JOURNAL OF AIRCRAFT
Vol. 37, No. 4, July–August 2000

Development of a Shallow-Water Icing Model in FENSAP-ICE

Y. Bourgault*
University of Ottawa, Ottawa, Ontario K1N 6N5, Canada
and
H. Beaugendre† and W. G. Habashi‡
McGill University, Montreal, Quebec H3A 2S6, Canada

As part of a modern comprehensive in-flight icing simulation code (FENSAP-ICE), a new thermodynamic model for ice accretion is developed (the Shallow-Water Icing Model or SWIM). SWIM is based on a system of partial differential equations (PDEs) and thus is thought to be superior to a control volume (no PDE) approach used in current icing codes. Flexibility in the physical modeling and flexibility in numerical algorithm selection make SWIM an attractive icing model. The complete model, numerical algorithms used, and results are presented.

Nomenclature

C_i = cell at node i of the dual surface mesh
C_{ice} = specific heat of ice
C_w = specific heat of water
h_f = film thickness
h_i = film thickness at node i
h_{ij} = midedge value of h
L_{evap} = evaporation latent heat of water
L_{fusion} = fusion latent heat of ice
L_{subl} = sublimation latent heat of ice
LWC = cloud liquid water content
$\dot{m}_{\text{evap}}$ = evaporative mass flux
$\dot{m}_i$ = ice accretion mass flux at node i
$\dot{m}_{\text{ice}}$ = ice accretion mass flux
n_{ij} = vector normal to the dual cell boundary along the edge ij
$\dot{Q}_{\text{cond}}$ = conductive heat flux from structure
$\dot{Q}_h$ = convective heat flux from air
S_i^h = right-hand side (RHS) of the discrete mass conservation equation
S_i^T = RHS of the discrete energy conservation equation
T = absolute temperature, °K
$\tilde{T}$ = scaled temperature ($= T - T_{\text{ref}}$)
$\tilde{T}_{d,\infty}$ = scaled temperature of supercooled droplets
$\tilde{T}_i$ = scaled temperature at node i
T_{ref} = reference temperature (triple point of water in °K)
T_∞ = absolute temperature of the air at infinity, K
U_∞ = velocity of air at infinity
u = velocity of the film
$\bar{u}$ = average (along y) velocity of the film
(u, v) = dependent variables in the icing plane
u_d = impact droplet velocity
(u_i, v_i) = dependent variables in the icing plane at node i
u_{ij} = midedge value of u
$x = (x_1, x_2)$ = coordinates along the wall
y = coordinates normal to the wall
β = local collection efficiency
μ_w = dynamic viscosity of water
ρ_w = water density
σ = Boltzmann constant
τ_{wall} = air wall shear stress

I. Introduction

THE simulation of ice accretion has been traditionally based on inviscid panel or Euler flow computations for the air, on Lagrangian particle-tracking techniques for the droplets impingement, and on a control volume analysis of the mass and heat transfer to model the ice accretion. The best known codes using these approaches are NASA's Lewice[1] and the ONERA icing code.[2] A wealth of experience on the physics of ice accretion makes these codes valuable tools for a broad spectrum of in-flight icing situations.[3] These codes are also efficient in terms of computer resources requirement, but they need fine tuning, especially of the heat-transfer coefficients, which cannot be predicted by potential or inviscid flow models.[4] The heat-transfer coefficient must thus be guessed or parameterized in some way, leaving the engineer with a difficult task in the presence of major flow separations, as with glaze ice.

The conventional icing simulation approach also suffers from other limitations, such as limited ability to handle full three-dimensional simulations. For example, the calculation of collection efficiency seems to be a mixture of two- and three-dimensional Lagrangian particle tracking, with the danger of missing three-dimensional effects on the collection efficiency close to blade or wing tips and in complex wing/nacelle configurations. Another limitation is the very crude geometrical modeling of the water film runback, which limits the applicability of the current icing simulation technology to two-dimensional. Apart from an effort on three-dimensional film modeling,[2] no three-dimensional film model is available in the icing literature or in icing codes.

Moreover, as far as the thermodynamics of ice and water on walls is concerned, current icing models[1,2,5] do not make any distinction between the physical model and the discretization of the model actually used in computer codes. As far as we know, the only effort in splitting the physical model and its numerical discretization is the work of Al-Khalil et al.,[6] but their work accounts only for the film runback with no ice accretion. The resulting icing model is a large system of discretized equations expressing the heat and mass balance on finite control volumes at the air/ice interface. It is unclear whether the system is well posed or even properly closed, or if unstable numerical solutions or numerical bifurcations can result from wrong physical assumptions or numerical schemes. In fact, the terms most likely to cause numerical instabilities, such as the time derivatives, are simply thrown away, and a sequence of steady problems is solved for what is clearly an unsteady problem, that is the growth of ice shapes.

Presented as Paper 99-0246 at the AIAA 37th Aerospace Sciences Meeting and Exhibit, Reno, NV, 11–14 January 1999; received 9 September 1999; revision received 10 January 2000; accepted for publication 13 January 2000.

*Assistant Professor, Department of Mathematics and Statistics. Member AIAA.

†Graduate Research Assistant, CFD Laboratory, Department of Mechanical Engineering.

‡Director, CFD Laboratory, Department of Mechanical Engineering. Associate Fellow AIAA.

Current computational fluid dynamics (CFD) technologies could, to a large extent, overcome the limitations just stated. By solving the compressible Navier–Stokes equations, one can automatically obtain the heat-transfer coefficients and, at the same time, fully and directly account for the influence of the viscous effects on the degradation of performance. A full three-dimensional approach to icing, with efforts put on physical models and on the independent choice of appropriate numerical techniques, is also possible through CFD technology, at a cost of solving models based on partial differential equations (PDEs). By adding and removing terms in the system of PDEs, one can model as much physics as necessary to have reliable predictions and make the information travel properly and naturally through space. The effect of the numerical scheme could also be investigated without having to compromise the physical modeling for the sake of code stability. Finally, a modern CFD approach to icing simulation would bring new computational and simulation technologies to the icing community, such as advanced iterative methods, moving meshes and mesh adaptation. This would help make the investigation of complex engineering problems, such as wing/body junctions, nacelle/inlet or tailplanes icing affordable.

The CFD Laboratory of Concordia University has developed an Eulerian model for impinging droplets.[7] The efficiency of this model has been shown through the investigation of impingement patterns on the nose of an aircraft, including fine details of the cockpit windows.[8] It is the purpose of the present paper to go a step further in the application of CFD techniques to the in-flight icing and propose a thermodynamic model for the ice accretion and water runback. The physical model, written as a system of PDEs, is described. Appropriate numerical methods and code interactions with the flow and droplet solvers are presented. Numerical results are also presented.

II. Shallow-Water Icing Model

An icing simulation code is generally composed of three to four modules, i.e., a flow solver, a droplets impingement solver, a thermodynamical module for ice accretion prediction, and optionally a structure code to account for conjugate heat transfer in anti-icing systems.[9] The classical compressible Navier–Stokes equations are at the basis of the present airflow computations. An Eulerian droplet code[7] is used to obtain the water collection efficiencies on walls.

A new equilibrium model is introduced in this paper to predict the ice accretion and water runback on the surface. Figure 1 shows the heat and mass transfer phenomena taken into account in the model.

The velocity u of the water in the film is a function of coordinates $x = (x_1, x_2)$ on the surface and y normal to the surface:

$$u(x, y) = f(\tau_{\text{wall}}, y) \tag{1}$$

where τ_{wall} is the main driving force for the water film. A simplifying assumption consists in taking f linear in y with $u(x, y)$ null at the wall, i.e.,

$$u(x, y) = (y/\mu_w)\tau_{\text{wall}}(x) \tag{2}$$

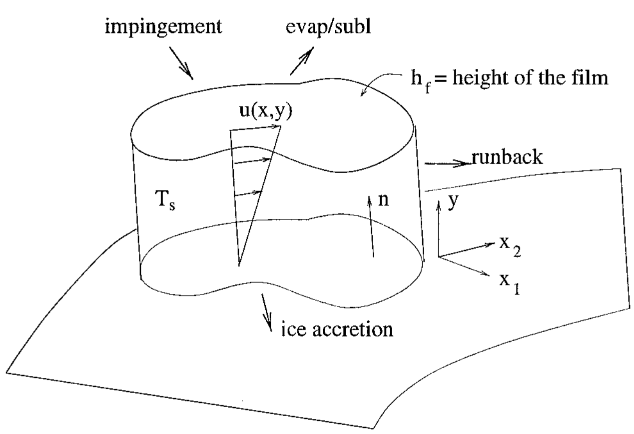

Fig. 1 Heat and mass balance in a thin film.

By averaging the physical quantities along the thickness of the film, a so-called Shallow-Water Icing Model (SWIM) is obtained. For the velocity this gives

$$\bar{u}(x) = \frac{1}{h_f}\int_0^{h_f} u(x, y)\,\mathrm{d}y = \frac{h_f}{2\mu_w}\tau_{\text{wall}}(x) \tag{3}$$

The system of PDEs is the following:

Mass Conservation

$$\rho_w\left[\frac{\partial h_f}{\partial t} + \operatorname{div}(\bar{u}h_f)\right] = U_\infty LWC\beta - \dot{m}_{\text{evap}} - \dot{m}_{\text{ice}} \tag{4}$$

where the three terms on the RHS model, respectively, the mass transfer caused by the water droplets impingement (source for the film), the evaporation, and ice accretion (sinks for the film).

Energy Conservation

$$\begin{aligned}\rho_w\left[\frac{\partial h_f C_w \tilde{T}}{\partial t} + \operatorname{div}(\bar{u}h_f C_w \tilde{T})\right] &= \left[C_w \tilde{T}_{d,\infty} + \frac{\|u_d\|^2}{2}\right]\\ &\quad\times U_\infty\, LWC\,\beta - 0.5(L_{\text{evap}} + L_{\text{subl}})\dot{m}_{\text{evap}}\\ &\quad+ (L_{\text{fusion}} - C_{\text{ice}}\tilde{T})\dot{m}_{\text{ice}} + \sigma\left(T_\infty^4 - T^4\right) + \dot{Q}_h + \dot{Q}_{\text{cond}}\end{aligned} \tag{5}$$

where the first three terms on the RHS model, respectively, the heat transfer caused by the super-cooled water droplets impingement, the evaporation, and ice accretion; the last three are the radiative, convective, and conductive heat transfer.

The coefficients ρ_w, C_w, $\tilde{T}_{d,\infty}$, U_∞, LWC, L_{evap}, L_{subl}, L_{fusion}, C_{ice}, σ, and T_∞ are constant parameters (at first) specified by the user. The substance involved (water near freezing point), and the ambient icing conditions completely determine those values. The Eulerian droplet module provides local values for the collection efficiency β and the droplet impact velocity u_d, and the flow solver provides local values for the wall shear stress τ_{wall} and the convective heat flux $\dot{Q}_h$. The evaporative mass flux is recovered from the convective heat flux using a parametric model.[5] The conductive heat flux could be given by a thermal structure analysis code or a simple model inside the icing module.

Thus, there remain three unknowns: the film thickness h_f, the equilibrium temperature $\tilde{T}$ within the air/water film/ice/wall interface, and the instantaneous mass accumulation of ice $\dot{m}_{\text{ice}}$. Only two PDEs are available to specify three unknowns, as obtained from the mass and energy conservation. The momentum conservation is hidden in the simplified and explicit form [Eq. (1)]. Its implicit solution, in conjunction with the other two conservation laws, would definitely allow the inclusion of more mechanical effects in the description of the film behavior (such as pressure gradient and surface tension effects), but each extra momentum equation would come with an extra unknown velocity component.

In fact, compatibility relations or state equations are needed to close the system. Up to this point, nothing in the model ensures that no ice accretes above the freezing point and no water film runs back below the freezing point. The compatibility relations strictly enforce these conditions on the solution. One way to write them is the following:

Compatibility Relations

$$h_f \geq 0 \tag{6}$$

$$\dot{m}_{\text{ice}} \geq 0 \tag{7}$$

$$h_f\tilde{T} \geq 0 \tag{8}$$

$$\dot{m}_{\text{ice}}\tilde{T} \leq 0 \tag{9}$$

The first compatibility relation is a natural assumption on the film thickness, as is the positivity of the air density for a compressible

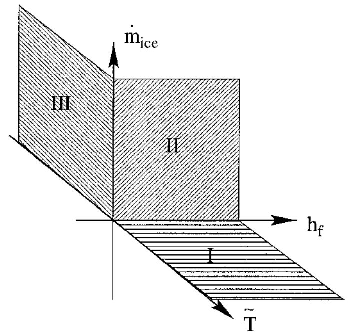

Fig. 2 Surface generated by the compatibility relations: I, running wet, no ice; II, glaze icing; III, rime icing.

flow. This first assumption is explicitly stated as there is no guaranty, at this time, that the model forces the film thickness to remain positive. The second compatibility relation, although not essential, just prevents the remelting of ice. It could probably be removed in case inward movements of the geometry (for ice remelting) can be properly handled. On the basis of the first two compatibility relations, the inequalities (8) and (9) are the true state (in)equations preventing undesirable freezing/runback behaviors above/below the triple point of water.

To investigate the well posedness of SWIM, one should first see if the compatibility relations lower the number of unknowns from 3 to 2. Indeed, as could be seen on Fig. 2, the regions of the $(h_f, \tilde{T}, \dot{m}_{\mathrm{ice}})$-icing space delimited by the compatibility relations generate a connected surface, called the icing surface. But two degrees of freedom are enough to describe a surface. It thus becomes clear that by restricting the solution of SWIM to the icing surface the number of unknowns matches the number of PDEs available.

The icing surface is composed of three contiguous quarters of plane. The first one, labeled I on Fig. 2, corresponds to a running wet-no ice growth condition above the freezing point. Some water may impinge on the walls, but no freezing occurs. Regions II and III correspond to wet and dry ice growth conditions, respectively. For wet or glaze ice growth a water film and some ice are present simultaneously, and the equilibrium temperature should be the triple point of water. For dry or rime ice growth all of the impinging water freezes; no water runback and the temperature of the interface could be below the freezing point. Of course, the solution is not restricted to any particular situation in space and in time, and switching is done automatically by the solver.

Up to now, we have looked at the following:

Icing Problem: Find h_f, $\tilde{T}$, and $\dot{m}_{\mathrm{ice}}$ satisfying the PDEs and the compatibility relations.

It is possible to switch the dependent variables. For example, for anti-icing applications, a possible variant would be the anti-icing problem.

Anti-Icing Problem: Assuming $\dot{m}_{\mathrm{ice}} = 0$, find h_f, $\tilde{T}$, and $\dot{Q}_{\mathrm{cond}}$ satisfying the three equations and such that $\dot{Q}_{\mathrm{cond}}$ minimizes the total heat flux

$$\int_{\mathrm{wall}} \dot{Q}_{\mathrm{cond}}(x)\,\mathrm{d}x$$

In this paper, we concentrate on the icing problem.

Naturally, SWIM has to interact with the other solvers because both the icing and the anti-icing problems are coupled with the air and droplets solutions. Figure 3 shows the code interactions together with the variables transferred between codes.

III. Computational Approach

Apart from the strategy used in SWIM, which is detailed next, the following methods are used for the numerical solution of the icing problem:

1) Each of the three systems of PDEs (air/droplet/SWIM) are solved independently from the others; selected variables are passed between solvers.

2) Both the flow and droplets solvers are weak-Galerkin finite element models on bricks, prisms, or tetrahedra.

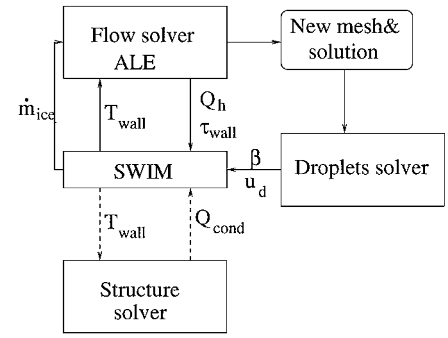

Fig. 3 Code interactions with SWIM.

3) A Newton-generalized minimal residual (GMRES) algorithm is used to solve each nonlinear system of equations.

4) The movement of the walls caused by ice growth is done in the flow solver through an arbitrary Lagrangian Eulerian (ALE) scheme,[10] using the instantaneous ice growth rate $\dot{m}_{\mathrm{ice}}$ to specify the wall nodes movement and could be complemented by a mesh optimization method.[11]

Special care has to be taken because the two PDEs of SWIM are expressed on the walls of the geometry, i.e., on a two-dimensional surface embedded in a three-dimensional one. So, the first derivatives, more precisely the div operator, of some unknowns have to be evaluated along the wall surface. A classical way to compute the divergence of some vector on a surface is through the introduction of a curvilinear coordinate system on the surface. To avoid the introduction of such curvilinear coordinates of an evolving (caused by ice growth) wall surface, the intrinsic definition of the div operator is applied:

$$\mathrm{div}\,\boldsymbol{u}(P) = \lim_{\mathcal{V}\to\mathcal{P}} \frac{\int_{\partial\mathcal{V}} \boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}s}{\mathrm{vol}(\mathcal{V})} \tag{10}$$

where $\mathcal{V}$ is an area element that shrinks to the point P on the surface.

The finite volume method is an application of that definition but at a discrete level. A finite volume cell is chosen as the area element $\mathcal{V}$ in Eq. (10) instead of a vanishing sequence of arbitrary area elements. Here a finite volume cell means a cell of the dual surface mesh, where the surface mesh is the hull of the three-dimensional mesh at the air-structure/ice shape interface and the dual surface mesh is the surface mesh obtained while connecting the barycenter of the surface mesh cells to the midedges of the cells (see Figs. 4 and 5).

The two PDEs of SWIM are discretized using a cell-vertex Roe finite volume scheme[12,13] based on h_f and $h_f\tilde{T}$ as solution variables. This gives

$$\mathrm{area}(C_i)\left(\frac{\partial h_i}{\partial t} - S_i^h\right) + \sum_j \left[\frac{u_{ij}\cdot n_{ij}}{2}\left(h_i^2 + h_j^2\right) - |u_{ij}\cdot n_{ij}|h_{ij}(h_j - h_i)\right] = 0 \tag{11}$$

$$\mathrm{area}(C_i)\left(\frac{\partial C_p h_i \tilde{T}_i}{\partial t} - S_i^T\right) + C_p \sum_j \left[\frac{u_{ij}\cdot n_{ij}}{2}\left(h_i^2\tilde{T}_i + h_j^2\tilde{T}_j\right) - |u_{ij}\cdot n_{ij}|h_{ij}(h_j\tilde{T}_j - h_i\tilde{T}_i)\right] = 0 \tag{12}$$

The summation is taken over all of the nodes j connected to the node i. The time derivatives could be discretized with any finite difference formula, provided it results in a fully implicit scheme.

The variables h_i and $h_i\tilde{T}_i$ are only needed to write down the Roe scheme. Indeed, their use as dependent variables is not recommended for the present system as the film thickness h_i will vanish in many icing situations, and $h_i\tilde{T}_i$ would also vanish, preventing the recovery of $\tilde{T}_i$ from $h_i\tilde{T}_i$ as when h_i is strictly positive. The variables h_i and $\tilde{T}_i$ seem to be a better choice. Even if h_i vanishes

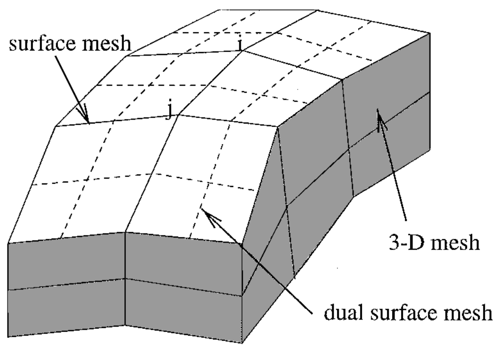

Fig. 4 Surface meshes for three-dimensional structured grids.

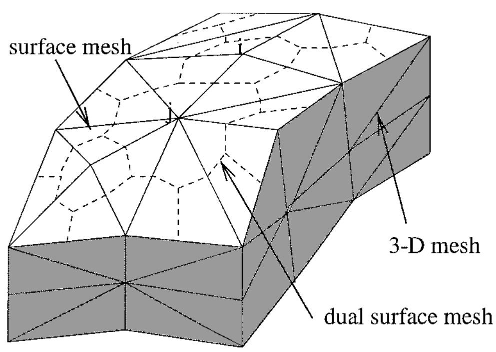

Fig. 5 Surface meshes for three-dimensional unstructured grids.

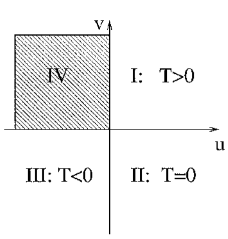

Fig. 6 Icing plane: I, running wet, no ice; II, glaze icing; III, rime icing; IV, dry air, no man's land.

at some grid points, the use of a fully implicit scheme will make possible the recovery of $\tilde{T}_i$ from the degenerated equations $S_i^h = 0$ and $S_i^T = 0$.

The instantaneous ice accretion rate $\dot{m}_i$ is also a dependent variable, although appearing only in the source terms. At each node three unknowns h_i, $\tilde{T}_i$, and $\dot{m}_i$ are thus to be computed, satisfying the system (11) and (12) and the compatibility relations. These latter could be addressed in their original form (6–9). Unfortunately, standard iterative methods like the Newton method cannot handle directly such a system of two equations and four inequalities. Even the application of nonlinear programming techniques[14] to the original problem is tedious and expensive as four generalized Lagrange multipliers would have to be added at each node to cope with the four inequalities.

Instead, we introduce a change of variable from a portion of the plane onto the icing surface. Figure 6 shows that portion of the plane, called the icing plane, with regions labeled I to III as for the icing surface in Fig. 2. An extra region, labeled IV, has appeared, which corresponds to dry air flight situations, i.e., without any droplets for an arbitrary temperature below or above freezing point.

The significance of each region is better understood by looking at the change of variable from the icing plane (u, v) to the icing surface:

$$\text{Region I}\quad \begin{cases} \tilde{T} = v^2 \\ h_f = u^2 \\ \dot{m}_{\text{ice}} = 0 \end{cases} \qquad \text{Region II}\quad \begin{cases} \tilde{T} = 0 \\ h_f = u^2 \\ \dot{m}_{\text{ice}} = v^2 \end{cases}$$

$$\text{Region III}\quad \begin{cases} \tilde{T} = -u^2 \\ h_f = 0 \\ \dot{m}_{\text{ice}} = v^2 \end{cases} \qquad \text{Region IV}\quad \begin{cases} \tilde{T} = v^2 - u^2 \\ h_f = 0 \\ \dot{m}_{\text{ice}} = 0 \end{cases}$$

This change of variable is invertible from the icing surface onto the region I ∪ II ∪ III of the icing plane. Moreover the transformation is differentiable, even on the u and v axis. These two features allow the solution of SWIM in (u_i, v_i) instead of $(\tilde{T}_i, h_i, \dot{m}_i)$ at each node. For example, using Newton's method the two unknowns (u_i, v_i) are updated with the residual of the two equations (11) and (12), without compromising the convergence of the iterative solver because the transformation is sufficiently regular. Whatever (u_i, v_i) is selected in the icing plane, the compatibility relations will be automatically satisfied.

Whenever required, the inverse transformation could be used to recover the physical variables $(\tilde{T}_i, h_i, \dot{m}_i)$ from (u_i, v_i). A difficulty arises from the fact that the inverse transformation is not defined in region IV, and the iterative solver could very well converge to values in that region. As just stated, the region IV corresponds to dry air situations, which are, in fact, already completely represented by the points on the portion of the u and v axis adjacent to this region. The region IV simply has to be avoided as otherwise the system (11) and (12) would be indefinite. Nonlinear programming techniques[14] could be used to avoid going out of regions I, II, and III.

IV. Numerical Results

Three test cases are presented, showing trends in the solution for SWIM and the potential capabilities of the model.

A. Water Runback over a NACA 0012 Airfoil

The first test case is a laminar flow a NACA 0012 airfoil at Mach number 0.399 and +15°C. Figure 7 shows the temperature contours of the flow. Figure 8 shows the three dependent variables of SWIM as well as derived variables. The variables are shown at four successive flow/SWIM iterations assuming 200 time steps into SWIM and full convergence to a steady solution in the flow solver. Intersolver communication is done as in Fig. 3.

The droplets impingement occurs within the first 10% along the chord. The water film will initially form in this impingement area, but will be advected all the way up to the trailing edge of the airfoil, as there is no flow separation.

The air temperature at infinity is +15°C, which gives a stagnation temperature of 24.6°C at the wall for this laminar flow. The temperature of the film is initialized with the stagnation temperature. The impingement area of the airfoil tends to be cooler because of

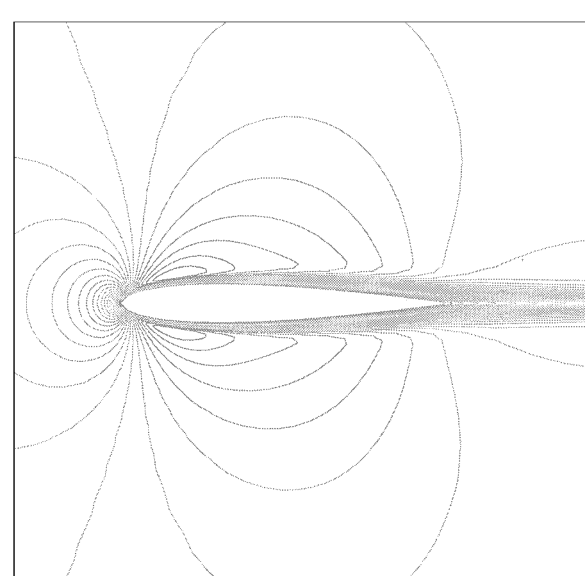

Fig. 7 Temperature contours for the flow around a NACA 0012 airfoil.

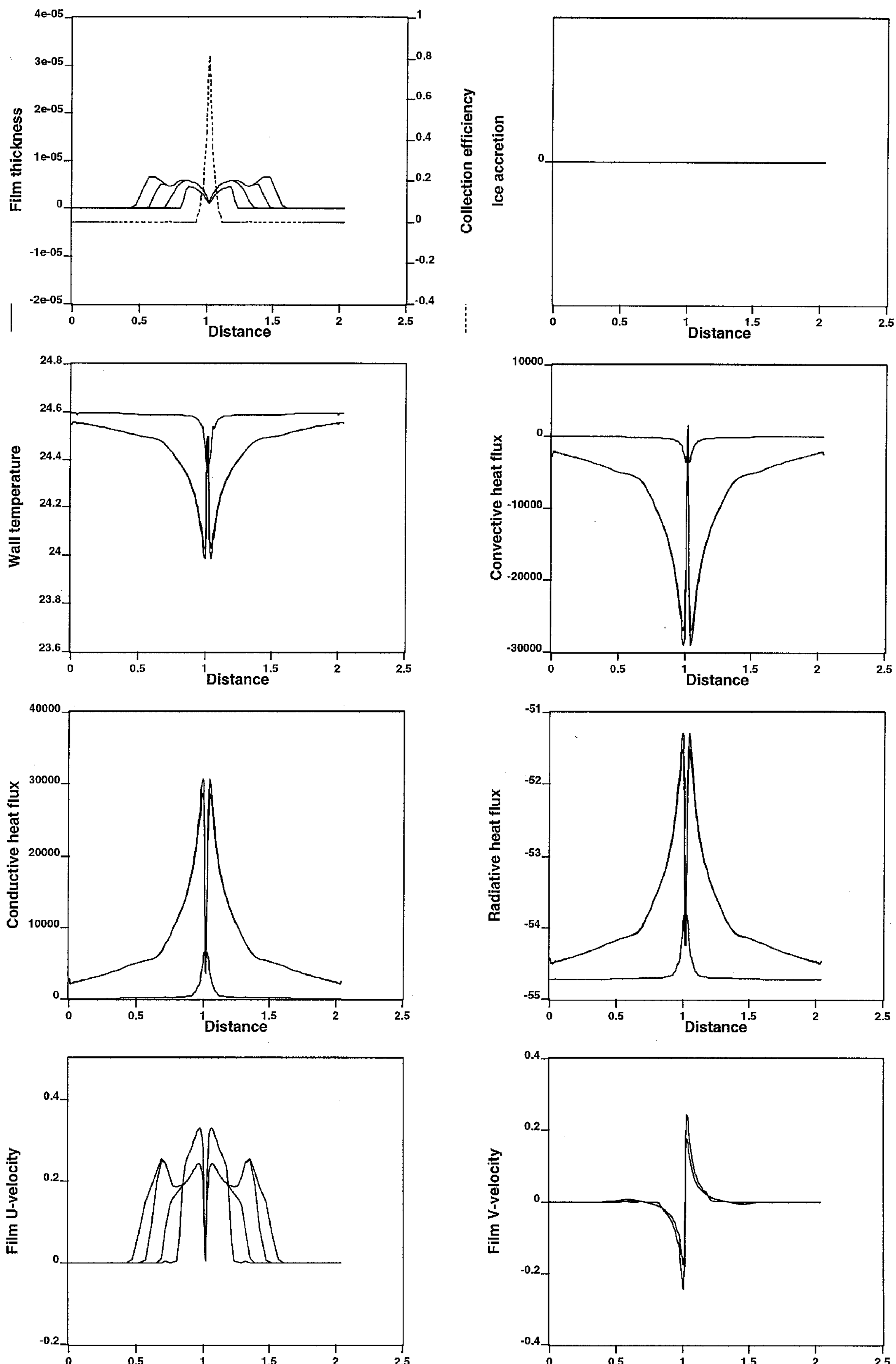

Fig. 8 Film runback over a NACA 0012 airfoil above freezing point. The variables are plotted against the arclength s. Flow conditions: T_∞ = +15°C, P_∞ = 90.76 KPa, U_∞ = 135.89 m/s, L_{ref} = 0.3302 m, LWC = 1 g/m^3, Δt = 2.43 ×10^{-6} s, and #time steps = 200, 400, 600, 800.

the impingement of droplets at $+15°C$. The film rapidly reaches the stagnation temperature as it flows aft the leading edge because it is assumed to be in conduction contact with a reservoir at that temperature. Of course, no ice accretes at that temperature.

The three next graphs of Fig. 8 shows heat source and sink terms. For this test case conduction appears to be of crucial importance for the convergence of SWIM. So, we had to implement a simple conduction model, assuming a linear temperature profile through a metal skin of a given thickness. The convection and conduction term are almost in equilibrium except maybe at the stagnation point, where convection is not as effective and conduction dominates. This might explain the peak in temperature at the stagnation point, even if the droplet impingement is a cooling factor in that area. A look at the film velocity shows that the water is almost stagnant at the stagnation point, increasing the response of the film to conduction. Radiative effects are not that important, as expected.

Finally, the film velocity shows patterns similar to the film thickness, as could be seen from the relation (3).

B. Dry Ice Accretion over a NACA 0012 Airfoil

The second test case is a laminar flow around a NACA 0012 airfoil at Mach number 0.420 and $-45°C$. Figure 9 shows the solution of SWIM and the derived variables after 100 time steps. All of the water freezes on impact, releasing more latent heat than the convective heat transfer and the sensible heat that goes into the ice and supercooled water temperature adjustment. A temperature increase is thus seen in the ice accretion region. The simple conduction model was not needed here.

C. Ice Accretion in a 90-deg Bend

Numerical results are shown on Fig. 10 for a Mach 0.1 laminar flow in a 90-deg bend. The air is coming from the left and follows the

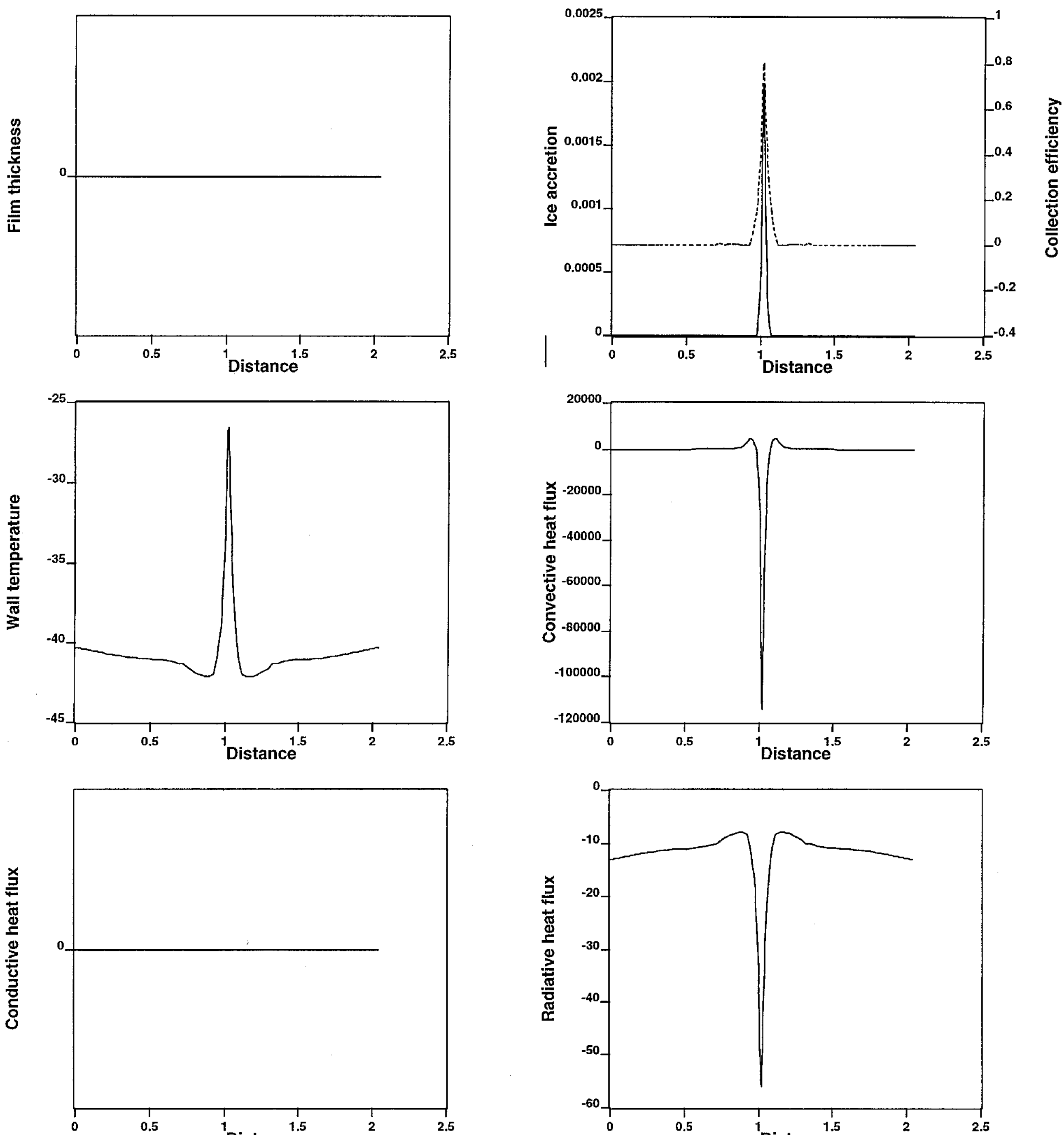

Fig. 9 Dry ice growth over a NACA 0012 airfoil. The variables are plotted against the arclength *s*. Flow conditions: $T_\infty = -45°C$, $P_\infty = 90.76$ KPa, $U_\infty = 127.30$ m/s, $L_{ref} = 0.3302$ m, $LWC = 0.04$ g/m^3, $\Delta t = 2.43 \times 10^{-6}$ s, and #time steps = 100.

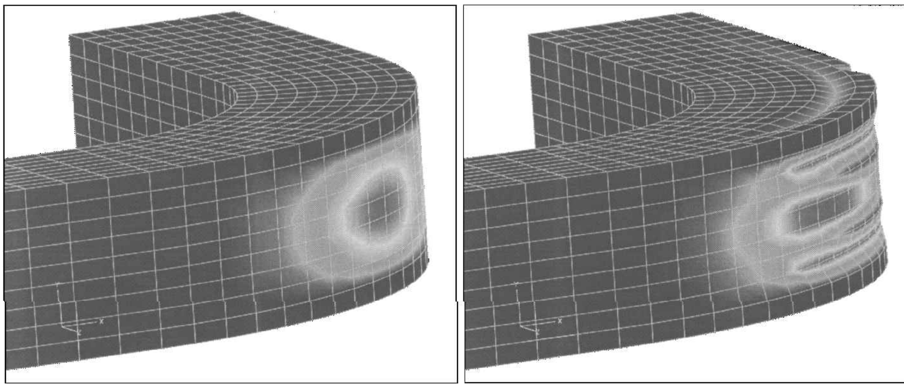

Fig. 10 Ice shape evolution with ALE: initial shape (left), iced bend (right).

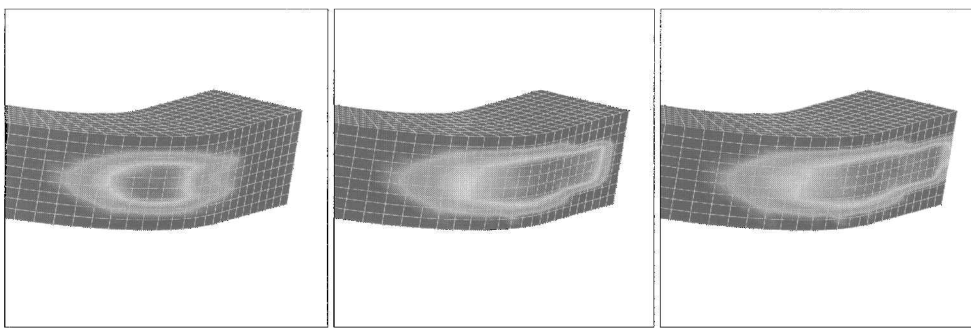

Fig. 11 From left to right: water film thickness after 50, 150, and 170 time steps.

bend. Because of their inertia, the droplets impinge on the outer wall, creating a high collection efficiency area. At a sufficiently low temperature the impinging water freezes right away on impact, deforming the wall as seen by the flow and the droplets solvers. An effect of the wall deformation is the evolution of the collection efficiency pattern, as could be seen by comparing the top and bottom pictures.

At a higher temperature a water film would grow as a result of the impinging droplets and start running back along the wall shear lines. Figure 11 shows the evolution of the film thickness at a temperature above the freezing point for the initial impingement pattern shown on Fig. 10.

V. Conclusion

A new equilibrium ice accretion model based on PDEs has been developed. This model includes some features for continuous film runback prediction on two- and three-dimensional geometries and for tight coupling with Navier–Stokes flow and droplets solvers. By using a PDE-based model, a distinction can be made between the physical modeling and the numerical resolution of the resulting problem. This allows the inclusion of the time derivatives in the model, thus carefully accounting for unsteady effects of the underlying phenomena.

Some numerical results have been presented for laminar flows, and coupling with a turbulence model (already implemented in FENSAP) is the next step. These results already show the potential of the method for simulating ice accretion on three-dimensional lifting surfaces, including water runback effects. This is something that has not been thoroughly investigated, at least from the numerical modeling standpoint, by the in-flight icing community. Of course, a lot of work is still needed for this new icing model to show its capabilities for a complete assessment of icing events. In particular, a more complete investigation of the tight coupling of the flow, droplets, and SWIM solvers as well as comparisons with experimental ice shapes are the next natural steps in evaluating the capabilities of the model.

Acknowledgments

The authors would like to thank Pratt and Whitney Canada, the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Fonds pour la Formation des Chercheurs et l'Aide à la Recherche (FCAR) for funding under which this work was supported.

References

[1]Ruff, G., and Berkowitz, B., "Users' Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)," NASA TR 185129, May 1990.

[2]Hedde, T., "Modélisation Tridimensionnelle des Dépôts de Givre sur les Voilures d'Aéronefs," Ph.D. Dissertation, Dépt. des Sciences de l'Atmosphère, Univ. Blaise Pascal, Clermont Ferrand, France, Dec. 1992.

[3]Wright, W., Gent, R., and Guffond, D., "DRA/NASA/ONERA Collaboration on Icing Research Part II-Prediction of Airfoil Ice Accretion," NASA TR CR-202349, May 1997.

[4]Potapczuk, M., "LEWICE/E: An Euler Based Ice Accretion Code," AIAA Paper 92-0037, Jan. 1992.

[5]Tran, P., Brahimi, M., Tezok, F., and Paraschivoiu, I., "Numerical Simulation of Ice Accretion on Multiple Element Configurations," AIAA Paper 96-0869, Jan. 1996.

[6]Al-Khalil, K., Keith, T., and Witt, K. D., "Further Development of an Anti-Icing Runback Model," AIAA Paper 91-0266, Jan. 1991.

[7]Bourgault, Y., Boutanios, Z., and Habashi, W. G., "Three-Dimensional Eulerian Approach to Droplet Impingement Simulation Using FENSAP-ICE, Part 1: Model, Algorithm, and Validation," *Journal of Aircraft*, Vol. 37, No. 1, 2000, pp. 95–103.

[8]Boutanios, Z., Bourgault, Y., Habashi, W., Isaac, G., and Cober, S., "3D Droplets Impingement Analysis Around an Aircraft's Nose and Cockpit Using FENSAP-ICE," AIAA Paper 98-0200, Jan. 1998.

[9]Croce, G., Habashi, W., Guèvremont, G., and Tezok, F., "3D Thermal Analysis of Anti-Icing Devices Using FENSAP-ICE," AIAA Paper 98-0193, Jan. 1998.

[10]Lepage, C., and Habashi, W., "Fluid-Structure Interactions Using the ALE Formulation," AIAA Paper 99-0660, Jan. 1999.

[11]Habashi, W., Fortin, M., Dompierre, J., Vallet, M.-G., and Bourgault, Y., "Certifiable CFD Through Mesh Optimization," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 703–711.

[12]Dick, E., "Introduction to Finite Volume Technique in Computational Fluid Dynamics," *Computational Fluid Dynamics*, edited by J. Wendt, A VKI Book, Brussels, Springer–Verlag, 1992, pp. 261–288.

[13]LeVeque, R., *Numerical Methods for Conservation Laws*, Lectures in Mathematics, Birkhäuser, Boston, Cambridge, MA, 1990, p. 214.

[14]Bertsekas, D., *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995, p. 780.