Weighted Essentially Non-Oscillatory spatial scheme. This choice was justified by many researchers for its ability to treat hyperbolic equations. We use the

formulation proposed by [32] along with the one proposed by [33] where he proposed a modified version of the classical WENO scheme. One important factor of the Level-Set function is being a signed distance function. Such a task is difficult at the initialization step, and to make it worse this property gets deteriorated instantly after advection. A numerical method was proposed by [18] to recover $|\Phi(\overrightarrow{x})| = d(\overrightarrow{x})$. One significant feature of a signed distance function is that $|\nabla\Phi| = 1$ and the normal direction is simply $\mathbf{n} = \nabla\Phi$. Consequently, the steady state solution of evolving Φ using Eq. 13, would reduce to the desired $|\nabla\phi| = 1$:

$$\frac{\partial\phi}{\partial t} + |\nabla\phi| = 1 \tag{13}$$

This reinitialization equation is rewritten in another format to allow for a conservative finite volume discretization :

$$\frac{\partial\phi}{\partial t} + sign(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\phi = sign(\phi) \tag{14}$$

Where the $sign(\phi)$ function is approximated numerically by the smooth function :

$$sign(\phi) = \begin{cases} -1 & \phi < -\epsilon \\ \frac{\phi}{\sqrt{\phi^2+\epsilon^2}} & |\phi| \leq \epsilon \\ 1 & \phi > \epsilon \end{cases} \tag{15}$$

The same schemes used for the Level-Set advection were used for the reinitialization equation. Implicit and semi-implicit schemes resulted in poor results. On the other hand, explicit (RK3) schemes were more capable of reproducing the required features. [33] showed that the reinitialization did not reduce the dissipation on the contrary, it increased dissipation. He further attributed such behaviour to the smoothed $sign(\phi)$ function. [34, 18] suggested the use of another function which was initially proposed by [35] written as :

$$sign(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2\epsilon^2}} \tag{16}$$

Thus we propose using an implicit Level-Set advection scheme since the dissipation is minuscule taking into account the simplifications used in modelling the droplet impingement and ice formation. However, an explicit (RK3), HJ WENO scheme is used for the reinitialization using Eq. 14 and 16, to initialize the signed distance function, and to restore it at the end of each ice layer (at the end of advecting Φ).

Other important functions related to the Level-Set are the Heaviside function $H(\phi)$, the solid characteristic function χ_s, and the Dirac function δ_s. The Heaviside function is used to calculate the solid characteristic function which are both given by :

$$H(\phi) = \begin{cases} 0 & \Phi < -\epsilon \\ 0.5\,(1 - (\Phi/\epsilon) - \sin(\pi\Phi/\epsilon)/\pi) & |\phi| \leq \epsilon \\ 1 & \phi > \epsilon \end{cases} \tag{17}$$

$$\chi_s = H(-\Phi) \tag{18}$$

The Dirac delta function δ_s is set to one at the interface and zero elsewhere. Mathematically the Dirac delta function is the normal derivative of the Heaviside function. Consequently, it is smoothed out at the interface.

One important ingredient is still missing and that is the propagated Level-Set velocity field. When using embedded-grid techniques propagating the velocity in the normal direction as proposed before is not an easy task. An additional PDE can be solved to propagate such velocity in the normal direction [31, 18] as shown in Eq. 19

$$\frac{\partial w}{\partial t} + sign(\phi)\overrightarrow{n} \cdot \nabla w = 0 \tag{19}$$

whose final solution gives the required Level-Set velocity $\mathbf{v} = w_{steady}$. To put the finishing lines we explain in the following section the IBM-LS technique used for the impingement and other implications.

IBM-LS Technique

We impose field variables on the first outer layer cell centres based on the required boundary condition. Consider the case shown on Fig. 4, where the required BC will be imposed on the centre of cells marked with an x and are henceforth called IB points. This procedure can be summarized as follows :

(1) First we search the points that fall on the interface by comparing $sign(\phi)$ with neighbours in all directions. These points are saved initially with their indices and the outward direction as the opposite to the point that had a different sign. In Fig. 4 point i has a neighbour with different sign to the left and thus the outside direction is to the right.

(2) An image point (ip) is generated at a distance Δ from the IB point through the normal direction. The normal direction is directly available in a Level-Set framework ans is given by $\mathbf{n} = \nabla\phi$.

(3) Following that we search for the closest point (cp) starting from the point located in the outer direction.

These are all the ingredients we generate initially. Steps 1 to 3 are repeated only when treating a modified geometry, or in other words for a different ice layer. The above ingredients can be used by the droplet module to impose impingement BC, and can also be used by the flow solver to construct wall laws.

The state vector is extrapolated from the closest point (cp) to the image point (ip) by the following formula :

$$\mathbf{Q}_{ip} = \mathbf{Q}_{cp} + \nabla\mathbf{Q}_{cp} \cdot \overrightarrow{\mathbf{r}} \tag{20}$$

where $\overrightarrow{\mathbf{r}}$ is the vector pointing from point (cp) to point (ip), and is equal to the distance times the normal vector $\overrightarrow{\mathbf{r}} = -\Delta \cdot \mathbf{n}$.

Based on the state vector and the required BC type we then calculate the required state vector at the IB point. For a Nuemman BC we impose $\mathbf{Q}_{IB} = \mathbf{Q}_{ip}$ to obtain a zero normal gradient. For a Dirichlet BC however, the value is extrapolated as $\mathbf{Q}_{IB} = \mathbf{Q}_{ip} * \phi_{ip}/\phi_{im}$. [36] proposed a higher order scheme by adding a projection point, a farther image point, an extra image point, and a mirrored point. But we currently stay with a first order extrapolation scheme. For the turbulent wall-modelling we are still studying different models. One promising model is the one proposed by [37].

Consequently, the IB points are not included in the solution of the conservation equation but are imposed explicitly. Interior points are decoupled and have no effect on the flow.

The two ingredients required by the Thermodynamic module, namely : the wall shear stress, and the heat transfer coefficient, are to be accuired by this IBM-LS formulation but are still under developement.

Références

[1] Reehorst, Al, National Transportation Safety Board Aircraft Accident Investigation Supported, 2005.

[2] Kays, William Morrow and Crawford, Michael E and Weigand, Bernhard, Convective heat and mass transfer, McGraw-Hill, 2012.

[3] Messinger, Bernard L, Equilibrium temperature of an unheated icing surface as a function of air speed, Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences), 20 (1) (1953).

[4] Hasanzadeh, K and Laurendeau, E and Paraschivoiu, I, Quasi-Steady Convergence of Multistep Navier-Stokes Icing Simulations, Journal of Aircraft, American Institute of Aeronautics and Astronautics 50 (4) (2013), pp. 1261–1274.

[5] Bidwell, Colin S and Potapczuk, Mark G, Users manual for the NASA Lewis three-dimensional ice accretion code (LEWICE 3D), 1993.

[6] Beaugendre, HÈloÔse and Morency, FranÁois and Habashi, Wagdi G., Development of a second generation in-flight icing simulation code, Journal of fluids engineering 128 (2) (2006), pp. 378–387.

[7] Montreuil, ONERA and A. Chazottes, ONERA and D. Guffond, ONERA and A. Murrone, ONERA and F. Caminade, Dassault-Aviation and S. Catris, Eurocopter, ECLIPPS : 1. Three-Dimensional CFD Prediction of the Ice Accretion, 2009.

[8] Scott, JN and Hankey, WL and Giessler, FJ and Gielda, TP, Navier-stokes solution to the flowfield over ice accretion shapes, Journal of Aircraft, 25 (8) (1988), pp. 710–716.

[9] Bourgault, Yves and Habashi, Wagdi G and Dompierre, Julien and Baruzzi, Guido S, A finite element method study of Eulerian droplets impingement models, International Journal for Numerical Methods in Fluids, Wiley Online Library, 29 (4) (1999), pp. 429–449.

[10] Jung, Sungki and Myong, Rho, Numerical Modeling for Eulerian Droplet Impingement in Supercooled Large Droplet Conditions, American Institute of Aeronautics and Astronautics, 2013.

[11] Kim, Jeewoong and Dennis, P Garza and Sankar, Lakshmi and Kreeger, Richard, Ice Accretion Modeling Using an Eulerian Approach for Droplet Impingement, American Institute of Aeronautics and Astronautics, 2013.

[12] Zhu, Chengxiang and Fu, Bin and Sun, Zhiguo and Zhu, Chunling, 3D ICE ACCRETION SIMULATION FOR COMPLEX CONFIGURATION BASING ON IMPROVED MESSINGER MODEL, International Journal of Modern Physics : Conference Series, 19 (2012) pp. 341–350.

[13] Cao, Yihua and Ma, Chao and Zhang, Qiang and Sheridan, John, Numerical simulation of ice accretions on an aircraft wing, Aerospace Science and Technology, 23 (1) (2012), pp. 296–304.

[14] Jung, SK and Myong, Rho Shin and Cho, Tae-Hwan, Development of Eulerian droplets impingement model using HLLC Riemann solver and POD-based reduced order method, AIAA Paper, 3907 (2011).

[15] Myers, Tim G., Extension to the Messinger model for aircraft icing, AIAA journal, 39 (2) (2001), pp. 211–218.

[16] H. Beaugendre and F Morency and W.G. Habashi, FENSAP-ICE's three-dimensional in-flight ice accretion module : ICE3D, Journal of Aircraft, 40 (2) (2003), pp. 239–247.

[17] Osher, S., Sethian, J.A., Fronts Propagating with Curvature Dependent Speed : Algorithms Based on Hamilton-Jacobi Formulations, Journal of Computational Physics, 79 (1988) pp. 12–49.

[18] Osher, Stanley and Fedkiw, Ronald, Level set methods and dynamic implicit surfaces, Springer Science & Business Media 153 (2006).

[19] Beaugendre, Héloïse and Morency, François and Gallizio, Federico and Laurens, Sophie, Computation of ice shedding trajectories using cartesian grids, penalization, and level sets, Modelling and Simulation in Engineering, Hindawi Publishing Corp., 2011 (2011).

[20] P. Angot, C.H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numer. Math., 81 (4) (1999), pp. 497–520.

[21] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech., 37 (1) (2005), pp. 239–261.

[22] X.Y. Hu, B.C. Khoo, N.A. Adams, F.L. Huang, A conservative interface method for compressible flows, J. Comput. Phys., 219 (2006), pp. 553–578.

[23] Pena, Dorian and Hoarau, Yannick and Laurendeau, Eric, Development of a Three-Dimensional Icing Simulation Code in the NSMB Flow Solver, International Journal of Engineering Systems Modelling and Simulation, 8 (2) (2016), pp.86–98.

[24] Pena, Dorian and Hoarau, Yannick and Laurendeau, Eric, A single step ice accretion model using Level-Set method, Journal of Fluids and Structures, 65 (2016), pp. 278–294.

[25] Vos, JB and Rizzi, AW and Corjon, A and Chaput, E and Soinne, E, Recent Advances in Aerodynamics inside the NSMB (Navier-Stokes Multi-Block) Consortium, AIAA paper, 802 (1998).

[26] W. Mulder, S. Osher, J. Sethian, Computing Interface Motion in Compressible Gas Dynamics, J. Comput. Phys, 100 (1992), pp. 209–228.

[27] R. Abgrall, H. Beaugendre, C. Dobrzynski, An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques, Journal of Computational Physics, 257 (2014), pp. 83–101.

[28] Schiller, L and Naumann, Z, A drag coefficient correlation, Vdi Zeitung, 77 (318) (1935).

[29] Francesco Capizzano and Emiliano Iuliano, A Eulerian Method for Water Droplet Impingement by Means of an Immersed Boundary Technique, J. Fluids Eng, 136 (4) (2014).

[30] Kreith, Frank and Manglik, Raj and Bohn, Mark, Principles of heat transfer, Cengage learning, 2010.

[31] P. Frolkovič, K. Mikula, J. Urbán, Semi-implicit finite volume level set method for advection motion of interface in normal direction, Applied Numerical Mathematics, 2014.

[32] C. W. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory schemes for Hyperbolic conservation laws, NASA/CR-97-206253 ICASE Report No. 97-65, 1997.

[33] S. Tanguy, Dèveloppement d'une mèthode de suivi d'interface. Applications aux ècoulements diphasiques. Modèlisation et simulation, Universitè de Rouen, 2004.

[34] K. VORONETSKA, Simulation numèrique directe des ècoulement á phases dispersèes, Universitè de Bordeaux, 2012.

[35] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, A PDE-based fast local level set method, Journal of Computational Physics, 155 (1999), pp. 410–438.

[36] C. Chi, B. J. Lee, and H. G. Im, An improved ghost-cell immersed boundary method for compressible flow simulations, International Journal for Numerical Methods in Fluids, 83 (2017), pp. 132–148.

[37] F. Capizzano, A turbulent Wall Model for Immersed Boundary Methods, 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposi, 2010.

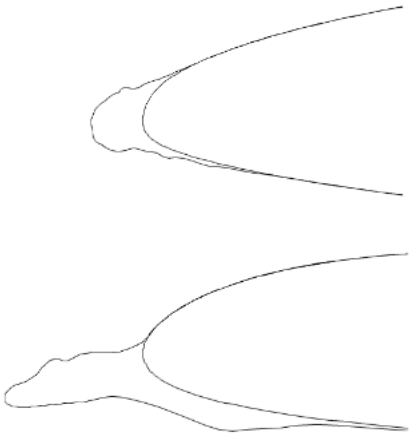

FIGURE 1 – Typical forms of icing simulated in NSMB. On top : rime icing on a NFL0414 airfoil at $Re = 6.95 \times 10^6$, $T = 257.59K$. On bottom : glaze icing on a SA13112 airfoil at $Re = 3.91 \times 10^6$, $T = 263.19K$.

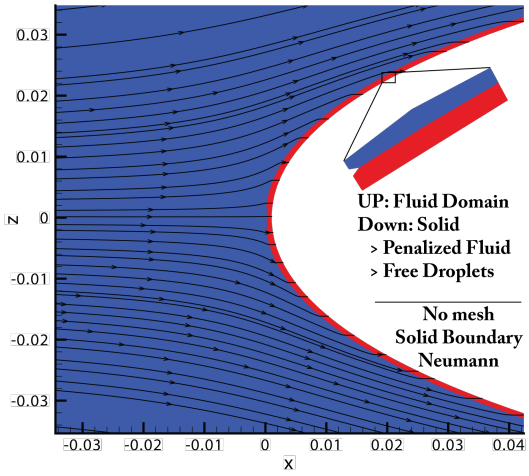

FIGURE 2 – Penalized air flow and free droplet flow, Streamlines shown are droplets trajectories.

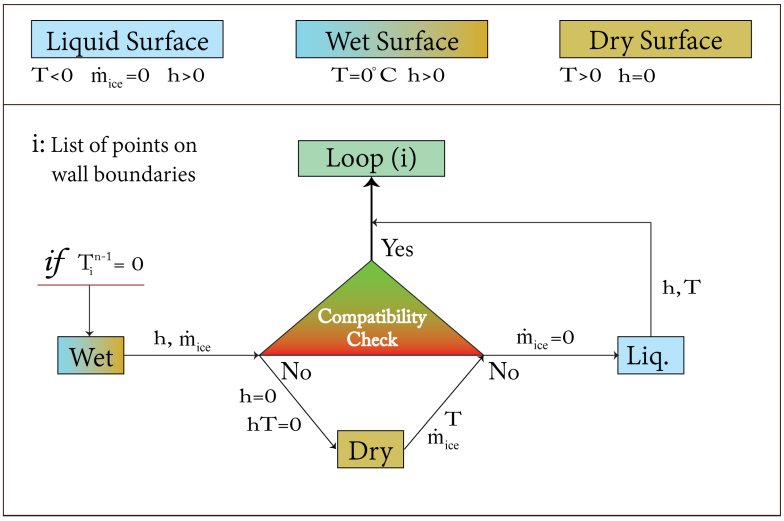

FIGURE 3 – Trial and error procedure at each wall node i.

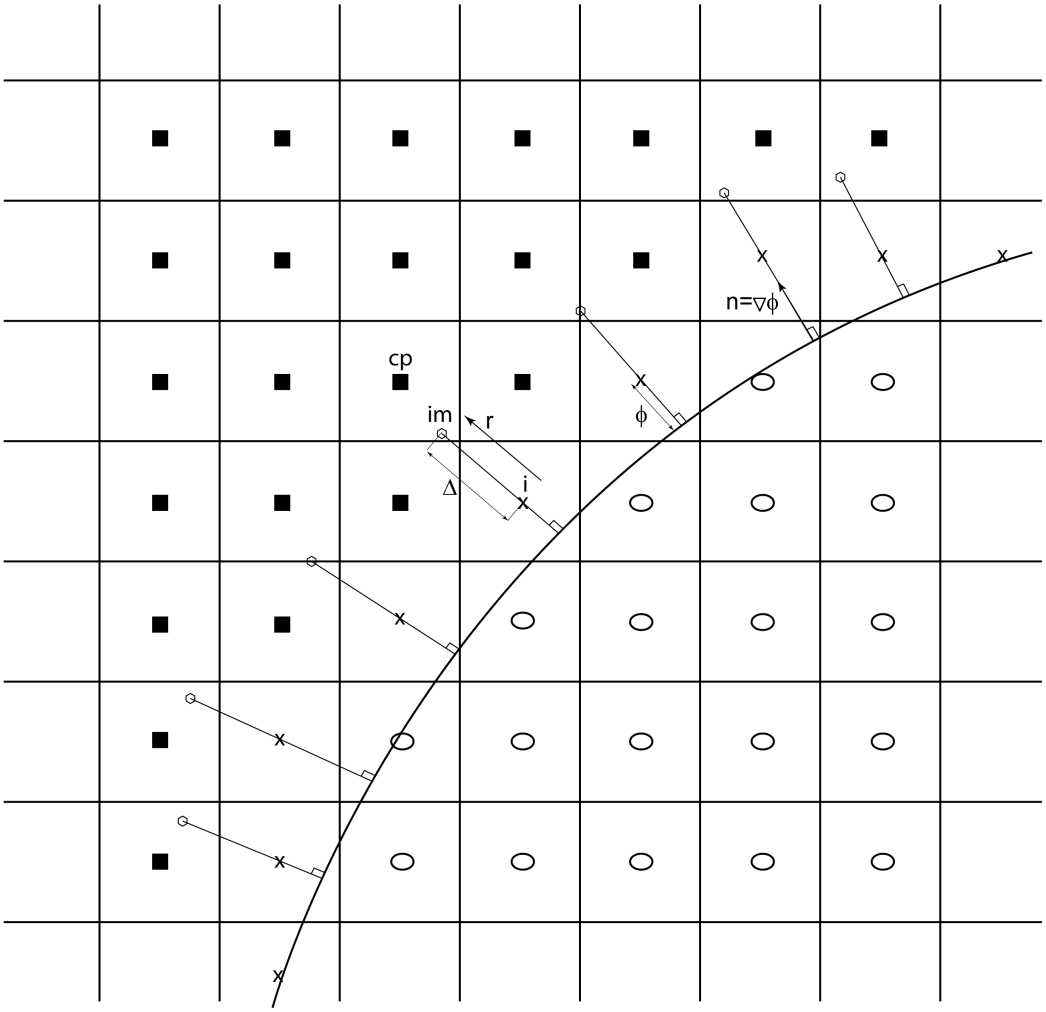

FIGURE 4 – Schematics of the computaion domain in an IBM-LS framework. Black squares are points in the physical flow domain, points marked by x, are the IB cells layer, circels are points inside the solid body given by negative ϕ, and hexagons represent the image points.