IN-03
55467
P-97

NASA Contractor Report 198355

Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6

William B. Wright
NYMA, Inc.
Brook Park, Ohio

June 1995

Prepared for
Lewis Research Center
Under Contract NAS3–27186

National Aeronautics and
Space Administration

N95-29132
Unclas
G3/03 0055467
(NASA-CR-198355) USERS MANUAL FOR THE IMPROVED NASA LEWIS ICE ACCRETION CODE LEWICE 1.6 Final Report (NYMA) 97 p

https://ntrs.nasa.gov/search.jsp?R=19950022711 2020-06-16T07:46:37+00:00Z

****
* By participating in the generation of the computer codes or other test data reported *
* herein, NASA makes no assurances that said code or other data is complete, *
* accurate, or sufficient for purposes of assuring safety of operation of aircraft. *
****

1.0 Acknowledgments 3
2.0 Abstract 3
3.0 Nomenclature 4
4.0 Summary 5
5.0 Background 6
6.0 Introduction 7
7.0 Description of the LEWICE 1.6 Model 9
7.1 Calculation of the Flow Field 9
7.2 Boundary Layer Calculation 9
7.2.1 Laminar Boundary Layer 10
7.2.2 Turbulent Boundary Layer 10
7.2.3 Integration Methodology 11
7.3 Droplet Trajectory Calculation 12
7.3.1 Equations of Particle Motion 12
7.3.2 Determination of Droplet Impingement 14
7.3.3 Calculation of the Local Collection Efficiency 15
7.3.4 Local Collection Efficiency Calculation for Multidispersed Particle Distributions 15
7.4 Discussion of the Ice Accretion Process 16
7.4.1 Ice Accretion Characteristics 16
7.4.2 Description of the Physical Model 16
7.4.3 Calculation of the Thermodynamic Characteristics 17
7.4.4 Solution of the Energy Equation 18
7.5 Calculation of the Iced Geometry 18
7.5.1 General Computational Procedure 19
7.6 Energy Balance Derivation 19
7.6.1 Derivation of Mass Balance for LEWICE 1.6 22
7.6.2 Water Shedding 23
7.6.3 Bead Height Calculation 23
7.6.4 Hot Air Anti-Ice 23
7.6.5 Anti-Ice with Internal Heat Source 24
7.6.6 Evaporative Anti-Icer 25
8.0 Input Files 26
8.1 Line 1 26
8.2 LEW16 Namelist 26
8.2.1 Example LEW16 Namelist 27
8.3 S24Y1 Namelist 28
8.3.1 Example S24Y Namelist 28
8.4 Body Geometry Input 28
8.4.1 NREAD=0 Example 28
8.4.2 NREAD=1 Example 31
8.5 TRAJ1 Namelist 32
8.5.1 Example TRAJ1 Namelist 32
8.6 DIST Namelist 33
8.6.1 Example DIST Namelist 33
8.7 ICE1 Namelist 33
8.8 Example ICE1 Namelist 33
8.9 LPRNT Namelist 33
8.9.1 Example LPRNT Namelist 34
8.10 Complete Example Case Input File 34
8.11 Deice Input File (Unit 19) 35

8.11.1Example Deicer Input (Unit 19) 35
8.12 Interactive Input 35
9.0 Output Files 37
10.0 Suggested Procedure for Using Different Modules 39
11.0 Using LEWICE 1.6 on a PC 41
12.0 User Tips 42
13.0 Numerical Improvements 45
13.1 LEWICE 1.0 - LEWICE 1.3 45
13.2 LEWICE 1.3 - LEWICE 1.6 54
13.2.1 Roughness Prediction 57
13.2.2 Multi-Element Ice Accretion 60
14.0 Example Cases 64
14.1 Example 1: Single Body Case 64
14.2 Example 2: Single Body with Multiple Drop Sizes 67
14.3 Example 3: Multi-Body Simulation 69
14.4 Example 4: Multi-body Ice Accretion Using Langmuir 'D' Drop Size Distribution 80
14.5 Example 5: Use of Deice/Anti-Ice Capability 81
14.6 Example 6: Large Drop Example 83
14.7 Example 7: Parameter Study 85
15.0 Programmer's Guide to LEWICE 1.6 Subroutines 89
16.0 Conclusions 93
17.0 References 94

1.0 Acknowledgments

LEWICE 1.6 is the result of several years worth of research, both in the development of theory and in testing those theories in the NASA Lewis Icing Research Tunnel. Improvement of this code has been a top priority in the Icing Technology Branch as industry seeks better tools for simulating flight into icing conditions. Recognition is given to the University of Illinois at Champagne, the Massachusetts Institute of Technology and the University of Toledo, who through grants sponsored by the Icing Branch provided special insight into icing physics. Specific recognition is given to Dr. Mike Bragg of Illinois, Dr. R. John Hansman and Dr. Ken Breyer of MIT and Dr. Cyril Masiulaniec, Dr. Ken DeWitt and Dr. Ted Keith of Toledo for their valuable research.

Special recognition goes to the members of the NASA Lewis Icing Branch, who have maintained funding for this development and provided unparalleled expertise in icing, both computational and experimental. This ongoing project is an exemplary example of teamwork in solving difficult problems. My thanks to:

Dave Anderson
Colin Bidwell
Randy Britton (NYMA, Inc.)
Tom Bond
HaeOk Lee
Dean Miller
Mark Potapczuk
Tom Ratvasky
Andy Reehorst
Jack Reinmann
Jaiwon Shin
Mario Vargas

2.0 Abstract

This report is intended as an update/replacement to NASA CR 185129 "User's Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)" (1) and as an update to NASA CR 195387 "Update to the NASA Lewis Ice Accretion Code LEWICE" (2). In addition to describing the changes specifically made for this version, information from previous manuals will be duplicated so that the user will not need three manuals to use this code.

3.0 Nomenclature

b = bead height/roughness (m)
F = wetness fraction
V = velocity (m/s)
s = surface distance (m)
r = water density
s = surface tension (kg/m/s)
t = time (sec)
c=chord (m)
ρ=density (kg/m^3)
V=velocity (m/sec)
α=angle of attack (degrees)
β=collection efficiency (dimensionless)
LWC = liquid water content (kg/m^3)
m = mass flux (kg/m^{2}s)
Q = heat flux (W/m^2)
L_v = heat of vaporization (Ws/kg)
L_f = latent heat of freezing (Ws/kg)
h = heat transfer coefficient (W/m^{2}K)
h_m = mass transfer coefficient (m/s)
c_p = heat capacity (Ws/kgK)
C = mass concentration (kg/m^3)
L = Lewis number (dimensionless)
MW = molecular weight (kg/kg-mol)
R = ideal gas constant =8314 kgm^2/(kg-mol s^{2}K)
Sc = Schmidt number (dimensionless)
Pr = Prandtl number (dimensionless)
P_v = vapor pressure (N/m^2)
r_h = relative humidity (dimensionless)
P = pressure (N/m^2)
γ = ratio of heat capacities c_p/c_v (dimensionless)
D_{AB} = diffusivity (m^2/s)
k = thermal conductivity (W/mK)
T_{rec} = recovery temperature (K)
T = temperature (K)
ΔT_m = melt temperature range
M = Mach number (dimensionless)
r = recovery factor (dimensionless)
N_f = freezing fraction (dimensionless)
A = area (m^2)
s = surface distance (m)
x = x-coordinate (m)
y = y-coordinate (m)
F = force (N)
v_f = water flow velocity
τ_f = shear stress (N/m^2)
θ = contact angle (radians)
b = bead height (m)
μ_f = viscosity of fluid (kg/ms)
W_e = Weber number (dimensionless)
σ = surface tension (N/m)
S = spread factor (dimensionless)
q''' = internal heat source term (W/m^3)
C_d = drag coefficient (dimensionless
Re = Reynolds number (dimensionless)
MVD = volumetric mead droplet diameter (m)
D = diameter (m)
δ = boundary layer thickness (m)
Nu = Nusselt number (dimensionless)
c'_f = friction coefficient (dimensionless)
k_s = equivalent sand-grain roughness (m)
ν = kinematic viscosity (m^2/s)

Subscripts

i = ice
evap = evaporative term
air = air
e = edge of boundary layer
s = surface
ke = kinetic term
o = total property
imp = impingement term
water = water
sens = sensible heat term
∞ = ambient condition
f = freezing
rb = runback
in = incoming term
out = outgoing term
shed = mass shed
remain = mass remaining
freeze = freeze

4.0 Summary

LEWICE 1.6 is an ice accretion prediction code that applies a time-stepping procedure to calculate the shape of an ice accretion. The potential flow field is calculated in LEWICE 1.6 using the Douglas Hess-Smith 2-D panel code (S24Y) (3). This potential flow field is then used to calculate the trajectories of particles and the impingement points on the body (4). These calculations are performed to determine the distribution of liquid water impinging on the body, which then serves as input to the icing thermodynamic code. The icing model, which was first developed by Messinger (5), is used to calculate the ice growth rate at each point on the surface of the geometry. By specifying an icing time increment, the ice growth rate can be interpreted as an ice thickness which is added to the body, resulting in the generation of new coordinates. This procedure is repeated, beginning with the potential flow calculations, until the desired icing time is reached.

The operation of LEWICE 1.6 is illustrated through the use of several examples. These examples are representative of the types of applications expected for LEWICE 1.6.

LEWICE 1.6 has been used to calculate a variety of ice shapes, but should still be considered a research code. The code should be exercised further to identify any shortcomings and inadequacies. Any modifications identified as a result of these cases, or of additional experimental results, should be incorporated into the model. Using it as a test bed for improvements to the ice accretion model is one important application of LEWICE 1.6.

5.0 Background

The evaluation of both commercial and military flight systems in icing conditions has become important in the design and certification phases of system development. These systems have been evaluated in flight in natural icing, in a simulated cloud produced by a leading aircraft, and in ground test facilities. All icing testing is relatively expensive, and each test technique, i.e., flight or ground testing, has operational limitations which limit the range of icing conditions that can be evaluated. It would benefit the aircraft or flight system manufacturer to be able to analytically predict the performance of the system for a range of icing conditions.

The first step in the prediction of the performance characteristics is the determination of the location, size, and shape of the ice that will form. An analytical ice accretion model would allow the evaluation of a wide range of proposed test conditions to identify those that will be most critical to the flight system. This could substantially reduce the amount of test time required to adequately evaluate a system and increase the quality and confidence level of the final evaluation. The analytically predicted ice accretion could also serve as the input to an advanced aerodynamic or system performance code to allow more complete evaluation in the design phases of the system.

Based on this need for an analytic model, in 1983 three computer codes were developed as a result of university grants and in-house research at the NASA Lewis Research Center: a potential flow code, a droplet trajectory code, and an energy balance code. This combined effort, which was called LEWICE, was used exclusively for in-house research at Lewis. In current nomenclature, this will be called LEWICE version 0.1.

Through funding by FAA and NASA Lewis, in 1987 the previous codes were combined into a form usable by industry and distribution began. This version will be called LEWICE 0.5.

Through additional funding by NASA Lewis, interactive graphics capabilities were added and a correlation for surface roughness were added. The code's capabilities and usefulness were documented in CR 185129, *Users Manual for the NASA Lewis Ice Accretion Code, LEWICE (1).* This version will be called LEWICE 1.0

As usage of the code increased, both in industry and at NASA Lewis, several errors were detected and fixed in the code and several new features were added (6). These new capabilities were documented in CR 195387, *Update to the NASA Lewis Ice Accretion Code LEWICE.* This version was initially released in June 1993 as LEWICE Beta, and will be called LEWICE 1.3 in current nomenclature.

The purpose of the current study was to improve the ice accretion capabilities of the LEWICE code, especially in the glaze ice regime, and to add features to the code which give it greater flexibility and usefulness. These improvements were primarily in four areas: the paneling of the surface and definition of the control volumes were improved to obtain more consistent results and to run more time steps; the impingement limit searches were improved to obtain better limits for multi-drop size cases; the roughness size routines were improved to eliminate the need for a sand-grain roughness input by the user; finally, a multi-body capability was added. These improvements were developed at NYMA, Inc. and at the NASA Lewis Research Center under NASA funding.

6.0 Introduction

The computer code, LEWICE 1.6, embodies an analytical ice accretion model that evaluates the thermodynamics of the freezing process that occurs when supercooled droplets impinge on a body. The atmospheric parameters of temperature, pressure, and velocity, and the meteorological parameters of liquid water content (LWC), droplet diameter, and relative humidity are specified and used to determine the shape of the ice accretion. The surface of the clean (uniced) geometry is defined by segments joining a set of discrete body coordinates. The code consists of four major modules. They are 1) the flow field calculation, 2) the particle trajectory and impingement calculation, 3) the thermodynamic and ice growth calculation, and 4) the modification of the current geometry by adding the ice growth to it.

LEWICE 1.6 applies a time-stepping procedure to "grow" the ice accretion. Initially, the flow field and droplet impingement characteristics are determined for the clean geometry. The ice growth rate on each segment defining the surface is then determined by applying the thermodynamic model. When a time increment is specified, this growth rate can be interpreted as an ice thickness and the body coordinates are adjusted to account for the accreted ice. This procedure is repeated, beginning with the calculation of the flow field about the iced geometry, then continued until the desired icing time has been reached.

Ice accretion shapes for cylinders and several single-element and multi-element airfoils have been calculated using this computer code. The calculated results have been compared to experimental ice accretion shapes obtained both in flight and in the Icing Research Tunnel at NASA Lewis Research Center. The comparisons using the improved code have been very encouraging.

This report will, in several places, duplicate information contained in the previous LEWICE Users Manuals so that the user does not need to read three manuals to fully utilize this code. This report will also document the modifications made, including changes to the physical model and improvements to the numerics of the program. It will also cover additional features of the code which users may find useful. The new capabilities of this code will now be described.

Due to the feed back of the user community, four major features have been added to the NASA Lewis ice accretion code LEWICE 1.6. These features include: first, further improvements to the numerics of the code so that more time steps can be run and so that the code is more stable; second, inclusion and refinement of the roughness prediction model described in an earlier paper (6); third, improvement of the impingement limit searches to improve droplet trajectories especially for multiple drop size cases; fourth, inclusion of multi-element trajectory and ice accretion capabilities to LEWICE 1.6. This report will describe each of these advancements in full and make comparisons with the experimental data available (7).

The numerics of the code have been improved in several ways to produce more accurate ice shapes. The convergence characteristics of the code have been improved by implementing an adaptive grid technique, a new ice growth algorithm and a new variable time stepping scheme. Improvements to the transition model and transition heat transfer calculations have been made to produce more realistic results. New additions include a "pseudo" surface which produces more realistic heat transfer for large glaze ice shapes and a mass addition routine which allows ice growth in arbitrary directions.

An adaptive grid scheme has been implemented, which allows more optimal tailoring of the individual surface models for each phase of the ice growth process yielding smoother more accurate ice shapes, better convergence characteristics and quicker run times. A highly refined "baseline" model is used to represent the geometry at each time step. This model is updated after each time step.

Adaptive griding techniques are used to generate optimal surface models from the baseline model for the heat transfer, the collection efficiency, mass balance, energy balance, and the mass addition phases. A typical baseline model may contain 4000 points. A typical flow panel model for accurate trajectory calculation may have constant leading edge spacing and require 100 points. Flow panel models used for the generation of velocities for the heat transfer coefficient calculation, which can be either of the "pseudo" surface type (which produces more

realistic heat transfer for glaze shapes) or keyed to radius of curvature, may typically require 150 panels. A typical surface model for the energy balance, the mass balance and mass addition algorithms may contain several thousand points.

A new ice growth scheme has been implemented. This scheme employs a separate time stepping procedure on a highly refined surface model. The method features local conservation of mass, accurate resolution of complex ice shapes, and eradication of the troublesome problem of iced lobes growing into each other.

The time stepping procedure has been automated. The user specifies the maximum ice thickness to be added at each time step. The time step is varied at each time step to match this maximum amount of ice thickness. This ice growth scheme yields better convergence characteristics by giving the user more precise control over the maximum geometric change and hence aerodynamic change between consecutive time steps.

Additional improvements include improvements to the transition model, the transition heat transfer calculation, incorporation of a more realistic "pseudo" surface and additions to the ice growth module. The transition model has been modified to produce more realistic transition locations for cases with large or multiple stagnation points. A more realistic treatment of the laminar and turbulent heat transfer coefficient in the transition region has been implemented. The new ice growth model allows ice growth in arbitrary directions to accommodate current and future ice growth models. An optional "pseudo" surface method has been installed which more accurately models flow and hence heat transfer for ice shapes with large stagnation zones.

The roughness prediction model used is the same model that was described in a previous paper (6). Previously, this model was not considered reliable due to the deteriorating accuracy of the code for multiple time steps. Due to the increased accuracy of the code for multiple time steps, this routine was reactivated. Comparisons will be made between this model and the measured roughness heights obtained by Shin (8).This routine is considered reliable enough that the standard input of sand-grain roughness into LEWICE 1.6 has been removed.

The third feature is the improvement made to the impingement limit search. For many complex geometries, and especially for multi-drop size cases, the code would not be able to find impingement limits for cases where it was known that they existed. This occurred because the code would lose track of where the airfoil was in relation to the drops. Many times, the code would exit due to this error. The current scheme for determining impingement limits does not have this error, as shown in the example cases.

The fourth feature is the addition of multi-element capability to LEWICE 1.6. The potential flow solver has always been capable of producing multi-element flows, but only now have the trajectory, energy balance and ice addition routines been correctly modified to produce multi-element ice accretions. A comparison will be made with experimental data obtained in the IRT (7).

7.0 Description of the LEWICE 1.6 Model

This section will present the physical model used in the LEWICE 1.6 ice accretion code, as well as derivation of equations where necessary.

7.1 Calculation of the Flow Field

The flow solver used in the LEWICE 1.6 code was first developed at Douglas by Hess and Smith (3). It is a two-dimensional potential flow code capable of handling up to 10 separate bodies and up to 10^5 panels in its modified form here. The use of a potential flow code is desirable due to speed of execution and its ability to calculate a flow field around the irregular geometries produced during ice accretion. Testing in flight and in the NASA Lewis Icing Tunnel (9) has shown that ice accretion is a bootstrap process, meaning that the ice growth rate at a given time is a function of the ice which has already formed. Therefore, the accuracy of an ice accretion code depends on being able to produce a flow solution several times during a case in order to simulate this process.

The Hess/Smith code produces a flow solution by using a distribution of sources, sinks, and/or vortices along the body geometry. In calculating the flow field, contributions from all the sources, sinks, and/or vortices are summed. The body surface itself is represented by several straight line segments called panels. The flow code is limited to cases where the Mach number and angle attack are low (Mach ≤ 0.5 and $\alpha \leq 11°$ [before stall]). Only limited details of the flow methodology are presented, as the mechanisms of potential flow are usually well understood by the users of the code, who tend to be aerospace engineers. Other readers are referred to the paper by Hess and Smith.

LEWICE 1.6 also performs a correction to the flow solution for compressibility. This is based on the Kármán-Tsien method (10). The equation for this is

$$c_{p_{comp}} = \sqrt{1 - M_\infty^2} + \frac{c_{p_{inc}}}{2} \frac{M_\infty^2}{1 + \sqrt{1 - M_\infty^2}}$$

where Cp_{inc} is the incompressible pressure coefficient calculated and Cp_{comp} is the pressure coefficient corrected for compressibility.

Several users of LEWICE 1.6 already have a flow field solver which their company uses for design. It is possible to remove/ignore the flow solution used in LEWICE 1.6 and use other codes. This has been done by NASA Lewis for ARC2D, a 2D Naviér-Stokes code developed by NASA Ames (11). This version of LEWICE is also available from the NASA Lewis Icing Branch.

Another important output from the flow solver is the location of the stagnation point. This is the point at which the local tangential velocity is zero, or most nearly so. Air, and thus surface 'runback' water, flow away from the stagnation point toward the trailing edge of the airfoil. For near-freezing temperatures, much of the incoming water does not freeze on impact, and prediction of surface water flow is critical to accurate ice prediction. For a clean airfoil, selection of the stagnation point is a trivial procedure, as the surface is smooth.

As ice builds into odd shapes, this procedure becomes more difficult. The tangential velocity predicted often changes direction over a region of very slow flow near the 'true' stagnation point. LEWICE 1.6 handles this problem by locally smoothing the results to find the 'true' stagnation point, and then treating the entire region of 'multiple' stagnation points as laminar flow for the calculation of the boundary layer profile.

7.2 Boundary Layer Calculation

LEWICE 1.6 performs an integral boundary layer calculation solely for the purpose of predicting the local heat transfer coefficient on the body(s). The equations used for this purpose are derived in Schlichting (10), and a summary of this procedure is repeated here. This calculation is made much more difficult by the accumulation of ice, which can make the surface roughened. This causes turbulence, which increases the heat transfer coefficient, thus causing more ice to accumulate. Also, the theory presented here assumes the roughness to be small in relationship to the boundary layer thickness. Tests performed in the NASA Lewis Icing Research Tunnel have shown that this is not the case. A more accurate

model for this phenomena will be available in future versions of the code.

7.2.1 Laminar Boundary Layer

The boundary layer equations are formulated in integral form and are solved using a standard von Kármán - Pohlhausen method. This method will be outlined briefly here. A more complete derivation is performed by Schlichting.

First, the boundary layer equation is written in terms of the momentum thickness and the displacement thickness,

$$U^2\frac{d\delta_2}{ds} + (2\delta_2 + \delta_1)\,U\frac{dU}{ds} = \frac{\tau_o}{\rho}$$

Then, a fourth degree polynomial is defined for the velocity near the surface,

$$\frac{u}{U} = a\eta + b\eta^2 + c\eta^3 + d\eta^4$$

where $\eta = y/\delta$, $\Lambda = \delta^2/\nu\ dU/ds$ and

$a = 2+\Lambda/6$, $b = -\Lambda/2$, $c = -2 + \Lambda/2$, $d = 1 - \Lambda/6$

By defining the shape factors

$K = \delta_2^2/\nu\ dU/ds$ and $Z = \delta_2^2/\nu$, the equations can be written in the form

$dZ/ds = F(K)/U$; $K = Z\ dU/ds$

where

$$F(K) = \frac{2\tau_o\delta_2}{\mu U} - 4K - 2K\frac{\delta_1}{\delta_2}$$

At s = 0, the following results are known

$F(K) = 0$, $K_o = 0.077$, $L_o = 7.052$, $Z_o = 0.077/(dU/ds)$, $(dZ/ds)_o = -0.0652\ (d^2U/ds^2)/(dU/ds)^2$

The solution procedure is as follows:

1) The potential flow function U(s) together with its derivative dU/ds are known;

2) Integration of the above equations gives the shape factors Z(s) and K(s) so that displacement thickness δ^2 can be calculated;

3)The first shape factor Λ(s) is found;

4) The displacement thickness, δ_1, and the shearing stress at the wall, τ_o, are found;

5) The boundary layer thickness δ(s) is found;

6) Finally, the velocity distribution is found.

For the thermal boundary layer, an approximate formula is used which was first developed by Smith and Spalding,

$$\frac{U d\delta_T^2}{\nu\, ds} = 46.72 - 2.87\frac{\delta_T^2 dU}{\nu\, ds}$$

This formula, which is exact for both flat plate and at stagnation for a Prandtl number of 0.7, is assumed to enjoy universal validity and compares well with the exact solution for a circular cylinder and with various angle wedges. This formula can be integrated directly to obtain

$$\left(\frac{\delta_T}{c}\right)^2\frac{U_\infty c}{\nu} = \frac{46.72}{\left(\frac{U}{U_\infty}\right)}\int_0^{\frac{s}{c}}\left(\frac{U}{U_\infty}\right)^{1.87} d\left(\frac{s}{c}\right)$$

The local Nusselt number at the surface is simply $Nu = 2\ (c/\delta_T)$.

At stagnation, this formula gives an indefinite answer, hence the limit is determined using L'Hospital's Rule. The result is

$$\left(\frac{\delta_T}{c}\right)^2\frac{U_\infty c}{\nu} = \frac{16.28}{\left|\frac{d}{d\left(\frac{s}{c}\right)}\left(\frac{U}{U_\infty}\right)\right|}$$

7.2.2 Turbulent Boundary Layer

Since ice is formed on the leading edge of an airfoil where the flow is originally laminar and since ice is known to have a roughened surface, the transition from laminar to turbulent heat transfer is assumed to be caused by this roughness. The accepted criteria for boundary layer transition on a rough surface is $Re_k \geq 600$ where $Re_k = U_k k_s/\nu$, U_k = velocity at the roughness height and k_s = the equivalent sand-grain roughness. The equation used for the Nusselt number is derived from experimental data on various sand-grain roughnesses (12,13). The form is

$$Nu_x = \frac{\frac{1}{2}c'_f Re_x Pr}{Pr_t + \sqrt{\frac{1}{2}c'_f}\,(0.52\,(Re_{k,r})^{0.45} Pr^{0.8})}$$

where $Re_{k,r} = u_r k_s/\nu$ and $u_r = U_e\sqrt{(c'_f/2)}$

The skin friction is derived by using the momentum law of the wall for fully rough flow (14)

$$\frac{d}{dy^+}u^+ = \frac{1}{\kappa(y^+ + (\delta y_o)^+)}$$

This is integrated to find u^+, noting that experimentally, $(\delta y_o)^+ = 0.031\ Re_k$

$$u^+ = \frac{1}{\kappa}\log\left(\frac{32.6y^+}{Re_k} + 1\right)$$

At the outer edge of the boundary layer, u^+ is always greater than the law of the wall by an additive 2.3 which gives

$$u_\infty^+ = \frac{1}{\kappa}\log\left(\frac{32.6y^+}{Re_k} + 1\right) + 2.3$$

Finally, $u^+_\infty = 1/\sqrt{(c'_f/2)}$ and $\delta_2/\delta = 0.097$ thus

$$\frac{c'_f}{2} = \left(\frac{0.41}{\log\left(\frac{864\delta_2}{k_s} + 2.568\right)}\right)^2$$

The turbulent momentum thickness is found by inserting the power law formula for velocity into the momentum integral equation. This yields, after integration,

$$\delta_2 = \frac{0.36\nu^{0.2}}{U^{3.29}}\left(\int_0^s U^{3.86}ds\right)^{0.8}$$

7.2.3 Integration Methodology

The standard integration techniques are derived assuming constant spaced intervals. It was desirable to obtain an integration formula for variable spacing so that the panels could be used as the integration steps. One such fifth order scheme for constant spacing is (15)

$$\int_{x_0}^{x_1} f(x)\,dx = h\left(\frac{55}{24}f_1 - \frac{59}{24}f_2 + \frac{37}{24}f_3 - \frac{9}{24}f_4\right)$$

This equation can be derived by replacing the numerical coefficients with unknown, then substituting f(x)=1, f(x)=x, f(x)=x^2, and f(x)=x^3 into the above equation and solving the resulting four linear polynomials for the four unknown coefficients. This procedure can be easily extended to variable spaced systems and should lead to a formulation of similar accuracy. Performing this task yields the following four equations to be solved:

$$1 = p + q + r + s$$

$$\frac{1}{2} = p + q(1+a) + r(1+a+b) + s(1+a+b+c)$$

$$\frac{1}{3} = p + q(1+a)^2 + r(1+a+b)^2 + s(1+a+b+c)^2$$

$$\frac{1}{4} = p + q(1+a)^3 + r(1+a+b)^3 + s(1+a+b+c)^3$$

where a, b, and c represent the known ratios

$a = \Delta s_{i+1}/\Delta s_i$, $b = \Delta s_{i+2}/\Delta s_i$, $c = \Delta s_{i+3}/\Delta s_i$

For constant spacing, a=b=c=1 and the equations when solved for p, q, r, and s give the numerical coefficients given above. For the variable spaced system given here, the solution to the above four equations is

$$p = \frac{1}{12}\left[\frac{12a^3 + A + B + C + D}{a(a+b)(a+b+c)}\right]$$

where

$$A = 6a^2(4b + 2c + 3)$$

$$B = 12a(b+1)(b+c+1)$$

C = 6b (b+c+1); D = 4c + 2b + 3

$$q = -\frac{1}{12}\left[\frac{(6a+6b+4)(a+b+c) + 4a + 4b + 3}{ab(b+c)}\right]$$

$$r = \frac{1}{12}\left[\frac{(6a+4)(a+b+c) + 4a + 3}{bc(a+b)}\right]$$

$$s = -\frac{1}{12}\left[\frac{(6a+4)\,(a+b)+4a+3}{c\,(b+c)\,(a+b+c)}\right]$$

7.3 Droplet Trajectory Calculation

The algorithm used in LEWICE 1.6 for calculation of droplet trajectories was originally developed by Frost, Chang, Shieh, and Kimble of FWG Associates under contract to NASA Lewis (4). Since that time, it has gone under substantial modification, especially for this version of the code.

The code was developed to calculate the shape of arbitrarily-shaped particles and although water droplets are the use of LEWICE 1.6 for icing simulations, the code retains the flexibility to handle other models. One such application which has benefitted industry is the use of the droplet trajectory for calculating sand particle trajectories. The same physics apply, with the only difference residing in the physical properties of the drop. In this case, the only property used in LEWICE 1.6 is the density of the particle.

The objective of a particle trajectory code is to determine the impingement limits and the local and total collection efficiencies. The upper and lower impingement limits are the last points aft of the stagnation point which are hit by a water droplet. The collection efficiency is a measure of the particle's deflection from the free stream and represents the fraction of the liquid water content captured by that location on the airfoil. Total collection efficiency represents the total fraction captured by the airfoil. The total collection efficiency is defined as the ratio of the actual mass of impinging water to the maximum value that would occur if the droplets followed straight-line trajectories. This is illustrated in Figure 7.1.

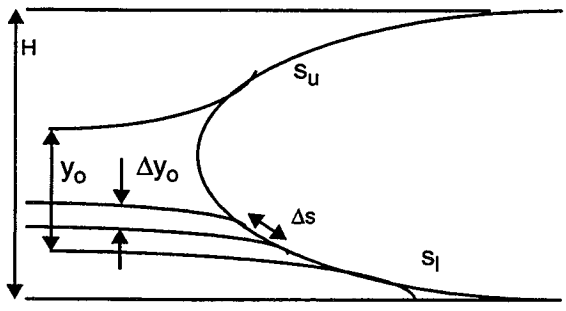


Figure 7.1 Definition of Impingement Limits

This definition can be given in equation form as

$$E_m = \frac{y_o}{H}$$

where y_o is the vertical distance between the droplet release points of the upper and lower surface tangent trajectories. The local collection efficiency, β, can be written in differential form as

$$\beta = \frac{dy_o}{ds}$$

It is related to the total collection efficiency by the equation

$$E_m = \frac{1}{H}\int_{s_l}^{s_u} \beta\, ds$$

where s_u and s_l are the upper and lower surface impingement limits, respectively. The following sections will cover the methods applied to calculate the variables discussed above.

7.3.1 Equations of Particle Motion

The motion of a particle is analyzed as a point mass particle that is acted on by the potential flow field but which itself does not affect the flow. The forces acting on the particle are considered to be those of lift, drag, pitching moment, and gravity. Figure 7.2 shows the forces acting on the particle and the velocity vectors relative to the motion of the particle. The flight reference line (FRL) is not significant for a spherical particle; however, for arbitrarily shaped particles, i.e., a snow flake, the FRL must be defined relative to the lift, drag, and moment coefficient data available.

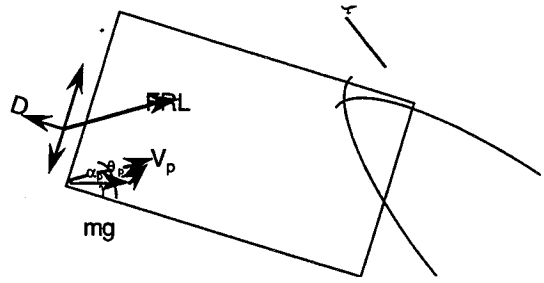


Figure 7.2 Forces acting on a particle

The equations of motion of an arbitrarily shaped particle are derived from a force balance on a point mass (16), as shown in Figure 7.2, and are as follows:

$$m\ddot{x} = -\vec{D}\cos\gamma - \vec{L}\sin\gamma + mg\sin\alpha$$

$$m\ddot{y} = -\vec{D}\sin\gamma + \vec{L}\cos\gamma - mg\cos\alpha$$

where

$$\gamma = \text{atan}\frac{\dot{y}_p - V_y}{\dot{x}_p - V_x}$$

In Figure 7.2, note that the coordinate system used in LEWICE 1.6 is fixed to the leading edge of the clean airfoil.

For an airfoil at an angle of attack α, the coordinate system is at an angle to the gravitational coordinate system. Therefore, the effect of gravity must be accounted for in the equations for both lift and drag.

The flow field velocity components in the x and y directions, i.e., V_x and V_y, respectively, are obtained from the potential flow program. The aerodynamic drag and lift forces are defined as

$$\vec{D} = c_d\frac{\rho_a V^2}{2}A_p$$

$$\vec{L} = c_l\frac{\rho_a V^2}{2}A_p$$

where A_p is a characteristic area of the particle, ρ_a, is the density of air at the position of the particle, and V is the particle velocity relative to the flow field and defined as

$$V = \sqrt{(\dot{x}_p - V_x)^2 + (\dot{y}_p - V_y)^2}$$

For arbitrarily shaped particles, the pitch angle, θ_p, is required to evaluate the angle of attack α_p, using the following equation

$$\alpha_p = \theta_p - \gamma$$

This motion is governed by the following equation

$$\ddot{\theta} = \frac{M}{I_{zz}}$$

where I_{zz}, is the moment of inertia of mass relative to the z axis. The moment of aerodynamic forces acting on the particle is

$$M = c_m\frac{\rho_a V^2}{2}A_p d_p$$

where c_m is the pitching moment coefficient which must also be specified by the user.

The lift, drag, and pitching moment coefficients, c_l, c_d, and c_m respectively, must be provided by the user for arbitrarily shaped particles. The coefficient data is input to the program through subroutine COEFF and should be functions of the particle angle of attack and the particle Reynolds number based on the particle diameter, given by the following equation:

$$Re_p = \frac{Vd_p}{\upsilon}$$

The diameter of the particle, d, and the kinematic viscosity of air, ν, are assumed constant along the trajectory of the particle.

Since water droplets are usually assumed to be rigid spheres in icing studies, the only forces considered to be acting on the particle are those of drag and gravity. The governing equations can therefore be simplified as follows:

$$m\ddot{x} = -\vec{D}\cos\gamma + mg\sin\alpha$$

$$m\ddot{y} = -\vec{D}\sin\gamma - mg\cos\alpha$$

In this case, the drag force, D, is determined using a steady-state drag coefficient for a sphere which is a function of the droplet Reynolds number, Re_p. Approximating droplets as rigid spheres is valid for drop radii less than 500 microns(17). A valid drag law for spherical particles is built into LEWICE 1.6 (18).

For particles with diameters of less than 10 microns, the ratio of particle diameter to the mean distance between air molecules is small enough so that molecular slip phenomena result in drag forces lower than those calculated by the drag law used in

LEWICE 1.6. The Cunningham correction factor(19), C_f, is therefore applied to correct the drag coefficient using the following equation:

$$c_{d_{slip}} = \frac{c_d}{C_f}$$

The values of C_f are input by the user when necessary and are given in Table 7.1.

Table 7.1 Cunningham Correction Factors

Diameter (μm)	Correction Factor
1	1.168
2	1.084
3	1.056
4	1.042
5	1.034
6	1.028
7	1.024
8	1.021
9	1.019
10	1.017

Substituting $\ddot{x} = \frac{d}{dt}\dot{x}$ and $\ddot{y} = \frac{d}{dt}\dot{y}$ gives

$$\frac{d}{dt}\dot{x} = -\frac{\vec{D}}{m}\cos\gamma + g\sin\alpha \text{ and}$$

$$\frac{d}{dt}\dot{y} = -\frac{\vec{D}}{m}\sin\gamma - g\cos\alpha$$

gives the equations as they are expressed in LEWICE 1.6.

These equations are integrated using the method of Gear developed for stiff equations (20,21). The details of the subroutines that make up the integration method can be found in Reference 16 and in COMMENT statements in the computer code.

7.3.2 Determination of Droplet Impingement

The calculation of the droplet trajectories is continued until the droplets impinge upon the body or move out of range. This section describes the procedure used to determine whether or not a droplet impacts the body and, if so, the location of impingement. This section will describe the method used to find trajectory impingement limits from the equations described in the previous section.

As previously discussed, the geometry is defined by segments joining a discrete set of body coordinates. A droplet is considered to impact the body when its trajectory intersects one of these body segments. The current model does not take account of grazing collisions or droplets that may impact the body so that they are re-introduced into the flow by bouncing, splashing, etc. (22).

The impact algorithm determines the intersection of the particle trajectory and the line connecting the ice shape points. Once the body segment through which the particle passed and the intersection (impingement) point have been determined, the surface distance, s, from the stagnation point to the impingement point is determined by interpolation.

After calculating the flow field about the body, the program enters the particle trajectory subroutine. First, the initial particle location x_o, y_o and velocity V_x, V_y is determined by the code. A particle is released at a location upstream of the airfoil where the flow field is essentially the same as the free stream conditions. The program will select an initial upstream x-coordinate, x_o, by searching for a position where the difference between the local velocity V_l and the freestream velocity V_∞ are within the tolerance specified by the input parameter VEPS.

In order to find the impingement limits, the code first defines a broad range from which to search. This is achieved by finding the starting points of a trajectory which passes above all the bodies and one which passes below all the bodies. Subsequent searches are contained within these limits. The next search is started half-way between these two limits. If this hits the surface, the upper limit lies between this trajectory and the upper limit. The code iterates by starting a trajectory half-way between the highest trajectory which hits and the lowest which misses until the distance between the two is less than YOLIM. If the trajectory misses, the code determines whether it passed above or below the body(s) and replaces the upper or lower range limit with the current value. A similar process is used to find the lower impingement limit.

This process is straight-forward for single bodies, but becomes much more complicated when several bodies are considered. First, the trajectory routine is set up to only handle one body at a time. Therefore,

trajectories for the first body must consider only hits on that body, and treat hits on other bodies as misses. For other bodies, particles which hit bodies past itself are treated as simple misses, but particles which strike bodies previous to it in space must be dealt with separately. Similarly, a particle which misses and passes between bodies must be treated differently than particles which miss above or below all bodies. For this reason, it is convenient to define a global range, as defined earlier and also a local range for each body. Particles which hit previous bodies are assumed to pass above the local range. If LBOTH=1 in the input field, the code will perform the calculation both ways and obtain two arrays of collection efficiencies for each body except the first. These two are then merged into one collection efficiency array for further calculation.

7.3.3 Calculation of the Local Collection Efficiency

The particle trajectories and impingement points, calculated as previously described, are used to establish the relations between the particle's initial position (x_0,y_0) and the position where it impinges on the body surface, specified by the surface distance, s, which is the length along the body surface measured from the stagnation point. The value of s is defined as negative on the lower surface and positive on the upper surface.

The local collection efficiency is calculated by first calculating droplet trajectories and producing a plot of particle release point, y_o, vs. surface impact distance, s, as shown in Figure 7.3. As was indicated earlier, the local collection efficiency is a function of the surface distance and can be determined by differentiating the curve shown in Figure 7.3 with respect to s. The derivative at the center of each body segment is calculated by linear interpolation.

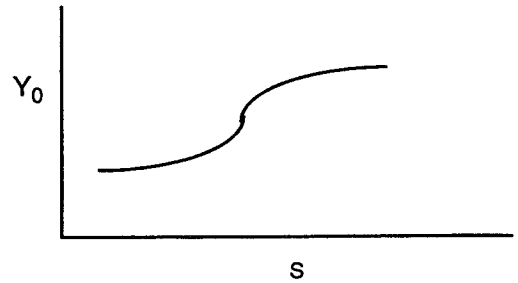


Figure 7.3 Example of a Y0 vs. S curve

7.3.4 Local Collection Efficiency Calculation for Multidispersed Particle Distributions

The previous section described how the local collection efficiency was calculated for a single droplet diameter. In icing applications, the mass median droplet diameter of the droplet size distribution is used to characterize the size of the droplets. A feature of the particle trajectory portion of the trajectory program is that it allows the user to analyze the local collection efficiency for a multidispersed particle distribution (23).

To perform this calculation, the user must input the droplet diameter and the associated mass fraction and Cunningham correction factor for each specified droplet size. A maximum of 10 droplet sizes can be used to characterize a droplet distribution. For example, the required input for a Langmuir 'D' distribution with a mass median of 20.0 microns is shown in Table 7.3.

Table 7.3: Langmuir D droplet size distribution with a mass median of 20 microns

% LWC	Ratio of Droplet Diameters	Droplet Diameter (μm)	Cunningham Correction Factor
0.05	0.31	6.2	1.0272
0.10	0.52	10.4	1.00
0.20	0.71	14.2	1.00
0.30	1.00	20.0	1.00
0.20	1.37	27.4	1.00
0.10	1.74	34.8	1.00
0.05	2.22	44.4	1.00

The solution procedure is begun by calculating the local collection efficiency distribution for each droplet size characterizing the distribution. The local collection efficiency for the distribution is determined by summing the contributions of each of the droplet sizes using the following equation:

$$\beta(s) = \sum_{i=1}^{N} n_i \beta_i(s)$$

where n_i is the mass fraction of liquid water associated with droplet diameter i and N is the number of droplet sizes used to characterize the distribution.

7.4 Discussion of the Ice Accretion Process

The model of the ice accretion process applied in LEWICE 1.6 is presented in this section, beginning with a discussion of some of the general characteristics of ice accretion shapes, and followed by a description of the physical model of the ice accretion process from which a mathematical model must be formulated.

7.4.1 Ice Accretion Characteristics

Before discussing the physical model of the ice accretion process, it is necessary to define some of the terms used in such a discussion.

Ice may form on the forward facing surfaces of an aircraft flying through clouds composed of supercooled water droplets. The type and shape of ice that forms are functions of the atmospheric parameters of velocity, pressure, and temperature, and the meteorological parameters of liquid water content, droplet diameter, and icing time.

Ice shapes are generally classified as glaze, mixed, and rime accretions. Rime ice is milky white and opaque. Glaze ice is generally clear and is characterized by the presence of larger protuberances, commonly known as glaze horns, as shown in Figure 7.4. A mixed ice accretion will have some of the characteristics of both glaze and rime ice accretions. As shown in Figure 7.5, the center portion of a mixed ice accretion will have the characteristics of glaze ice accretion. This glaze center will be surrounded by rime ice accretions, commonly called rime feathers because of their thin, feather-like shape and delicate structure.

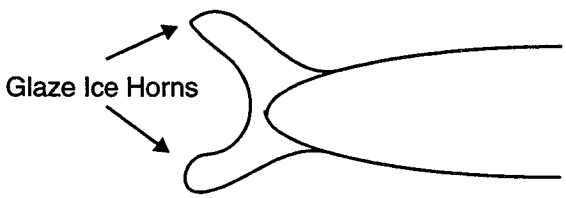


Figure 7.4 Example of a Glaze Ice Shape

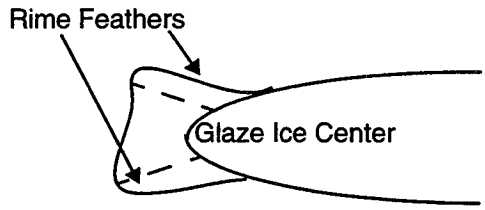


Figure 7.5 Example of Mixed Ice Accretion

The type of ice that will be formed is dependant on the atmospheric and meteorological conditions identified in the preceding paragraph. Predicting the type and shape of the ice accretion that will be formed for a specified set of icing conditions is difficult because of the complex interactions between the atmospheric and meteorological parameters. Typically, rime ice is formed at lower temperatures, velocities, and LWC than glaze ice. At the warmest temperatures, the accretion is composed exclusively of glaze ice. As the temperature decreases, areas of rime ice begin to form near the impingement limits. As the temperature decreases further, these rime portions increase in size until the accretion is composed solely of rime ice. The extent of icing and the locations at which ice forms on a surface are largely dictated by the size of the droplets impinging on the surface. For a given icing condition, and in the absence of ice shedding, the size of the accretion depends on the length of time ice is allowed to accrete. The general effects of temperature, droplet size, LWC, and angle of attack on ice shapes formed on a NACA 0012 airfoil in the NASA Lewis Icing Research Tunnel (IRT) are documented in Reference 24.

7.4.2 Description of the Physical Model

An understanding of the interactions between these parameters is required to predict the shape of an ice accretion that will be formed at a specified set of icing conditions. To develop this fundamental understanding, it is necessary to examine the physical model of the ice accretion process.

A model of the ice accretion process was first presented by Tribus (25) and developed further by Messinger (5). While many studies have been done to understand various aspects of the ice accretion process, the original physical model has been applied relatively unchanged. Recent close-up mov-

ies and photographs of the ice accretion process made at the NASA Lewis Research Center have increased our understanding of the process and indicated that modifications to the physical model may be necessary (26, 27). Conclusions drawn from the observations are included in the following discussion of the ice accretion process, and differences from the previous model are highlighted.

The ice accretion process is characterized by the presence of supercooled droplets entrained in the flow about a body. These droplets follow trajectories that will cause them to either be carried past or impinge upon a body.

Upon impact with a clean surface, the droplets coalesce into larger surface drops under the effects of surface tension and flow along the surface as dictated by the airflow along the surface of the body. These surface drops will then either freeze on the surface or be shed from the surface because of the aerodynamic forces on the drop. The ice accretions formed by this initial freezing form a rough surface which enhances the convective heat transfer and local collection efficiency of the surface, and therefore allows the ice accretion process to continue.

The type of ice that will form for a given set of conditions is determined primarily by the rate at which the freezing process occurs. For example, if the conditions are such that the droplets freeze rapidly, there is essentially no initial coalescing and flowing of the droplets. Instead, they freeze on impact and form the characteristic rime ice accretions. These accretions are opaque and milky white in color because of the presence of air bubbles that are trapped in the structure during the rapid freezing process.

As the rate of the freezing process decreases, the droplets begin to coalesce and flow on the surface. Upon freezing, these larger surface droplets form surface roughness elements which tend to enhance the convective heat transfer and local collection efficiency characteristics, which, in turn, enhance the continuing growth of the ice accretion in this region. These local areas of enhanced ice growth are, therefore, the beginnings of the characteristic horns found on mixed and glaze ice accretions. As the freezing rate decreases further, the drops flow further along the surface of the body before freezing, thus moving the regions of enhanced ice growth away from the stagnation point. This, in turn, causes the horns of the accretion to move further apart and forms the familiar glaze ice accretions. As the rate of the freezing process decreases, less air is trapped within the ice structure and the ice gradually becomes clearer until it is essentially transparent, as in glaze ice.

7.4.3 Calculation of the Thermodynamic Characteristics

The thermodynamic analysis of an icing surface was first developed by Tribus(18) from the physical model of the ice accretion process previously discussed. This model was used to calculate the heating requirements for icing protection and proposed LWC measurement systems. Messinger(5) developed the thermodynamic model further to include an analysis of the temperature of an unheated surface in icing conditions for three surface temperature regimes, i.e., less than 273.15 K, equal to 273.15 K, and above 273.15 K, and the concept of the freezing fraction, f, to be discussed later. These early formulations have been used in various icing applications.

As discussed earlier, microscopic movies of the ice accretion process made at NASA Lewis Research Center indicate that the process may be more accurately modeled by modifying the equations used in past icing studies. The observations reveal that, after the initial flow of the coalesced droplets on the surface, the liquid does not flow but is caught and frozen in the grooves between the individual surface roughness elements. Incorporating this observation into a mathematical model would probably require modeling the individual roughness elements and the freezing of pools of water surrounded on all sides by ice. A microscopic and possibly three-dimensional analysis of the icing surface would be required to mathematically apply this model to an ice accretion prediction method. The mathematical model used in this and previous studies is more macroscopic in nature because the roughness elements do not directly effect the freezing process except to enhance the convective heat transfer coefficient.

The equations that model the thermodynamics of the freezing process on a body undergoing icing are formulated by performing a First Law of Thermodynamic mass and energy balance on a control volume located on the surface. The control volume to be

analyzed is located on the surface of the body and extends from outside the boundary layer to the surface of the body, as shown in Figure 7.6. The lower boundary of the control volume is initially on the surface of the clean geometry and moves outward with the surface as the ice accretes. Therefore, the control volume is always situated on either the clean or ice surface. Computationally, the surface geometry is represented by several straight-line segments. In previous versions, the panel model was also used for the control volume model. In this version, the control volume surface is defined by several times more segments than the panel model. This increased resolution is necessary to capture the details of the ice accretion process. Computationally, the cost of this refined model lies in the memory requirements more so than the CPU time. This occurs because the time-consuming chores are the calculation of the flow field and the droplet trajectories, both of which use the larger segments associated with the panel model. The thermodynamic analysis, by comparison, is much faster.

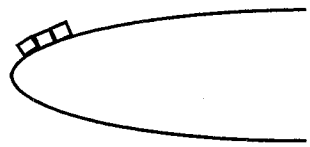


Figure 7.6 Control volume diagram

Before discussing the method involved, it is necessary to discuss several of the terms that are important to the solution procedure.

As discussed earlier, the atmospheric and meteorological parameters determine the type of ice that will form for a given icing condition. It has been found by various authors that the concept of a freezing fraction can be used to determine the type of ice that will form. The freezing fraction, f, was defined by Messinger as the fraction of impinging liquid that freezes within the region of impingement. In this application, f is defined as the fraction of the total liquid entering the control volume that freezes within the control volume. For colder icing conditions, the droplets tend to freeze immediately on impact, resulting in the formation of rime ice. Since all the water entering the control volume freezes within the control volume, the freezing fraction equals 1.0. Freezing fractions close to 0.0 characterize glaze or clear ice.

Freezing fractions between approximately 0.3 and 1.0 will normally indicate that the ice has some combination of glaze and rime characteristics. Ice accretions are often composed of glaze, rime, and intermediate regions. The local value of the freezing fraction therefore varies along the surface, and can be calculated using the mass and energy balances given below.

7.4.4 Solution of the Energy Equation

The evaluation of the freezing fraction is begun at the stagnation point because there will be no runback into the control volumes located on each side of the stagnation point. Therefore, it is first assumed that the equilibrium surface temperature, T_s, equals 273.15 K. The terms of the energy equation are then evaluated at this temperature, and the resulting expression is solved to determine the freezing fraction, f.

For $0.0 < f < 1.0$, $T_s = 273.15K$, and the initial assumption was correct. A value of $f < 0.0$ indicates that the surface temperature is greater than 273.15 K. Therefore, the solution is obtained by setting f = 0.0 and solving for T_s. Note that an iterative procedure is required since many of the terms are functions of T_s.

Similarly, $f > 1.0$ indicates that T_s is less than 273.15K, and f should be set equal to 1.0. Again, an iterative procedure must be applied to determine when the thermodynamic characteristics of the control volume are known, the mass balance is used to determine the mass flow rate of runback water out of the control volume. Any water flow out of the control volume will be away from the stagnation point and into the next control volume.

The above procedure is then repeated for the adjacent downstream control volume and continued along the upper surface of the body. The entire procedure is then repeated again, starting at the stagnation point and proceeding along the lower surface of the body.

7.5 Calculation of the Iced Geometry

When the freezing fraction has been determined for each segment (control volume) on the body, the local ice accumulation rate is determined from the mass balance. This ice growth rate must be inter-

preted as an ice thickness to form an ice accretion on the surface of the geometry.

The new ice surface is formed by first adding the ice thickness, d, perpendicular to each segment. The adjacent endpoints of each of these new segments are then averaged to obtain the coordinates describing the new ice surface. When the new surface is formed, the length of a segment increases.

The segments are allowed to grow but, at some point, must be split to maintain adequate definition of the surface. As the ice accretion grows, it is also possible for two lobes of the accretion to grow together, causing some of the points to lie in the interior of the body. These points must be removed in order to continue the calculations.

The point removal procedure is begun by applying the same procedure used to determine if two segments intersect. In this case, each body segment is checked with every other segment, excluding the two adjacent segments. If an intersection is found, all segments between the two intersecting segments are removed, and the set of coordinates is revised to reflect these changes. However, the mass removed is preserved by distributing this mass amongst the surrounding control volumes which remain.

7.5.1 General Computational Procedure

The previous sections discussed each of the individual phenomena of the ice accretion process that are evaluated in LEWICE 1.6. The purpose of this section is to describe how these individual calculations are implemented to form a complete ice accretion.

As discussed in the introduction, LEWICE 1.6 applies a time-stepping procedure to grow an ice accretion. The flow field and droplet impingement characteristics are initially determined for the clean geometry. The ice growth rate on each segment defining the surface is determined by applying the thermodynamic model. The new surface is then formed by specifying an icing time and applying the procedure described in the previous section to account for the accreted ice on the clean surface. After calculating this initial ice layer, the entire procedure is repeated, beginning with the calculation of the flow field about the iced geometry, to obtain revised local collection efficiency and thermodynamic data.

The above procedure is repeated by specifying discrete time increments until the desired icing time is reached. These time increments can be chosen by the user, or can be determined automatically by the code.

Ice accretions can have many geometrical shapes ranging from the smooth, aerodynamically-shaped rime accretions to rough glaze accretions with deep center grooves. LEWICE 1.6 is therefore required to calculate sufficiently accurate flow fields and particle trajectories about what can be very irregular geometries where viscous effects such as boundary layer separation and reattachment are important. This can, at times, exceed the capabilities of the potential flow code and produce non-physical results.

7.6 Energy Balance Derivation

Evaporation Term

Heat loss by evaporation is given by

$$Q_{evap} = m_{evap} * L_v$$

The mass loss is equal to the concentration difference across the boundary layer times the mass transfer coefficient, i.e.,

$$m_{evap} = h_m * (C_e - C_s)$$

The mass transfer coefficient is related to the heat transfer coefficient by the Chilton-Colburn analogy (28),

$$h_m = h_c / (\rho_e * c_{p,air} * \mathcal{L}^{2/3})$$

where

$$\mathcal{L} = \text{Lewis number} = Sc/Pr = k/(\rho * c_p * \mathcal{D}_{AB})$$

which is evaluated using film properties.The concentration is given as

$$C = P_v * MW/(R * T)$$

where P_v is the vapor pressure. Evaluating C at the surface and at the boundary layer edge and substituting gives

$$m_{evap} = h_c * MW_{water}/(\rho_e * c_{p,air} * R * \mathcal{L}^{2/3}) * (P_{v,e}/T_e - P_{v,s}/T_s)$$

The vapor pressure at the surface is the saturated vapor pressure by definition. Rather than com-

pute the relative humidity at the boundary layer edge, the vapor pressure at this point is related to the vapor pressure in the freestream via a mass balance which assumes no condensation or evaporation while the drop is travelling toward the airfoil. This mass balance can be written as (29)

$$P_{v,e}/P_e = P_{v,\infty}/P_\infty$$

The vapor pressure in the freestream is the product of the saturated vapor pressure times the relative humidity. The relative humidity in the freestream is an input variable to the code. The mass loss by evaporation is then

$$m_{evap} = h_c*MW_{water}/(\rho_e*c_{p,air}*R*\mathcal{L}^{2/3}) * (P_{v,\infty}* r_h*P_e/(T_e*P_\infty) - P_{v,s}/T_s)$$

Although the derivation could be stopped at this point, the form of the equations in LEWICE 1.6 replaces the density and temperature at the boundary layer edge using the ideal gas relationship and isotropic relations. Substituting

$$T_e = P_e*MW_{air}/(R*\rho_e)$$

$$\rho_e/\rho_o = (P_e/P_o)^{(1/\gamma)}$$

and

$$\rho_o*R = P_o*MW_{air}/T_o$$

gives

$$\mathbf{m_{evap} = h_c/c_{p,air}*(MW_{water}/MW_{air})*\mathcal{L}^{-2/3}) * (P_{v,\infty}/ P_\infty*r_h - P_{v,s}/P_o*T_o/T_s*(P_o/P_e)^{(1/\gamma)})}$$

which is the form LEWICE 1.6 uses. The heat loss term is then

$$\mathbf{Q_{evap} = L_v*h_c/c_{p,air}*(MW_{water}/MW_{air})*\mathcal{L}^{-2/3} * (P_{v,\infty}/P_\infty*r_h -P_{v,s}/P_o*T_o/T_s*(P_o/P_e)^{(1/\gamma)})}$$

As the energy equation solves for temperature, this can be put in the form

$$Q_{evap} = C_1-C_2*P_{v,s}/T_s$$

where

$$C_1 = L_v*h_c/c_{p,air}*(MW_{water}/MW_{air})*\mathcal{L}^{-2/3})*r_h*P_{v,\infty}/P_\infty$$

$$C_2=L_v*h_c*MW_{water}/(P_o*c_{p,air}*MW_{air}*\mathcal{L}^{2/3})$$

$$*T_o*(P_o/P_e)^{(1/\gamma)}$$

These terms (C_1 and C_2) are relatively constant with respect to surface temperature.

Conduction Term

Heat loss into the airfoil surface is modelled with the following assumptions:

1) Axial heat transfer is minimal, especially in the region of interest (glaze ice). The glaze ice surface temperature is, by definition T_{mp} (273 K).

2) Heat transfer effects can be modelled using a semi-infinite airfoil surface, since by the time the 'penetration thickness' reaches the inner surface of the airfoil, the magnitude of the heat flux at the surface is minimal.

3) The boundary condition at the ice surface assumes a stationary front, not a moving one as happens in reality. The time frame when heat conduction effects are important is short compared to the growth rate, making the assumption valid as an approximate calculation

4) The icing surface temperature exhibits a 'step change' at t=0 from the initial temperature (T_{rec}) to the icing temperature.

Using these assumptions, the heat loss on an unheated airfoil due to conduction during icing is given at each location (each control volume) as

$$\mathbf{Q_{cond} = - k*(T_s - T_{rec})/\sqrt{(\pi\alpha t)}}$$

(reference: Bird, Stewart & Lightfoot *Transport Phenomena* pp 352-4) (28)

Sensible and Latent Heat

These equations are derived by following the thermodynamic path of the water to its final state. If none of the water is going to freeze, there is only sensible heat transfer in heating the water to its final state.The equation is

$$\mathbf{Q_{sens} = m_{imp}*c_{p,water}*(T_s - T_\infty)}$$

for the impinging water and

$$\mathbf{Q_{sens} = m_{rb,in}*c_{p,water}*(T_s - T_{rb})}$$

for the runback water entering the control volume.

If part of the water is freezing, then there are two terms: 1) sensible heat needed to raise the water temperature to T_{mp} and 2) latent heat gain in freezing the water. The equations are

$$Q_{sens} = m_{imp}*c_{p,water}*(T_{mp} - T_{\infty})+m_{imp}*\Delta H_f*N_f$$

for the impinging water and

$$Q_{sens} = m_{rb,in}*c_{p,water}*(T_{mp} - T_{rb})+m_{rb,in}*\Delta H_f*N_f$$

for the runback water entering the control volume. The equations are solved in terms of temperature, not freezing fraction. The freezing fraction is replaced by temperature using the relationship

$$N_f = (T_{mp}+\Delta T_m-T_s)/\Delta T_m$$

where ΔT_m is a very small (10^{-5}) temperature range over which the ice freezes.

The term ΔH_f is not simply the heat of fusion because the formulation is based on a per volume basis instead of a per mass basis. The enthalpy per volume of water at temperature $T_{mp}+\Delta T_m$ is

$$H = \rho_{water}*(c_{p,ice}*(T_{mp}+\Delta T_m)+L_f)$$

The enthalpy per volume of ice at temperature T_{mp} is

$$H = \rho_{ice}*c_{p,ice}*T_{mp}$$

The difference in enthalpy per unit mass is then

$$\Delta H_f = (c_{p,ice}*(T_{mp}+\Delta T_m)+L_f) - \rho_{ice}*c_{p,ice}*T_{mp}/\rho_{water}$$

The sensible and latent heat terms are then

$$\mathbf{Q_{sens} = m_{imp}*[c_{p,water}*(T_{\infty}- T_{mp})+(c_{p,ice}*T_{mp}*(1-\rho_{ice}/\rho_{water})+c_{p,ice}*\Delta T_m+L_f)}$$

$$*(T_{mp}+\Delta T_m-T_s)/\Delta T_m]$$

for the impinging water and

$$\mathbf{Q_{sens} = m_{rb,in} * [c_{p,water} * (T_{rb}- T_{mp}) + (c_{p,ice}*T_{mp} * (1-\rho_{ice}/\rho_{water})+c_{p,ice}*\Delta T_m+L_f)}$$

$$*(T_{mp}+\Delta T_m-T_s)/\Delta T_m]$$

for the runback water entering the control volume.

If all of the incoming water freezes, there are three terms to account for: 1) sensible heat needed to raise the drop temperature to T_{mp}; 2) latent heat gain; and 3) sensible heat gain in lowering the ice temperature to T_s. The equations are

$$\mathbf{Q_{sens} = m_{imp}*[c_{p,water}*(T_{\infty} - T_{mp}) + c_{p,ice}*T_{mp}*(1-\rho_{ice}/\rho_{water}) +c_{p,ice}*(T_{mp}+\Delta T_m-T_s)+L_f]}$$

for the impinging water and

$$\mathbf{Q_{sens} = m_{rb,in}*[c_{p,water}*(T_{rb} - T_{mp}) +c_{p,ice}*T_{mp}*(1-\rho_{ice}/\rho_{water})+c_{p,ice}*(T_{mp}+\Delta T_m-T_s)+L_f]}$$

for the runback water entering the control volume.

For calculation purposes, all of these equations can be put in the form

$$Q_{sens} = C_1 + C_2*T_s$$

where C_1 and C_2 would be determined by the regime the control volume is in.

Note that this requires knowledge of the phase state prior to calculation of the temperature. This is performed in LEWICE 1.6 by performing a standard 'freezing fraction' calculation as the initial guess. This guess is then checked against the calculated temperature. Although this guess can be wrong theoretically, its occurrence is very unlikely. The check is made and corrected for if necessary, however.

Kinetic Heating

The two types of kinetic heating are kinetic heat gain from the air and the kinetic heat gain from the impinging water droplets. The kinetic heat gain due to the air is determined using a 'recovery temperature' as defined by Schlichting (10) (pp. 337-9 and 713-5). The heat gain is then defined from

$$\mathbf{Q_{ke,air} = h*(T_{rec}-T_{\infty})}$$

where the recovery temperature is defined as

$$T_{rec} = T_{\infty}*(1+r*(\gamma-1)*M^2/2)$$

and the recovery factor r is

$$r = \sqrt{Pr} \text{ (Laminar) and } r = \sqrt[3]{Pr} \text{ (Turbulent)}$$

LEWICE 1.6 uses the local pressure instead of Mach Number, hence

$$\mathbf{T_{rec} = T_{\infty}*(1+r*((P/P_0)^{(1-\gamma)/\gamma}-1))}$$

is the form used in LEWICE 1.6.

A smaller amount of kinetic heating is imparted by the impinging drops,

$$Q_{ke,water} = m_{imp} * V_\infty^2/2$$

Convection Heat Loss

The heat lost by convection is simply

$$Q_{conv} = h*(T_s - T_\infty)$$

Total Energy Balance

The energy balance is the sum of the previously derived terms, taking into account the correct signs.

$$0 = L_v * h_c / c_{p,air} * (MW_{water} * MW_{air}) * L^{-2/3}) * (r_h * P_{v,\infty}/P_\infty - T_o/T_s * P_{v,s}/P_o * (P_o/P_e)^{(1/\gamma)})$$

$$+ k*(T_{rec} - T_s) / \sqrt{(\pi \alpha t)} + m_{imp} * V_\infty^2/2 + h*(T_{rec} - T_s) +$$

$$\{$$

$$(m_{rb,in} * T_{rb} + m_{imp} * T_\infty) * c_{p,water} - (m_{imp} + m_{rb,in}) * c_{p,water} * T_s$$

$$(m_{imp} + m_{rb,in}) * [(c_{p,ice} * T_{mp} * (1 - \rho_{ice}/\rho_{water}) + c_{p,ice} * \Delta T_m + L_f - c_{p,water} * T_{mp}) * (T_{mp} + \Delta T_m - T_s)/\Delta T_m] + (m_{rb,in} * T_{rb} + m_{imp} * T_\infty) * c_{p,water}$$

$$(m_{imp} + m_{rb,in}) * [c_{p,ice} * T_{mp} * (1 - \rho_{ice}/\rho_{water}) + c_{p,ice} * (T_{mp} + \Delta T_m - T_s) + L_f - c_{p,water} * T_{mp}] + (m_{rb,in} * T_{rb} + m_{imp} * T_\infty) * c_{p,water}$$

$$\}$$

where one of the three terms in the brackets is used depending on the phase regime. The procedure is to assume the mid-phase ($0 < N_f < 1$) form of the equations, then change the terms based on whether or not the assumption is correct. As the conduction term is time dependant, the actual form used is the integrated average value for that time step.

As this equation is non-linear with respect to temperature, an iterative solution is necessary.This is performed by a Newton-Raphson iteration procedure. The RHS of the above equation is labeled $f(T_s)$ and its derivative is $\partial f(T_s)/\partial T_s$. The predicted temperature at each iteration is

$$T_{s,new} = T_{s,old} - f(T_{s,old})/[\partial f(T_{s,old})/\partial T_{s,old}]$$

This iteration is repeated until the difference between $T_{s,new}$ and $T_{s,old}$ is sufficiently small.

7.6.1 Derivation of Mass Balance for LEWICE 1.6

The mass balance at each control volume is

$$m_{imp} + m_{rb,in} = m_{rb,out} + m_{freeze} + m_{shed} + m_{evap} + m_{remain}$$

The term m_{shed} is determined independently from the other terms based on a Weber number calculation. The amount of water shedding is assumed to encompass both water shedding and splashing, as both are considered to be controlled by the Weber number. The small amount lost by this mechanism has been correlated to qualitative observations of icing physics by Bill Olsen (9). No quantitative data is available is available for comparison.

The term m_{evap} is determined by the vapor pressure as derived earlier and hence is dependant only on the surface temperature, as long as the incoming mass flow rate exceeds the evaporation rate.

The term m_{remain} is the amount of unfrozen water which is not allowed to leave the control volume due to surface tension (Weber number) effects. This amount is determined independently, as long its value does not exceed the amount of unfrozen water available, $m_{rb,out}$. In that case, no water leaves the control volume ($m_{rb,out} = 0$) and the value of the term m_{remain} is determined by the equation below for $m_{rb,out}$.

The terms m_{freeze} and $m_{rb,out}$ are determined by the freezing fraction. Regardless of the final temperature or freezing fraction, the sum of these terms is

$$m_{rb,out} + m_{freeze} + m_{remain} = m_{imp} + m_{rb,in} - m_{shed} - m_{evap}$$

The individual amounts are determined by the freezing fraction

$$m_{freeze} = N_f * (m_{imp} + m_{rb,in} - m_{shed} - m_{evap})$$

and

$$m_{rb,out} + m_{remain} = (1 - N_f) * (m_{imp} + m_{rb,in} - m_{shed} - m_{evap})$$

As explained earlier, the term $\mathbf{m_{remain}}$ is presumed known independently of this computation as long as its value does not exceed the RHS of the above equation. In that case, $\mathbf{m_{rb,out}}$ = 0 and $\mathbf{m_{remain}}$ is determined by the above equation.

This procedure results in an explicit, marching-type solution for m_{freeze} and m_{rb},out as long as there is a definitive starting point (stagnation point) acquired from the flow solution. This should always be the case for 2D potential flow. For 2D Naviér-Stokes, if there is a recirculating flow multiple starting points are possible. This can be handled by the current methodology by modifying the program to allow for integration from each of the multiple points. If runback water is entering a control volume from both sides, there can be no runback out, hence mass which does not freeze would be accounted for in either m_{remain} or m_{shed}. For 3D flows, a more involved solution mechanism may be necessary. The equations would remain the same, only the solution mechanism would change.

7.6.2 Water Shedding

Qualitative observations of the icing process reveal that some water is lost due to ice shedding. The shedding occurs at regions with a high Weber number. Based on that evidence, a routine was added to eliminate a small amount of runback water based on Weber number. If the Weber number is below a critical value (a value of 500 is used currently) no water is shed. Above this point, the percent mass lost is equal to the percent difference in Weber number,

$$\% \text{ loss} = (W_e - W_{e,c})/W_e * 100\%$$

It should be emphasized that this relation is not based on any quantitative measurements of mass loss. The amount of mass lost using this criteria is slight, which matches the qualitative experimental observations.

7.6.3 Bead Height Calculation

The code will calculate the height of a bead of water by assuming that the bead assumes the shape of a partial volume of a sphere. This volume is expressed by

$$V = \frac{\pi}{3}R^3(2 - 3\cos\theta + \cos^3\theta)$$

where θ = Contact Angle.

The height b of the drop is b = R (1-cos θ).This height is compared to the height needed for the drop to flow. This will occur when the aerodynamic force on the drop exceeds the surface tension (W_e = 1). For steady flow, the aerodynamic force is (30)

$$F_{flow} = v_f\frac{dm}{dt} = \rho b v_f^2 = \rho b^3\left(\frac{F\tau_f}{\mu_f}\right)^2$$

where τ_f = shear stress, F= wetness factor, and the velocity is given by

$$v_f = \frac{F\tau_f}{2\mu_f}$$

Applying the W_e = 1 criteria yields

$$b > \left(\frac{4\sigma\mu^2}{\rho F^2\tau^2}\right)^{1/3}$$

The wetness factor is defined here as the ratio of the 'spread factor' at a given ambient temperature with the 'spread factor' at a 10°contact angle. At this and lower contact angles the surface is said to be completely wetted. The 'spread factor' (31) is a function of contact angle only, and is given by

$$S = \left(\frac{4\sin\theta\,(1+\cos\theta)}{(2+\cos\theta)\,(1-\cos\theta)}\right)^{1/3}$$

7.6.4 Hot Air Anti-Ice

An equation to predict anti-icing performance is obtained by assuming: there is no lateral conduction in the airfoil; there is a continuous supply of anti-ice air; and, the internal heat transfer coefficients are known by the user. The user must also supply a desired surface temperature. The first step is to substitute the desired surface temperature into the icing heat balance. A more detailed analysis are presented in the LEWICE/Thermal code (32) and the ANTICE (30) code. This yields

$$\mathbf{q_s = L_v * h_c * MW_{water}/(P_o * c_{p,air} * MW_{air} * L^{2/3}) * (P_{v,e}/P_e * P_o - r_h * T_o * P_{v,s}/T_s * (P_o/P_e)^{(1/\gamma)}) + m_{imp} * V_\infty^2/2 + h * (T_{rec} - T_s) + m_{imp} * c_{p,water} * (T_\infty - T_s)}$$

where q_s is the heat which needs to be supplied at the surface to achieve the desired temperature. Note that since a constant surface temperature is assumed that there is no transfer of heat from one control volume to the next due to runback water flow. This system computes the heat requirements for a 'running wet' system, not an evaporative system. The temperature of the hot air is determined by a steady-state 1-D heat transfer analysis and is given by

$$T_{air} = T_s + q_s\left[\frac{1}{h_o} + \sum_{i=1}^{n}\frac{\Delta x_i}{k_i}\right]$$

where Δx is the thickness of each material and k is its thermal conductivity. Both variables are input by the user for each layer.

7.6.5 Anti-Ice with Internal Heat Source

An internal heat source such as an electrothermal heater can also be used to anti-ice the airfoil. An approximation to the heat requirements for an anti-icer from an internal heat source can also be obtained. The formulation for this model is slightly more complex as the location of the internal heat source can vary depending upon design.

The derivation starts by writing the differential equation for the heater layer,

$$\frac{dq}{dx} = q'''$$

where q''' is the volumetric heat source. Integrating this with the limits of $q = q_0$ at $x = x_k$ and $q = q_1$ at $x = x_{k+1}$ gives

$$q = q_0 + (x - x_k)\frac{q_1 - q_0}{x_{k+1} - x_k}$$

where q_1 and q_0 are related by

$$q_1 = q_0 + q'''(x_{k+1} - x_k)$$

For every other layer in the heater mat and airfoil, there is no heat source, hence dq/dx = 0 thus the heat flux through the layer is constant. For layers below the heater, this constant must be q_0 while for those above the heater the constant is q_1.

The constant q_1 can be calculated using the same equation as that used for the hot air system, as this represents the heat loss at the surface. The surface temperature is once again input by the user. The equation for q_1 is

$$q_1 = L_v * h_c * MW_{water}/(P_o * c_{p,air} * MW_{air} * L^{2/3}) * (P_{v,e}/P_e * P_o - r_h * T_o * P_{v,s}/T_s * (P_o/P_e)^{(1/\gamma)}) + m_{imp} * V_\infty^2/2 + h * (T_{rec} - T_s) + m_{imp} * c_{p,water} * (T_\infty - T_s)$$

The solution to this problem also requires knowledge of the inside surface heat transfer coefficient and the inside air temperature. The program assumes the inside air temperature to be the same as the outside air temperature and the inside heat transfer coefficient to be a minimal value, which assumes free convection. The solution then proceeds as follows: let T_h be the temperature at the bottom of the heater layer, which is layer h of n total layers and T_{h+1} be the temperature at the top of the heater layer. Using the fact that the heat flux through each of the layers below the heater is q_0 and the heat flux through each of the layers above the heater is q_1, the temperatures T_h and T_{h+1} are given by

$$T_h = T_\infty - q_0\left(\frac{1}{h_0} + \sum_{j=1}^{h-1}\frac{x_{j+1} - x_j}{k_j}\right)$$

$$T_{h+1} = T_s - q_1\sum_{j=h+1}^{n}\frac{x_{j+1} - x_j}{k_j}$$

Since q_1 has already been found, T_{h+1} can be calculated from this equation. Furthermore, the equation for the heat flux in the heater layer can be integrated to relate the two temperatures T_h and T_{h+1}. This yields

$$T = q_0 x + \left(\frac{x^2}{2} - x_h x\right)\frac{q_1 - q_0}{x_{h+1} - x_h} + c$$

Applying the boundary condition of $T = T_{h+1}$ at $x = x_{h+1}$ gives

$$c = T_{h+1} - \frac{(q_0 + q_1)}{2}\frac{x^2_{h+1}}{x_{h+1} - x_h} + x_h x_{h+1}\frac{q_1}{x_{h+1} - x_h}$$

The temperature at $x=x_h$ is then

$$T_h = T_{h+1} + \left(\frac{x_h + x_{h+1}}{2}\right)\frac{(q_1 + q_0)}{k_h}$$

There are now two equations which describe the two unknowns T_h and q_0. Solving these equations yields

$$q_0 = \frac{T_\infty - T_s + q_1\left(\sum_{j=h+1}^{n} \frac{x_{j+1}-x_j}{k_j} - \frac{x_{h+1}+x_h}{2}\right)}{\left(\frac{1}{h_0} + \sum_{j=1}^{h-1} \frac{x_{j+1}-x_j}{k_j} + \frac{x_{h+1}+x_h}{2}\right)}$$

The computational procedure is to first compute q_1, then q_0, q''', T_{h+1} and T_h in that order. The heater wattage is normally desired in units of W/in^2, so the output parameter of heat requirement is actually $q'''(x_{h+1} - x_h)$ which is then converted to the desired units. Either T_h or T_{h+1} must be the maximum temperature, thus the program compares the two and outputs the maximum heater temperature.

7.6.6 Evaporative Anti-Icer

For an evaporative system, the same equations can be applied, but the solution process is different. In this case, $m_{imp} = m_{evap}$ and the surface temperature is calculated by iteratively solving the following equation:

$$h_c/c_{p,air}*(MW_{water}/MW_{air})*L^{-2/3}) * (P_{v,\infty}/P_\infty*r_h - P_{v,s}/P_o*T_o/T_s*(P_o/P_e)^{(1/\gamma)}) - m_{imp} = 0$$

The process is iterative, as $P_{v,s}$ is a non-linear function of T_s. Once the surface temperature is known, it is substituted into one of the earlier heat balance equations (depending upon which type of anti-icer was selected) to find the required heat flux and the maximum temperature.

Ice Density

The current correlation for ice density was developed from experimental data on iced cylinders (33, 34). Its form is

$$\rho_i = 1000 exp\left(0.15\left(1 + \frac{6043}{S^{2.65}}\right)\right)$$

where S is a dummy parameter given by

$$S = \frac{MVD^{0.82} V^{0.59} LWC^{0.21}}{D^{0.48} (-T_c)^{0.23}}$$

and D = cylinder diameter (LEWICE 1.6 uses the diameter of the inscribed circle at the leading edge); T_c = surface temperature in degrees Celsius.

The model given above can predict very low ice densities which might be undesirable, hence a second correlation was produced which is based on freezing fraction. For a freezing fraction of 1, an ice density of 820 kg/m^3 is produced. For a freezing fraction of 0, a glaze ice density of 917 kg/m^3 is produced. For other freezing fractions, the density is interpolated from these two limits. Future experiments are planned to improve on these model.

8.0 Input Files

This section will define the variables in the input files to LEWICE 1.6. Several notes are also added to aid the user in properly setting up an input file for this code.

8.1 Line 1

Line 1 is the title you assign to the run. This title will appear in the output file MISC.DAT

8.2 LEW16 Namelist

The LEW16 namelist is a collection of new inputs for version 1.6, or variables which were previously input at the terminal.

IACCFL: IACCFL is an accuracy flag. If IACCFL=0 then normal collection efficiency and flow will be run. This mode is very quick but the least accurate. If IACCFL=1 then small equal segment lengths will be run throughout the leading edge region. This means a lot more panels and larger run times but more symmetry and accuracy. If IACCFL=2 then small equal segment lengths will be run throughout the leading region and additionally the panel centered collection efficiency routine will be run. This routine iterates on each panel within the calculated impingement limits to try and find a trajectory that hits its midpoint. This produces very symmetry collection efficiecies but is expensive. Two hundred trajectories is not uncommon.

ITIMFL: ITIMFL is a time stepping flag. If ITIMFL=0 then number of time steps must be input and the time steps will be of equal length. If ITIMFL=1 then an auto time step will be calculated and the time step will vary. If ITIMFL=1 then the ice time step is controlled by limiting the thickness of ice to be added at each time step. If ITIMFL=1 then the maximum number of steps to be calculated in the ice addtion routine (NSTEPI) must be input.

NSTEPI: NSTEPI is related to the refined grid surface spacing. this input is ignored unless ITIMFL=1. If NSTEPI=1 then the largest amount of ice to be added will be equal to the refined geometry segment length (i.e. DSMIN). If NSTEPI=2 then the largest amount of ice to be added will be equal to twice the refined geometry segment length (i.e. 2*DSMIN). A good value for NSTEPI is 4. This method essentially limits the ice growth to a certain percentage of chord. For NSTEPI=4 and DSMIN=0.0005*chord this is about 0.002*chord or 0.2% chord

TSTOP: TSTOP is the total time of the icing simulation in seconds.

IFLO: IFLO is the number of time steps to be used in the simulation. If ITIMFL=1, then this input is ignored, except in the roughness prediction model. In this routine, TSTOP/IFLO gives the time increment for roughness growth for the first time step only.

IMLSFL: IMLSFL is a flag to insure transition is occuring outside of the multiple stagnation point region. If IMLSFL=1 then transtion will be delayed until outside of all stagnation points If IMLSFL=0 then normal LEWICE 1.6 transition will be run.

ISTC: ISTC is a trajectory speed up flag if one standard collection efficiency scheme is run. If ISTC=0 then particles are released closer to the body in the collection efficiency routine giving improved speed with no loss in accuracy.

DSMN: DSMN is the minimum size of the control volumes (non-dimensionalized). It is also tied indirectly to the number of panels produced for the flow solution. Larger values create fewer control volumes/panels while smaller values create more control volumes/panels.

Note: in version 1.6 the number of control volumes will be much greater than the number of panels. The default value for DSMN is $5*10^{-4}$.

DSMAX: DSMAX is the maximum allowable control volume size. The default value is 10^{-3}.

DDANG: DDANG is the maximum allowable angle between control volumes. Smaller values produce more control volumes while larger values create fewer control volumes.

IDEN: IDEN is a flag used to select an ice density model. IDEN=0 uses a correlation based on rotating cylinders. IDEN=1 uses a correlation based on freezing fraction. These models are described in more detail later in this manual. Default is IDEN=0.

IRUNB: IRUNB is a flag used to control runback. If IRUNB=0, standing water which does not runback is carried over to the next time step. If IRUNB=1, this

mass is removed. Default is IRUNB=0 and is highly recommended.

NFLIT: NFLIT is a flag used to produce a smoother velocity distribution near the stagnation point. If NFLIT=0, then no adjustments are done. If NFLIT=1, this forces IMLSFL=1 and produces better heat transfer coefficients in the stagnation region. Default is NFLIT=1.

IGRID: IGRID is a flag which controls creation of a grid for the trajectory routine. If IGRID=0, off-body air velocities are determined directly from the potential flow solution. If IGRID=1, the program will create a grid, determine the off-body velocities at the grid points and then interpolate from these points to find the velocity at the desired point.

Note: This flag is useful if you want to use a grid-based flow solution as your input and bypass the potential flow solution.

IRC: IRC is a flag which chooses the type of grid to be used. IRC=0 creates a rectangular grid and IRC=1 creates a 'C' grid.

Notes: This input is ignored unless IGRID=1. Also, this flag is superceded if you are using a separate grid-based flow code.

SLWC: This is the time needed to 'ramp up' to the desired LWC in the IRT. Typical values are less than 20 seconds. If you are running a short spray time in the IRT, this parameter can become important. Consult with the tunnel operators for more accurate values for this input if needed.

NREAD: NREAD determines which file the body geometry is read from. If NREAD=0, coordinates are read from units 44-53. Unit 44 is used for the first body, unit 45 is used for the second body, etc. The program will prompt you for the filenames to be used for these units. If NREAD=1, coordinates are read from the main input file, unit 35.

The format for NREAD=0 is two spaces, followed by an x-coordinate which is at most 12 spaces long, followed by two more spaces and the y-coordinate which is at most 12 spaces long. The format is fixed, meaning exponential number format cannot be used. If NREAD=1, the number of points is on the first line. Following lines contain x,y coordinate pairs which are free-format.

Notes: Coordinates must be non-dimensionalized, start at the trailing edge then proceed along the bottom surface toward the leading edge, then back to the trailing edge along the top surface (clockwise input). The first and last points must be the same.

IBOD: IBOD is the number of bodies to be simulated. For example, a three body simulation can consist of a slat, main and flap. However, multi-body simulations are not limited to this example.

LBOTH: LBOTH is a flag which determines the thoroughness of the trajectory search and is used only for multibody simulations. If LBOTH=0, the code will look for impingement on the 2nd through 10th bodies assuming the trajectories which hit it travel beneath bodies which precede it. If LBOTH=1, it will look for two sets of impingement limits. The first set will look below the first body and the second set will look above the first body.

Note: Except for very unusual configurations, trajectories which hit the main body passed beneath the slat, not from above and almost never from both directions. Selecting LBOTH=0 will, in most cases, make the code run twice as fast as selecting LBOTH=1 with no loss of accuracy. LBOTH=0 is default.

8.2.1 Example LEW16 Namelist

```
 &LEW16
 IACCFL =    1
 ITIMFL =    1
 TSTOP  =  300.
 IFLO   =    1
 IMLSFL =    0
 ISTC   =    1
 NSTEPI =    4
 DSMN   =    5.D-4,5.D-4,5.D-4,5.D-4,
5.D-4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
 DSMAX  =    1.D-3
 DDANG  =    5.D-4
 IDEN   =    0
 IRUNB  =    0
 NFLIT  =    1
 IGRID  =    0
 IRC    =    1
 SLWC   =    0.0
 NREAD  =    0
 IBOD   =    3
 LBOTH  =    0
```

```
&END
```

Note: DSMN values should all be on the same line in your input file.

8.3 S24Y1 Namelist

The S24Y namelist defines input to the potential flow subroutine.

ILIFT: If ILIFT=0, this is not a lifting body. If ILIFT=1 (default), it is a lifting body.

IPARA: If IPARA=0, linear panels are used. If IPARA=1 (default), parabolic panels are used

IFIRST: If IFIRST=0, no first order terms are used. If IFIRST=1, first derivative term is used. If IFIRST=2, curvature term is used and if IFIRST=3 (default), both terms are used.

ISECND: If ISECND=0, no first order terms are used. If ISECND=1, second derivative term is used. If ISECND=2, curvature term is used and if ISECND=3 (default), both terms are used.

IPVOR: If IPVOR=0, a constant vorticity is used between body elements. If IPVOR=1 (default) a variable vorticity distribution is used.

INCLT: If INCLT=0 (default), then CLT is the angle of attack. If INCLT=1, then CLT is the total lift coefficient.

CLT: CLT is the value for angle of attack or lift coefficient, depending on the input for INCLT.

CCL: CCL is the reference length used in calculating lift coefficient.

Note: The program in many cases assumes that the x,y coordinates are non-dimensionalized, thus CCL=1.0. See the section on User Tips for the procedure for using dimensionalized coordinates.

IFLLL: If IFLLL =0, results of the parabolic integration are not printed. If IFLLL=1 (default), these results are printed.

8.3.1 Example S24Y Namelist

```
&S24Y1
ILIFT  =    1
IPARA  =    1
IFIRST =    3
ISECND =    3
IPVOR  =    1
INCLT  =    0
CLT    =  0.0
CCL    =  1.0
IFLLL  =    1
&END
```

8.4 Body Geometry Input

As described earlier in the NREAD input, the body geometry(s) will be read from the main input file if NREAD=1 and from separate file(s) if NREAD=0. If NREAD=1, those coordinates must be placed in-between the S24Y1 namelist and the TRAJ1 namelist. The format for NREAD=0 is 2(2X,F12.7) for the x,y coordinates, while for NREAD=1, the coordinates are free-format, and the number of points is also provided.

8.4.1 NREAD=0 Example

```
0.22017      0.042322
0.21501      0.040958
0.20997      0.039232
0.20505      0.037208
0.20020      0.034955
0.19543      0.032527
0.19070      0.029957
0.18601      0.027279
0.18135      0.024521
0.17668      0.021797
0.17202      0.019036
0.16737      0.016266
0.16272      0.013486
0.15807      0.010708
0.15340      0.0079681
0.14872      0.0052843
0.14400      0.0026744
0.13925      0.00016658
0.13447     -0.0022393
0.12965     -0.0045611
0.12480     -0.0067850
0.11992     -0.0089328
0.11501     -0.011003
0.11007     -0.012997
0.10511     -0.014920
0.10012     -0.016774
0.095111    -0.018564
0.090080    -0.020298
0.085029    -0.021980
0.079959    -0.023606
0.074866    -0.025171
```

```
0.069751    -0.026675
0.064608    -0.028101
0.059440    -0.029459
0.054234    -0.030715
0.048987    -0.031855
0.043682    -0.032837
0.038306    -0.033622
0.032813    -0.034090
0.029992    -0.034116
0.027086    -0.033912
0.024032    -0.033300
0.021994    -0.032368
0.021024    -0.031568
0.019928    -0.029802
0.019492    -0.027358
0.019581    -0.025736
0.020244    -0.022892
0.021854    -0.019536
0.023707    -0.016850
0.025710    -0.014574
0.030021    -0.010860
0.034532    -0.0076986
0.039152    -0.0048347
0.043846    -0.0021729
0.048591     0.00034696
0.053379     0.0027508
0.058202     0.0050567
0.063058     0.0072705
0.067943     0.0094064
0.072855     0.011468
0.077793     0.013460
0.082756     0.015382
0.087741     0.017242
0.092753     0.019030
0.097787     0.020755
0.10285      0.022409
0.10794      0.023977
0.11306      0.025467
0.11821      0.026863
0.12340      0.028183
0.12861      0.029411
0.13385      0.030572
0.13912      0.031654
0.14442      0.032660
0.14974      0.033600
0.15508      0.034482
0.16044      0.035314
0.16581      0.036106
0.17120      0.036865
0.17659      0.037609
0.18202      0.038229
0.18746      0.038847
0.19290      0.039451
0.19835      0.040043
0.20380      0.040628
0.20925      0.041216
0.21471      0.041782
0.22017      0.042322
```

As the S24Y1 example gives IBOD=3, the case selected contains three bodies. The coordinates listed above are for the first body and are read from unit 44. The following coordinate set is read from unit 45.

```
0.90000    -0.001652
0.88250    -0.000056
0.86500     0.000273
0.84750    -0.000441
0.83000    -0.001960
0.81250    -0.004088
0.79500    -0.006713
0.77750    -0.009716
0.76000    -0.012999
0.74250    -0.016163
0.72500    -0.019460
0.70750    -0.022785
0.69000    -0.026145
0.67250    -0.029498
0.65500    -0.032718
0.63750    -0.035742
0.62000    -0.038507
0.60250    -0.040915
0.58500    -0.042966
0.56750    -0.044723
0.55000    -0.046137
0.53250    -0.047285
0.51500    -0.048160
0.49750    -0.048769
0.48000    -0.049133
0.46250    -0.049252
0.44500    -0.049147
0.42750    -0.048846
0.41000    -0.048363
0.39250    -0.047684
0.37500    -0.046795
0.35750    -0.045689
0.34000    -0.044310
0.32250    -0.042693
0.30500    -0.040719
```

```
0.28750    -0.038339
0.27000    -0.035406
0.25250    -0.031787
0.23500    -0.027055
0.22625    -0.023961
0.21750    -0.020062
0.20875    -0.014735
0.20350    -0.009562
0.20140    -0.005999
0.20000     0.000693
0.20140     0.008736
0.20350     0.013650
0.20875     0.021693
0.21750     0.030254
0.22625     0.036470
0.23500     0.041251
0.25250     0.047880
0.27000     0.052577
0.28750     0.056231
0.30500     0.059178
0.32250     0.061628
0.34000     0.063672
0.35750     0.065373
0.37500     0.066752
0.39250     0.067858
0.41000     0.068705
0.42750     0.069307
0.44500     0.069664
0.46250     0.069804
0.48000     0.069692
0.49750     0.069363
0.51500     0.068782
0.53250     0.067900
0.55000     0.066745
0.56750     0.065261
0.58500     0.063511
0.60250     0.061439
0.62000     0.059136
0.63750     0.056553
0.65500     0.053704
0.67250     0.050624
0.69000     0.047341
0.70750     0.043883
0.72500     0.040285
0.74250     0.036575
0.76000     0.032809
0.77750     0.028609
0.79500     0.024402
0.81250     0.020146
0.83000     0.015848
0.84750     0.011529
0.86500     0.007217
0.88250     0.002828
0.90000    -0.001652
```

Finally, the coordinates of the third body are read from unit 46.

```
1.1496      -0.19391
1.1425      -0.18890
1.1351      -0.18443
1.1274      -0.18040
1.1195      -0.17672
1.1115      -0.17331
1.1033      -0.17010
1.0951      -0.16706
1.0868      -0.16414
1.0785      -0.16116
1.0702      -0.15824
1.0619      -0.15534
1.0535      -0.15245
1.0452      -0.14956
1.0369      -0.14661
1.0287      -0.14357
1.0205      -0.14043
1.0124      -0.13713
1.0044      -0.13368
0.99643     -0.13010
0.98858     -0.12638
0.98080     -0.12254
0.97308     -0.11858
0.96543     -0.11451
0.95784     -0.11034
0.95031     -0.10606
0.94284     -0.10168
0.93541     -0.097226
0.92803     -0.092689
0.92070     -0.088067
0.91342     -0.083356
0.90619     -0.078552
0.89904     -0.073631
0.89194     -0.068608
0.88492     -0.063432
0.87801     -0.058082
0.87124     -0.052495
0.86463     -0.046613
0.85831     -0.040255
0.85532     -0.036764
0.85254     -0.032928
0.85010     -0.028480
0.84913     -0.024964
```

```
0.84912     -0.022917
0.85017     -0.019703
0.85276     -0.016602
0.85488     -0.015016
0.85912     -0.012868
0.86499     -0.011364
0.87027     -0.010865
0.87521     -0.010981
0.88435     -0.012470
0.89301     -0.014788
0.90141     -0.017552
0.90964     -0.020619
0.91775     -0.023899
0.92575     -0.027353
0.93368     -0.030954
0.94152     -0.034693
0.94929     -0.038549
0.95700     -0.042517
0.96465     -0.046589
0.97224     -0.050766
0.97977     -0.055036
0.98724     -0.059414
0.99466     -0.063885
1.0020     -0.068464
1.0093     -0.073172
1.0165     -0.077998
1.0236     -0.082964
1.0307     -0.088044
1.0377     -0.093262
1.0446     -0.098579
1.0515     -0.10402
1.0583     -0.10957
1.0650     -0.11522
1.0717     -0.12095
1.0784     -0.12677
1.0850     -0.13264
1.0915     -0.13856
1.0981     -0.14450
1.1046     -0.15063
1.1110     -0.15677
1.1175     -0.16292
1.1239     -0.16909
1.1304     -0.17527
1.1368     -0.18145
1.1432     -0.18766
1.1496     -0.19391
```

LEWICE 1.6 will prompt you for the file names of these units. The coordinates listed represent the coordinates of the example case which is sent out with the code. Each body is a MS-317 airfoil cross-section. They have been rotated to look like a multi-element airfoil.

Note: In this example, each body has 89 points on input. It is NOT necessary for each body to have the same number of points. This example does, because each section was created from a MS-317 input file which had 89 points.

8.4.2 NREAD=1 Example

First Body

89

```
0.22017      0.042322
0.21501      0.040958
0.20997      0.039232
   .            .
   .            .
   .            .
```

(points same as earlier example and not duplicated in their entirety)

Second Body

89

```
0.90000    -0.001652
0.88250    -0.000056
0.86500     0.000273
   .            .
   .            .
   .            .
```

(points same as earlier example and not duplicated in their entirety)

Third Body

89

```
1.1496      -0.19391
1.1425      -0.18890
1.1351      -0.18443
   .            .
   .            .
   .            .
```

(points same as earlier example and not duplicated in their entirety)

These coordinates are all read from the main input file, unit 35.

8.5 TRAJ1 Namelist

The TRAJ1 namelist inputs variables required by the trajectory routines in LEWICE 1.6.

GEPS: GEPS is the convergence criteria for the particle trajectory integration. Smaller values will increase accuracy at the cost of additional computational time. The default value is $5*10^{-5}$.

VEPS: VEPS is the accuracy criteria for the freestream velocity comparison. For accuracy, drops are released far enough away from the airfoil such that the local velocity is within VEPS of the freestream value. The default value is 10^{-3}.

NEQ: NEQ defines the number of equations which must be solved in the particle trajectory routine. Valid inputs are 4 and 6. If NEQ=4, the particle is spherical and non-lifting. If NEQ=6, the particle is assumed to be rotating, which gives it some lift. The default value is 4.

NPL: NPL is the number of particle trajectories (not including the impingement limit values) which define the collection efficiency distribution.

Note: The actual number which the code uses may be different than this value. The code is limited to one trajectory strike per panel, so if more than one trajectory hits a given panel, it will be eliminated.

NSEAR: NSEAR is the maximum number of trajectories which are allowed for the computation of the upper and lower impingement limits. The default value is 199, which is more than sufficient for most cases. If the code exits and gives an error message stating that more than 199 are required, check to see that all of your input is accurate. If it is, contact the NASA Lewis Icing Branch to report the bug.

TSP: TSP is the initial timestep used in the trajectory integration. A smaller value will force the code to take more steps to reach the airfoil, while larger values will reduce the number of timesteps. There is a limit of 1000*IBOD timesteps for a given trajectory. The default value is 10^{-3}.

CHORD: CHORD is the distance from the leading edge to the trailing edge in meters. For a cylinder, this represents the cylinder diameter. For airfoils, it is the standard chord length.

G: G is the value for gravity in m/s^2. For cases which have been run so far, there are no discernible changes to the droplet trajectories due to gravity.

PIT: PIT is the initial particle pitch angle, and should be zero for non-rotating spheres, which is the default value for the code.

PITDOT: PITDOT is the first derivative of the pitch angle with respect to time, and should also be zero for non-rotating spheres.

YOLIM: YOLIM is the accuracy criteria for determining if an impingement limit has been found. It is the distance between two particles at their initial value. If two trajectories are released which are initially less than YOLIM apart, and one hits the airfoil while the other misses, the trajectory which hits is the impingement limit. Smaller values cause more trajectories to be required, while larger values need fewer trajectories. Values which are much smaller than the default value of $5*10^{-5}$ may cause the program to exceed NSEAR trajectories and stop.

RHOP: RHOP is the density of the water particle in kg/m^3. This has been placed in the input file to broaden the utility of this code to industry. As LEWICE 1.6 assumes water particles to be hard spheres, the physics of water droplet trajectories is the same as for sand particle trajectories. The only required change is the density of the particle, as sand has different properties than water. If sand density is substituted, the code can be used to predict deposition of sand (sand collection efficiency). In this mode, the ice accretion results should be ignored, and the code can be run using a single time step, as the geometry is not changing.

8.5.1 Example TRAJ1 Namelist

```
&TRAJ1
GEPS   = 0.5E-04
VEPS   = 1.E-03
NEQ    =    4
NPL    =   24
NSEAR  =  199
TSP    = 1.E-03
CHORD  = 0.9143
G      = 0.0
PIT    = 0.0
PITDOT = 0.0
YOLIM  = 0.5E-04
```

```
RHOP    = 1.0E3
&END
```

8.6 DIST Namelist

The DIST namelist defines the particle size and distribution. For each variable, there are 10 possible values, as the code can handle up to a 10 drop size distribution.

FLWC: FLWC is the fraction of the total liquid water content contained in each drop size.

DPD: DPD is the size, in microns, of the water drops. If only one size is input, it is the MVD (median volume droplet).

CFP: CFP is the Cunningham Correction Factor, which is used for small particle sizes. If a particle size is smaller than 10 μm, the drag on the particle has been reduced due to slip flow. This correction factor is used to correct the computed collection efficiency for this effect. The following is a list of this correction factor for different drop sizes.

Diameter (μm)	Correction Factor
1	1.168
2	1.084
3	1.056
4	1.042
5	1.034
6	1.028
7	1.024
8	1.021
9	1.019
10	1.017

A more complete list (for smaller particles) can be found in the original LEWICE manual (1), and in a paper by Carlson and Haglund[9].

8.6.1 Example DIST Namelist

```
 &DIST
 FLWC   = 0.05, 0.1, 0.2, 0.3, 0.2, 0.1,
0.05, 0.0, 0.0, 0.0
 DPD    = 6.2, 10.4, 14.2, 20.0, 27.4,
34.8, 44.4, 0.0, 0.0, 0.0
 CFP    = 1.0272, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 0.0, 0.0, 0.0
 &END
```

Note: The above input should all be on the same line for a given variable. The example provided is a Langmuir 'D' drop size distribution, with an MVD of 20 μm.

8.7 ICE1 Namelist

The ICE1 namelist provides the meteorological conditions of the icing simulation.

VINF: VINF is the velocity 'at infinity' (the flight speed) in m/s.

LWC: LWC is the liquid water content in g/m^3.

TINF: TINF is the static temperature 'at infinity' (i.e., ambient condition).

Note: The data supplied to researchers is often the total temperature, not the static temperature. Make certain the value input is correct!

PINF: PINF is the static pressure 'at infinity' (i.e., ambient condition) in Pascals (N/m^2).

RH: RH is the relative humidity in%. This is normally assumed to be 100%, unless the actual value is known.

TIME: TIME is the starting time for the simulation in seconds. If for some reason, the code stops prematurely, the simulation can be restarted by constructing an input file which consists of the last ice shape computed as the input geometry and the time of that shape input here.

8.8 Example ICE1 Namelist

```
&ICE1
VINF    = 89.5
LWC     = 0.34
TINF    = 269.
PINF    = 94540.0
RH      = 100.0
TIME    = 0.0
&END
```

8.9 LPRNT Namelist

The LPRNT namelist controls print options for LEWICE 1.6. Users can limit the amount of printout which saves space and computation time.

IPRT: If IPRT=0, the ice shape (geometry) will not be printed. If IPRT=1, it will be printed

FPRT: If FPRT=0, the flow solution (units 28, 29) will not be printed. IfFPRT=1, it will be printed

HPRT: If HPRT=0, the heat transfer coefficients will not be printed. If HPRT=1, it will be printed

BPRT: If BPRT=0, the collection efficiencies will not be printed. IfBPRT=1, it will be printed

EPRT: If EPRT=0, the energy balance output (units 40, 43) will not be printed. IfEPRT=1, it will be printed

MPRT: If MPRT=0, the mass balance output units 30, 33, 38, 41) will not be printed. IfMPRT=1, it will be printed

TPRT: If TPRT=0, the trajectories will not be printed. If TPRT=1, all trajectories will be printed. If TPRT=2, only trajectories which hit will be printed.

IRPRT: If IRPRT=0, the iris plotting files (86, 87, 88, 89, 90, 93) will not be printed. If IRPRT=1, it will be printed

8.9.1 Example LPRNT Namelist

```
 &LPRNT
 IPRT   =1
 FPRT   =1
 HPRT   =1
 BPRT   =1
 EPRT   =1
 MPRT   =1
 TPRT   =1
 IRPRT  =1
 &END
```

8.10 Complete Example Case Input File

```
 &LEW16
 IACCFL =    1
 ITIMFL =    1
 TSTOP  =  300.
 IFLO   =    1
 IMLSFL =    0
 ISTC   =    1
 NSTEPI =    4
 DSMN   =    5.D-4,5.D-4,5.D-4,5.D-4,
5.D-4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
 DSMAX  =    1.D-3
 DDANG  =    5.D-4
 IDEN   =    0
 IRUNB  =    0
 NFLIT  =    1
 IGRID  =    0
 IRC    =    1
 SLWC   =    0.0
 NREAD  =    0
 IBOD   =    3
 LBOTH  =    0
 &END
 &S24Y1
 ILIFT  =    1
 IPARA  =    1
 IFIRST =    3
 ISECND =    3
 IPVOR  =    1
 INCLT  =    0
 CLT    =   0.0
 CCL    =   1.0
 IFLLL  =    1
 &END
 &TRAJ1
 GEPS   = 0.5E-04
 VEPS   = 1.E-03
 NEQ    =    4
 NPL    =   24
 NSEAR  =  199
 TSP    = 1.E-03
 CHORD  = 0.9143
 G      = 0.0
 PIT    = 0.0
 PITDOT = 0.0
 YOLIM  = 0.5E-04
 RHOP   = 1.0E3
 &END &DIST
 FLWC  = 0.05, 0.1, 0.2, 0.3, 0.2, 0.1,
0.05, 0.0, 0.0, 0.0
 DPD   = 6.2, 10.4, 14.2, 20.0, 27.4,
34.8, 44.4, 0.0, 0.0, 0.0
 CFP    = 1.0272, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 0.0, 0.0, 0.0
 &END &ICE1
 VINF   = 89.5
 LWC    = 0.34
 TINF   = 269.
 PINF   = 94540.0
 RH     = 100.0
 TIME   = 0.0
 &END
 &LPRNT
 IPRT   =1
 FPRT   =1
 HPRT   =1
 BPRT   =1
 EPRT   =0
```

```
MPRT    =0
TPRT    =2
IRPRT   =0
&END
```

8.11 Deice Input File (Unit 19)

LEWICE 1.6 can perform a simplistic, pseudo 2D simulation of a thermal de-icer/anti-icer. A more complex, and thus more computationally expensive simulation can be provided by the LEWICE/Thermal code (10) and the ANTICE code (11). The analysis performed by LEWICE 1.6 is provided in another section of this manual. This section will describe the variables input.

Number of Layers: This is the number of different materials in the direction normal to the airfoil surface. The input is a 2-digit number which is placed after the equals sign, allowing one space after the equals sign. The limit is 50 layers.

Length: This is the thickness of the material in meters.

Note: The layers are input with the inner surface first, and the outer surface last.

Conductivity: This is the thermal conductivity of each material in W/m°K.

Number of Heat Transfer Coefficients: This tells the code how many heat transfer coefficients are to be read in. These values are the interior heat transfer coefficients, and are only used when modelling a hot-air design. External heat transfer coefficients are predicted by LEWICE 1.6. The format gives 4 digits as shown in the example, provided one space after the equals sign. The maximum input is therefore 9999.

S/C: S/C is the wrap distance as measured from the leading edge and is non-dimensionalized by the chord.

HTC: HTC is the interior heat transfer coefficient in $W/m^2{}^\circ K$ and should not be confused with the LEWICE 1.6 calculated heat transfer coefficient.

8.11.1 Example Deicer Input (Unit 19)

```
NUMBER OF LAYERS = 13
     LENGTH        CONDUCTIVITY
       (M)           (W/M*K)
 4.4450000E-03 1.7653000E+02
 2.5400000E-04 1.7307000E-01
 1.2700000E-03 1.7653000E+02
 2.5400000E-04 1.7307000E-01
 5.0800000E-04 1.5057000E+01
 2.0800000E-04 1.7307000E-01
 3.5100000E-04 3.8075000E-01
 2.0800000E-04 1.7307000E-01
 1.6500000E-04 1.0384000E+02
 2.0800000E-04 1.7307000E-01
 3.5100000E-04 3.8075000E-01
 4.2700000E-04 1.7307000E-01
 7.6200000E-04 1.5057000E+01
   NUMBER OF HEAT TRANSFER COEFFICIENTS
INPUT = 0015
       S/C          HTC(W/M*M*K)
-0.5500000E+00 1.0000000E+01
-0.4000000E+00 1.0000000E+01
-0.3000000E+00 1.5000000E+01
-0.2000000E+00 2.0000000E+01
-0.1500000E+00 3.0000000E+01
-0.1000000E+00 5.0000000E+01
-0.0500000E+00 8.0000000E+01
 0.0000000E+00 1.0000000E+02
 0.0500000E+00 8.0000000E+01
 0.1000000E+00 5.0000000E+01
 0.1500000E+00 3.0000000E+01
 0.2000000E+00 2.0000000E+01
 0.3000000E+00 1.5000000E+01
 0.4000000E+00 1.0000000E+01
 0.5500000E+00 1.0000000E+01
```

Note: Value for Number of Heat Transfer Coefficients Input is on the same line as comment.

8.12 Interactive Input

This section will describe the input required by the code during the run.

Enter Input File Name: This is the prompt asking for the name of unit 35, the main input file.

Case Studies: LEWICE 1.6 is capable of handling up to 10 case studies per run. This option is useful if the user wants to perform a parameter study. This means that the code will run (up to) 10 independent cases where all of the input is the same except for the input variable. The variables which can be parameterized in this manner are: TINF, LWC, PINF, CLT, MVD, RH, and DTIME. These are ambient temperature, liquid water content, ambient pressure, angle of attack, drop size (single size distribution only), relative humidity and time step (constant time

step mode only). If the value input is more than 1, the code will prompt you to define which variable you want to parameterize, and the values for that variable. This will supecede the value given in the input file for that variable. The interactive input is meant to be self-explanatory.

Interactive Deicer Input: The code will ask if you want to perform the deicing analysis. If you answer N, then you do not have to have a deicer input file, and the code will not ask for further information. If you answer Y, the code will prompt you for the desired surface temperature in °K. A survey of users has shown this to be a good design parameter, and answers the question "How much heat do I need to keep the surface at X temperature?".

The next question determines if you want to model an evaporative system. This is also a common question of users "What is the minimum required heat needed to evaporate everything?"

Note: Due to the simplified approach used, the answers to these questions may be extremely conservative. The LEWICE/Thermal and ANTICE codes perform a more thorough analysis.

The third question in this section determines if the model is an electrothermal one (internal heat source in one of the layers) or a hot air system (high heat transfer below the layers).

For an electrothermal system, the heater layer must be input. The output is then the required heat emanating from that layer. For a hot air system, the internal heat transfer coefficients are read in from unit 19.

Enter File for Body I: This is the prompt asking for the filename which contains the coordinate info for the Ith body.

9.0 Output Files

This section will define all output files and the information contained therein. It will also duplicate some of the input file information where necessary.

UNIT 7 (JUNK.DAT)

This file contains information which is useful for debugging the program. Unless you have an error in your case, this file need not be consulted.

UNIT 18 (DEICEO.DAT)

Anti-ice output file. Variables are: 'S/C' non-dimensionalized wrap distance, heat required (W/in^2), maximum temperature (°K) and surface temperature (°K).

UNIT 19 (DEICEI.INP)

Anti-ice input file. Supplies # of 'layers', thickness (m), thermal conductivity (W/m*K), and inside heat transfer coefficient (W/m^2*K) versus S/C for anti-icing heat requirement calculation.

UNITS 20-29 (ICE1.DAT, ICE2.DAT...)

X/C,Y/C

Non-dimensional geometry coordinates for airfoil and ice shape for bodies 1-5. **Note**: If you have more than 6 bodies, the unit numbers will overlap with other output files!

UNIT 26 (HTC.DAT)

I, S/C, HTC, FR; Heat transfer data. Variables are: panel #, 'S/C' distance, HTC - heat transfer coefficient (W/m^{2}K) FR - Frössling Number (Nu/$\sqrt{Re}$)

UNIT 28 (PRES.DAT)

I, S, VE, TE, PE, RA

Flow parameters at edge of boundary layer, corrected for compressibility. Variables are: panel #, 'S/C' distance, velocity (V/V_∞), temperature (T/T_0), pressure (P/P_0), air density (ρ/ρ_0)

UNIT 29 FLOW.DAT

I, X, Y, S, VT, CP, J, CSIG, VN

Incompressible flow solution. Variables are: panel # for that body, x,y at panel center, S relative to trailing edge, non-dimensional tangent velocity, pressure coefficient, panel #, sigma solution, non-dimensional normal velocity.

UNIT 30 (MASS.DAT)

I, S, EMFX, EMIX, EMRX, EMEX, MDOTTI, MDOTT, EMEXS

Mass balance terms. Variables are: panel #, 'S/C' distance from stagnation, ice mass, impinging mass, runback mass coming in, evaporating mass, mass available to freeze (in+runback-evap.), mass available to freeze times freezing fraction (should be same as EMFX value), and maximum evaporation (if enough water were available). All mass values are in kg/unit span.

UNIT 32 (BETA.DAT)

I, S, BETA

Collection efficiency data. Variables are: panel #, 'S/C' distance from stagnation, collection efficiency from all drop sizes.

UNIT 33 (FRACT.DAT)

S/C, XTOT, FFRAC, ENVAP, XVR

Fractional form of mass balance. Variables are: 'S' distance from stagnation, fraction stagnant film, freezing fraction, evap. fraction, runback fraction.

UNIT 38 (DICE.DAT)

S, DICE, VRB

Mass balance output. Variables are: 'S/C' distance from stagnation, ice height to be added, velocity of runback (m/s).

UNIT 40 (TEMP.DAT)

S, T, TREC

Energy balance output. Variables are: 'S/C' distance from stagnation, surface temperature (K), 'recovery' temperature (K).

UNIT 41 (DENS.DAT)

S, RI

Ice density output. Variables are: 'S/C' distance from stagnation, ice density (kg/m^3).

UNIT 43 (QENER.DAT)

S, QCX, QEX, QSX, QLX, QCOND

Energy balance terms. Variables are: 'S/C' distance from stagnation, net convective heat loss, evaporative heat loss, kinetic, sensible and latent heat gain from impinging water, sensible and latent heat gain from runback water, conduction heat loss.

UNITS 44-53 (BODY1.INP, BODY2.INP...)

Geometry files for alternate geometry input. A different file is used for each body. X,Y coordinates are non-dimensionalized.

UNIT 56 (MISC.DAT)

Other miscellaneous output from the code. Lift results from flow code, individual trajectory information among others.

UNITS 64-69 (TRAJ1.DAT, TRAJ2.DAT...)

These units contain the x,y coordinated of every time step of every droplet trajectory. File 64 is for body 1, 65 for body 2, etc. These files are very large. If this information is not needed, you can save a great deal of space by eliminating these files.

UNIT 70 (ECHO.DAT)

Contains an echo of all screen output for reference after the code is finished.

UNIT 86 (PLOT1.DAT)

X, Y, S, HTC, ENF, RI, TSURF, XK, DICE, VE, BETA, CP

Plot output file. These terms (all previously defined) are printed out in a format for plotting routines on an iris workstation and for creating postscript files.

UNIT 87 (PLOT2.DAT)

XH, YH, SH, Y0, XTRAJ, YTRAJ

Trajectory plot information. Variables are: x,y,s, y0 points of a trajectory hit and the x,y coordinates of each trajectory within the impingement limits. File is used to create plots on an iris workstation and for creating postscript files

UNIT 88 (PLOT3.DAT)

See UNIT 86

UNIT 89 (PLOT4.DAT)

See UNIT 87

UNIT 90 (PLOT5.DAT)

Impingement limit information for iris plotting routine or for postscript plotting.

UNIT 93 (PLOT6.DAT)

See UNIT 90

10.0 Suggested Procedure for Using Different Modules

Users will often have a flow code and/or a trajectory code which they are already familiar with and would like to use for ice accretion studies. This section will describe the variables which need to be passed or read in so that the user can replace one of the LEWICE 1.6 modules with a different module. There are two methods for replacing one of the modules. First, the user could integrate their flow code into the rest of the modules. Second, the user could run their flow code separately and read the solution into the code. The procedure for both methods is similar. NASA Lewis personnel would be available to perform this integration for you. If Lewis personnel perform this task, we would want to use the resulting code for our in-house research. We would not release the completed work to other sources without permission.

Replace Flow Code Only

The inputs to the flow code are essentially the items provided in the S24Y1 namelist, the geometry file input, the interactive user input, and the airfoil chord (input in TRAJ1 namelist). Outputs from this section are the sigma solution (CSIG), the flow variables at the edge of the boundary layer VE, TE, PE, RA (velocity, temperature, pressure, density). To replace the potential flow code with a grid-based flow code, whether this is an Euler or Naviér-Stokes formulation, simply replace the call to S24Y with a call to that routine, or read in a solution at this point. This procedure should eliminate the need for all routines between S24Y and STAG. The solution at the edge of the boundary layer should be read into the variables given above, while the entire grid solution should be read into the variables XG, YG, VDX, VDY where XG, YG are the grid points and VDX and VDY are the x,y components of the velocity solution at that point. These variables replace the routines VELC, VELR, and VELCTY.

Notes: The calls immediately after S24Y in the LEWICE 1.6 code, those to VEDGE and STAG may no longer be necessary after this operation. VEDGE will calculate the compressible correction to the potential solution, an unnecessary task with an Euler or Naviér-Stokes code. It will also compute the flow derivative dv/ds (DVEDS), the panel angle (surface grid angle), and the total properties such as total pressure and total temperature. This routine may still be necessary if any of these variables are not computed by the code you are inserting.

Similarly, routine STAG computes the stagnation point XSTAG, YSTAG and the stagnation panel ISTAG from the flow solution. If the flow solution you wish to use already supplies this information, this routine will also be unnecessary.

After reading in the grid solution, all calls to VELCTY should be replaced with calls to VEL2, which handles interpolation from an MxN grid solution. The grid information needs to be input before the call to routine RELEAS in routine TRAJ. Please note that the code will try to access the sigma solution if a particle travels outside of the defined grid. As a sigma solution is no longer present, this situation should be avoided.

All other input to the trajectory code is given in the input file or from user input at the start of the code.

In addition, the boundary layer integration BLINT should not be necessary when using a Naviér-Stokes solution as the heat transfer coefficient can be calculated from that solution. If the local heat transfer coefficient is input by the user (into variable HTC), the routine BLINT is unnecessary.

Replace Flow and Trajectory Codes

The user again has the same two options available. The flow and trajectory codes can be integrated with the ice accretion routines, or the user can read in the flow and trajectory solutions. The variables which are calculated by these two routines and are input into the ice accretion routine are: X,Y coordinates, S (distance from stagnation), NPTS (number of points), VE, TE, PE, RA as before, BETA (collection efficiency), HTC (heat transfer coefficient), NTHI, NTLOW (upper and lower transition points), THETA (panel angle), ISTAG, XSTAG, YSTAG (stagnation point information). The input file and user input supply all other variables to this set of routines. Output for the next time step is simply the new set of X,Y coordinates (new ge-

ometry). This procedure should eliminate the need for all routines between S24Y1 and CHOLES. In addition, if the heat transfer coefficient HTC is being supplied by the code you are inserting, the routine BLINT would also be unnecessary.

For either of these procedures, especially if the codes are being merged, care should be taken to make certain that COMMON blocks are aligned correctly, the array sizes being passed are equivalent, and the precision is the same for all variables. LEWICE 1.6 routines are all REAL*8.

There may be other tasks involved in combining the codes which was unintentionally overlooked here. The procedure is straight forward, but may be time consuming due to unanticipated incompatibility of the codes. Members of the NASA Lewis Icing Branch are available to assist you in this task by explaining variable definitions or providing our expertise in merging different codes.

11.0 Using LEWICE 1.6 on a PC

This program was successfully compiled and run on a 90MHz Pentium clone with 32MB RAM and took 11 minutes for one time step for the example case given in the input file description. Note that this is a three body simulation and a single body simulation will run much faster as it requires fewer panels. Various changes were made in this code to make conversion to personal computers much easier. This program should run on any personal computer system without modification or with only minor modification. For the user's convenience, an IBM executable file is included on the disk. The output is in columns of text, with a text header identifying the variable. This file format can be easily imported into any spreadsheet package for plotting.

Problem Shooting

Below are listed some of the problems that were encountered trying to run LEWICE 1.6 on a Macintosh IIci. Even though these problems were solved on this machine, you may encounter some of the same problems.

1) Too many open files - There was a limit on the number of 'units' that could be open at a given time. Should your compiler be more restrictive, you may need to reduce the number which are open at any given point in the program. The normal procedure is to close the unit {CLOSE(#)} when the program was finished reading from/ writing to the unit, and then open the unit up again when it was needed. One of the problems in doing this is that the program often identifies the unit as a variable name, i.e., it will open {OPEN(UNIT=MT} where MT is a variable declared in the program. There are certain points in the program where it is difficult to pinpoint the numerical value a certain variable has. If you need to reduce the number of open files, NASA personnel can help you identify where additional OPEN and CLOSE statements might be.

2) RAM / Hard disk requirements - The exact system requirements are unknown, however I believe a 386 is required at minimum. The program takes over 32MB of hard disk space for the executable, and potentially more than this for output files. To save space, the binary scratch files should be erased after each run, and depending on what you are looking for, you can remove most of the other files as well. Additionally, the variables which store the trajectory information can be eliminated in order to reduce the RAM requirement

3) Compiling / Executing - If your compiler has any options for large source codes or increased precision, those options may be necessary to run the program. For any one-time step run, the accuracy should be pretty good as compared to running the program on a workstation. Often, however, more than one time step/ flow solution is necessary to more accurately predict the ice shape. Previously, the program had unfortunately shown some accuracy problems when using many time steps. Although eliminated now on an iris workstation, this problem may be evident when running the program on a PC.

4) Specific compiler problems which may arise depending on the sophistication of your compiler are: a) NAMELIST incompatibility - Early versions of compilers sometimes do not support namelist input which is used in LEWICE 1.6; b) SAVE command - some compilers want the save command after any DATA statements, others want to have SAVE before the DATA statement.

12.0 User Tips

The tips provided in this section are given elsewhere in this manual and the original manual. They are listed again here for convenient reference by the user.

Old Input Files

Input files from previous versions of LEWICE will **not** work 'as is' with this version. Every variable listed in the input file section must be entered in the order in which it is given. Please follow the examples provided.

Coordinate Input

If you are using the separate geometry files for input, the format has x,y coordinate pairs in two columns of format 2(2X,F12.7). Each body will have a separate file in this format.

Coordinates are input clockwise for each body, starting at the trailing edge.

The code prefers to have non-dimensional coordinates for input.

Panel Criteria

The key to obtaining good ice shape prediction for glaze ice is to run multiple time step cases where each time step produces a flow solution which is acceptable. Poor flow solutions in potential flow are characterized by 'noise' in the CP vs. S curve. Spikes in this solution will result in irregular ice shape formations. In LEWICE 1.6, this is highly automated by the code, but the user has some control to attempt to obtain better flow solutions.

DSMIN will control the number of control volumes/panels used. For single body simulations, very few problems have been encountered. However, multiple bodies sometimes have problems when running multiple time step simulations. Common problems are for the user to specify a value for DSMIN which is too small or too large. Values in the range 0.003 < DSMIN < .0003 are recommended. Please check the geometry output file(s) (units 20-25) to determine how many panels the program is using for each body.

Turning Angle

The turning angle is defined as the acute angle between two panels. The larger this angle is, the more likely that a poor flow solution will be obtained. A small angle will create new panels in regions of high curvature while conserving the total iced area. A slight rounding of the ice shape is obtained with this procedure, although it is not normally visible. A very small turning angle is not practical, as an excessive number of panels will be produced which slows the solution considerably. This is controlled in LEWICE 1.6 by DDANG, which is the turning angle between two control volumes. The key criteria for this parameter is the quality of the flow solution and the number of panels produced.

The modification of the initial input points can sometimes have the adverse side effect of slightly changing the airfoil shape, especially for a sparse initial geometry. This geometry should be examined very carefully for anomalies regarding this side effect.

Time Step

As stated before, one of the keys to good ice shape prediction in glaze ice is the use of multiple time steps. The original manual states that the maximum amount of ice accreted in any time step should be no greater than 1% of the chord. This is still a reasonable value. The computation used is

$$\Delta t = \frac{0.01 c \rho_i}{V(LWC)\left(1 + \frac{\alpha}{20}\right)\beta_{max}}$$

This will give the user a rough idea of the time step size needed for an accurate simulation. The automated time step input will limit the maximum ice growth to NSTEPI times the panel size. Even for long runs (for example 45 min. hold conditions) small time steps can and should be used. A sample case with 270 time steps is included to show the robustness of this model.

Number of Trajectories

An input to the code is the number of trajectories used in the impingement region. A good approximation would be to first estimate how many panels you expect to be in the impingement region. The number of trajectories should not be less than one trajectory for every three panels and should not be greater than one for each panel. As the code adds panels in the geometry modification routines, it will automatically add trajectories in direct proportion to the panel addition. This is a necessary task to produce adequate collection efficiencies for iced airfoils. As the trajectory calculation is the slowest module in LEWICE 1.6, this will slow down the solution and may be noticeable on personal computers.

Droplet Distribution

Most cases run with LEWICE 1.6 will use a single drop size, the MVD for the flight condition. Although multiple drop size distributions can be run with LEWICE 1.6, this feature is only recommended for higher level computers. The procedure is to calculate a collection efficiency for each drop size, and then to superimpose the solutions. For a five drop size distribution, this feature essentially makes the code five times slower to obtain what is often a marginal effect. The main practical use would be to determine more accurate impingement limits on the clean airfoil.

However, preliminary results have shown that multiple drop size distributions have a large impact on the collection efficiencies of multi-body airfoils. See the example cases provided for an example of this effect.

PC Application

Personal computers are more limited in their capabilities than workstations or other high end computers such as vax or Cray. Engineers who use PCs to run this code may want to limit some of the capabilities to reduce RAM and storage requirements. Specifically, many output files are generated by LEWICE 1.6 for plotting purposes. Users of this code may find that the are only interested in a select few of these outputs such as the ice shape file and flow solution file. The other outputs can be commented out which can greatly reduce the amount of data produced and will increase the execution of the program.

Similarly, routines used solely to print out droplet trajectories contain large arrays which use a lot of RAM. If this information is not important to you, the RAM requirements can be greatly reduced by eliminating these arrays. Another limitation could be the number of open files a PC can have. Newer computers and compilers normally do not have a problem with this, but older machines and older compilers may. Reducing the output allows you to eliminate some of the open files if this is a problem.

Anti-Icing

This program will calculate the heat requirements and then compute the ice shape as if the surface were unheated. Layers are input with the inner surface first, and the outer surface last. The surface temperature input in LEWICE 1.6 must be above freezing (in Kelvin) for this option to work. NASA Lewis also has codes which perform more detailed analysis of deicer and anti-icer performance. The LEWICE/Thermal code (32) performs a 2D transient deicer simulation and the ANTICE code (30) performs a 2D steady-state anti-icing simulation. If you want a more detailed analysis, you are encouraged to try these codes.

Case Study

This option allows the user to perform a parameter sweep of one variable using one input file and one set of I/O. This allows one variable to change while all other variables remain the same. Trying to accomplish this task with several runs often leads to mistakes by the user in not supplying the exact same information for all the runs. The variables which can be parameterized in this manner are: temperature, liquid water content, pressure, angle of attack, median droplet diameter (only for one drop size cases), sand-grain roughness, and number of time steps. This list can be added to or changed based on user needs.

Parameter Arrays

Most of the arrays in the program are dimensioned using a PARAMETER statement. This allows the array sizes (total number of panels allowed) to be

easily increased or decreased by the user. The program currently allows 10001 panels and 10001 control volumes. The RAM requirements can be decreased by lowering these values.

Inactive Input

Several variables in the original LEWICE input file are no longer used by the LEWICE 1.6 code. LEWICE 1.6 will **not** read in this input so that old data files will have to be changed. Refer to the section on input files for a description of the current input files.

Multiple Stagnation Points

The criteria used by the program is to select the value closest to the stagnation point from the previous time step. If it finds more than one stagnation point on the first time step, or when using a restart file, the point closest to the hilite is used. If this is not satisfactory, the user should lower the turning angle criteria or otherwise smooth the input data so as to produce a single stagnation point value.

Flow Code Limitation

For glaze ice shapes at high subsonic velocities, it is possible for the code to compute a pressure coefficient which would lead to a negative local static pressure. The program will compute the static pressure needed for a local Mach number of 0.8, hence 'rounding off' the solution. The subsequent ice shape may not be an accurate representation. The user is encouraged to use a Euler/Naviér-Stokes flow solution for this case.

13.0 Numerical Improvements

13.1 LEWICE 1.0 - LEWICE 1.3

This section will describe improvements made to LEWICE between version 1.0 (original LEWICE) and version 1.3 (sometimes referred to as LEWICE-Beta). **All of these refinements are included in LEWICE 1.6**, except where noted in the second section of this chapter. Some of these changes were first described in Wright (11).

Last Modification Date

Each subroutine contains the last date a change was made to the routine, along with the initials of the programmer. WBW - William B. Wright; CSB - Colin S. Bidwell MGP - Mark G. Potapczuk. Most of these are WBW because I had responsibility for putting the package together, and often made changes to routines supplied by others to integrate subroutines.

Disclaimer

This is a notification, both printed out and in the source that this code is available only to U.S. Companies, Universities and Government Agencies.

Scratch Files

Most scratch files were eliminated in the code. These files were used to pass information between subroutines, a process which is done more efficiently using COMMON blocks as I/O is one of the slowest computer functions.

Split Output

Output from the program has been split into several files to facilitate plotting after the run. Most output files contain columns of text, with a header which describes the variables. A description of the variables in these output files is contained in this report.

Split Common Blocks

Large common blocks were split into smaller fragments to eliminate passing of variables which aren't used by that routine. Helps programming and readability of code.

Double Precision

All routines were modified so that the code is REAL*8 for accuracy purposes.

Remove Routine CNSTS

Physical constants are provided in the first routine in which they appear. Most of the definitions in this routine are ice/water properties and are now located in EBAL.

Replace Routine ASK with WINPUT

Interactive I/O at the beginning of the program has been expanded to include several options which are described in a previous section. These options were added to provide more features to the user and improve the models within the program.

Add Case Study

This option allows the user to perform a parameter sweep of one variable using one input file and one set of I/O. This allows one variable to change while all other variables remain the same. Trying to accomplish this task with several runs often leads to mistakes by the user in not supplying the exact same information for all the runs. The variables which can be parameterized in this manner are: temperature, liquid water content, velocity, angle of attack, median droplet diameter (only for one drop size cases), sand-grain roughness, and number of time steps. This list can be added to or changed based on user needs.

Perform Automatic Flow Recalculation

The program used to allow for ice to be accreted for multiple time steps without the flow field being recalculated. This is inaccurate, especially if panels (points) are added by the geometry modification routine. The program will perform a flow solution, a trajectory solution and an ice accretion for each time step.

Time and Time Step

Rather than ask the user for the time step after every flow solution, the program asks for the total accretion time and the number of flow solutions. By doing this, the program can run to completion after the initial input set. It also has the option (ITIMFL=1) to automatically calculate the timestep based on limiting ice growth to NSTEPI*DSMIN height for each time step.

Perform Multi-body Ice Accretion

The potential flow program in LEWICE 1.6 is capable of solving for the flow over multiple bodies, but the other routines were not able to handle this option. Routines were added to allow for trajectories and ice accretions to be performed on multiple bodies.

Eliminate Hardware Specific Plot Calls

LEWICE 1.6 contained internal plot calls to a graphics routine specific to NASA Lewis. These routines were removed. LEWICE 1.6 does not perform interactive graphics. All graphics are performed after the run by printing out generic text files which could be read into several plot packages. A post-processing graphics routine for workstations is available from NASA Lewis.

Create Echo File

All terminal I/O is printed out to UNIT 70. This allows the user to keep a permanent record of the terminal input. It is also useful for reviewing errors which could have been missed during the run.

Alternate Coordinate Input

There are two options for the initial input of the geometry coordinates. The first option is the option originally used by LEWICE 1.6 so that users who have been running the code do not have to change their input files. The second option allows the user to specify the input in x,y coordinate pairs from a separate file (for multi-body input each body is input from a separate file). This second option is often more convenient to use.

Parameter Arrays

Most of the arrays in the program are dimensioned using a PARAMETER statement. This allows the array sizes (total number of panels allowed) to be easily increased or decreased by the user. The program currently allows 3000 panels in the flow solution and 1000 for the trajectory and ice accretion. This discrepancy is needed for multi-body runs as the flow solution is solved for all bodies simultaneously, while the trajectories and ice accretions are handled individually.

Simplify Structure and Options of Code

The potential flow code used by LEWICE 1.6 contained several options which either could not be used for ice accretion or which were preferable for use with ice accretion calculation. The options which were removed are summarized below.

ISOL

Solution mechanism for the potential flow code. There were three options in the original code. All of them could be used for small arrays, while one was preferred in the original user's manual for arrays larger than 101 panels. The SOLVIT routine was kept, and the other options were removed.

IPRINT

Additional 'debug' printing switch. This option created excessive printout and was unnecessary for a working code.

ITYPE

Input along with x,y coordinates. Not necessary as the code assumes the x-coordinates are read in first.

LCMB

Default value (0) to use the method of S24Y was kept, the other method being duplicative.

LCMP

Compressibility corrections do not add significantly to the computational time and are therefore always performed (option 1).

LEQM

Droplets should always be released in equilibrium with the surrounding air (option 1).

LSYM

Symmetric results are normally obtained even when running a full solution. As the code is fast for most modern machines and most real applications are unsymmetrical, this option was removed (option 0 used).

LXOR

A particle is in equilibrium with the surrounding air when the computed off-body velocity is within ΔV_e (option 1).

LYOR

Rather than have the user unintentionally supply erroneous information, the program uses the YOMAX and YOMIN values calculated as initial guides for trajectories (option 1).

PRATK

As the particle is in equilibrium with the air, it will be travelling at the same angle of attack. Hence PRATK = α.

XORC

Not input when LXOR = LYOR = 1.

YORC

Not input when LXOR = LYOR = 1.

XSTOP

Inaccurate to use any value other than the end of the geometry Particles should be allowed to hit any portion of the geometry.

YOMAX

Initial guess calculated by the code.

YOMIN

Initial guess calculated by the code.

VXPIN

Not needed for LEQM = 1.

VYPIN

Not needed for LEQM = 1.

QCOND

Heat input for a thermal deicer/anti-icer system. This has been changed to interactive input. The user supplies a desired surface temperature (above freezing) and the code will calculate the heat required to achieve that temperature for each control volume. Electrothermal and hot air systems are modeled.

In addition, there were several options to the S24Y flow code which were not input from the LEWICE 1.6 input file, but were initialized in routine SETUP. Several of these were unnecessary and were removed. These variables are listed next.

ID, IBOD, IDOLD, LAST, ITYPE, MORE

Variables are not needed due to alternate (simpler) method of inputting multiple bodies. LEWICE 1.6 asks for the total number of bodies at the beginning and then inputs the coordinates in the order x(body1), y(body1), x(body2), y(body2) etc. The old routine looks like it dates from an era of punch card input.

ISV

Body save option. If the body needs to be saved by the program, it will be.

TTITLE, FTITLE

Title Input. Duplicative of original title input and hence unnecessary.

ITR

Transformation input. Default 0 (no transformation) used.

IOFF

Off-body points toggle. Off-body points are needed for trajectory calculation, hence this is performed (option 1).

NONU, NBNU

Non-uniform flow input. Non-uniform flow routines were commented out since 8/83, hence it was assumed this option did not work properly or was not needed.

Routines Removed/Renamed

The removal of these options, along with the elimination of most scratch files was performed in an effort to make the flow code more streamlined and hence more efficient. The process also makes the routines more readable and easier to modify. The following represents a list of the routines that were eliminated because they were duplicates of other routines, could be combined with other routines, performed a function no longer necessary or were involved only in producing scratch file I/O: RWND, REWYND, FILES, GETT, SAVE, NEW45, ASSEMB, PRNTEL, PRINTG, MAIN1, MAIN3, SOLVE, UPDT30, MIS2, QUASI, OFFPTS, VXYOFF, VPROFF, COMB2D, PLTRAJ, BORDER, PLOTD, INTPT, PSURF, COMPF, COMPT, CPW, PVW, PVI, ASK, READIN, NWPTS. As a result of these and other changes, the lines of code in the potential flow routine was cut approximately in half with no loss of useful functionality.

ASK was expanded and renamed WINPUT.

READIN no longer performs reading of information and its name was changed to SETLIM.

PVW and PVI were combined into PVAP.

CPW is performed in ACCRET.

COMPF and COMPT were combined into ACCRET.

OFFPTS, VXYOFF, VPROFF were combined into OFFBOD.

INTPT - This methodology for determining trajectory hits is no longer used.

PSURF, PLOTD, BORDER, PLTRAJ, COMB2D, MIS2, QUASI performed options which were no longer needed.

MAIN1, MAIN3, SOLVE - additional hierarchy not necessary.

PRINTG, PRNTEL, ASSEMB - mostly I/O functions. Any other functions of these routines are performed by ELFORM.

RWND, REWYND, FILES, GETT, SAVE - I/O routines not needed.

NWPTS was replaced by NWPTS2 and NWPTS4.

Simplify COMPS Routine

The 'S' distance from the stagnation point is computed in a simpler fashion than the previous code. The zero point is the interpolated stagnation point computed by STAG, not the panel center.

Improve STAG Routine

Error bound on identifying a stagnation point was changed from 10^{-2} to 10^{-10}.

Stagnation point is linearly interpolated from VT values to find VT=0 point. This reduces code dependency on panel size and location.

If more than one stagnation point is found, the VT vs. S curve is artificially smoothed in that region. If more than one stagnation point is still found, the process is repeated for up to three iteration. After this point, a stagnation point is selected by the program. The criteria used by the program is to select the value closest to the stagnation point from the previous time step. If it finds more than one stagnation point on the first time step, or when using a restart file, the point closest to the hilite is used. If this is not satisfactory, the user should lower the turning angle criteria or otherwise smooth the input data so as to produce a single stagnation point value.

Pseudo-surface generation was eliminated, as well as all terminal I/O in this section.

Compute $\partial V/\partial s$ from Flow Solution

The boundary layer routine requires the value of this derivative, which previously was computed in BDYLYR. It is now computed in VEDGE by performing a weighted central difference of the V vs. S curve to find the derivative at each panel. This array is then artificially smoothed to remove some of the 'noise' in the flow solution (12).

Compute Panel Angle

The angle between panels is computed and stored in an array instead of being calculated each time it is needed.

Add Checks on Computed Pressure

For glaze ice shapes at high subsonic velocities, it is possible for the code to compute a pressure coefficient which would lead to a negative local static pressure. The program will limit the local Mach number to M=0.8 and calculate local velocity and pressure from this.

Move RA (air density) computation to VEDGE. Air density array is computed along with other compressible arrays in VEDGE for uniformity.

Change Print Out Routines

Arrays are printed out from the routines in which they are computed. If the program bombs, this makes it easier to trace the location of the error. It also ensures that information is printed using the 'S' values from the existing geometry, not the new geometry after modification. The second method causes values to appear to be 'shifted'. The print outs are also sent to separate files to facilitate later plotting.

Add Routine VEFORM

A 100+ line routine associated with the flow field computation was performed by MAFORM, VELCTY, and OFFBOD. The creation of this subroutine elimi-

nated the duplication of coding and made the routines more readable.

Value Check in ABFORM Routine

The trajectory code requires an off-body air velocity at the droplet location. For a potential flow code, this is obtained by summing the contribution from each panel. When a drop is very close to a panel, the contribution from that panel will be very large. As the contribution from this panel will dominate the calculation, its value can be truncated without loss of accuracy to the off-body velocity calculation. The distance limit in the code is currently 10^{-5}. By using this simple check, the input DSHIFT is no longer necessary for trajectory calculation.

Remove DSHIFT Functionality

As a result of the change in routine ABFORM, computation and use of DSHIFT was removed from routines READIN and MODE.

Add Grid-Based Velocity

LEWICE 1.6 offers the option to the user of calculating off-body velocities by interpolating from a grid. This method is very useful in the 3D code as interpolation is much faster than summing the contribution from every panel. The decrease in computation time is less in 2D (around 20%), as there are fewer panels. The procedure is to calculate a grid, compute the off-body velocities at the grid-points, and then interpolate from the grid a value at a droplet location. The overhead incurred by calculating a grid and computing the off-body points nearly cancels the increased efficiency obtained from the interpolation process.

The user has a choice between a skewed rectangular grid and a 'C'-grid. When using the potential flow program, the rectangular grid is preferred as more of the grid points are located in the region where droplet trajectories are located. The 'C'-grid option can be modified by the user to supply a velocity field from a different source, for example, from a Naviér-Stokes solution.

Additional routines were created to:

1) create the grid;

2) compute the off-body velocities at the grid-points; and,

3) interpolate the air velocity at the drop location.

Add Temperature Variation of Physical Variables

Air density and viscosity were constant in the trajectory routine, but were computed as a function of temperature in the boundary layer routine. These variables are now temperature-dependant in both routines.

Correct Temperature Dependance of Variables

Air viscosity and thermal conductivity were temperature dependant in the boundary layer routine. However, their dependance on temperature was inaccurate due to a typographical error in the routine and was corrected.

Remove One Trajectory Option

Previously, LEWICE allowed one trajectory to be computed. For this case, RANGE, IMPLIM and ORDER were not accessed. Accurate collection efficiencies cannot be calculated using only one trajectory, so this option was removed. If one trajectory is selected in the input, the code will exit and notify the user that more trajectories are required.

Set Particle Angle Equal to Flow Angle

As water droplets are released in equilibrium with the surrounding air, they will be travelling at the airfoil angle of attack, hence separate values are not necessary.

Calculate Y0MAX, Y0MIN

The initial values of the upper and lower limit trajectories are set to the uppermost and lowermost coordinates of the airfoil plus (or minus) 1/2 the airfoil thickness. These are adjusted for angle of attack and are reset to previously computed missed trajectories during the impingement limit search.

Storage of Trajectories

Droplet trajectory information is stored for later printing for plotting purposes. These arrays are quite large and may cause memory problems on a personal computer. If a user has low memory requirements, these arrays and the print out routine can be eliminated to save memory.

Move VELCTY Call Statements

Routine VELCTY computes the off-body air velocity. Previously, this was called from routine DIFFUN, as well as from several other routines. However, this routine only needed to be called whenever the time step was changed, not on every call to DIFFUN. The CALL statement for VELCTY was placed at strategic locations within routines INTIG and DIFFUN to speed these routines. As a result, the integration of a single droplet trajectory is approximately 50% faster.

Add Save Routines to DIFSUB

Three routines, YSAV, YSAV1, and YSAV3 were added to DIFSUB to replace repetitive routines whose purpose were to save and recall the solution. This reduced lines and clarified the structure of the routine.

Adjust Integration Constants

Within DIFSUB, there are several fractions in COMMENT statements which are typed in as their decimal equivalents in the code for accuracy. These decimal equivalents were checked against their supposed fractional equivalent. As a result, some of the decimal equivalents were changed slightly.

Eliminate PEDERV Option

There was an unused option in DIFSUB which required a routine called PEDERV. As the option is not used for icing calculations, the routine and the CALL statement were eliminated.

Compressible Correction to Flow

The off-body velocities are computed from the incompressible potential flow solution. After this is performed, the value is corrected for compressibility using the Kármán-Tsien method used in VEDGE.

Drag Relations

An empirically-based correction was made to the sphere drag to account for compressibility effects. This is not a rigorous curve-fit, but a crude approximation based on numerical values of sphere drag at various Mach numbers and Reynolds numbers (35). A slight change in the incompressible drag relation was also made. The formula currently used is taken from White (18).

Correct Above/Below Trajectory

The previous determination of whether or not a trajectory was above or below a body was corrected by saving the maximum drop location as it passed by the surface. The previous determination did not always work for highly cambered airfoils.

Intersection Criteria

A droplet trajectory will intersect the geometry when the line formed by the last two drop locations intersects one of the panels. Currently computed by determining the intersection point of the two lines (the line of each panel and the trajectory line) and determining if the intersection point is on the panel. More accurate than performing a sum of the angles (performed in the previous version), which can fail due to numerical truncation.

Impingement Location

Due to the change in 'S' definition in COMPS and STAG, the computation of the 'S' location of a trajectory hit was corrected and led to a simpler formulation.

Change Criteria on Vertical Lines

The criteria for a vertical line was changed from a denominator of 10^{-5} to 10^{-9}.

Array Change for Multiple Drop Distribution

Due to the elimination of a scratch file, the arrays which contain the impingement data for each drop size in a distribution were changed from size (NPA) to (NPA,10) where NPA = number of panels and 10 is the maximum number of drop sizes in a distribution.

Alternative Collection Efficiency Calculation

LEWICE calculated collection efficiencies from the droplet hit locations by performing a least-squares polynomial fit of the points. The number of terms in the polynomial could have any value, as long as it was an even number. A 4th order polynomial is currently used by routine TERP, which performs the polynomial fit. An alternate method was developed which performed a simple central difference of the Y0 vs. S points to obtain a derivative at that point. Values at the panel centers are then interpolated from these values. This method tends to produce a 'smoother' collection efficiency distribution than the polynomial fit, which can sometimes produce erratic results if the correlation of the polyno-

mial is not high. The interpolation scheme used assumes no more than one trajectory per panel. If the interpolation scheme fails for any reason, a spline-fit routine is called.

Compute Trajectory Angle

An option was added to the geometry modification routine to grow ice in the direction of the impinging trajectory instead of normal to the surface. For this procedure, the angle of the trajectory when it impinges is saved to an array.

Output Files for Plotting

In addition to the printout of variables in columns of text, LEWICE 1.6 also outputs files in a format readable by GL plot routines used on a unix workstation. The 'C'-source routines which perform this plotting are available for users who have workstations.

Increased Number of Trajectories

Previously, the number of panels could increase during a run while the number of trajectories remained constant. Currently, the number of trajectories increases in direct proportion to the increase in panel number so that a more accurate collection efficiency curve is found for long cases.

Variable LWC

The Icing Research Tunnel at NASA Lewis has a short period at the beginning of a spray whereby the LWC steadily increases from zero up to the desired LWC value. The user can input a 'ramp-up' time at the beginning of a run. This will result in the LWC increasing linearly from zero to the LWC in the input file over the course of the time specified. Ramp-up time of up to 20 sec. have been noticed in the IRT. Numerically, the effect of using this parameter has been negligible.

Transition Movement

Close-up movies of the icing process (26), (27) have indicated that the transition point moves toward stagnation during the course of an icing run. Although currently commented out, the mechanism for forcing this transition movement has been coded in (36).

Allow Different Roughness Regions

An option which is currently commented out in WINPUT and BDYLYR would allow the user to input separate roughnesses for the upper and lower sections of the airfoil. Until a better empirical relationship for roughness exists, this option will remain off.

Allow Different Transition Criteria

The program will change from laminar flow in the boundary layer to turbulent flow when the local Reynolds number based on the sand-grain roughness height exceeds 600. An option in the program, currently commented out, would allow the user to specify different criteria above and below stagnation.

Non-Dimensionalize Routines

The flow solver was already non-dimensionalized so the integral boundary layer routine and trajectory routine were non-dimensionalized for debugging purposes.

Add Frössling Number to Print Out

The Frössling Number ($Nu/\sqrt{Re}$) was calculated and printed out for comparison to some NASA Lewis experimental data which was presented in this form (37).

Create Routine BLINT

Previously, the boundary layer integration for above and below stagnation were carried out by different routines within BDYLYR. The integration process is identical, hence routine BLINT was created to perform both integrations. This was performed as an aid to the debugging process.

Calculate Stagnation Heat Transfer Coefficient

Previously, as 'S' = 0 at stagnation, the heat transfer coefficient was assigned the same value as the panel immediately downstream from stagnation. However, the boundary layer integral can easily be evaluated at stagnation using L'Hopital's Rule. This procedure is currently used to calculate the stagnation point heat transfer coefficient.

Small Velocity Integration

An alternate procedure was developed to evaluate the boundary layer integral where $V/V_\infty<.1$. This was performed to avoid premature tripping of the boundary layer due to numerical error in the flow solution near stagnation. Terms in this integration have powers as high as V^6, hence this routine is highly susceptible to slight numerical errors.

Calculate Shape Factors

Previously, LEWICE used flat-plate values when calculating the relationship between momentum thickness and boundary layer thickness. Currently, a von Kármán-Poelhausen analysis is performed (10) using a Newton-Raphson iteration to find the shape factors.

Interpolate Transition Point Location

Previously, the transition point was placed at the panel center of the first panel which had a roughness Reynolds Number greater than 600. This was found to be dependent on panel size and location, causing numerical errors in the turbulent heat transfer coefficient. Currently, the upper and lower transition points are found by interpolation to find the 'S' location where $Re_k = 600$. This fixes a numerical error in finding the transition point location.

Redefine Transition Criteria

An option, currently commented out in the program, would replace the sand-grain roughness value with the boundary layer height value in the calculation of the roughness Reynolds Number for those cases where the sand-grain roughness value exceeds the boundary layer value at that location. Not currently used.

Newton-Coles Integration

For purposes of increasing accuracy, the integration of the boundary layer is performed by a 5th-order Newton-Coles procedure which was modified for variable spacing instead of a trapezoidal approach (15).

Additional Transition Criteria

Additional conditional statements were placed on the transition criteria to account for transition via roughness Reynolds Number criteria and forced time-dependent transition criteria explained previously.

Limit Sand-Grain Roughness

An option, currently commented out in the program, would truncate the sand-grain roughness value to the turbulent boundary layer thickness for those cases where the sand-grain roughness value used exceeded the turbulent boundary layer thickness. Not currently used.

Replace Roughness Staunton Number Definition

The definition of roughness Staunton Number previously used by LEWICE was recommended by Kays and Crawford for spheres packed in a tube. Experimental evidence from Dipprey and Sabersky (13) on actual sand-grain roughness gave a different correlation. As LEWICE 1.6 uses the equivalent sand-grain roughness approach, the second correlation is felt to be more appropriate until experimental data on ice shapes can be taken. The task of taking this type of data is currently under way.

Replace EBAL with Simplified LEWICE/Thermal Routine

The more rigorous heat balance methodology used by the LEWICE/Thermal deicer program replaces the routines EBAL, COMPF, COMPT, CPW, PVI and PVW routines in LEWICE 1.6. This routine has enhanced internal documentation, initializes variables, checks the flow and collection efficiency routines for errors, and iteratively computes surface temperature for all conditions using a Newton-Raphson iteration. In the glaze ice regime, where freezing fraction is between 0 and 1, the freezing fraction is replaced with temperature by using a high heat capacity approach. The heat capacity in this region is the latent heat of fusion divided by a small temperature difference, $\Delta T = 10^{-5}$. A phase check is incorporated into the calculation to further increase the accuracy of the freezing fraction computation. All terms of the energy balance and mass balance are computed and printed to separate output files. Physics improvements include: using the runback analysis performed by the boundary layer routine; adding some conduction effects; correcting a slight error in the evaporation term; adding droplet shedding via an empirical correlation with Weber Number; using the variable ice density routine; adding aerodynamic ice shedding via an empirical correlation (38) and performing a particle trajectory of the shed particle.

Shear-Driven Runback Flow

Instead of allowing runback water to freely pass into the next control volume, the rate is moderated by assuming the surface water is shear-driven by the surrounding air flow. If the force of the shear flow (or the force of gravity) is greater than the surface tension force, water is allowed to flow, else it remains in

the control volume and is added into the mass balance for the next time step. This procedure also has the capability of predicting a water bead height which could be used in place of the input sand-grain roughness. The geometry of the bead is determined from the contact angle and the 'spread factor' (31). Currently this option is commented out, and the program uses the input sand-grain roughness.

Phase Check

When using a high heat capacity method, as the surface temperature is not known beforehand, it is necessary to first assume a heat capacity, compute the surface temperature, and then check the assumption. For an unheated airfoil, the original assumption, which is based on a freezing fraction calculation, is not often wrong. The checks remain, however.

Conduction Effects

By performing a thermal analysis of an unheated airfoil using the LEWICE/Thermal code (30), it was discovered that the heat loss into the airfoil approximated an analytic solution for 1D transient heat transfer (28). This analytical solution was then added to LEWICE 1.6 as a means of calculating the heat loss by conduction.

Water Shedding

Close-up movies of the icing process by Olsen (26) showed some cases where water drops were shed from the surface. He showed that the shedding was qualitatively proportional to the local Weber Number. By matching his qualitative findings to quantitative terms in LEWICE 1.6, an empirical relationship was obtained which causes a small amount of mass loss at ambient temperatures approaching freezing. This relationship needs to be more quantitatively defined.

Ice Density Correlation

The ice density correlation given by Macklin is based on a computed parameter instead of a measured one. As a result, LEWICE would often give constant (glaze ice) values for ice density. An alternative correlation (again using low-speed rotating cylinders) was found to give qualitatively correct ice densities. An IRT experiment is planned to evaluate this correlation.

Curvature Calculation

The ice density correlation requires the local radius of curvature to make an analogy with the rotating cylinder data. This curvature is computed by assuming the five panels on either side of the panel in question form a partial arc of a circle. By comparing the arc distance with the straight-line distance, the radius of curvature can be calculated.

Ice Shedding

If the macroscopic aerodynamic force (found by summing surface static pressure * area) is greater than the adhesion force on the ice, the entire ice geometry will be shed. This routine is currently not used, as the empirical relationship between adhesion force and surface temperature is believed to be in error (38). In glaze ice conditions, where the ice surface temperature is 32 °F, the relationship would compute no adhesion, whereas qualitative experimental evidence shows that glaze ice is firmly held to the surface.

Ice Particle Trajectory

As the ice shedding routine is not used, this routine is also not used. It uses the sphere drag relationship used for water droplet particles and calculates a simple velocity and direction of the particle, which assumes that it is following the airflow streamlines.

Geometry Addition

LEWICE added ice to the surface by taking the computed ice height and adding that distance, in the unit normal direction, to the existing x,y coordinates of a panel. This results in two values for the corner coordinates which were then averaged to find the new coordinate pair. This procedure does not take into account the local curvature, which can be expressed by the turning angle. Currently, a similar procedure is used. However, an iteration loop was added to correct for the curvature. A correction coefficient is defined as the ratio of the required area to be added (ice height * Δs) divided by the actual area added. The ice heights are then multiplied by this coefficient and the process is repeated for twenty iterations. At the end of this procedure, the areas are very nearly identical. A small panel turning angle requirement aids this iteration because the initial geometry error is small for small turning angles.

Ice Addition Direction

A procedure was developed to grow ice in the direction of the incoming trajectory, in the direction of the flow, or in the unit normal direction. This procedure can also allow for rime ice to grow using a different methodology than glaze ice. Currently, the code grows ice normal to the surface. The other routines are commented out.

Additional Geometry Check

SEGSEC, which controls panel removal due to intersecting or nearly intersecting panels is called after the above procedure has been completed as well as after the panel addition routines.

Correct Jaggedness of Ice

The geometry modification routine described earlier can sometimes cause the iced geometry to appear 'jagged' even when the ice height curve is smooth. If left uncorrected, this will cause an excessive number of panels to be added. This is corrected by using a routine similar to the one used to add panels. This routine finds three sequential panels where the two turning angles are of a different sign ('jaggedness' criteria). It then adjusts the two interior points such that 1) the net area is the same; 2) the size of the three panels is the same; and, 3) the two turning angles are the same. This procedure is repeated until any region of three panels where the two turning angles have different signs has turning angles less than 2°. The output from this routine is an iced geometry with the same number of panels and the same iced area, with less jaggedness.

Check Transposition

After this routine and the other two panel modification routines, a routine is accessed to check for transposition of the new coordinates. In the early history of these routines this was a problem, but after some bug fixes it has not recurred. The checking routine remains, however.

Panel Size Check

A routine was added to check the relative size of neighboring panels and to adjust the coordinates if the ratio of sizes is greater than SEGTOL. If the ratio is greater than SEGTOL, the intersection point between the panels is changed such that 1) the area of the triangle formed by the three x,y coordinate points is the same after modification; and, 2) the panel sizes are the same. This procedure is repeated until all panel size ratios are within SEGTOL. The output from this routine is an iced geometry with the same number of panels but the ratio of the panel size is less than or equal to SEGTOL.

Panel Addition

A routine was created to add one panel where the turning angle is greater than the user input. An additional point (panel) is added to three existing points (two existing panels) such that 1) the area of the triangle formed by the three existing x,y coordinate points is the same as the area of the tetrahedron formed by the four new points (three new panels); 2) the two end points remain the same; 3) the size of the three panels is the same; and, 4) the two turning angles created are the same. This procedure is repeated until all turning angles are less than the input value. The output from this routine is an iced geometry which has the same area and a similar overall shape, but has a more well-defined surface due to the point (panel) addition.

13.2 LEWICE 1.3 - LEWICE 1.6

This section will describe improvements made to LEWICE between version 1.3 (sometimes referred to as LEWICE-Beta) and version 1.6 (current version). In any case where the information in this section conflicts with the previous section, the information in this section is correct, as this reflects further refinements/ changes to those described earlier.

Several improvements were made to the LEWICE code which allow for more accurate ice shape predictions. These include changes to the surface modeling, mass addition and time stepping algorithm to improve convergence characteristics of LEWICE 1.6. Modifications were also made to the transition model, the calculation of transition heat transfer transition heat models to produce more realistic heat transfer. New additions include an ice growth algorithm which allows ice growth in arbitrary directions and a new "pseudo" surface which features improve heat transfer prediction for large glaze shapes.

Several changes to the LEWICE calculation scheme have been made to achieve good convergence characteristics. A converged solution is one that ceases to change with increasing size and or increased number of time steps. Any good numerical method should have good convergence characteris-

tics. Convergence is generally controlled by increasing the number of grid points or time steps within the accuracy of the computer.

For LEWICE 1.6, convergence is measured using ice shapes and is controlled by the number of time steps and grid points. Convergence has occurred when an increase in time steps or grid points results in no appreciable change in the ice shape. LEWICE has had a history of poor convergence characteristics. The complex, nonlinear nature of ice growth has been a severe impediment in producing convergence. The number of possible time steps have been limited to 5 steps for some complex glaze shapes and 10 steps for some simple rime shapes. After these limits flow code failure and ice shape divergence were common. These changes have been made to LEWICE 1.6 to overcome the problems resulting in poor convergence and cases have been run using as many as 1000 time steps.

An adaptive grid technique was incorporated into LEWICE 1.6 to allow optimization for each phase of the ice accretion process. Surface models are generated from a highly refined baseline model using adaptive grid techniques. Separate surface models can be used for each step in the ice accretion calculation depending on the specific accuracy requirements for that step. The adaptive grid method is useful in producing surface models with point distributions that are sufficiently dense in regions of interest, vary smoothly and have a minimum of points, insuring computational accuracy and speed. The adaptive grid technique involves the use of weighting functions supplied by the user to produce weighted point distributions for a surface. Surface models can be generated which resolve any variable of interest in the icing process. Weight functions can be chosen to produce densely packed points in regions of interest and sparsely packed points away from the region of interest. Possible weighting functions could include: radius of curvature, velocity or velocity gradient, collection efficiency or collection efficiency gradient, ice thickness or ice thickness gradient.

Currently three surface models with various options are used for the icing calculation. For the heat transfer calculation two types of models can be used: a standard surface model and a "pseudo" surface model. For both models, point distributions are weighted using radius of curvature with user specified constraints on maximum and minimum point spacing. For the trajectory calculation, surface point distributions are optionally weighted to collection efficiency or towards a constant leading edge spacing. The baseline model is used for the remaining mass balance, energy balance and the mass addition calculations. This model, which features several thousand equally spaced points, was found to be sufficiently accurate for these calculations and reduced the error in the task of updating the baseline model after each time step.

The ice growth scheme has been modified to produce more conservative, accurate, and smoother ice shapes. The scheme involves a separate time stepping procedure for the ice growth module. Ice is added to the geometry in small increments to a highly refined surface model. During each ice addition step, the new surface model is checked for regions of convergence and divergence. Points in regions of high convergence are removed to keep the region from growing into itself. Points are added in regions of high divergence to preserve the resolution of the surface model. Step size is varied according to convergence and divergence rates of the surface model to produce a smooth ice shape. The ice addition routine continues, conserving mass locally, until all of the ice has been added for the current time step.

A new variable time stepping algorithm was installed which has better convergence characteristics, is more automated and is computationally more accurate and quicker then traditional constant time stepping algorithms. The new method involves restricting the maximum ice thickness for a given time step to a specified amount. Essentially the maximum allowable ice thickness is set and the time step is allowed to vary to produce this maximum thickness. A default value of 0.2% chord has been found to be reasonable, but this can be changed by the user.

This type of time stepping allows direct control over the most important parameter affecting convergence which is geometric change. Geometric change is the direct cause of change in both collection efficiency and heat transfer distribution which are the two main contributors to ice shape calculation. If the amount of geometric change can be controlled in a systematic way, then changes in collection efficiency

and heat transfer distribution can be controlled as well and good convergence characteristics can be achieved.

This time stepping technique should be highly independent of both geometry and icing condition lending itself to automation. Simply put, if the maximum geometric change which produces a discernible change in collection and heat transfer is known, then this value can be set and should not have to be changed for any condition or configuration. The technique produces larger times steps for periods of small geometric growth and smaller time steps for periods of large geometric growth than the traditional methods, producing fewer and more accurate time steps.

The transition model has been modified to produce realistic transition for cases with multiple stagnation points. Previously it was possible for transition to be predicted prematurely for cases with multiple stagnation points. This resulted in poor heat transfer distributions where transition would be predicted very near the leading edge. This would cause the heat transfer coefficient aft of the transition point to be less than the laminar value. This shortcoming is responsible for a good portion of the ice shape asymmetry observed for seemingly symmetric glaze icing conditions (e.g. NACA-0012 at 0 degrees angle-of-attack). The transition model is highly dependent on velocity gradients, which are realistically large between multiple stagnation points. Transition was predicted prematurely between consecutive stagnation points because of these large gradients even though the velocities were near zero. The newly incorporated model forces transition to be delayed until aft of the aftmost stagnation point at the leading edge producing more realistic heat transfer. This correction provides the desired result of producing turbulent heat transfer coefficients which are higher than the laminar values, which is a more realistic result.

Corrections were made to the transition heat transfer calculation so that the boundary layer would not prematurely transition from laminar to turbulent. A premature transition occurs for high roughness values and causes the turbulent heat transfer coefficient downstream of transition to be lower than the laminar value. Transition is delayed in these cases until the computed turbulent heat transfer is greater than or equal to the laminar value at this location.

A new ice growth algorithm which allows specification of arbitrary ice growth directions has been implemented. Current options include growth directions in the surface normal direction, flow direction and in the trajectory tangent directions. This algorithm will allow easy incorporation of future ice growth models.

Finally, a new optional 'pseudo' surface has been implemented which produces more realistic heat transfer for cases with large stagnation zones than the previous model. It is known that inviscid codes do not produce realistic surface velocity distributions for large concave forward facing regions such as a large glaze ice shape. Surface velocity and surface velocity gradients are overpredicted resulting in poor heat transfer prediction. A more realistic flow solution can be obtained using the inviscid panel codes by filling in the large concave region, essentially modelling it as a forward facing flat plate. A method has been implemented which scans the surface model for voids and fills them.

As evidence of the improved capabilities this model provides, the following case was run. A 21 inch NACA0012 was run at 0° angle of attack for 45 minutes using 10 second time steps, resulting in 270 time steps in the simulation. 10 seconds is smaller than the smallest time step used in the automated time step mode. The automated time step procedure used 83 time steps, an average of 32.5 seconds per time step.

Two conditions were run. Both had a velocity of 150 mph, liquid water content of 0.5 g/m^3, and a droplet diameter of 20 microns. The first case was a rime ice case with a total temperature of -8 °F while the second case was a glaze ice case with a total temperature of 28 °F.

The final shape of the rime ice case is shown in Figure 13.1. Although experimental data is not taken for this long of an icing time, the mass, shape and maximum thickness are proportional to shorter cases ran in the IRT to which LEWICE 1.6 has been compared. To demonstrate the high accuracy of the current model, the lower surface is reflected upwards to show the symmetry of this shape. This is shown in Figure 13.2. The symmetry of the ice is nearly per-

fect. It was unheard of to run the previous versions for more than 10 time steps, and even then accuracy of this level could not be obtained.

The glaze ice case, ran at a 28 °F total temperature, is shown in Figure 13.3. A 10 second time step was used for this case as well. Again, although experimental data is not available for this run, the horn growth proceeds logically as time progresses. Previous versions of LEWICE could only be run accurately for 5 or 6 time steps in glaze conditions, yet this version can easily handle 270 time steps with such large horns. Again, the lower surface is reflected upwards to show the symmetry of the solution. This result is shown in Figure 13.4.

The symmetry achieved numerically far exceeds the symmetry available experimentally for much shorter icing times. The difference between the upper and lower horn angles is 0.7° and the difference in horn length is 2 mm (4% of length). The symmetry achieved here shows that each module is being executed accurately. If any of the modules is not being performed correctly, then the shape would not be symmetric. This result provides confidence that the ice shape is being computed accurately for other cases.

Figure 13.1

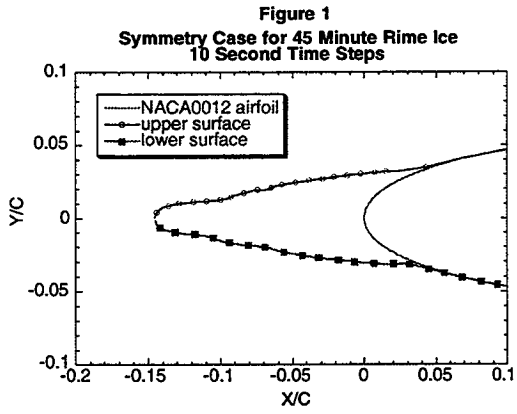


Figure 13.2

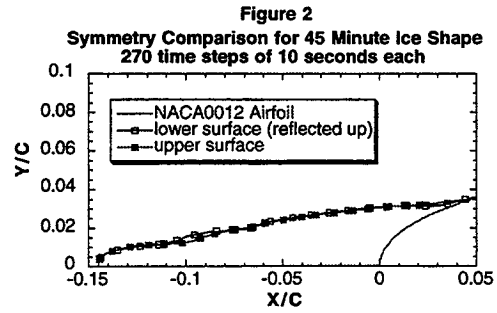


Figure 13.3

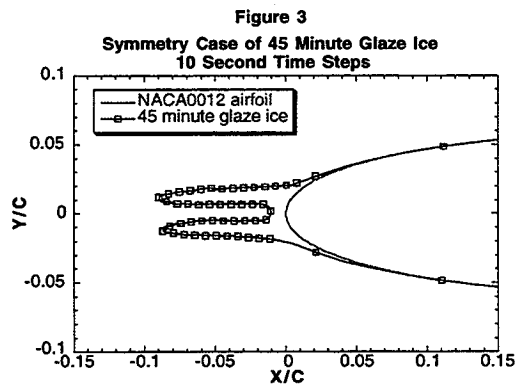
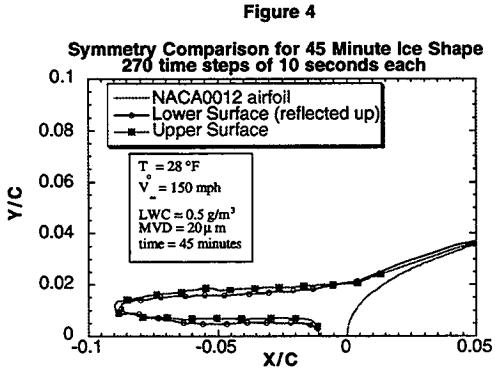


Figure 13.4



13.2.1 Roughness Prediction

The model used for predicting the distribution of roughness on an airfoil was developed in a previous paper, therefore a detailed derivation will not be presented. The height of roughness is determined by equating the force of the water flow with the force caused by surface tension. The runback model used to arrive at these forces was developed by Al-Khalil.[6]

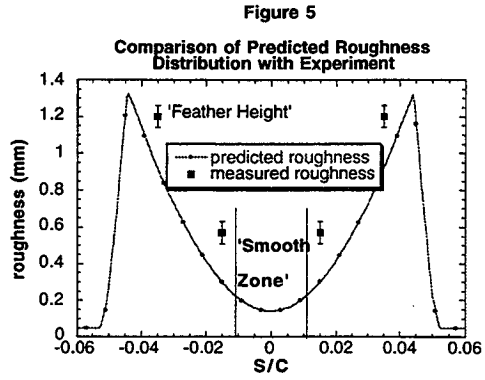
The advantages of this model are that it produces results which qualitatively agree with measured roughness data and that it relieves the burden on the user of having to input a crucial parameter for glaze ice. The disadvantage of the current model is that it does not predict some of the trends shown by the measured roughness data. The equation for the local height of roughness is

$$b = \left(\frac{4\sigma}{\rho\left(F\frac{d}{ds}V \right)^2} \right)^{\frac{1}{3}}$$

where the upper bound on this height is the local height of the ice shape. The use of this upper bound shows the first discrepancy in the experimental data. The measured roughness reached a stable height after two minutes and did not increase thereafter, whereas the predicted roughness will continue to increase.

An example of the roughness distribution obtained by this method is shown in Figure 13.5. The conditions represent the 'baseline' case used by Shin[3] in his IRT test. His measured data is represented by the squares. The size of the bars represents the margin of error in the data.

Figure 13.5



In that effort, he reported a 'smooth' zone near the leading edge, followed by a region of uniform roughness. This is qualitatively represented in the model, although the predicted distribution does not show an abrupt change in height. Shin then reports a feather height, which is also shown in this plot.

Despite the agreement of the roughness distribution, the distribution itself is not used in the code due to the numerical inaccuracy of the flow derivative. After several time steps, the quality of the solution decreases significantly if the distribution is used. Instead, an average roughness is used in the code. The equation for this is

$$b_{avg} = \frac{\int_0^s b_i ds}{\int_0^s ds}$$

This average roughness is then used as the sand-grain roughness for the entire airfoil. This average value is amazingly close to the measured roughness values found in Shin's experiment. In the test, parameterizations of LWC, velocity, temperature and time were performed to establish the variation of roughness with these parameters. The tests were conducted on a 21" chord NACA0012 at 0° angle of attack. The matrix centered around these conditions:

LWC = 0.5 g/m^3

V = 150 mph

T = 28 °F

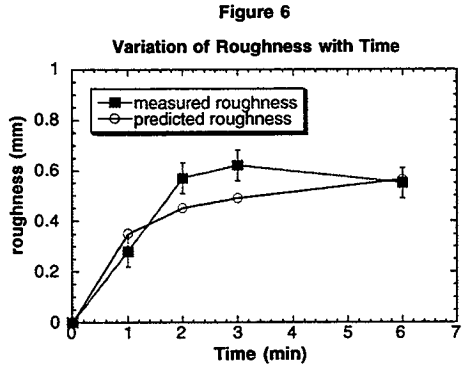
MVD = 20 μm

time = 6 min.

The parameterization used three velocities (150, 200, 250 mph), three temperatures (25, 28, 30 °F), four LWC's (0.5, 0.75, 1.0, 1.2 g/m^3) and four times (1, 2, 3, 6 minutes).

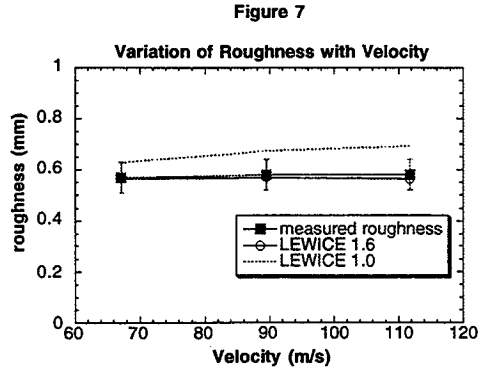
The first case shows the increase in roughness with icing time and is shown in Figure 13.6. For this plot, the experimental data is shown as solid squares with error bars. These error bars are 0.06 mm and represent an estimate by Shin of the error in the measuring technique. He also reported errors for each individual measurement, but it was more convenient to show a single error level for all measurements. The model predicts the level of roughness, although the increase with time is more gradual in the predicted values. It also continues to increase with time, whereas the experimental data shows no significant increase in roughness past 2 minutes. However, it is an improvement over previous models which had no variation in roughness with time.

Figure 13.6



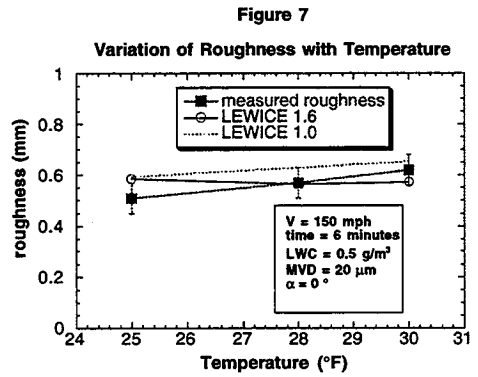
The variation of roughness with velocity is shown in Figure 13.7. The model shows excellent agreement with this parameter. This is an important result, as the previous model for sand-grain roughness predicted an increase in roughness, as shown by the dotted lines. The values are nearly constant over this range of conditions due to two competing effects in the theoretical model. The increase in velocity causes an increase in the amount of incoming water, which should increase the roughness. However, the higher velocity also causes a change in the flow derivative which results in a decrease in the roughness levels, especially near stagnation. The two effects cancel each other, causing essentially no change in roughness for these conditions.

Figure 13.7



The effect of temperature on roughness is shown in Figure 13.8. The roughness values predicted are reasonably close to the measured values, however the trend is opposite. The measured roughness increases very slightly with temperature, while the predicted levels show a minimum value at 28 °F, with increased values at both ends, especially at the 25 °F condition.

Figure 13.8

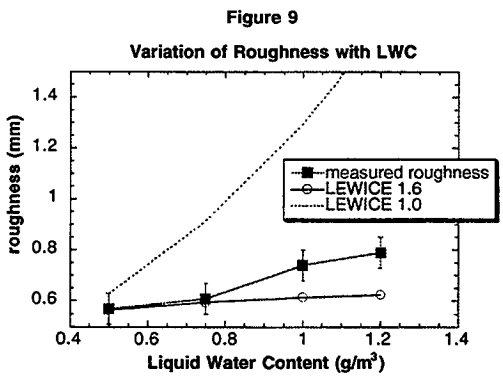


The increase in roughness from the 28 °F case to the 30 °F case is caused by a slight increase in surface tension with temperature. The increase in roughness from the 28°F case to the 25 °F case is caused by a decrease in the 'wetting factor' which is caused by a change in the predicted contact angle of the roughness element. This theory needs to be developed further to correct the predicted trend. The current roughness prediction and the previous corre-

lation are both within the experimental error over this range.

The variation of roughness with liquid water content (LWC) is shown in Figure 13.9. The roughness increases with LWC in both the predicted and measured values, but there is a significant (20%) underprediction in roughness at the higher LWC values. However, this model is a vast improvement over previous models which predicted extremely high roughness values at high LWC, as shown by the dotted lines.

Figure 13.9



13.2.2 Multi-Element Ice Accretion

The Hess-Smith potential flow code used by LEWICE 1.6 had the capability to predict flow on multi-element configurations. Often, the limitations of potential flow make these predictions much less accurate than more sophisticated models, but it may be adequate for the purpose of providing sample ice shapes. However, the trajectory and ice accretion routines were only applicable for single element geometries. As a result of the current effort, not only can LEWICE 1.6 perform multi-element trajectories, but these routines can also be converted for use with more accurate flow solvers.

LEWICE 1.6 performs multi-element trajectories by treating each element as a single entity. Impingement limits and collection efficiencies are determined on each body as though the other bodies are not there. Their influence on the trajectories is embedded in the flow solution, which takes into account all of the bodies. Any trajectory hits on other elements are treated as missed trajectories.

The hits on other bodies are, however, useful in determining the starting location of the next trajectory in the impingement limit search, especially hits on bodies which precede the one selected. For example, when the code looks for impingement limits on the flap(s), trajectory hits on the slat are useful in determining the starting location of the next trajectory. Routine MODE in LEWICE 1.6 determines if a trajectory hits or misses a body. It was modified so that it not only knew that a body was hit, but which one. As stated earlier, hits on other bodies are only used to select the starting location of the next trajectory.

An additional problem occurs especially on the main element of a multi-element airfoil. It is possible for trajectories to hit this element by passing above the slat as well as by passing below the slat. Therefore, for all bodies but the first one (the slat) LEWICE 1.6 will first look below the slat for an upper and lower impingement limit and determine one set of collection efficiencies for this set of impingement limits. LEWICE 1.6 will then look above the slat and attempt to find a second set of impingement limits. If two sets of limits are found, the two collection efficiency arrays are merged.

This process is made clearer by looking at Figures 13.10 and 13.11. Figure 13.10 shows the two sets of impingement limits on the main element of a sample slat and main element combination. The collection efficiency curve for this condition is shown in Figure 13.11. Both bodies are NACA0012 airfoils. This is not a realistic test case, but is simply representative of the code's capabilities. If the user knows that impingement is going to occur on other elements by travelling below the slat, the user can bypass this option as the code will run nearly twice as fast by bypassing this feature.

Figure 13.10

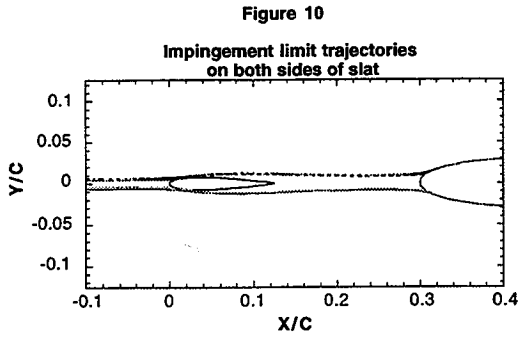
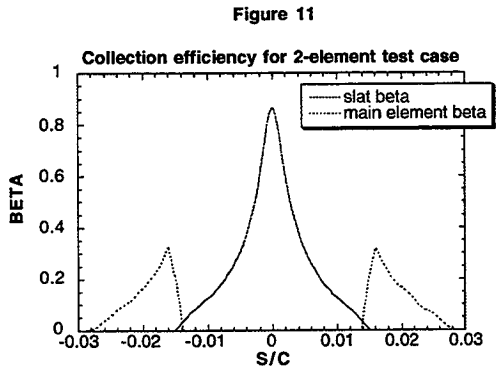


Figure 13.11



The process of converting LEWICE 1.6 to handle multi-element geometries was made more difficult by the addition of the multiple geometry scheme used to increase the accuracy of the code. After the trajectory routine is completed, LEWICE 1.6 creates a second set of panels for each element and resolves the flow field using this panel set. For a single element case, this second set of panels produces a smoother pressure distribution which increases the accuracy of the boundary layer integration. This is not always the case for multiple element geometries. Work is continuing on this routine so that this very useful feature can be used with multi-element geometries.

Once the collection efficiencies for each element has been found, and the flow recalculated, LEWICE 1.6 is ready to perform the boundary layer integration and the ice accretion. Once again, this procedure is performed on each element individually, without regard to the presence or influence of other elements. The geometry modification is performed on each element individually, hence the code does not currently check for different elements intersecting due to ice growth. In this case, the code will most likely crash when it tries to solve the flow field on the next time step.

A comparison between this code and experimental data taken on a 5-element Boeing 737-200 airfoil. The airfoil is shown in Figure 13.12. The experimental data was taken in the IRT in 1991 and is documented in reference (4). The conditions for the comparison were:

15° flap

V = 100 mph

T = 28 °F

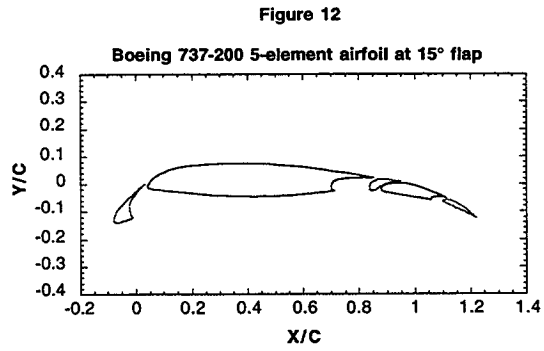
LWC = 0.92 g/m^3

MVD = 14.4 μm

0° angle of attack

time = 8 minutes

Figure 13.12



This case was selected because it was a glaze ice condition and ice was obtained on all five elements. The ice shape comparison is shown in Figs. 13.13-13.17. Due to the complex geometry, a Langmuir 'D' droplet distribution consisting of 7 drop sizes was used in the numerical prediction.

Figure 13.13

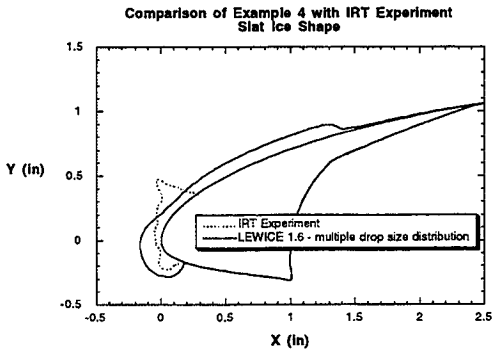


Figure 13.14

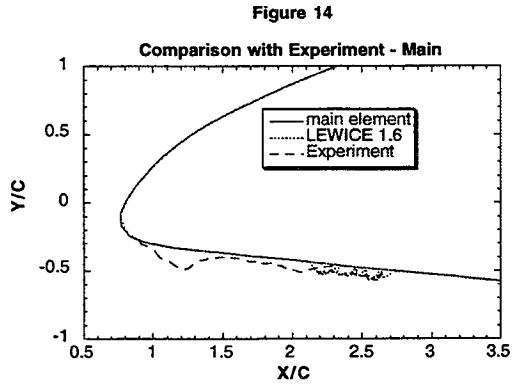


Figure 13.15

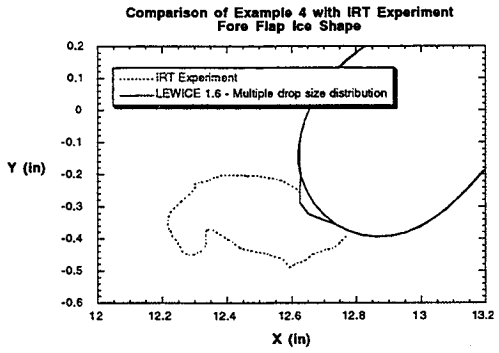


Figure 13.16

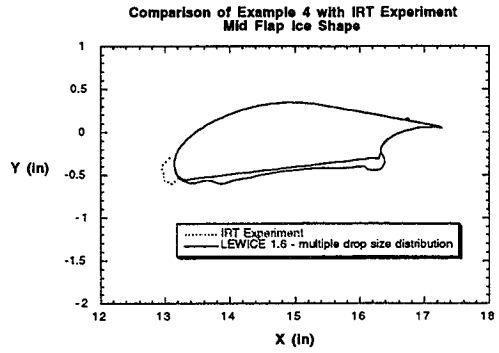
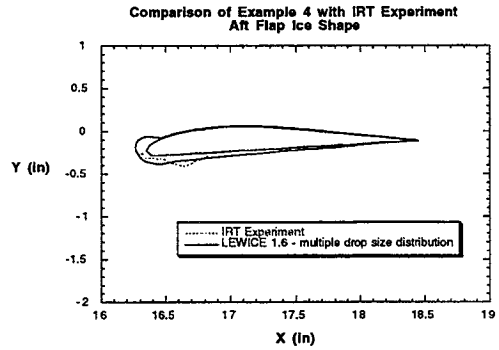


Figure 13.17



The predicted ice shape on the slat, shown in Figure 13.13 is representative of the very poor prediction when compared to the experimental data. This poor performance is somewhat surprising, as previous improvements to LEWICE had shown significant improvement in ice shape prediction for single element airfoils. The extent of predicted ice accretion in this figure is due to direct impingement, not due to extensive runback. This prediction is quite far from the experimental icing limits.

There are several reasons why this occurred. First, there are questions as to whether the configuration used in LEWICE 1.6 is the same as the test configuration. Airfoil coordinates are only available in the stowed reference plane. A rotation angle, x-offset, and gap distance are listed in a table in the report which allows coordinates for the test configuration to be computed. However, the 15° flap configuration is the only one of the three configurations for which the elements do not intersect with each other

when this transformation is applied. This leaves open the question as to the validity of all of the numbers in the table, including this configuration. By observing pictures of the airfoil in the 15° flap configuration, it is felt that the correct configuration was tested with LEWICE 1.6.

The second explanation for this discrepancy is the quality of the potential flow solution, which is plotted in Figure 13.18. The flow solution may not be adequate for this complex flow situation. A resolution of this question will not be known until more detailed analysis can be made with more accurate flow solvers.

Figure 13.18

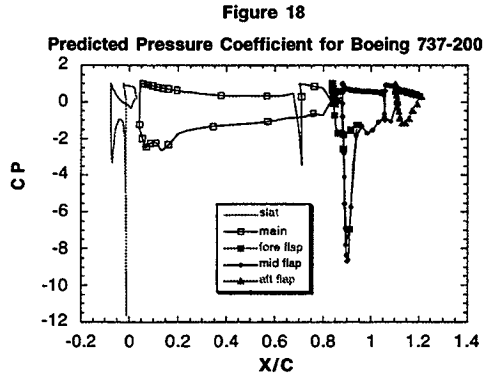
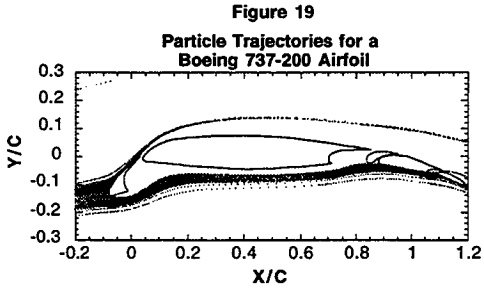


Figure 13.19



Finally, multiple time steps of this case were not ran due to the computational time needed. See Example 3 in the example case section for a mutli-time step simulation using a single time step. The changing iced geometry can change the collection efficiency distribution on a single element airfoil drastically and will no doubt have an impact on the results for this case.

The other elements also show major discrepancies between the predicted ice shape and the experimental ice shape. The ice shape for the main element is shown in Figure 13.14. In both the numerical and experimental shapes, a small amount of ice is produced just past the slat. The predicted shape is further down the chord than the experimental shape. This is consistent with the difference shown in the slat ice shapes. If the slat is raised slightly in space, more water will hit the main element and the impingement will be closer to the leading edge. It is unknown if this is due to errors in the flow or an error in the definition of the slat geometry.

The comparison for the fore flap is shown in Figure 13.15. The predicted iice shape for this element is also much less than the experimental shape. The prediction for the other two elements show the same trends. The prediction for the fore flap, shown in Figure 13.16, shows some significant runback freezing. The comparison for the aft flap shows the same trends as described earlier and are likely due to accuracy of the flow solution and use of a single icing time step.

14.0 Example Cases

14.1 Example 1: Single Body Case

```
 MS-317 AIRFOIL
 &LEW16
 IACCFL =    1
 ITIMFL =    0
 TSTOP  =  480.
 IFLO   =   48
 IMLSFL =    0
 ISTC   =    1
 NSTEPI =    4
 DSMN   =    3.D-4, 1.D-3, 5D-4, 7.D-4,
4D-4, 5.D-4, 5.D-4, 5.D-4, 5.D-4, 5.D-4
 DSMAX  =    5.D-4
 DDANG  =    5.D-4
 IDEN   =    0
 IRUNB  =    0
 NFLIT  =    1
 IGRID  =    0
 IRC    =    1
 SLWC   =    0.0
 NREAD  =    0
 IBOD   =    1
 LBOTH  =    0
 &END
 &S24Y1
 ILIFT  =    1
 IPARA  =    1
 IFIRST =    3
 ISECND =    3
 IPVOR  =    1
 INCLT  =    0
 CLT    =  0.0
 CCL    =  1.0
 IFLLL  =    1
 &END
 &TRAJ1
 GEPS   = 0.5E-04
 VEPS   = 1.E-03
 NEQ    =    4
 NPL    =   24
 NSEAR  =  199
 TSP    = 1.E-03
 CHORD  = 0.9144
 G      = 0.0
 PIT    = 0.0
 PITDOT = 0.0
 Y0LIM  = 1.E-04
 RHOP   = 1.0E3
 &END
 &DIST
 FLWC   = 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.00,0.0, 0.0, 0.0
 DPD    = 20., 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0
 CFP    = 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0
 &END
 &ICE1
 VINF   = 44.7
 LWC    = 0.92
 TINF   = 269.89
 PINF   = 94540.0
 RH     = 100.0
 TIME   = 0.0
 &END
 &LPRNT
 IPRT   = 1
 FPRT   = 1
 HPRT   = 1
 BPRT   = 1
 EPRT   = 0
 MPRT   = 0
 TPRT   = 1
 IRPRT  = 0
 &END
```

Notes: Data for a given variable should all be on one line, which the column spacing of this report does not permit. Since this is a single body, only the first value of DSMN is significant.

X, Y Coordinates for Example Cases 1 and 2

```
1.0000000    -0.0023600
0.9750000    -0.0000800
0.9500000     0.0003900
0.9250000    -0.0006300
0.9000000    -0.0028000
0.8750000    -0.0058400
0.8500000    -0.0095900
0.8250000    -0.0138800
0.8000000    -0.0185700
0.7750000    -0.0230900
0.7500000    -0.0278000
0.7250000    -0.0325500
0.7000000    -0.0373500
0.6750000    -0.0421400
```

0.6500000	-0.0467400
0.6250000	-0.0510600
0.6000000	-0.0550100
0.5750000	-0.0584500
0.5500000	-0.0613800
0.5250000	-0.0638900
0.5000000	-0.0659100
0.4750000	-0.0675500
0.4500000	-0.0688000
0.4250000	-0.0696700
0.4000000	-0.0701900
0.3750000	-0.0703600
0.3500000	-0.0702100
0.3250000	-0.0697800
0.3000000	-0.0690900
0.2750000	-0.0681200
0.2500000	-0.0668500
0.2250000	-0.0652700
0.2000000	-0.0633000
0.1750000	-0.0609900
0.1500000	-0.0581700
0.1250000	-0.0547700
0.1000000	-0.0505800
0.0750000	-0.0454100
0.0500000	-0.0386500
0.0375000	-0.0342300
0.0250000	-0.0286600
0.0125000	-0.0210500
0.0050000	-0.0136600
0.0020000	-0.0085700
0.0000000	0.0009900
0.0020000	0.0124800
0.0050000	0.0195000
0.0125000	0.0309900
0.0250000	0.0432200
0.0375000	0.0521000
0.0500000	0.0589300
0.0750000	0.0684000
0.1000000	0.0751100
0.1250000	0.0803300
0.1500000	0.0845400
0.1750000	0.0880400
0.2000000	0.0909600
0.2250000	0.0933900
0.2500000	0.0953600
0.2750000	0.0969400
0.3000000	0.0981500
0.3250000	0.0990100
0.3500000	0.0995200
0.3750000	0.0997200
0.4000000	0.0995600
0.4250000	0.0990900
0.4500000	0.0982600
0.4750000	0.0970000
0.5000000	0.0953500
0.5250000	0.0932300
0.5500000	0.0907300
0.5750000	0.0877700
0.6000000	0.0844800
0.6250000	0.0807900
0.6500000	0.0767200
0.6750000	0.0723200
0.7000000	0.0676300
0.7250000	0.0626900
0.7500000	0.0575500
0.7750000	0.0522500
0.8000000	0.0468700
0.8250000	0.0408700
0.8500000	0.0348600
0.8750000	0.0287800
0.9000000	0.0226400
0.9250000	0.0164700
0.9500000	0.0103100
0.9750000	0.0040400
1.0000000	-0.0023600

The results for the flow, heat transfer coefficient, collection efficiency, and trajectories are given for the first time step only. The ice shape shown was produced after 48 time steps of 10 seconds each. The user should be able to get a similar ice shape using the auto timestep option, which should use approximately 8 time steps.

The flow solution is plotted below in Figs. 14.1 and 14.2 for the entire airfoil and a second plot highlighting the flow at the leading edge.

Figure 14.1

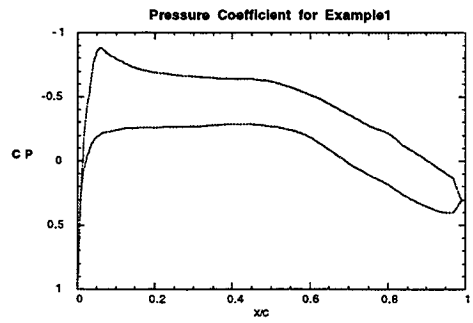
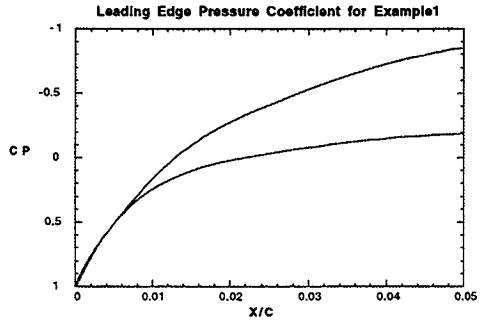


Figure 14.2



The heat transfer coefficients are plotted below in Figures 14.3 and 14.4, both at the leading edge, and for the whole airfoil.

Figure 14.3

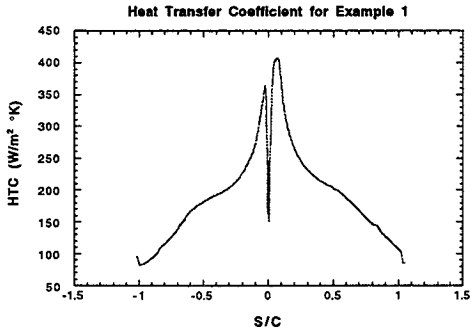
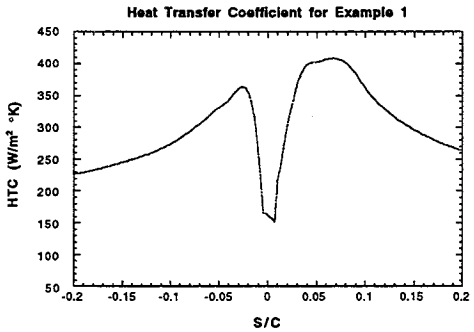
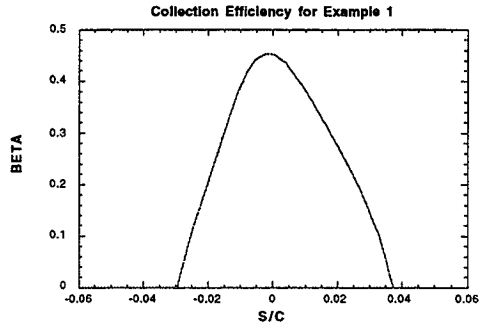


Figure 14.4



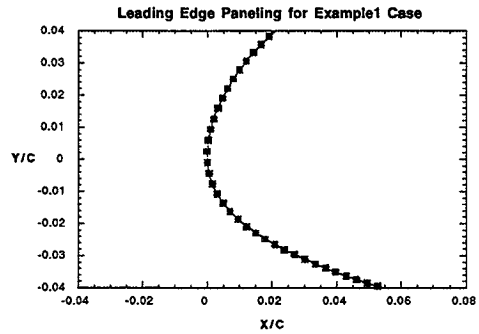
The collection efficiency curve is given below in Figure 14.5.

Figure 14.5



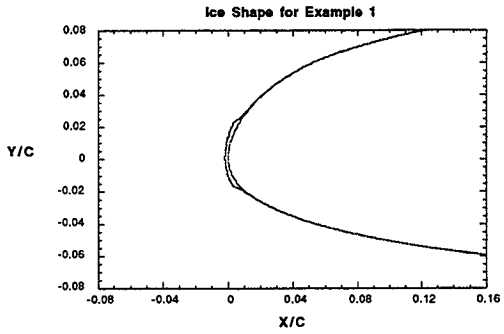
Even though the collection efficiency curve is smooth, the short impingement limits need to be verified. This is done by ensuring that there is adequate spacing near the leading edge. The spacing is not at all 'sparse' at the leading edge, as Figure 14.6 shows.

Figure 14.6



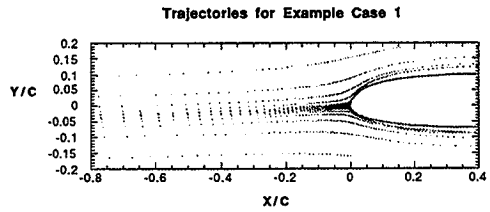
The ice shape after 8 minutes (480 seconds) is shown in Figure 14.7.

Figure 14.7



The trajectories are plotted i n Figure 14.8.

Figure 14.8



Note: A short explanation is needed for the user to understand how these last two plots were produced. All output in LEWICE 1.6 is sequential, meaning that data for the clean airfoil is given at the top of a file and data for the last time step is at the bottom, with output from intermediate time steps in-between. Because of this, the ice shape file contains every ice shape produced, but for clarity only the last time step ice shape was plotted. This was performed by deleting the intermediary data from within the spreadsheet used to create this plot. This task could also be performed with a text editor or word processor. The trajectory output file contains x,y coordinates of the trajectory position at each trajectory time step. The first trajectory is first, then the second, etc. In order to produce the plot, a scatter plot (which plots only data points and does not draw a line between points) is selected. Then a small marker size was selected, as it is useful to show the trajectories as small dots as the particle size is small. Even at that, the marker size is significantly larger than the actual drop size.

14.2 Example 2: Single Body with Multiple Drop Sizes

A useful feature of LEWICE 1.6 is its ability to model a multiple drop size distribution as would be seen in flight or in the Icing Research Tunnel. The following case is the same as example 1, except that it uses a Langmuir 'D' distribution instead of a single drop size.

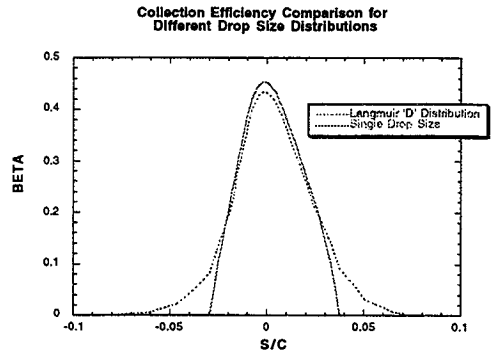
```
MS-317 AIRFOIL
&LEW16
IACCFL =    1
ITIMFL =    0
TSTOP  =  480.
IFLO   =   48
IMLSFL =    0
ISTC   =    1
NSTEPI =    4
DSMN   =    3.D-4,1.D-3,5D-4,7.D-4,4D-
4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
DSMAX  =    1.D-3
DDANG  =    7.D-4
IDEN   =    0
IRUNB  =    0
NFLIT  =    1
IGRID  =    0
IRC    =    1
SLWC   =    0.0
NREAD  =    0
IBOD   =    1
LBOTH  =    0
&END
&S24Y1
ILIFT  =    1
IPARA  =    1
IFIRST =    3
ISECND =    3
IPVOR  =    1
INCLT  =    0
CLT    =  0.0
CCL    =  1.0
IFLLL  =    1
&END
&TRAJ1
GEPS   = 0.5E-04
VEPS   = 1.E-03
```

```
NEQ    =    4
NPL    =   24
NSEAR  =  199
TSP    = 1.E-03
CHORD  = 0.9144
G      = 0.0
PIT    = 0.0
PITDOT = 0.0
Y0LIM  = 1.E-04
RHOP   = 1.0E3
&END
&DIST
FLWC   = 0.05, 0.1, 0.2, 0.3, 0.2,
0.1, 0.05, 0.0, 0.0, 0.0
DPD    = 6.2, 10.4, 14.2, 20., 27.4,
34.8, 44.4, 0.0, 0.0, 0.0
CFP    = 1.027, 1.0, 1.0, 1.0, 1.0,
1.0, 1.0, 0.0, 0.0, 0.0
&END
&ICE1
VINF   = 44.7
LWC    = 0.92
TINF   = 269.89
PINF   = 94540.0
RH     = 100.0
TIME   = 0.0
&END
&LPRNT
IPRT   = 1
FPRT   = 0
HPRT   = 0
BPRT   = 1
EPRT   = 0
MPRT   = 0
TPRT   = 1
IRPRT  = 0
&END
```

Notes: Due to the small drop sizes used, correction factors must be applied to account for slip flow effects. Since this case is identical to the previous case except for the trajectory portion, flow field and heat transfer results will not be shown, as they are identical (for the first time step) to the previous example case.

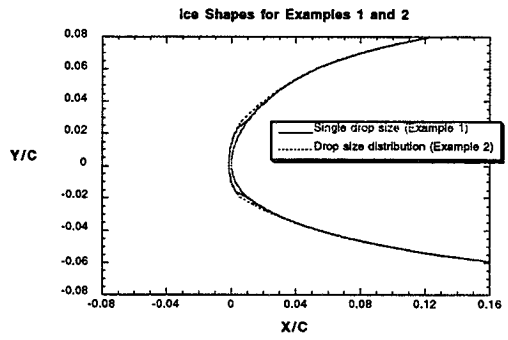
The collection efficiency curve for the first time step is shown in Figure 14.9 and is plotted in comparison to the Example 1 plot.

Figure 14.9



The impingement limits using a multi-drop size distribution are wider than in the single drop size mode, while the single drop size distribution captures the peak collection efficiency well. This is highly representative of other cases using single body geometries. However, as the ice shape plot demonstrates, as shown in Figure 14.10, the ice shape does not show much difference when using the drop size distribution. This happens because the ice shape is highly dependant on the maximum collection efficiency and the total water catch, which are not much different between the two cases. Also, the collection efficiency is dependant on the ice shape, so the multiple time steps ran have a large effect as well.

Figure 14.10

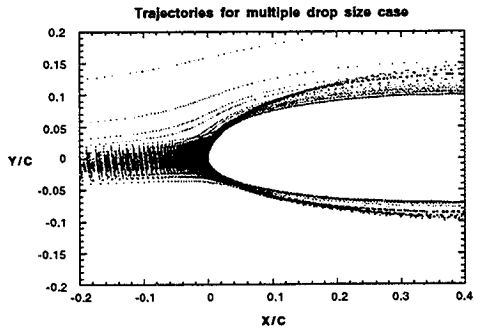


Note: A short explanation is again necessary for the user to understand how the collection efficiency plot was produced for this example. LEWICE 1.6 will output the cumulative collection efficiency after every drop size calculation. Therefore, the data plotted here is the last set of data points for the single time

step run. This set of points represents the local collection efficiency once all drop sizes have been calculated. The previous set of points represents the local collection efficiency up to that point in the calculation. This data is only useful if you want to know how much of the water is coming from a certain drop size. In equation form, the first set of points is $n_1*\beta_1$, the second set is $n_1*\beta_1 + n_2*\beta_2$, and so on. The last set is $\Sigma\, n_i*\beta_i$ for all drop sizes, which by definition is the collection efficiency.

The trajectory plot for Example 2 is shown in Figure 14.11.

Figure 14.11



It is difficult to see because of the plot scale, but on a larger scale you can see the different deflections from the surface that different particle sizes take.

14.3 Example 3: Multi-Body Simulation

This example models a Boeing 737-200 model which was ran in the IRT (7). There is a difficulty in running multi-body simulations such as this in LEWICE 1.6. It may be difficult for the user to select appropriate values for DSMN which will ensure a useful simulation without using excessive computational time. If you look closely, the user may notice regions where the paneling in this example is coarse. Due to the large number of time steps run, this was needed to obtain a solution in reasonable time.

```
BOEING 737-200 5 ELEMENT AIRFOIL
&LEW16
IACCFL =    1
ITIMFL =    0
TSTOP  =  480.
IFLO   =   48
IMLSFL =    0
ISTC   =    1
NSTEPI =    4
DSMN   =    4.D-4,1.D-3,5D-4,7.D-4,4D-
4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
DSMAX  =    1.D-3
DDANG  =    7.D-4
IDEN   =    0
IRUNB  =    0
NFLIT  =    1
IGRID  =    0
IRC    =    1
SLWC   =    0.0
NREAD  =    0
IBOD   =    5
LBOTH  =    0
&END
&S24Y1
ILIFT  =    1
IPARA  =    1
IFIRST =    3
ISECND =    3
IPVOR  =    1
INCLT  =    0
CLT    =  0.0
CCL    =  1.0
IFLLL  =    1
&END
&TRAJ1
GEPS   = 0.5E-04
VEPS   = 1.E-03
NEQ    =    4
NPL    =   24
NSEAR  =  199
TSP    = 1.E-03
CHORD  = 0.4572
G      = 0.0
PIT    = 0.0
PITDOT = 0.0
Y0LIM  = 1.E-04
RHOP   = 1.0E3
&END
&DIST
FLWC   = 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0
DPD    = 14.4, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0
CFP    = 1.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0
```

```
&END
&ICE1
VINF   = 44.7
LWC    = 0.92
TINF   = 269.89
PINF   = 94540.0
RH     = 100.0
TIME   = 0.0
&END
&LPRNT
IPRT   = 1
FPRT   = 1
HPRT   = 1
BPRT   = 1
EPRT   = 0
MPRT   = 0
TPRT   = 1
IRPRT  = 0
&END
```

Note: Output was not obtained for all time steps due to the excessive amount of space required for the output files. A case was ran using TSTOP=10. and IFLO=1 using the printout flags on, then a second case was ran using TSTOP=480 and IFLO=48 with all flags except for ice shape print out turned off. This saves much space and computational time to only print out the information which is desired.

The coordinates are listed for each body.

Slat

```
 0.0319900    -0.0003880
 0.0260850    -0.0055260
 0.0220870    -0.0097870
 0.0162090    -0.0161470
 0.0108823    -0.0220130
 0.0059837    -0.0275150
 0.0014036    -0.0327710
-0.0029500    -0.0378880
-0.0072060    -0.0430130
-0.0114030    -0.0481970
-0.0155520    -0.0534540
-0.0196240    -0.0587490
-0.0217070    -0.0615210
-0.0233800    -0.0667930
-0.0244600    -0.0714290
-0.0251850    -0.0761260
-0.0255380    -0.0808990
-0.0255030    -0.0857520
-0.0250570    -0.0906980
-0.0241760    -0.0957410
-0.0228300    -0.1009010
-0.0209820    -0.1061910
-0.0186030    -0.1116010
-0.0156360    -0.1171710
-0.0120500    -0.1228810
-0.0166550    -0.1245610
-0.0212590    -0.1262510
-0.0262240    -0.1281610
-0.0312040    -0.1300310
-0.0362020    -0.1318710
-0.0377570    -0.1324210
-0.0393130    -0.1329810
-0.0424330    -0.1340710
-0.0450400    -0.1349610
-0.0473650    -0.1357210
-0.0496980    -0.1364610
-0.0517050    -0.1370710
-0.0535020    -0.1376010
-0.0553060    -0.1381010
-0.0568730    -0.1385110
-0.0582860    -0.1388610
-0.0597050    -0.1391810
-0.0609480    -0.1394410
-0.0621970    -0.1396710
-0.0632690    -0.1398510
-0.0643460    -0.1400010
-0.0652700    -0.1401010
-0.0661990    -0.1401810
-0.0670000    -0.1402310
-0.0678050    -0.1402510
-0.0685040    -0.1402410
-0.0692040    -0.1402110
-0.0698160    -0.1401610
-0.0704270    -0.1400810
-0.0709660    -0.1399910
-0.0715020    -0.1398810
-0.0719780    -0.1397610
-0.0724500    -0.1396110
-0.0728730    -0.1394510
-0.0732900    -0.1392810
-0.0740370    -0.1388910
-0.0747060    -0.1384510
-0.0753040    -0.1379610
-0.0758370    -0.1374210
-0.0763220    -0.1368310
-0.0767520    -0.1361910
-0.0768620    -0.1360110
-0.0770400    -0.1356910
-0.0771070    -0.1355610
```

-0.0775530	-0.1346210
-0.0779280	-0.1336110
-0.0780940	-0.1330710
-0.0782370	-0.1325310
-0.0783670	-0.1319610
-0.0784740	-0.1313710
-0.0785660	-0.1307510
-0.0786350	-0.1301310
-0.0786860	-0.1294610
-0.0787150	-0.1288010
-0.0787220	-0.1280810
-0.0787060	-0.1273610
-0.0786660	-0.1266010
-0.0786030	-0.1258310
-0.0785110	-0.1250110
-0.0783970	-0.1241910
-0.0782500	-0.1233110
-0.0780810	-0.1224410
-0.0778750	-0.1215110
-0.0776470	-0.1205810
-0.0773770	-0.1195810
-0.0770860	-0.1185910
-0.0767500	-0.1175310
-0.0763920	-0.1164810
-0.0759840	-0.1153610
-0.0755550	-0.1142510
-0.0750720	-0.1130610
-0.0745680	-0.1118810
-0.0740050	-0.1106310
-0.0734230	-0.1093810
-0.0727780	-0.1080610
-0.0721120	-0.1067510
-0.0713800	-0.1053710
-0.0706290	-0.1039910
-0.0698070	-0.1025410
-0.0689670	-0.1011010
-0.0680540	-0.0995810
-0.0671210	-0.0980710
-0.0661130	-0.0964910
-0.0650860	-0.0949110
-0.0639800	-0.0932650
-0.0628570	-0.0916290
-0.0616540	-0.0899180
-0.0604340	-0.0882210
-0.0591370	-0.0864590
-0.0578240	-0.0847110
-0.0563680	-0.0828150
-0.0548940	-0.0809330
-0.0519380	-0.0772680
-0.0489950	-0.0737440
-0.0458350	-0.0700930
-0.0424470	-0.0663180
-0.0388320	-0.0624230
-0.0349720	-0.0583990
-0.0308690	-0.0542520
-0.0265130	-0.0499780
-0.0218960	-0.0455770
-0.0170160	-0.0410490
-0.0118740	-0.0363980
-0.0064550	-0.0316200
-0.0007600	-0.0267180
0.0052184	-0.0216900
0.0114770	-0.0165440
0.0180210	-0.0112750
0.0248620	-0.0058910
0.0319900	-0.0003880

Main

0.8513300	0.0228280
0.8388900	0.0228280
0.8222200	0.0228280
0.8055600	0.0228280
0.7972200	0.0228280
0.7901700	0.0224610
0.7831100	0.0220360
0.7759900	0.0218780
0.7688500	0.0217090
0.7617200	0.0212130
0.7466600	0.0203130
0.7374000	0.0186330
0.7276400	0.0160140
0.7218300	0.0145730
0.7165100	0.0126090
0.7115000	0.0100900
0.7071300	0.0071730
0.7035600	0.0036450
0.7014500	-0.0004880
0.7011500	-0.0055560
0.7011500	-0.0111110
0.7011500	-0.0166670
0.7011500	-0.0191670
0.7022000	-0.0207830
0.7051900	-0.0218280
0.7092200	-0.0220000
0.7107500	-0.0238490
0.7122800	-0.0256990
0.7021400	-0.0271010
0.6920000	-0.0285040
0.6741700	-0.0301620
0.6563300	-0.0318210

0.6469700	-0.0329670
0.6361700	-0.0336900
0.6036100	-0.0365610
0.5748300	-0.0388400
0.5485900	-0.0406480
0.5239700	-0.0420680
0.5001400	-0.0431560
0.4763300	-0.0439490
0.4515800	-0.0444640
0.4383900	-0.0446180
0.4120300	-0.0446960
0.3986300	-0.0446070
0.3786100	-0.0442610
0.3565700	-0.0435940
0.3335700	-0.0426020
0.3091400	-0.0412540
0.2827700	-0.0395010
0.2537800	-0.0372770
0.2213800	-0.0344940
0.1845100	-0.0310320
0.1478200	-0.0273930
0.1138700	-0.0238180
0.0805000	-0.0200360
0.0557060	-0.0170380
0.0478570	-0.0148590
0.0450780	-0.0133170
0.0430530	-0.0109000
0.0420470	-0.0079590
0.0419990	-0.0061170
0.0424310	-0.0041910
0.0434140	-0.0017230
0.0450410	0.0013180
0.0487960	0.0068520
0.0520620	0.0109060
0.0557500	0.0149780
0.0596280	0.0188190
0.0633610	0.0221360
0.0669780	0.0249990
0.0691830	0.0265760
0.0716610	0.0282430
0.0744330	0.0299990
0.0775110	0.0318460
0.0809110	0.0337830
0.0846940	0.0358390
0.0889220	0.0380320
0.0974220	0.0421780
0.1068700	0.0464670
0.1178900	0.0511520
0.1302500	0.0560830
0.1410000	0.0593500
0.1574600	0.0618220
0.1750100	0.0641390
0.1936100	0.0662720
0.2132500	0.0682060
0.2338700	0.0699280
0.2554300	0.0714110
0.2778600	0.0726440
0.3010800	0.0736170
0.3250300	0.0743060
0.3631200	0.0748060
0.3886600	0.0747560
0.4130900	0.0744280
0.4370200	0.0738220
0.4607800	0.0729170
0.4839700	0.0717330
0.5069700	0.0702670
0.5299600	0.0685060
0.5531500	0.0664280
0.5767200	0.0640110
0.6009400	0.0612390
0.6260000	0.0580720
0.6521100	0.0544730
0.6795600	0.0503850
0.7032200	0.0466790
0.7387200	0.0410880
0.7762800	0.0351860
0.8138300	0.0292830
0.8513300	0.0228280

Fore Flap

0.9485800	0.0091966
0.9456100	0.0090326
0.9416800	0.0087703
0.9378600	0.0084588
0.9341500	0.0080994
0.9306100	0.0076909
0.9271100	0.0072434
0.9237300	0.0067518
0.9204500	0.0062155
0.9172200	0.0056352
0.9140500	0.0050066
0.9108800	0.0043307
0.9078100	0.0035946
0.9061600	0.0031801
0.9045100	0.0027506
0.9016000	0.0019654
0.8989200	0.0011686
0.8962900	0.0003329
0.8937200	-0.0005363
0.8912500	-0.0014387

0.8888400	-0.0023621	0.8373500	-0.0157080
0.8866000	-0.0033090	0.8370800	-0.0148490
0.8844100	-0.0042640	0.8368000	-0.0139700
0.8823800	-0.0052300	0.8366500	-0.0130840
0.8804700	-0.0061960	0.8365400	-0.0121870
0.8786700	-0.0071590	0.8364900	-0.0112850
0.8769800	-0.0081120	0.8364900	-0.0103190
0.8754600	-0.0090480	0.8365200	-0.0099000
0.8740600	-0.0099670	0.8365600	-0.0094150
0.8727600	-0.0108790	0.8367000	-0.0083170
0.8720500	-0.0114520	0.8368900	-0.0072980
0.8712800	-0.0120380	0.8371400	-0.0062890
0.8703300	-0.0128600	0.8374300	-0.0052900
0.8696200	-0.0135120	0.8378300	-0.0043040
0.8689000	-0.0141990	0.8382400	-0.0033240
0.8674600	-0.0155670	0.8387000	-0.0023544
0.8658400	-0.0170290	0.8392700	-0.0013966
0.8644700	-0.0182560	0.8398400	-0.0004452
0.8632200	-0.0193270	0.8405200	0.0004926
0.8620400	-0.0202590	0.8412600	0.0014199
0.8609200	-0.0210720	0.8420500	0.0023376
0.8598600	-0.0217820	0.8429000	0.0032431
0.8588700	-0.0224010	0.8438000	0.0041362
0.8578800	-0.0229370	0.8447500	0.0050169
0.8569600	-0.0234030	0.8458200	0.0058791
0.8559800	-0.0237960	0.8469400	0.0067252
0.8550600	-0.0241290	0.8481700	0.0075504
0.8541500	-0.0244030	0.8494000	0.0083612
0.8532400	-0.0246190	0.8507500	0.0091468
0.8522800	-0.0247790	0.8522000	0.0099073
0.8513200	-0.0248750	0.8537000	0.0106432
0.8504300	-0.0249120	0.8552600	0.0113534
0.8495900	-0.0248980	0.8568700	0.0120353
0.8487000	-0.0248230	0.8585900	0.0126825
0.8478700	-0.0246940	0.8603700	0.0132973
0.8469900	-0.0245000	0.8622500	0.0138787
0.8461200	-0.0242450	0.8641900	0.0144350
0.8453000	-0.0239360	0.8661200	0.0149212
0.8444400	-0.0235680	0.8681100	0.0153541
0.8436300	-0.0231490	0.8701500	0.0157374
0.8428800	-0.0226810	0.8722400	0.0160717
0.8421400	-0.0221620	0.8743200	0.0163610
0.8414500	-0.0215980	0.8765200	0.0165991
0.8407700	-0.0209860	0.8788200	0.0167858
0.8401400	-0.0203350	0.8811800	0.0169241
0.8395200	-0.0196410	0.8835900	0.0170140
0.8390200	-0.0189160	0.8861600	0.0170463
0.8385100	-0.0181550	0.8888400	0.0170231
0.8380700	-0.0173640	0.8916200	0.0169436
0.8376800	-0.0165480	0.8945200	0.0168068

0.8976300	0.0165954
0.8996300	0.0164496
0.9016800	0.0162987
0.9052300	0.0159639
0.9087800	0.0155741
0.9124900	0.0151134
0.9166900	0.0145439
0.9215000	0.0138341
0.9269600	0.0129737
0.9327600	0.0120006
0.9395400	0.0108219
0.9483100	0.0092487
0.9485800	0.0091966

Mid Flap

1.1150000	-0.0526090
1.1120000	-0.0516120
1.1081000	-0.0505120
1.1060000	-0.0498840
1.1037000	-0.0492800
1.1013000	-0.0487200
1.0988000	-0.0482110
1.0963000	-0.0477580
1.0938000	-0.0473650
1.0913000	-0.0470340
1.0888000	-0.0467680
1.0864000	-0.0465670
1.0840000	-0.0464310
1.0818000	-0.0463590
1.0796000	-0.0463450
1.0776000	-0.0463880
1.0758000	-0.0464800
1.0741000	-0.0466080
1.0732000	-0.0467080
1.0723000	-0.0468250
1.0706000	-0.0470900
1.0691000	-0.0473780
1.0677000	-0.0477010
1.0664000	-0.0480440
1.0652000	-0.0484050
1.0642000	-0.0487860
1.0632000	-0.0491850
1.0624000	-0.0496030
1.0617000	-0.0500250
1.0610000	-0.0504670
1.0604000	-0.0509150
1.0599000	-0.0513850
1.0595000	-0.0518660
1.0591000	-0.0523430
1.0588000	-0.0528500
1.0585000	-0.0533780
1.0583000	-0.0539820
1.0582000	-0.0542290
1.0581000	-0.0548590
1.0579000	-0.0554150
1.0578000	-0.0559710
1.0576000	-0.0565260
1.0574000	-0.0575260
1.0573000	-0.0577780
1.0510000	-0.0567590
1.0447000	-0.0557450
1.0390000	-0.0547160
1.0334000	-0.0537400
1.0277000	-0.0527700
1.0220000	-0.0517970
1.0134000	-0.0503380
1.0050000	-0.0488780
0.9992700	-0.0479050
0.9935900	-0.0469320
0.9879100	-0.0459590
0.9821100	-0.0449440
0.9508200	-0.0395360
0.9195200	-0.0341280
0.8899200	-0.0290190
0.8885600	-0.0287810
0.8878200	-0.0286550
0.8870900	-0.0284970
0.8863200	-0.0282660
0.8855600	-0.0279750
0.8848800	-0.0276430
0.8842100	-0.0272520
0.8835500	-0.0268050
0.8829100	-0.0263040
0.8823400	-0.0257650
0.8818400	-0.0251890
0.8813500	-0.0245660
0.8808700	-0.0238980
0.8805200	-0.0232180
0.8801700	-0.0224990
0.8798900	-0.0217610
0.8796700	-0.0210060
0.8795200	-0.0202400
0.8794200	-0.0194650
0.8793900	-0.0186880
0.8793600	-0.0178980
0.8794400	-0.0171280
0.8796000	-0.0163070
0.8796900	-0.0159660
0.8797800	-0.0156270
0.8798000	-0.0155330

0.8800700	-0.0147360
0.8803900	-0.0139500
0.8808300	-0.0131880
0.8812700	-0.0124180
0.8817700	-0.0116510
0.8823800	-0.0109040
0.8830000	-0.0101430
0.8837300	-0.0093990
0.8845700	-0.0086690
0.8854200	-0.0079260
0.8864400	-0.0072120
0.8874600	-0.0064810
0.8886000	-0.0057650
0.8898500	-0.0050620
0.8912200	-0.0043750
0.8926500	-0.0036890
0.8942400	-0.0030320
0.8959000	-0.0023784
0.8976700	-0.0017427
0.8995500	-0.0011273
0.9016100	-0.0005474
0.9037100	0.0000234
0.9059900	0.0005554
0.9083800	0.0010591
0.9108800	0.0015328
0.9135500	0.0019604
0.9163200	0.0023516
0.9192000	0.0027056
0.9222000	0.0030175
0.9253500	0.0032702
0.9286100	0.0034726
0.9318600	0.0036300
0.9352700	0.0037315
0.9387800	0.0037732
0.9423400	0.0037449
0.9458800	0.0036693
0.9494700	0.0035297
0.9530900	0.0033233
0.9567100	0.0030630
0.9603600	0.0027320
0.9639400	0.0023567
0.9675700	0.0019079
0.9711200	0.0014115
0.9746500	0.0008529
0.9781200	0.0002455
0.9815100	-0.0004107
0.9848200	-0.0011173
0.9880200	-0.0018569
0.9911300	-0.0026450
0.9941300	-0.0034600
0.9971600	-0.0043460
1.0003000	-0.0053200
1.0035000	-0.0063860
1.0068000	-0.0075400
1.0104000	-0.0088440
1.0144000	-0.0103410
1.0187000	-0.0120560
1.0236000	-0.0140300
1.0298000	-0.0165630
1.0425000	-0.0218650
1.0567000	-0.0276050
1.0665000	-0.0316360
1.0762000	-0.0356850
1.0845000	-0.0391550
1.0929000	-0.0426600
1.1047000	-0.0476710
1.1085000	-0.0492790
1.1120000	-0.0508160
1.1150000	-0.0526090

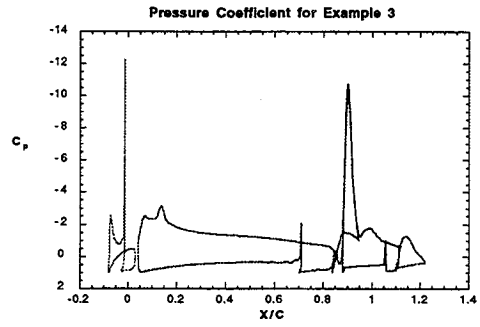
Aft Flap

1.2208000	-0.1222180
1.2146000	-0.1193280
1.2076000	-0.1160480
1.2009000	-0.1128880
1.1931000	-0.1092280
1.1849000	-0.1053980
1.1767000	-0.1015780
1.1688000	-0.0978380
1.1610000	-0.0941980
1.1528000	-0.0903780
1.1447000	-0.0865550
1.1367000	-0.0828030
1.1295000	-0.0794540
1.1219000	-0.0759030
1.1132000	-0.0718060
1.1046000	-0.0677660
1.1029000	-0.0669850
1.1025000	-0.0667800
1.1021000	-0.0665750
1.1016000	-0.0662970
1.1013000	-0.0659570
1.1010000	-0.0655400
1.1007000	-0.0650990
1.1005000	-0.0646390
1.1004000	-0.0641340
1.1003000	-0.0636250
1.1003000	-0.0631190
1.1003000	-0.0626270
1.1005000	-0.0621560

1.1007000	-0.0616500
1.1007000	-0.0614560
1.1009000	-0.0612410
1.1013000	-0.0606540
1.1018000	-0.0601610
1.1023000	-0.0597070
1.1028000	-0.0592850
1.1034000	-0.0588920
1.1041000	-0.0585580
1.1049000	-0.0582500
1.1057000	-0.0579660
1.1066000	-0.0577040
1.1077000	-0.0575290
1.1088000	-0.0573750
1.1100000	-0.0572420
1.1113000	-0.0571950
1.1128000	-0.0572030
1.1145000	-0.0572970
1.1162000	-0.0574140
1.1181000	-0.0576530
1.1191000	-0.0577990
1.1201000	-0.0579620
1.1219000	-0.0583060
1.1240000	-0.0587540
1.1261000	-0.0592830
1.1284000	-0.0599260
1.1308000	-0.0606610
1.1333000	-0.0614970
1.1359000	-0.0624380
1.1385000	-0.0634590
1.1411000	-0.0645640
1.1437000	-0.0657250
1.1462000	-0.0669450
1.1487000	-0.0682270
1.1512000	-0.0695390
1.1534000	-0.0708500
1.1556000	-0.0721890
1.1577000	-0.0735020
1.1607000	-0.0754850
1.1639000	-0.0777110
1.1656000	-0.0788680
1.1672000	-0.0800730
1.1687000	-0.0811850
1.1702000	-0.0823130
1.1729000	-0.0844220
1.1753000	-0.0863080
1.1803000	-0.0902460
1.1844000	-0.0934980
1.1880000	-0.0962680
1.1930000	-0.1002880
1.1981000	-0.1043080
1.2032000	-0.1083180
1.2083000	-0.1123380
1.2122000	-0.1154380
1.2164000	-0.1187680
1.2208000	-0.1222180

The flow solution for this case is shown in Figure 14.12.

Figure 14.12



This plot can be compared to Figure 13.18, which was ran using different values for DSMN.

The heat transfer coefficients for each body are shown in Figures 14.13 - 14.17

Figure 14.13

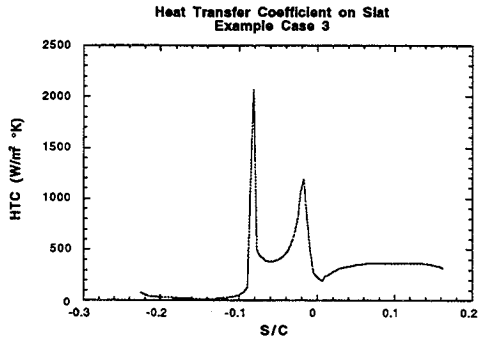


Figure 14.14

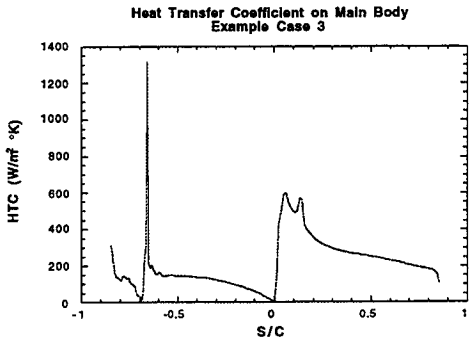


Figure 14.15

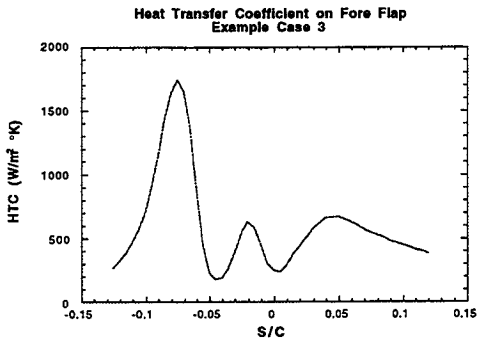


Figure 14.16

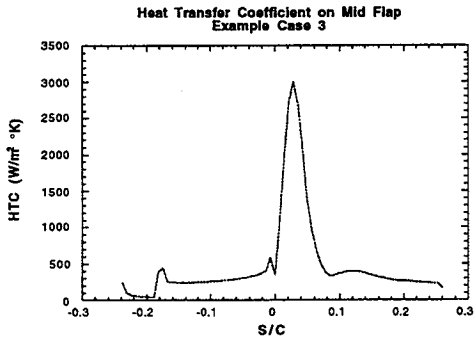
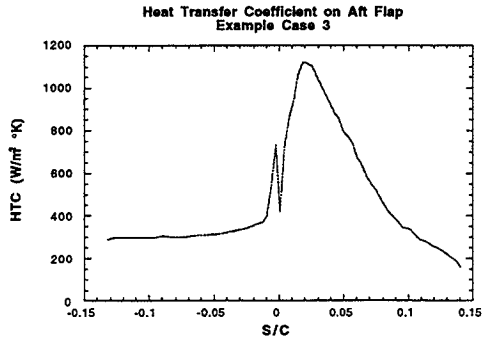


Figure 14.17



The chord in each case is the overall chord value, which in this case is 0.4572 m (18 in.). The collection efficiencies for this case are plotted in Figures 14.18-14.21. Note that there is no plot for the main element, which did not have any impingement for either example 3 or 4.

Figure 14.18

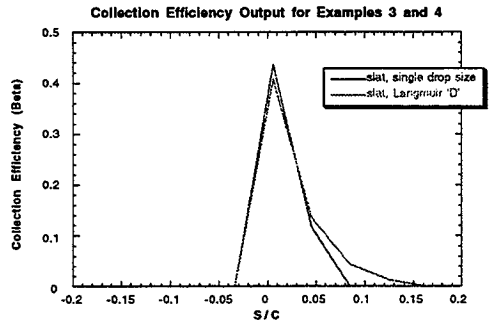
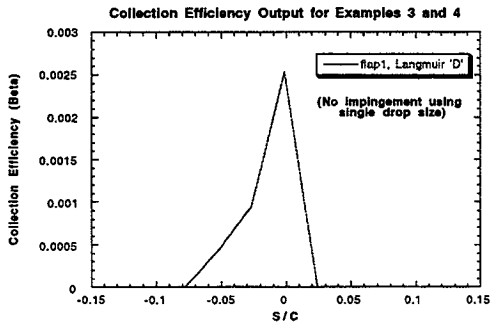


Figure 14.19



Note that there is no impingement on the first flap when using a single drop size.

Figure 14.20

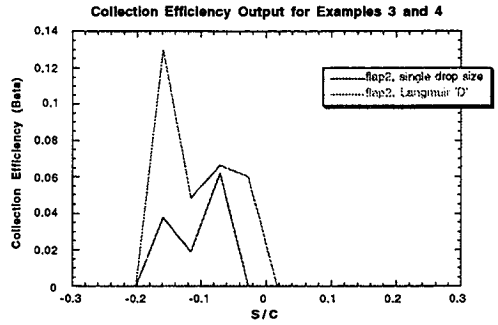
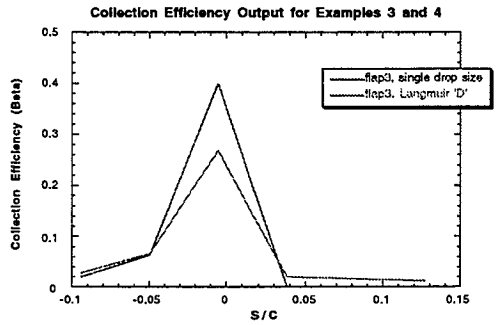


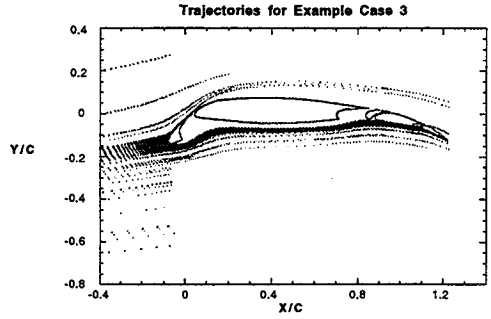
Figure 14.21



The slat acts very much like a single body geometry in that there is a small difference between the MVD value and the value using a distribution. The flaps show the largest deviation as more water catch is deflected to the second flap from the third. The first and second flaps show more water catch using a distribution while the water catch on the third flap is less.

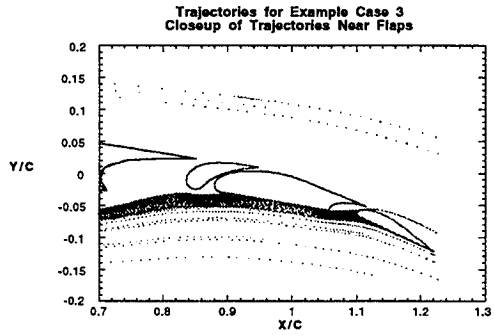
The trajectories for Example 3 are shown in Figure 14.22.

Figure 14.22



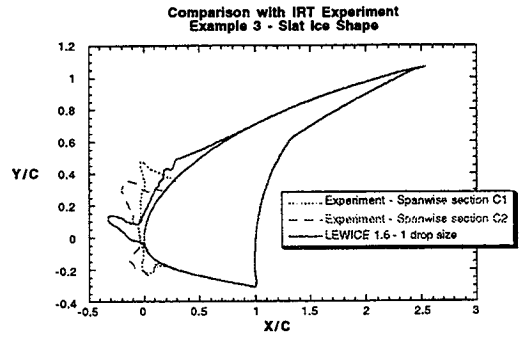
You may notice that some trajectories stop sooner than others in this plot. When the code is looking for impingement on the slat, it stops looking once it passes the slat. Due to the way the code is set up, it ignores hits on bodies other than the one it is currently trying to find impingement for. Also notice how it tracks the particles as they pass between the second and third flaps. Figure 14.23 shows a close-up of the trajectories near the flaps.

Figure 14.23



The ice shape for Example 3 is shown in Figures 14.24 - 14.28.

Figure 14.24



This plot shows that the heat transfer coefficient on the lower surface is over predicted, causing the lower horn to be much larger and closer to stagnation than the experiment. The impingement limits are also shifted aft on the section, indicating that the flow may not be modelled properly with this code.

Note: The experimental data provided is referenced to the 0° flap deflection coordinates and is in inches. The LEWICE 1.6 results were rotated and dimensionalized to produce this plot. To reproduce this plot, the user must also rotate and dimensionalize the data. The rotation equations are

$$x_{new} = \frac{x_{old} + y_{old}\tan\alpha}{1 + (\tan\alpha)^2}$$

and

$$y_{new} = \frac{y_{old} - x_{old}\tan\alpha}{1 + (\tan\alpha)^2}$$

where α = rotation angle of that body. For this case, the rotation angles are as follows: slat = 28.31°, main = 0°, fore flap = 4.1°, mid flap = 14.7°, aft flap = 30°. The chord is 18 inches, and then the coordinates need to be translated such that they align in space with the 0° flap coordinates. The following table gives the trailing edge points for each body.

Body	x	y
Slat	2.538	1.068326
Main	15.324045	0.410907
Fore Flap	14.603322	0.343473
Mid Flap	17.269679	0.049394
Aft Flap	18.446431	-0.112893

Figure 14.25 shows only the ice shape from the experiment, as no ice was formed in LEWICE 1.6 for this case.

Figure 14.25

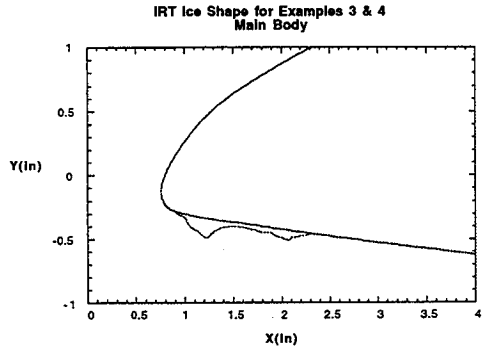


Figure 14.26 also shows only the ice shape from the experiment, as LEWICE 1.6 did not predict any ice on this body. However, in the experiment, only one of six span-wise sections had any ice, showing the variability of the experiment and possible three-dimensional effects.

Figure 14.26

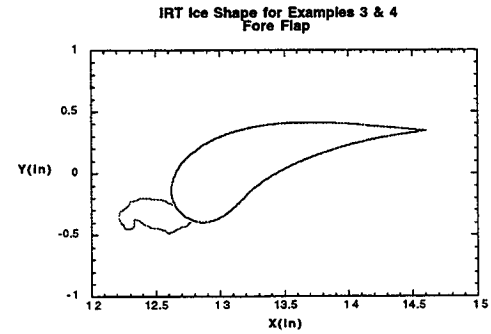


Figure 14.27

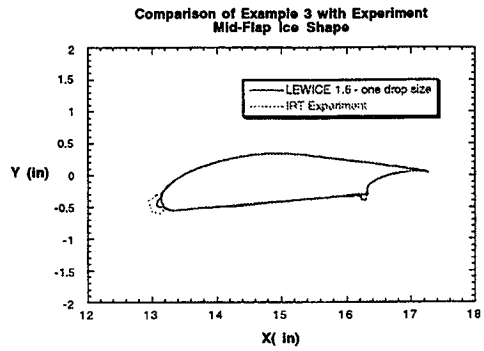
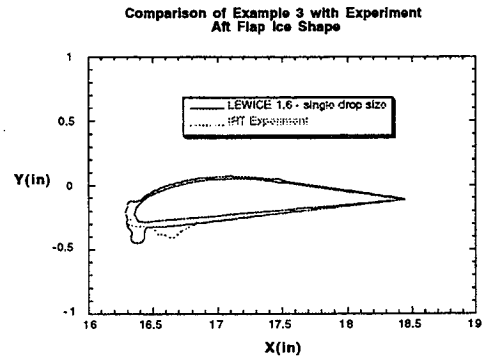


Figure 14.28



The mid-flap shape is smaller at the leading edge and there is a second impingement site near the back, This is consistent with the trends shown for previous elements. The last figure shows a significant deviation from the ice shape in the experiment. There is a shift in the impingement limits consistent with the shift shown in the slat results. Also, there is a considerable amount more water catch than in the experiment. Finally, the heat transfer coefficient is overpredicted at each location.

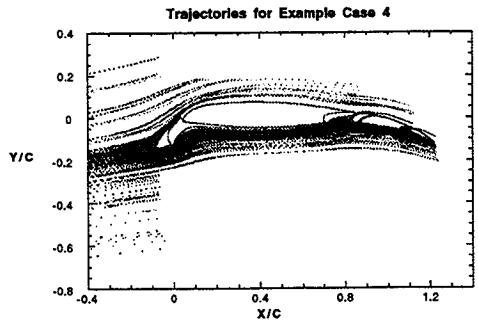
14.4 Example 4: Multi-body Ice Accretion Using Langmuir 'D' Drop Size Distribution

```
BOEING 737-200 5-ELEMENT AIRFOIL
&LEW16
IACCFL =    1
ITIMFL =    0
TSTOP  =  480.
IFLO   =    1
IMLSFL =    0
ISTC   =    1
NSTEPI =    4
DSMN   =    4.D-4,1.D-3,5D-4,7.D-4,4D-
4,5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
DSMAX  =    1.D-3
DDANG  =    7.D-4
IDEN   =    0
IRUNB  =    0
NFLIT  =    1
IGRID  =    0
IRC    =    1
SLWC   =    0.0
NREAD  =    0
IBOD   =    5
LBOTH  =    0
&END
&S24Y1
ILIFT  =    1
IPARA  =    1
IFIRST =    3
ISECND =    3
IPVOR  =    1
INCLT  =    0
CLT    =  0.0
CCL    =  1.0
IFLLL  =    1
&END
&TRAJ1
GEPS   = 0.5E-04
VEPS   = 1.E-03
NEQ    =    4
NPL    =   24
NSEAR  =  199
TSP    = 1.E-03
CHORD  = 0.4572
G      = 0.0
PIT    = 0.0
PITDOT = 0.0
Y0LIM  = 1.E-04
RHOP   = 1.0E3
&END
&DIST
FLWC   = 0.05, 0.1, 0.2, 0.3, 0.2,
0.1, 0.05, 0.0, 0.0, 0.0
DPD    = 4.5, 7.5, 10.2, 14.4, 19.7,
25.1, 32.0, 0.0, 0.0, 0.0
CFP    = 1.038, 1.022, 1.016, 1.0, 1.0,
1.0, 1.0, 0.0, 0.0, 0.0
```

```
&END
&ICE1
VINF    = 44.7
LWC     = 0.92
TINF    = 269.89
PINF    = 94540.0
RH      = 100.0
TIME    = 0.0
&END
&LPRNT
IPRT    = 1
FPRT    = 0
HPRT    = 0
BPRT    = 1
EPRT    = 0
MPRT    = 0
TPRT    = 1
IRPRT   = 0
&END
```

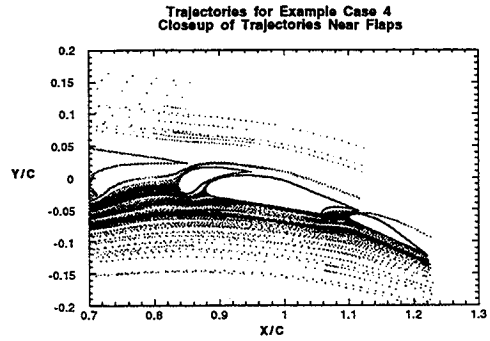
This case is identical to Example 3, so much of the output is the same. The differences are the use of a multi-drop size distribution and the use of a single time step. A single time step was selected due to the difficulty in running this case for more than one time step. The flow is shown in Figure 14.12, the heat transfer coefficients in Figures 14.13 - 14.17 and the collection efficiencies in Figures 14.18 - 14.21. The remaining plots are the trajectory plots and the ice shape plots. The trajectories are shown in Figure 14.29. The ice shapes were shown in Figures 13.13 - 13.17.

Figure 14.29



There are some very apparent differences between this plot and Figure 14.22 for the single drop size run, especially near the flaps. This confirms the collection efficiency plots, which also showed a difference in this region. Figure 14.30 shows the trajectories near the flaps for this example.

Figure 14.30



The trajectories for this case not only pass between the second and third flaps, but between the main element and the first flap. Some trajectories also impinge further back on the second flap, as they try to navigate between the first and second flaps.

14.5 Example 5: Use of Deice/Anti-Ice Capability

This example will cover the uses of the deicer capability in LEWICE 1.6. The codes LEWICE/Thermal and ANTICE will provide a more detailed analysis at the cost of increased computational time. This option was added at the request of several users who desired a computationally cheaper answer to their design questions.

The example case chosen is the same as shown in Example 1, except that in this case, the output from each option of the deicer section is presented. Also, only one time step needs to be run, since the geometry does not need to be changed if there is no ice. Hence this example comprises four separate cases: 1) electrothermal 'running wet'; 2) electrothermal 'evaporative'; 3) hot air 'running wet'; 4) hot air 'evaporative'.

Electrothermal 'Running Wet'

In this case, the interactive input is as follows (answers in bold-face print):

```
 HOW MANY RUNS (CASE STUDIES) DO YOU
WISH TO PERFORM?
     1
```

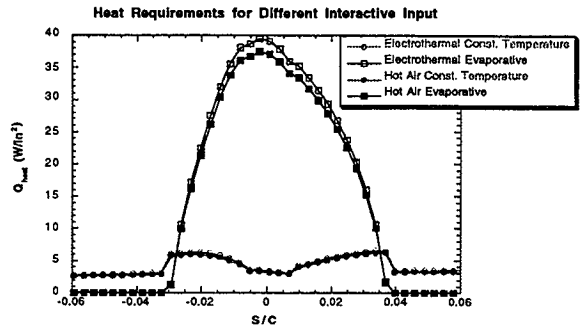
```
 DO YOU WISH TO CALCULATE HEAT REQUIRED
TO ANTI-ICE?
 ANSWER Y OR N
     Y
 ENTER DESIRED SURFACE TEMPERATURE (DEG
K)
    283.15
 ENTER 1 IF YOU WANT TO MODEL AN EVAPO-
RATIVE ANTI-ICER (TEMPERATURE ABOVE
WILL BE USED AS THE INITIAL GUESS)
     0
  ENTER 0 FOR ELECTROTHERMAL HEATER
(INTERNAL HEAT GEN.)
     0
  ENTER 1 FOR HOT AIR ANTI-ICE (MUST
SUPPLY INSIDE HTC VALUE)
     0
 ENTER HEATER LAYER
     9
 CURRENTLY, THE X-Y COORDINATES WHICH
DEFINE EACH BODY ARE NOT TAKEN FROM THE
STANDARD INPUT FILE. THEY ARE INPUT
FROM A SEPARATE FILE FOR EACH BODY.
 ENTER FILE NAME FOR BODY 1
   ms317.dat
```

The deice input file for all cases is

```
NUMBER OF LAYERS = 13
    LENGTH          CONDUCTIVITY
     (M)              (W/M*K)
 4.4450000E-03 1.7653000E+02
 2.5400000E-04 1.7307000E-01
 1.2700000E-03 1.7653000E+02
 2.5400000E-04 1.7307000E-01
 5.0800000E-04 1.5057000E+01
 2.0800000E-04 1.7307000E-01
 3.5100000E-04 3.8075000E-01
 2.0800000E-04 1.7307000E-01
 1.6500000E-04 1.0384000E+02
 2.0800000E-04 1.7307000E-01
 3.5100000E-04 3.8075000E-01
 4.2700000E-04 1.7307000E-01
 7.6200000E-04 1.5057000E+01
NUMBER OF HEAT TRANSFER COEFFICIENTS
INPUT = 0015
      S/C          HTC(W/M*M*K)
-0.5500000E+00 1.0000000E+01
-0.4000000E+00 1.0000000E+01
-0.3000000E+00 1.5000000E+01
-0.2000000E+00 2.0000000E+01
-0.1500000E+00 3.0000000E+01
-0.1000000E+00 5.0000000E+01
-0.0500000E+00 8.0000000E+01
 0.0000000E+00 1.0000000E+02
 0.0500000E+00 8.0000000E+01
 0.1000000E+00 5.0000000E+01
 0.1500000E+00 3.0000000E+01
 0.2000000E+00 2.0000000E+01
 0.3000000E+00 1.5000000E+01
 0.4000000E+00 1.0000000E+01
 0.5500000E+00 1.0000000E+01
```

For the second case, the interactive input answers '1' for evaporative deicer. The other input can be the same. Similarly, the input for hot-air 'running wet' answers '1' for hot air anti-ice. The heat flux results for these cases are shown in Figure 14.31.

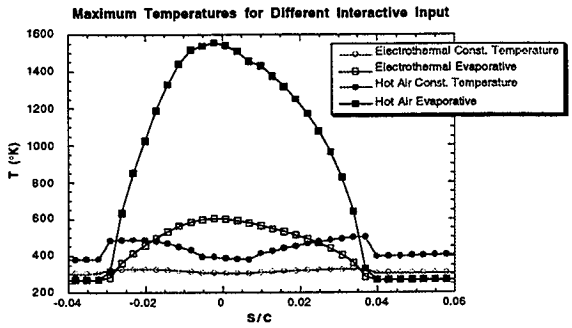
Figure 14.31



The constant temperature cases produce low heat requirements near the stagnation point, where there is more kinetic heating, and greater heat requirements downstream. Note that since a constant surface temperature was selected, heat is input even past the impingement limits, which are clearly discernible on this plot (where heat requirements go to zero for evaporative systems). Evaporative systems require more heat near stagnation where the collection efficiency is higher. It has the advantage of not requiring heat past the impingement limits. The hot air system in this case is slightly more efficient, but that conclusion is highly dependant on the materials used, and the interior heat transfer coefficient input.

The maximum temperature for each system is shown in Figure 14.32. For the electrothermal system, this is the heater temperature and for the hot air system, it is the bleed air temperature.

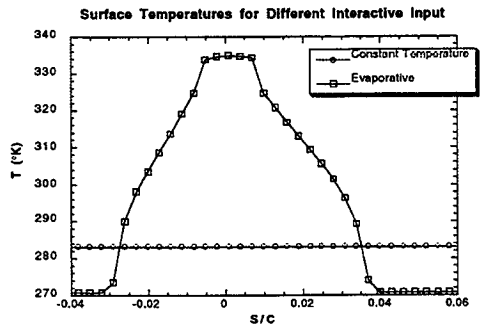
Figure 14.32



This plot shows that the limitations of the simplistic assumptions made in the analysis and the need for the more complex LEWICE/Thermal and ANTICE codes. The maximum temperatures for the hot air system are clearly excessive. However, no attempt was made to optimize the design of the system. It is seen that evaporative systems require higher temperatures near the stagnation point and lower downstream temperatures than running wet systems, which is consistent with the earlier plot.

Finally, Figure 14.33 shows the surface temperatures for these cases.

Figure 14.33



In this plot, there is no difference between a hot air or electrothermal system. Again, evaporative systems require higher temperatures near stagnation and lower values downstream.

14.6 Example 6: Large Drop Example

This case uses the MS-317 single body airfoil presented in Example 1 and 2, but uses a multi-drop size distribution with an MVD of 200 microns, which simulates a condition of concern to industry.

```
MS317-LARGE DROP CASE
&LEW16
IACCFL =    1
ITIMFL =    1
TSTOP  =  900.
IFLO   =   30
IMLSFL =    0
ISTC   =    1
NSTEPI =    4
DSMN   =    3.D-4, 8.D-4, 5D-4, 7.D-4,
3.5D-4, 5.D-4,5.D-4,5.D-4,5.D-4,5.D-4
DSMAX  =    8.D-4
DDANG  =    7.D-4
IDEN   =    0
IRUNB  =    0
NFLIT  =    1
IGRID  =    0
IRC    =    1
SLWC   =    0.0
NREAD  =    0
IBOD   =    1
LBOTH  =    0
&END
&S24Y1
ILIFT  =    1
IPARA  =    1
IFIRST =    3
ISECND =    3
IPVOR  =    1
INCLT  =    0
CLT    =  2.0
CCL    =  1.0
IFLLL  =    1
&END
&TRAJ1
GEPS   = 0.5E-04
VEPS   = 1.E-03
NEQ    =    4
NPL    =   24
NSEAR  =  199
TSP    = 1.E-03
```

```
CHORD  = 0.9144
G      = 0.0
PIT    = 0.0
PITDOT = 0.0
Y0LIM  = 1.E-04
RHOP   = 1.0E3
&END
&DIST
FLWC   = 0.05, 0.1, 0.2, 0.3, 0.2,
0.1, 0.05,0.0, 0.0, 0.0
DPD    = 62., 104., 142., 200., 274.,
348., 444., 0.0, 0.0, 0.0
CFP    = 1.000, 1.000,1.000,1.0, 1.0,
1.0, 1.0, 0.0, 0.0, 0.0
&END
&ICE1
VINF   = 87.4
LWC    = 0.34
TINF   = 263.56
PINF   = 100500.
RH     = 100.0
TIME   = 0.0
&END
&LPRNT
IPRT   = 1
FPRT   = 1
HPRT   = 1
BPRT   = 1
EPRT   = 0
MPRT   = 0
TPRT   = 1
IRPRT  = 0
&END
```

The output for this example is provided in Figures 14.34-39.

Figure 14.34

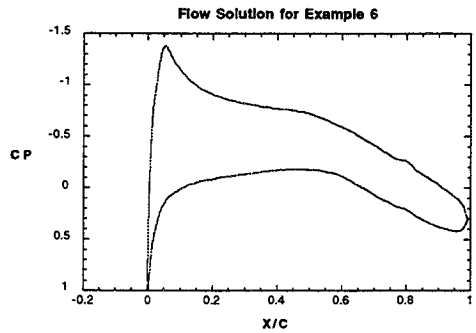


Figure 14.35

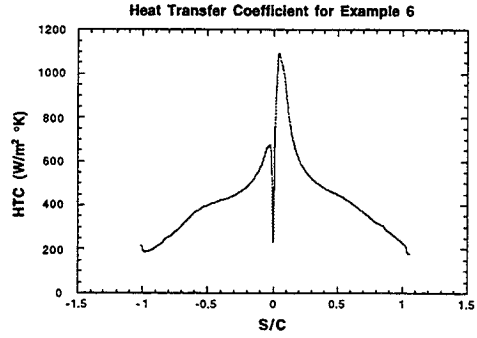


Figure 14.36

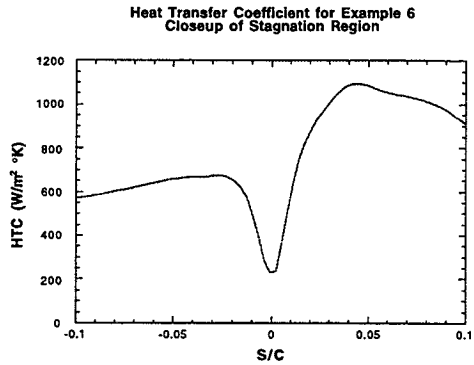


Figure 14.37

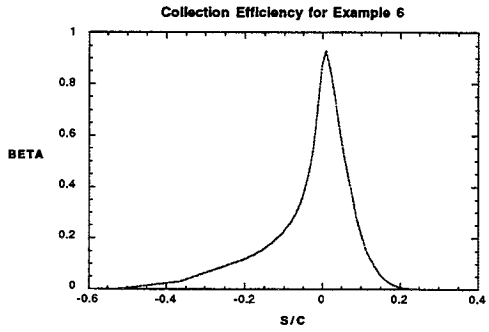


Figure 14.38

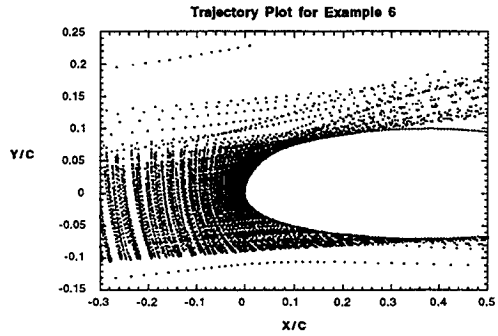
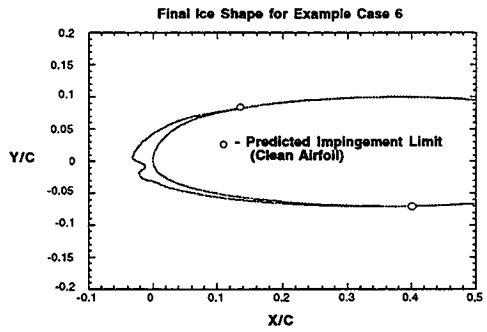


Figure 14.39

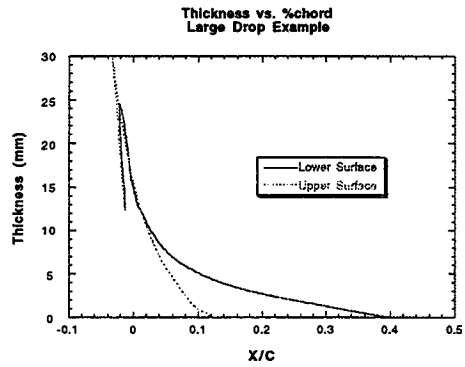


As can be seen by comparing Figure 14.38 with previous examples, larger drops are not deflected as much as smaller drops, which accounts for the higher collection efficiency and the wider impingement limits. This example shows that the upper surface limit is approximately 20% of chord, while the lower limit is 50% of chord. This is a lot of real estate to cover for an ice protection system. However, it should be noted that there is not much water for much of this distance. The collection efficiency drops below 10% at 13% of chord on the upper surface and past 23% chord on the lower surface, meaning that much of the area requires much less power than the leading edge and may potentially be left unprotected, depending on the sensitivity of the airfoil to different thicknesses of ice.

The ice shape plot, Figure 14.39, also shows this effect well. Despite the width of the initial collection efficiency curve, the 'visible' extent of icing is around 8% chord on the upper surface and 20-25% chord on the lower surface, depending on how good your eyesight is. There are several contributors to this 'shrinkage' of the icing extent. First, not all of the ice which impinges will freeze. Even with no heat, a small amount of evaporation takes place. Near the impingement limits, this is a significant percentage of the incoming mass. Second, as the ice grows, it will inhibit impingement further back from the existing ice shape. This is true even for a smooth shape as shown here. In reality, the effect could be even more dramatic. Ice feathers which grow near the impingement limits will 'catch' drops which are near the surface and provide a natural barrier to further impingement.

The ice shape file, unit 20 (ICE1.DAT), also prints out the ice thickness at each location and the wrap distance, S/C. The output is non-dimensionalized to the chord as are the ice shape coordinates. Figure 14.40 shows the ice thickness in millimeters versus the x-coordinate, which represents the extent of ice as a percentage of chord.

Figure 14.40



14.7 Example 7: Parameter Study

This example examines the use of the parameter study option in LEWICE 1.6. For this example, the input data is identical to Example 6, except that a single drop size is used. As the parameter study is performed on drop size, the number placed in the input file is immaterial - it will be overridden by the interactive input. The interactive input is listed below, with answers in bold-face print.

```
 HOW MANY RUNS (CASE STUDIES) DO YOU
WISH TO PERFORM?
          10
 ENTER THE NUMBER FOR YOUR PARAMETER
VARIABLE
 1      TEMPERATURE (K)
 2      LIQUID WATER CONTENT (G/M**3)
 3      VELOCITY (M/S)
 4      ANGLE OF ATTACK (DEG)
 5      DROPLET DIAMETER (MICRONS)
 6      ROUGHNESS (MILLIMETERS)
 7      NUMBER OF FLOW SOLUTIONS
        5
 ENTER VARIABLE NUMBER 1
    100.
 ENTER VARIABLE NUMBER 2
    150.
 ENTER VARIABLE NUMBER 3
    200.
 ENTER VARIABLE NUMBER 4
    250.
 ENTER VARIABLE NUMBER 5
    300.
 ENTER VARIABLE NUMBER 6
    400.
 ENTER VARIABLE NUMBER 7
    500.
 ENTER VARIABLE NUMBER 8
    600.
 ENTER VARIABLE NUMBER 9
    800.
 ENTER VARIABLE NUMBER 10
    1000.
 DO YOU WISH TO CALCULATE HEAT REQUIRED
TO ANTI-ICE?
 ANSWER Y OR N
     N
 CURRENTLY, THE X-Y COORDINATES WHICH
DEFINE EACH BODY ARE NOT TAKEN FROM THE
STANDARD INPUT FILE. THEY ARE INPUT
FROM A SEPARATE FILE FOR EACH BODY.
 ENTER FILE NAME FOR BODY 1
 ms317.dat
```

According to the User's Manual for LEWICE 1.0, the upper limit on drop size is 500 μm for the equations to be valid. A preliminary analysis of droplet physics reveals that LEWICE 1.6 can still be used for these very large drop sizes. Large drops will tend to be non-spherical, have internal convection patterns which cause rotation, and are more likely to splash on the surface. The degree of inaccuracy in the trajectory routines is still probably less than the inaccuracy induced by using a potential flow solver on multi-element airfoils. Future work on this code will look at these effects more thoroughly.

As the input data is identical to Example 6 except for drop size, only the collection efficiencies will be presented. Figures 14.41 and 14.42 show the collection efficiency curves for this example.

Figure 14.41

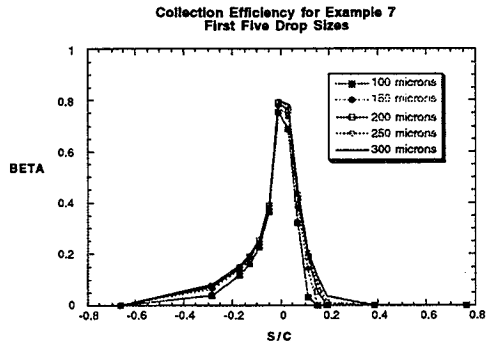
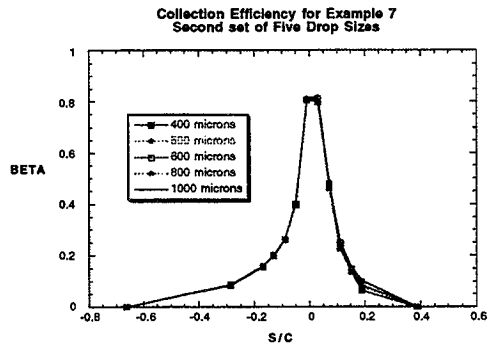


Figure 14.42



The panel resolution was very coarse for these runs to obtain faster turn-around time. The same cases were ran again using a DSMN of $3*10^{-4}$ instead of $7*10^{-4}$. The tolerance for impingement limit trajectories (Y0LIM) was also reduced to $5*10^{-5}$. The collection efficiencies for this case are plotted in Figures 14.43 and Figure 14.44

Figure 14.43

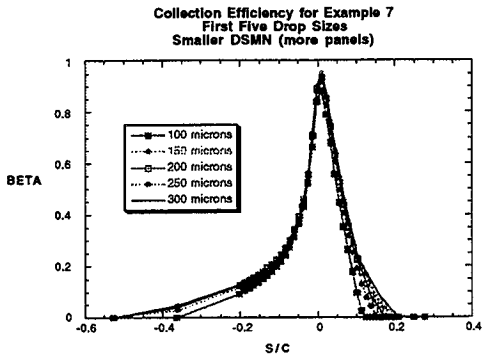


Figure 14.44

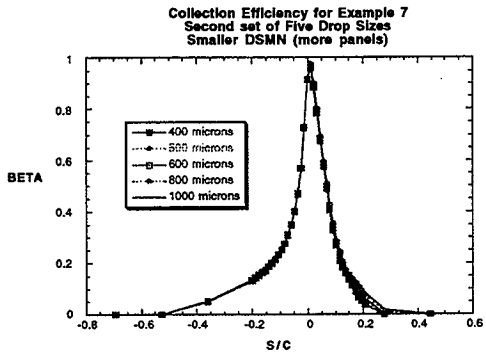


Figure 14.45

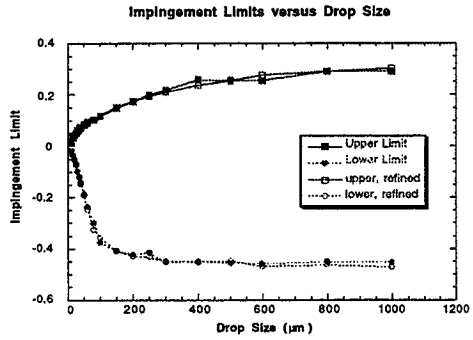
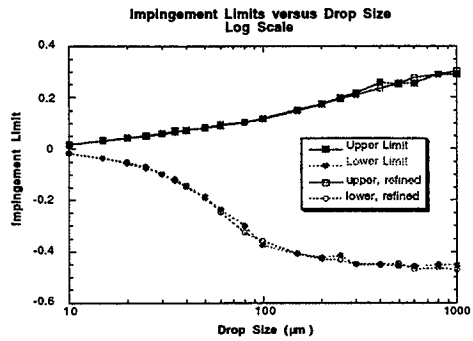


Figure 14.46



Figures 14.45 and14.46 plot upper and lower impingement limit versus drop size, where Figure 14.45 has a linear x-scale and Figure 14.46 has a log x-scale. The results from both panel resolution runs are presented. This plot was obtained by searching unit 56 (MISC.DAT) for the label 'UPPER SURFACE LIMIT' then entering by hand the upper and lower limit values into a spreadsheet. As there were only 20 values (10 upper and 10 lower), this was not too taxing. Also, a similar parameter run was performed for smaller drop sizes to make the plots more complete.

The maximum collection efficiency is plotted versus drop size in Figures 14.47 and 14.48. This plot shows a dramatic difference due to the increase in panelling. The larger DSMN spacing did not produce enough panels, as the results were visibly coarse. The coarsely spaced example was included to show that even experienced users do not always run the code optimally.

Figure 14.47

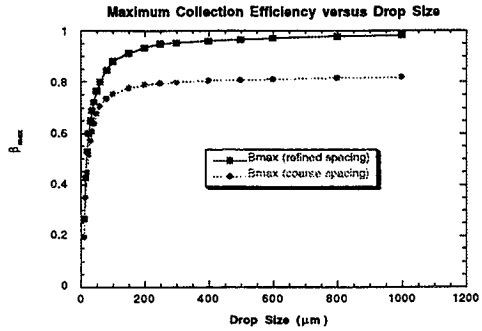
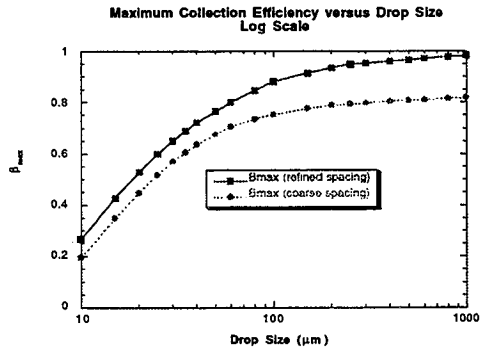


Figure 14.48



15.0 Programmer's Guide to LEWICE 1.6 Subroutines

LEWICE 1.6 (main program) calls:

WINPUT

Performs user interactive input

SETUP

Reads input file and computes the panel model for the clean airfoil. Calls STG1 (initial stagnation pt.), SURFD (surface distance) and NWDIST (point redistribution).

S24Y

Computes potential flow. Calls: ELFORM (forms elements), MAFORM (forms matrix), SOLVIT (solver), COMBO (finds combination constants), FLOWS (computes Cp, VT, etc.), OFFBOD (off-body points).

STAG

Computes the stagnation point and the 'S' distance from it for every panel. The procedure is similar to the previous version. However, when >1 point is found, the VT solution is smoothed ONLY over that region (smoothing the whole array produces undesirable side-effects). Calls COMPS and STG2.

VEDGE

Along with performing the Karman-Tsien compressible corrections, this routine defines an array DVEDS which contains the derivative with respect to S for the boundary layer integral. Calls DANG3.

TRAJ

Computes particle trajectories and computes collection efficiency. Calls: SETLIM (sets limits), RELEAS (finds release point), VELC ('C' grid), VELR (rectangular grid), RANGE (trajectory range), IMPLIM (impingement limits), COLLEC (collection efficiency), and COLLEK (alternate collection efficiency).

ICE

Integrates the boundary layer, computes the new ice shape, and adds points (control volumes) to the surface. Calls: BDY2 (boundary layer), BEAD (roughness prediction), EBAL (energy balance), INTRST (point intersection for ice thickness printout), NWF3 (creates new ice shape), and OUTPUT (outputs icing parameters).

PLTOUT

Writes out solution for iris plot routine. (Commented out for users who use PC's.) Calls TRJOUT (Trajectory output.)

TRJOUT

Outputs trajectories for plotting on iris. This is commented out currently, as individual trajectories are not often plotted and this file is very large.

RADC5

Determines Radius of Curvature for any control volume or panel.

NWDIST

Computes the new distribution of panels or control volumes. Calls CSPLIN (spline-fit), LNTP (linear interpolation), NWSD3 (surface distribution), NWSD9 (surface distribution), NWSD11 (surface distribution), SURFD (surface distance).

PMOD

Creates new panel model which is optimized for the boundary layer/energy balance calculation. Calls STAG, PRBIST (variable redistribution), and PRDIST (variable redistribution).

S24Y calls:

ELFORM

Forms elements. Combination of former routines ASSEMB, ELFORM, PRINTG, PRNTEL, MAIN1. Calls: GEOMCF, BOMB1, BOMB3

MAFORM

Forms Matrices. Calls: VEFORM

SOLVIT

Inverts matrices to find sigma solution. Combination of old routines SOLVE, SOLVIT, QUASI. The flow solver had multiple options on which solver to call. However, they were all remarkably similar in pro-

gramming, solution, and CPU times. Therefore, only SOLVIT is used.

COMBO

Finds combination constants. Calls MIS1. Note: Old routines MIS1 and MIS2 were exactly the same except for one (unnecessary) line. MIS2 was eliminated.

FLOWS

Finds CP and VT from combined solution.

OFFBOD

Finds off-body velocities. Combination of old routines OFFBOD, VPROFF, VXYOFF. Calls VEFORM

Routines called by S24Y routines:

VEFORM

This is a routine which does a large (>100 lines) do loop associated with the flow solution. It was noticed that MAFORM, OFFBOD, and VELCTY all contained this loop. This duplication of lines of code was removed by creating this routine. Calls WEIGHT, ABFORM.

GEOMCF

Geometry combination coefficients

BOMB1

Program stops due to incorrect input.

BOMB3

Program stops due to incorrect input.

MIS1

Flow solver for multiple bodies.

ABFORM

Calculated velocity arrays.

WEIGHT

Calculates weighting functions

TRAJ calls:

SETLIM

Formerly READIN. Previously, variables were passed from the flow and trajectory routines using scratch files. This is currently done using COMMON blocks, hence the name was changed to reflect its current function, which is to set the outer limits for the trajectories.

RELEAS

Checks the release point which is input to see if it is actually in the free stream. Moves the release point out in .5*chord increments (Up to 50) until the input criteria is met. Calls: VEL2, VELCTY

VELC

Creates a simple M by N 'C' grid around the airfoil if grid-based velocities are desired. If using a separate flow code, the user could alter this routine to read in the grid and off-body velocities from that code. The first point created for the 'C' grid is DSHIFT from the airfoil. This is the ONLY remaining use of this variable in the program. Calls: VELCTY

VELR

Creates an angled, rectangular grid if off-body velocities are desired. This is more efficient when using a panel code to create the points, as the trajectories which hit the body will all come from the same region in space. Calls: VELCTY (off-body velocity), INTRST (intersection routine)

RANGE

Determines an upper and lower limit for release points by finding two trajectories: one which passes above the airfoil and one which passes below. Calls: VELCTY, VEL2 (off-body velocity), INTIG (integration).

IMPLIM

Determines the upper and lower impingement limits. Calls: VELCTY, VEL2, INTIG.

COLLEC

Performs NPL evenly spaced trajectories between the impingement limits. Used to find the collection efficiency. Calls: VELCTY, INTIG, TRAJSV (saves trajectories for plotting), VEL2.

COLLEK

Performs collection efficiency calculation using trajectories which hit panel centers (IACCFL=2). Calls: VELCTY, VEL2, INTIG, TRAJSV.

Routines called by TRAJ routines:

VELCTY

Computes off-body velocity by summing the contribution from each panel. Calls: VEFORM, CORR

VEL2

Computes off-body velocity by interpolating from an M by N grid, regardless of the form ('C' or rectangular). Calls: CORR, VELCTY.

INTIG

Integrates momentum equation for each drop using Adams Predictor-Corrector Scheme. Calls: VELCTY, VEL2, DIFSUB, MODE.

DIFSUB

Performs Adams Predictor-Corrector integration. Calls: DIFF2N, YSAVE, YSAV2, YSAV3, VELCTY, VEL2, NDI08Z, NDI09Z

MODE

Determines if a trajectory has impinged on the surface. Calls INTRST

DIFF2N

Contains the differential equations solved by DIFSUB. Calls COEFF to find the drag coefficient (uses spherical drag).

YSAVE

Saves solution during integration. The same DO loops were used in DIFSUB before, but the use of this routine, along with YSAV2 and YSAV3 made DIFSUB more readable and less spaghetti.

YSAV2

See YSAVE

YSAV3

See YSAVE

NDI01Z

Sets up tri-diagonal matrix.

NDI02Z

Solves tri-diagonal matrix.

CORR

Compressibility correction for off-body velocities.

INTRST

Determines if two line segments intersect. Used for trajectory impact.

ICE1 calls:

BDY2

Finds transition movement from MIT criteria. Corrects heat transfer coefficients for multiple stagnation problems, early transition problems. Calls BLINT to integrate the boundary layer. Calls TERP.

EBAL

Control routine for performing mass and energy balance. Outputs mass balance terms, energy balance terms, density, surface temperature and ice height values. Calls: RADC5, DANG3, RH2ICE, RHOICE, TREC, NOICE (deicing), ACCRET (accretion), QVAP, SHED, CPW.

NWF3

Adds ice height to surface in an iterative fashion in an attempt to conserve area. This means that the computed area from EBAL = actual area added. Second half of routine performs 'smoothing' to eliminate jaggedness in ice shape not seen in area or ice height values. Criteria will automatically conserve the iced area. Calls: SNORMC2, LNTP, SURFD, ADJPT2.

OUTPUT

Prints out general information about the ice accretion. Previously, LEWICE would output everything from this routine. Array variables are now output from the routines generating them (FLOWS, VEDGE, TRAJ, BDY2, EBAL, ACCRET are the main ones.)

Routines called by ICE1 routines:

BLINT

Integrates the boundary layer and computes heat transfer coefficient. Uses von Kármán-Pohlhausen technique with special formulas at and near stagnation to reduce numerical error at these points. Formulas found using L'Hopital Rule. NOTE: transition points and stagnation point are not control volume corners as assumed before, but are interpolated. Stagnation point is interpolated to find exact x,y point where VT=0 and transition point is where roughness Reynold's number Re_k=600. Coef-

ficients for roughness Staunton number are taken from Dipprey and Sabersky paper on 'sand-grain' roughness not on original Kays and Crawford values which were for spheres. Calls ALTINT.

ALTINT

Performs integration outlines in a previous section.

TERP

Interpolation scheme using a least-squares fit polynomial. Must be an even-order polynomial. Currently 4th order. Calls: CHOLES.

CHOLES

Solves matrix set up by TERP

ACCRET

Calculates ice height from mass and energy balance. Finds evaporation amount, runback amount, ice density and ice height. Calls: QVAP, SOLVEW and RHOICE.

QVAP

Determines heat loss by evaporation. Calls PVAP.

SOLVEW

Solves the non-linear energy equation for surface temperature via Newton-Raphson iteration. Solves for temperature even when freezing fraction is between 0 and 1 by assuming a 'high heat capacity' of (Latent Heat/ ΔT melt) ΔT melt = 10^{-5} °F. Corrections for a wrong 'assumed state' is left-over from deicer equations where it is possible to incorrectly assume which phase (solid, liquid, in between) the water is in. For an unheated surface, this section will probably not be accessed (IF statements will never be true). The segment is left in 'just in case'. Calls: QVAP, RHOICE, TREC, CPW, PVAP.

SHED

Removes ice by aerodynamic forces exceeding adhesion force. **Warning:** experimental data for ice adhesion has too large of an experimental error for this correlation to work correctly. Calls: COEFF

DANG3

Finds the panel angle.

SNRMC2

Finds panel normals. Calls: NWDIST

ADJPT2

Determines if control volumes grow together and redistributes mass accordingly. Replaces SEGSEC.

LNTP

Linear interpolation

NOICE

Finds heat requirements and maximum temperatures for anti-icing systems. Called by EBAL. Calls CPW, TREC, PVAP.

PVAP

Calculates vapor pressure of ice or water.

TREC

Calculates recovery temperature.

CPW

Calculates specific heat of ice or water.

RHOICE

Calculates ice density from rotating cylinder correlation.

RH2ICE

Calculates ice density from correlation based on freezing fraction.

16.0 Conclusions

Three major improvements have been made to the LEWICE 1.6 code from LEWICE version 1.3 (LEWICE-Beta).LEWICE 1.6 has been modified to handle a large number of time steps for single element geometries and to perform the ice addition and recalculation of the flow accurately. A roughness model has been proposed which predicts measured roughness values much more accurately than previous models. Finally, multi-element trajectories and single time-step ice accretions on multi-element geometries are now possible.

Due to the ability of the code to handle a great number of time steps very accurately, little future work will be performed in this area due to the success of this routine. Work on the multi-element features is needed so that multiple time steps can be performed on these configurations as well.

The roughness prediction model shows much promise, but additional work is needed on a more robust model. Research is also being needed on the affect of this roughness on the heat transfer.

The code has shown the ability to predict collection efficiencies for a multiple drop size distribution on multi-element geometries for multiple time-step ice accretions. The solution is quite robust and should be able to handle most of the geometries in use by industry.

This new version of LEWICE is called LEWICE 1.6 and is available for use by industry. Development of this code is continuing in the areas of droplet trajectory physics, roughness modeling, roughness effects on heat transfer, ice density, surface tension effects and sublimation effects. Periodic updates to this document should be expected.

17.0 References

1) Ruff, G.A. and B.M. Berkowitz, "Users Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)," NASA CR185129, May 1990.

2) Wright, W.B., "Update to the NASA Lewis Ice Accretion Code LEWICE," NASA CR195387, Oct. 1994.

3) Hess, J. L. and A. M. O., "Calculation of Potential Flow About Arbitrary Bodies," Progress in Aeronautical Sciences, 8:1-138, (D. Kuchemann, editor), Elmsford, New York, Pergmon Press, 1967.

4) Frost, W., Chang, H., Shieh, C. and K. Kimble, "Two Dimensional Particle Trajectory Computer Program," Interim Report for Contract NAS3-224488, 1982.

5) Messinger, B. L., "Equilibrium Temperature of an Unheated Icing Surface as a Function of Airspeed," J. of the Aeronautical Sciences, vol. 20, no. 1, Jan. 1953, pp. 29-42.

6) Wright, W. B., "Advancements in the LEWICE Ice Accretion Model," NASA CR191019 (AIAA-93-0171), Jan. 1993.

7) Berkowitz, B.M., Potapczuk, M.G., Namdar, B.S. and Langhals, T.J., "Experimental Ice Shape and Performance Characteristics for a Multi-Element Airfoil in the NASA Lewis Icing Research Tunnel," NASA TM105380, Dec. 1991.

8) Shin, J., "Characteristics of Surface Roughness Associated With Leading Edge Ice Accretion," NASA TM106459 (AIAA-94-0799), Jan. 1994.

9) Olsen, W. and E. Walker, "Experimental Evidence for Modifying the Current Physical Model for Ice Accretion on Aircraft Surfaces," NASA TM 87184, 1986.

10) Schlichting, H., Boundary-Layer Theory. F. J. Cerra, ED. New York, New York: McGraw-Hill, 1979.

11) Potapczuk, M. G., Bragg, M. B., Kwon. O.J. and L. N. Sankar, "Simulation of Iced Wing Aerodynamics," NASA TM 104362, May 1991.

12)Von Doenhoff, A. E. and E. A. Horton, "Low-Speed Experimental Investigation of the Effect of Sandpaper Type Roughness on Boundary-Layer Transition," NACA TN 3858, 1956.

13) Dipprey, D. F., and R. H. Sabersky, "Heat and Momentum transfer in smooth and rough tubes at various Prandtl numbers," Int. J. Heat and Mass Transfer vol. 6, pp. 329-353, 1963.

14) Kays, W. M. and M. E. Crawford, Convective Heat and Mass Transfer, 2nd Edition, MacGraw-Hill Book Company, New York, 1980.

15) Press, W. H., Teukolsky, S. A., Vettering, W. T., and B. P. Flannery, "Numerical Recipes in Fortran - The Art of Scientific Computing," Second Edition, Cambridge University Press, New York, 1992.

16) Clift, R. Bubbles, Drops and Particles, Academic Press, New York, 1978.

17) Hobbs, P.V. Ice Physics, Oxford University Press, Ely House, London, 1974.

18) White, F. M., Viscous Fluid Flow, McGraw-Hill, Inc., 1974.

19) Carlson, D.J. and Haglund, R.F. "Particle Drag and Heat Transfer in Rocket Nozzles." AIAA Journal, Vol. 2,No. 11,1964,pp. 1980-1984.

20) Gear, C.W. "The Automatic Integration of Ordinary Differential Equations." Comm. ACM 14, 1971, pp. 176-179.

21) Gear, C.W. "DIFSUB for Solutions of Ordinary Differential Equations." Comm. ACM 14, 1971, pp. 185-190.

22) Bilanin, A. J. "Proposed Modifications to Ice Accretion/Icing Scaling Theory," AIAA88-0203, Jan. 1988.

23) Chang, H., Frost, W., and Shaw, R.J. "Influence of Multidroplet Size Distribution on Icing Collection Efficiency." AIAA-83-0110, 1983.

24) Olsen, W., Shaw, R., and Newton, J. "Ice Shapes and the Resulting Drag Increase for an NACA 0012 Airfoil." NASA Technical Memorandum 83556, 1984.

25) Tribus, M.V., et al. "Analysis of Heat Transfer Over a Small Cylinder in Icing Conditions on Mount Washington." American Society of Mechanical Engineers Transactions, Vol. 70, 1949, pp. 871-876.

26) Olsen, W. "Close-up Movies of the Icing Process on the Leading Edge of an Airfoil." NASA Lewis Research Center Movie C-313, 1985.

27) Hansman, R. J., A. Reehorst, and J. Sims, "Analysis of Surface Roughness Generation in Aircraft Ice Accretion," AIAA-92-0298, Jan. 1992.

28)Bird, R.B., Stewart, W. E., and E. N. Lightfoot, Transport Phenomena, John Wiley & Sons, Inc., New York, 1960.

29)Cansdale, J. T. and R. W. Gent, "Ice Accretion on Aerofoils in Two-Dimensional Compressible Flow - A Theoretical Model," Royal Aircraft Establishment Technical Report 82128, 1983.

30) Al-Khalil, K.M., "Numerical Simulation of an Aircraft Anti-Icing System Incorporating a Rivulet Model for the Runback Water," PhD Dissertation, University of Toledo, Toledo, Ohio, Jan. 1991.

31)Macklin, W. C., and G. S. Payne, "A Theoretical Study of the Ice Accretion Process," Quarterly Journal of the Royal Meteorological Society, 1967, Vol. 93, pp. 195-213.

32) Wright, W. B., "Simulation of Two-Dimensional Icing, De-Icing and Anti-Icing Phenomena," PhD Dissertation, University of Toledo, Toledo, Ohio, Dec. 1991.

33) Rios, M., "Icing Simulations Using Jones' Density Formula for Accreted Ice," AIAA-91-0556.

34) Jones, K. F., "The Density of Natural Ice Accretions," Fourth International Conference on Atmospheric Icing of Structures, 1988, pp. 114-117.

35) Anderson, D. A., Tannehill, J. C., and R. H. Pletcher," Computational Fluid Mechanics and Heat Transfer," McGraw-Hill Book Company, New York, 1984.

36) Yamaguchi, K., "Improved Ice Accretion Prediction Techniques Based on Experimental Observations of Surface Roughness Effects on Heat Transfer," MS Thesis, Massachusetts Institute of Technology, Cambridge Massachusetts, May 1990.

37) Poinsatte, P. E., "Heat Transfer Measurements From a NACA 0012 Airfoil in Flight and in the NASA Lewis Icing Research Tunnel," NASA CR 4278, March 1990.

38)Scavuzzo, R. J., M. L. Chu and C. J. Kellackey, "Impact Ice Stresses in Rotating Airfoils," AIAA 90-0198, Jan. 1990.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1995	Final Contractor Report

4. TITLE AND SUBTITLE

Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6

5. FUNDING NUMBERS

WU–505–68–10
C–NAS3–27186

6. AUTHOR(S)

William B. Wright

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NYMA, Inc.
2001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER

E–9729

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR–198355

11. SUPPLEMENTARY NOTES

Project Manager, Mark Potapczuk, Propulsion Systems Division, NASA Lewis Research Center, organization code 2720, (216) 433–3919.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories 34 and 03

This publication is available from the NASA Center for Aerospace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This report is intended as an update/replacement to NASA CR 185129 "User's Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)" (1) and as an update to NASA CR 195387 "Update to the NASA Lewis Ice Accretion Code LEWICE" (2). In addition to describing the changes specifically made for this version, information from previous manuals will be duplicated so that the user will not need three manuals to use this code.

14. SUBJECT TERMS

Icing; Aircraft; Heat transfer

15. NUMBER OF PAGES

98

16. PRICE CODE

A05

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	