# Update to the NASA Lewis Ice Accretion Code LEWICE

William B. Wright
NYMA, Inc.

Prepared for
Lewis Research Center

# 1.0 ABSTRACT

This report is intended as an update to NASA CR 185129 "User's Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)" (1). It describes modifications and improvements made to this code as well as changes to the input and output files, interactive input, and graphics output. The comparison of this code to experimental data is shown to have improved as a result of these modifications.

## 2.0 SUMMARY

LEWICE 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 using the Douglas Hess-Smith 2-D panel code (S24Y) (2). This potential flow field is then used to calculate the trajectories of particles and the impingement points on the body (3). 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 (4), 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.

LEWICE has been used to calculate a variety of ice shapes, but should still be considered a research code. The code should be excersised 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 accrtetion model is one important application of LEWICE.

## 3.0 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 techniqure, 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, a computer code named LEWICE and documented in an earlier report.. This code compared reasonably well with the available experimental ice shapes

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 usefullness. These improvements were developed at Sverdrup Technology, Inc. and at the NASA Lewis Research Center under NASA funding.

## 4.0 Nomenclature

$t$ = time (sec)

$c$=chord (m)

$\rho$=density (kg/m$^3$)

$V$=velocity (m/sec)

$\alpha$=angle of attack (degrees)

$\beta$=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$^3$)

$\mathcal{L}$ = Lewis number (dimensionless)

MW = molecolar 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$^2$)

$\gamma$ = ratio of heat capacities $c_p/c_v$ (dimensionless)

$\mathcal{D}_{AB}$ = diffusivity (m$^2$/s)

$k$ = thermal conductivity (W/mK)

$T_{rec}$ = recovery temperature (K)

$T$ = temperature (K)

$\Delta T_m$ = 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

$\tau_f$ = shear stress (N/m$^2$)

$\theta$ = contact angle (radians)

$b$ = bead height (m)

$\mu_f$ = viscosity of fluid (kg/ms)

$W_e$ = Weber number (dimensionless)

$\sigma$ = 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)

$\delta$ = boundary layer thickness (m)

Nu = Nusselt number (dimensionless)

$c'_f$ = friction coefficient (dimensionless)

$k_s$ = equivalent sand-grain roughness (m)

$\nu$ = 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

$\infty$ = ambient condition

f = freezing

rb = runback

in = incoming term

out = outgoing term

shed = mass shed

remain = mass remaining

freeze = freeze

## 5.0 INTRODUCTION

The computer code, LEWICE, 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 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 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 not, for the most part, duplicate information contained in the original LEWICE Users Manual unless it is deemed necessary to explain the changes made. Instead, this report will document the modifications made, incuding 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.Examples of the improved ice accretion prediction capabilities are also included.

## 6.0 INTERACTIVE I/O

This is intended as a brief overview of the user input and screen output of this program.

### SIMULATION TIME

This is the total time of the run.

### NUMBER OF CASE STUDIES

The program is set up to perform a sequence of runs. This is useful as engineers usually have other tasks to perform other than watching a program run. For example, if the user wants to see the change in ice shape with temperature, the user would select the number of cases to be run. The program will then prompt the user for the different temperatures. The program will then run each case one after the other until it is finished. Output will be sequential. For example, the ice shape file will contain: ice shape @ t=0, case1; @t=t1, case1;@t=t2, case1;.... @t=t1, case2 (t=0, the clean airfoil, need not be repeated);@t=t2, case2;... for all cases. No user input is necessary after the initial input. Terminal output is echoed to unit70, so that if errors occured, they will not be missed by the user.

### NUMBER OF FLOW SOLUTIONS

This is the number of time steps, for each time step performs a flow and trajectory calculation. The time step used by the program is equal to TOTAL TIME / # of TIME STEPS.

### NUMBER OF BODIES

This is the number of separate bodies being simulated for multi-element icing simulation. The program will not handle separate bodies which become attatched due to icing.

### GRID-BASED VELOCITIES

During the integration of the droplet trajectory, the program needs an off-body air velocity at that point. The default setting is to calculate the velocity every time from the sources and sinks of the potential solution (previous method). A possible faster approach is to calculate the off-body velocity on a grid and to interpolate the desired velocity from that grid. For short icing times or small # of flow solutions, there have not been any problems using a rectangular grid to interpolate air velocities. A 'C' grid gives too many points where they aren't needed. As the solution progresses, this method gets less accurate. The default setting is highly recommended at this point. Note that this recommendation for POTENTIAL FLOW. NASA Lewis has ice accretion codes which perform Naviér-Stokes and Euler calculations (5). As these calculations are more accurate and are performed on a grid in the first place, grid interpolation of velocity is much more accurate.

### RAMP-UP TIME

In the Icing Research Tunnel, there is a short initial period (about 20 sec.) before the desired LWC is reached (6). This option will account for that discrepancy. The program doesn't seem too sensitive to this. The net result is to decrease slightly the first time step's ice accretion.

### TURNING ANGLE

This is the angle between two flat panels. A zero turning angle represents no curvature (flat plate). The smaller this value, the more panels that will be generated during a run and the more detailed the resulting ice shape. The purpose is to obtain an adequate flow solution the next time step. A value of 10° has been found to be a reasonable value for a NACA 0012 airfoil during icing.

### ANTI-ICE HEAT REQUIREMENT

If 'Y' is selected, the program will compute an estimate of the heat required to keep the surface at a specified surface temperature. This estimate assumes a uniform temperature chordwise and computes the heat required from analytic solution of the resulting 1D steady state heat equation (7). A second input file (NOICE.INP) is required for this. It contains the number of layers, thicknesses, thermal conductivities, and inside heat transfer coefficients. The output contains heat flux and maximum temperature at each control volume. This program will calculate the heat requirements and then compute the ice shape as if the surface were unheated.NASA Lewis also has codes which perform more detailed analysis of deicer and anti-icer performance (8), (9).

### DESIRED SURFACE TEMPERATURE

This variable is input only if ani-ice mode is on. It is the temperature the user wants at the surface. This temperature must be above freezing (in Kelvin) for this option to work.

### HOT AIR / ELECTROTHERMAL CHOICE

If you choose the anti-ice option, the program will prompt you for a choice of a hot air system or an electrothermal system.

### HEATER LAYER

If an electrothermal deicer is selected, the program will prompt the user for which layer the heater resides in (counting from the inner surface).

### READING IN COORDINATES

This option allows the user to input the x,y coordinates of the body in the LEWICE input file (default), or from a separate file consisting of two columns of data. The left column for x-coordinates and the right column for y-coordinates. This is often a more convenient form of input.as this is the format of the ice shape output file.

## 6.1 VARIABLES IN INPUT FILE

These variables are given to the program from the input file. They control the numerics of the program and if reasonable values are not input can result in variations in the resulting ice shape.

### SEGTOL

This is the ratio of the size of a given panel to its neighbors. This definition differs from the previous definition of this variable. The program will adjust corner points to ensure that panel size ratios are within the given limit. Must be greater than 1. Be careful not to use a value too small or too large. A range between 1.2-2.0 is recommended. A value too large will not redistribute enough points, and you will start seeing effects such as blockish ice shapes and poor flow solutions. A lower value will create a more uniform set of panels, but may result in the generation of too many panels. A value of 1.5 has worked well for most simulations.

### XKINIT

This is the equivalent sand-grain roughness used in the heat transfer coefficient correlations (10). NOTE: Input is now in MILLIMETERS. For now, use the correlation provided in the LEWICE Manual for determining a representative value. The beta program is less sensitive to this parameter than LEWICE, so the user has a wider margin of error in this variable. Experiments are under way to determine a new correlation which takes into account variability of ice roughness with chord, leading-edge radius and meteorological conditions.

### NPL

This is the number of trajectories used to define a collection efficiency curve. Too few will result in poor collection efficiency prediction, while too many will slow the simulation, as this section remains the slowest part of the program. This value will be dependant on the geometry in question. A value of 24 trajectories for a 135 panel NACA 0012 has been used in testing this program. That value is quite a bit on the high side, but so far no analysis has been done to better define an acceptable lower limit. NOTE: The program will increase this value in proportion to the increase in panel number as the run progresses. If the program is generating a large number of new panels for a difficult case, the solution will slow down quite a bit, as both the flow and trajectory modules will be doing more work.

### TERMINAL OUTPUT

This section describes the output to the terminal during a run. The program will identify the routine it is running at the start of execution. Flow field, stagnation point, trajectory code, and ice accretion routines are identified. Multiple stagnation points (if any) are identified. The program automatically selects the value closest to the previous time step's value and continues. During geometry modification, the program outputs to UNIT 70 pertinent information on the panel addition. NADDED is the cumulative number of panels added from the initial number. NITER is the number of points changed that trip due to the panel size ratio being too large. NTRIP is the number of trips (iterations) needed to 'converge' on the panel model for the next time step. Normally, when the program exceeds the 1000 panel limit, this will be reflected in this output, especially the NADDED value. For debugging purposes, it would be quite useful if NASA Lewis could have access to the input files for these cases, as well as the terminal input selected.

## 7.0 OUTPUT FILES

UNIT 18

Anti-ice output file. Variables are: 'S' distance (m), heat required (W/in$^2$), maximum temperature ('K).

UNIT 19

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

UNIT 20

x/c, y/c

Non-dimensional geometry coordinates for airfoil and ice shape

UNIT 25

I, S, VE, TE, PE, RA

Flow parameters at edge of boundary layer, corrected for compressibility. Variables are: panel #, 'S' distance (m), velocity (m/s), temperature (K), pressure (Pa), air density (kg/m$^3$)

UNIT 26

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

UNIT 29

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

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

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

UNIT 31

I, S, Y0

Impingement data for each drop size. Variables are: Hit #, 'S' location of hit, deflection of particle.

UNIT 32

I, S, BETA

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

UNIT 33

S, 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

S, DICE, VRB

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

UNIT 40

S, T, H, TREC

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

UNIT 41

S, RI

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

**UNIT 43**

S, QCX, QEX, QSX, QLX, QCOND,QTX

Energy balance terms. Variables are: 'S' 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 net diffence of all terms (should be near 0).

**UNITS 44-54**

Geometry files for alternate geometry input. A different file is used for each body.

**UNIT 86**

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.

**UNIT 87**

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 withing the impingement limits. File is used to create plots on an iris workstation.

## 8.0 GRAPHICS CAPABILITIES

The code is no longer equipped with an interactive graphic capability as these routines tend to be hardware specific. Instead, output is printed to several files as described in the previous section. Graphs are then generated from these output files. For PC systems, the column formatted files are easily read into common spreadsheet programs. For workstations, a set of post-run graphics programs is included on the tape when the program is sent. This menu-driven package generates plots from the output units 86 and 87. The user simply types 'plot' at the end of the run to start the package.

## 9.0 Modifications to the LEWICE Program

This section is intended to give a description of modifications which were made to the LEWICE program and are part of the LEWICE-Beta program. 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 sollution 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.

**Perform Multi-body Ice Accretion**

The potential flow program in LEWICE 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 contained internal plot calls to a graphics routine specific to NASA Lewis. These routines were removed. LEWICE-Beta 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 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 seperate file (for multi-body input each body is input from a seperate file). This second option is often more convenient to use.

**Criteria Check for Input Geometry**

LEWICE-Beta has two user-specified criteria for all geometries, including the input geometry. These criteria are explained in more detail in another section. The first criteria, SEGTOL, is input near the bottom of the main input file. The criteria states that the ratio in size of one panel to either of its neighbors is within SEGTOL ( $1/\text{SEGTOL} \le \Delta s_{i+1} / \Delta s_i \le \text{SEGTOL}$). The second criteria states that the acute angle between two panels (called the turning angle) is less than the value supplied by the terminal input. For the first criteria, values in the range $1.2 \le \text{SEGTOL} \le 2$ are normally used, while for the second criteria the range is normally $5 \le \Delta\theta \le 15$. The purpose of these two criteria is to ensure a reasonably accurate flow solution and an accurate geometry modification for the run. The high end of the ranges above will produce fewer additional panels, but may lead to a degredation of the quality of the solution. The low end of the ranges should aid the quality of the solution, but may lead to an excessive addition of panels.

**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 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.

<u>ISOL</u>

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.

<u>IPRINT</u>

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

<u>ITYPE</u>

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

<u>LCMB</u>

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

<u>LCMP</u>

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

<u>LEOM</u>

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 unsymmetric, 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 $\Delta V_e$ (option 1).

LYOR

Rather than have the user unintentionally supply erroneous information, the program uses the Y0MAX and Y0MIN 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 = $\alpha$.

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.

Y0MAX

Initial guess calculated by the code.

Y0MIN

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 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-Beta 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 take the absolute value for pressure and continue with the simulation. A warning message is delivered to the terminal notifying the user of this occurrence.

Move RA 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 seperate files to facilitate later plotting.

**Change Panel Addition Routine**

After a new geometry has been found, it is necessary to add panels (points) so that future flow solutions and geometry additions are more accurate. These criteria are explained in more detail in another section. The first criteria, SEGTOL, is input near the bottom of the main input file. The criteria states that the ratio in size of one panel to either of its neighbors is within SEGTOL ( $1/\text{SEGTOL} \leq \Delta s_{i+1} / \Delta s_i \leq \text{SEGTOL}$). The second criteria states that the acute angle between two panels (called the turning angle) is less than the value supplied by the terminal input. For the first criteria, values in the range $1.2 \leq \text{SEGTOL} \leq 2$ are normally used, while for the second criteria the range is normally $5 \leq \Delta\theta \leq 15$. The purpose of these two criteria is to ensure a reasonably accurate flow solution and an accurate geometry modification for the run. The high end of the ranges above will produce fewer additional panels, but may lead to a degredation of the quality of the solution. The low end of the ranges should aid the quality of the solution, but may lead to an excessive addition of panels.

**Add Routine VELFORM**

A 100+ line routine associated with the flow field computation was performed by MAFORM, VELCTY, and OFFBOD. The creation of this subroutine eliminated 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-Beta 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 contibution 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 gridpoints; 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 tempera-

ture 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 (13). A slight change in the incompressible drag relation was also made. The formula currently used is taken from White (14).

**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 polynomial 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-Beta 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.

**Panel Removal**

Previously, panels could only be removed if non-adjacent panels intersected. In addition, panels are currently removed when the distance between non-adjacent panels is less than $10^{-7}$ (nearly intersecting panels) and when the non-dimensional panel size is less than $10^{-7}$. In the second case, the region where all panels are $\leq 10^{-7}$ is artificially smoothed, then spline-fit and repaneled with panels of size $10^{-7}$*SEGTOL. A glitch in the geometry addition routine exists when panels are growing together, causing large turning angles which then causes additional panels to be created which causes the panel size in this region to decrease. A resolution of this problem is being sought.

**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 (15), (16) have indicated that the transition point moves toward stagnation during the course of an icing run. Although currently com-

mented out, the mechanism for forcing this transition movement has been coded in (17).

**Allow Different Roughness Regions**

An option which is currently commented out in WINPUT and BDYLYR would allow the user to input seperate 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 (18).

**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 susceptable 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 (19) 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 (20).

**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 (21) on actual sand-grain roughness gave a different correlation. As LEWICE-Beta 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. 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 seperate 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 (not currently used) and performing a particle trajectory of the shed particle (not currently used).

**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' (21). 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, it was discovered that the heat loss into the airfoil approximated an analytic solution for 1D transient heat transfer (22). This analytical solution was then added to LEWICE-Beta as a means of calculating the heat loss by conduction.

**Evaporative Term**

In the LEWICE Manual, there is a derivation of the evaporative heat loss term. It contains an assumption that the ratio of the evaporative pressure divided by the static pressure was identical at the edge of the boundary layer and in the free-stream. Although close for most conditions, some sample calculations showed this assumption to be in error. As this term can easily be evaluated at the edge of the boundary layer using the potential flow solution, this assumption was removed.

**Water Shedding**

Close-up movies of the icing process by Olsen 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-Beta, 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. 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 * $\Delta s$) 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.

## 10.0 Programmer's Guide to LEWICE Subroutines

**LEWICE (main program) calls:**

WINPUT

Performs user interactive input

SETUP

Reads input file and checks the clean airfoil for turning angle and segment ratio (SEGTOL) criteria. Calls NWPTS2 and NWPTS4 to do this second part.

S24Y1

Computes potential flow. Calls: ELFORM, MAFORM, SOLVIT, COMBO, FLOWS, OFFBOD.

STAG

Determines Stagnation Point. Calls: COMPS and AMEAN

VEDGE

Performs compressible correction to flow. Also determines derivative which goes into the boundary layer integral. Calls: SPLIN2 and AMEAN.

TRAJ

Computes particle trajectories and computes collection efficiency. Calls: SETLIM, RELEAS, VELC, VELR, RANGE, IMPLIM, COLLEC, TRAJOUT, ORDER, EFFICY

ICE1

Integrates the boundary layer, computes the new ice shape, and adds points (panels) to the surface. Calls: BDYLYR, EBAL, PLOTOUT, NWFOIL, OUTPUT, SEGSEC, NWPTS4, NWPTS2, RMOVE.

RESTRT

Writes out restart file which can be used as an input file to LEWICE.

PLOTOUT

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

**S24Y1 calls:**

ELFORM

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

MAFORM

Forms Matricies. Calls: VELFORM

SOLVIT

Inverts matricies 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 programming, 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 VELFORM

VELFORM

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.

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). This is done 3 times. If there is still >1 point, the routine 'gives up' and automatically picks the point closest to the last time step point. NOTE: When using the restart file, the program will not know the previous stagnation location. The user should put some effort into making sure a multiple stagnation point problem will not occur in the first time step when using the restart file.

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. This is smoothed using AMEAN.

**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 trajecories.

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: 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, 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, VELCT2, VEL2, INTIG.

IMPLIM

Determines the upper and lower impingement limits. Calls: VELCTY, VELCT2, VEL2, INTIG.

COLLEC

Performs NPL evenly spaced trajectories between the impingement limits. Used to find the collection efficiency. Calls: VELCTY, VELCT2, VEL2, INTIG.

TRAJOUT

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

ORDER

Puts all trajectory hit locations in order for interpolation routines.

EFFICY

Computes and outputs collection efficiencies. Calls: TERP and LINER.

**Routines called by TRAJ routines:**

VELCTY

Computes off-body velocity by summing the contribution from each panel. Calls: VELFORM

VELCT2

Uses VELCTY until the drop is within DSHIFT and then uses last known velocity from then on. Not currently used. The circumvention of DSHIFT is currently handled in function routine ABFORM.

VEL2

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

INTIG

Integrates momentum equation for each drop using Adams Predictor-Corrector Scheme. Calls: VELCTY, VELCT2, VEL2, DIFSUB, MODE.

DIFSUB

Performs Adams Predictor-Corrector integration. Calls: DIFFUN, YSAVE, YSAV2, YSAV3, VELCTY, VELCT2, VEL2, NDI01Z, NDI02Z

MODE

Determines if a trajectory has impinged on the surface. Calls INTRST

DIFFUN

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 spagetti.

YSAV2

See YSAVE

YSAV3

See YSAVE

NDI01Z

Sets up tri-diagonal matrix.

NDI02Z

Solves tri-diagonal matrix.

TERP

Finds collection efficiency by fitting the trajectory hit points (Y0 vs S) with a least-squares fit polynomial. Must be an even-order polynomial. Currently 4th order. Calls: CHOLES. TERP also finds the trajectory angle for NWFOIL. Used only if ice is added normal to the trajectories, not normal to the surface. LEWICE currently adds ice normal to the surface.

CHOLES

Solves matrix set up by TERP

LINER

Finds collection efficiency by finding the central-differenced derivative at each hit point, then linearly interpolating to find collection efficiency. LINER is the current choice in the program.

SPLIN2

Finds collection efficieny by performing a spline-fit of the Y0 vs. S curve and evaluating the derivative. Used only when TERP fails to find an answer.

**ICE1 calls:**

BDYLYR

Finds effective LWC values if a ramp-up time is used instead of a constant LWC. Finds transition movement from MIT criteria (currently not used). Finds 'bead height' from surface tension and gravitational forces. Currently used only for water runback. This value could be used as the local 'sand-grain' roughness instead of the input value. This option is currently commented out. Calls BLINT to integrate the boundary layer.

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 panel 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. Coefficients 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.

EBAL

Control routine for performing mass and energy balance. Outputs mass balance terms, energy balance terms, density, surface temperature and ice height values. Calls: CURVE, ACCRET, QVAP, SHED. (Call to SHED commented out.)

PLOTOUT

See LEWICE subroutines

NWFOIL

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 which AMEAN would not do. Calls: SNORMC, SEGSEC, COMPS, DANGLE.

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, EFFICY, BDYLYR, EBAL, ACCRET are the main ones.)

SEGSEC

Removes points based on panels which intersect or 'nearly' intersect. This criteria is based on two non-neighboring points lying within the minimum panel size distance from each other. Reduces 'spikes' in ice surface. Calls: INTRST (intersection routine)

NWPTS4

Adjusts panel corner points (does not add panels) based on the ratio of the panel's size to its neighbor. If this ratio is greater than SEGTOL (in input file), the corner point is moved such that: 1) the two panels are of equal size; 2) the triangle formed from the three x,y coordinates has the same area as the original three x,y points; 3) the first and third points remain the same, only the middle point is adjusted. Calls: DANGLE, NCHK, NEWERX, COMPS.

NWPTS2

Adds a panel when the difference in panel angle (turning angle) is greater than the user input angle. Creates four corner points from an existing set of three corner points such that: 1) the three panels created are of equal size; 2) the two 'turning angles' created are equal; 3) the area of the trapezoid formed by those four points is the same as the triangle formed by the three original points 4) the first and last points are unchanged, the middle point is replaced by two points whose coordinates are calculated. These four criteria are used to solve for the four 'unknowns' (the two x,y coordinate pairs). The end result is a 4th order polynomial for panel size which was explicitly solved by MathCad 3.1 for Macintosh. Although not explicitly proven, this answer was verified and it can be shown that only one root is physically real. Calls: DANGLE, NCHK, NEWRX1, COMPS.

RMOVE

Removes panels which are smaller than the minimum allowable panels size. This value is currently $10^{-7}$ (non-dimensional) although it could be made a user input by removing comments in WINPUT. The routine first smooths the region (NOT the whole ice shape!) and then spline fits it. The region is replaced by panels of size PANMIN*SEGTOL. Calls: COMPS, DANGLE, AMEAN, SPLIN2.

**Routines called by ICE1 routines:**

CURVE

Determines local radius of curvature by assuming the nearest ±6 panels forms a partial arc of a circle. By comparing the arc length to the chord length, the radius can be determined. Used for ice density correllation (RHOICE) but could have other uses.

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.

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/ $\Delta$T melt) $\Delta$T 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

SHED

Removes ice by aerodynamic forces exceeding adhesion force. Currently commented out, as the experimental data for ice adhesion has too large of an experimental error for this correlation to work correctly. Calls: COEFF

DANGLE

Finds the panel angle.

SNORMC

Finds panel normals. Calls: CROSP, AREAP

CROSSP

Determines cross-product.

AREAP

Detemines area of tetrahedron. Calls: PLIN, DSTPLN

PLIN

Determines line parallel to a segment which passes through a known point.

DSTPLN

Finds minimum distance between a point and a line.

NCHK

Finds if two points from panel addition were calculated in the wrong order and switches them. May not be necessary any longer. Another 'just in case' routine.

NEWERX

Finds new x,y coordinates for NWPTS4.

NEWRX1

Finds two new x,y points for NWPTS2.

SPLIN2

Finds spline coefficients. Used by VEDGE, RMOVE and EFFICY.

NOICE

Finds heat requirements and maximum temperatures for anti-icing systems. Called by EBAL.

## 11.0 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-Beta 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 seperately 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 S24Y 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 compressible 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 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 geometry). 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 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.

## 12.0 Using LEWICE Beta on a PC

This program was successfully compiled and run on a 25MHz Macintosh IIci and took 15-20 minutes for one time step. A LEWICE user with a 486/66MHz machine reported a time of approximately 2 minutes for one time step. 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. Written output has been redistributed to a number of different output files instead of just one. This facilitates plotting on a personal computer. 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 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 program took approximately 1.5 MB of RAM and quite a bit of hard disk space for the output files. (not sure of the exact figure, but on the order of a couple of MB.) 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 mainframe. 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 mostly eliminated now on an iris workstation, this problem may be evident when running the program on a PC.

## 13.0 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.

**Coordinate Input**

In the original data file, all x-coordinates are provided first, with six values on each line except the last. The last line of x-coordinates will have the remaining values and the number of values followed by a 1. This information tells the program how many points are on this line. The 1 at the end tells the program that this is the end of the x-coordinates. The y-coordinates are formatted the same way. Multi-body input is the same, with input format: all x for body 1, all y for body 1; all x for body 2, all y for body 2, etc.

If you are using the seperate geometry files for input, the format has x,y coordinate pairs in two columns of format F12.7. Each body will have a seperate 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. The user has two parameters to control to attempt to obtain better flow solutions.

SEGTOL

The first parameter, SEGTOL, is located near the bottom of the main input file. SEGTOL is the upper limit to the ratio of a given panel size to its neighbor. This definition differs from the previous definition of this variable. After the ice has been added to the surface, those panels are wider than the surrounding panels, sometimes up to 10 times wider. A routine was created to move the intersection point between two panels such that the size of both panels is the same, while not changing the total iced area. This essentially 'shifts' the ice over to the next panel. For a reasonably paneled model, the user will not be able to notice a difference in the ice shape before and after this procedure.

A question remains as to the proper value for this parameter. For most simulations to date, a value of 1.5 has been used for SEGTOL. Values in the range 1.2 to 2 have also been used. SEGTOL must be greater than 1. A value near one means that all panels will be approximately the same size. Normally, the user will want more panels in the icing zone than downstream, so too low of a value should be avoided. The panels which have the greatest change are those where there is the most accretion, therefore a large value of SEGTOL would create ice shapes which appear 'blockish' and will produce a poor flow solution.

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. A value of 10° has been used for turning angle for the test cases. The key criteria for this parameter is the quality of the flow solution aand the number of panels produced. If an excessive number of panels is produced, it may become necessary to try to manually repanel the iced geometry. The number of panels which is considered excessive depends on the type of hardware used. A Cray computer would have essentially no limit, while users with personal computers would like to keep panel addition to a minimum.

A common method to create a better initial panel model is to use small values for SEGTOL and Turning Angle for a short icing run. The first set of coordinates in the ice shape file are the coordinates of the clean airfoil which meets the more stringent criteria. That geometry can then be used as the starting point for future runs.

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, although larger time steps can be run for longer runs. The computation used is

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

This will give the user a rough idea of the time step size needed for an accurate simulation. For long runs (for example 45 min. hold conditions) larger time steps can and should be used. The automated re-paneling procedure used in LEWICE will start to produce hundreds of new panels after a number of time steps. This will be counter productive to the purpose of producing a good flow solution as the panel code will not be able to digest the amount of detail it is provided. A suggested procedure for these types of runs would be to start with five 9-minute time steps for a sample condition and to compare that ice shape against the ice shape produced with nine 5-minute time steps. If the ice shapes are similar, the larger step can be used. If they are much different, the procedure can be repeated using smaller time steps until a consistent output is produced.

**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 trajctories should not be less than one trajectory for every three panels. 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, this will slow down the solution and will be noticable on personal computers. The default case, a 135 panel NACA0012 airfoil will take approximately 2 minutes on a 486/66MHz computer, while a more complex ice shape with nearly 1000 panels and a proportionately larger number of trajectories could take 15 minutes to 1/2 an hour.

**Droplet Distribution**

Most cases run with LEWICE will use a single drop size, the MVD for the flight condition. Although multiple drop size distributions can be run with LEWICE, 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.

**Sand-Grain Roughness**

LEWICE assumes that the roughness on ice can be approximated by standard roughness models for sandpaper type roughness. This roughness controls the laminar/turbulent interface and is very important in glaze ice accretions. **The units of input for roughness are MILLIMETERS, not meters as previously input.** The version of LEWICE on which this manual is based is much less sensitive to this variable than its predecessors. Currently, there is essentially two zones of roughness. The first zone is a low level roughness characteristic of the clean airfoil and rime ice accretions. The second zone is a high level roughness characteristic of glaze ice. For a 21" chord NACA0012 airfoil, there is essentially no 'intermediate zone'. Any input roughness will essentiall lie on either the smooth (laminar) side or the rough (turbulent) side. The transition occurs at a roughness of about 0.2 mm for this airfoil.

The transition point may change for different airfoils, but preliminary case studies suggest that the transition due to different roughness values is quite abrupt. Hence to produce a conservative ice shape for any condition, an arbitrarily high sand-grain roughness can be used. Comparison to experimental ice shapes show that some cases which are in 'mixed' icing conditions are better modeled using the 'smooth zone' roughness values, but all horn shape ice can be modeled only using a large roughness, where the qualifier 'large' is greater than .2 mm for a 21" chord NACA0012. As such, the correlation in the LEWICE manual can be discarded.

**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 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 soley 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.NASA Lewis also has codes which perform more detailed analysis of deicer and anti-icer performance. The surface ,temperature input must be above freezing (in Kelvin) for this option to work..

**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.

**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 independantly for each body.

Inactive Input

Several variables in the original LEWICE input file are no longer used by the LEWICE-Beta code. LEWICE-Beta will read in this input so that old data files do not have to be changed. However, the user should realize that changing these variables will not change the performance of the code, as this data is ignored. The inactive input variables are: ISOL, IPRINT, ITYPE, LCMB, LCMP, LEQM, LSYM, LXOR, LYOR, PRATK, XORC, YORC, XSTOP, YOMAX, YOMIN, VXPIN, VYPIN, QCOND.

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.

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 take the absolute value for pressure and continue with the simulation. A warning message is delivered to the terminal notifying the user of this occurrence. 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.

# 14.0 Equation Derivations

## 14.1 Derivation of Energy Balance

**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,

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

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 compute 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

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

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-Beta 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

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

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

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

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

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

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

**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

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

for the impinging water and

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

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 $\Delta T_m$ is a very small ($10^{-5}$) temperature range over which the ice freezes.

The term $\Delta H_f$ 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

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

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

for the impinging water and

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

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

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

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

for the impinging water and

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

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-Beta 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 occurrance 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 (pp. 337-9 and 713-5). The heat gain is then defined from

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

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-Beta uses the local pressure instead of Mach Number, hence

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

is the form used in LEWICE-Beta.

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.

$$0 = L_v*h_c/c_{p,air}*(MW_{water}*MW_{air})*\mathcal{L}^{-2/3}) * (r_h*P_{v,\infty}/P_\infty - T_0/T_s*P_{v,s}/P_0*(P_0/P_e)^{(1/\gamma)})$$

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

$$\{$$

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

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

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

$$\}$$

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 labeld $f(T_s)$ 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.

## 14.2 Derivation of Mass Balance for LEWICE-Beta

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 $\mathbf{m_{shed}}$ is determined independantly 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. No quantitaive data is available is available for comparison.

The term $\mathbf{m_{evap}}$ 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 $\mathbf{m_{remain}}$ 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 independantly, as long its value does not exceed the amont of unfrozen water available, $\mathbf{m_{rb,out}}$. In that case, no water leaves the control volume ($\mathbf{m_{rb,out}} = 0$) and the value of the term $\mathbf{m_{remain}}$ is determined by the equation below for $\mathbf{m_{rb,out}}$.

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

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

The individual amounts are determined by the freezing fraction

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

and

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

As explained earlier, the term $\mathbf{m_{remain}}$ is presumed known independantly 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.

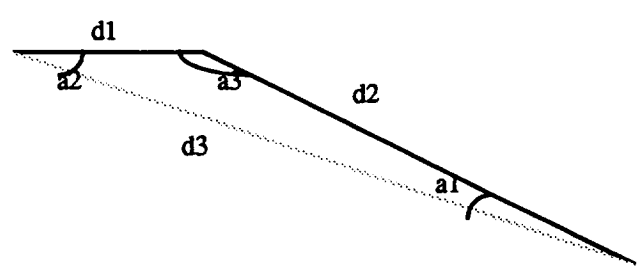
This procedure results in an explicit, marching-type solution for $\mathbf{m_{freeze}}$ and $\mathbf{m_{rb,out}}$ as long as there is a definitive starting point (stagnation point) acquired from the flow solution. This should always be the case for 2D potential flow. For 2D Naviér-Stokes, if there is 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 $\mathbf{m_{remain}}$ or $\mathbf{m_{shed}}$. For 3D flows, a more involved solution mechanism may be necessary. The equations would remain the same, only the solution mechanism would change.

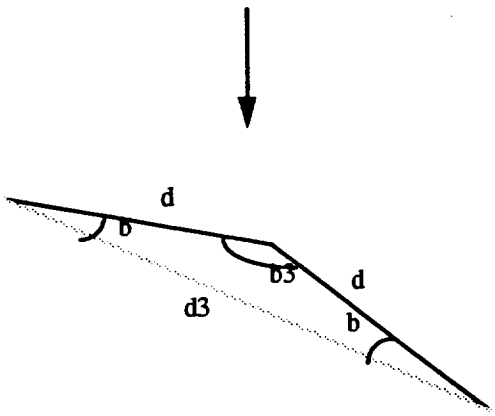
## 14.3 Derivation of Panel Modification for LEWICE-Beta

### 14.3.1 Size Ratio Correction

Purpose: change corner point of two neighboring panels.

A representative case is shown below.






As the two end points do not change, the rest of the ice shape is not affected by this operation.

If the area of the two triangles are equal, then the total iced area remains the same. Additionaly, the distance from the first point to the third point also remains the same.The area of a triangle is A = 1/2 (base)*(height). If the base is d3, the height is d1*sin(a2). From the law of sines,

$$\frac{\sin a1}{d1} = \frac{\sin a3}{d3}$$ Substituting in the area equation

yields $$A = \frac{1}{2}(d3)^2 \frac{(\sin a2)\ (\sin a1)}{\sin a3}$$

For the new triangle, the area is

$$A = \frac{1}{2}(d3)^2 \frac{(\sin b)\ (\sin b)}{\sin b3}$$

For a triangle, the sum of the angles is equal to $\pi$, hence b+b+b3=$\pi$. By subtituting for b3 in the above area equation, the angle b can be solved for. The solution is

$$\tan b = 2\frac{(\sin a2)\ (\sin a1)}{\sin a3}$$

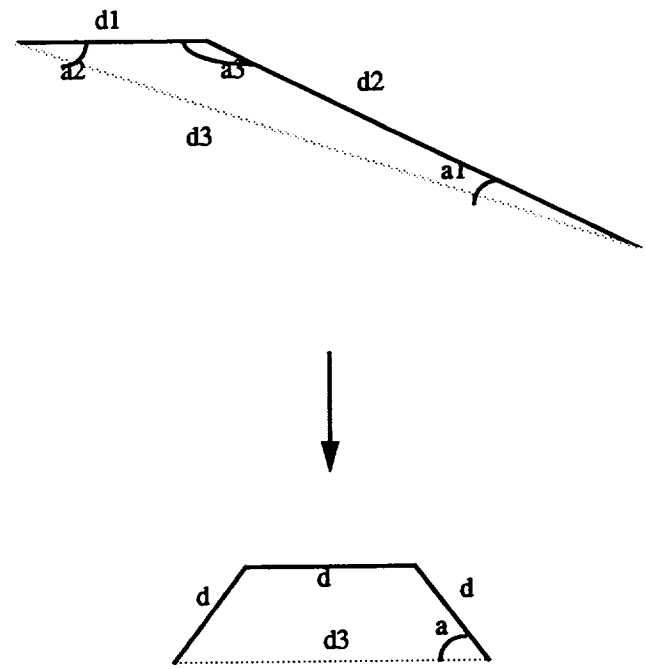
Once this angle is found, the other angle b3 and the panel size d are easily found.

$$d = \frac{d3}{2\cos b}; \quad b3 = \pi - 2b$$

Finally, the new x,y coordinate pair is found from the panel size and angle.

### 14.3.2 Derivation of Equations for Panel Addition

This procedure is similar to the previous derivation. This routine again starts with three coordinates (two panels). This time, the triangle area is equated with the resulting tetrahedron. The base of the tetrahedron is the same as the base of the triangle. As the two sided of the tetrahedron are the same, and the angles are the same, the tetrahedron is a trapezoid.




The area of the tetrahedron is

$$A = d^2(\sin a)\ (1 + \cos a)$$

and the base d3 is related to the panel width d by

$$d3 = d(1 + 2\cos a)$$

As the area is known and the distance d3 is known, these two equations can be combined and solved for d and the angle a. Letting

$$r = \frac{d}{d3}$$

and

$$B = \frac{(\sin a2)\ (\sin a1)}{\sin a3}$$

the following 4th order polynomial is obtained.

$$3r^4 + 8r^3 + 6r^2 = 4B^2 + 1$$

By using the following definitions,

$$C = 4B^2$$

$$D = C\sqrt{(C+1)} \quad ,$$

$$E = \sqrt[3]{D - C}$$

$$F = \sqrt[3]{D + C}$$

$$G = \sqrt{4 + 6(D + C)}$$

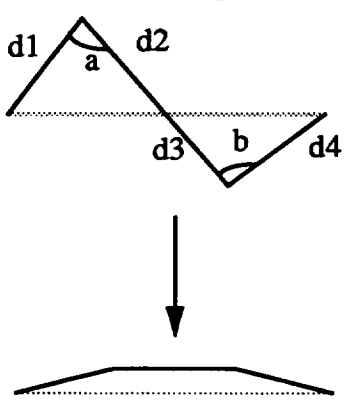
the solution of the polynomial is

$$r = \frac{\sqrt{\frac{12G - G^3 + 16}{G}} + G - 4}{6}$$

The turning angle is found from one of the previous relationships. Finally, the new x,y coordinate pairs are found from the panel size and angle.

### 14.3.3 Derivation of Equations for Correcting 'Jaggedness'

This procedure is similar to the previous derivation. This routine starts with four coordinates (three panels). The area of these panels is computed with respect to the straight line between the two endpoints, as shown below.




The area of the first region is found by subtracting the areas of the two triangles above. From before, a triangle's area can be expressed as

$$A = \frac{1}{2} d1\, d2 \sin a$$

The distances can be expressed as the x-distance divided by the cosine of the slope,

$$d1 = \frac{(x2 - x1)}{\cos\beta 1}$$

The triangle area is then

$$A = \frac{1}{2}(x2 - x1)\,(xm - x2)\,(\tan\beta 2 - \tan\beta 1)$$

where xm is the point where the second panel crosses the dashed line above. As the tangent is simply the slope of the line, the area can be expressed in terms of the three x,y coordinates of the triangle. This yields

$$A = \frac{1}{2}\left((x2 - x1)\,(ym - y2) - (xm - x2)\,(y2 - y1)\right)$$

The second triangle has a similar form. By subtracting the two areas and eliminating the point (xm, ym), the following area equation is obtained.

$$A = \frac{1}{2}\left((x3 - x1)\,(y2 - y4) + (x4 - x2)\,(y3 - y1)\right)$$

This area is equated to the trapezoid area given earlier. The factor B in this case is

$$B = (x3 - x1)\,(y2 - y4) + (x4 - x2)\,(y3 - y1)$$

The new panel size and x,y coordinate pairs are found using the same equations as before.

**Equation for Equal Area Ice Addition**

The predicted x,y coordinates are

$$xn_i = x_i + \cos\alpha\,(\Delta ice_i C_i + \Delta ice_{i+1} C_{i+1})$$

and

$$yn_i = y_i + \sin\alpha\,(\Delta ice_i C_i + \Delta ice_{i+1} C_{i+1})$$

where $\alpha$ = growth angle and the correction factors $C_i$ and $C_{i+1}$ are initially one and are equal to the ratio of the iced area $\Delta ice * \Delta s$ divided by the area of the tetrahedron added. This correction factor is applied to the predicted set of (xn,yn) coordinates in the next iteration. The iteration process is repeated for 20 steps, at which point the difference in area is presumed to be small.

## 14.4 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 xxx 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.

## 14.5 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 $\theta$ = Contact Angle.

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

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

where $\tau_f$ = 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' is a function of contact angle only, and is given by

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

## 14.6 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. This yields

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

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 $\Delta x$ is the thickness of each material and k is its thermal conductivity. Both variables are input by the user for each layer.

## 14.7 Anti-Ice with Internal Heat Source

An internal heat source suich 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 thru 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

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

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 procedes 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.

## 14.8 Drag Correction

The equation for drag on a sphere as a function of Reynolds number is normally given as

$$C_d = 24/Re + .4 + 6/(1+\sqrt{Re})$$

Plots of drag vs. Re for several Mach numbers show that drag increases with increasing Mach number above a Mach number of 0.3. A curve-fit of these values gives the approximation

$$C_{d,\,comp.} = C_{d,\,inc}*(.94572+.00124*M+.6027*M^2.$$

## 14.9 Ice Density

The current correlation for ice density was developed from experimental data on iced cylinders. Its form is

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

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 uses the diameter of the inscribed circle at the leading edge); $T_c$ = surface temperature in degrees Celcius.

## 14.10 Radius of Curvature

The radius of curvature for the above computation is found by assuming the points at the leading edge of an iced airfoil will fit an equation for the partial arc of a circle. The arc length is given by

$$S = R\theta$$

while the distance from the first to the last point is obtained from the law of cosines,

$$C = R\sqrt{[2(1-\cos\theta)]}$$

S and C are both measurable, so the two equations are solved for the two unknowns, one of which is the radius of curvature.

## 14.11 Flow Derivative

The derivative of velocity with respect to S is needed for the boundary layer integration. Standard difference formulas rely on equal spacing of panels. As the panel spacing is not uniform, a variable spaced difference formula must be used. The formula is derived from the Taylor polynomial expansions about panel i,

$$V_{i+1} = V_i + \frac{dV}{ds_i}(s_{i+1}-s_i) + \frac{d^2V}{ds_i^2}\frac{(s_{i+1}-s_i)^2}{2} + \frac{d^3V}{ds_i^3}\frac{(s_{i+1}-s_i)^2}{2}$$

and

$$V_{i-1} = V_i + \frac{dV}{ds_i}(s_{i-1}-s_i) + \frac{d^2V}{ds_i^2}\frac{(s_{i-1}-s_i)^2}{2} + \frac{d^3V}{ds_i^3}\frac{(s_{i-1}-s_i)^2}{2}$$

Arranging these so as to eliminate the second order terms gives

$$\frac{dV}{ds_i} = \frac{\frac{s_{i+1}-s_i}{s_i-s_{i-1}}(V_i-V_{i-1}) + \frac{s_i-s_{i-1}}{s_{i+1}-s_i}(V_{i+1}-V_i)}{s_{i+1}-s_{i-1}}$$

## 14.12 Integral 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 ploynomial 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 $\delta^2$ can be calculated;

3)The first shape factor $\Lambda(s)$ is found;

4) The displacement thickness, $\delta_1$, and the shearing stress at the wall, $\tau_o$, are found;

5) The boundary layer thickness $\delta(s)$ 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,

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

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

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

## 14.13 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. The form is

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

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

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

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,

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

## 14.14 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

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

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 coeffi-

cients 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

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

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

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

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

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

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

## 15.0 References


1)Ruff, G. A. and B. M. Berkowitz, "Users Manual for the NASA Lewis Ice Accretion Prediction Code (*LEWICE*)," NASA CR 185129, May 1990.

2)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.

3)Frost, W., Chang, H., Shieh, C. and K. Kimble, "Two Dimensional Particle Trajectory Computer Program," Interim Report for Contract NAS3-224488, 1982.

4)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.

5)Potapczuk, M. G., Bragg, M. B., Kwon. O.J. and L. N. Sankar, "Simulation of Iced Wing Aerodynamics," NASA TM 104362, May 1991.

6)Ide, R. F., "Liquid Water Content and Droplet Size Calibration of the NASA Lewis Icing Research Tunnel," NASA TM 102447, AVSCOM ZTM 89-C-014, AIAA 90-0669, Jan. 1990.

7)Bird, R.B., Stewart, W. E., and E. N. Lightfoot, Transport Phenomena, John Wiley & Sons, Inc., New York, 1960.

8)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.

9)Wright, W. B., "Simulation of Two-Dimensional Icing, De-Icing and Anti-Icing Phenomena," PhD Dissertation, University of Toledo, Toledo, Ohio, Dec. 1991.

10)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.

11)Wright, W. B., "Advancements in the LEWICE Ice Accretion Model," NASA CR 191019, Jan. 1993.

12)Anderson, D. A., Tannehill, J. C., and R. H. Pletcher, "Computational Fluid Mechanics and Heat Transfer," McGraw-Hill Book Company, New York, 1984.

13)White, F. M., Viscous Fluid Flow, McGraw-Hill, Inc., 1974.

14)Schlichting, H., *Boundary-Layer Theory.* F. J. Cerra, ED. New York, New York: McGraw-Hill, 1979.

15)Olsen, W. and E. Walker, "Experimental Evidence for Modifying the Current Physical Model for Ice Accretion on Aircraft Surfaces," NASA TM 87184, 1986.

16)Hansman, R. J., A. Reehorst, and J. Sims, "Analysis of Surface Roughness Generation in Aircraft Ice Accretion," AIAA-92-0298, Jan. 1992.

17)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.

18)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.

19)Kays, W. M. and M. E. Crawford, Convective Heat and Mass Transfer, 2nd Edition, MacGraw-Hill Book Company, New York, 1980.

20)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.

21)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.

22)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.

23)Minkowycz, W. J., Sparrow, E. M., Schneider, C. E., and R. H. Pletcher, "Handbook of Numerical Heat Transfer," John Wiley and Sons, Inc., New York, 1988.

24)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.

25)Macklin, W. C., "The Density and Structure of Ice Formed by Accretion," *Quarterly Journal of the Royal Meteorological Society*, January 1962, Vol. 88, No. 3375, pp. 30-50.

26)Rios, M., "Icing Simulations Using Jones' Density Formula for Accreted Ice," AIAA-91-0556.

27)Jones, K. F., "The Density of Natural Ice Accretions," *Fourth International Conference on Atmospheric Icing of Structures*, 1988, pp. 114-117.

28)Scavuzzo, R. J., M. L. Chu and C. J. Kellackey, "Impact Ice Stresses in Rotating Airfoils," AIAA 90-0198, Jan. 1990.

29)Shin, J. and T. H. Bond, "Results of an Icing Test on a NACA 0012 Airfoil in the NASA Lewis Icing Research Tunnel," AIAA-92-0647, Jan. 1992.

# 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**
October 1994

**3. REPORT TYPE AND DATES COVERED**
Final Contractor Report

**4. TITLE AND SUBTITLE**
Update to the NASA Lewis Ice Accretion Code LEWICE

**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.
Engineering Services Division
2001 Aerospace Parkway
Brook Park, Ohio 44142

**8. PERFORMING ORGANIZATION REPORT NUMBER**
E-9150

**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-195387

**11. SUPPLEMENTARY NOTES**
Project Manager, Mark Potapczuk, Propulsion Systems Division, organization code 2720, NASA Lewis Research Center, (216) 433-3919.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified - Unlimited
Subject Categories 34 and 64

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (*Maximum 200 words*)

This report is intended as an update to NASA CR-185129 "User's Manual for the NASA Lewis Ice Accretion Prediction Code (LEWICE)." It describes modifications and improvements made to this code as well as changes to the input and output files, interactive input, and graphics output. The comparison of this code to experimental data is shown to have improved as a result of these modifications.

**14. SUBJECT TERMS**
Icing; Computational methods

**15. NUMBER OF PAGES**
50

**16. PRICE CODE**
A03

**17. SECURITY CLASSIFICATION OF REPORT**
Unclassified

**18. SECURITY CLASSIFICATION OF THIS PAGE**
Unclassified

**19. SECURITY CLASSIFICATION OF ABSTRACT**
Unclassified

**20. LIMITATION OF ABSTRACT**