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 airfoil 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 $\rho_d$ and $U_d$ are considered; where $\rho_d$ is the droplet density in kilogram water per cubic meter of air, and $U_d$ is the droplet velocity. The general Eulerian equations for mass and momentum conservation are formulated as:

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

(1)

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

(2)
with \( \rho_w \) the density of water and \( \rho_a \) 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:

\[
\frac{\vec{U}_{s,i}}{|\vec{U}_{s,i}|} = \frac{\vec{U}_{a,i}}{|\vec{U}_{a,i}|},
\]

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 \( \text{[rank]} \) 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:

\[
\text{rank}_i = 1 : \begin{cases} 
 u_{a,i} < 1 \times 10^{-5} & \text{and} & v_{a,i} < 1 \times 10^{-5} \\
 v_{a,i} < 1 \times 10^{-5} & \text{and} & w_{a,i} < 1 \times 10^{-5} \\
 w_{a,i} < 1 \times 10^{-5} & \text{and} & w_{a,i} < 1 \times 10^{-5}
\end{cases}
\]

(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.

### Algorithm 1: Rank calculation

**Input:** Initial ranking  
**Output:** Final ranking  
\( \text{changed} = \text{True} \)

while \( \text{changed} \) do

\( \text{changed} = \text{False} \)

for \( i = 1, N_{\text{conn}} \) do

\( A = \) element left of \( i \)

\( B = \) element right of \( i \)

if \( \vec{U}_{a,A} \cdot (\vec{x}_B - \vec{x}_A) \) and \( \text{rank}_B < \text{rank}_A + 1 \) then

\( \text{rank}_B = \text{rank}_A + 1 \)

if not \( \text{changed} \) then

\( \text{changed} = \text{True} \)

else if \( \vec{U}_{a,B} \cdot (\vec{x}_A - \vec{x}_B) \) and \( \text{rank}_A < \text{rank}_B + 1 \) then

\( \text{rank}_A = \text{rank}_B + 1 \)

if not \( \text{changed} \) then

\( \text{changed} = \text{True} \)

end if
end if
end for

end while

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
Numerical Method for Ice Accretion on 3D Wings

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

order$_i$ = 1, for the first cell $i$, with rank$_i$ = 1. It increases for every cell with rank$_i$ = 1. If all cells have been checked, the process is repeated for 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  

\[
\begin{aligned}
&j = 1 \\
&\text{for } i = 1, N_{cell} \text{ do} \\
&\quad \text{if rank}_i = k \text{ then} \\
&\quad \quad \text{order}_i = j \\
&\quad \quad j = j + 1
\end{aligned}
\]

**Algorithm 2:** Order calculation

The resulting array [order] contains the surface element numbers, ordered in flow direction, such that:

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

**3 Catching Efficiency Calculation**

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

\[
\beta = \frac{\rho_d \vec{U}_d \cdot \vec{n}}{\text{LWC} |\vec{U}_{d,\infty}|}
\]

\[
\beta_i = \frac{\rho_{d,i} \vec{U}_{d,i} \cdot \vec{n}_i}{\text{LWC} |\vec{U}_{d,\infty}|}
\]

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, \( \dot{m}_{\text{out}} \), has to be converted to a 3D vector:

\[
\vec{\dot{m}}_{\text{out},i} = \dot{m}_{\text{out},i} \frac{\vec{U}_{a,i}}{\left| \vec{U}_{a,i} \right|},
\]

such that

\[
\dot{m}_{\text{in},B} = \sum_{i=1}^{N_{\text{conn}}} \dot{m}_{\text{out},i} \frac{\vec{U}_{a,B}}{\left| \vec{U}_{a,B} \right|} \vec{n}_{A,B}.
\]

\[\text{(6)}\]

\[\text{(7)}\]

![Fig. 3: Messinger control volume, 3D, mass flows](image)

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 efficiency obtained using the input from

**Table 1: Conditions for selected cases [7]**

<table>
<thead>
<tr>
<th>MVD</th>
<th>AoA</th>
<th>c</th>
<th>LWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 ( \mu )m</td>
<td>2.5(^{\circ})</td>
<td>0.9144 m</td>
<td>0.19 g/m</td>
</tr>
<tr>
<td>236 ( \mu )m</td>
<td>2.5(^{\circ})</td>
<td>0.9144 m</td>
<td>1.89 g/m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( U_\infty )</th>
<th>( T_\infty )</th>
<th>( p_\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>78.23 m/s</td>
<td>299 K</td>
<td>101330 Pa</td>
</tr>
<tr>
<td>78.23 m/s</td>
<td>299 K</td>
<td>101330 Pa</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Droplet</th>
<th>LW C [%]</th>
<th>20 ( \mu )m MVD</th>
<th>236 ( \mu )m MVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>bin 1</td>
<td>5.0</td>
<td>3.850397</td>
<td>16.25037</td>
</tr>
<tr>
<td>2</td>
<td>10.0</td>
<td>9.390637</td>
<td>63.65823</td>
</tr>
<tr>
<td>3</td>
<td>20.0</td>
<td>13.80175</td>
<td>135.4827</td>
</tr>
<tr>
<td>4</td>
<td>30.0</td>
<td>19.60797</td>
<td>298.5197</td>
</tr>
<tr>
<td>5</td>
<td>20.0</td>
<td>25.4820</td>
<td>508.4572</td>
</tr>
<tr>
<td>6</td>
<td>10.0</td>
<td>30.73474</td>
<td>645.4684</td>
</tr>
<tr>
<td>7</td>
<td>3.0</td>
<td>35.19787</td>
<td>715.8689</td>
</tr>
<tr>
<td>8</td>
<td>1.0</td>
<td>38.32569</td>
<td>747.3936</td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>40.66701</td>
<td>763.2455</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
<td>44.36619</td>
<td>1046.767</td>
</tr>
</tbody>
</table>
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 $T_\infty$ 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 runback water.
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 efficiencies 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


Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2012 proceedings or as individual off-prints from the proceedings.