Journal of Computational and Applied Mathematics 429 (2023) 115230

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Lagrangian and Eulerian algorithms for water droplets in in-flight ice accretion

Tommaso Bellosta [*], Giacomo Baldan, Giuseppe Sirianni, Alberto Guardone

Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, Milano, 20156, Italy

ARTICLE INFO

Article history:
Received 14 September 2022
Received in revised form 28 February 2023

Keywords:
Particle tracking
Ice accretion
Collection efficiency

ABSTRACT

A particle tracking framework for the computation of the collection efficiency for in-flight ice accretion is presented. Algorithms for the computation of the collection efficiency are presented for both Lagrangian and Eulerian descriptions of the droplets equations. Strengths and weaknesses of each method are analyzed and possible solutions are described and implemented. For the Lagrangian solver, a method for automatic resolution adaptation is introduced, and a strategy for exploiting parallelism is described. In the Eulerian frame a scheme is presented for the solution of the equations on unstructured 3D grids using a relaxation approach. An hybrid Lagrangian Eulerian approach for the computation of the collection efficiency is also introduced which is relevant for Super-cooled Large Droplet condition. All presented algorithms are thoroughly verified, and a comprehensive validation is performed.

1. Introduction

In flight ice accretion is a very relevant topic for both academia and industry. In the push for even more safe air transport systems, the hazard posed by icing must be reduced to a minimum. Certification authorities require aircraft manufacturers to prove their products can withstand operations in icing environments and require some kind of ice protection and/or detection system on all air-worthy vehicles. This led to a very active research path for both the physics behind ice accretion and its numerical simulation. Certification requirements are sometimes met by in-flight testing or experiments in icing-wind tunnels, but are often proved via numerical simulation. Moreover, manufacturers are now considering ice accretion early in their design cycle, making icing codes part of their design loop.

An aircraft flying through a cloud of super-cooled water droplets may be subject to mild or severe icing depending on the atmospheric and flying conditions such as temperature, velocity, droplets diameter and water content in the cloud (LWC). These droplets may freeze upon impact giving origin to rime ice, or adhere to the surface forming a liquid film which can later freeze forming glaze ice. Most of the current ice prediction models originated from the early work of Messinger [1] and later of Myers [2]. They proposed a method for predicting the layer of ice accreting on a surface by solving a one dimensional Stefan problem [3] on control elements summing up to the entire surface. This cells are coupled to each other via boundary conditions accounting for mass exchange. Examples of current numerical prediction tools employing such models are LEWICE [4], ONICE [5], MULTI-ICE [6], PoliMIce [7], FENSAP-ICE [8] and ONICE3D [9].

The standard approach to simulating ice accretion uses a quasi-steady approach exploiting the different time-scales at play. First the flow of air and water droplets is computed in order to obtain the amount of water mass captured by

[*] Corresponding author.
E-mail address: tommaso.bellosta@polimi.it (T. Bellosta).

https://doi.org/10.1016/j.cam.2023.115230
0377-0427/© 2023 Elsevier B.V. All rights reserved.

Nomenclature

Greek letters

α	Droplets volume fraction [-]
β	Collection efficiency [-]
μ	Viscosity [Pa s]
ρ	Density [kg/m^3]

Subscripts

d	Droplet
f	Fluid

Physical quantities

$\mathbf{v}$	Local velocity vector [m/s]
C_D	Drag coefficient [-]
d	Droplet diameter [m]
g	Gravitational acceleration [m/s^2]
LWC	Liquid water content [g/m^3]
m	Mass [kg]
MVD	Mean volume diameter [m]
s	Curvilinear abscissa [m]
t	Time [s]
T_∞	Freestream temperature [K]
V_∞	Freestream velocity [m/s]
A	Elemental cell surface area [m^2]
Re	Reynolds number [-]
We	Weber number [-]

the aircraft surface. Then, ice growth is computed and a new iced geometry is obtained. The process is repeated until the total exposure time is reached.

As water droplets are responsible for the mass of water fueling ice formation, their deposition on the exposed surfaces must be accurately predicted. Water deposition depends on the droplets trajectory, interaction with the flow field and their behavior at the impingement location. This is especially true for super-cooled large droplets (SLD), which are water droplets with a diameter larger than 50 μm. Due to their larger size, SLD can undergo deformation and breakup as their relative velocity with respect to the surrounding air can be higher than that of smaller droplets. Also, the primary outcome of impact with a solid surface is that of disintegrating upon impingement and depositing only a fraction of their mass on the surface. Therefore, water droplets must be carefully simulated in order to account for all phenomena that may effect the final ice prediction.

This work's focus is on the computation of the dispersed flow of air and water droplets. Different regimes exist identifying the type of coupling between particles and the carrier fluid [10]. The usual concentration of water droplets in clouds [11,12], is such that only a one way coupling is active. In short, particle concentration is low enough not to modify the carrier flow behavior. This is reflected in the way such dispersed flows are simulated: first the carrier fluid solution is obtained independently, and then droplets trajectories are computed. The dispersed phase field can be obtained by means of a Lagrangian or Eulerian simulation. If the Lagrangian formulation is employed, the overall solution method for the particle laden flow is called Euler–Lagrange. The Lagrangian formulation for the dispersed phase is the standard approach used in the ice-accretion field since it allows to easily treat first order effects such as droplet-wall interaction, as well as second order effects like particle deformation and aerodynamic breakup. The main drawback of the Lagrangian formulation is the amount of particles required to obtain meaningful results, which can be very large for 3D simulations. Examples of existing Lagrangian codes are found in [4,5]. A second modeling option is the Euler–Euler description. By casting the particles equations in the Eulerian frame, some limitations of the Lagrangian method can be avoided. These include long computation times for 3D cases, difficulty in determining impingement limits and the initial seeding of particles. Some limitations arise, first of them being the difficulty in accounting for splashing droplets, and the impossibility of dealing with trajectory crossing. That considered, many codes where published employing the Eulerian description [13–15], each implementing different strategies for solving the equations.

In aeronautical icing problems, the aim of particle tracking is to obtain the collection efficiency parameter β which expresses the fraction of water mass (contained in the free stream) collected at a given location over the aircraft surface.

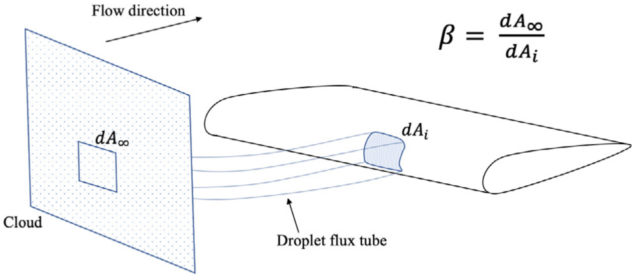

Fig. 1. Collection efficiency parameter β. Visual representation for a three-dimensional problem.

β is defined as the ratio of the cross section area of a droplet stream tube at the farfield to the area of impact on the body surface, as reported in Fig. 1.

In this work we introduce a comprehensive set of algorithms for the computation of the collection efficiency β. We focus on both Lagrangian and Eulerian methods, and propose an hybrid strategy involving both. The Lagrangian approach is presented highlighting the equations and algorithms involved. Focus is placed on the numerical solution of the problem and on issues related to particle resolution and initial seeding. A finite volume solver for the droplets equations in the Eulerian frame is also introduced. A relaxation formulation is used to obtain an approximate Riemann solver that allows an upwind discretization of the convective terms. Both solvers are thoroughly tested via an extensive validation and verification (V&V) campaign.

This paper is organized as follows. In Section 2 the Lagrangian scheme is introduced and properly verified. In Section 3 the Eulerian formulation is presented together with the numerical solution procedure. Again a proper verification test is performed via the method of manufactured solutions. Section 4 reports the validation results consisting in replicating the experimental collection efficiency measurements for both two and three dimensional cases. The work ends with the conclusions in Section 5.

2. Lagrangian description

In the Euler–Lagrange approach to solving particle laden flows, the dispersed phase is handled by means of the Discrete Parcel Method (DPM) [16]. Each particle, or parcel, represents a set of neighboring droplets and it is characterized by an average velocity $\mathbf{v}_p$ and an average diameter d_p.

2.1. Models formulation

The trajectory of each parcel is reconstructed by numerically integrating the following differential equation

$$m_{\mathrm{p}} \frac{d\mathbf{v}_{\mathrm{p}}}{dt} = \frac{\pi}{8} \mu_{\mathrm{air}} d_{\mathrm{p}} \mathrm{Re}_{\mathrm{p}} \left(\mathbf{v}_{\mathrm{air}} - \mathbf{v}_{\mathrm{p}}\right) C_D + \frac{\pi}{6} d_{\mathrm{p}}^3 \mathbf{g} \left(\rho_{\mathrm{p}} - \rho_{\mathrm{air}}\right), \tag{1}$$

where μ_{air}, ρ_{air}, $\mathbf{v}_{\mathrm{air}}$ are the air local viscosity, density and velocity, C_D is the parcel drag coefficient and $\mathbf{g}$ is the gravitational acceleration. Re_{p} is the particle's Reynolds number computed using the relative droplet-flow velocity

$$\mathrm{Re}_{\mathrm{p}} = \frac{\rho_{\mathrm{air}} \left|\mathbf{v}_{\mathrm{air}} - \mathbf{v}_{\mathrm{p}}\right| d_{\mathrm{p}}}{\mu_{\mathrm{air}}}. \tag{2}$$

Note that water is far denser than air ($\rho_{\mathrm{p}} \gg \rho_{\mathrm{air}}$) [17] and that the only forces considered in the equation of motion (1) are gravitation and aerodynamic drag.

Aerodynamic drag is modeled from empirical correlations for the drag coefficient C_D. The model takes into account deformations deviating the particle shape from a perfect sphere. Such feature is fundamental to simulate the motion of droplets of large dimensions. For spherical particles, the model by Morrison [18] is linked to that reported by Clift [19] at $\mathrm{Re}_{\mathrm{p}} = 10^6$ to best fit experimental data. For larger droplets, their deformation is modeled as a linear transition between a spherical shape and a oblate disk. The parameter governing the deformation is the eccentricity, which is a function of the Weber number $f = (1 + 0.07\sqrt{\mathrm{We}})^{-6}$, see [20]. The Weber number We expresses the magnitude of the aerodynamic forces relative to the surface tension of a droplet σ_p:

$$\mathrm{We} = \frac{\rho_{\mathrm{p}} d_{\mathrm{p}} \left|\mathbf{v}_{\mathrm{p}} - \mathbf{v}_{\mathrm{air}}\right|^2}{\sigma_{\mathrm{p}}}, \tag{3}$$

The drag coefficient of the deformed droplet can be obtained as a weighted average of that of a rigid sphere and that of a disk:

$$C_D = \begin{cases} (1-f)C_{D_{\text{Sphere}}} + fC_{D_{\text{Disk}}} & \text{if We} \leq 12 \\ C_{D_{\text{Disk}}} & \text{if We} > 12 \end{cases} \tag{4}$$

where the disk drag coefficient is reported in [19].

In aeronautical icing applications, the impingement of a droplet against a solid surface can result in three different outcomes [21], namely stick contact, rebound and splash. For stick contact, the particle adheres to the wall upon impingement. This happens with some degree of deformation for the impacting droplet, so that it is possible for the particle to spread over the surface in the vicinity of the impact location. Rebound and splash mechanisms instead lead to the re-emission of some or all of the droplet mass in the air flow. For rebounding droplets, at modest impact velocities, a thin film of air can get trapped between the droplet and the wall causing it to bounce right off the surface. Splashing droplets on the other hand, disintegrate upon impact and a liquid sheet is ejected, leading to the formation of a secondary group of droplets. The total ejected mass is a fraction of the initial droplet mass and increases as the impact velocity increases.

The modeling of the droplet-wall interaction upon impact is of paramount importance as it allows to accurately estimate the amount of water collected over the surface, to ultimately evaluate the thickness of the resulting ice layer. The occurrence of a particular mechanisms depends on many parameters such as the characteristics of the droplet (size, density, viscosity, surface tension), its impact velocity and impact angle, and the properties of the surface.

To describe the problem different dimensionless numbers are used, such as the Weber number and the Ohnesorge number defined as:

$$Oh = \frac{\mu_p}{\sqrt{\sigma \rho_p d_p}} \tag{5}$$

expressing the ratio of viscous forces to the product of inertial forces and surface tension. The strategy adopted is that of completely switching off the rebound model and considering bouncing as a special case of splashing, when only one secondary droplet is formed and no water is left on the wall. This choice is consistent with the scientific literature concerning the wall interaction problem, as many authors [22] identify only two results of the impingement: deposition and splashing. The model adopted is that employed in LEWICE [23]. In the following a brief overview of the model is reported. The parameter governing the outcome of the model is the dimensionless splashing number

$$K_L = \frac{0.859\sqrt{K}\left(\frac{\rho_p}{LWC}\right)^{0.125}}{\sin \xi^{1.25}} \tag{6}$$

where ξ is the droplet incidence angle, defined between the surface tangential direction and the particle impact velocity. $K = Oh\, Re_p^{1.25}$ is the Mundo parameter. A droplet splashes if $K_L \geq 200$. This formulation considers two splashing regimes to comply with experimental observation. The Mundo parameter K is proportional to the normal impact energy, linking the normal impact velocity with splashing phenomena. A second splashing regime is observed at low incidence angles ξ, defined as the complimentary of the angle between the droplet velocity and the surface normal. At low incidence, the normal impact velocity is not high enough to activate the condition on the Mundo parameter; nonetheless splashing can still be observed in this case. What happens is that the droplet does not spread much on the surface and a part of the particle is not attached to the wall during the impact. Hence, the tangential kinetic energy is not efficiently dissipated by the viscous forces and the liquid droplet may partially or completely bounce off the wall. The denominator of K_L considers this second regime by increasing the splashing parameter with a decrease of the incidence angle.

If the above conditions is satisfied, the fraction of mass that is reintroduced in the flow field, expressed through the mass loss coefficient Φ, is computed as:

$$\Phi = \frac{m_s}{m_p} = 0.7(1 - \sin \xi)\left[1 - e^{-0.0092026(K_L - 200)}\right] \tag{7}$$

The size of the splashing particles is computed as:

$$d_s = 8.72\; d_p\; e^{-0.0281K} \tag{8}$$

The splashing droplets velocities are computed as:

$$\begin{cases} u_{n_s} = u_{n_0}\,(0.3 - 0.002\;\xi) \\ u_{t_s} = u_{t_0}\,(1.075 - 0.0025\;\xi) \end{cases} \tag{9}$$

where the s refers to the splashing particle, 0 the impinging one; n and t are respectively the normal and tangent component of the velocity vector.

As stated in the introduction, the final objective of particle tracking is that of computing the collection efficiency. For a uniformly distributed cloud, it can be computed as the ratio of the local surface collection density i.e., the mass of water collected at a given location per unit surface (or per unit length), w.r.t. the nominal cloud front density.

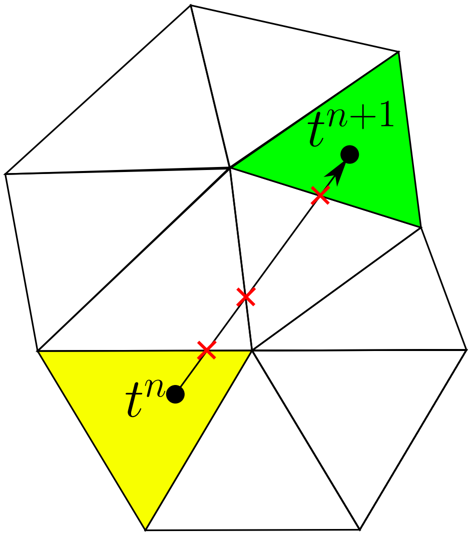

Fig. 2. Algorithm used in this work to get the owner cell during time integration. Intersections are computed between the particle trajectories and the faces of the current owner cell (here in yellow). If a valid intersection is found, the neighboring cell is queried until the new owner is found (green triangle).

2.2. Algorithms

In order to integrate the particles trajectories one has to first compute the carrier fluid solution (namely the velocity, density and viscosity) at the particle current position. Due to the different description employed for the continuous (Eulerian) and the disperse (Lagrangian) phases, the corresponding solution variables are not known at the same locations, but need to be interpolated accordingly. This interpolation step is the main building block of every (Lagrangian) particle tracking code. Since the flowfield solution is defined over the computational mesh, first the particle position needs to be localized in the grid. This localization step has the aim of finding the mesh cell that contains the particle, in order to properly perform the interpolation. Different strategies exist in literature, namely methods that make use of Cartesian support grids, tree based search and known-vicinity algorithms. Due to the evolution nature of the particle tracking problem, where particles are integrated from one position to the next many times, known-vicinity algorithms are the most widely employed for unstructured meshes. In short, information about the previous location of the particle is used to find the cell containing the new position. In Lagrangian particle tracking such algorithms are extremely efficient due to the on average short distance traveled by particles at a given time step. Our implementation of the known vicinity algorithm computes the intersections between the ray starting from the current particle position in the direction of the new position and the faces of the current cell being tested (Fig. 2). It starts with the cell containing the start position, if the intersection distance is greater than the path the particle needs to travel, then the new position is in the cell containing the previous one. Otherwise the mesh connectivity is used to fetch the cell on the other side of the intersected face, and the procedure is repeated until the new cell owner is found. This algorithm allows to identify intersections with the boundaries unambiguously, which is of paramount importance for computing the collection efficiency. The algorithm needs careful thought in the way it is implemented in order to avoid some possible problems. First, round-off errors in computations can lead to failure of geometric predicates like the ones used to compute the ray-face intersections. In our case it means losing particles during the localization step. In the present work, this is avoided by computing the intersections using arbitrary precision algebra. Filtered predicates are used to perform computations adaptively in order to adjust the precision of number types when needed [24]. This allows to keep the computational overhead manageable. The second problem is the use of hybrid grids. These may contain elements with quadrilateral faces, which are often non planar and need to be properly handled. One possible solution is that of computing the actual ray-bilinear patch intersection, the other is to split each face in two triangles which is less computationally expensive. The latter approach is adopted here. In order to split quadrilateral faces consistently on both sides, face splitting is done by selecting the diagonal starting from the vertex with the lowest global id. This avoids the problem of gaps forming in between cells sharing a quadrilateral face not split consistently.

Once the owner cell is known, the fluid solution can be interpolated to the particle position. In the node-centered finite volume approach considered here, the solution is defined at grid vertices and different strategies are available for interpolating nodal values to the interior of an element. One solution would be using the cell nodes to perform an inverse distance weighted interpolation, which works well for hybrid grids since it does not depend on the element type. The other option is to use linear (for simplex elements) or non-linear (for non-simplex elements) interpolation. The former method is simpler but the reconstructed field is not continuous across cell faces, which we found being extremely important for the convergence of the collection efficiency with the number of particles. The latter method allows to reconstruct a continuous field but can be expensive for non simplex elements since it requires solving a non-linear system of equations to invert the interpolation functions. In this work, the strategy adopted is to perform the interpolation by subdividing each element in triangles or tetrahedra and to use barycentric coordinates to perform a linear interpolation. This allows to retain the mentioned benefit of shape function interpolation avoiding the cost for non simplex cells. The subdivision is done online only for the interpolation step and done so to always have a conformal grid. The subdivision of elements is done only based on the ids of the vertices and it is consistent with the quadrilateral faces subdivision used in the tracking algorithm [25].

2.2.1. Parallelism

One way coupled particle tracking is, ideally, an embarrassingly parallel task. To take advantage of the available computational power of modern machines, an hybrid MPI-OpenMP approach is employed. At the higher level, standard message passing interface is used to achieve parallelism via domain decomposition. This allows to run simulations on large grids, where the data structure for the mesh and particles cannot fit in the memory of a single machine. Balancing the computational effort of Euler–Lagrange formulations is no easy task. In standard Eulerian descriptions, the computational cost is proportional to the number of vertices/elements of the mesh, while communication costs are proportional to cut edges. Therefore standard mesh partitioning tools subdivide the domain by ensuring a similar number of nodes in each partition while trying to minimize the number of cut edges. In particle tracking applications instead, the computational cost is proportional to the number of particles currently associated to a given domain partition while the number of mesh nodes only adds to the memory footprint. In the target application of this work (i.e. the computation of the collection efficiency for simulating in-flight ice accretion), particles occupy only a small subset of the whole domain, making balancing an even harder task. In this work the problem is tackled in a two step procedure. First the mesh is partitioned using the standard (i.e. Eulerian metric) approach, and a preliminary particle tracking simulation is performed with a relatively small number of droplets. As these particles are tracked in the domain, their owner cells are tagged and a positive weight is assigned to them proportional to the number of particles that pass through them during the computation. Cell weights are then mapped to the mesh nodes, and the mesh is re-partitioned according to the updated weights. Additional information and scaling properties of this mesh re partition strategy can be found in Ref. [26]. Inside one sub-domain, each particle can be tracked concurrently since their trajectory does not depend on other droplets trajectories. Leveraging the multicore architecture of modern CPUs, each MPI process can spawn as many threads as there are cores available. Each thread takes care of a dynamically selected batch of particles.

2.2.2. Collection efficiency computation and particle resolution

For the computation of the collection efficiency, clouds are represented as a single front initially injected at an arbitrary distance in front of the aircraft, see Fig. 1. This distance is set so that droplets are traced starting from an unperturbed region of the domain and so that the computational burden related to the trajectory time integration, proportional to the integration length, is reasonable. In three dimensional problems, this front consists of a two-dimensional layer of droplets uniformly distributed over a plane oriented perpendicularly to the free stream direction. In a two-dimensional setting the layer degenerates into a straight line normal to the mean flow direction. In practical applications, clouds are poly-dispersed. In ice accretion applications, usually particles are treated as mono-dispersed, with a diameter set equal to the cloud Median Volume Diameter (MVD) i.e., the diameter corresponding to the median value of the distribution characterizing the size of droplets contained in the cloud.

By integrating the equation of motion numerically it is possible to reconstructs the trajectory of each droplet from its initial position, at the cloud front, to its impinging point over the aircraft surface. As mentioned, the collection efficiency at each surface element i can then be obtained as

$$\beta_i = \frac{\sum m_j}{A_i \, \mathrm{LWC}}$$

where LWC is the computational cloud liquid water content obtained as the ratio between the total cloud mass and the area of the injection front. The value of β strongly depends on the number of particles. Moreover, we are not interested in particles not impinging the aircraft surface, since we would rather simulate particles that contribute to the computation of β. Therefore, the seeding and resolution of the injection front is a delicate task. In order to alleviate this task an automatic cloud front adaptation was developed. A uniform grid is constructed having its vertices at the particles injection position. The particle grid is made of line elements for 2D cases, and quadrilateral elements for 3D computations. Successive simulations are performed, refining each element if at least one of its particles hit the target surface. The computation stops automatically when the difference in norm L2 of two successive β calculations falls below a user supplied threshold. Since the formula used for computing β requires a uniform mass distribution on the injection front, each parcel's population size (i.e. the number of actual droplets that are represented by the parcel) need to be adjusted to guarantee a uniform LWC distribution. This is achieved by computing the dual area of each vertex in the particle mesh (in analogy with the area of dual cells used in median-dual node centered finite volume schemes) and obtaining the parcel population size by dividing its area with the particle mass (the result is multiplied by a target LWC in order to obtain a pure number). The mass to sum in the expression for beta becomes the population size n_p times the mass of the single particle m_p.

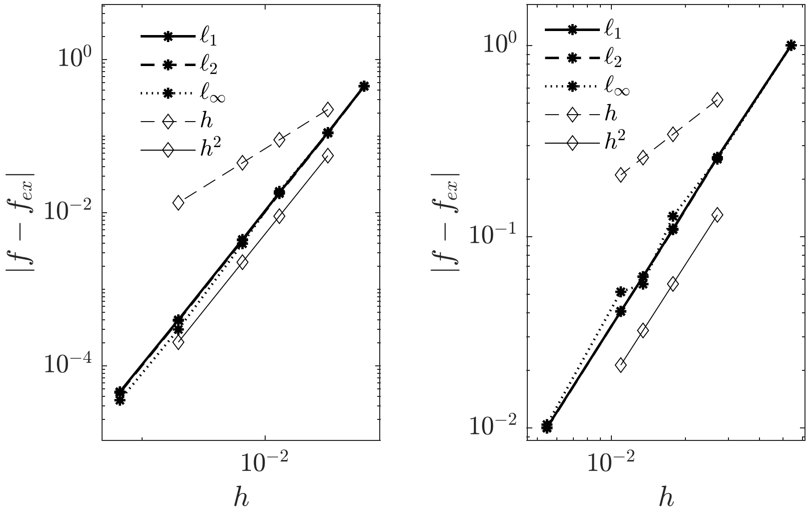

Fig. 3. Convergence of the interpolation against the grid size. Left: unstructured 2D grid. Right: unstructured 3D grid.

More accurate representation of the particle size distribution can be taken in to account by subdividing the droplets diameter PDF in bins. For each bin the MVD can be computed and a mono-dispersed particle tracking simulation can be run. The total collection efficiency can then be computed as the weighted sum of the β of each bin. This is the standard way of simulating poli-dispersivity for the computation of the collection efficiency in in-flight ice accretion.

2.3. Verification

Being at the heart of the particle tracking code, the localization algorithm and the interpolation need to be extensively verified before proceeding with a proper code validation in order to discover possible implementation mistakes. Two verification tests are performed. The first deals with the localization and interpolation routines. One thousand particles at random positions are initialized inside an unstructured mesh and an analytical function known at the nodes of the grid is interpolated to the particle locations. The chosen function is

$$f(x, y, z) = \cos(xy)e^{\frac{z}{10}} + 0.3(x^2 + y^2 + z^2) \tag{10}$$

The test was performed on both a two and three dimensional mesh and the results are reported in Fig. 3. The error of the interpolation is plotted against the spacing of the mesh, showing the theoretical order of convergence for the ℓ_1, ℓ_2 and ℓ_∞ norms.

A second test was performed in order to verify the time integration algorithm. A particle in the Stokesian regime is injected with zero initial velocity in a one-dimensional accelerating flowfield such that $u_f = x$ for 3.5 s. The simple form of the flowfield and of Stokes drag, in addition to neglecting the gravity term, allows to solve the equation of motion of a particle analytically. It is then possible to compare the computed final position with the exact one and plot the convergence curves in Fig. 4. Here the convergence of the time integration implicitly depends also on the integration and localization routines employed. Achievement of the expected order of convergence is treated as a proof of verification of the full particle tracking algorithm.

3. Eulerian description

If the number of particles in each computational cell is high enough [17], they can be described as a continuum and their equations of motion can be cast in the Eulerian framework. This allows to use many of the standard tools developed for CFD, and avoids having to deal with the carrier and dispersed phases being known at different points.

3.1. Model equations

As for the Lagrangian description, the hypothesis of one-way coupled flow is made and therefore the carrier fluid equations remain the same and can be solved as usual. In conservative form the particles equations can be written as a function of the volume fraction α and the particle's momentum $\alpha\mathbf{v}_p$

$$\begin{cases} \frac{\partial}{\partial t}(\alpha) & +\nabla\cdot\left(\alpha\mathbf{v}_p\right) = 0 \\ \frac{\partial}{\partial t}\left(\alpha\mathbf{v}_p\right) & +\nabla\cdot\left(\alpha\mathbf{v}_p\otimes\mathbf{v}_p\right) = \alpha\mathbf{f}_d \end{cases} \tag{11}$$

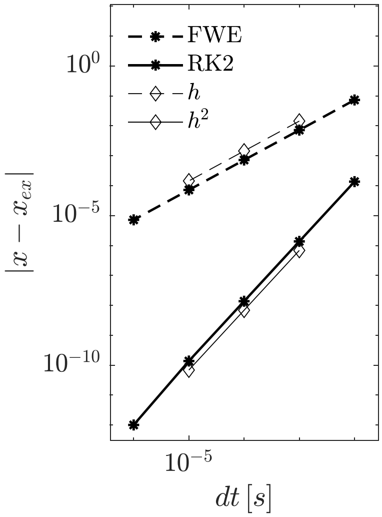

Fig. 4. Convergence of time integration of the Lagrangian implementation. 2D unstructured grid with forward Euler time integration (FWE) and a second order Runge–Kutta scheme (RK2).

The hyperbolic part of the above system is known as pressureless gasdynamics (PGD), due to its resemblance to the Euler system without the pressure term in the momentum equation. The PGD system is notoriously non strictly hyperbolic, which makes it hard to develop upwind schemes since its Jacobian matrix is non diagonalizable. This complicates stuff for solving the equations in a finite volume framework. A technique recurrent in literature is that of making the equations strictly hyperbolic by adding a pressure like term in the momentum equation and subtracting the same term as a source on the right hand side. An example of this approach is reported in Ref. [27]. A similar approach is taken in this work. A relaxed version of the PGD system [28], is solved in order to have a strictly hyperbolic system of equation. A new primitive variable π is introduced and the system is augmented as:

$$\begin{cases} \frac{\partial}{\partial t}(\alpha) & +\nabla\cdot\left(\alpha\mathbf{v}_p\right) = 0 \\ \frac{\partial}{\partial t}\left(\alpha\mathbf{v}_p\right) & +\nabla\cdot\left(\alpha\mathbf{v}_p\otimes\mathbf{v}_p + \pi\mathbf{I}\right) = \alpha\mathbf{f}_d \\ \frac{\partial}{\partial t}(\alpha\pi) & +\nabla\cdot\left(\alpha\pi\mathbf{v}_p + c^2\mathbf{v}_p\right) = -\lambda\alpha\pi \end{cases} \tag{12}$$

where λ is the relaxation rate. This relaxed system has the advantage of having all eigenvalues linearly degenerate (all waves are contact discontinuities), thus making the solution of the Riemann problem straightforward. The modified system (12) is used to derive an approximate Riemann solver for the original PGD Eqs. (11). A splitting scheme is employed. Starting from the solution at time t_n, the intermediate solution is obtained by solving system (12) for $\lambda = 0$. The solution at time t_{n+1} is obtained in a second step by solving the system of ODE

$$\begin{cases} \frac{\partial}{\partial t}(\alpha) = 0 \\ \frac{\partial}{\partial t}\left(\alpha\mathbf{v}_p\right) = 0 \\ \frac{\partial}{\partial t}(\alpha\pi) = -\lambda\alpha\pi \end{cases} \tag{13}$$

for the relaxation rate going to infinity. This results in keeping the volume fraction and momentum unchanged and in setting $\pi = 0$, which if substituted back into the relaxed system yields the original PGD equations. It can be seen that the zero pressure space in the limit of infinite relaxation rate, is stable in the sense that for initial data with zero pressure, the solution naturally evolves with zero pressure [29]. The original work in Ref. [28] proves that in 1D a Godunov method for the presented scheme is total variation diminishing and strictly positive for some mild conditions on the relaxation parameter c. In this work, the presented scheme is implemented in a FV framework. The Eulerian particle tracking equations are discretized using the vertex-centered finite volume method with a standard edge-based data structure on a median-dual grid. The convective fluxes needed in the first step of the splitting are discretized using a limited second order MUSCL reconstruction with the Venkatakrishnan slope limiter. The Riemann problem is solved at each edge to compute the fluxes. Source terms are approximated at each node using a piecewise-constant reconstruction within each control volume. Gradients are obtained via a weighted least-squares method. A time-marching approach is used to drive the system to steady state using an implicit Euler scheme. Local time stepping and automatic CFL adaptation coupled with under-relaxation is employed to accelerate convergence. Boundary conditions need to be imposed on the farfield and on the solid surfaces. In this work, boundary conditions are enforced in a weak sense by computing the proper boundary flux. At the farfield, a standard characteristic based condition is enforced. At solid surfaces, the boundary condition must

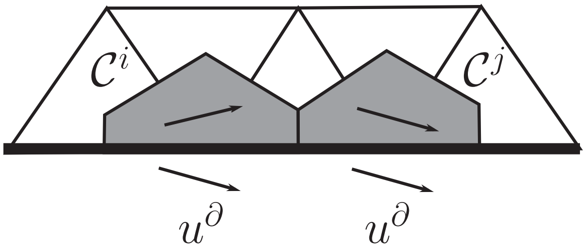

Fig. 5. Boundary conditions are imposed at the wall in a weak sense. In control volume $\mathcal{C}^i$ symmetry is imposed by reflecting the velocity vector. On $\mathcal{C}^j$ the boundary flux is computed using the solution at the boundary node.

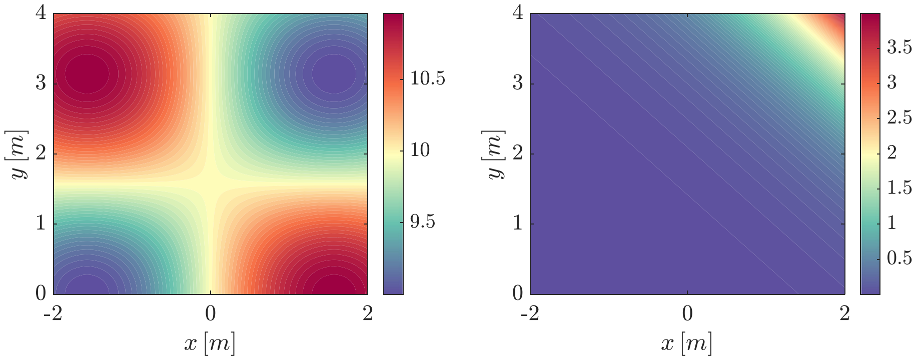

Fig. 6. Analytical solution used for the verification studies of the Eulerian particle tracking solver. Left: particle velocity in x direction. Right: droplet volume fraction.

allow for droplet impingement and should not inject any mass in the domain. The imposed condition changes according to the direction of the droplets velocity with respect to the surface normal, see Fig. 5. If the velocity is pointing towards the surface, then an outflow condition in imposed. Instead if the velocity is pointing towards the computational domain, symmetry is enforced to obtain a slip condition.

The collection efficiency can be computed as

$$\beta = \frac{\alpha \mathbf{v}_p \cdot \mathbf{n}}{\mathrm{LWC}_\infty V_\infty} \tag{14}$$

Poly-dispersed clouds can be accounted for in the same way as done for the Lagrangian particle tracking, i.e. by subdividing the droplets size PDF in different bins and running a simulation for each of them.

3.2. Verification

The presented scheme was subject to a similar Verification procedure as done for the Lagrangian solver. The method of manufactured solutions is used to obtain an exact solution to system (11). An analytical expression for the conservative variable made of trigonometric and exponential terms (Fig. 6 shows the expression for the volume fraction and the first component of the velocity) is substituted in the system. Residual terms are used as sources to drive the solution to the imposed exact expression. The modified system is solved varying the grid spacing (both for structured and unstructured meshes) to obtain the convergence plots of Fig. 7 in L2 norm. The plot shows that the scheme is first order when the nodal values are used to solve the Riemann problem, whereas a linear reconstruction of the interface values, yields a second order scheme as expected.

3.3. Dealing with secondary droplets

Secondary droplets, namely those created as a result of splashing, can have a large impact on the catch efficiency as it will be shown later in the validation section. In the Eulerian formulation, treating those droplets is not as straightforward as it is for the Lagrangian approach. The outcome of the splashing model is the re-injection of new smaller droplets in the computational domain. The size of secondary droplets depends on the impact angle of primary droplets and therefore it is different for every point of the surface. Since the Eulerian formulation deals with a mono-dispersed spray, accounting for splashing droplets as done for the primary ones would be computationally prohibitive, and would require the specification of a boundary condition that depends on the surface location. In this work, an hybrid strategy is employed, for which

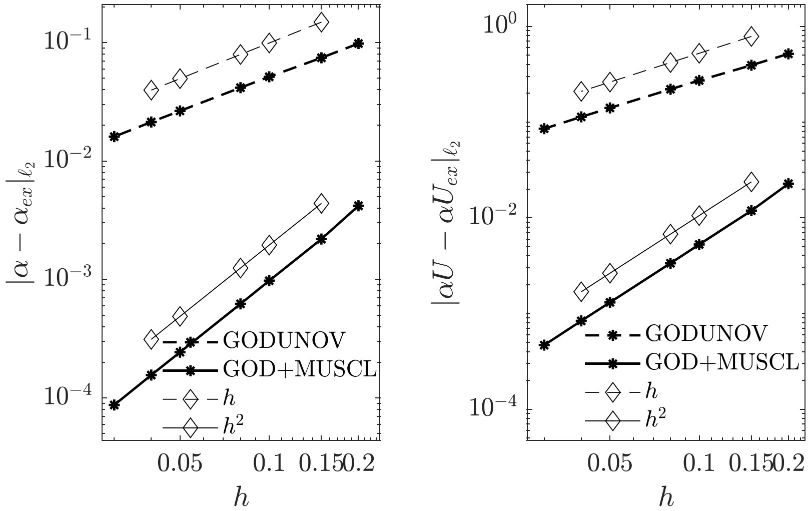

Fig. 7. Convergence of the FV scheme for the Eulerian droplet solver. Left: L2 norm of the error of the droplet volume fraction. Right: error of the particles x momentum.

secondary droplets are tracked using the Lagrangian approach. The total collection efficiency at a certain point on the surface is therefore computed as the sum of the Eulerian catch efficiency (possibly of every bin if a poly-dispersed characterization of the spray is used) plus the additional Lagrangian re-injection collection efficiency.

4. Validation results

The particle tracking solvers presented are now validated against multiple experimental results. The carrier phase solution for both the Lagrangian and Eulerian implementations is obtained by solving the Reynolds Averaged Navier–Stokes equations by means of a vertex centered second order FV scheme [30]. The convective fluxes are obtained by a standard Roe scheme, whereas viscous fluxes are computed using the corrected average of gradients approach. Effect of turbulence on the mean fields are accounted for by the SST model [31].

4.1. Two dimensional impingement on a NACA23012 model in SLD conditions

The code developed was used to replicate the collection efficiencies obtained during an experimental campaign conducted by NASA Glenn [32]. Their work aimed at providing the first extensive impingement database for SLD as well as smaller droplets. The collection efficiency was measured on a NACA23012 straight wing model installed at the NASA Glenn Ice Research Tunnel. The experiments were conducted at a total air temperature of 283 K and an air speed of 78 m/s, corresponding to a Reynolds number of approximately 5.25 million per meter. The model used had a chord of 91.44 cm and was mounted with an angle of 2.5 degrees with respect to the incoming flow. The collection efficiency was measured for MVDs of 20 μm, 52 μm, 111 μm, 154 μm and 236 μm. For both solvers, an hybrid mesh of around 100 thousands cells was used for the computation. Such grid was chosen as to get a numerical result of the carrier flow that is independent on the mesh resolution. For the Eulerian solver, the sparse system obtained from the discretization introduced in Section 3, is solved using the generalized method of residuals coupled with a standard incomplete lower upper (ILU) preconditioner. The pseudo in time problem is solved until the L2 norm of the residual of the volume fraction equation drops below the value of 10^{-10}. Computation using the Lagrangian solver was done until the difference in norm L2 of two successive β iterations reaches the value of 10^{-6}, which accounts for a total of around 200 thousands particles. Fig. 8 presents the computed and measured values of β for an MVD of 20 μm, 111 μm and 236 μm against the curvilinear coordinate originating from the airfoil leading edge. For all tested MVDs, the solver is able to remarkably predict the experimental results. Comparison is satisfactory both for the maximum value of the collection efficiency, although being slightly overestimated for the 111 μm droplets, and for the prediction of the impingement limits, the latter being very important since it is strongly correlated to the surface subject to ice accretion. A slight underestimation of the wetted area can be noted for the 20 μm droplets case. To the authors knowledge this may be attributed to uncertainties in the nominal values of the experimental conditions. A previous work [33], showed that uncertainties in the operating conditions can lead to some variability of the collection efficiency especially near the suction side impingement limits. Fig. 9 shows the volume fraction and the trajectories of 52 μm droplets around the naca23012 model computed with the Eulerian solver. The Eulerian formulation is able to properly represent the shadow regions appearing past the impingement limits on the

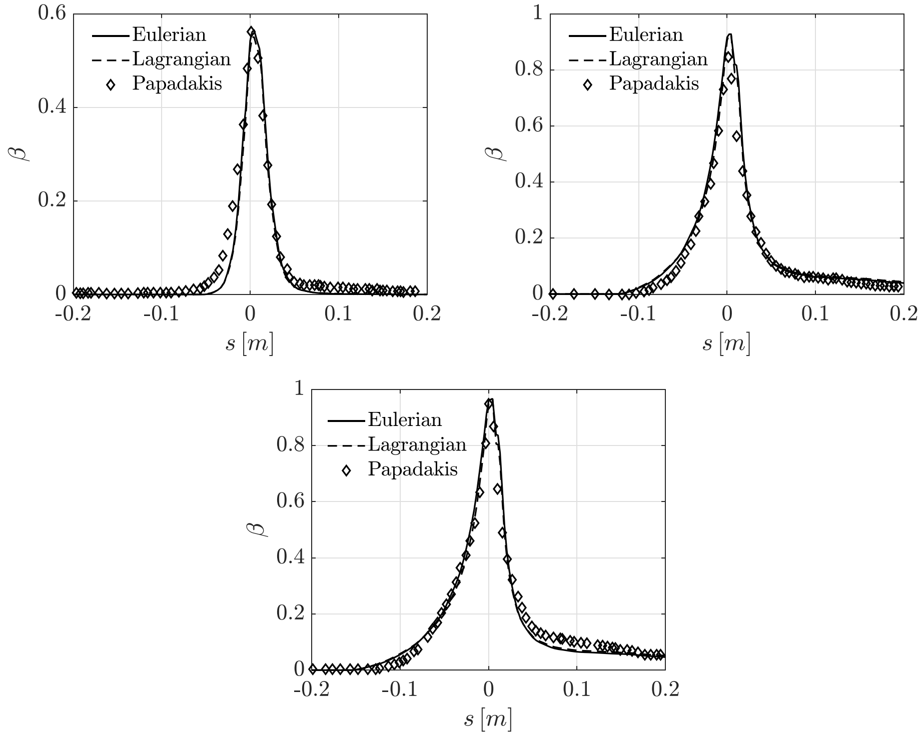

Fig. 8. Eulerian and Lagrangian collection efficiency on a NACA23012 airfoil. Comparison with experimental measurements for a spray with MVD of 20 μm (top left), 111 μm (top right) and 236 μm (bottom).

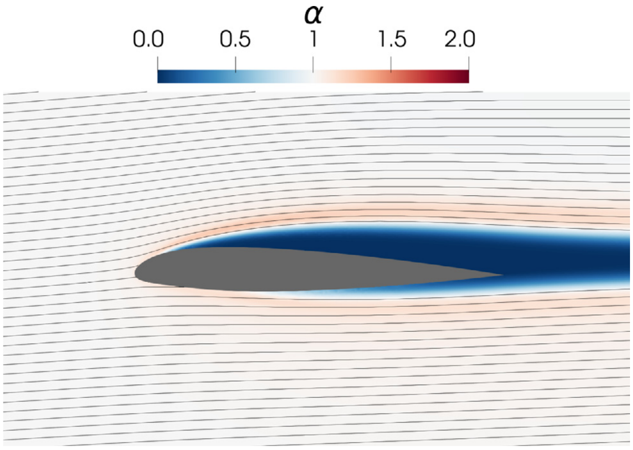

Fig. 9. Volume fraction field and droplet trajectories for the NACA23012 52 μm case.

wing that are characterized by the absence of particles. In then Eulerian solution, no-particle regions are in fact very low volume fraction regions due to numerical viscosity of the scheme.

The presented results are obtained by tracking a poly-dispersed cloud as explained in Section 2. The droplets size distribution of the spray was measured in the experimental campaign and reported as a discrete 10 or 27 bin distribution. Fig. 10 shows the effects of the different size discretizations on the 20 μm MVD case. With the finer characterization of the spray, the computed β better agrees with the experimental data. The effect is that of diffusing the value obtained with the mono-dispersed description to obtain a lower maximum value and more extended impingement limits. No sensible changes are noticeable between the results obtained with the 10 and 27 bin distributions.

The computational cost of the methodologies presented is briefly discussed for the computation of β on 2D geometries. The cost is assessed on the computation of the collection efficiency for a spray of 20 μm MVD discretized using a 10 bin distribution. The simulation wall time was measured on a PC equipped with an Intel i7-8700 CPU and 16 GB of DDR4

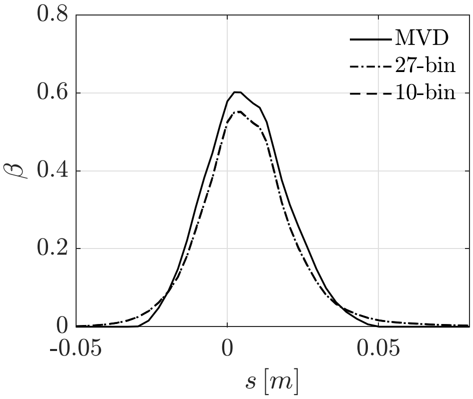

Fig. 10. Effects of different droplets size distribution on the collection efficiency. 20 μm MVD case.

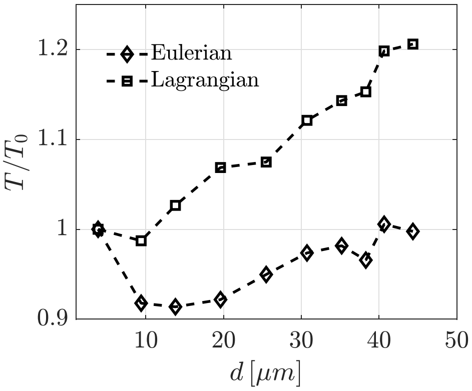

Fig. 11. Computational time for the multibin computation of the collection efficiency on a NACA23012 airfoil. Normalized wall time vs. droplets' diameter. The time to compute each bin is normalized by the wall time of the first diameter in the discrete distribution. T_0 is 3.3 and 12.5 s for the Lagrangian and Eulerian implementations, respectively. 20 μm MVD case.

memory. Simulations are run in parallel on all the available cores of the CPU. The Eulerian computations are parallelized via domain decomposition and the MPI library. The embarrassingly parallel nature of the Lagrangian implementation is exploited by tracking particles concurrently via the OpenMP library. Fig. 11 reports the wall time to simulate each bin in the spray distribution normalized by that of the smallest diameter in the discrete distribution T_0. T_0 amounts to 3.3 s or the Lagrangian methodology, and 12.5 for the Eulerian implementation. The time to solution is approximately 4 time faster with the Lagrangian code. The computational cost of each bin in the Eulerian frame does not show any dependency on the droplets size, whereas a linear relationship is observed for the Lagrangian computation. The increase in the solution time using the Lagrangian approach s due to its cost being proportional to the impinged surface area, which is larger for larger droplets. This aspect make has a large influence on the simulation cost for 3D configurations, as will be show in the following sections.

4.2. Droplet impingement on a horizontal tail

This validation case presents the seeding front adaptation of the Lagrangian algorithm and compares the Eulerian and Lagrangian collection efficiency prediction against experimental data on a 3D geometry. Droplet impingement was measured on a horizontal swept tail at the NASA IRT facility [34]. The tail model was mounted in the wind tunnel with an angle of attack of 6° in an airflow at Mach 0.23, temperature of 280 K and pressure of 95 147 Pa. Tests were run for a spray MVD of 21 μm and 92 μm. For second case, the collection efficiency was computed by approximating the impinging cloud as a mono-dispersed spray with diameter equal to the case MVD. The smaller MVD spray was instead discretized using a discrete 7-bins Langmuir-D distribution.

Fig. 12 reports the quad-tree like mesh used in the adaptive droplet seeding algorithm. Initially droplets are injected uniformly on a plane at a distance of 3 chord lengths from the model leading edge. Droplets are added incrementally by refining the seeding mesh as explained in Section 2. The distribution of the initial seeding points is isotropic in order to mimic the surface mesh at the impingement location. For an anisotropic target surface, an anisotropic initial seeding

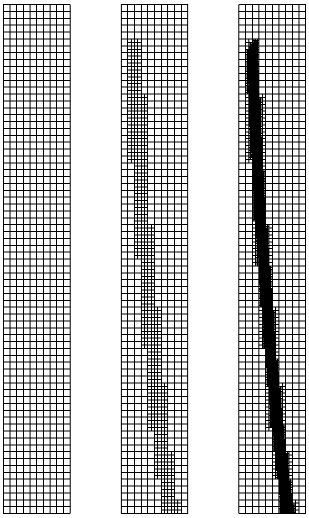

Fig. 12. Adaptation of the seeding front for the horizontal tail case. Left, initial injection positions. Center, first adaptation step. Right, fourth adaptation step.

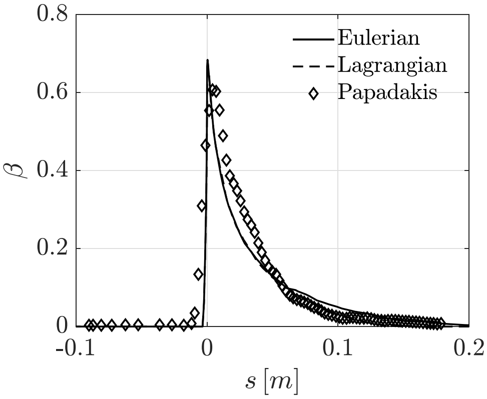

Fig. 13. Comparison of the Eulerian and Lagrangian collection efficiency algorithms against experimental data for the horizontal tail case. MVD of 21 μm.

plane should instead be used for optimal convergence of the collection efficiency. For the 21 μm case, the computation stopped at the 7th refinement level due to a residual on the β difference lower than 10^{-5}. The total number of droplets tracked is 1 613 024. In order to obtain the same accuracy on the collection efficiency, a uniform seeding of the droplets would have required 12 134 913 particles, which amounts to an almost ten fold increase in computational time. Similarly for the 92 μm MVD, target residual was reached at the 7th iteration with a total number of tracked droplets of 5 339 953. The quite large difference with respect to the number of trajectories needed for the smaller MVD case is due to the larger impingement area of the larger droplets. Nevertheless, the number of particles needed is still substantially lower than those needed for a uniformly refined cloud.

Figs. 13 and 16 show the comparison of the Eulerian and Lagrangian solvers against experimental data. Data was sampled at station placed 37 inches off the wind tunnel floor. As before, Eulerian and Lagrangian results match remarkably well. For the 21 μm case, comparison with the experiment is again satisfactory. Impingement limits are well captured both on the suction, where β presents a strong discontinuity, and pressure sides, where most of the impinging mass sticks to the tail. A small over prediction of the peak of β is again noticeable. Fig. 14 gives a glimpse of the particles volume fraction field. A large shadow zone is noticeable behind the whole tail. Particles are also centrifuged away from the vortex core that sheds from the tail tip. An additional step of the Lagrangian refinement procedure was performed in order to asses convergence of the computed collection efficiency. Fig. 15 compares the β along the measurement section obtained with 6.4 and 1.6 million trajectories, showing an almost complete overlap.

The computation was performed also for the larger droplet case and is reported in Fig. 16. Here a larger discrepancy can be seen with respect to the experimental measurements. Impingement limits are well captured on both sides and the value of β on the pressure side tail is also accurately computed. Differences can be seen past the bulk of the collection

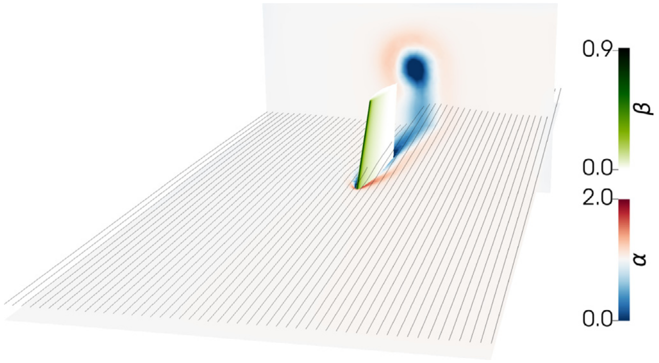

Fig. 14. Particle field and collection efficiency for the horizontal tail case. MVD of 21 μm. Eulerian solver.

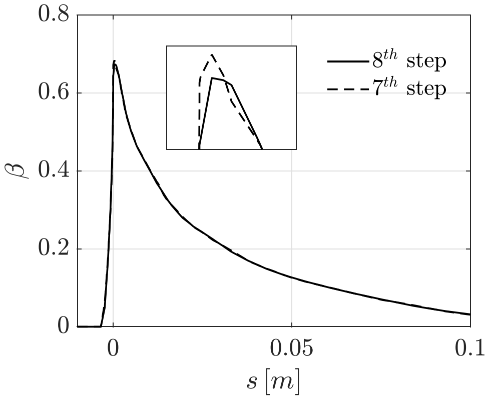

Fig. 15. Lagrangian collection efficiency for an MVD of 21 μm. Comparison of β at 7 and 8 adaptation steps. The number of particles is respectively 1.6 and 6.4 million.

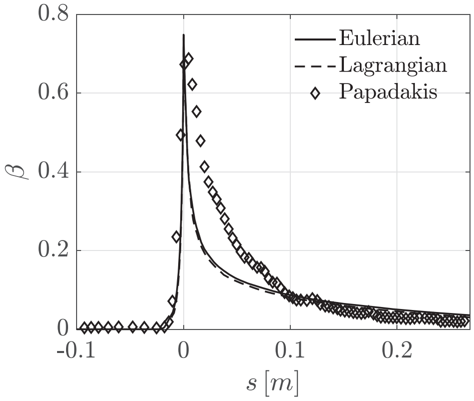

Fig. 16. Comparison of the Eulerian and Lagrangian collection efficiency algorithms against experimental data for the horizontal tail case. MVD of 92 μm.

efficiency on the pressure side between curvilinear coordinates of 0 and 0.1 m where β is underestimated. This result is in line with the output of the 1st AIAA Ice Prediction Workshop where diverse numerical code produced similar discrepancies at the tail. To the authors knowledge this is due to the splashing model employed.

The computational cost of the two methodologies is assessed on the computation of the collection efficiency for the 21 μm case. MVD discretized using a 7 bin distribution. The simulation wall time was measured on a PC equipped with an Intel i7-8700 CPU and 16 GB of DDR4 memory. Simulations are run in parallel on all the available cores of the CPU. The Eulerian computations are parallelized via domain decomposition and the MPI library. The embarrassingly parallel nature of the Lagrangian implementation is exploited by tracking particles concurrently via the OpenMP library. Fig. 17

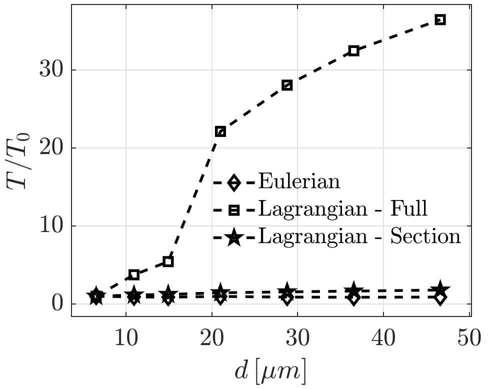

Fig. 17. Computational time for the multibin computation of the collection efficiency on a swept horizontal tail. Normalized wall time vs. droplets' diameter. The time to compute each bin is normalized by the wall time of the first diameter in the discrete distribution. T_0 is 150 and 79.3 s for the Lagrangian and Eulerian implementations, respectively. Computational cost of the Eulerian implementation and that of the Lagrangian approach for the computation of β in the full geometry and on a 10 cm span-wise section. 21 μm MVD case.

reports the wall time to simulate each bin in the spray distribution normalized by that of the smallest diameter in the discrete distribution T_0. T_0 amounts to 150 s or the Lagrangian methodology, and 79.3 for the Eulerian implementation. The outcome of the analysis is opposite of the 2D results. The Eulerian approach allows a much faster time to solution, especially for larger diameters. The figure reports the wall time per bin of the Eulerian and Lagrangian computations on the full horizontal tail and that of the Lagrangian for the evaluation of β on a 10 cm span-wise section of the geometry. As per the 2D computation, the Eulerian approach does not show a strong dependency of the computational time with the size of the droplets. The full simulation of 7 distinct particle sizes, takes under 10 min on the machine presented above. The same computation with the Lagrangian approach takes almost 6 hours on the same hardware. The cost dramatically increases with particle size. The cost to compute the larger droplet size, equal to 46.62 μm, was almost 40 times that of the smaller size of 6.5 μm. The Lagrangian approach is instead a competitive alternative if β is only needed on a portion of the target surface. The figure shows also the wall clock time for this case, where the collection efficiency is only computed on a 10 cm thick span-wise section of the tail. In this configuration the result is obtained in around 17 min, making the cost comparable to that of the Eulerian counterpart.

4.3. Secondary droplets effects on a high lift device

Ref. [34] reports collection efficiency measurements for SLD droplets on an high lift system made of three lifting surfaces: a slat, a main wing and a flap. Experiments were conducted at NASA IRT facility for MVDs of 92 μm on a 91.44 cm chord model at 4° angle of attack in a flow of air with velocity 78.2 m/s, temperature 278 K and pressure 95 630 Pa. This case, also part of the AIAA Ice Prediction Workshop, was selected in order to showcase the effects of secondary droplets coming as a result of splashing on the final value of β. In the previously presented results, particles only impinge on a restricted region near the leading edge and secondary droplets are advected downstream without ever impacting again on the model. On the high lift device instead, particles splashing on the slat element can impact on the rear elements increasing the total catch efficiency. Also, the more complex flowfield makes for a more interesting simulation, with SLD droplets being in a large velocity non equilibrium with the air due to the locally high curvature of the flow streamlines Fig. 18.

The hybrid Eulerian–Lagrangian algorithm is compared to the fully Lagrangian implementation on the prediction of the collection efficiency. Simulations were performed using a poly-dispersed spray. The cumulative LWC distribution is reported in Fig. 19, highlighting the 27 diameters values that were sampled to perform the simulations.

Lagrangian calculations were run with a tolerance on β of 10^{-5}. For the case at MVD 92 μm this amounts to tracking just 20 000 particles for the first LWC bin but almost 5 million for the last bin. The large number of particles required, is due to the larger impingement area compared to a standard single element wing and also to the secondary droplets that impinge again on the rear elements as it is clearly shown in Fig. 18. The Lagrangian re-injection step in the hybrid Eulerian–Lagrangian computations was performed by injecting n particles along each boundary face where splashing happens.

Fig. 20 shows the computed collection efficiency on slat, main wing and flap. The curves shown in the plot represent β computed with the Eulerian and Lagrangian solvers both with and without accounting for the secondary droplets. On the slat portion of the high lift device, splashed droplets have no influence on the final β as can be seen from the overlapping curves in the plot. Re-injected droplets are advected downstream and do not impinge again on the leading edge lift device. Both Eulerian and Lagrangian computations accurately predict the experimental collection efficiency throughout the whole surface.

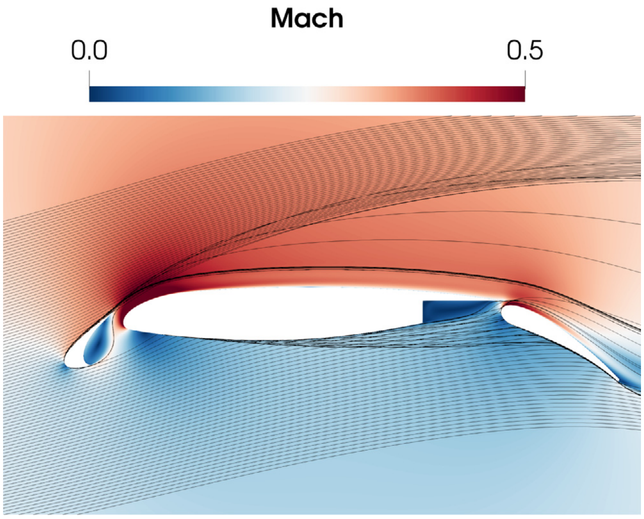

Fig. 18. Mach field around high lift device and particles trajectories of diameter 92 μm.

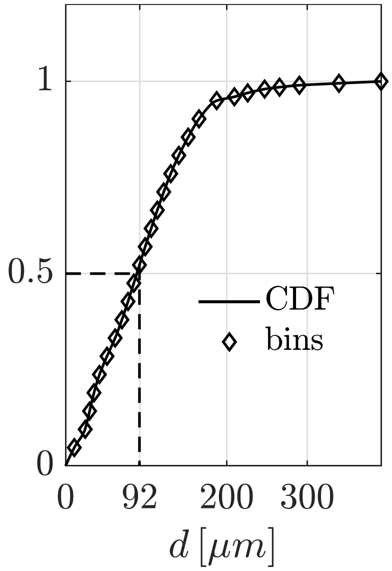

Fig. 19. Droplet size cumulative distribution for the 92 μm MVD case of the three element airfoil.

On the main element the collection efficiency is grossly under estimated. There is a quite large contribution due to secondary droplets near the stagnation point and for negative values of the surface coordinate but that is not enough to compensate for the missing mass. Again, Eulerian and Lagrangian results are in great agreement. Differences between prediction and experimental results may be caused by the relatively complex flowfield of this case. In the region near the main wing stagnation point, large non-equilibrium in the droplet and air velocities may lead to aerodynamic instabilities and droplet breakup, a phenomenon that is not modeled in the present work. Another reason could be the large turbulent nature of the flow, especially in the vicinity of the shear layer that separates at the slat leading edge. Turbulence effects on the particles are modeled only implicitly via the mean aerodynamic field. An explicit modeling of droplet-turbulence interaction, for instance via semi-empiric particle eddy interaction models, may be beneficial for improving the prediction.

On the flap the numerical simulations are remarkably accurate. The impingement limits and peak of beta are well predicted, and most of the mass on the pressure side is also captured. Here, the effects of secondary droplets are clearly noticeable as they allow to recover the missing mass at the stagnation point and on the pressure side.

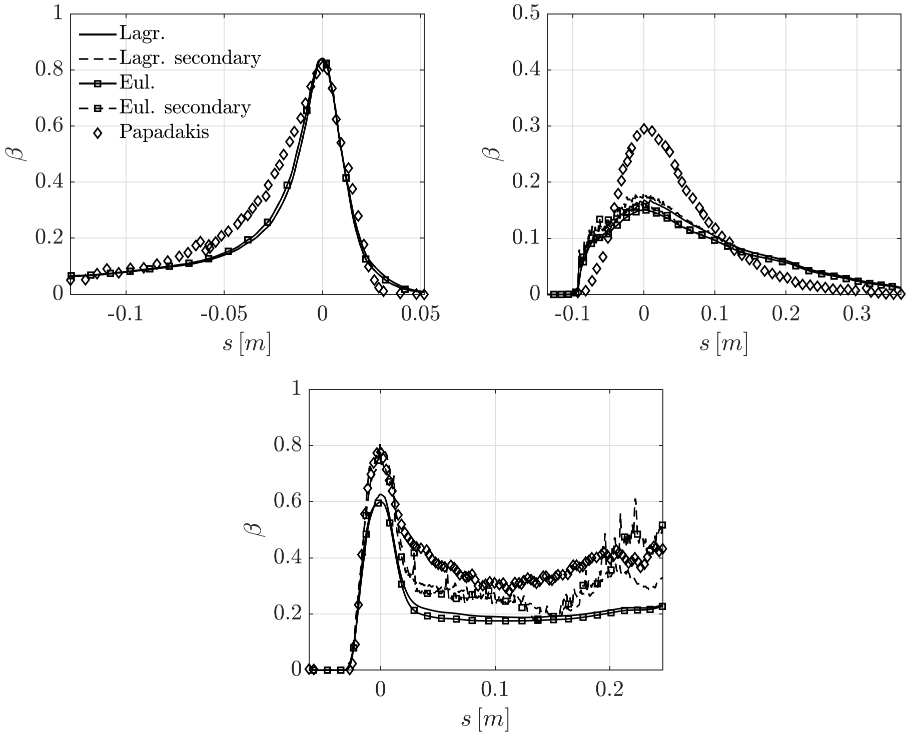

Fig. 20. Comparison of the Eulerian and Lagrangian collection efficiency algorithms against experimental data for the three element airfoil case. Top left, top right and bottom figures refer respectively to slat, main wing and flap. MVD of 92 μm.

5. Conclusion

In this work, Lagrangian and Eulerian algorithms for computing the collection efficiency in in-flight liquid icing applications are presented. Both approaches allow for the use of unstructured body fitted grids thus allowing computations on complex three dimensional geometries. A Lagrangian algorithm is presented together with an AMR implementation for solving the initial seeding and resolution shortcomings of Lagrangian methods. Arbitrary precision arithmetic is used in geometric computations to achieve robust tracking algorithms. The Eulerian FV approach presented allows to easily reuse standard CFD data structures and tools such as linear solvers, convergence acceleration methods and parallelization strategies since the system is cast in a conservative form. The problem of treating secondary splashed droplets is solved by tracking these particles using the Lagrangian implementation. Verification tests are performed to assess the correct implementation of the algorithms prior to presenting the validation results. The comparison against experimental collection efficiency data, shows that the presented algorithms are capable of predicting the water catch efficiency correctly for diverse conditions, small and SLD droplets, and geometries.

Tests on a NACA23012 airfoil highlighted no particular lack of the present methodology due to the simple geometry and flow conditions. On the three dimensional impingement on a horizontal tail, the splashing model employed predicted an excessive mass loss for the 92 μm case. On the more complex high lift device simulation, results are encouraging on the front and rear surfaces, whereas the collection efficiency on the main element is largely underestimated. This highlights the need of additional investigation in splashing ad physics models for three dimensional geometries and complex configurations.

In every test performed, the Eulerian formulation yielded results that are in complete agreement with the Lagrangian implementation. This justifies the use of the Eulerian approach in large three dimensional cases where β is needed on large surface areas which would require a prohibitively high number of particles with the Lagrangian method, as shown in the numerical experiments. The Lagrangian approach is still attractive due to the ease of testing new models and adding additional effects that were neglected in this work, such as lifting forces, breakup, and turbulent dispersion. Also, it can be used to compute the collection efficiency on a subset of the whole geometry, thus allowing the use of a smaller computational domain.

The V&V effort reported in this paper is a much needed step towards current and future work in ice accretion physics and modeling. Current efforts include, but are not limited to, novel droplet-wall interaction models, parallelization approaches for particle laden flows, virtual certification of aircraft and helicopters and uncertainty quantification analysis.

CRediT authorship contribution statement

Tommaso Bellosta: Conceptualization, Methodology, Software, Validation, Investigation, Visualization, Writing – original draft. **Giacomo Baldan:** Methodology, Software, Investigation. **Giuseppe Sirianni:** Methodology, Software, Investigation. **Alberto Guardone:** Conceptualization, Supervision, Funding acquisition, Project administration, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

The ICE GENESIS project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 824310.

References

[1] B.L. Messinger, Equilibrium temperature of an unheated icing surface as a function of air speed, J. Aeronaut. Sci. 20 (1) (1953) 29–42, http://dx.doi.org/10.2514/8.2520.

[2] T.G. Myers, Extension to the messinger model for aircraft icing, AIAA J. 39 (2) (2001) 211–218, http://dx.doi.org/10.2514/2.1312.

[3] J. Stefan, Ueber die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere, Ann. Phys. 278 (2) (1891) 269–286, http://dx.doi.org/10.1002/andp.18912780206.

[4] W. Wright, Validation results for LEWICE 3.0, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, http://dx.doi.org/10.2514/6.2005-1243.

[5] P. Villedieu, P. Trontin, D. Guffond, D. Bobo, SLD Lagrangian modeling and capability assessment in the frame of ONERA 3D icing suite, in: 4th AIAA Atmospheric and Space Environments Conference, http://dx.doi.org/10.2514/6.2012-3132.

[6] F. Petrosino, G. Mingione, A. Carozza, T. Gilardoni, G. D'Agostini, Ice accretion model on multi-element airfoil, J. Aircr. 48 (6) (2011) 1913–1920, http://dx.doi.org/10.2514/1.C031346.

[7] A. Donizetti, T. Bellosta, A. Rausa, B. Re, A. Guardone, Level-set mass-conservative front-tracking technique for multistep simulations of in-flight ice accretion, J. Aircr. (2023) 1–11, http://dx.doi.org/10.2514/1.C037027.

[8] H. Beaugendre, F. Morency, W.G. Habashi, FENSAP-ICE's three-dimensional in-flight ice accretion module: ICE3D, J. Aircr. 40 (2) (2003) 239–247, http://dx.doi.org/10.2514/2.3113.

[9] E. Radenac, H. Gaible, H. Bezard, P. Reulet, IGLOO3D computations of the ice accretion on swept-wings of the SUNSET2 database, SAE Technical Paper (2019) http://dx.doi.org/10.4271/2019-01-1935.

[10] S. Elghobashi, On predicting particle-laden turbulent flows, Appl. Sci. Res. 52 (4) (1994) 309–329.

[11] M. Hess, P. Koepke, I. Schult, Optical properties of aerosols and clouds: The software package OPAC, Bull. Am. Meteorol. Soc. 79 (5) (1998) 831–844.

[12] D. Rosenfeld, I.M. Lensky, Satellite-based insights into precipitation formation processes in continental and maritime convective clouds, Bull. Am. Meteorol. Soc. 79 (11) (1998) 2457–2476.

[13] Y. Bourgault, W.G. Habashi, J. Dompierre, G.S. Baruzzi, A finite element method study of Eulerian droplets impingement models, Internat. J. Numer. Methods Fluids 29 (4) (1999) 429–449.

[14] E. Iuliano, G. Mingione, F. Petrosino, F. Hervy, Eulerian modeling of large droplet physics toward realistic aircraft icing simulation, J. Aircr. 48 (5) (2011) 1621–1632, http://dx.doi.org/10.2514/1.C031326.

[15] D. Sotomayor-Zakharov, S. Bansmer, Finite-volume Eulerian solver for simulation of particle-laden flows for icing applications, Comput. & Fluids 228 (2021) 105009, http://dx.doi.org/10.1016/j.compfluid.2021.105009.

[16] C.T. Crowe, J.D. Schwarzkopf, M. Sommerfeld, Y. Tsuji, Multiphase Flows with Droplets and Particles, second ed., in: Engineering & Technology, Physical Sciences, CRC Press, Boca Raton, 2011, p. 509, http://dx.doi.org/10.1201/b11103.

[17] E. Loth, Numerical approaches for motion of dispersed particles, droplets and bubbles, Prog. Energy Combust. Sci. 26 (3) (2000) 161–223, http://dx.doi.org/10.1016/S0360-1285(99)00013-1.

[18] F.A. Morrison, An Introduction to Fluid Mechanics, Cambridge University Press, 2013, http://dx.doi.org/10.1017/CBO9781139047463.

[19] J.R.G. R. Clift, M.E. Weber, Bubbles, drops and particles, J. Fluid Mech. 94 (4) (1978) 795–796, http://dx.doi.org/10.1017/S0022112079221290.

[20] R. Honsek, W.G. Habashi, FENSAP-ICE: Eulerian modeling of droplet impingement in the SLD regime of aircraft icing, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006, http://dx.doi.org/10.2514/6.2006-465.

[21] C. Bai, A.D. Gosman, Development of methodology for spray impingement simulation, SAE Trans. 104 (1995) 550–568.

[22] A. Yarin, Drop impact dynamics: splashing, spreading, receding, bouncing, Annu. Rev. Fluid Mech. 38 (2006).

[23] W. Wright, M. Potapczuk, L. Levinson, Comparison of LEWICE and glennice in the SLD regime, in: 46th AIAA Aerospace Sciences Meeting and Exhibit, http://dx.doi.org/10.2514/6.2008-439.

[24] The CGAL Project, CGAL User and Reference Manual, 5.4 ed., CGAL Editorial Board, 2022.

[25] J. Dompierre, P. Labbè, M. Vallet, R. Camarero, How to subdivide pyramids, prisms and hexahedra into tetrahedra, in: 8th International Meshing Roundtable, 1999.

[26] G. Baldan, T. Bellosta, A. Guardone, Efficient parallel algorithms for coupled fluid-particle simulation, in: 9th Edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering, CIMNE, 2021, http://dx.doi.org/10.23967/coupled.2021.021.

[27] S. Jung, R. Myong, A second-order positivity-preserving finite volume upwind scheme for air-mixed droplet flow in atmospheric icing, Comput. & Fluids 86 (2013) 459–469, http://dx.doi.org/10.1016/j.compfluid.2013.08.001.

[28] C. Berthon, M. Breuß, M.-O. Titeux, A relaxation scheme for the approximation of the pressureless Euler equations, Numer. Methods Partial Differential Equations 22 (2) (2006) 484–505, http://dx.doi.org/10.1002/num.20108.

[29] M. Boileau, C. Chalons, M. Massot, Robust numerical coupling of pressure and pressureless gas dynamics equations for Eulerian spray DNS and LES, SIAM J. Sci. Comput. 37 (1) (2015) 79–102, http://dx.doi.org/10.1137/130945740, URL https://hal.archives-ouvertes.fr/hal-00906220.

[30] T.D. Economon, F. Palacios, S.R. Copeland, T.W. Lukaczyk, J.J. Alonso, SU2: An open-source suite for multiphysics simulation and design, AIAA J. 54 (3) (2016) 828–846, http://dx.doi.org/10.2514/1.J053813.

[31] F. Menter, Zonal two equation $k-\omega$ turbulence models for aerodynamic flows, in: 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference, 1993, http://dx.doi.org/10.2514/6.1993-2906.

[32] M. Papadakis, A. Rachman, S.-C. Wong, H.-W. Yeong, K.E. Hung, C.S. Bidwell, Water Impingement Experiments on a NACA 23012 Airfoil with Simulated Glaze Ice Shapes, American Institute of Aeronautics and Astronautics, Inc., 2004.

[33] G. Gori, P.M. Congedo, O. Le Maître, T. Bellosta, A. Guardone, Modeling in-flight ice accretion under uncertain conditions, J. Aircr. 59 (3) (2022) 799–813, http://dx.doi.org/10.2514/1.C036545.

[34] M. Papadakis, K.E. Hung, G. Vu, H.-W. Yeong, C.S. Bidwell, M.D. Breer, T.J. Bencic, Experimental Investigation of Water Droplet Impingement on Airfoils, Finite Wings, and an S-duct Engine Inlet, Technical Memorandum NASA/TM-2002-211700, NASA, 2002.