DOI: 10.13009/EUCASS2017-400

3D ice accretion modeling using an integral boundary layer method

G. Blanchard[⋆†], *E. Radenac*[⋆], *N. Bempedelis*[‡], *C. Bayeux*[⋆], *P. Villedieu*[⋆]
[⋆]*Onera - The French Aerospace Lab, 2 Avenue Edouard Belin, 31055 Toulouse, France*
[‡]*Mechanical Engineering Department, University College London, Gower St, London WC1E 6BT, UK*
ghislain.blanchard@onera.fr and emmanuel.radenac@onera.fr
[†]Corresponding author

Abstract

A three-dimensional integral boundary layer code was developed to allow fast computations of boundary layer flows for 3D ice accretion modeling. The model and its numerical discretization based on a surface Finite-Volume approach are derived in the present paper. The unsteady equations of momentum deficit and kinetic energy deficit are solved until convergence is reached, preventing from specifying explicitly the stagnation point or separation line. This approach was integrated in IGLOO3D, which is the fully 3D ice accretion suite developped at ONERA, and the code was used to demonstrate the ability of the present method to predict the heat exchange and the ice accretion for a NACA0012 glaze ice configuration.

1. Introduction

Icing certification cost can be significantly reduced by developing simulation tools to evaluate the ice accretion effects for a wide range of icing conditions. 2D tools are already integrated in the process of certification. Since numerous simulations must be performed, these icing tools must have little computational cost.

The improvement of 3D tools would allow to reduce the safety margins used when performing multiple 2D simulations around 3D iced elements. But 3D icing suites are still not as efficient as 2D tools. ONERA is developing a fully 3D ice accretion suite called IGLOO3D. This tool couples codes solving the air flow, the trajectories of water droplets and the ice accretion (Messinger approach), respectively. The surface codes of IGLOO3D, such as the ice accretion solver MESSINGER3D, solve 3D systems of equation on surface grids (contrary to partially 3D icing suites solving 2D equations along streamlines).

The aerodynamic computation produces both the flowfield transporting the supercooled water droplets and the skin friction and heat transfer coefficient on the iced walls. To that end, it is possible to use a Navier-Stokes solver, which is the current option used in IGLOO3D[15] . However, this approach is time-consuming. For the 2D codes used for certification, the computational cost is far cheaper: obviously, IGLOO3D also computes an additional dimension, but more significantly, the 2D aerodynamic resolution is usually based on an inviscid (or potential) computation coupled to a boundary layer integral method, which is much cheaper than a Navier-Stokes computation.

Consequently, a 3D integral boundary layer code is being developed in order to allow a very similar approach in IGLOO3D. Currently, simplified integral boundary layer models, such as the well-known method of Thwaites,[9] are commonly employed in 2D icing suites. However, the direct extension of the simplified integral method to 3D is not straightforward.

Consequently, a general Finite-Volume formulation was employed for the solution of the system of equations. As a first step, the aforementioned discretization approach was developed and implemented in a 2D prototype of the ONERA's 2D icing suite IGLOO2D[1] . The extension and validation of this integral boundary layer method in three dimensions is presented in the present article. It must be noted that it is common practice in icing codes to weakly couple the inviscid and boundary layer codes (the effect of the boundary layer on the inviscid flow is neglected). The issue of 3D viscous-inviscid interaction methods will thus not be addressed in this paper.

The derivation of the model is presented in section 2. The Finite-Volume Method used to discretize the system will be exposed in section 3. Validation and cross-checking of the 3D code against the 2D code in 2D configurations and in 3D configurations is not addressed in this paper but interested readers can refer to the following one[2] . The use of the integral boundary layer code for ice accretion computation will be shown in section 4.

2. Derivation of the model

2.1 Brief state-of-the-art of 3D integral boundary-layer methods

The integral form of the boundary layer equations is obtained by integrating their differential form along a direction normal to the surface of the body. The obtained equations, referred to as "Integral boundary layer (IBL) equations" are differential equations whose variables are integral quantities. The information that is lost upon the process of integration has to be replaced by some set of closure relations so as to make the problem determinate.

At least for usual (and simple enough) closure relations, it can be demonstrated that the IBL equations are hyperbolic[1, 13] . This allows to define stable spatial discretization schemes. This also makes the Finite Volume Method a good choice for the resolution of the IBL equations. This method was first employed by Mughal[12] . Previous works on the 3D IBL equations mainly employed Finite Difference Method in curvilinear coordinates[3, 14] . With such methods, it is difficult to handle surface curvature discontinuities[7] and the IBL equations become very complex due to the introduction of numerous metric coefficients and geodesic curvature terms[19] . These difficulties are eliminated by the use of the Finite Volume Method.

In his first approach, Mughal solved the steady versions of the IBL equations in their conservative form. In the present paper, the unsteady version will be solved, as was also presented in Mughal's PhD thesis[13] and used in more recent articles of Drela and co-workers[7] . In addition to the simplicity of implementation, this approach prevents the need to explicitly locate the attachment line or the stagnation point prior to the computation. The derivation of the unsteady conservative IBL equations will be presented in section 2.2.

Moreover, the integral boundary layer variables are calculated by on-the-fly integration of the velocity profiles like in Mughal's PhD thesis[13] . This will allow defining closure relations for arbitrary velocity profiles if required and numerically assess some integrals for the computations of numerical fluxes for instance. The used velocity profiles will be exposed in section 2.3.

Finally, it can be noticed that Lokatt and Eller recently developed a Finite-Volume scheme for unstructured grids which they applied to the 3D IBL equations[10] . The same kind of method will be used here in order to ensure conservation on arbitrarily curved surfaces, as explained in section 3.

The expected outputs of the IBL method are mainly the skin friction and the heat transfer coefficients over the iced surfaces, which are major parameters for the ice accretion computation. The skin friction is linked to the boundary layer dynamics and will be computed directly from the dynamic integral equations exposed in section 2.2. A similar approach could be used for solving the integral heat transfer inside the boundary layer. This topic is poorly addressed in the literature. As a first attempt, Reynolds-like analogies often used in icing codes will be employed in this work (section 2.4). Consequently, only the derivation of the 3D dynamic integral boundary layer equations is addressed.

2.2 3D integral boundary layer system

The 3D unsteady incompressible differential boundary layer equations constitute the basis of the integral formulation. In a body-fitted coordinate system where z is normal to the surface, they read:

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \tag{1}$$

$$\frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xz}}{\partial z} \tag{2}$$

$$\frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \frac{1}{\rho}\frac{\partial \tau_{yz}}{\partial z} \tag{3}$$

$$0 = -\frac{1}{\rho}\frac{\partial P}{\partial z} \tag{4}$$

where $\boldsymbol{q} = \left(u_x, u_y, u_z\right)$ is the velocity vector, ρ is the density, τ is the shear stress vector $\tau_{\cdot z} = \mu\frac{\partial u_{\cdot}}{\partial z}$, μ is the dynamic viscosity of air and P is the pressure.

Like in the 2D code,[1] the integral system is based on the momentum equation and the kinetic energy equation. The latter equation is derived with the following procedure (where the subscript $()_e$ stands for the external flow quantities):

$$\left(u_{xe}^2\times \text{(eq. (1))} - 2u_x\times \text{(eq. (2))} + u_{ye}^2\times \text{(eq. (1))} - 2u_y\times \text{(eq. (3))} \right)$$

DOI: 10.13009/EUCASS2017-400

Assuming that the external velocity field is irrotational, the kinetic energy conservation equation may be written as follows:

$$\begin{aligned}&\frac{\partial u_x^2}{\partial t}+\frac{\partial u_x^3}{\partial x}+\frac{\partial u_x^2 u_y}{\partial y}+\frac{\partial u_x^2 u_z}{\partial z}=u_x\frac{\partial|\boldsymbol{q}_e|^2}{\partial x}+\frac{2u_x}{\rho}\frac{\partial\tau_{xz}}{\partial z}-\\&\frac{\partial u_y^2}{\partial t}-\frac{\partial u_x u_y^2}{\partial x}-\frac{\partial u_y^3}{\partial y}-\frac{\partial u_z u_y^2}{\partial z}+u_y\frac{\partial|\boldsymbol{q}_e|^2}{\partial y}+\frac{2u_y}{\rho}\frac{\partial\tau_{yz}}{\partial z}\end{aligned} \tag{5}$$

The 3D unsteady incompressible differential boundary layer equations are integrated in the direction normal to the wall, yielding a system of three equations involving integral quantities.

More specifically, $\int_0^\infty$ (eq. (2)) dz - $\int_0^\infty u_{xe}\times$ (eq. (1)) dz and $\int_0^\infty$ (eq. (3)) dz - $\int_0^\infty u_{ye}\times$ (eq. (1)) dz produce the transport equation (6) for the momentum deficit in the surface tangential directions.

$\int_0^\infty$ (eq. (5)) dz - $\int_0^\infty |\boldsymbol{q}_e|^2\times$ (eq. (1)) dz yields the transport equation (7) for the deficit of kinetic energy.

$$\frac{\partial\boldsymbol{M}}{\partial t}+\tilde{\nabla}\cdot\bar{\bar{\boldsymbol{T}}}=-\tilde{\nabla}\boldsymbol{q}_e\cdot\boldsymbol{M}+\frac{\boldsymbol{\tau_w}}{\rho} \tag{6}$$

$$\frac{\partial}{\partial t}\left(tr(\bar{\bar{\boldsymbol{T}}})\right)+\tilde{\nabla}\cdot\left(\boldsymbol{E}-\bar{\bar{\boldsymbol{T}}}\boldsymbol{q}_e\right)=2D-\bar{\bar{\boldsymbol{T}}}:\tilde{\nabla}\boldsymbol{q}_e+\boldsymbol{q}_e\cdot\left(\tilde{\nabla}\boldsymbol{q}_e\cdot\boldsymbol{M}-\frac{\boldsymbol{\tau_w}}{\rho}\right) \tag{7}$$

where $\tilde{\nabla}$ denotes the in-plane gradient. The involved variables are:

$$\boldsymbol{M}=\int_0^\infty(\boldsymbol{q}_e-\boldsymbol{q})\,\mathrm{d}z \qquad \text{Mass flux defect} \tag{8}$$

$$\bar{\bar{\boldsymbol{T}}}=\int_0^\infty((\boldsymbol{q}_e-\boldsymbol{q})\otimes\boldsymbol{q}))\,\mathrm{d}z \qquad \text{Momentum flux defect} \tag{9}$$

$$\boldsymbol{E}=\int_0^\infty\left(\boldsymbol{q}\left(|\boldsymbol{q}_e|^2-|\boldsymbol{q}|^2\right)\right)\,\mathrm{d}z \qquad \text{Kinetic energy defect} \tag{10}$$

$$D=\frac{1}{\rho}\int_0^\infty\boldsymbol{\tau_w}\cdot\frac{\partial\boldsymbol{q}}{\partial z}\,\mathrm{d}z \qquad \text{Dissipation integral} \tag{11}$$

$$\boldsymbol{\tau_w}=\begin{cases}\mu\frac{\partial\boldsymbol{q}}{\partial z}\Big|_{z=0}, & \text{laminar regime}\\ \mu\frac{\partial\boldsymbol{q}}{\partial z}\Big|_{z=0}-\rho\overline{\boldsymbol{q}'q'_z}, & \text{turbulent regime}\end{cases} \qquad \text{Shear stress vector} \tag{12}$$

The system is expressed in a tensorial form, making it independent of the coordinate system used. The notations are quite similar to the ones of Drela,[7] although Drela accounts for compressibility and for additional equations. The link between these notations and more usual displacement $\delta_{1.}$ and momentum $\theta_{..}$ integral boundary layer thicknesses is (expressed in a global coordinate system $(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$):

$$\boldsymbol{M}=\begin{pmatrix}|\boldsymbol{q}_e|\delta_{1X}\\|\boldsymbol{q}_e|\delta_{1Y}\\|\boldsymbol{q}_e|\delta_{1Z}\end{pmatrix}\qquad\bar{\bar{\boldsymbol{T}}}=|\boldsymbol{q}_e|^2\begin{pmatrix}\theta_{XX}&\theta_{XY}&\theta_{XZ}\\\theta_{YX}&\theta_{YY}&\theta_{YZ}\\\theta_{ZX}&\theta_{ZY}&\theta_{ZZ}\end{pmatrix} \tag{13}$$

2.3 Closure relations

Since the closure relations of the literature have initially been derived for 2D boundary layers and the effect of the transverse flow is expected to be low, closure is provided in a local coordinate system aligned with the external streamlines,

$$\boldsymbol{s}=\frac{\boldsymbol{q}_e}{|\boldsymbol{q}_e|},\qquad\boldsymbol{c}=-\frac{\boldsymbol{s}\times\boldsymbol{n}}{|\boldsymbol{s}\times\boldsymbol{n}|},\qquad\boldsymbol{n}=\boldsymbol{n}_w \tag{14}$$

where $\boldsymbol{n_w}$ is the vector normal to the surface. The in-plane velocity thus reads:

$$\tilde{\boldsymbol{q}}(\eta)=|\boldsymbol{q}_e|(\hat{u_s}(\eta)\boldsymbol{s}+\hat{u_c}(\eta)\boldsymbol{c}) \tag{15}$$

where $\eta=z/\delta$. δ is an estimate of the boundary layer thickness.

DOI: 10.13009/EUCASS2017-400

2.3.1 Laminar regime

The closure relations used for the velocity profiles in laminar regime read:

$$\hat{u_s}(\eta) = 1 - [1 + a_s\eta]\,(1-\eta)^{p_s-1} \tag{16}$$

$$\hat{u_c}(\eta) = a_c\eta\,(1 - \hat{u_s}(\eta)) \tag{17}$$

For the streamwise velocity $\hat{u_s}$, it is possible to express a_s and p_s as functions of the shape factor $H = \frac{\delta_{1s}}{\theta_{ss}}$.[1] For instance:

$$a_s(H) = \sqrt{p_s(H)^2 - p_s(H)(p_s(H)+1)Hg(H)} - 1 \tag{18}$$

The function $g(H)$ was derived by Cousteix[4] to fit the skin friction coefficient obtained for the Falkner-Skan solutions. $p_s(H)$ and $g(H)$ are given in appendix.

At the wall ($\eta = 0$), $\hat{u_s}$ satisfies the no-slip condition and the derivative of the velocity is consistent with the friction obtained in the self-similar Falkner-Skan solutions. At the edge of the boundary layer, $\hat{u_s}(1) = 1$ and the derivatives of the velocity are vanishing up to (p_s-2)th order. Moreover, both δ_{1s} and θ_{ss} are consistent with this velocity profile.

A 3D boundary layer is characterized by the presence of a cross-stream velocity component, due to transverse pressure gradients. For the crosswise velocity $\hat{u_c}$, the profile derived by Mughal[13] for unidirectional crossflow calculations is employed. From the calculation of $\delta_{1c} = \delta \int_0^1 -\hat{u_c}(\eta)\mathrm{d}\eta$ (since $u_{ce} = 0$ in the streamline-aligned coordinate system), a_c is linked to the solved variables through:

$$a_c = -\frac{\delta_{1c}}{\delta_{1s}}\frac{Hg(H)}{b(H)}\frac{p_s(H)\,(p_s(H)+1)}{1 + \dfrac{2a_s(H)}{p_s(H)+2}} \tag{19}$$

where $b(H)$ is given in appendix and the following relation has been used to express δ: $\delta = \delta_{1s}\frac{b(H)}{Hg(H)}$.[1] $\hat{u_c}$ then satisfies the no-slip condition at the wall and $\hat{u_c} = 0$ at the edge of the boundary layer. The derivatives of the velocity are vanishing at this location up to (p_s-2)th order. Finally, δ_{1c} and θ_{cc} are consistent with this velocity profile.

The two profiles are used to compute the dissipation integral (equation (11)) and the shear stress vector (equation (12)).

2.3.2 Turbulent regime

The streamwise velocity profile employed for the turbulent regime is based on the work of Tai,[17] whereas the crosswise component is neglected, as a first attempt:

$$\hat{u_s}(\eta) = \eta^{(H-1)/2} \tag{20}$$

$$\hat{u_c}(\eta) = 0 \tag{21}$$

It is worth mentioning that more evolved approaches have been developed to model turbulent boundary layers (velocity profiles of Swafford[16] or Drela's approach of transporting turbulent shear stress[7] for instance). In particular, the relations used for the current paper do not allow the turbulent boundary layer separation. Those approaches could be assessed in the future.

Besides, empirical relations are used for the friction and dissipation coefficients. Two options are available for the streamwise skin friction, the relation proposed by Ludwieg and Tillmann:[11]

$$C_{fs} = \frac{\tau_{ws}}{\rho|\boldsymbol{q}_e|^2/2} = 0.246 \times 10^{-0.678H} {Re_{\theta_{ss}}}^{-0.268}, \tag{22}$$

which is assumed valid for $Re_{\theta_{ss}} > 1200$, or the one proposed by White:[20]

$$C_{fs} = \frac{\tau_{ws}}{\rho|\boldsymbol{q}_e|^2/2} = \frac{0.3\ \mathrm{e}^{-1.33H}}{(\log Re_{\theta_{ss}})^{1.74+0.31H}}. \tag{23}$$

The dissipation coefficient is given by Drela's relation:[5]

$$D = |\boldsymbol{q}_e|^3\frac{H^*}{2}\left[\frac{C_{fs}}{6}\left(\frac{4}{H} - 1\right) + 0.03\left(\frac{H-1}{H}\right)^3\right], \tag{24}$$

DOI: 10.13009/EUCASS2017-400

where:

$$H^* = 1.505 + \frac{4}{Re_{\theta_{ss}}} + \left(0.165 - \frac{1.6}{\sqrt{Re_{\theta_{ss}}}}\right)\frac{(H_0 - H)^{1.6}}{H} \qquad \text{if } H < H_0 \tag{25}$$

$$H^* = 1.505 + \frac{4}{Re_{\theta_{ss}}} + (H - H_0)^2\left(\frac{0.04}{H} + 0.007\frac{\ln Re_{\theta_{ss}}}{\left(H - H_0 + \frac{4}{\ln Re_{\theta_{ss}}}\right)^2}\right) \qquad \text{if } H > H_0 \tag{26}$$

and

$$H_0 = 4 \qquad \text{if } Re_{\theta_{ss}} < 400 \tag{27}$$

$$H_0 = 3 + \frac{400}{Re_{\theta_{ss}}} \qquad \text{if } Re_{\theta_{ss}} \geq 400 \tag{28}$$

Again, the crosswise components are cancelled as a first approximation.

Regarding prediction of transition, the local criterion of Drela[6] was implemented for flows on smooth walls.

$$Re_{\theta_{ss}T} = 155 + 89\left[0.25\tanh(\frac{10}{H-1} - 5.5) + 1\right]\widetilde{n}^{1.25} \tag{29}$$

$$\widetilde{n} = -8.43 - 2.4\ln\left(\frac{\tau'}{100}\right) \tag{30}$$

$$\tau' = 2.7\tanh\left(\frac{Tu}{2.7}\right), \tag{31}$$

where Tu is the turbulence rate (in %). However, the code is expected to be run mostly on rough walls. The criterion widely used in icing suites[18] is thus also proposed: transition occurs for a roughness Reynolds number $\mathrm{Re}_k = \frac{k_s|\boldsymbol{q}_e|}{\nu_e}$ larger than 600, where k_s is the equivalent sand grain roughness height and ν_e is the kinematic viscosity of air.

2.3.3 Rotation to the global coordinate system

Once the closure relations have been computed, the velocity profiles are rotated to the global coordinate system $((\boldsymbol{q})_{LCS} \rightarrow (\boldsymbol{q})_{GCS})$ by using the rotation matrix $\bar{\bar{\boldsymbol{R}}}$:

$$(\boldsymbol{q})_{GCS} = \begin{pmatrix} u_X \\ u_Y \\ u_Z \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}\cdot\boldsymbol{s} & \boldsymbol{X}\cdot\boldsymbol{c} & \boldsymbol{X}\cdot\boldsymbol{n} \\ \boldsymbol{Y}\cdot\boldsymbol{s} & \boldsymbol{Y}\cdot\boldsymbol{c} & \boldsymbol{Y}\cdot\boldsymbol{n} \\ \boldsymbol{Z}\cdot\boldsymbol{s} & \boldsymbol{Z}\cdot\boldsymbol{c} & \boldsymbol{Z}\cdot\boldsymbol{n} \end{pmatrix}\begin{pmatrix} u_s \\ u_c \\ u_n \end{pmatrix} = \bar{\bar{\boldsymbol{R}}}(\boldsymbol{q})_{LCS} \tag{32}$$

In a similar manner, the primary variables are transformed from the global to the local coordinate system (in order to construct the velocity profiles) as follows:

$$(\boldsymbol{M})_{LCS} = \bar{\bar{\boldsymbol{R}}}^T(\boldsymbol{M})_{GCS} \tag{33}$$

2.3.4 Calculation of the shape factor

The proposed closure of the problem requires knowledge of the following parameters: $H, \delta_{1s}, \delta_{1c}$. δ_{1s} and δ_{1c} are easily obtained from the solved variables $(\boldsymbol{M})_{GCS}$ by the rotation described earlier. However, in 3D, the shape factor $H = \frac{\delta_{1s}}{\theta_{ss}}$ is not explicitly present in the formulation since θ_{ss} is only linked to the solved variables through:

$$tr(\bar{\bar{\boldsymbol{T}}}) = |\boldsymbol{q}_e|^2(\theta_{ss} + \theta_{cc}) \tag{34}$$

Thus, an iterative process regarding the calculation of the shape factor H has to follow. To that end, the definition of $\theta_{cc} = \delta\int_0^1 -\hat{u_c}(\eta)^2 \mathrm{d}\eta$ is used:

$$\frac{1}{H} = \frac{tr(\bar{\bar{\boldsymbol{T}}})}{|\boldsymbol{q}_e|^2\delta_{1s}} - \frac{\theta_{cc}}{\delta_{1s}} = \frac{tr(\bar{\bar{\boldsymbol{T}}})}{|\boldsymbol{q}_e|^2\delta_{1s}} + \frac{\delta\int_0^1 \hat{u}_c^2\,\mathrm{d}\eta}{\delta_{1s}} \tag{35}$$

Since $\hat{u}_c$ depends on H and the primary variables, it is now possible to obtain H by solving equation (35) in an iterative way. For instance, for the case of the laminar flow, equation (35) to be solved becomes:

$$\frac{1}{H} = \frac{tr(\bar{\bar{\boldsymbol{T}}})}{|\boldsymbol{q}_e|^2\delta_{1s}} + \frac{b(H)a_c(H)^2}{Hg(H)p_s(H)(2p_s(H)-1)(2p_s(H)+1)}\left(1 + \frac{3a_s(H)}{p_s(H)+1} + \frac{6a_s(H)^2}{(p_s(H)+1)(2p_s(H)+3)}\right) \tag{36}$$

2.4 Computation of heat transfer

The approach used here is commonly used in 2D ice accretion codes, as shown by Gent et al.[8] It is also the way heat transfer coefficients are inferred from dynamic data in the ONERA's 2D code IGLOO2D.[18] The heat transfer coefficient h_c is inferred from a Reynolds-like analogy:

$$h_c = \mathrm{St}\rho_e c_{p_e} |\boldsymbol{q}_e| \tag{37}$$

where ρ_e is the air density, c_{p_e} is the specific heat capacity at constant pressure for air and the Stanton number St depends on the regime.

For a laminar flow, the equation of Smith and Spalding is employed. It links the heat transfer coefficient to the evolution of the velocity at the edge of the boundary layer along a streamline (s is the wrap distance from the attachment line):

$$\mathrm{St} = \frac{0.2926\sqrt{\nu_e}}{|\boldsymbol{q}_e|\mathrm{Pr}}\sqrt{|\boldsymbol{q}_e|^{2.87} / \int_0^s |\boldsymbol{q}_e|^{1.87} ds} \tag{38}$$

where Pr is the Prandtl number.

The heat transfer coefficient in turbulent rough wall conditions is obtained with:

$$\mathrm{St} = \frac{C_{fs,r}/2}{\mathrm{Pr}_t + \sqrt{C_{fs,r}/2}/\mathrm{St}_k} \tag{39}$$

where $\mathrm{St}_k = 1.92\mathrm{Pr}^{-0.8}\mathrm{Re}_k^{-0.45}$, Pr_t is the turbulent Prandtl number and $\mathrm{Re}_k = \dfrac{k_s u_\tau}{\nu_e}$. u_τ is the friction velocity. $C_{fs,r}$ is the streamwise skin friction coefficient expected on a rough wall, which is computed from the streamwise momentum thickness computed by the integral method:

$$\frac{C_{fs,r}}{2} = \frac{0.168}{\left(\log\left(864\theta_{ss}/k_s + 2.568\right)\right)^2} \tag{40}$$

3. Finite-Volume resolution

The system of equations (6), (7) can be written in the following manner:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \tilde{\nabla} \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}) \tag{41}$$

where the components of the flux vector $\boldsymbol{F}$ can be written:

$$\boldsymbol{F} = \begin{pmatrix} \bar{\bar{\boldsymbol{T}}} \\ \left(\boldsymbol{E} - \bar{\bar{\boldsymbol{T}}}\boldsymbol{q}_e\right)^T \end{pmatrix} = \begin{pmatrix} |\boldsymbol{q}_e|^2 \bar{\bar{\boldsymbol{\theta}}} \\ |\boldsymbol{q}_e|^3 \left(\boldsymbol{\delta}_3 - \bar{\bar{\boldsymbol{\theta}}}\dfrac{\boldsymbol{q}_e}{|\boldsymbol{q}_e|}\right) \end{pmatrix}$$

where $\bar{\bar{\boldsymbol{\theta}}}$ and $\boldsymbol{\delta}_3$ are defined as follows:

$$\bar{\bar{\boldsymbol{\theta}}} = \int_0^\infty \left(\left(\frac{\boldsymbol{q}_e}{|\boldsymbol{q}_e|} - \frac{\boldsymbol{q}}{|\boldsymbol{q}_e|}\right) \otimes \frac{\boldsymbol{q}}{|\boldsymbol{q}_e|}\right) \tag{42}$$

$$\boldsymbol{\delta}_3 = \int_0^\infty \left(\frac{\boldsymbol{q}}{|\boldsymbol{q}_e|}\left(1 - \frac{|\boldsymbol{q}|^2}{|\boldsymbol{q}_e|^2}\right)\right) \tag{43}$$

Following an unstructured-mesh Finite-Volume formulation, the system is integrated over a cell Ω_i between time t^n and t^{n+1} ($\Delta t^n = t^{n+1} - t^n$):

$$\int\limits_{\Omega_i} \boldsymbol{U}(t^{n+1}) \mathrm{d}\Omega - \int\limits_{\Omega_i} \boldsymbol{U}(t^n)\, \mathrm{d}\Omega = - \sum_{j \in \mathcal{N}(i)} \int\limits_{t^n}^{t^{n+1}} \int\limits_{\Gamma_{ij}} \boldsymbol{F} \cdot \boldsymbol{n}_{ij}\, \mathrm{d}\Gamma \mathrm{d}t + \int\limits_{t^n}^{t^{n+1}} \int\limits_{\Omega_i} \boldsymbol{S}\, \mathrm{d}\Omega \mathrm{d}t \tag{44}$$

where the divergence theorem has been used for the transformation of the volume integral to surface integral, Γ_{ij} is the edge shared by cell Ω_i with one of its neighbors Ω_j. The vector $\boldsymbol{n}_{ij}$ is the local unit normal to the edge Γ_{ij} in the tangent plane of cell Ω_i and pointing outward.

Equation (44) can be reshaped in the following discrete form:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t^n}{|\Omega_i|} \sum_{j \in \mathcal{N}(i)} \boldsymbol{F}_{ij} |\Gamma_{ij}| + \Delta t^n \boldsymbol{S}_i \tag{45}$$

where $\boldsymbol{U}_i^n$ is the discrete mean value of the unknowns in cell Ω_i at time t^n, $\boldsymbol{F}_{ij}$ is the numerical flux at the edge Γ_{ij} and $\boldsymbol{S}_i$ is the discrete source term in cell Ω_i:

$$\boldsymbol{U}_i^n \overset{\text{def}}{\simeq} \frac{1}{|\Omega_i|} \int_{\Omega_i} \boldsymbol{U}(t^n) \mathrm{d}\Omega \tag{46}$$

$$\boldsymbol{F}_{ij} \overset{\text{def}}{\simeq} \frac{1}{\Delta t^n} \frac{1}{|\Gamma_{ij}|} \int_{t^n}^{t^{n+1}} \int_{\Gamma_{ij}} \boldsymbol{F} \cdot \boldsymbol{n}_{ij} \mathrm{d}\Gamma \mathrm{d}t \tag{47}$$

$$\boldsymbol{S}_i \overset{\text{def}}{\simeq} \frac{1}{\Delta t^n} \frac{1}{|\Omega_i|} \int_{t^n}^{t^{n+1}} \int_{\Omega_i} \boldsymbol{S} \, \mathrm{d}\Omega \mathrm{d}t \tag{48}$$

To complete the discretization of the continuous model, the spatial and temporal schemes used to express the numerical flux $\boldsymbol{F}_{ij}$ are detailed in the following paragraphs. Regarding the source term $\boldsymbol{S}_i$, the computation of the velocity gradient is performed with a linear least-squares method (first order in space).

3.1 Spatial scheme

Conservation In the employed Finite Volume formulation, the fluxes' contribution to the residuals has to be expressed in the plane of every associated cells to obtain a conservative scheme. However, in the general case of an embedded surface, two neighboring cells may not be co-planar. Therefore, two different expressions for the flux through the shared edge are required as, in this general case, the following relationship is only true for a planar mesh:

$$\boldsymbol{F}_{ji} = -\boldsymbol{F}_{ij} \tag{49}$$

In the present study, the conservative character of the equations is preserved by employing a set of rotations, similarly to what is proposed by Lokatt and Eller.[10] A schematic depicting the involved variables is provided in figure 1.

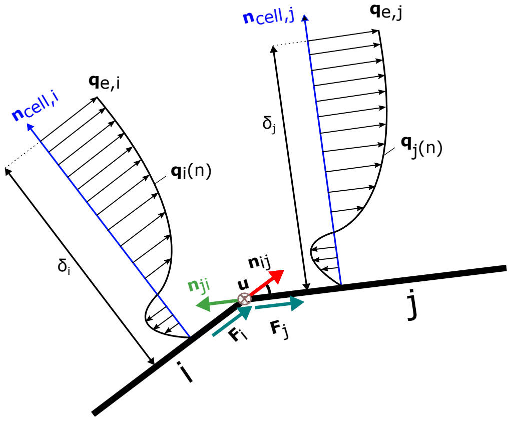


Figure 1: Schematic of fluxes' calculation method in a 2D cut of an embedded surface.

Therefore, we introduce the rotation matrix $\bar{\bar{\boldsymbol{Q}}}_{ij}$ which is used to express tensorial quantities from the plane associated to cell Ω_i in the plane associated to cell Ω_j. The process to build the matrix was adapted from the method described in Lokatt[10] (where the fluxes are expressed in the local coordinate system contrary to the present method) and consists of the following steps:

- Definition of the axis of rotation, $\boldsymbol{u}_{ij} = \dfrac{\boldsymbol{n}_{cell,i} \times \boldsymbol{n}_{cell,j}}{|\boldsymbol{n}_{cell,i} \times \boldsymbol{n}_{cell,j}|}$

DOI: 10.13009/EUCASS2017-400

- Definition of the angle of rotation, $\alpha_{ij} = \cos^{-1}\left(\boldsymbol{n}_{cell,i} \cdot \boldsymbol{n}_{cell,j}\right)$
- Definition of the rotation matrix $\bar{\bar{\boldsymbol{Q}}}_{ij}$ around the previously defined axis,

$$\bar{\bar{\boldsymbol{Q}}}_{ij} = \cos\alpha_{ij}\bar{\bar{\boldsymbol{I}}} + \sin\alpha_{ij}\left[\boldsymbol{u}_{ij}\right]_{\times} + \left(1 - \cos\alpha_{ij}\right)\boldsymbol{u}_{ij} \otimes \boldsymbol{u} \tag{50}$$

where $\left[\boldsymbol{u}_{ij}\right]_{\times}$ is the cross product matrix of $\boldsymbol{u}_{ij}$, $\otimes$ is the tensor product and $\bar{\bar{\boldsymbol{I}}}$ is the identity matrix.

Then, the following relation holds:

$$\boldsymbol{F}_{ji} = -\bar{\bar{\boldsymbol{Q}}}_{ij}\boldsymbol{F}_{ij} \quad \text{and} \quad \boldsymbol{F}_{ij} = -\bar{\bar{\boldsymbol{Q}}}_{ji}\boldsymbol{F}_{ji} = -\bar{\bar{\boldsymbol{Q}}}_{ij}^{T}\boldsymbol{F}_{ji} \tag{51}$$

Flux expression Based on the formulation employed in the 2D version of the code,[1] a first order upwind scheme is used in order to ensure stability. The upwinding of the numerical flux is based on the edge velocity $\boldsymbol{q}_{e,ij}$ which is defined in the tangential plane associated to cell Ω_i by:

$$\boldsymbol{q}_{e,ij} = \frac{1}{\omega_i + \omega_j}\left(\omega_i \boldsymbol{q}_{e,i} + \omega_j \bar{\bar{\boldsymbol{Q}}}_{ji}\boldsymbol{q}_{e,j}\right) \tag{52}$$

where ω_i and ω_j are inverse distance weighting factors that rely only on the mesh geometry:

$$\omega_i = |G_i G_{ij}|^{-1}, \quad \omega_j = |G_j G_{ji}|^{-1} \tag{53}$$

where G_i, G_j and $G_{ij}(= G_{ji})$ are the centers of gravity of cells Ω_i, Ω_j and edge Γ_{ij} respectively.

Then, the numerical flux is calculated with upstream values according to the sign of the face velocity:

- If $(\boldsymbol{q}_{e,ij} \cdot \boldsymbol{n}_{ij}) \geq 0$ then :

$$\boldsymbol{F}_{ij} = \begin{pmatrix} |\boldsymbol{q}_{e,ij}|^2 \bar{\bar{\boldsymbol{\theta}}}_i \boldsymbol{n}_{ij} \\ |\boldsymbol{q}_{e,ij}|^3 \left(\boldsymbol{\delta_3}_i - \bar{\bar{\boldsymbol{\theta}}}_i \dfrac{\boldsymbol{q}_{e,i}}{|\boldsymbol{q}_{e,i}|}\right) \cdot \boldsymbol{n}_{ij} \end{pmatrix} \quad \text{and} \quad \boldsymbol{F}_{ji} = -\bar{\bar{\boldsymbol{Q}}}_{ij}\boldsymbol{F}_{ij} \tag{54}$$

- else:

$$\boldsymbol{F}_{ji} = \begin{pmatrix} |\boldsymbol{q}_{e,ji}|^2 \bar{\bar{\boldsymbol{\theta}}}_j \boldsymbol{n}_{ji} \\ |\boldsymbol{q}_{e,ji}|^3 \left(\boldsymbol{\delta_3}_j - \bar{\bar{\boldsymbol{\theta}}}_j \dfrac{\boldsymbol{q}_{e,j}}{|\boldsymbol{q}_{e,j}|}\right) \cdot \boldsymbol{n}_{ji} \end{pmatrix} \quad \text{and} \quad \boldsymbol{F}_{ij} = -\bar{\bar{\boldsymbol{Q}}}_{ji}\boldsymbol{F}_{ji} \tag{55}$$

where $\bar{\bar{\boldsymbol{\theta}}}$ and $\boldsymbol{\delta_3}$ are respectively computed by numerical integration of (42) and (43) with a Simpson method in the upwind cell.

As also mentioned in Bayeux's paper,[1] the present scheme cannot capture the separation of the boundary layer. This issue will be addressed in future works.

Treatment of the stagnation point It must be mentioned that it was identified in the 2D code that the discretization of the equations (and more specifically the one of kinetic energy conservation) needs to be corrected[1] to have a better numerical treatment of the stagnation point. This correction consists of an additional source term aimed at recovering the consistency of the numerical method in the vicinity of the stagnation point. It was adapted as follows from the work of Bayeux:[1]

$$S_{stag} = \frac{\boldsymbol{E} - \bar{\bar{\boldsymbol{T}}}\boldsymbol{q}_e}{|\boldsymbol{q}_e|^3} \cdot \left[\tilde{\nabla} \cdot \left(|\boldsymbol{q}_e|^3 \bar{\bar{\boldsymbol{I}}}\right) - 3|\boldsymbol{q}_e|^2\, \tilde{\nabla} \cdot \left(|\boldsymbol{q}_e| \bar{\bar{\boldsymbol{I}}}\right)\right], \tag{56}$$

which has been discretized consistently with the employed numerical flux:

$$S_{stag,i} = \frac{1}{|\Omega_i|}\frac{\boldsymbol{E}_i - \bar{\bar{\boldsymbol{T}}}_i \boldsymbol{q}_{e,i}}{|\boldsymbol{q}_{e,i}|^3} \cdot \left[\sum_{j \in \mathcal{N}(i)} \left(|\boldsymbol{q}_{e,ij}|^3 |\Gamma_{ij}| \boldsymbol{n}_{ij}\right) - 3|\boldsymbol{q}_{e,i}|^2 \sum_{j \in \mathcal{N}(i)} \left(|\boldsymbol{q}_{e,ij}||\Gamma_{ij}|\boldsymbol{n}_{ij}\right)\right] \tag{57}$$

The current formulation was derived for streamwise aligned meshes. It will be fully adapted to unstructured meshes in the future.

DOI: 10.13009/EUCASS2017-400

3.2 Temporal scheme

An explicit Euler method has been employed for the discretization of the transport term while the source terms can be implicit, in an attempt to increase the stability of the method:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t^n}{|\Omega_i|} \sum_{j\in\mathcal{N}(i)} \boldsymbol{F}_{ij}^n |\Gamma_{ij}| + \Delta t_i^n \boldsymbol{S}_i^{n+1} \tag{58}$$

After linearization of the source term, the solution is given by:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n + \left[\boldsymbol{I} - \Delta t_i^n \nabla_U \boldsymbol{S}_i^n\right]^{-1} \Delta t_i^n \left(-\frac{1}{|\Omega_i|} \sum_{j\in\mathcal{N}(i)} \boldsymbol{F}_{ij}^n |\Gamma_{ij}| + \boldsymbol{S}_i^n \right) \tag{59}$$

where $\nabla_U \boldsymbol{S}$ is the jacobian matrix of the source term which is calculated by numerical differenciation.

In equation (59), a local time stepping approach is used to obtain a faster convergence to the steady-state solution. The timestep value Δt_i is computed in each cell Ω_i to satisfy the following empirical CFL condition based on the convective time scale:

$$\Delta t_i < CFL \, \frac{\Delta x_i}{|\boldsymbol{q}_{e,i}|} \tag{60}$$

where Δx_i is the characteristic cell length given by:

$$\Delta x_i = \frac{|\Omega_i|}{\sum\limits_{j\in\mathcal{N}_i} |\Gamma_{ij}|} \tag{61}$$

4. Ice accretion computation using the 3D Integral Boundary Layer code

The 3D integral boundary layer code, BLIM3D, was included in ONERA's 3D icing suite, IGLOO3D. This means that BLIM3D is fed by an inviscid code compatible with IGLOO3D structure, here elsA. Then it provides the heat transfer coefficient and skin friction to the 3D Messinger code MESSINGER3D.

A NACA0012 glaze ice case was used to demonstrate the capabilities of BLIM3D to be used for ice accretion computation. A glaze ice case was used because the shape of this kind of rather warm ice depends greatly on the computation of the heat transfer in the boundary layer. The conditions of the case are given in table 1. Contrary to a previous study,[2] the 2D flow has a positive angle of attack and it is a good test case to assess the code to predict automatically the separation line position. The flow is 2D but a 3D unstructured grid was generated as shown in figure 2. The mesh is composed of 69452 triangles, and a local refinement is used in the vicinity of the leading edge.

	AOA	**Chord**	**P$_\infty$**	**T$_\infty$**	**M$_\infty$**	**LWC**	**MVD)**	**Δt**
	(^{o})	**(m)**	**(Pa)**	**(K)**		**(g/m^3)**	**(μm)**	(s)
NACA0012 airfoil	4	0.53	95610	266.45	0.1770	1.3	20	480

Table 1: Conditions of the test-case, AOA: angle of attack; P$_\infty$, T$_\infty$, M$_\infty$: static pressure, static temperature and Mach number of incoming airflow; LWC: Liquid Water Content of incoming flow; MVD: Median Volume Diameter of supercooled water droplets; Δt: accretion time.

The rough models of BLIM3D were used for laminar-turbulent transition and heat transfer (the roughness height is 0.53 mm) and, as shown in table 2, the closure relation proposed by White (equation 23) for skin friction was assessed.

Table 2 shows that several computations were performed on the NACA0012 case. The ONERA's 2D icing code IGLOO2D was run with both predictor (a single aerodynamics - droplet - accretion loop is made) and predictor-corrector approach (two loops are made and the grid is updated to account for the effect of the ice shape on the airflow and the droplet trajectories). The in-house solvers STRMESH2D, EULER2D, TRAJL2D and MESSINGER2D[18] were used for the structured mesh, the inviscid flow, the Lagrangian trajectography and the ice acrretion computations, respectively. Two methods were assessed for the boundary layer simulation. First, the usual approach was employed.

DOI: 10.13009/EUCASS2017-400

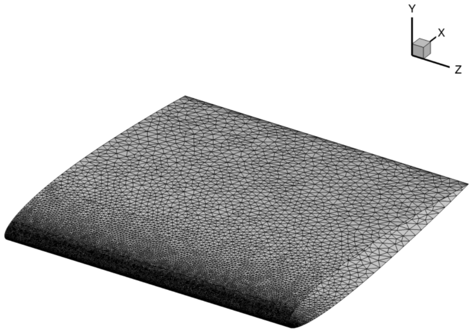


Figure 2: NACA0012 surface mesh for BLIM3D

Code	**Gas flow**	**Droplet trajectories**	**Friction closure**	h_c **closure**	**Grid**	**Predictor-corrector**	**Output**
IGLOO3D	elsA RANS	SPIREE		section 2.4	3D structured 1312 elements	Predictor	Boundary layer, ice
IGLOO3D	elsA Euler +BLIM3D	SPIREE	White	section 2.4	3D unstructured 69452 elements	Predictor	Boundary layer, ice
IGLOO2D	EULER2D +SIM2D	TRAJL2D		section 2.4	2D structured 128 elements	Predictor-Corrector	Boundary layer, ice
IGLOO2D	EULER2D +BLIM2D	TRAJL2D	Ludwieg-Tillmann	section 2.4	2D structured 128 elements	Predictor-Corrector	Boundary layer, ice
IGLOO2D	EULER2D +BLIM2D	T RAJL2D	White	section 2.4	2D structured 128 elements	Predictor-Corrector	Boundary layer, ice

Table 2: Various computations performed on the NACA0012 glaze ice case.

It consists of a simplified integral metod (code SIM2D). Second, the 2D version of the method presented here[1] was tested. Both the closure relations of White (equation 23) and Ludwieg-Tillman (equation 22) were assessed in 2D.

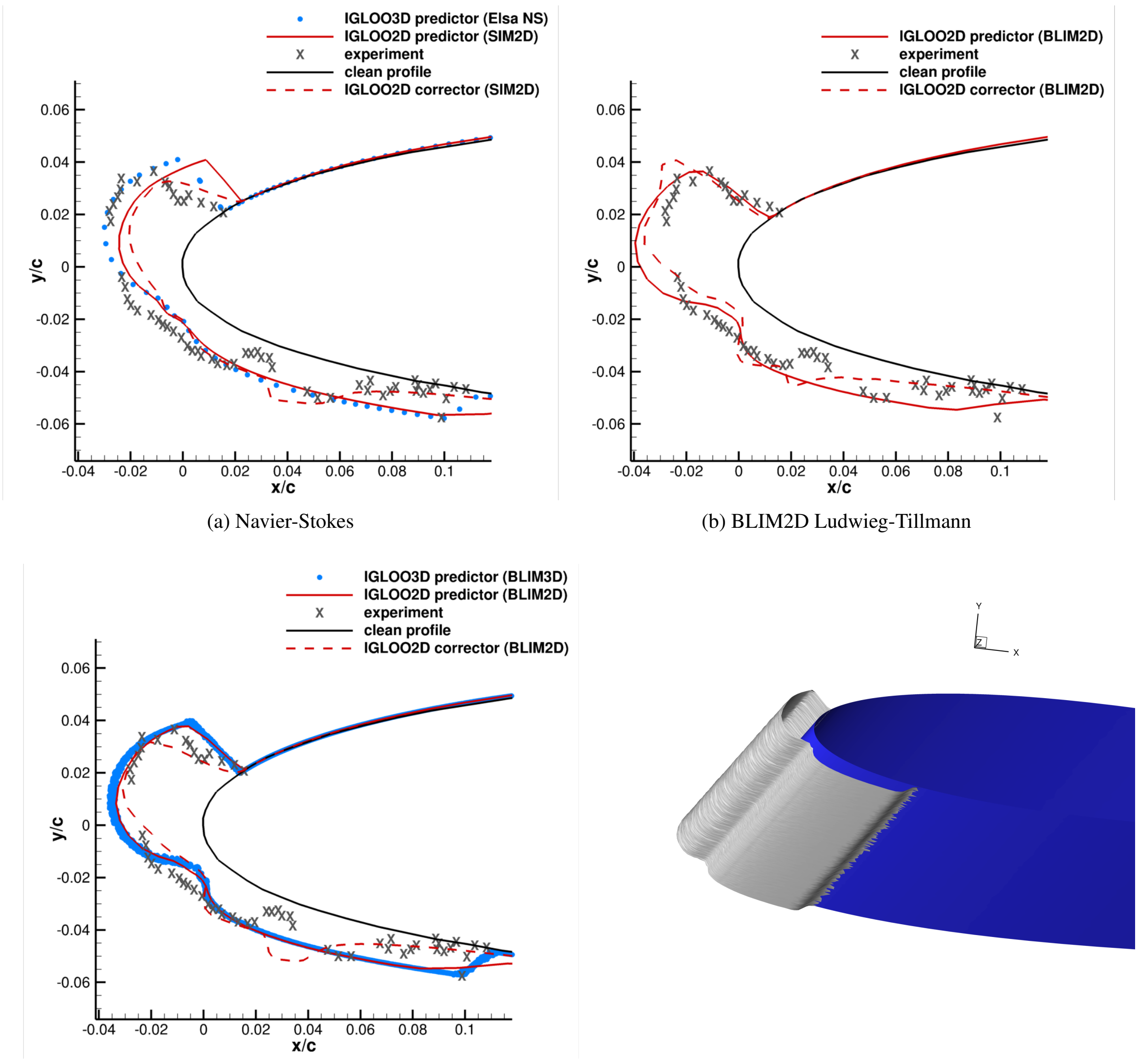
In addition, the same test-case was also run with the usual approach of IGLOO3D, consisting of computing the airflow with elsA code (RANS approach on a structured grid), and the droplet trajectories with the Eulerian code SPIREE (monodisperse approach here). For this approach, several methods are available to compute the heat transfer coefficient. For this paper, the relations of section 2.4, fed by the smooth-wall momentum thickness provided by elsA, were used. This approach is very similar to IGLOO2D and thus often produces satisfactory results on 2D configurations. For both the RANS and BLIM3D approaches, IGLOO3D was used in predictor mode. Finally, computations with the ONERA's 2D Prandtl boundary layer solver CLICET were also performed as a basis for the validation of the boundary layer results.

Figure 3 gathers the results produced by the various approaches. The horns are globally well captured with IGLOO2D and IGLOO3D for this test case. The corrector phase does not bring a major improvement for this test-case compared to the predictor step. This justifies the use of a single predictor step with IGLOO3D.

Regarding the IGLOO3D computations, the usual Navier-Stokes approach produces an ice shape which is very similar to IGLOO2D (SIM2D), despite a small underestimation of the horn close to the leading edge with the 2D approach (figure 3a). This can be explained by the smaller heat transfer coefficient obtained in 2D (see figure 4a, comparison between "IGLOO2D SIM2D" and "IGLOO3D elsa NS"). Consequently, the ice is less cooled, the runback is thus enhanced and the solidification lessened.

The influence of the closure model retained for the skin friction was studied with BLIM2D. For this icing condition, White model (figure 3c) gives an iced shape which is a little bit closer to the experimental data than the one given by the Ludwieg-Tillmann model (figure 3b). Nevertheless, the superiority of the White model is not as noticeable as it was observed in an other configuration[2] . Figure 4a shows that Ludwieg-Tillman model leads to an over-estimated heat transfer coefficient compared with the elsA result close to the stagnation point. This can be explained by the fact that this model is not valid for low $Re_{\theta_{ss}}$ and, close to the stagnation point, the rough walls model forces the transition

DOI: 10.13009/EUCASS2017-400



(a) Navier-Stokes

(b) BLIM2D Ludwieg-Tillmann

(c) BLIM3D White

(d) BLIM3D White

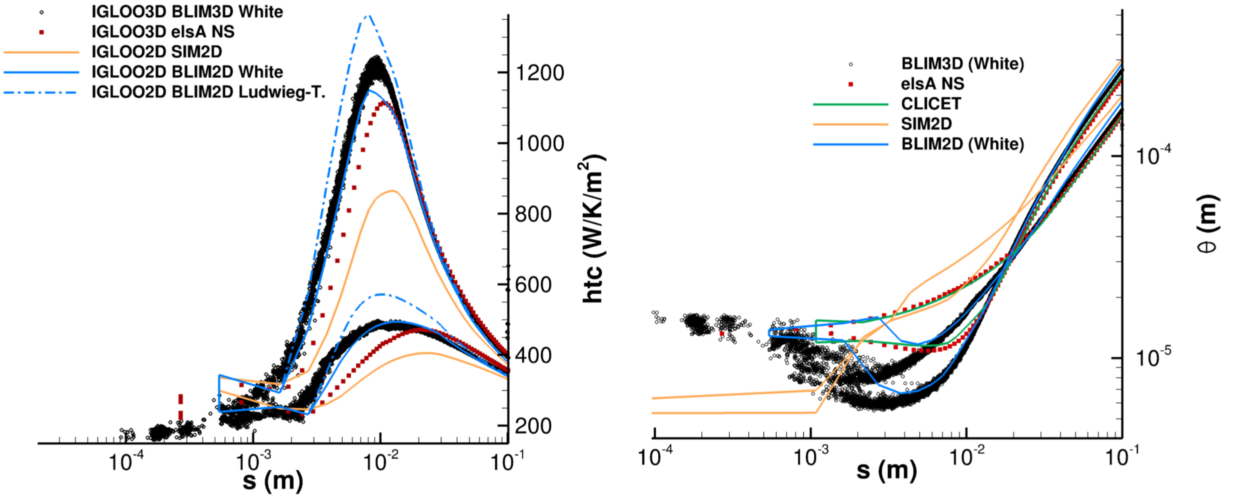
Figure 3: IGLOO3D computations of the ice shape with Navier-Stokes or BLIM3D approaches

to occur very rapidly in a region where the order of magnitude of $Re_{\theta_{ss}}$ is only a few tens. With White model, the agreement becomes much better between BLIM3D and IGLOO3D used along with elsA.

It is worth recalling that the heat transfer coefficient is dependent on the momentum thickness calculated by the boundary layer code. Figure 4b shows that BLIM3D and its 2D counterpart BLIM2D produce very similar results, although the BLIM3D results are a little noisy due to the unstructured grid and BLIM2D results are less resolved due to the coarser grid. The agreement between BLIM2D, BLIM3D, elsA and CLICET is rather satisfactory for low wrap distance s and for $s > 2 \times 10^{-2} m$ but it is worse for intermediate values of s, which remains to be investigated. SIM2D produces larger momentum thickness values in a large range of s and a value which is too low at the stagnation point. Small discrepancies are often observed in turbulent regime between BLIM2D and SIM2D[1] . However, close to the stagnation point, this discrepancy becomes significant because the momentum thickness is small. This is again a consequence of the acceleration of transition due to roughness.

BLIM3D was thus included successfully in IGLOO3D and gave promising result in the present test case. But a proper choice of the turbulent closure relations had to be made and more work is necessary on the turbulent regime to further improve the results.

DOI: 10.13009/EUCASS2017-400



(a) Heat transfert coefficient

(b) Momentum thickness

Figure 4: Heat transfer and momentum thickness computed with the different approaches employed for the computation of the NACA0012 glaze case

5. Conclusion

The present article described a 3D integral boundary layer method and its application to icing problems. A 3D code was developed and included in the ONERA's 3D icing suite, IGLOO3D in order to reduce the computational cost of the airflow during the process of ice accretion computations.

The solved equations which were presented are an extension of the 2D system of equations of Bayeux et al.[1] The unsteady momentum and kinetic energy equations are written in conservation form. Regarding heat transfer, a 2D-like approach is used to infer the heat transfer coefficient from the dynamics of the boundary layer.

The solver is based on the Finite Volume method, using an upwind scheme, which was presented in the present paper. It is worth mentioning that the equations are solved over the whole iced surface.

An unstructured NACA0012 test-case was presented to demonstrate the ability of the method to be efficiently included in ice accretion computations. The results show a sensitivity of the heat transfer coefficient and thus of the glaze ice shapes to the momentum thickness produced by the code. Since laminar-turbulent transition occurs very rapidly on ice, it has been shown that proper turbulent closure relations have to be employed.

References

[1] C. Bayeux, E. Radenac, and P. Villedieu. *Theory and validation of a 2D Finite-Volume integral boundary layer method intended for icing applications*. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, June 2017.

[2] N. Bempedelis, C. Bayeux, G. Blanchard, E. Radenac, and P. Villedieu. *A 3D Finite-Volume Integral Boundary Layer method for icing applications*. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Jun 2017.

[3] J. Cousteix. Three-dimensional boundary layers: Introduction to calculation methods. In *AGARD Computation of Three-Dimensional Boundary Layers Including Separation 49 p (SEE N87-22145 15-34)*, February 1987.

[4] J. Cousteix. *Couche Limite Laminaire*. Cepadues Editions, 1989.

[5] M. Drela. Two-dimensional transonic aerodynamic design and analysis using the euler equations. PhD Massachusetts Institute of Technology, 1985.

[6] M. Drela. MISES implementation of modified Abu-Ghannam/Shaw transition criterion. online database, 1998.

DOI: 10.13009/EUCASS2017-400

[7] M. Drela. *Three-Dimensional Integral Boundary Layer Formulation for General Configurations*. Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jun 2013.

[8] R. W. Gent, N. P. Dart, and J. T. Cansdale. Aircraft icing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 358(1776):2873–2911, 2000.

[9] W.M. Kays and M.E. Crawford. *Convective Heat and Mass Transfer*. McGraw Hill, New York, 1993.

[10] M. Lokatt and D. Eller. Finite-volume scheme for the solution of integral boundary layer equations. *Computers & Fluids*, 132:62 – 71, 2016.

[11] H. Ludwieg and W. Tillmann. Investigations of the wall-shearing stress in turbulent boundary layers. Technical Report NACA-TM-1285, National Advisory Committee for Aeronautics; Washington, DC, United States, May 1950.

[12] B.H. Mughal. A calculation method for the three-dimensional boundary layer equations in integral form. Master Thesis, Massachusetts Institute of Technology, 1992.

[13] B.H. Mughal. Integral methods for three-dimensional boundary layers. PhD Thesis, Massachusetts Institute of Technology, 1998.

[14] D.F. Myring. An integral prediction method for three dimensional turbulent boundary layers in incompressible flow. Technical Report 70147, 1970.

[15] E. Radenac. *Validation of a 3D ice accretion tool on swept wings of the SUNSET2 program*. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Jun 2016.

[16] T. W. Swafford. Analytical approximation of two-dimensional separated turbulent boundary-layer velocity profiles. *AIAA Journal*, 21(6):923–926, Jun 1983.

[17] T. Tai. *An integral prediction method for three-dimensional flow separation*. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, Jan 1984.

[18] P. Trontin, G. Blanchard, A. Kontogiannis, and P. Villedieu. *Description and assessment of the new ONERA 2D icing suite IGLOO2D*. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Jun 2017.

[19] A. Van Garrel. Integral boundary layer method for wind turbine aerodynamics,. Technical Report ECN-C–04-004, Decmber 2003.

[20] F. M. White. *Viscous fluid flow*. McGraw-Hill 2nd Edition, 1974.