A 3D Finite-Volume Integral Boundary Layer method for icing applications

Nikolaos Bempedelis*
Charlotte Bayeux†
Ghislain Blanchard‡
Emmanuel Radenac‡
Philippe Villedieu‡

ONERA, Toulouse, 31055, France

A three-dimensional integral boundary layer code was developed to allow fast computations of boundary layer flows for the purpose of ice accretion modelling. The model is derived in this paper. It is based on a surface Finite-Volume approach. 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. A validation of the code is also presented in the present article. First, the 3D solver is cross-checked against a 2D solver on test cases of self-similar flows and on a NACA0012 configuration. The modelling of the effects of three-dimensionality is also assessed on a self-similar flow test-case. Moreover, the use of unstructured grids is also validated. Finally, an example of the use of the code for the computation of ice accretion is presented.

I 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.[1] 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

*PhD student, Mechanical Engineering Department, University College London, Gower St, London WC1E 6BT, UK
†PhD student, Aerodynamics and Energetic Modeling Dept., BP 4025 2 avenue Ed. Belin, Toulouse, F.
‡Research engineer, PhD, Aerodynamics and Energetic Modeling Dept., BP 4025 2 avenue Ed. Belin, Toulouse, F.

of Thwaites,[2] 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.[3] 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 will be presented in section II. The Finite-Volume Method used to discretize the system will be exposed in section III. Validation and cross-checking of the 3D code against the 2D code in 2D configurations will be addressed in section IV, as well as a validation on a 3D configuration. The use of the integral boundary layer code for ice accretion computation will be shown in section V.

II Derivation of the model

II.A 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.[3,4] 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.[5] Previous works on the 3D IBL equations mainly employed Finite Difference Method in curvilinear coordinates.[6,7] With such methods, it is difficult to handle surface curvature discontinuities[8] and the IBL equations become very complex due to the introduction of numerous metric coefficients and geodesic curvature terms.[9] 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[4] and used in more recent articles of Drela and co-workers.[8] 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 II.B.

Moreover, the integral boundary layer variables are calculated by on-the-fly integration of the velocity profiles like in Mughal's PhD thesis.[4] 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 II.C.

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 III.

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 II.B. 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 II.D). Consequently, only the derivation of the 3D dynamic integral boundary layer equations is addressed.

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

Like in the 2D code,[3] 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):

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

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:

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

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

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

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

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

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,[8] 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 ($\boldsymbol{X}$,$\boldsymbol{Y}$,$\boldsymbol{Z}$)):

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

II.C 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.

II.C.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 $H = \frac{\delta_{1s}}{\theta_{ss}}$.[3] 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[11] 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[4] 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: $\delta = \delta_{1s}\frac{b(H)}{Hg(H)}$.[3] $\hat{u_c}$ 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)).

II.C.2 Turbulent regime

The streamwise velocity profile employed for the turbulent regime is based on the work of Tai,[12] 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[13] or Drela's approach of transporting turbulent shear stress[8] 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:[14]

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

which is assumed valid for $Re_{\theta_{ss}} > 1200$, or the one proposed by White:[15]

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

The dissipation coefficient is given by Drela's relation:[16]

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

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[17] 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 $\mathrm{Re}_k = \dfrac{k_s|\boldsymbol{q}_e|}{\nu_e}$ larger than 600, where k_s is the equivalent sand grain roughness height and ν_e is the kinematic viscosity of air.

II.C.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:

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

II.C.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:

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

II.D Computation of heat transfer

The approach used here is commonly used in 2D ice accretion codes, as shown by Gent et al.[19] 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:

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

III 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:

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

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

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

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).

III.A 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.

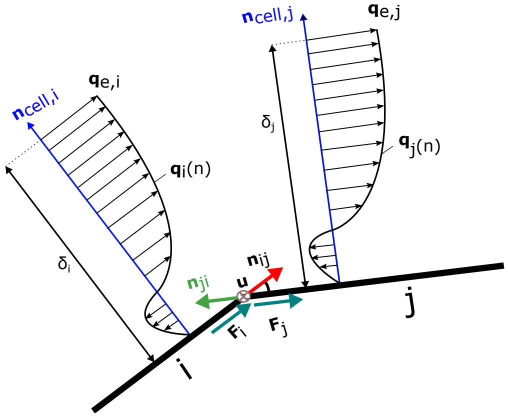


Figure 1: *Schematic of fluxes' calculation method in a 2D cut of an embedded surface.*

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, $\boldsymbol{u}_{ij} = \dfrac{\boldsymbol{n}_{cell,i}\times\boldsymbol{n}_{cell,j}}{|\boldsymbol{n}_{cell,i}\times\boldsymbol{n}_{cell,j}|}$
- 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,[3] 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:

$$\omega_i = |G_iG_{ij}|^{-1}, \quad \omega_j = |G_jG_{ji}|^{-1} \tag{53}$$

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,[3] 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[3] 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:[3]

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

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.

III.B 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:

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

IV Validation of the method

A first step towards the validation of the method is to compare it against 2D theoretical test-cases (self-similar boundary layer solutions of Falkner and Skan). The impact of the structured or unstructured grid will be evaluated on the theoretical test-case, as well as the grid refinement and the effect of rotations.

The 3D code, BLIM3D, will also be cross-checked against its already validated 2D version, BLIM2D, on a NACA0012 airfoil test-case to validate the laminar-turbulent transition. Numerical results are also compared with a reference solution given by an ONERA in-house code (CLICET) based on the full Prandtl equations.[21]

The last test case is focused on the study of a self-similar solution for a 3D boundary layer as proposed by Cooke[22] to assess the ability of the proposed method to deal with 3D flows.

It has to be noted that in all of the subsequent simulations, the CFL value was set to $CFL = 0.9$.

IV.A Falkner-Skan conditions

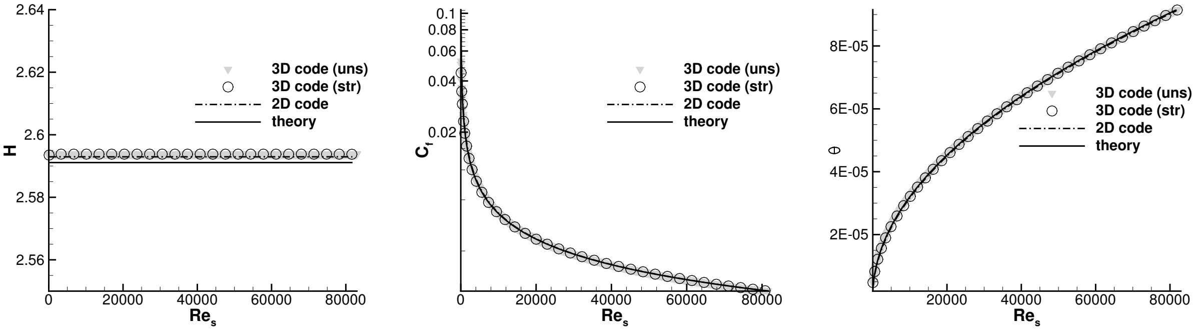
The theoretical test-cases are the self-similar boundary layer solutions of Falkner and Skan. This family of solutions corresponds to 2D laminar boundary layers developing along the x direction when the external flow velocity is $u_e(x) = cx^m$, where c and m are two constants. The pressure gradient is thus dependent on m. It is positive (adverse) for negative values of m. $m = 0$ corresponds to a flow over a flat plate (zero pressure gradient). The pressure gradient is negative (favorable) for positive values of m. One may notice that $m = 1$ represents a 2D stagnation point flow.

A very important parameter that characterizes the state of the boundary layer is the shape factor H. It can be proven that for the special cases of self-similar solutions, the shape factor is constant all along the surface. Besides, the skin friction $C_f = C_{fs}$ and $\theta = \theta_{ss}$ are major parameters for icing (since the heat transfer coefficient can be inferred from θ as seen in section II.D). The comparisons will thus focus on H, C_f and θ.

IV.A.1 Computations on structured and unstructured grids

The computations were performed with both the 2D and 3D codes. A mesh with 512 equidistant points was used in the streamwise direction. In order to properly compare the results between the two codes, the same mesh was used in 3D, with only one cell in the transverse direction for structured meshes. The results given by BLIM3D on unstructured Delaunay meshes (with the same characteristic cell size) are also presented.

Zero pressure gradient (Blasius solution) The case of a flat plate subjected to a constant external velocity field was studied. The Reynolds number of the computation is $Re = 81800$, the flow is expected to be laminar over the entire length of the plate.

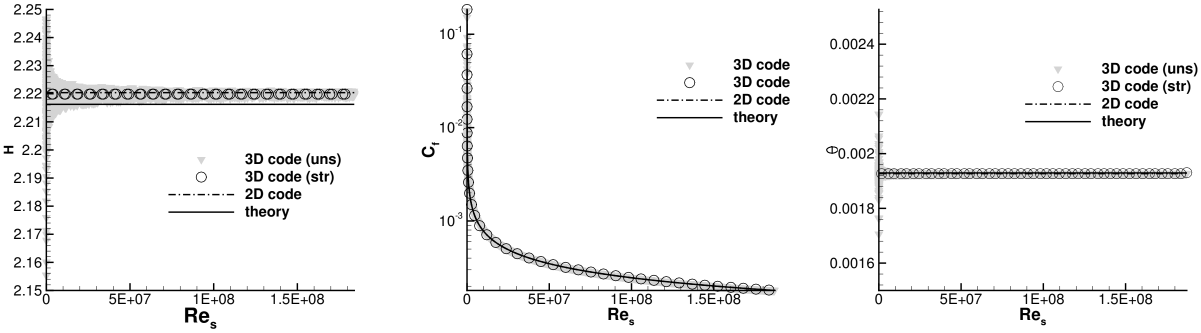


(a) *Shape factor H* **(b)** *Skin friction coefficient C_f* **(c)** *Momentum thickness θ_{ss} (m)*

Figure 2: *BLIM2D and BLIM3D results, on structured (str) and unstructured (uns) meshes, for the case $m = 0$ plotted against the local Reynolds number Re_s.*

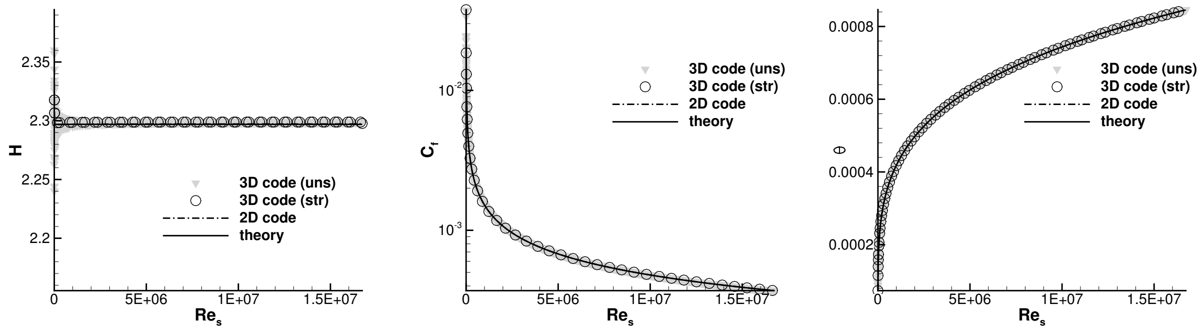
The shape factor H, the skin friction coefficient C_f and the (streamwise) momentum thickness θ obtained from the 2D and 3D versions of the code are plotted against the local Reynolds number in the direction of the external streamlines $Re_s = q_e s/\nu_e$ in figure 2. One can observe the accordance in the results between the two versions of the code. At the same time, one can see how the integral method compares against the analytical solution to the differential boundary layer equations, as proposed by Blasius (relative error on H lower than 0.1% according to values reported in table 1).

Non-zero pressure gradient Two test-cases were studied with favorable pressure gradient, for two different values of the exponent m. The first one corresponds to a 2D stagnation point flow (case $m = 1$) and the second one to an accelerated boundary layer (case $m = 1/3$). The corresponding numerical results are presented in figures 3 and 4 respectively.



(a) *Shape factor H* **(b)** *Skin friction coefficient C_f* **(c)** *Momentum thickness θ_{ss} (m)*

Figure 3: *BLIM2D and BLIM3D results, on structured (str) and unstructured (uns) meshes, for the 2D stagnation point flow (case $m = 1$) plotted against the local Reynolds number Re_s.*



(a) *Shape factor H* **(b)** *Skin friction coefficient C_f* **(c)** *Momentum thickness θ_{ss} (m)*

Figure 4: *BLIM2D and BLIM3D results, on structured (str) and unstructured (uns) meshes, for the accelerated boundary layer (case $m = 1/3$) plotted against the local Reynolds number Re_s.*

Cases	**m=0**	**m=1/3**	**m=1**
theory	2.59110	2.29694	2.21623
2D code	2.59294	2.29726	2.22039
3D code (str)	2.59380	2.29890	2.21986
3D code (uns)	2.59380	2.29892	2.22008

Table 1: *Results on shape factor H (asymptotic values).*

For both cases, the results obtained are almost identical for both 2D and 3D methods on structured meshes. The relative error on the shape factor is negligible (table 1). Due to the simplifications introduced in the treatment of the stagnation point (section III.A), small discrepancies can be observed for the shape factor (figure 3a) and the momentum thickness (figure 3c) for low Reynolds numbers of the 2D stagnation point flow (case $m = 1$) for the 3D results with an unstructured mesh. Nevertheless, the results produced on the structured and the unstructured meshes are in good agreement for higher local Reynolds numbers.

For the case of an accelerated boundary layer ($m = 1/3$), a small error is observed for both the 2D and the 3D method at low Reynolds numbers, which is consistent with the fact that the stagnation point correction is active in this region and it was developed to correctly handle the $m = 1$ case. The evolution of this error against the mesh size is detailed in the next paragraph.

IV.A.2 Mesh convergence

The ability of the method to converge when the mesh size decreases is now presented. To that end, the rate of convergence of the method was assessed in the case of an external velocity field belonging in the Falkner-Skan family with an exponent equal to $m = 1/3$. Uniform structured grids were used with different refinements in the streamwise direction and one cell in the transverse direction.

For a constant grid refinement ratio r, the expression of the convergence rate p may be written as follows:[20]

$$p = \frac{\log\left(\frac{f_3 - f_2}{f_2 - f_1}\right)}{\log r} \tag{62}$$

where f_i is the average of the shape factor H over the grid i and r is the characteristic cell size ratio between two meshes (i.e. in our case $r = 0.5$). Figure (5) shows that the numerical solution for the shape factor converges as expected to a constant value when the mesh is refined. Quantitative results are also reported in table 2. The convergence rate is close to 1, which is in good agreement with the fact that a first order scheme is used for the spatial discretization.

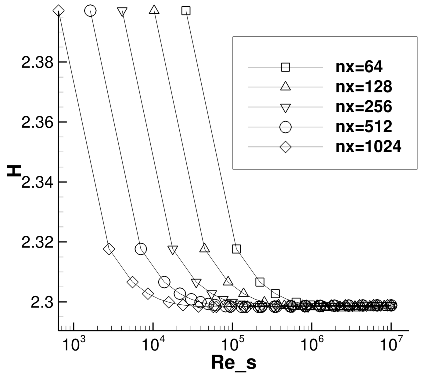


Figure 5: *BLIM3D convergence assessed on the shape factor*

Mesh size	$< H >$	p
64	2.30050	-
128	2.29965	-
256	2.29925	1.09
512	2.29907	1.16
1024	2.29899	1.22

Table 2: *BLIM3D convergence rate (based on the average shape factor $< H >$).*

IV.A.3 Coordinate system transformation

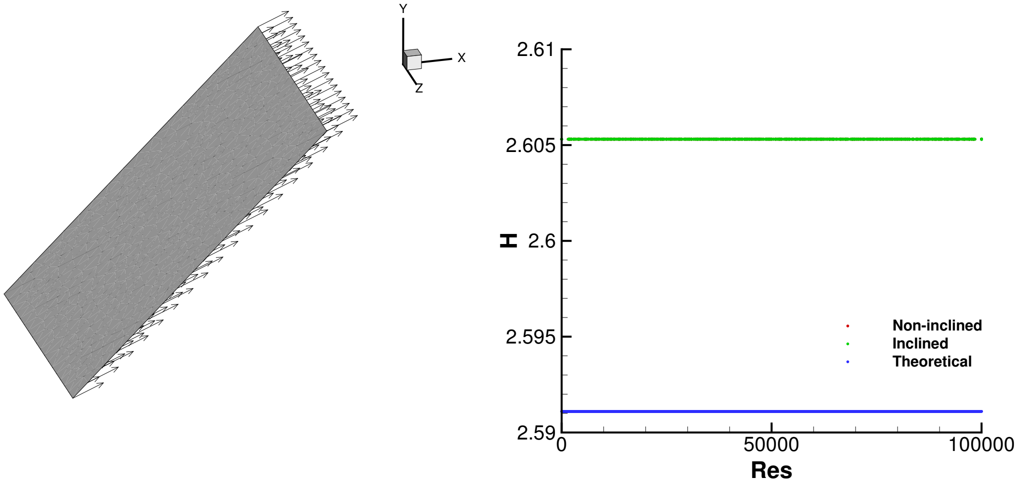
Another interesting test case to be studied before proceeding with more complex geometries and external velocity fields was to re-run the previous simulations with the flat plate being arbitrarily placed in the 3D space. Additionally, the velocity vector was not aligned to the flat plate but the incoming flow was creating an angle of 45 degrees with it. The goal of this test case was to verify that the process of rotation between the local and global coordinate systems was properly performed (since the global and the local coordinate systems were no more the same). In the aforementioned simulation, the velocity vector was $\boldsymbol{q}_e = \left(1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}\right)\, m/s$ and was tangential to the plane of the flat plate.

The geometry of the problem along with the external velocity field are presented in figure 6a. As observed in figure 6b, the results obtained for the inclined and non-inclined cases are identical and therefore the coordinate system transformations are deemed to be properly executed.

IV.B NACA0012 case

The goal of this test case is to assess the ability of the 3D method to correctly compute the laminar-turbulent transition over a NACA0012 profile. The numerical results of BLIM3D are compared against BLIM2D and the code CLICET which is considered as the reference.

A 2D mesh of 512 cells, with a more refined region close to the leading edge, was used for the 2D computations with BLIM2D and CLICET. The 3D mesh was generated from the 2D mesh thanks to an extrusion in the



(a) *Problem's geometry* **(b)** *Shape factor H*

Figure 6: *(a) Problem's geometry and (b) shape factor H plotted against the local Reynolds number Re_s for the inclined and non-inclined test cases.*

transverse direction over 3 cells. In this way, both meshes have the same refinement along the streamwise direction in order to have comparable results.

The data set for this test case is given in table 3. The wall is assumed to be smooth. The local criterion proposed by Drela (equation (29)) was thus used for this test case. Besides, Ludwieg-Tillmann relation (equation (22)) was employed for the turbulent skin friction closure. An arbitrary initial condition was imposed at the beginning of the calculation. It is important to notice that the position of the stagnation point and the position of the transition are automatically determined during the 3D code calculation.

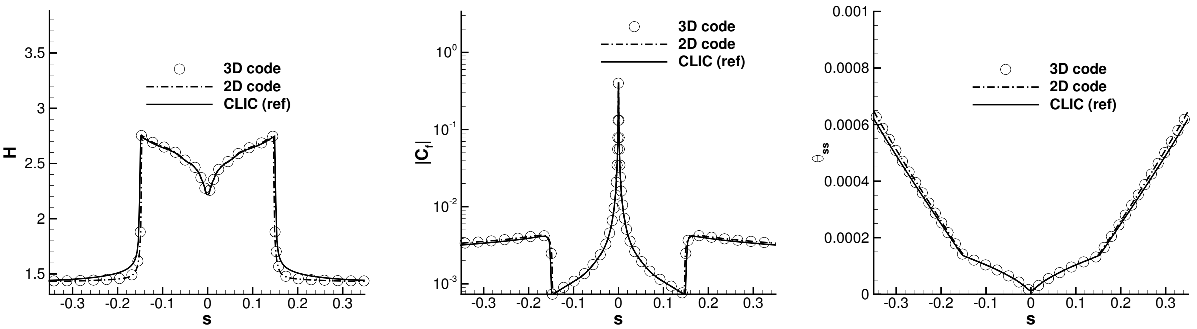
Profile	NACA0012
chord length (m)	0.5
AoA ()	0
M_∞	0.15
T_∞ (K)	263
P_∞ (Pa)	80000

Table 3: *Operating conditions for the NACA0012 test case*

Figure 7 shows that the results produced by BLIM3D are in very good agreement with BLIM2D and CLICET. The location of the laminar-turbulent transition, where the skin friction abruptly increases and the shape factor decreases, is well predicted. The laminar regime is very well captured, whereas the turbulent region is correctly reproduced despite small discrepancies which were often observed with BLIM2D. The use of more evolved closure relations should improve these results.

IV.C Falkner-Skan-Cooke conditions

The validation of the 3D character of the method was carried out against an extension of the self-similar solutions to 3D as proposed by Cooke,[22] which essentially represents the solution for a laminar boundary layer flow over an infinite swept wing. According to their work, an analytical solution is obtained in 3D when the streamwise component of the external velocity field is following the 2D Falkner-Skan family of velocity profiles, while the crosswise velocity is constant, i.e. $\boldsymbol{q}_e = \left((X/L)^m, cst, 0\right)$.



(a) *Shape factor H*

(b) *Magnitude of the skin friction coefficient $|C_f|$*

(c) *Momentum thickness θ_{ss} (m)*

Figure 7: *Comparisons of the numerical results with respect to the curvilinear abscissa of the NACA0012 profile.*

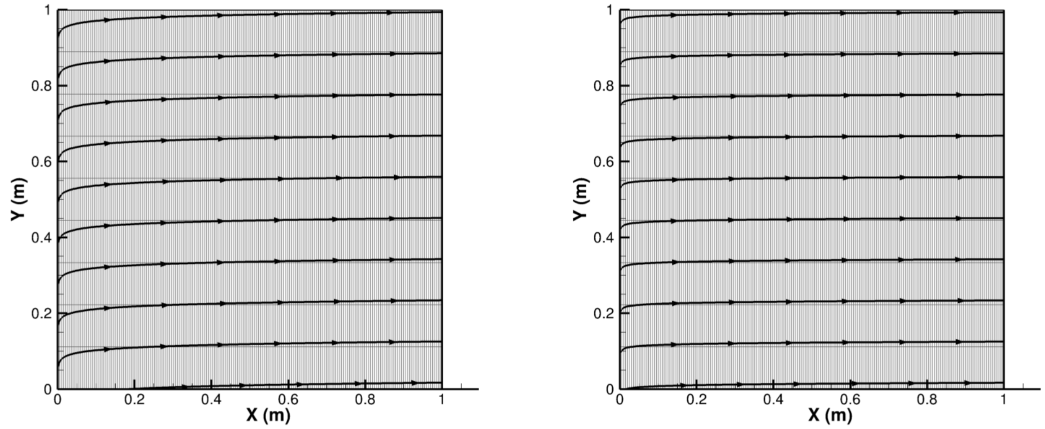
In such cases, it can be shown after the work of Cooke[22] that the shape factors defined as

$$H_{XX} = \frac{u_{Xe}\delta_{1X}}{|\boldsymbol{q}_e|\theta_{XX}}\ , \qquad H_{YY} = \frac{u_{Ye}\delta_{1Y}}{|\boldsymbol{q}_e|\theta_{YY}} \tag{63}$$

are expected to remain constant along the surface of the body.

A square flat grid was used for the test case (figure 8). In order to simulate the infinite conditions of the flow, periodicity boundary conditions were applied at the boundaries $Y_{\min}$, $Y_{\max}$. The velocity profile was set to : $\boldsymbol{q}_e = (100x, 1, 0)$.

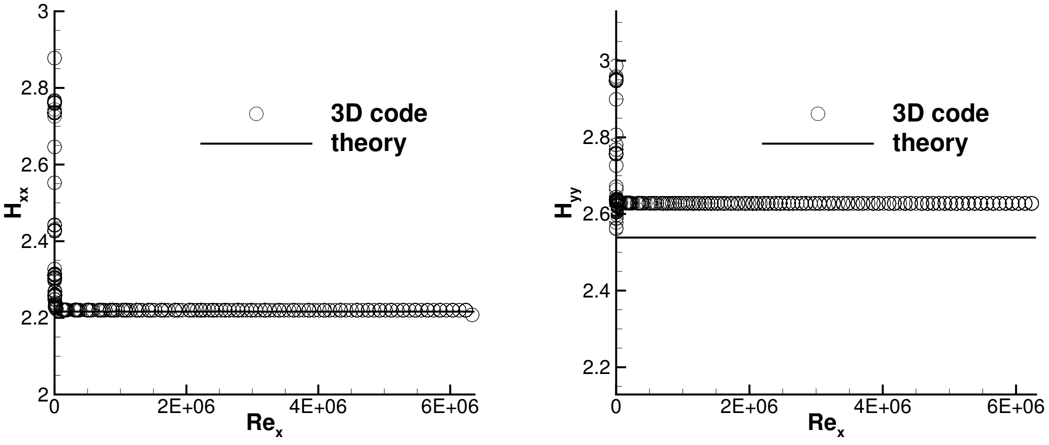
As it can be observed in figure 8, BLIM3D is able to calculate the expected skewing between the external and the wall limiting streamlines. The calculated values for the shape factor are well calculated when compared to the theoretical ones (figure 9). However, and possibly connected to the discretization error appearing in the region of the leading edge, the boundary layer requires a certain length to fully develop and reach its self-similar state as it was already discussed in section IV.A.1. The stagnation point correction does not act properly here, which could be due to the fact that it was developed for a real stagnation point and not for a separation line as is computed here.



(a) *External streamlines*

(b) *Wall limiting streamlines*

Figure 8: *External and wall limiting (skin friction) streamlines, $\boldsymbol{q}_e = (100x, 1, 0)$.*



(a) *Shape factor H_{XX}*

(b) *Shape factor H_{YY}*

Figure 9: *Shape factor H_{XX} and H_{YY} plotted against the local Reynolds number Re_x, $\boldsymbol{q}_e = (100x, 1, 0)$.*

V 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 4. The flow is 2D but a 3D unstructured grid was generated as shown in figure 10. It is composed of 39940 triangles, and a local refinement is used in the vicinity of the stagnation point.

	AOA	**Chord**	**P**$_\infty$	**T**$_\infty$	**M**$_\infty$	**LWC**	**MVD)**	Δt
	(^{o})	**(m)**	**(Pa)**	**(K)**		**(g/m^3)**	**(μm)**	(s)
NACA0012 airfoil	0	0.533	95937	264.82	0.2056	0.61	40	672

Table 4: *Conditions of the test-case, AOA: angle of attack; P_∞, T_∞, M_∞: 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 0.533 mm) and, as shown in table 5, the two closure relations for skin friction were assessed (Ludwieg-Tillmann, equation (22), and White, equation (23)).

Table 5 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, SIM2D, TRAJL2D and MESSINGER2D[18] were used for the structured mesh, the inviscid flow, the simplified integral boundary layer, the Lagrangian trajectography and the ice acrretion computations, respectively.

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 II.D, fed by the smooth-wall momentum thickness provided by elsA, were used. This approach is very similar to IGLOO2D and thus

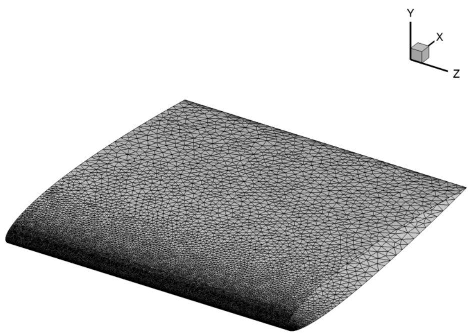


Figure 10: *NACA0012 surface mesh for BLIM3D*

Code	**Gas flow**	**Droplet trajectories**	**Friction closure**	h_c closure	**Grid**	**Predictor-corrector**	**Output**
IGLOO3D	elsA RANS	SPIREE		section II.D	3D structured 1312 elements	Predictor	Boundary layer, ice
IGLOO3D	elsA Euler BLIM3D	SPIREE	Ludwieg-Tillmann	section II.D	3D unstructured 39940 elements	Predictor	Boundary layer, ice
IGLOO3D	elsA Euler BLIM3D	SPIREE	White	section II.D	3D unstructured 39940 elements	Predictor	Boundary layer, ice
IGLOO2D	EULER2D SIM2D	TRAJL2D		section II.D	2D structured 128 elements	Predictor-Corrector	Boundary layer, ice
IGLOO2D	EULER2D BLIM2D		Ludwieg-Tillmann		2D structured 128 elements		Boundary layer
IGLOO2D	EULER2D BLIM2D		White		2D structured 128 elements		Boundary layer

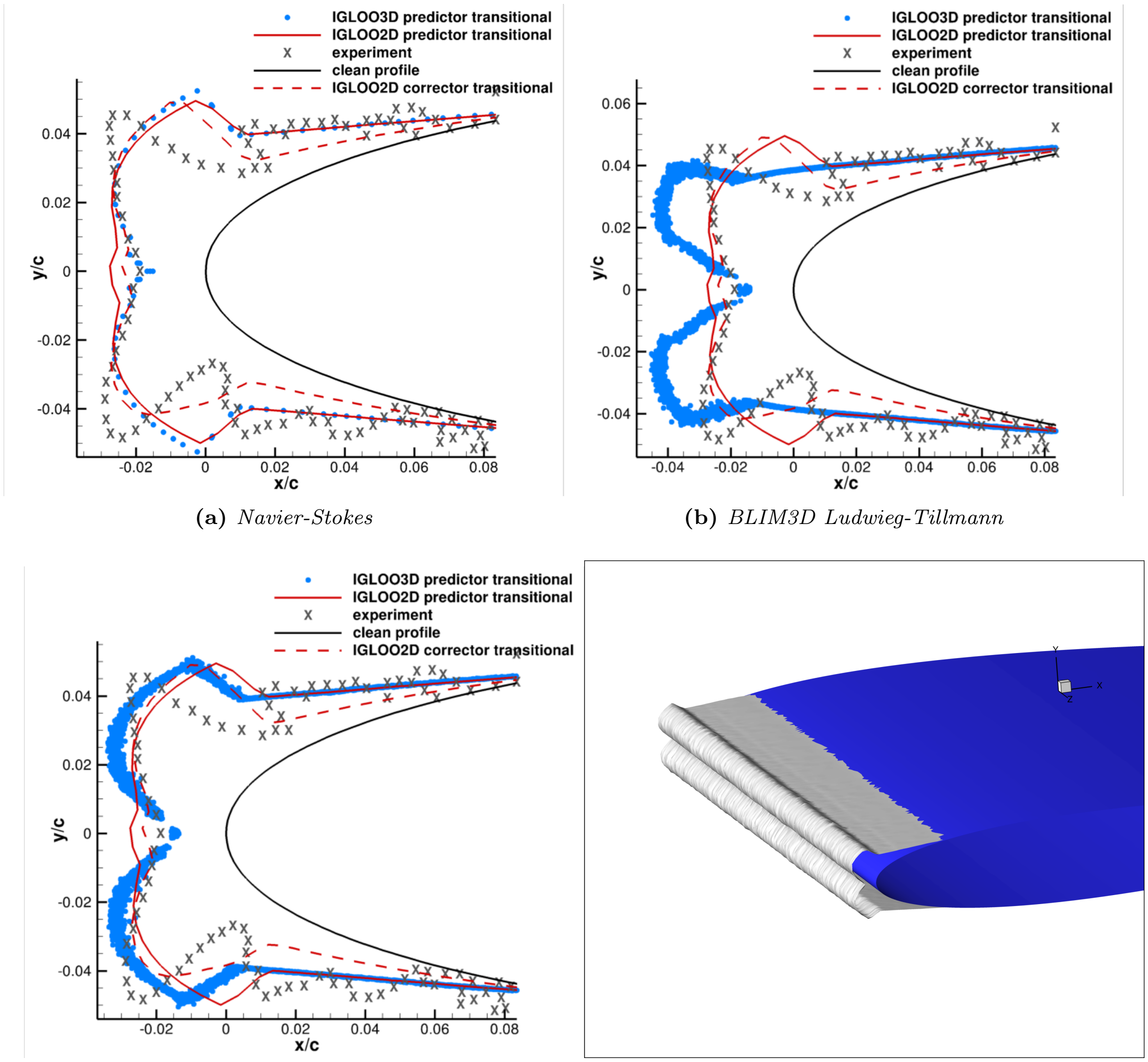
Table 5: *Various computations performed on the NACA0012 glaze ice case.*

often produces satisfactory results on 2D configurations. For both the RANS and BLIM3D approaches, IGLOO3D was used in predictor mode.

Figure 11 gathers the results produced by the various approaches. The horns are globally well captured with IGLOO2D, although they are a little too large because too much water runback occurs. Moreover, 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, despite small discrepancies in the vicinity of the stagnation point (figure 11a). This can be explained by the smaller heat transfer coefficient obtained with IGLOO3D compared to IGLOO2D (figure 12). Consequently, the ice is less cooled, the runback is thus enhanced and the solidification lessened.

When BLIM3D is used, the ice shapes given by IGLOO3D are very dependent on the closure model retained for the skin friction (figures 11b and 11c). Ludwieg-Tillmann model fails to capture the correct location of the horns, whereas White model is much more satisfactory. This can be linked to the fact that the latter is not valid for low $Re_{\theta_{ss}}$. Indeed, the transition on rough walls forces the transition to occur very rapidly, and the order of magnitude of $Re_{\theta_{ss}}$ is only a few tens. As a consequence, figure 12a shows that the heat transfer coefficient given by Ludwieg-Tillmann model is largely overestimated over the horns (see figure 13 to identify the location of the horns in terms of the wrap distance s from the stagnation point). Although it



(a) *Navier-Stokes*

(b) *BLIM3D Ludwieg-Tillmann*

(c) *BLIM3D White*

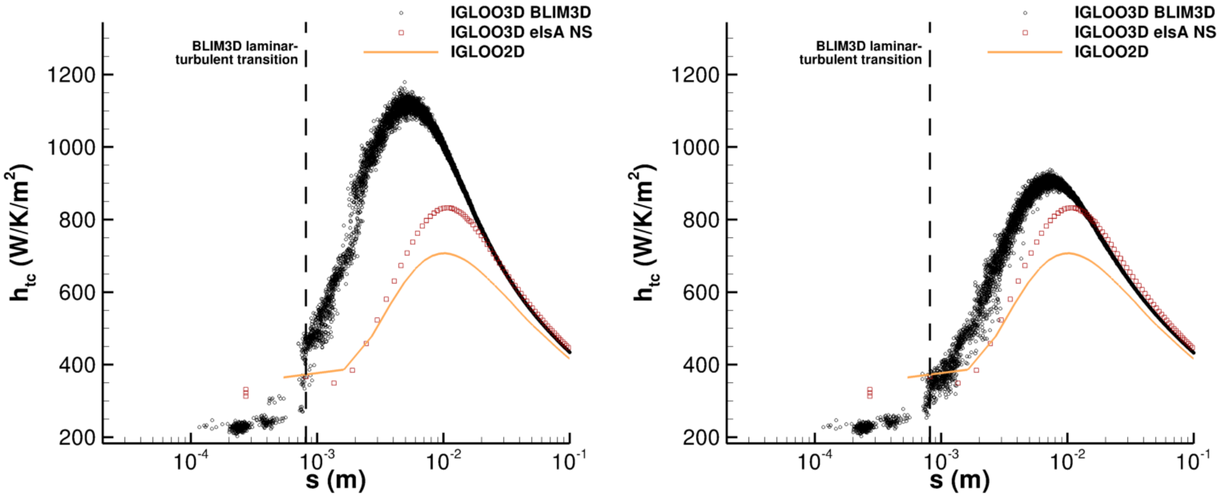
(d) *BLIM3D White*

Figure 11: *IGLOO3D computations with Navier-Stokes or BLIM3D approach and two different closure relations*

is still overestimated with White model (figure 12b), the agreement becomes much better between BLIM3D, IGLOO2D (SIM2D) and IGLOO3D used along with elsA.

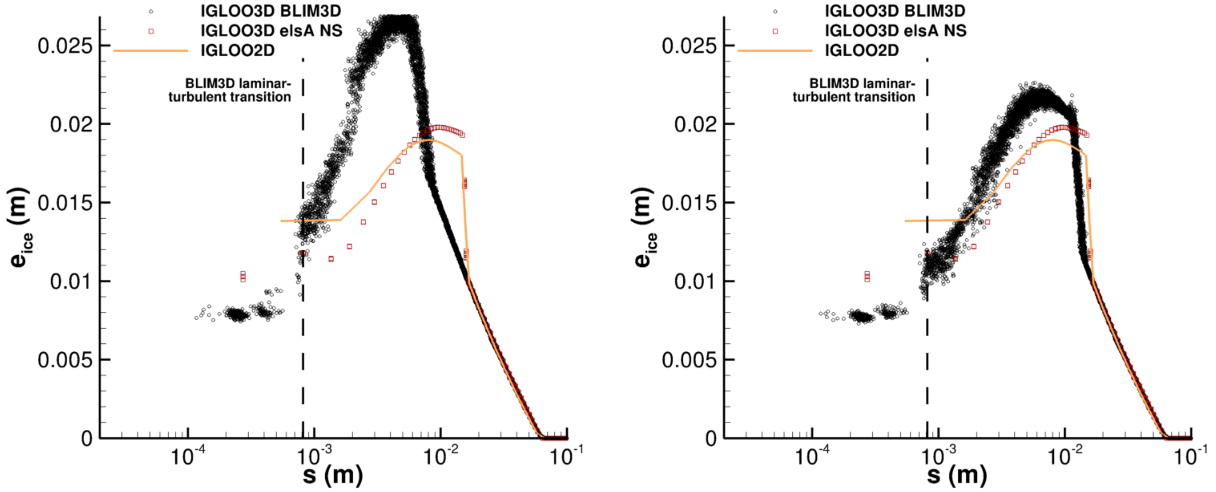
It is worth recalling that the heat transfer coefficient is dependent on the momentum thickness calculated by the boundary layer code. Figure 14 shows that BLIM3D and its 2D counterpart BLIM2D (run on the same mesh as IGLOO2D SIM2D with the same inviscid data) produce very similar results, although the BLIM3D results are a little noisy due to the unstructured grid. However, the momentum thickness given by SIM2D and elsA are larger. Small discrepancies are often observed in turbulent regime between BLIM2D and SIM2D.[3] 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. 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.



(a) *BLIM3D Ludwieg-Tillmann* **(b)** *BLIM3D White*

Figure 12: *Heat transfer computed with the different approaches employed for the computation of the NACA0012 glaze case*



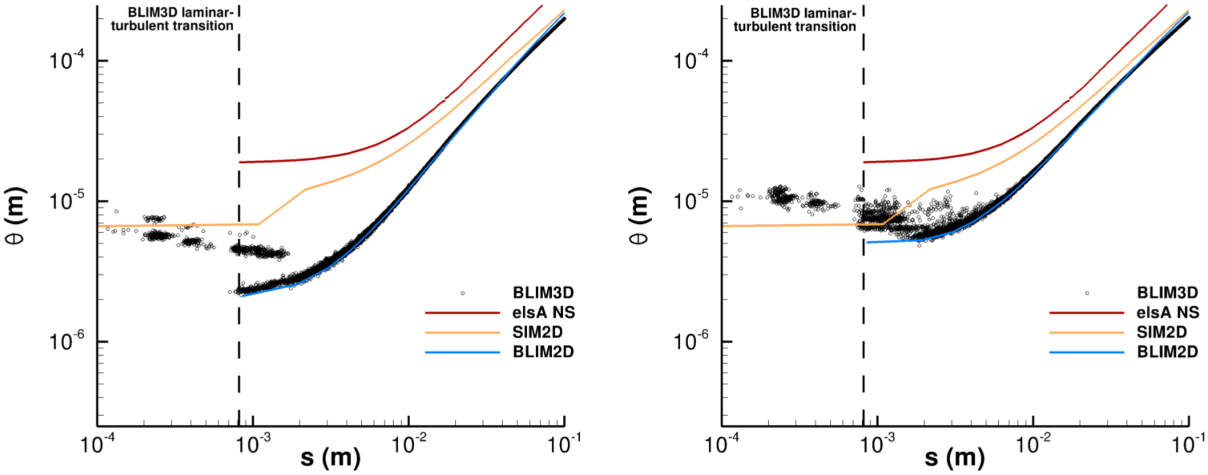
(a) *BLIM3D Ludwieg-Tillmann* **(b)** *BLIM3D White*

Figure 13: *Ice thickness computed with the different approaches employed for the computation of the NACA0012 glaze case*

VI 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.[3] 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



(a) *BLIM3D Ludwieg-Tillmann*

(b) *BLIM3D White*

Figure 14: *Streamwise momentum thickness computed with the different approaches employed for the computation of the NACA0012 glaze case*

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.

Several validation cases were presented to ensure that the code properly catches the boundary layer characteristics. Moreover, 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 great 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.

Appendix

The functions used in this paper for the laminar velocity profile exponent $p_s(H)$, the streamwise skin friction coefficient $g(H) = \frac{C_{fs}}{2} Re_{\theta_{ss}}$ and $b(H)$ are:

$$p_s(H) = \begin{cases} 2.4834 + \dfrac{0.7877}{(H-1.9538)^{1.6001}} \text{ if } H \leq H_{crit} \\ 2 + \dfrac{2.0411 \times 10^{11}}{(H+25.89)^{7.7560}} \text{ if } H > H_{crit} \end{cases} \tag{64}$$

$$g(H) = \begin{cases} 2.99259 \left[\left(\dfrac{1}{H} - \dfrac{1}{2H_{crit}} \right)^{1.7} - \left(\dfrac{1}{2H_{crit}} \right)^{1.7} \right], & \text{for } H \leq H_{crit} \\ 0.20644 - 90.30936 \left(\left(\dfrac{1}{H_{crit}} \right)^{1.3} - \dfrac{1}{H^{1.3}} \right)^{3.35661} + & \\ (H-1) \left[-0.06815 + 46.34236 \left(\dfrac{1}{H_{crit}^2} - \dfrac{1}{H^2} \right)^{2.338238} \right], & \text{for } H > H_{crit} \end{cases} \tag{65}$$

$$b(H) = p_s(H) - \sqrt{p_s(H)^2 - p_s(H)\left(p_s(H)+1\right) H g(H)} \tag{66}$$

$H_c = 4.02923$ is the value of the shape factor at separation of the boundary layer.

References

[1] Radenac, E., *Validation of a 3D ice accretion tool on swept wings of the SUNSET2 program*, 8th AIAA Atmospheric and Space Environments Conference - AVIATION 2016 WASHINGTON D.C., USA, 2016.

[2] Kays W.M. and Crawford M.E., *Convective heat and mass transfer*, McGraw-Hill, 1993.

[3] Bayeux, C., Radenac, E. and Villedieu, P., *Theory and Validation of a 2D Finite-Volume integral boundary layer method intended for icing applications*, 9th AIAA Atmospheric and Space Environments Conference - AVIATION 2017 DENVER, USA, 2017.

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

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

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

[7] Cousteix, J., *Three-dimensional Boundary Layers. Introduction to calculation methods*, AGARD R-741, paper 1, 1986.

[8] Drela M., *Three-dimensional integral boundary layer formulation for general configurations*, 21st AIAA computational fluid dynamics conference, 2013.

[9] Van Garrel A., *Integral boundary layer method for wind turbine aerodynamics*, Technical report ECN-C–04-004, 2003.

[10] Lokkat M. and Eller D., *Finite-volume scheme for the solution of integral boundary layer equations*, Computer and fluids, 132 (2016) 62-71, 2016.

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

[12] Tai T. C., *An integral prediction method for three-dimensional flow separation*, 22nd AIAA Aerospace Sciences Meeting, 1984.

[13] Swafford, T., *Analytical approximation of two-dimensional separated turbulent boundary layer velocity profiles*, AIAA Journal, 21(6), 1983.

[14] Ludwieg, H., Tillmann, W., *Investigations of the wall-shearing stress in turbulent Boundary layers.*, NACA-TM-1285, 1950.

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

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

[17] Drela, M., *MISES Implementation of Modified Abu-Ghannam/Shaw Transition Criterion.*, online database, 1998.

[18] Trontin, P., Blanchard, G., Kontogiannis, A., Villedieu, P. *Description and assessment of the new ONERA 2D icing suite IGLOO2D*, 9th AIAA Atmospheric and Space Environments Conference - AVIATION 2017 DENVER, USA, 2017.

[19] Gent R.W., Dart N.P. and Cansdale J.T., *Aircraft icing*, Phil. Trans. R. Soc. Lond. A, 358, 2000.

[20] Schwer L.E., *Is your mesh refined enough? Estimating discretization error using GCI*, LS-DYNA Anwenderforum, Bamberg, 2008.

[21] Aupoix B., *Couches Limites Bidimensionnelles Compressibles. Descriptif et mode d'emploi du code CLICET - Version 2015*, Technical report, 2015.

[22] Cooke J.C., *The boundary layer of a class of infinite yawed cylinders*, Mathematical Proceedings of the Cambridge Philosophical Society, 46(4), 1950.