Three Dimensional Surface Redefinition Method for Computational Ice Accretion Solvers

Christopher E. Porter[1]
NASA Glenn Research Center, Cleveland, OH, 44135

and

David L. Rigby[2]
HX5 LLC, Brook Park, OH 44142

Computational tools have been increasing in maturity and are thus commonly used in the engineering design process. Advancements in NASA's current state of the art computational ice accretion tool are required to tackle the icing challenges of tomorrow. GlennICE, a next generation ice accretion solver, is under development at NASA to tackle these challenges. One of the elements of this migration is transitioning from a quasi-three dimensional strip theory based ice accretion methodology to a fully three dimensional methodology. This requires construction of a method to redefine or extrude a discretized or tessellated surface geometry based on the predicted volumetric ice growth for each tessellated surface triangle. This paper describes the methodology that is employed in the GlennICE software, and assesses the performance of the method in replicating a well-defined analytical test case.

I. Introduction

DUE to the increasing performance and decreasing cost of computational infrastructure, modeling tools have enabled the integration of many design considerations earlier in the design process. Icing is one such design consideration. Computational icing tools have been primarily applied to simulating ice accretion on a cantilever wing. For these configurations NASA's two dimensional and quasi-three dimensional tools, LEWICE[1] and LEWICE3D,[2] are well respected having been validated against a large database of ice shape geometries.[3,4]

Current and future icing challenges are more general in nature. These challenges including icing on probes and radomes, engine icing, as well as unconventional configurations like truss braced wings and urban air mobility vehicles. NASA's Advanced Air Transport Technology project has an interest in tackling these more general icing challenges to enable the next generation of vehicle and engine designs. To meet these needs a shift from two and quasi-three dimensional approaches to a more generalized fully three dimensional approach is required. One such generalized module required within this framework is a method to extrude the initial surface to a subsequent iced surface in three space.

Commercial tools such as ANSYS's FENSAP-ICE and SIEMENS have addressed this challenge using various methods. FENSAP-ICE utilizes an arbitrary lagrangian eulerian method to deform the surface mesh.[5] While SIEMENS utilizes their mesh morphing technology.[6] Tong, et al.[7] use elements from Jiao's face offsetting methodology,[8,9] in conjunction with a prismatoid extrusion method with various smoothing options to deduce the height field. A volume correction is applied to ensure volume conservation after the non-conservative smoothing algorithms. This is followed by a mesh moving technique to allow for a subsequent flow field computation to generate a loosely coupled ice shape - flow field interaction.

In this paper, a prismatoid method with a node based smoothing is the method favored here and the method implemented in GlennICE Version 0.1.1. This is a development version of the software and as such is considered pre-release. It should be noted that currently the end desire is not to integrate this functionality with a mesh moving capability. Instead, the desire is to build a methodology that generates a consistent ice shape that is insensitive to the level of refinement of the input computational fluid dynamics mesh. The scope of this paper is to detail the method as implemented in GlennICE Version 0.1.1 and investigate the error associated with this approach.

[1] Research Aerospace Engineer, Icing Branch, 21000 Brookpark Rd, MS 11-2.

[2] Aerospace Engineer V, 3000 Aerospace Parkway.

II. Methodology

The three dimensional surface redefinition process described in this document is a purely geometrical problem. A requirement of this algorithm is that the information needed to perform the three dimensional surface redefinition be inherently available in the mesh and solution file output of a typical computational fluid dynamics (CFD) solver. Note that this paper does not describe the process used to compute the ice volume, instead it describes how the surface is altered after the ice growth volume is known.

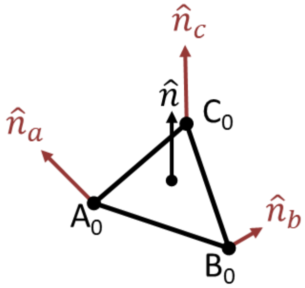


Figure 1: Illustration of the Variables Required to Compute the Surface Redefinition.

The geometrical parameters required for the method utilized in this paper are depicted in Fig. 1. They are: the coordinates of the three nodes that make up the triangular tessellated face, the face centered unit normal vector, and the three node centered unit normal vectors. The coordinates of the nodes are known and provided in the CFD solution file. From the coordinate data, the face centered unit normal vector is well defined, and it is simply the unit normal vector of the plane defined by the three nodal coordinates. This value is depicted mathematically by Eq. 1.

$$\hat{n} = \frac{\vec{e}_{A_0B_0} \times \vec{e}_{A_0C_0}}{\left\|\vec{e}_{A_0B_0} \times \vec{e}_{A_0C_0}\right\|} \tag{1}$$

where,

$$\vec{e}_{A_0B_0} = \vec{B}_0 - \vec{A}_0$$

$$\vec{e}_{A_0C_0} = \vec{C}_0 - \vec{A}_0$$

Note that the coordinates of the nodes, A_0, B_0, and C_0 are being mathematically treated as vectors from the origin, i.e. $\vec{A}_0$, $\vec{B}_0$, and $\vec{C}_0$. To add a bit of clarity, uppercase characters will be used to denote the mathematical vectors from the origin that physically represent coordinate locations. Lowercase characters will be used to denote relative vectors.

The remaining unknowns are the node centered unit normal vectors, depicted as the red vectors in Fig. 1. These vectors ideally should be equivalent to the normal vector at that point on the continuous surface that the discretized mesh approximates. However, these values are not well defined from the available data set and must be estimated.

A. Computation of the Node Centered Surface Normal Vector

There are many ways to estimate the value of the node centered unit normal vectors, and perhaps one of the most well-known and robust methods is the arithmetic averaging of the face centered unit normal vectors of all the triangular facets that touch the node.[10] This type of approach is typically favored over other methods due to its simplicity, computational efficiency, compactness, and robustness. There are other weighting options for the averaging besides an arithmetic weight. Cenanovic et al.[11] compare their method for estimating node centered normal vectors with a variety of the simpler weighted averaging processes. One process that performs quite well, and will be leveraged in this paper, is the angle weighted averaging introduced by Thürrner and Wüthrich.[12] The mathematical representation of this method can be seen in Eq. 2. It should be noted that the $\sum_{i=1}^{m} \theta_i$ terms cancel out, however, they were included here for clarity.

$$\hat{n}_{nc} = \frac{\left(\frac{\sum_{i=1}^{m} \theta_i * \hat{n}_{fc,i}}{\sum_{i=1}^{m} \theta_i}\right)}{\left\|\frac{\sum_{i=1}^{m} \theta_i * \hat{n}_{fc,i}}{\sum_{i=1}^{m} \theta_i}\right\|} \tag{2}$$

where m is the number of faces that contain the node of interest as a vertex.

For visual reference, a diagram can be seen in Fig. 2 that illustrates the face centered normal vectors, $\hat{n}_{fc}$, the node centered normal vector, $\hat{n}_{nc}$, and the interior angle used for weighting, θ.

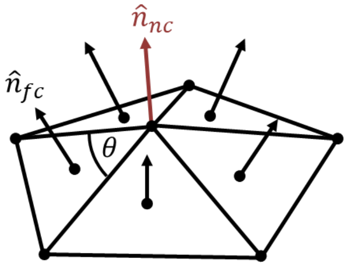


Figure 2: **Illustration of the Variables Used for the Estimation of the Node Centered Surface Normal Vector.**

B. Faceted Extrusion Process

The faceted extrusion process occurs in the direction of the face centered unit normal vector of the triangle of interest. This process makes use of Eq. 3, which describes the volume of the prismatoid the extrusion generates. In order to analytically integrate Eq. 3 there needs to be a formulation of the area of the triangle as a function of the extrusion height, h, which is the distance in the face centered normal direction from the initial surface triangle. Figure 3 provides a diagram of the extrusion of a triangle in a convex region of the surface.

$$Volume = \int Area\ \partial h \tag{3}$$

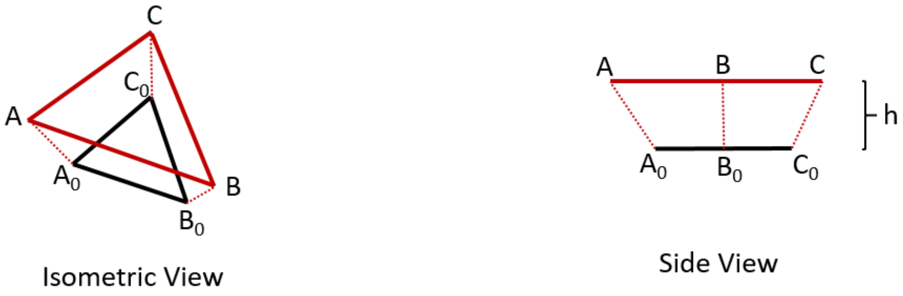


Figure 3: Diagram of the Extruded Triangle at an Extruded Distance of ‘h’.

Mathematically the area of the triangle is given by Eq. 4, and is half the magnitude of the vector resulting from the cross product of the two edge vectors, $\vec{e}_{AB}$ and $\vec{e}_{AC}$. The inclusion of the square root operation in the computation of the magnitude makes the analytical integration of Eq. 4 quite difficult. Numerical schemes would likely be required to computationally solve this form of the equation. This leads to robustness issues, and the identification of degenerate concave extrusions is not straight forward.

$$Area = \frac{1}{2}\|\vec{e}_{AB} \times \vec{e}_{AC}\| \tag{4}$$

The magnitude of a vector is simply the scalar length of the vector. This can be done using the classical magnitude computation as depicted in Eq. 4. However, the dot product of a vector with a parallel unit vector will also return the scalar length of that vector, albeit a signed scalar length. That is to say if the parallel unit vector is in the opposite direction of the vector of interest, the scalar returned will be negative. Taking the absolute value of this scalar will remove the sign, if any. Based on the problem definition, i.e. that the extrusion is in the direction of the face centered unit normal vector of the triangle of interest, one can notice that the face centered unit normal vector of $\Delta A_0 B_0 C_0$ is the same as the face centered unit normal vector of ΔABC. Because of this property Eqs. 4 and 5 are functionally equivalent. The usefulness of Eq. 5 lies in the fact that $\hat{n}$ is a known constant, and not a function of *h*.

$$Area = \frac{1}{2}\left|(\vec{e}_{AB} \times \vec{e}_{AC}) \cdot \hat{n}\right| \tag{5}$$

The absolute value operation makes the integration of Eq. 5 still analytically difficult. However, this absolute value is included solely to satisfy the property that negative area is physically extraneous. If one ignores this physical requirement and notices that negative area simply identifies a concave degenerate extrusion, then that mathematical property actually becomes quite useful. Equation 6 is the mathematical formulation that will be used here for the computation of area as a function of height. It is simply Eq. 5 without the absolute value operation.

$$Area = \frac{1}{2}(\vec{e}_{AB} \times \vec{e}_{AC}) \cdot \hat{n} \tag{6}$$

$\vec{A}$, $\vec{B}$, and $\vec{C}$ as a function of *h* are mathematically depicted in Eqs. 7-9. Again, note that the coordinates of the nodes, A, B, and C are being mathematically treated as vectors from the origin, i.e. $\vec{A}$, $\vec{B}$, and $\vec{C}$. For clarity, uppercase characters are used to denote the mathematical vectors from the origin that physically represent coordinate locations, and lowercase characters are used to denote relative vectors.

$$\vec{A} = \vec{A}_0 + \frac{h}{\hat{n} \cdot \hat{n}_a}\hat{n}_a \tag{7}$$

$$\vec{B} = \vec{B}_0 + \frac{h}{\hat{n} \cdot \hat{n}_b}\hat{n}_b \tag{8}$$

$$\vec{C} = \vec{C}_0 + \frac{h}{\hat{n} \cdot \hat{n}_c}\hat{n}_c \tag{9}$$

The edge vectors $\vec{e}_{AB}$ and $\vec{e}_{AC}$ are defined similarly to the edge vectors in Eq. 1, but make use of Eqs. 7-9 and are now also functions of *h*. Substituting $\vec{e}_{AB}$ and $\vec{e}_{AC}$ into Eq. 6 and simplifying yields Eq. 10.

$$Area = Eh^2 + Fh + G \tag{10}$$

where,

$$E = \frac{\left(\hat{n}_a \cdot (\hat{n}_b \times \hat{n}_c)\right)}{2 * (\hat{n} \cdot \hat{n}_a)(\hat{n} \cdot \hat{n}_b)(\hat{n} \cdot \hat{n}_c)}$$

$$F = \frac{\left(\vec{B}_0 \cdot (\hat{n} \times \hat{n}_a)\right) + \left(\vec{C}_0 \cdot (\hat{n}_a \times \hat{n})\right)}{2 * (\hat{n} \cdot \hat{n}_a)} + \frac{\left(\vec{A}_0 \cdot (\hat{n}_b \times \hat{n})\right) + \left(\vec{C}_0 \cdot (\hat{n} \times \hat{n}_b)\right)}{2 * (\hat{n} \cdot \hat{n}_b)} + \frac{\left(\vec{A}_0 \cdot (\hat{n} \times \hat{n}_c)\right) + \left(\vec{B}_0 \cdot (\hat{n}_c \times \hat{n})\right)}{2 * (\hat{n} \cdot \hat{n}_c)}$$

$$G = \frac{\left(\hat{n} \cdot (\vec{A}_0 \times \vec{B}_0)\right) + \left(\hat{n} \cdot (\vec{B}_0 \times \vec{C}_0)\right) + \left(\hat{n} \cdot (\vec{C}_0 \times \vec{A}_0)\right)}{2}$$

Note that *E*, *F*, and *G* are all functions of the parameters illustrated in Fig. 1, and are all known quantities. With respect to Eq. 10, at *h*=0, the area will always be positive as by definition the clockness of $\Delta A_0 B_0 C_0$ is such that the cross product, $\vec{e}_{A_0B_0} \times \vec{e}_{A_0C_0}$, is in the same direction as $\hat{n}$. Negative values of *h* are extraneous solutions and will be ignored here.

If one exists, the smallest, positive, real root of Eq. 10 denotes the location where $\vec{e}_{AB} \times \vec{e}_{AC}$ goes from pointing in the direction of $\hat{n}$, to pointing in the opposite direction of $\hat{n}$. But a positive, real root of Eq. 10 may not exist. For instance, consider the case of a tessellated planar surface. The node centered unit normal vectors point in the same direction as the face centered unit normal vector, i.e. $\hat{n} = \hat{n}_a = \hat{n}_b = \hat{n}_c$. Examining the effect of this on Eq. 10, one can see that this leads to E and F being equal to zero, and thus the area is constant as a function of height, which is the result one should expect. Thus Eq. 10 states that the order of the area equation is at maximum quadratic, but linear and constant formulations are perfectly valid if E or E and F are respectively zero.

Considering the case where at least one positive, real root exists, the extrusion height h must be less than the smallest, positive, real root of Eq. 10 for the extrusion to be physically meaningful. Values of h larger than this root denote a degenerate extrusion. Figure 4 illustrates how area varies as a function of h for a second order degenerate extrusion. All values of h larger than the first zero crossing are values where the extrusion is degenerate and therefore the solution is extraneous.

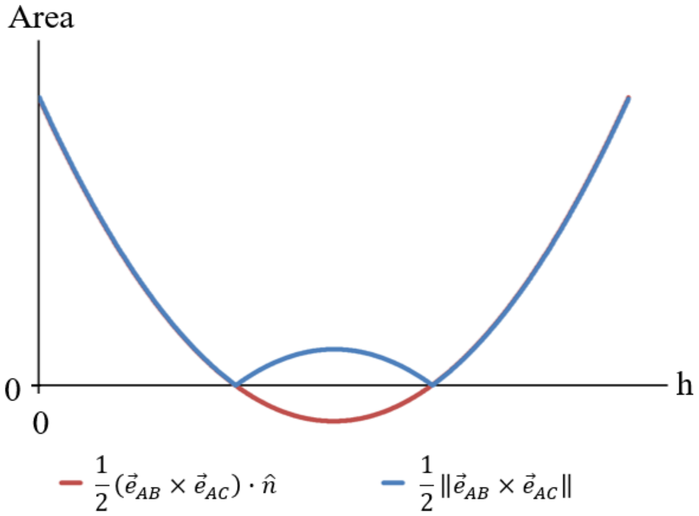


Figure 4: Area as a Function of Height for a Second Order Degenerate Extrusion

Substituting Eq. 10 into Eq. 3, integrating, and noting that the constant of integration must be equal to zero to satisfy the boundary condition that the volume is zero at an extrusion height of zero yields Eq. 11.

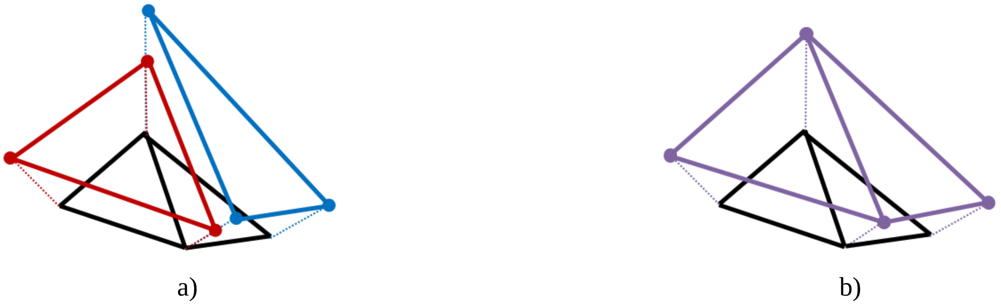
$$Volume = \frac{E}{3}h^3 + \frac{F}{2}h^2 + Gh \tag{11}$$

where E, F, and G are the same as those defined for Eq. 10.

One can now substitute in the desired ice volume and solve for the desired extrusion height, h. The correct value of h corresponds to the smallest, positive, real root of Eq. 11 with the substituted value of the ice volume. If the smallest, positive, real root of Eq. 11 is less than the smallest, positive, real root of Eq. 10, then a valid solution is found. If the smallest, positive, real root of Eq. 11 is larger than the smallest, positive, real root of Eq. 10, or if there is no positive, real root of Eq. 11, then the extrusion is degenerate. Currently there is no algorithmic process to handle degenerate extrusions, but it is promising that they are easily identifiable.

C. Node Based Smoothing

Due to the locality of the extrusion process, there will be occurrences where the mesh is not water tight, i.e. stair-stepping will occur between neighboring extrusions. This can be seen in Fig. 5a. A water tightness correction in the form of a node based smoothing operation is applied to the extruded surface faces. Noting that all the extruded nodes at a given nodal location lie in the direction of the node centered unit normal vector, the actual nodal extrusion will be the average distance of all the nodal extrusions of the given faces that touch that particular node. This average value also lies in the direction of the node centered unit normal vector. The resulting surface map can be seen in Fig. 5b. This node based smoothing can affect the volume of the extrusion making the process non-conservative with respect to the integrated value of ice volume. Since the desired volume and actual extruded volume are known quantities, the error is known, and can potentially be corrected. Currently the algorithm as implemented in GlennICE does not correct for this error.



a)

b)

Figure 5: Depiction of the Surface Extrusions a) Before the Node Based Smoothing and b) After the Node Based Smoothing

D. Analytical Test Case

A desirable test case is one that is sufficiently complex to flex the algorithm, but well understood enough to enable quantification of the numerical error of the algorithms being used to approximate the analytical solution. The test case presented here aims to meet those needs, and is defined by three properties:

1) the geometry is a unit sphere
2) the particles are purely ballistic (arbitrarily traveling in the positive x direction)
3) the leading edge ice thickness is 10% of the radius

By assuming a purely ballistic particle, one allows for the collection efficiency, β, to vary on the sphere. Specifically, β on the impinged hemisphere is equal to the dot product of the negative of the surface unit normal vector with <1,0,0>, the unit vector of the flow direction. The total volume of ice grown is given by the equation:

$$Volume\ of\ Ice = \bar{\beta} A \gamma \tag{12}$$

where $\bar{\beta}$ is the average value of collection efficiency on the the impinged hemisphere, A is the surface area of the impinged hemisphere and γ is a coefficient that captures the effects of the icing related parameters: liquid water content, free stream velocity, ice density, and time. The coefficient γ has a dimension of length.

The ice shape can be computed by using Eq. 13. Note that the polar angle is being measured from the x-axis for mathematical convenience. The left hand side of the equation is the desired volume of ice as computed by Eq. 12 over a small circumferential strip, $\delta\theta$. Note that the local value of β has been replaced by $-cos(\theta)$, which is equivalent to the dot product described above. The right hand side of the equation is the general form for computing the volume between two concentric spheres over a small circumferential strip, $\delta\theta$. Note that this equation can be posed for a small discrete element for generality, where the limits of integration on phi are from ϕ to $\phi + \delta\phi$. However, the solution doesn't vary azimuthally, and therefore the integration is performed from 0 to 2π, i.e. along a small circumferential strip.

$$\left[\frac{\int_{\theta}^{\theta+\delta\theta}\int_{0}^{2\pi} -\cos(\theta) {r_o}^2 \sin(\theta) \delta\phi\delta\theta}{\int_{\theta}^{\theta+\delta\theta}\int_{0}^{2\pi} {r_o}^2 \sin(\theta) \delta\phi\delta\theta}\right]\left[\int_{\theta}^{\theta+\delta\theta}\int_{0}^{2\pi} {r_o}^2 \sin(\theta)\ \delta\phi\ \delta\theta\right]\ [\gamma] = \int_{\theta}^{\theta+\delta\theta}\int_{0}^{2\pi}\int_{r_o}^{r_{ice}} r^2\ \sin(\theta)\ \delta r\ \delta\phi\delta\theta \tag{13}$$

Integrating Eq. 13, simplifying, taking the limit as $\delta\theta \rightarrow 0$, and solving for r_{ice} yields Eq. 14.

$$r_{ice} = [-3\cos(\theta)\,\gamma\,{r_o}^2 + {r_o}^3]^{\frac{1}{3}} \tag{14}$$

Note that one of the definitions of the analytical test case was that the leading edge ice growth is 10% of the radius of the sphere. Substituting in values of $r_{ice} = 1.1$ and $\theta = \pi$ into Eq. 14 and solving for γ yields $\gamma = 0.110\bar{3}$. Thus the analytical definition of the ice shape is given by Eq. 15 noting that r_o is unity.

$$r_{ice} = [-0.331\cos(\theta) + 1]^{\frac{1}{3}} \tag{15}$$

Now that γ is known, one can now calculate the total volume of ice on the impinged hemisphere using Eq. 12. This gives $Volume\ of\ Ice = \frac{1}{2}\ (2\pi 1^2)\ 0.110\bar{3}$.

E. Discretized Sphere

In order to identify the accuracy of the method when applied to a discretized geometry, one must discretize the geometry. The discretized spheres were generated using Pointwise,[13] a commercial mesh generator. The topology of the mesh can be seen in Fig. 6. The thick black lines denote the four longitudinal connectors and eight latitudinal connectors that define the level of refinement of the discretized sphere. Changing the number of nodes on these connectors changes the number of triangles that make up the surface mesh.

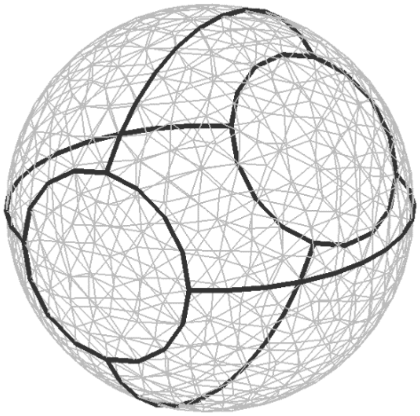


Figure 6: Mesh Topology of the Discretized Sphere. The Pictured Surface Mesh Corresponds to Refinement Level 1.

For this particular investigation, four levels of refinement were chosen. Table 1 documents the number of nodes on the longitudinal and latitudinal connectors, as well as the resulting number of surface faces on the sphere. The first level of refinement was the default level of refinement from Pointwise's intrinsic utility that generates simple geometric shapes, in this case a sphere. Each subsequent level of refinement doubles the number of nodes on both the longitudinal and latitudinal dimensions. The surface mesh on the impingement side of the sphere for the four levels of mesh refinement can be seen in Figure 7.

Table 1. Connector Dimensions and Number of Surface Faces for the Four Refinement Levels.

Refinement Level	Longitudinal Dimension	Latitudinal Dimension	Number of Faces
1	5	12	1152
2	10	24	5434
3	20	48	22998
4	40	96	94190

For the discretized case, the face centered value of β can be specified in two distinct ways. One way would be to assign the average value of β over the surface face as defined from the analytical case. The second way, and the method favored here, is to assign the value of β that GlennICE would predict. Thus β on each face is defined as the dot product of the surface normal of that discretized face with <1,0,0>, the assumed flow direction. This preference is done to maintain a connection to both the analytical problem, as well as the discretized version of the analytical problem that GlennICE simulates. It should be noted that this preference does not limit identifying the error due to the node based smoothing described above, as the error due to the smoothing and the error due to the discretization can be examined independently.

Figure 7 illustrates the variation of β on the surface of the discretized spheres. These contours were created with the post processing tool Tecplot.[14] Although the contours illustrate a banded variation across the surface face, this is purely a visualization artifact, β is constant within each surface face. These contours illustrate the value of β is highest at the leading edge and decays to zero at the bounds of the hemisphere.

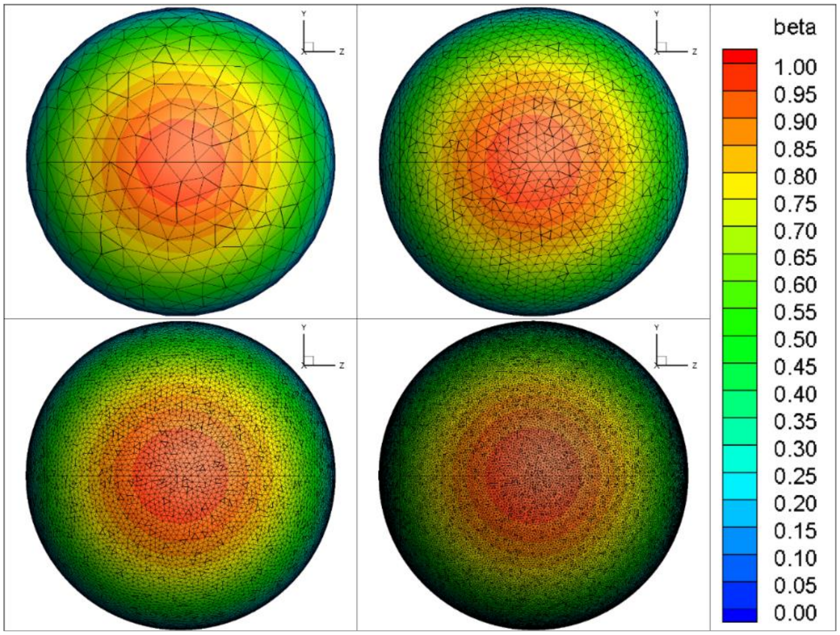


Figure 7: Surface Meshes and β Contours for the Four Levels of Mesh Refinement.

III. Results

Figure 8 illustrates the iced surface geometry extracted from the XY plane for the impinged side of the sphere. This impinged side of the sphere is the hemisphere that is visible in Fig. 7 For the non-analytical cases the data extraction was performed with the post processor Tecplot on the generated stereolithography, STL, file.

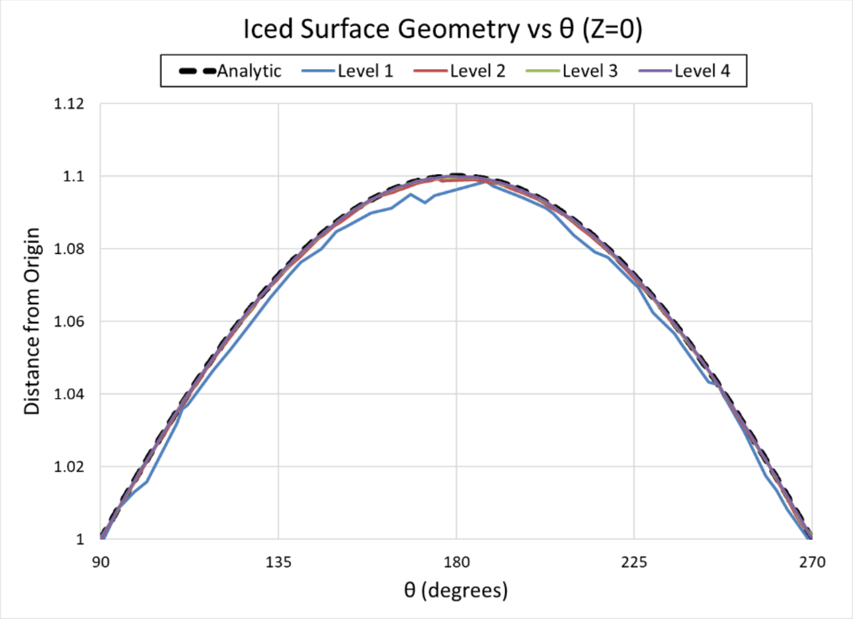


Figure 8: Iced Surface Geometry Versus θ on the XY-Plane.

The data is presented in an r – θ coordinate system. This allowed for discrepancies between the predicted and analytical results to be more visibly apparent. Note that above θ was defined as the polar angle, and therefore has a mathematical range of 0 to 180. However the data being presented in Fig. 8 has a range of 90 degrees to 270 degrees. This is done for convenience, and denotes the fact that two azimuthal angles that are 180 degrees out of phase, are being presented on the same plot. From visual inspection, the Level 1 refinement is insufficient to qualitatively capture the ice formation. The Level 1 refinement is quite coarse, and therefore is not completely unexpected. The Level 2-4 refinements compare quite well qualitatively to the analytical solution defined by Eq. 15, suggesting that a mesh refinement that is sufficient for a computational fluid dynamics (CFD) analysis should generate a qualitatively sufficient ice shape using the method described in this paper.

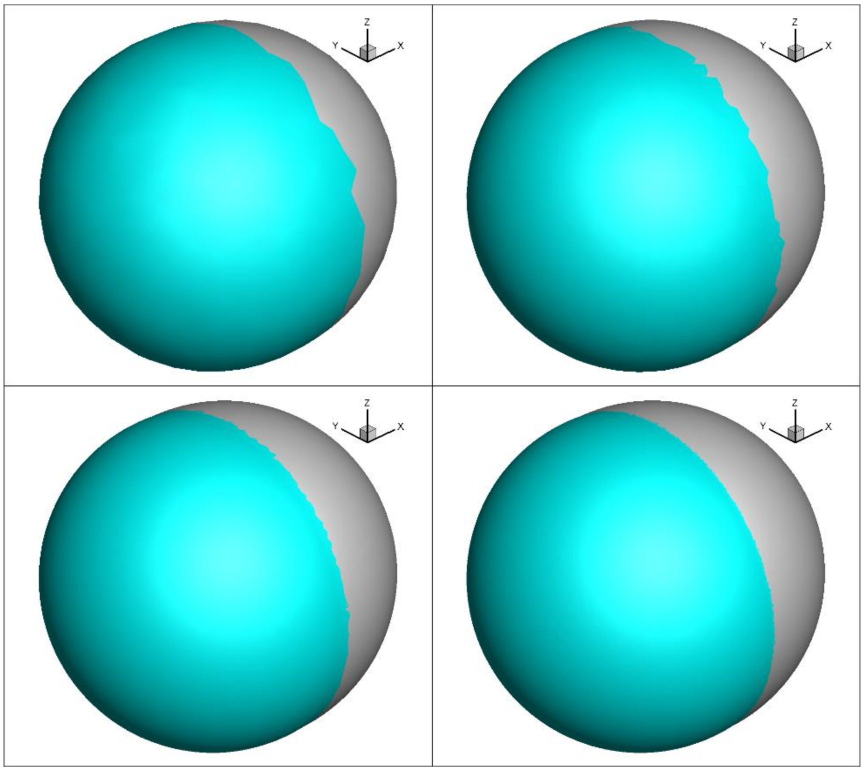


Figure 9: Revised Iced Geometry for the Four Levels of Mesh Refinement

Figure 9 depicts the three dimensional surface for the four levels of mesh refinement. The light blue indicates the surface of the ice shape, i.e. triangular facets where at least one node was altered. The grey denotes the clean surface, i.e. triangular facets where no nodes were altered. The ice shape refinement levels are depicted in the same order as those shown in Fig. 7. Observing Fig. 9, one can infer an increase in ice shape fidelity due to the increased smoothness in the lighting on the geometric surface. The increase in sharpness at the impingement limit is another indication of increased ice shape fidelity. While the impingement limit for the second refinement level is not very sharp, Fig. 8 shows that this refinement level still compares quite favorably qualitatively to the analytical solution.

Table 2. Ice Volumes of Various Definitions for the Four Refinement Levels.

Refinement Level	Analytical	Predicted	Discretized
1	0.34662239	0.34414313	0.34459796
2	0.34662239	0.34602108	0.34617904
3	0.34662239	0.34643054	0.34652116
4	0.34662239	0.34652759	0.34659723

It is quite common place in icing literature to judge simulation results on their ability to qualitatively represent an ice shape. Due to the variability inherent in the ice accretion process, this type of validation is not without merit. However, the well-defined analytical test case described here provides the ability to numerically compare the methods being used to simulate the ice accretion process. Table 2 depicts various definitions of the iced volume for the four refinement levels investigated here.

The 'analytical' volume is the volume of ice that would grow analytically. It does not change with mesh refinement as it is not related to the mesh, but the analytical case. It is defined above in the section describing the test case.

The ‘predicted’ volume is the ice volume generated from the surface redefinition method described in this document. While the volume of each of the polyhedral extrusions can be calculated and summed, this functionality is not yet currently implemented in GlennICE and therefore the post processing in this paper relied on third party computer-aided design (CAD) software. Thus ice volume for each refinement level was computed as the difference of the volume of the discretized iced geometry with the volume of the discretized clean geometry. These volumes were computed using the ANSYS SpaceClaim[15] CAD software.

The ‘discretized’ volume is the volume that would be generated from the impingement on the discretized geometry. This is not equal to the analytical volume as β is not smoothly varying, it is constant across each individual triangular surface face. Also, due to the tessellation, the surface area of the impinged hemisphere is not exactly 2π.

Figure 10 depicts three sets of data that describe volumetric error as a function of surface mesh refinement. As the analytic solution is the desired solution, one may view the error between the predicted volume and the analytical volume as the error associated with the surface redefinition method presented here, depicted by the line with blue dots in Fig. 10.

However, there are two sources of error that contribute to the volume error between the predicted and analytical solutions. The first source of error is the error due to the mesh discretization. This error is depicted by the line with green dots in Fig. 10. If the surface redefinition method presented here was perfectly accurate and had no smoothing error, the predicted ice shape would still not be equal to the analytical solution due to the error in water impingement that arises due to the surface discretization.

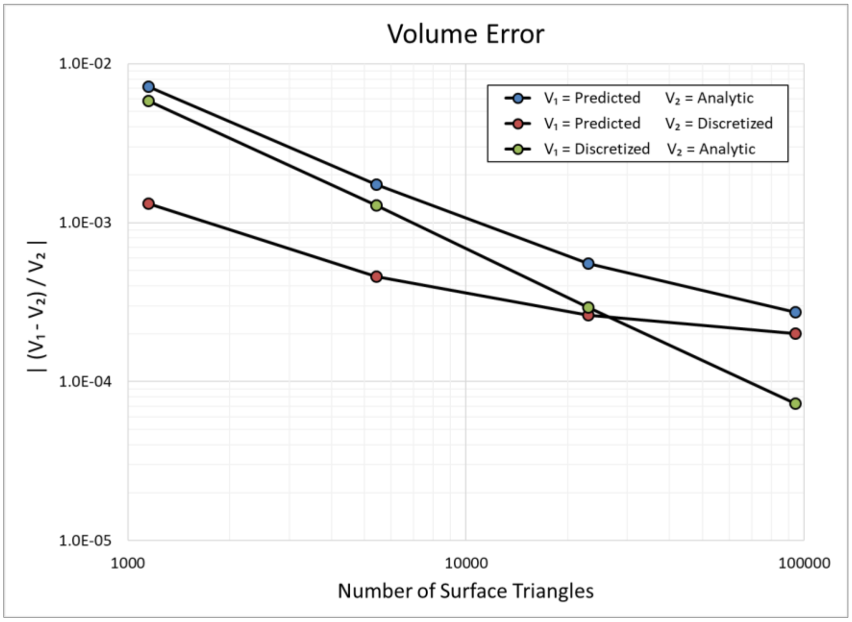


Figure 10: Volume Error Versus Surface Mesh Refinement. The Symbol V Denotes the Volume as Defined in the Legend

The second source of error is the error due to the smoothing operation of the algorithm described in this document which ensures the mesh is watertight. This error is depicted by the line with red dots in Fig. 10. This error is significantly smaller than the inherent discretization error for the coarser surface meshes. However, it does not decay at a similar rate with increases in surface mesh refinement, and becomes the dominant source of error for the most refined mesh. The fact that the smoothing error dominates as the surface mesh is refined means that a correction should seriously be considered to make the method conservative.

IV. Discussion

This paper presents the algorithm currently implemented in GlennICE to redefine a discretized surface mesh. This method extrudes a surface triangle in the normal direction to the triangular face generating a prismatoid. This method captures the surface curvature effects on the extruded volume and can identify degenerate extrusions. A node based smoothing process is employed to remove stair stepping and generate a watertight surface mesh. This node based smoothing is not guaranteed to conserve volume, and as such can introduce an error.

A well-defined analytical test case was described that was used here for error analysis. This test cases was defined by a sphere with a unit radius being impinged by purely ballistic particles with an operating condition that leads to a leading edge ice growth that is 10% of the sphere's radius. Pointwise was used to generate a discretized version of this sphere for four levels of surface mesh refinement.

Classically, ice accretion solvers have judged their accuracy using a qualitative comparison of the geometric ice shapes. Using a qualitative based approach the coarsest refinement level did a poor job replicating the analytical ice growth. However, all of the other refinement levels compared well from a qualitative perspective.

Using a more rigorous numerical based validation method, it was illustrated that increasing the level of surface mesh refinement decreased the numerical error. The error between the predicted ice shape and the analytical solution was generated by two sources. The first source being the discretization error that manifests itself in β. The value of β for the discretized cases was defined to be the value of β that one would expect from a numerical solver such as GlennICE. This illustrates that even if the surface redefinition method described here were perfectly accurate, it would not capture the analytical solution due to the error in β.

The second source of error is due to the smoothing process that produces the watertight iced surface mesh. This error is much less than the discretization error at the coarser mesh refinements. While this error does reduce as refinement increases, the improvement isn't linearly proportional to the surface mesh refinement. As such this error term becomes dominant at the higher refinement levels.

It is noted that both the desired volume at each face is known, as well as the extruded volume after the smoothing process. As such the error in the ice shape is known locally and can be corrected by various methods to ensure the volume in the extrusion process is conserved. However, these methods are not yet included in GlennICE and are therefore outside the scope of this paper and not discussed here.

Acknowledgments

The authors would like to thank NASA's Advanced Air Transport Technology project for its financial support of this research effort, as well as the larger GlennICE development group (Eric Galloway, William Wright, and Mark Potpaczuk) for their efforts that aided in the success of this paper.

References

[1] Wright, William B. "User Manual for the NASA Glenn Ice Accretion Code LEWICE. Version 2.2. 2." (2002).

[2] Bidwell, Colin S. "User's Manual for the NASA Glenn Three-Dimensional Grid Based Ice Accretion Code (LEWI3DGR Ver. 1.7)." (2005).

[3] Wright, William. "A summary of validation results for LEWICE 2.0." 37th Aerospace Sciences Meeting and Exhibit. 1998.

[4] Wright, William. "Validation results for LEWICE 3.0." 43rd AIAA Aerospace Sciences Meeting and Exhibit. 2005

[5] Morency, Francois "FENSAP-ICE-A comprehensive 3D simulation system for in-flight icing." *15th AIAA Computational Fluid Dynamics Conference*. 2001.

[6] Snyder, D. "Streamlining Aircraft Icing Simulations." *STAR Global Conference*. Vienna, Austria (2014)

[7] Tong, Xialoing "Three-dimensional surface evolution and mesh deformation for aircraft icing applications." *Journal of Aircraft* 54.3 (2017): 1047-1063.

[8] Jiao, J., "Volume and Feature Preservation in Surface Mesh Optimization," Proceedings of the 15th International Meshing Roundtable, Springer–Verlag, Berlin, 2006, pp. 359–374.

[9] Jiao, J., "Face Offsetting: A Unified Approach for Explicit Moving Interfaces," Journal of Computational Physics, Vol. 220, No. 2, 2007, pp. 612–625.

[10] Gouraud, Henri. "Continuous shading of curved surfaces." IEEE transactions on computers 100.6 (1971): 623-629.

[11] Cenanovic, Mirza, Peter Hansbo, and Mats G. Larson. "Finite element procedures for computing normals and mean curvature on triangulated surfaces and their use for mesh refinement." *eprint arXiv:1703.05745* (2017).

[12] Thürrner, Grit, and Charles A. Wüthrich. "Computing vertex normals from polygonal facets." Journal of Graphics Tools 3.1 (1998): 43-46.

[13] Pointwise, Software Package, V18.2R2, Pointwise Inc., Fort Worth, TX, 2019.

[14] Tecplot 360, Software Package, EX 2018 R1, Tecplot Inc., Bellevue, WA,2018.

[15] Ansys SpaceClaim, Software Package, R19.1, Ansys Inc., Canonsburg, PA, 2018.