Improvements to GlennICE Collection Efficiency Algorithm

Wright, William B.[1]
HX5, LLC, Cleveland, OH, 44135, USA

GlennICE (Glenn Icing Computational Environment) is a computational tool designed to calculate ice growth on complex three-dimensional geometries using the input from a user-supplied computational fluid dynamics (CFD) solution for the geometry of interest. The NASA John H. Glenn Research Center at Lewis Field is developing this tool to aid those evaluating, designing and certifying aircraft, engines, and aircraft components for flight in icing conditions. This domestically available software is being developed to enable the introduction of new icing physics into a computational environment in a manner that is open for evaluation and eventual use by industry, academia, and other government organizations. This paper will document recent improvements that reduce the number of trajectories needed to converge on collection efficiency. These improvements are documented using example cases with complex geometries.

I. Nomenclature

x	x-direction, m
y	y-direction, m
z	z-direction, m
beta	collection efficiency, dimensionless

II. Introduction

A research project is underway at the NASA Glenn Research Center (GRC) to produce computer software that can accurately predict ice growth under any meteorological conditions for any aircraft surface. At present, the software does not have ice crystal capability, nor does it have any adjustments for large drops such as splashing or bouncing. The current software has only been exercised in Appendix C conditions. The Icing Branch at GRC has previously produced computer software [1] [2] over the last thirty years for performing icing simulation. While some of these tools have been collaborative projects, most have been developed primarily by one person with some assistance by others. The state of computing has also changed dramatically in that time period. Early on, design decisions were made in these tools that took into consideration the limited capabilities of early computers. Additionally, the personnel tasked to design and write this software were icing experts, not software experts. As these codes have grown in complexity and have been accepted by users as production icing tools, there has arisen a need for the developers to adhere to standard software practices used to develop commercial software. The goals of these practices, as they relate to the GlennICE effort, include creating a team-based development process rather than the previous individual-based process; building modular code elements that can be re-used or extended as needed by several processes; designing or redesigning code elements with future capabilities in mind; and developing efficient testing methodologies for testing and validation of the code. This report will, in part, provide an update on efforts to improve computational efficiency by optimizing the trajectory search capability with minimal effects on accuracy of the results.

III. Methodology

Particle Trajectories

GlennICE integrates particle trajectories using a method developed by Dormand and Prince [3]. It is a 4^{th} order Runge-Kutta scheme that is designed to minimize error of the 5^{th} order term. This means that it computes both the fifth order result and the 4^{th} order result and returns the difference as the error. The algorithm was developed by Hairer et. al.[4] and is the default integrator used by MATLAB [5] and GNU Octave [6]. The interface between the DOPRI5 routines and GlennICE resides in a subroutine that contains the derivative function, which in this case is the equation

[1] Research Engineer, Icing Branch, Senior Member AIAA.

of motion in the Lagrangian reference frame. Particles are released from a user-supplied bounding box and subsequently refined based on the distance of the trajectory to the surface or based on whether it hits. Details of this method were presented previously[7].

Convergence Criteria

GlennICE performs convergence by use of refinement iterations. Convergence in this context is the ability to reduce the error on the rate of water collected on each CFD face and is distinct from convergence as applied to a CFD flow solver. Within each iteration, GlennICE determines where new trajectories are released and how many are released as previously described. Prior to the release of GlennICE 3.2 there was no default convergence criteria that was used. The user would simply keep running refinement iterations until the results looked acceptable in their engineering judgment. Previous publications from Rigby [8] and Porter [9] demonstrated means of accelerating convergence by looking at the convergence of stream tubes and bounded faces respectively. As this paper expands on those definitions, this paper will present those definitions again as well as describe the modifications made for GlennICE 4.0 and the branch bounded_faces.

An example of the stream tube area is shown in Figure 1. The outer triangles are CFD faces on the surface and the red dots are trajectory impact locations. The connectivity of the impact locations is based on the triangulation of the upstream release points. In GlennICE, a stream tube is said to be contained when all the red dots shown impact the same CFD face. The metric GlennICE uses for convergence states that if a given fraction of the stream tubes for a CFD face are contained (as defined by variable FRACTION_CONTAINED_TOL), then convergence was achieved. This is a very conservative definition, since the stream tube should be contained if the gray circle in the figure resides on a single CFD face, but this area is difficult to compute. Additionally, if all three hit locations from an upstream release point triangle hit the same CFD face, that triangle is said to be bounded and that triangle is no longer refined. This is controlled by the input boolean TURN_OFF_BOUNDED_FACES. In his paper, Porter showed that this did not affect the accuracy of collection efficiency. The three small triangles on the left side of Figure 1 are bounded faces. Other convergence criteria implemented for GlennICE 3.2 that are addressed in this paper are MAX_FACE_COUNT and USE_NONUNIFORM_TOLERANCE. The parameter MAX_FACE_COUNT simply sets an upper limit to the number of trajectories needed to find collection efficiency on a given CFD face. The current default value is 600. The parameter USE_NONUNIFORM_TOLERANCE relaxes the convergence criteria near the impingement limit. Numerous cases show that GlennICE can expend a large number of trajectories trying to resolve collection efficiency in regions of low mass flow and this boolean relaxes the fraction contained tolerance in these regions. This will be demonstrated in this report.

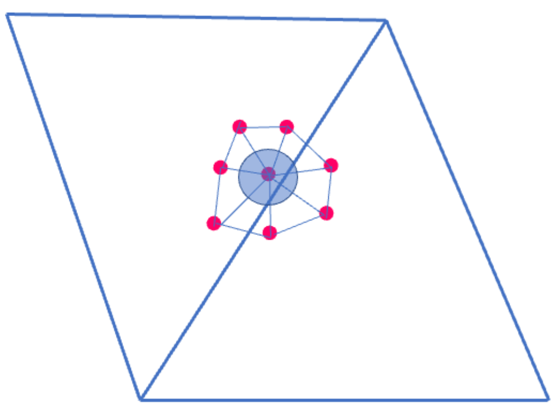


Figure 1 Definition of a Stream Tube

For GlennICE 4.0, convergence was improved by enforcing the MAX_FACE_COUNT criteria and by using the boolean USE_NONUNIFORM_TOLERANCE to turn off refinement in impingement limit regions rather than simply relax the criteria itself. Previously, the MAX_FACE_COUNT criteria was not being strictly enforced. The issue is illustrated in Figure 2. In this figure, if the seed point represented by the center (red) circle is set to refine then all the green points are added as release points of trajectories even though all of the blue seed points hit faces that exceed the value of MAX_FACE_COUNT. The fix was to prevent refinement of a seed point edge if either point hit a surface that exceeded the value of MAX_FACE_COUNT.

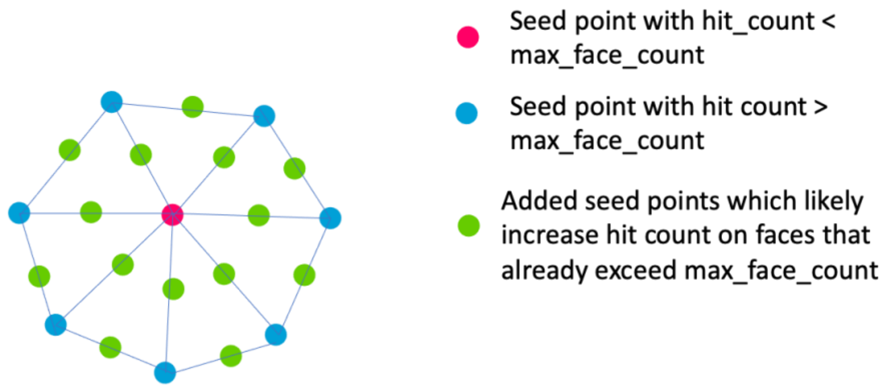


Figure 2 Stream Tube with Added Refinement

The second change made for GlennICE 4.0 was to turn off refinement of faces in low impingement regions when the boolean USE_NONUNIFORM_TOLERANCE was selected. The change reduces the value of MAX_FACE_COUNT near the limit so fewer trajectories are computed in this region. This could potentially impact collection efficiency accuracy, so evaluation of this change for sample cases is important.

For test branch bounded_faces, additional computational efficiency was achieved by leveraging the concept of bounded faces described earlier. This branch is still under review and has not been accepted as part of GlennICE. Since it was shown in the paper by Porter [9] that turning off refinement of bounded faces did not affect collection efficiency results, then it is hypothesized that the mass flux impinging a face can be calculated by extrapolating the mass hitting the bounded faces to the entire CFD face. This is illustrated in Figure 3. In this diagram, the mass hitting the area controlled by the smaller triangles is known, since the faces are bounded. In branch bounded_faces, if the fraction of the CFD face that is bounded exceeds the value of the parameter FRACTION_BOUNDED_TOL then the CFD face is said to be converged and that the mass impinging that CFD face can be extrapolated from the mass hitting the bounded region by multiplying the mass hitting the bounded region by the ratio of the two areas. In contrast to the fraction_contained tolerance, all of the smaller triangles are bounded, but only the nodes in red are contained.

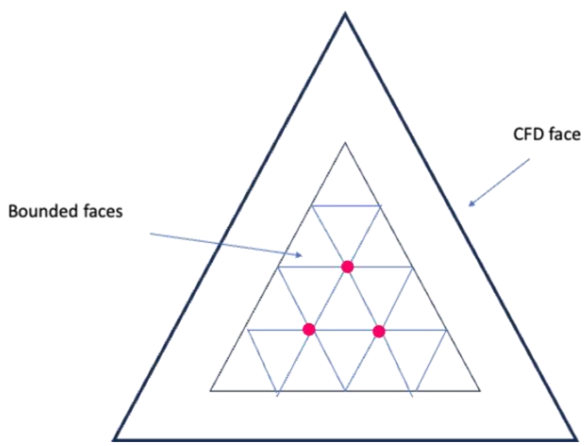


Figure 3 Bounded Faces Diagram

IV.Results

The first case is an analysis of a multiwire probe reported by Chen [10]. The geometry is shown in Figure 4 below. The Multi-Element Probe (Multiwire) consists of three heated sensing elements mounted inside of a heated shroud, as seen in this figure. One of these wires is cylindrical and has a diameter of 0.5334 mm (0.021 inch). The other two elements are known as half-pipes and have a half-circle geometry with an outer diameter of 2.0182 mm (0.083 inch). One of these elements has its concave surface facing the flow while the other element has its convex surface facing the. The length of each wire varies. The shroud has an outside diameter of 28.58 mm (1.13 inch) and an inside diameter of 26.82 mm (1.06 inch).

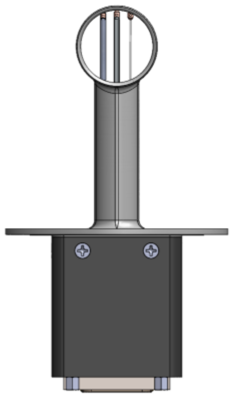
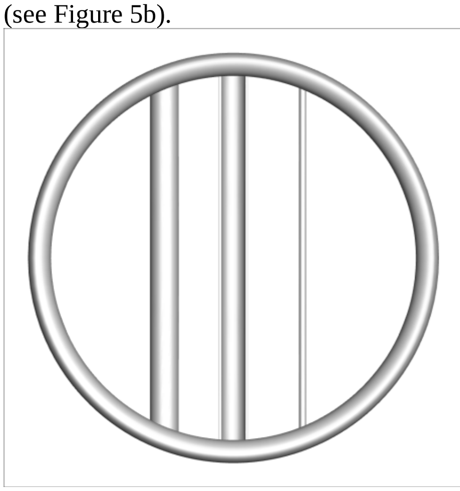
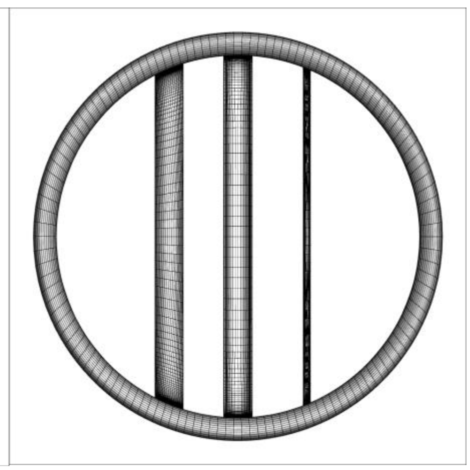


Figure 4 SEA Inc. Multi-Element Probe (Multiwire)

The Multiwire model used for simulation does not include the strut or the compensation wire (see Figure 5a). The compensation wire is in the shadow of the half-pipe and is therefore relatively unaffected by the cloud. In addition, the half-pipe with its convex surface facing the flow has been modeled as a solid cylinder. The computational domain for the Multiwire is a square block with side length 1.27 m (50 in). The probe is centered in the block. This is large enough to capture upstream effects. Structured surface meshes was used with an unstructured volume mesh (see Figure 5b).



(a) Frontal View of Multiwire Shroud and Elements Geometry Used for Simulations



(b) Frontal View of Multiwire Shroud and Elements Mesh Used for Simulations

Figure 5 SEA Inc. Multi-Element Probe

NASA's FUN3D 13.7 CFD software was used to numerically calculate a flow solution [11]. All probe surfaces are set as non-slip. The farfield is set as Riemann invariant. A Spalart-Allmaras turbulance model was used. A second-order backward differencing scheme was used for time-accurate temporal time integration. The frequency of vortex shedding was estimated from the diameter of the largest sensing element. The volume grid consists of 2,957,775 elements and 1,495,045 points. The casing and each of the three wires are separate surfaces. Results using GlennICE were based on running 21 refinement iterations with a monodispersed drop size of 20 µm. This number of iterations was sufficient to converge the collection efficiency results using version 3.2 for the three wires, but the casing was only 69.63% converged. Further refinement using version 3.2 could not be performed due to a failure with the Delaunay triangulation package, delaunator [12]. For version 4.0, this package was replaced with the CDT [13] package. After 21 iterations, 35,608,886 trajectories were calculated using GlennICE version 3.2. Due to this limitation, the same number of refinement iterations was performed for each version of GlennICE for consistency. For GlennICE version 4.0, 4,130,499 trajectories were performed, a speed up of 8.6x fewer trajectories. This occurs because the inner wall of the casing has a small number of hits, possibly due the nature of the time averaged flow solution calculated. The impingement from these hits does not show up on any plot of collection efficiency. Despite this, GlennICE version 3.2 kept trying to refine those hits, while the improvements implemented in GlennICE 4.0 stops this refinement. Figure 6 shows that the collection efficiency plots were unaffected by this change. The data for this slice was taken at the centerline of the probe (y=0). For this case, branch bounded_faces required 20 iterations and 3,129,564 trajectories to obtain converged collection efficiencies. Additionally, it can be seen in the figure that collection efficiency was not impacted by computing the value from the extrapolated value from the bounded faces mass flux calculated, validating that this change does not affect results. Figure 9 shows the trajectories required for convergence for a range of drop sizes. On average, GlennICE 4.0 was 11 times faster than GlennICE 3.2 while branch bounded_faces was 14.8 times faster than GlennICE 3.2 for this geometry over the range of drop sizes studied. Figure 10 shows the total mass flow collected by the casing and each of the wires for the drop size range shown in the previous figure. This plot demonstrates that the calculated mass flow is unaffected by reducing the number of trajectories calculated.

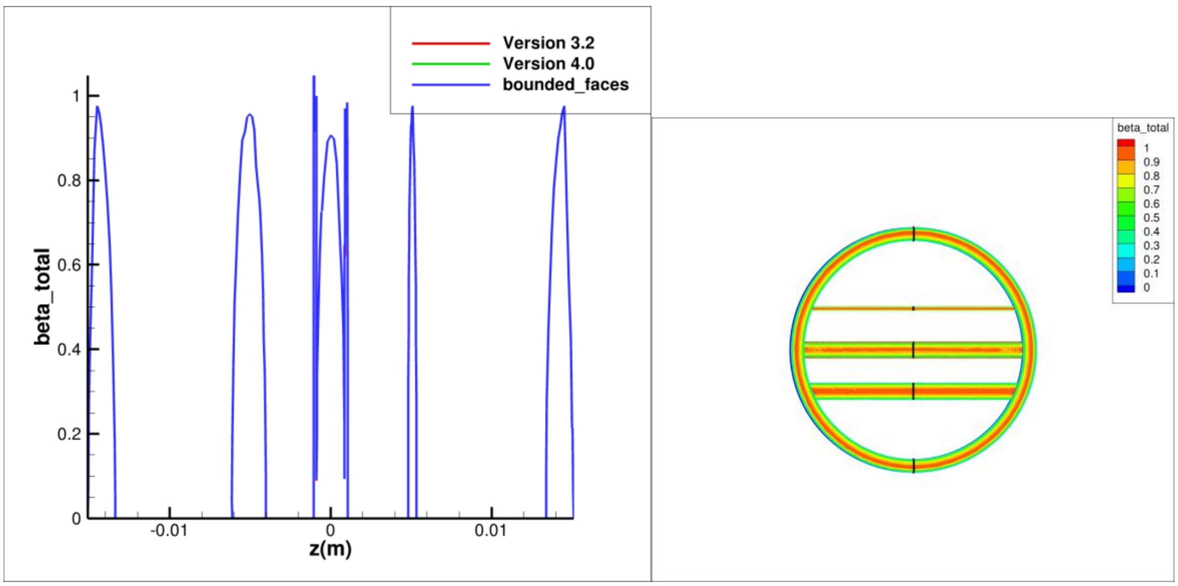


Figure 6 Collection Efficiency for Three Versions of GlennICE

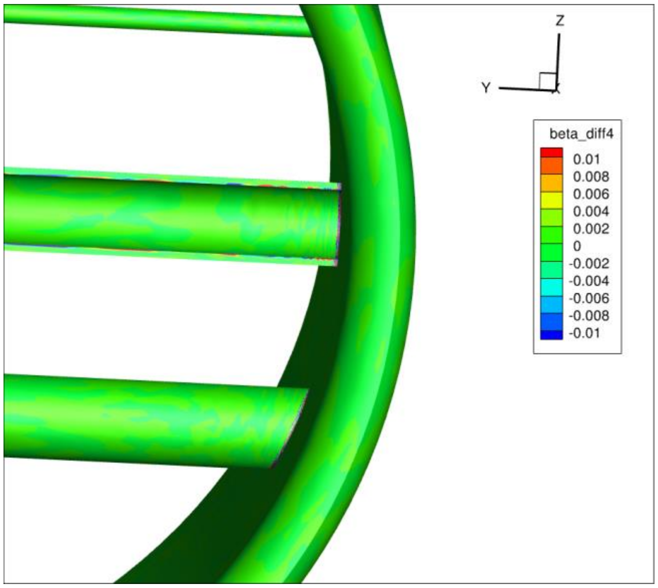


Figure 7 Difference in Collection Efficiency – Version 4 vs Version 3.2

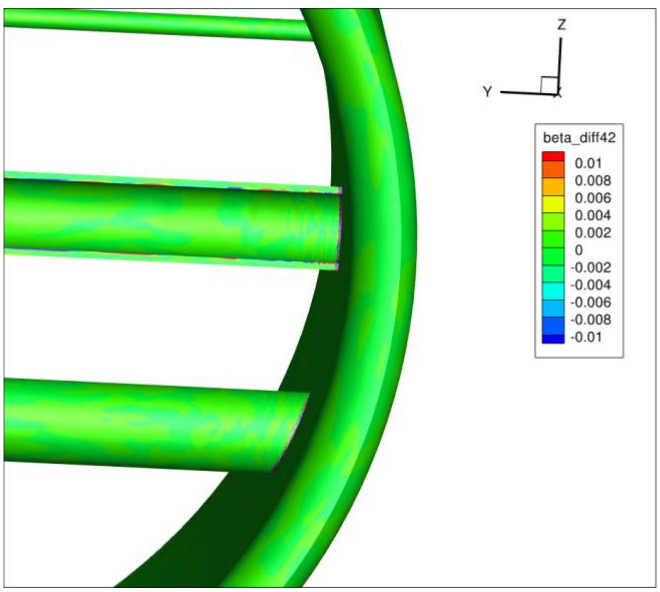


Figure 8 Difference in Collection Efficiency – Bounded_faces vs Version 3.2

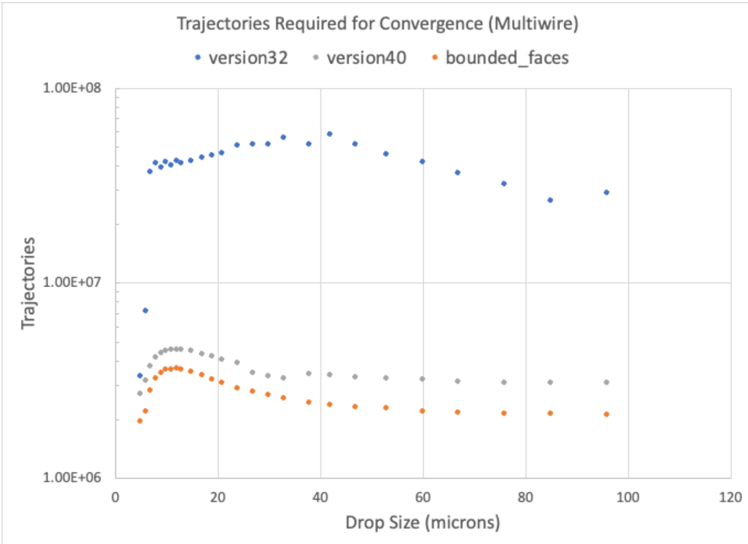


Figure 9 Required Trajectories for Convergence

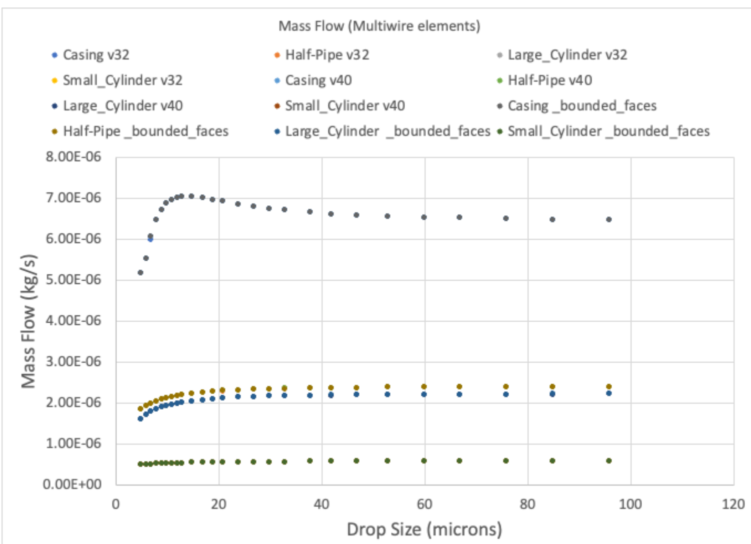


Figure 10 Mass Flow Impinging on Each Element for Different GlennICE Versions

The second case involved computing collection efficiency on a CRM-HL geometry used in high lift workshops [14]. The geometry is shown in Figure 11. This mesh is also a mixed element grid with 65,893,479 elements and 11,102,088 points. This flow solution was also produced with FUN3D version 13.7. The GlennICE analysis was performed once again using a monodispersed 20 µm drop size. For this case, computation was continued until GlennICE 3.2 reached convergence using default parameters. For GlennICE 3.2, this required 19 refinement iterations and 16,353,707 trajectories. For Glennice 4.0 the same case required 20 iterations but only 7,797,625 trajectories. For branch bounded_faces, 6,327,942 trajectories were required to reach the same convergence. Figure 12 shows collection efficiencies for the 21 µm case with the slice location. Figure 13 shows collection efficiency slices on the wing (left) and engine inlet (center and right). Once again, the new metrics did not visually affect collection efficiency results. Figure 14 shows the trajectories required for convergence for a range of drop sizes for this case. This plot does not show drop sizes less than 10 microns as those cases did not converge for any version of GlennICE. The reason for this is currently under investigation. For the same number of refinement iterations, similar efficiency improvements were seen for drop sizes smaller than 10 microns, however. On average, GlennICE 4.0 was

1.9 times faster than GlennICE 3.2 while branch bounded_faces was 2.5 times faster than GlennICE 3.2 for this geometry and the range of drop sizes studied. The computational savings for this geometry were not as high as the multiwire case as the CRM_HL geometry did not use as many trajectories for GlennICE 3.2 as the multiwire case did. As mentioned earlier, the multiwire case contained many faces with very small number of hits due to the unsteady nature of the flow solution. Figure 15 shows the total mass flow collected by aircraft for the drop size range shown in the previous figure. This plot demonstrates that the calculated mass flow is unaffected by reducing the number of trajectories calculated. This grid had a single surface, so collection on each section was not obtained.

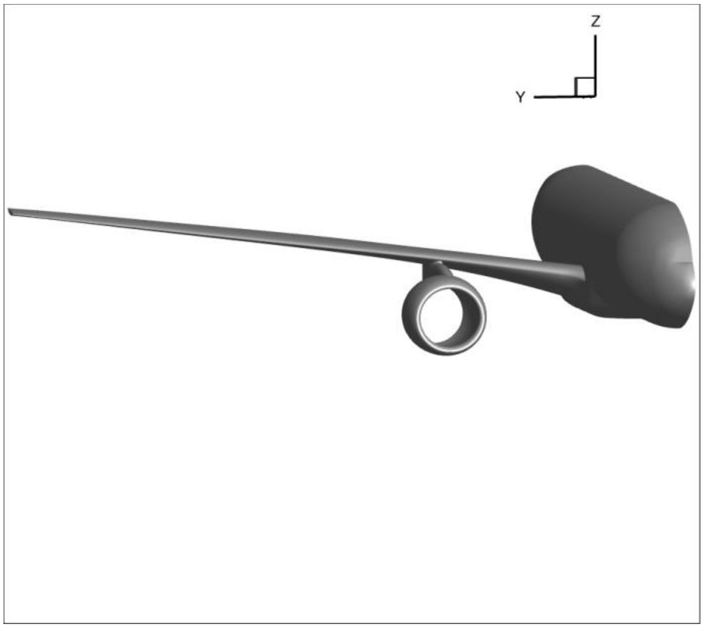


Figure 11 CRM-HL geometry

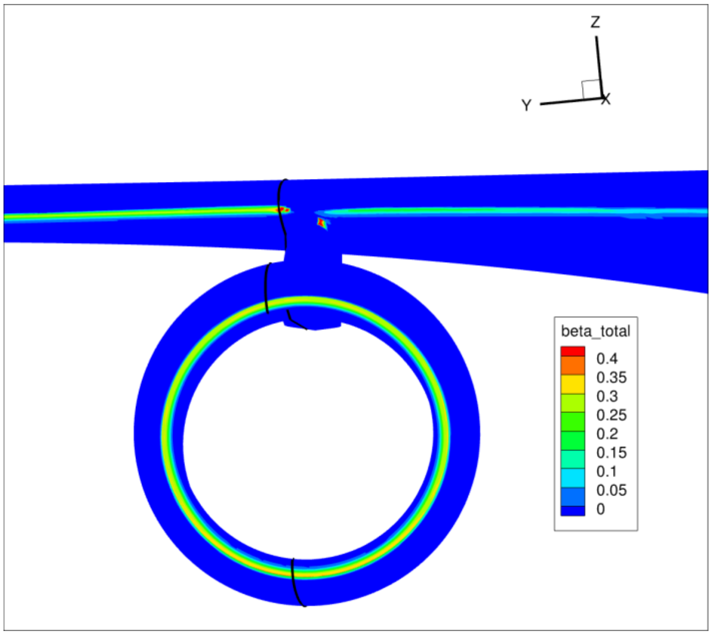


Figure 12 Slice Location and Collection Efficiencies for the 21 μm Drop Case

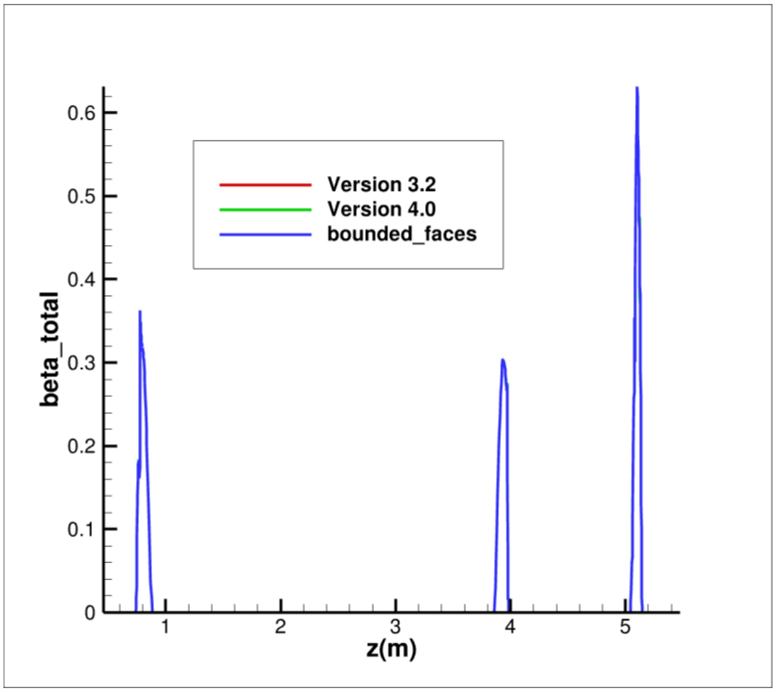


Figure 13 Collection Efficiency Results on CRM-HL Geometry

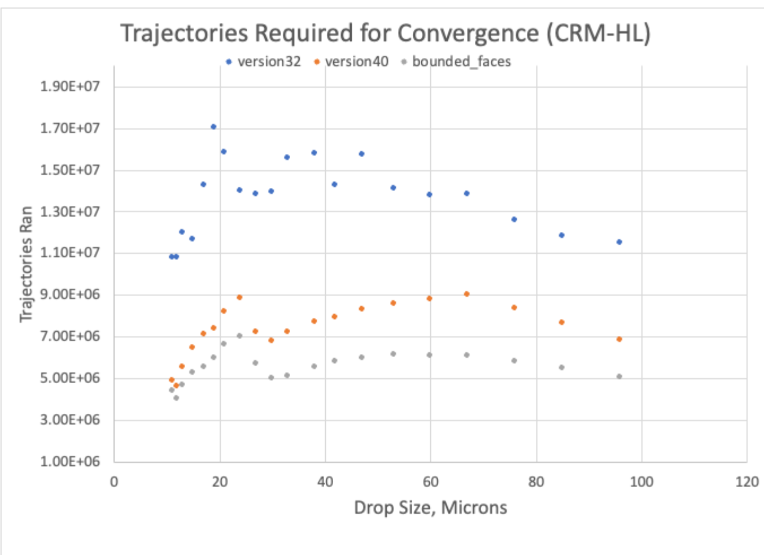


Figure 14 Trajectories Required for Convergence for CRM_HL

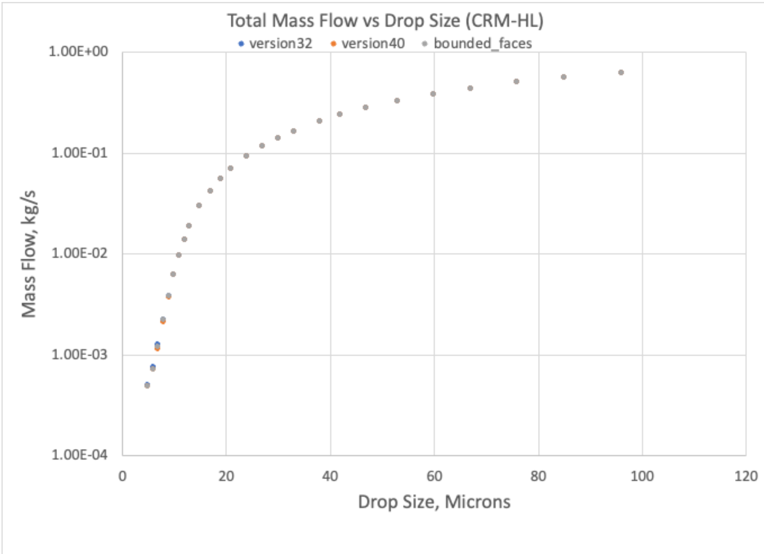


Figure 15 Mass Flow Collected on CRM-HL for Different Versions of GlennICE

V. Conclusion

This paper shows calculation of collection efficiencies for three different versions of GlennICE (3.2, 4.0 and test branch bounded_faces). The results show that computational efficiency was improved, sometimes dramatically, by reducing the number of trajectories necessary for convergence. The results also show that accuracy of the results was not noticeably affected by these changes. This result suggests that the user can take advantage of the new metrics to obtain results with fewer trajectories. However, the metrics are optional and the user can turn them off if it can be demonstrated that the results are adversely affected.

VI.Acknowledgements

The authors would like to acknowledge the support of the Advanced Air Transport Technology project for support of this research effort. The authors would also like to acknowledge the work of the rest of the Icing Branch, especially the GlennICE development team for their efforts in developing this software.

VII. References

[1] Wright, W. B. "Users Manual for LEWICE Version 3.2, NASA CR-2008-214255, Aug. 2008

[2] Bidwell, C. S., and Potapczuk, M. G., "Users Manual for the NASA Lewis Three-Dimensional Ice Accretion Code (LEWICE 3D), NASA TM-105974, Dec., 1993.

[3] Dormand, J. R., and P. J. Prince, "A Family of Embedded Runge-Kutta Formulae," J. of Computational and Applied Mathematics, Vol. 6, Issue 1, March 1980 pp. 19-26

[4] E. Hairer, S.P. Norsett and G. Wanner, Solving Ordinary Differential Equations i. Nonstiff Problems. 2nd edition. Springer Series in Computational Mathematics, Springer-Verlag (1993).

[5] https://www.mathworks.com/help/matlab/ref/ode45.html

[6] https://octave.org/doc/interpreter/Matlab_002dcompatible-solvers.html#Matlab_002dcompatible-solvers

[7] Wright, W., Porter, C., Potapczuk, M., Galloway, E. and Rigby, D., "An Automated Refinement Process for Trajectory Methods in GlennICE," AIAA Aviation Conference, June 2021.

[8] Rigby, David L. and Wright, William B., "Convergence Criteria for Lagrangian Collection Efficiency Simulations.", AIAA 2022-3693, June, 2022.

[9] Porter, Christopher E. "A Comparison of Trajectory Refinement Schemes for GlennICE.", AIAA 2022-3692, June, 2022.

[10] Chen, Ru-Ching and Porter, Christopher E., "Simulation of Fluid Flow and Collection Efficiency for a SEA Inc. Multi-Element Probe and Ice Crystal Detector Using GlennICE." ,AIAA 2022-3694, June, 2022.

[11] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jacobson, K. E., Jones,W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., Thompson, K. B., Walden, A. C., Wang, L., and Wood, W. A., "FUN3D Manual: 13.7," NASA Technical Memorandum 20205010139, 2020.

[12] https://github.com/delfrrr/delaunator-cpp

[13] https://github.com/artem-ogre/CDT

[14] https://hiliftpw.larc.nasa.gov/ind-workshop4.html