A Penalization Method for 2D Ice Accretion Simulations

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

► To cite this version:

Pierre Lavoie, Ghislain Blanchard, Emmanuel Radenac, Éric Laurendeau, Philippe Villedieu. A Penalization Method for 2D Ice Accretion Simulations. International Conference on Icing of Aircraft, Engines, and Structures, Jun 2019, Minneapolis, United States. 10.4271/2019-01-1939 . hal-02747032

HAL Id: hal-02747032

<https://hal.science/hal-02747032>

Submitted on 3 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Penalization Method for 2D Ice Accretion Simulations

P. Lavoie^{1,2}, E. Radenac¹, G. Blanchard¹, E. Laurendeau², P. Villedieu¹

¹ONERA - The French Aerospace Lab, F-31055 Toulouse, France

²Department of Mechanical Engineering, Polytechnique Montréal, C.P. 6079, succ. Centre-ville, Montréal (Qc.), H3C 3A7, Canada

Abstract

Numerical tools for 3D in-flight icing simulations are not straightforward to automate when seeking robustness and quality of the results. Difficulties arise from the geometry and mesh updates which need to be treated with care to avoid folding of the geometry, negative volumes or poor mesh quality. This paper aims at solving the mesh update issue by avoiding the re-meshing of the iced geometry. An immersed boundary method (here, penalization) is applied to a 2D ice accretion suite for multi-step icing simulations. The suggested approach starts from a standard body-fitted mesh, thus keeping the same solution for the first icing layer. Then, instead of updating the mesh, a penalization method is applied including: the detection of the immersed boundary, the penalization of the volume solvers to impose the boundary condition and the extraction of the surface data from the field solution. Although the current work is intended for 3D simulations, the content of this paper is limited to 2D applications to show the feasibility of the method. Tests are performed on multi-step rime and glaze ice cases where it is observed that the penalization method is able to produce sensible ice shapes compared to a body-fitted approach. However, accurate results are only achieved if the mesh is *a priori* refined in the zone where ice accretion is expected. The paper will further discuss the computational time, mesh refinement requirements, accuracy, limitations and recommendations for improvement of the penalization method applied to icing.

Introduction

Ice accretion simulation tools are nowadays widely used in the design and certification of aircraft. The recent reinforcement in icing certification norms and the aeronautical industry interest in the simulations of complex 3D ice shapes, such as scallops or lobster-tails typical of swept-wing icing, lead to the development of new and improved 3D ice accretion suites.

Ice accretion tools are typically based on a quasi-steady assumption where modules are called sequentially and solved to steady state. The main modules can be listed as shown in Figure 1: (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, (5) a solver to perform a heat and mass balance to obtain the ice thickness and finally (6) a tool to update the geometry based on the ice thickness. In a multi-step approach, the process can be repeated several times during the ice accretion simulation, where re-meshing and mesh-adaption technique are used from the second step further.

These tools are difficult to automate when seeking robustness and quality of the results, especially in 3D. The issue generally comes from an unphysical update of the iced geometry (e.g. geometry overlap) or from the difficulty to re-mesh the ice shape. This paper is concerned by the methods required for the automatization of 2D and 3D icing suites using a multi-step approach, more specifically by the re-meshing/mesh update step. It is assumed that using an Immersed Boundary Method (IBM) could help in the automatization of ice accretion tools by avoiding the re-meshing or at least, minimizing the work required to adapt the mesh.

Figure 1: Sequential call to modules in a multi-step icing simulation

Body-fitted (BF) meshes are typically used in ice accretion tools. In this case, work is spent on the generation of a good quality mesh while the accuracy of the boundary condition is not an issue, since the mesh matches the boundary. With an Immersed Boundary Method, a good quality mesh is still required but most of the work is spent on the imposition of the boundary condition. Also, because the mesh does not have to match the boundary, mesh adaptation is easier to perform.

The application of IBM to a full ice accretion suite is rare. Ice accretion related work is performed by CIRA, where a discrete forcing immersed boundary method is applied to the aerodynamic solver [6] and to an Eulerian droplet trajectory solver [7]. No application to ice accretion is presented. Al-Kebsi et al. [2] use a continuous forcing immersed boundary method (penalization) to solve the RANS equations. On the other hand, a discrete forcing approach is used for the droplet trajectory solver (Ghost-Cell). Again, no result is yet presented for ice accretion application. At ONERA [10], immersed boundary methods are available under elsA and CASSIOPEE software. Also, an IBM is used by [12] to estimate the

aerodynamic degradation due to icing and a penalization method is used by [3] for the evaluation of ice shedding trajectories.

In this paper, the application of a penalization method to an ice accretion suite is investigated. Although the penalization method is intended to solve issues predominant for 3D simulations, the content of this paper is limited to 2D applications to show the feasibility of the method. The current approach differs from the literature since the penalization is applied to the Euler equations instead of RANS and the penalization is also applied to the droplet solver which was not performed by previous authors.

This paper first presents the ice accretion suite used as the development platform. Then, implementation details for the penalization method are covered: preprocessing, penalization, interpolation. Rime ice and glaze ice multi-step icing cases are used for verification where the body-fitted and penalized solutions are compared. The paper also discusses the computational cost, mesh refinement requirements, accuracy, limitations and recommendations for improvement.

Methodology

The 2D ice accretion suite IGLOO2D [15] is used as the development environment. IGLOO2D is an unstructured ice accretion suite. The mesh generation can be handled using GMSH to generate unstructured meshes. Otherwise, an in-house structured mesh generation tool can be used [15]. The aerodynamic field can be evaluated using the potential flow theory or the Euler equations. The convective heat transfer is evaluated using either a simplified integral boundary layer method (SIM2D) [15] or the full boundary layer equations. For the droplet trajectories and impingement evaluation, both Lagrangian and Eulerian solvers are available. 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. In this paper, the results are presented for unstructured meshes using the compressible and inviscid Euler equations in combination with the simplified integral boundary layer method and the Eulerian formulation for the droplet trajectories and impingement rates.

Figure 2: Sequential call to modules in a multi-step icing simulation using an Immersed Boundary Method.

For the application of the immersed boundary method, the suggested approach is to start the multi-step ice accretion from a standard Body-Fitted mesh, thus keeping the original BF solution for the 1st icing layer. For the subsequent steps, instead of performing the mesh update, a penalization method is used to impose the correct boundary condition on the immersed boundary. The use of the penalization method requires some modification to the ice accretion suite. It can be divided in: a volume solver modification, a preprocessing step and an extraction step, as illustrated in Figure 2.

Surface Data Extraction

The modules can be classified either as volume or surface solvers. The aerodynamics (EULER2D) and the Eulerian droplet trajectories (TRAJE2D) are solved on volume meshes (2D). On the other hand the simplified integral boundary layer method (SIM2D) and the ice accretion (MESSINGER2D) are solved on surface grids (1D). Relevant surface information from the volume solvers is communicated to the surface solvers at each step of the multi-step loop (e.g. pressure, velocity, droplet impingement rate, etc.).

However, the penalization method does not explicitly provide the data on the immersed boundary (IB). Instead, the variables are known in the surrounding cells. An additional extraction step is thus required to recover the data and pass it to the surface solvers. 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.

Penalization

With a 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 term is turned on and off using a mask function (χ) which can take several forms, the simplest one being a sharp Heaviside function ($\chi = 0$ in the fluid and $\chi = 1$ in the solid). An example for the Penalized Euler equations is as follows, where $\vec{v}$ is the air velocity, I is the identity matrix, E is the total energy, λ is a penalization parameter and $\vec{v}_n$ is the normal velocity to the immersed boundary:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) &= 0 \\ \frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v} \otimes \vec{v} + P I) &= -\lambda \chi \rho \vec{v}_n \\ \frac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + P) \vec{v}) &= -\lambda \chi \rho \vec{v}_n \cdot \vec{v} \end{aligned} \quad (1)$$

The RHS of equation 1 represents the penalization terms. These enforce a slip boundary condition at the IB ($\vec{v} \cdot \vec{n} = 0$). Natural Neumann boundary conditions are applied on the density (ρ) and Pressure (P), thus no penalization is required for these variables. In summary the following boundary conditions are applied at the IB:

$$\frac{\partial \rho}{\partial n} = 0, \frac{\partial P}{\partial n} = 0 \text{ and } \vec{v} \cdot \vec{n} = 0 \quad (2)$$

Note that for an ideal flow, the boundary condition on pressure should include the curvature of the boundary [11, 8]. In this paper, the curvature is neglected as an approximation.

The equations must be modified in the aerodynamic and droplet trajectory solvers to account for the source terms. Because the source terms are stiff in a penalization method, an implicit formulation is used to solve the system of equations. The surface-based modules such as the ice accretion and boundary layer solvers are kept unmodified.

The penalization of the Euler equations for this paper is slightly more complex than the example presented in Eq. (1). The implementation is inspired on the work of [4, 5] and use additional terms to avoid a jump in the solution in the IB vicinity. For the Eulerian droplet field, a more classical penalization is used where the equations are penalized only if the droplets are traveling from the solid to the fluid zone. In this case a zero flux condition is imposed at the wall. Otherwise the usual equations are solved.

IBM Preprocessing

As a mask function is required to apply the penalization source terms, a preprocessing step is required. The preprocessing is performed as a replacement to the mesh update. First, the inside (solid) and outside (fluid) cells are detected. Knowing the list of edges defining a closed immersed boundary, a ray casting algorithm can be used for this matter [14]. Once this information is known, it can be used to determine the sign of the signed distance field. The signed distance (ϕ) is useful to evaluate the mask function and to determine the normals to the IB (given by $\vec{n} = \nabla\phi/||\nabla\phi||$). For instance, the normals are required to enforce the slip velocity condition in the Euler flow solver. The signed distance to the immersed boundary can be evaluated using the Eikonal equation or via an algebraic approach. The latter is used in this paper which consists in seeking the minimum projected distance to the edges defining the IB.

As stated earlier, the penalization also requires a data interpolation step. The weights used in the least-square approach are geometry dependent only. Hence, they can be evaluated and stored at the preprocessing step. The neighborhood of each interpolation point is also determined at this stage.

Table 1: Simulation parameters for the icing simulations

	Case 1	Case 2	Case 3
Ice Type	Rime	Glaze	Glaze
Profile	NACA0012	GLC305	NACA0012
Chord	0.5334 m	0.9144 m	0.5334 m
AoA	4.0°	4.0°	4.0°
Mach number	0.321	0.273	0.317
Static Pressure	101 325 Pa	101 325 Pa	101 325 Pa
Static Temperature	256.8 K	268.3 K	262.3 K
TWC	0.34 g/m ³	0.54 g/m ³	0.60 g/m ³
MVD	20 μ m	20 μ m	15 μ m
Icing Time	690 s	1350 s	384 s

Results

The penalization method is applied to the entire icing suite to perform multi-step icing simulations. First, a 2 steps simulation is selected to ease the comparison and analysis (mesh refinement study). Second, a 10 steps simulation is performed to assess the method on a more representative case. The selected rime and glaze ice cases are presented in Table 1. For all cases, a body-fitted simulation is used as a comparison base. The initial BF mesh is exactly the same for the body-fitted and the immersed boundary simulations. For the BF approach re-meshing is performed between each ice layers. For the penalization approach, the re-meshing is totally avoided and the immersed boundary method is applied on the 2nd step further.

Mesh Generation

The simulations are performed on unstructured grids made solely of triangles. Because it is desired to avoid the re-meshing step, the mesh is initially refined in the area where ice accretion is expected, typically the leading edge (LE). The wall cell size is kept constant from the wall up to a height representing 10% of the chord length. In this way, the same wall cell size is seen by the immersed boundary as the icing layers are added, allowing a better comparison with the body-fitted approach. The BF approach uses the same initial mesh and its subsequent meshes (generated by re-meshing) also exhibit this LE refinement zone. The wall cell size is kept constant over the entire airfoil surface and selected to capture the leading edge curvature adequately. Consequently, a smaller cell size is used for the GLC305 airfoil due to its higher LE curvature. It also provides a better discretization of the expected ice horn which is also of high curvature at its tip. The far-field is located at 50 chord lengths.

Figure 3: Extra coarse mesh example for the NACA0012 airfoil (with LE refinement)

A family of 4 meshes is generated for each airfoil to support the mesh refinement study. A reference mesh is used as a starting point for the refinement study. From this mesh two coarser meshes and one finer mesh are produced. Since unstructured meshes are used, the strategy is to divide the characteristic cell size by 2 to obtain a finer mesh. In turns the number of cells is roughly multiplied by 4 for each refinement. An example is provided for the NACA0012 airfoil in Figure 3 (2842 cells) for the extra-coarse mesh. The cell count for each refinement level is listed in Table 2.

Table 2: Number of cells for each mesh level (with LE refinement)

	NACA0012	GLC305
Extra-coarse	2 842	11 350
Coarse	9 498	41 820
Reference	35 547	168 896
Fine	141 160	633 461

Note on Accuracy

The penalization method is known to be 1st order accurate in the imposition of the boundary condition while the body-fitted approach is 2nd order. To alleviate this drawback a finer mesh can be used near the wall, or automatic mesh refinement technique can be used [1]. It is also possible to impose a 2nd order boundary condition as suggested by [13, 9] where a new discretization of the penalization term is used. In the current paper, the results are limited to a 2nd order spatial discretization in the field with a classical 1st order implementation of the penalization method. It will be shown that satisfactory results can still be achieved for ice accretion simulations.

The Effect of Surface Refinement

When considering discretization errors, two effects can be listed which play a role on accuracy, (1) the refinement of the volume mesh and (2) the refinement of the surface mesh. For the BF method, re-meshing of the ice geometry is performed for each new layer, thus the surface as well as the computational domain are refined according to the ice shape. For the IBM, the volume mesh refinement is handled by *a priori* refining the initial mesh near the LE. As the initial mesh is re-used, and because the ice shape is updated via a Lagrangian displacement of the nodes, the surface mesh undergoes a coarsening for each new ice layer (for a convex ice shape). A more suitable solution is to apply a surface refinement after the node displacement to achieve a similar level of discretization as the BF approach.

The effect of surface refinement can be observed for instance on Figure 5 which presents the rime ice results of Case 1 on different mesh refinement level. When no refinement is applied (comparing curves labeled “BF layer 2” and “IB NoSurfRef layer 2”), there is a large discrepancy between the methods, but as the mesh is refined the ice shapes are getting closer to the body-fitted solution. It means that, when avoiding surface refinement, a finer volume mesh is required (the initial surface mesh is also finer) in order to obtain similar results to the BF approach with re-meshing.

When surface refinement is used (curves labeled “IB layer 2”), the body-fitted and penalization methods provide comparable results for the same mesh refinement level. Hence, in the following sections, the focus will be on the results with surface refinement (curves labeled “IB layer 2”). For these results the errors are induced by: (1) the 1st order nature of the boundary condition as imposed by the penalization method, (2) the mesh which is no longer identical from the 2nd step further but shares a comparable cell size and surface discretization at the boundary, (3) the way the boundary condition is enforced (immersed boundary vs body-fitted).

Sensitivity to Mesh Refinement

The goal of the suggested IB ice accretion method is to reproduce the ice shape from the BF approach, but avoiding the re-meshing. Because a 1st order boundary condition is used with the penalization method, a discretization error is expected. In this section, the consistency is verified. The mesh refinement study will also give some insight on the mesh requirement to obtain satisfactory ice shape with the penalization method. Note that for the 1st ice layer (curves labeled “layer 1”), the ice shape is exactly the same for the body-fitted and penalization methods since the application of the boundary condition and the initial mesh are the same. Thus, the analysis and comparisons will focus on the 2nd ice layer.

For rime conditions, the ice shape is mostly governed by the droplet trajectory solver as the droplets freeze immediately upon impact. For the rime ice Case 1 (Figure 5), it was observed that the collection efficiency (β) is reproduced with very good accuracy at the IB compared to the BF solution. This leads to a very good match between the Body-Fitted and penalization methods independently of the mesh refinement level.

For glaze ice conditions, the aerodynamic solution has a large effect on the ice shape as it controls the convective heat transfer coefficient and in turn, influences the behavior and/or presence of runback water. For the glaze ice Case 3 (Figure 6), the penalization method can fairly reproduce the ice shape from the BF method on all mesh refinement level and with a particularly good match on the finest mesh (Figure 6d). The ice shape is not as good as for the rime condition but the difference in ice shape is reduced as the mesh is refined. The larger difference is caused by a discrepancy in the aerodynamic solution in the vicinity of the boundary. This can be explained by the order of accuracy of the BC for the immersed boundary method or by the mesh characteristics near the boundary. Although the cell size is comparable for the BF and IB approaches, the location of those cells is not strictly the same.

The glaze ice Case 2 (Figure 7) shows a limitation of the current penalization method. For this case, a large flow separation may occur downstream of the ice horn (Figure 4). It is well known that inviscid simulations cannot capture flow separations. This issue must nonetheless be handled by ice accretion suites whether it uses a body-fitted or immersed boundary approach, which means that the two methods should also provide the same result. The aerodynamic field in Figure 4 clearly shows that it is not the case. It must be noted that this configuration affects the ice shape because glaze conditions may occur even at the horn tip depending on the separation pattern (which makes Case 2 very difficult). As the flow separation is sensitive to the accuracy of the boundary condition, it is believed that the implementation of a 2nd order penalization method could help in obtaining similar results between the BF and IB methods independently of the mesh refinement level.

Figure 4: Icing case 2, 2nd layer, reference mesh, comparison of the separation bubble for the IB (top) and BF (bottom) approaches

Figure 5: 2-layer rim ice accretion (Case 1), comparison between the Body-Fitted (BF) and the penalization methods. Mesh refinement: (a) Extra-coarse, (b) Coarse, (c) Reference, (d) Fine

Figure 6: 2-layer glaze ice accretion (Case 3), comparison between the Body-Fitted (BF) and the penalization methods. Mesh refinement: (a) Extra-coarse, (b) Coarse, (c) Reference, (d) Fine

Figure 7: 2-layer glaze ice accretion (Case 2), comparison between the Body-Fitted (BF) and the penalization methods. Mesh refinement: (a) Extra-coarse, (b) Coarse, (c) Reference, (d) Fine

Multi-Step Icing over 10 Layers

In the previous section, 2-step icing simulations were performed using the penalization method and compared with the standard body-fitted approach. In this section, 10 icing layers are computed to assess the penalization method over a more realistic simulation. The reference mesh is selected for this study as it provides a satisfactory ice shape compared to the fine mesh and for a lower simulation cost.

It can be observed that for 10 layers the penalization method still compares well to the BF approach for both rime ice (Case 1, Figure 8) and glaze ice (Case 3, Figure 9), although the difference in ice shape is accentuated as icing layers are added. A larger discrepancy is observed for Case 3. This can again be explained by the influence of the aerodynamic solution which is greater for a glaze ice case. Thus the accuracy of the BC imposition also has a greater impact.

For the more complex glaze ice shape (Case 2, Figure 10), the 10-step simulation exhibits too much runback water and the height of the ice horn is greatly reduced. Again, this is due to the difficulty in capturing the separation bubble for both body-fitted and immersed boundary methods. The BF and IB methods behave in a similar way, with again a difference in ice shape that can be explained by a discrepancy in the aerodynamic solution near the boundary.

Figure 8: 10-layer rime ice accretion (Case 1) on the reference mesh, comparison between BF and IB solution (with surface refinement)

Figure 9: 10-layer glaze ice accretion (Case 3) on the reference mesh, comparison between BF and IB solution (with surface refinement)

Figure 10: 10-layer glaze ice accretion (Case 2) on the reference mesh, comparison between BF and IB solution (with surface refinement)

Simulation Costs

Considering the simulation costs, the use of a penalization method might increase the workload for two main reasons: (1) the higher cell count and (2) the computation of additional penalization terms in the equations (more operations per cell). With the penalization method, the mesh is refined in the zone where ice accretion is expected to maintain a reasonable cell size, typically near the leading edge (LE), see Figure 3. Otherwise, the ice shape sees a mesh which is locally coarser as the ice grows. Overall it increases the cell count comparing to a classical BF approach with re-meshing, where no special LE refinement is required (Figure 11).

Figure 11: Extra coarse mesh example for the NACA0012 airfoil (without LE refinement)

Table 3 presents the cell count for each mesh refinement level when no LE refinement is used. It also shows the ratio of cells between the Immersed Boundary mesh (with LE refinement) and the standard BF mesh (without LE refinement) in the column labeled (IB/BF). The number of cells increases by 16% to 26% depending on the icing case and mesh level. However, note that the cell count can be reduced for the penalization method by using an automatic mesh refinement strategy as a replacement to the initial LE refinement. This type of method is easier to implement with IBM because the cells are not required to match the boundary.

Table 3: Number of cells for each mesh level (without LE refinement)

	NACA0012		GLC305	
	Cells	IB/BF	Cells	IB/BF
Extra-coarse	2 462	1.16	9 288	1.22
Coarse	8 026	1.18	33 574	1.25
Reference	29 071	1.22	135 468	1.25
Fine	114 528	1.23	501 587	1.26

In the current approach the cell count does not change much between icing layers when re-meshing is employed. For the penalization method, the initial mesh is re-used as is and therefore, the cell count remains the same. Also, both the body-fitted and penalization methods use a mesh refined at the LE, leading to a comparable cell count which is independent on the number of ice layers. Thus, the higher simulation cost is driven by the increase in operation per cell due to the additional penalization terms.

Table 4 shows CPU time per iteration per cell for the aerodynamic and droplet trajectory solvers normalized by the body-fitted CPU time. As the 1st ice layer is exactly the same for both approaches, only the 2nd ice layer is compared. According to Table 4, the penalization method is 8% to 14% times as costly compared to the body-fitted approach, which is not a significant increase. It represents the additional work needed to compute the penalization terms in the equations. It also includes the work required to invert the matrix of the implicit system as more iterations are generally required with the penalization method. Also note that the penalization method exhibits a convergence rate very similar to the body-fitted approach, although on the icing case 2 the convergence to steady state is hindered by the presence of the separation bubble.

Table 4: Normalized CPU time per iteration per cell for the 2nd ice layer with the base/reference mesh (BF vs IBM)

	NACA0012		GLC305	
	Euler2D	TrajE2D	Euler2D	TrajE2D
BF	1.00	1.00	1.00	1.00
IBM	1.13	1.10	1.08	1.11
IBM (surface refinement)	1.11	1.12	1.14	1.11

Summary/Conclusions

In this paper, a penalization method is implemented in the Euler flow solver and the Eulerian droplet solver of the ice accretion suite IGLOO2D. To obtain the new software using the immersed boundary method, the extraction of the surface data is also required to communicate the information to the surface solvers (boundary layer and ice accretion solvers). This extraction step is performed using a weighted least-square interpolation technique. Additionally, the re-meshing step is avoided and replaced by a pre-processing step for the IBM.

In order to compare the penalization and body-fitted methods applied to ice accretion, unstructured meshes were generated including a refinement zone at the leading edge (where ice accretion is expected). In this way, the cell size seen by the immersed boundary is the same during the entire icing simulation.

Compared to a classical body-fitted mesh, refining the leading edge zone increases the number of cells by 16% to 26% for the cases presented in this paper. To reduce the associated computational cost, an automatic mesh adaptation technique could be used. It would ensure an adequate mesh quality near the ice shape while keeping the cell count fairly low (close to a classical BF mesh). The use of the penalization method also induces more operations per cell since terms are added to the equations. The increase in workload is however not significant (8% to 14%) compared to the BF approach.

It was shown that the penalization method using a simple 1st order boundary condition (BC) is an interesting candidate for ice accretion

software. It is able to reproduce rime ice and simple glaze ice cases with good accuracy for both 2-step and 10-step icing simulations. On the other hand, more complex glaze ice cases might require a 2nd order BC in order to reproduce the BF results in the presence of flow separation induced by large ice horns (high curvature). For this case, a fully coupled Euler/BL method or even RANS could be used to accurately capture the flow separation.

The penalization method showed to be promising for 2D ice accretion simulation, although requiring some improvements to handle more complex scenarios. The application to 3D ice accretion suites should be straightforward and could greatly simplify their automation.

References

1. Abgrall, R., H. Beaugendre, et C. Dobrzynski. «An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques.» *Journal of Computational Physics* 257 (2014): 83-101.
2. Al-Kebsi, A., D. Pena, E. Laurendeau, R. Mose, et Y. Hoarau. «Multi-Step Level-Set Ice Accretion Simulation with the NSMB solver.» *23e Congrès Français de Mécanique*. 2017.
3. Beaugendre, Héloïse, François Morency, Federico Gallizio, et Sophie Laurens. «Computation of Ice Shedding Trajectories Using Cartesian Grids, Penalization, and Level Sets.» *Model. Simul. Eng.* (Hindawi Publishing Corp.) 2011 (1 2011): 3:1--3:15.
4. Brown-Dymkoski, Eric, Nurlybek Kasimov, et Oleg V. Vasilyev. «A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows.» *Journal of Computational Physics* 262 (2014): 344-357.
5. —. «A Characteristic-Based Volume Penalization Method for Arbitrary Mach Flows Around Solid Obstacles.» Édité par Jochen Fröhlich, Hans Kuerten, Bernard J. Geurts et Vincenzo Armenio. *Direct and Large-Eddy Simulation IX*. Cham: Springer International Publishing, 2015. 109-115.
6. Capizzano, Francesco. «Turbulent Wall Model for Immersed Boundary Methods.» *AIAA Journal* 49 (2011): 2367-2381.
7. Capizzano, Francesco, et Emiliano Iuliano. «A Eulerian Method for Water Droplet Impingement by Means of an Immersed Boundary Technique.» *Journal of Fluids Engineering* (ASME) 136 (2 2014): 040906--040906-8.
8. Dadone, A., et B. Grossman. «Surface boundary conditions for the numerical solution of the Euler equations.» *AIAA Journal* (American Institute of Aeronautics and Astronautics) 32 (2 1994): 285-293.
9. Etcheverlepo, Adrien. «Développement de méthodes de domaines fictifs au second ordre.» Ph.D. dissertation, Université Bordeaux I, 2013.
10. Péron, Stephanie, Thomas Renaud, Ivan Mary, Christophe Benoit, et Marc Terracol. «An Immersed Boundary Method for preliminary design aerodynamic studies of complex configurations.» Dans *AIAA AVIATION Forum*, --. American Institute of Aeronautics and Astronautics, 2017.
11. Rizzi, A. «Numerical Implementation of Solid-Body Boundary Conditions for the Euler Equations.» *ZAMM* 58 (1978): 301-304.
12. Sarofeen, Christian, Ralph Noack, et Richard Kreeger. «A Non-Cut Cell Immersed Boundary Method for Use in Icing Simulations.» Dans *Aerospace Sciences Meetings*, --. American Institute of Aeronautics and Astronautics, 2012.
13. Sarthou, A., S. Vincent, et J. P. Caltagirone. «A second-order curvilinear to Cartesian transformation of immersed interfaces and boundaries. Application to fictitious domains and multiphase flows.» *Computers & Fluids* 46 (2011): 422-428.

14. Schneider, Philip J., et David H. Eberly. «Chapter 13 - computational geometry topics.» Dans *Geometric Tools for Computer Graphics*, édité par DAVID H. SCHNEIDER, 673-825. San Francisco: Morgan Kaufmann, 2003.
15. Trontin, Pierre, Ghislain Blanchard, Alexandros Kontogiannis, et Philippe Villedieu. «Description and assessment of the new ONERA 2D icing suite IGLOO2D.» *9th AIAA Atmospheric and Space Environments Conference*. 2017.

Contact Information

Pierre Lavoie, pierre.lavoie@polymtl.ca

Acknowledgments

The work presented in this paper is co-financed by the French Aerospace Lab (ONERA) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

Definitions/Abbreviations

AoA	Angle of Attack
BC	Boundary Condition
BF	Body-Fitted
BFIB	Body-Fitted Immersed Boundary
BL	Boundary Layer
IB	Immersed Boundary
IBM	Immersed Boundary Method
LE	Leading Edge
MVD	Median Volume Diameter
RANS	Reynolds Averaged Navier-Stokes
TWC	Total Water Content