JOURNAL OF AIRCRAFT
Vol. 37, No. 1, January–February 2000

Three-Dimensional Eulerian Approach to Droplet Impingement Simulation Using FENSAP-ICE, Part 1: Model, Algorithm, and Validation

Yves Bourgault,* Ziad Boutanios,† and Wagdi G. Habashi‡
Concordia University, Montreal, Quebec H3G 1M8, Canada

To realistically compute three-dimensional droplet impingement on aircraft and engines, an Eulerian model for diphasic airflows containing water droplets is proposed as an alternative to the traditional Lagrangian particle tracking approach. The partial differential equations-based model is presented, together with details on the numerical methods and its algorithmic implementation in three dimensions within the finite element Navier–Stokes analysis package for icing. Code validations in two and three dimensions are presented in comparison with published NASA experimental impingement results, and numerical accuracy requirements are discussed.

Nomenclature

C_D = spherical droplet drag coefficient
c_u, $c_{\|u}$ = stabilization parameters for $\boldsymbol{u}$
c_α, $c_{\|\alpha}$ = stabilization parameters for α
d = droplet diameter
Fr = Froude number, $U_\infty / \sqrt{(Lg_0)}$
$\boldsymbol{f}$ = right-hand side of droplet momentum equation
g_0 = gravitational acceleration, 9.8 m/s^2
h_K = size of element K
K = droplet inertia parameter, $\rho d^2 U_\infty / 18 L\mu$
L = reference length of the geometry
N = space dimension; equal to 2 for two dimensions, 3 for three dimensions
$\boldsymbol{n}$ = normal vector to the domain (CAD) surface
$\boldsymbol{n}_h$ = normal vector to the surfacic grid
$\tilde{\boldsymbol{n}}_h$ = vector orthogonal (nonnormalized) to the surfacic grid
Re_d = droplet Reynolds number, $\rho_a d U_\infty |\boldsymbol{u}_a - \boldsymbol{u}| / \mu$
U_∞ = reference velocity
$\boldsymbol{u}$ = nondimensional velocity of droplets
$\boldsymbol{u}_a$ = nondimensional velocity of air
$\boldsymbol{u}^n$ = droplet velocity at the time step n
$\|\boldsymbol{u}\|_{\infty,K}$ = maximum of the Euclidean norm of the droplet velocity on element K
V_h = finite element space
$\boldsymbol{x}$ = parametric representation of the element face, $\boldsymbol{x}(u, v)$
x_i = surfacic grid node
$\boldsymbol{x}_u$, $\boldsymbol{x}_v$ = u and v derivatives of the parametric function
α = droplet volume fraction
α^n = droplet volume fraction at the time step n
β = local collection efficiency
Δt = local time step seen as a function of grid nodes
Δt_K = local time step on element K
$\Delta t_{\max}$ = truncation maximal local time step allowed
ϵ = turbulent dissipation
κ = turbulent kinetic energy
λ = target Courant–Friedrichs–Lewy number
μ = dynamic viscosity of air
ρ = density of water
ρ_a = density of air
τ_K = streamline upwinding Petrov–Galerkin stabilization function on element K
φ = mass finite element test functions
ψ = momentum finite element test functions
Ω = spatial flow domain

Received 11 December 1998; revision received 11 June 1999; accepted for publication 12 June 1999.

*Research Assistant Professor, Computational Fluid Dynamics Laboratory, Department of Mechanical Engineering; currently at University of Ottawa, Ottawa. Member AIAA.

†Graduate Research Assistant, Computational Fluid Dynamics Laboratory, Department of Mechanical Engineering; currently at Real Numerix, Ltd., Montreal. Member AIAA.

‡Director, Computational Fluid Dynamics Laboratory, Department of Mechanical Engineering. Associate Fellow AIAA.

I. Introduction

AIRCRAFT in-flight icing remains an operational and certification issue that is far from being completely resolved. Recent trends in the aviation industry, such as the advent of lower altitude regional aircraft, have increased the potential for icing incidents and accidents. In addition, the suspicion about the dramatic effects of supercooled large droplets has forced the icing community to reconsider certification minimum requirements and to re-evaluate the icing environment and the simulation tools used to comply with those requirements. Such concerns have reanimated the research, as well as the review of engineering and certification efforts to tackle the in-flight icing problem.[1]

One of the urgent needs identified is for improved numerical simulation that could lead to better designs and shorter certification through the investigation of a broader catalog of icing scenarios. Such improved simulation not only calls for better physical modeling of icing phenomena, but also for the introduction of more modern computational fluid dynamics (CFD) approaches.[2] The need to investigate complex three-dimensional geometries such as engine inlets, nacelles, swept wings, empennages, radomes, and even complete aircraft, however, task the geometric capabilities of current icing simulation codes.

Today's icing physical models, numerical methods, software implementation, and design integration methods have some catching up to do with the achievements in aircraft flow simulation and in the numerical design of gas turbine engines. To demonstrate this assertion, one only has to compare the general review of icing numerical models given in a recent AGARD report[2] to a similar review, done at about the same time, of simulation techniques for propulsion systems.[3] Advanced CFD algorithms and turbulence models, conjugate heat transfer (simultaneous solution of fluid–solid heat transfer), CAD-based representation, concurrent engineering design approaches, multidisciplinary optimization, high-performance computing, advanced scientific visualization, now part of the daily culture of the aerodynamic design process, have not yet completely found their way in the icing analysis–design–certification loop.

Icing is still being treated as a post-aerodynamic design exercise, leading to a disconnection of the aerodynamics and icing. This

segregation has thus led icing simulation codes to evolve independently from the CFD class of codes most commonly used in the aerodynamic design process in industry, proprietary or commercial, leading to difficulties in formulating a combined aeroicing approach or even creating easy interfacing via independent modules.

A comprehensive in-flight icing simulation capability requires, at a minimum, codes or modules for droplet impingement limits, accreted ice shape prediction, de- and anti-icing heat loads, and performance degradation calculations. Flow analysis for impingement calculations are currently being carried out by incompressible potential (panel methods) or compressible inviscid (Euler) methods, interactively coupled with boundary-layer (BL) techniques.[4,5] Droplet tracking is done through a Lagrangian approach, some of whose limitations will be detailed later. Accreted ice shapes are also predicted with inviscid flow plus BL-based icing codes[6–8] or experimentally in icing tunnels.[7–12] Heat transfer calculations, as well as performance degradation assessments, are also mostly carried out through viscous–inviscid interaction codes. In addition, the thermodynamic analysis of ice accretion is mostly done in two dimensions, by using a two-dimensional code along cuts or the surface streamlines.[13,14] Finally, note that when a method is liberally labeled three dimensional in icing research it most likely indicates an attempt at quasi three dimensionality through a series of patches and compromises, somewhat different from what aerodynamicists understand a truly three-dimensional analysis to be.

Thus, no capability readily exists to simulate the impingement–accretion–de-icing processes as the coupled, three-dimensional, viscous, compressible, turbulent, unsteady phenomena they truly are. The argument for the continuing use of viscous–inviscid interaction solvers is that satisfactory results have been obtained on airfoils,[8,15–17] and thus, the high cost associated with turbulent flow simulations is not justified, especially in three dimensions.[6] However, lack of cost-effectiveness arguments, repeated over two decades, now sound hollow and have a natural way of being seen to be out of context: Efficient Navier–Stokes simulation techniques are here, now, and are here to stay.

Many serious pitfalls are hidden in today's simplistic icing simulation approaches, and one can only hope to scratch the surface in this Introduction. The computation of droplet impingement by a Lagrangian tracking approach introduces a different numerical technique than the flow solver, with the consequence that some icing codes can require up to five grids.[4] This hit-or-miss method requires the painstaking launching of individual droplets upstream of an object and tracking each droplet to determine if, and then where, it hits the object. Panel methods are often used to calculate the flow because they are cost effective for surface velocities. However, a Lagrangian tracking method requires the determination of the velocity along a droplet's path, necessitating the summation of all panels' contributions, on all surfaces, for each point of the arbitrary path, something that is fairly expensive. Ways and means can be found to speed up the velocity spatial recovery from the panels, such as the use of Cartesian grid to precalculate the velocity vectors in three-dimensional space prior to the Lagrangian droplet analysis. Then the velocities can be interpolated from the Cartesian grid along the trajectories whenever needed. Nevertheless, this remains an inefficient patch to continue using an imposed method. Furthermore, once particles hit the geometry, the collection efficiency is recovered through an averaging process that has to be carefully defined and implemented in three dimensions so as to avoid an inordinate number of particles to be launched.[4,13] The accuracy of determining impingement limits also depends on the density and placement of launched particles and on the ability to calculate impingement on components in the aerodynamic shadow of others, such as the main wing or a flap in the shadow of their deployed slat during climb or descent. Finally, the traditional plague of panel-based methods is that results can seldom be improved by mesh refinement (panels would blow up), making the accuracy of a particle trajectory as it approaches the surface singularities, especially at the crucial limits of impingement, at best, doubtful.

As for ice accretion, most available analyses are based on very simple unidimensional control volume balance approaches that cannot accurately account for many of the geometrically induced features of the complex freezing–melting–runback–beading process. For heat loads, convective heat transfer coefficients used in icing codes are based on tuned correlations[6,18] that cannot cover the entire spectrum of icing conditions and that are not readily available for three-dimensional geometries. Such limited approaches may have served their course in giving qualitative assessments of heat transfer in simple geometries and must now cede their place to established modern methods for quantitative predictions of complex de- and anti-icing devices.

As demand grows for realistic and reliable icing simulation on complex three-dimensional geometries and passages, it is increasingly evident that three-dimensional Navier–Stokes flow and heat transfer approaches are the answer. This is more true if the focus is not only design but also optimization. This class of solvers would also make it possible to use the same code for external icing (wing, fuselage, nose cone, radomes, probes, etc.) internal icing (engine), and, ultimately, to analyze the aircraft, engine or both, as systems. CFD advances in large-scale solvers, turbulence modeling and their implementation, unstructured meshes and automated adaptation, are the de facto design norm and are rapidly becoming the aeronautics industry's norm. Their introduction in the icing arena can only go toward a much needed tighter integration of icing-related phenomena with the aerodynamic design process.

Thus, it is abundantly clear that the eclectic array of simulation tools now available to analyze the major aspects of icing will eventually be replaced by a modular multidisciplinary approach, whose cornerstones could be anticipated to be the following.

1) Multiphase flow solvers (air/droplets mixture) for impingement could replace the potential flow plus Lagrangian tracking approach.

2) A more complete ice accretion model can account for all mass and heat transfer mechanisms and accurately predict ice shapes in three dimensions (effectively two dimensions, on the surface of the ice-accreting object). In addition, the model would be an unsteady one, continually tracking the ice-accreting free surface and automatically modifying the mesh through moving grids.[19]

3) Use conjugate heat transfer to calculate simultaneously the flow inside and outside the wing, as well as conduction through its skin. This approach would dispense with correlations and eventually lead to major optimization benefits.

4) Use Navier–Stokes analysis to estimate performance degradation, with accuracy sharpened via automatic mesh adaptation.

In a series of successive papers, the CFD Laboratory proposes to present the scientific advances incorporated in the modules of the finite element Navier–Stokes analysis package for icing (FENSAP-ICE). The two-part first paper addresses the development and validation of the Eulerian droplet impingement model as an alternative to the Lagrangian particle tracking techniques. The approach is validated through comparison to published experimental results on a two-dimensional cylinder and a three-dimensional sphere. Part 2 will examine the connection of the simulation methodology to actual flight conditions by analyzing impingement patterns on the nose and cockpit of an aircraft, the Convair-580 used by the Canadian Atmospheric Environment Service to collect in-flight icing data, for the Canadian freezing drizzle experiment held in the winter of 1997–1998.

II. Three-Dimensional Eulerian Model for Droplet Impingement

The droplet impingement module in FENSAP-ICE, DROP3D, is based on an Eulerian model, has been introduced for two-dimensional applications by Bourgault et al.,[20] and is briefly reviewed here for completeness. This is essentially a two-fluid model consisting of the set of Navier–Stokes or Euler equations, augmented by droplet-related continuity and momentum equations. The latter are, in nondimensional form, respectively,

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \boldsymbol{u}) = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = (C_D Re_d / 24K)(\boldsymbol{u}_a - \boldsymbol{u}) + (1 - \rho_a / \rho)(1/Fr^2)\boldsymbol{g} \tag{2}$$

where the variables $\alpha(x, t)$ and $\boldsymbol{u}(x, t)$ are mean values, respectively, of the water volume fraction and of the droplet velocity over a small fluid element around the location x at time t. In these equations, $C_D = (24/Re_d)(1 + 0.15Re_d^{0.687})$ for $Re_d \leq 1000$ is an empirical drag coefficient for spherical droplets. The first term on the right-hand side of Eq. (2) represents the air drag force on the droplets, whereas the second term represents the buoyancy and gravity forces. The two-fluid model assumes spherical droplets of one size, usually chosen equal to the median volume diameter (MVD) of the sample size distribution. Handling multiple droplet sizes has been done by duplicating the number of equations solved, using the system once for each class of diameters. No collision or mixing between the droplets is accounted for because it can be shown not to be important for icing situations.

The Lagrangian tracking method requires the integration of individual droplet trajectories from a computational domain inlet to the impacted geometry's surface, something that might give rise to elaborate algorithmic developments and large computational resources for complex three-dimensional geometries. Indeed, a large number of particles may have to be launched for impingement analysis on geometries with small intricate details. On the other hand, an Eulerian model based on partial differential equations is discretized with the same numerical techniques as the companion (field) flow solver and uses the same grid. The collection efficiency recovery on the walls is as simple as the pressure coefficient computation from the flow solution. It involves standard flow postprocessing. Of course, for simple geometries such as single element airfoils, a Lagrangian method provides the collection efficiency by tracking 20–40 droplets, as opposed to an Eulerian approach that requires solving the system of Eqs. (1) and (2) over the whole grid. In such simple situations, Lagrangian tracking would be the most efficient. However, a complete contrast of the performance of both methods must include considerations on the complexity of the geometry and whether a local assessment of the impingement, for example, only on a given section of a wing, or a more global picture, for example, over a whole aircraft, is needed. Each method has the potential to give a partial or a global picture, but one of the two methods will be more efficient for a given purpose on a given geometry with a required amount of detail.

One advantage of the Eulerian approach is its ability to do a reverse analysis (where does a droplet that impinges come from?) directly in the postprocessor by observing the streamlines of the droplet velocity field ending at a given location on the wall (in the steady case at least). This is to be compared to the multiple shooting methods needed with the Lagrangian tracking; that is, the reverse analysis could only be done by computing a droplet trajectory in a direct fashion starting from a guessed initial position, updating the initial position on the basis of the difference between the expected and the actual impingement locations, and repeating these operations until the expected and actual final positions agree.

III. Algorithmic Implementation

The solution α and $\boldsymbol{u}$ of the Eulerian droplet model are made based on a precalculated inviscid or viscous airflow. However, using an inviscid flow may result in inaccurate impingement patterns in situations where viscous effects are dominant, for example, with separated flows. In many cases, however, such as those close to cruise conditions for aircraft, neglecting the BL should have negligible effects on the quality of the collection efficiency, as will be demonstrated by the validation test cases.

The general approach to impingement computations is as follows: Starting from a CAD model, generate a structured or unstructured grid with a mesh generation package (in our case we use ICEM CFD); compute a fully converged viscous or an inviscid flow solution with a CFD analysis package (in our case, we use our proprietary FENSAP); by using the same mesh as for the airflow solution, compute droplet solutions and impingement patterns with DROP3D (seen as a plug-in module to FENSAP or any other flow code); then use a flow visualizer to analyze the droplet solution and impingement patterns. For a distribution of droplet sizes, one computes as many droplet solutions as there are size classes and postprocesses the collection efficiencies from all classes to recover a compound collection efficiency.

In the following, the different features of the flow solver are first reviewed, and then the numerical ideas behind the implementation of DROP3D are presented: in particular, the stabilized finite element used, the three-dimensional recovery of the collection efficiency, the iterative solver, and the local time-stepping techniques.

A. FENSAP Navier–Stokes Module

FENSAP has capabilities for analyzing steady or unsteady compressible turbulent flows, with or without recirculation regions, and can revert to inviscid flow simulation as an option, whenever this assumption proves sufficient in a given situation. The same flexibility is built in with respect to two-dimensional vs three-dimensional simulations: the code being a fully three-dimensional one but with a capability to analyze two-dimensional flows. The following algorithmic developments in FENSAP make a three-dimensional Navier–Stokes approach affordable and cost effective.

1) There are capabilities for structured and unstructured grids, based on hexahedral/tetrahedral/prismatic finite elements, with the geometrical flexibility needed for complex industrial applications.

2) Implicit time-stepping methods are used combined with efficient iterative solvers and preconditioners (such as an algebraic multigrid preconditioner) with optimized memory requirements.

3) Large portions of the code have shared-memory parallelization.

4) A κ–ϵ high-Reynolds number turbulence model with logarithmic finite elements at the walls considerably reduces the size of the necessary grids for turbulent flows. In addition, low-Reynolds number models are available to examine the near-wall detailed physics of flows over surfaces with sand-paper roughness.[9]

5) There is arbitrary Lagrangian–Eulerian formulation for problems with changing boundaries,[21] such as in ice-growth situations.

Moreover, the code is interfaced with an efficient anisotropic mesh optimization methodology (MOM3D) package, also developed at the CFD Laboratory, that has been shown to give highly accurate user-, solver-, and initial mesh-independent results in two-[22] and three dimensions.[23]

B. DROP3D: Finite Element Method

A finite element Galerkin formulation is used to numerically discretize Eqs. (1) and (2), with a streamline upwind Petrov–Galerkin (SUPG)[24] term added. In the Galerkin approach, given a mesh over a domain Ω and the space V_h of continuous piecewise linear elements (quasi linear for bricks), the volume fraction $\alpha^n \in V_h$ and the droplet velocity $\boldsymbol{u}^n \in V_h^N$ at time t^n are solutions of

$$\int_\Omega \left[\frac{\alpha^n - \alpha^{n-1}}{\Delta t} + \nabla \cdot (\alpha^n \boldsymbol{u}^n) \right] \varphi \, \mathrm{d}x + a_\alpha(\alpha^n, \varphi) = 0 \tag{3}$$

$$\int_\Omega \left[\frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n \right] \cdot \psi \, \mathrm{d}x + a_u(\boldsymbol{u}^n, \psi) = \int_\Omega \boldsymbol{f} \cdot \psi \, \mathrm{d}x \tag{4}$$

for all $\varphi \in V_h$, $\psi \in V_h^3$. Here, $\boldsymbol{f}$ are the drag and gravity forces on droplets and, thus, depend on $\boldsymbol{u}$ and $\boldsymbol{u}_a$.

The expressions $a_\alpha(\cdot, \cdot)$ and $a_u(\cdot, \cdot)$ are the SUPG stabilization terms just mentioned. These terms are written as

$$a_\alpha(\alpha, \varphi) = c_\alpha \sum_K \int_K \left[\frac{\alpha^n - \alpha^{n-1}}{\Delta t} + \nabla \cdot (\alpha^n \boldsymbol{u}^n) \right] \tau_K (\boldsymbol{u}^n \cdot \nabla) \varphi \, \mathrm{d}x \tag{5}$$

$$a_u(\boldsymbol{u}, \psi) = c_u \sum_K \int_K \left[\frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n - \boldsymbol{f} \right] \cdot \tau_K (\boldsymbol{u}^n \cdot \nabla) \psi \, \mathrm{d}x \tag{6}$$

with τ_K equal to $h_K / \sqrt{(1 + |\boldsymbol{u}|^2)}$ and c_α and c_u parameters chosen by the user. It is mainly the volume fraction that requires some

extra stabilization because the velocity itself suffers little numerical instability. As a result, the parameter c_u is usually chosen smaller than c_α, with c_u usually in [0.1, 10] as opposed to c_α in [1, 100] but very often with $c_\alpha \approx 2c_u$. The oscillations on the volume fraction are mainly due to the highly nonlinear convective nature of the Eulerian droplet model.

By examining three-dimensional results, it becomes evident that artificial viscosity strictly along the droplets streamline direction, as provided by an SUPG method, may not be enough for some situations. An example occurs when a relatively coarse grid is not aligned with a contact discontinuity appearing in the droplet solution. In that case, the volume fraction of water is discontinuous while crossing the contact discontinuity, but the droplet velocity is continuous and aligned with that curve. The result of the SUPG stabilization is the addition of artificial viscosity along the discontinuity curve only, without any crosswind diffusion, although some stabilization effects may still be needed in that direction. For these very rare occasions where crosswind diffusion is needed (as observed in practice), discontinuity-capturing terms[25,26] have been added to the finite element formulation.[27] They are rarely used in our calculations and, when used, are used with great moderation.

The terms $a_{\|\alpha}(\cdot, \cdot)$ and $a_{\|u}(\cdot, \cdot)$ are the discontinuity-capturing crosswind diffusion terms and are expressed as

$$a_{\|\alpha}(\alpha^n, d\varphi) = c_{\|\alpha} \sum_K \int_K \left[\frac{\alpha^n - \alpha^{n-1}}{\Delta t} + \nabla \cdot (\alpha^n \boldsymbol{u}^n) \right] \tau_K (\boldsymbol{u}^n_{\|\alpha} \cdot \nabla) \varphi \, dx \quad (7)$$

$$a_{\|u}(\boldsymbol{u}^n, d\psi) = c_{\|u} \sum_K \int_K \left[\frac{\boldsymbol{u}^n - \boldsymbol{u}^{n-1}}{\Delta t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n - \boldsymbol{f} \right] \tau_K (\boldsymbol{u}^n_{\|u} \cdot \nabla) \psi \, dx \quad (8)$$

where $c_{\|\alpha}$ and $c_{\|u}$ are user-specified parameters as multiples of c_α and c_u, respectively and $\boldsymbol{u}^n_{\|\alpha}$ and $\boldsymbol{u}^n_{\|u}$ are calculated using the expression

$$\boldsymbol{u}^n_{\|x} = \begin{cases} \dfrac{(\boldsymbol{u} \cdot \nabla x^h)}{|\nabla x^h|_2^2} \nabla x^h & \text{if} \quad \nabla x^h \neq 0 \\ 0 & \text{if} \quad \nabla x^h = 0 \end{cases} \quad (9)$$

C. DROP3D: Recovering the Collection Efficiency

An important parameter that controls the accretion of ice on a surface is the local collection efficiency β, that is, the normalized flux of water on the surface. The droplet solution yields α and $\boldsymbol{u}$ everywhere in the solution domain, and the surface collection efficiency β can then be calculated as

$$\beta = -\alpha \boldsymbol{u} \cdot \boldsymbol{n} \quad (10)$$

where $\boldsymbol{n}$ is the surface normal into the computational domain at mesh points on solid boundaries.

There is a difficulty in defining the normal $\boldsymbol{n}$ at a mesh point on the wall. On an element face, the normal vector comes naturally into play and is uniquely defined as follows:

$$\boldsymbol{n} = \frac{\boldsymbol{x}_u \times \boldsymbol{x}_v}{\|\boldsymbol{x}_u \times \boldsymbol{x}_v\|} \quad (11)$$

where $\boldsymbol{x} = \boldsymbol{x}(u, v)$ is a parametric representation of that face and $\boldsymbol{x}_u$ ($\boldsymbol{x}_v$) is the derivative of the parametric function with respect to u (v). It is implicitly understood through that definition that the parameterization of the surface is continuously differentiable. For a discretized boundary, however, at least three element (wall) faces meet at any grid node on a wall, preventing in most cases the continuity of the parameterization derivatives on edges between adjacent faces. In the two-dimensional implementation of the Eulerian model (DROP2D), collection efficiencies have been computed at the center of the edge on the solid boundaries and then transformed into values at the wall nodes by averaging the values on adjacent edges using the length of the edges as weighting factors. That strategy could probably be implemented in three dimensions, but a simpler technique has been adopted.[27]

To turn around the normal discontinuity problem in three dimensions, a discrete normal $\boldsymbol{n}_h$ has been defined at the grid node x_i as the normalized vector,

$$\boldsymbol{n}_h = \tilde{\boldsymbol{n}}_h / \|\tilde{\boldsymbol{n}}_h\| \quad (12)$$

with

$$\tilde{\boldsymbol{n}}_h = \sum_{\substack{\partial K \text{ neighbor of } x_i \\ \partial K \text{ on the wall}}} \boldsymbol{a}_i \times \boldsymbol{b}_i \quad (13)$$

and ∂K a boundary face of the element K, and with $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ the two edges on that face starting from x_i such that their cross product points into the computational domain. The vector $\tilde{\boldsymbol{n}}_h$ is nothing but the weighted average of the normal vectors of all of the wall faces around the node x_i, with the weights being equal to the area of the faces. Hence, by refining of the grid, the vector $\boldsymbol{n}_h$ converges to the actual normal $\boldsymbol{n}(x_i)$, except on sharp edges of a CAD surface where the continuous normal $\boldsymbol{n}$ is not defined, although the discrete normal $\boldsymbol{n}_h$ is. On such sharp edges of a CAD model one should expect $\boldsymbol{n}_h$ to be at best the average of the normal vector on both sides of the edge.

By replacing the continuous normal $\boldsymbol{n}$ by the discrete normal $\boldsymbol{n}_h$ in Eq. (10), a good approximation of the collection efficiency can be computed. According to the discussion on the value of $\boldsymbol{n}_h$ on sharp edges of a CAD geometry, some precaution should be taken on such sharp edges when interpreting the computed collection efficiency.

D. Iterative Solution and Local Time Stepping

The finite element formulation [Eqs. (3) and (4)] is implicit in time and requires the solution of a nonlinear system at each time step. The system is linearized using a Newton–Raphson method. Usually, a linear system, whose matrix is the Jacobian of the nonlinear system of equations, is solved for each Newton step. However, instead of computing this Jacobian directly, an iterative solver is used that requires only the product of the Jacobian with a vector, and then this matrix–vector product is approximated directly by a finite difference formula. This has been possible because the linear system is solved through a generalized conjugate gradient iterative method based on descent directions, in our case the generalized minimal residual (GMRES) method. The combination of the Newton–Raphson with the finite difference approximation of the Jacobian-vector product is often called the nonlinear GMRES method.[28]

The nonlinear GMRES method is used in DROP3D in conjunction with a diagonal preconditioner. In general, 1–3 Newton iterations, with 20–50 descent directions per time step, are sufficient to guarantee the convergence of the iterative solver at each time step. With such strategy it is possible to tackle three-dimensional test cases (engine inlet and nacelle, aircraft nose and cockpit, wing, etc.) with grids of several hundred thousand nodes using less than half a gigabyte of RAM.

A steady solution of the Eulerian model is reached by iterating in time. Steadiness rapidly occurs in the impingement area simply because the information travels along the streamlines from upstream to downstream for the Eulerian model and the exposed surfaces are located upstream. The critical region for convergence in time to the steady state is the wake behind the geometry, where the water droplets are more progressively, and hence more slowly, washed out.

To attain the steady-state faster over the entire computational domain, a local time-stepping strategy has been implemented. It consists in computing a local time step Δt_K on each element K, for example, using

$$\Delta t_K = \frac{\lambda h_K}{\max(\|\boldsymbol{u}\|_{\infty,K}, \lambda h_K / \Delta t_{\max})} \quad (14)$$

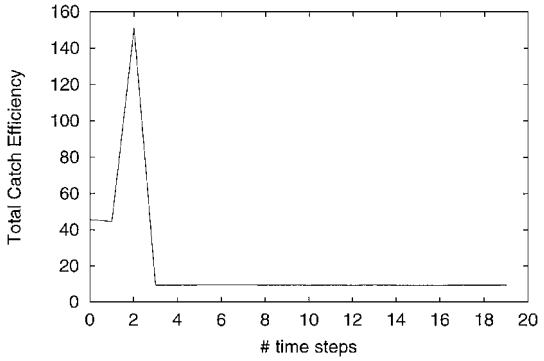

Fig. 1a PTCE.

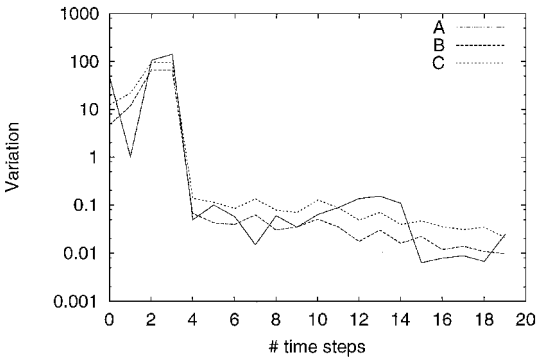

Fig. 1b Variation of the TCE line A, maximum line B, and Euclidean line C norm of the variation of the local catch efficiency from time step to time step.

where λ is the target Courant–Friedrichs–Lewy number and $\Delta t_{\max}$ is a truncation maximal time step. Both λ and $\Delta t_{\max}$ are specified by the user, in practice of the order of 2–1000 and 100–1000, respectively, depending on the stiffness of the problem.

The local time step is more or less set so that the information associated with the solution of the Eulerian model travels locally through λ elements at each time step. Before its use in the scheme, Eqs. (3) and (4), the time step Δt_K is locally transported to neighboring nodes by averaging the Δt_K over all of the element K neighboring a node using the volume of the element as a weighting factor.

Figure 1 shows the evolution of the total catch efficiency (TCE) as a function of the number of time steps fot the two-dimensional cylinder test case, where each time step is 10-s CPU time. The TCE is the integral of the collection efficiency coefficient over the whole surface. In other words, the TCE is the total mass flux of water impinging on the geometry, scaled by the liquid content of the cloud (LWC) $\times U_\infty$. By comparing the variation of the TCE from time step to time step with the norm (maximum and Euclidean) of the local catch efficiency again from time step to time step (Fig. 1b), it should be expected that the TCE is a good indicator of the code's convergence. That is, the convergence of the local catch efficiency seems to be closely connected with the convergence of the global catch efficiency. As can be seen from the graphs, four local time steps are enough to guarantee the convergence of the TCE for that simple example. The variation of the TCE from time step to time step is of the order of 0.1% of the final TCE only after the first four time steps. Usually some more iterations are done to account for potential local variation of the collection efficiency that may not be seen in the TCE.

To further improve the computing efficiency of DROP3D, a grid coloring scheme has been implemented to parallelize the assembly of the residual of the nonlinear system. This has proved sufficient on shared-memory computers with less than 16 CPUs, inasmuch as most of the effort is spent in computing residuals inside the nonlinear GMRES routine.

IV. Validation Test Cases

Two test cases from the literature have been selected to validate the impingement code. The first one is a two-dimensional cylinder taken from Ref. 5, and the second one is a three-dimensional sphere taken from Bidwell and Mohler.[29] These test cases were selected because the geometries are nonproprietary and fairly accurate experimental data are available in the public domain. The matrix of the validation test case conditions is detailed in Table 1.

The two-dimensional cylinder and three-dimensional sphere geometries were built with B-spline surfaces and curves inside circular domains. To do so, the CAD system used is ICEM-CFD's DDN module, which is an integral part of a CAD/mesher package commercialized by ICEM CFD Engineering. Inviscid airflow numerical solution were obtained for the test cases using FENSAP, as only droplet impingement is investigated in this paper. Of course, as stressed in the Introduction, an inviscid flow solution would not be sufficient for a complete icing analysis.

The meshes used for the calculations consist of bilinear hexahedral elements: a 17,168-node mesh for the cylinder and a 201,735-node mesh for the sphere (see Fig. 2). Inviscid flow solutions have been obtained on both grids, and results are presented in the following subsections. Droplet solutions presented in this section have been calculated for MVD solutions and Langmuir-D distributions. The complete details on the validation are given by Boutanios.[27]

A. Two-Dimensional Cylinder

DROP3D was validated in two dimensions using experimental results from Ref. 5 for impingement data on the surface of a two-dimensional cylinder. The air and droplet flow parameters are given in Table 1 in the column labeled "Cylinder."

Figure 3 shows the pressure isolines for the inviscid airflow around the two-dimensional cylinder. Figure 4 shows the collection efficiency profiles for the MVD = 16 μm solution and the resulting solution of a Langmuir-D distribution with an MVD of 16 μm, respectively. The details of the Langmuir-D distribution are given in Table 2. Not much difference can be seen, and the two solutions are indeed similar. However, should one plot the collection efficiency on the surface of the cylinder for both cases and compare to experimental results,[5] some differences come to light. Such a plot is shown in Fig. 5, and it can be seen that the Langmuir-D distribution is slightly more effective than the MVD solution in matching the experimental data. The solution based on the Langmuir-D distribution captures both maximum collection efficiency and impingement limits values and stays within the experimental repeatability range. The MVD solution, on the other hand, overestimates the maximum impingement value, underestimates the impingement

Table 1 Matrix of validation test cases

Parameters	Cylinder	Sphere
Reference length, m	0.1016	0.1504
Airspeed, m/s	80	75
Static temperature, °C	12	7
Static pressure, Pa	89,867	95,840
MVD, μm	16.0	18.6

Table 2 Langmuir-D distribution of droplet diameters with an MVD of 16 μm as used for the two-dimensional cylinder test case

Percentage LWC	Ratio of diameters	Droplet diameter, μm
5	0.31	5.0
10	0.52	8.3
20	0.71	11.4
30	1.00	16.0
20	1.37	21.9
10	1.74	27.8
5	2.22	35.5

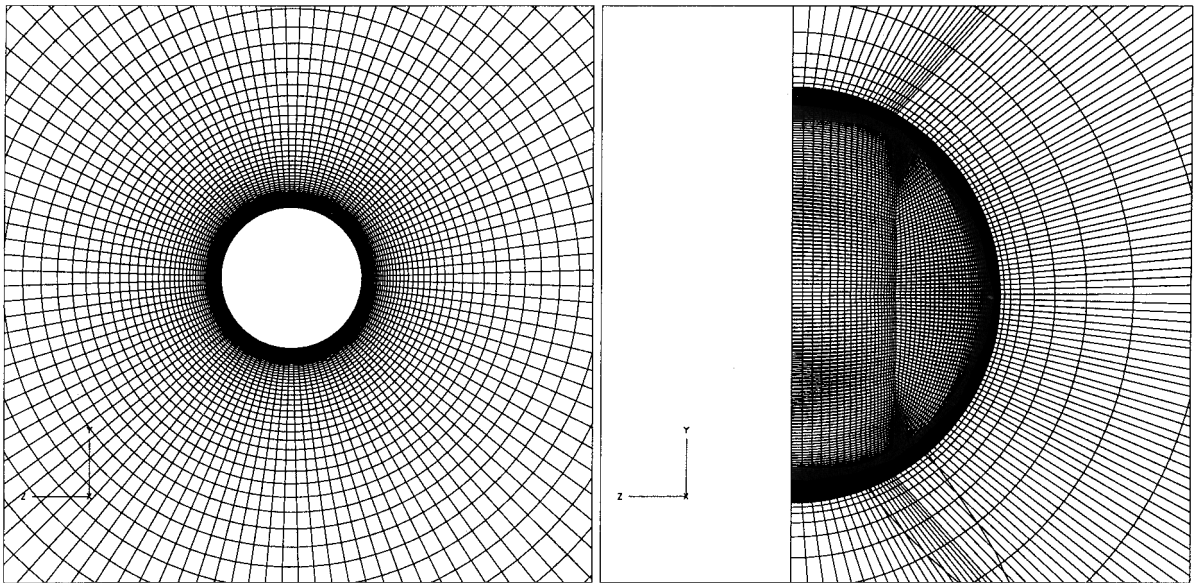

Fig. 2 Cylinder (left) and sphere mesh (right) for inviscid airflow solutions.

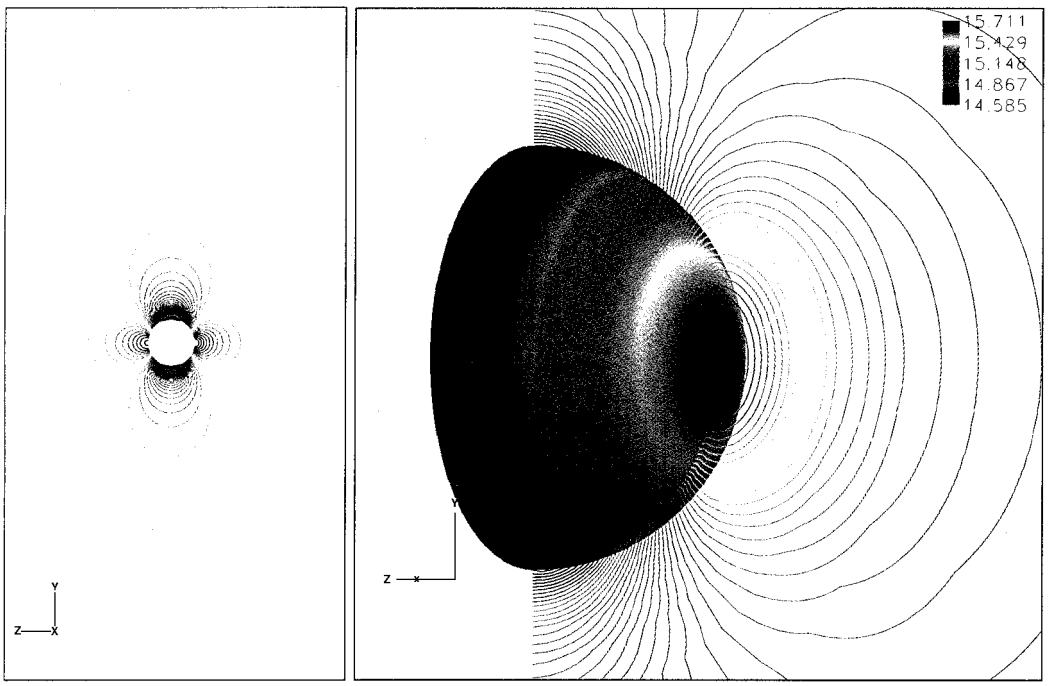

Fig. 3 Pressure distribution for inviscid flow solution around the two-dimensional cylinder (left) and the three-dimensional sphere (right).

limits, and exceeds the experimental repeatability range in some areas. One can conclude that DROP3D is capable of providing quality two-dimensional droplet impingement data and that the quality of the results improves by using an adequate droplet size distribution rather than simply the MVD of the distribution. This last conclusion is not new and is also true with Lagrangian methods on such simple geometries.[30]

B. Three-Dimensional Sphere

To validate the impingement module in three dimensions, experimental results on a three-dimensional sphere[29] test case were used. The experimental air and droplet flowfields parameters are given in Table 1 in the "Sphere" column.

An inviscid airflow solution was carried out using the mesh shown in Fig. 2, with the pressure profile from this solution shown in Fig. 3. Figure 6 shows the surface collection efficiency on the sphere, as calculated in DROP3D for the MVD $= 18.6$ μm solution and the Langmuir-D distribution with the same MVD, respectively. The details of the Langmuir-D distribution are given in Table 3. As with the two-dimensional cylinder, the solutions look qualitatively similar. Figure 7 compares the experimental data with both MVD and Langmuir-D distribution solutions. Again, the Langmuir-D distribution is slightly better than the MVD solution in matching the maximum collection efficiency and the impingement limits. Both numerical curves show small excursions outside the experimental range of repeatability, but the results are quite satisfactory, particularly for the Langmuir-D distribution. The discrepancies between numerical and experimental results can

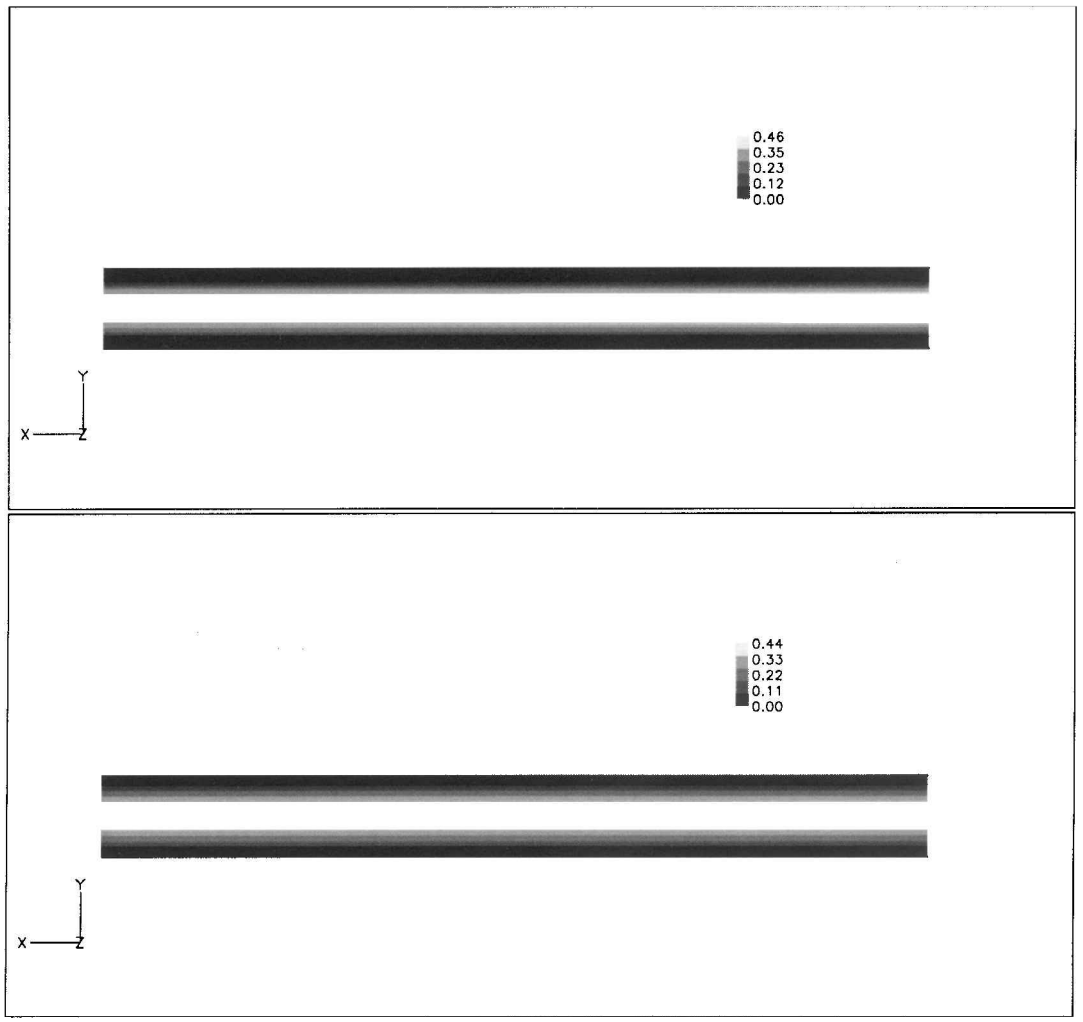

Fig. 4 Collection efficiency at the surface of a two-dimensional cylinder for one MVD = 16 μm solution (top) and a Langmuir-D distribution with the same MVD (bottom).

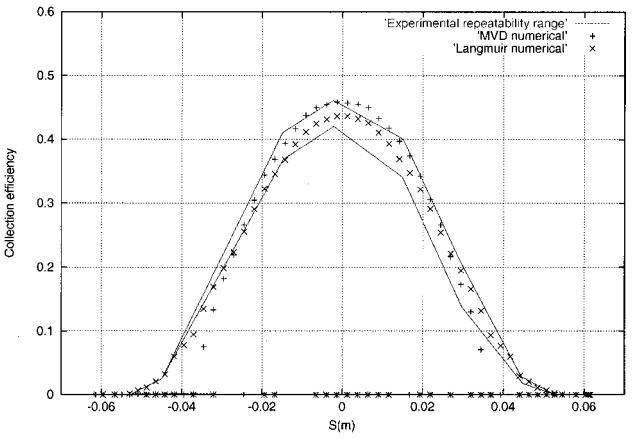

Fig. 5 Comparison of collection efficiency on the surface of the two-dimensional cylinder for the Langmuir-D distribution and the MVD solution to experimental data.

probably be reduced with a more appropriate droplet distribution such as the one measured in the icing tunnel during the experiment. Note the following comments about problems encountered with experimental data not specifically obtained or documented for CFD validation.

1) The experimental range of repeatability reported was an average over the whole range of measurements and not a local range as for the cylinder data. A more accurate estimate of the experimental range of repeatability would be desirable for better comparison to numerical results.

2) The experimental data dates back to the mid-1950s, and recent studies pointed to uncertainties in the data measurement techniques[31] resulting in over- and underestimation at different points within the same set.

3) The exact droplet distribution found in the icing tunnel, again as the experimental data dates back to the mid-1950s, is not given and may not be known, so that the Langmuir-D distribution is a best estimate of the actual distribution.

These remarks increase the level of confidence in DROP3D. Overall, one can say that the FENSAP-ICE droplet module, DROP3D, is capable of providing quality results in three dimensions, at least for similar geometries, given an adequate droplet distribution and flow experimental conditions.

Table 3 Langmuir-D distribution of droplet diameters with an MVD of 18.6 μm as used for the three-dimensional sphere test case

Percentage LWC	Ratio of diameters	Droplet diameter, μm
5	0.31	5.8
10	0.52	9.7
20	0.71	13.2
30	1.00	18.6
20	1.37	25.5
10	1.74	32.4
5	2.22	41.3

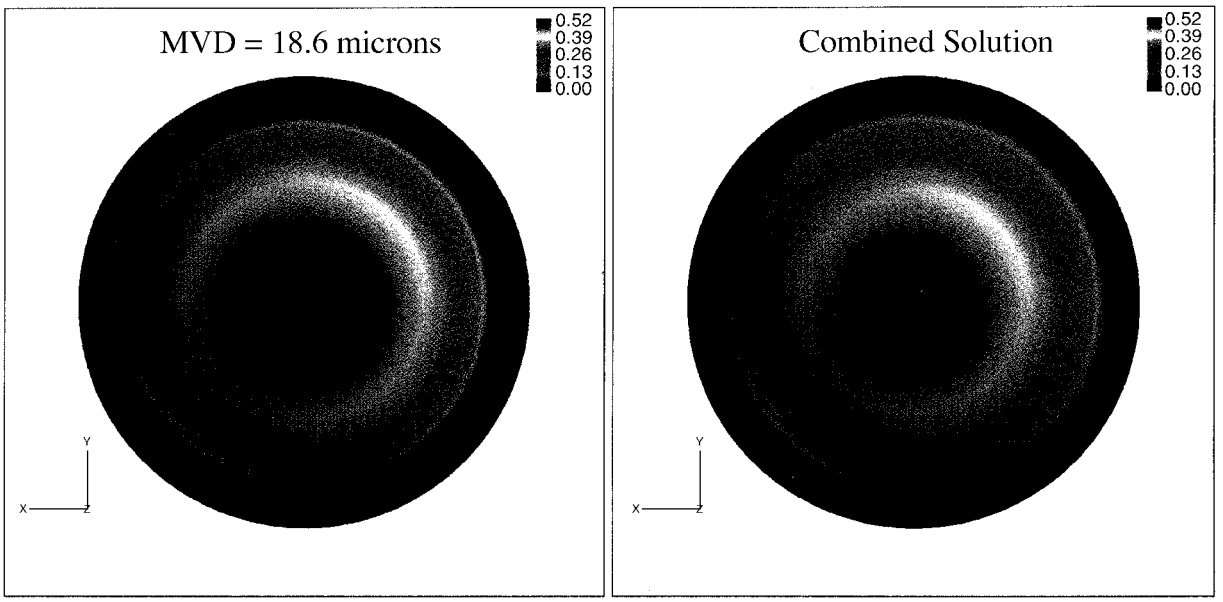

Fig. 6 Collection efficiency at the surface of a three-dimensional sphere for one MVD = 18.6 μm solution (left) and a Langmuir-D distribution with the same MVD (right).

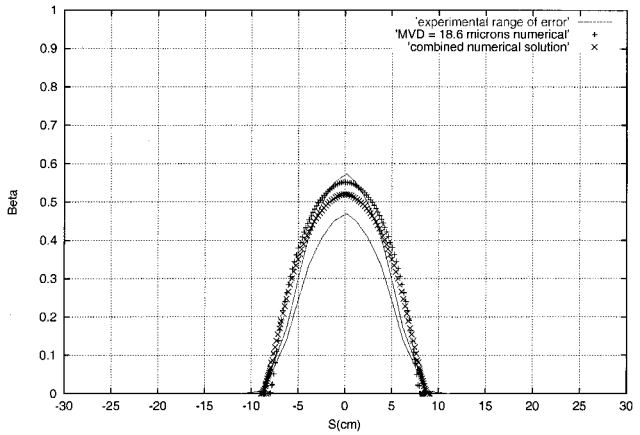

Fig. 7 Comparison of collection efficiency on the surface of the three-dimensional sphere for the Langmuir-D distribution and the MVD solution to experimental data.

V. Conclusions

The FENSAP-ICE three-dimensional droplet impingement module, DROP3D, has been shown to be a suitable tool to study water impingement over arbitrary aerodynamic surfaces. The code results are shown to be consistent with experimental data both in two and three dimensions. The advantage of DROP3D over current Lagrangian codes is in its Eulerian formulation, where the impingement phenomenon is represented in a natural and appropriate way by a set of partial differential equations. More physics can be added to the formulation as needed, and no particle tracking has to be done, avoiding particle localization problems and uncertainties arising in three-dimensional situations.

It has been shown that MVD solutions give less accurate approximations of the droplet impingement coefficient and that better results can consistently be obtained with diameter distributions that approach the actual one.

Acknowledgments

The authors would like to thank Atmospheric Environment Services, the National Sciences and Engineering Research Council and Pratt & Whitney Canada for funding under which this work was supported.

References

[1]"FAA Inflight Aircraft Icing Plan," Federal Aviation Administration, U.S. Dept. of Transportation, Washington, DC, April 1997.

[2]"Effect of Icing, De-Icing, and Heavy Rain on Aircraft Performance," AGARD Fluid Dynamics Panel, FDP Working Group 20, Advisory Group for Aerospace Research and Development, NATO, 1997.

[3]Sindir, M., and Lynch, E., "Overview of the State-of-the-Practice of Computational Fluid Dynamics in Advanced Propulsion System Design," AIAA Paper 97-2124, June 1997.

[4]Hedde, T., "Modélisation tridimensionnelle des dépôts de givre sur les voilures d'aéronefs," Ph.D. Thesis, Dept. des Sciences de l'Atmosphère, Univ. Blaise Pascal, 1992.

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

[6]Potapczuk, M., Al-Khalil, K., and Velazquez, M., "Ice Accretion and Performance Degradation Calculations with LEWICE/NS," AIAA Paper 93-0173, Jan. 1993.

[7]Wright, W., and Potapczuk, M., "Computational Simulation of Large Droplet Icing," *FAA International Conference on Aircraft Inflight Icing*, Vol. 2, Federal Aviation Administration, U.S. Dept. of Transportation, Washington, DC, May 1996, pp. 545–555.

[8]Wright, W., and Potapczuk, M., "Comparison of LEWICE 1.6 and LEWICE/NS with IRT Experimental Data from Modern Airfoil Tests," AIAA Paper 97-0175, Jan. 1997.

[9]Dompierre, J., Cronin, D. J., Bourgault, Y., Baruzzi, G., and Habashi, W., "Numerical Simulation of Performance Degradation of Ice Contaminated Airfoils," AIAA Paper 97-2235, June 1997.

[10]Kwon, O., and Sankar, L., "Numerical Investigation of Performance Degradation of Wings and Rotors due to Icing," AIAA Paper 92-0412, Jan. 1992.

[11]Lee, S., Dunn, T., Gurbacki, H., Bragg, M., and Loth, E., "An Experimental and Computational Investigation of Spanwise-Step-Ice Shapes on Airfoil Aerodynamics," AIAA Paper 98-0490, Jan. 1998.

[12]Sankar, L., Phaengsook, N., and Bangalore, A., "Effects of Icing on the Aerodynamic Performance of High Lift Airfoils," AIAA Paper 93-0026, Jan. 1993.

[13]Bidwell, C., Pinella, D., and Garrison, P., "Ice Accretion Calculations for a Commercial Transport Using the LEWICE3D, ICEGRID3D and CMARC Programs," AIAA Paper 99-0250, Jan. 1999.

[14]Potapczuk, M., "A Review of NASA Lewis' Development Plans for Computational Simulation of Aircraft Icing," AIAA Paper 99-0243, Jan. 1999.

[15]Korkan, K., and Britton, R., "On Ice Shape Prediction Methodologies and Comparison with Experimental Data," AIAA Paper 89-0732, Jan. 1989.

[16]Addy, H., Potapczuk, M., and Sheldon, D., "Modern Airfoil Ice Accretion," AIAA Paper 97-0174, Jan. 1997.

[17]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, 1997.

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

[19]Habashi, W., Bourgault, Y., Boutanios, Z., Baruzzi, G., Wagner, G., and Croce, G., "Putting Computers on Ice: A CFD Integrated Approach to the In-Flight Icing Problem," *Fourth ECCOMAS Conference on Computational Fluid Dynamics*, Wiley, New York, 1998, pp. 512–517.

[20]Bourgault, Y., Habashi, W. G., Dompierre, J., and Baruzzi, G. S., "A Finite Element Method Study of Eulerian Droplets Impingement Models," *International Journal for Numerical Methods in Fluids*, Vol. 29, No. 4, 1999, pp. 429–449.

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

[22]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.

[23]Kozel, V., Habashi, W., Tam, A., Robichaud, M., Bogstad, M., Wulf, A., and Hohmeyer, M., "Mesh Optimization: Tight Coupling of Mesh Generation and Solver, with CAD Integrity," *Fourth ECCOMAS Conference on Computational Fluid Dynamics*, Wiley, New York, 1998, pp. 114–118.

[24]Hughes, T., and Mallet, M., "A New Finite Element Formulation for Computational Fluid Dynamics: III The Generalized Streamline Operator for Multidimensional Advective–Diffusive Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, 1986, pp. 305–328.

[25]Hughes, T., and Mallet, M., "A New Finite Element Formulation for Computational Fluid Dynamics: IV A Discontinuity-Capturing Operator for Multidimensional Advective–Diffusive Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, 1986, pp. 329–339.

[26]Hughes, T., Mallet, M., and Mizukami, A., "A New Finite Element Formulation for Computational Fluid Dynamics: I Beyond SUPG," *Computer Methods in Applied Mechanics and Engineering*, Vol. 54, 1986, pp. 341–355.

[27]Boutanios, Z., "A 3-D Eulerian Investigation of the Water Droplets Collection Efficiency on the Nose and Cockpit Windows of a Convair-580 Aircraft in Flight Using FENSAP-ICE," M.Sc. Thesis, Concordia Univ., Montreal, QC, Canada, March 1999.

[28]Brown, P., and Saad, Y., "Hybrid Krylov Methods for Nonlinear Systems of Equations," TR UCRL-97645, Lawrence Livermore National Lab., Nov. 1987.

[29]Bidwell, C., and Mohler, S., "Collection Efficiency and Ice Accretion Calculations for a Sphere, a Swept MS(1)-317 Wing, a Swept NACA-0012 Wing Tip, an Axisymmetric Inlet, and a Boeing 737-300 Inlet," AIAA Paper 95-0755, Jan. 1995.

[30]Bidwell, C., "LEWICE-Based Codes," *NASA–Industry Workshop on Aircraft Icing*, NASA Lewis Research Center, Cleveland, OH, July 1993.

[31]Reichhold, J., Bragg, M., and Sweet, D., "Experimental Determination of the Droplet Impingement Characteristics of a Propeller," AIAA Paper 97-0179, Jan. 1997.