NUMERICAL METHOD FOR ICE ACCRETION ON 3D WINGS

J.M. Hospers , H.W.M. Hoeijmakers
Group of Engineering Fluid Dynamics, University of Twente
Enschede, The Netherlands
j.m.hospers@utwente.nl; h.w.m.hoeijmakers@utwente.nl

Keywords: *Icing, Multi-Phase, Splashing, CFD, Eulerian*

Abstract

*Computer simulations of the ice accretion process provide an attractive method for analyzing a wide range of icing conditions at low cost. An ice accretion model that accurately predicts ice growth shapes on arbitrary airfoils sections is valuable for the analysis of the sensitivity of airfoils for ice accretion. Furthermore the analysis of the influence of flow variables such as airspeed and angle of attack, pressure, temperature and humidity on ice accretion is studied easily. Such an approach can also be used to assess the energy requirements necessary to prevent and/or remove ice from an airfoil. Once the method has been validated, it will provide a cost-effective means of performing icing research studies which now rely, for an important part, on experimental techniques. In this paper, a computational method is presented that computes three dimensional ice accretion on multiple-element airfoils in specified icing conditions. The main part of the method is the method to compute the distribution of the supercooled water impinging on the wing surface, which is a challenge especially for so-called super-cooled large droplets (SLD). To this aim, for a given flow field solution, the numerical method (*Droplerian*) uses an Eulerian method to determine the spatial distribution of the Liquid Water Content (LWC) and the droplet velocities. To solve the equations for the droplet velocities and liquid water content distribution, Droplerian uses a Finite Volume Method for unstructured grids. Through the droplet velocities and Liquid Water Content at the surface of the airfoil the droplet catching efficiency is calculated. The method can handle a multi-disperse droplet distribution with an arbitrary number of droplet classes (bins) and contains a droplet splashing and droplet rebound model. The splashing and rebound models are indispensable for correctly treating impingement of SLD's. Once the droplet-catching efficiency and droplet impact velocity are determined, they are used as input for the icing model, which is based on Messinger's model for ice accretion.*

1 3D Eulerian Droplet Tracking

In previous studies the Eulerian droplet tracking method was validated by comparing its results with results from previous computational models and from two dimensional experimental data from Papadakis et al. for a NACA 23012 airfoil at 2.5°angle of attack (AoA) [3–6]. The originally two dimensional method has been extended to three dimensions and was compared to the same experimental impingement data and to the previous two dimensional results. The droplet quantities ρ_d and $\vec{U}_d$ are considered; where ρ_d is the droplet density in kilogram water per cubic meter of air, and $\vec{U}_d$ is the droplet velocity. The general Eulerian equations for mass and momentum conservation are formulated as:

$$\frac{\partial \rho_d}{\partial t} + \vec{\nabla}\cdot\rho_d\vec{U}_d = 0, \tag{1}$$

$$\frac{\partial \rho_d\vec{U}_d}{\partial t} + \vec{\nabla}\cdot\left(\rho_d\vec{U}_d\right)\vec{U}_d = \rho_d\vec{f}_D + \rho_d\left(1-\frac{\rho_a}{\rho_w}\right)\vec{g}, \tag{2}$$

with ρ_w the density of water and ρ_a the density of air. The only other source-term considered, besides the drag force $\vec{f}_D$, is due to gravity; for the present case other forces, such as lift force and Basset history force, are neglected.

A splashing model is employed in the Eulerian method, which uses the method by Honsek and Habashi [2], based on work of Trujille [8]; which has been described in [6]. A rebound model based on work from Bai and Gosman [1] was also included, both models are present in both the 2D and 3D methods.

In the following sections, the necessary changes to the 2D method are described as well as the resulting 3D catching efficiencies and ice accretion shapes. These results are compared to results of the 2D method and to experimental results.

2 Boundary Ordering

In 2D, the ice accretion method is fairly simple: starting at the stagnation point, the temperature T_s and freezing fraction f can be calculated, one control volume after another. In 3D the stagnation point has become a stagnation line, leading to a 2D surface flow.

This means that the sequential ordering of control volumes becomes less than trivial. This would mean that the surface flow could; either be solved iteratively, or, an alternative boundary ordering has to be found. In order to save on the computational cost that would be implied by solving the surface flow iteratively, while recursively iteration for temperature, the boundary order will be determined.

Considering the surface flow on an airfoil: similar to the 2D case, surface flow starts at a stagnation panel, continuing downstream in multiple directions. Assuming that direction of the surface flow is completely determined by the flow of air along the surface:

$$\frac{\vec{U}_{s,i}}{\left|\vec{U}_{s,i}\right|} = \frac{\vec{U}_{a,i}}{\left|\vec{U}_{a,i}\right|}, \tag{3}$$

for the i^{th} control volume.

This means that by following the flow of air along the surface, the order in which the surface flow passes through the control volumes can be determined. As an intermediate step an array $[\mathsf{rank}]$ can be determined, starting at 1, increasing for each control volume the flow has passed.

As a first step, the approximate stagnation point (or stagnation line in 3D) is determined:

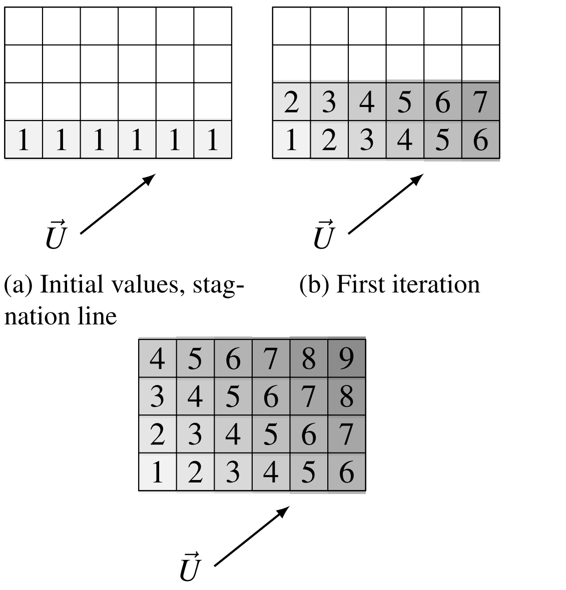
$$\mathsf{rank}_i = 1 : \begin{cases} u_{a,i} < 1\times10^{-5} & \& \quad v_{a,i} < 1\times10^{-5} \\ u_{a,i} < 1\times10^{-5} & \& \quad w_{a,i} < 1\times10^{-5} \\ v_{a,i} < 1\times10^{-5} & \& \quad w_{a,i} < 1\times10^{-5} \end{cases}. \tag{4}$$

Starting from these points, which are most likely stagnation points, the flow can be followed downstream, numbering the control volumes consecutively: if the velocity of the flow is in the same direction as the vector from one control volume to the next, the rank is determined. This process is repeated until no control volumes change rank. This is clarified in Algorithm 1 and illustrated in Fig 1.

Input: Initial ranking
Output: Final ranking
changed = **True**
while *changed* **do**
  changed = **False**
  for $i = 1, N_{conn}$ **do**
    $A =$ element left of i
    $B =$ element right of i
    if $\vec{U}_{a,A} \cdot (\vec{x}_B - \vec{x}_A)$ **and** $\mathsf{rank}_B < \mathsf{rank}_A + 1$ **then**
      $\mathsf{rank}_B = \mathsf{rank}_A + 1$
      if not *changed* **then**
        changed = **True**
    else if $\vec{U}_{a,B} \cdot (\vec{x}_A - \vec{x}_B)$ **and** $\mathsf{rank}_A < \mathsf{rank}_B + 1$ **then**
      $\mathsf{rank}_A = \mathsf{rank}_B + 1$
      if not *changed* **then**
        changed = **True**

Algorithm 1: Rank calculation

The order of calculation is then determined by looping over all ranked cells, shown in Algorithm 2 and Fig. 2. The order-counter starts at



(a) Initial values, stagnation line

(b) First iteration

(c) Last iteration

Fig. 1 : Illustration of rank calculation on a flat plate

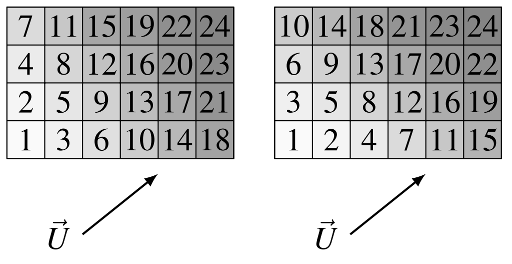
$\text{order}_i = 1$, for the first cell i, with $\text{rank}_i = 1$. It increases for every cell with $\text{rank}_i = 1$. If all cells have been checked, the process is repeated for $\text{rank}_i = 2$. Note that the order can change depending on the grid cell ordering.

Input: Final ranking
Output: Order
$k = 1$
while $k \leq \text{maxval}\,[\text{rank}]$ **do**
  $j = 1$
  for $i = 1, N_{cell}$ **do**
    if $\text{rank}_i = k$ **then**
      $\text{order}_i = j$
      $j = j + 1$

Algorithm 2: Order calculation

The resulting array $[\text{order}]$ contains the surface element numbers, ordered in flow direction, such that:

$$\begin{bmatrix} \text{order}_1 \\ \text{order}_2 \\ \vdots \\ \text{order}_{N_{cell}} \end{bmatrix} = \begin{bmatrix} \text{first cell to process} \\ \text{second cell to process} \\ \vdots \\ \text{last cell to process.} \end{bmatrix}$$



(a) Cells ordered up-down-left-right

(b) Cells ordered left-right-up-down

Fig. 2 : Illustration of order calculation on a flat plate

3 Catching Efficiency Calculation

Apart from the preprocessing of the boundary the catching efficiency (β) has to be determined on the surface. For each control volume on the surface the catching efficiency is determined:

$$\beta = \frac{\rho_d \vec{U}_d \cdot \vec{n}}{\text{LWC}\left|\vec{U}_{d,\infty}\right|}$$
$$\beta_i = \frac{\rho_{d,i} \vec{U}_{d,i} \cdot \vec{n}_i}{\text{LWC}\left|\vec{U}_{d,\infty}\right|}. \tag{5}$$

Any droplets that splash or rebound are already accounted for due to the change in mass and momentum introduced in [3], so the mass loss coefficient should not be included in Eq. 5.

4 Messinger Iteration

With the boundary order and catching efficiencies known, it is possible to calculate the heat and mass balance in every control volume. Because this is a 3D method, the Messinger model has to be modified somewhat.

4.1 Mass Flow

The Messinger model assumes a 2D flow, with surface flow entering the control volume (CV) from one side and leaving it from the other. In 3D, the surface flow enters and leaves as determined by the surface flow direction. However, for the local mass and energy balances the control volume is still considered 2D, see Fig 3. The 3D flow comes into play in determining which control volumes receive the run back mass going out of the current control volume. The mass flowing out of the control volume, $\dot{m}_{out}$, has to be converted to a 3D vector:

$$\vec{\dot{m}}_{out,i} = \dot{m}_{out,i}\frac{\vec{U}_{a,i}}{\left|\vec{U}_{a,i}\right|}, \tag{6}$$

such that

$$\dot{m}_{in,B} = \sum_{i=1}^{N_{conn}} \dot{m}_{out,A}\frac{\vec{U}_{a,B}}{\left|\vec{U}_{a,B}\right|}\cdot\vec{n}_{A,B}. \tag{7}$$

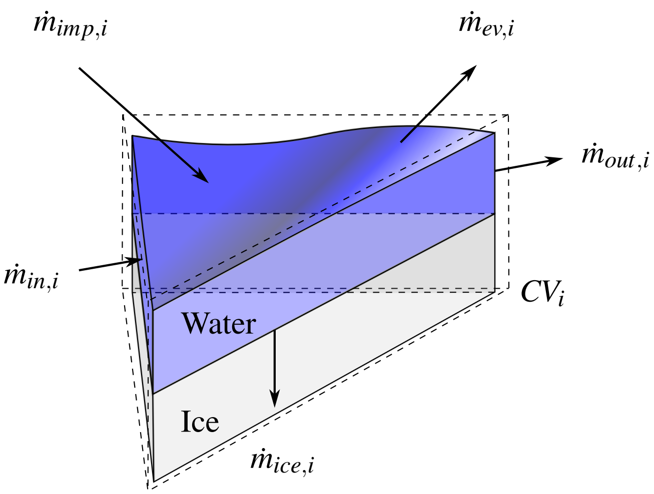


Fig. 3 : Messinger control volume, 3D, mass flows

Assuming that all surface flow is instantaneous, Algorithm 3 can be used to determine the inflow into every control volume.

5 Results

To validate the 3D model results are compared with results obtained with a similar 2D model, validated and explained in [6]. For a first comparison, the 2D geometry of [7] is expanded by 1 meter in the third dimension. A zero flux wall condition is imposed on the sides of the domain. The 3D results should be similar if not identical to the 2D results.

The input conditions from the Papadakis cases are listed in Table 1 and Table 2. An example of a catching effiency obtained using the input from

Input: Order, $\dot{m}_{out,i}$
Output: $[\dot{m}_{in}]$
$UX = 0$
for $j = 1, N_{conn}$ **do**
  $A =$ element left of j
  $B =$ element right of j
  $UXj = \frac{1}{2}\left(\vec{U}_{a,A}+\vec{U}_{a,B}\right)\cdot\frac{\vec{x}_B-\vec{x}_A}{|\vec{x}_B-\vec{x}_A|}$
  if $A == i$ **and** $UXj > 0$ **then**
    $UX = UX + UXj$
  else if $B == i$ **and** $UXj < 0$ **then**
    $UX = UX - UXj$
for $j = 1, N_{conn}$ **do**
  $A =$ element left of j
  $B =$ element right of j
  $UXj = \frac{1}{2}\left(\vec{U}_{a,A}+\vec{U}_{a,B}\right)\cdot\frac{\vec{x}_B-\vec{x}_A}{|\vec{x}_B-\vec{x}_A|}$
  if $A == i$ **and** $UXj > 0$ **then**
    $\dot{m}_{in,B} = \dot{m}_{in,B} + \frac{UXj}{UX}\dot{m}_{out,A}$
  else if $B == i$ **and** $UXj < 0$ **then**
    $\dot{m}_{in,A} = \dot{m}_{in,A} - \frac{UXj}{UX}\dot{m}_{out,B}$

Algorithm 3: Inflow calculation

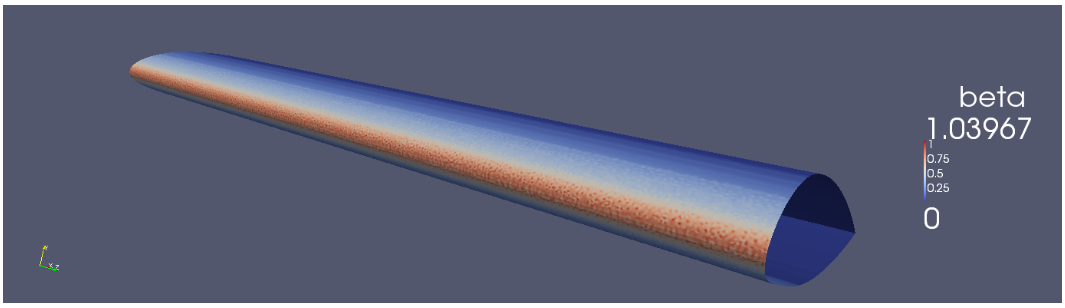
Table 1: Conditions for selected cases [7]

MVD	AoA	c	LWC
20 µm	2.5°	0.9144 m	0.19 g/m
236 µm	2.5°	0.9144 m	1.89 g/m

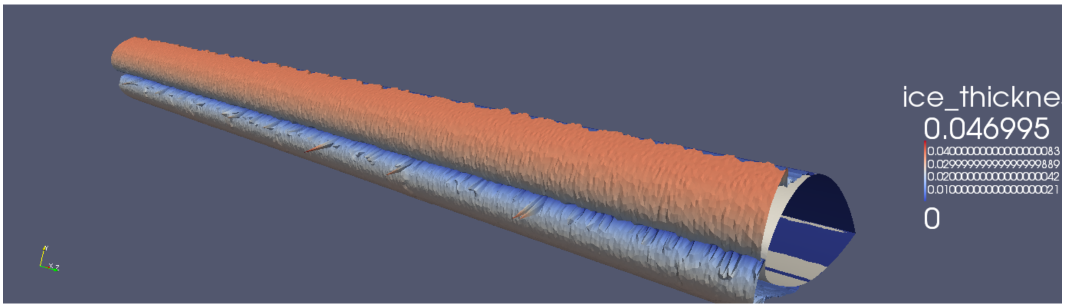
U_∞	T_∞	p_∞
78.23 m/s	299 K	101330 Pa
78.23 m/s	299 K	101330 Pa

Table 2: 10-Bin droplet distributions for selected cases [7]

Droplet bin	LWC [%]	Droplet size [µm] 20 µm MVD	236 µm MVD
1	5.0	3.850397	16.25037
2	10.0	9.390637	63.65823
3	20.0	13.80175	135.4827
4	30.0	19.60797	298.5197
5	20.0	25.4820	508.4572
6	10.0	30.73474	645.4684
7	3.0	35.19787	715.8689
8	1.0	38.32569	747.3936
9	0.5	40.66701	763.2455
10	0.5	44.36619	1046.767



(a) Catching efficiency



(b) Ice accretion shape

Fig. 4 : Three dimensional results for NACA 23012 at an MVD of 236 μm

Table 1 and Table 2 is shown for the 236 μm MVD in Fig. 4. As an example, the 3D ice accretion shape, obtained by lowering T_∞ to 263 K is also shown.

The same catching efficiency is plotted in two dimensions in Fig. 5. The points mark the 3D results, the solid line marks the 2D results. This shows that 3D and 2D are very similar, however, due to the large variation in orientation of the 3D surface element a larger variation is observed in the catching efficiency.

A more interesting case can be created by sweeping the geometry. The NACA 23012 airfoil is swept by 25°. Furthermore, the wing is now finite, it fills only half the width of the computational domain. This should allow for a difference in catching efficiency from root to tip and show some 3D effects. All other parameters are kept according to Table 1 and Table 2.

Looking at Fig. 6, the droplet trajectories are observed having a 3D component. The droplet move from root to tip. For the ice accretion shape, an increase in ice thickness is observed near the root of the wing, due to a smaller amount of run-back water.

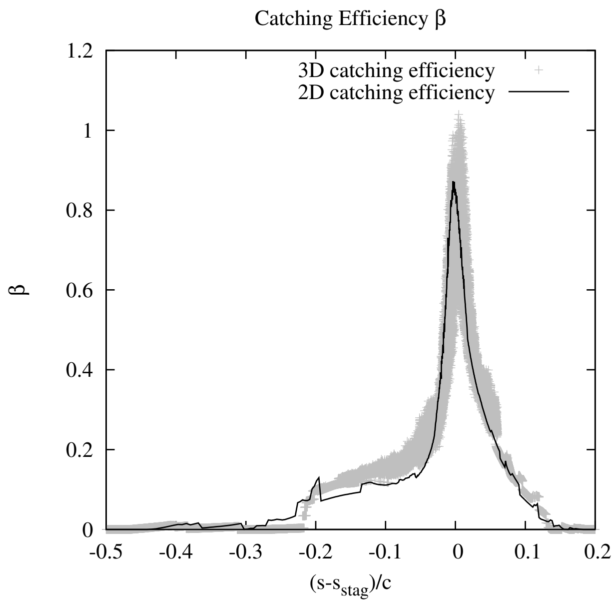
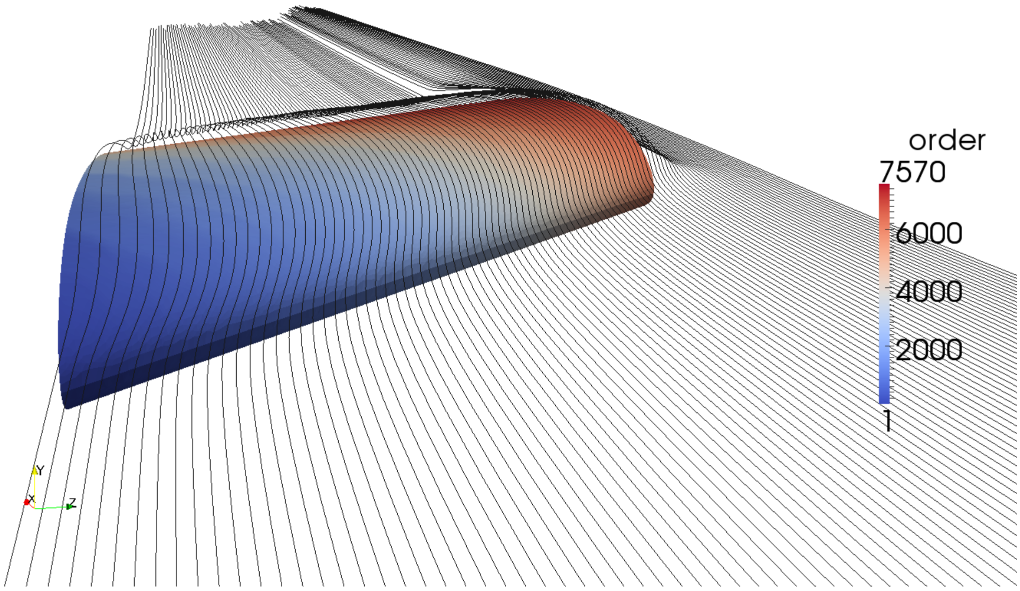
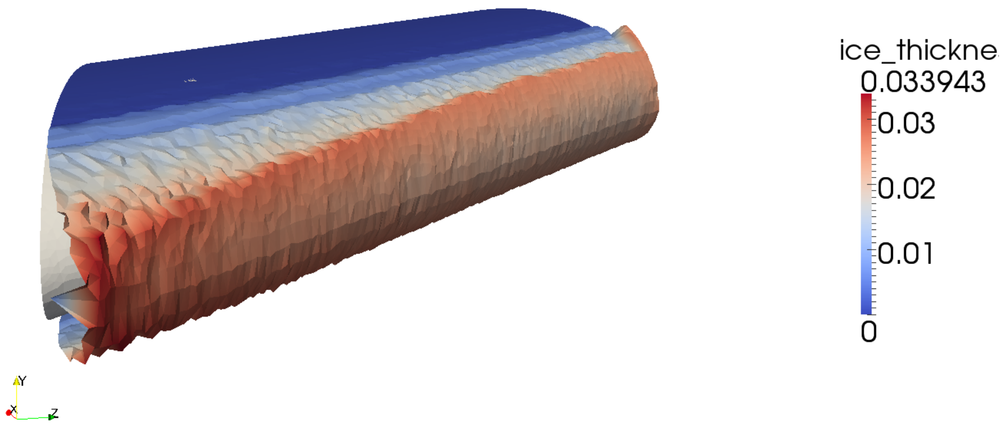


Fig. 5 : Three dimensional catching efficiency for NACA 23012 at an MVD of 236 μm



(a) Element order and droplet trajectories



(b) Ice accretion shape

Fig. 6 : Three dimensional results for NACA 23012 at an MVD of 236 µm

6 Conclusion

A 3D ice accretion method has been created by extending an existing 2D Eulerian ice accretion code. It has been shown that this extended model, including a splashing and rebound model, provides similar results to its 2D predecessor. For a semi-2D straight airfoil ice accretion shapes, droplet trajectories, and catching effiencies are just as easily obtained in three dimensions; however, due to the large variation in element orientation for this unstructured method, leads to results which vary along the span of the airfoil. For swept wings an expected variation in ice thickness was observed.

References

[1] C. Bai and A.D. Gosman. Development of methodology for spray impingement simulation. Technical Report 950283, SAE, 400 Commonwealth Drive, Warrendale, PA, 1995.

[2] R. Honsek, W.G. Habashi, and M.S. Aubé. Eulerian modeling of in-flight icing due to supercooled large droplets. *Journal of Aircraft*, 45(4):1290–1296, July-August 2008. doi: 10.2514/1.34541.

[3] J.M. Hospers and H.W.M. Hoeijmakers. Numerical prediction of SLD ice accretion on multi-element airfoils. In *48th Aerospace Sciences Meeting*, Orlando, FL, January 2010.

[4] J.M. Hospers and H.W.M. Hoeijmakers. Numerical prediction of SLD ice accretion on multi-element airfoils. In *7th International Conference on Multiphase Flow*, Tampa, FL, May 2010.

[5] J.M. Hospers and H.W.M. Hoeijmakers. Numerical simulation of SLD ice accretion. In *27th International Congress of the Aeronautical Sciences, ICAS 2010*, Nice, France, September 2010.

[6] J.M. Hospers and H.W.M. Hoeijmakers. Numerical simulation of SLD ice accretion. In *SAE 2011 International Conference on Aircraft and Engine Icing and Ground Deicing*, Chicago, IL, June 2011.

[7] Michael Papadakis, Arief Rachman, See-Cheuk Wong, Hsiung-Wei Yeong, Kuohsing E. Hung, Giao T. Vu, and Colin S. Bidwell. Water droplet impingement on simulated glaze, mixed and rime ice accretions. Technical Report NASA/TM-2007-213961, NASA, October 2007.

[8] M.F. Trujillo, W.S. Mathews, C.F. Lee, and J.F. Peters. Modelling and experiment of impingement and atomization of a liquid spray on a wall. *International Journal of Engine Research*, 1(1):87–105, 2000. ISSN 1468-0874. doi: 10.1243/1468087001545281.

Copyright Statement