NASA/CR—1999-209409

User Manual for the NASA Glenn Ice Accretion Code LEWICE
Version 2.0

William B. Wright
Dynacs Engineering Company, Inc., Brook Park, Ohio

September 1999

NASA/CR—1999-209409

User Manual for the NASA Glenn Ice Accretion Code LEWICE
Version 2.0

William B. Wright
Dynacs Engineering Company, Inc., Brook Park, Ohio

Prepared under Contract NAS3–98008

National Aeronautics and
Space Administration

Glenn Research Center

September 1999

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A09

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A09

Abstract

A research project is underway at NASA Glenn to produce a computer code which can accurately predict ice growth under a wide range of meteorological conditions for any aircraft surface. This report will present a description of the code inputs and outputs from version 2.0 of this code, which is called LEWICE. This version differs from previous releases due to its robustness and its ability to reproduce results accurately for different spacing and time step criteria across computing platform. It also differs in the extensive effort undertaken to compare the results against the database of ice shapes which have been generated in the NASA Glenn Icing Research Tunnel (IRT)[1].

This report will only describe the features of the code related to the use of the program. The report will not describe the inner workings of the code or the physical models used. This information is available in the form of several unpublished documents which will be collectively referred to as a Programmers Manual for LEWICE[2] in this report. These reports are intended as an update/ replacement for all previous user manuals of LEWICE. In addition to describing the changes and improvements made for this version, information from previous manuals may be duplicated so that the user will not need to consult previous manuals to use this code.

Summary

The baseline module of 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 can be calculated in LEWICE using the Douglas Hess-Smith 2-D panel code (S24Y)[3]. In the current version, the potential flow module can be bypassed by setting a flag in the user input file. In this mode, the user has the option to call a grid generator and grid-based flow solver (Euler or Naviér-Stokes) or to read in the solution file from this flow solver. For any of the methods chosen, the flow solution is then used to calculate the trajectories of particles and the impingement points on the body[4]. These calculations are performed to determine the distribution of liquid water impinging on the body, which then serves as input to the icing thermodynamic code. The icing model, which was first developed by Messinger[5], is used to calculate the ice growth rate at each point on the surface of the geometry. By specifying an icing time increment, the ice growth rate can be interpreted as an ice thickness which is added to the body, resulting in the generation of new coordinates. This procedure is repeated, beginning with the flow calculations, until the desired icing time is reached.

The operation of LEWICE is illustrated through the use of several examples. These examples are representative of the types of applications expected for LEWICE. A more extensive set of example cases are provided in a validation report on this version of LEWICE[1]. LEWICE has been used to calculate a variety of ice shapes, and is considered a validated production code. However, development continues toward improvement of the physical models. Any modifications identified as a result of this research, or of additional experimental results, will be incorporated into the model.

Table of Contents

Abstract ii
Summary iii
Table of Contents iv
List of Tables viii
List of Figures ix
Chapter 1: Introduction 1
Chapter 2: Installation Procedure 3
Chapter 3: Running LEWICE on a PC 4
3.1 LEWICE Quick Start Guide 4
3.2 PC Requirements 5
Chapter 4: Running LEWICE on Other Systems 6
4.1 Source Code Modification 6
Chapter 5: Interactive Input 7
5.1 Enter Input File Name 7
5.2 Enter File Name for Body 1 8
5.3 Confirmation of Warnings 9
Chapter 6: Main Input File 11
6.1 Listing Variables in a Namelist 11
6.1.1 Variable not in namelist error 11
6.1.2 Namelists Listed Out of Order or Missing Namelist 12
6.2 Line 1 13
6.3 LEW20 Namelist 13
6.3.1 ITIMFL 13
6.3.2 TSTOP 15
6.3.3 IBOD 16
6.3.4 IFLO 17
6.3.5 DSMN 18
6.3.6 NPL 20
6.3.7 RHOP 20
6.3.8 IGRID 21
6.3.9 IDEICE 22
6.4 DIST Namelist 23
6.4.1 FLWC 24
6.4.2 DPD 25
6.5 ICE1 Namelist 27
6.5.1 CHORD 27
6.5.2 AOA 28
6.5.3 VINF 28
6.5.4 LWC 29
6.5.5 TINF 30
6.5.6 PINF 31
6.5.7 RH 32
6.6 LPRNT Namelist 33
6.6.1 FPRT 33

6.6.2 HPRT 33
6.6.3 BPRT 34
6.6.4 EPRT 34
6.6.5 MPRT 34
6.6.6 TPRT 35
6.6.7 IDBF 35
6.7 Complete Example Case Input File 36
6.8 Extent of Data in Validation Database 38
Chapter 7: Body Geometry Input 39
7.1 Number of Points 44
7.2 Blank Lines 44
7.3 Too Few Points 44
7.4 Too Many Points 45
7.5 Body Not Closed 45
7.6 Small Point Spacing 45
7.7 Body Input Backwards 46
7.8 Large Angles Between Neighboring Segments 46
7.9 Intersecting Bodies 48
7.10 Concentric Bodies 48
7.11 Bodies Out of Order 48
Chapter 8: Optional Input Files 50
8.1 Deice Input File (deicei.inp) 50
8.1.1 Deicer Input File Name 50
8.1.2 TSURF 51
8.1.3 IEVAP 52
8.1.4 ITHERM 53
8.1.5 Number of Layers 53
8.1.6 Length 54
8.1.7 Conductivity 55
8.1.8 LHEAT 56
8.1.9 Number of Heat Transfer Coefficients 56
8.1.10 S/C 57
8.1.11 HTC 58
8.2 Grid Input Files (xy.plt and q.plt) 60
8.2.1 Grid read statements (xy.plt) 60
8.2.2 Flow solution read statements (q.plt) 61
Chapter 9: Output Files 62
9.1 beta.dat 62
9.2 dens.dat 63
9.3 dyice.dat 63
9.4 final1.dat 63
9.5 final2.dat, final3.dat, final4.dat, final5.dat 63
9.6 fixed.dat 64
9.7 flow.dat 64
9.8 fract.dat 64
9.9 htc.dat 65

9.10 ice1.dat 65
9.11 ice2.dat, ice3.dat, ice4.dat, ice5.dat 65
9.12 imp.dat 65
9.13 junk.dat 66
9.14 mass.dat 66
9.15 misc.dat 67
9.16 pres.dat 67
9.17 qener.dat 67
9.18 temp.dat 68
9.19 thick.dat 68
9.20 traj1.dat 68
9.21 traj2.dat, traj3.dat, traj4.dat, traj5.dat 69
9.22 xkinit.dat 69
9.23 xkinit2.dat 69
Chapter 10: Additional Output Files 70
10.1 noice.dat 70
10.2 ctemp.dat 70
10.3 geometry.dat 70
Chapter 11: Utility Programs 71
11.1 Program THICK 71
11.1.1 Interactive Input for Program THICK 71
11.1.2 File Format for Clean Geometry and Iced Geometry 73
11.2 Program EXPAND 74
11.2.1 Interactive Input for Program EXPAND 74
11.2.2 File Format for Input and Output Geometries 75
11.3 Program CONVERT 76
11.3.1 Interactive Input for Program CONVERT 76
11.3.2 Format for the LEWICE 1.6 Input File 77
11.3.3 Warning Message for CONVERT 77
11.4 Program TOBINARY 77
11.5 Program TOASCII 78
Chapter 12: Runtime Errors 79
12.1 Flow Solver Errors 79
12.2 Errors when Using Grid Solutions 81
12.3 Trajectory Errors 82
12.4 Energy Balance Errors 84
12.5 Ice Growth Errors 85
Chapter 13: Example Cases 89
13.1 Case 1: Run 072501 from Validation Report 89
13.2 Case 2: Run 072504 from Validation Report 93
13.3 Case 3: Cylinder Benchmark Case 97
13.4 Case 4: Langmuir 'D' distribution 102
13.5 Case 5: Exceedence condition 106
13.6 Case 6: Exceedence condition w/Langmuir 'D' distribution 111
13.7 Case 7: Run 072504 without automated time step and fewer time steps 116
13.8 Case 8: Run 072504 with Larger Point Spacing 120

13.9 Case 9: First benchmark conditions with an evaporative hot air anti-icer. 124
13.10 Case 10: First benchmark condition with a running wet electrothermal anti-icer. 129
13.11 Case 11: 3-body example 133
13.12 Case 12: Grid input 139
13.12.1 Grid Input Problems 143
13.12.2 LEWICE Errors Associated with Grid Input Problems 145
Chapter 14: User Tips and Notes 146
14.1 Old Input Files 146
14.2 Input File Errors 146
14.3 Porting ASCII Files 147
14.4 PC Application 147
14.5 Listing Variables in a Namelist 148
14.6 Multiple Bodies / Multi-Element Airfoils 148
14.7 Panel Criteria 148
14.8 Time Step 149
14.9 DSMN (Point Spacing) 150
14.10 Number of Trajectories 151
14.11 Grid-Based Flow Solutions 151
14.12 Anti-Icing 151
14.13 Droplet Distribution 152
14.14 Collection Efficiencies and Droplet Trajectories 153
14.15 Chord Length 154
14.16 Multiple Stagnation Points 154
14.17 Flow Code Limitation 154
14.18 Static Pressure/Altitude 155
14.19 Temperature 155
14.20 Relative Humidity 155
14.21 Printer Flags 156
14.22 Geometry Input 156
Chapter 15: Procedure for Acquiring LEWICE or other NASA Icing Codes 158
15.1 Current Address for Icing Code Requests 158
Chapter 16: Sample Code Request Letter 159
Chapter 17: Technical Support 160
References 161

List of Tables

Table 1: Example LEW20 Namelist 23
Table 2: Example of bad input for FLWC 24
Table 3: Example of correct input for FLWC 24
Table 4: Example DIST Namelist 27
Table 5: Example ICE1 Namelist 33
Table 6: Example LPRNT Namelist 36
Table 7: Example Test Input File 37
Table 8: Example Body Geometry (NACA 23014) 39
Table 9: Example Deicer Input (deicei.inp) 59
Table 10: Example of Data Format in Output Files 62
Table 11: Main Input File for Example Case 1 90
Table 12: Main Input File for Example Case 2 94
Table 13: Main Input File for Example Case 3 98
Table 14: Main Input File for Example Case 4 103
Table 15: Main Input File for Example Case 5 107
Table 16: Main Input File for Example Case 6 112
Table 17: Main Input File for Example Case 7 117
Table 18: Main Input File for Example Case 8 121
Table 19: Main Input File for Example Case 9 125
Table 20: Deicei.inp input file for Example Case 9 126
Table 21: Main Input File for Example Case 10 129
Table 22: Deicei.inp Input File for Example Case 10 130
Table 23: Main Input File for Example Case 11 134
Table 24: Main Input File for Example Case 12 140

List of Figures

Figure 1: Test Conditions in Database Used for Code Validation 26
Figure 2: Example of a 45 degree panel angle 47
Figure 3: Example of a 135 degree panel angle 47
Figure 4: Ice Shape for Example Case 1 91
Figure 5: Ice Thickness for Example Case 1 91
Figure 6: Heat Transfer Coefficient for Example Case 1 92
Figure 7: Pressure Coefficient for Example Case 1 92
Figure 8: Collection Efficiency for Example Case 1 93
Figure 9: Ice Shape for Example Case 2 95
Figure 10: Ice Thickness for Example Case 2 95
Figure 11: Heat Transfer Coefficient for Example Case 2 96
Figure 12: Pressure Coefficient for Example Case 2 96
Figure 13: Collection Efficiency for Example Case 2 97
Figure 14: Ice Shape for Example Case 3 99
Figure 15: Ice Thickness for Example Case 3 100
Figure 16: Heat Transfer Coefficient for Example Case 3 100
Figure 17: Pressure Coefficient for Example Case 3 101
Figure 18: Collection Efficiency for Example Case 3 101
Figure 19: Collection Efficiency as a Function of Wrap for Example Case 3 102
Figure 20: Ice Shape for Example Case 4 104
Figure 21: Ice Thickness for Example Case 4 104
Figure 22: Heat Transfer Coefficient for Example Case 4 105
Figure 23: Pressure Coefficient for Example Case 4 105
Figure 24: Collection Efficiency for Example Case 4 106
Figure 25: Ice Shape for Example Case 5 108
Figure 26: Ice Thickness for Example Case 5 109
Figure 27: Heat Transfer Coefficient for Example Case 5 109
Figure 28: Flow Solution for Example Case 5 110
Figure 29: Collection Efficiency for Example Case 5 110
Figure 30: Ice Shape for Example Case 6 113
Figure 31: Ice Thickness for Example Case 6 113
Figure 32: Comparison of Ice Shapes for Monodispersed and Langmuir 'D' Cases 114
Figure 33: Comparison of Ice Thicknesses for Monodispersed and Langmuir 'D' Cases 114
Figure 34: Heat Transfer Coefficient for Example Case 6 115
Figure 35: Pressure Coefficient for Example Case 6 115
Figure 36: Collection Efficiency for Example Case 6 116
Figure 37: Ice Shape for Example Case 7 118
Figure 38: Ice Thickness for Example Case 7 118
Figure 39: Heat Transfer Coefficient for Example Case 7 119
Figure 40: Pressure Coefficient for Example Case 7 119
Figure 41: Collection Efficiency for Example Case 7 120
Figure 42: Ice Shape for Example Case 8 122
Figure 43: Ice Thickness for Example Case 8 122

Figure 44: Heat Transfer Coefficient for Example Case 8 123
Figure 45: Pressure Coefficient for Example Case 8 123
Figure 46: Collection Efficiency for Example Case 8 124
Figure 47: Heat Requirement for the Evaporative Hot Air System in Example Case 9 127
Figure 48: Surface Temperature for the Evaporative Hot Air System in Example Case 9 128
Figure 49: Hot Air Temperature for the Evaporative Hot Air System in Example Case 9 128
Figure 50: Heat Requirement for the Running Wet Thermal Deicer in Example Case 10 132
Figure 51: Heater Temperature for the Running Wet Thermal Deicer in Example Case 10 132
Figure 52: Ice Shape on Slat and Main Elements for Example Case 11 135
Figure 53: Ice Shape on Flap for Example Case 11 135
Figure 54: Ice Thickness on Slat for Example Case 11 136
Figure 55: Ice Thickness on Main Element for Example Case 11 136
Figure 56: Ice Thickness on Flap for Example Case 11 137
Figure 57: Heat Transfer Coefficient for Example Case 11 137
Figure 58: Pressure Coefficient for Example Case 11 138
Figure 59: Collection Efficiency for Example Case 11 138
Figure 60: Ice Shape for Example Case 12 141
Figure 61: Ice Thickness for Example Case 12 141
Figure 62: Heat Transfer Coefficient for Example Case 12 142
Figure 63: Pressure Coefficient for Example Case 12 142
Figure 64: Collection Efficiency for Example Case 12 143

Chapter 1: Introduction

The evaluation of both commercial and military flight systems in icing conditions continues to be important in the design and certification phases of aircraft systems. These systems are evaluated in flight in natural icing, in a simulated cloud produced by a lead aircraft, and in ground test facilities equipped with a droplet spray system. Icing testing is relatively expensive, and each test technique, i.e., flight or ground testing, has operational considerations which limit the range of icing conditions that can be evaluated. As a result, it benefits 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 on the surface of interest. Analytical modeling of the ice accretion process allows the evaluation of a wide range of proposed test conditions in order to identify those that will be most critical to the flight system. This can 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 ice protection code to allow more complete evaluation in the design phases of the aircraft.

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 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, and continued until the desired icing time has been reached.

Ice accretion shapes for cylinders and several single-element and multi-element airfoils have been calculated using this computer code. The calculated results have been compared to experimental ice accretion shapes obtained both in flight and in the Icing Research Tunnel at NASA Glenn Research Center. The results of this comparison with the experimental database is described in a recent contractor report[1].

Chapter 2: Installation Procedure

The PC executable for LEWICE is distributed on a CD-ROM compatible with PC, Macintosh and Unix systems. The PC executable will run only on the PC systems or under PC emulation with the other operating systems. Simply copy the executable and input files to a suitable directory onto a hard disk to install LEWICE. LEWICE cannot be run from the CD. Users who wish to run LEWICE on other systems should consult Chapter 4.

Chapter 3: Running LEWICE on a PC

The PC executable provided on the disk can be run by double-clicking the icon in Windows 3.1 or higher. It can also be run by typing "lewice" at a DOS prompt if the user is in the directory where LEWICE resides or if the proper path has been set in the AUTOEXEC.BAT file. However, it must be run from the user's hard disk and not from the CD-ROM. When run from Windows, a console shell opens for interactive input and output. This console shell disappears when the run is finished. For this reason, it is highly recommended that the user run the PC executable from a DOS Shell instead of from the console shell.

3.1 LEWICE Quick Start Guide

This section is intended for users unfamiliar with LEWICE and/or DOS Shell commands.

The commands below (indented bold lines) should be typed at the C:\ prompt in a DOS Shell window on a Windows95 machine. Alternatively, the user can use the Windows interface for any of the commands shown.

1) Create a directory on the hard drive to store output

md lewice

2) Insert the LEWICE CD-ROM and copy all files from this disk to the lewice directory as described in the Installation Procedure.

3) Inside the lewice directory, make additional directories for each run

md case1

4) Run LEWICE

lewice <return>

- program will prompt for input file name. Enter the following:

case1.inp <return>

- after printing a copyright notice, the program will prompt for a geometry file name. Enter the following:

```
case1.xyd <return>
```

- if any warning messages appear, type

```
Y <return>
```

to continue the simulation.

5) copy data files to the proper directory

```
copy *.dat case1
```

6) repeat steps 3-5 for each case to be run

3.2 PC Requirements

This program was run successfully on a Compaq Presario® with a 133MHz Pentium® CPU and 24MB of RAM running Windows95®. This machine took 9 minutes to run the first example case provided. Lower end systems were not available for testing. It is believed that the program should run on at least a 486 and Windows 3.1, although this has not been verified. LEWICE 2.0 was developed on a Silicon Graphics Indigo2 running IRIX 6.2. Various changes were made in this code to make conversion to personal computers much easier. The validation report[1] shows comparisons of ice shape predictions on a personal computer and a Silicon Graphics workstation. The ice shape predictions on various PCs using the executable provided on the distribution disk were identical to the results shown in the validation report.

Most of the output data is provided in columns of text, with a text header identifying the variable. This file format can be easily imported into any spreadsheet package for plotting. The program takes about 650 KB of hard disk space for the executable, and several megabytes for output files. The second example case shows the program's potential to produce large output files. The output files for this case take only 3.3 MB of disk space. However, several of the larger output files were not printed in this example and output was further reduced using the print flags in the main input file. If this same case were to be run with all of the outputs activated, the output for this case would occupy over 45 MB of disk space.

Chapter 4: Running LEWICE on Other Systems

Source code is not included in the general release of LEWICE. The code has been successfully compiled and tested on Sun and SGI workstations. The executables for those systems are included on the distribution CD-ROM. The SGI executable was created on the SGI under IRIX 6.2. Test cases were run on a SGI Indigo2 and a SGI Power Challenge. The SUN executable was created under Solaris 2.6. Test cases were run on a Sun Enterprise 3500 system. Users who are interested in running LEWICE on these systems or other platforms should contact the Icing Branch office for the source code and the documents describing the benchmark procedure for different platforms. Due to a small variability in output for different compilers and platforms, it is important for users to revalidate LEWICE using the benchmark tests if the code is recompiled. The complete set of benchmark tests is included on the distribution CD-ROM. The run procedure described in Chapter 3 can also be used on unix machines, with the exception that DOS commands should be replaced with unix commands.

4.1 Source Code Modification

Developers who wish to obtain source code for LEWICE 2.0 for modification should also contact the Icing Branch office for access to source code. If the code is modified in any way and not simply recompiled, the developer will need to repeat the complete set of validation tests[1] in order to establish the validity of the modified code. Several different combinations of compiler flags were used during testing to establish code sensitivity to those flags. It was discovered that the variation was very minimal to non-existent, depending upon the flag set. For the PC executable, the default flags were used in Digital's Virtual Fortran 5.0.A with the configuration set to "Win32 release". For the Sun and SGI executables, the only flag used was the optimization flag O2. The precise procedure used to compile the codes and test the executable will be supplied with any source code distribution.

Chapter 5: Interactive Input

This section will describe the interactive input required by LEWICE. Error messages are also described along with suggested corrective measures. The error messages are generated by the operating system and are unique to the platform being used. Since LEWICE can be compiled and run on a wide variety of operating systems and hardware platforms, it is not possible to identify the error messages for every platform. Two operating systems were chosen to provide representative examples of the operating system errors. The systems chosen were those generated by Windows95/DOS for PCs and those generated by IRIX 6.2 for SGI workstations. The latter example should be representative of error messages provided by other unix operating systems. The error generated by the system will be listed in italics, while the explanation of the error is listed in plain text. The errors listed in this document are those which result from running the executable provided on the LEWICE distribution disks. Users who recompile the program for their system may generate different errors.

5.1 Enter Input File Name

The first interactive input directs the user to input the name of the main input file. The name can be up to 80 characters long. This filename length is necessary as the user must also input the directory path of this file if it is different from the directory containing the program. Please read the error messages in this section concerning the proper form of the input using a directory path. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 35 named
last format:
Unit 35 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 35, file C:\Lewice\test.inp
```

This error indicates that the file name input does not exist, or does not exist in this directory. Common problems:

1) The file name was not typed correctly (in the example case, remember to include the extension - i.e. use "case1.inp" not simply "case1");

2) The input file is in a different directory than the program. The input file can be in a different directory than the program, but in order for LEWICE to recognize the input file the path must be specified. For example, use "inputs\naca0012\case1.inp" instead of simply "case1.inp" to read the input file "case1.inp" in the directory "inputs" and subdirectory "naca0012". **Note:** The above example used the DOS directory convention of backward slashes "\" to list subdirectories. IRIX and many other unix systems use forward slashes "/" instead.

PC Note: To get to the root directory, first type a backward slash "\", then the path and file name. For example, the command "\lewice\inputs\naca0012\case1.inp" can also be used to read the file "case1.inp" in the directory "C:\lewice\inputs\naca0012".

Unix Note: It is common practice in unix to place all programs in a predefined directory such as /usr/bin so that everyone using that system can run the program. The path for specifying the input file in this case is to provide the path from the directory the user is in. For example, if the user is in their home directory and the input file is in the home directory, no path should be provided. If the user is in their home directory and the input files are in directory ../inputs/naca0012, then the proper path to input is "inputs/naca0012/case1.inp". If the user is in directory ../inputs/naca0012 and the input file is in this directory, then no path needs to be provided in this case either. **P.S.:** This sequence is correct based on the IRIX 6.2 operating system. Behavior for other unix operating systems are expected to be the same, but potentially could be different.

5.2 Enter File Name for Body 1

This interactive input directs the user to input the name of the file containing the body geometry points for the first body being used. If only one body is being run, the program will not ask for additional geometry files. If more than one body is being simulated, the code will prompt the user for additional file names for each body. If the file cannot be accessed, the error message listed previously for the main input file name will also be generated. In this case however, the unit number referenced in the error message will be unit 44 for the first body, unit 45 for the second body, unit 46 for the third body, unit 47 for the fourth body and 48 for the fifth body. The corrective measures concerning file name and directory path listed earlier for this error will also apply to errors involving the body geometry.

5.3 Confirmation of Warnings

The file name inputs for the main input file name and geometry input file name(s) are the only interactive inputs to LEWICE unless the program has a problem reading the input files. LEWICE may issue a warning message or an error message in accordance with the nature of the problem. These warning messages and error messages are listed in the following chapters which describe the input files for LEWICE. After any warning messages are issued, the user will have to confirm the settings to continue the run from the following interactive input:

There is a problem with your input.(value) warnings have been issued. Do you wish to continue? Answer Y or N

This question will be asked after all of the input fields have been processed within the code. The listing (value) in the above statement will be replaced by the actual number of warning messages issued. The program will only continue running if the response to this question is Y or y. Other responses (including no response) will be interpreted as a negative response and the program will exit. This message will not appear if there are no warnings issued. If the user wishes to correct the input file based on a warning message, simply respond negatively to the prompt and edit the file after the program quits.

For every error message, the following line will be printed:

The code will stop because of the above error.

This message indicates that the statement above this error message has caused the program to exit. The program will then continue processing the input file. Once all of the input fields have been processed, the following line will be printed:

(value) problems have occurred with the input which cannot be corrected by the code. Please correct the input and resubmit the case.

Once again, the (value) field in the above statement will be replaced with the actual number of errors detected. After this statement prints out, the user must correct the errors listed. Refer to the sections on that variable for assistance with correcting the error.

Chapter 6: Main Input File

This section will define the variables in the main input file to LEWICE. Several notes are also added to aid the user in properly setting up an input file for this code. Any warning messages or error messages based on improper inputs will also be described in this section.

In the examples provided in this section, the input file was named “test.inp” and was located in a directory called “Lewice”. In every instance where a screen message is referenced, the message is also written to the debug file, “junk.dat”.

6.1 Listing Variables in a Namelist

The main input file for LEWICE is formatted as a series of namelists. Variables in namelist format are input on separate lines. Each line contains a unique variable which is listed in that namelist. The line should contain the variable name followed by an equal sign (=) followed by the value to be assigned to that variable. The value can be in integer, real or exponential format regardless of the definition used within the program. For example, an integer variable does not have to be input as an integer. The value will be truncated for use in the program. In addition, the user is not required to list every variable in the namelist. If a variable is not listed in the input file, the program will use the default value. Default values are listed in this section for each variable. Examples of valid inputs are provided for each namelist section.

6.1.1 Variable not in namelist error

If the user lists a variable which is not in one of the namelists, the following system error is generated:

IRIX 6.2 Message

```
namelist read: variable not in namelist
apparent state: unit 35 named test.inp
last format: namelist io
Unit 35 is a sequential unformatted external file
*** Execution Terminated (119) ***
```

Windows95 DOS Shell Message

```
forrtl: severe (19): invalid reference to variable in NAMELIST input, unit 35,
file C:\Lewice\test.inp
```

Common causes for this error occur when the user mistypes the variable name or when the user enters a variable from a previous LEWICE version which is no longer input into that namelist. IRIX 6.2 has an additional constraint that the first character of each namelist and each namelist variable resides in column 2. This constraint was not noticed on the PC. Also note that the LEWICE input files are ASCII text. PCs, Macs and Unix workstations all have different formats for treating line breaks with ASCII files which may cause problems when transferring input files to different platforms. Specifically, when PC ASCII files are moved to an SGI with IRIX 6.2, there is an extraneous character (^M) at the end of each line. This character must be removed from each line to use the file on the SGI.

6.1.2 Namelists Listed Out of Order or Missing Namelist

The namelists used for LEWICE 2.0 are LEW20, DIST, ICE1, and LPRNT in that order. If a namelist appears out of order or is missing, the following system error is generated:

IRIX 6.2 Message

```
namelist read: cannot position within current file
apparent state: unit 35 named test.inp
last format: namelist io
Unit 35 is a sequential unformatted external file
*** Execution Terminated (170) ***
```

Windows95 DOS Shell Message

```
forrtl: severe (24): end-of-file during read, unit 35, file C:\Lewice\test.inp
```

Each section of the main input file will now be explained. Statements which are output to the screen are in *italics*. Statements which contain the phrase (value) indicate that a numerical value will be output where the phrase (value) is located.

6.2 Line 1

Line 1 is the title assigned to the run by the user. The title can be up to 80 characters in length and will be written to the output files "misc.dat" and "junk.dat". If no title exists, the following system error will be generated:

IRIX 6.2 Message

```
namelist read: cannot position within current file
apparent state: unit 35 named test.inp
last format: namelist io
Unit 35 is a sequential unformatted external file
*** Execution Terminated (170) ***
```

Windows95 DOS Shell Message

```
forrtl: severe (24): end-of-file during read, unit 35, file C:\Lewice\test.inp
```

6.3 LEW20 Namelist

The LEW20 namelist contains a collection of inputs from a number of separate namelists from version 1.6[6]. The inputs have been rearranged for clarity. Input for the LEW20 namelist is identified by the line:

&LEW20

This line should immediately follow the title line. Refer to the list of namelist errors at the beginning of this section if there are problems with this input.

LEW20 Namelist Variables

6.3.1 ITIMFL

ITIMFL is a flag indicating whether LEWICE will use automatic time stepping or will use a user-defined number of time steps. If ITIMFL=0 then the number of time steps will remain as input by the user in the IFLO variable. If ITIMFL=1 then the time step will be calculated based on the accumulation parameter. In either case, the time steps are of equal length throughout the run.

When ITIMFL=1, the minimum number of time steps is calculated internally in the program by Equation 1.

$$N = \frac{(LWC)(V)(Time)}{(chord)(\rho_{ice})(0.01)} \quad (1)$$

where

LWC = liquid water content (g/m^3)

V = velocity (m/s)

Time = accretion time (s)

chord= airfoil chord (m)

ρ_{ice}= ice density = 9.17*10^5 g/m^3

When ITIMFL=1 and the number of time steps input by the user is less than the number calculated, the number of time steps will be increased and a warning message will be generated. If the number of time steps input by the user is greater than the value calculated, no adjustment is made and no warning message will be generated. The variability of LEWICE results for various time steps and point spacings is discussed in the section on Numerical Variability in the validation report[1]. Due to this variability, LEWICE 2.0 selects automatic time stepping to be **on** as its default setting. The warning message listed below will be printed whenever the user sets automatic time stepping to the off value:

You are not using automated time stepping procedure. The accuracy of the code in this situation is unknown.

In addition, if the number of time steps selected (see IFLO input) is less than the recommended value, the following additional warning message will print out:

You are running fewer time steps than recommended. Number of time steps recommended = (value). Number of time steps selected = (value). Ice shapes produced may be different from those used to validate this code.

The valid inputs for ITIMFL are 0 and 1. If any other value is input, the following warning occurs:

Valid inputs of itimfl are 0 and 1. Your input value of (value) is out of range. Setting itimfl to 1.

If ITIMFL is not specified in the namelist, its default value will be set to 1 (automatic time stepping is **on**).

6.3.2 TSTOP

TSTOP is the total time of the icing simulation in seconds. This value must be > 0. An input value < 0 will generate the following error:

Severe input error: Time cannot be negative! tstop = (value) is an invalid input.

After checking the other inputs, the program will exit due to this error.

An input of zero for TSTOP will cause a "divide by zero" problem because the calculated time step will also be zero. The following error message will be generated:

Calculated time step is <= zero! dtime = (value)

After checking the other inputs, the program will exit due to this error.

An extensive validation effort has been performed to define the regimes where LEWICE produces acceptable results. The existing experimental data base does not contain any data points for icing times greater than 45 minutes (2700 seconds). If a TSTOP value greater than 2700 seconds is input, the following warning message will be generated:

The maximum icing time used for evaluation of this code was 45 min. Your time of tstop = (value) exceeds this value. The accuracy of the code in this situation is unknown.

The default time used by LEWICE is 60 seconds if the user does not specify a value.

6.3.3 IBOD

IBOD is the number of bodies to be simulated. For example, a three body simulation can consist of a slat, main element and flap. However, multi-body simulations are not limited to this example.Valid input values for IBOD are 1 ≤ IBOD ≤ 5. If the input value for IBOD is ≤ 0, the following error message will be generated:

Severe input error: You must run at least 1 body! ibod = (value)

After checking the other inputs, the program will exit due to this error.

If the input value for IBOD is >5, the following error message will be generated:

Array size limits the number of bodies to 5. Your input value of (value) is out of range. Decrease the number of bodies input or increase the array size to fix this problem. The latter suggestion requires changes to the source code.

After checking the other inputs, the program will exit due to this error. The last statement is intended for code developers. The size of the array can be easily increased to run more than five bodies. This change would require that the code be recompiled.

As stated earlier, LEWICE 2.0 can run multiple body simulations including multi-element airfoils. A report of its capabilities in this region shows very encouraging results[7]. However, much of the development effort for version 2.0 has centered on validating the existing features of the program. Even though the results to date have been encouraging, there is not enough data available to consider LEWICE 2.0 validated for multi-body flows. Therefore, if the input value for IBOD is in the region 2 ≤ IBOD ≤ 5, the following warning message will be generated:

NASA does not have enough data to thoroughly validate LEWICE for multi-body conditions.The built-in flow solver is for incompressible and inviscid (potential) flow. This is usually inadequate for the flow around multi-element wings. It is often necessary to alter the angle of attack and flap angle to analyze these cases. Number of bodies input = (value)

The default number of bodies is equal to 1 if the user does not specify a value.

6.3.4 IFLO

IFLO is the number of time steps to be used in the simulation. A value greater than 0 is required for this input. If the input value is ≤ 0, the following warning message will be generated:

Number of flow solutions input is <= zero! Resetting number of flow solutions (IFLO) = 1. iflo input = (value)

As stated in this message, the program will continue with a single time step. This value may be changed again if the auto-time step flag is on (ITIMFL = 1).

If the automatic time step option is used (ITIMFL = 1), then the number of time steps may be overwritten with the calculated number of time steps. The program will use the value of IFLO input if it is ≥ the calculated value and no warning will be issued. If the number of time steps is less than the recommended value and ITIMFL = 1, the following warning message is generated:

The input number of time steps (5) has been changed to the calculated value of (19). Unless otherwise noted, this occurred because the auto-time stamp itimfl is set = 1.

In the above example, the calculated number of time steps was 19 and the input value was 5. The actual values printed out will depend on the conditions input by the user.

If the automatic time step option is not used (ITIMFL = 0) and the number of time steps input less than the recommended value, the following warning message is generated:

You are running fewer time steps than recommended. Number of time steps recommended = (value). Number of time steps selected = (value). Ice shapes produced may be different from those used to validate this code.

If no time step value is specified by the user, the default number of time steps is set equal to 1 if auto-time stepping is off and will be calculated if auto-time stepping is set on.

Note: For very small (< 6") chord geometries such as cylinders, the number of time steps recommended by the code may be considered prohibitive by the user. It may be possible (and even necessary) to decrease the number of time steps for these cases. It should be noted though that the smallest chord airfoil in the validation database is a 14" NACA0015 for which 8 ice shapes have been digitized. Therefore, results for chord lengths smaller than this have not been validated.

6.3.5 DSMN

DSMN is the minimum size of the control volumes (non-dimensionalized). It is also tied indirectly to the number of panels produced for the flow solution. The exact number of panels and control volumes used will depend on the surface area and complexity of the input geometry. Larger values of DSMN create fewer control volumes and fewer panels while smaller values of DSMN create more control volumes and more panels.

Part of the validation effort for LEWICE 2.0 centered on defining practical ranges for the values of DSMN. The results of these tests are provided in the validation report[1]. For all of the airfoils studied, the range of DSMN values was $\mathbf{2*10^{-4} \leq DSMN \leq 8*10^{-4}}$. The lower limit reflects the current limits of the array sizes in the program and is not a reflection of the accuracy for low DSMN values. For DSMN values of $8*10^{-4}$ or higher, quantitative differences occur due the coarse spacing provided. An exception was found for cylinders which have a very large surface area compared to similarly sized airfoils. However, results on cylinders have not been validated against experimental data. Larger DSMN values are necessary for cylinders due to the limitations on array sizes. The value of DSMN input must be greater than zero however. If a value of DSMN ≤ 0 is input, the following error message is generated:

Severe input error: DSMN must be greater than zero! dsmn =(value)

After checking the other inputs, the program will exit due to this error.

If a value of DSMN greater than $8*10^{-4}$ is input, the following warning message is generated:

The point spacing input is larger than those tested. dsmn(i) = (value). Ice shapes produced may be different from those used to validate this code.

There is no warning message generated for DSMN values below $2*10^{-4}$. However, the user should expect that for values of $2*10^{-4}$ or below, the code may generate an error message that the array size has been exceeded. See the section in this manual on runtime errors for a listing of errors produced when the array sizes have been exceeded. All of the validation tests were run with a DSMN value of $4*10^{-4}$.

The following notes also contain useful information regarding the use of DSMN.

Note: In version 2.0 the number of control volumes will be much greater than the number of panels. The ratio of control volumes to panels based on the validation tests is approximately 50 to 1. The default value for DSMN is $4*10^{-4}$. As the total wrap distance around a typical airfoil is slightly greater than 2 (dimensionless), the default DSMN value will produce over 5000 control volumes. The number of panels produced in this case will be approximately 100.

Note: There is one value of DSMN for each body. If only one body exists, only the first value input is used by the code. For multi-body simulations, it is to the user's advantage to use smaller DSMN values on the smaller bodies and larger values on the larger bodies. Unless otherwise indicated, the user should select values within the appropriate range.

Note: The number of panels and control volumes used are virtually independent of the number of points contained in the geometry input file.

Note: DSMN values larger than $8*10^{-4}$ are sometimes needed when the ice shape is extremely large in comparison to the airfoil. An example of this condition occurs for cylinders below six inches in diameter, especially when a lengthy accretion time is used (20 minutes or more). The time step for this case may have to be increased from the recommended values in order to run this case as well.

6.3.6 NPL

NPL is the number of particle trajectories (including the impingement limit trajectories) which define the collection efficiency distribution. LEWICE needs at least 10 trajectories in order to calculate an accurate collection efficiency curve. If the input value for NPL is less than 10, the following warning message will be generated:

Number of trajectories input (5) is insufficient for this code. Increasing to default value (24).

In this example, the value input for NPL was 5. In addition, LEWICE will issue a warning if the user attempts to run an excessive number of trajectories. This warning message reads:

The number of trajectories input (500) is quite large. Although not a real problem, you can likely reduce NPL without loss of accuracy. Default NPL = 24

In this example, a value of 500 for NPL was input. The warning message will occur for any value of NPL over 50. It should be noted that in this case the value input by the user is not overwritten (the extra trajectories will be performed). All of the validation tests were run with a value of 24 for NPL, which is the default value if none is specified by the user.

Note: The actual number which the code uses for the collection efficiency calculation may be different than the value input. The code is limited to one trajectory strike per panel. If more than one trajectory hits a given panel, only the first hit will be saved. Thus the use of large NPL values may result in unnecessary computations which do not enhance the accuracy of the final result.

6.3.7 RHOP

RHOP is the density of the water particle in kg/m^3. This has been placed in the input file to broaden the utility of this code to industry. Except for very large particle sizes, the physics of water droplet trajectories is the same as for sand particle trajectories. The only required change to model sand particle trajectories is the density of the particle, as sand has different properties than water. If sand density is substituted, the code can be used to predict deposition of sand (sand collection efficiency). In this mode, the ice accretion results should be ignored and the code can be run using a single time step.

This input value must be greater than zero. If a value of RHOP ≤ 0 is input, the following error message will be generated:

Severe input error: Density must be greater than zero! rhop = (value)

After checking the other inputs, the program will exit due to this error.

In addition, LEWICE will warn the user when this input is not equal to the density of water (1000 kg/m^3).

The value input for particle density (value) is different than for water (1000). Most likely, this was done in order to simulate sand particle trajectories. The output for this run should not be used for ice accretion studies. Ice shapes produced may be different from those used to validate this code.

6.3.8 IGRID

IGRID is a flag which allows a grid solution to be used in place of the potential flow code. If IGRID=0, off-body air velocities are determined directly from the potential flow solution. If IGRID=1, the panel solution will not be used. Instead, a grid solution will be read in from files xy.plt and q.plt which are supplied by the user. These files are the grid and flow solution files in PLOT3D format. LEWICE will then interpolate from these points to find the air velocity at the drop location when calculating trajectories. Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

Valid inputs of igrid are 0 and 1. Your input value of (value) is out of range. Setting to 0.

Some cases have been made using a grid solution as input to verify that the routines function as designed. However, this module has not been thoroughly tested and could be buggy. In addition, since only one time step can be used with this option, it has limited use for ice accretion results. Therefore if the user selects IGRID =1, the following warning message is issued:

You are bypassing the potential flow solution to use a grid solution. This option has been tested on exactly two grids: one single body and one multi-body grid. This procedure may still be buggy and is not recommended unless you are willing to customize the code for your use. This option can only be used with a single time step. Setting IFLO = 1

Since the grid read in can only be applicable for the clean airfoil, the number of time steps will be set to 1 if the user has not already done so. The default value for IGRID is 0 if not specified.

6.3.9 IDEICE

IDEICE is a flag which controls a simplistic deicer model. If IDEICE=0 (default), this routine will not be run. If IDEICE=1, then a 1D steady state anti-icer will be run to generate an estimate of the heat required to keep the surface ice free. This solution can then be used as a starting point for using the LEWICE/Thermal[8] or ANTICE[9] models. A value of IDEICE=1 also requires an additional input file called "deicei.inp" which contains inputs needed for the anti-icing module. The information in the deicei.inp input file will be used for each body in a multi-element calculation. Valid input values are 0 and 1. Other input values will produce the following warning message:

Valid inputs of ideice are 0 and 1. Your input value of (value) is out of range. Setting to 0.

In order to distinguish the attributes of this routine versus the more sophisticated programs LEWICE/Thermal[8] and ANTICE[9], the following warning message is generated when IDEICE = 1 is selected:

The anti-icing analysis performed with this option provides only an approximate solution. Surface temperature and heat flux predictions are reasonable. Heater temperature and hot air temperature predictions are much higher than actual. The codes ANTICE and LEWICE/Thermal *should be used for more accurate predictions. These two codes will be integrated into future versions of LEWICE.*

Note: This routine does not affect the ice accretion routine. The code will output an ice shape as if no heat had been applied. This routine generates a separate file containing the temperatures and heat fluxes needed to maintain a desired surface temperature which is input by the user.

Note: This routine treats the current geometry as the airfoil and does not distinguish an iced airfoil from an un-iced airfoil. Therefore, only the results obtained in the first time step are applicable to an anti-icing problem.

&END

This line concludes the section for the LEW20 namelist. The following table lists an example input for this namelist.

Table 1: Example LEW20 Namelist

```
&LEW20
ITIMFL =    1
TSTOP  =  300.
IBOD   =    1
IFLO   =    5
DSMN   =    4.D-4
NPL    =    24
RHOP   =  1000.
IGRID  =    0
IDEICE =    0
&END
```

6.4 DIST Namelist

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

&DIST

This line identifies the start of this namelist section. Refer to the list of namelist errors at the beginning of this section if there are problems with this input.

6.4.1 FLWC

FLWC is the volume fraction of the total liquid water content contained in each drop size. The sum of these values must equal one. If the sum of the FLWC values is not one, the following warning message is generated:

The flwc values are the Fractional Liquid Water Content attributed to each droplet size. These values must add to one (1). Your input of (value) does not add up to 1. The program will adjust your FLWC values proportionately so they add to 1.

If the sum of the FLWC values is greater than zero, the individual values input will be increased/decreased so that their sum equals one. If this sum is ≤ 0, then the water mass will be equally distributed for each of the drop sizes input (FLWC = 1/[number of drop sizes input]).

The program will determine the number of drop sizes in the distribution by looking for the first occurrence where FLWC = 0. Therefore, the user should not place zeros until the end of the distribution is reached.

Table 2: Example of bad input for FLWC

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

Table 3: Example of correct input for FLWC

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

6.4.2 DPD

DPD is the size, in microns, of the water drops. If only one size is input, it is the MVD (median volume droplet). MVD is not an input variable to LEWICE. The MVD is calculated from the individual drop sizes input in this section. The individual drop sizes and the calculated MVD must both be greater than zero. An input drop size less than zero will generate the following error message:

Severe input error: Drop size must be > 0 ! DPD = (value) for bin number (value).

After checking the other inputs, the program will exit due to this error.

A calculated MVD less than zero will generate the following additional error message:

Severe input error: MVD drop size <= 0! MVD = (value)

In addition, LEWICE will generate warning messages if the drop size input is outside of the range in the validation database. If the MVD drop size is below 15 microns, the following warning message is generated:

Your MVD value of (value) is below 15 microns. No validation data is available. The accuracy of the code in this situation is unknown.

If the MVD drop size is greater than 270 microns, the following warning message is generated:

The MVD of your drop size distribution exceeds 270 microns. No validation data is available for your drop size of (value) microns. The accuracy of the code in this situation is unknown.

Due to the recent popularity of drop size inputs outside the FAA certification envelope, it is worth emphasizing the above warning message. This statement does **not** imply that LEWICE cannot run the drop size distribution input. It most likely can. The warning statement does **not** imply that the results will be inaccurate. LEWICE results for exceedence conditions are quite encouraging in this respect. The statement simply points out that limited experimental data is yet available at these drop sizes. Since the results cannot be experimentally validated, the true accuracy of the results cannot be verified. The following plot shows the range of drop sizes in the validation database versus liquid water content.

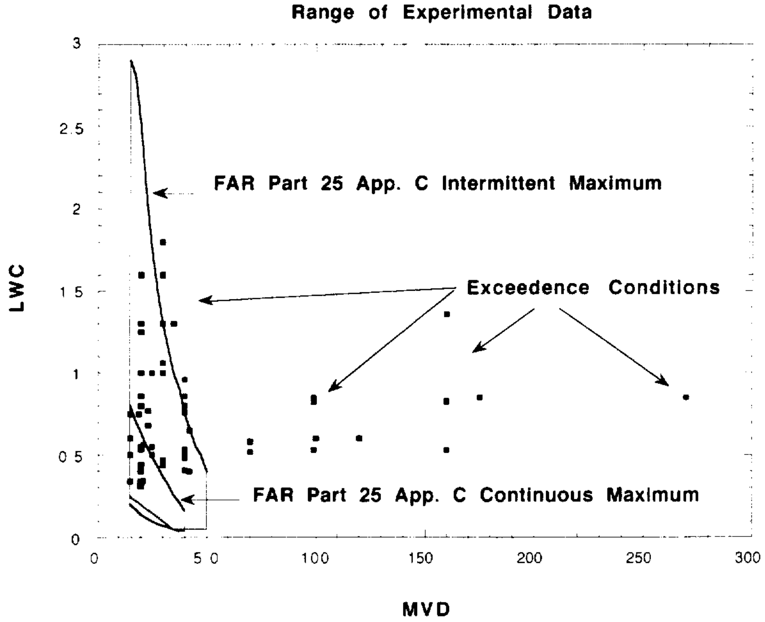

Figure 1: Test Conditions in Database Used for Code Validation

If the MVD drop size is greater than 50 microns, the following warning message is generated:

Your MVD of (value) exceeds the FAA intermittent maximum drop size of 50 microns. Although some data has been collected in this regime and used for code validation, there is not enough data available to consider the code validated for this drop size. The accuracy of the code in this situation is unknown.

Once again, consult the earlier statement concerning the implications of running exceedence conditions. An analysis of the capabilities of LEWICE in this regime can also be found in the proceedings of an FAA conference in 1996[10].

&END

This line concludes the section for the DIST namelist. The following table lists an example input for this namelist.

Table 4: Example DIST Namelist

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

Note: The example provided is a Langmuir 'D' drop size distribution, with an MVD of 20 μm.

Note: If the user does not specify input values, the default value for this namelist is a mono-dispersed drop size of 20 μm.

6.5 ICE1 Namelist

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

&ICE1

This line identifies the start of this namelist section. Refer to the list of namelist errors at the beginning of this section if there are problems with this input.

6.5.1 CHORD

CHORD is the distance from the leading edge to the trailing edge in meters. For a cylinder, this represents the cylinder diameter. For airfoils, it is the standard chord length. For a multi-body simulation, CHORD represents the reference length used to nondimensionalize the coordinates input. A typical value used for multi-element airfoils is the length of the airfoil in the stowed position. The input must be greater than zero. For input values of CHORD ≤ 0, the following error message is generated:

Severe input error: Chord must be greater than zero! Chord = (value)

After checking the other inputs, the program will exit due to this error.

If no value is input for chord, its default value is 0.9144 m (36"). The range of chord lengths in the experimental database is 13.9" to 78".

6.5.2 AOA

This is the angle of the body(s) as input with respect to the flow in degrees. There are no error messages associated with this input. If the angle of attack is greater than 6° (or less than -6°), the following warning message will be generated:

Angle of attack value of (value) may incur separation once ice forms. This may not be modeled well by the code. The accuracy of the code in this situation is unknown.

Potential flow cannot model stall or post-stall behavior. The user should also note that in the validation test procedure, the angle of attack input into the code was sometimes different from the actual angle of attack value. This difference was made to compensate for the difference in predicted lift using a potential flow code and the actual lift of the clean airfoil.

If no value of AOA is supplied by the user, the value will be set to 0 degrees. The range of angle of attack values in the validation database is -4 to +7 degrees.

6.5.3 VINF

VINF is the ambient velocity (the flight speed) in m/s.

Note: Knots * 0.51481 = m/s; MPH * 0.447 = m/s

The input value for VINF must be greater than zero. If a value of VINF ≤ 0 is input, the following error message will be generated:

Severe input error: Velocity must be greater than zero! vinf = (value)

After checking the other inputs, the program will exit due to this error.

An upper limit on Mach number exists within the code. If the ambient Mach number is ≥ 1, the following error message is generated:

Mach number is => 1! Cannot calculate supersonic flow with this code! Mach No. = (value)

After checking the other inputs, the program will exit due to this error.

In addition, high subsonic Mach numbers have not been validated against experimental data. Problems may exist due to the limitations of potential flow. The following warning message is generated if the ambient Mach number exceeds 0.45 (the highest value in the validation database).

High mach number of (value) may not be modeled well by the code.The accuracy of the code in this situation is unknown.

If no value of VINF is supplied by the user, its default value is 90 m/s. The range of velocity values in the validation database was 56 m/s to 146 m/s. In terms of Reynolds number, the range of data was $2.26*10^6$ to $1.3*10^7$. In terms of Mach number, the range of data was 0.17 to 0.45.

6.5.4 LWC

LWC is the liquid water content of the air in g/m^3. This value must be ≥ 0. If a negative value is input, the following error message will be generated:

Severe input error: LWC cannot be negative! lwc = (value)

After checking the other inputs, the program will exit due to this error.

In addition, if the LWC value input is greater than 2 g/m^3, the following warning message will be generated:

There is no data available to validate the code for LWC values this high. lwc = (value). The accuracy of the code in this situation is unknown.

The user should also consult the statement concerning the use of exceedence conditions listed in the description of the drop size input.

If no value of LWC is input by the user, its default value is 0.54 g/m^3. The range of values for Liquid Water Content in the validation database was 0.31 to 1.8 g/m^3. See Figure 1 for a plot of LWC versus MVD for the validation database test points.

6.5.5 TINF

TINF is the ambient *static* temperature in degrees Kelvin. This input value must be greater than zero. A value of TINF ≤ 0 will generate the following error message:

Temperature input is <= zero! Make sure temperature is in degrees Kelvin. tinf = (value)

After checking the other inputs, the program will exit due to this error.

In addition, warning messages are generated when the static temperature is outside the normal icing regime. If the input value of TINF is less than 240 K, the following warning message is generated:

It is unlikely that supercooled droplets exist below 240 Kelvin. tinf = (value) Make sure your input value is in degrees Kelvin. The accuracy of the code in this situation is unknown.

If the input value of TINF is greater than 273.15 K, the following warning message is generated:

No ice will form at above freezing temperatures! Is this what you want? tinf = (value)

An example of a case where the use of above freezing temperatures is warranted would be to match experimental data on droplet collection efficiency which is taken at above freezing temperatures.

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

$$T_s = T_o - \frac{V_\infty^2}{2c_p} \text{ where}$$

T_s = static temperature, K

T_o = total temperature, K

V_∞ = velocity, m/s

c_p = specific heat of air, J/kg/K

If no value of TINF is input by the user, its default value is 268.15 K. The range of values in the validation database was 241.3 K to 270.2 K.

6.5.6 PINF

PINF is the ambient static pressure in Pascals (N/m^2). The input value for PINF must be greater than zero. An input value of PINF ≤ 0 will generate the following error message:

Severe input error: Pressure input is <= zero! pinf = (value)

After checking the other inputs, the program will exit due to this error.

No other warning messages or error messages are generated for this input variable. However, it should be noted that all of the validation tests used an input of 10^5 N/m^2.

Note: Ambient pressure is not recorded as part of the tunnel data, so the exact value during the tests is unknown. However, since ambient pressure is at best a secondary effect on the ice accretion process and since the IRT is not a pressurized tunnel, a representative value near atmospheric pressure was used for the comparison.

Note: To a good approximation, for a 'standard atmosphere', the following equation can be used:

$$P = 100920 - 11.35H + 0.00039456H^2 \text{ where}$$

P = pressure in N/m^2 and

H=height (altitude) in meters

Note: $lb_f/in^2 * 6894.7 = N/m^2$; atmospheres * 101330 = N/m^2; in. Hg * 3386.4 = N/m^2

If no value of PINF is input by the user, its default value is 10^5 N/m^2.

6.5.7 RH

RH is the relative humidity and is input in percent relative humidity. This input value is normally assumed to be 100%, unless the actual value is known. Relative humidity is not recorded as part of the tunnel data, so the exact value during the tests is unknown. However, since relative humidity is at best a secondary effect on the ice accretion process, a value of 100% can be assumed. The value of relative humidity must be in the range 0% ≤ RH ≤ 100%. Input values outside this range will produce the following error message:

Relative humidity must be between 0% and 100%. Your input of (value) is outside this range.

After checking the other inputs, the program will exit due to this error.

If no value of relative humidity is input by the user, its default value is 100%.

&END

This line concludes the section for the ICE1 namelist. The following table lists an example input for this namelist.

Table 5: Example ICE1 Namelist

```
&ICE1
CHORD   = 0.9144
AOA     = 0.0
VINF    = 89.5
LWC     = 0.34
TINF    = 269.
PINF    = 100000.0
RH      = 100.0
&END
```

6.6 LPRNT Namelist

The LPRNT namelist controls output file print options for LEWICE 2.0. Users can limit the amount of information sent to each printout file which saves disk space and computation time.

&LPRNT

This line identifies the start of this namelist section. Refer to the list of namelist errors at the beginning of this section if there are problems with this input.

6.6.1 FPRT

If FPRT=0, the flow solution output (flow.dat and pres.dat) will not be written. If FPRT=1, every 10th control volume will be written to reduce the size of the output. If FPRT = 2, every control volume will be output. Valid inputs for FPRT are 0, 1 and 2. Other input values will produce the following warning message:

Valid inputs of fprt are 0, 1, and 2. Your input value of (value) is out of range. Setting to 1.

If no value of FPRT is input by the user, its value will default to 1.

6.6.2 HPRT

If HPRT=0, the heat transfer coefficients (htc.dat) will not be written. If HPRT=1, every 10th control volume will be written to reduce the size of the output. If HPRT = 2, every control volume will be output. Valid inputs for HPRT are 0, 1 and 2. Other input values will produce the following warning message:

Valid inputs of hprt are 0, 1, and 2. Your input value of (value) is out of range. Setting to 1.

If no value of HPRT is input by the user, its value will default to 1.

6.6.3 BPRT

If BPRT=0, the collection efficiencies will not be written. If BPRT=1, they will be written. Collection efficiencies are only written for the panel geometry, not for the control volume geometry. There is no need for a reduced input flag.Valid inputs for BPRT are 0 and 1. Other input values will produce the following warning message:

Valid inputs of bprt are 0 and 1. Your input value of (value) is out of range. Setting to 1.

If no value of BPRT is input by the user, its value will default to 1.

6.6.4 EPRT

If EPRT=0, the energy balance output (temp.dat, qener.dat, xkinit2.dat) will not be written. If EPRT=1, every 10th control volume will be written to reduce the size of the output. If EPRT = 2, every control volume will be output. Valid inputs for EPRT are 0, 1 and 2. Other input values will produce the following warning message:

Valid inputs of eprt are 0, 1, and 2. Your input value of (value) is out of range. Setting to 0.

If no value of EPRT is input by the user, its value will default to 0. Note that the default state results in no printout to these files.

6.6.5 MPRT

If MPRT=0, the mass balance output (mass.dat, fract.dat, dyice.dat, dens.dat) will not be written. If MPRT=1, every 10th control volume will be written to reduce the size of the output. If

MPRT = 2, every control volume will be output. Valid inputs for MPRT are 0, 1 and 2. Other input values will produce the following warning message:

Valid inputs of mprt are 0, 1, and 2. Your input value of (value) is out of range. Setting to 0.

If no value of EPRT is input by the user, its value will default to 0. Note that the default state results in no data written to these files.

6.6.6 TPRT

If TPRT=0, the x,y coordinates of individual droplet trajectories will not be written (traj1.dat, traj2.dat, traj3.dat, traj4.dat, traj5.dat). If TPRT=1, only trajectories used for the collection efficiency calculation will be written. If TPRT=2, all trajectories will be written. Valid inputs for TPRT are 0, 1 and 2. Other input values will produce the following warning message:

Valid inputs of tprt are 0, 1, and 2. Your input value of (value) is out of range. Setting to 0.

If no value of TPRT is input by the user, its value will default to 0. Note that the default state results in no data written to these files.

Note: The trajectories calculated for a given body are all written sequentially to the file. If only one body exists, only traj1.dat will be created.

Note: The definition of this flag has been reversed from version 1.6!

6.6.7 IDBF

If IDBF=0, debug information will not be written. If IDBF=1, debug info will be written to the screen and to the file "junk.dat".Valid inputs for IDBF are 0 and 1. Other input values will produce the following warning message:

Valid inputs of idbf are 0 and 1. Your input value of (value) is out of range. Setting to 0.

If no value of IDBF is input by the user, its value will default to 0. Note that the default state results in only limited data written to this file.

&END

This line concludes the section for the LPRNT namelist. The following table lists an example input for this namelist.

Table 6: Example LPRNT Namelist

```
&LPRNT
FPRT    =1
HPRT    =1
BPRT    =1
EPRT    =0
MPRT    =0
TPRT    =0
IDBF    =0
&END
```

6.7 Complete Example Case Input File

This concludes the section describing the variables in the main input file. The following table lists an example input for this file.

Table 7: Example Test Input File

```
Example 1
&LEW20
ITIMFL =    1
TSTOP  =  300.
IBOD   =    1
IFLO   =    5
DSMN   =    4.D-4
NPL    =    24
RHOP   =  1000.
IGRID  =    0
IDEICE =    0
&END
&DIST
FLWC   = 0.05, 0.1, 0.2, 0.3, 0.2, 0.1, 0.05, 0.0, 0.0, 0.0
DPD   = 6.2, 10.4, 14.2, 20.0, 27.4, 34.8, 44.4, 0.0, 0.0, 0.0
&END
&ICE1
CHORD  = 0.9144
AOA    = 0.0
VINF   = 89.5
LWC    = 0.34
TINF   = 269.
PINF   = 100000.0
RH     = 100.0
&END
&LPRNT
FPRT   =1
HPRT   =1
BPRT   =1
EPRT   =0
MPRT   =0
TPRT   =0
IDBF   =0
&END
```

6.8 Extent of Data in Validation Database

This section will summarize the range of experimental data used to validate LEWICE 2.0. LEWICE will most likely produce results outside of these ranges. However, since no experimental data is available for validation, the true accuracy of the results cannot be verified. This statement does **not** imply that LEWICE cannot run the data input. It most likely can. The warning statement(s) generated do **not** imply that the results will be inaccurate. The statement simply points out that no experimental data is yet available.

Table 8: Range of Experimental Data for Validation

Variable	Range
Time:	2 min. to 45 min.
Chord:	13.9" to 78"
AOA:	-4° to 7°
Velocity:	56 m/s to 146 m/s
Reynolds Number:	$2.26*10^6$ to $1.3*10^7$
Mach Number:	0.17 to 0.45
LWC:	0.31 g/m^3 to 1.8 g/m^3
MVD:	15 μm to 270 μm
Temperature (static):	-25.3°F to 26.7°F
Temperature (total):	-15°F to 33°F

Chapter 7: Body Geometry Input

This section will describe the proper format for the input to the geometry file(s). It will also describe warning messages and error messages which could result from an improperly formatted file. It should be noted that all of the validation data uses airfoils. Although LEWICE can simulate any enclosed body (or bodies), the validation performed to date has been limited to the available data[1].

In the interactive input to the program, LEWICE 2.0 will prompt the user for the file name(s) of the geometry input file(s). A separate input file must be provided for each body being simulated. If only one body is simulated, only one geometry file will be read in. Each line of the geometry input file contains an x,y coordinate pair for the body geometry. The x-coordinate is listed first. The format of the data is free-format for the x,y coordinates. A sample body input geometry is listed in Table 9 below.

Table 9: Example Body Geometry (NACA 23014)

1.0006	-0.0025295
0.98460	-0.0049304
0.96895	-0.0072317
0.95325	-0.0094792
0.93745	-0.011684
0.92169	-0.013835
0.90590	-0.015936
0.89007	-0.017990
0.87424	-0.019997
0.85838	-0.021958
0.84252	-0.023873
0.82664	-0.025742
0.81073	-0.027568
0.79485	-0.029346
0.77894	-0.031082
0.76300	-0.032776
0.74707	-0.034426
0.73111	-0.036035
0.71518	-0.037598

0.69924	-0.039116
0.68326	-0.040598
0.66732	-0.042033
0.65137	-0.043424
0.63540	-0.044773
0.61946	-0.046076
0.60351	-0.047332
0.58753	-0.048547
0.57164	-0.049710
0.55575	-0.050817
0.53965	-0.051897
0.52374	-0.052927
0.50808	-0.053878
0.49217	-0.054777
0.47619	-0.055636
0.46031	-0.056440
0.44448	-0.057185
0.42891	-0.057849
0.41336	-0.058424
0.39764	-0.058929
0.38194	-0.059367
0.36661	-0.059710
0.35146	-0.059932
0.33633	-0.060041
0.32141	-0.060016
0.30661	-0.059853
0.29100	-0.059549
0.27610	-0.059218
0.26159	-0.058688
0.24626	-0.058063
0.23351	-0.057365
0.21934	-0.056303
0.20485	-0.055197
0.19167	-0.053961
0.17837	-0.052466
0.16365	-0.050700

0.15032	-0.049074
0.13639	-0.047096
0.12273	-0.045124
0.10983	-0.043093
0.10580	-0.042443
0.10062	-0.041576
0.094406	-0.040509
0.088393	-0.039431
0.081019	-0.038064
0.072426	-0.036360
0.065461	-0.034918
0.057876	-0.033223
0.048544	-0.030964
0.041900	-0.029167
0.035180	-0.027125
0.028754	-0.024882
0.022044	-0.022105
0.016585	-0.019234
0.013220	-0.017132
0.010646	-0.015314
0.0083173	-0.013440
0.0059869	-0.011249
0.0037424	-0.0085095
0.0025648	-0.0066507
0.0014672	-0.0044964
0.00058515	-0.0022445
0.00022853	-0.0010844
3.0568e-05	-0.00027511
-0.00015575	0.00062445
-0.00039301	0.0021397
-0.00053857	0.0042402
-0.00046579	0.0061892
-0.00021252	0.0082460
0.00014993	0.0099811
0.00083843	0.012354
0.0018108	0.014868

0.0029083	0.017135
0.0048311	0.020258
0.0067977	0.022936
0.0087496	0.025330
0.012939	0.029991
0.015252	0.032287
0.017814	0.034591
0.020953	0.037204
0.023386	0.039080
0.026571	0.041418
0.030689	0.044188
0.033220	0.045771
0.036504	0.047745
0.038999	0.049176
0.041659	0.050643
0.044448	0.052116
0.047613	0.053697
0.050539	0.055095
0.053291	0.056346
0.056412	0.057696
0.058997	0.058773
0.061712	0.059879
0.064857	0.061087
0.066489	0.061718
0.073202	0.063984
0.080154	0.066505
0.088125	0.068774
0.099735	0.071726
0.11462	0.074895
0.12940	0.077445
0.14454	0.079325
0.15802	0.080718
0.18655	0.083164
0.20491	0.084047
0.22594	0.084694
0.24642	0.085044

0.26546	0.085239
0.28616	0.085354
0.30437	0.084924
0.32198	0.084534
0.34505	0.083789
0.36169	0.083010
0.37983	0.082035
0.40332	0.080534
0.42432	0.079198
0.45166	0.076774
0.48189	0.074390
0.51205	0.071520
0.53792	0.068763
0.56294	0.065910
0.58539	0.062993
0.63827	0.056776
0.67977	0.051450
0.70802	0.047307
0.72582	0.044603
0.74476	0.041645
0.76651	0.038213
0.78926	0.034527
0.80617	0.031748
0.82429	0.028683
0.84640	0.024814
0.86233	0.022217
0.87581	0.020070
0.88985	0.017608
0.90557	0.014849
0.92131	0.012070
0.93709	0.0092346
0.95298	0.0063447
0.96881	0.0034328
0.98460	0.00048670
1.0006	-0.0025295

It is quite common for problems to arise when inputting a new geometry for the first time. The following discussion will describe some of the common errors made by users in generating an input file. Error messages and warning messages printed to the screen will also be covered. As mentioned in the previous section, all warning and error messages are also written to the debug file "junk.dat".

7.1 Number of Points

An empty (blank) file will produce the following error:

Severe input error: Number of points must be greater than zero!

The code will stop because of the above error.

After checking the other inputs, the program will exit due to this error. A file which does not exist at run time (missing file) should generate a system error. That error is listed in Chapter 5: Interactive Input.

7.2 Blank Lines

Blank lines in the geometry file should be avoided. On an SGI workstation running IRIX 6.2, blank lines are ignored by the code. Other systems may read blank lines as an additional data point of x=0, y=0. The code is set to check for this occurrence at the end of the file. If blank lines exist at the end of the file, the following warning message may print out:

Removing blank line from geometry input file.

7.3 Too Few Points

If the body geometry is too coarse, the panel model created may not replicate the body geometry input. The initial set of coordinates output to file "ice1.dat" contain the initial panel model of the body geometry. The user should check that this shape matches their input file for any new geometry or if the user increases the point spacing (DSMN). If the number of points input is less than 30, the following warning message will be generated:

The number of geometry points input (5) is less than 30. A coarse geometry such as this may not run well.

In the example above, a input data file consisting of five points was used.

7.4 Too Many Points

The only true upper limit to the number of points which the user can input is 10000, which is the internal array size. Standard geometry input files used for testing purposes range from 50 to 150 points. The program will print the following warning if more than 1000 points are input:

The number of geometry points input (2000) is very high. You probably do not need this many points.

In this example, the number of points input was 2000.

7.5 Body Not Closed

The panel solution used in this code assumes that the body(s) being simulated are closed bodies. Several tests have been run using airfoils with open trailing edges and most of the results appear acceptable. The following warning message is generated when the program detects an unclosed body:

Warning! The trailing edge is not closed. Check the flow solution carefully.

If the flow solution calculated with an open trailing edge is acceptable to the user, then there may be no need to alter the trailing edge simply to enclose the body.

7.6 Small Point Spacing

LEWICE may have problems with input points which are very close together or exactly the same (duplicate point). This can easily occur when points are typed into the computer or when two airfoil segments are joined together. LEWICE 2.0 will automatically correct this problem and generate the following warning message:

Two of the points on the geometry are too close together. Removing point (value) from body (value).

This statement informs the user that one of the input points has been removed.

7.7 Body Input Backwards

LEWICE requires that the body geometry points should be input in a clockwise fashion. This means that the points are input starting at the trailing edge and proceed sequentially toward the leading edge along the lower surface up to the leading edge, then traverse back to the trailing edge along the upper surface. LEWICE 2.0 will automatically correct for the case where the geometry is input counterclockwise. If LEWICE detects this situation, the following warning message will be generated:

Body points have been input counterclockwise. The program needs clockwise points. Reversing points.

7.8 Large Angles Between Neighboring Segments

Several errors can occur when points are typed in. These errors may cause the geometry to be different from the one intended by the user. Some common errors include: points input in reverse order; missing or misplaced decimal points; or mistyped numbers. LEWICE cannot check for all possible errors in the file. The user should always check the first panel geometry printed to "ice1.dat" to ensure that the body being used by LEWICE closely resembles the intended geometry. A common consequence of geometry input errors are large angles between neighboring segments. LEWICE will print two warning messages to aid the user in identifying potential problems. The following error will print out if this angle is greater than 45°:

You have two segments which form an angle greater than 45 degrees. Check the panel geometry and flow solution thoroughly. Check input point number (value) on body (value).

Figure 2 provides a diagram showing a 45 degree panel angle.

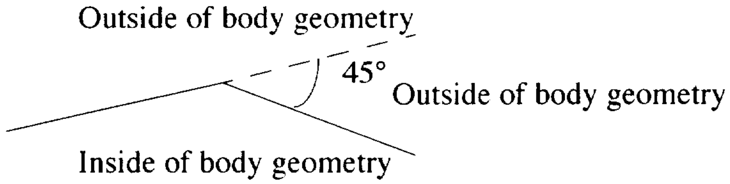

Figure 2: Example of a 45 degree panel angle

LEWICE does not correct for this occurrence since the large angle may be intentional on the part of the user. If this angle is intentional, the user should ensure that there are sufficient input points in regions of high curvature. The point distribution methodology used in LEWICE will tend to "round off" corners if an insufficient number of points are used. A more severe warning is issued by LEWICE if the segment angle exceeds 135°:

You have two segments which form an angle greater than 135 degrees. Check the panel geometry and flow solution thoroughly. Also check that points have been entered correctly. Check point number (value) on body (value).

Figure 3 shows an example of a 135° panel angle.

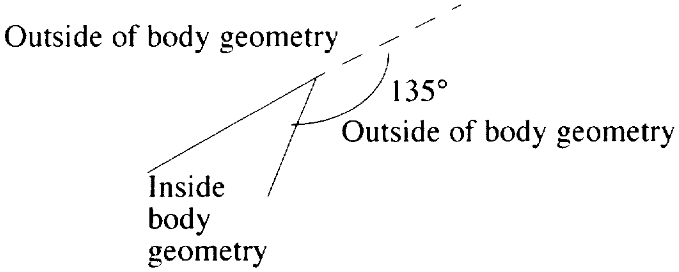

Figure 3: Example of a 135 degree panel angle

In this case, it is unlikely that the user intended to have this large of a segment angle. LEWICE will not make any corrections to the input as it is possible that this large angle was also intentional on the part of the user. If this large angle is intentional on the part of the user, it is even more important that the user supply sufficient input points in the region around this angle so that it does not get rounded off. The flow solution and ice accretion results should also be thoroughly checked to ensure that an acceptable result was obtained.

7.9 Intersecting Bodies

Additional input problems may arise when the user attempts to input more than one body. One such problem can occur when the bodies intersect. This can easily occur with multi-element airfoils if the user does not properly rotate the flap or use the proper gap settings. LEWICE cannot correct for this problem. The following error message will be generated:

Bodies (value) and (value) intersect. Program cannot run.

After performing the remaining checks on the geometry, the program will exit due to this error.

7.10 Concentric Bodies

LEWICE cannot run multiple bodies where one body is completely inside another body. This can occur if the coordinates for the bodies are supplied relative to different points of origin rather than relative to the same point of origin. LEWICE cannot correct for this error. If this situation is found, the following error message will be generated:

Body (value) is inside body (value).LEWICE cannot run this case.

After performing the remaining checks on the geometry, the program will exit due to this error.

7.11 Bodies Out of Order

The logic used by the trajectory module dictates that multiple bodies need to be input in sequential order in the x-direction. This means that the first body a particle could encounter must be listed first, the second body it could encounter must be listed second and so on. This criteria is based upon the leading edge of each body, not on the trailing edge as particles are most likely to impinge on the leading edge of each body. LEWICE will correct for this problem and automatically put the bodies in the proper order. The following warning message will be generated:

Bodies (value) and (value) are out of order. Bodies must be input so that the first body will be the first to be hit by drops; the second body must be the second to be hit and so on. Putting bodies in correct order.

If the program finds any correctable problems with the geometry file, the corrected geometry will be written to file "fixed.dat".

Chapter 8: Optional Input Files

A case submitted by the user containing default values for each input in the main input file will not read additional input files. The user can choose two flags in the main input file which will require the input of an additional input file. An input value of 1 for variable IDEICE will read in the file "deicei.inp" while an input value of 1 for variable IGRID will read in the files "xy.plt" and "q.plt". The formats for these additional input files will now be discussed in detail.

8.1 Deice Input File (deicei.inp)

LEWICE 2.0 can perform four different models of a thermal anti-icer. The information in the deicei.inp input file will be used for each body in a multi-element calculation. The four models are an evaporative hot air system, an evaporative electrothermal system, a running wet hot air system and a running wet electrothermal system. This section will describe the variables input for this analysis.

8.1.1 Deicer Input File Name

The name of the input file (deicei.inp) is fixed and cannot be changed without altering the program. The file must also reside in the directory where the user is running LEWICE. If this file does not exist, the following system error message will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 19 named
last format:
Unit 19 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: severe (24): end-of-file during read, unit 19, file
C:\Lewice\deicei.inp
```

The user should verify that the file "deicei.inp" is in the proper directory.

8.1.2 TSURF

This is the temperature at the airfoil surface which the user wants to maintain. Typical running wet anti-icing systems operate in the region 5°C-10°C while an evaporative system may operate at 50°C or even higher. The number must be listed in columns 51-58. The text accompanying this input field in the examples listed is provided for informational purposes. Only the numerical value is read by LEWICE. If the numerical value does not reside in this column range, a value of TSURF = 0 will be read which will generate the following error:

Surface temperature in deice input is <= zero! Program cannot continue. tsurf = 0.

If the user places non-numerical values within columns 51-58, the following system errors will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (50x,f7.3)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

The input value for TSURF must be greater than zero. If a value of TSURF ≤ 0 is input, the following error message is generated:

Surface temperature in deice input is <= zero! Program cannot continue. tsurf = (value).

After checking the rest of this input file, the program will stop.

In addition, the program assumes that the user wants to perform an anti-icing analysis. The desired surface temperature should therefore be above freezing. If a value of TSURF ≤ 273.15 is entered, the following warning message is generated:

Surface temperature in deice input is below freezing. tsurf = (value). Is this what you want?

Note: The results for a desired surface temperature below freezing will be inaccurate since the anti-icing module does not have provisions to freeze surface water as does the LEWICE icing module.

8.1.3 IEVAP

This flag indicates if the system is running wet (IEVAP = 0) or evaporative (IEVAP = 1). The input is read from column 51.

Important Note: If the value entered on this line is not located in column 51, then the program will read a value of 0, even if this was not the intended value.

If the user places a non-numerical value in column 51, the following system errors will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (50x,i1)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

Valid input values for IEVAP are 0 and 1. If the value input is outside this range, the following warning message will be generated:

Evaporation flag out of range. ievap = (value). Setting ievap = 0

8.1.4 ITHERM

This flag indicates whether the anti-icer is electrothermal (ITHERM=0) or uses hot air (ITHERM=1). Input is read from column 51.

Important Note: If the value entered on this line is not located in column 51, then the program will read a value of 0, even if this was not the intended value.

If the user places a nonnumerical value in column 51, the following system errors will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (50x,i1)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

Valid input values for ITHERM are 0 and 1. If the value input is outside this range, the following warning message will be generated:

Deicer flag out of range. itherm = (value). Setting itherm = 0

8.1.5 Number of Layers

This variable is the number of different materials in the direction normal to the airfoil surface. The input is a 2-digit number starting in column 21. The limit is 50 layers. If a value outside the range 1 ≤ # Layers ≤ 50 is input (or if no value is input in these columns), the error message will read:

Number of layers input is out of range. layer = (value). Program cannot continue.

After checking the rest of this input file, the program will stop.

If the user places non-numerical values in one of these columns, the following system errors will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (20x,i2//)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

8.1.6 Length

This input represents the thickness of the material in meters. The data is read from columns 3-17 and should be in exponential format.

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

This input value must be > 0. If a value is input outside this range (or if no value is input in these columns), the following error message will be generated:

Thickness of layer in deice input file is <= zero. The program cannot continue. dy(i) = (value) in layer i = (value)

After checking the rest of this input file, the program will stop.

If the user places non-numerical values within these columns, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (2(2x,e15.7))
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

8.1.7 Conductivity

This input represents the thermal conductivity of each material in W/mK. The data is read from columns 20-34 and should be in exponential format.

This input value must be > 0. If a value is input outside this range (or if no value is input in these columns), the following error message will be generated:

Thermal conductivity of layer in deice input file is <= zero. The program cannot continue. ak(i) = (value) in layer i = (value)

After checking the rest of this input file, the program will stop.

If the user places non-numerical values within these columns, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (2(2x,e15.7))
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

8.1.8 LHEAT

This is the layer in which an electrothermal heater is located. For a hot-air system, the input is not used. The input is read from column 51. Valid input values are 1 ≤ LHEAT ≤ LAYERS where LAYERS is the total number of layers input. If the value input is outside this range (or if no value is input in this column), then the following error message will be generated:

Heater layer input is out of range. lheat = (value). Program cannot continue.

After checking the rest of this input file, the program will stop.

If the user places non-numerical values within this column, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (50x,i1)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

8.1.9 Number of Heat Transfer Coefficients

This input value tells the code how many heat transfer coefficients are to be read in. These heat transfer coefficient values are the *interior* heat transfer coefficients, and are only used when modeling a hot-air design. External heat transfer coefficients are predicted by LEWICE 2.0. The value for the number of heat transfer coefficients is read from columns 51-54. This input value must be ≥1. If the value input is outside this range (or if no value is input in this column), then the following error message will be generated:

Number of HTC values to input is out of range. nheat = (value). Program cannot continue.

After checking the rest of this input file, the program will stop.

If the user places non-numerical values within these columns, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (50x,i4)
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

Since the format for this variable allows only four spaces for the input, the maximum value for NHEAT is 9999.

8.1.10 S/C

S/C is the wrap distance at which the heat transfer coefficient is specified and is measured from the leading edge. The expected values for this input are non-dimensionalized by the chord. The data is read from columns 3-17 and should be input in exponential format. If the data is not placed in these columns, the wrap distance will be read as zero. In addition, the heat transfer coefficient for regions outside those specified by the user will be zero. If the user specifies values for regions outside the wrap distance on the body, the following warning message will be generated:

Wrap distance input is out of range. st(i) = (value) at location i = (value). Will be set to actual range of st(i) = (value)

For example, if the wrap distance on the body has a range of -1.012 ≤ S/C ≤ 1.012 and a heat transfer coefficient is specified at a wrap location of S/C = -1.05, the code will use the actual limit of S/C = -1.012.

If the user places non-numerical values within these columns, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (2(2x,e15.7))
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

8.1.11 HTC

HTC is the *interior* heat transfer coefficient in W/m^{2}K and should not be confused with the LEWICE 2.0 calculated external heat transfer coefficient. The data is read from columns 20-34 and should be in exponential format. The heat transfer coefficient for wrap distance regions outside those specified by the user will be zero. The input values for HTC must be ≥ 0. Negative values will generate the following error message:

Heat transfer coefficient input is out of range htc(i) = (value) at location i = (value). Program will stop because of this error.

After checking the rest of this input file, the program will stop.

If the user places non-numerical values within these columns, the following system error will be generated:

IRIX 6.2 Message

```
fmt: read unexpected character
apparent state: unit 19 named deicei.inp
last format: (2(2x,e15.7))
Unit 19 is a sequential formatted external file
*** Execution Terminated (115) ***
```

Windows95 DOS Shell Message

forrtl: severe (64): input conversion error, unit 19, file C:\Lewice\deicei.inp

This completes the section on the deicer input file. An example input for this input file is given in Table 10.

Table 10: Example Deicer Input (deicei.inp)

```
DESIRED SURFACE TEMPERATURE              TSURF = 320.000  K
=1 FOR EVAPORATIVE, =0 FOR RUNNING WET   IEVAP = 1
=0 FOR ELECTROTHERMAL, =1 FOR HOT AIR   ITHERM = 1
NUMBER OF LAYERS = 01
       LENGTH        CONDUCTIVITY
        (M)             (W/M*K)
   1.7500000E-03    1.7653000E+02
    LAYER IN WHICH HEATER RESIDES        LHEAT = 1
    NUMBER OF HEAT TRANSFER COEFFICIENTS INPUT = 0027
       S/C           HTC(W/M*M*K)
  -1.0500000E+00    1.0000000E+01
  -0.4000000E+00    1.0000000E+01
  -0.3000000E+00    1.5000000E+01
  -0.2000000E+00    2.0000000E+01
  -0.1500000E+00    3.0000000E+01
  -0.1000000E+00    5.0000000E+01
  -0.0500000E+00    7.5000000E+01
  -0.0400000E+00    1.7500000E+02
```

```
-0.0300000E+00     2.5000000E+02
-0.0250000E+00     3.5000000E+02
-0.0225000E+00     5.0000000E+02
-0.0175000E+00     7.5000000E+02
-0.0125000E+00     8.7500000E+02
-0.0062500E+00     9.5000000E+02
 0.0000000E+00     1.0000000E+03
 0.0062500E+00     9.5000000E+02
 0.0125000E+00     8.7500000E+02
 0.0175000E+00     5.0000000E+02
 0.0225000E+00     3.0000000E+02
 0.0250000E+00     1.0000000E+02
 0.0500000E+00     7.5000000E+01
 0.1000000E+00     5.0000000E+01
 0.1500000E+00     3.0000000E+01
 0.2000000E+00     2.0000000E+01
 0.3000000E+00     1.5000000E+01
 0.4000000E+00     1.0000000E+01
 1.0500000E+00     1.0000000E+01
```

8.2 Grid Input Files (xy.plt and q.plt)

LEWICE 2.0 has the option to bypass the potential flow solution and read in a flow solution from a grid-based flow solver. This option limits the user to a single time step. This option has also seen very limited testing and has not been validated against the database of experimental ice shapes. It can read up to 10 grid blocks and each block can contain up to 600x200 grid points. The import format for the grid conforms to the PLOT3D[11] format with iblanking. The input data files must contain REAL*4 binary data. The read statements needed to create the files are as follows:

8.2.1 Grid read statements (xy.plt)

```
 Read(2) mg
 Read(2) (imns(mqf),jmns(mqf),mqf=1,mg)
 Read(2) (( xg(i,j,mqf),i=1,im),j=1,jm),
1  (( yg(i,j,mqf),i=1,im),j=1,jm),
2  (( iblank(i,j,mqf),i=1,im),j=1,jm)
```

8.2.2 Flow solution read statements (q.plt)

```
 Read(2) mg
 Read(2) (imns(mqf),jmns(mqf),mqf=1,mg)
 Read(2) mach,alpha,reair,time
 Read(2) (( q(i,j,1),i=1,im),j=1,jm),
1         (( q(i,j,2),i=1,im),j=1,jm),
2         (( q(i,j,3),i=1,im),j=1,jm),
3         (( q(i,j,4),i=1,im),j=1,jm)
```

Note: For a single body, LEWICE 2.0 assumes that the grid is in the single block format used by PLOT3D. Therefore, the number of grid blocks (mg) is not read in.

There is no error checking of the grid and solution file. The user should independently verify that the grid and solution files are correct for the case being run. In particular, the angle of attack and velocity should match the values input in the main input data file. The program will also not run with a grid solution unless the grid surface geometry is very similar to the body geometry read in from the geometry input file(s). An example case is provided in this manual which uses the grid input function. The example case will further describe errors which can occur when using this feature.

Chapter 9: Output Files

This list contains a description of the LEWICE output files included on the distribution disk. It will also duplicate some of the input file information where necessary. Data is output sequentially with regard to time. This means that the first set of outputs in a given file are the results for the first time step, the second set of outputs are the results for the second time step and so on. This is illustrated in Table 11. The values in parenthesis in the following descriptions are the titles used for the columns of data in the output files.

Table 11: Example of Data Format in Output Files

```
Header Text
data from time step 1
data from time step 2
.
.
.
data from last time step
```

9.1 beta.dat

This file contains collection efficiency output for each time step. Columns are dimensionless wrap distance from the stagnation point (s/c), collection efficiency (beta), dimensionless wrap distance as measured from the airfoil leading edge (sle/c), dimensionless x-coordinate of the airfoil (x/c) and dimensionless y-coordinate of the airfoil (y/c). This output will be generated if the output flag BPRT is set to 1. When a drop size distribution is used, only the composite collection efficiency is written. The collection efficiency for each individual drop size is not output.

Note: The format for this output file has changed since the validation runs were made. At that time, the format for this file conformed to the output format specified in the LEWICE 1.6 User Manual[6]. The current output format is based upon several requests from users.

Note: The wrap distance from the leading edge may lose physical meaning past the first time step.

Note: The impingement limit listed in the output file "imp.dat" may not match the wrap location where the collection efficiency (beta) goes to zero in the output file "beta.dat". The difference is due to the resolution of surface points in the impingement limit region. The value quoted in the file "imp.dat" is the computed impingement limit. The location where the collection efficiency goes to zero is a function of both the computed impingement limit and the body geometry coordinates.

9.2 dens.dat

This file contains predicted ice density at each location for each time step. Column are wrap distance from stagnation (s/c) and ice density (density) in kg/m^3. If the input flag MPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.3 dyice.dat

This file contains ice thickness and other results from the mass balance. Columns are: wrap distance from stagnation (s/c), ice height to be added at each location (dice), velocity of runback water (vrunback) in m/s, and the ice area to be added at each location (aice) in m^2. If the input flag MPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.4 final1.dat

This file contains the final ice shape produced by LEWICE on the first body. If the program stopped due to an error, this file will not be output. The first row of data contains the number of points on the ice shape and the remaining lines contain the x,y coordinates of the ice shape in inches. This file format can also be used as input to the utility program THICK which calculates parameters of the ice shape for comparison with digitized ice shapes from experiments.

9.5 final2.dat, final3.dat, final4.dat, final5.dat

These files contain the final ice shape for the other bodies. Final2.dat contains the iced geometry for the second body, final3.dat contains the iced geometry for the third body and so on. These files will be generated only if more than one body is being simulated. The data is formatted the same as for file final1.dat.

9.6 fixed.dat

This file contains the clean airfoil geometry after the initial geometry checks. This file will only be output if the geometry has changed as a result of this error checking. See the section on the geometry input file for a listing and description of these checks. The data for this file consists of the new x,y coordinates for every body. Each body is offset by the header "Body # (value)". The first column of the output file contains the new x-coordinates while the second column contains the y-coordinates of the changed geometry. The user should check this geometry to ensure that the corrections made by the code are acceptable. If they are not, the user will need to fix the geometry input file and resubmit this case.

9.7 flow.dat

This file contains the output from the potential flow solution at each time step. Columns contain the panel index (i), dimensionless x,y coordinates (x/c,y/c) at the panel center (not at the endpoints as with other files), dimensionless wrap distance as measured from the lower surface trailing edge (s/c), dimensionless tangent velocity (vt), pressure coefficient (cp), a separate panel index for each body (j), the panel source/sink value (sigma), and the dimensionless normal velocity (vn). One flow solution is written to disk for each time step and an additional flow solution is generated on the final ice shape before the program exits. Previously, the last flow solution performed by LEWICE was on the iced geometry at the next-to-last time step. This output will be generated if the input flag FPRT is set to 1 or 2. However, no output will be generated to this file if the user sets the grid flag **on** (IGRID=1) since the potential flow solution will not be performed if this flag is set on.

9.8 fract.dat

This file contains mass fraction output from the LEWICE mass balance at each time step. Columns are dimensionless wrap distance from stagnation (s/c), mass fraction of incoming water which does not freeze, evaporate, or runback (xtot), mass fraction of incoming water which freezes (ffrac), mass fraction of incoming water which evaporates (envap), and scaling factor for runback (xvr). The mass fraction of incoming water which freezes is commonly known as the freezing fraction. If the input flag MPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

Note: The output is generated starting from the stagnation point toward the lower surface trailing edge. Output from the stagnation point toward the upper surface trailing edge follows this. Output from subsequent time steps will follow after the output from the first time step, as is the case for the other output files.

9.9 htc.dat

This file contains the convective heat transfer coefficient at each time step. Columns are segment number (seg), dimensionless wrap distance from stagnation (s/c), heat transfer coefficient (htc) in W/m^2/K, and Frössling number (fr). The Frössling number output is the local Nusselt number divided by the square root of the ambient Reynolds number. If the input flag HPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.10 ice1.dat

This file contains the ice shape for the first body at each time step. Columns contain the coordinates of the ice shape (x,y) in inches, ice thickness (thick) in inches and wrap distance from the stagnation point (s) in inches. LEWICE calculates these variables in dimensionless units but outputs the data in inches as all of the experimental data is taken in inches. The first set of points contains the airfoil prior to any ice build up. Each successive set of data contains the ice shape at a specified point in time. The last set of data contains the final ice shape, which is also output to the file "final1.dat".

9.11 ice2.dat, ice3.dat, ice4.dat, ice5.dat

These files contain the ice shape for the other bodies at each time step. Ice2.dat contains the iced geometry for the second body, ice3.dat contains the iced geometry for the third body and so on. These files will be generated only if more than one body is being simulated. The data is formatted the same as for file ice1.dat.

9.12 imp.dat

This file contains information related to the impingement limit for each time step. Columns are droplet size (size) in microns, dimensionless x-coordinate of the lower impingement limit (xlow), dimensionless y-coordinate of the lower impingement limit (ylow), dimensionless wrap distance from stagnation of the lower impingement limit (slow), dimensionless wrap distance from the leading edge of the lower impingement limit (slowle), dimensionless x-coordinate of the upper impingement limit (xhi), dimensionless y-coordinate of the upper impingement limit (yhi), dimensionless wrap distance from stagnation of the upper impingement limit (shi), and dimensionless wrap distance from the leading edge of the upper impingement limit (shile). This output will be generated if the output flag BPRT is set to 1. If more than one body is being simulated, data for the second body will be listed beneath the output for the first body and data for each subsequent body will follow this output.

Note: The wrap distance from the leading edge may lose physical meaning past the first time step.

Note: The impingement limit listed in the output file "imp.dat" may not match the wrap location where the collection efficiency (beta) goes to zero in the output file "beta.dat". The difference is due to the resolution of surface points in the impingement limit region. The value quoted in the file "imp.dat" is the computed impingement limit. The location where the collection efficiency goes to zero is a function of both the computed impingement limit and the number of points on the surface.

9.13 junk.dat

This file contains information useful for debugging the code and any screen outputs including warning or error messages. All of this output will be generated if the output flag IDBF is set to 1. This flag only needs to be set on if problems occur when running the code. By default in version 2.0, much of this printout is turned **off**. When the input flag IDBF is set to 0, only the most important debug information is sent to this file.

9.14 mass.dat

This file contains mass flux terms from the LEWICE mass balance at each time step. Columns are dimensionless wrap distance from stagnation (s/c), mass flux of incoming water which freezes (mdotf) in kg/m^{2}s, mass flux of impinging water (mdotc) in kg/m^{2}s, mass flux of incoming water which evaporates (mdote) in kg/m^{2}s, mass flux of runback water (mdotri) in kg/m^{2}s, mass flux of incoming water (mdotti) in kg/m^{2}s, mass flux of incoming water minus mass flux which evaporates (mdott) in kg/m^{2}s, and potential mass flux which could be evaporated (emexs) in kg/m^{2}s. If the input flag MPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.15 misc.dat

This file contains other miscellaneous output from the code. It contains a complete copy of the main input data file, lift results from the flow code, and individual trajectory information among other information.

9.16 pres.dat

This file contains the compressible flow solution at the edge of the boundary layer. Columns are segment number (seg), dimensionless wrap distance from stagnation (s/c), dimensionless velocity at the edge of the boundary layer (ve), dimensionless temperature at the edge of the boundary layer (te), dimensionless pressure at the edge of the boundary layer (press) and dimensionless density at the edge of the boundary layer (ra). Reference variables which were used to nondimensionalize these quantities are chord length, ambient velocity, freestream total temperature, freestream total pressure and freestream total density, calculated from the equation

$$\rho_o = \frac{P_o}{RT_o} \tag{2}$$

If the input flag HPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.17 qener.dat

This file contains individual terms from the energy balance. Columns are wrap distance from stagnation (s/c), net convective heat loss (qconv), evaporative heat loss (qevap), sensible heat loss/gain (qsens), latent heat gain (qlat), conduction heat loss/gain (qcond) and the sum of these individual terms (qtot). The numbers in this last column should be very close to zero as an indicator that energy was balanced at that control volume. If the input flag EPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.18 temp.dat

This file contains the surface temperature output from the energy balance. Columns are wrap distance from stagnation (s/c), surface temperature (t) in degrees Kelvin, and 'recovery' temperature (t_rec) in degrees Kelvin. If the input flag EPRT is set to 1, the output from every 1/10th control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

9.19 thick.dat

This file contains the ice thickness for each time step as measured from the clean surface. The ice thickness output in the ice1.dat file provides the ice thickness measured from the current ice shape. The thick.dat file was created to show the ice thickness from a common reference, i.e., the clean airfoil. This output file is similar to the ice thickness output file "clean.dat" created by the utility program THICK. However, output is sent to thick.dat only for every 1/10th control volume, so the output to this file will appear coarse compared to the output from program THICK as it outputs the ice thickness at every point. Columns are the x-coordinates of the clean surface (xsav) in inches, the y-coordinates of the clean surface (ysav) in inches, the ice thickness as measured from the clean surface (ditot) in inches, the cumulative ice area (area) in inches and the wrap distance from the leading edge of the clean surface (s) in inches.

9.20 traj1.dat

This file contains the dimensionless x,y coordinates of individual droplet trajectories. Droplet trajectory output is sequential. The first set of coordinates contain the coordinates of the first trajectory calculated. Subsequent output contains coordinates for each successive trajectory calculated. No indicator is present in the output file to offset trajectory output from one time step to another. Hence, this output is only recommended for single time step cases. If the input flag TPRT is set to 1, only droplet trajectories used for the collection efficiency calculation will be generated. If this flag is set to 2, all droplet trajectories will be generated. Note that the definition of this flag has changed from version 1.6[6]. These files are very large. If this information is not needed, the user can save a great deal of disk space by not generating this file.

9.21 traj2.dat, traj3.dat, traj4.dat, traj5.dat

These files contain the droplet trajectories calculated for the other bodies at each time step. Traj2.dat contains the droplet trajectories for the second body, traj3.dat contains the droplet trajectories for the third body and so on. These files will be generated only if more than one body is being simulated. The data is formatted the same as for file traj1.dat. These files are very large. If this information is not needed, the user can save a great deal of disk space by not generating this file.

9.22 xkinit.dat

This file contains the predicted sand-grain roughness at each time step. Columns are time in seconds (time), and two predictions for sand-grain roughness which are calculated by different sets of equations (xkinit1) and (xkinit2). LEWICE 2.0 uses the last value listed on each line (xkinit2) as the sand-grain roughness, which is also dimensionless.

9.23 xkinit2.dat

The previous file (xkinit.dat) contains the average sand grain roughness which is used by LEWICE in the heat transfer coefficient calculation. LEWICE calculates a local roughness value but uses the average for numerical purposes. The file xkinit2.dat contains the local roughness coefficients. Columns are the wrap distance from stagnation (s/c), the local sand grain roughness (xk) in millimeters, the film thickness (hflow) in millimeters, and the water bead height (hbead) in millimeters. The LEWICE Programmers Manual[2] will go into more detail on the differences between these terms and how they are calculated.

Chapter 10: Additional Output Files

These files will be generated if nonstandard values have been selected for flags IDEICE and IGRID in the main input file.

10.1 noice.dat

This file contains output from the anti-icing calculation. Columns are dimensionless wrap distance from stagnation (s/c), heat required at that control volume (qheat) in W/in^2, maximum temperature (T_max) in degrees Kelvin and surface temperature (T_surf) in degrees Kelvin. This output will be generated if the deicing flag is set **on** (IDEICE = 1). For a multi-body case, output for each body will be generated in succession. For a hot air anti-ice system, the maximum temperature column will contain the local temperature of the air stream. For an electrothermal system, the maximum temperature column will contain the local temperature of the heater. In either case, the predicted temperature will be extremely high due to the simplistic assumptions made in the solution process. The user is strongly advised to use the ANTICE[9] and LEWICE/Thermal[8] codes for more accurate prediction of these maximum temperatures.

10.2 ctemp.dat

This file contains surface velocities from the grid solution file. This output will be generated if the grid flag is on (IGRID = 1). Columns are grid index value (i), body index value (ii), dimensionless x-coordinate (xoc), dimensionless y-coordinate (yoc), dimensionless surface velocity (ve), and surface pressure coefficient (cp).

10.3 geometry.dat

This file contains the surface geometry of the body(s) read in from the grid solution file. The columns are the dimensionless x,y coordinates of the surface geometry. This output will be generated if the grid flag is on (IGRID = 1). LEWICE will not run with a grid solution unless this geometry is very similar to the one read in from the geometry input file(s). No error checking has been added to ensure this.

Chapter 11: Utility Programs

This section will describe five utility programs which are released along with LEWICE 2.0. The inputs and outputs to these utility programs will also be discussed.

11.1 Program THICK

Program THICK generates an ice thickness distribution using as input a clean airfoil geometry and an iced geometry. The iced geometry can come from LEWICE or from experiment. The program also outputs the quantitative parameters which were used for the validation tests. The results of these tests have been published in a recent contractor report[1]. Only one body can be processed for this program. A multi-body case can be processed by running THICK for each individual body.

Basically, the program calculates the minimum distance from each point on the ice shape to the clean surface. For small ice thicknesses on the same order of magnitude as the point spacing, the thickness is determined by using the unit normal from the surface. It is possible for complex ice shapes to have more than one ice thickness at a given location on the clean surface. In this case, the maximum of these individual ice thicknesses is used. The program will be able to calculate an accurate approximation to the ice thickness at each location if the number of points on the clean surface is sufficiently large. For all of the validation tests, at least 5000 points were used on each clean airfoil. The user should consult the report on the validation tests[1] and the LEWICE Programmers Manual[2] for further details on the methodology used in this program.

11.1.1 Interactive Input for Program THICK

Input Clean Geometry File Name

The first interactive input directs the user to input the name of the input file containing the clean geometry. The name can be up to 80 characters long. This filename length is necessary as the user must also input the directory path of this file if it is different from the directory containing the program. Please read the error messages in this section which describes the proper form of the input when using a directory path. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 8 named
last format:
Unit 8 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 8, file C:\Lewice\test.xyd
```

This error indicates that the file name input by the user does not exist, or does not exist in this directory. Common problems:

1) The file name was not typed correctly (remember to include the extension - use "case1.xyd" not simply "case1");

2) The input file is in a different directory than the program and the user did not specify the directory. The input file can be in a different directory than the program, but in order for THICK to recognize the input file the path must be specified. For example, use "inputs\naca0012\case1.xyd" instead of simply "case1.xyd" to read the input file "case1.xyd" in the directory "inputs" and subdirectory "naca0012". **Note:** The above example used the DOS directory convention of backward slashes "\" to list subdirectories. IRIX and many other unix systems use forWard slashes "/" instead.

PC Note: To get to the root directory, first type a backward slash "\", then the path and file name. For example, the command "\lewice\inputs\naca0012\case1.inp" can be also be used to read the file "case1.inp" in the directory "C:\lewice\inputs\naca0012".

Unix Note: It is common practice in unix to place all programs in a predefined directory such as /usr/bin so that everyone using that system can run the program. The path for specifying the input file in this case is to provide the path from the directory the user is in. For example, if the user is in their home directory and the input file is in the home directory, no path should be provided. If the user is in their home directory and the input files are in directory ../inputs/naca0012, then the proper path to input is "inputs/naca0012/case1.inp". If the user is in directory ../inputs/naca0012 and the input file is in this directory, then no path needs to be provided in this case either. **P.S.:** This sequence is correct based on the IRIX 6.2 operating system. Behavior for other unix operating systems are expected to be similar, but potentially could be different.

Input Iced Geometry File Name

This interactive input directs the user to input the name of the file containing the iced geometry. The iced geometry can come from experiment or from an ice shape prediction and does not have to be a complete airfoil. Digitized ice shapes from experiments are often supplied only up to the end of the measured ice shape. If the file cannot be accessed, the error message listed previously for the clean geometry input file name will be generated. In this case however, the unit number being referenced will be unit 9. The corrective measures concerning file name and directory path listed earlier for this error will also apply to errors involving the iced geometry.

This completes the section on interactive inputs for program THICK. The file format for the input files will now be discussed.

11.1.2 File Format for Clean Geometry and Iced Geometry

The file format for the clean geometry and the iced geometry are the same. The first line of the file must be the number of points in the geometry. Many text editors and word processors are capable of providing a line count for a file. Each of the following lines of the file contains an x,y coordinate pair for the body geometry. The x-coordinate is listed first. The format of the data is free-format for the x,y coordinates. It is quite common for problems to arise when inputting a new geometry for the first time. Consult the section in this report on entering body geometries into LEWICE if there are problems with this input. Although the file format for the input files are the same, the number of points on the clean geometry input file should be much higher in order to capture the detail in the ice thickness distribution. The validation tests on LEWICE used over

5000 points for the clean geometry files when running program THICK. In order to aid the user in constructing input files with this many points, a second utility program called EXPAND was created. This program will now be discussed in more detail.

11.2 Program EXPAND

Program EXPAND will read in a set of x,y coordinates from a file and output the same geometry with a different number of points. The additional points are calculated by linear interpolation. This program is useful for creating clean geometry input files for use with program THICK.

11.2.1 Interactive Input for Program EXPAND

Enter Input File Name

The first interactive input directs the user to input the name of the input file containing the geometry. The name can be up to 80 characters long. This filename length is necessary as the user must also input the directory path of this file if it is different from the directory containing the program. Please read the error messages in the previous section concerning the proper form of the input using a directory path. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 8 named
last format:
Unit 8 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 8, file C:\Lewice\test.xyd
```

Enter Output File Name

This interactive input directs the user to input the name of the file containing the output geometry. **Note:** If the file name input already exists, it will be overwritten.

Enter Number of Points

This interactive input directs the user to input the number of points for the output geometry. The number can be free-format, but must be within the range 1 ≤ NPOINTS ≤ 10000. Input values outside this range will generate the following warning message:

Number of points input (value) is out of range. Using default value of 5000.

If a nonnumerical value is input for the number of points, it will be converted to integer for use in the program on the SGI system. The likely result of this action is that the default number of points will be used. On the PC, the following system error will be generated:

forrtl: severe (59): list-directed I/O syntax error, unit 5, file CONIN$

This completes the section on interactive inputs for program EXPAND. The file format for the input files will now be discussed.

11.2.2 File Format for Input and Output Geometries

The file format for the input and output geometries are the same. The first line of the file must be the number of points on the geometry. Many text editors and word processors are capable of providing a line count for a file. Each of the following lines of the file contains an x,y coordinate pair for the body geometry. The x-coordinate is listed first. The format of the data is free-format for the x,y coordinates. It is quite common for problems to arise when inputting a new geometry for the first time. Consult the section in this report on entering body geometries into LEWICE if there are problems with this input.

Note: The input file format for this program is not the same as the geometry input file format for LEWICE 2.0. Read the previous description on the input file format.

11.3 Program CONVERT

Input files from previous versions of LEWICE will not work with version 2.0. Program CONVERT will read in an input file from LEWICE version 1.6 and convert it for use in LEWICE version 2.0. The program will prompt the user for the name of the LEWICE 1.6 input file and the name to be given to the LEWICE 2.0 input file.

11.3.1 Interactive Input for Program CONVERT

Enter Version 1.6 Input File Name

The first interactive input directs the user to enter the name of the LEWICE 1.6 input file to be converted. The name can be up to 80 characters long. This filename length is necessary as the user must also input the directory path of this file if it is different from the directory containing the program. Please read the error messages in the section on the main input file concerning the proper form of the input using a directory path. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 8 named
last format:
Unit 8 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 8, file C:\Lewice\test.xyd
```

Enter LEWICE 2.0 File Name

This interactive input directs the user to input the name of the file which will contain the LEWICE 2.0 input. **Note:** If the file name input already exists, it will be overwritten.

11.3.2 Format for the LEWICE 1.6 Input File

Program CONVERT will read input files which conform to the format listed in the LEWICE 1.6 User Manual^6. No error checking of this input file is performed.

Note: Variable IDEICE will be set to its default value (0) as it was not present in the LEWICE 1.6 inputs.

Note: Many of the print flags in the LPRT namelist have changed definition from version 1.6 to version 2.0. Refer to those sections describing this input for a more detailed explanation.

11.3.3 Warning Message for CONVERT

The flag INCLT in the LEWICE 1.6 input file is not used in LEWICE 2.0. If the flag INCLT=1, then the following warning message will appear:

The flag INCLT is set to 1 in the LEWICE 1.6 input file. This means that the value of CLT in the LEWICE 1.6 input file is probably the lift coefficient instead of the angle of attack. Since the value of CLT was used as the value of AOA in the LEWICE 2.0 input file, make sure that the value of AOA is the desired angle of attack.

This message serves only as a warning and the LEWICE 2.0 file will be generated.

11.4 Program TOBINARY

PLOT3D files are read into LEWICE in binary format. This utility program converts PLOT3D files in text format to binary. It is useful for transferring the PLOT3D files to different hardware platforms. The program requires no interactive input. The text file names are "xy.txt" for the grid and "q.txt" for the flow solution. The binary output files are named "xy.plt" and "q.plt" respectively. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 8 named
```

```
last format:
Unit 8 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 8, file C:\Lewice\test.xyd
```

Note: If the output files "xy.plt" and "q.plt" already exists, they will be overwritten.

Note: This program was designed to be used to convert files in the PLOT3D format from text to binary format. It is not a general purpose conversion utility.

11.5 Program TOASCII

PLOT3D files are read into LEWICE in binary format. This utility program converts PLOT3D files in binary format to text format. It is useful for transferring the PLOT3D files to different hardware platforms. The program requires no interactive input. The binary file names are "xy.plt" for the grid and "q.plt" for the flow solution. The text output files are named "xy.txt" and "q.txt" respectively. If the file cannot be accessed, the following system error will be generated:

IRIX 6.2 Message

```
open(name): No such file or directory
apparent state: unit 8 named
last format:
Unit 8 is a sequential formatted external file
*** Execution terminated (2) ***
```

Windows95 DOS Shell Message

```
forrtl: The system cannot find the file specified.
forrtl: severe (29): file not found, unit 8, file C:\Lewice\test.xyd
```

Note: If the output files "xy.txt" and "q.txt" already exists, they will be overwritten.

Note: This program was designed to be used to convert files in the PLOT3D format from binary to text format. It is not a general purpose conversion utility.

Chapter 12: Runtime Errors

This section will describe errors which may occur during a run. Errors during a run are very infrequent with this code, but may still occur if a problem arises which the code cannot handle. The most likely cause of these errors is that the user is analyzing a case outside of the range of data used for validation. Over 400 runs were made with LEWICE 2.0 during the validation testing phase. In one case, an error occurred because the DSMN value was too small and the array size was exceeded. This case ran successfully with a slightly higher DSMN value. In no other case did any error occur which caused the code to stop prior to its normal completion. This occurred even though many of the tests were in fact selected in an attempt to create runtime errors. Corrective actions which can be taken are discussed following the listing of each individual error message. All of the errors listed will stop the run before completion unless otherwise noted. The runtime error which is generated will be in *italics*, followed by a description of the error. Every possible error message is listed and they have been organized by type. Any error messages which occurred during the validation testing phase are identified in this section.

12.1 Flow Solver Errors

The flow solver has encountered an error. Usually, this means that there is something wrong with the body geometry(s) such as two bodies intersecting. Please check the last ice shape print-out for problems and check that the case conforms to the recommended limits on inputs. Because of this error, the program cannot continue.

The error listed above can only occur for multi-body simulations such as multi-element airfoils. One potential cause of this error can occur when multi-element airfoils begin to intersect due to ice growth. It is also possible that errors occurred in the ice growth process during the previous time step. The user can run a different spacing (DSMN) on the geometry or different time step (IFLO) for this case. However, it may be the case that this is a correct prediction and that the elements have fused together due to ice growth. The run can only be continued from this point if the user resubmits the case using the last iced geometry and treating the two individual bodies as a single body.

This situation can only occur when the number of panels is greater than 1001 and you have increased both NPAF and NBF parameters. To solve this problem increase KORE in routine SOLVIT and recompile.

If this error occurs, one of the array size dimensions has been exceeded. As stated in the error message, this can occur only if the user has modified the program. The user should also check the iced geometry for abnormalities.

Attempted to load the (value)th body. Maximum allowable number of bodies is (value).

Because of the above error, this run is terminated.

This error should never occur since LEWICE now checks the input for the number of bodies.

The number of elements (value) will exceed allowable storage (value) when added to the data set.

Because of the above error, this run is terminated.

The array size had been exceeded. LEWICE should have stopped when adding ice during the previous time step. The last iced geometry is likely bad. The user will need to increase DSMN to run this case.

The number of elements (value) for the new body exceeds the number (value) for the body it is replacing.

Because of the above error, this run is terminated.

The array size had been exceeded. LEWICE should have stopped when adding ice during the previous time step. The last iced geometry is likely bad. The user will need to increase DSMN to run this case.

12.2 Errors when Using Grid Solutions

This section covers specific errors which can occur when attempting to use a grid-based flow solution.

Cannot find the drop in the grid. Usually, this means the grid was read wrong or there is a problem searching multi-block grids. Did not find interpolation stencils.

.... the point: xpt= (value)   ypt= (value)

program stop in searchpt

During the droplet trajectory calculation, LEWICE attempted to find the grid cell containing the drop so that the air velocity at that point could be determined. This search was unsuccessful. Check that the grid matches the airfoil input in the geometry input file and that the flight conditions in the main input file match those in the flow solution input file. The user should also refer to the trouble shooting given for Example Case 12.

Cannot find the drop in the grid. Usually, this means the grid was read wrong or there is a problem searching multi-block grids.

```
*****************************************
         program stop. !!!
         subroutine:  stencils!
*****************************************
```

This error occurs for the same reason as the previous error but in a different subroutine.

Cannot find the drop in the grid. Usually, this means the grid was read wrong or there is a problem searching multi-block grids.

**** *big trouble!* ****

This error occurs for the same reason as the previous error but in a different subroutine.

12.3 Trajectory Errors

This section lists errors which can occur during the droplet trajectory calculation.

(value) trajectories are calculated in range. There is something wrong with the flow solution for this to happen. Program cannot continue with run.

In the initial phase of the droplet trajectory routine, droplets are sent toward the model to find an upper and lower bound for the impingement limit search. This error occurs when this range cannot be found. Usually, there is an error in the iced geometry from the previous time step. The user may need to adjust the point spacing (DSMN) or time step (IFLO) for this case.

****kflag= (value) difsub failed****

Integration of trajectory stopped before a result could be found. Program will assume that the trajectory missed which may not be correct. Please check output carefully.

During each step in the trajectory integration, the program needs to iterate on the droplet location for the next step. This message will appear if convergence was not achieved. This message occurs very rarely. In the validation test cases, it occurred once in over 400 LEWICE runs. In that case, the program assumed the particle had missed the body and continued the run without further incident. If this message occurs repeatedly during a run, the user should stop the case manually (using <ctrl> C) as a problem has occurred. This may occur for very complex ice shapes. The user may need to adjust the point spacing (DSMN) or time step (IFLO) for this case. Below are the definitions of the "kflag" error values.

```
-1  the step was taken with h = hmin, but the requested error was not achieved.
-2  the maximum order specified was found to be too large.
-3  corrector convergence could not be achieved for h .gt. hmin.
-4  the requested error is smaller than can be handled for this problem.
```

*** 1000*ibod steps in intig ***

Integration of trajectory stopped before a result could be found. Program will assume that the trajectory missed which may not be correct. Please check output carefully.

end point xp = (value) yp = (value)

Each trajectory is limited to no more than 1000*ibod steps, where "ibod" is the number of bodies input by the user in the main input file. It is very rare for a trajectory to need more than 200 steps for completion. This upper limit is normally reached because the step size has gotten very small. In certain multi-element cases, more steps are needed as the program will start the trajectories far upstream in order to obtain free stream conditions for the initial air flow around the drop. As with the previous warning message, the program will continue as though the trajectory missed the body. This assumption can be checked as the program prints out the last droplet location to the screen and to the debug file. If this message occurs repeatedly during a run, the user should stop the case manually (using <ctrl> C) as a problem has occurred. This may occur for very complex ice shapes on multi-element airfoils. The user may need to adjust the point spacing (DSMN) or time step (IFLO) for this case.

Newton-Raphson did not converge in vterm. minimum error = (value), last calculated value (value), value at minimum error (value).

The program attempted to calculate the freefall (terminal) velocity of a particle to be used as an initial condition for the trajectory integration. This error may be indicative of problems with the flow solution. The value at the minimum error will be used and the code will continue to run.

---

number of trajectories is more than (value). Flow solution is probably bad. Program cannot continue. Check output carefully

The program is limited to 199 trajectories during a single impingement limit search. It is highly likely that there is a problem with the iced geometry from the previous time step which caused a poor flow solution. The user will likely need to adjust the point spacing (DSMN) or time step (IFLO) for this case.

Impingement limit could not be found. Flow solution is probably bad. Continuing with rest of run. Please check output thoroughly.

y0max =(value), y0min = (value)

The impingement limit is determined by a binomial search algorithm. This error will occur if the starting point for the next trajectory is the same as the last trajectory calculated. This may have occurred if the iced geometry from the previous time step caused a poor flow solution. The program will continue as if no impingement limit exists.

Program cannot decide on path for next trajectory. Output may not be reliable. Check results carefully.

This error can only occur if a multi-body simulation is being performed. At the end of each trajectory, the program must decide where to start the next trajectory based on where hit trajectories have occurred and if this particle passed over or under the body for which the impingement limit is being sought. The error occurs when the data is conflicting on the proper decision. The program will choose one of the two possible paths and continue with the run. The user should check for other trajectory errors and also check the iced geometry produced at the last time step.

12.4 Energy Balance Errors

This section covers errors which occur during the calculation of the mass and energy balance. An error in this section is usually indicative of problems elsewhere in the code. The iced geometry from the last time step and the flow solution for this time step should be checked for possible abnormalities.

Maximum ice thickness is much too large. ymax = (value)

The calculated ice thickness at one of the control volumes is too large to be added to the ice shape. This may indicate that the number of time steps should be increased for this case.

The temperature sent to function PVAP is outside the range for this routine. There is a problem with this case. Be careful interpreting results.

big error in pvap t = (value)

This error can only occur if the calculated surface temperature is below 32 K. Obviously, there is an error in the case for this to occur. The program will attempt to continue, but the user should manually terminate the case (<ctrl> C) and investigate the problem with the ice shape.

In NOICE, Newton-Raphson did not converge. minimum error = (value)

last temp. predicted = (value), temp. at min. error = (value)

The output data for anti-icing may be incorrect.

The program must iteratively solve for surface temperature for evaporative anti-icing cases. This error occurs when convergence was not found. The program will continue with the temperature value at the minimum error.

In SOLVEW, Newton-Raphson did not converge. minimum error = (value)

last temperature calculated (value), temperature at minimum error = (value)

The output data may be incorrect.

The program must iteratively solve for surface temperature during the energy balance. This error occurs when convergence was not found. The program will continue with the temperature value at the minimum error.

12.5 Ice Growth Errors

If the errors listed in this section occur, the most likely cause of the error is a problem when the predicted ice growth is added to the current ice shape.

Stopping because nsteps too large.

nsteps = (value)

This error occurs because the code is trying to add too much ice at one time. This could be due to a bad solution from one of the earlier routines (such as FLOW, TRAJ or ICE). Check that your case conforms to the guidelines for time step (IFLO) and point spacing (DSMN).

If it does and the case still bombs here then you have found a legitimate bug in LEWICE. It would be very helpful for future versions if we (NASA) could have a copy of the input file so this bug can be fixed. Also include what machine the code ran on and if it was recompiled.

The user most likely input fewer time steps than recommended if this message appears. Increase the number of time steps (IFLO) for this case.

Stopping because negative nsteps.

nsteps = (value)

This error occurs because the code is trying to add too much ice at one time. This could be due to a bad solution from one of the earlier routines (such as FLOW, TRAJ or ICE). Check that your case conforms to the guidelines for time step (IFLO) and point spacing (DSMN).

If it does and the case still bombs here then you have found a legitimate bug in LEWICE. It would be very helpful for future versions if we (NASA) could have a copy of the input file so this bug can be fixed. Also include what machine the code ran on and if it was recompiled.

This error can also occur if too few time steps are used. Increase the number of time steps (IFLO) for this case.

WARNING!!! The point spacing at the following location is zero. The code will attempt to continue, but the output may be bad or the code will bomb later. This statement is in routine NWF3 and is usually caused by a memory allocation error.

i= (value), xbtdot= (value), dsdd= (value).

This problem should never occur in LEWICE 2.0. Please consider sending this case to us (NASA) so that this case can be investigated.

---

spline routine failed. doing linear interpolation

The program uses a cubic spline to generate surface points. If this fails, this warning informs the user that linear interpolation will be used. This is a minor error and the output is likely ok.

---

danger you are now extrapolating. Problem in routine lntp.

i= (value) j= (value), xnl(j)= (value), xol(1)= (value), xol(no)= (value)

The program will also use linear interpolation to find new surface points. This warning informs the user that the code needed to extrapolate to find points. This should not normally happen. The iced geometry should be checked. Also consider changing the point spacing (DSMN) and/or the number of time steps (IFLO) for this case.

---

error nwfoil (value) (value) (value) (value) (value) (value)

The program attempted to smooth the ice shape before the next time step, but was unsuccessful. This is a minor error and the output is likely ok.

Array size exceeded. Increase npa to at least npa= (value)

The program needed more points than are currently allowed. The user should increase the DSMN value for this case. Alternatively, the programmer may wish to consider increasing the array size and recompiling the code.

ARRAY STORAGE EXCEEDED!!!!!

Strange things may have happened to ice shape! Do one of the following:

(1) Rerun the case with a larger DSMN input

(2) Increase the value of NPA in file param.inc

For the second option, LEWICE must be recompiled. You will need at least (value) points to run this case.

This error is the one most likely to occur during a run. It will appear when the user selects a DSMN value which is too small for the case being run. For a typical clean airfoil, the total wrap distance is slightly greater than 2 (dimensionless). For a DSMN value of $4*10^{-4}$, this will create over 5000 control volumes on that shape. As the array size limits are 10000, the total wrap distance around the iced geometry needs to be more than four times greater than the chord length for this error to appear. For a DSMN value of $8*10^{-4}$, the total wrap distance around the iced geometry needs to be more than eight times greater than the chord length. The ratio of wrap distance to chord length for a cylinder is by definition pi (3.1415926536...). This ratio will increase as the ice grows on the cylinder. This example shows why larger DSMN values are likely needed for a cylinder.

Chapter 13: Example Cases

The following example cases have been included to illustrate different features of LEWICE. Many of the cases correspond to runs made for the validation report. Those examples which correspond to a particular validation case are identified. The cases are included for illustration purposes only. The user does not have to run all of the cases provided. Output for the cases are provided on the distribution disk. The results shown in this section were generated using the PC executable on the distribution disk. The validation report showed minor differences in output across platform, but the results from two different PCs using the PC executable on the distribution disk were exactly the same.

In each of the examples, all of the possible input fields have not been specified. Those which have not been specified will be assigned to their default value as listed in the description of the input variables in this report. The plots for each example case show the ice shape for each time step, the ice thickness distribution for each time step, the flow solution for the clean airfoil and the final ice shape, the heat transfer coefficient for the clean airfoil and next-to-last ice shape and the collection efficiency for the clean airfoil and the next-to-last ice shape. The ice thickness plots are the result of plotting the data in the "thick.dat" output file, not the thickness listed in the ice shape file. The difference between the two outputs is the definition of wrap distance as defined for those files. Refer to the description of these output files for further explanation. As can be seen from the print flags in the main input files, many of the other output files were not produced. The heat transfer coefficient and collection efficiency plots show the solution on the next-to-last ice shape since the code stops after calculating the ice shape for the last time step.

13.1 Case 1: Run 072501 from Validation Report

Computation Time: Pentium II 400MHz, 1min. 46 sec.; Pentium 133MHz, 8 min. 34 sec.

Disk Space: 942 KB

The first example case is run number 072501 from the validation report. This example case is illustrative of the tests performed for validation against the experimental data. The run number corresponds to the designation used by the test engineer. It is a six minute glaze ice accretion and was the first benchmark case used to assess cross-platform variability and CPU usage in the validation report. The main input file is listed below. The main input file and geometry input file are provided on the distribution disk as “case1.inp” and “case1.xyd” respectively.

Table 12: Main Input File for Example Case 1

```
Test Case
&LEW20
TSTOP  =  360.
IBOD   =    1
IFLO   =   6
DSMN   =    4.0D-4
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

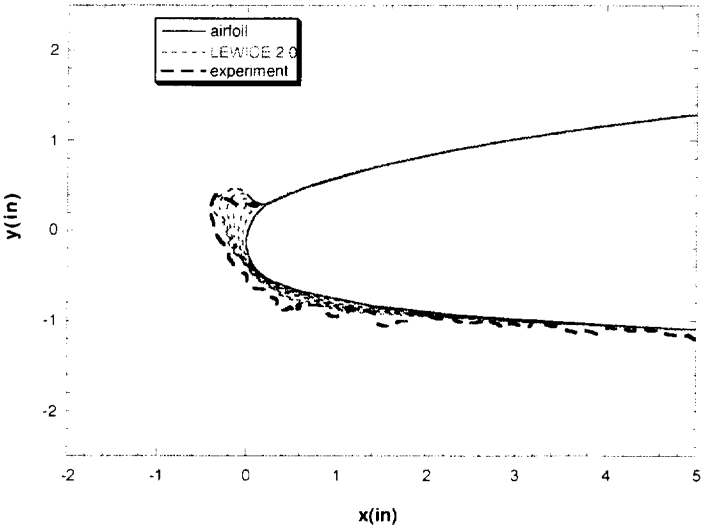

Figure 4: Ice Shape for Example Case 1

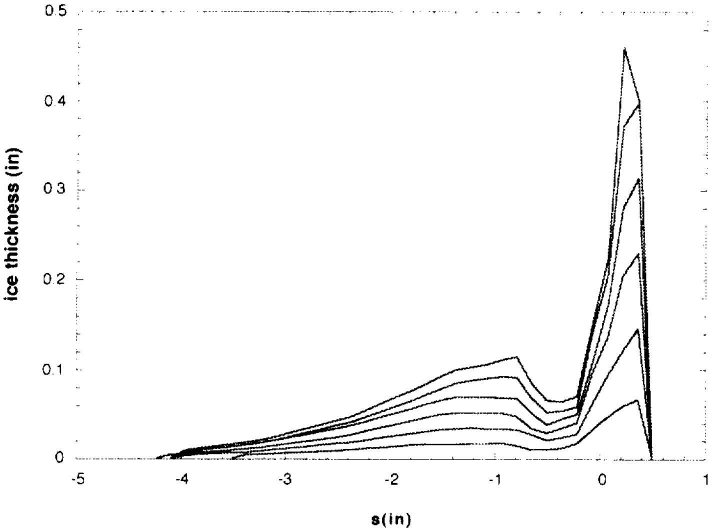

Figure 5: Ice Thickness for Example Case 1

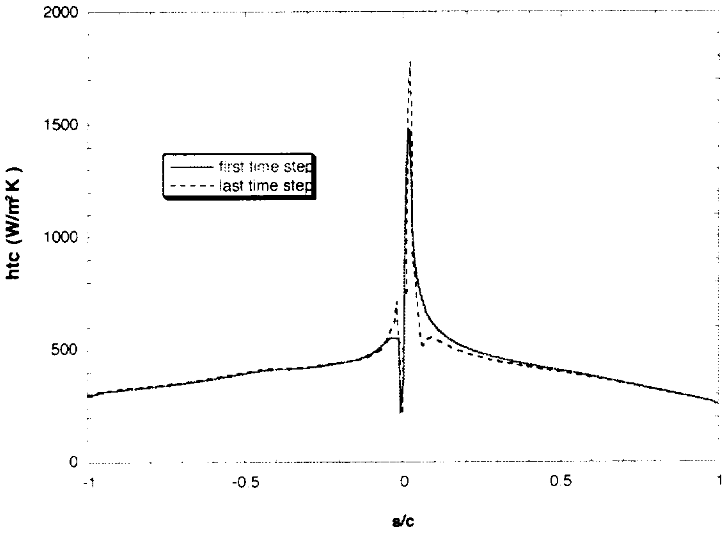

Figure 6: Heat Transfer Coefficient for Example Case 1

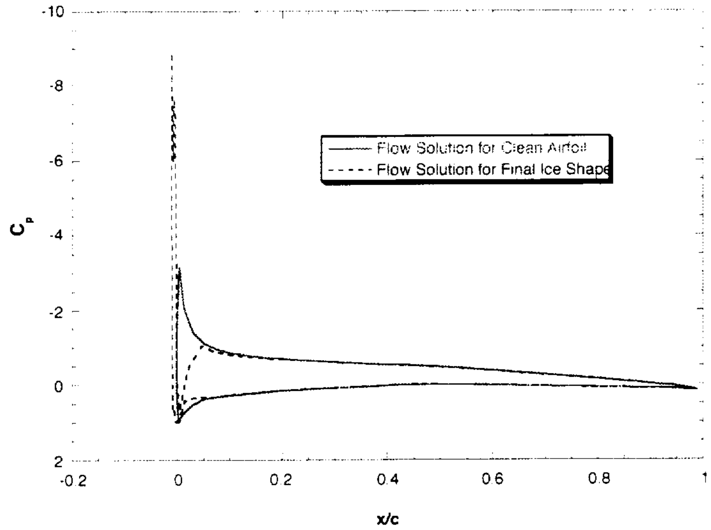

Figure 7: Pressure Coefficient for Example Case 1

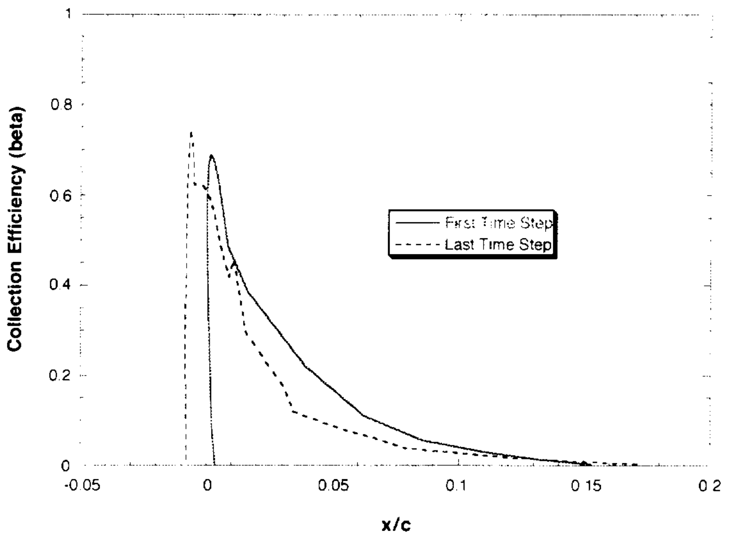

Figure 8: Collection Efficiency for Example Case 1

13.2 Case 2: Run 072504 from Validation Report

Computation Time: Pentium II 400MHz, 5 min. 44 sec.; Pentium 133MHz, 30 min.

Disk Space: 2.67 MB

This example case is taken from Run 072504 from the validation report. It is the second case used in that report for benchmarking results on different platforms. The case is a 45 minute glaze ice condition. This is representative of a long icing time which might be run for certification purposes. The main input file for this case is listed in Table 13. The main input file and geometry input file are provided on the distribution disk as "case2.inp" and "case2.xyd" respectively. Note that the time step selected in the input file is overridden during the run due to the use of the automatic time step flag. The user should confirm this when the warning message appears on the screen.

Table 13: Main Input File for Example Case 2

```
Test Case
&LEW20
TSTOP  =  2700.
IBOD   =    1
IFLO   =   15
DSMN   =    4.0D-4
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

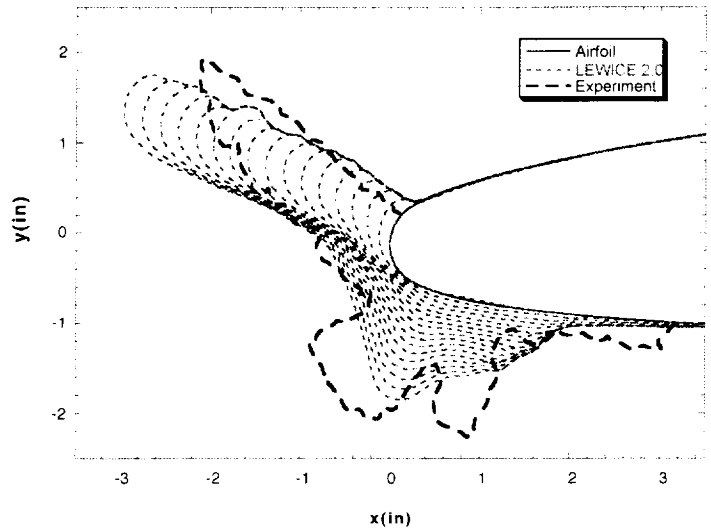

Figure 9: Ice Shape for Example Case 2

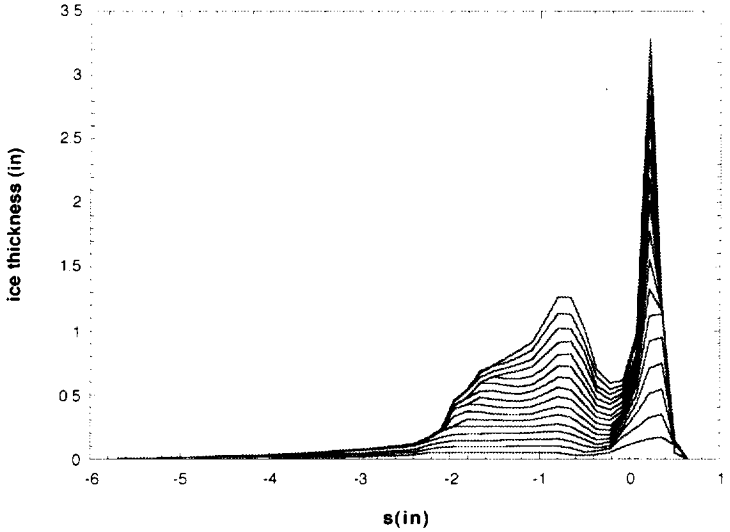

Figure 10: Ice Thickness for Example Case 2

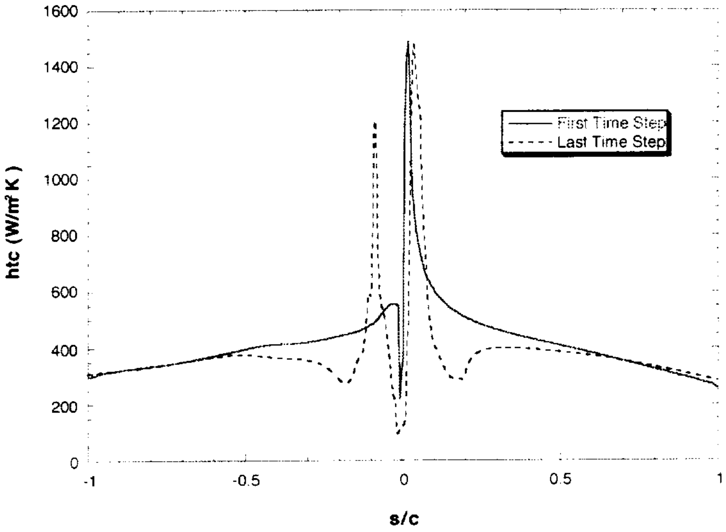

Figure 11: Heat Transfer Coefficient for Example Case 2

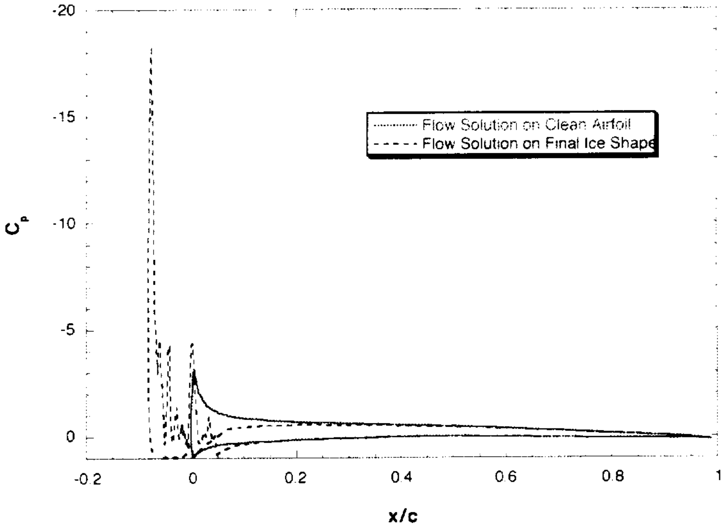

Figure 12: Pressure Coefficient for Example Case 2

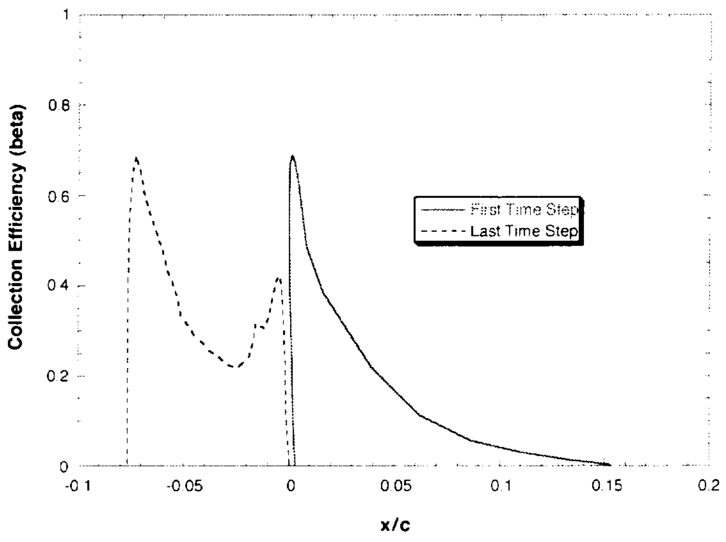

Figure 13: Collection Efficiency for Example Case 2

13.3 Case 3: Cylinder Benchmark Case

Computation Time: Pentium II 400MHz, 16 min. 41 sec.; Pentium 133MHz, 92 min.

Disk Space: 3.86 MB

This example case uses a 6" diameter cylinder for the geometry input. This example was included in the validation report as benchmark case number 16. It is a much longer case computationally and as such is not well suited for slower machines as the times above illustrate. The slow execution time is due to the relative size of the ice shape to the cylinder. Slower execution times can be expected for any small model, not just cylinders. The small chord size also causes the program use a large number of time steps for this case if the automated time-step flag (ITIMFL) is on. The automatic time step flag was not used for this case as the computation time would be prohibitive. The cylinder case imposes a further constraint due the high ratio of wrap distance to chord length. This ratio necessitates the use of a larger DSMN value than desired due to array size limitations. The main input file for this case is listed in Table 14. The main input file and geometry input file are provided on the distribution disk as "case3.inp" and "case3.xyd" respectively. Note that the program will warn the user that the autotime step flag is not being used. The user should confirm this when the warning message appears on the screen. The effect of the compromises made to this case can be seen in the ice shape output as the ice shape on the cylinder is not symmetrical. As such, this case would not be appropriate for comparison with experimental data if such data exists.

Table 14: Main Input File for Example Case 3

```
Test Case
&LEW20
ITIMFL =  0
TSTOP  =  2700.
IBOD   =  1
IFLO   =  15
DSMN   =  0.0008
NPL    =  24
RHOP   =  1000.
IGRID  =  0
IBOE   =  0
IDEICE =  0
&END
&DIST
FLWC   =  1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
DPD    =  20.,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
&END
&ICE1
CHORD  =  0.1524
AOA    =  0.0
VINF   =  90.
LWC    =  0.540
TINF   =  268.30
PINF   =  100000.00
RH     =  100.0
&END
```

```
&LPRNT
FPRT    =  1
HPRT    =  1
BPRT    =  1
EPRT    =  0
MPRT    =  0
TPRT    =  0
IDBF    =  1
&END
```

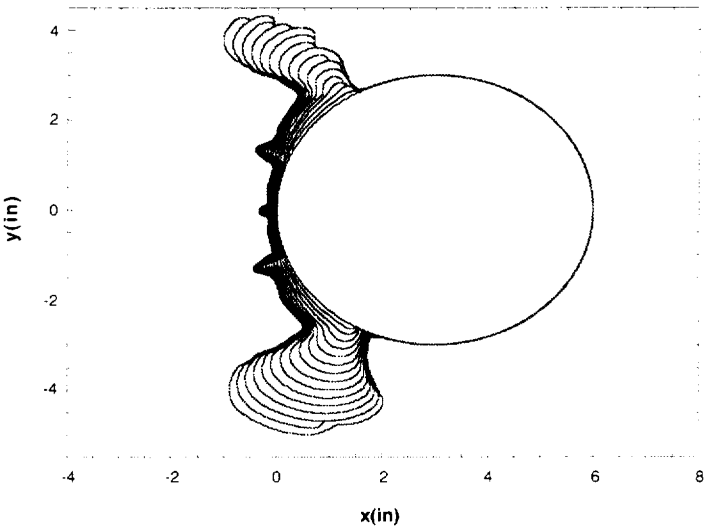

Figure 14: Ice Shape for Example Case 3

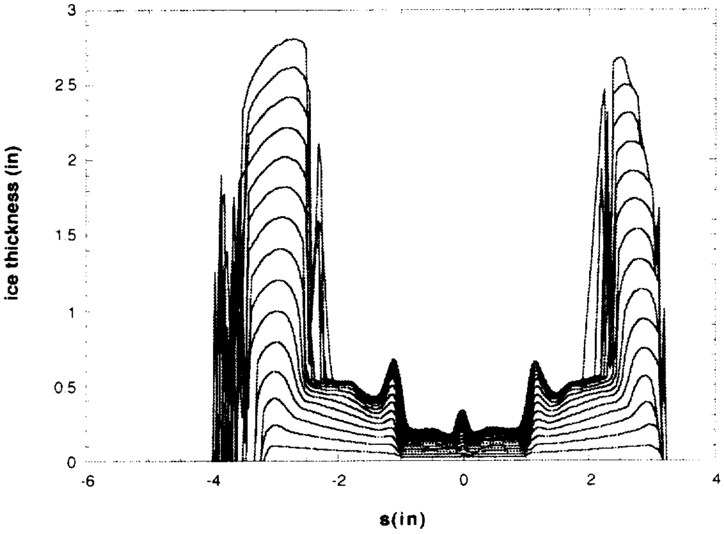

Figure 15: Ice Thickness for Example Case 3

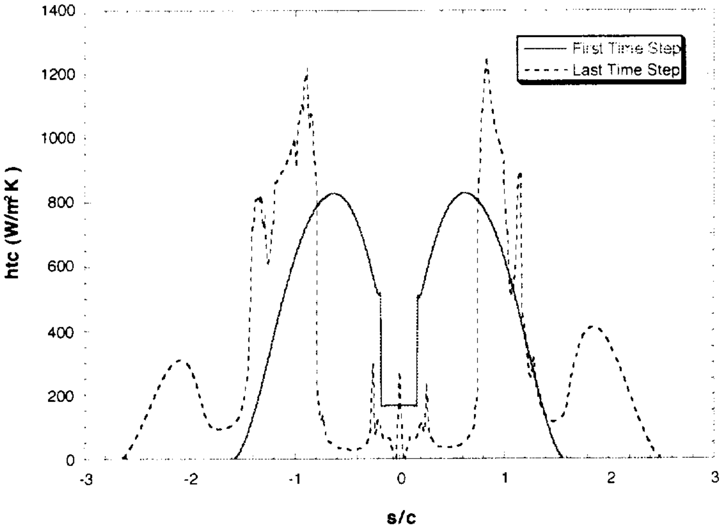

Figure 16: Heat Transfer Coefficient for Example Case 3

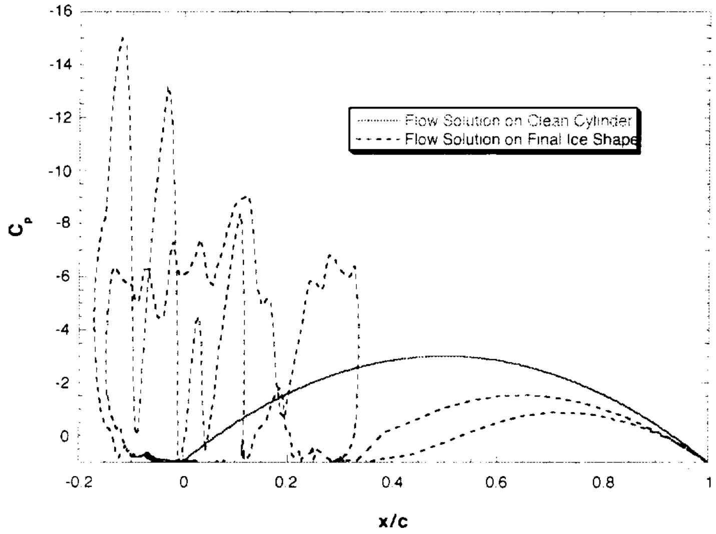

Figure 17: Pressure Coefficient for Example Case 3

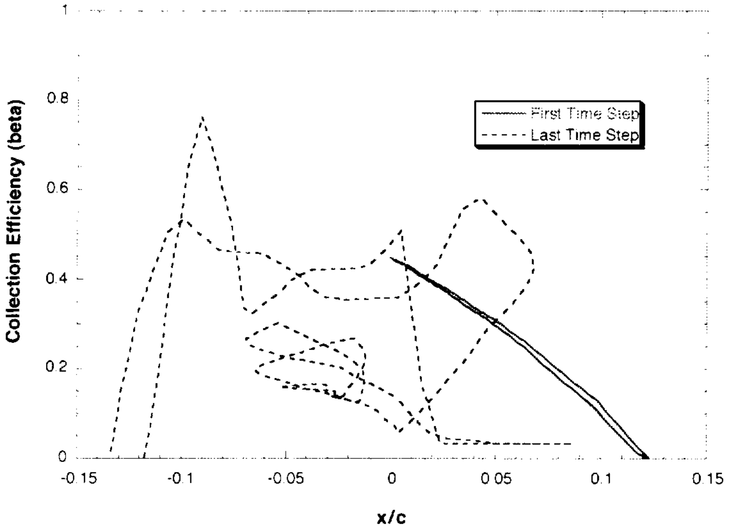

Figure 18: Collection Efficiency for Example Case 3

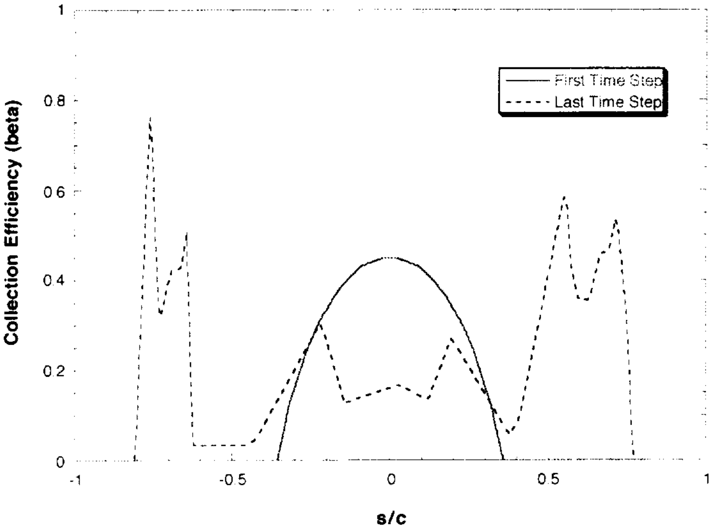

Figure 19: Collection Efficiency as a Function of Wrap for Example Case 3

The plots for collection efficiency and pressure coefficient are quite complex due the complexity of the ice shape. For this case, it is more appropriate to plot collection efficiency as a function of wrap instead of x-distance. This is shown in Figure 19.

The wrap distance plotted in Figure 19 was the wrap distance from stagnation, not the wrap distance from the leading edge. Both wrap distances are listed in the output file.

13.4 Case 4: Langmuir 'D' distribution

Computation Time: Pentium II 400MHz, 3 min. 42 sec.; Pentium 133MHz, 21 min. 22 sec.

Disk Space: 993 KB

This case shows an example using a drop size distribution. The case was originally run as part of the Numerical Effects section in the validation report. That section showed the qualitative difference in ice shape and ice thickness by using a droplet distribution rather than a monodispersed drop size as was the case for the main validation runs. The main input file is listed in Table 15. The main input file and geometry input file are provided on the distribution disk as "case4.inp" and "case4.xyd" respectively.

Table 15: Main Input File for Example Case 4

```
Test Case
&LEW20
TSTOP  =  360.
IBOD   =    1
IFLO   =   6
DSMN   =    4.0D-4
NPL    =   24
&END
&DIST
FLWC   = 0.05,  0.1,  0.2, 0.3,  0.2,  0.1, 0.05
DPD    =  6.2, 10.4, 14.2, 20., 27.4, 34.8, 44.4
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

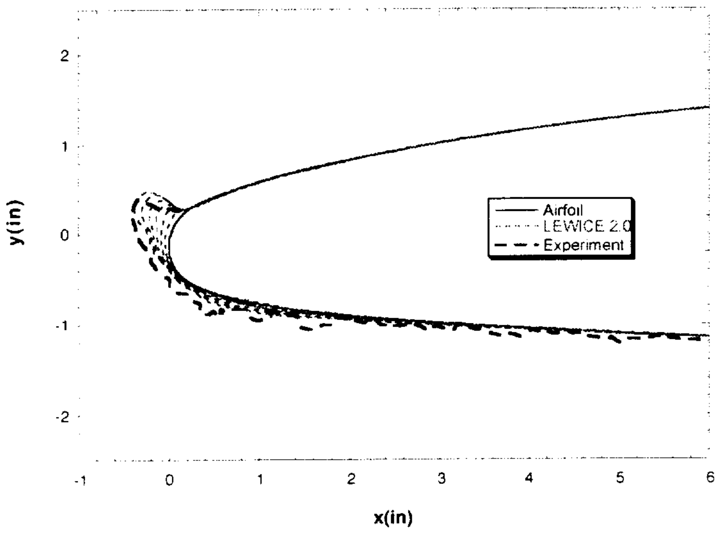

Figure 20: Ice Shape for Example Case 4

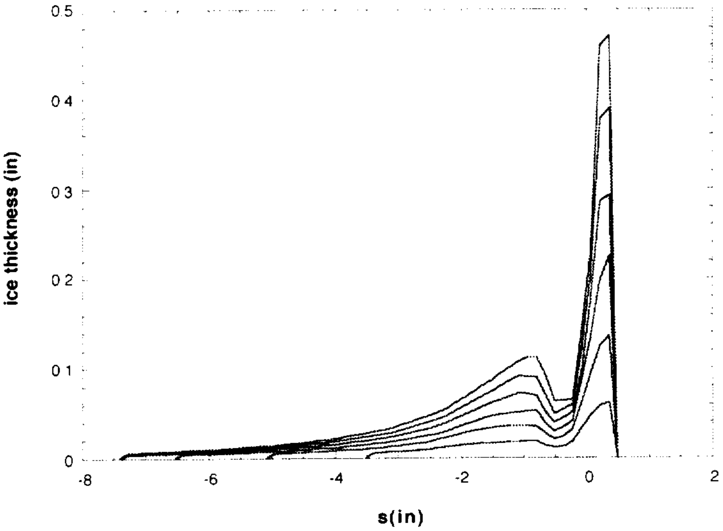

Figure 21: Ice Thickness for Example Case 4

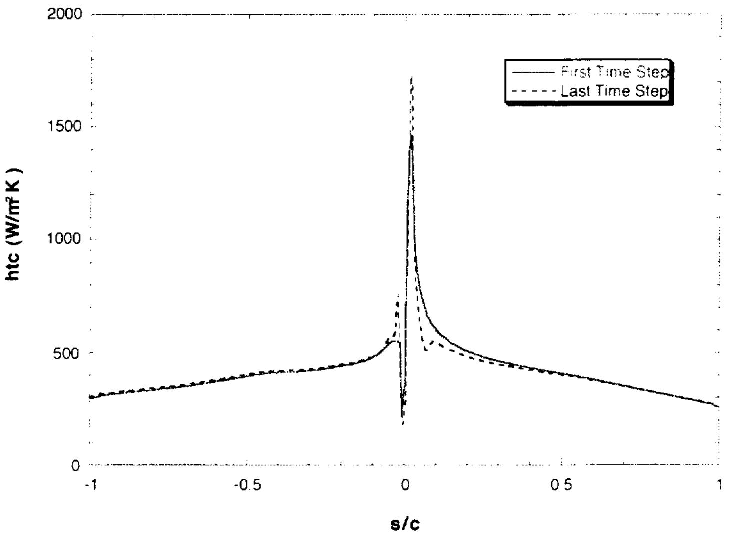

Figure 22: Heat Transfer Coefficient for Example Case 4

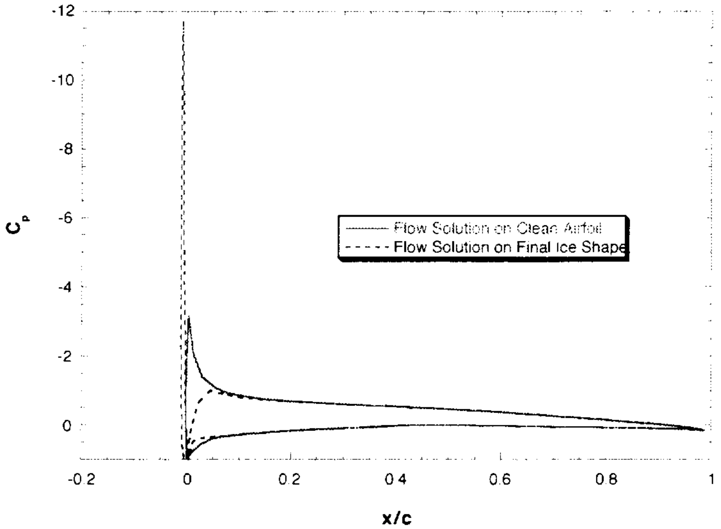

Figure 23: Pressure Coefficient for Example Case 4

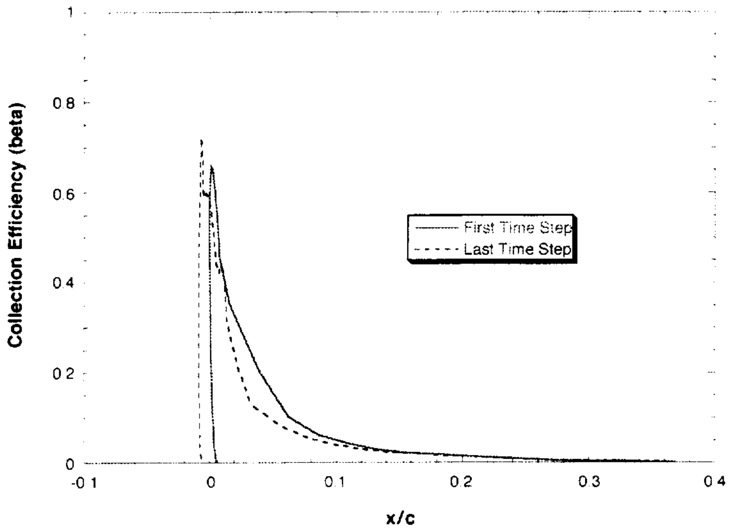

Figure 24: Collection Efficiency for Example Case 4

13.5 Case 5: Exceedence condition

Computation Time: Pentium II 400MHz, 1 min. 40 sec.; Pentium 133MHz, 8 min. 29 sec.

Disk Space: 1.08 MB

This case shows an example using an exceedence condition as input. This condition is Run DC-2 from the NACA4415(mod) database in the validation report[1]. When this case is run, a warning message will appear to indicate that the condition is outside of the FAA certification envelope. This warning must be confirmed by the user for this case to run.

There are some interesting features to note on this ice accretion. First, the ice shape from the IRT is much rougher than the one calculated by LEWICE, even more so than ice shapes generated within the FAA Appendix C envelope. Second, the lower surface icing limit for the experiment extends about 5 inches (6% chord) past the end of the ice shape predicted by LEWICE. This is a curious result as water collection tests have shown the opposite trend concerning the prediction of impingement limits[12]. It should be noted though that the experimental data for lower icing limit for this particular condition ranges from 7.3 inches (9.4% chord) to 13.3 inches (17% chord) while the predicted lower icing limit from LEWICE is 7.9 inches (10% chord) using a monodispersed drop size. Granted, there is very little ice on the ice shape from the experimental data in this region, but it exists nevertheless.

This example illustrates that it is the ice accretion limit which is the important parameter to consider and not simply the water collection limit (they are not the same!). Indeed, the collection efficiency prediction for this case shows water collection past 14% chord for this case while the icing limit is closer to 10% chord. Another interesting observation is that although the condition indicates a glaze ice condition, there is not a true glaze ice "horn" for this case. This observation can be seen throughout the validation database for many of the exceedence conditions, even for longer exposure times. The higher water collection rate for these drop sizes tends to distribute the ice more evenly throughout the impingement region. However, the conclusions listed in these observations may be premature due to the scarcity of data in this regime. Table 16 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as "case5.inp" and "case5.xyd" respectively.

Table 16: Main Input File for Example Case 5

```
Test Case
&LEW20
TSTOP  =  420.
IBOD   =    1
IFLO   =    7
DSMN   =    4.0D-4
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 160.0
&END
```

```
&ICE1
CHORD   = 1.9812
AOA     = 0.0
VINF    = 87.2
LWC     = 0.82
TINF    = 266.85
PINF    = 100000.00
RH      = 100.0
&END
&LPRNT
FPRT    = 1
HPRT    = 1
BPRT    = 1
TPRT    = 0
&END
```

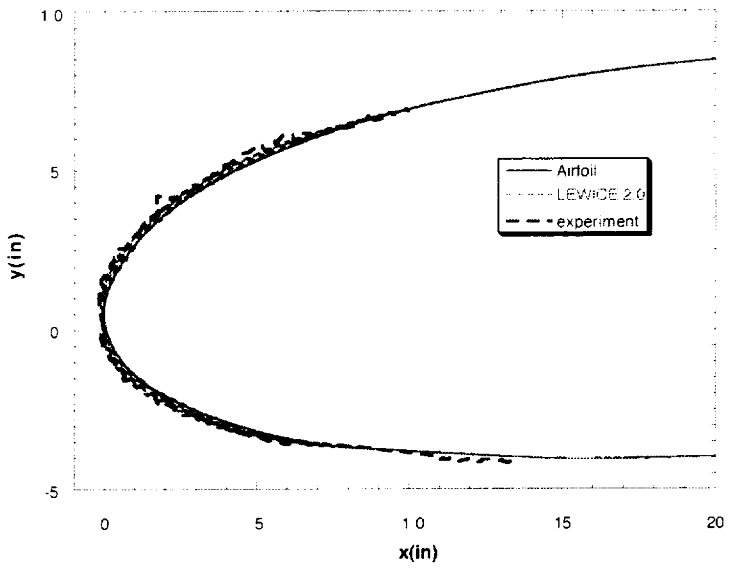

Figure 25: Ice Shape for Example Case 5

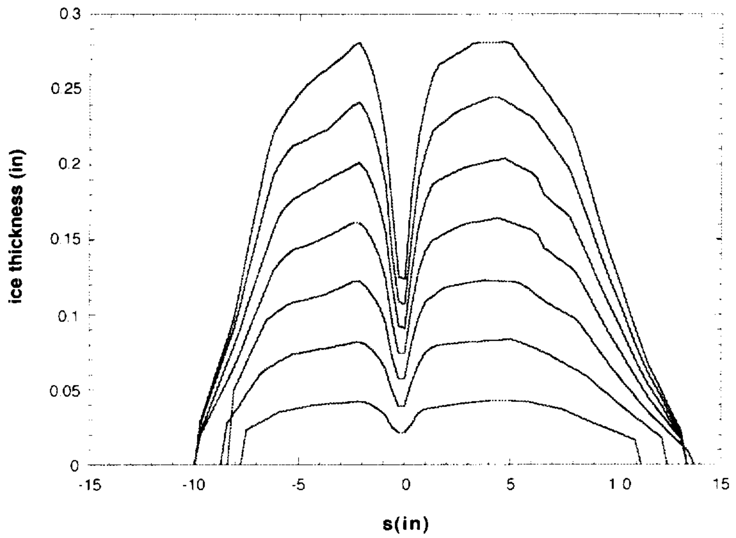

Figure 26: Ice Thickness for Example Case 5

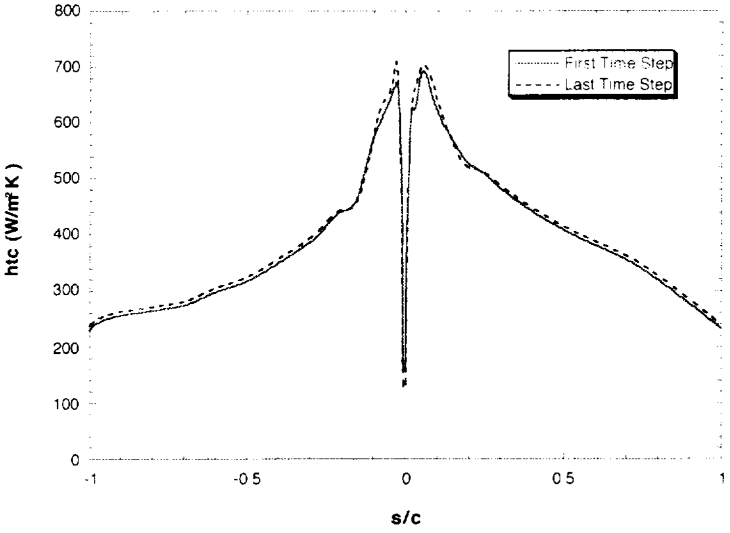

Figure 27: Heat Transfer Coefficient for Example Case 5

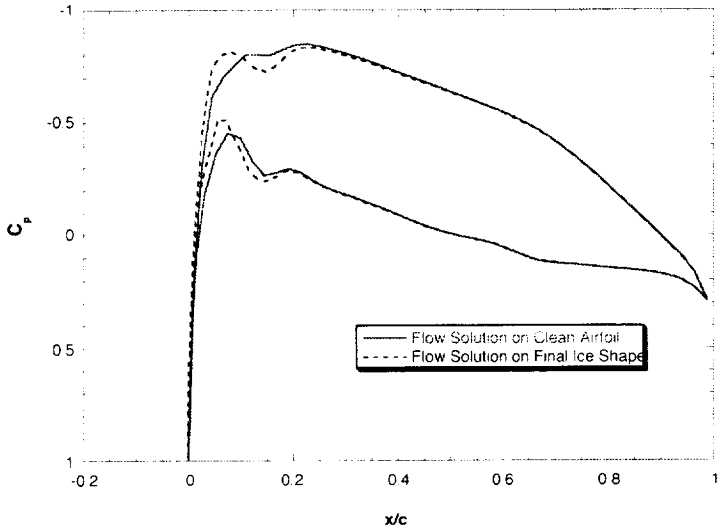

Figure 28: Flow Solution for Example Case 5

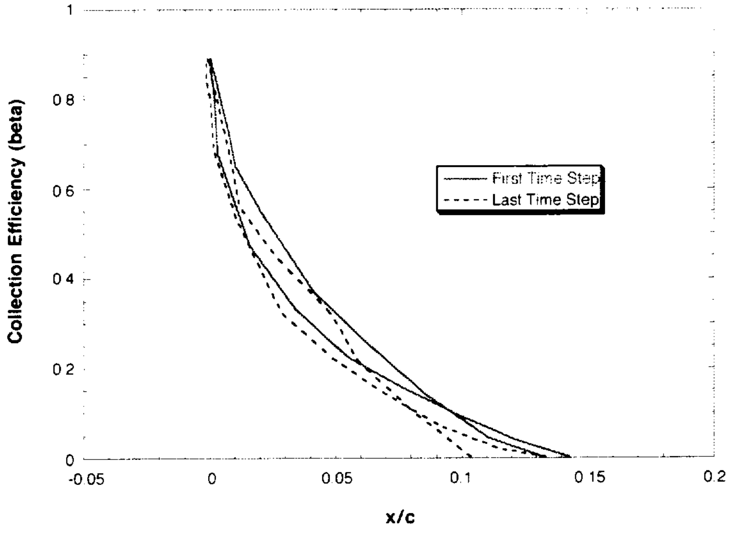

Figure 29: Collection Efficiency for Example Case 5

13.6 Case 6: Exceedence condition w/Langmuir 'D' distribution

Computation Time: Pentium II 400MHz, 3 min. 10 sec.; Pentium 133MHz, 17 min. 43 sec.

Disk Space: 1.14 MB

This example case uses the same input as the previous example, except that a Langmuir 'D' drop size distribution is used instead of a monodispersed drop size. This example was provided to show the effect of a drop size distribution for an exceedence condition. As with the previous example, the code will print a warning concerning the input drop size. The user will have to confirm the warning to continue the case. The collection efficiency plot for this example shows a greater extent of water collection than the previous example as expected. However, the ice shape comparison for this case is virtually identical to the previous example, as a comparison of the two ice shapes shows.

A plot of the ice thickness distribution for these two cases shows that the Langmuir 'D' case extends the ice by 1 inches on the lower surface and 3 inches on the upper surface. However the panel spacing is very course in this region for both cases. Whether this quantitative difference is due more to the use of a drop size distribution or whether it is due to the course spacing is unknown. The algorithms in LEWICE will place a higher density of points in regions of high curvature. This results in higher accuracy in regions where the ice thickness is greatest. Near the icing limits, the surface curvature is flat, reducing the accuracy of the result. Table 17 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as "case6.inp" and "case6.xyd" respectively.

Table 17: Main Input File for Example Case 6

```
Test Case
&LEW20
TSTOP   =  420.
IBOD    =    1
IFLO    =    7
DSMN    =    4.0D-4
NPL     =   24
&END
&DIST
FLWC    = 0.05,  0.1,   0.2,  0.3,   0.2,   0.1, 0.05
DPD     = 49.6, 83.2, 113.6, 160., 219.2, 278.4, 355.2
&END
&ICE1
CHORD   = 1.9812
AOA     = 0.0
VINF    = 87.2
LWC     = 0.82
TINF    = 266.85
PINF    = 100000.00
RH      = 100.0
&END
&LPRNT
FPRT    = 1
HPRT    = 1
BPRT    = 1
TPRT    = 0
&END
```

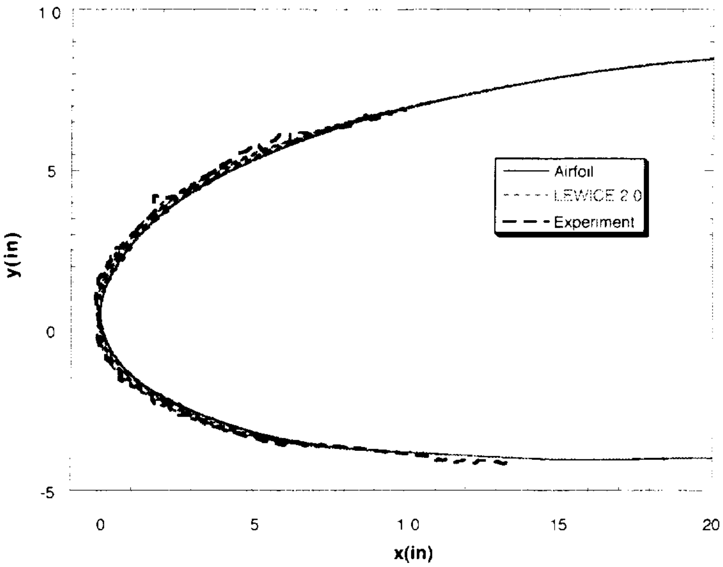

Figure 30: Ice Shape for Example Case 6

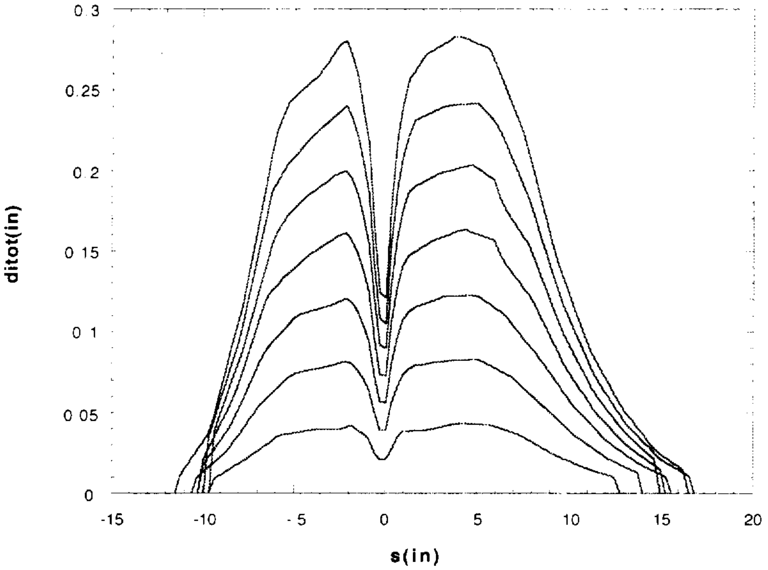

Figure 31: Ice Thickness for Example Case 6

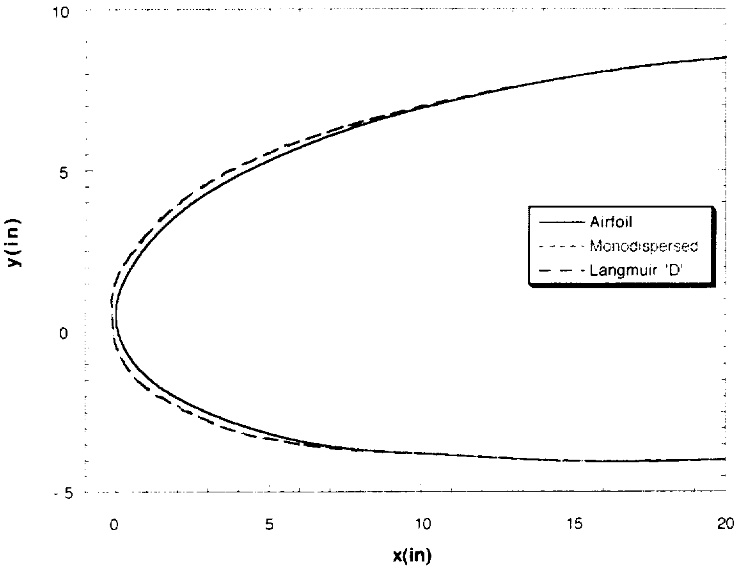

Figure 32: Comparison of Ice Shapes for Monodispersed and Langmuir 'D' Cases

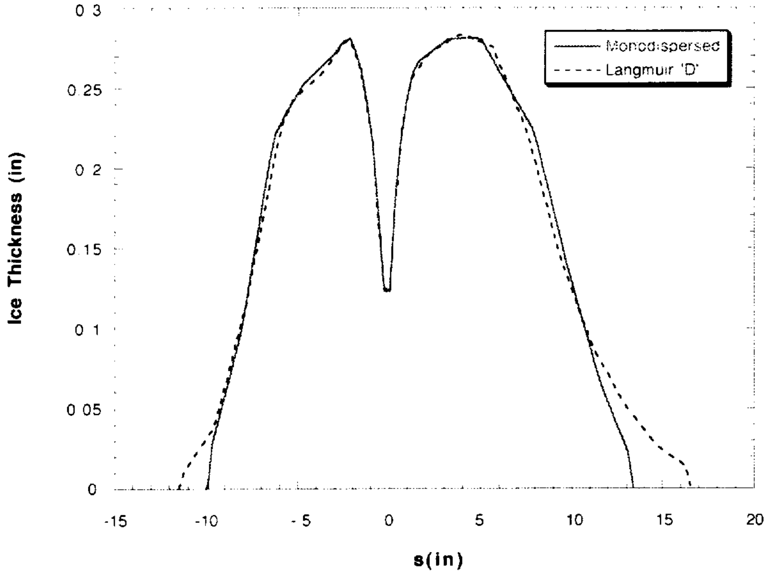

Figure 33: Comparison of Ice Thicknesses for Monodispersed and Langmuir 'D' Cases

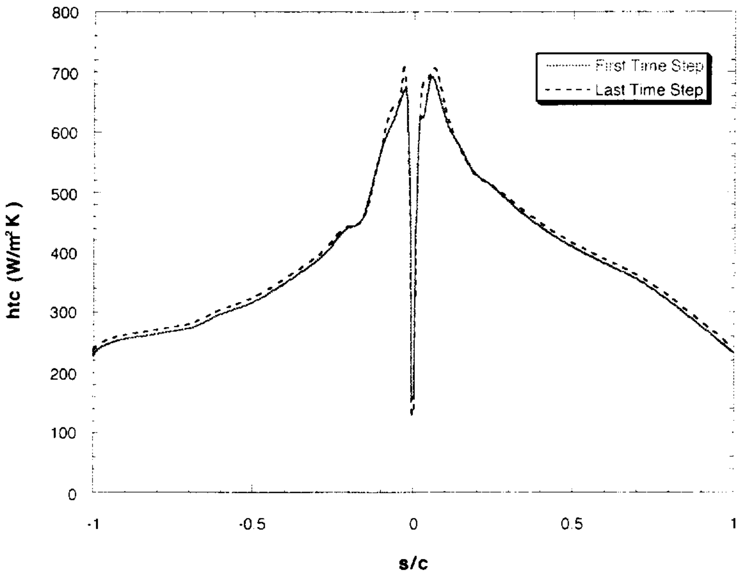

Figure 34: Heat Transfer Coefficient for Example Case 6

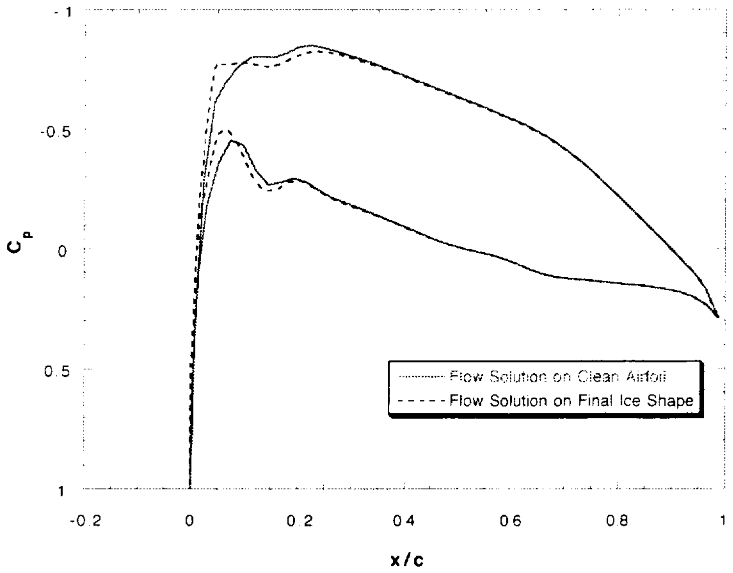

Figure 35: Pressure Coefficient for Example Case 6

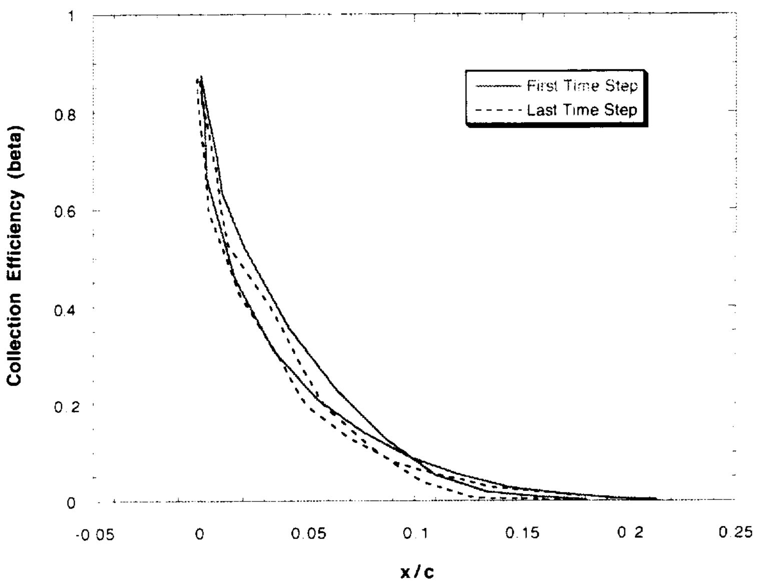

Figure 36: Collection Efficiency for Example Case 6

13.7 Case 7: Run 072504 without automated time step and fewer time steps

Computation Time: Pentium II 400MHz, 1 min. 9 sec.; Pentium 133MHz, 6 min. 15 sec.

Disk Space: 556 KB

This example case illustrates the problems which can occur when the user bypasses the recommended operating procedure for LEWICE. In this case, the automatic time step flag has been set to **off** (ITIMFL = 0) and the number of time steps has been reduced to three. Otherwise, the example case uses the same inputs as Example Case 2. Warning messages will be issued for this case to inform the user that the guidelines are not being followed. This message must be confirmed by the user to run this case. Since the current example case uses fewer time steps, it runs much faster than Example Case 2. However, an examination of the ice shape prediction shows the pitfalls of this approach. Because the ice shape is not allowed to progress in time relative of the proper ice accretion physics, the glaze ice horn which develops is not as large as the experimental ice shape or the prediction from Example Case 2. Also note that there is no development of a distinctive lower horn which was observed in Case 2. Table 18 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as “case7.inp” and “case7.xyd” respectively.

Table 18: Main Input File for Example Case 7

```
Test Case
&LEW20
ITIMFL = 0
TSTOP  =  2700.
IBOD   =    1
IFLO   =    3
DSMN   =    4.0D-4
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

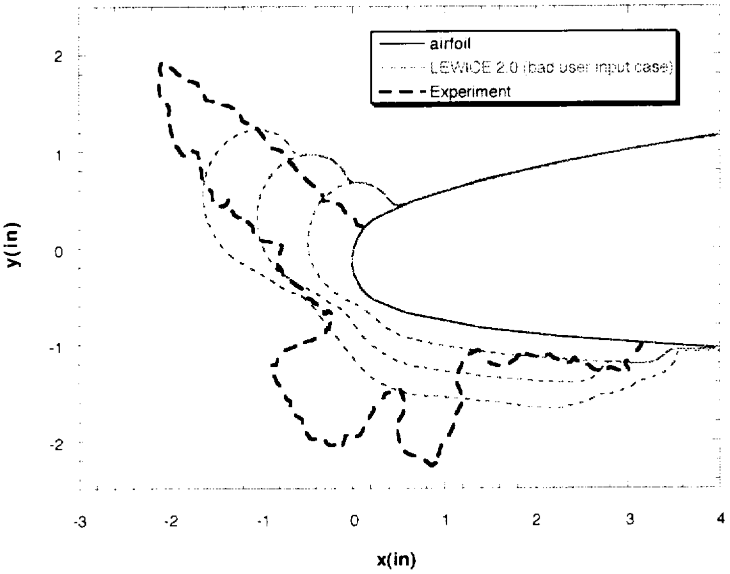

Figure 37: Ice Shape for Example Case 7

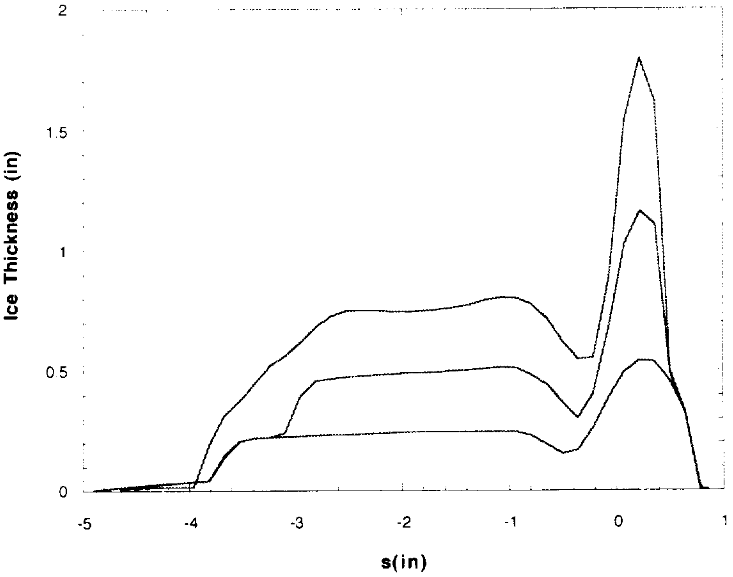

Figure 38: Ice Thickness for Example Case 7

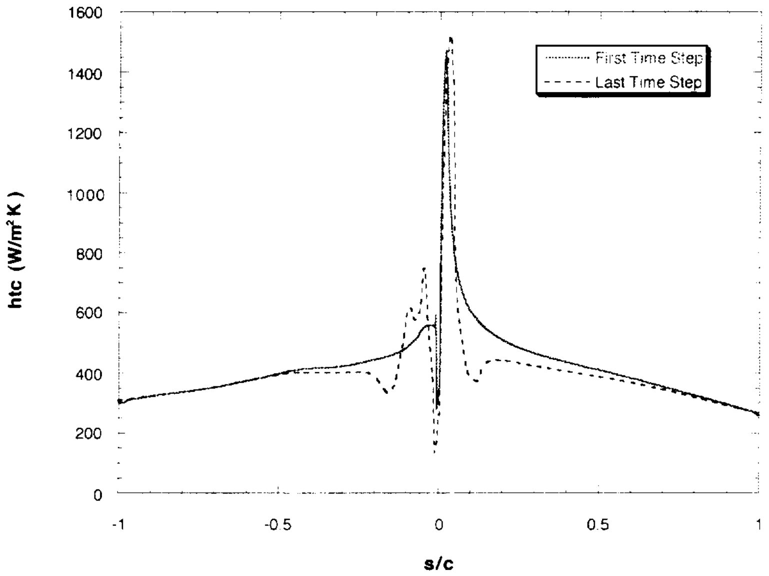

Figure 39: Heat Transfer Coefficient for Example Case 7

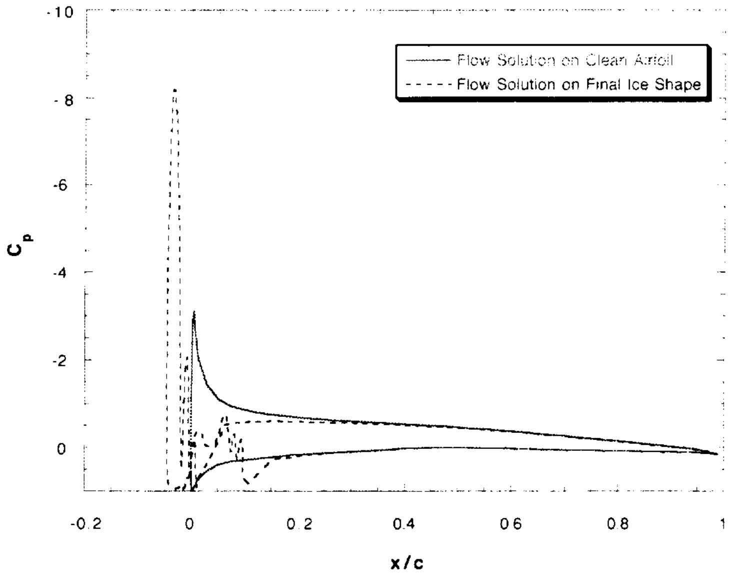

Figure 40: Pressure Coefficient for Example Case 7

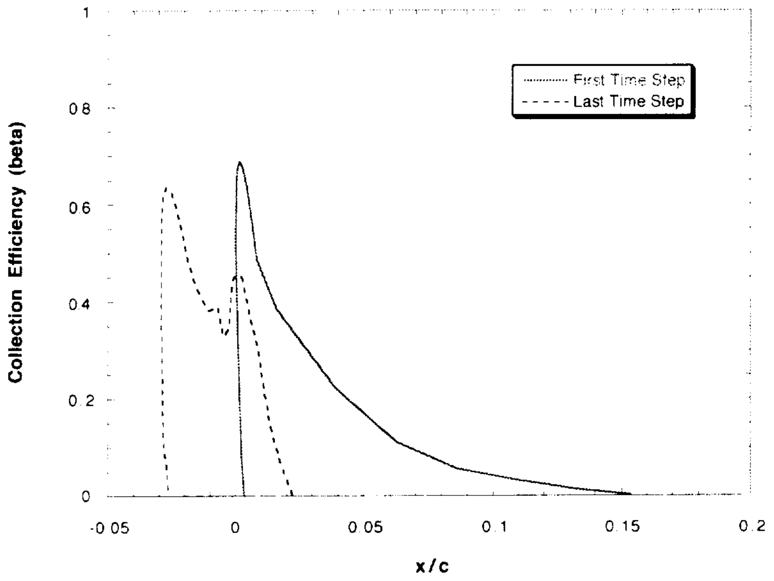

Figure 41: Collection Efficiency for Example Case 7

13.8 Case 8: Run 072504 with Larger Point Spacing

Computation Time: Pentium II 400MHz, 1 min. 9 sec.; Pentium 133MHz, 6 min. 38 sec.

Disk Space: 1.07 MB

As with the previous example, this example case illustrates the problems which can occur when the user bypasses the recommended operating procedure for LEWICE. In this case, the point spacing has been increased past the recommended limits. Otherwise, the inputs for this case also correspond to Example Case 2. A warning will be issued by the program when this example case is run due to the large point spacing. The warning message must be confirmed by the user to run this case. Since the point spacing is much sparser than Case 2, the program also runs much faster, again at some cost to the accuracy of the solution. The effect on the ice shape prediction due to the increased point spacing is not as great for this case as the time step effect, but the effects are still noticeable. The main effect on the ice shape prediction for this case is a change in the predicted horn angle. The upper horn tends to "droop" for this case whereas the original case predicted a more regular horn angle throughout the run. This "drooping" effect on glaze ice horns is typical of a case with poor numerics. The user should increase the number of points (decrease DSMN) for this case to counter this effect. Table 19 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as "case8.inp" and "case8.xyd" respectively.

Table 19: Main Input File for Example Case 8

```
Test Case
&LEW20
TSTOP  =  2700.
IBOD   =    1
IFLO   =   15
DSMN   =    1.2D-3
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

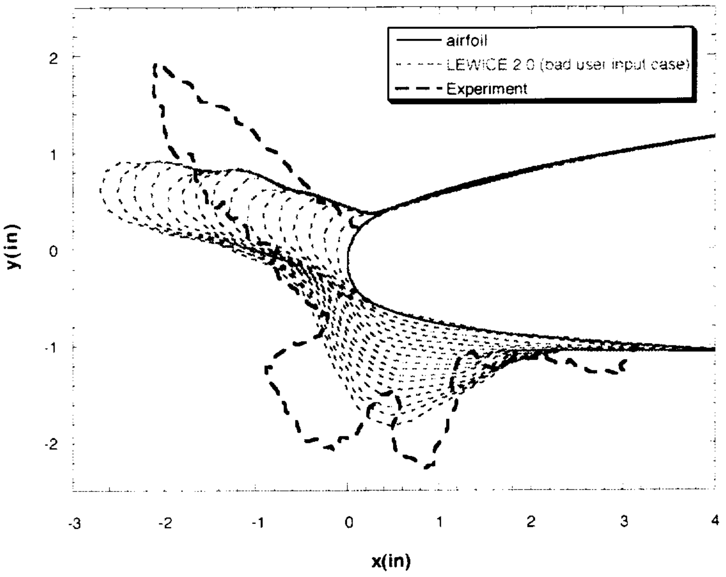

Figure 42: Ice Shape for Example Case 8

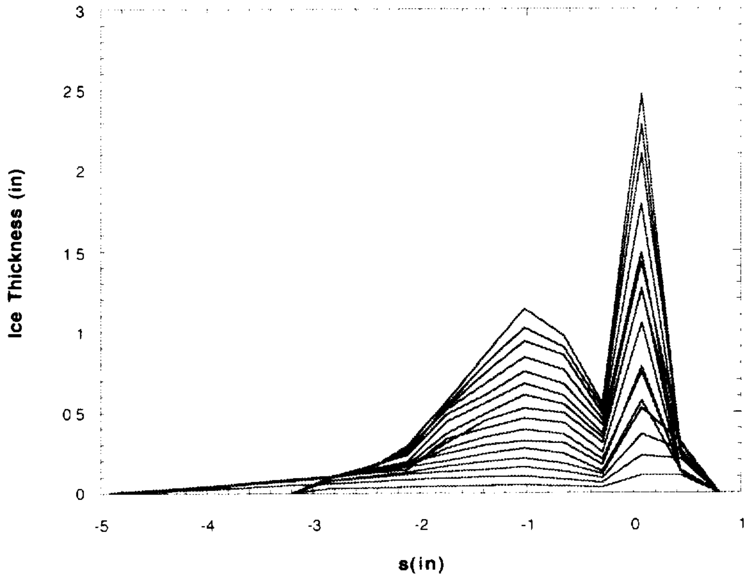

Figure 43: Ice Thickness for Example Case 8

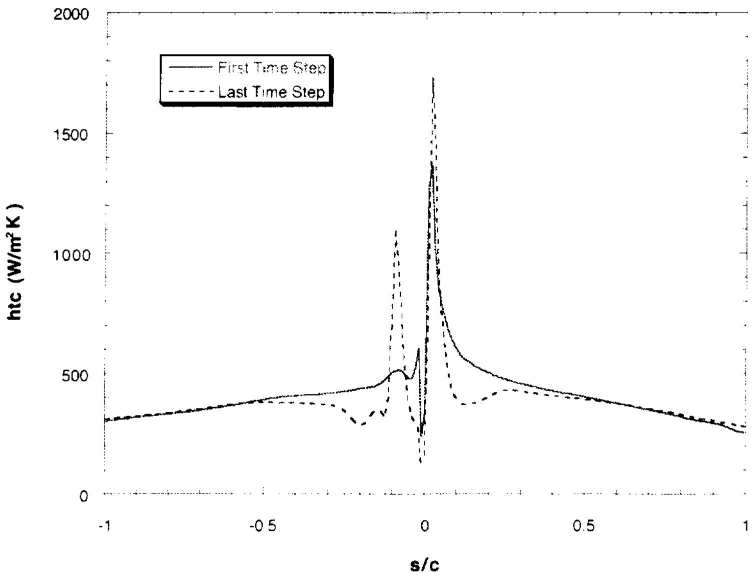

Figure 44: Heat Transfer Coefficient for Example Case 8

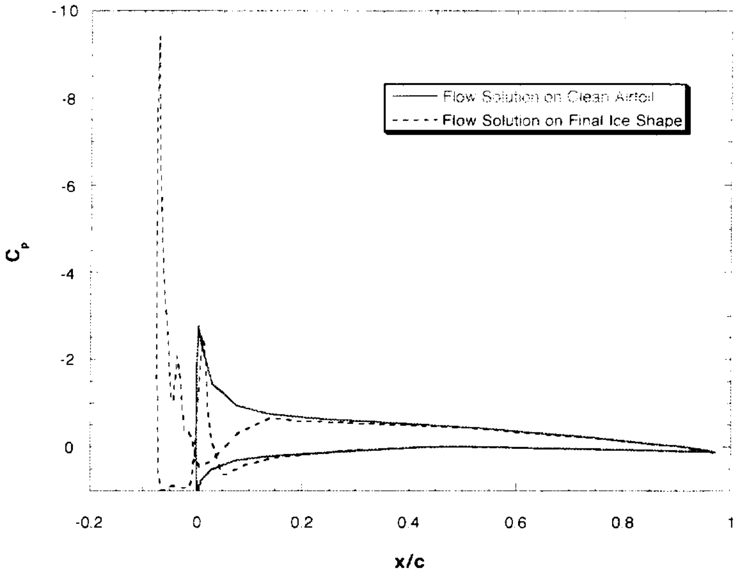

Figure 45: Pressure Coefficient for Example Case 8

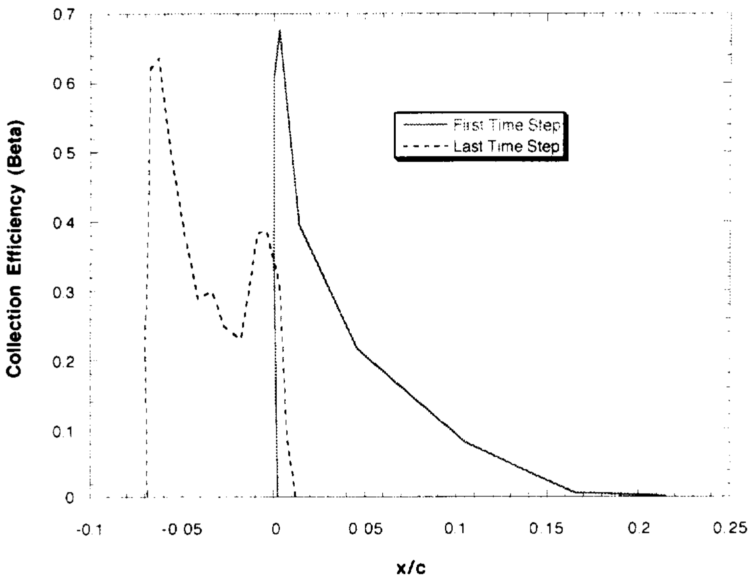

Figure 46: Collection Efficiency for Example Case 8

13.9 Case 9: First benchmark conditions with an evaporative hot air anti-icer.

Computation Time: Pentium II 400MHz, 16 sec.; Pentium 133MHz, 1 min. 18 sec.

Disk Space: 420 KB

This example case illustrates the use of the anti-ice capabilities of LEWICE. As the procedure used in LEWICE for anti-icing is less involved than the procedure used in ANTICE[9] or LEWICE/ Thermal[8] and since this feature has not been validated with experimental data, a warning will appear when this case is run. The user must confirm the warning message to continue the run. Also note that a second warning will be generated when the "deicei.inp" input file is read by LEWICE. This warning must also be confirmed to continue the run. The conditions for this case are the same as Example Case 1, except the simulated icing time has been reduced to 1 minute and the deicer flag has been set **on** (IDEICE=1). The use of a 1 minute icing time was arbitrary. The anti-ice solution provided is a steady-state solution so the choice of time step is irrelevant to that output. Since this example was created to illustrate the use of the anti-ice solution, a single time step was chosen for expediency.

Table 20 shows the main input file for this case while Table 21 shows the deice input file. The main input file and geometry input file are included on the distribution disks as “case9.inp” and “case9.xyd” respectively. The deice input file is listed on the distribution disk as “deice9.inp”. Its name must be changed to “deicei.inp” and placed in the same directory as the executable for this case to run. This case shows an example of an evaporative anti-icer using bleed air. Since the input flag states that all of the water must all evaporate, the wattages shown are very high. Wattages in this range should be expected for evaporative systems. The useful outputs from this example are the heat requirements and the surface temperature. As shown in Figure 49, the bleed air temperature solution will be inaccurate due to the simplistic assumptions used. The NASA Glenn code ANTICE[9] should be used to predict bleed air temperatures for this case. Other outputs for this case such as heat transfer coefficient, collection efficiency and pressure coefficient are the same as shown in the first example case and are not shown again here.

Table 20: Main Input File for Example Case 9

```
Test Case
&LEW20
ITIMFL = 0
TSTOP  =   60.
IBOD   =    1
IFLO   =   1
DSMN   =    4.0D-4
NPL    =   24
IDEICE =   1
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
```

```
VINF    =  90.
LWC     = 0.540
TINF    = 268.30
PINF    = 100000.00
RH      = 100.0
&END
&LPRNT
FPRT    = 1
HPRT    = 1
BPRT    = 1
TPRT    = 0
&END
```

Table 21: Deicei.inp input file for Example Case 9

```
DESIRED SURFACE TEMPERATURE              TSURF = 320.000  K
=1 FOR EVAPORATIVE, =0 FOR RUNNING WET   IEVAP = 1
=0 FOR ELECTROTHERMAL, =1 FOR HOT AIR   ITHERM = 1
NUMBER OF LAYERS = 01
       LENGTH        CONDUCTIVITY
        (M)             (W/M*K)
   1.7500000E-03    1.7653000E+02
    Layer in which heater resides        lheat = 1
    NUMBER OF HEAT TRANSFER COEFFICIENTS INPUT = 0027
       S/C            HTC(W/M*M*K)
  -1.0500000E+00    1.0000000E+01
  -0.4000000E+00    1.0000000E+01
  -0.3000000E+00    1.5000000E+01
  -0.2000000E+00    2.0000000E+01
  -0.1500000E+00    3.0000000E+01
  -0.1000000E+00    5.0000000E+01
  -0.0500000E+00    7.5000000E+01
  -0.0400000E+00    1.7500000E+02
  -0.0300000E+00    2.5000000E+02
  -0.0250000E+00    3.5000000E+02
  -0.0225000E+00    5.0000000E+02
  -0.0175000E+00    7.5000000E+02
```

```
-0.0125000E+00    8.7500000E+02
-0.0062500E+00    9.5000000E+02
 0.0000000E+00    1.0000000E+03
 0.0062500E+00    9.5000000E+02
 0.0125000E+00    8.7500000E+02
 0.0175000E+00    5.0000000E+02
 0.0225000E+00    3.0000000E+02
 0.0250000E+00    1.0000000E+02
 0.0500000E+00    7.5000000E+01
 0.1000000E+00    5.0000000E+01
 0.1500000E+00    3.0000000E+01
 0.2000000E+00    2.0000000E+01
 0.3000000E+00    1.5000000E+01
 0.4000000E+00    1.0000000E+01
 1.0500000E+00    1.0000000E+01
```

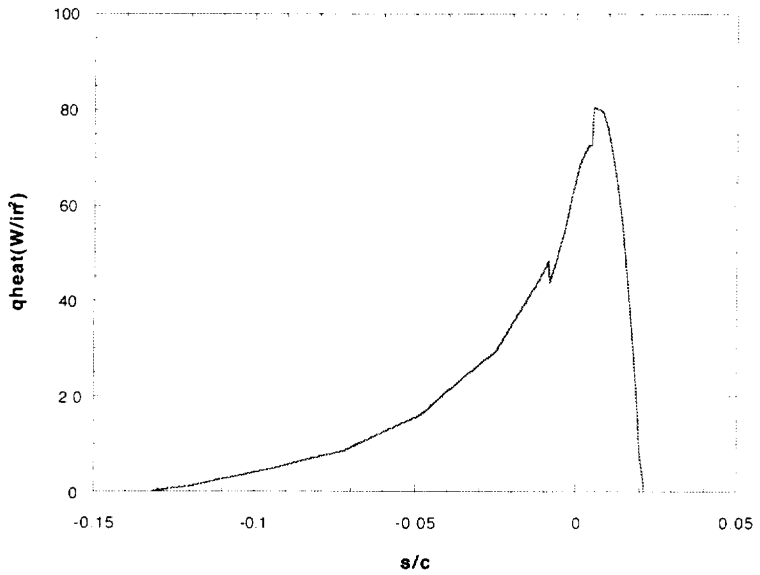

Figure 47: Heat Requirement for the Evaporative Hot Air System in Example Case 9

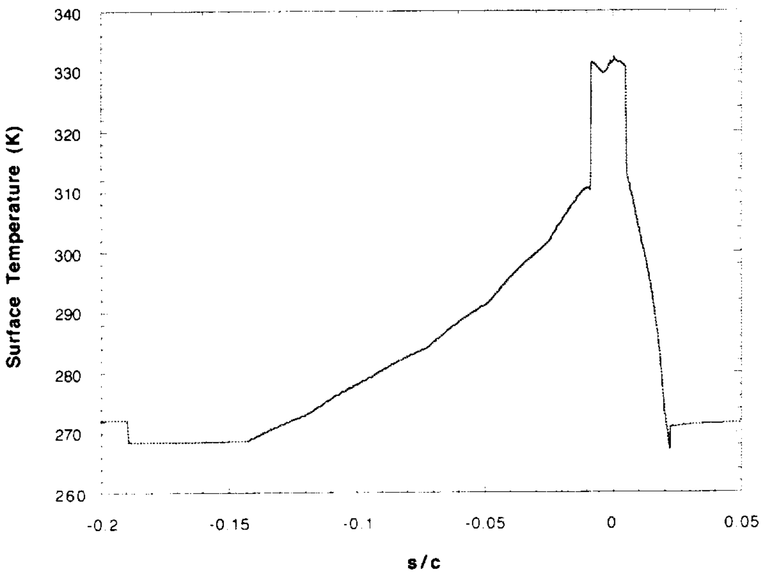

Figure 48: Surface Temperature for the Evaporative Hot Air System in Example Case 9

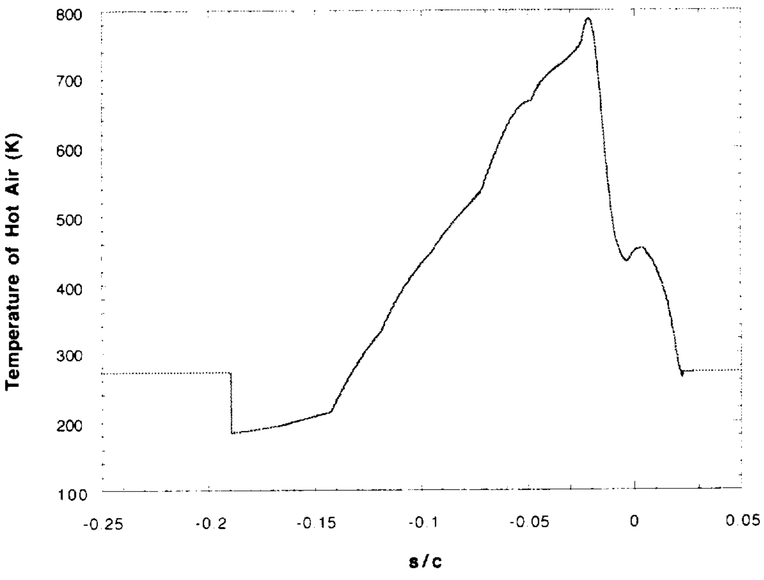

Figure 49: Hot Air Temperature for the Evaporative Hot Air System in Example Case 9

13.10 Case 10: First benchmark condition with a running wet electrothermal anti-icer.

Computation Time: Pentium II 400MHz, 15 sec.; Pentium 133MHz, 1 min. 19 sec.

Disk Space: 420 KB

Example Case 10 also shows an example using the deicer input file. The conditions for this case are the same as those shown for Example Case 9, except in this case a running wet thermal deicer is used instead of a hot air anti-icer as in the previous example. A warning is again issued to the user to indicate that the solution provided is only a first approximation and has not been validated. The warning must be confirmed by the user to run this case. Also note that a second warning will be generated when the "deicei.inp" input file is read by LEWICE. This warning must also be confirmed to continue the run. Table 22 shows the main input file for this case while Table 23 lists the deicer input file. The main input file and geometry input file are included on the distribution disks as "case10.inp" and "case10.xyd" respectively. The deice input file is listed on the distribution disk as "deice10.inp". Its name must be changed to "deicei.inp" to run this case.

The useful outputs for this case are the local heater wattages shown in Figure 50. For a running wet system, the assumption made is that the surface temperature is constant and is input by the user. Therefore, a plot of surface temperature is not provided. As was the case for the previous example, the heater temperatures plotted are inaccurate due to the simplistic assumptions used in order to achieve a fast solution. The NASA Glenn codes ANTICE[9] or LEWICE/Thermal[8] should be used to predict heater temperatures for this case.

Table 22: Main Input File for Example Case 10

```
Test Case
&LEW20
ITIMFL = 0
TSTOP  =   60.
IBOD   =    1
IFLO   =   1
DSMN   =    4.0D-4
NPL    =   24
IDEICE =   1
```

```
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  4.5
VINF   =  90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

Table 23: Deicei.inp Input File for Example Case 10

```
DESIRED SURFACE TEMPERATURE                TSURF = 283.150  K
=1 FOR EVAPORATIVE, =0 FOR RUNNING WET     IEVAP = 0
=0 FOR ELECTROTHERMAL, =1 FOR HOT AIR     ITHERM = 0
NUMBER OF LAYERS = 06
       LENGTH        CONDUCTIVITY
        (M)             (W/M*K)
   3.4300000E-03    0.1200000E+00
   8.9000000E-03    0.2940000E+00
   2.8000000E-03    0.2560000E+00
   1.3000000E-03    4.1000000E+01
   2.8000000E-03    0.2560000E+00
   2.0300000E-03    1.6270000E+01
    Layer in which heater resides        lheat = 4
    NUMBER OF HEAT TRANSFER COEFFICIENTS INPUT = 0027
       S/C             HTC(W/M*M*K)
  -1.0500000E+00    1.0000000E+01
```

```
-0.4000000E+00     1.0000000E+01
-0.3000000E+00     1.5000000E+01
-0.2000000E+00     2.0000000E+01
-0.1500000E+00     3.0000000E+01
-0.1000000E+00     5.0000000E+01
-0.0500000E+00     7.5000000E+01
-0.0400000E+00     1.7500000E+02
-0.0300000E+00     2.5000000E+02
-0.0250000E+00     3.5000000E+02
-0.0225000E+00     5.0000000E+02
-0.0175000E+00     7.5000000E+02
-0.0125000E+00     8.7500000E+02
-0.0062500E+00     9.5000000E+02
 0.0000000E+00     1.0000000E+03
 0.0062500E+00     9.5000000E+02
 0.0125000E+00     8.7500000E+02
 0.0175000E+00     5.0000000E+02
 0.0225000E+00     3.0000000E+02
 0.0250000E+00     1.0000000E+02
 0.0500000E+00     7.5000000E+01
 0.1000000E+00     5.0000000E+01
 0.1500000E+00     3.0000000E+01
 0.2000000E+00     2.0000000E+01
 0.3000000E+00     1.5000000E+01
 0.4000000E+00     1.0000000E+01
 1.0500000E+00     1.0000000E+01
```

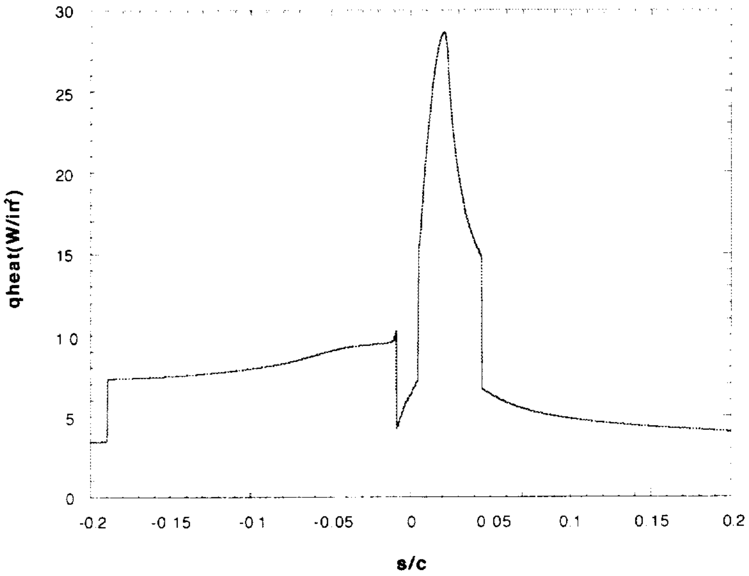

Figure 50: Heat Requirement for the Running Wet Thermal Deicer in Example Case 10

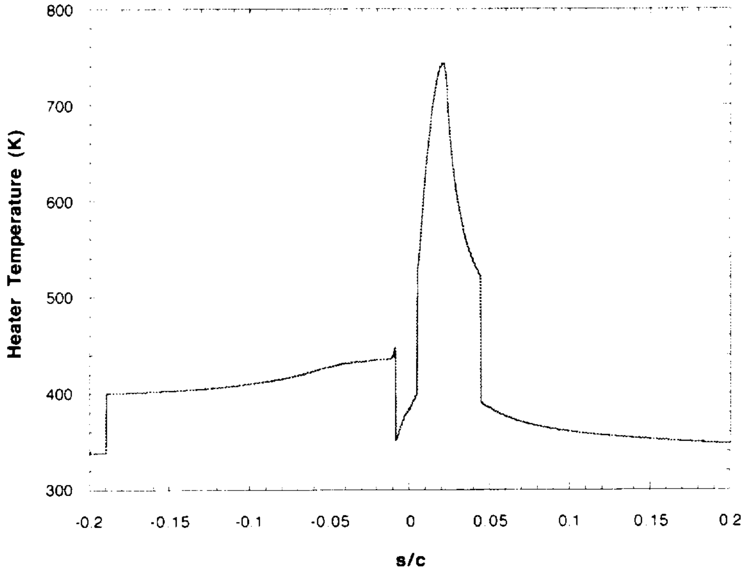

Figure 51: Heater Temperature for the Running Wet Thermal Deicer in Example Case 10

13.11 Case 11: 3-body example

Computation Time: Pentium II 400MHz, 13 min. 31 sec.; Pentium 133MHz, 78 min. 22 sec.

Disk Space: 1.82 MB

This example case shows a condition using a multi-element airfoil instead of the single element airfoils used in the previous example and in the validation database. This condition represents one data point of only a handful of experimental data points which are available on multi-element airfoils. The other data points available are for the same airfoil. A warning will be issued to the user due to the use of multiple body input. An additional warning will be made due to the high angle of attack used. Warnings will also be issued since the trailing edge is not closed on two of the bodies. The user must confirm these warnings in order to run the case.

The use of a potential flow solution for this case is questionable as the flow physics clearly indicate that a more complex flow solution should be sought. However, it is quite difficult and time consuming even with current technology to regrid an iced airfoil multiple times in order to obtain the more accurate solution. Icing physics dictate that in most cases a multiple time step solution (which takes into account the change in flow due to ice accretion) is preferable to a single time step solution which does not. In addition, parametric studies have also shown that the user should make use of a drop size distribution for multi-element cases rather than use a monodispersed drop size as this example shows. The ice shape predicted for this case shows some of the limitations of using the built-in flow solver for a multi-element airfoil. The ice shape predicted on the slat shows an accuracy similar to those shown for single element airfoils. This result should be expected, since the viscous effects are not as dominant in this region. The ice shape predicted on the main element shows a different horn angle than the experiment, but the icing region is similar. The largest deviation from the experimental data is shown on the flap. This result also makes sense physically as the viscous forces are dominant in this region. Additionally, the trailing edge of the flap is open and no corrections were made to the input angle of attack for this case as the actual lift coefficient was not known for this case. These two corrections could make a substantial increase in accuracy for these cases. One point which should be noted is that the ice accretion on the aft elements is somewhat an artifact of the subscale conditions being used. Parametric studies using a full-scale airfoil at flight Reynolds numbers show much less ice accretion on aft elements. The main input file and geometry input file are included on the distribution disks as “case11.inp” and “case11.xyd” respectively.

Table 24: Main Input File for Example Case 11

```
Test Case
&LEW20
ITIMFL =   1
TSTOP  =   360.
IBOD   =    3
IFLO   =   6
DSMN   =    2.0D-4, 4.0D-4, 2.5D-4
NPL    =   24
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 0.9144
AOA    =  8.0
VINF   =  88.5
LWC    = 0.60
TINF   = 268.15
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

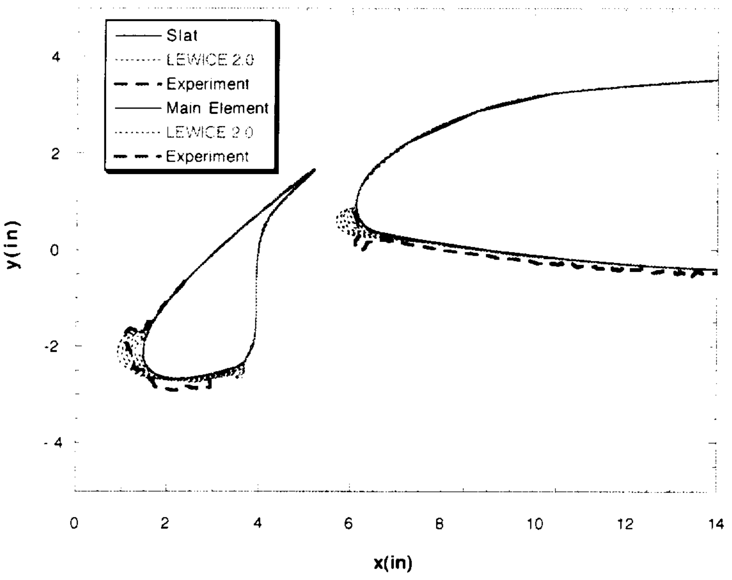

Figure 52: Ice Shape on Slat and Main Elements for Example Case 11

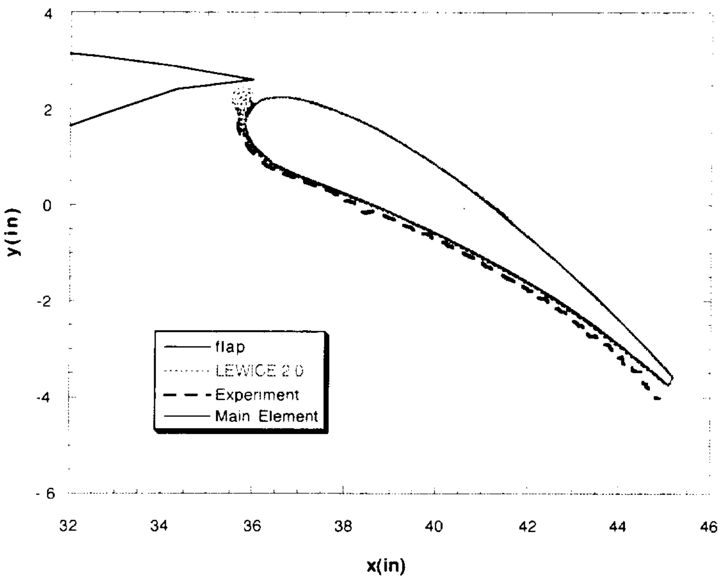

Figure 53: Ice Shape on Flap for Example Case 11

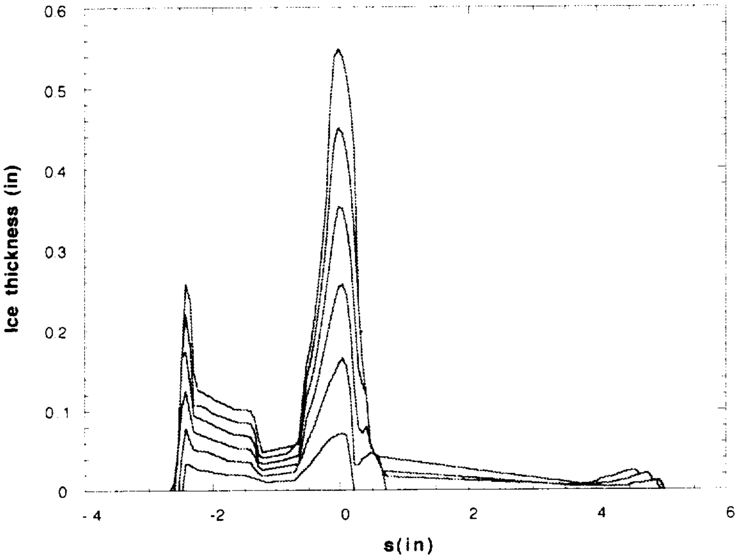

Figure 54: Ice Thickness on Slat for Example Case 11

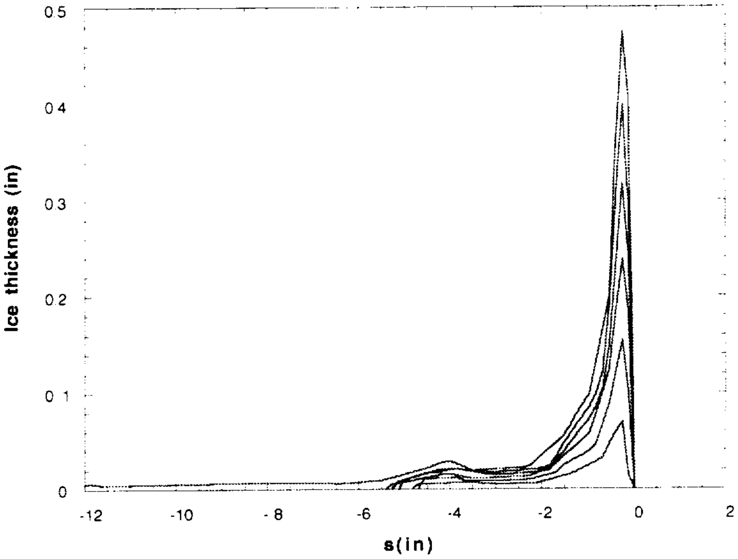

Figure 55: Ice Thickness on Main Element for Example Case 11

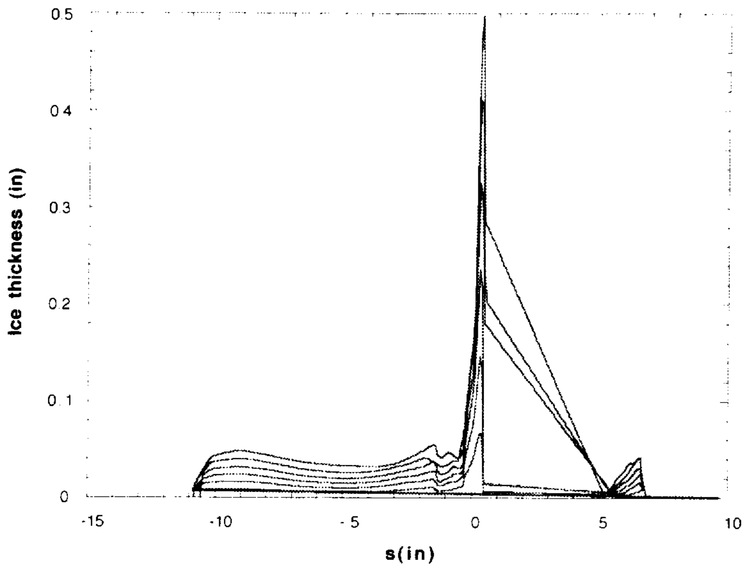

Figure 56: Ice Thickness on Flap for Example Case 11

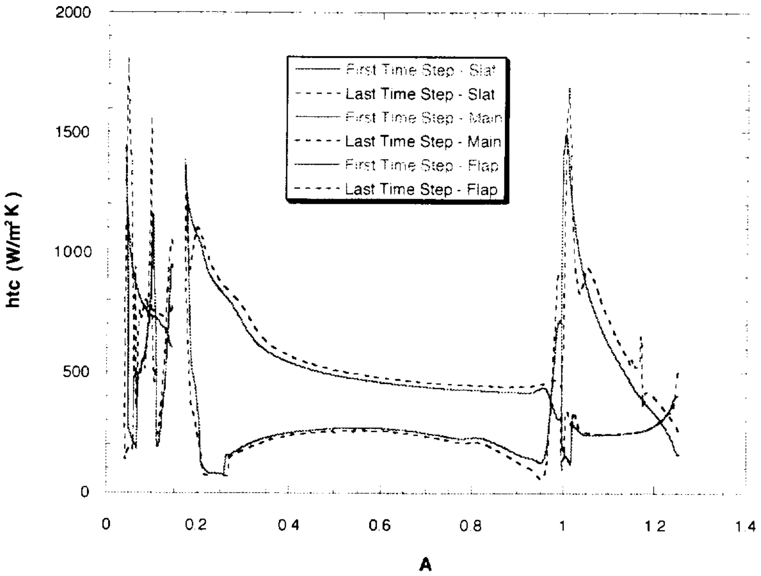

Figure 57: Heat Transfer Coefficient for Example Case 11

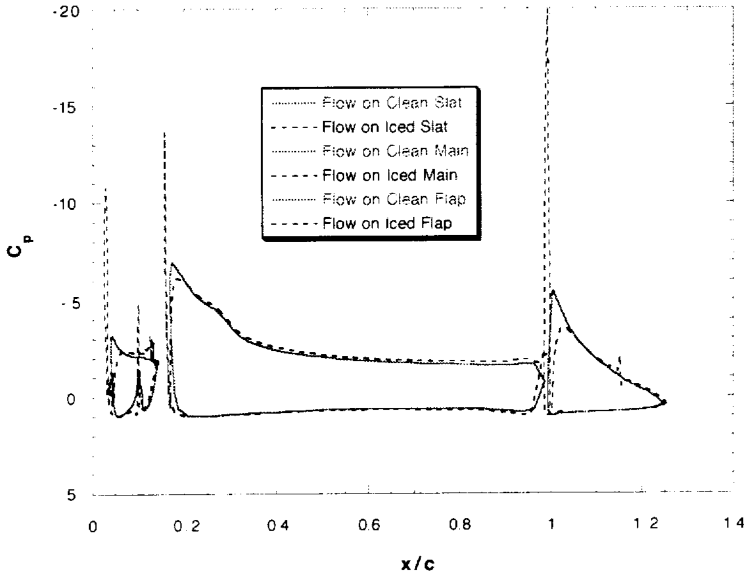

Figure 58: Pressure Coefficient for Example Case 11

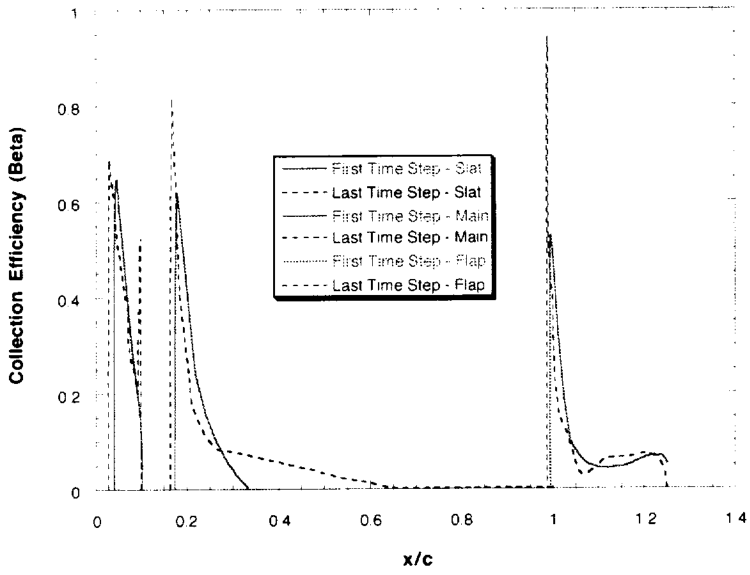

Figure 59: Collection Efficiency for Example Case 11

13.12 Case 12: Grid input

Computation Time: Pentium II 400MHz, 40 sec.; Pentium 133MHz, 3 min. 55 sec.

Disk Space: 177 KB

This example case illustrates a new feature of LEWICE. The user may import a grid and grid-based flow solution from another source in order to perform a more detailed analysis. The drawback of this feature is that since the flow solver is not integrated into the model, only a single time step can be performed. Its use is thus limited to examining differences in water collection efficiency and heat transfer coefficient prediction. A process for integrating a flow solver into LEWICE to perform multi-time step ice accretions using a grid-based flow solver is described in detail in the Programmers Manual for LEWICE[2]. This process will require modifications both to LEWICE and to the flow solver, therefore it is best left to code developers to examine this feature.

The computation times listed for this case represent the time taken for LEWICE to run. The computation time does not include the computation time necessary to create the flow solution. In addition, the disk space listed represents the disk space occupied by the output files from LEWICE. The grid file and flow solution file are considered input files to LEWICE and are much larger. Table 25 shows the main input file for this case. The complete input for this case, including the grid and flow solution are included on the distribution disk. The grid file is a single block structured grid output by GRIDGEN[13]. LEWICE 2.0 can handle up to ten grid blocks, each of which can be as large as 600 by 200 grid points. LEWICE is currently set up to use structured grids, although a programmer familiar with unstructured grids should be able to easily modify the code to handle unstructured grids. This aspect is also covered in the Programmers Manual.

The flow solution provided was generated by the Naviér-Stokes code NPARC[14]. The output format of the flow solution file provided is different from the original output produced by NPARC to conform to the format needed by LEWICE. Other codes may also be used to generate the grid and flow solution. The codes selected for this example were chosen based on their accessibility. This feature is quite new and may require programming knowledge on the part of the user to get a grid solution to read in correctly. At the end of this example case, a section is provided describing some of the pitfalls encountered in reading the original flow solution into LEWICE.

The output for this case shows that the predicted lower impingement limit has been reduced significantly due to the use of this grid solution as compared to the output obtained by using the potential flow solution. A parametric study should be performed on the resolution of the grid and on the panel resolution to determine if this result is due to the flow physics or if it is an artifact of the point spacing near the impingement limit.

Table 25: Main Input File for Example Case 12

```
Test Case
&LEW20
ITIMFL = 0
TSTOP  =   60.
IBOD   =    1
IFLO   =   1
DSMN   =    4.0D-4
NPL    =   24
IGRID  =   1
&END
&DIST
FLWC   = 1.0
DPD    = 20.
&END
&ICE1
CHORD  = 1.745
AOA    =  5.0
VINF   =  76.
LWC    = 0.80
TINF   = 258.00
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
```

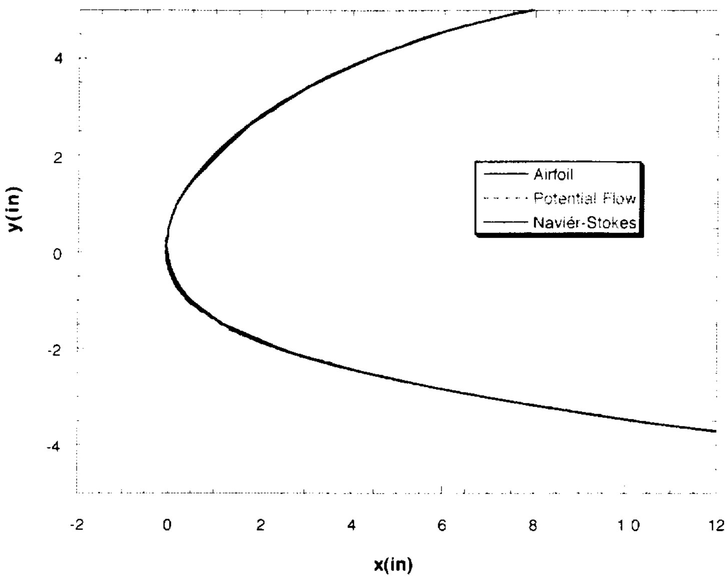

Figure 60: Ice Shape for Example Case 12

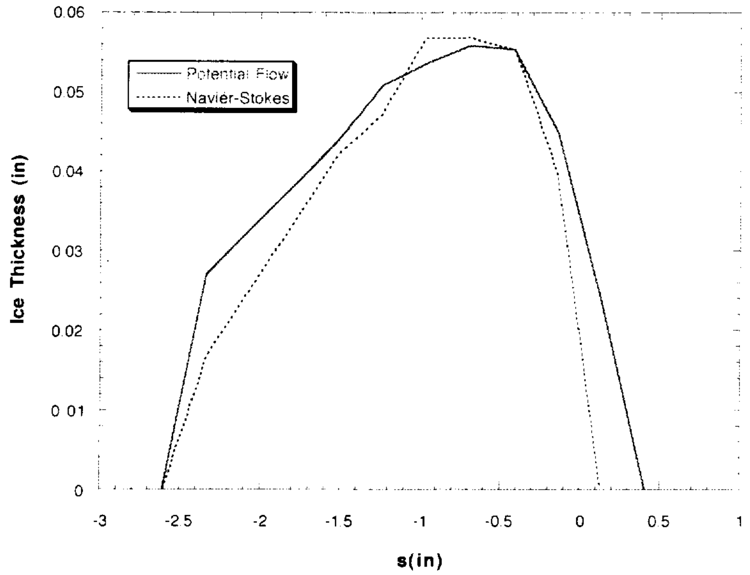

Figure 61: Ice Thickness for Example Case 12

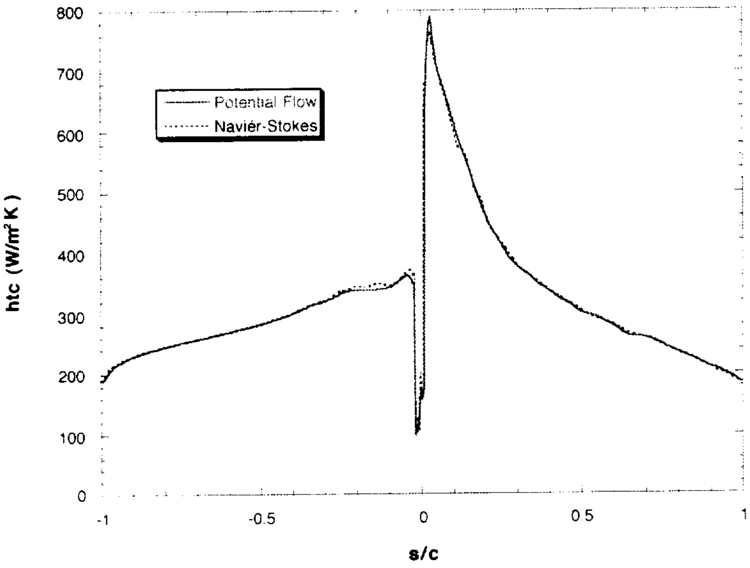

Figure 62: Heat Transfer Coefficient for Example Case 12

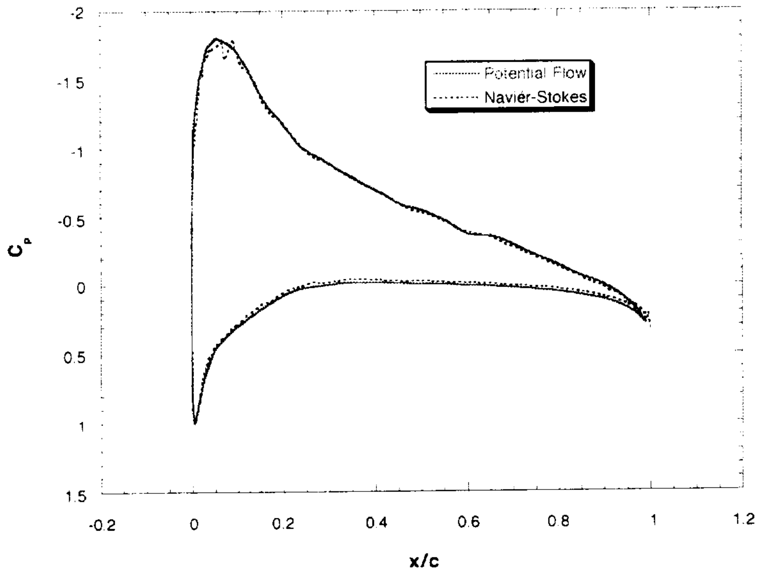

Figure 63: Pressure Coefficient for Example Case 12

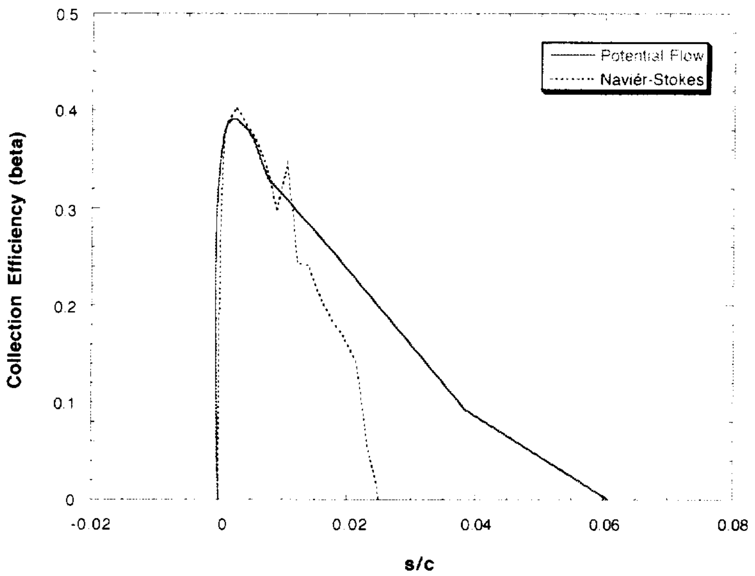

Figure 64: Collection Efficiency for Example Case 12

13.12.1 Grid Input Problems

This section describes the problems encountered when trying to import the NPARC flow solution into LEWICE for Example Case 12. The problems associated with this feature will be discussed in more detail in the Programmers Manual[2]. However, many of the problems do not require extensive programming knowledge to rectify. The problems encountered by the user may be different than those described here. This section is intended to provide some general explanations which will aid the user in reading input files.

Problem 1: Translated Grid Geometry

The airfoil for this case was the NACA23014(mod) airfoil used in the validation report. Therefore, the geometry file originally input into LEWICE used the same airfoil geometry as the validation cases. The grid however used a surface geometry which was offset from this geometry file. It is necessary that the airfoil input by the user matches the surface geometry of the grid. In order to rectify this problem, the airfoil geometry was shifted by 0.15 inches (0.0218 in nondimensional format) in the y-direction so that the airfoil is aligned with the grid surface.

Problem 2: Input Format

LEWICE expects the grid and flow solution to be in PLOT3D binary format with the IBLANK feature **on** as described in the PLOT3D User Manual[11]. For a single body airfoil, LEWICE is set up to read only single block grids. Although the original file was a single block grid, it was written using the multi-block grid format described in the PLOT3D manual. After this problem was corrected, it was discovered that the original input grid had the first index in the second dimension as the outer grid boundary and the last index value as the surface grid. LEWICE expects the grid to be input such that the first index in the second dimension contains the surface geometry. Also, LEWICE expected a 'C' or 'O' grid input clockwise in the first dimension while the actual grid was supplied in the counterclockwise direction. These problems required that a small utility code be written to read the file in its existing format and to output it in the desired format.

Problem 3: Unit Conversions

The next problem which was encountered was the use by NPARC of dimensionless variables which were nondimensionalized by different factors than those expected by LEWICE. The second and third 'Q' functions (ρv_x and ρv_y) had to be divided by the ambient Mach number while the fourth 'Q' function had to be multiplied by γ (1.4) to convert the file to the quantities expected by LEWICE.

Problem 4: Platform Problems

Once this conversion was made, the case ran successfully on an SGI Indigo2. When the case was repeated on a PC, it was discovered that the binary file format for SGIs and for PCs were incompatible. The grid and flow solution files had to be converted to text format before transfer to a PC and then reconverted to binary format on the PC before the grid and solution files could be used on that platform. This problem also required that small utility programs be written to perform the conversion. Two such programs (TOBINARY and TOASCII) are provided on the distribution disk for conversion to and from binary format.

13.12.2 LEWICE Errors Associated with Grid Input Problems

For the problems described earlier LEWICE would, for the most part, read the grid and flow solution file and start running. The problems occurred when runtime errors were generated for these cases. These errors usually occurred during the droplet trajectory routine as the code attempted to interpolate air velocities from this grid. The user should review the runtime error messages which are generated by LEWICE specific to grid usage. For the cases run thus far, the runtime errors which were generated were caused by an incorrect interpretation of the information in the grid and flow solution files. Once these files were converted to the file format expected by LEWICE, the code ran without incident.

Chapter 14: User Tips and Notes

Many of the tips and notes provided in this section are listed in this manual in the description of the input and output files. They are summarized here for convenient reference by the user. Each paragraph may contain a user note which is not directly related to other notes in that section.

14.1 Old Input Files

Input files from previous versions of LEWICE will **not** work 'as is' with this version. Please follow the examples provided or use the utility program CONVERT.

14.2 Input File Errors

An error reading the input file indicates that the file name input does not exist, or does not exist in this directory. Common problems:

1) The file name was not typed correctly (remember to include the extension - use "case1.inp" not simply "case1");

2) The input file is in a different directory than the program. The input file can be in a different directory than the program, but in order for LEWICE to recognize the input file the path must be specified. For example, use "inputs\naca0012\case1.inp" instead of simply "case1.inp" to read the input file "case1.inp" in the directory "inputs" and subdirectory "naca0012". **Note:** The above example used the DOS directory convention of backward slashes "\" to list subdirectories. IRIX and many other unix systems use forward slashes "/" instead.

PC Note: To get to the root directory, first type a backward slash "\", then the path and file name. For example, the command "\lewice\inputs\naca0012\case1.inp" can also be used to read the file "case1.inp" in the directory "C:\lewice\inputs\naca0012".

Unix Note: It is common practice in unix to place all programs in a predefined directory such as /usr/bin so that everyone using that system can run the program. The path for specifying the input file in this case is to provide the path from the directory the user is in. For example, if the user is in their home directory and the input file is in the home directory, no path should be provided. If the user is in their home directory and the input files are in directory ../inputs/naca0012, then the proper path to input is "inputs/naca0012/case1.inp". If the user is in directory ../inputs/naca0012 and the input file is in this directory, then no path needs to be provided in this case either. **P.S.:** This sequence is correct based on the IRIX 6.2 operating system. Behavior for other unix operating systems are expected to be similar, but potentially could be different.

14.3 Porting ASCII Files

The LEWICE input files are ASCII text. PCs, Macs and Unix workstations all have different formats for treating line breaks with ASCII files which may cause problems when transferring input files to different platforms. Specifically, when PC ASCII files are moved to an SGI with IRIX 6.2, there is an extraneous character (^M) at the end of each line. This character must be removed from each line to use the file on the SGI.

The conversion programs provided were designed to be used to convert files in the PLOT3D format from text to binary format and from binary to text format. They are not general purpose conversion utilities.

14.4 PC Application

When run from Windows, a console shell opens for interactive input and output. This console shell disappears when the run is finished. For this reason, it is highly recommended that the user run the PC executable from a DOS Shell instead of from the console shell.

Most of the output data is provided in columns of text, with a text header identifying the variable. This file format can be easily imported into any spreadsheet package for plotting. The program takes about 650 KB of hard disk space for the executable, and several megabytes for output files. The second example case shows the program's potential to produce large output files. The output files for this case takes only 3.3 MB of disk space. However, several of the larger output files were not printed in this example and output was further reduced using the print flags in the main input file. If this same case were to be run with all of the outputs activated, the output for this case would occupy over 45 MB of disk space.

14.5 Listing Variables in a Namelist

Variables in namelist format are input on separate lines. Each line contains a unique variable which is listed in that namelist. The line should contain the variable name followed by an equal sign (=) followed by the value to be assigned to that variable. The value can be in integer, real or exponential format regardless of the definition used within the program. For example, an integer variable does not have to be input as an integer. The value will be truncated for use in the program. In addition, the user is not required to list every variable in the namelist. If a variable is not listed in the input file, the program will use the default value. Default values are listed in this section for each variable. Examples of valid inputs are provided for each namelist section. Common causes for errors occur when the user mistypes the variable name or when the user enters a variable from a previous LEWICE version which is no longer input into that namelist.

14.6 Multiple Bodies / Multi-Element Airfoils

As stated earlier, LEWICE 2.0 can run multiple body simulations including multi-element airfoils. A report of its capabilities in this region shows very encouraging results[7]. However, much of the development effort for version 2.0 has centered on validating the existing features of the program. Even though the results to date have been encouraging, there is not enough data available to consider LEWICE 2.0 validated for multi-body flows.

14.7 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 which is caused by the rough surface. Spikes in this solution will result in irregular ice shape formations. In LEWICE 2.0, this is highly automated by the code, but the user has some control to attempt to obtain better flow solutions.

The number of panels and control volumes used are virtually independent of the number of points contained in the geometry input file.

The input parameter DSMN will control the number of control volumes and panels used. For single body simulations, very few problems have been encountered. The effect of DSMN on ice shape has been documented in the validation report[1]. However, multiple bodies sometimes have problems when running multiple time step simulations. Common problems are for the user to specify a value for DSMN which is too small or too large. Values in the range $2*10^{-4} \leq$ DSMIN $\leq$ $8*10^{-4}$ are recommended. The lower limit reflects the current limits of the array sizes in the program and is not a reflection of the accuracy for low DSMN values. For DSMN values of $8*10^{-4}$ or higher, quantitative differences occur due the coarse spacing provided. An exception was found for cylinders which have a very large surface area compared to similarly sized airfoils. Larger DSMN values are necessary for this geometry due the limitations on array sizes. Please check the geometry output file(s) to determine how many panels the program is using for each body.

It is recommended that the user supply approximately 100 or more panels for each body input.

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. The initial geometry written to the geometry output file(s) should be examined very carefully for anomalies regarding this side effect.

14.8 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 LEWICE manual stated that the maximum amount of ice accreted in any time step should be no greater than 1% of the chord. This is still a reasonable value. The computation used is

$$N = \frac{(LWC)(V)(Time)}{(chord)(\rho_{ice})(0.01)} \tag{3}$$

This will give the user a rough idea of the time step size needed for an accurate simulation. Even for long runs (for example 45 min. hold conditions) small time steps can and should be used.

The variability of LEWICE results for various time steps and point spacings is discussed in the section on Numerical Variability in the report on the validation tests. Due to this variability, LEWICE 2.0 selects automatic time stepping to be **on** as its default setting.

For very small (< 6 inch) chord geometries such as cylinders, the number of time steps recommended by the code may be considered prohibitive by the user. It may be possible (and even necessary) to decrease the number of time steps for these cases. It should be noted though that the smallest chord airfoil in the validation database is a 14 inch chord NACA0015 for which 8 ice shapes have been digitized.

14.9 DSMN (Point Spacing)

The number of panels and control volumes used are virtually independent of the number of points contained in the geometry input file.

In version 2.0 the number of control volumes will be much greater than the number of panels. The ratio of control volumes to panels is approximately 50 to 1. The default value for DSMN is $4*10^{-4}$.

There is one value of DSMN for each body. If only one body exists, only the first value input is used by the code. For multi-body simulations, it is to the user's advantage to use smaller DSMN values on the smaller bodies and larger values on the larger bodies which are input. Unless otherwise indicated, the user should not select excessively large values for DSMN. Values smaller than $2*10^{-4}$ have been used for small airfoil elements such as slats and flaps.

DSMN values larger than $8*10^{-4}$ are sometimes needed when the ice shape is extremely large in comparison to the airfoil. An example of this condition occurs for cylinders below six inches in diameter, especially when a lengthy accretion time is used (20 minutes or more). The time step for this case may have to be increased from the recommended values in order to run this case as well.

The total wrap distance around a typical airfoil is slightly greater than 2 (dimensionless value). For a DSMN value of $4*10^{-4}$, the number of control volumes produced will exceed 5000. Since the array size limit is 10000 points, the total wrap distance around the iced geometry needs to be more than four times greater than the chord length for the array bounds to be exceeded. For a DSMN value of $8*10^{-4}$, the total wrap distance around the iced geometry needs to be more than eight times greater than the chord length. The ratio of wrap distance to chord length for a cylinder is by definition pi (3.1415926536...). This ratio will increase as the ice grows on the cylinder. This example shows why larger DSMN values are likely needed for a cylinder.

14.10 Number of Trajectories

The number of trajectories used in the impingement region is an input to the code. A good approximation would be to first estimate how many panels are expected to be in the impingement region. The number of trajectories should not be less than one trajectory for every three panels and should not be greater than one for each panel. An excessive number of trajectories should be avoided as this will slow down the solution. The actual number which the code uses for the collection efficiency calculation may be different than the value input. The code is limited to one trajectory strike per panel. If more than one trajectory hits a given panel, only the first hit will be saved.

14.11 Grid-Based Flow Solutions

Some cases have been made using a grid solution as input to verify that the routines function as designed. This option has seen very limited testing and **has not been validated** against the database of experimental ice shapes. This procedure may still be buggy and is not recommended unless the user is willing to customize the code for their use. Since only one time step can be used with this option, the use of this feature is limited to collection efficiency and heat transfer coefficient prediction. Even for these uses, the user should perform grid resolution studies in the impingement region before drawing any conclusions.

There is no error checking of the grid and solution file. The user should independently verify that the grid and solution files are correct for the case being run. In particular, the angle of attack and velocity should match the values input in the main input data file. The program will also not run with a grid solution unless the grid surface geometry is very similar to the body geometry read in from the geometry input file(s).

14.12 Anti-Icing

This program will calculate the heat requirements using a hot air or an electrothermal anti-icer. It will then compute the ice shape as if the surface were unheated. Layers are input with the inner surface first, and the outer surface last. The information in the deicei.inp input file will be used for each body in a multi-element calculation. The desired surface temperature input in the "deicei.inp" input file must be above freezing (in Kelvin) for this option to work properly. NASA Glenn also has codes which perform more detailed analysis of deicer and anti-icer performance. The LEWICE/Thermal code[8] performs a 2D transient deicer simulation and the ANTICE code[9] performs a 2D steady-state anti-icing simulation. If the user needs a more detailed analysis than provided with this LEWICE function, they are encouraged to try these codes. These two codes will be integrated into future versions of LEWICE.

This routine does not affect the ice accretion routine. The code will output an ice shape as if no heat had been applied. This routine generates a separate file containing the temperatures and heat fluxes needed to maintain a desired surface temperature which is input by the user.

This routine treats the current geometry as the airfoil and does not distinguish an iced airfoil from an un-iced airfoil. Therefore, only the results obtained in the first time step are applicable to an anti-icing problem.

The text accompanying the input fields in the examples listed is provided for informational purposes. Only the numerical value is read by LEWICE.

Typical running wet anti-icing systems operate in the region 5-10 °C while an evaporative system may operate at 50 °C or even higher.

The internal heat transfer coefficient for regions outside those specified by the user will be zero.

14.13 Droplet Distribution

Most cases run with LEWICE 2.0 in the validation database use a single drop size, the MVD for the flight condition. Although multiple drop size distributions can be run with LEWICE 2.0, execution times will be increased. The difference in ice shape and icing limits are documented in the validation report[1]. This report shows that the effect is very slight for single body simulations. The procedure used by LEWICE for multiple drop size distributions 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 for this feature would be to determine more accurate impingement limits on the clean airfoil. Preliminary results have shown that multiple drop size distributions can have a large impact on the collection efficiencies and ice accretion of multi-element airfoils.

The FLWC values input in the main input file are Fraction Liquid Water Content. These values must add to one (1). The program will adjust the FLWC values proportionately so they add to 1.

The program will determine the number of drop sizes in the distribution by looking for the first occurrence where FLWC = 0. Therefore, the user should not place zeros in the FLWC field until the end of the distribution is reached.

MVD is not an input variable to LEWICE. The MVD is calculated from the individual drop sizes input in this section.

Important: The validation database contains MVD drop sizes in the range 15≤ MVD ≤ 270. Although some experimental data has been collected in the 50 < MVD ≤ 270 micron range and has been used for code validation, there is not enough data available to consider the code validated for these drop sizes. Caution should be exercised when running cases outside this region. Due to the recent popularity of drop size inputs in exceedence conditions, it is worth emphasizing the above statement. This statement does **not** imply that LEWICE cannot run the drop size distribution input by the user. It most likely can. The warning statement does **not** imply that the results will be inaccurate. LEWICE results for exceedence conditions are quite encouraging in this respect. The statement simply points out that insufficient experimental data is available at these drop sizes. Since the results cannot be experimentally validated, the true accuracy of the results cannot be verified.

14.14 Collection Efficiencies and Droplet Trajectories

The wrap distance from the leading edge output in the collection efficiency data file (beta.dat) and the impingement limit data file (imp.dat) may lose physical meaning past the first time step.

The impingement limit listed in the output file "imp.dat" may not match the wrap location where the collection efficiency (beta) goes to zero in the output file "beta.dat". The difference is due to the resolution of surface points in the impingement limit region. The value quoted in the file "imp.dat" is the computed impingement limit. The location where the collection efficiency goes to zero is the surface point closest to this value.

Droplet trajectory output is sequential. The first set of coordinates contain the coordinates of the first trajectory calculated. Subsequent output contains coordinates for each successive trajectory calculated. No indicator is present in the output file to offset trajectory output from one time step to another. Hence, this output is only recommended for the first time step.

14.15 Chord Length

CHORD is the distance from the leading edge to the trailing edge in meters. For a cylinder, this represents the cylinder diameter. For airfoils, it is the standard chord length. For a multi-body simulation, CHORD represents the reference length used to nondimensionalize the coordinates input. A typical value used for multi-element airfoils is the length of the airfoil in the stowed position.

14.16 Multiple Stagnation Points

A main cause of error in LEWICE occurs when multiple stagnation points are predicted by the flow solution. 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, the point closest to the leading edge is used. If this is not satisfactory, the user should lower the DSMN input variable or increase the number of points in the input data so as to produce a single stagnation point value.

14.17 Flow Code Limitation

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

Naviér-Stokes code capable of handling transonic conditions becomes available, the user is encouraged to use it for this case. In addition, no experimental data is available for Mach Numbers above M=0.45. Therefore the code has not been validated against experimental data above this value. Problems may exist with the solution due to the limitations of potential flow.

Potential flow cannot model stall or post-stall behavior. The user should also note that in the validation test procedure, the angle of attack input into the code was sometimes different from the actual angle of attack value. This difference was made to compensate for the difference in predicted lift using a potential flow code and the actual lift on the clean airfoil.

14.18 Static Pressure/Altitude

Ambient pressure is not recorded as part of the tunnel data, so the exact value during the tests is unknown. However, since ambient pressure is at best a secondary effect on the ice accretion process, a representative value near atmospheric pressure was used for the comparison.

To a good approximation, for a 'standard atmosphere', the following equation can be used:

$P = 100920 - 11.35H + 0.00039456H^2$ where

P = pressure in N/m^2 and

H=height (altitude) in meters

14.19 Temperature

The input variable for temperature in LEWICE is the ambient *static* temperature in degrees Kelvin. The data supplied to researchers is often the total temperature, not the static temperature. Make certain the value input is correct!

14.20 Relative Humidity

Relative humidity is not recorded as part of the tunnel data, so the exact value during the tests is unknown. However, since it is at best a secondary effect on the ice accretion process, a representative value of 100% humidity can be used.

14.21 Printer Flags

Output files from LEWICE can be very large. If all of this information is not needed, the user can save a great deal of disk space by not generating individual files or by reducing the amount of information which is sent to those files. Example Case 2 illustrates this effect. As listed in this example, the case produces 3.3 MB of output. If all of the printer flags are activated, the output will exceed 45 MB. Finally, it should be noted that the definition for the print flag TPRT has changed from version 1.6. The current definition of TPRT has the opposite meaning for input values of TPRT=1 and TPRT=2 than the definition used in version 1.6.

14.22 Geometry Input

It should be noted that all of the validation data uses airfoils. Although LEWICE can simulate any enclosed body (or bodies), the validation performed to date has been limited to the available data.

A separate input file must be provided for each body being simulated. If only one body exists, only one geometry file will be read. Each line of the geometry input file contains an x,y coordinate pair for the body geometry. The x-coordinate is listed first. The format of the data is free format for the x,y coordinates. It is quite common for problems to arise when inputting a new geometry for the first time. The following discussion will describe some of the common errors made by users in generating an input file.

If the body geometry is too coarse, the panel model created may not replicate the body geometry input. The initial set of coordinates output to file "ice1.dat" contain the initial panel model of the first body geometry. The user should check that this airfoil matches their geometry input file whenever a new geometry is input or if the point spacing (DSMN) has increased. Standard geometry input files used for testing purposes range from 50 to 150 points.

The panel solution used in this code assumes that the body(s) being simulated are closed bodies. Several tests have been run using airfoils with open trailing edges and some of the results appear acceptable. If the flow solution calculated with an open trailing edge is acceptable to the user, then there may be no need to alter the trailing edge simply to enclose the body.

LEWICE requires that the body geometry points should be input in a clockwise fashion. This means that the points are input starting at the trailing edge and proceed sequentially toward the leading edge along the lower surface up to the leading edge, then traverse back to the trailing edge along the upper surface.

Several errors can occur when points are typed in. These errors may cause the geometry to be different from the one intended by the user. Some common errors include: points input in reverse order; missing or misplaced decimal points; or mistyped numbers. LEWICE cannot check for all possible errors. The user should always check the first panel geometry printed to "ice1.dat" to ensure that the body being used by LEWICE closely resembles the intended geometry. Specifically, the user should ensure that there are sufficient input points in regions of high curvature. The point distribution methodology used in LEWICE will tend to "round off" corners if an insufficient number of points are used.

Additional input problems may arise when the user attempts to input more than one body. One such problem can occur when the bodies intersect. This can easily occur with multi-element airfoils if the user does not properly rotate the flap or use the proper gap settings.

LEWICE cannot run multiple bodies where one body is completely inside another body. This can occur if the coordinates for the bodies are supplied relative to different points of origin rather than relative to the same point of origin.

The logic used by the trajectory module dictates that multiple bodies need to be input in sequential order in the x-direction. This means that the first body a particle could encounter must be listed first, the second body it could encounter must be listed second and so on. This criteria is based upon the leading edge of each body, not on the trailing edge as particles are most likely to impinge on the leading edge of each body.

Chapter 15: Procedure for Acquiring LEWICE or other NASA Icing Codes

Funding for LEWICE development comes from the NASA budget which is provided for by US tax dollars. LEWICE is therefore provided free of charge to US corporations, universities or individuals. A request letter such as that provided in the next section should be sent to the Icing Branch Chief. The letter may be sent by regular mail, electronic mail or fax. A current mailing address is provided below.

15.1 Current Address for Icing Code Requests

Mr. Thomas H. Bond
Branch Chief, Icing Branch
NASA Glenn Research Center
21000 Brookpark Rd.
MS 11-2
Cleveland OH 44135
E-mail: Thomas.H.Bond@grc.nasa.gov
Phone: (216) 433-3900
FAX: (216) 977-1269

Chapter 16: Sample Code Request Letter

Dear Sir,

Our company/university would like to request the ice accretion code LEWICE 2.0 for use in design and/or certification of our products for flight in icing conditions. We are a US corporation with offices in __________ (place). Our immediate need is for the certification of ____ (fill in blank). The code will be run primarily on ________ (name the system) and it is preferred that the code be distributed on _________ (preferred media). Thank you.

Sincerely,

Note: Code requests are normally filled 2-3 weeks from the request date on average.

Note: As of March 1, 1999, the NASA Lewis Research Center has been renamed to the John H. Glenn Research Center at Lewis Field. The address listed above reflects this change.

Note: The code developers are involved only in technical support of the codes and are not directly involved in the distribution process. Questions concerning distribution should be sent to the branch office.

Chapter 17: Technical Support

Technical support is available from 9AM to 5 PM Eastern time from the code developer(s). Questions on running the code or how to simulate difficult problems with LEWICE should be sent to:

William Wright
NASA Glenn Research Center
21000 Brookpark Rd.
MS 11-2
Cleveland OH 44135
E-mail: William.B.Wright@grc.nasa.gov
Phone: (216) 433-2161
FAX: (216) 977-1269

E-mail and fax are the preferred methods of communication. The user should provide a description of the problem and error messages obtained and also provide an input file (including geometry) so the error can be reproduced. If correspondence is confidential, please mark it as such. Code improvements including bug fixes which result from this correspondence will be incorporated into future code versions which can be released by NASA.

References

1. Wright, W. B. and Rutkowski, A., "Validation Results for LEWICE 2.0," NASA CR 208690, Nov. 1998.

2. Wright, W. B., Potapczuk, M. G., Levinson, L. H. and Mellor, P, "Programmers Manual for LEWICE 2.0", internal NASA documentation, 1998.

3. Hess, J. L. and Smith, A. M. O., "Calculation of Potential Flow About Arbitrary Bodies," Progress in Aeronautical Sciences, 8:1-138, (D. Kuchemann, editor), Pergamon Press, Elmsford, New York, 1967.

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

5. Messinger, B. L., "Equilibrium Temperature of an Unheated Icing Surface as a Function of Air Speed," J. Aeronaut. Sci., vol. 20, no. 1, Jan. 1953, pp. 29-42.

6. Wright, W.B., "Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6," NASA CR 198355, June 1995.

7. Wright, W.B., "Capabilities of LEWICE 1.6 and Comparison with Experimental Data," AHS International Icing Symposium, Montreal, Canada, pp. 347-61, Sept. 1995.

8. Masiulaniec, K. C. and Wright, W. B., "User's Manual for the NASA Lewis Ice Accretion/ Heat Transfer Prediction Code with Electrothermal Deicer Input", NASA CR 4530, 1994.

9. 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.

10. Wright, W.B., and Potapczuk, M. G., "Computational Simulation of Large Droplet Icing," Proceedings of the FAA International Conference on Aircraft Inflight Icing, pp. 545-55, Aug. 1996.

11. Walatka, P. P., et. al., "PLOT3D User Manual," NASA TM 101067, Mar. 1990.

12. Papadakis, M., et. al., "Progress in Measuring Water Impingement Characteristics on Aircraft Surfaces," AIAA98-0488, Jan. 1998.

13. Chawner, J. and Steinbrenner, J., "Automatic Grid Generation Using GRIDGEN," NASA CP-3291, Surface Modeling, Grid Generation, and Related Issues in CFD Solutions, May 1995.

14. NPARC Version 3.0 User's Manual, NPARC Alliance, Sept. 1996

A

angle 17, 46, 47, 120, 133

angle of attack 17, 28, 61, 133, 151, 155

ANTICE 22, 70, 124, 125, 129, 152

anti-ice 22, 50, 53, 70, 124, 125, 129, 151

array size 16, 18, 19, 45, 79, 80, 87, 88, 97, 149, 150

ASCII 78

automatic 14, 15, 17, 93, 97, 116, 149

B

benchmark 90, 93, 97, 124, 129

beta.dat 62, 66, 153

binary 77, 78

body 1, 2, 8, 16, 19, 22, 28, 35, 39, 44, 45, 46, 47, 48, 50, 57, 58, 61, 63, 64, 65, 69, 70, 71, 73, 75, 79, 80, 82, 83, 84, 133, 144, 148, 149, 150, 151, 152, 156, 157

boundary layer 67

BPRT 34, 62, 66

bug 22, 86, 151, 160

C

CD-ROM 3, 4, 6

certification 1, 25, 93, 106, 159

chord 14, 18,27, 28, 57, 67, 88, 97, 106, 107, 149, 150, 154

collection efficiency 20, 21, 30, 35, 62, 68, 89, 93, 97, 101, 102, 106, 107, 110, 111, 116, 120, 124, 125, 138, 139, 143, 151, 152

computation time 33, 89, 93, 97, 102, 106, 111, 116, 120, 124, 129, 133, 139

conductivity 55

control volume 18, 19, 33, 34, 63, 64, 65, 67, 68, 70, 84, 88, 148, 150

CONVERT 76, 77, 146

ctemp.dat 70

cylinder 2, 18, 19, 27, 88, 97, 98, 149, 150, 154

D

debug 11, 35, 44, 66, 83

default 11, 14, 15, 16, 17, 18, 19, 20, 22, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 50, 66, 75, 77, 89, 148, 149, 150

deicei.inp 22, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 125, 126, 129, 130, 152

dens.dat 34, 63

density 14, 20, 21, 63, 67, 111

design 1, 22, 56, 90, 151, 159

disk space 5, 33, 68, 69, 89, 93, 97, 102, 106, 111, 116, 120, 124, 129, 133, 139, 147, 156

distribution 20, 23, 24, 25, 27, 62, 102, 111, 133, 152, 153

distribution disk 5, 7, 89, 90, 93, 98, 103, 107, 111, 117, 121, 125, 129, 134, 139, 145

DOS 4, 7, 8, 72, 146

DOS Shell 4, 7, 12, 13, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 72, 74, 76, 78, 147

DPD 25

drop 1, 21, 23, 24, 25, 26, 27, 30, 35, 49, 62, 65, 68, 69, 81, 82, 83, 102, 107, 111, 133, 145, 152, 153

DSMN 18, 19, 44, 79, 80, 82, 83, 85, 86, 87, 88, 97, 121, 148, 149, 150, 154, 156

dyice.dat 34, 63

E

electrothermal 50, 53, 56, 70, 129, 151

energy balance 34, 67, 68, 84, 85

EPRT 34, 35, 67

error 7, 8, 9, 10, 11, 12, 15, 16, 18, 19, 21, 24, 25, 27, 28, 29, 30, 31, 32, 33, 39, 44, 46, 48, 51, 52, 53, 54, 55, 56, 57, 58, 61, 63, 64, 66, 70, 71, 73, 74, 76, 77, 79, 80, 81, 82, 83, 84, 85, 86, 87, 145, 146, 148, 151, 154, 156, 157, 160

evaporative 50, 51, 52, 67, 85, 124, 125, 127, 128, 152

example case 5, 36, 61, 89, 91, 92, 93, 95, 96, 97, 99, 100, 101, 103, 104, 105, 107, 108, 109, 110, 112, 113, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 147, 156

exceedence 25, 26, 30, 106, 111, 153

EXPAND 74, 75

experiment 71, 73, 116, 151

experimental data 2, 15, 25, 28, 29, 30, 38, 65, 89, 98, 107, 124, 133, 153, 155

F

file name 4, 8, 39, 72, 73, 75, 76, 78, 146

final.dat 63, 65

fixed.dat 49, 64

flow 1, 16, 18, 28, 29, 33, 45, 46, 47, 60, 64, 67, 79, 81, 82, 83, 84, 89, 133, 139, 143, 144, 145, 148, 154, 156

flow.dat 33, 64

FLWC 24, 27, 153

FPRT 33, 64

fract.dat 34, 64

freezing fraction 64

Frössling number 65

G

geometry 1, 4, 8, 18, 19, 34, 39, 44, 46, 47, 48, 61, 64, 70, 71, 73, 75, 79, 82, 90, 93, 97, 103, 107, 111, 117, 121, 125, 129, 134, 143, 144, 148, 149, 150, 151, 156, 157, 160

geometry.dat 70

glaze ice 90, 93, 107, 116, 148, 149, 154

grid 21, 60, 61, 64, 70, 81, 133, 139, 143, 144, 145, 151

GRIDGEN 139, 162

H

heat transfer coefficient 33, 56, 57, 58, 65, 69, 89, 92, 96, 100, 105, 109, 115, 119, 123, 125, 137, 139, 142, 151, 152

hot air 22, 50, 53, 70, 124, 127, 128, 129, 151

HPRT 33, 34, 65, 67
htc.dat 33, 65

I

ice accretion 1, 21, 22, 23, 31, 32, 47, 90, 106, 116, 133, 134, 139, 152, 153, 155, 159
ice growth 1, 79, 85
ice shape 5, 15, 18, 19, 21, 23, 60, 63, 64, 65, 68, 71, 73, 79, 82, 83, 84, 85, 87, 89, 91, 95, 97, 99, 102, 104, 106, 108, 111, 113, 114, 116, 118, 120, 122, 133, 135, 141, 148, 149, 150, 151, 152, 154
ice thickness 2, 63, 65, 68, 71, 73, 84, 89, 91, 95, 100, 102, 104, 109, 111, 113, 114, 118, 122, 136, 137, 141
ice1.dat 44, 46, 65, 68, 156, 157
iced geometry 2, 63, 64, 65, 71, 73, 79, 80, 83, 84, 87, 88, 150
icing 1, 2, 15, 22, 27, 30, 52, 93, 106, 111, 133, 152, 158, 159
icing conditions 1
icing limit 107
IDBF 35, 66
IDEICE 22, 50, 70, 77, 124
IEVAP 52
IFLO 14, 17, 22, 82, 83, 85, 86
IGRID 21, 50, 64, 70
imp.dat 65, 154
impingement 1, 65, 107, 151
impingement limit 20, 65, 66, 82, 83, 84, 107, 140, 153
input file 3, 4, 5, 7, 8, 9, 11, 12, 19, 20, 22, 36, 39, 44, 45, 50, 51, 54, 55, 56, 57, 58, 60, 61, 62, 64, 70, 71, 72, 73, 74, 75, 76, 77, 81, 83, 86, 89, 90, 93, 94, 98, 103, 107, 111, 112, 117, 121, 125, 126, 129, 130, 134, 139, 140, 143, 146, 147, 148, 150, 151, 153, 156, 160
interpolation 74, 81, 86, 87
IRIX 6, 7, 8, 11, 12, 13, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 72, 73, 74, 76, 77, 78, 146, 147
IRT 31, 106
ITHERM 53
ITIMFL 13, 15, 17, 116

J

junk.dat 11, 13, 35, 44, 66

L

LAYERS 56
layers 53, 54, 151
leading edge 27, 46, 48, 57, 66, 68, 154, 157
LEWICE/Thermal 22, 70, 124, 129, 152
LHEAT 56
LWC 1, 14, 29, 30

M

Mach number 29, 144, 154
mass balance 34, 63, 64, 66

mass.dat 34, 66
meteorological 1, 27
misc.dat 13, 67
monodispersed 27, 102, 107, 111, 114, 133
MPRT 34, 63, 64, 67
multi-body 16, 19, 22, 27, 70, 71, 79, 84, 148, 150, 154
multi-element 2, 16, 22, 27, 48, 50, 79, 83, 133, 148, 152, 153, 154, 157

N

namelist 11, 13, 15, 23, 27, 32, 33, 36, 148
Naviér-Stokes 139, 155
Newton-Raphson 83, 85
noice.dat 70
note 8, 18, 19, 20, 23, 27, 28, 30, 31, 32, 34, 35, 36, 52, 53, 54, 61, 62, 65, 66, 68, 72, 75, 76, 77, 78, 93, 98, 107, 133, 146, 150, 155, 156, 159
NPARC 139, 143, 144, 162
NPL 20
Nusselt number 65

O

operating system 3, 7, 8, 73, 147

P

panel 18, 19, 20, 21, 34, 44, 45, 46, 47, 64, 80, 111, 140, 148, 149, 150, 151, 156, 157
PC 3, 4, 5, 7, 89, 144, 147
PINF 31, 32
platform 7, 89, 90, 93, 144
PLOT3D 21, 60, 144, 162
points 8, 15, 19, 21, 30, 44, 45, 46, 47, 48, 60, 63, 64, 65, 71, 73, 74, 75, 87, 111, 133, 139, 148, 149, 150, 153, 154, 156, 157
potential flow 21, 28, 29, 60, 64, 133, 140, 148, 155, 161
pres.dat 33, 67
pressure 1, 31, 67, 154, 155
pressure coefficient 64, 70, 92, 96, 101, 102, 105, 115, 119, 123, 125, 138, 142, 154
Programmers Manual 69, 71, 139, 143, 161

Q

q.plt 21, 50, 60
qener.dat 34, 67

R

relative humidity 1, 32, 155
request 158, 159
Reynolds number 29, 65, 133
RH 32
RHOP 20
roughness 69
runback 63, 64, 66, 162
running wet 50, 51, 52, 129, 132, 152
runtime error 19, 79, 145

S

SGI 6, 7, 44, 144
stagnation 62, 63, 64, 65, 66, 67, 69, 70, 102, 154
Sun 6
system error 7, 11, 12, 13, 44, 50, 51, 52, 53, 54, 55, 56, 57, 59, 72, 74, 76, 77, 78

T

technical support 159, 160
temp.dat 34, 68
temperature 1, 22, 30, 38, 51, 67, 68, 70, 84, 85, 125, 128, 129, 132, 152, 155, 161
thermodynamic 1
THICK 63, 68, 71, 72, 73, 74
thick.dat 68, 89
time step 13, 14, 15, 17, 18, 20, 21, 22, 60, 62, 63, 64, 65, 66, 68, 69, 79, 80, 82, 83, 84, 85, 86, 87, 89, 93, 97, 116, 120, 124, 125, 139, 148, 149, 150, 151, 154
TINF 30, 31
TPRT 35, 68, 156
traj1.dat 35, 68, 69
trajectory 1, 20, 21, 35, 48, 67, 68, 81, 82, 83, 84, 145, 151, 153, 157, 161
TSTOP 15, 16
TSURF 51, 52
tsurf 51

U

unix 7, 8, 12, 72, 73, 146, 147

V

validation 5, 14, 18, 19, 20, 25, 26, 28, 29, 30, 31, 38, 39, 62, 71, 73, 79, 82, 89, 93, 97, 102, 106, 107, 133, 143, 149, 152, 153, 155, 156, 161
variability 14, 90, 149
velocity 1, 14, 21, 28, 29, 31, 61, 63, 64, 67, 70, 81, 83, 151
VINF 28, 29

W

warning 5, 9, 11, 14, 15, 16, 17, 19, 20, 21, 22, 24, 25, 26, 28, 29, 30, 31, 33, 34, 35, 38, 39, 44, 45, 46, 47, 48, 52, 53, 57, 66, 75, 83, 86, 87, 93, 98, 106, 111, 116, 120, 124, 129, 133, 153
Windows 4, 5, 7, 12, 13, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 72, 74, 76, 78, 147
wrap distance 19, 57, 58, 62, 63, 64, 65, 66, 67, 68, 69, 70, 88, 89, 97, 102, 150

X

xkinit.dat 69
xkinit2.dat 34, 69
xy.plt 21, 50, 60

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
September 1999

3. REPORT TYPE AND DATES COVERED
Final Contractor Report

4. TITLE AND SUBTITLE
User Manual for the NASA Glenn Ice Accretion Code LEWICE
Version 2.0

5. FUNDING NUMBERS
WU-548-20-23-00
NAS3-98008

6. AUTHOR(S)
William B. Wright

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Dynacs Engineering Company, Inc.
2001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER
E-11949

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR—1999-209409

11. SUPPLEMENTARY NOTES

Project Manager, Thomas Bond, Turbomachinery and Propulsion Systems Division, NASA Glenn Research Center, organization code 5840, (216) 433-3900.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category: 03 Distribution: Nonstandard

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

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

A research project is underway at NASA Glenn to produce a computer code which can accurately predict ice growth under a wide range of meteorological conditions for any aircraft surface. This report will present a description of the code inputs and outputs from version 2.0 of this code, which is called LEWICE. This version differs from previous releases due to its robustness and its ability to reproduce results accurately for different spacing and time step criteria across computing platform. It also differs in the extensive effort undertaken to compare the results against the database of ice shapes which have been generated in the NASA Glenn Icing Research Tunnel (IRT) 1. This report will only describe the features of the code related to the use of the program. The report will not describe the inner workings of the code or the physical models used. This information is available in the form of several unpublished documents which will be collectively referred to as a Programmers Manual for LEWICE 2 in this report. These reports are intended as an update/replacement for all previous user manuals of LEWICE. In addition to describing the changes and improvements made for this version, information from previous manuals may be duplicated so that the user will not need to consult previous manuals to use this code.

14. SUBJECT TERMS
Icing

15. NUMBER OF PAGES
182

16. PRICE CODE
A09

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT