

NASA/CR—2008-214255



User's Manual for LEWICE Version 3.2

William Wright
QSS Group, Inc., Cleveland, Ohio

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected

papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at 301-621-0134
- Telephone the NASA STI Help Desk at 301-621-0390
- Write to:
NASA Center for AeroSpace Information (CASI)
7115 Standard Drive
Hanover, MD 21076-1320

NASA/CR—2008-214255



User's Manual for LEWICE Version 3.2

William Wright
QSS Group, Inc., Cleveland, Ohio

Prepared under Contract NAS3-00145

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

November 2008

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by an expert reviewer(s).

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076-1320

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Available electronically at <http://gltrs.grc.nasa.gov>

Contents

Chapter 1:	Introduction	1
Chapter 2:	Installation Procedure	3
Chapter 3:	Running LEWICE on a PC	5
3.1.	LEWICE Quick Start Guide.....	5
3.2.	PC Requirements	6
Chapter 4:	Running LEWICE on Other Systems	7
4.1.	Source Code Modification	7
Chapter 5:	Changes From Version 2.0.....	9
5.1.	Changes From Version 3.0.....	10
Chapter 6:	Recommended Practices	11
6.1.	Panel Criteria.....	11
6.2.	Time Step	12
6.3.	DSMN (Point Spacing).....	12
6.4.	Number of Trajectories.....	13
6.5.	One-Dimensional Anti-Icing.....	13
6.6.	Droplet Distribution.....	14
6.7.	Collection Efficiencies and Droplet Trajectories	15
6.8.	Static Pressure/Altitude.....	15
6.9.	Temperature	16
6.10.	Relative Humidity	16
6.11.	Printer Flags	16
6.12.	Geometry Input.....	16
6.13.	De-icer Input	18
6.14.	Pneumatic Boot Option.....	21
6.15.	Additional Input Files	21
Chapter 7:	Interactive Input.....	23
7.1.	Enter Input File Name.....	23
7.2.	Enter File Name for Body 1.....	24
7.3.	Confirmation of Warnings	24
7.4.	Enter De-icer File Name for Body 1	25
7.5.	Confirmation of Other Warnings	25
Chapter 8:	Main Input File	27
8.1.	Listing Variables in a Namelist.....	27
8.1.1.	Variable not in namelist error	27
8.1.2.	Namelist Listed Out of Order or Missing Namelist.....	28
8.2.	Line 1	28
8.3.	LEW20 Namelist.....	28
8.3.1.	ITIMFL	29
8.3.2.	TSTART	30
8.3.3.	TSTOP	31
8.3.4.	IBOD.....	31
8.3.5.	IFLO	32
8.3.6.	DSMN	33
8.3.7.	NPL.....	34
8.3.8.	RHOP.....	34

8.3.9.	SLD.....	35
8.3.10.	IGRID	35
8.3.11.	IDEICE	36
8.3.12.	ICP.....	37
8.3.13.	IBETA.....	38
8.3.14.	IHTC	39
8.3.15.	IQEX.....	39
8.3.16.	IBOOT	40
8.4.	DIST Namelist.....	42
8.4.1.	FLWC	42
8.4.2.	DPD	42
8.5.	ICE1 Namelist	44
8.5.1.	CHORD	44
8.5.2.	AOA.....	44
8.5.3.	VINF.....	45
8.5.4.	LWC	45
8.5.5.	TINF	46
8.5.6.	PINF.....	47
8.5.7.	RH.....	48
8.5.8.	GRAV	48
8.5.9.	SREF.....	48
8.6.	LPRNT Namelist.....	49
8.6.1.	FPRT.....	49
8.6.2.	HPRT	49
8.6.3.	BPRT	50
8.6.4.	EPRT.....	50
8.6.5.	MPRT	50
8.6.6.	TPRT.....	50
8.6.7.	IDBF	51
8.6.8.	KWARN	51
8.7.	RDATA Namelist.....	52
8.7.1.	ISCOLC, JSCOLC, KSCOLC, SSLOPC, SZEROC.....	52
8.7.2.	ISCOLB, JSCOLB, KSCOLB, SSLOPB, SZEROB.....	54
8.7.3.	ISCOLH, JSCOLH, KSCOLH, SSLOPH, SZEROH.....	55
8.7.4.	ISCOLX, JSCOLX, KSCOLX, SSLOPX, SZEROX.....	56
8.8.	BOOT Namelist.....	59
8.8.1.	XBOOTUP.....	59
8.8.2.	XBOOTLOW	59
8.8.3.	HRES	59
8.9.	Complete Example Case Input File.....	60
8.10.	Extent of Data in Validation Database	62
Chapter 9:	Body Geometry Input.....	63
9.1.	Number of Points.....	67
9.2.	Blank Lines	68
9.3.	Too Few Points.....	68
9.4.	Too Many Points	68

9.5.	Body Not Closed	68
9.6.	Small Point Spacing.....	69
9.7.	Body Input Backwards.....	69
9.8.	Large Angles Between Neighboring Segments	69
9.9.	Intersecting Bodies	70
9.10.	Concentric Bodies	71
9.11.	Bodies Out of Order	71
9.12.	Bodies Improperly Sized.....	71
Chapter 10:	Optional Input Files	73
10.1.	De-icer Input File	73
10.1.1.	De-icer Section 1	73
10.1.2.	De-icer Section 2.....	81
10.1.3.	De-icer Section 3.....	86
10.1.4.	De-icer Section 4.....	89
10.1.5.	De-icer Section 5.....	114
10.1.6.	Complete example for de-icer input file.....	120
10.2.	Hot Air Heat Transfer Coefficient Input File (hi.inp)	126
10.3.	Hot Air Heat Flux Input File (qain.inp).....	127
10.4.	3D Streamline Input file (stream.inp).....	127
10.5.	Grid Input Files (XY.PLT and Q.PLT).....	128
10.5.1.	Grid read statements (XY.PLT)	128
10.5.2.	Flow solution read statements (Q.PLT).....	128
10.6.	Pressure Coefficient Input File (rflow.inp).....	130
10.7.	Collection Efficiency Input File (rbeta.inp).....	130
10.8.	External Heat Transfer Coefficient Input File (rhtc.inp)	131
10.9.	External Heat Flux Input File (qextin.inp).....	131
10.10.	Surface Temperature Input File (tsurf.inp)	131
Chapter 11:	Output Files	133
11.1.	beta.dat.....	133
11.2.	dens.dat	134
11.3.	dyice.dat.....	134
11.4.	final1.dat	134
11.5.	final2.dat, final3.dat, final4.dat, final5.dat.....	134
11.6.	fixed.dat	134
11.7.	flow.dat	135
11.8.	fract.dat	135
11.9.	htc.dat.....	135
11.10.	ice1.dat.....	135
11.11.	ice2.dat, ice3.dat, ice4.dat, ice5.dat.....	136
11.12.	imp.dat	136
11.13.	junk.dat.....	136
11.14.	limit.dat	137
11.15.	mass.dat.....	137
11.16.	misc.dat	137
11.17.	pres.dat.....	137
11.18.	qgener.dat.....	138

11.19.	temp.dat.....	138
11.20.	thick.dat.....	138
11.21.	traj1.dat	138
11.22.	traj2.dat, traj3.dat, traj4.dat, traj5.dat	139
11.23.	xkinit.dat	139
11.24.	xkinit2.dat	139
Chapter 12:	Additional Output Files.....	141
12.1.	splash.dat.....	141
12.2.	noice.dat.....	141
12.3.	ctemp.dat.....	142
12.4.	geometry.dat.....	142
12.5.	force.dat	142
12.6.	force2.dat.....	142
12.7.	qopt.dat.....	143
12.8.	dummy.inp	143
12.9.	temps1.dat, temps2.dat, temps3.dat, temps4.dat, temps5.dat, temps6.dat, temps7.dat	143
12.10.	fort.54.....	144
12.11.	fort.80, fort.81, etc.....	144
12.12.	piccolo1.dat	145
12.13.	piccolo2.dat	145
Chapter 13:	Utility Programs.....	147
13.1.	Program THICK.....	147
13.1.1.	Interactive Input for Program THICK.....	147
13.1.2.	File Format for Clean Geometry and Iced Geometry.....	148
13.1.3.	THICK output files.....	149
13.1.4.	THICK Judgment Errors.....	152
13.2.	Program EXPAND	155
13.2.1.	Interactive Input for Program EXPAND	155
13.2.2.	File Format for Input and Output Geometries.....	156
13.3.	Program CONVERT.....	156
13.3.1.	Interactive Input for Program CONVERT.....	156
13.3.2.	Format for the LEWICE 1.6 Input File	157
13.3.3.	Warning Message for CONVERT	157
13.4.	Program CONDEICE	157
13.4.1.	Interactive Input for Program CONDEICE	157
13.4.2.	Format for the LEWICE 2.2.2 De-icer Input File.....	158
13.5.	Programs TOBINARY, TOBINARYM	158
13.6.	Programs TOASCII, TOASCIIM.....	159
13.7.	Program ANTICON	159
13.8.	Program ANTICON2.....	159
13.9.	Program REDUCE	160
13.10.	Program LewCon.....	160
13.11.	LewCon Example Case.....	161
13.12.	LewCon Missing Features	163
Chapter 14:	Runtime Errors	165

14.1.	Flow Solver Errors.....	165
14.2.	Errors when Using Grid Solutions	166
14.3.	Trajectory Errors	166
14.4.	Energy Balance Errors.....	168
14.5.	Ice Growth Errors.....	169
14.6.	De-icer Routine Errors.....	171
14.6.1.	Other de-icer routine messages	173
Chapter 15:	Example Cases.....	175
15.1.	Case 1: Run 072501 from Validation Report.....	175
15.2.	Case 2: Langmuir ‘D’ distribution	179
15.3.	Case 3: Run 072504 from Validation Report.....	182
15.4.	Case 4: Run 072504 without automated time step and fewer time steps	186
15.5.	Case 5: Run 072504 with Larger Point Spacing	190
15.6.	Case 6: Cylinder Benchmark Case	194
15.7.	Case 7: Exceedence condition.....	199
15.8.	Case 8: Exceedence condition with Langmuir ‘D’ distribution.....	203
15.9.	Case 9: Exceedence condition with splashing model.....	208
15.10.	Case 10: Comparison of Splashing Model with Impingement Data	213
15.11.	Case 11: Three body example	216
15.12.	Case 12: Pneumatic boot case with freezing rain distribution.	222
15.13.	Case 13: First benchmark conditions with an evaporative hot air anti-icer.....	226
15.14.	Case 14: First benchmark condition with a running wet electrothermal anti-icer.	233
15.15.	Case 15: Example Using Detailed Thermal Module	240
15.16.	Case 16: Laminar Example Using Detailed Thermal Module	254
15.17.	Case 17: Running Wet Example Using Detailed Thermal Module	265
15.18.	Case 18: Evaporative Example Using Detailed Thermal Module	276
15.19.	Case 19: Hot Air Evaporative Example Using Detailed Thermal Module.....	287
15.20.	Case 20: Hot Air Piccolo Tube Example Using Detailed Thermal Module.....	289
15.21.	Case 21: Hot Air Example Interfaced with LEWICE3D.....	291
15.21.1.	Procedure for 2.5D Anti-Icing and De-Icing Analysis.....	303
15.22.	Case 22: Grid input.....	305
15.22.1.	Grid Input Problems	309
15.22.2.	LEWICE Errors Associated with Grid Input Problems.....	310
15.23.	Case 23: Automated Navier-Stokes Ice Accretion Using ICEG2D	311
15.23.1.	ICEG2D Process.....	325
15.23.2.	ICEG2D Input Problems.....	328
15.23.3.	Using Other Flow Solvers in ICEG2D	330
Chapter 16:	User Tips and Notes.....	331
16.1.	Old Input Files.....	331
16.2.	Input File Errors	331
16.3.	Porting ASCII Files	331
16.4.	PC Application	332
16.5.	Listing Variables in a Namelist.....	332
16.6.	Multiple Bodies / Multi-Element Airfoils.....	333
16.7.	Grid-Based Flow Solutions.....	333
16.8.	Chord Length.....	333

16.9. Multiple Stagnation Points.....	333
16.10. Flow Module Limitation.....	333
16.11. Ice Density	334
16.12. Icing Limit.....	334
Chapter 17: Procedure for Acquiring LEWICE or other NASA Icing Software.....	335
17.1. Current Address for Icing Software Requests.....	335
Chapter 18: Sample Software Request Letter	337
Chapter 19: Technical Support.....	339
Chapter 20: References	340
Chapter 21: Appendix A	343
21.1. Functional Requirements for LEWICE version 3.2	343
21.2. Geometry Smoothing.....	344
21.2.1. Description/Purpose.....	344
21.2.2. Smoothing Requirements.....	344
21.3. Point Generation.....	345
21.3.1. Description/Purpose.....	345
21.3.2. Point Generation Requirements	345
21.4. Ice Addition.....	346
21.4.1. Description/Purpose.....	346
21.4.2. Ice Addition Requirements	346
21.5. Trajectory Integration	347
21.5.1. Description/Purpose.....	347
21.5.2. Trajectory Integration Requirements.....	347
21.6. Droplet Breakup*	351
21.6.1. Description/Purpose.....	351
21.6.2. Droplet Breakup Requirements.....	351
21.7. Droplet Splashing	353
21.7.1. Description/Purpose.....	353
21.7.2. Droplet Splashing Requirements.....	353
21.8. Impingement Limits	357
21.8.1. Description/Purpose.....	357
21.8.2. Impingement Limit Requirements.....	357
21.9. Collection Efficiency	359
21.9.1. Description/Purpose.....	359
21.9.2. Collection Efficiency Requirements.....	359
21.10. Other Droplet Trajectory Requirements	361
21.10.1. Description/Purpose.....	361
21.10.2. Other Droplet Trajectory Requirements	361
21.11. Potential Flow	363
21.11.1. Description/Purpose.....	363
21.11.2. Potential Flow Requirements	363
21.12. Grid-based Flow	365
21.12.1. Description/Purpose.....	365
21.12.2. Grid-based Flow Requirements.....	365
21.13. Compressibility Correction.....	367
21.13.1. Description/Purpose.....	367

21.13.2. Compressibility Requirements	367
21.14. Stagnation Point	370
21.14.1. Description/Purpose.....	370
21.14.2. Stagnation Point Requirements	370
21.15. Roughness Height.....	371
21.15.1. Description/Purpose.....	371
21.15.2. Roughness Height Requirements	371
21.16. Heat Transfer Coefficient	373
21.16.1. Description/Purpose.....	373
21.16.2. Heat Transfer Coefficient Requirements	373
21.17. Energy Balance Equation.....	379
21.17.1. Description/Purpose.....	379
21.17.2. Energy Balance Equation Requirements	379
21.18. Mass Balance Equation.....	385
21.18.1. Description/Purpose.....	385
21.18.2. Mass Balance Equation Requirements	385
21.19. One Dimensional Anti-Icing	389
21.19.1. Description/Purpose.....	389
21.19.2. One Dimensional Anti-Icing Requirements.....	389
21.20. Effective Heat Transfer Coefficient	393
21.20.1. Description/Purpose.....	393
21.20.2. Effective Heat Transfer Requirements	393
21.21. Internal Heat Transfer Coefficient.....	397
21.21.1. Description/Purpose.....	397
21.21.2. Internal Heat Transfer Requirements	397
21.22. Two Dimensional De-icing/Anti-Icing.....	399
21.22.1. Description/Purpose.....	399
21.22.2. Two Dimensional De-Icing Requirements	399
21.23. Pneumatic Boot Requirements	405
21.23.1. Description/Purpose.....	405
21.23.2. Boot Requirements	405
21.24. Error Checking	407
21.24.1. Description/Purpose.....	407
21.24.2. Error Checking Requirements.....	407
References for Appendix A	419
Subject Index	423

List of Tables

Table 1:	Example LEW20 Namelist	41
Table 2:	Example of bad input for FLWC	42
Table 3:	Example of correct input for FLWC	42
Table 4:	Example DIST Namelist.....	44
Table 5:	Example ICE1 Namelist	49
Table 6:	Example LPRNT Namelist	52
Table 7:	Example input for RDATA namelist.....	58
Table 8:	Example Test Input File	60
Table 9:	Range of Experimental Data for Validation	62
Table 10:	Example Body Geometry (NACA 23014).....	63
Table 11:	First section of data file with comments.....	74
Table 12:	Sample data for each layer.....	76
Table 13:	Sample data for each layer with comments	76
Table 14:	Sample data for each section with comments.....	78
Table 15:	Example data for first section of de-icer input.....	80
Table 16:	Example data line	83
Table 17:	Complete input data for de-icer section 2.....	85
Table 18:	Complete input data for de-icer section 2 using temperature controlled heaters.....	86
Table 19:	Example input for boundary condition data.....	89
Table 20:	Sample input line for hot air anti-icer.....	91
Table 21:	Sample input line for first piccolo tube input	93
Table 22:	Constants for piccolo tube correlation.....	93
Table 23:	Sample input line for piccolo tube correlation.....	95
Table 24:	Sample input line for piccolo hole locations.....	96
Table 25:	Piccolo liner input	97
Table 26:	Sample input for anti-icing flags.....	100
Table 27:	Cross Reference for Allowable Flags.....	103
Table 28:	Example input for the conduction and phase change flag input line.....	105
Table 29:	Example input for 1D anti-icing approximation (IDEICE = 1).....	106
Table 30:	Example input for rotating body information.	107
Table 31:	Example input for time step data input line.....	109
Table 32:	Example data for spray time data input line	110
Table 33:	Example input for hot air data file characteristics.....	111
Table 34:	Example input for wall heat flux data file characteristics	112
Table 35:	Example input for surface temperature data file characteristics	112
Table 36:	Example input for 3D streamline data file characteristics.....	113
Table 37:	Complete input example for fourth data section	113
Table 38:	Example input for PLOT3D style output.....	115
Table 39:	Example input for temperature output flags	118
Table 40:	Example of complete de-icer input file	120
Table 41:	Example of Data Format in Output Files.....	133
Table 42:	Excerpt of Script Commands	151
Table 43:	Main Input File for Example Case 1	176

Table 44:	Main Input File for Example Case 2	179
Table 45:	Main Input File for Example Case 3	183
Table 46:	Main Input File for Example Case 4	186
Table 47:	Main Input File for Example Case 5	191
Table 48:	Main Input File for Example Case 6	195
Table 49:	Main Input File for Example Case 7	199
Table 50:	Main Input File for Example Case 8	203
Table 51:	Main Input File for Example Case 9	208
Table 52:	Main Input File for Example Case 10	213
Table 53:	Main Input File for Example Case 11	217
Table 54:	Main Input File for Example Case 12	223
Table 55:	Main Input File for Example Case 13	227
Table 56:	De-ice input file for Example Case 13	228
Table 57:	Main Input File for Example Case 14	234
Table 58:	De-ice Input File for Example Case 14	235
Table 59:	Main Input File for Example Case 15	241
Table 60:	De-ice Input File for Example Case 15	242
Table 61:	Main Input File for Example Case 16	255
Table 62:	De-ice Input File for Example Case 16	256
Table 63:	Main Input File for Example Case 17	266
Table 64:	De-ice Input File for Example Case 17	267
Table 65:	Main Input File for Example Case 18	277
Table 66:	De-ice Input File for Example Case 18	278
Table 67:	Main Input File for Example Case 21-1	292
Table 68:	De-ice Input File for Example Case 21-1	293
Table 69:	Main Input File for Example Case 22	306
Table 70:	Main Input File for Example Case 23	312
Table 71:	ICEG2D Input File for Example Case 23.....	313
Table 72:	ICEG2D Batch Script for Example Case 23.....	313
Table 73:	Path Definitions in .tcshrc File.....	328
Table 74:	Example of batch structure	329
Table 75:	lewice.inp Input File.....	329
Table 76:	wind.inp input File	329

List of Figures

Figure 1:	Test Conditions in Database Used for Software Validation.....	46
Figure 2:	Example of a 45 degree panel angle	70
Figure 3:	Example of a 135 degree panel angle	70
Figure 4:	Grid block in the computational domain (left) and the physical domain (right).....	73
Figure 5:	Additional thickness case	78
Figure 6:	Example of heater cycle types.....	83
Figure 7:	Temperature location for temperature controlled heaters	84
Figure 8:	Boundaries on Sample Geometry	87
Figure 9:	Boundaries on Computational Geometry.....	87
Figure 10:	First hot air model.....	90
Figure 11:	Side and front view of piccolo tube system.....	91
Figure 12:	Piccolo tube variables (NHOLE = 3).....	92
Figure 13:	XHOLE and YHOLE definitions.....	96
Figure 14:	Piccolo tube with liner.....	96
Figure 15:	Potential IOTYPE = 1 (thermocouple) locations in a grid block	117
Figure 16:	Representation of 3D streamline data	128
Figure 17:	Diagram of temperature output for IOTYPE = 4 and IOTYPE = 5 output.....	144
Figure 18:	Horn Angle Definition for THICK	150
Figure 19:	Ice Shape Misalignment	152
Figure 20:	Shifted Ice Shape Plot	153
Figure 21:	Expanded View of Airfoil Misalignment.....	153
Figure 22:	THICK Horn Angle Illustration.....	154
Figure 23:	Ice Shape for Example Case 1	177
Figure 24:	Ice Thickness for Example Case 1	177
Figure 25:	Heat Transfer Coefficient for Example Case 1.....	178
Figure 26:	Pressure Coefficient for Example Case 1	178
Figure 27:	Collection Efficiency for Example Case 1	179
Figure 28:	Ice Shape for Example Case 2	180
Figure 29:	Ice Thickness for Example Case 2.....	181
Figure 30:	Heat Transfer Coefficient for Example Case 2.....	181
Figure 31:	Pressure Coefficient for Example Case 2.....	182
Figure 32:	Collection Efficiency for Example Case 2	182
Figure 33:	Ice Shape for Example Case 3	184
Figure 34:	Ice Thickness for Example Case 3.....	184
Figure 35:	Heat Transfer Coefficient for Example Case 3.....	185
Figure 36:	Pressure Coefficient for Example Case 3.....	185
Figure 37:	Collection Efficiency for Example Case 3	186
Figure 38:	Ice Shape for Example Case 4	188
Figure 39:	Ice Thickness for Example Case 4.....	188
Figure 40:	Heat Transfer Coefficient for Example Case 4.....	189
Figure 41:	Pressure Coefficient for Example Case 4.....	189
Figure 42:	Collection Efficiency for Example Case 4	190
Figure 43:	Ice Shape for Example Case 5	192

Figure 44:	Ice Thickness for Example Case 5.....	192
Figure 45:	Heat Transfer Coefficient for Example Case 5.....	193
Figure 46:	Pressure Coefficient for Example Case 5.....	193
Figure 47:	Collection Efficiency for Example Case 5.....	194
Figure 48:	Ice Shape for Example Case 6.....	196
Figure 49:	Ice Thickness for Example Case 6.....	196
Figure 50:	Heat Transfer Coefficient for Example Case 6.....	197
Figure 51:	Pressure Coefficient for Example Case 6.....	197
Figure 52:	Collection Efficiency for Example Case 6.....	198
Figure 53:	Collection Efficiency as a Function of Wrap Distance for Example Case 6.....	198
Figure 54:	Ice Shape for Example Case 7.....	201
Figure 55:	Ice Thickness for Example Case 7.....	201
Figure 56:	Heat Transfer Coefficient for Example Case 7.....	202
Figure 57:	Flow Solution for Example Case 7.....	202
Figure 58:	Collection Efficiency for Example Case 7.....	203
Figure 59:	Ice Shape for Example Case 8.....	205
Figure 60:	Ice Thickness for Example Case 8.....	205
Figure 61:	Comparison of Ice Shapes for Monodispersed and Langmuir ‘D’ Cases.....	206
Figure 62:	Comparison of Ice Thicknesses for Monodispersed and Langmuir ‘D’ Cases.....	206
Figure 63:	Heat Transfer Coefficient for Example Case 8.....	207
Figure 64:	Pressure Coefficient for Example Case 8.....	207
Figure 65:	Collection Efficiency for Example Case 8.....	208
Figure 66:	Ice Shape for Example Case 9.....	210
Figure 67:	Ice Thickness for Example Case 9.....	210
Figure 68:	Comparison of Ice Shapes for Splashing and Non-splashing Cases.....	211
Figure 69:	Comparison of Ice Thicknesses for Splashing and Non-splashing Cases.....	211
Figure 70:	Heat Transfer Coefficient for Example Case 9.....	212
Figure 71:	Pressure Coefficient for Example Case 9.....	212
Figure 72:	Collection Efficiency for Example Case 9.....	213
Figure 73:	Collection Efficiency for Example Case 10.....	215
Figure 74:	Heat Transfer Coefficient for Example Case 10.....	215
Figure 75:	Pressure Coefficient for Example Case 10.....	216
Figure 76:	Ice Shape on Slat and Main Elements for Example Case 11.....	218
Figure 77:	Ice Shape on Flap for Example Case 11.....	218
Figure 78:	Ice Thickness on Slat for Example Case 11.....	219
Figure 79:	Ice Thickness on Main Element for Example Case 11.....	219
Figure 80:	Ice Thickness on Flap for Example Case 11.....	220
Figure 81:	Slat Heat Transfer Coefficient for Example Case 11.....	220
Figure 82:	Main Heat Transfer Coefficient for Example Case 11.....	221
Figure 83:	Flap Heat Transfer Coefficient for Example Case 11.....	221
Figure 84:	Pressure Coefficient for Example Case 11.....	222
Figure 85:	Collection Efficiency for Example Case 11.....	222
Figure 86:	Ice Shape With and Without Boot for Example Case 12.....	224
Figure 87:	Ice Thickness With and Without Boot for Example Case 12.....	225
Figure 88:	Heat Transfer Coefficient for Example Case 12.....	225
Figure 89:	Pressure Coefficient for Example Case 12.....	226

Figure 90:	Collection Efficiency with Pneumatic Boot for Example Case 12.....	226
Figure 91:	Heat Requirement for the Evaporative Hot Air System in Example Case 13.....	232
Figure 92:	Surface Temperature for the Evaporative Hot Air System in Example Case 13....	232
Figure 93:	Hot Air Temperature for the Evaporative Hot Air System in Example Case 13 ...	233
Figure 94:	Effective Heat Transfer in Example Case 13	233
Figure 95:	Heat Requirement for the Running Wet Thermal De-icer in Example Case 14	239
Figure 96:	Heater Temperature for the Running Wet Thermal De-icer in Example Case 14..	239
Figure 97:	Effective Heat Transfer Coefficient in Example Case 14	240
Figure 98:	Schematic of Heaters for Example Case 15	248
Figure 99:	Heater Power for Example Case 15	249
Figure 100:	Heat Transfer Coefficient for Example Case 15.....	249
Figure 101:	Pressure Coefficient for Example Case 15	250
Figure 102:	Collection Efficiency for Example Case 15	250
Figure 103:	Heater A Temperature for Example Case 15.....	251
Figure 104:	Heater B Temperature for Example Case 15.....	251
Figure 105:	Heater C Temperature for Example Case 15.....	252
Figure 106:	Heater D Temperature for Example Case 15.....	252
Figure 107:	Heater E Temperature for Example Case 15	253
Figure 108:	Heater F Temperature for Example Case 15	253
Figure 109:	Heater G Temperature for Example Case 15.....	254
Figure 110:	Heat Transfer Coefficient for Example Case 16.....	261
Figure 111:	Heater A Temperature for Example Case 16.....	262
Figure 112:	Heater B Temperature for Example Case 16.....	262
Figure 113:	Heater C Temperature for Example Case 16.....	263
Figure 114:	Heater D Temperature for Example Case 16.....	263
Figure 115:	Heater E Temperature for Example Case 16	264
Figure 116:	Heater F Temperature for Example Case 16	264
Figure 117:	Heater G Temperature for Example Case 16.....	265
Figure 118:	Heater A Temperature for Example Case 17.....	272
Figure 119:	Heater B Temperature for Example Case 17.....	273
Figure 120:	Heater C Temperature for Example Case 17.....	273
Figure 121:	Heater D Temperature for Example Case 17.....	274
Figure 122:	Heater E Temperature for Example Case 17	274
Figure 123:	Heater F Temperature for Example Case 17	275
Figure 124:	Heater G Temperature for Example Case 17.....	275
Figure 125:	Comparison of One-Dimensional Assumption to 2D Result	276
Figure 126:	Heater A Temperature for Example Case 18.....	283
Figure 127:	Heater B Temperature for Example Case 18.....	284
Figure 128:	Heater C Temperature for Example Case 18.....	284
Figure 129:	Heater D Temperature for Example Case 18.....	285
Figure 130:	Heater E Temperature for Example Case 18	285
Figure 131:	Heater F Temperature for Example Case 18	286
Figure 132:	Heater G Temperature for Example Case 18.....	286
Figure 133:	Comparison of 2D Anti-Icing Results to 1D Anti-Icing Results.....	287
Figure 134:	Jet Locations on Bleed Air Model	288
Figure 135:	Internal Heat Transfer Coefficients for Bleed Air Case.....	288

Figure 136:	Surface Temperature Distribution for Bleed Air Case.....	289
Figure 137:	Piccolo Tube on Bleed Air Model	290
Figure 138:	Internal Heat Transfer Coefficients for Bleed Air Case.....	290
Figure 139:	Surface Temperature Distribution for Bleed Air Case.....	291
Figure 140:	Engine Inlet Cross-Section for Example Case 21-1.....	299
Figure 141:	Internal Heat Transfer Coefficient for Example Case 21-1.....	299
Figure 142:	Surface Temperature Distribution for Example Case 21-1	300
Figure 143:	Ice Shape for Example Case 21-1	300
Figure 144:	Ice Thickness for Example Case 21-1.....	301
Figure 145:	Heat Transfer Coefficient for Example Case 21-1.....	301
Figure 146:	Pressure Coefficients for Example Case 21-1	302
Figure 147:	Collection Efficiency for Example Case 21-1	302
Figure 148:	3D Engine Inlet (Half-model) Surface Grid (Ice is Shaded).....	303
Figure 149:	Ice Shape for Example Case 22	307
Figure 150:	Ice Thickness for Example Case 22.....	308
Figure 151:	Heat Transfer Coefficient for Example Case 22.....	308
Figure 152:	Pressure Coefficient for Example Case 22.....	309
Figure 153:	Collection Efficiency for Example Case 22	309
Figure 154:	Ice Shape for Example Case 23	323
Figure 155:	Ice Thickness for Example Case 23.....	323
Figure 156:	Heat Transfer Coefficient for Example Case 23.....	324
Figure 157:	Pressure Coefficient for Example Case 23.....	324
Figure 158:	Collection Efficiency for Example Case 23	325

Chapter 1: Introduction

The evaluation of both commercial and military aircraft systems in icing conditions continues to be important during both design and certification. These systems can be 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 software to allow more complete evaluation in the design phases of the aircraft.

The computer software LEWICE contains 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 software 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 software. 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 report³.

The thermal module of LEWICE 3.2 can model any number of heaters, any heater chordwise length, and any heater gap desired. The heaters may be fired in unison, or they may be cycled with periods independent of each other. The heater intensity may also be varied. In addition, the user may specify any number of layers and thicknesses depthwise into the airfoil. Thus, the new addition has maximum flexibility in modeling virtually any electrothermal de-icer installed into any airfoil. It can also model a bleed air anti-icing system where the user can specify the heat

flux (or heat transfer coefficient) from the bleed air or the user can set an “optimal” operating temperature and have the program iteratively solve for the required heat flux

The new module has several enhancements over previous de-icing and anti-icing models. These features include heaters (or bleed air supply) which are temperature-controlled, heaters with a variable (temperature-dependant) resistance, improved stability as well as improved ice shedding and runback freezing. A correlation has also been added specifically for piccolo tube applications.

As mentioned in the previous paragraph, the model simulates both shedding and runback. With the runback capability, it can simulate the anti-icing mode of heater performance, as well as detect icing downstream of the heaters due to runback over unprotected portions of the geometry. The shedding model allows for the ice to be shed as a whole, by control volume or by heater section.

This version of LEWICE can be run in two modes. In the first mode, all heat transfer is considered to be caused by conduction, but no heaters are firing. This is equivalent to the LEWICE 2.0 model. In the second mode, conduction heat transfer where the heaters are engaged is modeled, with subsequent ice shedding. When run in the first mode, there is virtually identical agreement with LEWICE 2.0 in the prediction of accreted ice shapes. Output from the software has been expanded to include support for PLOT3D, TECPLOT and selected temperature distributions either depthwise at a specific chord location or chordwise at a specific depth.

The purpose of the current research has been to add several features to the program so that the user could model the complete icing problem. First, LEWICE 2.0 had an option where the user could bypass the potential flow module and read in results from a Naviér-Stokes program. The user simply selects the grid option in the current input file and a grid-based flow solver of the user’s choice can be used with any of the other modules of LEWICE. In LEWICE 3.2, an option has been created for automated use of a Naviér-Stokes program such as WIND⁹ by use of the program ICEG2D¹⁰. This option allows for more accurate generation of ice shapes albeit at an increase in computation time. Second, the thermal de-icing and anti-icing options in LEWICE 2.2.2¹¹ have been expanded to include additional piccolo tube capabilities. One of these additional options allows the user to select a piccolo tube correlation for internal heat transfer coefficients¹². Third, an empirical model was added to account for the physical effects of supercooled large droplets (SLD)¹³. The model allows for breakup of drops prior to impact, the splashing of drops on the surface, and the reimpingement of splashed droplets. Finally, an option was added to model a simplistic pneumatic boot so that the user could assess the effect of ice impingement behind protected surfaces. This option is primarily of use for SLD conditions as well.

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. A DOS Shell is called a “Command Prompt” by Windows XP and can be opened by selecting “Start” in the lower left hand corner, then selecting the “Programs” submenu then selecting the “Accessories” submenu and finally selecting “Command Prompt”.

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 Windows machine. Alternatively, the user can use the Windows interface for any of the commands shown. Windows XP refers to a DOS Shell window as a “Command Prompt” and can be accessed from the Program menu under the “Accessories” submenu.

- 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 output 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® with a 2.4GHz Pentium IV® CPU and 512MB of RAM running Windows2000®. This machine took 20 seconds to run the first example case provided. LEWICE 2.0 was run successfully on a Compaq® with a 133MHz Pentium® CPU and 24MB of RAM running Windows95. That machine took 9 minutes to run the first example case provided. It is believed that LEWICE 3.2 should run on any system that had run LEWICE 2.0. 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 3.2 was developed on a Silicon Graphics Octane running IRIX 6.5. Various changes were made in this software to make conversion to personal computers much easier. The validation report³ 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 900 KB of hard disk space for the executable, and several megabytes for output files. The third 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 software has been successfully compiled and tested on LINUX PCs, 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 Octane 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 3.2 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 tests³ in order to establish the validity of the modified code. Several different combinations of compiler flags were used during testing to establish software 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 Visual Fortran 6.0.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: Changes From Version 2.0

LEWICE 3.2 differs from LEWICE 2.0 in the following areas: 1) the procedure for analyzing hot air or electrothermal ice removal (de-icing) or prevention (anti-icing) systems has been added; 2) the procedure for interfacing with Naviér-Stokes solvers has been enhanced; 3) an attempt has been made to model supercooled large droplet (SLD) phenomena; and 4) a simplistic model of a pneumatic boot has been added. Version 3.2 differs from LEWICE 3.0 primarily due to updates in the SLD empirical model. Other aspects of the software have seen minor changes and the ice shapes generated using default options in LEWICE 3.2 should be very similar if not the same as those generated using LEWICE 2.0. However, enough discrepancies exist such that a complete revalidation is warranted to consider this version validated. This validation has been performed and will be published at a later date. Section 7.4 will describe the additional interactive input needed for the de-icing/anti-icing option. Section 8.3.1 will describe a change made to the automatic time step criteria. Section 8.3.9 will describe a flag for activating a model for supercooled large droplets (SLD). Section 8.3.11 will describe an additional setting for the input flag IDEICE. Sections 8.3.12 through 8.3.16 describe additional input flags added since the last release. Section 8.5.9 will describe different options for outputting wrap distance. Section 8.6.8 will describe an option for suppressing some confirmation warnings. Sections 8.7 and 8.8 will describe the addition of two new namelists which can be read. Section 10.1 will cover the inputs for the revised de-icing function. Sections 10.2 through 10.10 describe additional input files which can be read. Chapter 12 will cover the additional output files for this option. Chapter 13 describes some additional utility programs written for this version. Section 14.6 will summarize the runtime errors for the de-icer input. Sections 15.9 and 15.10 will describe examples using the SLD option. Section 15.12 will cover a boot example case. Sections 15.13 to 15.21 will describe de-icing and anti-icing example cases. Sections 15.22 and 15.23 will cover the Naviér-Stokes examples. Chapter 16 will summarize the user tips for these options.

Note: At a minimum, the user must add the following four lines to the end of the main input file for existing LEWICE 2.0 input files for them to work with LEWICE 3.2:

```
&RDATA
&END
&BOOT
&END
```

See Section 8.7 and 8.8 for details of the new namelists.

Note: LEWICE 2.2 users need only add the last two lines while LEWICE 2.4 users do not have to change the LEWICE input files. However, flag SREF is now input in the ICE1 namelist.

Note: De-icer inputs files must be modified to include the following data:

Below the line containing tairh, airmd, airup, and xairmd, add:

```
nhole  dhole    dtube  rdist  ycase  xuhnd  xlhnd
  3    0.00132  0.0381  50.    0.0    0.23   0.26
icase  hconst  rep    rp     zp    rrp    dp
  0     1.0   0.76   1.394  1.    0.    0.
```

```

xhole(m)    yhole(m)    zdist(m)
  0.0176    -0.033     0.0107
  0.000     0.000     0.0091
  0.0176    -0.033     0.0076
iliner  xuliner xlliner  xulst    xulnd  xllst  xllnd
  0      0.01   0.01   0.07546  0.1485 0.07546 0.1485

```

Line which read

```

i3d  iqain  ibleed  iopt
  0   0      0      0

```

must now include an additional variable

```

i3d  iqain  ibleed  iopt  itsur
  0   0      0      0     0

```

Below the line containing iscolq, jscolq, kscolq, sslopq and szeroq add:

```

iscolt, jscolt, kscolt, sslopt and szerot
1      2      2      1.d0    0.d0

```

See Section 10.1.4 for more information about the new variables.

5.1. Changes From Version 3.0

The SLD model reflect comparisons of calculated ice shapes and water impingement curves with the corresponding IRT experimental data. It is also believed that the current correlation behaves better when conditions outside the correlation database are selected as the trends appear to be qualitatively more appropriate. Details of the correlation are given in Appendix A.

The output files **piccolo1.dat** and **piccolo2.dat** now output effective heat transfer coefficient instead of internal heat transfer coefficient. This output allows for more seamless integration of LEWICE with Navier-Stokes solutions of the internal piccolo tube flow.

The utility program THICK has an improved horn-finding algorithm. Errors with THICK use reduced substantially during the comparison process with experimental data.

Two new conversion utility programs are now available. TOASCII and TOBINARYM are multi-element versions of TOASCII and TOBINARY.

A prototype user interface has been created called LewCon. This program takes input conditions from an Excel spreadsheet to run LEWICE and creates Excel charts of all LEWICE outputs. Multiple LEWICE runs can be ran in sequence by simply selecting multiple cases from the spreadsheet cases. See Section 13.10 for details.

Chapter 6: Recommended Practices

This section will list a set of recommended practices for running LEWICE 3.2. These recommendations are provided at various locations in this manual. They are summarized here for convenient reference by the user. Each paragraph in this chapter may contain a recommended practice which is not directly related to the surrounding notes. This section differs from Chapter 16 which lists user notes in that this section contains those notes considered most important for accurate use of the software.

6.1. 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 3.2, this is highly automated by the software, 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 report³. 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 $2 \cdot 10^{-4} \leq \text{DSMN} \leq 8 \cdot 10^{-4}$ 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 $8 \cdot 10^{-4}$ 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) (**ice1.dat**, **ice2.dat** etc.) to determine how many panels the program is using for each body.

It is recommended that the user supply approximately 100 or more points 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.

6.2. 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

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

where N is the number of time steps, LWC is the liquid water content, V is the velocity (air-speed), and ρ_{ice} is the ice density. The user should insure that they are using consistent units for this calculation.

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 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 3.2 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 software 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.

Time steps smaller than 30 seconds are discouraged as they can cause undesirable numerical effects. The user should not choose a time step this small unless the chord length is very small as well.

The procedure for the validation report set the number of time steps equal to the number of minutes of icing or 15, whichever was **lower**. This procedure is highly recommended to maintain consistency with the validation report results. The automated time step procedure will use this methodology by default.

The TSTART option can be useful for analysis of flight cases with variable conditions. TSTART can also be used in conjunction with the IGRID = 1 option to run the ICEG2D automatic grid generator.

6.3. 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 3.2 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 software. 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.

6.4. Number of Trajectories

The number of trajectories used in the impingement region is an input to the software. 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 software uses for the collection efficiency calculation may be different than the value input. The software is limited to one trajectory strike per panel. If more than one trajectory hits a given panel, only the first hit will be saved.

6.5. One-Dimensional 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 de-icer input file will be used for each body in a multi-element calculation. The desired surface temperature input in the de-icer input file must be above freezing (in Kelvin) for this option to work properly. NASA Glenn also had programs which performed more detailed analysis of de-icer and anti-icer performance. The LEWICE/Thermal program⁷ performed a 2D transient de-icer simulation and the ANTICE program⁸ performed a 2D steady-state anti-icing simulation. These two programs have been integrated into this version of LEWICE. If the user needs a more detailed analysis than provided with the one-dimensional anti-icing function, they are encouraged to try IDEICE = 2 to IDEICE = 4 options in the main input file.

When IDEICE = 1 and IEVAP = 2, the program will override the normal ice accretion process and calculate a residual ice shape with the 1D heat transfer assumption by using the heat input specified. If the amount of heat is high enough, no ice will form. For other IEVAP values, this routine does not affect the ice accretion routine. The software 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.

The results for a desired surface temperature at freezing may be inaccurate since the 1D anti-icing module has limited provisions to freeze surface water as compared to the LEWICE icing module.

For high water loadings, the program may not be able to calculate the evaporative temperature. This occurs because the evaporative temperature is approaching the boiling point of water. In this case, the user can select $IEVAP = 0$ and $TSURF = 370$. If all of the water did not evaporate, then the $IEVAP = 1$ option should be avoided.

6.6. Droplet Distribution

Most cases run with LEWICE 3.2 in the validation database use a single drop size, the Median Volume Diameter (MVD) for the flight condition. Although multiple drop size distributions can be run with LEWICE 3.2, execution times will be increased. The difference in ice shape and icing limits are documented in the validation report³. This report shows that the effect is very slight for single body simulations in Appendix C conditions. 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 distribution of five drop sizes, this feature essentially makes the software 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 and for SLD conditions with bi-modal distributions. 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 and in SLD conditions with bi-modal droplet distributions.

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 15d MVD d 270. **Although experimental data has been collected in this regime, the software has been calibrated (but not validated) using the existing data. Your conditions may not be covered in the current experimental database.** Caution should be exercised when running cases above 50 micron MVD. Due to the recent popularity of drop size inputs in these 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 experimental data is limited at these drop sizes. Since the results cannot be experimentally validated, the true accuracy of the results cannot be verified.

6.7. 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. This occurs since the calculated “leading edge” of a glaze ice shape will likely be at the tip of the upper ice horn and nowhere near the leading edge of the clean geometry.

The impingement limit listed in the output file “**imp.dat**” may not match the wrap distance 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.

The SLD flag may be used for any drop size as the models employed will have negligible effect on Appendix C drop sizes.

The baseline empirical model for splashing is based upon experiments performed at lower velocities (20 m/s and below). These correlations have been modified based upon collection efficiency and ice shape data taken in the Icing Research Tunnel (IRT). While the NASA database in SLD is growing, it is by no means comprehensive and the behavior of the correlations outside the validation test matrix is unknown.

6.8. 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

6.9. 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!

6.10. Relative Humidity

Relative humidity is not normally 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.

6.11. 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 $\text{TPRT} = 1$ and $\text{TPRT} = 2$ than the definition used in version 1.6.

If LEWICE has issued a warning when reading other input files, it will ask for a confirmation of those warnings. The syntax of these warning and error messages is similar to Section 7.3 and will not be repeated. LEWICE has the option to suppress the interactive confirmation of these warnings by using the KWARN flag described in Section 8.6.8. This allows for smoother operation of LEWICE when using batch scripts. However, it is not recommended for general use as these warnings may be indicative of legitimate problems with the input. Errors in the input files will still cause the program to stop and warnings will still be printed to the screen and to **junk.dat**.

6.12. 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 software 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. The recommended procedure for closing the trailing edge is to extend the airfoil slope. This will result in an airfoil slightly longer than the original airfoil. Do not create a vertical line segment to connect the trailing edge.

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.

LEWICE expects that the body will be input in dimensionless format (x/c , y/c). While the x -coordinates do not have to traverse from exactly $x/c = 0$ to exactly $x/c = 1$, they should reside in this general vicinity. Additionally, the y/c coordinates are generally input such that the leading edge of the airfoil is at $y/c = 0$ although this is not required. For multi-body airfoils, the chord has been defined using the airfoil in the retracted position. Therefore runs with slats employed will have x/c values less than zero and flaps employed will have x/c values greater than one.

6.13. De-icer Input

The program will only ask for the de-icer input file name for cases where IDEICE is not equal to zero.

The IDEICE = 2, IDEICE = 3 and IDEICE = 4 options will perform a detailed thermal analysis within the airfoil, ice, and water. The IDEICE = 2 option is recommended for failed thermal cases with significant ice accretion. The laminar flag, IDEICE = 3, provides consistently better temperature predictions for cases with a clean leading edge. However, it is a less conservative option and has a tendency to overpredict temperatures for some cases. The boundary layer trip flag IDEICE = 4 provides a compromise value between these two options.

The de-icer routines in LEWICE assume the body geometry is a clean airfoil with surface properties as described in Section 10.1. Therefore if the input geometry is an iced airfoil, this option may not run properly. The ICEG2D option may not work for de-icer cases as a result.

The output file “**noice.dat**” contains output from the simplified 1D thermal analysis. This analysis is provided for all cases where IDEICE > 0. When IDEICE = 1, it is the only thermal analysis performed. It is also performed for IDEICE = 2 to IDEICE = 4 cases as the 1D option does not take significant computation time. Also see Section 10.1.4.4 for additional capabilities when IDEICE = 2 to 4.

The de-icer geometry is specified by the user in the (rectangular) computational domain, not in the physical domain.

Unlike thermal conductivity and density, the specific heat of materials is often difficult to find in references. Manufacturers often do not measure (or list) this quantity in spec sheets. However, this parameter is only important for transient (time-dependant) cases.

Data is read free-form. The numbers do not have to reside in any particular column and do not have to line up with the comments. The numbers must be input in the correct order however.

The input units for LEWICE 3.2 are metric. Due to the relative scale of certain values, the input units can be watts or kilowatts.

$$\text{W/in}^2 * 1.55 = \text{kW/m}^2$$

Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K respectively.

The first layer input is the innermost layer and the last layer will be the top surface.

While performing de-icing or anti-icing simulations, make certain that the top layer contains the thermal properties of ice. If the ice melts, the software will change the properties accordingly.

For any icing, de-icing, or anti-icing simulation, the thickness given to the ice layer in this section will only be used to determine the grid spacing within the ice. The actual ice thickness at each grid point is part of the output from this module.

The total number of points in the normal direction is currently limited to 310 points. The total number of points is the sum of the points in each individual layer minus the number of layers, plus 2.

In the de-icer input file, only the lines with data are used by the software. The comments included in the examples are extraneous and are ignored by LEWICE 3.2.

The section data is entered from the left side of the computational domain toward the right side. In the physical domain, data is entered starting from the lower surface trailing edge clockwise toward the leading edge and ending back at the trailing edge.

The length of the end segments will be changed so that the total segment length equals the total wrap distance calculated by LEWICE 3.2. An identical amount will be added (or subtracted) from each end segment. Segments may be eliminated, if needed, to match the airfoil wrap distance.

The center of the middle section is **not** aligned with the body leading edge by default. This can be changed in the next set of input data using variable OFFSET. It is not necessary for the input data to align on a given column. This was done in the example cases to improve readability.

Comment lines have been added to improve readability. The software will ignore the comments and only read the data lines.

If there was an error reading data in the first section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

Normal operation of this software uses one parameter study per run ($IPAR = 1$). While a multiple parameter study option has been included, it is possible that output from a multiple parameter study run may be different from the output that would have been achieved if the case were run separately. This option has seen limited testing. Section 13.1.3.1 describes a potentially better way to perform a parameter study by using external scripts.

Temperature controlled heaters as modeled in this software do not necessarily determine the “optimum” heat flux. Initially, it uses the wattages input until the temperature exceeds the ON temperature. It then turns off all heat until the temperature drops below the OFF temperature. Heat is then turned on, but at a lower wattage. This process continues until the simulation time (TSTOP) is reached. If the simulation time is sufficiently long and the wattage has not changed for a long time, then the wattage reported in “**qopt.dat**” may be close to the optimum heat flux. **Results to date have been inconsistent for this option.**

To model a section which does not have a heater, simply input $QDE = 0$.

To model an electrothermal anti-icing system or parting strip, set $TOFF = 0$.

The heat transfer coefficient input at the top surface may be overwritten using the IBOUND flag in the next input section.

The inside heat transfer coefficient, H1, can be used to model a hot air anti-icer. Section 10.1.4 provides a better way to model of this type of system.

The wall heat flux, QW1, can be used to model a hot air anti-icer. Section 10.1.4 provides a better way to model of this type of system.

The IQW flag does not necessarily determine the “optimum” heat flux. Initially, it uses the wattages input until the temperature exceeds the ON temperature. It then turns off all heat until the temperature drops below the OFF temperature. Heat is then turned on, but at a lower wattage. This process continues until the simulation time (TSTOP) is reached. If the simulation time is sufficiently long and the wattage has not changed for a long time, then the wattage reported in “**qopt.dat**” may be close to the optimum heat flux. Also see the description of temperature controlled heaters in Section 10.1.2.2. **Results to date have been inconsistent for this option.**

IBLEED and IQAIN flags are most often used when the input data is known. Examples of this condition occur when experimental test data is available or when values are calculated by another program or via correlation. IQW is most often used when this data is not available.

Use of the liner flag (ILINER) will increase the velocity of the flow in that region using the Bernoulli equation. The increased velocity then provides an increase to the heat transfer coefficient through the correlation provided earlier. The validity of this assumption is not known.

The applicability of the jet correlations provided to piccolo tube anti-icing systems is unknown.

IBOUND values are ignored by the software unless IBC2 = 2 boundary condition is used.

Most of the cases performed to date (especially those which have experimental data) use INIT = 2 and ICOND = 1. Other combinations have seen limited testing.

If a node of ice melts, the software will automatically change the thermal properties of that node to water.

If phase change is considered (IGDE = 1), the length (ELDE) given to the top layer in Section 10.1.1.2 will only be used to define the normal direction grid spacing in the ice layer. In this case, the ice thickness at time = 0 will be zero.

Some combinations of flags in Table 28 may generate warning messages but the flag values will not be changed

To model a fixed-body problem, simply set RPM = 0.

The software uses the rotor data input for three calculations: 1) the rotational speed is used to calculate an increase in the aerodynamic heating term in the energy balance; 2) the rotational force is used to shed ice; and, 3) the rotational force is used to find the resultant force of the shed ice particle, which is used to track the particle after it sheds.

The rotating body information is not used by the potential flow solver in LEWICE 3.2 nor is the rotating body information used by the trajectory equation.

If STA = 0 or RPM = 0, then there is no difference between IROT = 0 and IROT = 1 (rotational force is zero).

De-icer time steps less than 10^{-4} seconds are not allowed as the software will run very slowly at such de-icer time steps. De-icer time steps greater than 1 second cannot accurately capture temperature distributions even for cases where only the steady-state output (an anti-icing case) is simulated. This occurs because convergence cannot be maintained for larger de-icer time steps.

DTAUI values other than 0.1 seconds have been used in test cases which have a large initial transient in the temperature distribution. Smaller initial time steps were chosen in these cases to accurately model the temperature distribution. Larger initial time steps were chosen for steady-state anti-icing cases. Note that since the phase of each node in the ice layer must be iterated upon, a larger time step does not always provide a faster solution.

Times for temperature output selected by the user do not have to be in equal time increments. Equal increments were used in the example cases for convenience.

Previous versions of the LEWICE/Thermal software used time step instead of time.

The last two output types (IOTYPE = 4 or 5) are intended for debugging purposes and should not normally be needed by the user.

The shed trajectory computation assumes that shed ice particles are spheres.

The shed force data files can be extremely large and should be deleted if they are not needed.

6.14. Pneumatic Boot Option

Since the user inputs residual height and extent of coverage, the boot model does not predict the efficiency of a mechanical de-icing system. It simply allows the user to generate residual and intercycle ice shapes.

The term 'residual ice' has been used to refer to all ice remaining after boot activation, including ice accretion past the boot limits. 'Intercycle ice' refers to the ice accretion just prior to boot activation.

The pneumatic boot function does not predict boot efficiency since the residual ice height is specified. This option simply allows the user to generate residual ice shapes.

Ridge ice formations behind boots tend to be larger when HRES = 0.

The boot cycle time is fixed to the LEWICE time step size.

6.15. Additional Input Files

The user should input values for the entire wrap distance region. The program needs values for the entire body geometry and extrapolation from the values provided can yield poor results. Specifically, the values outside the region input are not set equal to zero.

The use of the KWARN flag is not recommended except when necessary for batch processes such as those ran by ICEG2D.

Chapter 7: 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 Windows95/DOS for PCs and IRIX 6.2 for SGI workstations. The latter example should be representative of error messages provided by other UNIX operating systems. The error message generated by the system is 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.

7.1. 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) ***
```

Windows 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 “\ewice\inputs\naca0012\case1.inp” can also be used to read the file “case1.inp” in the directory “C:\ewice\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.

7.2. 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 software 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.

7.3. 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 or if a de- icer input file is being used. 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 software. 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 program 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 program. 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.

7.4. Enter De-icer File Name for Body 1

This interactive input directs the user to input the name of the file containing the internal geometry of the body, physical properties, heater wattages, and various flags. If only one body is being run, the program will not ask for additional de-icer input files. If more than one body is being simulated, the program will prompt the user for additional file names for each body. The same file can be used for different bodies if the user desires. 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 10 for the first body, unit 11 for the second body, unit 12 for the third body, unit 13 for the fourth body, and 14 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 de-icer input. The de-icer input file will be checked for any errors or warning messages. If any problems occur with this input file, the warning and error messages described in the previous section will be issued.

Note: The program will only ask for the de-icer input file name for cases where IDEICE is not equal to zero.

7.5. Confirmation of Other Warnings

If LEWICE has a issues a warning when reading other input files, it will ask for a confirmation of those warnings. The syntax of these warning and error messages is similar to Section 7.3 and will not be repeated. LEWICE has the option to suppress the interactive confirmation of these warnings by using the KWARN flag described in Section 8.6.8. This allows for smoother operation of LEWICE when using batch scripts. However, it is not recommended for general use as these warnings may be indicative of legitimate problems with the input. Errors in the input files will still cause the program to stop and warnings will still be printed to the screen and to **junk.dat**.

Chapter 8: 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 software. 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**”.

8.1. 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.

8.1.1. 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) ***
```

Windows 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.

8.1.2. Namelists Listed Out of Order or Missing Namelist

The namelists used for LEWICE 3.2 are LEW20, DIST, ICE1, LPRNT, RDATA and BOOT 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) ***
```

Windows 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.

8.2. 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) ***
```

Windows DOS Shell Message

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

8.3. LEW20 Namelist

The LEW20 namelist contains a collection of inputs from a number of separate namelists from version 1.6¹⁴. 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

8.3.1. ITIMFL

Default Value: ITIMFL = 1

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 the following procedure. First, a preliminary number is calculated based upon the accumulation parameter as given below.

$$N = \frac{(LWC)(V)(Time)}{(chord)(\rho_{ice})(0.01)} \quad (2)$$

where

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

V = velocity (m/s)

$Time$ = accretion time (s)

$chord$ = airfoil chord (m)

ρ_{ice} = ice density = $9.17 \times 10^5 \text{ g}/\text{m}^3$

A second time step number is calculated by

$$N2 = \frac{Time}{60} \quad (3)$$

IFLO is then calculated by the following expression

$$IFLO = \text{Max}[\text{Min}(N, 30), \text{Min}(N2, 15)] \quad (4)$$

The values of IFLO calculated in this fashion will correspond to the values used in the LEWICE validation report. 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 report³. Due to this variability, LEWICE 3.2 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 program 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 program.

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**).

8.3.2. TSTART

Default Value: TSTART = 0

TSTART is the initial time of the icing simulation in seconds. At time = 0, LEWICE performs some estimates of transient behavior. The effect of these estimates is usually small. With variable TSTART, the user can run LEWICE using an iced geometry as the initial geometry and avoid applying those transient assumptions. The TSTART option can be useful for analysis of flight cases with variable conditions. TSTART can also be used in conjunction with the IGRID = 1 option to run the ICEG2D automatic grid generator. See Section 15.23 and the ICEG2D manual¹⁰ for details of the ICEG2D software. This variable has seen limited testing. Therefore, the following warning message will appear if TSTART is greater than zero:

TSTART is greater than zero. This option has only been used for ICEG2D cases.

TSTART must be greater than or equal to zero. If a negative value is input, the following error message will appear:

Severe input error: Time cannot be negative! TSTART = (value)

The program will stop because of the above error.

The maximum icing time in the validation database is 45 minutes (2700 seconds). If TSTART is greater than 2700 seconds, the following warning message will appear:

The maximum icing time used for testing this program was 45 min. Your time of TSTART = (value) exceeds this value.

Finally, TSTART must be less than the value of TSTOP input below. If TSTART is not less than TSTOP, the following warning message will appear:

TSTART must be less than TSTOP. Setting TSTART = 0.

The default value of TSTART used by LEWICE is 0 seconds if the user does not specify a value.

Note: The de-icer routines in LEWICE assume the body geometry is a clean airfoil with surface properties as described in Section 10.1. Therefore if the input geometry is an iced airfoil, this option may not run properly.

8.3.3. TSTOP

Default Value: TSTOP = 60

TSTOP is the ending 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 \leq 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 software was 45 min. Your time of tstop = (value) exceeds this value. The accuracy of the program in this situation is unknown.

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

8.3.4. IBOD

Default Value: IBOD = 1

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 d IBOD d 5. If the input value for IBOD is d 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 3.2 can run multiple body simulations including multi-element airfoils. A report of its capabilities in this region shows very encouraging results¹⁵. However, much of the development effort for version 3.2 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 3.2 validated for multi-body flows. Therefore, if the input value for IBOD is in the region 2 d IBOD d 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.

8.3.5. IFLO

Default Value: IFLO = 1

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 d 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 S 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 program.

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 program 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.

Note: Time steps smaller than 30 seconds are discouraged as they can cause undesirable numerical effects. The user should not choose a time step this small unless the chord length is very small as well.

Note: The procedure for the validation report set the number of time steps equal to the number of minutes of icing or 15, whichever was **lower**. This procedure is highly recommended to maintain consistency with the validation report results.

8.3.6. DSMN

Default Value: $DSMN = 4 \times 10^{-4}$

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 3.2 centered on defining practical ranges for the values of DSMN. The results of these tests are provided in the validation report³. For all of the airfoils studied, the range of DSMN values was 2×10^{-4} to 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 to 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 of 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 program.

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 program 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 3.2 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 program. 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.

8.3.7. NPL

Default Value: NPL = 24

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 program. 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 software uses for the collection efficiency calculation may be different than the value input. The program 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.

8.3.8. RHOP

Default Value: RHOP = 1000

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 software 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 software can be used to predict deposition of sand (sand collection efficiency). In this mode, the ice accretion results should be ignored and the program 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/m³).

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 software.

8.3.9. SLD

Default Value: SLD = 0

SLD is a flag which activates physical models and correlations for supercooled large droplets. If SLD = 0, the program will function with the same droplet model used in previous LEWICE versions. If SLD = 1, drops may break up prior to impact and splash upon impact and alter the water collection. The user is referred to Appendix sections 5 through 8 for more details on the models. Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

Note: The SLD flag may be used for any drop size as the models employed will have negligible effect on Appendix C drop sizes.

Note: The baseline empirical model is based upon experiments performed at lower velocities (20 m/s and below). It has been modified based upon collection efficiency and ice shape data taken in the Icing Research Tunnel (IRT). While the NASA database in SLD is growing, it is by no means comprehensive and the behavior of the correlations outside the validation test matrix is unknown.

This model is based upon experiments performed at lower velocities (20 m/s and below). As such, when SLD = 1, the following warning message will appear:

This option will activate SLD correlations that reflect comparisons of calculated ice shapes and water impingement curves with the corresponding experimental database. Please check your conditions against this data.

The default value for SLD is 0 if not specified.

8.3.10. IGRID

Default Value: IGRID = 0

IGRID is a flag which allows a grid solution to be used in place of the potential flow solver. 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. The user is referred to Section 10.5 for details on the format for these files. 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 IGRID = 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 contain errors. 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 seen limited testing. This procedure is not recommended for users unfamiliar with Navier-Stokes solvers. If you are using ICEG2D to automate grid generation, refer to the ICEG2D manual if you have problems. Note that IFLO still equals one when using ICEG2D.

Since the grid read in can only be applicable for a single time step, the number of time steps will be set to 1 if the user has not already done so. If the user is running ICEG2D or otherwise attempting to run a multi-time step Navier-Stokes solution and TSTART is greater than zero, the following warning message will appear:

ICEG2D is currently available only on the SGI platform. Multi-time step, grid-based option may not work properly otherwise.

Note: The ICEG2D program is currently being ported to LINUX and will eventually be available on other systems.

The default value for IGRID is 0 if not specified.

8.3.11. IDEICE

Default Value: IDEICE = 0

IDEICE is a flag that controls which de-icer model will be invoked. 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 de-icing or anti-icing models. If IDEICE = 2, 3, or 4 the software will access a module to analyze 2D transient icing with heater inputs. For IDEICE = 2, the software will perform the analysis using the standard heat transfer coefficients predicted assuming an ice roughened surface. This option is often preferable for de-icing simulations where ice will form. For IDEICE = 3, the software will use the laminar heat transfer coefficient, which assumes a clean surface. This option is recommended for anti-icing simulations or cases which generate a very small ice shape. For IDEICE = 4, the software will trip the boundary layer at the end of the heated section. This boundary layer trip will work for electrothermal as well as bleed air cases. If heat is applied to the entire surface, this option should produce the same output as the IDEICE = 3 option. IDEICE = 4 option may be useful for cases with significant runback ice. A value of IDEICE > 0 also requires an additional input file which contains information needed for the additional module. The software will prompt the user for the name of the de-icer input file. The software will ask for an input file for each body in a multi-element calculation. However, the user can input the same file if the information is the same for each body. Valid input values are 0 - 4. Other input values will produce the following warning message:

Valid inputs of IDEICE are 0 to 4. Your input value of (value) is out of range. Setting IDEICE = 0.

In order to distinguish the attributes of the various options, 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. IDEICE = 2 to IDEICE = 4 should be used for more accurate predictions.

Note: Setting IDEICE = 1 does not affect the ice accretion routine, except when the IEVAP = 2 option is used. Otherwise, the software 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, for IDEICE = 1, only the results obtained in the first time step are applicable to an anti-icing problem.

When IDEICE = 2, 3, or 4, the following warning message will be displayed:

This option accesses a module which will run MUCH more slowly than a standard (IDEICE = 0) LEWICE run.

When IDEICE = 3, the following additional warning message will be displayed:

This option will use laminar heat transfer coefficients. Some users believe this option will produce more accurate results for anti-icing cases. However, it is less conservative than the IDEICE = 2 option.

When IDEICE = 4, the following additional warning message will be displayed:

This option will trip the boundary layer at the end of the heated area. Some users believe this option will produce more accurate results for running wet cases. However, it is less conservative than the IDEICE=2 option.

Note: The IDEICE = 2, IDEICE = 3, and IDEICE = 4 options will perform a detailed thermal analysis within the airfoil, ice, and water. The IDEICE = 2 option is recommended for failed thermal cases with significant ice accretion. The laminar flag, IDEICE = 3, provides consistently better temperature predictions for cases with a clean leading edge. However, it is a less conservative option and has a tendency to overpredict temperatures for some cases. The boundary layer trip flag IDEICE = 4 provides a compromise value between these two options.

Note: The output file “**noice.dat**” contains output from the simplified 1D thermal analysis. This analysis is provided for all cases where IDEICE > 0. When IDEICE = 1, it is the only thermal analysis performed. It is also performed for IDEICE = 2, IDEICE = 3, and IDEICE = 4 cases as the 1D option does not take significant computation time. Also see Section 10.1.4 for additional capabilities when IDEICE = 2, 3 or 4.

8.3.12. ICP

Default Value: ICP = 0

ICP is a flag which allows surface pressure coefficients from another program to be used in place of the potential flow module. If $ICP = 0$, surface pressure coefficients (and hence air velocities) are determined directly from the potential flow solution. If $ICP = 1$, the panel solution will not be used. Instead, a file will be read in from file “**rflow.inp**” which is supplied by the user. This file contains surface pressure coefficients referenced to specified wrap distance values. The wrap distance is the distance along the surface of the body geometry as measured from some initial reference location. All of the input files for LEWICE 3.2 use the trailing edge on the lower surface as the reference location. Output files will normally use the stagnation point as the reference location. The format for the input file is covered in Section 10.6 and additional information is given in Section 8.7. LEWICE 3.2 will then bypass the potential flow module and use these values. Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

In addition, since pressure coefficients alone are insufficient to calculate collection efficiencies, this option can only be used in conjunction with the IBETA flag or the IGRID flag. If $ICP = 1$ and $IBETA = 0$ and $IGRID = 0$, the following warning message is issued:

The program cannot calculate trajectories using only surface pressure coefficients. Either read in collection efficiencies (IBETA = 1) or read in a grid-based flow solution (IGRID = 1). Setting ICP = 0

Some cases have been made using surface pressure coefficients as input to verify that the routines function as designed. In addition, since the pressure coefficients read in can only be applicable for the initial geometry, only one time step can be used with this option thus it has limited use for ice accretion results. Therefore if the user selects $ICP = 1$, the following warning message is issued:

You are bypassing the potential flow solution to use individual pressure coefficients. This option has seen limited testing. This option can only be used with a single time step. Setting IFLO = 1

The number of time steps will be set to 1 if the user has not already done so. The default value for ICP is 0 if not specified.

8.3.13. IBETA

Default Value: $IBETA = 0$

IBETA is a flag which allows collection efficiencies from another program to be used in place of the particle trajectory module. If $IBETA = 0$, collection efficiencies are determined directly from the particle trajectory module. If $IBETA = 1$, the trajectory module will not be used. Instead, a file will be read in from file “**rbeta.inp**” which is supplied by the user. This file contains collection efficiencies referenced to specified wrap distance values. The format for this file is covered in Section 10.7 and additional information is given in Section 8.7. LEWICE 3.2 will then bypass the particle trajectory module and use these values. Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

Some cases have been made using collection efficiencies as input to verify that the routines function as designed. 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 IBETA =1, the following warning message is issued:

You are bypassing the trajectory module to use individual collection efficiencies. This option has seen limited testing. This option can only be used with a single time step. Setting IFLO = 1

Since the collection efficiencies read in can only be applicable for the initial geometry, the number of time steps will be set to 1 if the user has not already done so. The default value for IBETA is 0 if not specified.

8.3.14. IHTC

Default Value: IHTC = 0

IHTC is a flag which allows convective heat transfer coefficients from another program to be used in place of the integral boundary layer in LEWICE 3.2. If IHTC = 0, convective heat transfer coefficients are determined directly from the integral boundary layer. If IHTC = 1, the integral boundary layer will not be used. Instead, a file will be read in from file “**rhtc.inp**” which is supplied by the user. This file contains convective heat transfer coefficients referenced to specified wrap distance values. The format for this file is covered in Section 10.8 and additional information is given in Section 8.7. LEWICE 3.2 will then bypass the integral boundary layer routines and use these values. Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

The IHTC flag and the IQEX flag from Section 8.3.15 are mutually exclusive. If both are set equal to one, the following warning message will appear:

This program can use either a convective heat transfer coefficient on the outer surface or a heat flux, but not both. Setting both IHTC = 0 and IQEX = 0

Some cases have been run using convective heat transfer coefficients as input to verify that the routines function as designed. 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 IHTC = 1, the following warning message is issued:

You are bypassing the integral boundary layer to use individual convective heat transfer coefficients. This option has seen limited testing. This option can only be used with a single time step. Setting IFLO = 1

Since the convective heat transfer coefficients read in can only be applicable for the initial geometry, the number of time steps will be set to 1 if the user has not already done so. The default value for IHTC is 0 if not specified.

8.3.15. IQEX

Default Value: IQEX = 0

IQEX is a flag which allows external heat fluxes from another program to be used in place of the integral boundary layer in LEWICE 3.2. If IQEX = 0, convective heat transfer coefficients are determined directly from the integral boundary layer. If IQEX = 1, the integral boundary layer will not be used. Instead, a file will be read in from file “**qextin.inp**” which is supplied by the user. This file contains external heat fluxes referenced to specified wrap distance values. The format for this file is covered in Section 10.9 and additional information is given in Section 8.7. LEWICE 3.2 will then bypass the integral boundary layer routines and calculate convective heat transfer coefficients from the equation

$$h = \frac{q}{(T_s - T_\infty)} \quad (5)$$

where q is the heat flux read in, T_s is the surface temperature at a given wrap distance location, and T_∞ is the ambient temperature. Some users may find external heat flux input more convenient than the convective heat transfer coefficient.

Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

The IHTC flag from Section 8.3.14 and the IQEX flag from Section 8.3.15 are mutually exclusive. If both are set equal to one, the following warning message will appear:

This program can use either a convective heat transfer coefficient on the outer surface or a heat flux, but not both. Setting both IHTC = 0 and IQEX = 0

Some cases have been run using external heat fluxes as input to verify that the routines function as designed. 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 IQEX = 1, the following warning message is issued:

You are bypassing the integral boundary layer to use individual heat flux values. This option has seen limited testing. This option can only be used with a single time step. Setting IFLO = 1

Since the external heat fluxes read in can only be applicable for the initial geometry, the number of time steps will be set to 1 if the user has not already done so. The default value for IQEX is 0 if not specified.

8.3.16. IBOOT

Default value: IBOOT = 0

IBOOT is a flag which activates a simplistic model of a mechanical boot process. It can also be used to model other de-icing processes as the methodology is not specific to any particular technology. This process is also independent of the thermal de-icing processes described elsewhere. If IBOOT=0, no mechanical de-icing occurs. If IBOOT=1, ice accumulation within the boot limits specified in Section 8.8 is limited to the residual height. The residual height is also input by the user in Section 8.8. The ice shapes output to **ice1.dat** (or other ice shape output files) will contain the residual ice shape. An additional output file, **intercycle.dat**, will contain

the intercycle ice shape. The de-icer cycle time will be the same as the LEWICE time step chosen earlier.

Valid inputs for this variable are 0 and 1. Other input values will produce the following warning message:

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

If IBOOT=1 is selected, the following warning message will be generated:

You have activated a crude boot model. File ICE1.DAT will contain the residual shape and INTERCYCLE.DAT will contain the intercycle shape. This option has not been validated.

Note: Since the user inputs residual height and extent of coverage, the boot model does not predict the efficiency of a mechanical de-icing system. It simply allows the user to generate residual and intercycle ice shapes.

Note: The term ‘residual ice’ has been used to refer to all ice remaining after boot activation, including ice accretion past the boot limits. ‘Intercycle ice’ refers to the ice accretion just prior to boot activation.

&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
TSTART = 0.
TSTOP = 300.
IBOD = 1
IFLO = 5
DSMN = 4.D-4
NPL = 24
RHOP = 1000.
SLD = 0
IGRID = 0
IDEICE = 0
ICP = 0
IBETA = 0
IHTC = 0
IQEX = 0
IBOOT = 0
&END
```

8.4. DIST Namelist

The DIST namelist defines the particle size and distribution. For each variable, there are 10 possible values, as the software 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.

8.4.1. FLWC

Default Value: FLWC = 1.

FLWC is the volume fraction of the total liquid water content contained in each drop size. The associated drop sizes are discussed in the next section. The sum of the FLWC 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
```

8.4.2. DPD

Default Value: DPD = 20

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 software 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 software 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.

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 experimental data has been collected in this regime, the software has been calibrated (but not validated) using the existing data. Your conditions may not be covered in the current experimental database. Output from this model should be used with caution.

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 1996¹⁶. A more recent publication illustrates use of the SLD = 1 empirical model¹³.

&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 .

8.5. 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.

8.5.1. CHORD

Default Value: CHORD = 0.9144

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 configuration. 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”.

8.5.2 AOA

Default Value: AOA = 0.

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 software. The accuracy of the software 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 software 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 software 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.

8.5.3. VINF

Default Value: VINF = 90

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 d 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 software. If the ambient Mach number is S 1, the following error message is generated:

Mach number is => 1! Cannot calculate supersonic flow with this software! 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 software. The accuracy of the software 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 $2.26 \cdot 10^6$ to $1.3 \cdot 10^7$. In terms of Mach number, the range of data was 0.17 to 0.45.

8.5.4. LWC

Default Value: LWC = 0.54

LWC is the liquid water content of the air in g/m^3 . This value must be S 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 software for LWC values this high. LWC = (value). The accuracy of the software 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.

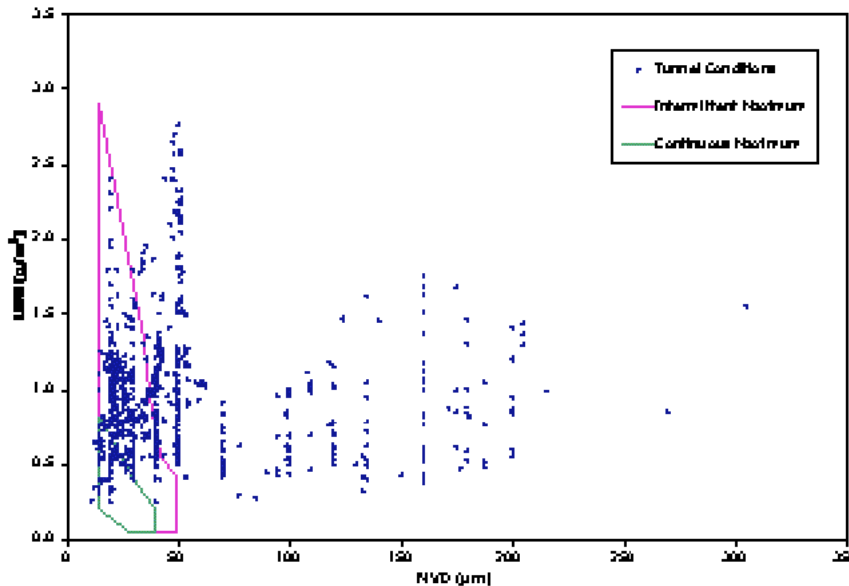


Figure 1: Test Conditions in Database Used for Software Validation

8.5.5. TINF

Default Value: $TINF = 268.15$

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 \leq 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 software 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!

$$T_s = T_o - \frac{V_\infty^2}{2c_p} \quad (6)$$

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.

8.5.6. PINF

Default Value: PINF = 100000.

PINF is the ambient static pressure in Pascals (N/m²). The input value for PINF must be greater than zero. An input value of PINF d 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 10⁵ N/m².

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 NASA Glenn Research Center Icing Research Tunnel (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:

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

where

P = pressure in N/m² and

H = height (altitude) in meters

Note: lb_f/in² * 6894.7 = N/m²; atmospheres * 101330 = N/m²; in. Hg * 3386.4 = N/m²

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

8.5.7. RH

Default Value: $\text{RH} = 100$.

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 most 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 $0\% \leq \text{RH} \leq 100\%$. 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%.

8.5.8. GRAV

Default Value: $\text{GRAV} = 9.8$

GRAV is the acceleration due to gravity. The default value is 9.8 m/s^2 . Users can choose to turn off gravity in order to compare LEWICE with cases generated in the Icing Research Tunnel (IRT) since models are typically mounted vertically in that tunnel. This may make a small difference for some supercooled large droplet cases but is not considered necessary. If GRAV is not 9.8 m/s^2 , the program will issue the following warning message:

Gravity is not 9.8 m/s^2. Is this what you want?

If no value for gravity is input by the user, its default value is 9.8.

8.5.9. SREF

Default Value: $\text{SREF} = 0$

If $\text{SREF} = 0$, wrap distance values in all output files will be referenced to the stagnation point of that time step. If $\text{SREF} = 1$, wrap distance will be referenced to the leading edge of the clean geometry for that body. If $\text{SREF} = 2$, wrap distance will be referenced to the lower surface trailing edge. Valid inputs for SREF are 0, 1 and 2. Other input values will produce the following warning message:

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

Since ice normally forms at the leading edge, SREF loses physical significance if applied for multiple time steps. Therefore, the following warning message will be generated when $\text{SREF} = 1$:

Wrap distance values are referenced to the leading edge. Your reference location loses physical meaning past the first time step.

If no value of SREF is input by the user, its value will default to 0.

Note: In LEWICE 2.2, variable SREF was in the LPRNT namelist.

&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
VINFL  = 89.5
LWC    = 0.34
TINFL  = 269.
PINFL  = 100000.0
RH     = 100.0
GRAV   = 9.8
SREF   = 0
&END
```

8.6. LPRNT Namelist

The LPRNT namelist controls output file print options for LEWICE 3.2. 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.

8.6.1. FPRT

Default Value: FPRT = 1

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.

8.6.2. HPRT

Default Value: HPRT = 1

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.

8.6.3. BPRT

Default Value: BPRT = 1

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.

8.6.4. EPRT

Default Value: EPRT = 0

If EPRT = 0, the energy balance output (**piccolo1.dat**, **piccolo2.dat**, **temp.dat**, **qener.dat**, **xkinit2.dat**) will not be written. **Piccolo1.dat** and **piccolo2.dat** also require the bleed air flag IOTHERM in Section 10.1.4.8. 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.

8.6.5. MPRT

Default Value: MPRT = 0

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

8.6.6. TPRT

Default Value: TPRT = 0

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!

8.6.7. IDBF

Default Value: IDBF = 0

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.

8.6.8. KWARN

Default Value: KWARN = 0

If KWARN = 0, all warning messages described in this manual will be output. If KWARN = 1, the program will still output the warnings, but will not ask for confirmation of warnings described in Sections 10.2 - 10.10. This allows for smoother operation of the scripts used when running ICEG2D. While warning messages and interactive confirmation can become annoying, deactivating the warning confirmation may mask legitimate problems in the input file. The use of KWARN = 0 is strongly recommended except when using ICEG2D. Valid inputs for KWARN are 0 and 1. Other inputs will generate the following warning message:

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

If KWARN = 1, the following warning message will be generated:

This option will suppress warnings (but NOT errors) when reading external input files. It is only recommended for sophisticated users running batch scripts.

If no value of KWARN is input by the user, its value will default to zero.

&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
KWARN  = 0
&END
```

8.7. RDATA Namelist

The RDATA namelist controls additional input file options for LEWICE 3.2. Users can bypass various calculations within LEWICE 3.2 by reading in data from external files. This approach is useful for reading in results from other programs such as a 2D Navier-Stokes flow program or 2D streamlines extracted from a 3D CFD package. The input flags were presented in the LEW20 Namelist in Section 8.3.

&RDATA

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.

8.7.1. ISCOLC, JSCOLC, KSCOLC, SSLOPC, SZEROC

Default values: ISCOLC=1, JSCOLC=2, KSCLOC=2, SSLOPC=1., SZEROC=0.

The first five lines of this data input section contains variables used when reading pressure coefficient data (i.e. when ICP = 1). This input option can be useful to transfer data from a Navier-Stokes program such as WIND for a 2D anti-icing analysis with LEWICE 3.2. These variables can be read regardless of the value of ICP however. The data line contains five variables which define the format of the pressure coefficient data. The first three variables (ISCOLC, JSCOLC and KSCOLC) define the column numbers in the pressure coefficient input data file which contain the requested values. ISCOLC defines which column contains the wrap distance. JSCOLC defines the column which contains the pressure coefficients, and KSCOLC defines the total number of columns in the file. A file which contained only the necessary data would therefore have ISCOLC = 1, JSCOLC = 2, and KSCOLC = 2. (ISCOLC = 2, JSCOLC = 1 and KSCOLC = 2 would also be valid.) This type of data input format was used since the data probably came from a separate program and this format style reduces the amount of reformatting needed for LEWICE 3.2. ISCOLC must be at least 1 and no greater than KSCOLC. If a value is read which is outside this range, the following warning message will be generated:

Columns for 1st variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting ISCOLC = 1

Similarly, JSCOLC must be at least 1 and no greater than KSCOLC. It also cannot be the same as ISCOLC. If a value is read which is outside this range, the following warning message will be generated:

Columns for 2nd variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting JSCOLC = 1

If the values for ISCOLC and JSCOLC are the same, the following warning message will be generated:

Columns for 1st and 2nd variable cannot be the same.

Setting ISCOLC = 1 and JSCOLC = KSCOLC

The total number of columns, KSCOLC must be at least 2, but can be higher. If a value is read which is outside this range, the following warning message will be generated:

Number of columns must be at least two (2)

Your value of (value) is out of range. Setting KSCOLC = 2

The fourth variable, SSLOPC, defines the conversion factor which will be applied to the wrap distance values input. LEWICE 3.2 uses wrap distances in dimensionless values (value divided by chord). The wrap distance values input can be in any set of units if the user supplies the correct conversion to SSLOPC. The slope can be positive or negative, but not zero. If a value of zero is read for this data input, the following warning message will be generated:

Slope cannot be zero.

Setting slope = 1.

The fifth value in this data input section, SZEROC, defines the offset of the input wrap distances from those needed for LEWICE 3.2. If SZEROC = 0., the software will assume that a wrap distance value of 0 in the input data corresponds to the lower surface of the trailing edge. If this is not the case, the user should supply a value (in dimensionless distance) which supplies the correct offset. The value for SZEROC should not be greater than the wrap distance from the leading edge to the trailing edge. If the value read is greater than 1.2 (1.2 chord lengths), the following warning message will be generated:

Offset may be past the trailing edge. Setting SZEROC = 1.

The following equation is used within LEWICE to convert wrap distance values using SSLOPC and SZEROC:

$$s_{LEWICE} = s_{input}sslopc + szeroc \quad (7)$$

where s_{LEWICE} is the wrap distance used by LEWICE and s_{input} is the wrap distance in the input file.

The format of the input data file “**rflow.inp**” and corresponding error messages will be described in Section 10.6.

8.7.2. ISCOLB, JSCOLB, KSCOLB, SSLOPB, SZEROB

Default values: ISCOLB=1, JSCOLB=2, KSCLOB=2, SSLOPB=1., SZEROB=0.

The next five lines of this data input section contains variables used when reading collection efficiency data (i.e. when IBETA = 1). This input option can be useful to transfer data from a 3D trajectory program such as LEWI3DGR for a 2D anti-icing analysis with LEWICE 3.2. These variables can be read regardless of the value of IBETA however. The data line contains five variables which define the format of the collection efficiency data. The first three variables (ISCOLB, JSCOLB and KSCOLB) define the column numbers in the collection efficiency input data file which contain the requested values. ISCOLB defines which column contains the wrap distance. JSCOLB defines the column which contains the collection efficiencies, and KSCOLB defines the total number of columns in the file. A file which contained only the necessary data would therefore have ISCOLB = 1, JSCOLB = 2, and KSCOLB = 2. (ISCOLB = 2, JSCOLB = 1 and KSCOLB = 2 would also be valid.) This type of data input format was used since the data probably came from a separate program and this format style reduces the amount of reformatting needed for LEWICE 3.2. ISCOLB must be at least 1 and no greater than KSCOLB. If a value is read which is outside this range, the following warning message will be generated:

Columns for 1st variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting ISCOLB = 1

Similarly, JSCOLB must be at least 1 and no greater than KSCOLB. It also cannot be the same as ISCOLB. If a value is read which is outside this range, the following warning message will be generated:

Columns for 2nd variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting JSCOLB = 1

If the values for ISCOLB and JSCOLB are the same, the following warning message will be generated:

Columns for 1st and 2nd variable cannot be the same.

Setting ISCOLB = 1 and JSCOLB = KSCOLB

The total number of columns, KSCOLB must be at least 2, but can be higher. If a value is read which is outside this range, the following warning message will be generated:

Number of columns must be at least two (2)

Your value of (value) is out of range. Setting KSCOLB = 2

The fourth variable, SSLOPB, defines the conversion factor which will be applied to the wrap distance values input. LEWICE 3.2 uses wrap distances in dimensionless values (value divided by chord). The wrap distance values input can be in any set of units if the user supplies the correct conversion to SSLOPB. The slope can be positive or negative, but not zero. If a value of zero is read for this data input, the following warning message will be generated:

Slope cannot be zero.

Setting slope = 1.

The last value in this data input subsection, SZEROB, defines the offset of the input wrap distances from those needed for LEWICE 3.2. If SZEROB = 0., the software will assume that a wrap distance value of 0 in the input data corresponds to the lower surface trailing edge. If this is not the case, the user should supply a value (in dimensionless distance) which supplies the correct offset. The value for SZEROB should not be greater than the wrap distance from the leading edge to the trailing edge. If the value read is greater than 1.2 (1.2 chord lengths), the following warning message will be generated:

Offset may be past the trailing edge. Setting SZEROB = 1.

The following equation is used within LEWICE to convert wrap distance values using SSLOPB and SZEROB:

$$s_{LEWICE} = s_{input}sslop + szero \quad (8)$$

where s_{LEWICE} is the wrap distance used by LEWICE and s_{input} is the wrap distance in the input file. The format of the input data file “**rbeta.inp**” and corresponding error messages will be described in Section 10.7.

8.7.3. ISCOLH, JSCOLH, KSCOLH, SSLOPH, SZEROH

Default values: ISCOLH=1, JSCOLH=2, KSCLOH=2, SSLOPH=1., SZEROH=0.

The next five lines of this data input section contains variables used when reading external heat transfer coefficient data (i.e. when IHTC = 1). This input option can be useful to transfer data from a 2D Navier-Stokes program or a 3D trajectory program such as LEWI3DGR for a 2D icing or anti-icing analysis with LEWICE 3.2. These variables can be read regardless of the value of IHTC however. The data line contains five variables which define the format of the external heat transfer coefficient data. The first three variables (ISCOLH, JSCOLH and KSCOLH) define the column numbers in the external heat transfer coefficient input data file which contain the requested values. ISCOLH defines which column contains the wrap distance. JSCOLH defines the column which contains the external heat transfer coefficients, and KSCOLH defines the total number of columns in the file. A file which contained only the necessary data would therefore have ISCOLH = 1, JSCOLH = 2, and KSCOLH = 2. (ISCOLH = 2, JSCOLH = 1 and KSCOLH = 2 would also be valid.) This type of data input format was used since the data probably came from a separate program and this format style reduces the amount of reformatting needed for LEWICE 3.2. ISCOLH must be at least 1 and no greater than KSCOLH. If a value is read which is outside this range, the following warning message will be generated:

Columns for 1st variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting ISCOLH = 1

Similarly, JSCOLH must be at least 1 and no greater than KSCOLH. It also cannot be the same as ISCOLH. If a value is read which is outside this range, the following warning message will be generated:

***Columns for 2nd variable must be at least one (1)
and no greater than the number of columns.***

Your value of (value) is out of range. Setting JSCOLH = 1

If the values for ISCOLH and JSCOLH are the same, the following warning message will be generated:

Columns for 1st and 2nd variable cannot be the same.

Setting ISCOLH = 1 and JSCOLH = KSCOLH

The total number of columns, KSCOLH must be at least 2, but can be higher. If a value is read which is outside this range, the following warning message will be generated:

Number of columns must be at least two (2)

Your value of (value) is out of range. Setting KSCOLH = 2

The fourth variable, SSLOPH, defines the conversion factor which will be applied to the wrap distance values input. LEWICE 3.2 uses wrap distances in dimensionless values (value divided by chord). The wrap distance values input can be in any set of units if the user supplies the correct conversion to SSLOPH. The slope can be positive or negative, but not zero. If a value of zero is read for this data input, the following warning message will be generated:

Slope cannot be zero.

Setting slope = 1.

The last value in this data input subsection, SZEROH, defines the offset of the input wrap distances from those needed for LEWICE 3.2. If SZEROH = 0., the software will assume that a wrap distance value of 0 in the input data corresponds to the lower surface trailing edge. If this is not the case, the user should supply a value (in dimensionless distance) which supplies the correct offset. The value for SZEROH should not be greater than the wrap distance from the leading edge to the trailing edge. If the value read is greater than 1.2 (1.2 chord lengths), the following warning message will be generated:

Offset may be past the trailing edge. Setting SZEROH = 1.

The following equation is used within LEWICE to convert wrap distance values using SSLOPH and SZEROH:

$$s_{LEWICE} = s_{input}ssloph + szeroH \quad (9)$$

where s_{LEWICE} is the wrap distance used by LEWICE and s_{input} is the wrap distance in the input file. The format of the input data file “**rhtc.inp**” and corresponding error messages will be described in Section 10.8.

8.7.4. ISCOLX, JSCOLX, KSCOLX, SSLOPX, SZEROX

Default values: ISCOLX=1, JSCOLX=2, KSCLOX=2, SSLOPX=1., SZEROX=0.

The final five lines of this data input section contains variables used when reading external heat flux data (i.e. when IQEX = 1). This input option can be useful to transfer data from a

Naviér-Stokes program such as WIND for a 2D anti-icing analysis with LEWICE 3.2. This option was provided since the convective heat flux may be easier for the user to obtain from the Naviér- Stokes flow solver. These variables can be read regardless of the value of IQEX however. The data line contains five variables which define the format of the external heat flux data. The first three variables (ISCOLX, JSCOLX and KSCOLX) define the column numbers in the external heat flux input data file which contain the requested values. ISCOLX defines which column contains the wrap distance. JSCOLX defines the column which contains the external heat fluxes, and KSCOLX defines the total number of columns in the file. A file which contained only the necessary data would therefore have ISCOLX = 1, JSCOLX = 2, and KSCOLX = 2. (ISCOLX = 2, JSCOLX = 1 and KSCOLX = 2 would also be valid.) This type of data input format was used since the data probably came from a separate program and this format style reduces the amount of reformatting needed for LEWICE 3.2. ISCOLX must be at least 1 and no greater than KSCOLX. If a value is read which is outside this range, the following warning message will be generated:

Columns for 1st variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting ISCOLX = 1

Similarly, JSCOLX must be at least 1 and no greater than KSCOLX. It also cannot be the same as ISCOLX. If a value is read which is outside this range, the following warning message will be generated:

Columns for 2nd variable must be at least one (1)

and no greater than the number of columns.

Your value of (value) is out of range. Setting JSCOLX = 1

If the values for ISCOLX and JSCOLX are the same, the following warning message will be generated:

Columns for 1st and 2nd variable cannot be the same.

Setting ISCOLX = 1 and JSCOLX = KSCOLX

The total number of columns, KSCOLX must be at least 2, but can be higher. If a value is read which is outside this range, the following warning message will be generated:

Number of columns must be at least two (2)

Your value of (value) is out of range. Setting KSCOLX = 2

The fourth variable, SSLOPX, defines the conversion factor which will be applied to the wrap distance values input. LEWICE 3.2 uses wrap distances in dimensionless values (value divided by chord). The wrap distance values input can be in any set of units if the user supplies the correct conversion to SSLOPX. The slope can be positive or negative, but not zero. If a value of zero is read for this data input, the following warning message will be generated:

Slope cannot be zero.

Setting slope = 1.

The last value in this data input section, SZEROX, defines the offset of the input wrap distances from those needed for LEWICE 3.2. If SZEROX = 0., the software will assume that a wrap distance value of 0 in the input data corresponds to the lower surface of the trailing edge. If this is not the case, the user should supply a value (in dimensionless distance) which supplies the correct offset. The value for SZEROX should not be greater than the wrap distance from the leading edge to the trailing edge. If the value read is greater than 1.2 (1.2 chord lengths), the following warning message will be generated:

Offset may be past the trailing edge. Setting SZEROX = 1.

The following equation is used within LEWICE to convert wrap distance values using SSLOPX and SZEROX:

$$s_{LEWICE} = s_{input}sslopx + szerox \quad (10)$$

where s_{LEWICE} is the wrap distance used by LEWICE and s_{input} is the wrap distance in the input file. The format of the input data file “**qextin.inp**” and corresponding error messages will be described in Section 10.9. Table 7 shows an example of the input data for this section.

Table 7: Example input for RDATA namelist

```
&RDATA
ISCOLC = 4
JSCOLC = 6
KSCOLC = 6
SSLOPC = 3.937d0
SZEROC = 0.d0
ISCOLB = 1
JSCOLB = 2
KSCOLB = 2
SSLOPB = 3.937d0
SZEROB = 0.d0
ISCOLH = 1
JSCOLH = 2
KSCOLH = 2
SSLOPH = 3.937d0
SZEROH = 0.d0
ISCOLX = 1
JSCOLX = 2
KSCOLX = 2
SSLOPX = 3.937d0
SZEROX = 0.d0
$END
```


8.8. BOOT Namelist

This namelist contains inputs which will be used if IBOOT=1. IBOOT was described in the LEW20 Namelist in Section 8.3.16. The user can input three variables in this namelist; the upper and lower boot limits and the maximum residual ice height.

&BOOT

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.

8.8.1. XBOOTUP

Default value: XBOOTUP=0.1

XBOOTUP specifies the location of the upper boot limit for each body input. The variable is the x-distance, normalized by chord, from the leading edge (minimum x-location of that body) to the upper boot limit. For example, if the upper boot limit is at 8% chord, the input should be XBOOTUP = 0.08. As stated earlier, this distance is measured from the leading edge of each body, not simply from the overall leading edge. Valid inputs for XBOOTUP are between 0. and 1. Values outside this range will generate the following warning message:

A valid upper boot limit is between $x/c=0.$ and $x/c=1.$ Setting XBOOTUP=0.

8.8.2. XBOOTLOW

Default value: XBOOTLOW=0.1

XBOOTLOW specifies the location of the lower boot limit for each body input. The variable is the distance, normalized by chord, from the leading edge (minimum x-location of that body) to the lower boot limit. For example, if the lower boot limit is at 8% chord, the input should be XBOOTLOW = 0.08. As stated earlier, this distance is measured from the leading edge of each body, not simply from the overall leading edge. Valid inputs for XBOOTLOW are between 0. and 1. Values outside this range will generate the following warning message:

A valid lower boot limit is between $x/c=0.$ and $x/c=1.$ Setting XBOOTLOW=0.

8.8.3. HRES

Default value: HRES=0.005

HRES specifies the maximum residual ice height for each body input. The variable is the dimensionless height relative to the chord length input. Valid inputs for HRES are greater than or equal to zero. Values outside this range will generate the following warning message:

Residual height must be $> or = 0.$ Setting HRES = 0.

Note: The pneumatic boot function does not predict boot efficiency since the residual ice height is specified. This option simply allows the user to generate residual ice shapes.

Note: Ridge ice formations behind boots tend to be larger when HRES = 0.

Note: The boot cycle time is fixed to the LEWICE time step size.

8.9. 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 8: Example Test Input File

Example 1

&LEW20

```
ITIMFL = 1
TSTART = 0.
TSTOP = 300.
IBOD = 1
IFLO = 5
DSMN = 4.D-4
NPL = 24
RHOP = 1000.
SLD = 0
IGRID = 0
IDEICE = 0
ICP = 0
IBETA = 0
IHTC = 0
I3D = 0
IQEX = 0
IBOOT = 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
VINP = 89.5
LWC = 0.34
TINF = 269.
PINP = 100000.0
RH = 100.0
GRAV = 9.8
```

```
SREF = 0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
EPRT = 0
MPRT = 0
TPRT = 0
IDBF = 0
KWARN = 0
&END
&RDATA
ISCOLC = 4
JSCOLC = 6
KSCOLC = 6
SSLOPC = 3.937d0
SZEROC = 0.d0
ISCOLB = 1
JSCOLB = 2
KSCOLB = 2
SSLOPB = 3.937d0
SZEROB = 0.d0
ISCOLH = 1
JSCOLH = 2
KSCOLH = 2
SSLOPH = 3.937d0
SZEROH = 0.d0
ISCOLX = 1
JSCOLX = 2
KSCOLX = 2
SSLOPX = 3.937d0
SZEROX = 0.d0
&END
&BOOT
XBOOTUP = 0.
XBOOTLOW = 0.
HRES = 5.d-3
```

&END

8.10. Extent of Data in Validation Database

This section will summarize the range of experimental data used to validate LEWICE 3.2. 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 9: Range of Experimental Data for Validation

Variable	Range
Time:	2 min. to 45 min.
Chord:	13.9 inches to 78 inches
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

Note: While many additional data points have been collected, especially in the SLD environment, the upper limits of the validation database has not been extended from LEWICE 2.0.

Chapter 9: 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 data³.

In the interactive input to the program, LEWICE 3.2 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. LEWICE expects the coordinates to be normalized by chord. 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 10 below.

Table 10: 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**”.

9.1. Number of Points

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

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

The software 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.

9.2. 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 software. Other systems may read blank lines as an additional data point of $x = 0$, $y = 0$. The software 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.

9.3. 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.

9.4. 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.

9.5. Body Not Closed

The panel solution used in this software 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. The recommended procedure for closing the trailing edge is to extend the airfoil slope. This will result in an airfoil slightly longer than the original airfoil. Do **not** create a vertical line segment to connect the trailing edge.



Preferred method for closing the trailing edge



Don't do this!

9.6. 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 3.2 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.

9.7. 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 3.2 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.

Note: The correction in LEWICE may not work for body geometries with an open trailing edge.

9.8. 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.

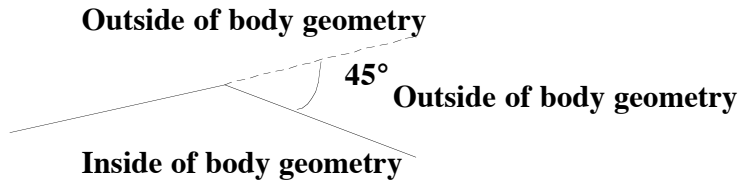


Figure 2: Example of a 45 degree panel angle

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.

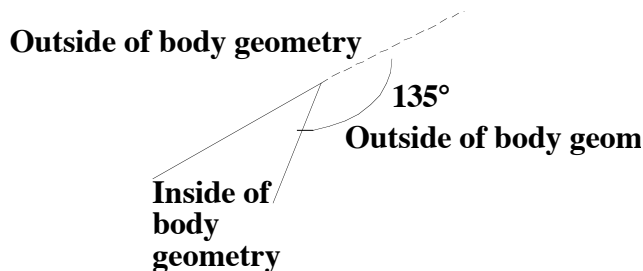


Figure 3: Example of a 135 degree panel angle

In this case, it is unlikely that the user intended to have this large of a segment angle. However, LEWICE will still 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.

9.9. 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.

9.10. 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.

9.11. 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**”.

9.12. Bodies Improperly Sized

LEWICE expects that the body will be input in dimensionless format (x/c , y/c). While the x -coordinates do not have to traverse from exactly $x/c = 0$ to exactly $x/c = 1$, they should reside in this general vicinity. Additionally, the y/c coordinates are generally input such that the leading edge of the airfoil is at $y/c = 0$ although this is not required. For multi-body airfoils, the chord has been defined using the airfoil in the retracted position. Therefore runs with slats employed will have x/c values less than zero and flaps employed will have x/c values greater than one. If the x/c coordinates are less than $x/c = -0.5$ or greater than $x/c = 1.5$, the following warning message will be generated:

Geometry input may not run properly in LEWICE. Check to make sure the coordinates have been normalized by chord.

Chapter 10: 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 several flags in the main input file which will require the input of an additional input file. An input value of 1, 2, 3, or 4 for variable IDEICE will read in a de-icer input file. An input value of 1 for variable IGRID will read in the files “XY.PLT” and “Q.PLT”. An input value of 1 for variables ICP, IBETA, IHTC, and IQEX will read in values from files “**rflow.inp**”, “**rbeta.inp**”, “**rhtc.inp**”, and “**qextin.inp**” respectively. Additionally, an input value of 1 for variables IBLEED, IQAIN, I3D, and ITSUR in the de-icer input file will read in the files “**hi.inp**”, “**qain.inp**”, “**stream.inp**”, and “**tsurf.inp**”. The formats for all of these additional input files will now be discussed in detail.

10.1. De-icer Input File

For IDEICE values from 1 to 4, LEWICE 3.2 will read in a de-icer input file. The name of this file is supplied by the user interactively. This is described in more detail in Section 7.4. The de-icer input consists of free-form (unformatted) ASCII text and can contain comment lines for user comments or to improve readability. The example cases used in this manual will contain comment lines which are used to further explain the features of this module. The description of the de-icer input file is divided into five sections: 1) description of the de-icer geometry and physical properties, 2) definition of heater power and cycle times, 3) definition of boundary conditions, 4) definition of various flags which control different features, and 5) description of time step and input/ output options. The division of the descriptions into five sections was done solely for clarity. The data is read from a single file for each body. Each of these sections will now be described in more detail.

10.1.1. De-icer Section 1

The first section of the de-icer input file contains the definition of the internal geometry of the body and the physical properties. The internal geometry is defined using rectangular blocks in the computational domain. This is illustrated in Figure 4. Each block contains grid spacing in the wrap direction and the normal direction.

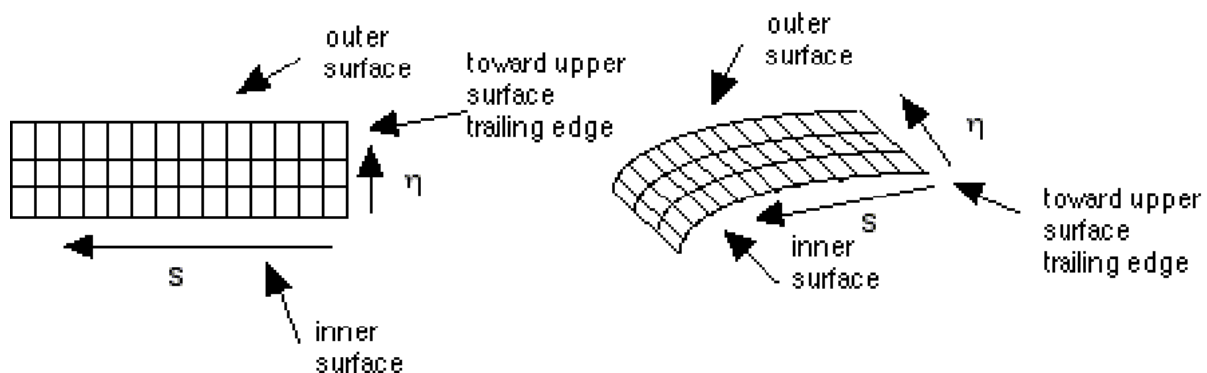


Figure 4: Grid block in the computational domain (left) and the physical domain (right)

Note: The de-icer geometry is specified by the user in the (rectangular) computational domain, not in the physical domain.

10.1.1.1. LDE and NXDE

The first row of data in the de-icer file contains the number of “layers” (blocks in the normal direction, variable named LDE) and the number of “sections” (blocks in the wrap direction, variable named NXDE). The total number of blocks simulated will be the product of these two numbers (LDE * NXDE). A sample data line is given below.

```
007 009
```

Note: In the subsequent description, the blocks in the normal direction will be referred to as “layers” while the blocks in the wrap direction will be referred to as “sections”.

The data line given above does not have to be the first line in the data file. Comment lines can be added for clarity. The software detects a comment line when it identifies a non-numeric character in the first 30 columns. An example is given in Table 11 below.

Table 11: First section of data file with comments

```
C      data   for   the   composite   body
C
C      lde= total number of layers in the y-direction
C      nxde= total number of heater sections in the x-direction
C
lde    nxde
007    009
```

Note: Data is read free-form. The numbers do not have to reside in any particular column and do not have to line up with the comments. However they must be in the prescribed order.

The number of layers and the number of sections must both be between 1 and 29. If the value read for these variables is outside this range, the following error message will appear:

Layer number must be greater than zero (0) and less than thirty (30). Your input value of (value) is outside this range. Blank lines are read as data with a value of zero. Check input file for blank lines.

The software will stop because of the above error.

As the message indicates, the most common cause of this error is a blank line in the input file which will be read as data with a value of zero.

10.1.1.2. NODE, ELDE, AK, ALP, AL, SLOPE

The next LDE number of data lines then contains six columns of numbers. The columns are: 1) the number of points in the normal direction for the layer; 2) the “thickness” of the layer (m); 3) the thermal conductivity of the layer (W/mK) at 0 °C in the normal direction; 4) the thermal diffusivity of the layer (m²/s); 5) the anisotropy ratio; and 6) the slope of thermal conductivity with temperature (W/mK²). The number of points defines the grid spacing in the normal direction for that layer. All sections will have the same grid spacing in the normal direction for a given layer.

Note: The first layer input is the innermost layer and the last layer will be the top surface.

The thermal conductivity of a layer in the normal direction is defined as a function of temperature, as shown in Equation 11.

$$k = k_{ref} + m(T - T_{ref}) \quad (11)$$

where

k = thermal conductivity (W/mK)

k_{ref} = thermal conductivity (W/mK) at the reference temperature

m = slope of temperature dependence (W/mK²)

T_{ref} = reference temperature = 0 °C

The user inputs the conductivity at the reference temperature as well as the slope in the equation above. For many composite materials, the conductivity can be anisotropic, which means that it is different in the wrap direction than it is in the normal direction. Because of this, the user also inputs a anisotropy ratio, which is the ratio of the thermal conductivity in the wrap direction to the thermal conductivity in the normal direction. Most materials will have an anisotropy ratio of 1.

Note: The anisotropy ratio is constant, meaning that the variation of conductivity with temperature is the same in each direction.

The user also inputs the thermal diffusivity, which is defined as

$$a = \frac{k}{\rho C_p} \quad (12)$$

where

a = thermal diffusivity (m²/s)

k = thermal conductivity (W/mK)

r = density (kg/m³)

C_p = specific heat (J/kgK)

Note: Unlike thermal conductivity and density, the specific heat of materials is often difficult to find in references. Manufacturers often do not measure (or list) this quantity in spec sheets. However, this parameter is only important for transient (time-dependant) cases.

An example of the data for the layers is given in Table 12 below.

Table 12: Sample data for each layer

15	3.430d-3	0.120d0	1.652d-7	1.d0	0.d0
08	8.900d-4	0.294d0	1.045d-7	1.d0	0.d0
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0
07	1.300d-5	41.000d0	1.194d-5	1.d0	0.d0
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0
08	2.030d-4	16.270d0	4.035d-6	1.d0	0.d0
21	2.540d-3	2.232d0	1.151d-6	1.d0	0.d0

Once again, the use of comment lines can be used as an aide in identifying the input, as shown in Table 13.

Table 13: Sample data for each layer with comments

C					
C Data for each layer: C					
C nodes = # of points in that layer or section					
C length = length (thickness) of layer or section					
C					
# of nodes	length (m)	conductivity (W/m/K)	diffusivity (m ² /s)	anisotropy factor	slope (b) of temp. eqn.
"node"	"elde"	"ak"	"alp"	"al"	"slope"
C substrate					
15	3.430d-3	0.120d0	1.652d-7	1.d0	0.d0
C insulation					
08	8.900d-4	0.294d0	1.045d-7	1.d0	0.d0
C insulation					
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0
C heater					
07	1.300d-5	41.000d0	1.194d-5	1.d0	0.d0
C insulation					
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0
C abrasion shield					
08	2.030d-4	16.270d0	4.035d-6	1.d0	0.d0
C ice					
21	2.540d-3	2.232d0	1.151d-6	1.d0	0.d0

Note: LDE = 7 in this example, so there are seven lines of data.

Note: The input units for LEWICE 3.2 are metric.

Note: While performing de-icing or anti-icing simulations, make certain that the top layer contains the thermal properties of ice. If the ice melts, the software will change the properties accordingly.

Note: For any icing, de-icing, or anti-icing simulation, the thickness given to the ice layer in this section will only be used to determine the grid spacing within the ice. The actual ice thickness at each grid point is part of the output from this module.

Note: The total number of points in the normal direction is currently limited to 310 points. The total number of points is the sum of the points in each individual layer minus the number of layers, plus 2 (see equation below).

$$N_{tot} = (\sum n) - lde + 2 \quad (13)$$

The number of points for a given layer must be at least two and less than 310. In addition, the total number of points as given in Equation 13 must be less than 310. If the input data for a given layer is outside this range, the following error message will occur:

Number of points in a layer must be greater than one (1) and less than 310. Your input value of (value) for layer number (value) is outside this range. Blank lines are read as data with a value of zero. Check input file for blank lines.

The software will stop because of the above error.

If the total number of points input is too high, the following error message will occur:

Total number of points in normal direction is too high. Number of points input = (value)

The software will stop because of the above error.

The thickness of each layer must be greater than zero. If the value read for this variable is outside this range, the following error message will occur:

Thickness of a layer must be greater than zero (0).Your input value of (value) for layer number (value) is outside this range.

The software will stop because of the above error.

The thermal conductivity must be greater than zero. If the value read for this variable is outside this range, the following error message will occur:

Thermal conductivity of layer must be greater than zero (0).Your input value of (value) for layer number (value) is outside this range.

The software will stop because of the above error.

The thermal diffusivity must be greater than zero. If the value read for this variable is outside this range, the following error message will occur:

Thermal diffusivity of layer must be greater than zero (0).Your input value of (value) for layer number (value) is outside this range.

The software will stop because of the above error.

The anisotropic ratio must be greater than zero. If the value read for this variable is outside this range, the following error message will occur:

Anisotropic ratio of layer must be greater than zero (0).Your input value of (value) for layer number (value) is outside this range.

The software will stop because of the above error.

10.1.1.3. NOX, ELX, AKX, ALPX, ALX, SLOPEX, ELXY, IVXY

The next NXDE number of lines contains the grid spacing and properties for each section. The primary definition of the thermal properties was provided in the previous set of data. The thermal properties input in this set of lines will be used only for the heater layer of an electrothermal system. For that layer, the thermal properties in this section will override those provided earlier. Each line consists of eight (8) columns of data. The columns are: 1) the number of points in the wrap direction for each section; 2) the width of each section (m); 3) the thermal conductivity of the layer (W/mK) at 0 °C in the normal direction; 4) the thermal diffusivity of the layer (m²/s); 5) the anisotropy ratio; 6) the slope of thermal conductivity with temperature (W/mK²); 7) the additional thickness in the normal direction for that section; and 8) the layer number to which the additional thickness is added.

The first six columns have the same definition as the layer data. The number of points determines the grid spacing in the normal direction. The sum of the widths should be the same as the total wrap distance around the input geometry. The program will automatically adjust the lengths of the two end segments so that this sum is equal to the total wrap distance. The heater locations may have shifted during this process. The user can avoid this by manually adjusting the end segments or by using the variable OFFSET in the next section. The thermal properties identified here will be used only in the heater layer of an electrothermal system as mentioned earlier. The last two columns of data are used to identify one section which can have a larger thickness than the thickness given in the layer data. This concept is illustrated in Figure 5. In this example, section 4 of layer 1 is thicker than the other sections. This feature can be used to model a nose block in an airfoil or other geometric anomalies. Table 14 shows an example input for the section data input.

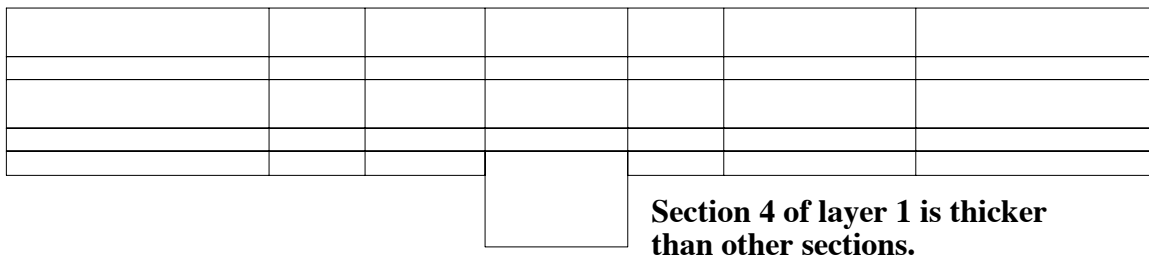


Figure 5: Additional thickness case

Table 14: Sample data for each section with comments

```

C
C   Data for each heater/gap section:
C
# of length cond. diff. anisotropy slope(b) of add. layer

```

nodes	(m)	(W/m/K)	(m ² /s)	of heater	temp. eqn.	length (m)	number
"nox"	"elx"	"akx"	"alpx"	"alx"	"slopex"	"elxy"	"ivxy"
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
C	heater G						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater E						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater C						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	parting strip - heater A						
36	1.905d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater B						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater D						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater F						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
C							

Note: NXDE = 9 in this example, so there are nine lines of data.

Note: Only the lines with data are used by the software. The comments included in this example are extraneous and are ignored by LEWICE 3.2.

Note: The section data is entered from the left side of the computational domain toward the right side. In the physical domain, data is entered starting from the lower surface trailing edge clockwise toward the leading edge and ending back at the trailing edge.

Note: The length of the end segments will be changed so that the total segment length equals the total wrap distance calculated by LEWICE 3.2. An identical amount will be added (or subtracted) from each end segment. Segments may be eliminated, if needed, to match the airfoil wrap distance.

Note: The center of the middle section is **not** aligned with the body leading edge by default. This can be changed in the next set of input data using variable OFFSET or by modifying the lengths of the end segments.

The first six data columns have the same limits on the inputs that the layer data contained. Therefore, the error messages will be the same as well. The additional thickness input cannot be negative. If a negative value is read, the following error message will occur:

Additional length of a section must be greater than or equal to zero (0). Your input value of (value) for section number (value) is outside this range.'

The software will stop because of the above error.

The layer number for the additional thickness must be at least 1 and no greater than the number of layers. If the value input is outside this range, the following error message will occur:

Layer number for additional thickness must be greater than zero (0) and less than or equal to the number of layers. Your input value of (value) for section number (value) is outside this range.'

The software will stop because of the above error.

Example Input

Table 15 shows an example of the complete input data for the first data section.

Table 15: Example data for first section of de-icer input.

```

C          data   for   the   composite   body
-----
C
C   lde= total number of layers in the y-direction
C   nxde= total number of heater sections in the x-direction
C
lde   nxde
007   009
C
C   Data for each layer:
C
C   nodes = # of points in that layer or section
C   length = length (thickness) of layer or section
C
# of   length   conductivity   diffusivity   anisotropy   slope (b) of
nodes  (m)      (W/m/K)      (m^2/s)      factor      temp. eqn.
C   substrate
15    3.430d-3   0.120d0      1.652d-7     1.d0       0.d0
C   insulation
08    8.900d-4   0.294d0      1.045d-7     1.d0       0.d0
C   insulation
18    2.800d-4   0.256d0      1.473d-7     1.d0       0.d0
C   heater
07    1.300d-5   41.000d0     1.194d-5     1.d0       0.d0

```

```

C insulation
18 2.800d-4 0.256d0 1.473d-7 1.d0 0.d0
C abrasion shield
08 2.030d-4 16.270d0 4.035d-6 1.d0 0.d0
C ice
21 2.540d-3 2.232d0 1.151d-6 1.d0 0.d0
C
C Data for each heater/gap section:
C
# of length cond. diff. anisotropy slope(b) of add. layer
nodes (m) (W/m/K) (m^2/s) of heater temp. eqn. length (m) number
C gap (no heat input)
10 0.9144d0 0.256d0 1.637d-7 1.d0 0.d0 0.d0 3
C heater G
14 3.175d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C heater E
21 2.540d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C heater C
21 2.540d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C parting strip - heater A
36 1.905d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C heater B
21 2.540d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C heater D
21 2.540d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C heater F
14 3.175d-2 41.000d0 1.194d-5 1.d0 0.d0 0.d0 3
C gap (no heat input)
10 0.9144d0 0.256d0 1.637d-7 1.d0 0.d0 0.d0 3
C

```

Note: It is not necessary for the input data to align on a given column. This was done in this example to improve readability.

Note: Comment lines have been added to improve readability. The software will ignore the comments and only read the data lines.

10.1.2. De-icer Section 2

This section of input data describes the heater power and cycle times. There are two types of heater modes. In the first mode, a heater can be given a specific wattage, an ON time, an OFF time, and a LAG time. In the second mode, it will be temperature controlled. In this mode, the

user supplies an initial wattage, an ON temperature, an OFF temperature, and the layer number which is used to control the heaters. This means that the heater can be controlled by the surface temperature instead of the heater temperature if desired. The inputs for this section will now be discussed in more detail.

Note: If there was an error reading data in the first section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

10.1.2.1. IJDE, OFFSET, IPAR

The first line of data in this section contains three variables: the layer number for an electrothermal heater, the offset for the parting strip heater, and the number of parameter studies. The heater layer number defines the layer from which heat will emanate for an electrothermal system. It also defines the layer which will use the thermal properties entered in the section data input earlier. A value for this variable must be entered even when using a hot air system. The input for this variable must be between 1 and the total number of layers. If the value input is outside this range, the following error message will occur:

Heater number must be greater than zero (0) and less than or equal to number of layers.

Your input value of (value) is outside this range.

Blank lines are read as data with a value of zero. Check input file for blank lines.

The software will stop because of the above error.

Additionally, if ice is the top layer, it cannot also be used as the heater layer. If this occurs, the following error message will be generated:

The ice layer cannot be the same as the heater layer.

The software will stop because of the above error.

The offset variable contains the distance, in meters, of the center of the computational domain from the leading edge. If $OFFSET = 0$, then the midpoint (determined by distance) of the de-icer sections will be aligned with the midpoint (determined by distance) of the airfoil. For values less than 0, the midpoint will be moved toward the lower surface while for values greater than 0, it will be moved toward the upper surface. If the absolute value of $OFFSET$ is greater than the chord length, the following warning message will occur:

The offset value may be past the trailing edge. $OFFSET = (value) s/c$

The number of parameter studies can be used to run more than one heater configuration using the same flight and meteorological conditions. The user can also opt to run different conditions as separate cases. The value for this variable must be between 1 and 29. If the value input is outside this range, the following error message will occur:

Number of parameter studies must be greater than zero (0) and less than thirty (30). Your input value of (value) is outside this range. Blank lines are read as data with a value of zero. Check input file for blank lines.

The software will stop because of the above error.

Note: Normal operation of this software uses one parameter study per run (IPAR = 1). While a multiple parameter study option has been included, it is possible that output from a multiple parameter study run may be different from the output that would have been achieved if the case were run separately. This option has seen limited testing. Section 13.1 describes a potentially better way to perform a parameter study by using external scripts.

Table 16 shows an example of this data line with comments.

Table 16: Example data line

c	data for the heater		
ijde	offset	ipar	
004	0.005	1	

10.1.2.2. QDE, TON, TOFF, TLAG, ICFLAG

The next NXDE lines of data contain the heater power and cycle times. The user must input a line of data for each section, even if there is no heat generation in that section. Each line must contain five columns of data. These columns represent: 1) the heater power (kW/m²); 2) heater ON time (sec) or heater ON temperature (K) (for temperature controlled heaters); 3) heater OFF time (sec) or heater OFF temperature; 4) heater LAG time (sec); 5) flag for using temperature controlled heaters. Figure 6 shows an example of each type of heater cycle

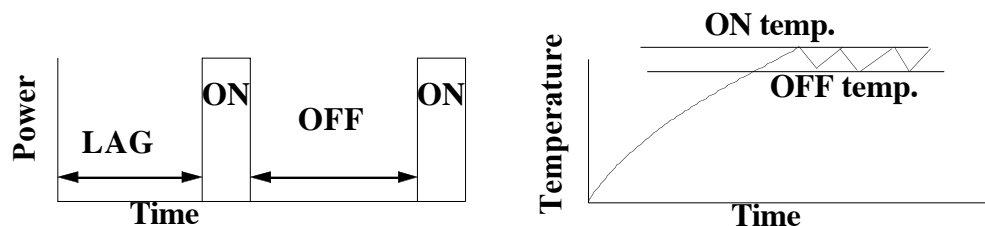


Figure 6: Example of heater cycle types

The heater power input can also represent the initial wall heat flux for an optimal hot air system. See IQW in de-icer section 4 input for more details. The heater power is input in units of kW/ m² and must be S 0. If the value input is outside this range, the following error message will occur:

Heater wattage must be greater than or equal to zero (0).Your input value of (value) for layer number (value) and parameter study (value) is outside this range.

The software will stop because of the above error.

Note: Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K respectively.

The heater ON time must also be S 0. If the value input is outside this range, the following error message will occur:

Heater on time must be greater than or equal to zero (0). Your input value of (value) for layer number (value) and parameter study (value) is outside this range.

The software will stop because of the above error.

The heater OFF time must also be ≥ 0 . If the value input is outside this range, the following error message will occur:

Heater off time must be greater than or equal to zero (0). Your input value of (value) for layer number (value) and parameter study (value) is outside this range.

The software will stop because of the above error.

The heater LAG time must also be ≥ 0 . If the value input is outside this range, the following error message will occur:

Heater lag time must be greater than or equal to zero (0). Your input value of (value) for layer number (value) and parameter study (value) is outside this range.

The software will stop because of the above error.

The temperature control flag has the following definition. If the flag = 0, then heaters are not temperature controlled and the heater will operate with a given wattage and ON/OFF cycle time. If the flag is > 0 , then it represents the layer number which is used to control the heater cycle time. In this mode, the wattage will be adjusted until the temperature remains between the temperature range input. The exact temperature used will reside on the lower surface of the layer at the midpoint of the section, as illustrated in Figure 7.

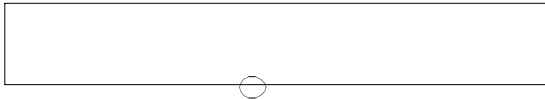


Figure 7: Temperature location for temperature controlled heaters

Valid input values for this flag are then 0 up to the number of layers. Enter the top layer number to control the heater with the surface temperature. If the input value is outside this range, the following error message will occur:

Temperature control flag must be greater than or equal to zero (0) and less than or equal to the number of layers. Your input value of (value) for layer number (value) and parameter study (value) is outside this range.

The program will stop because of the above error.

If temperature controlled heater is used, the ON temperature must be greater than the OFF temperature. If it is not, the following error message will occur:

Temperature control flag is on. ON temperature must be greater than OFF temperature. T_on = (value) T_off = (value)

In addition, if a temperature controlled heater is used, both temperatures must be greater than the ambient temperature. To effectively de-ice the surface, these temperatures should also be greater than freezing but this is not required for proper operation of the software. If either value is below the ambient temperature, the following error message will occur:

Temperature for temperature controlled heater cannot be less than the ambient temperature.

Temperature = (value)

The user should also refer to Section 10.1.4.2 for an additional use of the temperature controlled heater function.

Note: Temperature controlled heaters as modeled in this software do not necessarily determine the “optimum” heat flux. Initially, it uses the wattages input until the temperature exceeds the ON temperature. It then turns off all heat until the temperature drops below the OFF temperature. Heat is then turned on, but at a lower wattage. This process continues until the simulation time (TSTOP) is reached. If the simulation time is sufficiently long and the wattage has not changed for a long time, then the wattage reported in “**qopt.dat**” may be close to the optimum heat flux. **Results to date have been inconsistent for this option.**

10.1.2.3. Example Input

Table 17 then gives a complete input example for the second de-icer section, including comments.

Table 17: Complete input data for de-icer section 2

C	data for the heater				
C					
	ijde	offset	ipar		
	004	0.005	1		
		"QDE"	"TON"	"TOFF"	"TLAG"
		heater density	time on	time off	lag time
		(kWatts/m^2)	(sec)	(sec)	temp cntrl.
					flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	12.02d0	5.d0	115.d0	115.d0	0
C	heater E				
	11.19d0	5.d0	115.d0	115.d0	0
C	heater C				
	11.25d0	5.d0	115.d0	110.d0	0
C	parting strip (heater A) does not turn off				
	07.78d0	150.d0	0.d0	0.d0	0
C	heater B				
	11.68d0	5.d0	115.d0	110.d0	0
C	heater D				
	11.27d0	5.d0	115.d0	115.d0	0
C	heater F				
	11.75d0	5.d0	115.d0	115.d0	0
C	gap (no heat input)				

0.0d0 0.d0 0.d0 0.d0 0

Table 18 shows an example using temperature controlled heaters

Table 18: Complete input data for de-icer section 2 using temperature controlled heaters

C	data for the heater				
C					
	ijde	offset	ipar		
	004	0.005	1		
		"QDE"	"TON"	"TOFF"	"TLAG"
		heater density	temp on	temp off	lag time
		(kWatts/m^2)	(K)	(K)	(sec)
					temp cntrl.
					flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	12.02d0	277.d0	275.d0	0.d0	6
C	heater E				
	11.19d0	277.d0	275.d0	0.d0	6
C	heater C				
	11.25d0	277.d0	275.d0	0.d0	6
C	parting strip (heater A)				
	07.78d0	277.d0	275.d0	0.d0	6
C	heater B				
	11.68d0	277.d0	275.d0	0.d0	6
C	heater D				
	11.27d0	277.d0	275.d0	0.d0	6
C	heater F				
	11.75d0	277.d0	275.d0	0.d0	6
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0

Note: To model a section which does not have a heater, simply input QDE = 0.

Note: To model an electrothermal anti-icing system or parting strip, set TOFF = 0.

Note: $W/in^2 * 1.55 = kW/m^2$

10.1.3. De-icer Section 3

The third section of de-icer input is used to define boundary conditions for the simulation. There are four boundary conditions which need to be defined: at the top and bottom and on both sides. The four sides are illustrated in Figure 8.

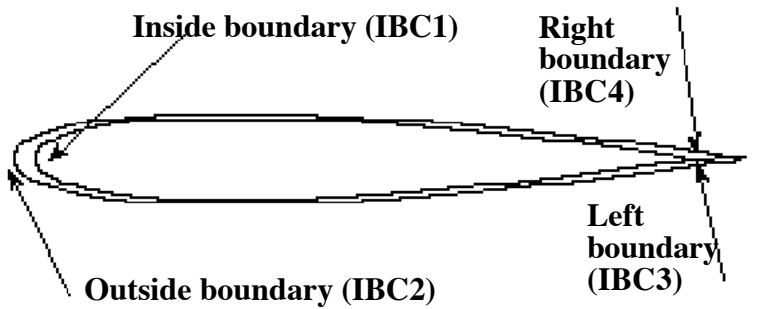


Figure 8: Boundaries on Sample Geometry

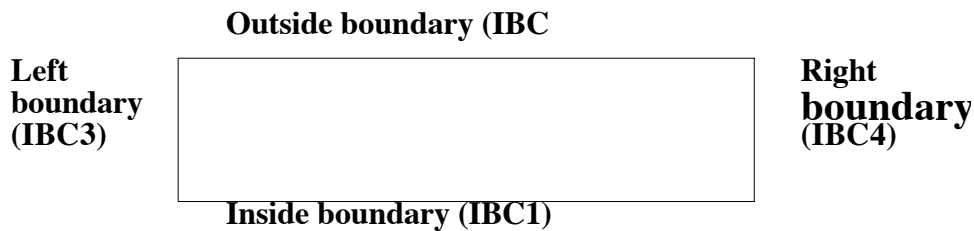


Figure 9: Boundaries on Computational Geometry

Note: If there was an error reading data in a previous section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

10.1.3.1. IBC1, IBC2, IBC3, IBC4

The first line of this section contains four flags which represent the boundary condition on the inside, outside, “left” and “right” boundaries as shown in the previous figure. The “left” boundary occurs at the lower surface trailing edge while the “right” boundary occurs at the upper surface trailing edge. Each flag can have a value from 1 to 3 where IBC = 1 represents a constant temperature boundary, IBC = 2 represents a convective boundary, and IBC = 3 represents a constant wall flux boundary. A value outside this range will cause the following warning message to appear:

Boundary condition index must be greater than zero (0) and less than or equal to three (3)

Your input value of (value) is outside this range. Setting boundary condition = 2 (convection)

Blank lines are read as data with a value of zero. Check input file for blank lines.

As stated in the warning, the boundary condition will be set to the default (convection) boundary. The standard boundary conditions for de-icing and anti-icing conditions are: IBC1 = 2 or 3 (IBC1 = 3 is used to model a hot air system.); IBC2 = 2; IBC3 = 2; IBC4 = 2. If a constant temperature boundary condition is used, the following warning message will appear:

Constant temperature boundary condition is not normally used in icing calculations.

If IBC2 = 3, IBC3 = 3, or IBC4 = 3, the following warning message will appear:

Constant heat flux boundary condition is not normally used in icing calculations.

10.1.3.2. TG1, TG2, TG3, TG4

The second line of data in this section also contains four values which represent the ambient temperature (in degrees Kelvin) at the boundary. If IBC = 1 or 2, this temperature should be the same as the ambient temperature entered in the main input file (variable TINF). If any of these temperatures are not equal to the ambient temperature, the following warning will appear:

Temperature at boundary is not equal to the ambient temperature input from the main LEWICE input file. Setting temperature equal to TINF.

The temperature on the inside surface (IBC1) will not be reset to the ambient temperature as this option could be used to model a hot air anti-icer.

10.1.3.3. H1, H2, H3, H4

The third line of data in this section contains the four values of the heat transfer coefficients ($\text{W}/\text{m}^2/\text{K}$) at the four boundaries. Each value must be ≥ 0 . If the data input is outside this range, the following error message will occur:

Heat transfer coefficient must be greater than or equal to zero (0).

The software will stop because of the above error.

Additionally, since the software is designed to emulate an enclosed boundary, the heat transfer coefficients on the “left” and “right” boundaries should be set to zero (insulated/reflective boundary). If a value > 0 is read, the following warning message will appear:

Convective heat flux boundary condition is not normally used in icing calculations.

Note: The heat transfer coefficient input at the top surface may be overwritten using the IBOUND flag in the next input section.

Note: The inside heat transfer coefficient, H1, can be used to model a hot air anti-icer. Section 10.1.4 provides a better way to model of this type of system.

10.1.3.4. QW1, QW2, QW3, QW4

The fourth line in this data section contains the four wall heat fluxes (kW/m^2) for each boundary. These values are read even if a wall heat flux boundary is not used. These values should be ≥ 0 . If a negative heat flux is read, the following warning message will appear:

Negative heat flux boundary condition is not normally used in icing calculations.

The software will allow the use of this boundary condition, however negative heat flux values have not been tested.

Note: The wall heat flux, QW1, can be used to model a hot air anti-icer. Section 10.1.4 provides a better way to model of this type of system.

Note: Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K respectively.

10.1.3.5. Example Input

An example of this data section is given in Table 19 below.

Table 19: Example input for boundary condition data

C	boundary	condition	data
C			
C	Type of boundary condition:		
C			
	"ibc1"	"ibc2"	"ibc3" "ibc4"
2	2	2	2
C			
C	Temperature data for bottom, left, and side boundaries		
C			
	"tg1"	"tg2"	"tg3" "tg4"
263.3	263.3	263.3	263.3
C			
C	Heat transfer coeff. in W/m ² /K		
C			
	"h1"	"h2"	"h3" "h4"
3.d0	3.d2	0.d0	0.d0
C			
C	Wall heat flux in kW/m ²		
C			
	"qw1"	"qw2"	"qw3" "qw4"
0.d0	0.d0	0.d0	0.d0

10.1.4. De-icer Section 4

The fourth section of the de-icer input file contains several flags and other variables which are used to access specialized features of the software. Since these features modify the basic way the software operates, care should be taken to ensure that the proper flags are set for the system the user wants to model.

Note: If there was an error reading data in a previous section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

10.1.4.1. TAIRH, AIRMD, AIRUP, XAIRMD

The first line of this section is used to input variables for an anti-icing system. There are two ways to model a hot air system with LEWICE 3.2. The user may supply the boundary condition as either a heat transfer coefficient or as a heat flux. In this first method, the user specifies the internal heat transfer coefficient (using variable H1 from Section 10.3), the incoming temperature of the bleed air (TAIRH), the mass flow rate (AIRMD), the mass fraction flowing up from the supply location (AIRUP) and the air supply location (XAIRMD). This model is diagrammed in Figure 10.

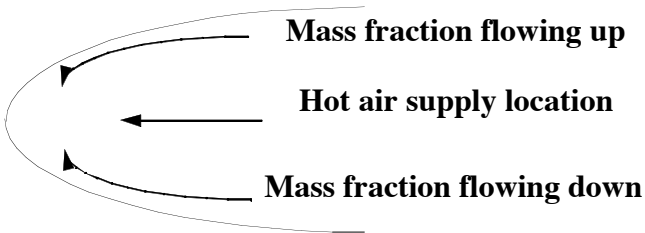


Figure 10: First hot air model

The first variable on this input line, TAIRH, is the hot air supply temperature in degrees Kelvin. This value should be greater than the melting point of ice (273.15 K) for this model to work properly. If the value read is less than this, the following warning message will appear:

Temperature for bleed air is less than freezing. This is not normally used in icing calculations.

The next variable on this line, AIRMD is the mass flow rate per unit span of the hot air supply in kg/ms. This variable must be ≥ 0 . If a negative mass flow rate is entered, the following error message will appear:

Mass flow rate for bleed air must be greater than or equal to zero (0). The software will stop because of the above error.

The third variable on this line, AIRUP, is the mass fraction flowing up (or to the right in the computational domain) as shown in Figure 10. The fraction flowing down is then fixed as $(1 - \text{AIRUP})$. Since this variable is a fraction, its value must be $0 \leq \text{AIRUP} \leq 1$. Values outside this range will generate the following warning message:

Mass fraction going up must be greater than or equal to zero (0) and less than or equal to one (1). Setting AIRUP = 0.5

As stated in the warning message, the value for AIRUP will be set to the default value of 0.5. The user should terminate the case and change the input file if this value is unacceptable.

The last variable on this line, XAIRMD, is used to define the hot air supply location in meters as measured from the leading edge. A negative value should be used to define a location on the lower surface of the body and a positive value should be used to define a location on the upper surface of the body. If the absolute value of XAIRMD is greater than the chord length of the body, the following warning message will be generated:

Mass flow rate input location is greater than one chord length. This value will be set to the trailing edge.

As suggested by the warning message, the supply location will be placed at the trailing edge of the airfoil. Table 20 shows an example input for this data line with comments.

Table 20: Sample input line for hot air anti-icer

C				miscellaneous data			
tairh	airmd	airup	xairmd				
4.5d2	1.49d-2	0.5d0	0.d0				

10.1.4.2. NHOLE, DHOLE, DTUBE, RDIST, YCASE, XUHND, XLHND

The second line of data in this section provides information for modeling a “piccolo tube” anti-ice system. A piccolo tube system consists of a pipe near the inside surface of the leading edge that contains hot air from the compressor. Air is directed from this tube toward the surface via small holes in the tube. This system is diagrammed in Figure 11.

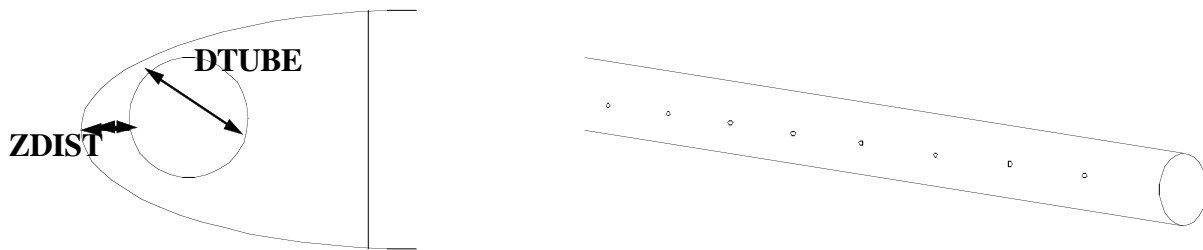


Figure 11: Side and front view of piccolo tube system

The data in this line of input will be used if variable ICORR in Section 10.1.4.3 is greater than zero. The data is used for the correlations given in the next section. The variables on this line are defined as follows: number of holes (NHOLE), hole diameter (DHOLE), tube diameter (DTUBE), spanwise distance between holes (RDIST), spanwise location of simulation (YCASE), end of heated section on upper surface (XUHND) and end of heated section on lower surface (XLHND). The user should only include one set of holes in the input and use variable RDIST for the distance between holes since the program already assumes a repeating hole pattern. The spanwise location of the run is included so that the user can simulate the heat transfer along the holes or at spanwise locations. The number of holes must be between 1 and 10. If a number outside this range is read, the following warning message is issued:

Number of jet holes must be between 1 and 10. Do not repeat hole pattern. Setting NHOLE = 1

The second variable on this line, DHOLE, contains the hole diameter in meters. This value must be greater than zero. If a value less than zero is read, the following error message will be generated:

Hole diameter must be greater than zero.

The program will stop due to this error.

Similarly, DTUBE represents the piccolo tube diameter in meters. This value must be greater than zero. If a value less than zero is read, the following error message will be generated:

Tube diameter must be greater than zero.

The program will stop due to this error.

The fourth variable, RDIST, is the spanwise distance between two holes that are at the same chordwise location. This variable is input in number of hole diameters. This means that if holes are 1.5 mm in diameter and are spaced 3 cm apart, then RDIST should be equal to 20 (0.03/0.0015). This variable must also be greater than zero. If a value less than zero is read, the following error message will be generated:

Distance between holes must be greater than zero.

The program will stop due to this error.

The next variable on this line, YCASE, contains the spanwise location of the simulation relative to the holes. This variable is read so that users may determine the solution between holes as well as directly over the hole. This variable may be positive or negative, but its absolute value must be less than the distance between holes (RDIST). The spanwise locations in this context are relative to the hole locations and do not represent an absolute spanwise value. The user may set YCASE = 0 and then input the hole locations (YHOLE as described below) as negative or positive relative to this spanwise location or the user may set one of the hole locations or any arbitrary point as the relative zero span location. If the value read is outside this range, the following warning message will be generated:

Span distance to hole cannot be greater than the distance between holes. Setting YCASE = RDIST.

Figures 11 and 12 show these variables in more detail.

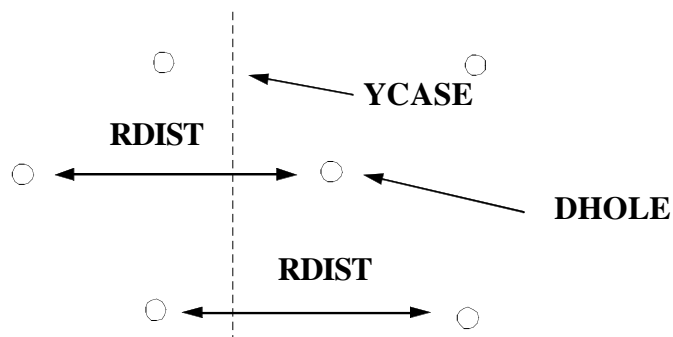


Figure 12: Piccolo tube variables (NHOLE = 3)

Variables XUHND and XLHND define the chordwise distance from the leading edge to the end of the heated section in meters for a piccolo tube system. This value is relative to the leading edge of each body. The values must be greater than the liner end locations (XULND, XLLND) if

ILINER = 1 in the section below and less than the trailing edge of that body. If the heated end location is less than the liner end location, the error message in Section 10.1.4.5 is displayed. If the heated end location is past the trailing edge, the following warning message will be generated:

End of heated section is past the trailing edge. Setting end of heated section to the trailing edge.

Table 21 shows an example of this input line.

Table 21: Sample input line for first piccolo tube input

nhole	dhole	dtube	rdist	ycase	xuhnd	xlhnd
3	0.00132	0.0381	20.	0.	0.138	0.138

10.1.4.3. ICORR, HCONST, REP, RP, ZP, RRP, DP

The next line of data in this section contains information for the piccolo tube heat transfer correlation. This correlation can take one of two forms depending on the input value of ICORR. For ICORR = 1, the Nusselt number is calculated from the following equation:

$$Nu = HCONST(Re^{REP}) \frac{\left(24 - \left(\frac{z}{d}\right)^{ZP} - 7.75\right)}{533 + 44\left(\frac{r}{d}\right)^{RP}}$$

where Re and Nu are the Reynolds number and Nusselt number based on the hole diameter, z is the distance from the hole to the inside surface of the airfoil, r is the radial distance from the hole, and d is the hole diameter. These variables are raised to the input powers REP, ZP, and RP. Goldstein et. al.¹⁷ give the following values for the constants:

Table 22: Constants for piccolo tube correlation

HCONST	1
REP	0.76
ZP	1
RP	1.394

This correlation was determined experimentally for impinging jets similar to piccolo holes for Reynolds numbers up to 1.28×10^5 , wall distances from 2 to 20 hole diameters, and radial distances from 0 to 6 hole diameters. All of these parameters are expected to be within the range needed for piccolo tube simulation. For ICORR = 2, the Nusselt number correlation takes the following form:

$$Nu = HCONST(Pr)^{\frac{1}{3}}(Re^{REP})\left(\frac{z}{d}\right)^{ZP}(RDIST)^{RRP}\left(\frac{r}{d}\right)^{RP}\left(\frac{DTUBE}{d}\right)^D$$

where RDIST is the spanwise distance between holes divided by the hole diameter and DTUBE is the tube diameter. RDIST is raised to the RRP power and DTUBE is raised to the DP

power. If Nusselt number does not depend on one or more of these variables, simply set the power to zero. The form of this correlation follows several references in the literature where Nusselt number is measured experimentally. The majority of the correlations in the open literature quote an upper limit of 4×10^4 for Reynolds number which may still be applicable for many piccolo tube applications. Brown et. al.¹⁸ developed a correlation which matched the form given above. Their correlation was specifically developed for a piccolo tube application but they did not publish the correlation coefficients. Users can also determine the constants for this equation experimentally. The table below gives a summary of the constants found in literature and their source.

Correlation constants from various references

Author	HCONST	REP	ZP	RRP	z/d range	RDIST range	RE limit
Tawfek ¹⁹	0.453	0.69	-0.22	-0.38	6-58	2-30	41,000
Gau & Chung ²⁰	0.251	0.68	0.15	-0.38	2-8	8-46	35,000
Gau & Chung ²⁰	0.394	0.68	-0.32	-0.38	8-16	8-46	35,000
Huber & Viskanta ²¹	0.285	0.71	-0.123	-0.725	0.25-6	4-8	20,500
Goldstein & Seol ²²	0.182	0.7	-0.16	-0.32	2-8	4-8	40,000
Hrycak ²³	0.337	0.7	-0.22	-0.16	2-8	2-7	30,000

Note: The applicability of the jet correlations provided to piccolo tube anti-icing systems is unknown.

For ICORR = 0, a correlation will not be used for the internal heat transfer and the user must supply heat transfer coefficient values or heat flux values by using the flags IBLEED, IQAIN, or IQW defined below. ICORR can be 0, 1, or 2. For ICORR values outside this range, the following warning message will be generated:

ICORR must be between zero (0) and two (2). Your value of (value) is outside this range. Setting ICORR = 0

Additionally, since these correlations have not been validated for piccolo tube application, when the value of ICORR is 1 or 2 the following warning message will be generated:

Heat transfer correlations have not been validated.

The value of HCONST must be greater than zero. If it is not, the following error message will be generated:

Correlation constant must be greater than zero.

The program will stop due to this error.

The absolute values on the powers (REP, ZP, RP, RRP, DP) in the correlation above should be less than 10. If any value is greater than 10, one of the following warning messages will be generated:

Strange power on Reynolds Number. Value read is (value). Is this what you want?

Strange power on radial distance. Value read is (value). Is this what you want?

Strange power on hole distance to wall. Value read is (value). Is this what you want?

Strange power on hole distance. Value read is (value). Is this what you want?

Strange power on tube diameter. Value read is (value). Is this what you want?

An example of this input line is given below.

Table 23: Sample input line for piccolo tube correlation

```
icorr hconst rep   rp   zp rrp dp
0      1.0  0.76  1.394  1.  0.  0.
```

10.1.4.4. XHOLE, YHOLE, ZDIST

The next set of data provides the locations of the piccolo tube holes relative to the airfoil surface. The program will read in NHOLE lines of data for this set of inputs. The first variable, XHOLE, provides the wrap distance in meters from the leading edge of the airfoil to the impact location of the jet on the airfoil surface. XHOLE will be negative for the lower surface and positive for the upper surface. If the value read is not on the airfoil, the following warning message will be generated:

Chordwise hole location is past the trailing edge. Setting XHOLE = 0

The second variable, YHOLE, provides the spanwise location of the jet impact in meters. The spanwise distance is relative to the value of YCASE input. This value will be negative if the hole is to the left of YCASE and positive if the hole input is to the right as shown in the figure below. It is not necessary for the user to input the hole closest to the YCASE line as the program uses the holes on both sides of the YCASE line to calculate the heat transfer coefficient. Each YHOLE value must be less than the distance between holes. If a value is read which is outside this limit, the following warning message will be generated:

Span distance to hole cannot be greater than the distance between holes. Setting YHOLE = RDIST.

The last variable on this line, ZDIST, defines the distance from the airfoil surface to the jet hole in meters. This distance must be greater than zero. If a value outside this range is read, the following error message will be generated:

Distance from piccolo to surface must be greater than zero.

The program will stop due to this error. These distances are illustrated in Figure 13.

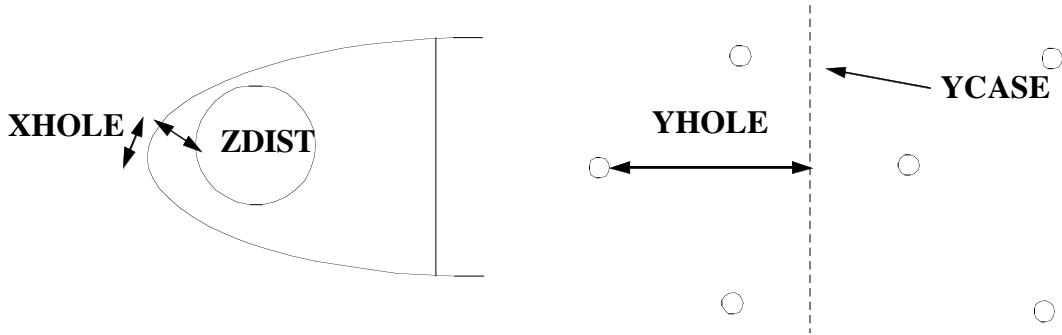


Figure 13: XHOLE and YHOLE definitions

Table 24: Sample input line for piccolo hole locations

xhole(m)	yhole(m)	zdist(m)
0.0176	-0.033	0.0107
0.000	0.033	0.0091
0.0176	-0.033	0.0076

10.1.4.5. ILINER, XULINER, XLLINER, XULST, XULND, XLLST, XLLND

The next line of data in this section defines inputs for a piccolo tube liner. A liner is an insert that increases the efficiency of heat transfer downstream of the tube. An example of a piccolo tube liner is presented in the figure below.

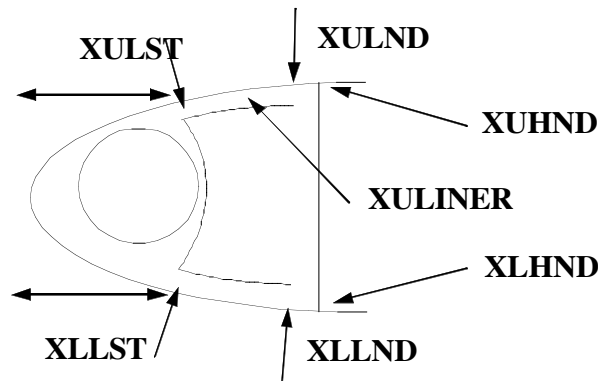


Figure 14: Piccolo tube with liner

The first variable, ILINER, is a flag which indicates that a liner exists. If ILINER = 0, no liner exists and if ILINER = 1, then a liner will be modeled. If any other value is read, the following warning message will be generated:

ILINER must be either zero (0) or one (1). Your value of (value) is outside this range. Setting ILINER = 0.

Use of the liner flag will increase the velocity of the flow in that region using the Bernoulli equation. The increased velocity then provides an increase to the heat transfer coefficient through the correlation provided earlier. The validity of this assumption is not known. Therefore if $ILINER = 1$, the following warning message will be generated:

Correlation for liner has not been validated.

The next two variables, XULINER and XLLINER, define the average distance from the wall to the liner in meters. XULINER is the distance between the surface and the liner on the upper surface while XLLINER defines the distance between the surface and the liner on the lower surface. Both variables must be greater than zero and less than the airfoil thickness at the liner location. If the value is read is outside this range, the following error message will be generated:

Liner distance is outside specified range. Airfoil thickness here is (value).

The program will stop due to this error.

The next set of variables defines the upper and lower limits for the liner. Variable XULST is the chordwise location of the start of the liner on the upper surface in meters while XULND is the end location of the liner. These variables are measured as the distance chordwise as diagrammed Figure 14 above. XLLST and XLLND are the starting and ending distances in meters of the liner on the lower surface respectively. All measurements are assumed to be in meters from the leading edge of each body. Variables XULST and XLLST must be behind (greater than) the last row of jets and in front of (less than) both the end locations (XULND, XLLND) of the liner and the end locations of the heated zone. If a value is read that is outside this range, the following error message will be generated:

Liner start location is outside the defined range. Value must be between (value) and (value).

The program will stop due to this error.

Variables XULND and XLLND must be greater than the start locations (XULST, XLLST) and less than the end of the heated section (XUHND, XLHND). If the end location is less than the start location, the error message above will be generated. If it is greater than the end of the heated section, the following error message will be generated:

Liner end location is outside the defined range. Value must be less than (value).

The program will stop due to this error.

The table below provides an example of this row of input.

Table 25: Piccolo liner input

iliner	xuliner	xlliner	xulst	xulnd	xllst	xllnd
0	0.01	0.01	0.052	0.128	0.052	0.128

10.1.4.6. *13D, IQAIN, IBLEED, IQW, ITSUR*

The next line of data in this section defines flags which determine the type of anti-icer which is to be modeled. These flags also control reading of external input files into LEWICE. It is recommended that the user only select one type of anti-icing model and not use a combination of models. The software will accept the input, but the resulting model may not be realistic. The first

variable on this line, I3D, controls the input of a file used for performing a pseudo-3D analysis. If I3D = 0, then no additional input data file will be read. If I3D = 1, the program will read in a file called “**stream.inp**” (described in Section 10.4) which contains the dimensionless spanwise distance from the current 2D cross-section to the next. The distance may also represent the axial distance between two 2D cross-sections of an engine inlet. The program will use this data in the mass and energy balance and is especially useful for determining the height and extent of runback ice formations. This option is often used when supplying other 2D streamline information from a 3D flow field and 3D trajectory analysis. Valid inputs for I3D are 0 and 1. If a value outside this range is read, the following warning message will be generated:

3D streamline flag must be zero (0) or one (1). Your value of (value) is out of range. Setting I3D = 0

The second variable on this line, IQAIN, controls the input of a wall heat flux in kW/m² at the inside surface. Valid inputs for this flag are 0 (OFF) and 1 (ON). If IQAIN = 1, the software will read a separate file called “**qain.inp**” (described in Section 10.3) which contains the wall heat flux on the inside surface as a function of the wrap distance. Also see Section 10.1.4.8 for additional options for the input from this file. If a value outside the valid range (0 or 1) is input, the following warning message will be generated:

Heat flux flag must be zero (0) or one (1) Your value of (value) is out of range. Setting IQAIN = 0

IQAIN will then be set to the default value of 0 (no data read).

The next flag on this line, IBLEED, can be used to read in a file containing heat transfer coefficients in W/m²K instead of a wall heat flux. If IBLEED = 0, no data will be read. If IBLEED = 1, the software will read in a separate file called “**hi.inp**” (described in Section 10.2) containing the heat transfer coefficients on the inside surface as a function of the wrap distance. See Section 10.1.4.8 for additional options on the input from this file. If a value outside the valid range (0 or 1) is input, the following warning message will be generated:

Bleed air flag must be zero (0) or one (1) Your value of (value) is out of range. Setting IBLEED = 0

IBLEED will then be set to the default value of 0 (no data read).

Note: Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K respectively.

The next flag on this line, IQW, can be used to estimate the “optimal” wall heat flux for an anti-icing system. If IQW = 0, no optimization is performed. If IQW = 1, then the software will use the wattages input in Section 10.1.2.2 along with the temperature control information described in that section to determine the wall heat flux required to maintain the selected temperature within the range specified. In contrast to Section 10.1.2.2, where the heat generation was supplied from layer IJDE, the heat generation case for IQW = 1 comes from the inside wall heat flux. For two cases which both use the temperature control option, IQW = 0 would then be used to model an electrothermal system while IQW = 1 would be used to model a hot air system. For a hot air system, the user must select either heat flux input (IQAIN = 1) or heat transfer coefficient input (IBLEED = 1). ITSUR = 1 as described below performs a similar iteration using the

surface temperatures input. If a value outside the valid range (0 or 1) is input, the following warning message will be generated:

***Optimum bleed air flag must be zero (0) or one (1) Your value of (value) is out of range.
Setting IQW = 0***

IQW will then be set to the default value of 0 (no data read).

Note: The IQW flag does not necessarily determine the “optimum” heat flux. Initially, it uses the wattages (or heat fluxes) input until the temperature exceeds the ON temperature. It then turns off all heat until the temperature drops below the OFF temperature. Heat is then turned on, but at a lower value. This process continues until the simulation time (TSTOP) is reached. If the simulation time is sufficiently long and the wattage has not changed for a long time, then the wattage reported in “**qopt.dat**” may be close to the optimum heat flux. Also see the description of temperature controlled heaters in Section 10.1.2.2. **Results to date have been inconsistent for this option.**

When IQW = 1, the following warning message will be generated:

IQW function has not been thoroughly tested. Results to date have been inconsistent for this option.

The last flag on this line, ITSUR, allows the user to read in a surface temperature distribution. If ITSUR = 1, the software will read in a separate file called “**tsurf.inp**” as described in Section 10.10. This file contains the surface temperature at each wrap distance location on the surface. The program will then calculate the heat flux necessary to achieve that distribution and the residual ice (if any) that results from that distribution. The heat flux can be electrothermal or bleed air as determined by the IQW flag described previously. When IQW = 0, the program will use temperature-controlled electrothermal heating as described in Section 10.1.2.2. When IQW = 1, the program will assume a wall heat flux. In either mode, the program will use the temperatures input as the controlling mechanism for the heat flux. Users have requested this option for calibration with tunnel test output. Valid inputs for this flag are 0 and 1. If the user inputs any other value, the following warning message will be generated:

***Surface temperature flag must be zero (0) or one (1). Your value of (value) is out of range.
Setting ITSUR = 0***

Use of this flag has been limited to the intended calibration feature. This function also requires an iteration similar to the IQW function above to determine heat flux. Therefore, the same caveats apply about the software’s ability to converge on heat flux. When ITSUR = 1, the following warning message will be generated:

ITSUR function has not been thoroughly tested. Results to date have been inconsistent for this option.

In addition, these flags are mutually exclusive. That is, only one of these flags can be set for a given input file. If IQAIN and IBLEED are both equal to one, the following warning message will be generated:

Heat flux flag and bleed air flag cannot both be on. Setting IBLEED = 0

If ITSUR and IBLEED are both equal to one, the following warning message will be generated:

Surface temperature flag and bleed air flag cannot both be on. Setting IBLEED = 0

If IQAIN and IQW are both equal to one, the following warning message will be generated:

Heat flux flag and optimum bleed flag cannot both be on. Setting IQAIN = 0

Note: IBLEED and IQAIN flags are most often used when the input data is known. Examples of this condition occur when experimental test data is available or when values are calculated by another program or via correlation. IQW is most often used when this data is not available. ITSUR has only been used for calibration with tunnel data.

An example input for this line is shown in Table 26.

Table 26: Sample input for anti-icing flags

i3d	iqain	ibleed	iqw	itsur
0	0	0	0	0

10.1.4.7. ICOND, IBOUND, INIT, ISH, ISTD, IGDE

The next line of this data section contains six flags which control different features within the software. The user should consider each flag carefully to ensure that their problem is being properly modeled. The default values represent those most likely to be used to set up a de-icing or anti-icing simulation. There is, however, no “best” value for these flags as they represent ways to model different problems.

Note: If there was an error reading data in a previous section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

The first variable on this data line, ICOND, controls the use of the slower de-icer model during an initial time lag. ICOND = 0 will turn off conduction during an initial time lag and can be used in a de-icing simulation where the body is allowed to accrete ice before any heaters turn on. When ICOND = 0, the software will use the standard LEWICE accretion routines for the length of the shortest time lag input in Section 10.1.2. If ICOND = 1, the de-icer model will be used starting at time = 0. If there is an initial accretion period as described above, the user should select ICOND = 0 to accelerate the solution. Valid inputs for this flag are 0 and 1. If a value outside this range is read, the following warning message will be generated:

Conduction flag must be zero (0) or one (1) Your input value of (value) is out of range. Setting ICOND = 1

If ICOND = 0 but there is a zero second time lag on any of the heaters (or if the user selected a wall heat flux boundary condition), the following warning message will appear:

Conduction flag must be = 1 since heaters will turn on immediately.

Setting ICOND = 1

The second variable in this data line, IBOUND, is used to refine the upper boundary condition set with IBC2 in Section 10.1.3.1. For any of the IBOUND values, IBC2 must equal two for IBOUND to be used. Otherwise, this flag is read in but not used by the software. If IBOUND = 1, the convective heat transfer coefficient at the top surface is defined by the input variable H2 in Section 10.1.3.2. This option is only useful for comparison with analytical heat transfer solutions and is not applicable to icing simulation. If IBOUND = 2, the heat transfer coefficient is defined using the integral boundary layer equations in LEWICE 3.2 or by a user read file (if IHTC = 1 in Section 8.3) but no ice accretion will occur. This option is useful for comparison with dry air data with heaters on. If IBOUND = 3, the heat transfer coefficient is defined using the integral boundary layer equations in LEWICE 3.2 and ice accretion will occur. IBOUND = 3 is recommended for any icing simulation. Valid input values are 1, 2, and 3. If a value outside this range is read, the following warning message will appear:

Boundary flag must be one (1), two (2) or three (3) Your input value of (value) is out of range. Setting IBOUND = 3

If IBOUND does not equal 3, the following reminder is generated:

Upper boundary is set to dry air. Ice will not form.

Note: IBOUND values are ignored by the software unless IBC2 = 2 boundary condition is used.

The third flag on this data line determines the initial temperature profile which should be used. If INIT = 1, all temperatures will be set equal to the ambient temperature. This option has primarily been used for comparison with analytical heat transfer cases. If INIT = 2, the temperature distribution in the wrap direction will be set to the recovery temperature. This option should be used when the body is initially in dry air. This occurs for dry air cases or for accretion cases where at least one heater is on at time = 0. INIT = 2 should be used for all cases where ICOND = 1. If INIT = 3, the temperature distribution in the wrap direction will be set to the accretion temperature (wet surface temperature). This option must be used if ICOND = 0 and IBOUND = 3. For any of these options, there is no initial temperature gradient in the body normal direction. Valid input values for INIT are 1, 2, and 3. If a value is read outside this range, the following warning message will be generated:

Initial temperature flag must be one (1), two (2) or three (3) Your input value of (value) is out of range. Setting INIT = 3

IF INIT = 1, the following reminder warning is generated:

Initial temperature selected is not normally used in icing simulations.

This warning will also be generated when INIT = 3 and ICOND = 1 since, if conduction is considered at the start of an accretion, the proper value for INIT is two.

Note: Most of the cases performed to date (especially those which have experimental data) use INIT = 2 and ICOND = 1. Other combinations have seen limited testing.

The fourth and fifth flags on this data line control ice shedding behavior for a de-icing system. The fourth flag, ISH, turns ice shedding on (ISH = 1) and off (ISH = 0) while the fifth flag, ISTD, determined the preferred type of ice shedding. ISTD values will be read but not used if ISH = 0.

Valid input values for ISH are 0 and 1. A value outside this range will generate the following warning message:

Ice shed flag must be zero (0) or one (1) Your input value of (value) is out of range. Setting ISH = 1

If ISH = 0 (no ice shed), the following warning message will be generated:

Ice will not shed. This is usually too conservative.

The values of ISTD have the following meaning: if ISTD = 0, ice will only shed if the software determines that the whole ice shape will shed (conservative); if ISTD = 1, ice will shed in a given section if the software determines that the ice in that section should shed; if ISTD = 2, ice can shed node by node if the software determines shedding of that node is appropriate (non-conservative). The recommended value is ISTD = 1 for the cases considered thus far. The other options are included as this result may not be universal since the number of test cases is small. Valid inputs for ISTD are 0, 1, and 2. If a value outside this range is read, the following warning message will be generated:

Shed type flag must be zero (0), one (1), or two (2) Your input value of (value) is out of range. Setting ISTD = 1

Additionally, if ISTD = 0 or 2, the following reminder warning is generated:

The preferred shedding mode is by sections (ISTD = 1).

The sixth and final flag on this data line, IGDE, is used to determine if phase change is considered. Phase change allows ice to melt or sublimate and allows water to evaporate. IGDE = 0 turns phase change off and should be used only when analyzing dry air cases. Additionally, the top layer properties entered in Section 10.1.1.2 should reflect the top layer of the de-icer or anti-icer construction. If IGDE = 1, phase change is considered and the top layer properties entered in Section 10.1.1.2 should reflect the properties for ice.

Note: If a node of ice melts, the software will automatically change the thermal properties of that node to water.

Note: If phase change is considered (IGDE = 1), the length (ELDE) given to the top layer in Section 10.1.1.2 will only be used to define the normal direction grid spacing in the ice layer. In this case, the ice thickness at time = 0 will be zero. Valid input values for IGDE are 0 and 1. If a value is read which is outside this range, the following warning message will appear:

Phase change flag must be zero (0) or one (1) Your input value of (value) is out of range. Setting IGDE = 1

If IGDE = 0, the following reminder warning will be generated:

Phase change is not selected. This is not normally used in icing simulations.

Some combinations of the flags in this data line will result in cases which are not physically meaningful. The software will check for these occurrences and take the following actions. If ice is the top layer (IGDE = 1) and conduction is initially off (ICOND = 0), then the upper boundary condition must be set so that ice accretion is considered (IBOUND = 3) and the initial

temperature distribution must be the accretion temperature (INIT = 3). If IGDE = 1 and ICOND = 0 but either IBOUND=3 or INIT=3, then the following warning message will be generated:

Since conduction flag is off (ICOND = 0), then the boundary condition flag (IBOUND) must be = 3 and temperatures must be initialized to the accretion temperature (INIT = 3). Resetting those flags.

Conversely, if ice is **not** the top layer (IGDE = 0) then ice accretion cannot occur and IBOUND = 3 or INIT = 3 are not allowed. If either of these flags has an incorrect value, the following warning will be generated:

Since phase change flag is off (IGDE = 0), then the boundary condition flag (IBOUND) must be = 2 and temperatures must be initialized to the recovery temperature (INIT = 2). Resetting those flags.

In addition, if ice is **not** the top layer (IGDE = 0), then ice shedding cannot occur. If IGDE = 0 and ISH = 1, the following warning message will be generated:

Ice cannot shed without phase change. Setting ISH = 0

Finally, if ice is the top layer (IGDE = 1), it cannot also be the source of an electrothermal heater. If IGDE = 1 and IJDE (from Section 10.1.2.1) equals the total number of layers (LDE from Section 10.1.1.1), then the following error message will be generated:

The ice layer cannot be the same as the heater layer. The software will stop because of the above error.

Table 27 contains a convenient cross reference showing the allowable combinations of flags. Each row contains the valid values for selected flag in the first column. For example, the first row shows the valid flags for that option. A cell containing the value “N/A” means “Not Applicable” because this flag is not used by the software for the other flag values set.

Note: Some combinations of flags in this table may generate warning messages but the flag values will not be changed

Table 27: Cross Reference for Allowable Flags

Flag	ICOND	IBOUND	INIT	ISH	ISTD	IGDE
ICOND = 0	0	1,2	1,2	0	N/A	0
ICOND = 0	0	3	3	0	N/A	1
ICOND = 0	0	3	3	1	0,1,2	1
ICOND = 1	1	1,2	1,2	0	N/A	0
ICOND = 1	1	3	1,2	0	N/A	1
ICOND = 1	1	3	1,2	1	0,1,2	1
IBOUND = 1	0	1	1,2	0	N/A	0
IBOUND = 1	1	1	1,2	0	N/A	0

IBOUND = 1	0	1	3	0	N/A	1
IBOUND = 1	0	1	3	1	0,1,2	1
IBOUND = 1	1	1	1,2	0	N/A	1
IBOUND = 1	1	1	1,2	1	0,1,2	1
IBOUND = 2	0	2	1,2	0	N/A	0
IBOUND = 2	1	2	1,2	0	N/A	0
IBOUND = 2	0	2	3	0	N/A	1
IBOUND = 2	0	2	3	1	0,1,2	1
IBOUND = 2	1	2	1,2	0	N/A	1
IBOUND = 2	1	2	1,2	1	0,1,2	1
IBOUND = 3	0	3	3	0	N/A	1
IBOUND = 3	0	3	3	1	0,1,2	1
IBOUND = 3	1	3	1,2	0	N/A	1
IBOUND = 3	1	3	1,2	1	0,1,2	1
INIT = 1	0,1	1,2	1	0	N/A	0
INIT = 1	0,1	3	1	0	N/A	1
INIT = 1	0,1	3	1	1	0,1,2	1
INIT = 2	0,1	1,2	2	0	N/A	0
INIT = 2	0,1	3	2	0	N/A	1
INIT = 2	0,1	3	2	1	0,1,2	1
INIT = 3	0	3	3	0	N/A	1
INIT = 3	0	3	3	1	0,1,2	1
ISH = 0	0,1	1,2	1,2	0	N/A	0
ISH = 0	1	3	1,2	0	N/A	1
ISH = 0	0	3	3	0	N/A	1
ISH = 1	0	3	3	1	0,1,2	1
ISH = 1	1	3	1,2	1	0,1,2	1
ISTD = 0	0	3	3	1	0	1
ISTD = 0	1	3	1,2	1	0	1
ISTD = 1	0	3	1,2	1	1	1
ISTD = 1	1	3	3	1	1	1

ISTD = 2	0	3	1,2	1	2	1
ISTD = 2	1	3	3	1	2	1
IGDE = 0	0,1	1,2	1,2	0	N/A	0
IGDE = 1	0	3	1,2	0	N/A	1
IGDE = 1	1	3	3	0	N/A	1
IGDE = 1	0	3	1,2	1	0,1,2	1
IGDE = 1	1	3	3	1	0,1,2	1

Table 28 shows an example of this data line with comments.

Table 28: Example input for the conduction and phase change flag input line.

```

icond  ibound  init  ish  istd  igde
1      3      2    1   1    1

```

10.1.4.8. *TSURF, IEVAP, I THERM*

This input data line contains variables used for the 1D anti-icing approximation (IDEICE = 1). If IDEICE = 1, the software will use the following variables from the de-icer input file to generate its solution: LDE (Section 10.1.1.1), ELDE and AK (Section 10.1.1.2), IJDE (Section 10.1.2.1), the upper and lower surface boundary conditions (Section 10.1.3) and the three variables listed in the title. The definitions for these variables (except for IEVAP which has an additional option) are the same in version 3.2 as they were in version 2.0. The input data format, however, conforms to the free-form format described earlier in the de-icer input file rather than the rigid format used in version 2.0.

The first value on this input data line, TSURF, is the temperature at the airfoil surface which the user wants to maintain. Typical running wet anti-icing systems operate in the region 278°K - 283°K while an evaporative system may operate at 320°K or even higher. The software 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 d 273.15 is entered, the following warning message is generated:

Desired surface temperature is less than freezing. Value read in was (value) K. Setting TSURF = 273.15 K

Note: The results for a desired surface temperature at freezing may be inaccurate since the 1D anti-icing module has limited provisions to freeze surface water as compared to the LEWICE icing module.

The second value on this input data line, IEVAP, indicates if the system is running wet (IEVAP = 0) or evaporative (IEVAP = 1). A third option (IEVAP = 2) will use the electrothermal wattages input in Section 10.1.2 if I THERM = 0 and the bleed air inputs if I THERM = 1. For I THERM = 1, bleed air input is needed via the heat flux input (IQAIN = 1),

heat transfer coefficient input (IBLEED = 1) or by correlation (ICORR = 1). The IEVAP = 2 option is convenient for direct comparison of the 1D option (IDEICE = 1) with the 2D option (IDEICE = 2, 3, or 4). Valid input values for IEVAP are 0, 1 or 2. If the value input is outside this range, the following warning message will be generated:

Evaporation flag must be zero (0), one (1), or two (2). Your value of (value) is out of range. Setting IEVAP = 0

Note: IEVAP = 2 can be used for any value of IDEICE. **When IEVAP = 2 and IDEICE = 1, the program will override the normal ice accretion process and calculate a residual ice shape with the 1D heat transfer assumption by using the heat input specified. If the amount of heat is high enough, no ice will form.** The IEVAP = 2 option uses the same external heat transfer coefficients as the IDEICE = 4 option, therefore a direct comparison can be made between the 1D and 2D options.

Finally, if an internal heat flux distribution is specified (IQAIN = 1), then the program may not have enough heat to fully evaporate the incoming water. Therefore, IEVAP = 1 cannot be used when both IITHERM = 1 and IQAIN = 1. The following warning message will be generated:

Cannot force evaporative condition when specifying internal heat flux or surface temperature. Setting IEVAP = 0.

Note: For high water loadings, the program may not be able to calculate the evaporative temperature. This occurs because the evaporative temperature is approaching the boiling point of water. In this case, the user can select IEVAP = 0 and TSURF = 370. If all of the water did not evaporate, then the IEVAP = 1 option should be avoided.

The third value on this input data line, IITHERM, indicates whether the anti-icer is electrothermal (IITHERM = 0) or uses hot air (IITHERM = 1). IITHERM will also control output of bleed air heat transfer coefficients as described in Sections 12.12 and 12.13. Valid input values for IITHERM are 0 and 1. If the value input is outside this range, the following warning message will be generated:

De-ice flag must be zero (0) or one (1). Your value of (value) is out of range. Setting IITHERM = 0

Additionally, the user must select a bleed air option when IITHERM = 1. This means that one of IBLEED = 1, IQAIN = 1, or ICORR = 1 must also be selected when using this option. If the value of IITHERM input is outside this range, the following warning message will be generated:

You must choose one of IBLEED=1, IQAIN=1, or ICORR=1 for hot air anti-icing simulations. Setting IITHERM = 0.

Table 29 contains an example input for this data line with comments.

Table 29: Example input for 1D anti-icing approximation (IDEICE = 1)

```
tsurf  ievap  itherm
278.d0  1      0
```


10.1.4.9. STA, RPM, IROT

The next line in this data input section defines three variables which are used to simulate rotational effects in the de-icing system. The software does not simulate a fully rotational system hence these options should be used with caution. The first variable, STA, defines the distance of the 2D cross-section being simulated from the hub of the rotating device (moment arm) in meters. This value is different from the value of YCASE input earlier. STA is used specifically to determine the rotational force for ice shedding while YCASE is used only in conjunction with piccolo tube applications. Valid inputs for STA are STA S 0. If a value which is outside this range is read, the following warning message will be generated:

Span location on rotor cannot be less than zero (0) Your value of (value) is out of range. Setting STA = 0.

The second variable on this data input line is the rotational speed, RPM, in revolutions per minute. Valid inputs for RPM are RPM S 0. If a value which is outside this range is read, the following warning message will be generated:

Rotor speed cannot be less than zero (0). Your value of (value) is out of range. Setting RPM = 0.

Note: To model a fixed-body problem, simply set RPM = 0.

The last variable on this data input line, IROT, contains a flag which defines the body as rotating vertically (propeller, IROT = 0) or horizontally (rotor blade, IROT = 1). Valid inputs for IROT are 0 and 1. If a value which is outside this range is read, the following warning message will be generated:

Rotor flag must be zero (0) or one (1). Your value of (value) is out of range. Setting IROT = 0

Note: The software uses the data input on this line for three calculations: 1) the rotational speed is used to calculate an increase in the aerodynamic heating term in the energy balance; 2) the rotational force is used to shed ice; and, 3) the rotational force is used to find the resultant force of the shed ice particle, which is used to track the particle after it sheds.

Note: The rotating body information is not used by the potential flow solver in LEWICE 3.2 nor is the rotating body information used by the trajectory equation.

Note: If STA = 0 or RPM = 0, then there is no difference between IROT = 0 and IROT = 1 (rotational force is zero).

Table 30 shows an example of this data line with comments.

Table 30: Example input for rotating body information.

```
sta      rpm      irot
10.d0    0.d0    1
```

10.1.4.10. DTAUI, NISP, DTAUM, NMSP, DTAUF, JCOUNTD

The sixth line of data input in this section defines six variables which are used to control the de-icer time step and convergence of the solution. The de-icer time step discussed in this section

is completely different from the icing time step defined in Section 8.3. In this section, the de-icer time step is used to calculate the transient (time-accurate) temperature distribution in the de-icer and the ice shape. In order to accurately capture these distributions, the de-icer time step must be much smaller than the icing time step. For example, a typical icing time step is 60 seconds while the standard de-icer time step is 0.1 seconds. This means that during a 60 second icing cycle, the de-icer routine will produce 600 temperature solutions, the last of which contains the local ice growth which is then used to produce an ice shape for the next icing time. The first variable on this line, DTAUI, defines the time step which is used at time = 0. This value must be 0.0001 d DTAUI d 10 and a typical value is 0.1 seconds. If the value read from the de-icer input file is outside this range, the following warning message will be generated:

Time steps less than 0.0001 or greater than 1.0 should not be used. Your value of (value) is out of range. Setting DTAUI = 0.1 sec Blank lines are read as data with a value of zero. Check input file for blank lines

Note: De-icer time steps less than 10^{-4} seconds are not allowed as the software will run very slowly at such de-icer time steps and de-icer time steps greater than 10 second cannot accurately capture temperature distributions even for cases where only the steady-state output (an anti-icing case) is simulated. This occurs because convergence cannot be maintained for larger de-icer time steps.

Note: DTAUI values other than 0.1 seconds have been used in test cases which have a large initial transient in the temperature distribution. Smaller initial time steps were chosen in these cases to accurately model the temperature distribution. Larger initial time steps were chosen for steady-state anti-icing cases. Note that since the phase of each node in the ice layer must be iterated upon, a larger time step does not always provide a faster solution.

The second variable on this data input line, NISP, defines the number of time steps for which DTAUI will be used as the time step. To use a 0.1 second time step for the first ten seconds, use NISP = 100. To use a 0.01 second time step for the first ten seconds, use NISP = 1000. NISP must be greater than zero. If a value is read which is outside this range, the following warning message will appear:

Number of time steps must be greater than zero (0). Your value of (value) is out of range. Setting NISP = 1000 Blank lines are read as data with a value of zero. Check input file for blank lines.

The third variable on this line, DTAUM, defines the time step for the period of time after DTAUI is used. This variable has a similar time range as DTAUI, namely 0.0001 d DTAUM d 1. De-icing time steps greater than 1 should only be used in anti-icing simulations where only one time step is needed. If a value is read which is outside this range, the following warning message will appear:

Time steps less than 0.0001 or greater than 1.0 should not be used. Your value of (value) is out of range. Setting DTAUM = 0.1 sec Blank lines are read as data with a value of zero. Check input file for blank lines.

The fourth variable on this line, NMSP, defines the number of time steps for which DTAUM will be used as the time step. Valid inputs for NMSP must be > 0 . If a value is read which is outside this range, the following warning message will appear:

Number of time steps must be greater than zero (0). Your value of (value) is out of range. Setting NMSP = 1000.

The fifth variable on this line, DTAUF, defines the time step for the remainder of the de-icing time, if any. This variable has the same range as DTAUM, namely 0.0001 d DTAUF d 1. If a value is read which is outside this range, the following warning message will appear:

Time steps less than 0.0001 or greater than 1.0 should not be used. Your value of (value) is out of range. Setting DTAUF = 0.1 sec Blank lines are read as data with a value of zero. Check input file for blank lines.

The last variable on this line, JCOUNT, defines the number of iterations for the solution technique and the phase change iteration. The temperature distribution at each time step is solved using the Modified Strongly Implicit Procedure (MSIP) which requires the use of some iteration. In addition, the software must predict the phase (rime, glaze, water) of each ice node and compare the solution to these predicted values. In most cases, JCOUNT = 10 has been found to be sufficient. The valid range for JCOUNT is 1 to 50. If a value is read which is outside this range, the following warning message will be generated:

Number of iterations must be greater than zero (0) and less than (value). Your value of (value) is out of range. Setting JCOUNTD = 10 Blank lines are read as data with a value of zero. Check input file for blank lines

Table 31 shows an example of this line of data input.

Table 31: Example input for time step data input line

```
dtaui  nisp  dtaum  nmisp  dtauf  jcountd
1.d-1  1000  1.d-1  1000  1.d-1  10
```

10.1.4.11. TSPRAY, TAFTER

The next line of this data input section contains two variables which control the on and off times of the heater relative to the immersion in the cloud. The first variable, TSPRAY, is defined as the time the heaters are on before water hits the surface. In a simulation of the real environment, this translates to the heaters being turned on before entering a cloud. In the tunnel, the heaters could also turn on before the water spray. TSPRAY = 0 has been used in the large majority of simulations to date. This value must be ≥ 0 . If a value is read which is outside this range, the following warning message is generated:

Time before spray cannot be less than zero (0). Your value of (value) is out of range. Setting TSPRAY = 0

Since TSPRAY = 0 has been used for the majority of simulations, the following reminder warning will be generated if TSPRAY > 0:

Time before spray is greater than zero (0). Water impingement will not start until TSPRAY.

The second variable on this data input line, TAFTER, is defined as the time the heaters are on after water stops hitting the surface. In a simulation of the real environment, it is the period of time in which the de-icers are still on after a cloud has been exited. TAFTER = 0 has been used

in the large majority of simulations to date. This value must be ≤ 0 . If a value is read which is outside this range, the following warning message is generated:

Time after spray cannot be less than zero (0).

Your value of (value) is out of range. Setting TAFTER = 0.

Since TAFTER = 0 has been used for the majority of simulations, the following reminder warning will be generated if TAFTER > 0:

Time after spray is greater than zero (0).

Water impingement will end and de-icing will continue for (value) seconds.

Table 32 shows an example of this data input line.

Table 32: Example data for spray time data input line

```
tspray  tafter
0.d0    0.d0
```

10.1.4.12. ISCOLA, JSCOLA, KSCOLA, SSLOPA, SZEROA

The eighth line of this data input section contains variables used when reading bleed air heat transfer coefficients (i.e. when IBLEED = 1). The data on this line is read regardless of the value of IBLEED however. The data line contains five variables which define the format of the bleed air data. The first three numbers (ISCOLA, JSCOLA, and KSCOLA) define the column numbers in the bleed air input data file which contain the requested values. ISCOLA defines which column contains the wrap distance. JSCOLA defines the column which contains the bleed air heat transfer coefficients in $W/m^2/K$, and KSCOLA defines the total number of columns in the file. A file which contained only the necessary data would therefore have ISCOLA = 1, JSCOLA = 2, and KSCOLA = 2. (ISCOLA = 2, JSCOLA = 1, and KSCOLA = 2 would also be valid.) This type of data input format was used since the data probably came from a separate program and this format style reduces the amount of reformatting needed for LEWICE 3.2. Section 10.2 contains further information on the format of the input data file “**hi.inp**” which contains the bleed air heat transfer coefficient values. ISCOLA must be at least 1 and no greater than KSCOLA. If a value is read which is outside this range, the following warning message will be generated:

Columns for 1st variable must be at least one (1) and no greater than the number of columns. Your value of (value) is out of range. Setting ISCOLA = 1 Blank lines are read as data with a value of zero. Check input file for blank lines.

Similarly, JSCOLA must be at least 1 and no greater than KSCOLA. It also cannot be the same as ISCOLA. If a value is read which is outside this range, the following warning message will be generated:

Columns for 2nd variable must be at least one (1) and no greater than the number of columns. Your value of (value) is out of range. Setting JSCOLA = 1 Blank lines are read as data with a value of zero. Check input file for blank lines.

If the values for ISCOLA and JSCOLA are the same, the following warning message will be generated:

Columns for 1st and 2nd variable cannot be the same. Setting ISCOLA = 1 and JSCOLA = KSCOLA

The total number of columns, KSCOLA, must be at least 2, but can be higher. If a value is read which is outside this range, the following warning message will be generated:

Number of columns must be at least two (2) Your value of (value) is out of range. Setting KSCOLA = 2

The fourth value on this data input line, SSLOPA, defines the conversion factor which will be applied to the wrap distance values input. LEWICE 3.2 uses wrap distances in dimensionless values (value divided by chord). The wrap distance values input can be in any set of units if the user supplies the correct conversion to SSLOPA. The slope can be positive or negative, but not zero. If a value of zero is read for this data input, the following warning message will be generated:

Slope cannot be zero. Blank lines are read as data with a value of zero. Check input file for blank lines.

The last value on this data input line, SZEROA, defines the offset of the input wrap distances from those needed for LEWICE 3.2. The software will assume that a wrap distance value of 0 in the input data corresponds to the lower surface of the trailing edge. If this is not the case, the user should supply a value (in dimensionless distance) which supplies the correct offset. The value for SZEROA should not be greater than the wrap distance from the leading edge to the trailing edge.

The following equation is used within LEWICE to convert wrap distance values using SSLOPA and SZEROA:

$$s_{LEWICE} = s_{input}sslopa + szeroa \quad (14)$$

where s_{LEWICE} is the wrap distance used by LEWICE and s_{input} is the wrap distance in the input file. If the value read is greater than 1.2 (1.2 chord lengths), the following warning message will be generated:

Offset may be past the trailing edge. Setting SZERO = 1.

Table 33 shows an example of this data input line.

Table 33: Example input for hot air data file characteristics

```
input parameters for reading in hot air HTC info
iscola  jscola  kscola  sslopa  szeroa
      2      3      3      1.d0    0.d0
```

10.1.4.13. ISCOLQ, JSCOLQ, KSCOLQ, SSLOPQ, SZEROQ

This input data line consists of the same five variables as the previous input line, except the variables on this line pertain to reading in heat flux data (IQAIN = 1). That is, ISCOLQ contains the column number containing the wrap distances for heat flux data input, JSCOLQ contains the

column number containing the heat fluxes in kW/m², and KSCOLQ contains the total number of data columns in the input data file. Similarly, SSLOPQ and SZEROQ are the conversion factor for the wrap distances and the offset, respectively. The limits on input and warning messages generated are the same as those listed in Section 10.1.4.8 and therefore will not be repeated here. The relationship between the wrap distances read in Section 10.3 and the wrap distances within LEWICE is also given by the equation in the previous section. Section 10.3 contains further information on the format of the input data file “**qain.inp**” which contains the bleed air heat flux values. Table 34 shows an example of this data input line.

Note: Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K respectively.

Table 34: Example input for wall heat flux data file characteristics

```
input parameters for reading in hot air q_wall info
iscolq   jscolq   kscolq   sslopq   szeroq
    1         2         2       1.d0     0.d0
```

10.1.4.14. *ISCOLT, JSCOLT, KSCOLT, SSLOPT, SZEROT*

This input data line consists of the same five variables as the previous input line, except the variables on this line pertain to reading in surface temperature data (ITSUR = 1). That is, ISCOLT contains the column number containing the wrap distances for surface temperature data input, JSCOLT contains the column number containing the surface temperatures in Kelvin, and KSCOLT contains the total number of data columns in the input data file. Similarly, SSLOPT and SZEROT are the conversion factor for the wrap distances and the offset, respectively. The limits on input and warning messages generated are the same as those listed in Section 10.1.4.12 and therefore will not be repeated here. The relationship between the wrap distances read in Section 10.10 and the wrap distances within LEWICE is also given by the equation in Section 10.1.4.12. Section 10.10 contains further information on the format of the input data file “**tsurf.inp**” which contains the surface temperature values. The table below shows an example of this data input line.

Table 35: Example input for surface temperature data file characteristics

```
input parameters for reading in hot air surf. temperature info
iscolt   jscolt   kscolt   sslopt   szerot
    1         2         2       1.d0     0.d0
```

10.1.4.15. *ISCOLS, JSCOLS, KSCOLS, SSLOPS, SZEROS*

This input data line consists of the same five variables as the previous input lines, except the variables on this line pertain to reading in 3D streamline data (I3D = 1). That is, ISCOLS contains the column number containing the wrap distances for 3D streamline data input, JSCOLS contains the column number containing the 3D streamline ratios, and KSCOLS contains the total number of data columns in the input data file. Similarly, SSLOPS and SZEROS are the conversion factors for the wrap distances and the offset, respectively. The limits on input and warning messages generated are the same as those listed in Section 10.1.4.12 and therefore will

not be repeated here. The relationship between the wrap distances read in Section 10.4 and the wrap distances within LEWICE is also given by the equation in Section 10.1.4.12. Section 10.4 contains further information on the format of the input data file “**stream.inp**” which contains the 3D streamline values. Table 36 shows an example of this data input line.

Table 36: Example input for 3D streamline data file characteristics

```
input parameters for reading in 3D streamline info
iscols   jscols   kscols   sslops   szeros
  1         2       2       1.d0    0.d0
```

10.1.4.16. Complete input section example

This completes the descriptions for the fourth input data section. Table 38 shows an example with all of the input data read in this section.

Table 37: Complete input example for fourth data section

```
C                               miscellaneous data
C
tairh  airmd   airup   xairmd
  5.d2  0.1d0   0.5d0   0.d0
nhole  dhole   dtube   rdist   ycase   xuhnd   xlhnd
  3     0.00132 0.0381  50.     0.0     0.23   0.26
C
icase  hconst  rep     rp      zp     rrp    dp
  0     1.0   0.76   1.394  1.     0.     0.
C
xhole(m)  yhole(m)  zdist(m)
.         0.0176   -0.033   0.0107
  0.000   0.000    0.0091
  0.0176  -0.033   0.0076
C
i3d   iqain  ibleed  iqw  itsur
  0    0     0      1   0
C
icond  ibound  init   ish  istd  igde
  1     3     2     1   1    1
C
tsurf  ievap  itherm
278.d0  1     0
C
sta    rpm   irot
```

```

10.d0  0.d0  1
C
dtaui  nisp  dtaum  nmsp  dtauf  jcountd
1.d-1  1000  1.d-1  1000  1.d-1  10
C
tspray  tafter
0.d0  0.d0
C
input parameters for reading in hot air HTC info
iscola  jscola  kscola  sslopa  szeroa
  2      3      3      1.d0   0.d0
input parameters for reading in hot air q_wall info
iscolq  jscolq  kscolq  sslopq  szeroq
  1      2      2      1.d0   0.d0
input parameters for reading in hot air surf. temperature info
iscolt  jscolt  kscolt  sslopt  szerot
  1      2      2      1.d0   0.d0
input parameters for reading in 3D streamline info
iscols  jscols  kscols  sslops  szeros
  1      2      2      1.d0   0.d0

```

10.1.5. De-icer Section 5

The last section of the de-icer input file defines parameters which control how temperatures will be output from LEWICE 3.2. There are two main styles of temperature output from the software. The first style will output temperatures in the entire domain using a specified file format. This style is suitable for importing results into post-processor software. The second style outputs temperatures at selected locations in space and time in column format consistent with the output style used for the other output files generated by LEWICE 3.2.

Note: If there was an error reading data in a previous section, the software will continue to read the rest of the input file. However, an error in an earlier section can cause incorrect error messages to appear for subsequent sections. It is highly recommended that the user fix the first errors encountered and resubmit the case.

10.1.5.1. NPRT, NPRF

The first line of data input in this section defines how many post-processor files the user wants and a flag defining the file format to be used. The first variable, NPRT, defines the number of times the software will output two very large files: one containing the grid for the de-icer and ice shape; the other containing the 2D solution file. This number must be 0 ≤ NPRT ≤ 50. If a value is input which is outside this range, the following warning message will be generated:

Number of output files must be between zero (0) and 50.

Your value of (value) is out of range. Setting NPRT = 0

The second variable on this line, NPRF, defines the output format for these files. If NPRF = 0, the files will be output in PLOT3D format. This is a standard output file format for post-processing software on Silicon Graphics and other UNIX platforms. If NPRF = 1, the files will be output in TECPLOT format. TECPLOT is a widely used post-processing software on the PC. Users who wish to have additional output file formats should contact the NASA Glenn Icing Branch. Valid input values for NPRF are 0 and 1. If a value is input which is outside this range, the following warning message will be generated:

Output file format flag must be zero (0) or one (1).

Your value of (value) is out of range. Setting NPRF = 0.

10.1.5.2. ITIME

The following NPRT number of lines contains the times at which this style output will be generated. Values should be input as integers (no decimal points) and cannot be negative values. If a value is input which is outside this range, the following warning message will be generated:

Printout time number (value) must be greater than or equal to zero (0).

Your value of (value) is out of range. Setting ITIME = 1000.

Table 38 contains an example of the first series of data input for this section.

Table 38: Example input for PLOT3D style output

```
nprt  nprf
10    0
itime values (seconds)
10
20
30
40
50
60
70
80
90
100
```

Note: Output times do not have to be in equal time increments as is shown in the example input.

Note: Previous versions of the LEWICE/Thermal software used time step instead of time.

10.1.5.3. NTYPE

The next line of data input defines the number of output types the user wants. There are five output types which can be generated by the software. These are defined as follows: 1)

temperature is output as a function of time at user-selected x,y locations (IOTYPE = 1); 2) temperature is output as a function of wrap distance at user-selected times and layer locations (IOTYPE = 2); 3) temperature is output as a function of the normal distance at user-selected times and wrap distance locations (IOTYPE = 3); 4) all temperatures are output in ASCII format (IOTYPE = 4); and, 5) temperatures at every node are output at user-specified times (IOTYPE = 5). This line of data input selects the number of output types desired. The user should select NTYPE = 1 if only one of these output types is needed. Valid inputs for NTYPE are 1 to 4. If a value is input which is outside this range, the following warning message will be generated:

Print type must be between one (1) and four (4).

Your value of (value) is out of range. Setting NTYPE = 1

Note: The last two output types (IOTYPE = 4 or 5) are intended for debugging purposes and should not normally be needed by the user.

10.1.5.4. IOTYPE, NPTSD

The next line of data input contains the type of output being requested (IOTYPE) and the number of points which the user wants output (NPTSD). The output type, IOTYPE, defines which of the five outputs will be generated. IOTYPE = 1 provides the same type of output as a thermocouple, namely it outputs temperature at a specific location as a function of time. IOTYPE = 2 outputs temperature as a function of wrap distance at specified times and given layers. For example, it can provide the surface temperature as a function of wrap distance at specified times and/or it can output heater temperatures as a function of wrap distance. IOTYPE = 3 outputs temperature as a function of the normal direction at specified wrap distance locations and times. For example, the user could output the temperature at the parting strip for selected times as a function of the normal distance through the heater. This output type may be useful to look for “hot spots” in a heater section. IOTYPE = 4 outputs the temperatures at every node at all time steps and should only be used for debugging purposes. Similarly, IOTYPE = 5 outputs the temperature at every node, but only at user-selected times. Valid inputs for IOTYPE are 1 to 5. If a value is input which is outside this range, the following warning message will be generated:

Printout type must be between one (1) and five (5).

Your value of (value) is out of range. Setting IOTYPE = 1

The next variable on this line, NPTSD, defines the number of points the user wants to output for type IOTYPE. Valid inputs for NPTSD are 0 d NPTSD d 50. If a value is input which is outside this range, the following warning message will be generated:

Number of printouts must be between zero (0) and 50.

Your value of (value) is out of range. Setting NPTSD = 0

10.1.5.5. ILAYER, IWHERE, JLAYER, JWHERE

The user must input NTYPE number of lines which contain the values for IOTYPE and NPTSD. These data lines are not input sequentially, however. After each data line containing values of IOTYPE and NPTSD, there must be NPTSD number of lines containing the output locations for that data type. These locations are read in as the variables ILAYER, IWHERE,

JLAYER and JWHERE. Each data type has a different input format type, each of which will be described in this subsection.

For IOTYPE = 1, the user must input NPTSD number of lines, each of which contains four variables. Rather than input node numbers or distances which may be unknown, the input has been set up to have the user input the layer and section numbers where temperature is to be output. The first variable, ILAYER, defines the section in the wrap direction where the user wants temperature output. The second variable, IWHERE, tells the software to take the left (1), middle (2) or right (3) side of the section. The third variable, JLAYER, defines the layer number where the user wants temperature output and the last variable, JWHERE, tells the software to take the bottom (1), middle (2), or top (3) side of the layer. This means of selecting temperature output is diagramed in Figure 15. In this figure, a sample grid block is shown with 15 nodes in the wrap direction and five in the normal direction. The open circles denote the nodes in the grid, while the solid circles represent the locations in this block which can be output using the IOTYPE = 1 designation.

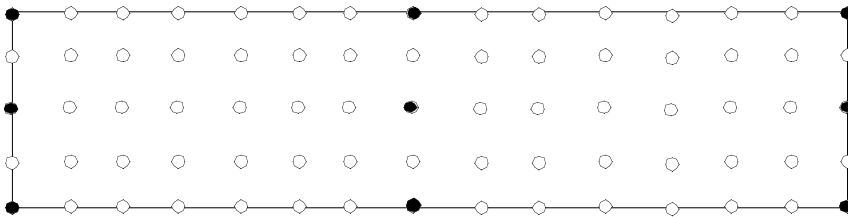


Figure 15: Potential IOTYPE = 1 (thermocouple) locations in a grid block

For IOTYPE = 1, valid inputs for ILAYER are from 1 to the number of sections, NXDE (Section 10.1.1.1). If a value is input which is outside this range, the following warning message will be generated:

Printout section must be between one (1) and the number of sections. Your value of (value) is out of range. Setting ILAYER = NXDE.

For IOTYPE = 1, valid inputs for JLAYER are from 1 to the number of layers, LDE (Section 10.1.1.1). If a value is input which is outside this range, the following warning message will be generated:

Printout layer must be between one (1) and the number of layers. Your value of (value) is out of range. Setting ILAYER = LDE.

Valid inputs for IWHERE are from 1 to 3. If a value is input which is outside this range, the following warning message will be generated:

Printout flag must be between one (1) and three (3) Your value of (value) is out of range. Setting IWHERE = 2

Similarly, valid inputs for JWHERE are from 1 to 3. If a value is input which is outside this range, the following warning message will be generated:

Printout flag must be between one (1) and three (3) Your value of (value) is out of range. Setting JWHERE = 2

For IOTYPE = 2, the user must input NPTSD number of lines which contain the time and location of temperatures for this output type. In this case, however, the user inputs only three values for each line: ILAYER, JLAYER, and JWHERE. For IOTYPE = 2, ILAYER contains the times at which output is to take place. JLAYER and JWHERE have the same definition as for IOTYPE = 1. For IOTYPE = 2, ILAYER need only be S 0. If a value is input which is outside this range, the following warning message will be generated:

Printout time cannot be negative. Your value of (value) is out of range. Setting ILAYER = 1000

For IOTYPE = 3, the user must input NPTSD number of lines which contain the time and location of temperatures for this output type. In this case, however, the user inputs only three values for each line: ILAYER, JLAYER, and JWHERE. For IOTYPE = 3, ILAYER contains the times at which output is to take place. JLAYER and JWHERE now contain the section number and location where temperature output will occur for this output type. Valid inputs are: ILAYER S 0, 1 d JLAYER d NXDE, 1 d JWHERE d 3. If a value is input which is outside this range, the warnings previously given will generated for the proper variable.

For IOTYPE = 4, the user does not input any additional lines of data, since no descriptors are needed. For IOTYPE = 5, the user must input NPTSD number of lines which contain the time at which temperatures will be output. In this case, the user inputs one variable on each line, namely ILAYER. Valid inputs for ILAYER are S 0 and the previously listed warning message will be generated if the data input is outside this range. Finally, Table 39 shows an example of the data input in this subsection.

Table 39: Example input for temperature output flags

```

C      ntype = number of types of output
C
ntype
3
C
C      iotype = type of output requested
C      nptsd = number of points to be output
C
iotype  nptsd
1      20
C      section where      layer  where  (20 data lines to follow)
          5      2          6      3
          5      2          7      3
          4      2          6      3
          4      2          7      3
          6      2          6      3
          6      2          7      3
          3      2          6      3

```

3	2	7	3
7	2	6	3
7	2	7	3
5	2	4	1
4	2	4	1
6	2	4	1
3	2	4	1
7	2	4	1
5	2	1	1
4	2	1	1
6	2	1	1
3	2	1	1
7	2	1	1

C

C iotype = type of output requested

C nptsd = number of points to be output

C

iotype nptsd

2 10

C

C time layer where

C (sec) (10 data lines to follow)

10 6 1

20 6 1

30 6 1

40 6 1

50 6 1

60 6 1

70 6 1

80 6 1

90 6 1

100 6 1

C

C iotype = type of output requested

C nptsd = number of points to be output

C

iotype nptsd

3 10

```

C      time      section  where
C      (sec)     (10 data lines to follow)
      10         5         2
      20         5         2
      30         5         2
      40         5         2
      50         5         2
      60         5         2
      70         5         2
      80         5         2
      90         5         2
     100         5         2

```

10.1.6. Complete example for de-icer input file

This completes the description of the input for the de-icer input file. Table 40 shows a complete example of an input file with comments.

Table 40: Example of complete de-icer input file

```

C      data for the composite body

```

```

C
C      lde= total number of layers in the normal direction
C      nxde= total number of heater sections in the wrap direction
C
lde   nxde
007   009
C
C      Data for each layer:
C
C      nodes = # of points in that layer or section
C      length = length (thickness) of layer or section
C
# of  length  conductivity  diffusivity  anisotropy  slope (b) of
nodes (m)      (W/m/K)      (m^2/s)      factor      temp. eqn.
C      substrate
15    3.430d-3  0.120d0      1.652d-7    1.d0        0.d0
C      insulation
08    8.900d-4  0.294d0      1.045d-7    1.d0        0.d0
C      insulation
18    2.800d-4  0.256d0      1.473d-7    1.d0        0.d0

```

```

C heater
07 1.300d-5 41.000d0 1.194d-5 1.d0 0.d0
C insulation
18 2.800d-4 0.256d0 1.473d-7 1.d0 0.d0
C abrasion shield
08 2.030d-4 16.270d0 4.035d-6 1.d0 0.d0
C ice
21 2.540d-3 2.232d0 1.151d-6 1.d0 0.d0

```

```

C
C Data for each heater/gap section:
C

```

# of nodes	length (m)	cond. (W/m/K)	diff. (m ² /s)	anisotropy of heater	slope(b) of temp. eqn.	add. length (m)	layer number
C gap (no heat input)							
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
C heater G							
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C heater E							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C heater C							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C parting strip - heater A							
36	1.905d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C heater B							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C heater D							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C heater F							
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C gap (no heat input)							
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3

```

C data for the heater

```

```

C
ijde offset ipar
004 0.005 1
      "QDE"      "TON"      "TOFF"      "TLAG"      "ICFLAG"
heater density time on time off lag time temp cntrl.

```

	(kWatts/m ²)	(sec)	(sec)	(sec)	flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	12.02d0	5.d0	115.d0	115.d0	0
C	heater E				
	11.19d0	5.d0	115.d0	115.d0	0
C	heater C				
	11.25d0	5.d0	115.d0	110.d0	0
C	parting strip (heater A) does not turn off				
	07.78d0	150.d0	0.d0	0.d0	0
C	heater B				
	11.68d0	5.d0	115.d0	110.d0	0
C	heater D				
	11.27d0	5.d0	115.d0	115.d0	0
C	heater F				
	11.75d0	5.d0	115.d0	115.d0	0
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0

C boundary condition data

C

C Type of boundary condition:

C

"ibc1" "ibc2" "ibc3" "ibc4"

2 2 2 2

C

C Temperature data for bottom, left, and side boundaries

C

"tg1" "tg2" "tg3" "tg4"

263.3 263.3 263.3 263.3

C

C Heat transfer coeff. in W/m²/K

C

"h1" "h2" "h3" "h4"

3.d0 3.d2 0.d0 0.d0

C

C Wall heat flux in kW/m²

C

"qw1" "qw2" "qw3" "qw4"
0.d0 0.d0 0.d0 0.d0

C miscellaneous data

C

tairh airmd airup xairmd
5.d2 0.1d0 0.5d0 0.d0
nhole dhole dtube rdist ycase xuhnd xlhnd
3 0.00132 0.0381 50. 0.0 0.23 0.26

C

icase hconst rep rp zp rrp dp
0 1.0 0.76 1.394 1. 0. 0.

C

xhole(m) yhole(m) zdist(m)
. 0.0176 -0.033 0.0107
0.000 0.000 0.0091
0.0176 -0.033 0.0076

C

i3d iqain ibleed iqw itsur
0 0 0 1 0

C

icond ibound init ish istd igde
1 3 2 1 1 1

C

tsurf ievap itherm
278.d0 1 0

C

sta rpm irot
10.d0 0.d0 1

C

dtaui nisp dtaum nmsp dtauf jcountd
1.d-1 1000 1.d-1 1000 1.d-1 10

C

tspray tafter
0.d0 0.d0

C

input parameters for reading in hot air HTC info

iscola jscola kscola sslopa szeroa
2 3 3 1.d0 0.d0

input parameters for reading in hot air q_{wall} info

iscolq	jscolq	kscolq	sslopq	szeroq
1	2	2	1.d0	0.d0

input parameters for reading in hot air surf. temperature info

iscolt	jscolt	kscolt	sslopt	szerot
1	2	2	1.d0	0.d0

input parameters for reading in 3D streamline info

iscols	jscols	kscols	sslops	szeros
1	2	2	1.d0	0.d0

C input/output parameters

nprt nprf

10 1

itime values (seconds)

10

20

30

40

50

60

70

80

90

100

C n_{type} = number of types of output

C

n_{type}

3

C

C i_{otype} = type of output requested

C n_{ptsd} = number of points to be output

C

i_{otype} n_{ptsd}

1 20

C section where layer where (20 data lines to follow)

 5 2 6 3

 5 2 7 3

 4 2 6 3

 4 2 7 3

6	2	6	3
6	2	7	3
3	2	6	3
3	2	7	3
7	2	6	3
7	2	7	3
5	2	4	1
4	2	4	1
6	2	4	1
3	2	4	1
7	2	4	1
5	2	1	1
4	2	1	1
6	2	1	1
3	2	1	1
7	2	1	1

C

C iotype = type of output requested

C nptsd = number of points to be output

C

iotype nptsd

2 10

C

C time layer where

C (sec) (10 data lines to follow)

10 6 1

20 6 1

30 6 1

40 6 1

50 6 1

60 6 1

70 6 1

80 6 1

90 6 1

100 6 1

C

C iotype = type of output requested

C nptsd = number of points to be output

```

C
iotype  nptsd
  3      10
C      time      section  where
C      (sec)     (10 data lines to follow)
      10         5         2
      20         5         2
      30         5         2
      40         5         2
      50         5         2
      60         5         2
      70         5         2
      80         5         2
      90         5         2
     100         5         2

```

10.2. Hot Air Heat Transfer Coefficient Input File (hi.inp)

LEWICE 3.2 has two methods for reading in data to be used in analyzing hot air anti-icing performance. In this first approach, if IBLEED = 1 in the de-icer input file as specified in Section 10.1.4.6, then the program will read in a file named “**hi.inp**” which contains the values of the hot air heat transfer coefficients in W/m²K. Alternatively, the user can set ICORR = 1 or 2 to activate the correlation built into LEWICE. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the hot air heat transfer coefficients. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 10.1.4.2 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The heat transfer coefficients must be in units of W/m²K. Other options on this input file are also discussed in Section 10.1.4.2.

The number of data points must be at least 2 and cannot be greater than 10000 (10⁴). If the value read is outside this range, the following error message will be generated:

Number of hot air values input is out of range. Fix hi.inp input and resubmit. NEL = (value) Program will stop because of this error

If one or more of the wrap distance values do not reside on the body geometry, the following warning message will be generated:

Wrap distance value input is out of range.srj(i) = (value) at location i = (value) limit = (value)

The variables output are the wrap distance input (after being translated using SSLOPA and SZEROA), the index of the wrap distance, and the wrap distance limit which was exceeded. For example if the first wrap distance value is negative after translation, then this warning will appear since the lower wrap distance limit is zero.

Note: It is normal for there to be a slight difference in wrap distance values between two different programs. Small differences may be ignored.

Note: The user should input values for the entire wrap distance region. The program needs values for the entire body geometry and extrapolation from the values provided can yield poor results. Specifically, the values outside the region input are not set equal to zero.

After the file has been processed, if any warning messages are issued the following confirmation will appear:

(value) warnings have been issued when reading HI values.

Do you wish to continue (Y / N)?

The user must input the character ‘y’ or ‘Y’ to continue the run.

If any error messages were generated, the following message will appear:

(value) errors have been found when reading HI values.

Program will stop because of this error.

The user must fix the errors and resubmit the case.

10.3. Hot Air Heat Flux Input File (qain.inp)

The second method of reading in data for a hot air anti-icer can be accessed by specifying IQAIN = 1 in the de-icer input file as described in Section 10.1.4.6. In this case, the program will read in a file named “**qain.inp**” which contains the values of the hot air heat fluxes. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the heat fluxes. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 10.1.4.13 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The heat fluxes must be in units of kW/m². Other options on this input file are also discussed in Section 10.1.4.13.

The number of data points must be at least 2 and cannot be greater than 10000 (10⁴). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

10.4. 3D Streamline Input file (stream.inp)

LEWICE 3.2 can be used to analyze a three dimensional geometry using a pseudo-3D approach. The user inputs a 2D cross-section in the geometry input file and then specifies in a

file named “**stream.inp**” the dimensionless distance between two cross-sections at each wrap distance location. This is diagrammed in Figure 16.

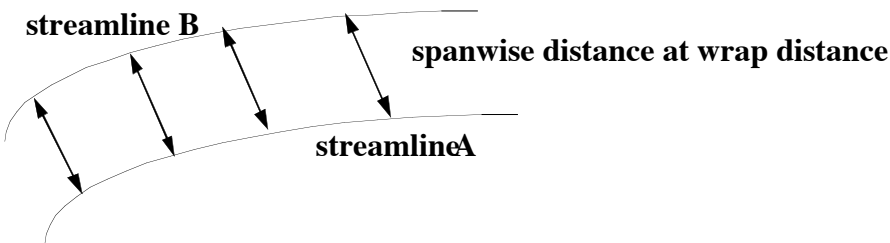


Figure 16: Representation of 3D streamline data

This input file will be read when $I3D = 1$ as described in Section 10.1.4.6. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the spanwise distances. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 10.1.4.15 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The spanwise distances must be nondimensionalized by the chord length. Other options on this input file are also discussed in Section 10.1.4.15.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

10.5. Grid Input Files (XY.PLT and Q.PLT)

LEWICE 3.2 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. LEWICE 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 PLOT3D²⁴ format with iblanking. The input data files must contain REAL*4 binary data. The read statements needed to create the files are as follows:

10.5.1. 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 ((iblack(i, j, mqf), i=1, im), j=1, jm)
```

10.5.2. 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 3.2 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 freestream 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.

The variables conform to the standard PLOT3D with multiple grids with IBLANKing. The first grid variable, MG, contains the number of grid blocks being read. The second set of grid variables, IMNS and JMNS, contain the number of grid points in each direction of the grid block. The variables XG and YG contain the grid coordinates and the variable IBLANK is an integer value which describes details about that grid node. Within LEWICE, IBLANK = 1 activates the grid node and IBLANK = 0 deactivates the node.

The flow solution file also contains the number of grid blocks and the grid dimensions. The following line contains ambient Mach number, angle of attack, Reynolds number, and time in seconds. These values are not used by LEWICE. However, if these values are different from the corresponding values read or calculated from the LEWICE main input file then the answers obtained from the analysis may not be correct. The four flow solution variables are defined as follows:

$$q1 = \frac{\rho}{\rho_{\infty}} \quad (15)$$

$$q2 = \frac{\rho}{\rho_{\infty}} \frac{u}{a} = \frac{\rho}{\rho_{\infty}} \frac{u}{V_{\infty}} M_{\infty} \quad (16)$$

$$q3 = \frac{\rho}{\rho_{\infty}} \frac{v}{a} = \frac{\rho}{\rho_{\infty}} \frac{u}{V_{\infty}} M_{\infty} \quad (17)$$

$$q^4 = \frac{P}{P_\infty} \frac{1}{\gamma - 1} + \frac{1}{2} \frac{\rho}{\rho_\infty} \left(\left(\frac{u}{a} \right)^2 + \left(\frac{v}{a} \right)^2 \right) \quad (18)$$

10.6. Pressure Coefficient Input File (**rflow.inp**)

LEWICE 3.2 will read in this optional input file if $ICP = 1$ in the main input file, as specified in Section 8.3.12. In this case, the program will read in a file named “**rflow.inp**” which contains the values of the pressure coefficients and bypass the calculation of this value in the software. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the pressure coefficients. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 8.3.12 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The pressure coefficients are by definition dimensionless. Other options on this input file are also discussed in Section 8.3.12.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

Note: As stated in Section 8.3.12, surface pressures are insufficient for calculating droplet trajectories. The user must include either a grid flow solution (which makes the “**rflow.inp**” file redundant) or supply collection efficiencies ($IBETA = 1$) to use this option.

10.7. Collection Efficiency Input File (**rbeta.inp**)

LEWICE 3.2 will read in this optional input file if $IBETA = 1$ in the main input file as specified in Section 8.3.13. In this case, the program will read in a file named “**rbeta.inp**” which contains the values of the collection efficiencies and bypass the calculation of this value in the software. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the collection efficiencies. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 8.3.13 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The collection efficiencies are by definition dimensionless. Other options on this input file are also discussed in Section 8.3.13.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

10.8. External Heat Transfer Coefficient Input File (rhtc.inp)

LEWICE 3.2 will read in this optional input file if $IHTC = 1$ in the main input file as specified in Section 8.3.14. In this case, the program will read in a file named “**rhtc.inp**” which contains the values of the external heat transfer coefficients and bypass the calculation of this value in the software. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the external heat transfer coefficients. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 8.3.14 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The external heat transfer coefficients are expected to be in $W/m^2/K$. Other options on this input file are also discussed in Section 8.3.14.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

10.9. External Heat Flux Input File (qextin.inp)

LEWICE 3.2 will read in this optional input file if $IQEX = 1$ in the main input file as specified in Section 8.3.15, then the program will read in a file named “**qextin.inp**” which contains the values of the external heat fluxes and bypass the calculation of the external heat transfer coefficient in the software. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the external heat fluxes. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 8.3.15 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The external heat fluxes are by expected to be in kW/m^2 . Other options on this input file are also discussed in Section 8.3.15.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

Note: Heat flux values are input (and output) in units of kW/m^2 while thermal conductivity and heat transfer coefficients are input (and output) in units of $W/m/K$ and $W/m^2/K$ respectively.

10.10. Surface Temperature Input File (tsurf.inp)

LEWICE 3.2 will read in this optional input file if $ITSUR = 1$ in the de-icer input file as specified in Section 10.1.4.6, then the program will read in a file named “**tsurf.inp**” which contains the values of the temperature distribution on the outside surface. The program will then

calculate the heat flux needed to achieve this temperature distribution. The heat flux can be supplied by either bleed air or electrothermal means. The procedure used for this option is identical to the procedures described in Sections 10.1.2 and 10.1.4.6. The first line of this data file contains the number of points to be read. Subsequent lines contain at least two columns of data: one column which contains the wrap distance locations and a second column containing the surface temperature. The wrap distance values are expected to be nondimensionalized by chord with the zero wrap distance value at the lower surface trailing edge of the body, with values proceeding clockwise from that location. These defaults can be changed as described in Section 10.1.4 except that the data must be input clockwise starting from the lower surface trailing edge. Since only wrap distance locations are read in, clockwise input cannot be checked for by the software. The surface temperatures are by expected to be in Kelvin. Other options on this input file are also discussed in Section 10.1.4.14.

The number of data points must be at least 2 and cannot be greater than 10000 (10^4). The warning and error messages associated with this file are the same as those described in Section 10.2 and will not be repeated.

Chapter 11: 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 41. The values in parenthesis in the following descriptions are the titles used for the columns of data in the output files.

Table 41: Example of Data Format in Output Files

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

Note: All input and output file names have been presented in uppercase to make them stand out in the manual. The actual input and output files on the CD-ROM have lower case filenames and all output files from LEWICE 3.2 have lowercase filenames.

11.1. 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 (β), 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 Manual¹⁴. 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 distance location where the collection efficiency (β) 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.

11.2. dens.dat

This file contains predicted ice density at each location for each time step. Columns 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.

Note: Ice density in LEWICE is fixed at a value of 917 kg/m^3 unless the SLD = 1 option is used. There exists two more correlations for ice density within the source code which are currently inactive due to undesirable numerical side effects. Developers who wish to activate these correlations can do so by changing an internal flag in the source code.

11.3. 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.

11.4. 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. Each line contains the dimensionless x,y coordinates of the final ice shape. 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.

Note: The file format has changed from the format used in LEWICE 2.0.

11.5. 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**”.

Note: The file format has changed from the format used in LEWICE 2.0.

11.6. 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 software are acceptable. If they are not, the user will need to fix the geometry input file and resubmit this case.

11.7. 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. Prior to version 2.0, the last flow solution performed by LEWICE was on the iced geometry at the next-to-last time step. This output file 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) or the flow field input flag **on** (ICP = 1) since the potential flow solution will not be performed if either flag is set on.

11.8. 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.

11.9. 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/m²/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.

11.10. ice1.dat

This file contains the ice shape for the first body at each time step. Columns contain the dimensionless coordinates of the ice shape (x/c, y/c), dimensionless ice thickness (thick/c) and wrap distance from the stagnation point (s/c). 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**”.

Note: The file format has changed from the format used in LEWICE 2.0.

11.11. **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**”.

Note: The file format has changed from the format used in LEWICE 2.0.

11.12. **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 to the lower impingement limit (*slow*), dimensionless wrap distance from the leading edge to 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 to the upper impingement limit (*shi*), and dimensionless wrap distance from the leading edge to 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 distance 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.

11.13. **junk.dat**

This file contains information useful for debugging the software 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 software. By default in version 3.2, 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.

11.14. limit.dat

This file contains the lower and upper icing limits for each time step. Due to runback or evaporation effects, this limit will be different than the impingement limit output to file “**imp.dat**”. Columns are dimensionless c coordinates of the lower icing limit (x_{low}/c , y_{low}/c), dimensionless wrap distance of the lower icing limit from the leading edge of the clean airfoil (s_{low}/c), dimensionless coordinates of the upper icing limit (x_{hi}/c , y_{hi}/c) and dimensionless wrap distance of the upper icing limit from the leading edge of the clean airfoil (s_{hi}/c).

Note: The “**limit.dat**” output file was not generated in previous versions. The value for the final time step should agree with output from the utility program THICK although roundoff errors may cause the utility program to report an erroneous limit.

11.15. 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 of incoming water which freezes ($mdotf$) in kg/m, mass of impinging water ($mdotc$) in kg/m, mass of incoming water which evaporates ($mdote$) in kg/m, mass of runback water ($mdotri$) in kg/m, mass of incoming water minus mass flux which evaporates ($mdotti$) in kg/m, and mass flux of standing water ($mdott$) in kg/m. 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 file has not changed from version 2.0. The output file description in the LEWICE 2.0 manual were wrong.

11.16. misc.dat

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

11.17. 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

$$\rho_o = \frac{P_o}{RT_o} \quad (19)$$

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.

11.18. qener.dat

This file contains individual terms from the energy balance. Columns are wrap distance from stagnation (s/c), net convective heat loss (q_{conv}), evaporative heat loss (q_{evap}), sensible heat loss/gain (q_{sens}), latent heat gain (q_{lat}), conduction heat loss/gain (q_{cond}) and the sum of these individual terms (q_{tot}). The numbers in this last column should be very close to zero as an indicator that energy was balanced at that control volume. Note that the net convective term includes heat gained by kinetic heating. If the input flag EPRT is set to 1, the output from every $1/10^{\text{th}}$ control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

11.19. 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 $1/10^{\text{th}}$ control volume will be generated. If this flag is set to 2, the output from every control volume will be generated.

11.20. 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 $1/10^{\text{th}}$ 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.

11.21. 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 1.6¹⁴. 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.

11.22. 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.

11.23. 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 3.2 uses the last value listed on each line (xkinit2) as the sand-grain roughness, which is also dimensionless.

11.24. 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, the water bead height (hbead) in millimeters and the water bead diameter (bdiam) in millimeters. The user is referred to Appendix A for more details on these terms and how they are calculated.

Chapter 12: Additional Output Files

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

12.1. splash.dat

This file contains output from the splashing model. Columns are dimensionless wrap distance from stagnation (s/c), drop size (dpd) in microns, collection efficiency prior to splash (beta), splash fraction (fsplash), re-impinging fraction (freimp), re-impinging efficiency (breimp), and the surface index where the re-impinged mass came from (iimp). These values will be output for each drop size in the distribution and for each time step in the simulation.

The final collection efficiency for that drop size is then calculated from the equation:

$$\beta_d = \beta_i(1 - f_i)$$

where β_d is the collection efficiency for that drop size, β_i is the collection efficiency prior to splash, f_i is the fraction splashed, β_r is the re-impinged efficiency, and f_r is the fraction reimpinged. This output will be generated if the splashing flag is set **on** (SLD = 1). For a multi-body case, output for each body will be generated in succession. The collection efficiency output in **beta.dat** is then calculated by the following equation:

$$\beta = \sum_{d=1}^{ND} \beta_d F$$

where ND is the number of drop sizes and FLWC refers to the distribution fraction input in Section 8.4.

12.2. 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 kW/m², maximum temperature (T_max) in degrees Kelvin, surface temperature (T_surf) in degrees Kelvin, effective heat transfer coefficient (h_eff) in W/m²/K, freezing fraction (f_fract), and evaporative fraction (f_evap). The effective heat transfer coefficient can be used to transfer information from LEWICE to other programs. It is the “effective” value which would need to be input as the convective boundary condition for many commercial or non-commercial CFD programs in order to thermally account for icing effects. Freezing fraction is the fraction which does not evaporate that freezes. For cases where IEVAP = 2, the program takes the amount of heat specified by the user. If this amount of heat is insufficient, ice can form. When IEVAP = 2 and IDEICE = 1, the program will override the standard LEWICE ice accretion and use the result from the anti-icing routine to provide the user with an estimate of runback ice formation neglecting conduction. This result can be compared with the IDEICE = 4 option that includes conduction effects. Evaporative fraction may be slightly less than one for IEVAP = 1 cases since the program iterates on the final temperature.

This output will be generated if the de-icing 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 can appear high due to the simplistic assumptions made in the solution process. The user is strongly advised to use a value of IDEICE = 3 or 4 for more accurate prediction of these maximum temperatures.

Note: Heat flux values are input (and output) in units of kW/m² while thermal conductivity and heat transfer coefficients are input (and output) in units of W/m/K and W/m²/K, respectively.

12.3. 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).

12.4. 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.**

12.5. force.dat

This file contains output from the ice shedding algorithm and will be output for IDEICE = 2 to IDEICE = 4 cases. The columns are time (time), total force due to adhesion (fad), net force on the ice shape in the x-direction (fx), net force in the y-direction (fy), and net force in the z-direction (fz). This file will be output if ISH = 1 in the de-ice input file. Units on time are seconds and all force units are in kN/m (kilo Newtons per meter).

12.6. force2.dat

This file contains output from the ice shed trajectories and will be output for IDEICE = 2 to IDEICE = 4 cases where ice forms then sheds. The columns are shed time (time), trajectory time step (tadd), x-coordinate of shed ice particle (x/c), y-coordinate of shed ice particle (y/c), z-coordinate of shed ice particle (z/c), and resultant force on the ice shape (ftot). This file will be output if ISH = 1 in the de-ice input file.

Units on time are seconds; distances are non-dimensionalized by chord; and force is in kN/m.

Note: The shed trajectory computation assumes that shed ice particles are spheres.

Note: The shed force data files can be extremely large and should be deleted if they are not needed.

12.7. qopt.dat

This file contains output from heater optimization runs and will be output for IDEICE = 2 to IDEICE = 4 cases where heaters are temperature controlled (including IQW = 1 cases). The columns are wrap distance index value [i], normal index value [j], temperature at control location [t(K)], and heat flux [qopt(kW/m²)]. When the temperature control option is being used in the de-icer input file, the software will use the wattages input by the user initially and then increase or decrease those wattages such that the temperature at the control location remains in the range specified. The wattages needed to maintain the heater section in that temperature range are output to the user in this file. As noted in the file header, units of temperature are in Kelvin and wattages are in kW/m².

Note: There does not exist a “convergence criteria” for the heater optimization process. If the user does not input a large enough time for the algorithm to fully converge, then the wattages output may not be optimal. Judgment is required by the user to decide if the wattages shown have converged.

12.8. dummy.inp

This file contains an echo of the de-icer input data without any user comments (i.e., only the numbers are written to this file). This file may be useful for debugging de-icer input data files. The software reads in the de-icer input file and strips off all the comments and then writes this file to disk. It then reads this file when it reads the variables into the program. The variables listed in this file are described in Section 10.1.

12.9. temps1.dat, temps2.dat, temps3.dat, temps4.dat, temps5.dat, temps6.dat, temps7.dat

These files contain temperature output as specified by Section 10.1.5.3 to 10.1.5.5 of this manual. If thermocouple output was specified (IOTYPE = 1), then the first column will be time (time), and the remaining ten columns will contain the temperature in Kelvin for the first ten thermocouple locations specified. The file header will denote these temperatures with the variables ‘t1’, ‘t2’, ‘t3’, ‘4’, ‘t5’, ‘t6’, ‘t7’, ‘t8’, ‘t9’, ‘t10’. If more than ten thermocouple locations are specified, then the next ten thermocouples will be written to the next file (for example, the first 10 will be in “**temps1.dat**”, the next 10 will be in “**temps2.dat**”, etc.). The file header for each case will remain the same.

If a temperature profile as a function of wrap distance was specified (IOTYPE = 2), then the first column will contain the wrap distance (s/c) and the remaining ten columns will contain the first ten temperature profiles specified in the de-icer input file for the IOTYPE = 2 option. The file header will denote these temperatures with the variables ‘ta’, ‘tb’, ‘tc’, ‘td’, ‘te’, ‘tf’, ‘tg’, ‘th’, ‘ti’, ‘tj’ to distinguish them from the IOTYPE = 1 output. Once again, if more than ten profiles were requested, the second ten profiles will be in the next highest number output file (for example, if the first ten profiles were output to “**temps3.dat**” then the second set of ten outputs would be written to “**temps4.dat**” and so on).

If a normal direction temperature profile was specified (IOTYPE = 3), then the first column will contain the normal distance (eta/c) and the remaining ten columns will contain the first ten temperature profiles specified in the de-icer input file for the IOTYPE = 3 option. The reference

value for eta is the inside surface of the de-icer. The file header will denote these temperatures with the variables 'taa', 'tbb', 'tcc', 'tdd', 'tee', 'tff', 'tgg', 'thh', 'tii', 'tjj' to distinguish them from the IOTYPE = 1 or IOTYPE = 2 output. Once again, if more than ten profiles were requested, the second ten profiles will be in the next highest number output file (for example, if the first ten profiles were output to “**temps3.dat**” then the second set of ten would be output to “**temps4.dat**” and so on).

Note: For a case where the user selects a number which is not divisible by 10, then extraneous data columns will be written which should be ignored. The software will always write out 11 columns in each of these files. The first column will always contain the independent variable with the next 10 containing temperature values. Example: The user specifies 25 locations for thermocouple style output (IOTYPE = 1) and 12 locations for wrap distance style output (IOTYPE = 2). Files “**temps1.dat**”, “**temps2.dat**”, and “**temps3.dat**” will contain the IOTYPE = 1 output while “**temps4.dat**” and “**temps5.dat**” will contain the IOTYPE = 2 style output. In file “**temps3.dat**”, the last five columns of numbers should be ignored in this example while in file “**temps5.dat**”, the last eight columns should be ignored.

If IOTYPE = 4 or IOTYPE = 5 output is specified, then the software will output the time at the top of the output section [time (sec)] on a single line. Following this line will be the temperatures at every node. Each column of data represents one column of nodes in the wrap direction. The first column will contain the normal distance (eta/c) and the remaining ten columns will contain temperature in Kelvin for the first ten columns of nodes. The second set of ten nodes will follow the first set and so on, as diagrammed in Figure 17.

```
time (sec)
eta(j) t(1,j) t(2,j) t(3,j) t(4,j) t(5,j) t(6,j) t(7,j) t(8,j) t(9,j) t(10,j)
-----
eta(j) t(11,j) t(12,j) t(13,j) t(14,j) t(15,j) t(16,j) t(17,j) t(18,j)
t(19,j) t(20,j)
-----
eta(j) t(21,j) t(22,j) t(23,j) t(24,j) t(25,j) t(26,j) t(27,j) t(28,j)
t(29,j) t(30,j)
etc
```

Figure 17: Diagram of temperature output for IOTYPE = 4 and IOTYPE = 5 output

12.10. fort.54

The file “fort.54” contains debug information concerning the phase change convergence of the program. The user should not need the output from this file unless the user is also a developer looking to improve the software’s capabilities. There are 10 columns of data output to this file but there is no file header. The columns are: wrap distance index value, normal index value, current temperature at node, previous temperature at node, previous time step temperature at node, current phase index, previous phase index, previous time step phase index, time, and iteration number.

12.11. fort.80, fort.81, etc.

The file “fort.80” will contain the de-icer grid and ice grid for the first time specified with the variable ITIME in Section 10.1.5.2. File “fort.81” then contains the temperature solution file for

the time specified. The format for these files will be determined by the value of the NPRF flag entered in Section 10.1.5.1. If more than one ITIME value was specified, additional output files will be generated. The file names for the second time specified will be “fort.82” and “fort.83” respectively. If a third time is specified, the output file names will be “fort.84” and “fort.85” and so on. Geometry grids are generated for each time specified since the ice grid may change due to ice growth.

12.12. piccolo1.dat

This file will output the surface temperatures and internal heat transfer coefficients for bleed air cases when I THERM = 1. The data will be output to this file for the first time step only. Columns are x-coordinate, y-coordinate, spanwise section, surface temperature in Kelvin, and internal heat transfer coefficient in kW/m²K. The coordinates are normalized by chord. There is no column header for this output since outputs from several spanwise runs are usually concatenated and used as input to TECPLOT (or similar software). The user is referred to Examples 20 and 21 in this manual for more information on this process.

12.13. piccolo2.dat

This file will output the surface temperatures and internal heat transfer coefficients for bleed air cases when I THERM = 1. The data will be output to this file for the last time step only. Columns are x-coordinate, y-coordinate, spanwise section, surface temperature in Kelvin, and internal heat transfer coefficient in kW/m²K. The coordinates are normalized by chord. There is no column header for this output since outputs from several spanwise runs are usually concatenated and used as input to TECPLOT (or similar software). The user is referred to Examples 20 and 21 in this manual for more information on this process.

Chapter 13: Utility Programs

This section will describe five utility programs which are released along with LEWICE 3.2. The inputs and outputs to these utility programs will also be discussed.

13.1. 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 for LEWICE 2.0³. Only one body can be processed for this program. A multi-body case can be processed by running THICK for each individual body geometry.

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 current version of THICK included on the CD can use the same input geometry that was used in LEWICE. There no longer exists a need to input a refined geometry. The refined geometry will be calculated internally in the program. The user should consult the report on the validation tests³ and Appendix A for further details on the methodology used in this program.

13.1.1. 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) ***
```

Windows 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 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 is 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.

13.1.2. File Format for Clean Geometry and Iced Geometry

The file format for the clean geometry and the iced geometry are the same. Each line 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. **The number of points in the geometry is no longer input.** The input file format for the clean and iced geometries is identical to the format for the body geometry input for LEWICE. 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. There is no longer any need to create a more refined body geometry for input into THICK. The input geometry for the clean airfoil can (and should) be the same as the LEWICE input geometry.

Note: The units of the input variables for THICK can be any consistent set of units. The program will normalize the data for processing and then convert the data back to the original units for output. Therefore the program should give consistent output regardless of the input units.

13.1.3. THICK output files

THICK produces five output files, each of which contains different methods for looking at the ice thickness output. The first two output files, “**clean.dat**” and “**iced.dat**”, contain the ice thickness on the clean and iced geometries. “**clean.dat**” contains four columns of data. The first two columns contain the x,y coordinates (xsav, ysav) of the body geometry input. The number of data points on the geometry may have changed, however. The units of the output data will be the same units as the input data file. The third column contains the ice thickness (ditot) at that location and the fourth column contains the wrap distance (s) as measured from the leading edge (minimum x- coordinate value). The units for these variables will be the same as the units for the body geometry. “**iced.dat**” contains the x,y coordinates (xice, yice) of the iced geometry and the ice thickness (yptot) to the body surface at each location. This output should have the same number of points and the same units as the input iced geometry.

Note: THICK does not perform any error checking on the input files. If there are any problems with the input files, then the output of THICK will be in error. The most common error occurs when the clean body geometry and the iced geometry are in different units.

The next output file, “**echo.dat**”, contains an echo of all of the screen output from THICK, including any program error messages. This output includes a summary of the icing parameters which THICK has been set up to calculate. Upper and lower icing limits, horn heights, horn angles (labeled upper and lower maximum ice thickness) are output as well as leading edge minimum thickness and ice area. The units of these variables will coincide with the units of the input geometry (except horn angles which are in degrees). Each of these parameters were defined in the LEWICE validation report. The definitions of horn angle and ice area have changed for this version. The current definitions correspond to those given in an SAE Aerospace Recommended Practices (ARP) report (ref.). The current definitions are also shown in the diagram and equation 24 below. Since THICK does not input the icing parameters or chord length, the normalized variables described in the LEWICE validation report cannot be output directly. If THICK fails to find a given parameter, it will print “N/A” for the value. The outputs from THICK cannot supplant good engineering judgment. THICK occasionally cannot find some of the parameters which exist and can also output erroneous information. The user is referred to the following section for details.

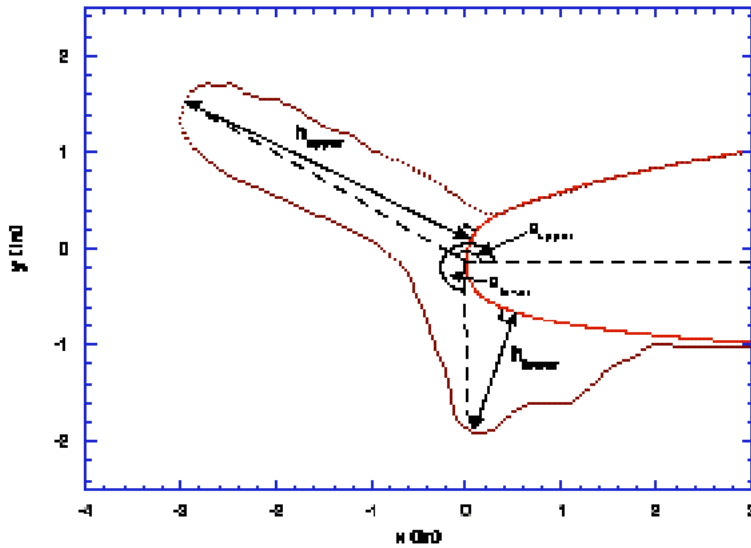


Figure 18: Horn Angle Definition for THICK

$$A = \left(\int_{xlow}^{xhi} y dx \right)_{iced} - \left(\int_{xlow}^{xhi} y dx \right)_{clean}$$

Ice Area definition for THICK, where xlow = lower icing limit and xhi = upper icing limit.

The fourth output file, “**peaks.dat**”, was added to THICK to aid the user in correcting problems where THICK does not label the correct peak as the upper or lower horn. This file contains all of the local maxima in ice thickness found by THICK. Experimental ice shapes will often have several peaks due to the irregularity of the ice shape. Ice shapes from LEWICE or other programs may have multiple peaks as well. Each peak found is labeled with the header “Found a Peak”. The index location and x,y coordinate location (on both the clean and iced surface) are output as well as the ice thickness and angle (in degrees). The coordinates and thicknesses output will have the same units as the input data. If the user believes that THICK chose the wrong location for either the upper or lower horn, then the user can choose to select one of the peaks given in this file.

The last output file, “**total.txt**”, contains only the eight icing parameters described in the SAE ARP mentioned earlier. The variables output are, in order: lower icing limit, upper icing limit, lower horn height, leading edge minimum thickness, upper horn height, ice area, lower horn angle and upper horn angle. The numbers are output in the same units as the input data and have not been normalized. Only the numbers are output. There do not exist any text header descriptions. The output was formatted in this manner in order to facilitate the automation of quantifying large numbers of ice shapes and importing the results into a spreadsheet software such as Microsoft Excel. This option is best described by the example below.

13.1.3.1. THICK Example Case

Computation Time: Approximately 15 minutes on a 550 MHz Pentium III.

In this example, the user has a large number of data files for which the user wants the eight quantitative icing parameters defined in the SAE report mentioned earlier. In this example, the experimental ice shapes from the LEWICE validation CD were chosen. An ASCII text script called “**exppc.bat**” was then manually generated which contains the necessary DOS commands for running THICK for each of the data files. This script was generated in approximately one day. An excerpt from the script is given below.

Table 42: Excerpt of Script Commands

```
copy GEOMETRY\C23014.XYD cgeom.xyd
copy 23014MOD\120R2.TXT exp.txt
thick < thick.inp
mkdir 23014MOD\120R2
move *.dat 23014MOD\120R2
erase exp.txt
copy 23014MOD\126R8.TXT exp.txt
thick < thick.inp
mkdir 23014MOD\126R8
move *.dat 23014MOD\126R8
erase exp.txt
```

Each set of script commands contains the following steps:

- 1) The geometry file is copied to a generic name, in this case “**cgeom.xyd**”.
- 2) For each experimental data file which uses that geometry, the file name is likewise copied to file “**exp.txt**”.
- 3) THICK is run with the interactive input coming from file “**thick.inp**”. “**thick.inp**” contains two lines which read “**cgeom.xyd**” and “**exp.txt**” (without the quotes).
- 4) Appropriate subdirectories are created to organize the output
- 5) The output data files (excluding “**total.txt**”) are moved to the subdirectory.
- 6) File “**exp.txt**” is removed.
- 7) Steps 2-6 are repeated for each data file which uses that geometry, then Step 1 is repeated for each airfoil in the database.

Once the script has finished, then “**total.txt**” will contain the output for **all** of the cases in a column format which can easily be imported into a spreadsheet software such as Microsoft Excel®. Note that in this script, the clean geometry and the iced geometry are in the same units since no conversions have been performed. The output can then be normalized from within the spreadsheet software. This process, including differences and percent differences has been performed for the experimental data on the LEWICE Validation CD-ROM. This spreadsheet, which contains values for Version 2.0 as well as Version 3.2 and experimental data, has been put on the

distribution CD-ROM for LEWICE 3.2 and is named “**total.xls**”. The astute user will notice that several of the quantitative values generated using this process do not agree with the values given in the LEWICE validation report spreadsheet “**matrix.xls**”. This demonstrates that **the automated process cannot (yet) be substituted for good engineering judgment**. The following section will describe some of the problems and potential solutions.

13.1.4. THICK Judgment Errors

The utility program THICK performs a simplistic geometric comparison between two data files and outputs the results. As such, it is prone to errors based upon a lack of icing knowledge. It is possible for any (or all!) of the eight parameters to be incorrectly output. Even so, THICK has proven to be a useful tool for reducing the amount of time needed to quantify ice shape parameters. There are two basic problems which occur repeatedly when analyzing output from THICK: misalignment of the geometries and misidentification of the horns.

First, the iced shape may be misaligned with the airfoil. This is illustrated in the following figure.

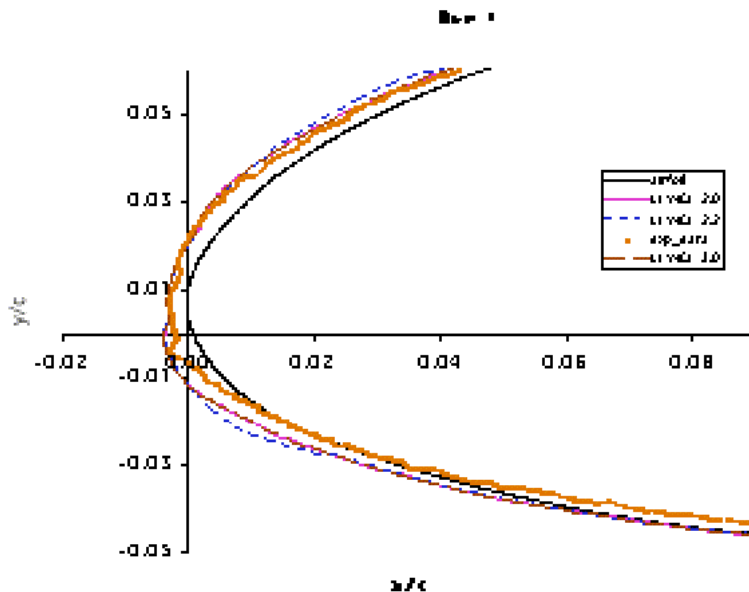


Figure 19: Ice Shape Misalignment

This figure shows a comparison for Run 15 in the NACA 4415MOD database. It appears as though the experimental data goes inside the airfoil on the lower surface. This will cause the lower icing limit to be smaller than actual. The upper icing limit may also be larger than actual and the misalignment will subtly change the area, horn heights, and possibly horn angles. This error can possibly be fixed manually by shifting the experimental ice shape to match the airfoil as shown in Figure 20.

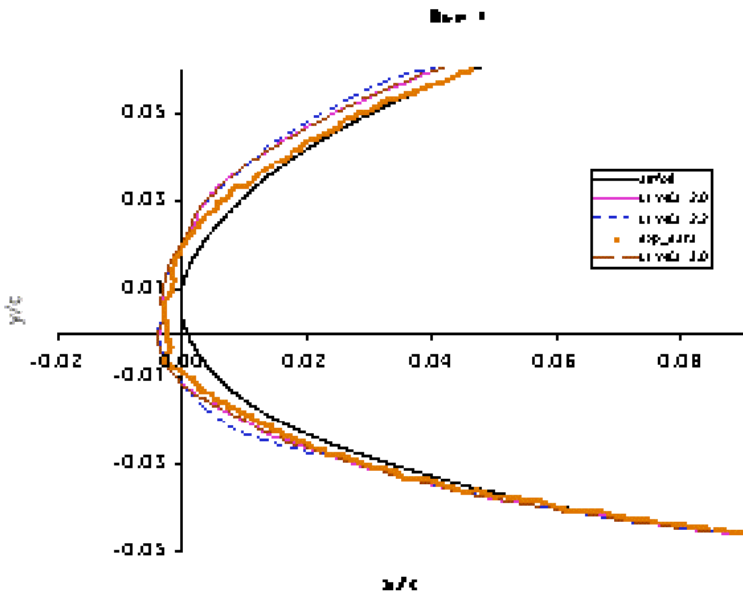


Figure 20: Shifted Ice Shape Plot

The problem with this procedure is that the translation chosen has no scientific basis - it just makes the ice shape “fit” better. This problem arises since the reference locations used for digitizing the ice shapes may not be the same reference point used for defining the airfoil coordinates. This problem can also occur with ice shapes generated by LEWICE, although the effect is more subtle as shown in the following figure. Since LEWICE outputs the entire iced geometry, any misalignment can cause the icing limits reported by THICK to be located as far back as the airfoil trailing edge.

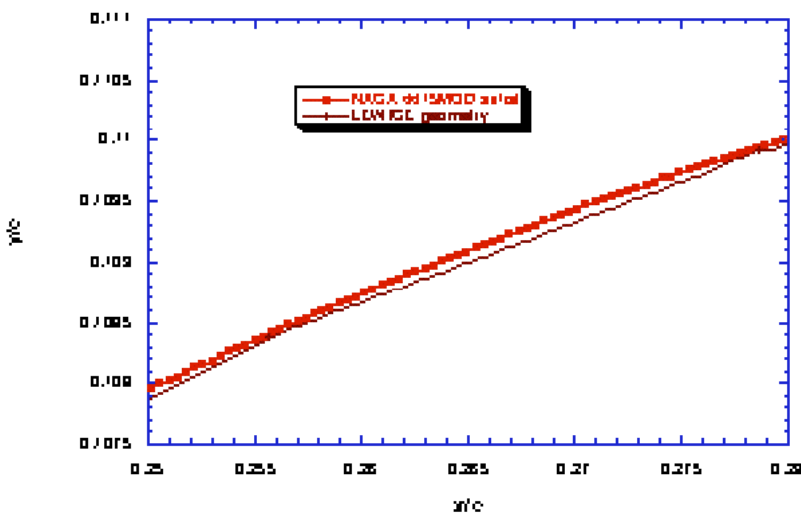


Figure 21: Expanded View of Airfoil Misalignment

Once again, the “iced” geometry resides inside the airfoil. It should be noted that the scale has been greatly expanded in this figure to show a very small geometric discrepancy. THICK will ignore values inside the airfoil, so in this case the THICK output may be accurate. If the discrepancy puts the iced geometry outside the airfoil, THICK may misidentify this small level as “ice” and report an erroneous icing limit. THICK has been set up to ignore some small discrepancies, but this tolerance cannot be set too high or else THICK will under report the limit. The icing limits reported by THICK should be checked with the icing limits output by LEWICE 3.2 in the file “**limit.dat**”. Ice area can also be in error due to this type of misalignment.

The second problem which occurs occasionally with THICK is that the user may decide to choose an ice horn that THICK did not initially identify as such. This problem occasionally occurs with SLD cases that have more than two large peaks due to large feather formations. THICK will choose the two largest horns and label them as the “upper” and “lower” even if the two largest peaks are both on the lower part of the airfoil. This is not a bug. Figure 22 shows a case where both horns are below the leading edge of the clean airfoil. This version of THICK correctly identifies the horns for this case. If the user wishes to choose peaks that THICK did not initially identify as horns, the user can choose the horns from the list of all known peaks which are output to “**peaks.dat**”.

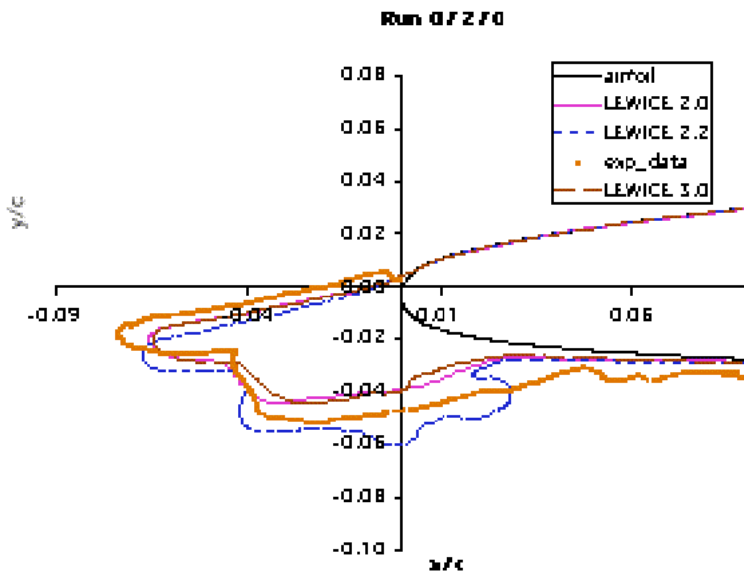


Figure 22: THICK Horn Angle Illustration

In this example, the LEWICE validation report lists the “upper” horn as the peak location of the upper “finger”. The “lower” horn height would be given by the maximum thickness of the lower block feature and the leading edge minimum would have been listed as the minimum thickness between those two values. Horn angles would then be calculated from those locations. The version of THICK included with LEWICE 3.2 now correctly identifies the horns for this case. In any case where the user has a discrepancy, the LEWICE validation report numbers should be used as the reference values when comparing to an automated process such as the one described here.

13.2. 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 spline interpolation. This program had been useful for creating clean geometry input files for use with program THICK, but may not be as useful since THICK has incorporated this feature.

13.2.1. 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) ***
```

Windows 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 d NPOINTS d 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.

13.2.2. File Format for Input and Output Geometries

The file format for the input and output geometries are the same. Each line 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. **The number of points in the geometry is no longer input.** 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 the same as the geometry input file format for LEWICE 3.2.

13.3. Program CONVERT

Input files from previous versions of LEWICE will not work with version 3.2. Program CONVERT will read in an input file from LEWICE version 1.6 and convert it for use in LEWICE version 3.2. 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 3.2 input file. LEWICE 2.0 users simply need to add the following four lines to the end of the main input file:

```
&RDATA  
&END  
&BOOT  
&END
```

LEWICE 2.2.2 users only need to add the last two lines to the end of the main input file.

LEWICE 2.4 users do not need to change the main input file.

13.3.1. 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) ***
```

Windows 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 3.2 Input File Name

This interactive input directs the user to input the name of the file which will contain the LEWICE 3.2 input. **Note:** If the file name input already exists, it will be overwritten.

13.3.2. 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 Manual¹⁴. 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 3.2. Refer to those sections describing this input for a more detailed explanation.

13.3.3. Warning Message for CONVERT

The flag INCLT in the LEWICE 1.6 input file is not used in LEWICE 3.2. 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 3.2 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 3.2 file will be generated.

13.4. Program CONDEICE

De-icer input files from previous versions of LEWICE will not work with version 3.2. Program CONDEICE will read in a de-icer input file from LEWICE version 2.2.2 and convert it for use in LEWICE version 3.2. The program will prompt the user for the name of the LEWICE 2.2.2 de-icer input file and the name to be given to the LEWICE 3.2 de-icer input file. LEWICE 2.4 used the same de-icer input file as LEWICE 2.0, so the conversion program will work on those de-icer input files as well.

13.4.1. Interactive Input for Program CONDEICE

Enter Version 2.2.2 De-icer Input File Name

The first interactive input directs the user to enter the name of the LEWICE 2.2.2 de-icer 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) ****

Windows 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 3.2 De-icer Input File Name

This interactive input directs the user to input the name of the file which will contain the LEWICE 3.2 de-icer input. **Note:** If the file name input already exists, it will be overwritten.

13.4.2. Format for the LEWICE 2.2.2 De-icer Input File

Program CONDEICE will read input files which conform to the format listed in the LEWICE 2.0 User Manual¹. No error checking of this input file is performed.

Note: Additional inputs will be set to their default values. Please refer to Section 10.1 for a description of the current de-icer input file.

13.5. Programs TOBINARY, TOBINARYM

Naviér-Stokes solution files in PLOT3D format 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) ****

Windows 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 exist, 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.

Note: TOBINARY works with single block grids where the number of grids blocks is not output to the file. TOBINARYM works with the PLOT3D multi-block format.

13.6. Programs TOASCII, TOASCIIM

Naviér-Stokes solution files in PLOT3D format 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) ***
```

Windows DOS Shell Message

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

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.

Note: TOASCII works with single block grids where the number of grids blocks is not output to the file. TOASCIIM works with the PLOT3D multi-block format.

13.7. Program ANTICON

This program reads the LEWICE3D files named “fort.2” and “fort.26” and outputs the files “**flow.dat**”, “**beta.dat**”, “**htc.dat**”, and “**stream.dat**” which contain the 3D results along 2D streamlines which can then be read into LEWICE using the input flags ICP, IBETA, IHTC, and I3D. The files “**flow.dat**”, “**beta.dat**”, “**htc.dat**”, and “**stream.dat**” must first be segmented into several files, each of which contains only one of the streamlines output to the other files. For example, if the file “**flow.dat**” contains five 2D streamlines, the user must first create five separate files, each of which contains only one of the streamlines. This function can be performed with any text editor or word processor software. These files need to have the proper name as described in Sections 8.3 and Sections 10.5 through 10.8 in order to be read properly by LEWICE. See the example in Section 15.18 for more details.

13.8. Program ANTICON2

This program reads the LEWICE3D files named “fort.2” and “fort.26” as well as the LEWICE 3.2 output files “**dyice.dat**” and “**temp.dat**” and outputs files named “PLOT3D.XYZ” and “PLOT3D.Q” which can be read by the NASA software PLOT3D. If more than one streamline exists in the “fort.26” file, then several output files from LEWICE 3.2 must be concatenated

prior to running ANTICON2. This function can be performed with any text editor or word processor software. For example, if five 2D streamlines were output to “fort.26”, then the user must perform five runs of LEWICE 3.2 in order to produce output for ANTICON2. Then the five “**dyice.dat**” output files must be combined into a single “**dyice.dat**” file prior to running ANTICON2. See Section 15.18 for more details.

13.9. Program REDUCE

This program reads the LEWICE 3.2 output files “**temps1.dat**”, “**temps2.dat**”, and “**temps3.dat**” and outputs files “**temps1a.dat**”, “**temps2a.dat**”, and “**temps3a.dat**” which are more easily compared to the existing validation data. The program converts the temperatures in these files from Kelvin to Fahrenheit and reduces the output from one temperature every 1/10th second to once every second. The program assumes that “**temps1.dat**”, “**temps2.dat**”, and “**temps3.dat**” all contain thermocouple type output (IOTYPE = 1) and that the de-icer time step is 0.1 seconds.

13.10. Program LewCon

Currently, LEWICE runs as a DOS program that accepts many different inputs such as cloud conditions, wing shapes, and thermal deicing inputs. However input conditions and experimental data are often stored in spreadsheets. LEWICE inputs are text files; therefore, they must be generated by the user. LEWICE’s outputs (collection efficiency, ice shapes and thicknesses) are also text files; to plot the data, users must generate a spreadsheet with this output. Because all LEWICE I/O is in the form of text files, using LEWICE for a large number of runs can be time-consuming.

Our goal was to improve LEWICE’s usability by creating a user interface that would automatically generate LEWICE input from a spreadsheet and automatically put LEWICE output into spreadsheets with charts. Additionally, this user interface would automatically convert units (as LEWICE only accepts input in certain units) and offer several output options. This program will be called the “LEWICE Console”.

The LEWICE Console is an easy to use interface for LEWICE written in Visual Basic. It allows users to run LEWICE given a spreadsheet listing experimental conditions. It automatically generates the input to LEWICE, does necessary unit conversions, runs LEWICE, and produces a spreadsheet with charts plotting the data. It allows users to import data from previously generated LEWICE inputs into a spreadsheet. It also allows users to batch run LEWICE on several different inputs to automatically generate multiple output spreadsheets. You can also generate plots of LEWICE output data vs. experimental data.

These capabilities are just the beginning for the LEWICE Console. LEWICE is capable of running a full de-icing simulation given the geometry and heating apparatus information. However, users find it difficult to run such simulations due to the number of inputs and the difficult input file format. The LEWICE Console could simplify this simulation by allowing the user to interactively draw a geometry, place heating apparatus, and specify information about each part. The input to LEWICE would be automatically generated from the experiment the user draws on the screen. The console version included on the CD will only perform icing simulations. De-icing cases must be performed using the normal input procedures.

13.11. LewCon Example Case

This example performs a simple walk-through of the capabilities of the LEWICE Console. Along the way, various notes and bug will be pointed out. This console was initially developed during a 10-week period by Aaron Armstrong, a senior in Computer Science during a summer internship and later modified to add certain features. It is meant to serve as a prototype for the GlennIce project, however I thought it would be useful to send it out for user feedback. Please send all feedback and bugs to William.B.Wright@grc.nasa.gov

- 1) Copy LewCon.exe to your hard drive
- 2) Open (double-click) LewCon.exe
- 3) Click “Load Spreadsheet”
- 4) Select “Otter_Cases.xls”

Note: You can use any Excel spreadsheet. The only requirement is that the first column is the case name. The console hasn’t been tested if the user has extraneous information at the top of the file.

- 5) Select columns and units for input variables

Note: Typically, a column header will be put in the file as shown in the example with the appropriate units. The program does not recognize those units, the user has to specify them. The user can choose either total temperature or static temperature for the input. You can also choose to enter altitude instead of pressure. You can also get to this dialog box by selecting “Specify Spreadsheet Data Source Columns” on the main window. Selecting “Save Preferences” will save the order and units so that you don’t have to change this setting if all of your spreadsheets are formatted similarly.

Note: Multiple drop size distributions will work. The multiple values of FLWC and DPD will reside in a single cell in the spreadsheet and must be separated by commas. The values can be entered in the spreadsheet cell or the console box in the main window.

- 6) Click on case CG0951F in the window. Note that the conditions appear in the boxes at the top. You can also run a single case by typing numbers into these boxes rather than selecting them from the spreadsheet. Select more than one case by using Shift-Click.

Note: This last sentence demonstrates the main feature of the LEWICE Console. The user can have hundreds of cases listed in a spreadsheet and then Shift-Click to select all of the cases and LEWICE will run every case selected and produce Excel Charts of your choosing with the LEWICE output.

- 7) Click “Run LEWICE” A quick run-down of other buttons on this page:

- a) “Import Data” imports conditions from an existing LEWICE input file.

- b) “Output Data to File” exports the selected condition and creates a valid LEWICE input file. The console will prompt you for a Title (line 1 in the LEWICE input file) but the input file itself will be called “lewin.txt”. Also, if you Shift-Click cases and select “Output Data to File”, the console does not output multiple LEWICE input files (known bug).

c) “Experimental Data” - Browse. This option looks for an Excel file containing the experimental data to be plotted against the LEWICE inputs. The Excel file should only contain the data and a header file. This feature has not been tested if there is extraneous info in the file. Also, it only chooses one file for each case, so you cannot Shift-Click and have it ask for multiple files. Select a single case (for example CG0951F) and then select “Experimental Data”.

d) “De-icer Input” This option allows the user to set up a deicer input file using the GUI. This is still in a very primitive stage at this point and requires a separate walk-through.

Note: This feature does not run de-icer cases. It conceptualizes what a de-icer GUI would look like.

e) “LEWICE Preferences” allows the user to set LEWICE inputs other than conditions. A separate screen will appear with input boxes and check boxes for LEWICE inputs. This screen can be used to set the SLD flag, time step flag (ITIMFL) and other inputs.

f) “Quit” exits the program.

8) Once “Run LEWICE” has been selected, the console opens a new window. The first option selects a LEWICE input file. If spreadsheet cases were selected, check the box labeled “Use Data Selected on LEWICE Console”

Note: If more than one case was selected in the main window, there will be a check mark labeled “Flight Simulator Mode”. If this box is checked, LEWICE will use the final ice shape from the first case as the input for the second case rather than restarting with the clean airfoil. In this manner, the console allows LEWICE to simulate cases with variable conditions such as occur in flight.

9) Click “Browse” beside the “specify a geometry file” button to select a geometry file. This is a standard LEWICE geometry file. The geometry file name can also be supplied from the spreadsheet containing the other LEWICE inputs.

10) Check the box “Run THICK” if you want the Console to run the THICK utility on the final ice shape for each case.

Note: THICK must reside in a subdirectory called “Utility” within the directory containing LEWICE.

11) Click “Output Preferences”

12) A third window appears. The top box allows you to choose the plotting options. You can choose “First Time Step”, “Last Time Step”, “First & Last Time Step” and “All Time Steps”.

13) Check each box beside the file you want plotted.

Note: Clicking the upside-down triangles give you the choice of plot options for each file. For example, “beta” can be plotted versus x/c , s/c , or sl/c . Check the box “Use Exper. Data” if you selected experimental data in the main screen. Also note that if you select “clean” to plot ice thickness from THICK, make sure to select the “Run THICK” check box or the program will crash (known bug). Also, “final” will crash the program. Select “ice” and pick “Last Time Step” to plot the final ice shape.

14) Click “Run LEWICE” in third window. The first time you do this, a standard Windows browser will appear so that you can tell the program the location of LEWICE.EXE. THICK.EXE must reside in subdirectory “UTILITY” underneath the LEWICE directory. LEWICE will then run for the cases you selected. A spreadsheet named the same as the case name will be created in the LEWICE directory and will contain all of the output and the selected plots.

13.12. LewCon Missing Features

This is a list of some features that have not yet been implemented. The list is not in any order and bugs previously mentioned are not listed.

1) There is no error checking of inputs other than what LEWICE will perform when it runs. You do not see the LEWICE warning messages on screen and the console will auto-accept the warning message. Any warnings or errors will be sent to JUNK.DAT as normal.

2) Any conditions not in the spreadsheet will have the LEWICE default value. When setting up cases manually, input files tend to be complete because the user sees all the inputs there and remember that you need a value for chord for example. When running cases through the spreadsheet, it’s easy to forget variables. This is a human bug, not a program bug, so it may be difficult to fix this.

3) The de-icer option has not been thoroughly tested. It will not write out a de-icer input file, so the console will not run de-icer cases. Deicer cases are frustrating to set up because the console crashes frequently. The materials database has not been filled in either, so while the user can select “Aluminum”, that is only a label and properties must be input manually. It is planned to change this so that when common materials are selected, the properties automatically show up. While bugs are fixed in the de-icer input section, NASA would like to have feedback on the layout of the graphical display. Is it understandable? What would you change to make it more intuitive?

Chapter 14: Runtime Errors

This section will describe errors which may occur during a run. Errors during a run are very infrequent with this software, but may still occur if a problem arises which the software 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 3.2 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 software 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.

14.1. 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 printout 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.

14.2. 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 droplet position 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 droplet position 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 droplet position 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.

14.3. 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 $h = h_{min}$, 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 $h.gt. h_{min}$.
- 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 software 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.

14.4. 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 software. 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.

14.5. 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 software 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 error in LEWICE. It would be very helpful for future versions if we (NASA) could have a copy of the input file so this error can be fixed. Also include what machine the software 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 software 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 error in LEWICE. It would be very helpful for future versions if we (NASA) could have a copy of the input file so this error can be fixed. Also include what machine the software 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 software will attempt to continue, but the output may be bad or the software will bomb later. This statement is in routine NWF3 and is usually caused by a memory allocation error.

i= (value), xbt-dot= (value), dsdd= (value).

This problem should never occur in LEWICE 3.2. 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 software 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 software.

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.

14.6. De-icer Routine Errors

The errors listed in this section are produced by the de-icer routine. They can only be generated for IDEICE = 1, through IDEICE = 4 cases. Some of the errors listed in this section are not errors, but additional information written to “**junk.dat**” for informational purposes. Those messages will be identified in their descriptions.

In NOICE_d, 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.

This error occurs for IDEICE = 1 cases where the program could not converge on the anti-icing temperature. Check the input data for possible errors, as this message did not appear during testing of the program.

The time step is <= zero! This is usually a memory allocation problem

Check output carefully. Setting dtime = 60 sec. Current dtime = (value)

This error occurs when the time step has somehow been set to zero. This should never occur, since the time step input is checked by the program.

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.

This error occurs when the energy balance equations did not converge. This error occurred during testing only after other serious warning messages listed in this section had been issued by the program. It usually occurred during anti-icing runs with an extremely high wattage. The run should be terminated and resubmitted with different inputs. See the recommendations below for the error “program is having problems with this case”.

Bad stfr = (value) (value) (value) (value)

This is a warning message only which indicates possible problems with the water runback routines. Check all output carefully, especially runback ice formations.

Bad mass flux (value) (value) (value) (value) (value) (value) (value)

This is a warning message only which indicates possible problems with the water runback routines. Check all output carefully, especially runback ice formations.

program is having problems with this case.

please check the output thoroughly. time = (value)

This warning message can occur when the program is having convergence problems in the phase change algorithm. When this message is immediately followed by six numbers (5 integers

and one real number), then the program is having serious convergence problems and may soon crash. The six numbers are: wrap distance index value, normal index value, temperature, time step index, phase change index, and previous phase change index. In any case where this message appears, the user should check the temperature output thoroughly. **If this warning message appears more than 10 times during a run or if the user detects anomalous temperature output, the output from this run should be abandoned and the case ran again with different inputs.** The user should consider the following remedies (in recommended order): reduce or eliminate LEWICE time stepping (IFLO = 1); a larger number of iterations (JCOUNTD) and/or a smaller time step (DTAUI, DTAUM, DTAUF); increased nodal spacing (NODE) especially in the ice layer and the layer adjacent; simplifying the case by reducing sections, layers or other variables.

temp below tinf (value) (value) (value) (value) (value) (value)

This error message occurs when the de-icing module crashes (stops running prematurely). It occurs when the temperature at a node drops below the ambient temperature, which should never occur under normal operation. The six values are: wrap distance index value, normal index value, temperature, time, phase change index and previous phase change index. The user should contact the developer if this error occurs and supply a sample input file for further program debugging.

phase change algorithm did not converge

This warning message is generated if the program repeatedly fails to converge on the phase for the problem at hand. The user should consider resubmitting this case with a larger number of iterations (JCOUNTD) and/or a smaller time step (DTAUI, DTAUM, DTAUF).

not converged. diftemp = (value), (value), (value), (value), (value)

This warning message occurs when the matrix solver MSIP did not converge on a temperature profile. This has occurred during testing when an anti-icing case was run with a large time step (only the steady-state solution was desired). In this case, no action is required of the user since the transient temperature profile was not needed. If this message occurs for a case where the transient profile is needed (de-icing cases), then the user should resubmit the case with a smaller time step. The user may need to examine grid refinement for this case as well. The five values output are: maximum temperature difference, temperature, previous temperature, wrap distance index value and normal index value.

bleed air option is on but flow rate is zero

either set IBLEED = 0 in the input file, or select a value AIRMD > 0

This error message would occur if the bleed air mass flow rate was zero for a case where bleed air was considered. This error should never occur as the program checks this when the data is read.

an error occurred while reading the input file

This error is generated when the program could not read the de-icer input file for some reason. Check the de-ice input file for possible errors and resubmit the case.

14.6.1. Other de-icer routine messages

This section will describe some of the other informational messages generated by the de-icer routine. These messages are not warnings or indicators of a problem with the output.

ice has shed (value) (value) (value) (value) (value) (value)

This is an informational printout only and is not a warning or error message. It is written out when the ice shape sheds. The six values are: body index, net force, force in normal direction, force in wrap direction, force in span direction and force of adhesion.

Phase has matched previous iteration

Iteration = (value)

This message occurs during the phase change iteration when the program cannot decide on the phase of a particular node or node(s) and is simply alternating between two values. The program is set up to handle this occurrence, so it is not a warning or a problem with the output.

(value) iterations were required at time = (value)

This message occurs when the program uses more than two iterations for a particular time step. It can be helpful in identifying if the program used too many iterations for a given time step. During testing, the number of iterations was almost always below 10 and usually below 5.

time = (value) sec

This message is written every 20 iterations to inform the user of the program's progress.

Chapter 15: 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. A script has also been included on the distribution disk called “EXAMPLES.BAT” which will automatically run all of the example cases. The computation time for running the script will be the sum of the computation times for all of the individual cases. The results shown in this section were generated using the SGI 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 de-icing and anti-icing cases will additionally show plots for selected temperature distributions. 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 software stops after calculating the ice shape for the last time step.

15.1. Case 1: Run 072501 from Validation Report

Computation Time: Pentium IV 2.4GHz, 6 s.

Disk Space: 1 MB

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 43: Main Input File for Example Case 1

Example 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

&RDATA

&END

&BOOT

&END

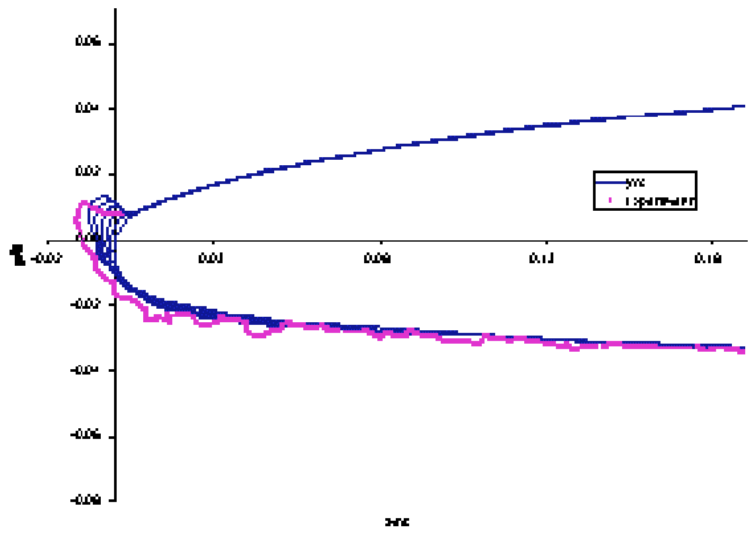


Figure 23: Ice Shape for Example Case 1

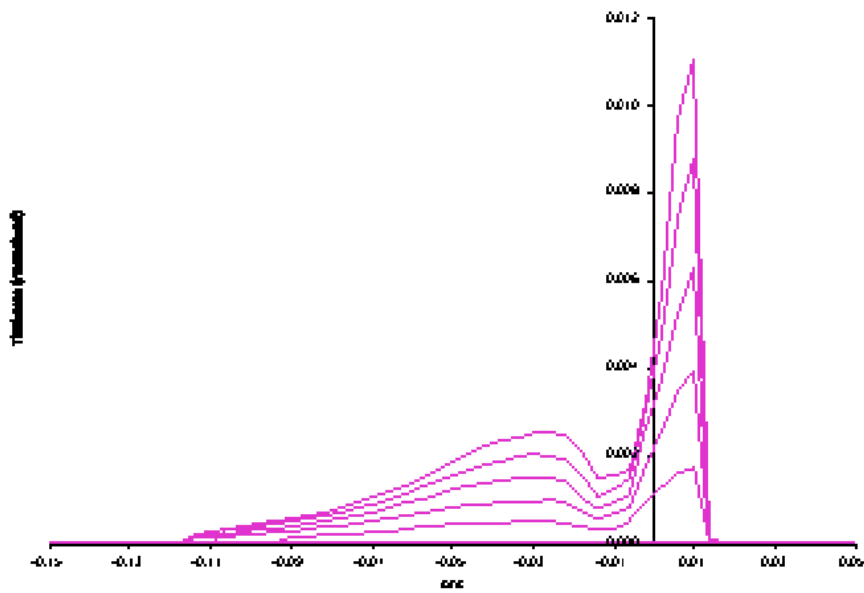


Figure 24: Ice Thickness for Example Case 1

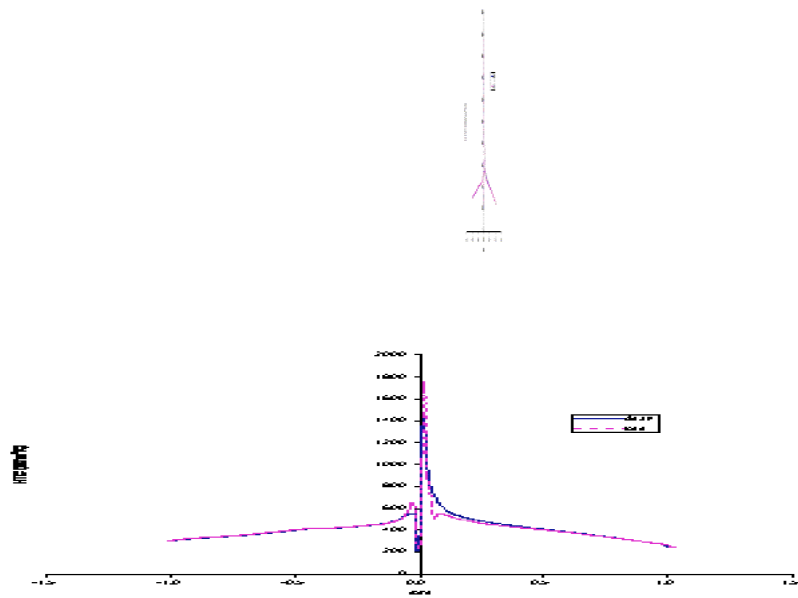


Figure 25: Heat Transfer Coefficient for Example Case 1

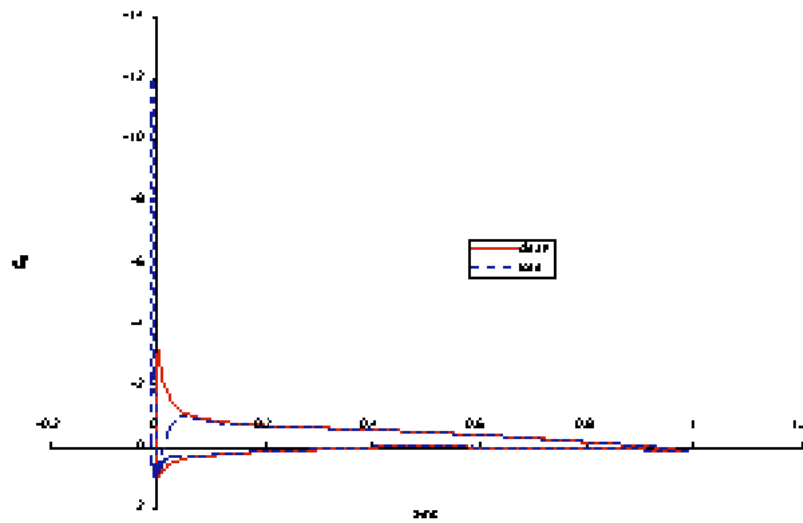


Figure 26: Pressure Coefficient for Example Case 1

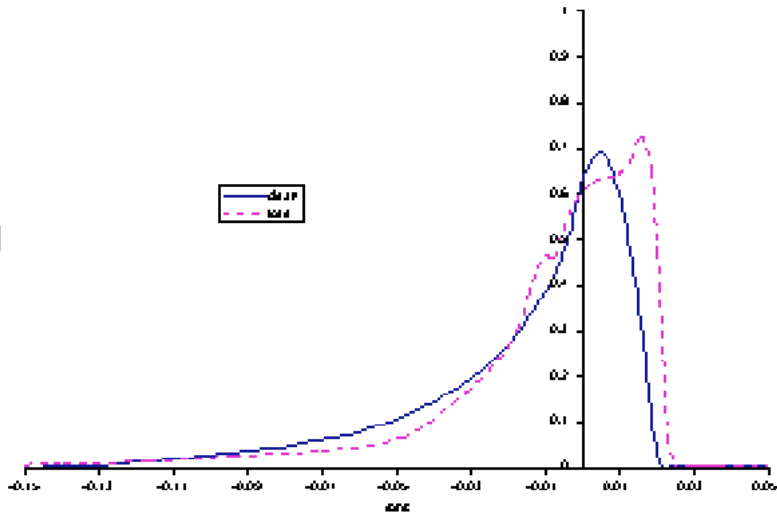


Figure 27: Collection Efficiency for Example Case 1

15.2. Case 2: Langmuir ‘D’ distribution

Computation Time: Pentium IV 2.4GHz, 34 s.

Disk Space: 993 KB

This case shows an example using a drop size distribution. The conditions are identical to Example Case 1 except for the droplet 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 45. The main input file and geometry input file are provided on the distribution disk as “**case2.inp**” and “**case2.xyd**” respectively.

Table 44: Main Input File for Example Case 2

```

Example 2 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
VINFL  = 90.
LWC    = 0.540
TINF   = 268.30
PINF   = 100000.00
RH     = 100.0
&END
&LPRNT
FPRT   = 1
HPRT   = 1
BPRT   = 1
TPRT   = 0
&END
&RDATA
&END
&BOOT
&END

```

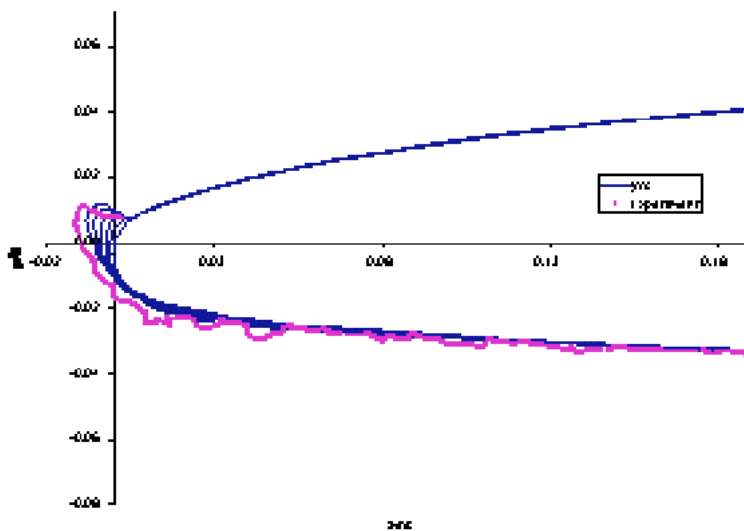


Figure 28: Ice Shape for Example Case 2

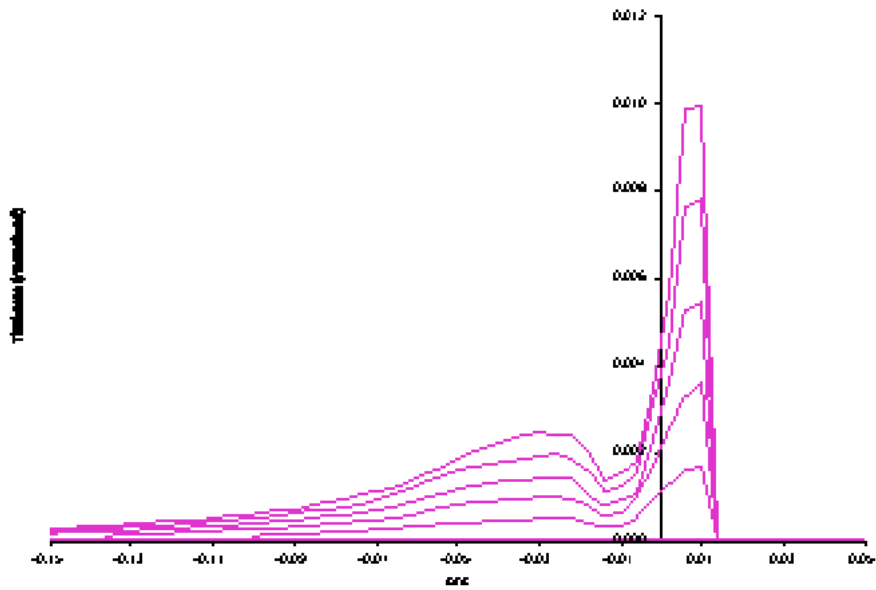


Figure 29: Ice Thickness for Example Case 2

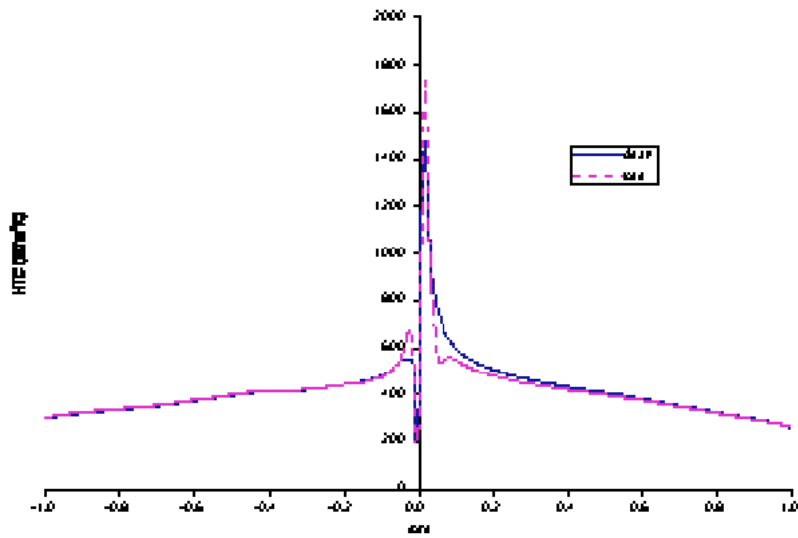


Figure 30: Heat Transfer Coefficient for Example Case 2

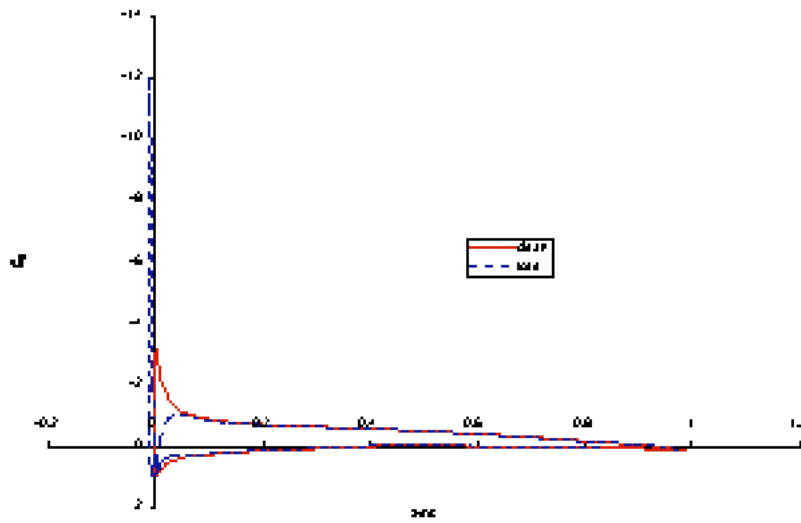


Figure 31: Pressure Coefficient for Example Case 2

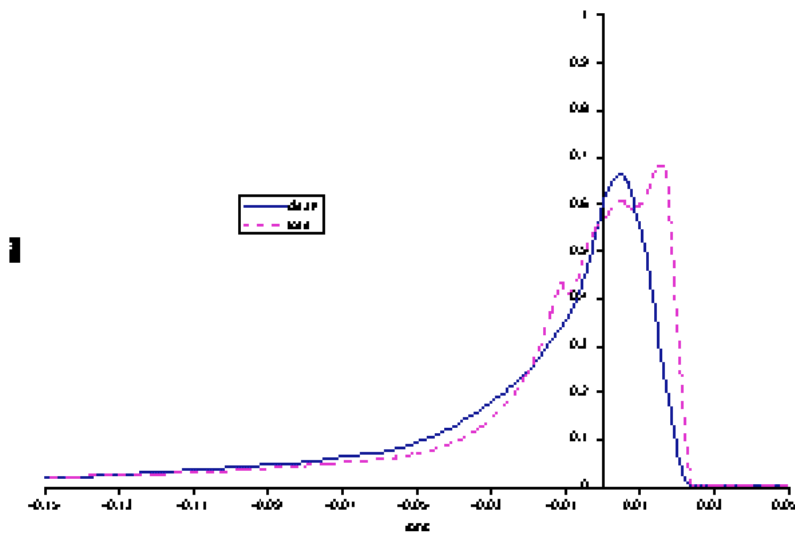


Figure 32: Collection Efficiency for Example Case 2

15.3. Case 3: Run 072504 from Validation Report

Computation Time: Pentium IV 2.4 GHz, 19 s.

Disk Space: 2.7 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 meteorological conditions for this case are identical to Example Case 1. It is interesting to note that the experimental icing limit is less for this case than the first example even though it has a longer exposure time. This effect is a by-product of the variability of ice formations in the tunnel and the variability in the measurement technique. The main input file for this case is listed in Table 46. The main input file and geometry input file are provided on the distribution disk as “**case3.inp**” and “**case3.xyd**” respectively.

Table 45: Main Input File for Example Case 3

Example 3 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

&RDATA

&END

&BOOT

&END

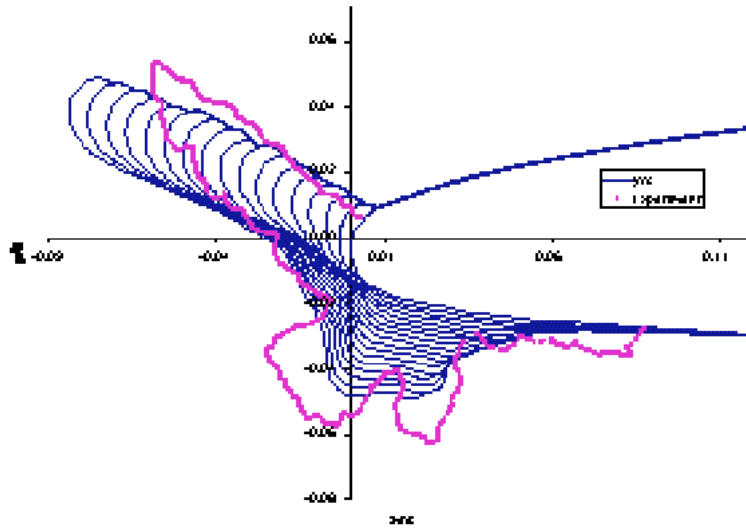


Figure 33: Ice Shape for Example Case 3

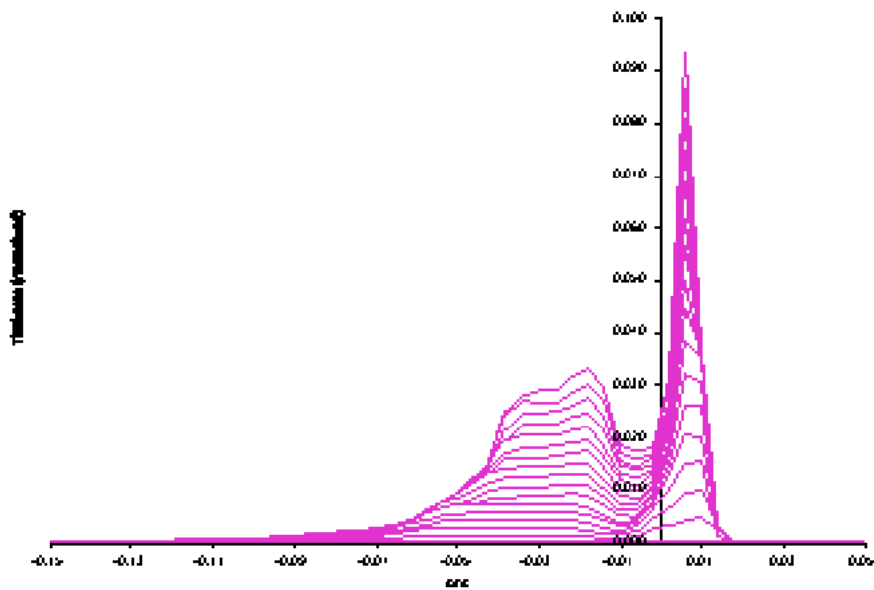


Figure 34: Ice Thickness for Example Case 3

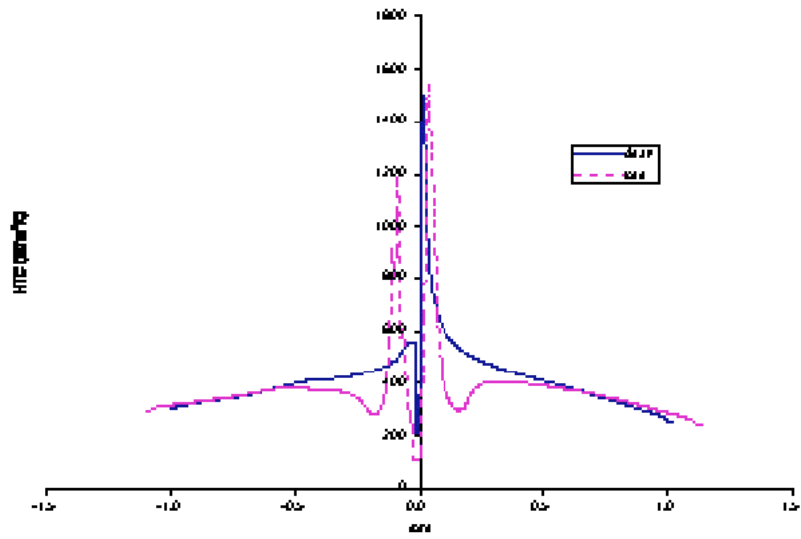


Figure 35: Heat Transfer Coefficient for Example Case 3

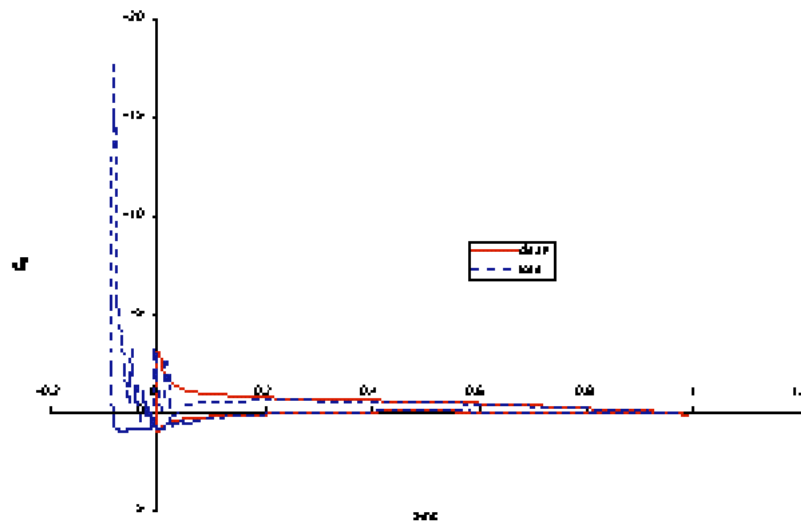


Figure 36: Pressure Coefficient for Example Case 3

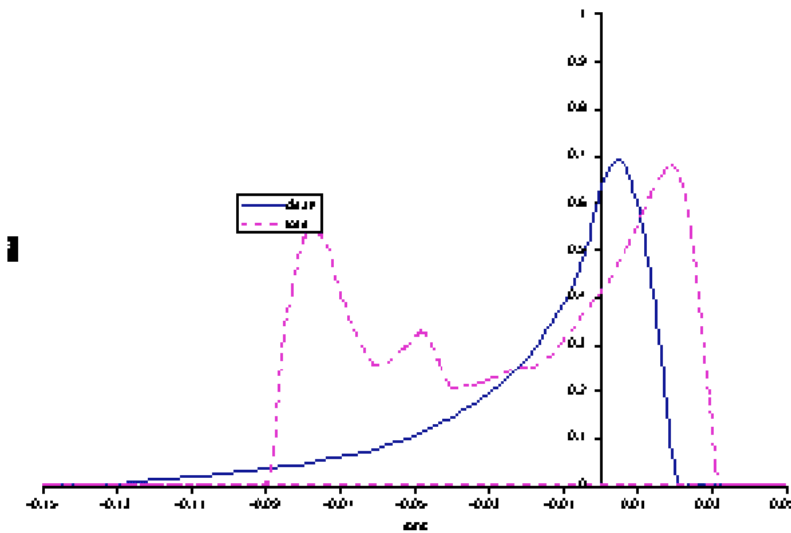


Figure 37: Collection Efficiency for Example Case 3

15.4. Case 4: Run 072504 without automated time step and fewer time steps

Computation Time: Pentium IV 2.4GHz, 6 s.

Disk Space: 640 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 3. 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 3. 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 3. Also note that there is no development of a distinctive lower horn which was observed in Case 3. Table 47 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as “**case4.inp**” and “**case4.xyd**” respectively.

Table 46: Main Input File for Example Case 4

```
Example 4 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
VINP = 90.
LWC = 0.540
TINF = 268.30
PINP = 100000.00
RH = 100.0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
TPRT = 0
&END
&RDATA
&END
&BOOT
&END
```

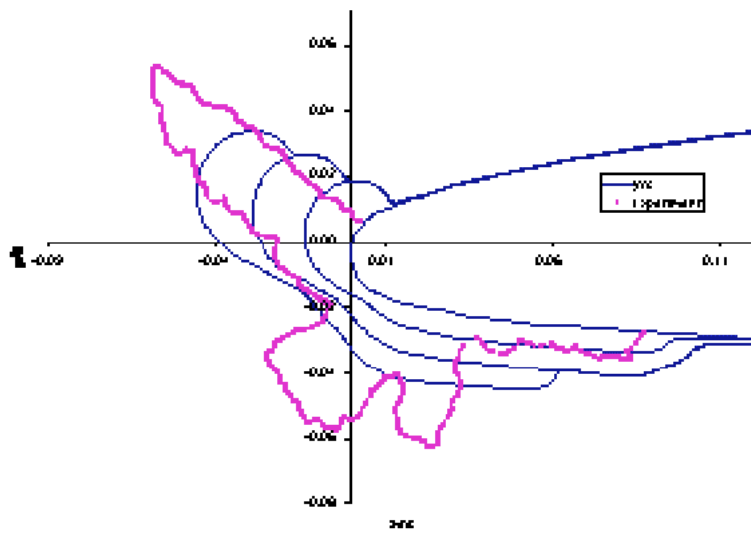


Figure 38: Ice Shape for Example Case 4

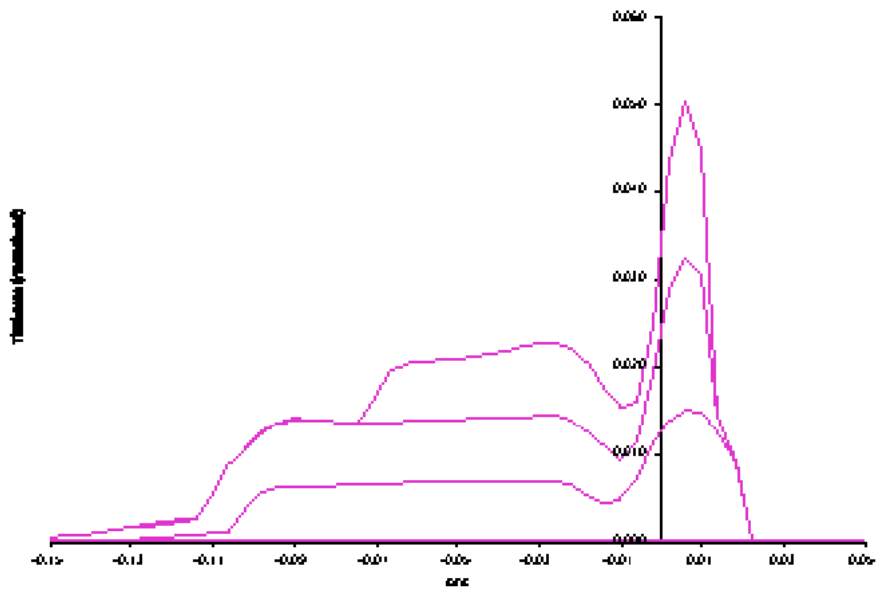


Figure 39: Ice Thickness for Example Case 4

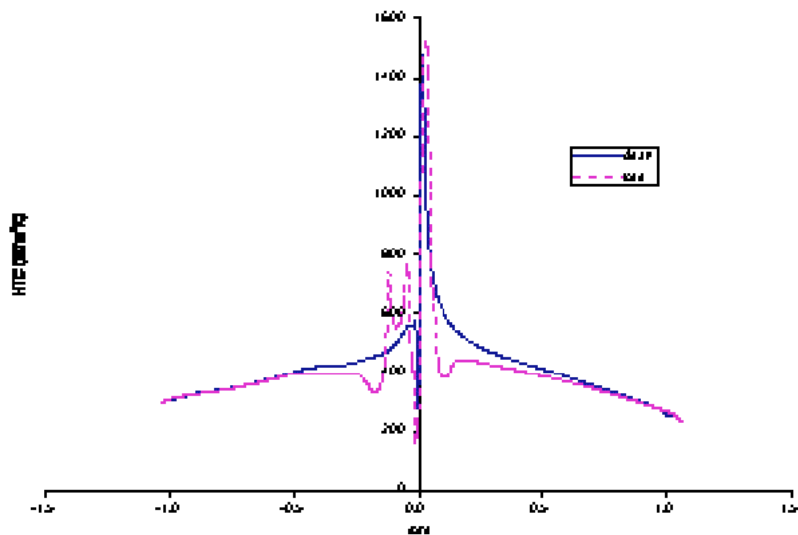


Figure 40: Heat Transfer Coefficient for Example Case 4

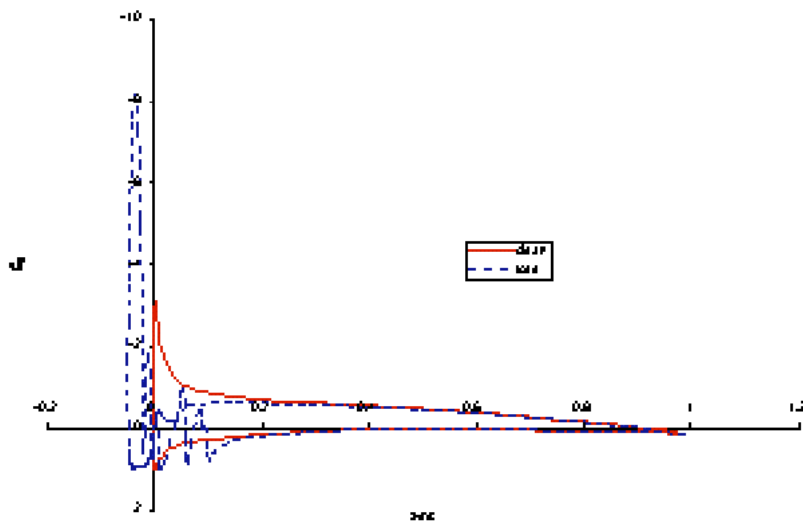


Figure 41: Pressure Coefficient for Example Case 4

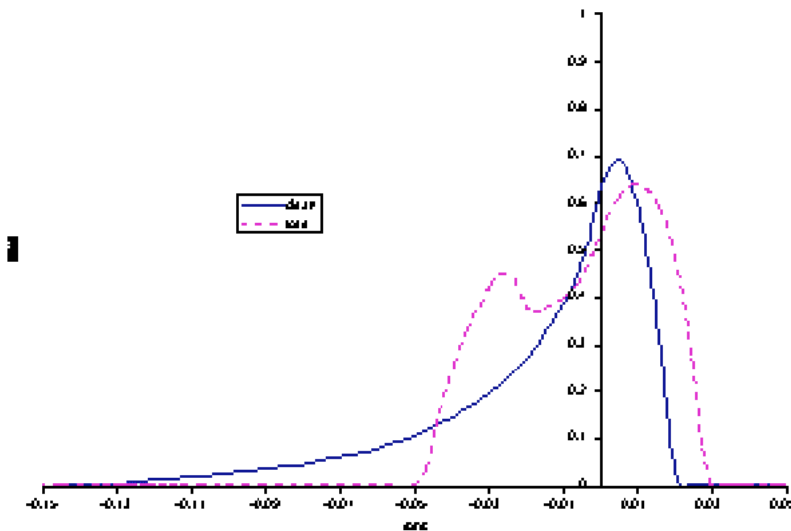


Figure 42: Collection Efficiency for Example Case 4

15.5. Case 5: Run 072504 with Larger Point Spacing

Computation Time: Pentium IV 2.4GHz,9 s.

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 3. 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 3, 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 48 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 47: Main Input File for Example Case 5

```
Example 5 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
  VINP = 90.
  LWC = 0.540
  TINP = 268.30
  PINP = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  TPRT = 0
&END
&RDATA
&END
&BOOT
&END
```

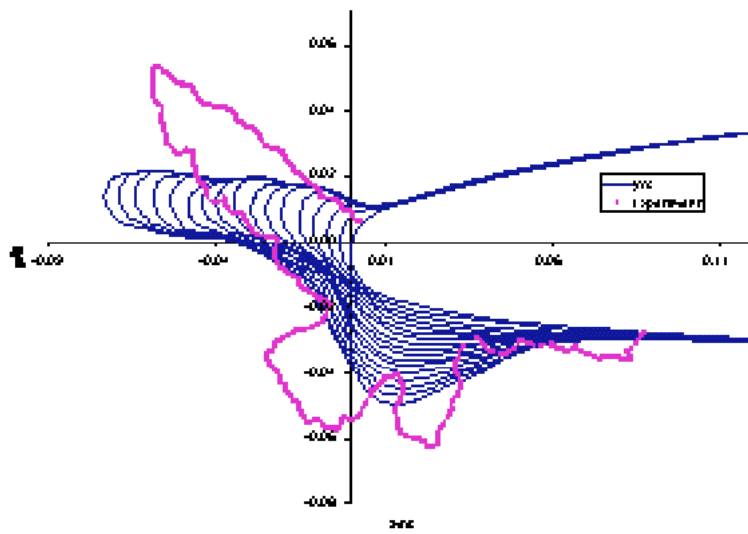


Figure 43: Ice Shape for Example Case 5

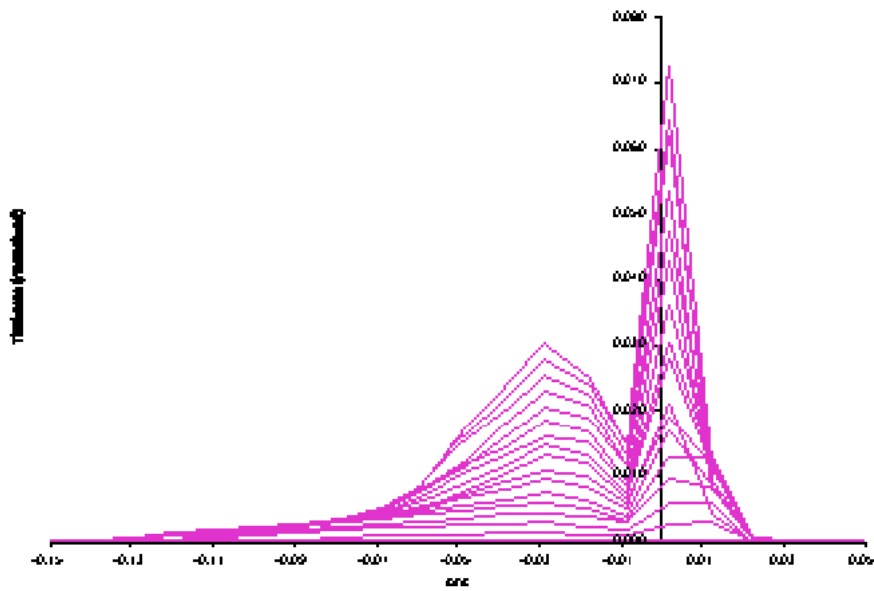


Figure 44: Ice Thickness for Example Case 5

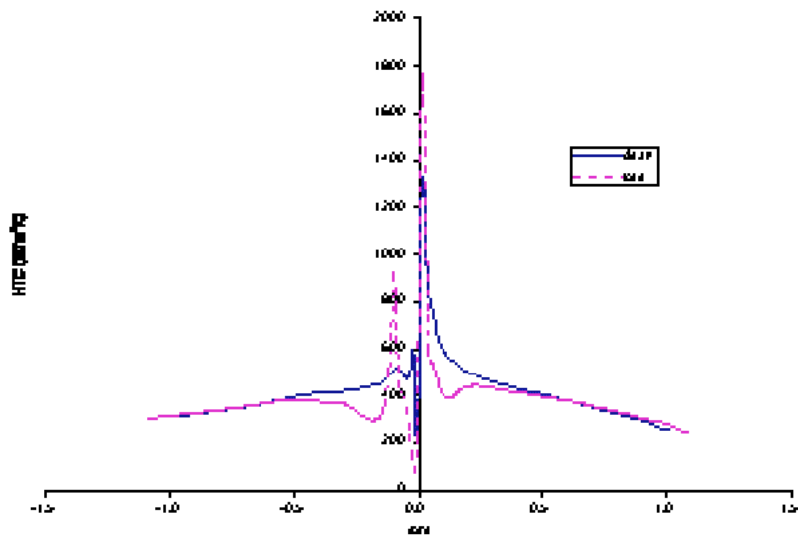


Figure 45: Heat Transfer Coefficient for Example Case 5

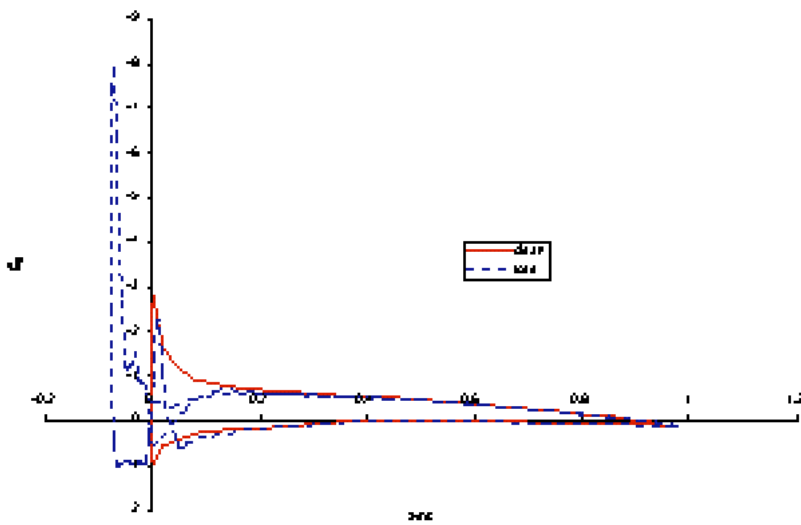


Figure 46: Pressure Coefficient for Example Case 5

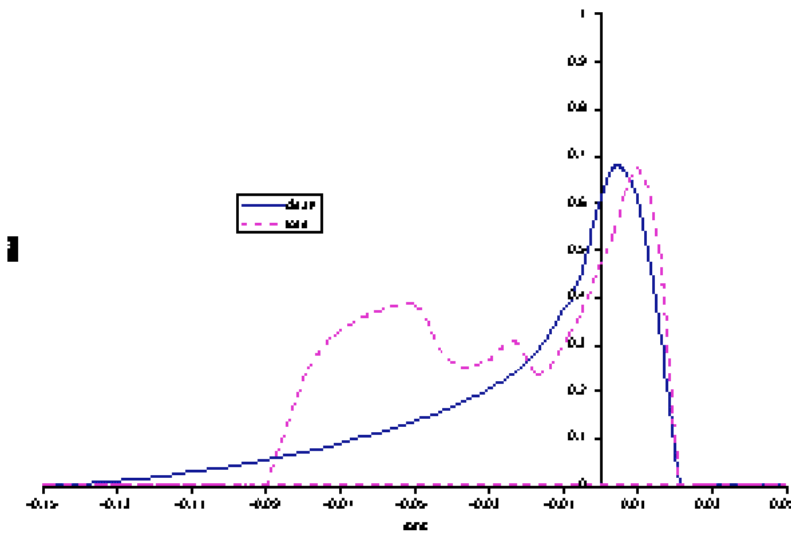


Figure 47: Collection Efficiency for Example Case 5

15.6. Case 6: Cylinder Benchmark Case

Computation Time: Pentium IV 2.4GHz, 27 s.

Disk Space: 3.9 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 longer case computationally and may not be suited for older processors. The execution time is due to the relative size of the ice shape to the cylinder. The small chord size also causes the program to 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 to 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 49. The main input file and geometry input file are provided on the distribution disk as “**case6.inp**” and “**case6.xyd**” respectively. Note that the program will warn the user that the automated time 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 48: Main Input File for Example Case 6

Example 6 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

DPD = 20.

&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

&RDATA

&END

&BOOT
&END

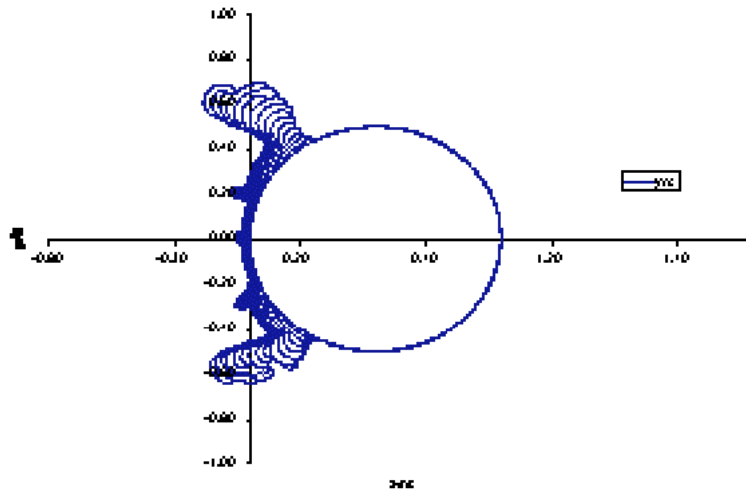


Figure 48: Ice Shape for Example Case 6

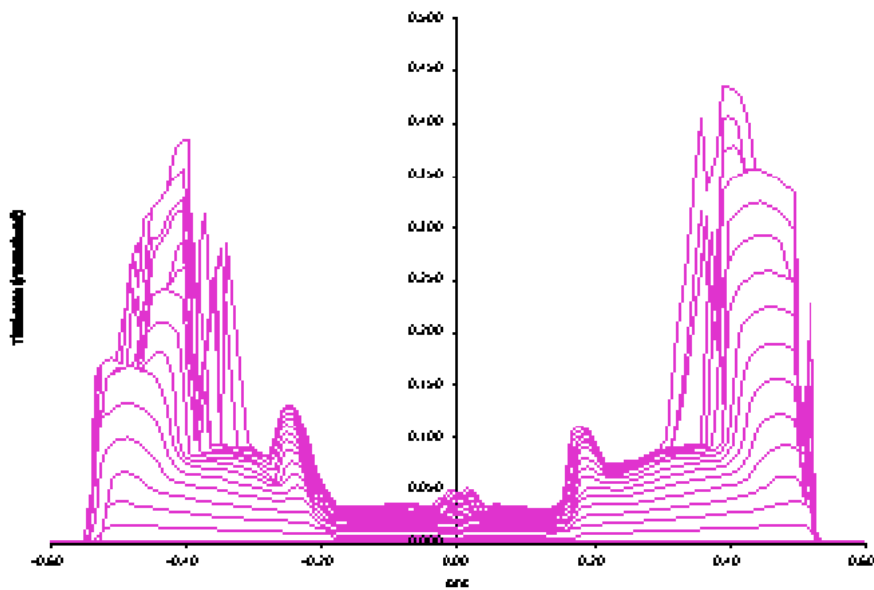


Figure 49: Ice Thickness for Example Case 6

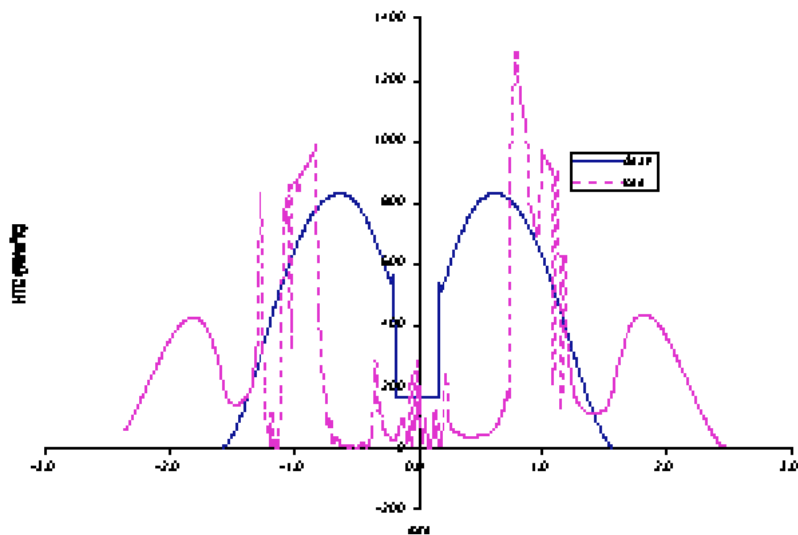


Figure 50: Heat Transfer Coefficient for Example Case 6

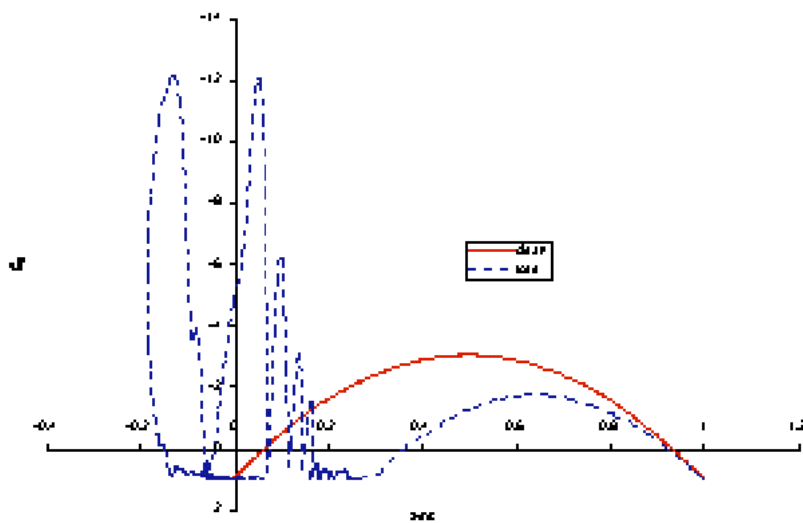


Figure 51: Pressure Coefficient for Example Case 6

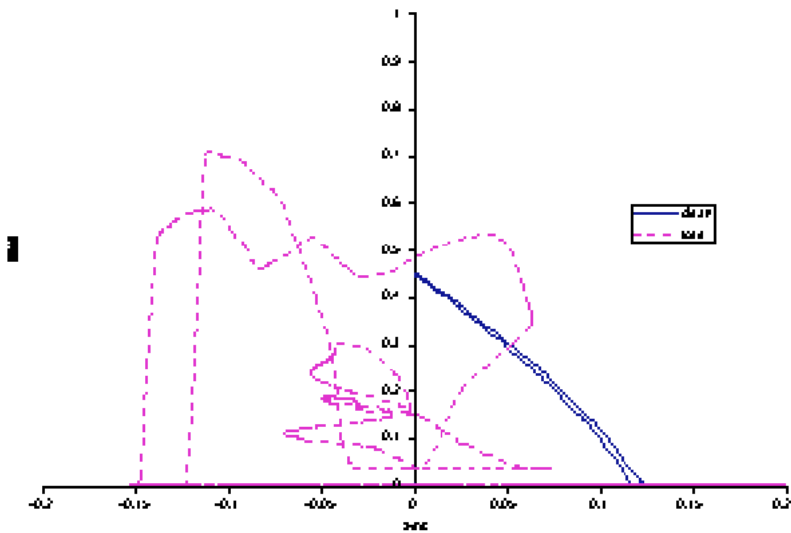


Figure 52: Collection Efficiency for Example Case 6

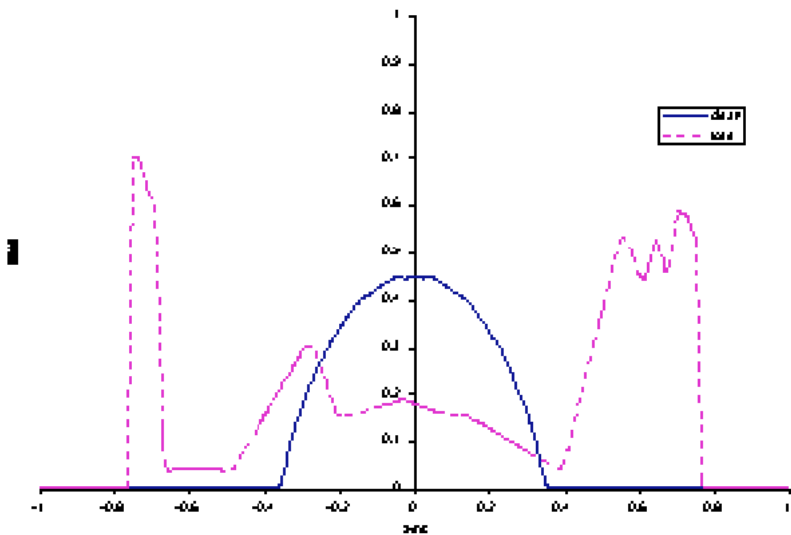


Figure 53: Collection Efficiency as a Function of Wrap Distance for Example Case 6

The plots for collection efficiency and pressure coefficient are quite complex due to the complexity of the ice shape. For this case, it is more appropriate to plot collection efficiency as a function of wrap distance instead of x-distance. This is shown in Figure 53.

The wrap distance plotted in Figure 53 was the wrap distance from stagnation, not the wrap distance from the leading edge. Both wrap distances are listed in the output file.

15.7. Case 7: Exceedence condition

Computation Time: Pentium IV 2.4GHz, 5 s.

Disk Space: 2.9 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 report³. 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 3.2. This is a curious result as water collection tests have shown the opposite trend concerning the prediction of impingement limits²⁵. 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 50 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 49: Main Input File for Example Case 7

Example 7 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
VINFL = 87.2
LWC = 0.82
TINF = 266.85
PINF = 100000.00
RH = 100.0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
EPRT = 1
MPRT = 1
TPRT = 0
&END
&RDATA
&END
&BOOT
&END
```

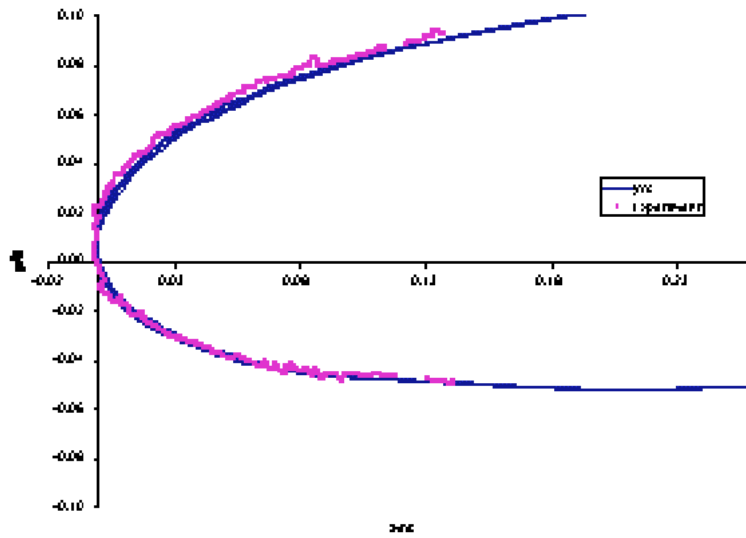


Figure 54: Ice Shape for Example Case 7

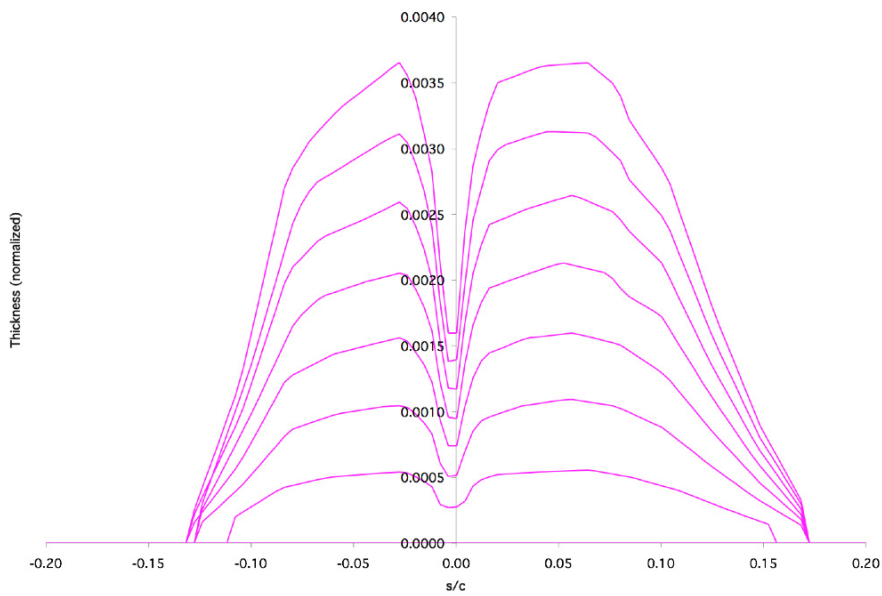


Figure 55: Ice Thickness for Example Case 7

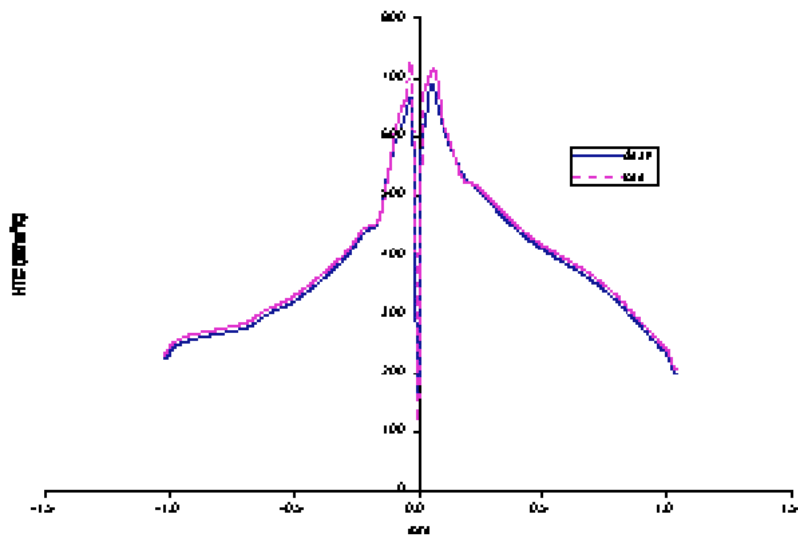


Figure 56: Heat Transfer Coefficient for Example Case 7

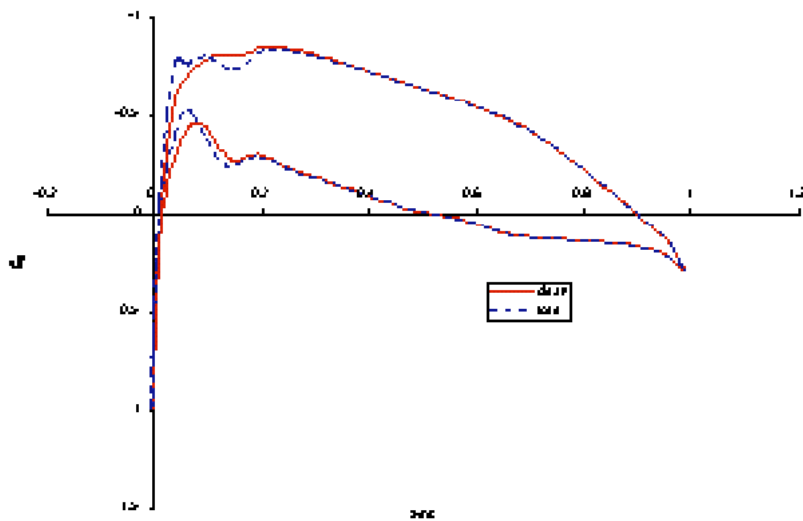


Figure 57: Flow Solution for Example Case 7

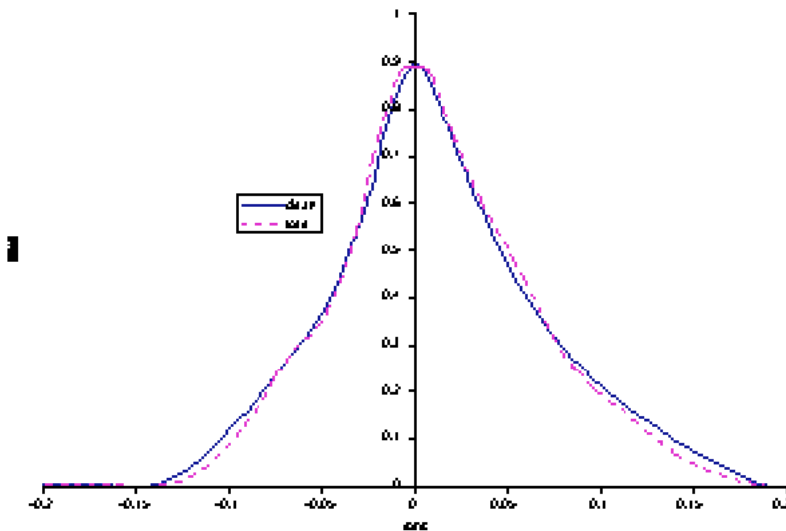


Figure 58: Collection Efficiency for Example Case 7

15.8. Case 8: Exceedence condition with Langmuir ‘D’ distribution

Computation Time: Pentium IV 2.4GHz, 27 s.

Disk Space: 3.4 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 software 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 inch on the lower surface and 3 inches on the upper surface. However the panel spacing is very coarse 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 coarse 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 50 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 50: Main Input File for Example Case 8

```
Example 8 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
&RDATA
&END
&BOOT
&END
```

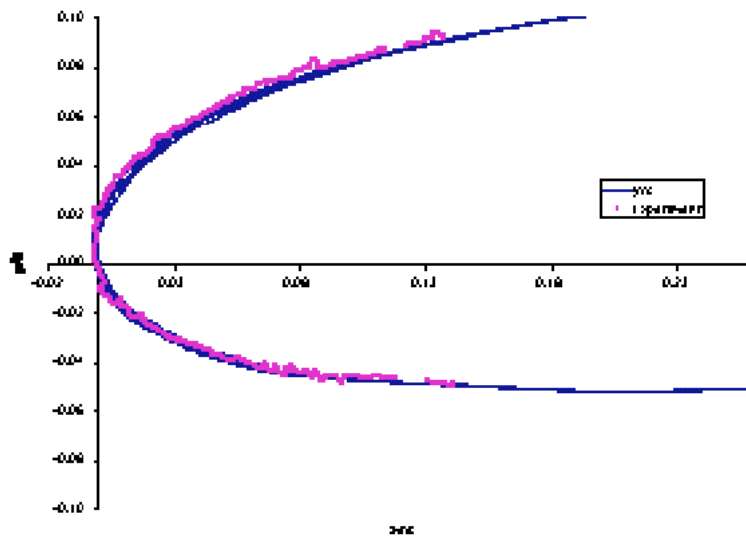


Figure 59: Ice Shape for Example Case 8

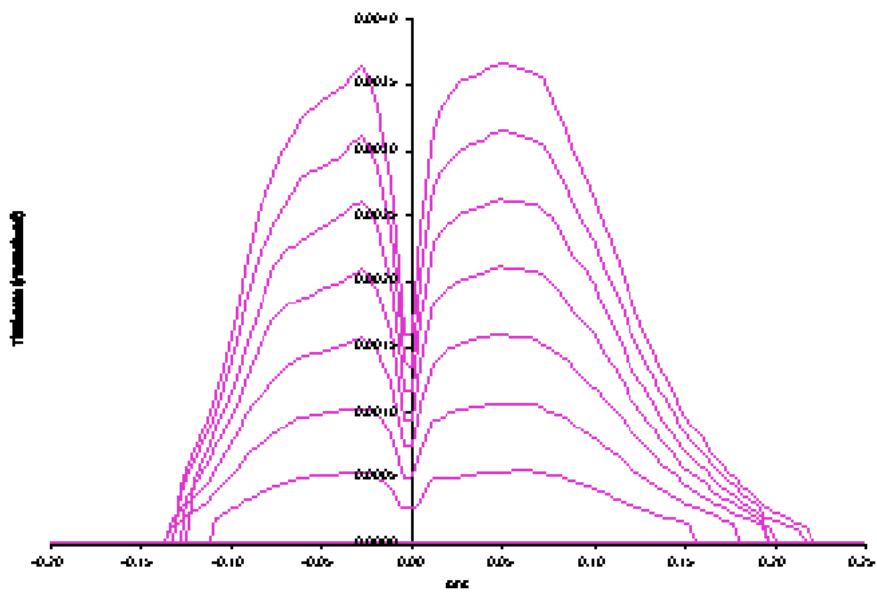


Figure 60: Ice Thickness for Example Case 8

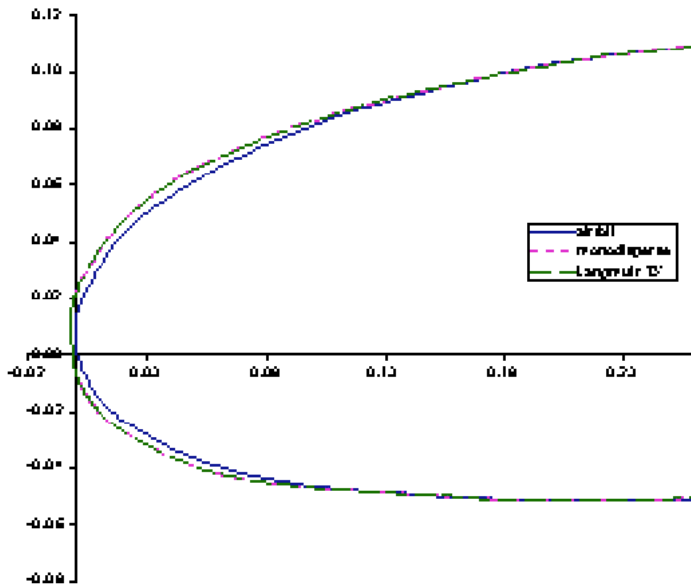


Figure 61: Comparison of Ice Shapes for Monodispersed and Langmuir ‘D’ Cases

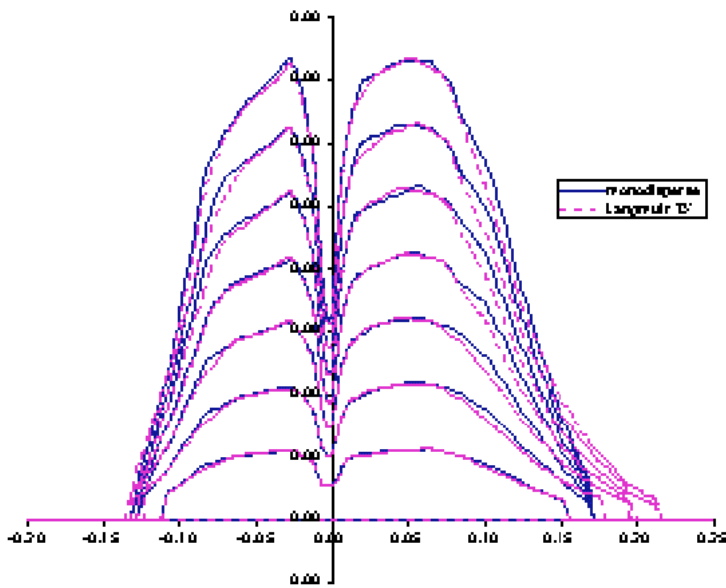


Figure 62: Comparison of Ice Thicknesses for Monodispersed and Langmuir ‘D’ Cases

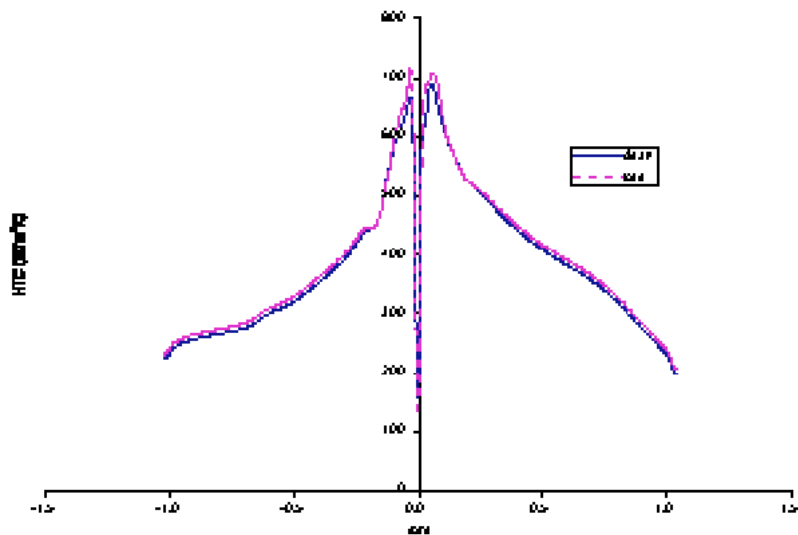


Figure 63: Heat Transfer Coefficient for Example Case 8

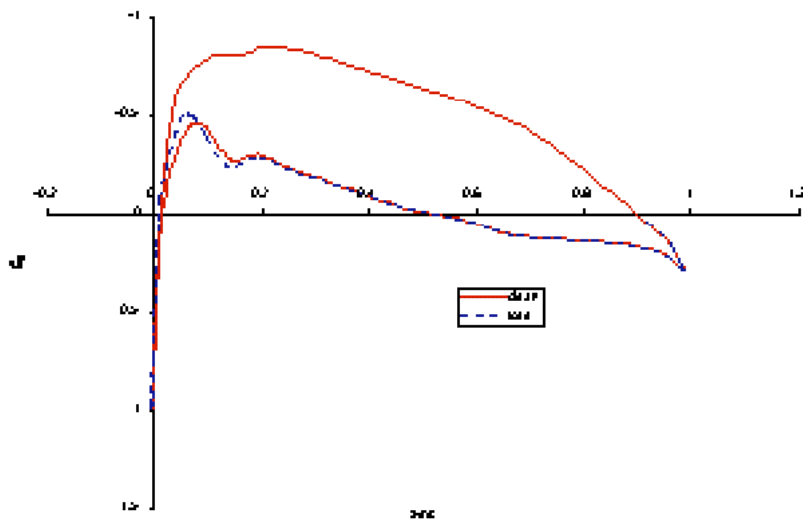


Figure 64: Pressure Coefficient for Example Case 8

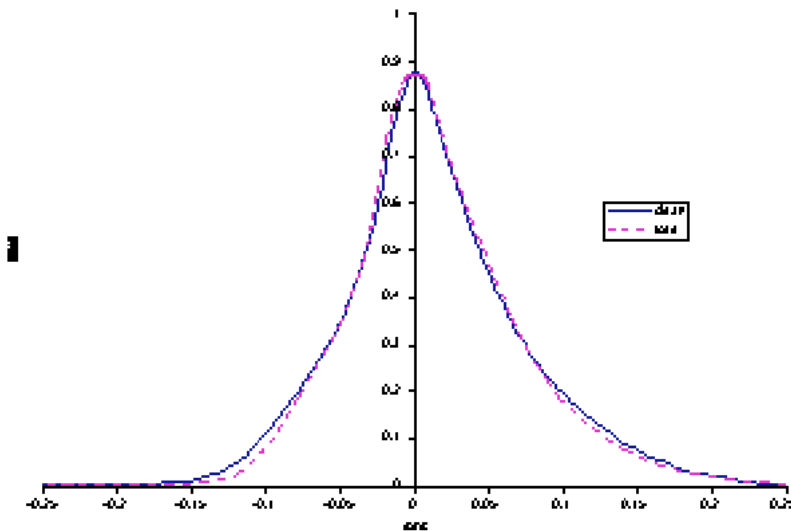


Figure 65: Collection Efficiency for Example Case 8

15.9. Case 9: Exceedence condition with splashing model

Computation Time: Pentium IV 2.4GHz, 48 s.

Disk Space: 1.46 MB

This example case uses the same input as the previous example, except that an empirical model for drop breakup, splashing and re-impingement has been activated. This example was provided to illustrate the effect of this model on a typical exceedence condition ice shape. As with the previous example, the software will print a warning concerning the input drop size. A warning will also appear regarding the use of the splashing model since it was based upon data at velocities below 20 m/s. The user will have to confirm the warnings to continue the case. The collection efficiency plot for this example shows a lesser extent of water collection than the previous example, as expected. The ice shape comparison for this case shows the same amount of ice at the leading edge but a smaller icing limit, 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 2 inches on the lower surface and 4 inches on the upper surface. However the panel spacing is very coarse in this region for both cases. 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 52 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as “**case9.inp**” and “**case9.xyd**” respectively.

Table 51: Main Input File for Example Case 9

```
Example 9 Test Case
&LEW20
```

```
TSTOP = 420.
IBOD = 1
IFLO = 7
DSMN = 4.0D-4
NPL = 24
SLD = 1
&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
VINFL = 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
&RDATA
&END
&BOOT
&END
```

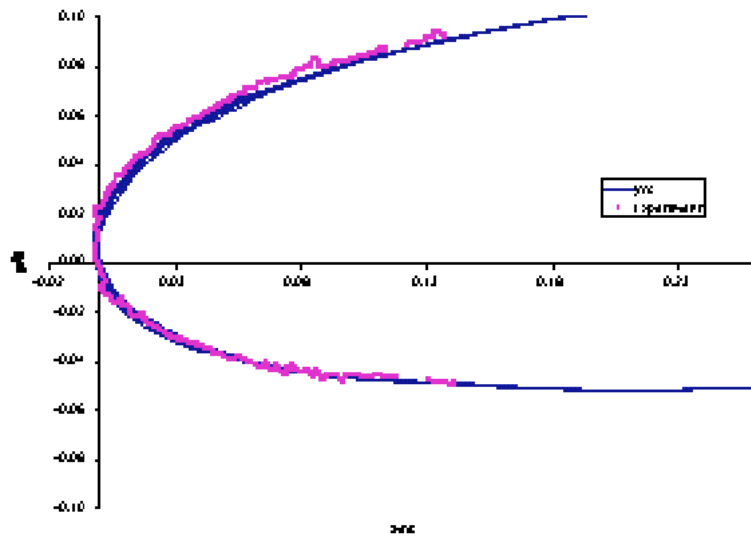


Figure 66: Ice Shape for Example Case 9

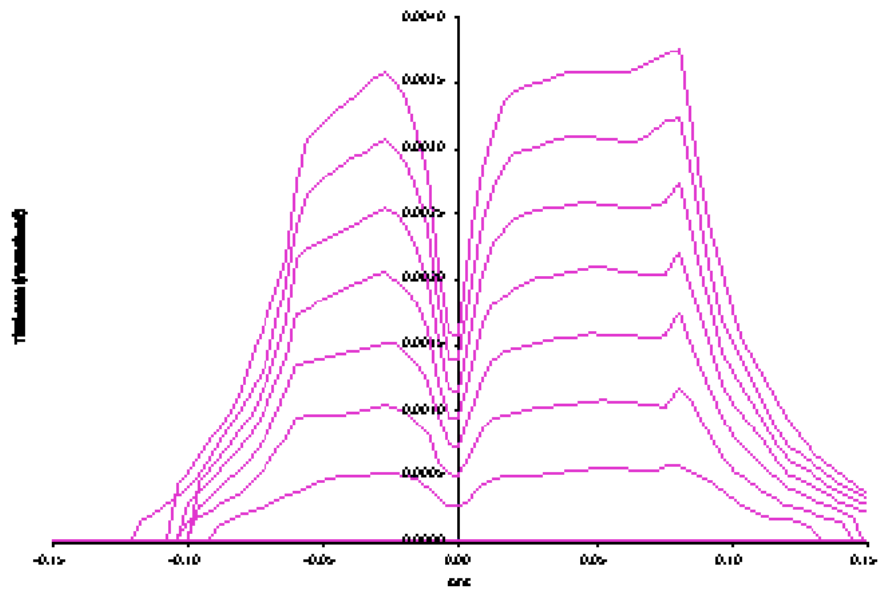


Figure 67: Ice Thickness for Example Case 9

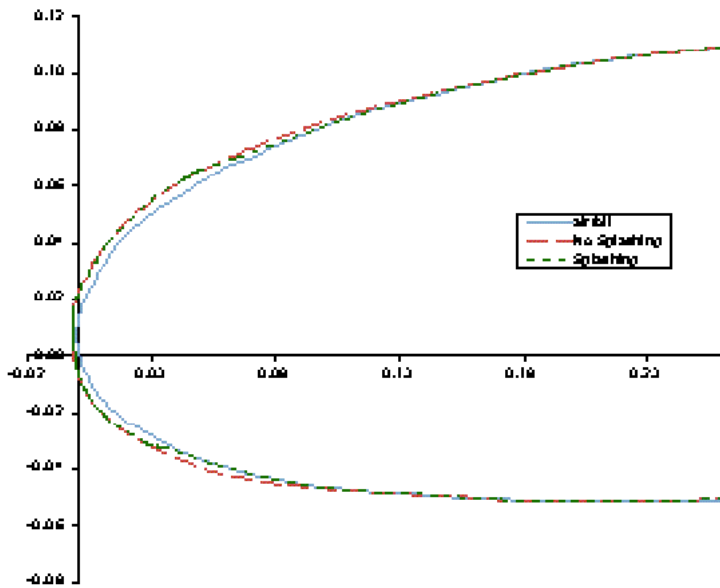


Figure 68: Comparison of Ice Shapes for Splashing and Non-splashing Cases

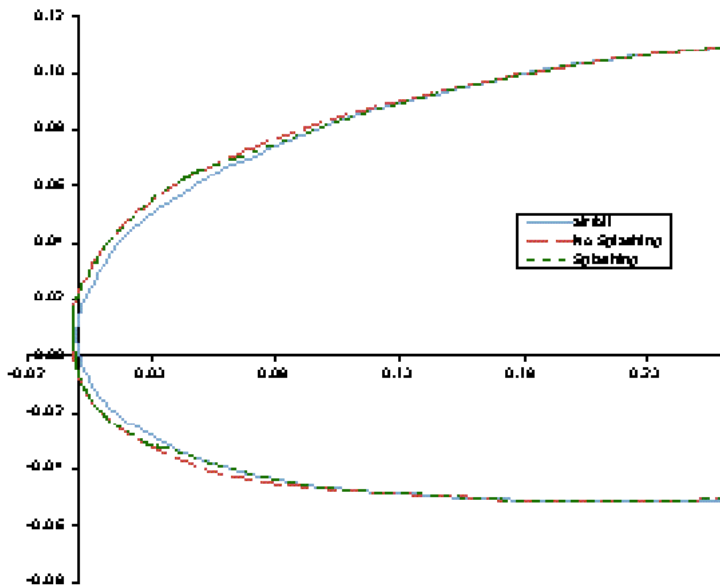


Figure 69: Comparison of Ice Thicknesses for Splashing and Non-splashing Cases

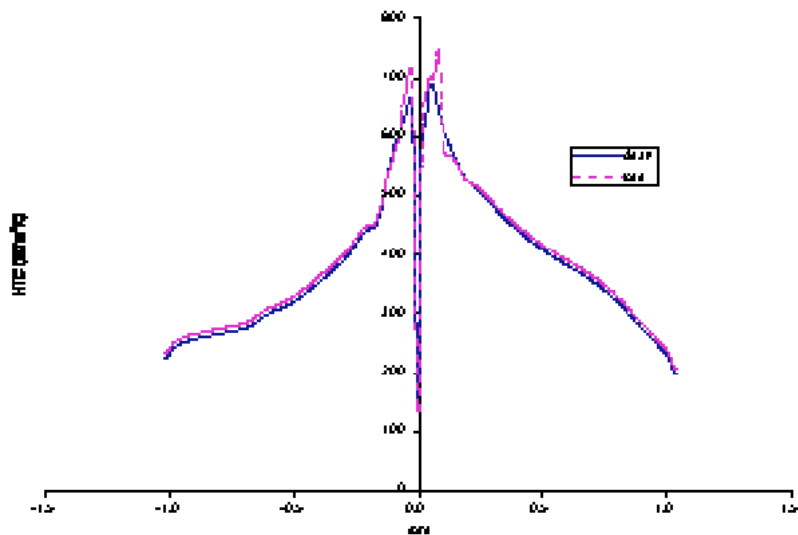


Figure 70: Heat Transfer Coefficient for Example Case 9

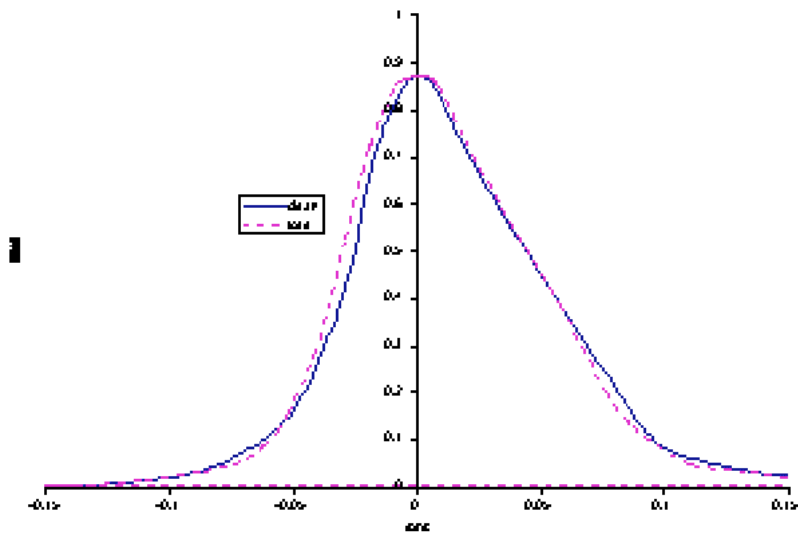


Figure 71: Pressure Coefficient for Example Case 9

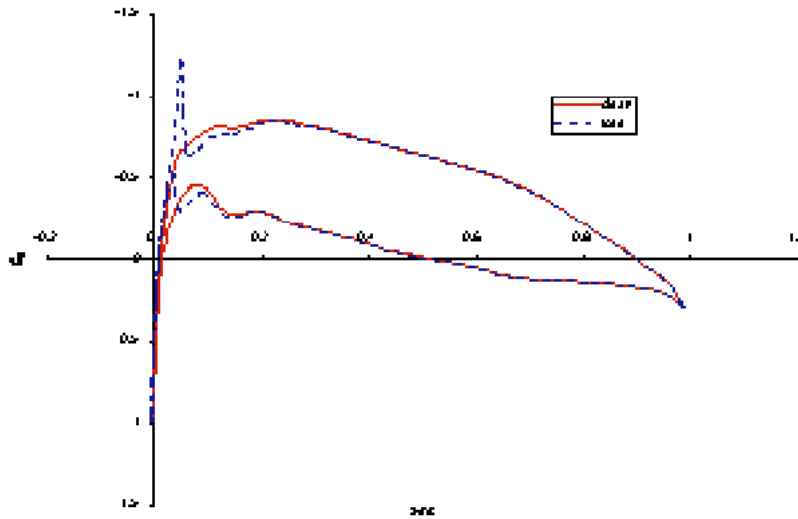


Figure 72: Collection Efficiency for Example Case 9

15.10. Case 10: Comparison of Splashing Model with Impingement Data

Computation Time: Pentium IV 2.4GHz, 32 s.

Disk Space: 900 KB

This example case also illustrates the use of the drop breakup and splashing model. In this example however, a comparison is made between the collection efficiency produced using this model with impingement test data taken in the IRT. The airfoil chosen for this case was the MS-317 airfoil which is used as the reference model for impingement tests. A comparison is also made between the collection efficiency predicted with and without splashing. As with the previous example, the software will print a warning concerning the input drop size. Since impingement tests are performed at above freezing temperatures, a warning will appear due to the temperature selected. A warning will also appear regarding the use of the splashing model since the model was based upon data at velocities below 20 m/s. The user will have to confirm the warnings to continue the case. The collection efficiency plot for this example shows a lesser extent of water collection than the run where splashing was not considered. The collection efficiency predicted with the new model is much closer to the experimental data. This is to be expected since the empirical splashing model employed was adjusted to take into account the effects shown in the IRT impingement data. Since the ambient temperature is above freezing for this case, no ice shape was generated. Table 52 shows the main input file for this case. The main input file and geometry input file are included on the distribution disks as “**case10.inp**” and “**case10.xyd**” respectively.

Table 52: Main Input File for Example Case 10

```
Example 10 Test Case
&LEW20
```

```
ITIMFL = 1
TSTOP = 60.
IBOD = 1
IFLO = 1
DSMN = 2.0D-4
NPL = 24
SLD = 1
&END
&DIST
FLWC = 0.010, 0.064, 0.106, 0.073, 0.135, 0.164, 0.149, 0.103, 0.130, 0.065
DPD = 10.9, 21.8, 31.8, 43.8, 69.2, 98.2, 126.7, 157.8, 196.3, 241.4
&END
&ICE1
CHORD = 0.9144
AOA = 0.
VINP = 78.68
LWC = 0.47
TINF = 293.15
PINP = 89232.3
RH = 70.0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
TPRT = 0
IDBF = 1
&END
&RDATA
&END
&BOOT
&END
```

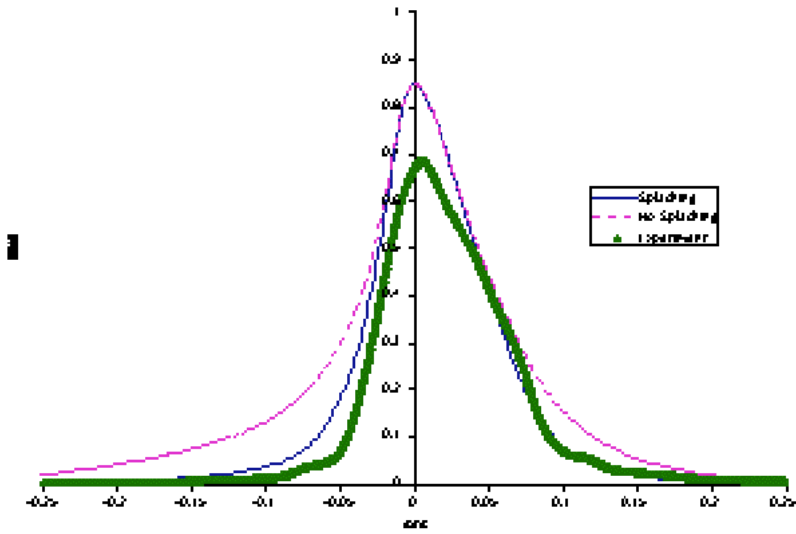


Figure 73: Collection Efficiency for Example Case 10

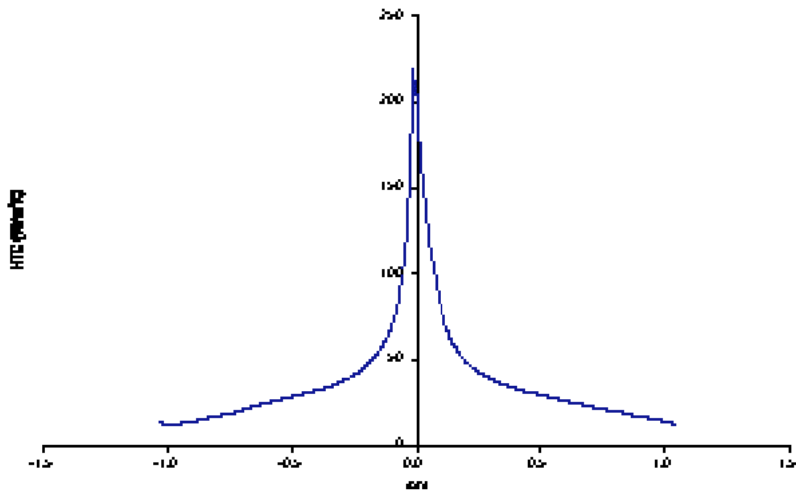


Figure 74: Heat Transfer Coefficient for Example Case 10

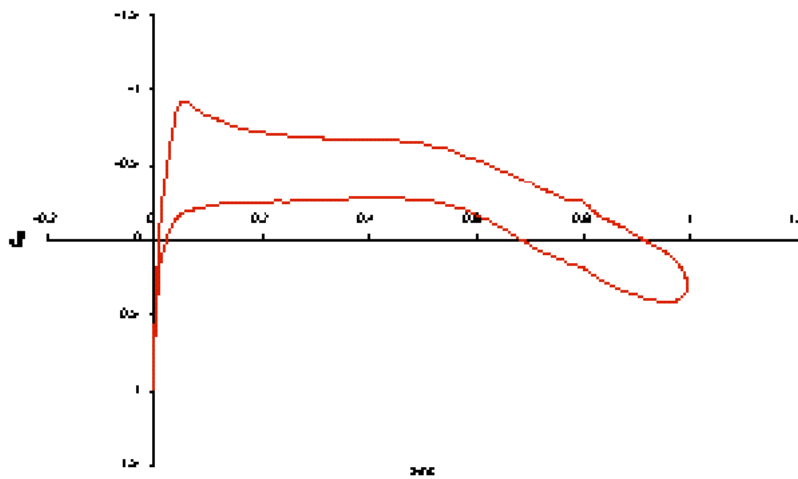


Figure 75: Pressure Coefficient for Example Case 10

15.11. Case 11: Three body example

Computation Time: Pentium IV 2.4GHz, 49 s.

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 53: Main Input File for Example Case 11

Example 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
  VINFL  = 88.5
  LWC    = 0.60
  TINFL  = 268.15
  PINFL  = 100000.00
  RH     = 100.0
&END
&LPRNT
  FPRT   = 1
  HPRT   = 1
  BPRT   = 1
  TPRT   = 0
&END
&RDATA
&END
&BOOT
```

&END

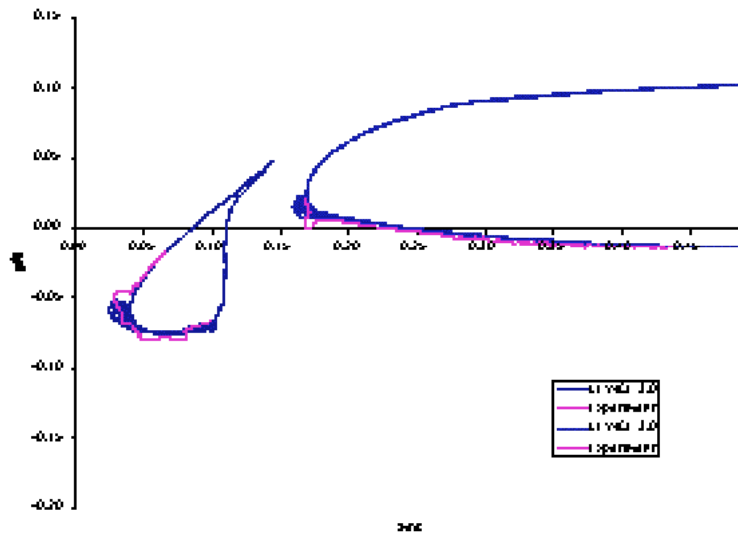


Figure 76: Ice Shape on Slat and Main Elements for Example Case 11

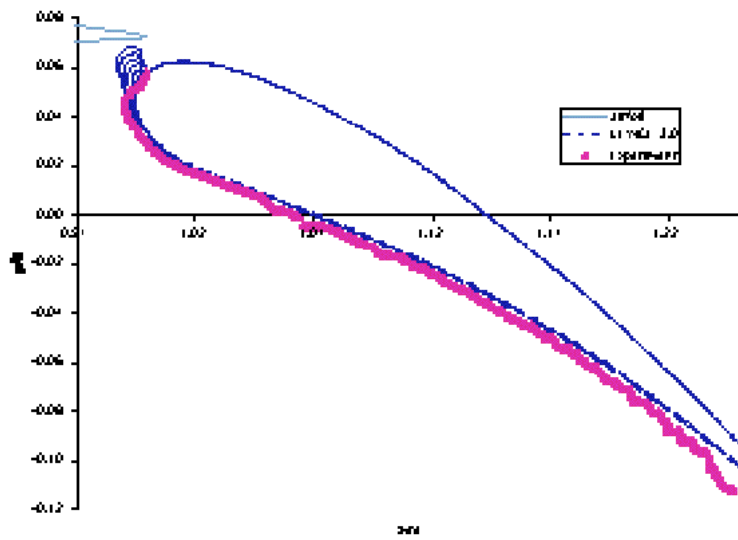


Figure 77: Ice Shape on Flap for Example Case 11

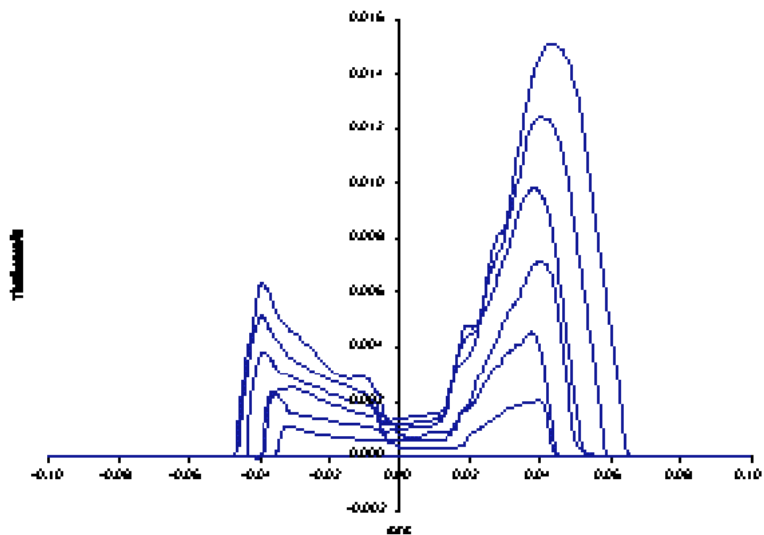


Figure 78: Ice Thickness on Slat for Example Case 11

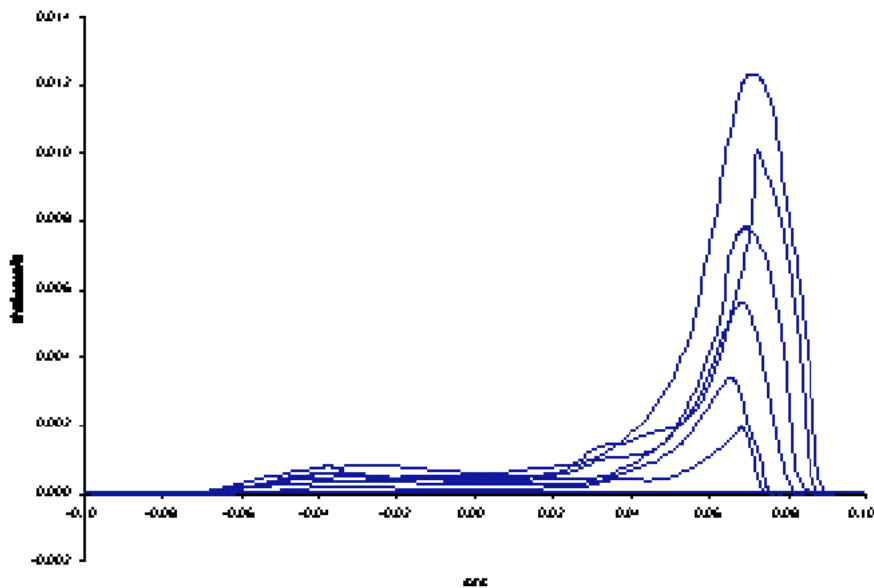


Figure 79: Ice Thickness on Main Element for Example Case 11

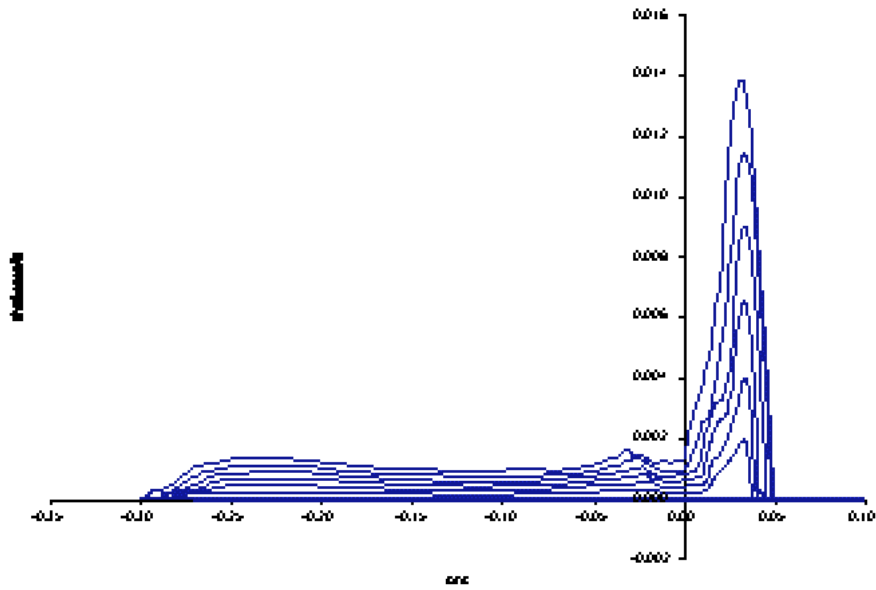


Figure 80: Ice Thickness on Flap for Example Case 11

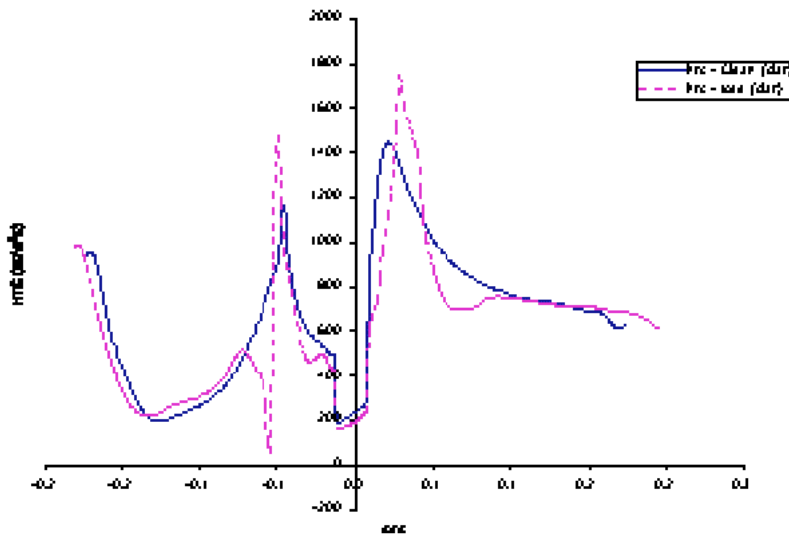


Figure 81: Slat Heat Transfer Coefficient for Example Case 11

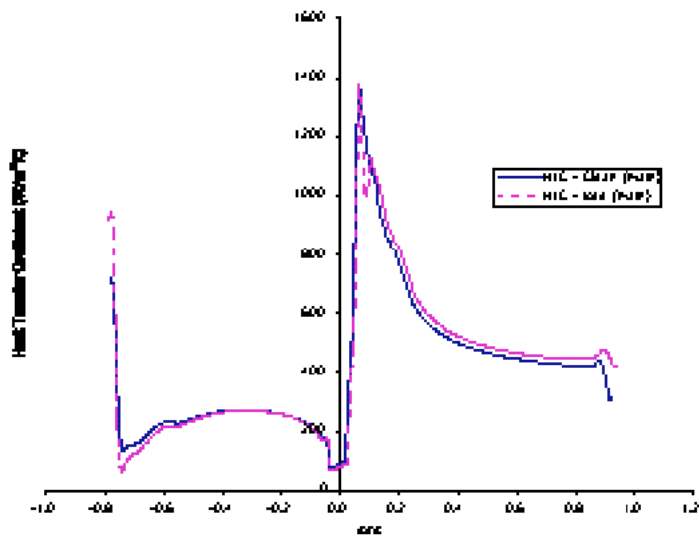


Figure 82: Main Heat Transfer Coefficient for Example Case 11

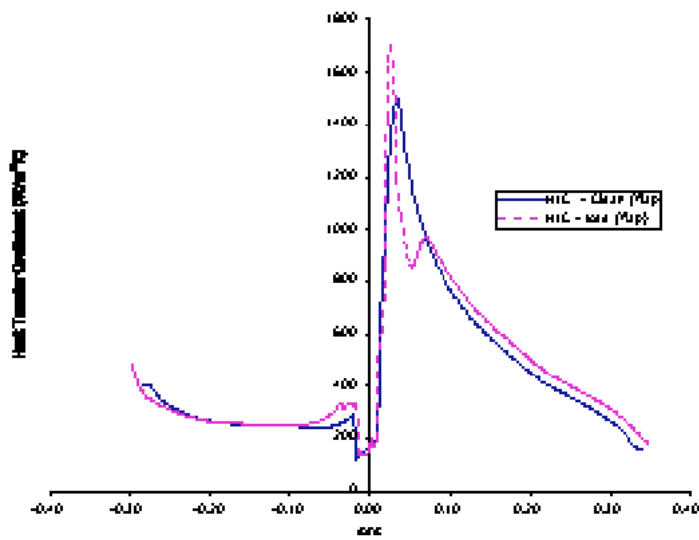


Figure 83: Flap Heat Transfer Coefficient for Example Case 11

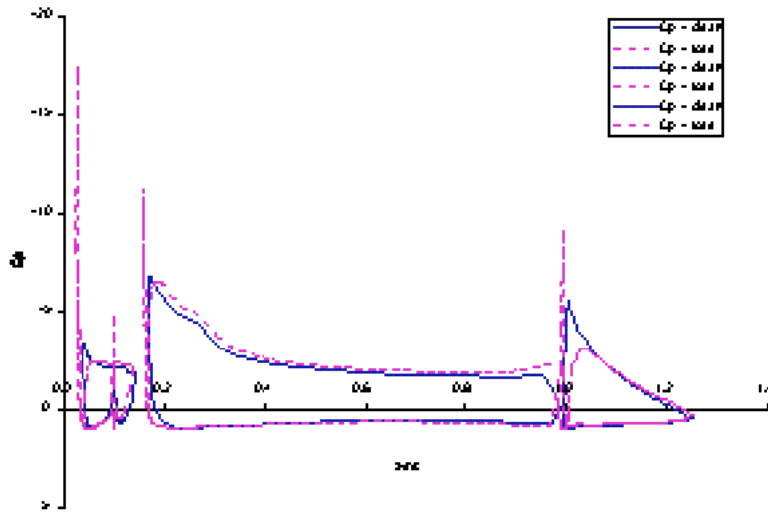


Figure 84: Pressure Coefficient for Example Case 11

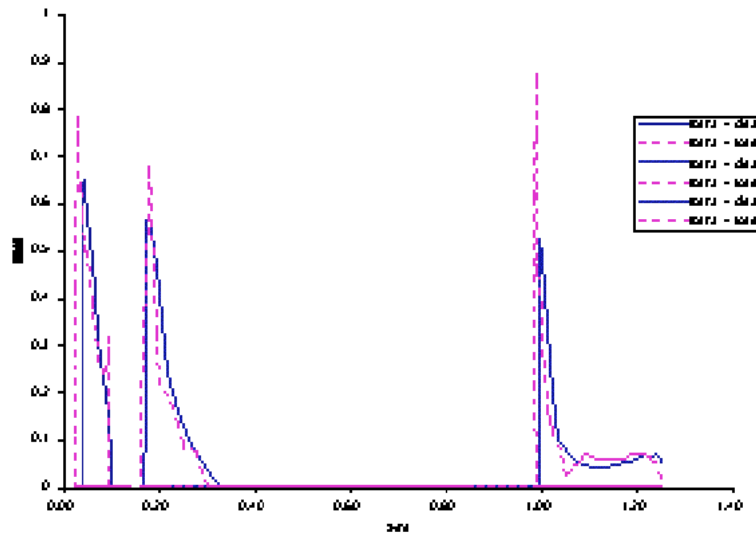


Figure 85: Collection Efficiency for Example Case 11

15.12. Case 12: Pneumatic boot case with freezing rain distribution.

Computation Time: Pentium IV 2.4GHz, 4 min. 13 s.

Disk Space: 7.6 MB

This example case utilized a simplistic pneumatic boot model that has been added to LEWICE 3.2. In this model, the user selects $IBOOT = 1$ and then specifies the residual ice height

and the upper and lower boot limits in the BOOT namelist. Note that the user specifies residual height, thus this module does not predict how well the boot performs. The model is simply a means of generating residual ice shapes past the boot limits. The drop size distribution for this case has been suggested as a freezing rain case for Appendix X. The boot limits were set to $\pm 7.5\%$ chord with a zero residual height. Since this model does not predict the effectiveness of a boot, the user is warned when this option is selected and this warning must be confirmed for the case to run.

Table 54 shows the main input file for this case. The main input file and geometry input file are included on the distribution disk as “**case12.inp**” and “**case12.xyd**” respectively. Figure 86 shows the residual ice shape compared to the ice shape without a boot. Since the meteorological conditions are very extreme for this case, a substantial ice ridge appears behind the boot on the lower surface. The ice height of this ridge is comparable to the lower horn ice height of the original ice shape, as shown in the ice thickness plot. The ridge height is larger than the original ice height at the ridge location since the water will collect at a higher rate as the ridge shape grows, as shown in Figure 90.

Note: No validation has been performed using this feature.

Table 54: Main Input File for Example Case 12

Example 12 Test Case

```

&LEW20
ITIMFL = 0
TSTOP = 2700
  IBOD = 1
  IFLO = 45
  DSMN = 4.0D-4
  NPL = 24
  IBOOT = 1
&END
&DIST
FLWC = 0.030, 0.133, 0.152, 0.048, 0.083, 0.354, 0.103, .067, .030
DPD = 4, 17, 72, 235, 345, 560, 895, 1120, 1405
&END
&ICE1
CHORD = 3.96
  AOA = 6.0
  VINP = 76.8
  LWC = 0.47
  TINP = 268.15
  PINP = 89232.3
  RH = 100.0

```

```

&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  TPRT = 0
  MPRT = 1
  EPRT = 0
&END
&RDATA
&END
&BOOT
XBOOTUP = 0.075
XBOOTLOW = 0.075
HRES = 0.00
&END

```

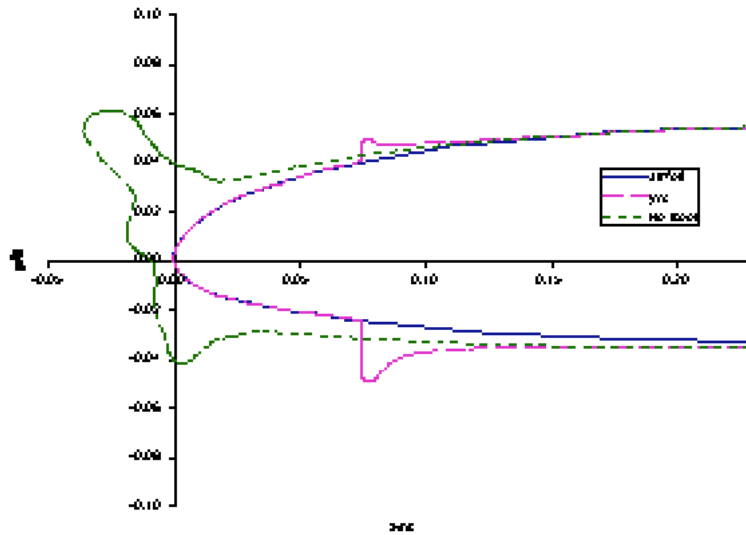


Figure 86: Ice Shape With and Without Boot for Example Case 12

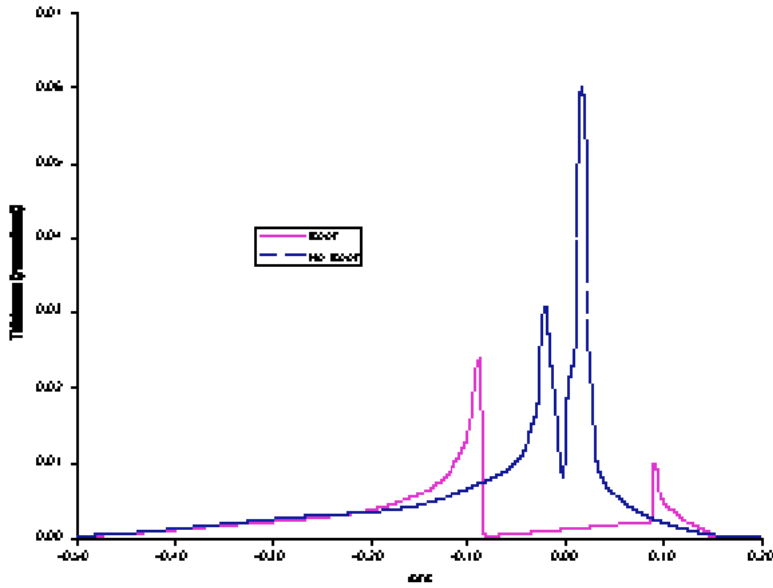


Figure 87: Ice Thickness With and Without Boot for Example Case 12

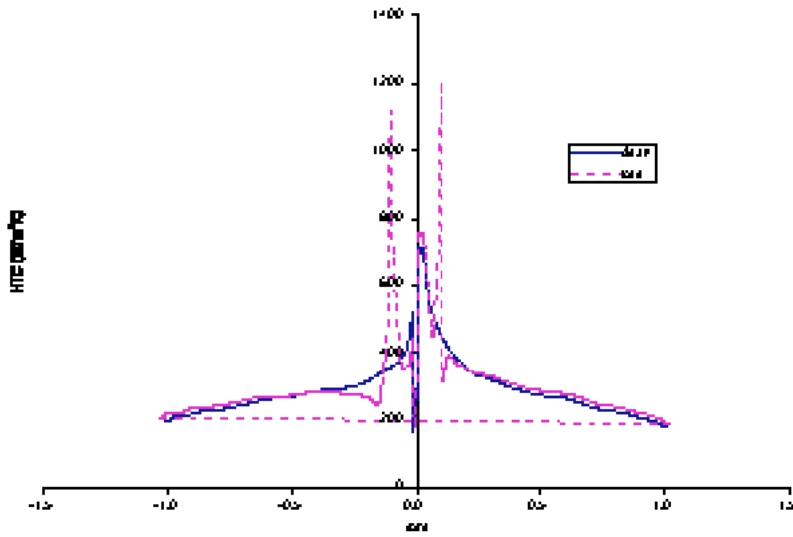


Figure 88: Heat Transfer Coefficient for Example Case 12

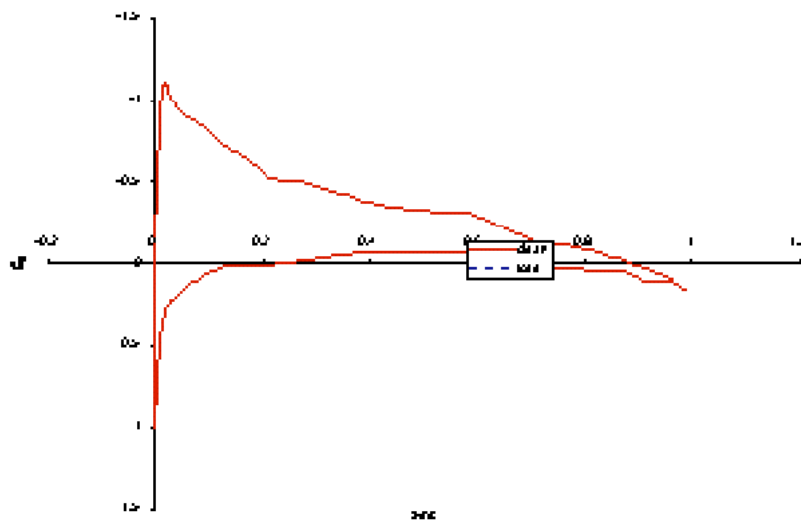


Figure 89: Pressure Coefficient for Example Case 12

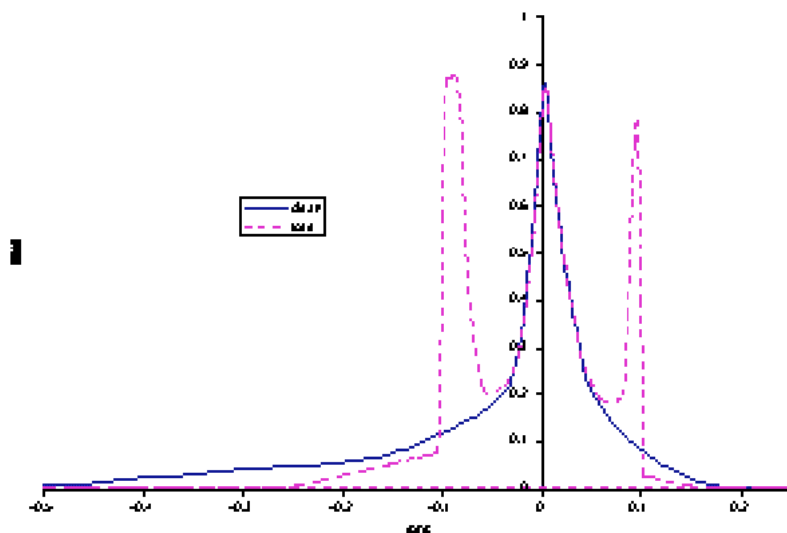


Figure 90: Collection Efficiency with Pneumatic Boot for Example Case 12

15.13. Case 13: First benchmark conditions with an evaporative hot air anti-icer.

Computation Time: Pentium IV 2.4GHz, 2 s.

Disk Space: 1.8 MB

This example case illustrates the use of the IDEICE = 1 anti-ice capabilities of LEWICE. As the procedure used in LEWICE for IDEICE = 1 anti-icing is less involved than the procedure

used for the IDEICE = 2 to IDEICE = 4 options 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 because the IDEICE = 1 option is being used. The conditions for this case are the same as Example Case 15, except the de-icer flag has been set to IDEICE = 1. 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 55 shows the main input file for this case while Table 56 shows the de-ice input file. The main input file and geometry input file are included on the distribution disks as “**case13.inp**” and “**case13.xyd**” respectively. The de-ice input file is listed on the distribution disk as “**case13d.inp**”. This case shows an example of an evaporative anti-icer using bleed air. Since the input flag states that all of the water must evaporate, the wattages shown are very high. Wattages in this range should be expected for evaporative systems. The useful outputs from “**noice.dat**” in this example are the heat requirements and the surface temperature. As shown in Figure 93, the bleed air temperature solution will be overpredicted due to the simplistic assumptions used. The input flag IDEICE should be set to 3 to predict bleed air temperatures for this case. The user could also consider the IEVAP = 2 option for this case. Another output from “**noice.dat**” which may be useful is the effective heat transfer coefficient, h_{eff} . The effective heat transfer coefficient can be used to transfer information from LEWICE to other programs. It is the “effective” value which would need to be input as the convective boundary condition for many commercial or non-commercial CFD programs in order to thermally account for icing effects. Other outputs for this case such as heat transfer coefficient, collection efficiency and pressure coefficient are the same as shown in Example 15 and are not shown again here.

Note: The full de-ice input file is read even though the program will only perform a one dimensional steady-state analysis.

Note: The de-ice input file name no longer is fixed as “**deice.inp**” as it was in version 2.0.

Table 55: Main Input File for Example Case 13

Example 13 Test Case

```
&LEW20
ITIMFL = 0
TSTOP = 600
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 = 0.
VINFL = 44.7
LWC = 0.780
TINF = 265.50
PINFL = 100000.00
RH = 100.0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
EPRT = 1
MPRT = 1
TPRT = 0
IDBF = 1
&END
&RDATA
&END
&BOOT
&END

```

Table 56: De-ice input file for Example Case 13

```

1  nx
2  9
# of length conductivity diffusivity anisotropy slope (b) of
nodes (m) (W/m/K) (m**2/s) factor temp. eqn.
C  substrate(aluminum)
25  1.750d-3  176.530d0  4.260d-5  1.d0  0.d0
C  ice
61  2.540d-3  2.232d0  1.151d-6  1.d0  0.d0
# of length cond. diff. anis. slope of add. layer
nodes (m) (W/m/K) (m**2/s) temp. eqn. thick. number
C  gap (no heat input)
10  0.9144d0  176.56d0  4.260d-5  1.d0  0.d0  0.d0  1
C  heater G
14  3.175d-2  176.56d0  4.260d-5  1.d0  0.d0  0.d0  1

```

```

C heater E
21 2.540d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C heater C
21 2.540d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C parting strip - heater A
36 1.905d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C heater B
21 2.540d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C heater D
21 2.540d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C heater F
14 3.175d-2 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1
C gap (no heat input)
10 0.9144d0 176.56d0 4.260d-5 1.d0 0.d0 0.d0 1

```

ijde offset ipar

```
001 0.000 1
```

C	Q	TON	TOFF	TLAG	ICFLAG
	heater density	time on	time off	lag time	temp cntrl.
	(kWatts/m**2)	(sec)	(sec)	(sec)	flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	00.00d0	600.d0	0.d0	0.d0	0
C	heater E				
	0.00d0	600.d0	0.d0	0.d0	0
C	heater C				
	00.00d0	600.d0	0.d0	0.d0	0
C	parting strip (heater A) does not turn off				
	00.00d0	600.d0	0.d0	0.d0	0
C	heater B				
	00.00d0	600.d0	0.d0	0.d0	0
C	heater D				
	00.00d0	600.d0	0.d0	0.d0	0
C	heater F				
	00.00d0	600.d0	0.d0	0.d0	0
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0

ibc1 ibc2 ibc3 ibc4

```

2    2    2    2
  tg1    tg2    tg3    tg4
268.3  268.3  268.3  268.3
  h1     h2     h3     h4
 3.d0   3.d2   0.d0   0.d0
  qw1    qw2    qw3    qw4
 0.d0   0.d0   0.d0   0.d0
  tairh  airmd  airup  xairmd
   5.d2  0.1d0  0.5d0  0.d0
  nhole  dhole  dtube  rdist  ycase  xuhnd  xlhnd
   3    0.00132  0.0381  50.    0.0    0.23  0.26
  icase  hconst  rep    rp    zp  rrp  dp
   0    1.0  0.76  1.394  1.    0.    0.
  xhole(m)  yhole(m)  zdist(m)
.    0.0176    -0.033    0.0107
   0.000    0.000    0.0091
   0.0176    -0.033    0.0076
  iliner  xuliner  xlliner  xulst    xulnd  xllst  xllnd
   0    0.01    0.01  0.07546  0.1485  0.07546  0.1485
  i3d  iqain  ibleed  iopt  itsur
   0    0    1    0    0
  icond  ibound  init  ish  istd  igde
  1    3    2    1    1    1
  tsurf  ievap  itherm
   320.d0  1    1
  sta    rpm  irot
 10.d0  0.d0  1
  dtaui  nisp  dtaum  nmsp  dtauf  jcount
 1.d-1  1000  1.d-1  1000  1.d-1  10
  tspray  tafter
   0.d0  0.d0
  iscol  jscol  kscol  sslope  szero
input parameters for reading in hot air HTC info
 1    2    2    1.d0    1.0118860d0
input parameters for reading in hot air q_wall info
 1    2    2    1.d0    0.d0
input parameters for reading in surf. temperature info
 1    2    2    1.d0    0.d0

```

input parameters for reading in 3D streamline info

1 2 2 1.d0 0.d0

nprt nprf

0 0

ntype

1

iotype nptsd

1 14

C	section	where	layer	where
	5	2	1	3
	5	2	1	1
	4	2	1	3
	4	2	1	1
	6	2	1	3
	6	2	1	1
	3	2	1	3
	3	2	1	1
	7	2	1	3
	7	2	1	1
	2	2	1	3
	2	2	1	1
	8	2	1	3
	8	2	1	1

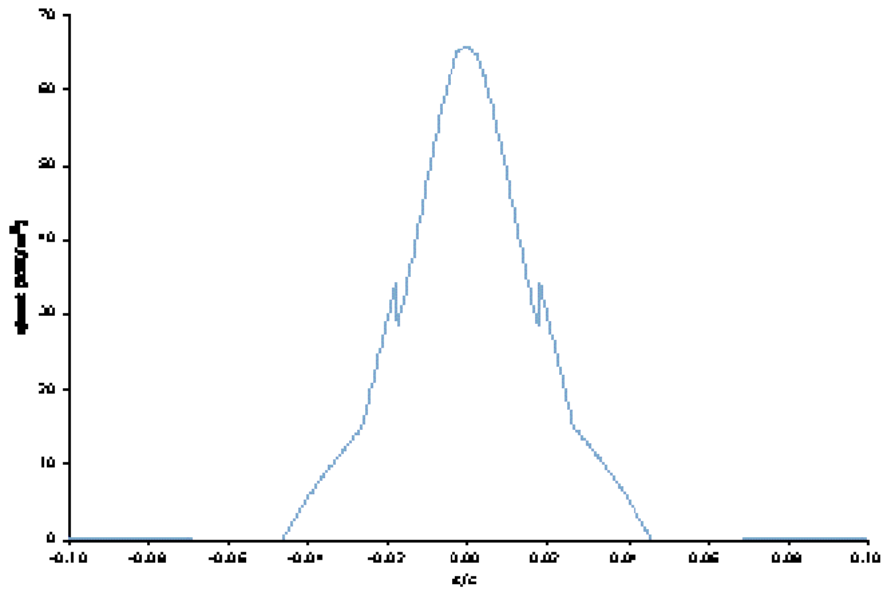


Figure 91: Heat Requirement for the Evaporative Hot Air System in Example Case 13

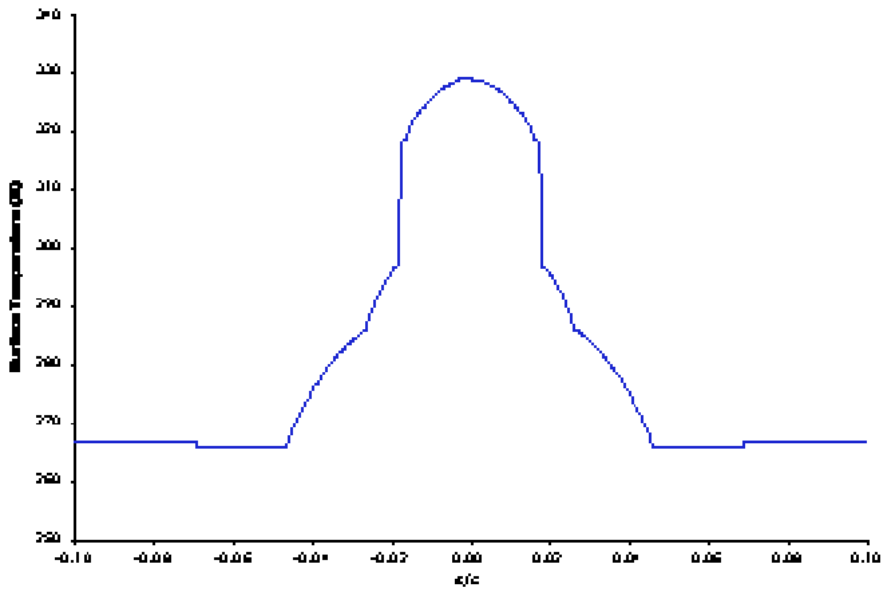


Figure 92: Surface Temperature for the Evaporative Hot Air System in Example Case 13

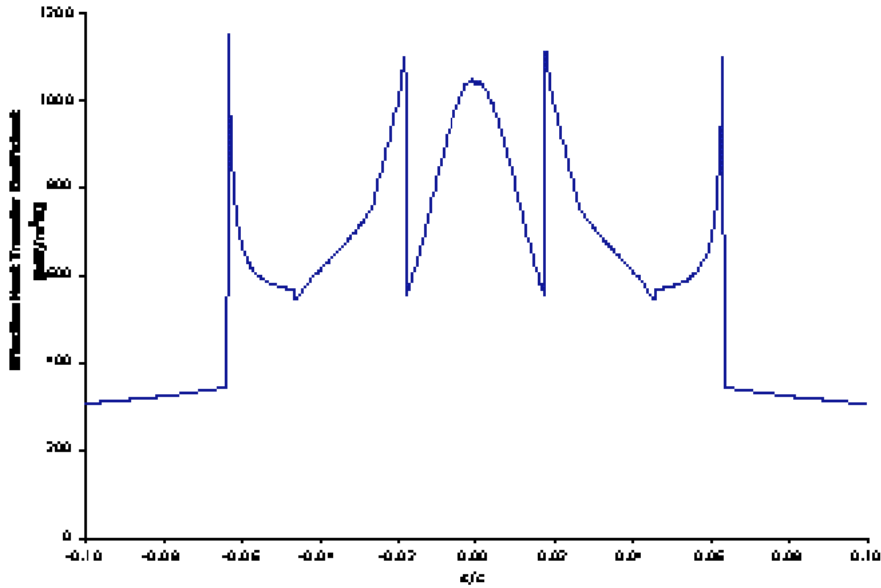


Figure 93: Hot Air Temperature for the Evaporative Hot Air System in Example Case 13

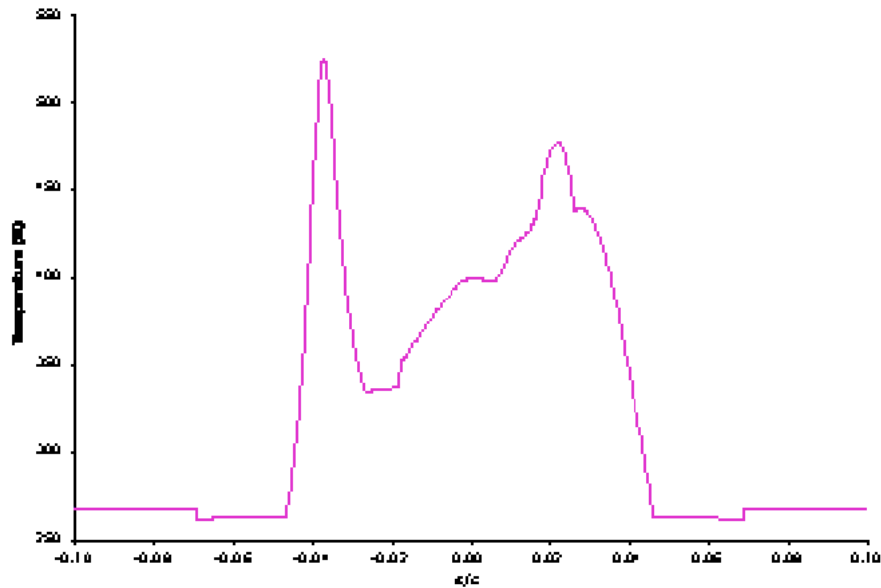


Figure 94: Effective Heat Transfer in Example Case 13

15.14. Case 14: First benchmark condition with a running wet electrothermal anti-icer.

Computation Time: Pentium IV 2.4GHz, 2 s.

Disk Space: 1.8 MB

Example Case 14 also shows an example using the IDEICE = 1 option and the de-icer input file. The conditions for this case are the same as those shown for Example Case 13, except in this

case a running wet thermal de-icer 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 because the IDEICE = 1 option was selected. This warning must also be confirmed to continue the run. Table 57 shows the main input file for this case while Table 58 lists the de-icer input file. The main input file and geometry input file are included on the distribution disks as “**case14.inp**” and “**case14.xyd**” respectively. The de-ice input file is listed on the distribution disk as “**case14d.inp**”.

The useful outputs for this case are the local heater wattages shown in Figure 95 and the effective heat transfer coefficient shown in Figure 97. 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 input flag IDEICE should be set to 3 to predict heater temperatures for this case.

Note: The full de-ice input file is read even though the program will only perform a one dimensional steady-state analysis.

Note: The de-ice input file name no longer is fixed as “**deicei.inp**” as it was in version 2.0.

Table 57: Main Input File for Example Case 14

Example 14 Test Case

```

&LEW20
ITIMFL = 0
TSTOP = 600
  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 = 0.
  VINP = 44.7
  LWC = 0.780
  TINP = 265.50

```



```

PINF = 100000.00
RH = 100.0
&END
&LPRNT
FPRT = 1
HPRT = 1
BPRT = 1
EPRT = 1
MPRT = 1
TPRT = 0
IDBF = 1
&END
&RDATA
&END
&BOOT
&END

```

Table 58: De-ice Input File for Example Case 14

# of nodes	length (m)	conductivity (W/m/K)	diffusivity (m**2/s)	anisotropy factor	slope (b) of temp. eqn.		
1	nx						
7	9						
C	substrate						
15	3.430d-3	0.120d0	1.652d-7	1.d0	0.d0		
C	insulation						
08	8.900d-4	0.294d0	1.045d-7	1.d0	0.d0		
C	insulation						
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0		
C	heater						
07	1.300d-5	41.000d0	1.194d-5	1.d0	0.d0		
C	insulation						
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0		
C	abrasion shield						
08	2.030d-4	16.270d0	4.035d-6	1.d0	0.d0		
C	ice						
61	2.540d-3	2.232d0	1.151d-6	1.d0	0.d0		
# of nodes	length (m)	cond. (W/m/K)	diff. (m**2/s)	anis. temp. eqn.	slope of temp. eqn.	add. thick.	layer number

C	gap (no heat input)						
10	0.9244d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
C	heater G						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater E						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater C						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	parting strip - heater A						
36	1.905d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater B						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater D						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater F						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3

ijde offset ipar

004 0.005 1

C	Q	TON	TOFF	TLAG	ICFLAG
	heater density	time on	time off	lag time	temp cntrl.
	(kWatts/m**2)	(sec)	(sec)	(sec)	flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	12.41d0	600.d0	0.d0	0.d0	0
C	heater E				
	8.87d0	600.d0	0.d0	0.d0	0
C	heater C				
	19.26d0	600.d0	0.d0	0.d0	0
C	parting strip (heater A) does not turn off				
	46.58d0	600.d0	0.d0	0.d0	0
C	heater B				
	31.95d0	600.d0	0.d0	0.d0	0
C	heater D				
	12.70d0	600.d0	0.d0	0.d0	0
C	heater F				

```

11.27d0      600.d0      0.d0      0.d0      0
C   gap (no heat input)
      0.0d0      0.d0      0.d0      0.d0      0
ibc1 ibc2 ibc3 ibc4
2   2   2   2
  tg1   tg2   tg3   tg4
268.3 268.3 268.3 268.3
  h1    h2    h3    h4
3.d0   3.d2  0.d0  0.d0
  qw1   qw2   qw3   qw4
0.d0   0.d0  0.d0  0.d0
  tairh  airmd  airup  xairmd
  5.d2  0.1d0  0.5d0  0.d0
  nhole  dhole  dtube  rdist  ycase  xuhnd  xlhnd
  3   0.00132 0.0381  50.   0.0   0.23  0.26
  icase  hconst  rep   rp    zp   rrp  dp
  0     1.0  0.76  1.394  1.   0.   0.
  xhole(m)  yhole(m)  zdist(m)
.   0.0176   -0.033   0.0107
  0.000     0.000   0.0091
  0.0176   -0.033   0.0076
  iliner  xuliner  xlliner  xulst   xulnd  xllst  xllnd
  0     0.01   0.01  0.07546  0.1485 0.07546 0.1485
i3d  iqain  ibleed  iopt  itsur
  0   0     0     0     0
icond  ibound  init  ish  istd  igde
  1   3     2     1   1   1
tsurf  ievap  itherm
  283.15d0  0     0
sta    rpm   irot
10.d0  0.d0  1
dtaui  nisp  dtaum  nmisp  dtauf  jcount
1.d-1  1000  1.d-1  1000  1.d-1  10
tspray  tafter
  0.d0  0.d0
iscol   jscol   kscol   sslope  szero
input parameters for reading in hot air HTC info
2     3     3     1.d0  0.d0

```

```

input parameters for reading in hot air q_wall info
1      2      2      1.d0    0.d0
input parameters for reading in surf. temperature info
1      2      2      1.d0    0.d0
input parameters for reading in 3D streamline info
1      2      2      1.d0    0.d0
nprt  nprf
0     0
nctype
1
iotype  nptsd
1      21
C      section where      layer  where
      5      2      6      3
      5      2      1      1
      4      2      6      3
      4      2      1      1
      6      2      6      3
      6      2      1      1
      3      2      6      3
      3      2      1      1
      7      2      6      3
      7      2      1      1
      2      2      6      3
      2      2      1      1
      8      2      6      3
      8      2      1      1
      5      2      4      1
      4      2      4      1
      6      2      4      1
      3      2      4      1
      7      2      4      1
      2      2      4      1
      8      2      4      1

```

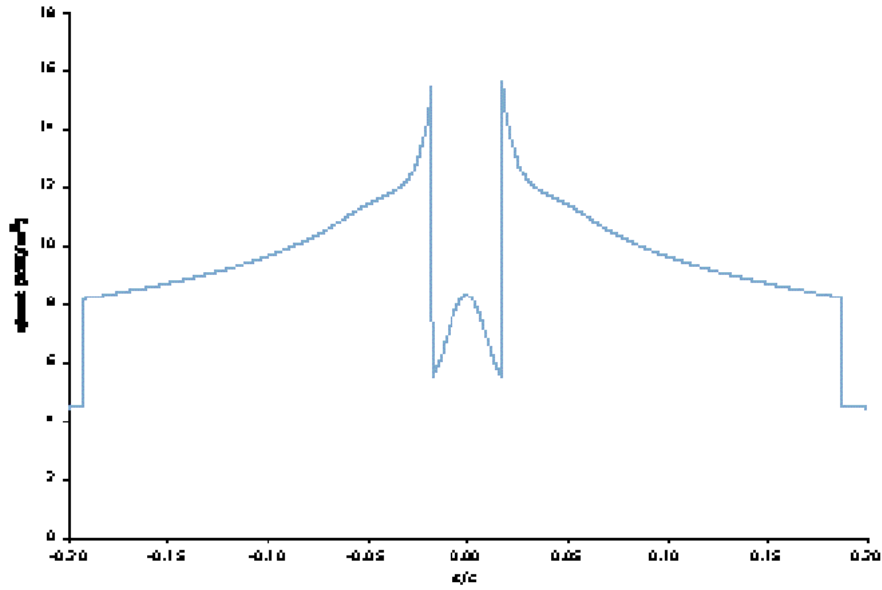


Figure 95: Heat Requirement for the Running Wet Thermal De-icer in Example Case 14

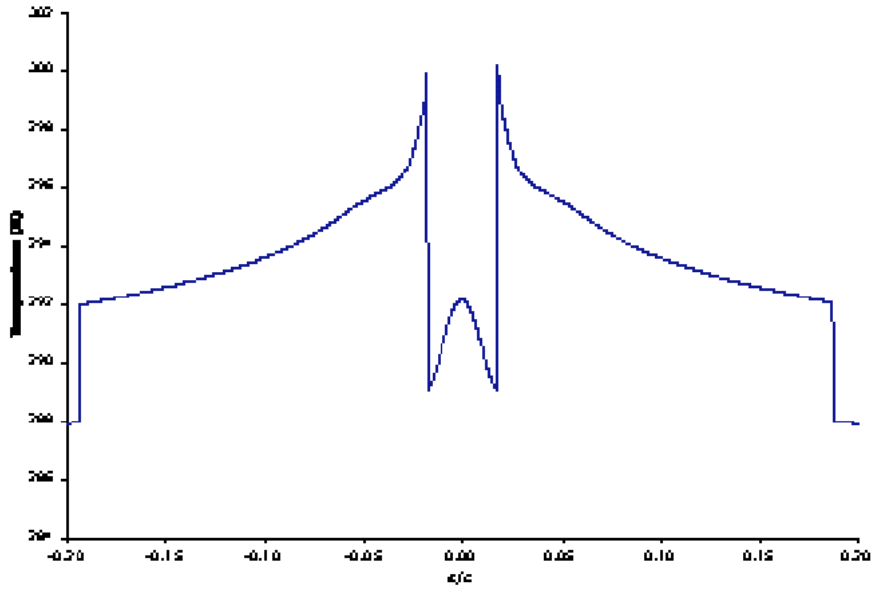


Figure 96: Heater Temperature for the Running Wet Thermal De-icer in Example Case 14

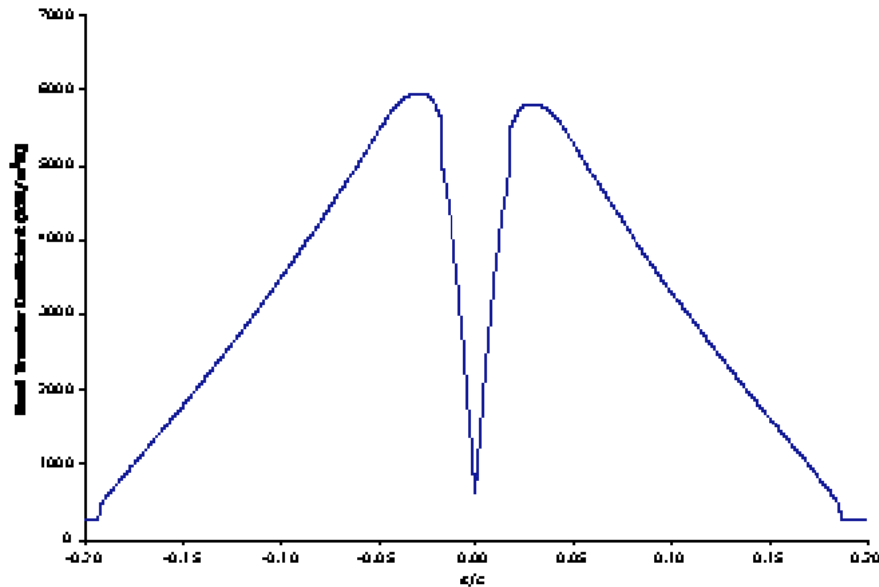


Figure 97: Effective Heat Transfer Coefficient in Example Case 14

15.15. Case 15: Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 47 min.15 s.

Disk Space: 26.5 MB

This example case illustrates a feature in version 3.2 of LEWICE that was added as of LEWICE 2.2. In the previous anti-icing examples, a simplified analysis was employed which assumed that heat transfer was one dimensional in the direction normal to the airfoil. This approach yielded heat flux values which appear reasonable, but results in heater (or bleed air) temperatures which appear excessive. This result shows the importance of the chordwise component of heat transfer on the process. The approach used in this and subsequent examples solves the two dimensional unsteady heat conduction equation within the airfoil geometry and couples this to the standard mass and energy balance used by LEWICE for ice accretion prediction. This approach is an extension of the capabilities which previously resided in the LEWICE/Thermal and ANTICE programs.

The first example using this approach models an electrothermal de-icer embedded within an airfoil as illustrated in Figure 98. The airfoil is a 36" chord NACA 0012 airfoil with seven heaters wrapped around the leading edge. The middle heater is 0.75" wide and is on for the duration of the run which simulates a parting strip heater. The parting strip heater separates the accreted ice into upper and lower segments for easier removal. The heaters on each side of the parting strip are turned on for 10 seconds and are off for 110 seconds (two minute cycle time). This set of heaters will first turn on 100 seconds into the run. The remaining four heaters will also turn on for 10 seconds out of the two minute cycle, but will first engage 110 seconds into the simulation so that they will be on immediately after the second set of heaters turn off. This heater cycle is diagrammed in Figure 99.

The experimental data plotted in these figures comes from Run 28 from a 1996 IRT test on the NACA0012 geometry described. Each heater zone pictured below had thermocouples just below the airfoil surface and on the inside surface of the de-icer. An RTD was placed under the heater to record heater temperature. For all thermocouples and RTDs, there was a set of readings at two spanwise locations on the two dimensional model. There are over 100 cases from this test which are available for comparison. A complete comparison of all runs has been performed and a report on the results is currently being prepared. These comparisons show that the software underpredicts the temperatures from some of the thermocouples. This is believed to be caused by using the standard LEWICE heat transfer coefficients which result from an assumption that ice will form on the surface causing roughness which causes the boundary layer to turn turbulent. The next example will explore an assumption using a laminar heat transfer coefficient instead.

Note: In this example, there is no residual ice. Therefore, only one LEWICE time step was performed (IFLO = 1) and neither the ice shape nor ice thickness plots were included.

Note: In the IRT run of this condition, residual ice did form. Due to the fragile nature of the runback ice formations, quantitative tracings could not be obtained and only qualitative observations of the extent of runback ice were recorded.

Note: Only the heater temperatures have been shown in this example. Surface temperature and substrate temperature comparisons are similar.

Note: In Example Cases 15-20, the user will receive five temperature output files named “**temps1.dat**” through “**temps5.dat**”. Since 21 locations were chosen for temperature versus time output, the first three files (“**temps1.dat**” to “**temps3.dat**”) will contain that data. “**temps4.dat**” will contain the temperature versus wrap distance location at the surface of the de-icer for 30 different times. “**temps5.dat**” will contain temperature as a function of the distance normal to the surface at the center of the parting strip heater for the same 30 time steps. All of these options are controlled by the user and are discussed in Section 10.1.5.

The 21 locations chosen for temperature versus time output correspond to the approximate locations of the 21 thermocouples in the experimental setup. Furthermore, they are output in same order as the original experimental data files to avoid confusion. This order is given as follows: Section A, outer surface; Section A, inner surface; Section B, outer surface; Section B, inner surface; Section C, outer surface; Section C, inner surface; Section D, outer surface; Section D, inner surface; Section E, outer surface; Section E, inner surface (end of “**temps1.dat**” file); Section F, outer surface; Section F, inner surface; Section G, outer surface; Section G, inner surface; Section A, heater; Section B, heater; Section C, heater; Section D, heater; Section E, heater; Section F, heater (end of “**temps2.dat**” file); Section G, heater. The remaining columns of “**temps3.dat**” contain zero values to avoid confusion. Temperature is output to these files in Kelvin every 0.1 seconds as specified in the input file. Experimental data was recorded in degrees Fahrenheit once per second. The utility program REDUCE was used to convert the LEWICE 3.2 output for easier comparison to the experimental data.

Table 59: Main Input File for Example Case 15

Example 15 Test Case

&LEW20

ITIMFL = 0

```

TSTOP = 600
  IBOD = 1
  IFLO = 1
  DSMN = 4.0D-4
  NPL = 24
  IDEICE = 2
&END
&DIST
  FLWC = 1.0
  DPD = 20.
&END
&ICE1
CHORD = 0.9144
  AOA = 0.
  VINP = 44.7
  LWC = 0.780
  TINP = 265.50
  PINP = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  EPRT = 1
  MPRT = 1
  TPRT = 0
  IDBF = 1
&END
&RDATA
&END
&BOOT
&END

```

Table 60: De-ice Input File for Example Case 15

```

1  nx
7  9
# of length conductivity diffusivity anisotropy slope (b) of

```


nodes	(m)	(W/m/K)	(m**2/s)	factor	temp. eqn.		
C	substrate						
15	3.430d-3	0.120d0	1.652d-7	1.d0	0.d0		
C	insulation						
08	8.900d-4	0.294d0	1.045d-7	1.d0	0.d0		
C	insulation						
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0		
C	heater						
07	1.300d-5	41.000d0	1.194d-5	1.d0	0.d0		
C	insulation						
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0		
C	abrasion shield						
08	2.030d-4	16.270d0	4.035d-6	1.d0	0.d0		
C	ice						
21	2.540d-3	2.232d0	1.151d-6	1.d0	0.d0		
# of nodes	length (m)	cond. (W/m/K)	diff. (m**2/s)	anis.	slope of temp. eqn.	add. thick.	layer number
C	gap (no heat input)						
10	0.9244d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
C	heater G						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater E						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater C						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	parting strip - heater A						
36	1.905d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater B						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater D						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater F						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
ijde	offset	ipar					
004	0.000	1					
C	Q	TON	TOFF	TLAG	ICFLAG		

	heater density (kWatts/m**2)	time on (sec)	time off (sec)	lag time (sec)	temp cntrl. flag
C	gap (no heat input)	0.0d0	0.d0	0.d0	0
C	heater G	11.75d0	10.d0	110.d0	0
C	heater E	11.27d0	10.d0	110.d0	0
C	heater C	11.68d0	10.d0	110.d0	0
C	parting strip (heater A) does not turn off	07.78d0	600.d0	0.d0	0
C	heater B	11.25d0	10.d0	110.d0	0
C	heater D	11.19d0	10.d0	110.d0	0
C	heater F	12.02d0	10.d0	110.d0	0
C	gap (no heat input)	0.0d0	0.d0	0.d0	0

ibc1 ibc2 ibc3 ibc4

2 2 2 2

tg1 tg2 tg3 tg4

265.5 265.5 265.5 265.5

h1 h2 h3 h4

3.d0 3.d2 0.d0 0.d0

qw1 qw2 qw3 qw4

0.d0 0.d0 0.d0 0.d0

tairh airmd airup xairmd

5.d2 0.1d0 0.5d0 0.d0

nhole dhole dtube rdist ycase xuhnd xlhnd

3 0.00132 0.0381 50. 0.0 0.23 0.26

icase hconst rep rp zp rrp dp

0 1.0 0.76 1.394 1. 0. 0.

xhole(m) yhole(m) zdist(m)

· 0.0176 -0.033 0.0107

0.000 0.000 0.0091

0.0176 -0.033 0.0076

```

iliner  xuliner xlliner  xulst    xulnd  xllst  xllnd
  0     0.01    0.01    0.07546  0.1485 0.07546 0.1485
i3d  iqain  ibleed  iopt  itsur
  0     0     0     0     0
icond  ibound  init  ish  istd  igde
  1     3     2     1     1     1
tsurf  ievap  itherm
 278.d0  1     0
sta     rpm  irot
10.d0  0.d0  1
dtaui  nisp  dtaum  nmsp  dtauf  jcount
1.d-1  1000  1.d-1  1000  1.d-1  10
tspray  tafter
 0.d0  0.d0
iscol  jscol  kscol  sslope  szero
input parameters for reading in hot air HTC info
 2     3     3     1.d0  0.d0
input parameters for reading in hot air q_wall info
 1     2     2     1.d0  0.d0
input parameters for reading in surf. temperature info
 1     2     2     1.d0  0.d0
input parameters for reading in 3D streamline info
 1     2     2     1.d0  0.d0
nprrt  nprf
 0     0
nctype
 3
iotype  nptsd
 1     21
C       section where    layer  where
      5     2           6     3
      5     2           1     1
      4     2           6     3
      4     2           1     1
      6     2           6     3
      6     2           1     1
      3     2           6     3
      3     2           1     1

```

7	2	6	3
7	2	1	1
2	2	6	3
2	2	1	1
8	2	6	3
8	2	1	1
5	2	4	1
4	2	4	1
6	2	4	1
3	2	4	1
7	2	4	1
2	2	4	1
8	2	4	1

iotype nptsd

2 30

C

C time layer where

C (sec)

5	6	1
10	6	1
15	6	1
20	6	1
25	6	1
30	6	1
35	6	1
40	6	1
45	6	1
50	6	1
55	6	1
60	6	1
65	6	1
70	6	1
75	6	1
80	6	1
85	6	1
90	6	1
95	6	1
100	6	1

105	6	1
110	6	1
115	6	1
120	6	1
125	6	1
130	6	1
135	6	1
140	6	1
145	6	1
150	6	1

iotype nptsd

3 30

C time section where

C (sec)

5	5	2
10	5	2
15	5	2
20	5	2
25	5	2
30	5	2
35	5	2
40	5	2
45	5	2
50	5	2
55	5	2
60	5	2
65	5	2
70	5	2
75	5	2
80	5	2
85	5	2
90	5	2
95	5	2
100	5	2
105	5	2
110	5	2
115	5	2
120	5	2

125	5	2
130	5	2
135	5	2
140	5	2
145	5	2
150	5	2

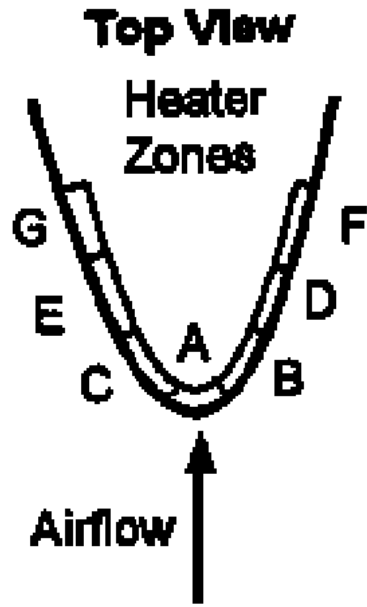


Figure 98: Schematic of Heaters for Example Case15

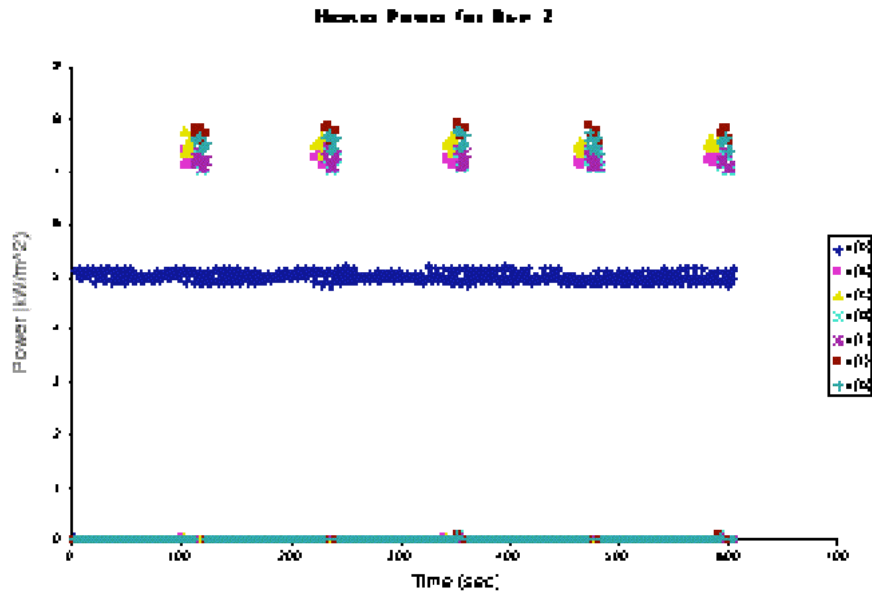


Figure 99: Heater Power for Example Case 15

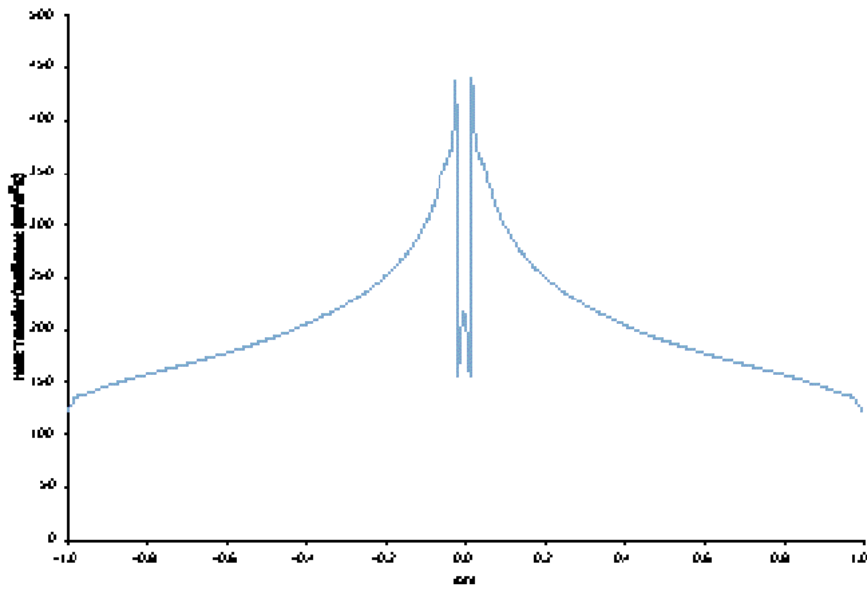


Figure 100: Heat Transfer Coefficient for Example Case 15

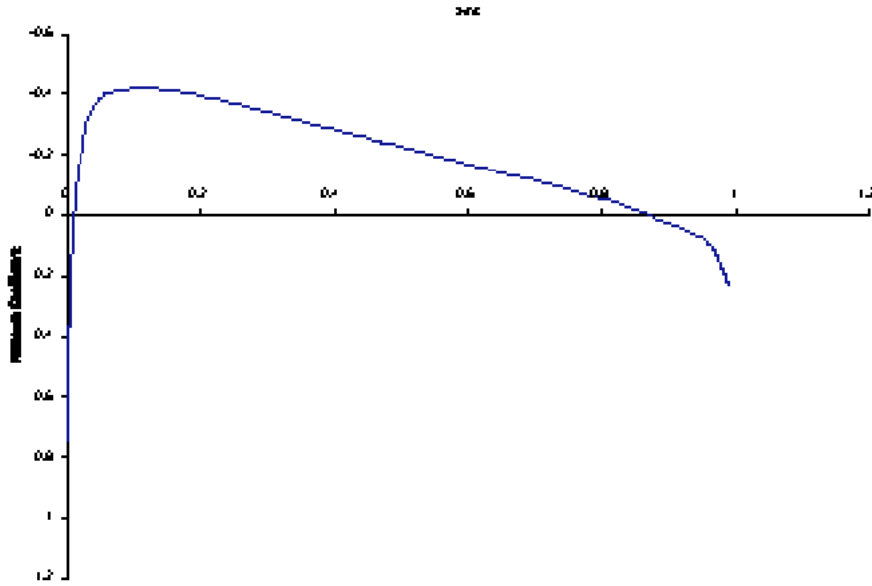


Figure 101: Pressure Coefficient for Example Case 15

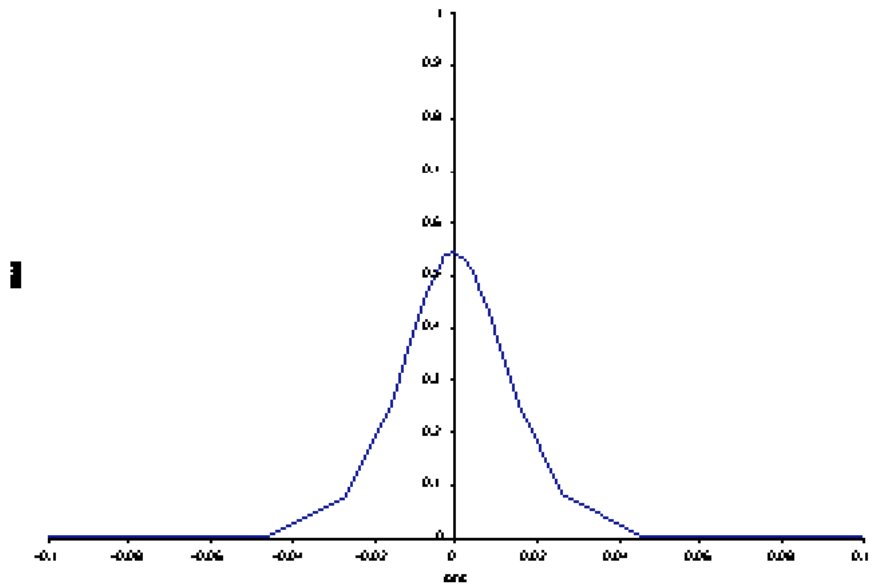


Figure 102: Collection Efficiency for Example Case 15

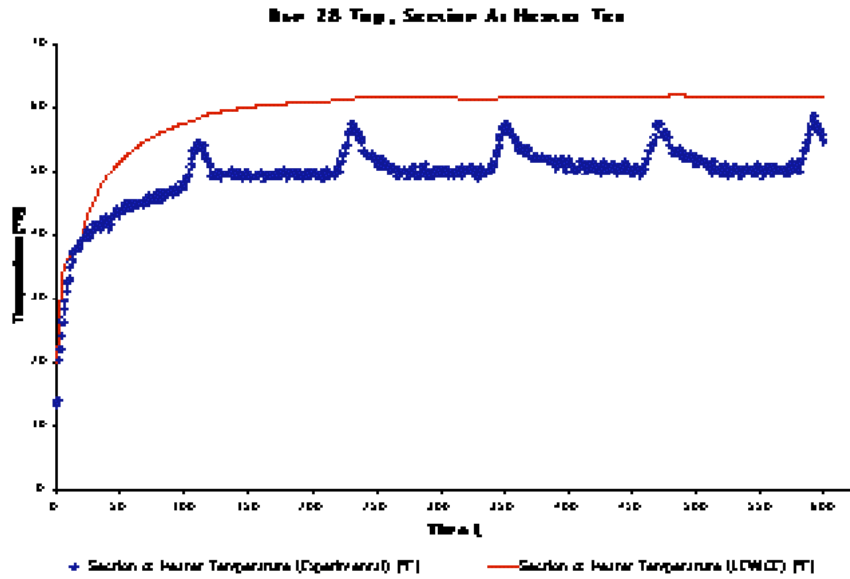


Figure 103: Heater A Temperature for Example Case 15

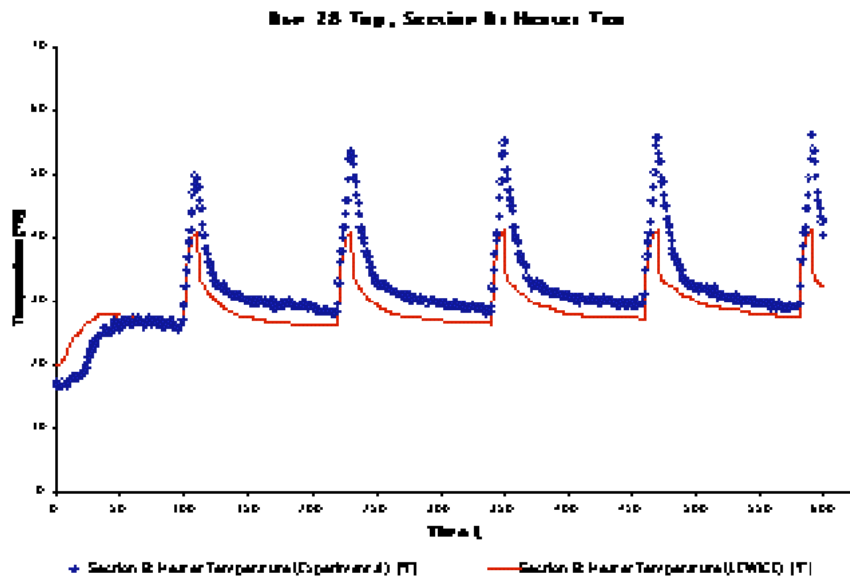


Figure 104: Heater B Temperature for Example Case 15

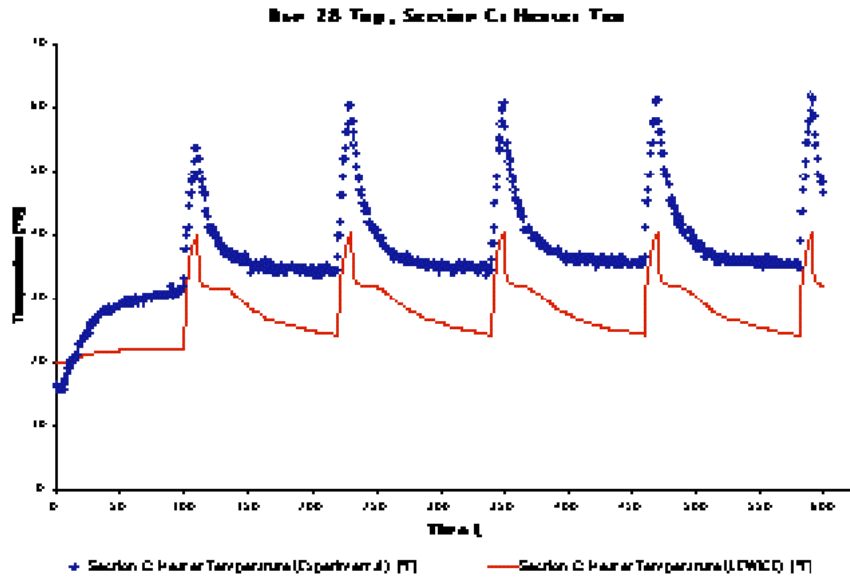


Figure 105: Heater C Temperature for Example Case 15

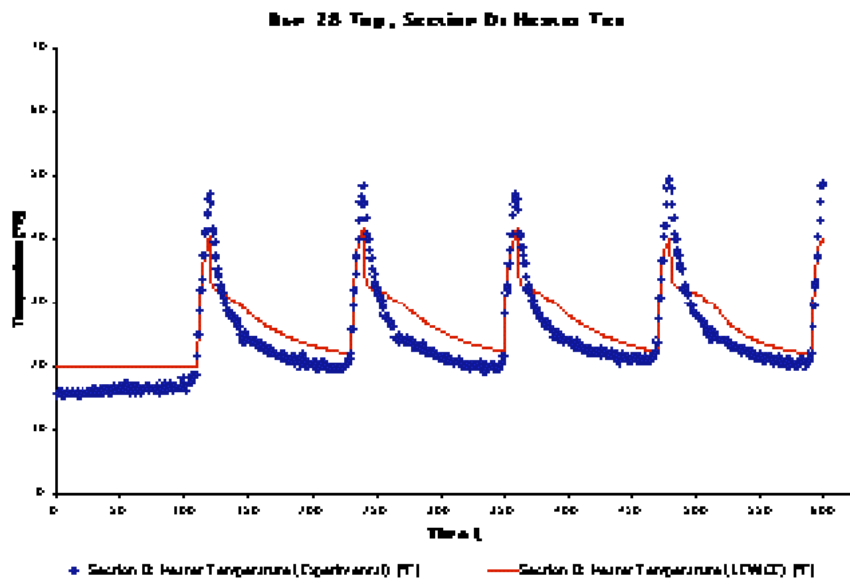


Figure 106: Heater D Temperature for Example Case 15

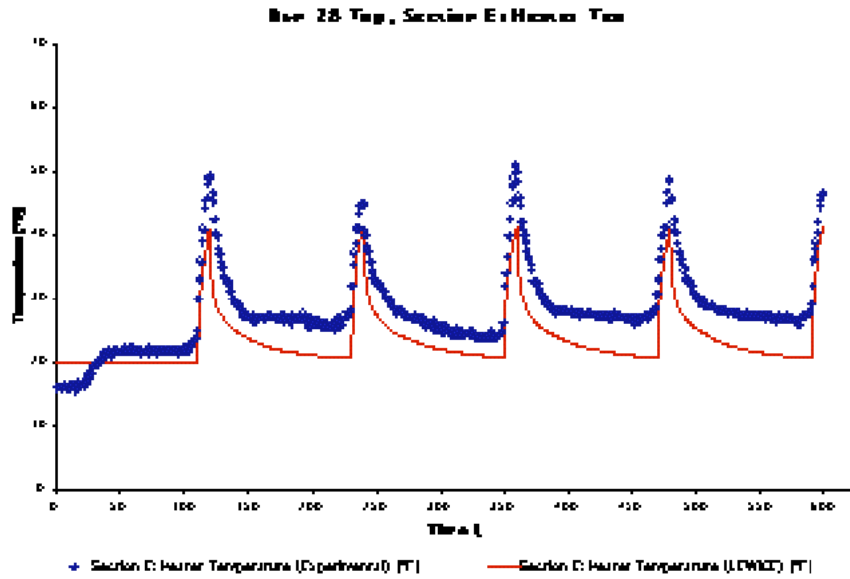


Figure 107: Heater E Temperature for Example Case 15

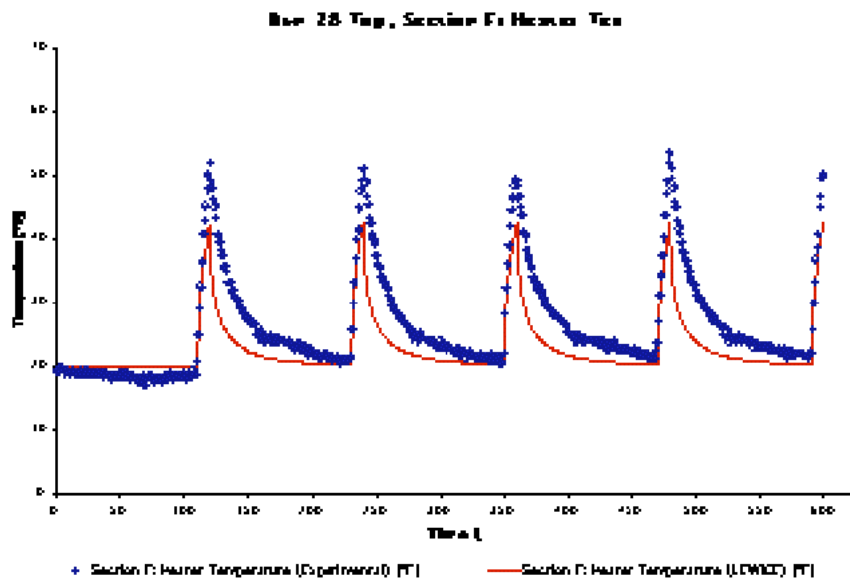


Figure 108: Heater F Temperature for Example Case 15

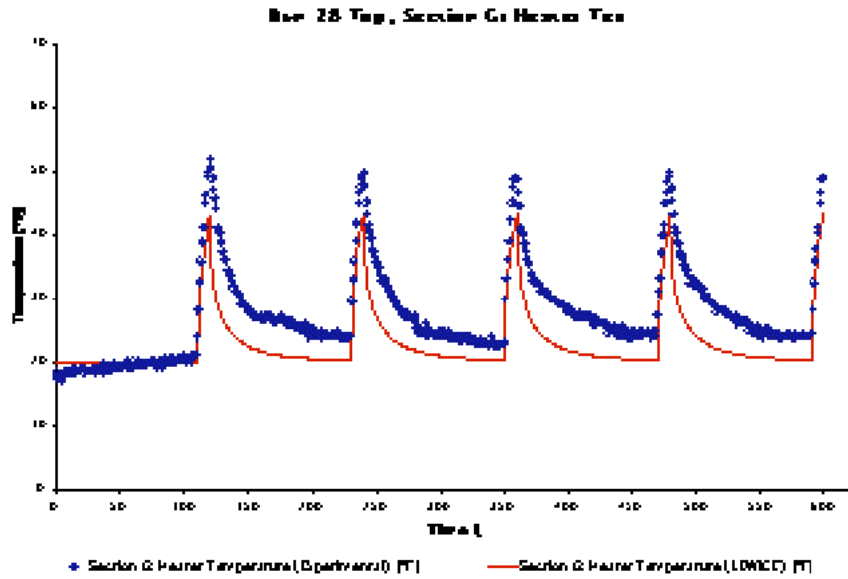


Figure 109: Heater G Temperature for Example Case 15

15.16. Case 16: Laminar Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 46 min.59 s.

Disk Space: 26.5 MB

This example case was created in conjunction with the previous example. In this case, the external heat transfer coefficient was forced to remain laminar on the airfoil. Within the icing limits, this is believed to be a more accurate assumption over the de-icer thermocouples since the de-icer maintains an ice-free surface after the ice is shed. The example inputs for this case are exactly the same as the previous example, except that IDEICE = 3 in the main LEWICE input file. The results from this case show that while the temperatures over heater zones “B” and “C” are better predicted using the laminar assumption, the results for heater zones “A”, “D”, “E”, “F” and “G” were better predicted using the turbulent assumption. This indicates that the best prediction could be obtained by forcing transition after heaters “B” and “C”. Forced transition, however, was not included as an input option to LEWICE 3.2 as this creates user to user variability in the results which was considered undesirable from a certification standpoint. Since the temperature in zone “A” (parting strip) is overpredicted with this option while neighboring zones “B” and “C” are underpredicted, this may indicate that the program may be underpredicting transverse heating in regions of high curvature. This assumption could not be verified however.

Note: In this example, there is no residual ice. Therefore, only one LEWICE time step was performed (IFLO = 1) and neither the ice shape nor ice thickness plots were included.

Note: In the IRT run of this condition, residual ice did form. Due to the fragile nature of the runback ice formations, quantitative tracings could not be obtained and only qualitative observations of the extent of runback ice were recorded.

Note: Only the heater temperatures have been shown in this example. Surface temperature and substrate temperature comparisons are similar.

Note: Pressure coefficients and collection efficiencies are the same as Example 15 and are not shown again.

Table 61: Main Input File for Example Case 16

Example 16 Test Case

```
&LEW20
ITIMFL = 0
TSTOP = 600
  IBOD = 1
  IFLO = 1
  DSMN = 4.0D-4
  NPL = 24
  IDEICE = 3
&END
&DIST
  FLWC = 1.0
  DPD = 20.
&END
&ICE1
CHORD = 0.9144
  AOA = 0.
  VINFL = 44.7
  LWC = 0.780
  TINF = 265.50
  PINFL = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  EPRT = 1
  MPRT = 1
  TPRT = 0
  IDBF = 1
&END
&RDATA
```

&END

&BOOT

&END

Table 62: De-ice Input File for Example Case 16

1	nx							
7	9							
# of nodes	length (m)	conductivity (W/m/K)	diffusivity (m**2/s)	anisotropy factor	slope (b) of temp. eqn.			
C	substrate							
15	3.430d-3	0.120d0	1.652d-7	1.d0	0.d0			
C	insulation							
08	8.900d-4	0.294d0	1.045d-7	1.d0	0.d0			
C	insulation							
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0			
C	heater							
07	1.300d-5	41.000d0	1.194d-5	1.d0	0.d0			
C	insulation							
18	2.800d-4	0.256d0	1.473d-7	1.d0	0.d0			
C	abrasion shield							
08	2.030d-4	16.270d0	4.035d-6	1.d0	0.d0			
C	ice							
21	2.540d-3	2.232d0	1.151d-6	1.d0	0.d0			
# of nodes	length (m)	cond. (W/m/K)	diff. (m**2/s)	anis. factor	slope of temp. eqn.	add. thick.	layer number	
C	gap (no heat input)							
10	0.9244d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3	
C	heater G							
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	
C	heater E							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	
C	heater C							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	
C	parting strip - heater A							
36	1.905d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	
C	heater B							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	
C	heater D							
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3	

```

C   heater F
14   3.175d-2  41.000d0  1.194d-5  1.d0    0.d0    0.d0    3
C   gap (no heat input)
10   0.9144d0  0.256d0  1.637d-7  1.d0    0.d0    0.d0    3
ijde offset ipar
004  0.000    1
C   Q          TON          TOFF          TLAG          ICFLAG
heater density time on      time off      lag time      temp cntrl.
(kWatts/m**2) (sec)         (sec)         (sec)         flag
C   gap (no heat input)
0.0d0          0.d0          0.d0          0.d0          0
C   heater G
11.75d0         10.d0         110.d0        110.d0        0
C   heater E
11.27d0         10.d0         110.d0        110.d0        0
C   heater C
11.68d0         10.d0         110.d0        100.d0        0
C   parting strip (heater A) does not turn off
07.78d0         600.d0        0.d0          0.d0          0
C   heater B
11.25d0         10.d0         110.d0        100.d0        0
C   heater D
11.19d0         10.d0         110.d0        110.d0        0
C   heater F
12.02d0         10.d0         110.d0        110.d0        0
C   gap (no heat input)
0.0d0          0.d0          0.d0          0.d0          0
ibc1 ibc2 ibc3 ibc4
2    2    2    2
tg1   tg2   tg3   tg4
265.5 265.5 265.5 265.5
h1    h2    h3    h4
3.d0  3.d2  0.d0  0.d0
qw1   qw2   qw3   qw4
0.d0  0.d0  0.d0  0.d0
tairh airmd airup xairmd
5.d2  0.1d0 0.5d0 0.d0
nhole dhole dtube rdist ycase xuhnd xlhnd

```

```

3 0.00132 0.0381 50. 0.0 0.23 0.26
icase hconst rep rp zp rrp dp
0 1.0 0.76 1.394 1. 0. 0.
xhole(m) yhole(m) zdist(m)
0.0176 -0.033 0.0107
0.000 0.000 0.0091
0.0176 -0.033 0.0076
iliner xuliner xlliner xulst xulnd xllst xllnd
0 0.01 0.01 0.07546 0.1485 0.07546 0.1485
i3d iqain ibleed iopt itsur
0 0 0 0 0
icond ibound init ish istd igde
1 3 2 1 1 1
tsurf ievap itherm
278.d0 1 0
sta rpm irot
10.d0 0.d0 1
dtaui nisp dtaum nmsp dtauf jcount
1.d-1 1000 1.d-1 1000 1.d-1 10
tspray tafter
0.d0 0.d0
iscol jscol kscol sslope szero
input parameters for reading in hot air HTC info
2 3 3 1.d0 0.d0
input parameters for reading in hot air q_wall info
1 2 2 1.d0 0.d0
input parameters for reading in surf. temperature info
1 2 2 1.d0 0.d0
input parameters for reading in 3D streamline info
1 2 2 1.d0 0.d0
nprt nprf
0 0
ntype
3
iotype nptsd
1 21
C section where layer where
5 2 6 3

```


5	2	1	1
4	2	6	3
4	2	1	1
6	2	6	3
6	2	1	1
3	2	6	3
3	2	1	1
7	2	6	3
7	2	1	1
2	2	6	3
2	2	1	1
8	2	6	3
8	2	1	1
5	2	4	1
4	2	4	1
6	2	4	1
3	2	4	1
7	2	4	1
2	2	4	1
8	2	4	1

iotype nptsd

2 30

C

C time layer where

C (sec)

5	6	1
10	6	1
15	6	1
20	6	1
25	6	1
30	6	1
35	6	1
40	6	1
45	6	1
50	6	1
55	6	1
60	6	1
65	6	1

70	6	1
75	6	1
80	6	1
85	6	1
90	6	1
95	6	1
100	6	1
105	6	1
110	6	1
115	6	1
120	6	1
125	6	1
130	6	1
135	6	1
140	6	1
145	6	1
150	6	1

iotype nptsd

3 30

C time section where

C (sec)

5	5	2
10	5	2
15	5	2
20	5	2
25	5	2
30	5	2
35	5	2
40	5	2
45	5	2
50	5	2
55	5	2
60	5	2
65	5	2
70	5	2
75	5	2
80	5	2
85	5	2

90	5	2
95	5	2
100	5	2
105	5	2
110	5	2
115	5	2
120	5	2
125	5	2
130	5	2
135	5	2
140	5	2
145	5	2
150	5	2

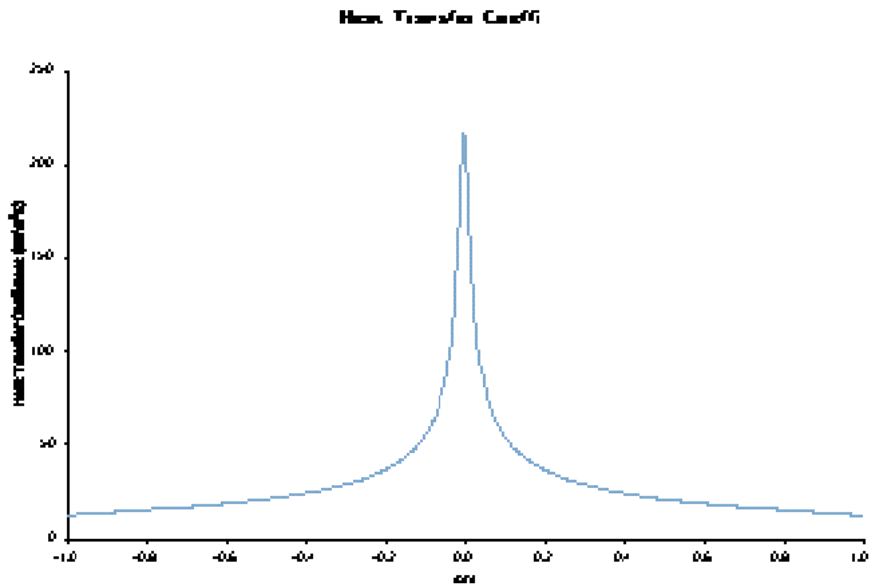


Figure 110: Heat Transfer Coefficient for Example Case 16

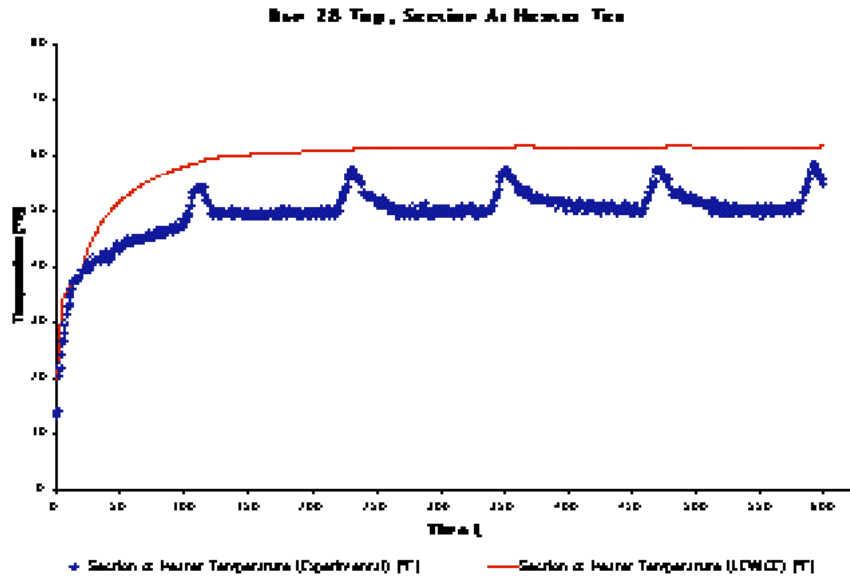


Figure 111: Heater A Temperature for Example Case 16

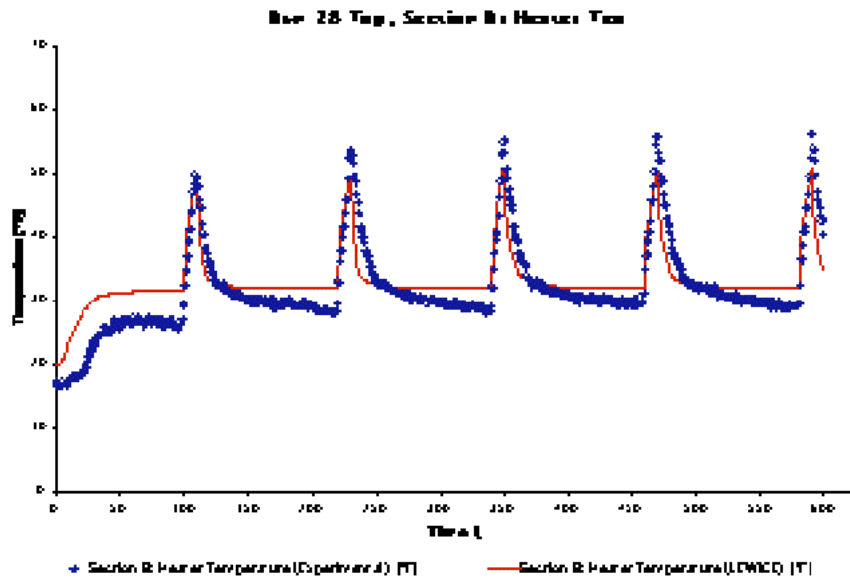


Figure 112: Heater B Temperature for Example Case 16

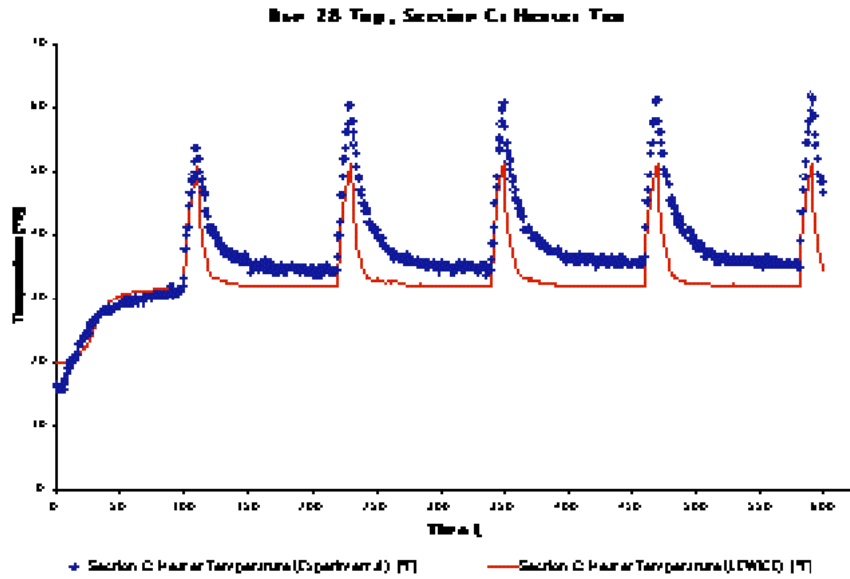


Figure 113: Heater C Temperature for Example Case 16

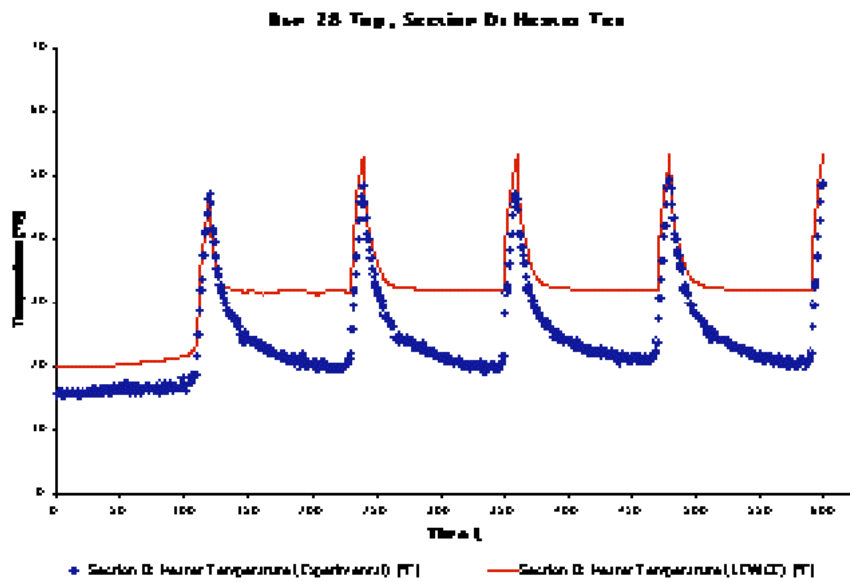


Figure 114: Heater D Temperature for Example Case 16

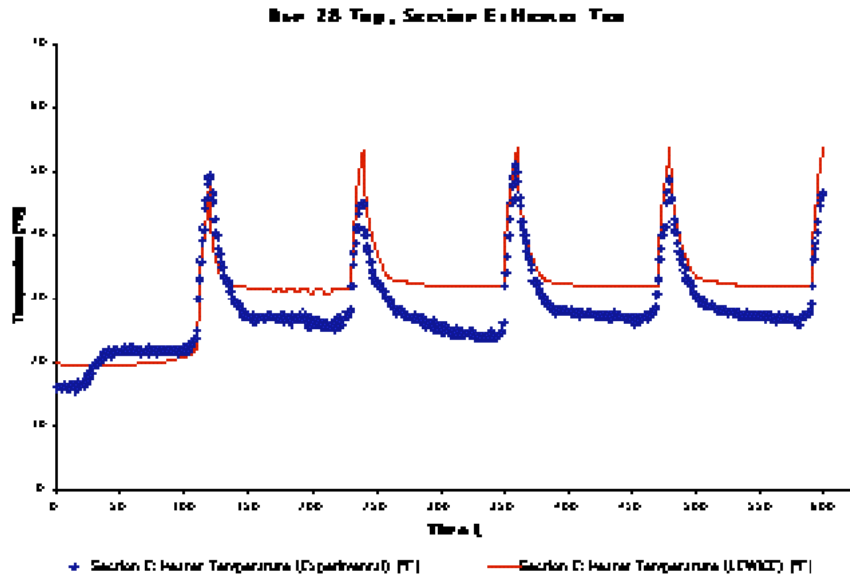


Figure 115: Heater E Temperature for Example Case 16

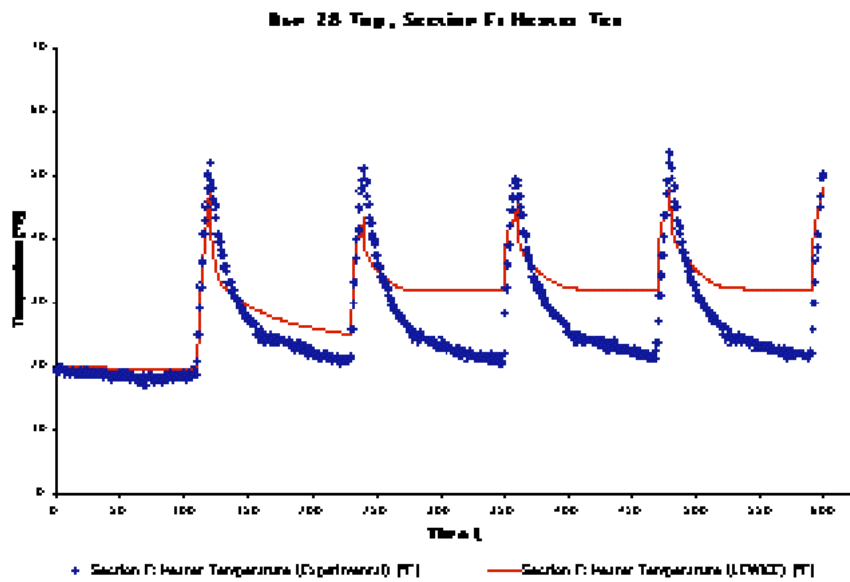


Figure 116: Heater F Temperature for Example Case 16

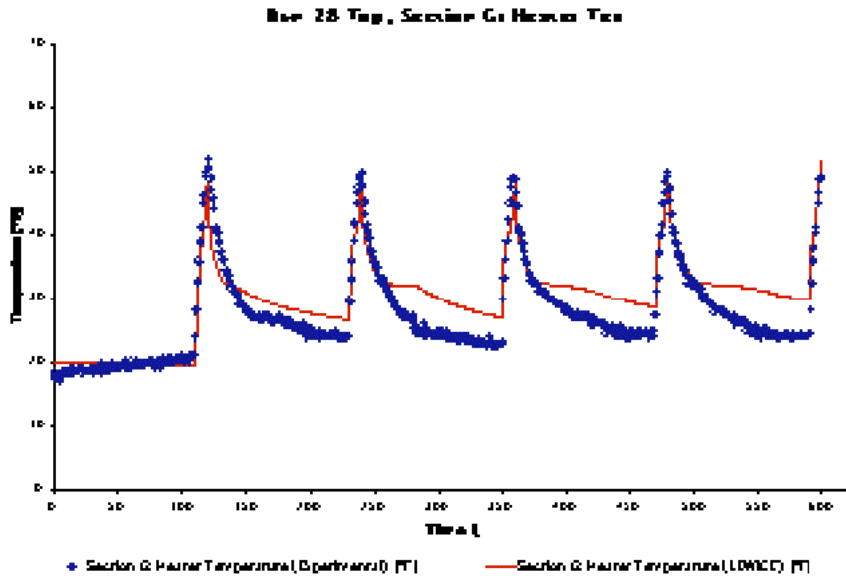


Figure 117: Heater G Temperature for Example Case 16

15.17. Case 17: Running Wet Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 40 min.33 s.

Disk Space: 11.5 MB

This example case uses the same NACA0012 de-icer model as used in the previous two cases. In this example, however, the heaters are on for the duration of the simulation. The wattages used were high enough in the experiment (Run 22F) to keep the de-icer region free of ice, but was not high enough to evaporate all of the water. In this mode, the electrothermal unit operates similarly to a hot air anti-icing system, albeit with specified heat fluxes instead of specified flow rates. In this example, the laminar heat transfer assumption was used since the de-icer region was completely free of ice.

The results show an excellent comparison to the experimental results using the laminar assumption. A case using the turbulent heat transfer assumption was also ran (but not shown) which again resulted in an underprediction of the temperatures. Similar results were obtained for all of the anti-icing cases in this database, although many of the laminar cases showed an overprediction of temperature. This would seem to indicate that the actual heat transfer coefficient lies between the two predicted values. A second explanation would be a physical effect not modeled which would cause an increase of the convective heat transfer coefficient due to the evaporation of water above and beyond that which is already modeled. In either case, this increased heat transfer coefficient can be reverse-engineered from the experimental data. However, since the database contains only NACA0012 airfoils, most cases of which are at 0° AOA, it is unknown how applicable this experimentally adjusted coefficient would be outside this region. As a result, the adjusted coefficient was left out of the LEWICE 3.2 model.

Another feature modeled in this example is the comparison of the 1D anti-icing model to the 2D anti-icing model. This comparison can be achieved for any case by using the IEVAP = 2

option in the de-icer input file. Many users have requested this option, as the 1D anti-icing model is known to overpredict heater temperatures. This comparison is shown in Figure 125. While the temperatures are higher in the 1D model, the difference is not as great as some have suggested.

Note: The heat transfer coefficients, pressure coefficients and collection efficiencies are the same as Example Case 16 and are not shown again.

Table 63: Main Input File for Example Case 17

Example 17 Test Case

```
&LEW20
ITIMFL = 0
TSTOP = 600
  IBOD = 1
  IFLO = 1
  DSMN = 4.0D-4
  NPL = 24
  IDEICE = 3
&END
&DIST
  FLWC = 1.0
  DPD = 20.
&END
&ICE1
CHORD = 0.9144
  AOA = 0.
  VINFL = 44.7
  LWC = 0.780
  TINFL = 265.50
  PINFL = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  EPRT = 1
  MPRT = 1
  TPRT = 0
  IDBF = 1
&END
```



```

&RDATA
&END
&BOOT
&END

```

Table 64: De-ice Input File for Example Case 17

```

1    nx
7    9
# of length conductivity diffusivity anisotropy slope (b) of
nodes (m) (W/m/K) (m**2/s) factor temp. eqn.
C    substrate
15   3.430d-3  0.120d0  1.652d-7  1.d0  0.d0
C    insulation
08   8.900d-4  0.294d0  1.045d-7  1.d0  0.d0
C    insulation
18   2.800d-4  0.256d0  1.473d-7  1.d0  0.d0
C    heater
07   1.300d-5  41.000d0  1.194d-5  1.d0  0.d0
C    insulation
18   2.800d-4  0.256d0  1.473d-7  1.d0  0.d0
C    abrasion shield
08   2.030d-4  16.270d0  4.035d-6  1.d0  0.d0
C    ice
21   2.540d-3  2.232d0  1.151d-6  1.d0  0.d0
# of length cond. diff. anis. slope of add. layer
nodes (m) (W/m/K) (m**2/s) temp. eqn. thick. number
C    gap (no heat input)
10   0.9244d0  0.256d0  1.637d-7  1.d0  0.d0  0.d0  3
C    heater G
14   3.175d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C    heater E
21   2.540d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C    heater C
21   2.540d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C    parting strip - heater A
36   1.905d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C    heater B
21   2.540d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C    heater D

```

21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater F						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3
ijde	offset ipar						
004	0.000	1					
C	Q	TON	TOFF	TLAG	ICFLAG		
	heater density	time on	time off	lag time	temp cntrl.		
	(kWatts/m**2)	(sec)	(sec)	(sec)	flag		
C	gap (no heat input)						
	0.0d0	0.d0	0.d0	0.d0	0		
C	heater G						
	2.27d0	600.d0	0.d0	0.d0	0		
C	heater E						
	3.40d0	600.d0	0.d0	0.d0	0		
C	heater C						
	2.94d0	600.d0	0.d0	0.d0	0		
C	parting strip (heater A) does not turn off						
	4.82d0	600.d0	0.d0	0.d0	0		
C	heater B						
	3.84d0	600.d0	0.d0	0.d0	0		
C	heater D						
	2.95d0	600.d0	0.d0	0.d0	0		
C	heater F						
	2.59d0	600.d0	0.d0	0.d0	0		
C	gap (no heat input)						
	0.0d0	0.d0	0.d0	0.d0	0		
ibc1	ibc2	ibc3	ibc4				
2	2	2	2				
tg1	tg2	tg3	tg4				
265.5	265.5	265.5	265.5				
h1	h2	h3	h4				
3.d0	3.d2	0.d0	0.d0				
qw1	qw2	qw3	qw4				
0.d0	0.d0	0.d0	0.d0				
tairh	airmd	airup	xairmd				
5.d2	0.1d0	0.5d0	0.d0				

```

nhole  dhole    dtube  rdist  ycase  xuhnd  xlhnd
  3  0.00132  0.0381  50.    0.0    0.23  0.26
icase  hconst  rep    rp     zp    rrp    dp
  0    1.0  0.76  1.394  1.    0.    0.
xhole(m)  yhole(m)  zdist(m)
.  0.0176  -0.033  0.0107
  0.000  0.000  0.0091
  0.0176  -0.033  0.0076
iliner  xuliner  xlliner  xulst  xulnd  xllst  xllnd
  0    0.01  0.01  0.07546  0.1485  0.07546  0.1485
i3d  iqain  ibleed  iopt  itsur
  0    0    0    0    0
icond  ibound  init  ish  istd  igde
  1    3    2    1    1    1
tsurf  ievap  itherm
  298.d0  2    0
sta    rpm  irot
10.d0  0.d0  1
dtaui  nisp  dtaum  nmisp  dtauf  jcount
1.d-1  1000  1.d-1  1000  1.d-1  10
tspray  tafter
  0.d0  0.d0
iscol  jscol  kscol  sslope  szero
input parameters for reading in hot air HTC info
  2    3    3    1.d0  0.d0
input parameters for reading in hot air q_wall info
  1    2    2    1.d0  0.d0
input parameters for reading in surf. temperature info
  1    2    2    1.d0  0.d0
input parameters for reading in 3D streamline info
  1    2    2    1.d0  0.d0
nprt  nprf
  0    0
ntype
  3
iotype  nptsd
  1    21
C      section where  layer  where

```

5	2	6	3
5	2	1	1
4	2	6	3
4	2	1	1
6	2	6	3
6	2	1	1
3	2	6	3
3	2	1	1
7	2	6	3
7	2	1	1
2	2	6	3
2	2	1	1
8	2	6	3
8	2	1	1
5	2	4	1
4	2	4	1
6	2	4	1
3	2	4	1
7	2	4	1
2	2	4	1
8	2	4	1

iotype nptsd

2 30

C

C time layer where

C (sec)

5	6	1
10	6	1
15	6	1
20	6	1
25	6	1
30	6	1
35	6	1
40	6	1
45	6	1
50	6	1
55	6	1
60	6	1

65	6	1
70	6	1
75	6	1
80	6	1
85	6	1
90	6	1
95	6	1
100	6	1
105	6	1
110	6	1
115	6	1
120	6	1
125	6	1
130	6	1
135	6	1
140	6	1
145	6	1
150	6	1

iotype nptsd

3 30

C time section where

C (sec)

5	5	2
10	5	2
15	5	2
20	5	2
25	5	2
30	5	2
35	5	2
40	5	2
45	5	2
50	5	2
55	5	2
60	5	2
65	5	2
70	5	2
75	5	2
80	5	2

85	5	2
90	5	2
95	5	2
100	5	2
105	5	2
110	5	2
115	5	2
120	5	2
125	5	2
130	5	2
135	5	2
140	5	2
145	5	2
150	5	2

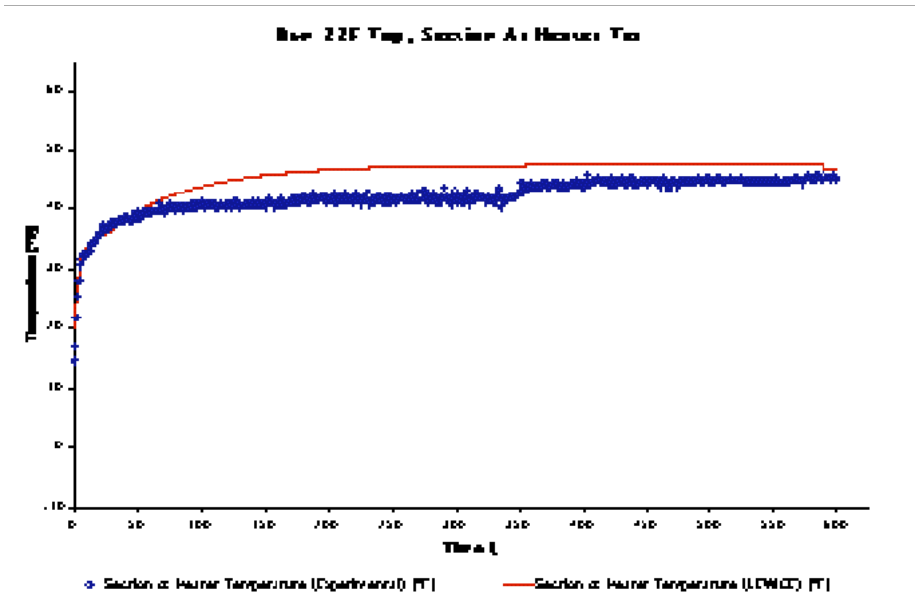


Figure 118: Heater A Temperature for Example Case 17

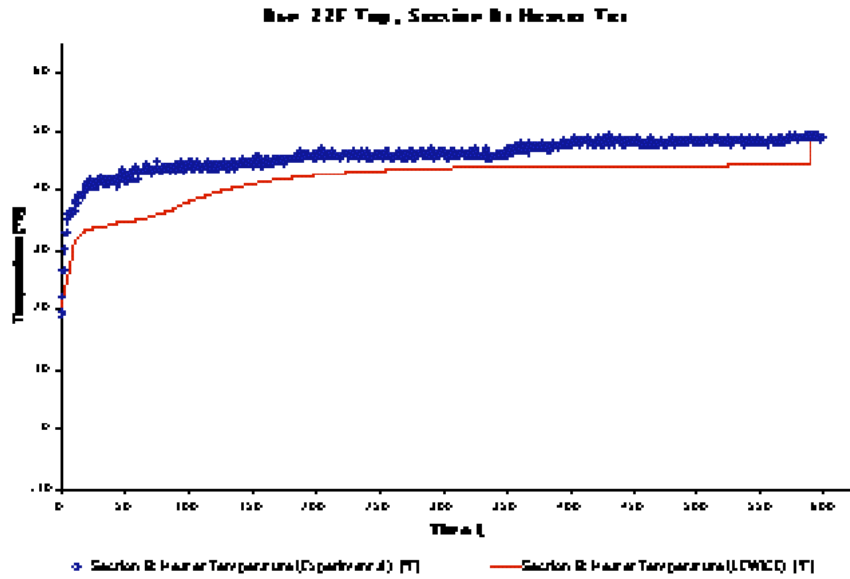


Figure 119: Heater B Temperature for Example Case 17

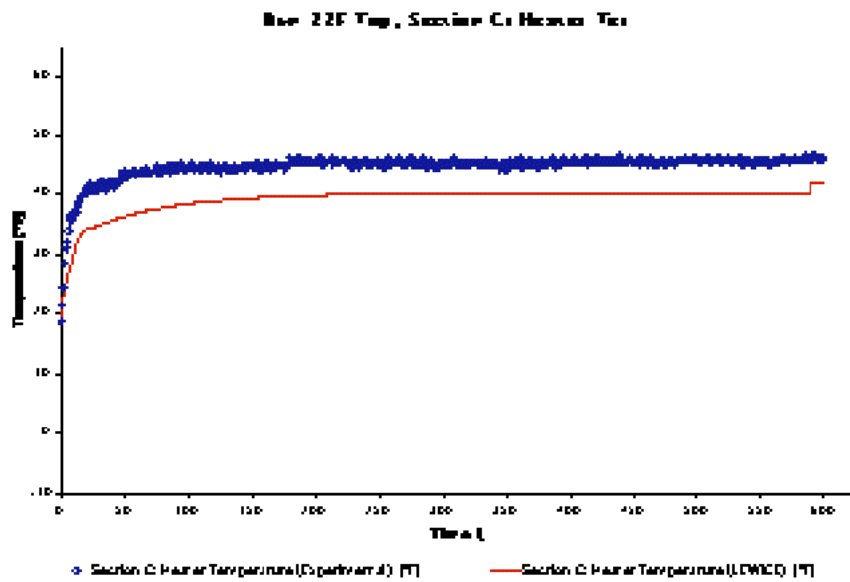


Figure 120: Heater C Temperature for Example Case 17

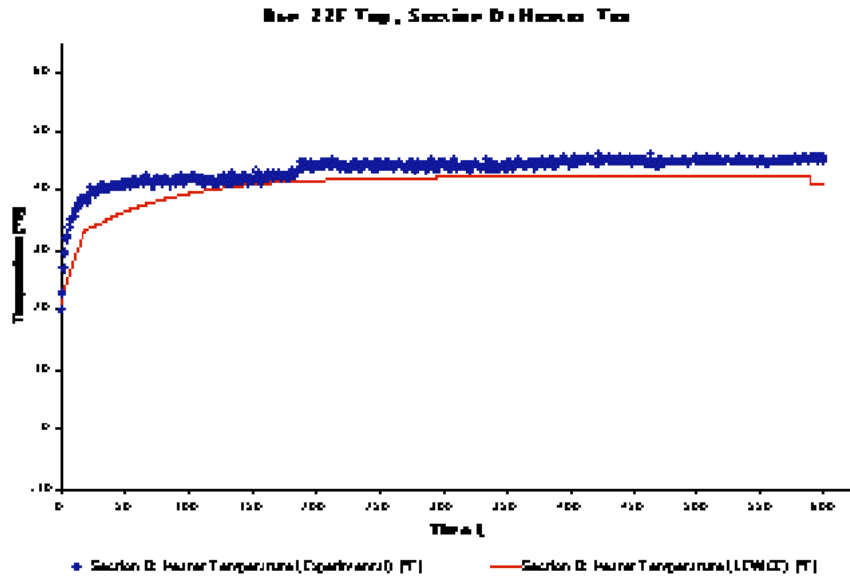


Figure 121: Heater D Temperature for Example Case 17

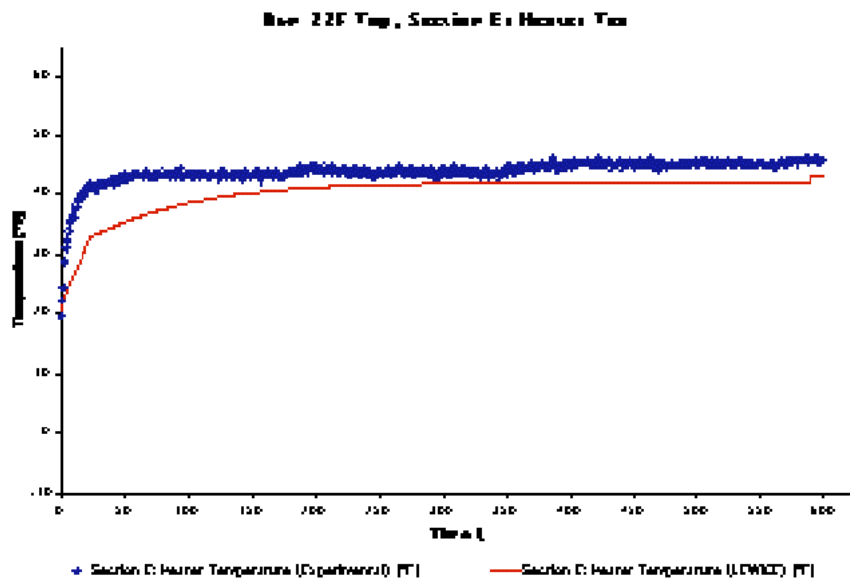


Figure 122: Heater E Temperature for Example Case 17

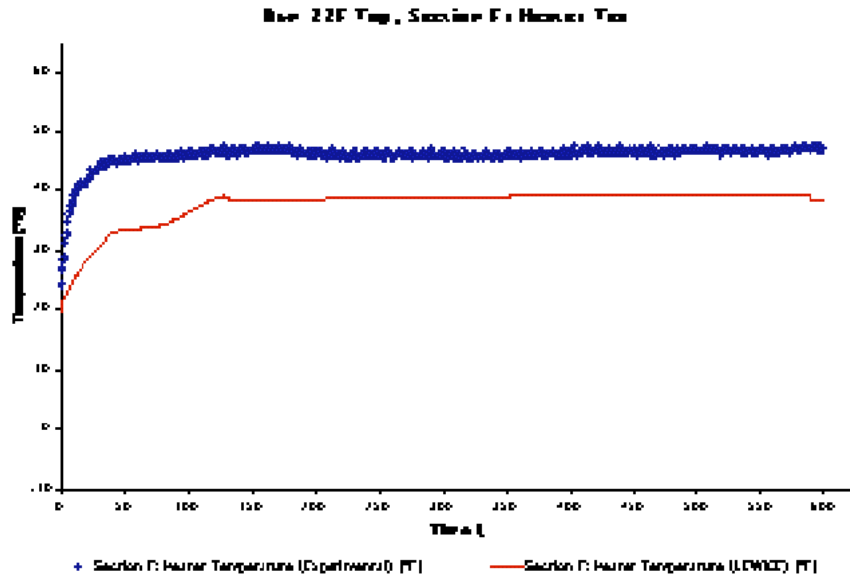


Figure 123: Heater F Temperature for Example Case 17

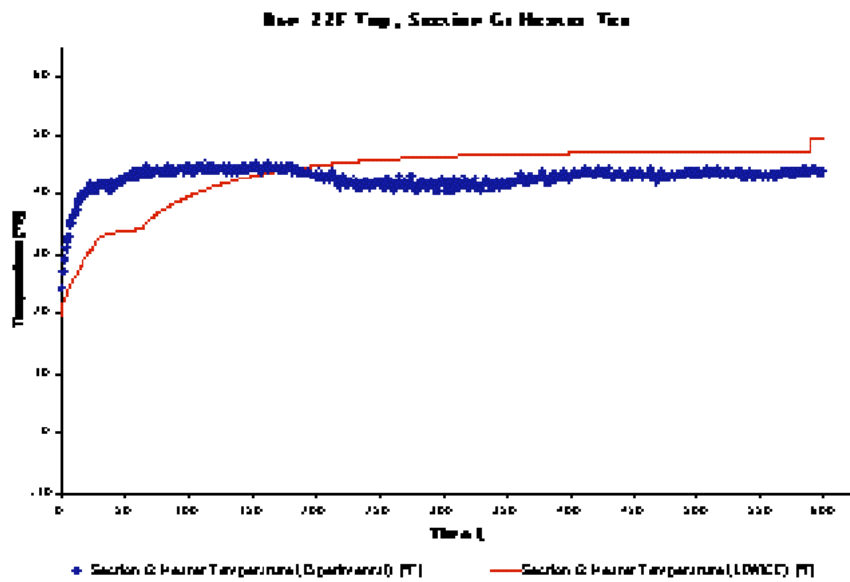


Figure 124: Heater G Temperature for Example Case 17

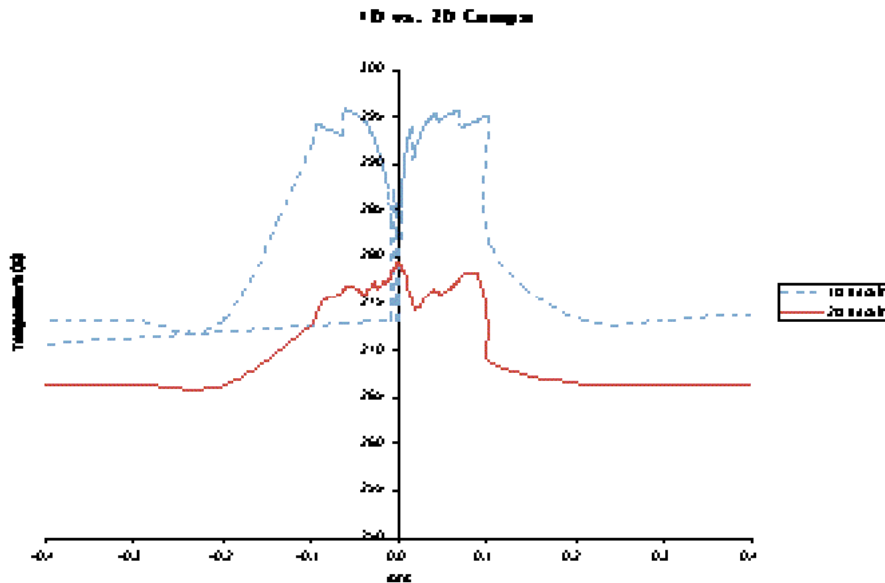


Figure 125: Comparison of One-Dimensional Assumption to 2D Result

15.18. Case 18: Evaporative Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 26 min.1 s.

Disk Space: 4.9 MB

This example case uses the same NACA0012 de-icer model as used in the previous two cases. In this example, however, the heaters are on for the duration of the simulation. The wattages used were high enough in the experiment (Run 22E) to completely evaporate all of the water within the de-icer region. In this mode, the electrothermal unit operates similarly to a hot air anti-icing system, albeit with specified heat fluxes instead of specified flow rates. In this example, the laminar heat transfer assumption was used since the de-icer region was completely free of ice.

The LEWICE results show an underprediction of the experimental results on the lower surface despite using the laminar assumption. The prediction on the upper surface and at the parting strip are reasonable considering the fluctuation of the heater wattages in the experiment. Since the heaters are offset slightly from the leading edge, the results will not be symmetric. Evaporative results were also obtained for all of the anti-icing cases in this database. While many of these cases show a similar result to this example, other cases showed an overprediction of temperature by a similar amount as the underprediction indicated in this case. This discrepancy would seem to indicate a physical effect which is not modeled that would account for this behavior. One possible cause is a change in the convective heat transfer coefficient due to the evaporation of water beyond that which is already modeled. The necessary heat transfer coefficient can be reverse-engineered for each case from the experimental data. However, the resulting values do not follow a known physical trend which could be placed in the model. Additionally, since the database contains only NACA0012 airfoils, most cases of which are at 0° AOA, it is unknown how applicable this experimentally adjusted coefficient would be outside this region. As a result, the adjusted coefficient was left out of the LEWICE 3.2 model.

Another feature modeled in this example is the comparison of the 1D anti-icing model to the 2D anti-icing model. This comparison can be achieved for any case using the IEVAP = 2 option in the de-icer input file. Many users have requested this option, as the 1D anti-icing model is known to overpredict heater temperatures. This comparison is shown in Figure 133. While the temperatures are higher in the 1D model, the difference is not as great as some have suggested, at least within the impingement limits.

Note: The heat transfer coefficients, pressure coefficients and collection efficiencies are the same as Example Case 16 and are not shown again.

Table 65: Main Input File for Example Case 18

Example 18 Test Case

```
&LEW20
ITIMFL = 0
TSTOP = 600
  IBOD = 1
  IFLO = 1
  DSMN = 4.0D-4
  NPL = 24
  IDEICE = 3
&END
&DIST
  FLWC = 1.0
  DPD = 20.
&END
&ICE1
CHORD = 0.9144
  AOA = 0.
  VINFL = 44.7
  LWC = 0.780
  TINFL = 265.50
  PINFL = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  EPRT = 1
  MPRT = 1
```

```

TPRT   = 0
IDBF   = 1
&END
&RDATA
&END
&BOOT
&END

```

Table 66: De-ice Input File for Example Case 18

```

1  nx
7  9
# of length conductivity diffusivity anisotropy slope (b) of
nodes (m) (W/m/K) (m**2/s) factor temp. eqn.
C  substrate
15  3.430d-3  0.120d0  1.652d-7  1.d0  0.d0
C  insulation
08  8.900d-4  0.294d0  1.045d-7  1.d0  0.d0
C  insulation
18  2.800d-4  0.256d0  1.473d-7  1.d0  0.d0
C  heater
07  1.300d-5  41.000d0  1.194d-5  1.d0  0.d0
C  insulation
18  2.800d-4  0.256d0  1.473d-7  1.d0  0.d0
C  abrasion shield
08  2.030d-4  16.270d0  4.035d-6  1.d0  0.d0
C  ice
61  2.540d-3  2.232d0  1.151d-6  1.d0  0.d0
# of length cond. diff. anis. slope of add. layer
nodes (m) (W/m/K) (m**2/s) temp. eqn. thick. number
C  gap (no heat input)
10  0.9244d0  0.256d0  1.637d-7  1.d0  0.d0  0.d0  3
C  heater G
14  3.175d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C  heater E
21  2.540d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C  heater C
21  2.540d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3
C  parting strip - heater A
36  1.905d-2  41.000d0  1.194d-5  1.d0  0.d0  0.d0  3

```

C	heater B						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater D						
21	2.540d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	heater F						
14	3.175d-2	41.000d0	1.194d-5	1.d0	0.d0	0.d0	3
C	gap (no heat input)						
10	0.9144d0	0.256d0	1.637d-7	1.d0	0.d0	0.d0	3

ijde offset ipar

004 0.000 1

C	Q	TON	TOFF	TLAG	ICFLAG
	heater density	time on	time off	lag time	temp cntrl.
	(kWatts/m**2)	(sec)	(sec)	(sec)	flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	12.41d0	600.d0	0.d0	0.d0	0
C	heater E				
	8.87d0	600.d0	0.d0	0.d0	0
C	heater C				
	19.26d0	600.d0	0.d0	0.d0	0
C	parting strip (heater A) does not turn off				
	46.58d0	600.d0	0.d0	0.d0	0
C	heater B				
	31.95d0	600.d0	0.d0	0.d0	0
C	heater D				
	12.70d0	600.d0	0.d0	0.d0	0
C	heater F				
	11.27d0	600.d0	0.d0	0.d0	0
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0

ibc1 ibc2 ibc3 ibc4

2 2 2 2

tg1 tg2 tg3 tg4

265.5 265.5 265.5 265.5

h1 h2 h3 h4

3.d0 3.d2 0.d0 0.d0

qw1 qw2 qw3 qw4

```

0.d0    0.d0    0.d0    0.d0
tairh  airmd   airup   xairmd
  5.d2  0.1d0   0.5d0   0.d0
nhole  dhole   dtube   rdist   ycase   xuhnd   xlhnd
  3    0.00132 0.0381  50.    0.0    0.23   0.26
icase  hconst  rep     rp     zp    rrp    dp
  1     1.0  0.76  1.394  1.    0.    0.
xhole(m)  yhole(m)  zdist(m)
.    0.0176   -0.033   0.0107
    0.000     0.000   0.0091
    0.0176   -0.033   0.0076
iliner  xuliner xlliner  xulst   xulnd  xllst  xllnd
  1     0.01   0.01   0.07546 0.1485 0.07546 0.1485
i3d    iqain  ibleed  iopt   itsur
  0     0     0     0     0
icond  ibound  init   ish    istd   igde
  1     3     2     1     1     1
tsurf  ievap  itherm
  328.d0  2     0
sta    rpm    irot
10.d0  0.d0  1
dtaui  nisp   dtaum  nmisp  dtauf  jcount
1.d-1  1000  1.d-1  1000  1.d-1  10
tspray  tafter
  0.d0  0.d0
iscol  jscol  kscol  sslope  szero
input parameters for reading in hot air HTC info
  2     3     3     1.d0   0.d0
input parameters for reading in hot air q_wall info
  1     2     2     1.d0   0.d0
input parameters for reading in surf. temperature info
  1     2     2     1.d0   0.d0
input parameters for reading in 3D streamline info
  1     2     2     1.d0   0.d0
nprt  nprf
  0    0
ntype
  3

```

ioctype	nptsd			
1	21			
C		section	where	layer
				where
		5	2	6
		5	2	1
		4	2	6
		4	2	1
		6	2	6
		6	2	1
		3	2	6
		3	2	1
		7	2	6
		7	2	1
		2	2	6
		2	2	1
		8	2	6
		8	2	1
		5	2	4
		4	2	4
		6	2	4
		3	2	4
		7	2	4
		2	2	4
		8	2	4

ioctype	nptsd			
2	30			
C				
C		time		layer
C		(sec)		where
		5		6
		10		6
		15		6
		20		6
		25		6
		30		6
		35		6
		40		6
		45		6

50	6	1
55	6	1
60	6	1
65	6	1
70	6	1
75	6	1
80	6	1
85	6	1
90	6	1
95	6	1
100	6	1
105	6	1
110	6	1
115	6	1
120	6	1
125	6	1
130	6	1
135	6	1
140	6	1
145	6	1
150	6	1

iotype nptsd

3 30

C	time	section	where
C	(sec)		
	5	5	2
	10	5	2
	15	5	2
	20	5	2
	25	5	2
	30	5	2
	35	5	2
	40	5	2
	45	5	2
	50	5	2
	55	5	2
	60	5	2
	65	5	2

70	5	2
75	5	2
80	5	2
85	5	2
90	5	2
95	5	2
100	5	2
105	5	2
110	5	2
115	5	2
120	5	2
125	5	2
130	5	2
135	5	2
140	5	2
145	5	2
150	5	2

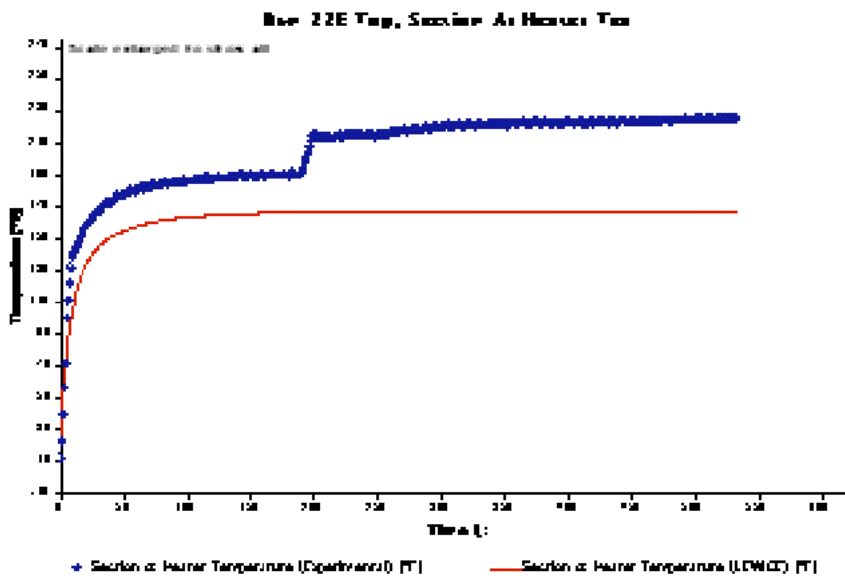


Figure 126: Heater A Temperature for Example Case 18

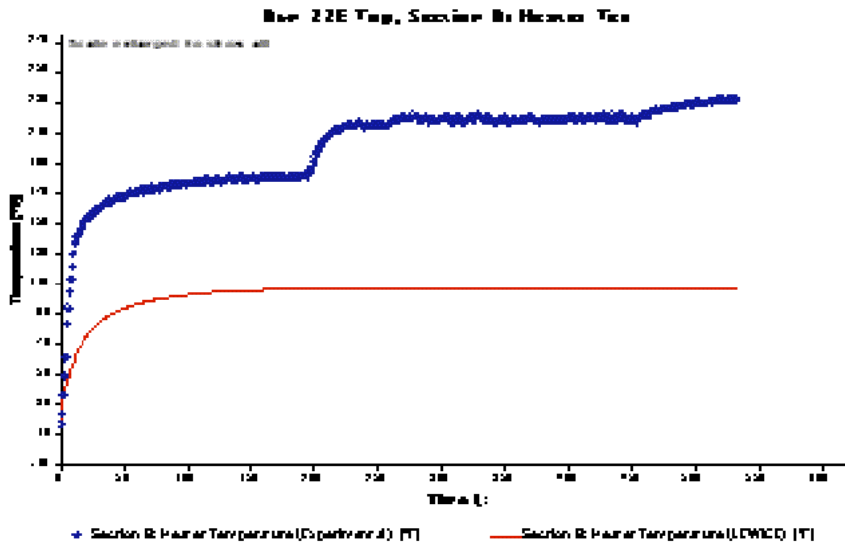


Figure 127: Heater B Temperature for Example Case 18

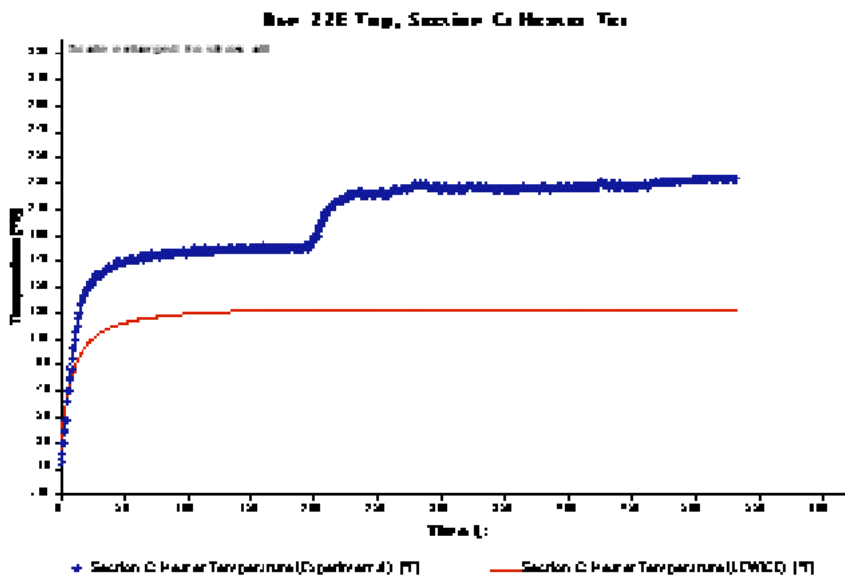


Figure 128: Heater C Temperature for Example Case 18

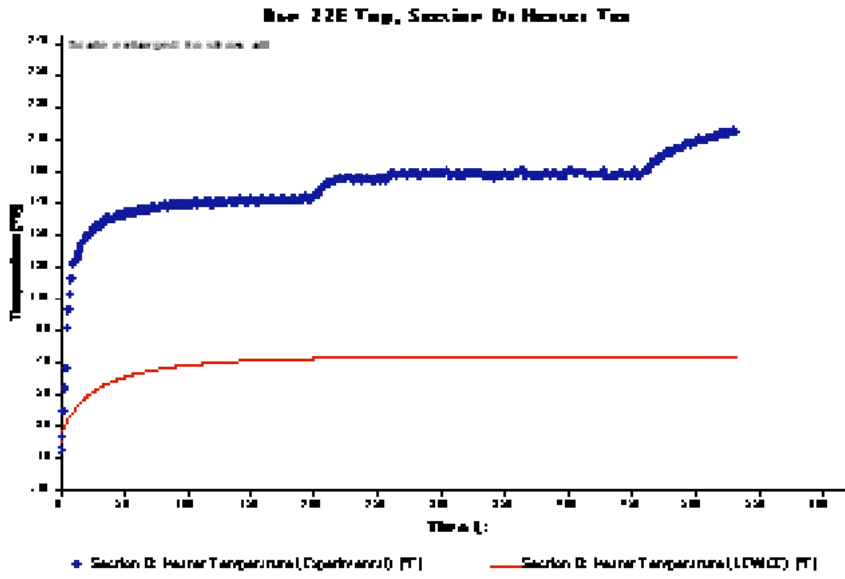


Figure 129: Heater D Temperature for Example Case 18

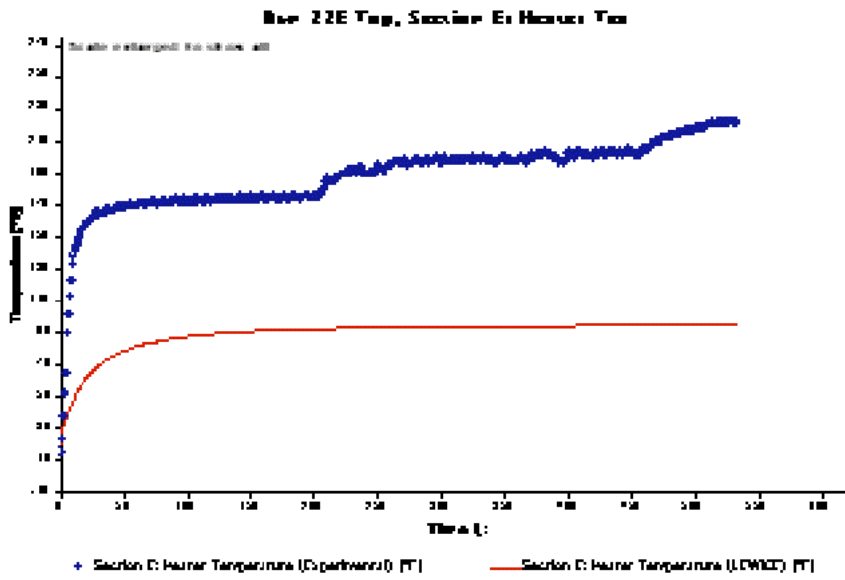


Figure 130: Heater E Temperature for Example Case 18

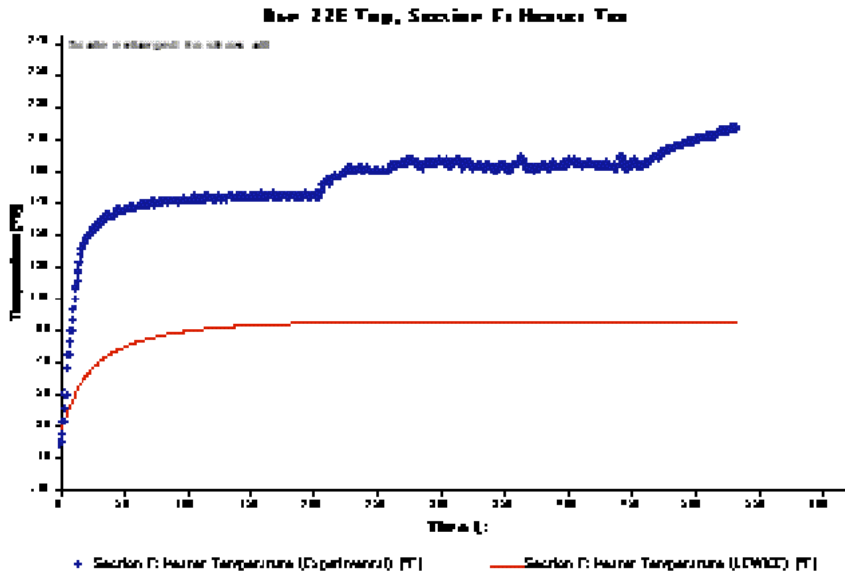


Figure 131: Heater F Temperature for Example Case 18

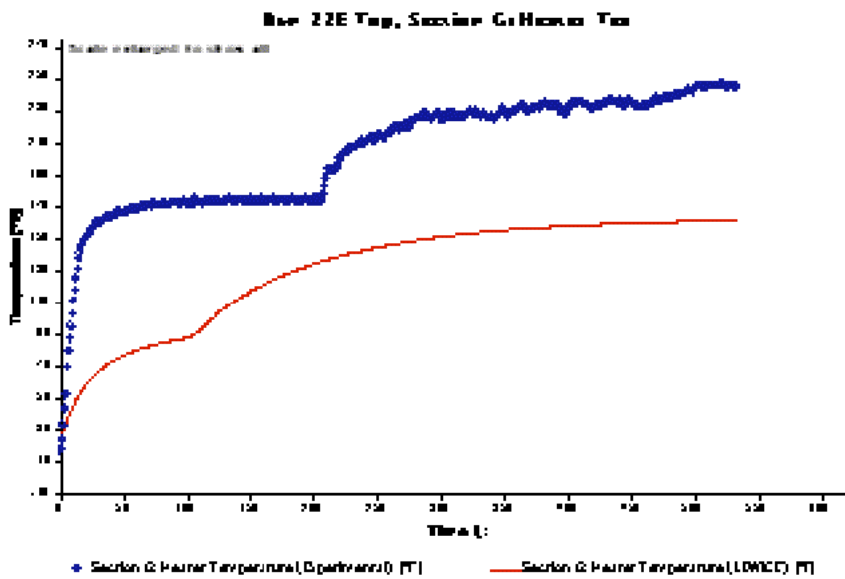


Figure 132: Heater G Temperature for Example Case 18

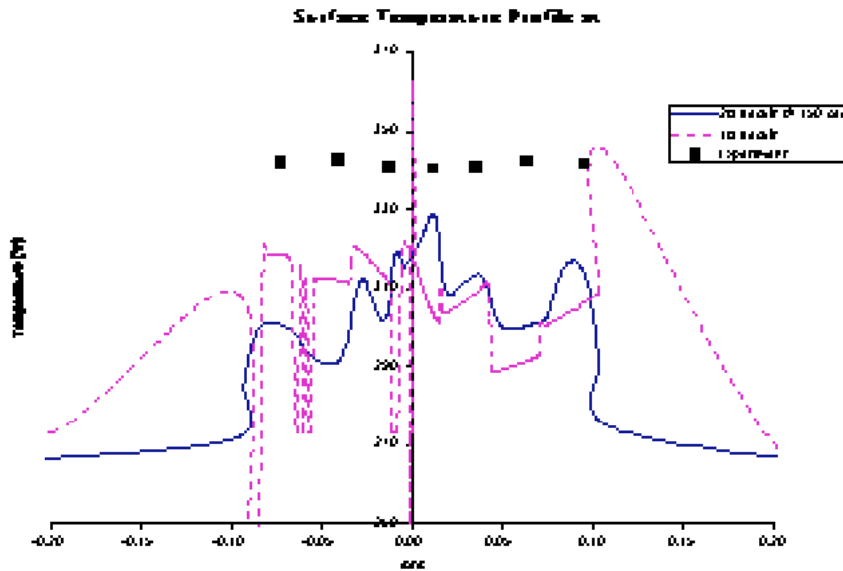


Figure 133: Comparison of 2D Anti-Icing Results to 1D Anti-Icing Results

15.19. Case 19: Hot Air Evaporative Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 48 min.37 s.

Disk Space: 22.9 MB

This example case uses a NACA0012 airfoil as the previous thermal cases, but in this example the electrothermal de-icer is replaced by the hot air model shown in Figure 134. This model was designed by NASA specifically to validate the bleed air capabilities in LEWICE and does not resemble any existing bleed air capability used by industry. The plenum section of the bleed air occupies most of the inside area of the airfoil within the impingement region. The air is then propelled to the inside surface via small holes which provide jet flow perpendicular to the surface. The bleed air then flows chordwise through a small channel to the exit. The test hardware has many more holes than a typical piccolo tube anti-icing system and the flow rates were well below choked flow. The advantage of this model was that the constant thickness of this channel made measurement of the internal heat transfer coefficients easier. Experimental data has been taken on the internal heat transfer coefficients for this model and are shown in Figure 135. The peaks in this curve show the locations of the air jets. The bleed air flow rate for this example was $0.11 \text{ kg}/(\text{m}^2\cdot\text{s})$ at a supply temperature of 440°F . This data is then used as an input to LEWICE 3.2. This example case uses the same LEWICE input case as the previous example, so that a direct comparison of the bleed air case could be made to the electrothermal case. No data is currently available for comparison for this type of anti-icing case.

Note: The external heat transfer coefficients, pressure coefficients and collection efficiencies are identical to the values in Example 18 and are not shown again in this example.

Note: All anti-icing cases are ran until TSTOP is reached or until steady-state convergence is achieved. This example case stops prior to the TSTOP value because the latter criteria was reached.

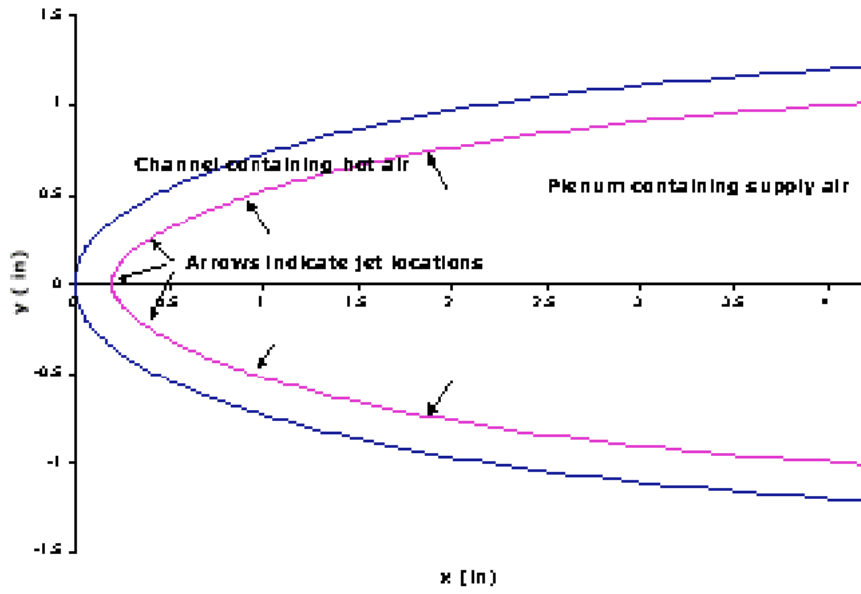


Figure 134: Jet Locations on Bleed Air Model

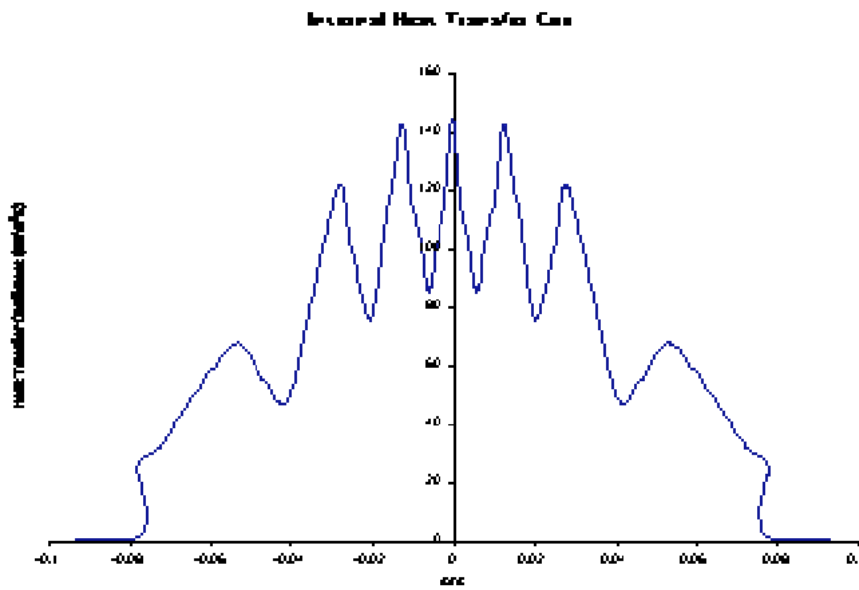


Figure 135: Internal Heat Transfer Coefficients for Bleed Air Case

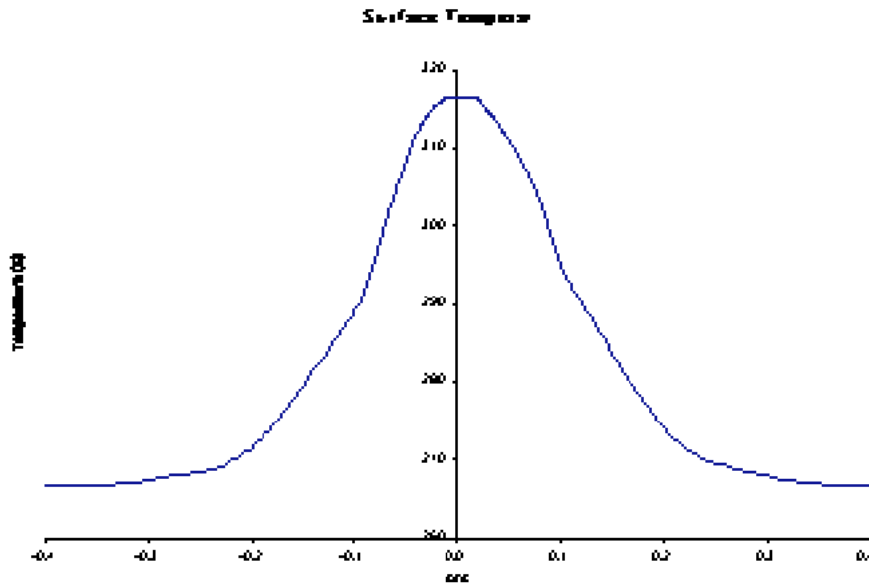


Figure 136: Surface Temperature Distribution for Bleed Air Case

15.20. Case 20: Hot Air Piccolo Tube Example Using Detailed Thermal Module

Computation Time: Pentium IV 2.4GHz, 53 min.6 s.

Disk Space: 21.8 MB

This example case uses a NACA0012 airfoil with bleed air as the previous example. In this case however, the bleed air heat transfer coefficients are obtained via a correlation rather than being read from an input file. The additional inputs needed for this option include the number of holes and hole locations. An inner liner was also used downstream from the holes. Mass flow rate and bleed air flow rate were also input. This case used an air flow rate of 0.01488 kg/m*s and a bleed air temperature of 453 K. Both values are much lower than the previous example, but are more typical for a piccolo tube anti-icing system. This example case uses the same LEWICE input case as the previous examples, so that a direct comparison of the piccolo tube case could be made to the electrothermal case and the previous bleed air example. The spanwise location (YCASE) chosen for this example was the location halfway between the jet locations and represents the worst case scenario for a piccolo tube system. Since the internal heat transfer will vary for different values of YCASE, a pseudo-3D solution can be obtained by running a number of 2D cases with different values of YCASE. This process was demonstrated in a recent report¹².

Note: The external heat transfer coefficients, pressure coefficients and collection efficiencies are identical to the values in Example 18 and are not shown again in this example.

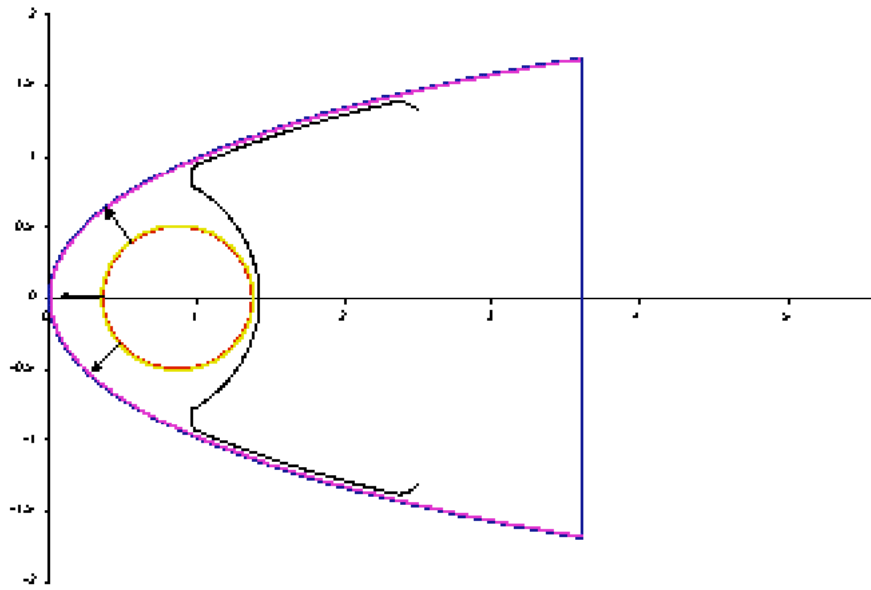


Figure 137: Piccolo Tube on Bleed Air Model

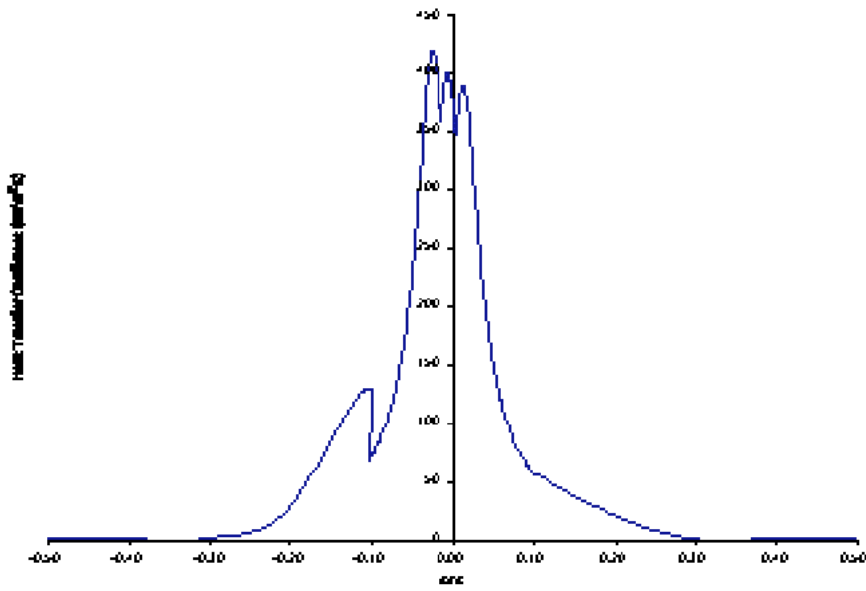


Figure 138: Internal Heat Transfer Coefficients for Bleed Air Case

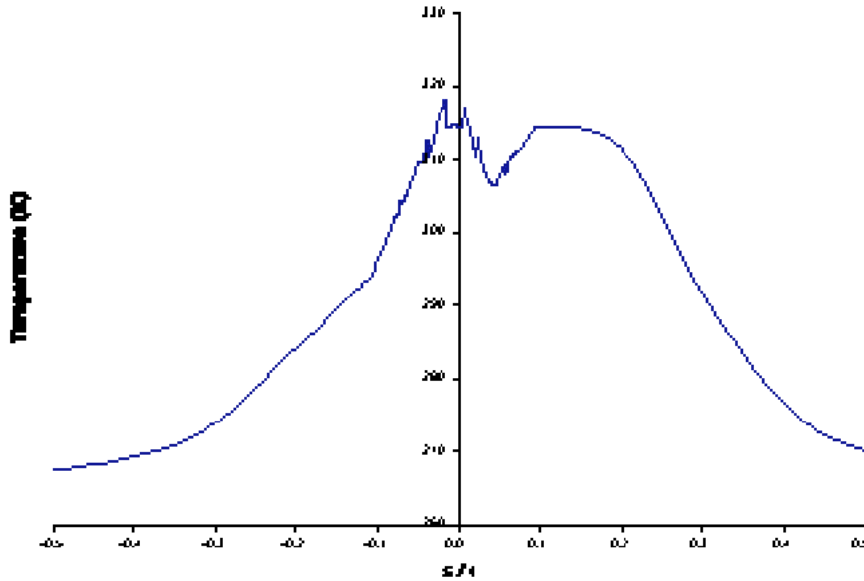


Figure 139: Surface Temperature Distribution for Bleed Air Case

15.21. Case 21: Hot Air Example Interfaced with LEWICE3D

Computation Time: Pentium IV 2.4GHz, 73 min.25 s.

Disk Space: 45.4 MB

This example case uses five 2D cross sections of an engine inlet as the body geometries. The first of these cross-sections is shown in Figure 140. This model was taken from a LEWICE3D example case. The surface pressure coefficients, collection efficiencies, and external heat transfer coefficients were supplied from a LEWICE3D output file at each 2D cross-section shown. The computation time shown included only the time needed to run the LEWICE 3.2 cases and does not include the computation time needed to generate the flow solution or the collection efficiencies. The internal heat fluxes for this model were determined by the same correlation as the previous example. Although the inlet cross sections are at different azimuthal locations, they were all assumed to be aligned with a piccolo tube hole. Since the inputs and results for each cross-section are very similar, only the inputs and results for the first section are shown in this report. The CD containing LEWICE 3.2 also contains the complete input and output files for this case as well as the previous cases. Figure 142 shows surface temperature predictions for the first section. The temperatures and the resultant residual ice can then be loaded into the utility program ANTICON2 for the generation of a simulated 3D residual ice shape. The simulated 3D ice shape is shown in Figure 148. The resolution of the ice shape in this figure can be improved by the addition of more streamlines to the model or by interpolation from the existing results. Additional streamlines would also allow the correlation to display a 3D variation in temperatures that is not seen in these results. Additionally, LEWICE 3.2 also contains a prediction of rivulet formation in its model, it should be possible to display that information as well. That process would require a much more sophisticated utility program than ANTICON2 and has not been developed yet.

Note: The computation time and disk space listed above includes only the resources needed to run the five thermal anti-icing cases. It does not include the resources needed to perform the 3D flow solution or collection efficiency analysis.

Note: If the bleed air is shut off, the user should get similar results to using the LEWICE3D program directly. A detailed discussion of the procedure and problems with this type of analysis follow this example case.

Note: Although pressure coefficients, heat transfer coefficients and collection efficiencies are inputs in this example, they are plotted and not tabulated like the other input files.

Table 67: Main Input File for Example Case 21-1

Example 21-1 Test Case

```
&LEW20
ITIMFL = 0
TSTOP = 180.
  IBOD = 1
  IFLO = 1
  DSMN = 4.0D-4
  NPL = 24
  IDEICE = 2
  ICP = 1
  IBETA = 1
  IHTC = 1
&END
&DIST
  FLWC = 1.0
  DPD = 20.36
&END
&ICE1
CHORD = 0.254
  AOA = 0.0
  VINFL = 76.86
  LWC = 0.200
  TINFL = 255.00
  PINFL = 84680.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1
```

```

BPRT = 1
EPRT = 1
MPRT = 1
TPRT = 0
IDBF = 1
&END
&RDATA
ISCOLB = 1
JSCOLB = 2
KSCOLB = 2
SSLOPB = 3.937d0
SZEROB = 1.078573d0
ISCOLH = 1
JSCOLH = 2
KSCOLH = 2
SSLOPH = 3.937d0
SZEROH = 1.078573d0
ISCOLC = 4
JSCOLC = 6
KSCOLC = 6
SSLOPC = 3.937d0
SZEROC = 1.078573d0
&END
&BOOT
&END

```

Table 68: De-ice Input File for Example Case 21-1

```

1  nx
2  10
# of length conductivity diffusivity anisotropy slope (b) of
nodes (m) (W/m/K) (m**2/s) factor temp. eqn.
C abrasion shield
18  2.030d-4  102.000d0  7.332d-5  1.d0  0.d0
C ice
21  2.540d-3  2.232d0  1.151d-6  1.d0  0.d0
# of length cond. diff. anis. slope of add. layer
nodes (m) (W/m/K) (m**2/s) temp. eqn. thick. number
C gap (no heat input)
10  0.1873d0  102.00d0  7.332d-5  1.d0  0.d0  0.d0  1

```

C	heater G						
14	3.175d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	heater E						
21	2.540d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	heater C						
21	2.540d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	parting strip - heater A						
18	0.9525d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	parting strip - heater A						
18	0.9525d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	heater B						
21	2.540d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	heater D						
21	2.540d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	heater F						
14	3.175d-2	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1
C	gap (no heat input)						
10	0.1873d0	102.00d0	7.332d-5	1.d0	0.d0	0.d0	1

ijde offset ipar1

001 0.000 1

C	Q	TON	TOFF	TLAG	ICFLAG
	heater density	time on	time off	lag time	temp cntrl.
	(kWatts/m**2)	(sec)	(sec)	(sec)	flag
C	gap (no heat input)				
	0.0d0	0.d0	0.d0	0.d0	0
C	heater G				
	00.00d0	00.d0	0.d0	0.d0	0
C	heater E				
	00.00d0	00.d0	0.d0	0.d0	0
C	heater C				
	08.00d0	278.d0	277.d0	0.d0	1
C	parting strip (heater A) does not turn off				
	8.38d0	278.d0	277.d0	0.d0	1
C	parting strip (heater A) does not turn off				
	1.75d0	278.d0	277.d0	0.d0	1
C	heater B				
	4.15d0	278.d0	277.d0	0.d0	1
C	heater D				

```

08.00d0      278.d0      277.d0      0.d0      1
C   heater F
00.00d0      00.d0      0.d0      0.d0      0
C   gap (no heat input)
0.0d0        0.d0      0.d0      0.d0      0
ibc1 ibc2 ibc3 ibc4
3   2   2   2
tg1   tg2   tg3   tg4
255.0 255.0 255.0 255.0
h1    h2    h3    h4
3.d0  3.d2  0.d0  0.d0
qw1   qw2   qw3   qw4
0.d0  0.d0  0.d0  0.d0
tairh airmd airup xairmd
5.d2  0.1d0 0.5d0 0.d0
nhole dhole dtube rdist ycase xuhnd xlhnd
3   0.00132 0.0381 50.  0.0  0.23  0.26
icase hconst rep  rp  zp rrp dp
1     1.0  0.76  1.394 1.  0.  0.
xhole(m)  yhole(m)  zdist(m)
.   0.0176   -0.033   0.0107
0.000     0.000   0.0091
0.0176   -0.033   0.0076
iliner xuliner xlliner xulst  xulnd xllst xllnd
1     0.01   0.01  0.07546 0.1485 0.07546 0.1485
i3d   iqain  ibleed iopt  itsur
1     0     0     1     0
icond ibound  init  ish  istd  igde
1     3     2     1     1     1
tsurf  ievap  itherm
278.0d0  0     1
sta    rpm   irot
10.d0  0.d0  1
dtaui nisp  dtaum  nmsp  dtauf  jcount
1.d-1 1000 1.d-1 1000 1.d-1  30
tspray tafter
0.d0  0.d0
iscol  jscol  kscol  sslope  szero

```

```

input parameters for reading in hot air HTC info
1      2      2      3.937d0      1.078573d0
input parameters for reading in hot air q_wall info
1      2      2      3.937d0      1.078573d0
input parameters for reading in surf. temperature info
1      2      2      3.937d0      1.078573d0
input parameters for reading in 3D streamline info
1      2      2      3.937d0      1.078573d0

```

```
nprt nprf
```

```
1 0
```

```
PLOT3D output times (sec)
```

```
180
```

```
ntype
```

```
3
```

```
ioctype nptsd
```

```
1 20
```

C	section	where	layer	where
	5	2	1	1
	4	2	1	1
	6	2	1	1
	3	2	1	1
	7	2	1	1
	2	2	1	1
	8	2	1	1
	1	2	1	1
	9	2	1	1
	10	2	1	1
	5	2	2	1
	4	2	2	1
	6	2	2	1
	3	2	2	1
	7	2	2	1
	2	2	2	1
	8	2	2	1
	1	2	2	1
	9	2	2	1
	10	2	2	1

```
ioctype nptsd
```

```

2      30
C
C      time      layer  where
C      (sec)
      5          2      1
      10         2      1
      15         2      1
      20         2      1
      25         2      1
      30         2      1
      35         2      1
      40         2      1
      45         2      1
      50         2      1
      55         2      1
      60         2      1
      65         2      1
      70         2      1
      75         2      1
      80         2      1
      85         2      1
      90         2      1
      95         2      1
     100         2      1
     105         2      1
     110         2      1
     115         2      1
     120         2      1
     125         2      1
     130         2      1
     135         2      1
     140         2      1
     145         2      1
     150         2      1

```

```

iotype nptsd
3      30
C      time      section  where
C      (sec)

```

5	5	2
10	5	2
15	5	2
20	5	2
25	5	2
30	5	2
35	5	2
40	5	2
45	5	2
50	5	2
55	5	2
60	5	2
65	5	2
70	5	2
75	5	2
80	5	2
85	5	2
90	5	2
95	5	2
100	5	2
105	5	2
110	5	2
115	5	2
120	5	2
125	5	2
130	5	2
135	5	2
140	5	2
145	5	2
150	5	2

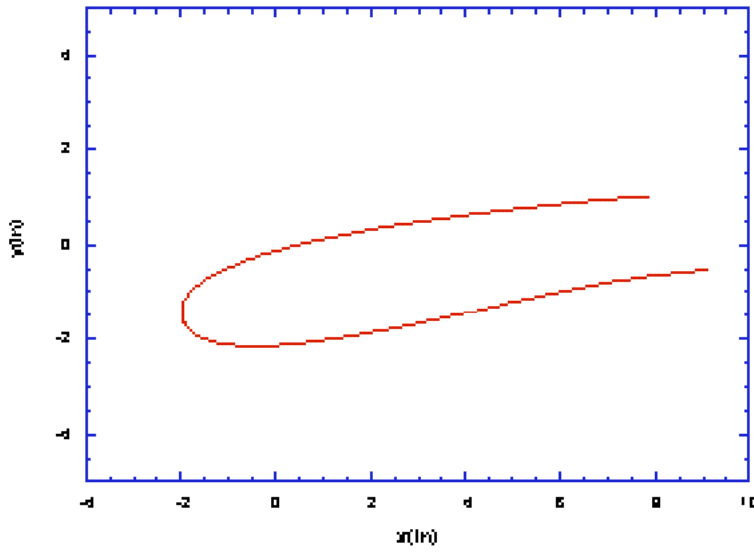


Figure 140: Engine Inlet Cross-Section for Example Case 21-1

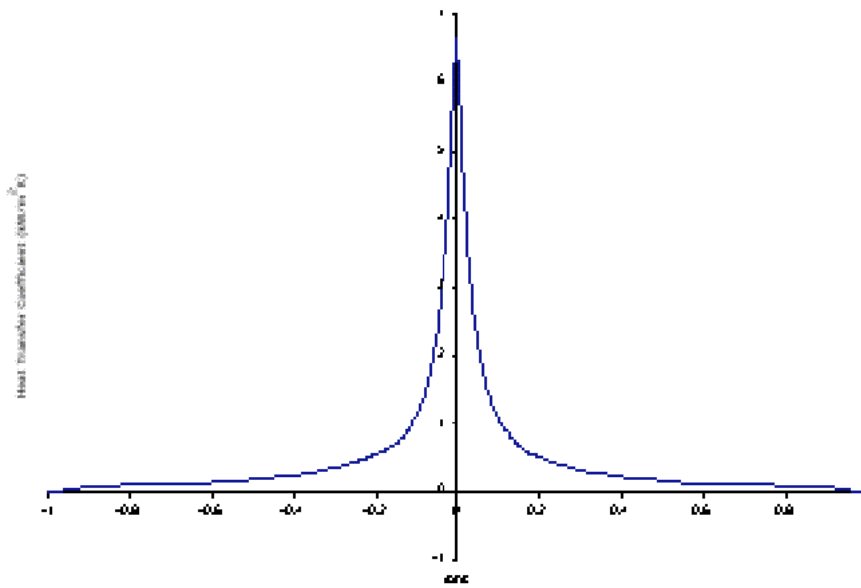


Figure 141: Internal Heat Transfer Coefficient for Example Case 21-1

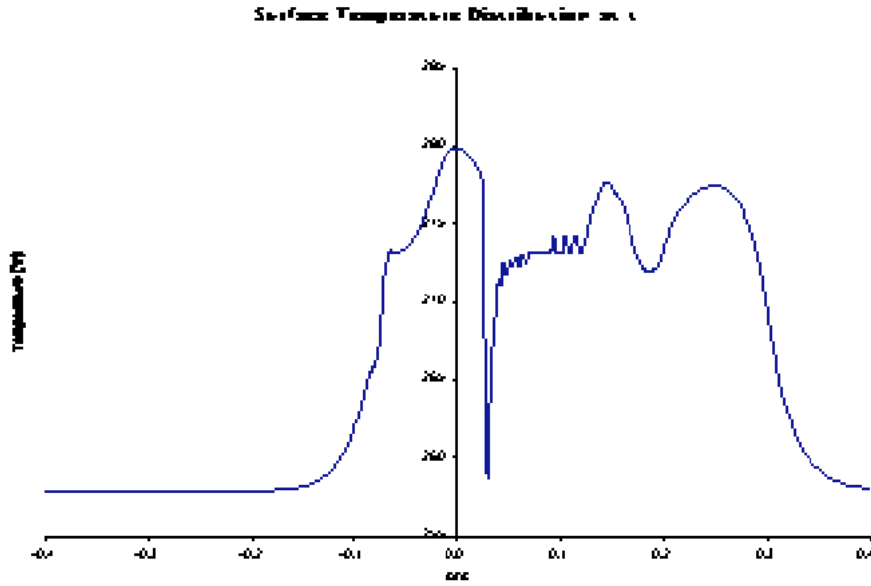


Figure 142: Surface Temperature Distribution for Example Case 21-1

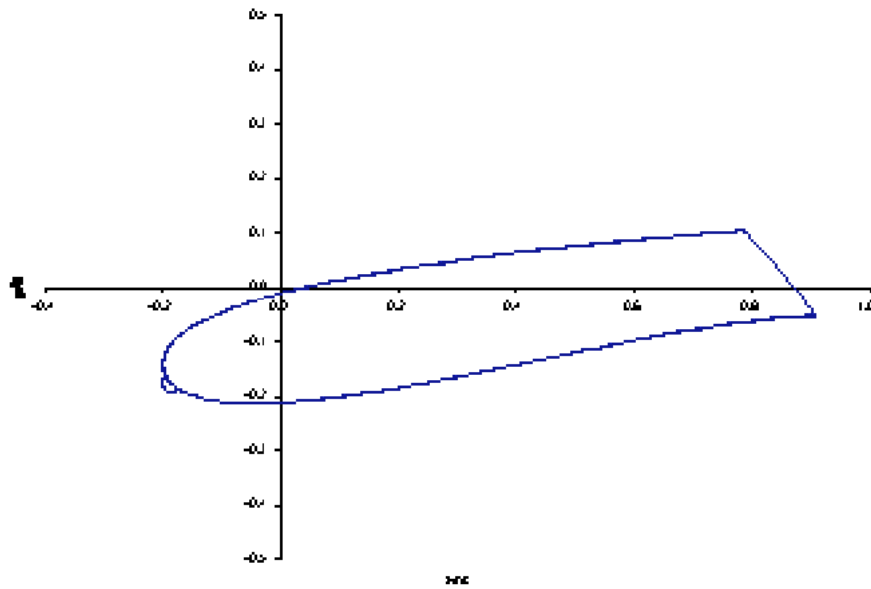


Figure 143: Ice Shape for Example Case 21-1

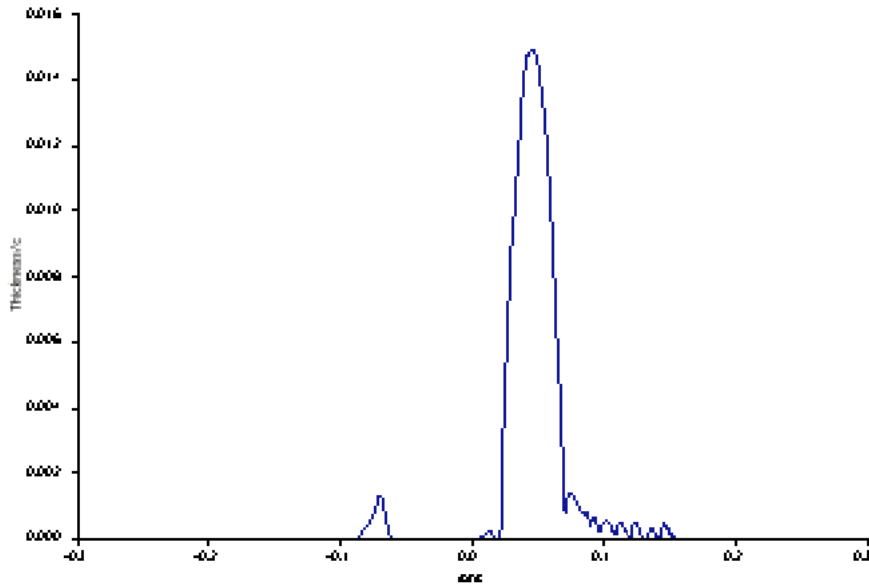


Figure 144: Ice Thickness for Example Case 21-1

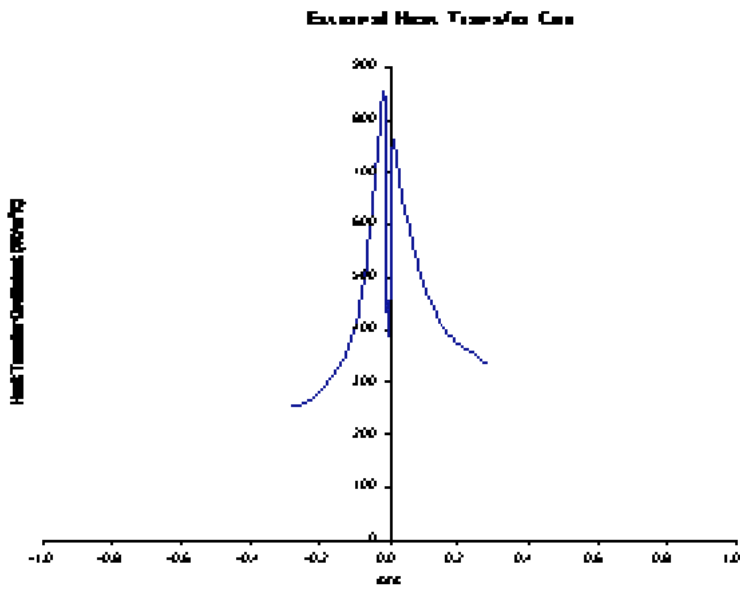


Figure 145: Heat Transfer Coefficient for Example Case 21-1

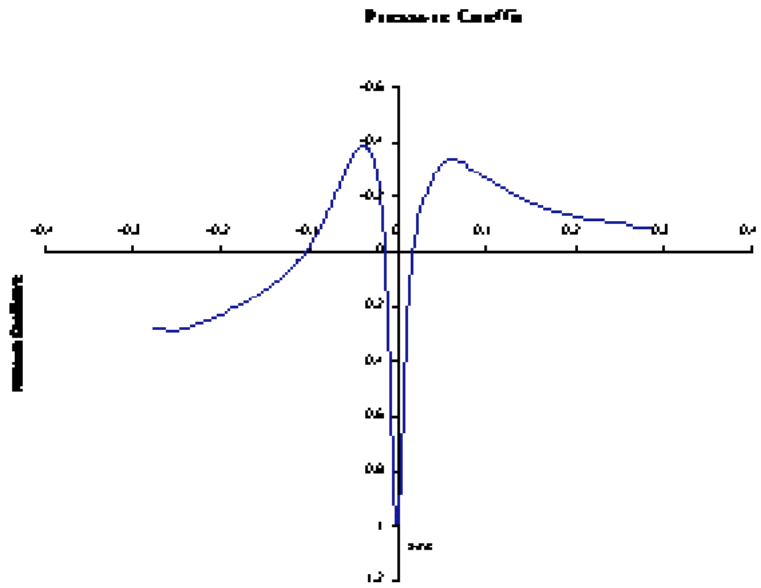


Figure 146: Pressure Coefficients for Example Case 21-1

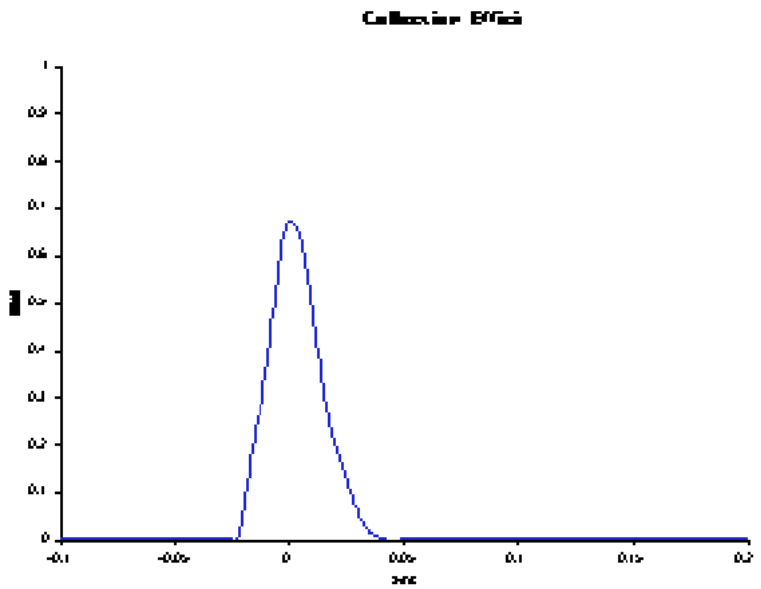


Figure 147: Collection Efficiency for Example Case 21-1

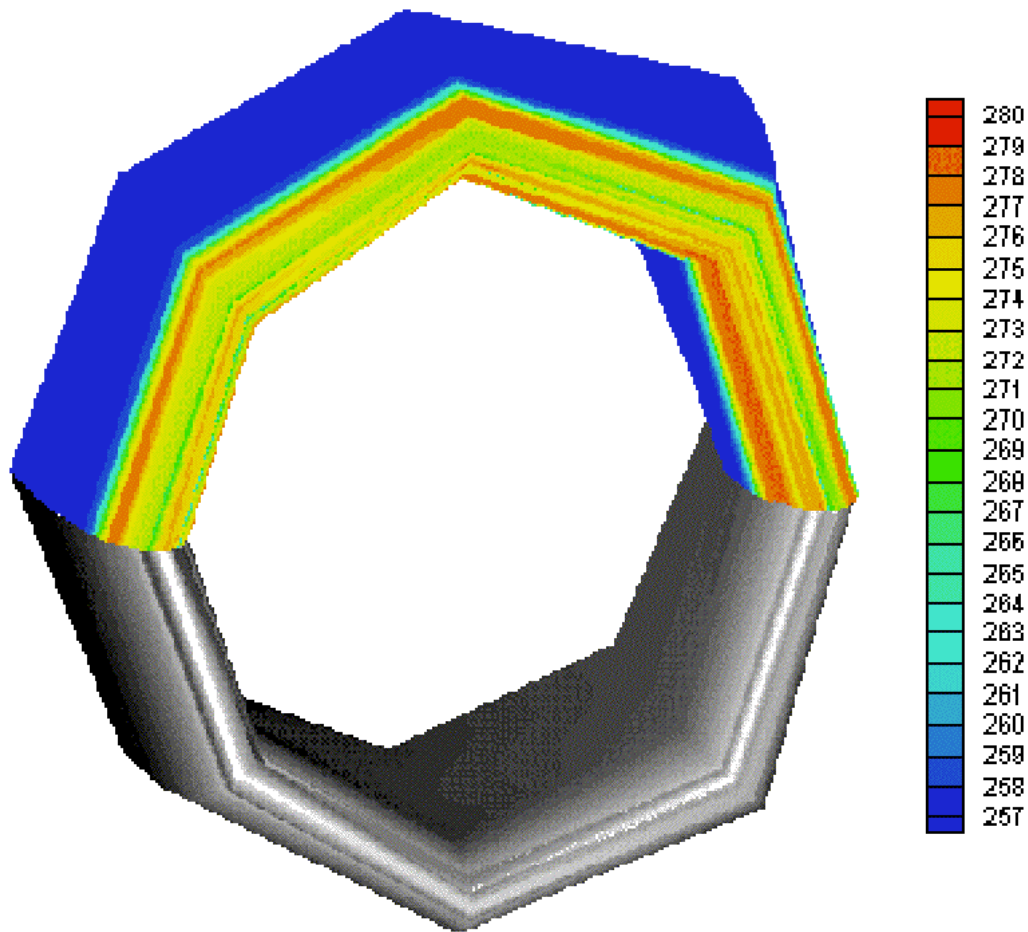


Figure 148: 3D Engine Inlet (Half-model) Surface Grid (Ice is Shaded)

15.21.1. Procedure for 2.5D Anti-Icing and De-Icing Analysis

This procedure will describe, in general, the process for obtaining a pseudo-3D anti-icing analysis using NASA software. The emphasis in this document is on the details of the anti-icing interface to the 3D software already used for icing analysis. These steps will not describe how to run a particular software. The user is referred to the User Manuals for the respective programs for details.

Step 1: Create a 3D surface model of your geometry. This is often performed by commercial software. Depending upon the requirements of your flow solver, you may need to use a CAD model, equations or points for this geometry

Step 2: Create a grid in 3D around the model. If using a potential flow solver, it may only be necessary to construct a “trajectory grid” for the particle trajectory program LEWI3DGR. Trajectory grids can be generated using programs supplied by NASA, while grids for Euler or Navier-Stokes are often generated using commercial software such as GRIDGEN.

Step 3: Create a 3D flow solution for the model. This flow solution can be potential flow, Euler, or Navier-Stokes depending upon your needs. The flow solvers can be commercial software packages such as VSAERO or FLUENT.

Step 4: Create a 3D collection efficiency distribution using LEWI3DGR. This may require using supplemental utility programs supplied by NASA to create “trajectory grids” or otherwise translate data from the flow module for use with NASA’s 3D trajectory modules.

Step 5 (optional): Calculate the wall heat flux or convective heat transfer coefficient on the surface of the model. This can either be calculated directly from a Navier-Stokes solution from

$$q_w = -\left(k \frac{\partial T}{\partial \eta}\right)_{\eta=0} \quad (21)$$

$$h = \frac{-\left(k \frac{\partial T}{\partial \eta}\right)_{\eta=0}}{(T_{\eta=0} - T_{\infty})} \quad (22)$$

or from using a 3D boundary layer program to get the convective heat transfer coefficient “h” or by using a 2D boundary layer program along streamline “cuts” as described in Step 6. This step is optional, as LEWICE 3.2 can generate “h” on the streamline “cuts”.

Step 6: Run the conversion program ANTICON to create input files compatible with LEWICE 3.2. This requires the files “fort.2” and “fort.26” from the LEWI3DGR solution.

Note: For inlets, a customized version of this program was required to obtain the 2D “cuts” normal to the local geometry, not simply taking the ‘X’ and ‘Z’ coordinates as is normally done by that software. The customized utility ANTICON is on the distribution disk. This program creates the files geom.xyd, “flow.dat”, “beta.dat”, “htc.dat”, and “stream.dat”. “stream.dat” contains the spanwise distance between streamlines at each chord location.

Step 7: Using a text editor, separate the ANTICON output into several files, one for each 2D “cut” and file type. For example, if there are five 2D “cuts” generated, the user will need to generate five geometry files, five flow solution files and so on. Each of these files must contain only the output data. Any text headers must be removed prior to running LEWICE 3.2. The general format for each file is as follows:

```
(number of points)
wrap distance1(s1)      data value1
wrap distance2(s2)      data value2
wrap distance3(s3)      data vaulle3
...
```

Step 8 (optional): Create a CFD model of the internal hot air bleed flow using a procedure similar to Steps 1-3 which are used for generating the external flow field. For each of the 2D “cuts” defined in Step 6, generate a file containing either the wall heat flux or wall heat transfer coefficient calculated by the internal flow software. These are defined by Equations 21 and 22 earlier. This step is optional, as the user may be modelling an electrothermal system or has selected the correlation in LEWICE for heat transfer coefficient.

Step 9: Create a 2D anti-icing model for each of the “cuts” generated in the previous step. Since the same system in each section is probably being modeled, the input files may be identical, but they are not required to be so.

Step 10: Run LEWICE 3.2 for each of the 2D “cuts” which have been created. Create directories to store each of the cases, as the output files generated by LEWICE 3.2 will be the same for each case. Set the flags in the LEWICE 3.2 input file to bypass calculation of the flow field (ICP = 1), collection efficiency (IBETA = 1) and heat transfer coefficient (both external [IHTC = 1] and internal [IBLEED = 1] if desired). Move the output for each run into the directory created for it prior to the next case.

Step 11: Concatenate the output files “**dyice.dat**” and “**temp.dat**” created from each of the streamline “cuts”. These files must also be called “**dyice.dat**” and “**temp.dat**”. Run the conversion program ANTICON2 which will combine the outputs from LEWICE 3.2 and LEWI3DGR and create a PLOT3D output file for the surface geometry, the iced geometry, and the temperature distribution on the surface. This requires the “fort.2” and “fort.26” output files used in Step 6.

Step 12: Plot the output files PLOT3D.XYZ and PLOT3D.Q in PLOT3D or a program which can read those files. Since PLOT3D was designed to visualize a flow solution, not a temperature solution, temperature is output from ANTICON2 in the “density” field as density and temperature are both scalar variables. The other PLOT3D information in PLOT3D.Q contains garbage values. The output files are read in PLOT3D using the string “read/mg/blank/form/xyz=PLOT3D.XYZ/ q=PLOT3D.Q”. An advanced user/developer could also write a small utility program to convert these output files into a different format such as the format used by TECPLOT.

15.22. Case 22: Grid input

Computation Time: Pentium IV 2.4GHz, 25 s.

Disk Space: 1 MB

This example case illustrates a feature of LEWICE whereby the user can 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 Appendix A. This process may require modifications both to LEWICE and to the flow solver, therefore it is best left to more advanced users or software 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 69 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 GRIDGEN²⁵. LEWICE 3.2 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 software to handle unstructured grids.

The flow solution provided was generated by the Navier-Stokes software NPARC²⁶. 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 software packages may also be used to generate the grid and flow solution. The programs 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 69: Main Input File for Example Case 22

Example 22 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
VINFL  = 76.
LWC    = 0.80
TINFL  = 258.00
PINFL  = 100000.00
RH     = 100.0
```



```

&END
&LPRNT
  FPRT = 1
  HPRT = 1
  BPRT = 1
  TPRT = 0
&END
&RDATA
&END
&BOOT
&END

```

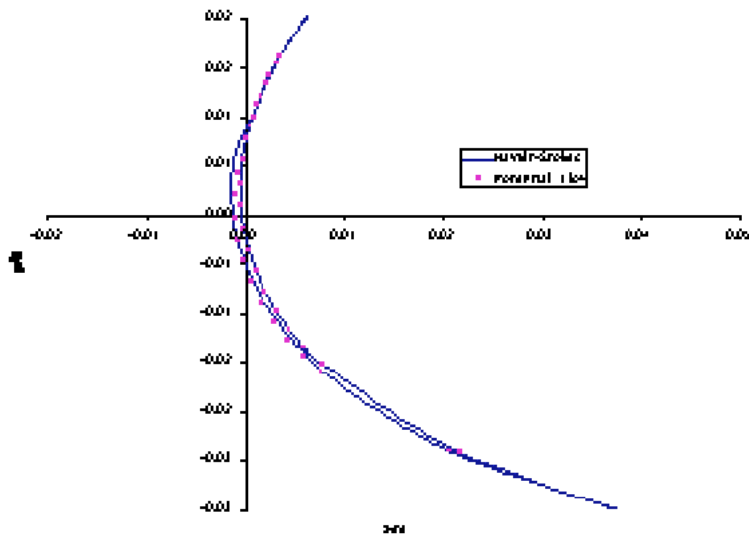


Figure 149: Ice Shape for Example Case 22

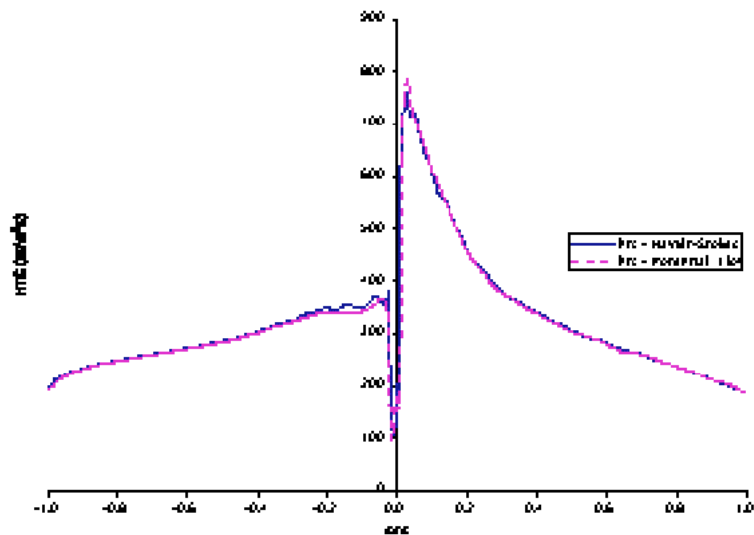


Figure 150: Ice Thickness for Example Case 22

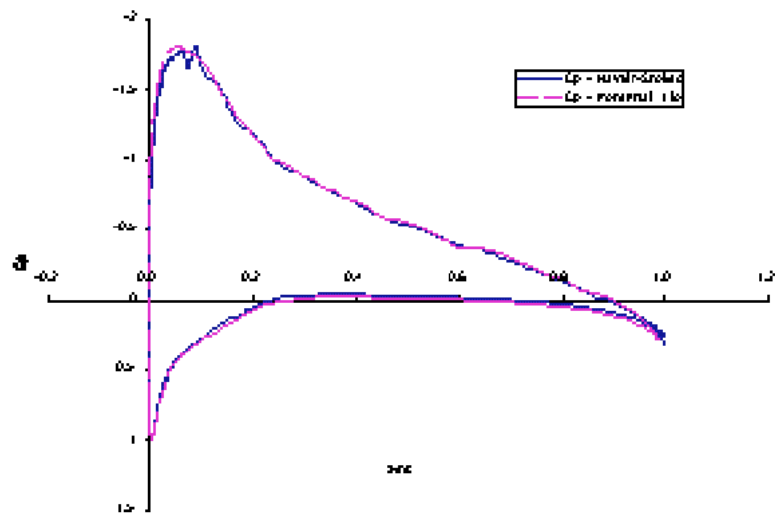


Figure 151: Heat Transfer Coefficient for Example Case 22

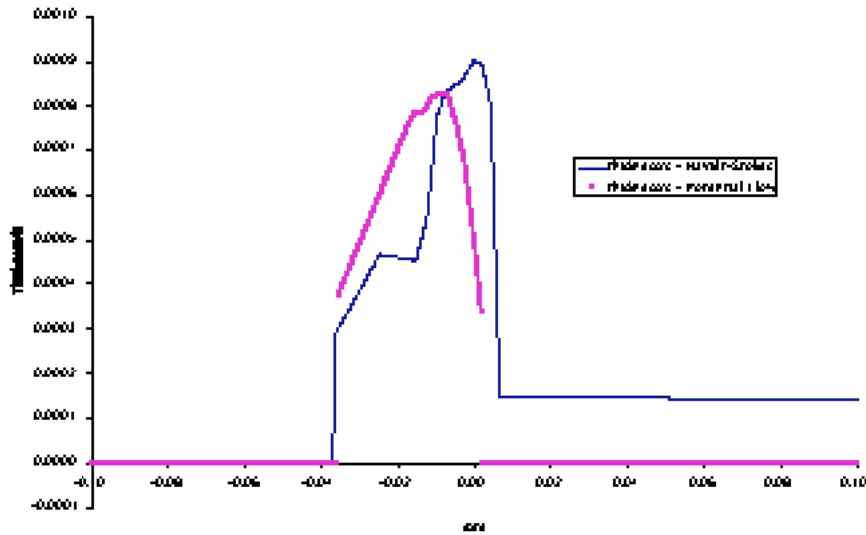


Figure 152: Pressure Coefficient for Example Case 22

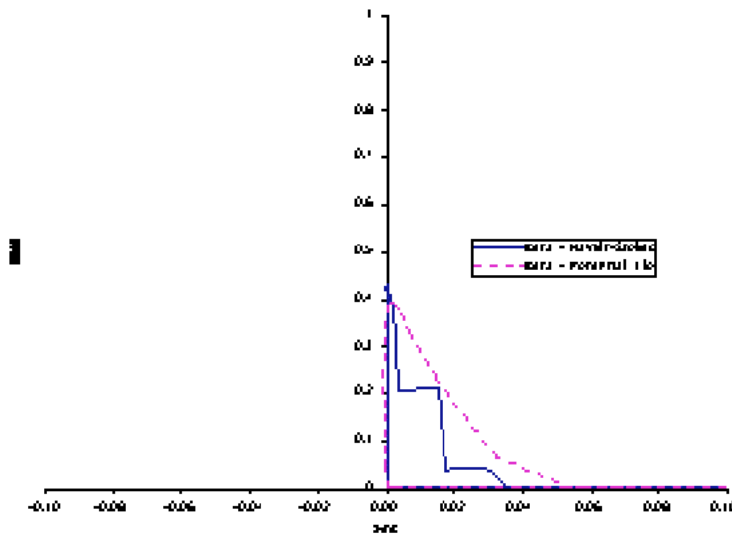


Figure 153: Collection Efficiency for Example Case 22

15.22.1. Grid Input Problems

This section describes the problems encountered when trying to import the NPARC flow solution into LEWICE for Example Case 22. 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. See Example 23 for an alternate path for incorporating Navier-Stokes analysis.

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 Manual²⁷. 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 program 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 been divided by the ambient Mach number while the fourth 'Q' function was multiplied by γ (1.4) as compared to the quantities expected by LEWICE.

Problem 4: Platform Problems

Once this conversion was made, the case ran successfully on an SGI Octane. 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.

15.22.2. 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 software 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 software ran without incident.

15.23. Case 23: Automated Naviér-Stokes Ice Accretion Using ICEG2D

Computation Time: SGI Octane R12000 Single Processor, 6 hrs., 30 min., 24 sec.

Disk Space: 128.2 MB

This example case illustrates a new feature of LEWICE. In this example a multi-time step ice accretion is generated by calling LEWICE and the flow solver multiple times using a batch script. The process is similar to the previous example in that a Naviér-Stokes flow solver is used instead of the potential flow solver in LEWICE. In order to accomplish a multi-time step ice accretion, the Naviér-Stokes solver requires a new grid for each time step due to the changing shape of the airfoil. The batch script works by automating the grid generation using the software ICEG2D¹⁰. ICEG2D automatically generates a structured C-type grid for each geometry and generates input files for the flow solver and for LEWICE. It can only be used for single body geometries and runs on the SGI platform. The software is being ported to LINUX and other platforms.

Another drawback of this feature is that since the flow solver is not integrated into the model, only a single time step can be performed with LEWICE and multiple time steps can only be accomplished using the script. If the script does not run properly, it is often difficult to determine exactly where the failure occurred without going through the procedure again using interactive inputs. Performing each step interactively defeats the purpose of the automatic grid generation since the user must monitor the process. This process has not been thoroughly tested and is recommended only for users who are familiar with both LEWICE and the flow solver being used. Currently, the process supports the NPARC and WIND solvers. The example case provided uses the WIND⁹ solver. It should be noted that WIND is not provided on the LEWICE CD and must be requested separately from the NPARC Alliance. It is likely that modifications must be made to the ICEG2D source code in order to support other flow solvers.

The computation times listed for this case were obtained by summing the times taken by LEWICE and WIND. Interim steps such as grid generation are nearly instantaneous by comparison and were not included. Table 71 shows the main LEWICE input file for the first time step of this case. Table 72 shows the ICEG2D input file that contains information needed by the flow solver. Table 73 shows the batch script used to run ICEG2D, LEWICE and WIND. The next section will cover this script in more detail. The complete inputs for this case are included on the distribution disk. 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.

Figure 154 shows the predicted ice shape for this case and for the case that uses only potential flow. Other figures show the comparison between potential flow and Naviér-Stokes for pressure coefficient, collection efficiency and heat transfer coefficient. While there is a difference between the two ice shapes, some may question if this difference was worth the extra computation time. A more thorough parameter study utilizing cases with high speeds and/or high angles of attack are needed to investigate the utility of this feature. There are seven main input files for this case, one for each time step. They are called “**case23_1.inp**” through

“**case23_7.inp**” on the distribution CD. A de-icer input file, “**case23d_lew.inp**”, is required by the ICEG2D script even though de-icing is not used in this case. The ICEG2D script file is called “**case23.inp**”, the ICEG2D initial input file is called “**case23_start.inf**” and the ICEG2D restart file is called “**case23_start.inf**”. The ICEG2D geometry input file is called “**case23.cln**” to distinguish it from other geometry input files as noted below.

Note: The geometry input file format for this case conforms to the ICEG2D geometry format rather than the LEWICE geometry format. The ICEG2D geometry input file includes the number of blocks and the number of points as well as the x-coordinate and y-coordinate values.

Note: The disk storage for this case can be reduced significantly by erasing interim output files if they are not needed.

Table 70: Main Input File for Example Case 23

Example 23-1 Test Case

```

&LEW20
TSTOP = 60.
  TSTART = 0.
  IBOD = 1
  IFLO = 1
  IGRID = 1
  DSMN = 4.0D-4
  NPL = 24
  IHTC = 1
&END
&DIST
  FLWC = 1.0
  DPD = 20.
&END
&ICE1
CHORD = .5334
  AOA = 3.5
  VINFL = 102.8
  LWC = 0.55
  TINFL = 262.04
  PINFL = 100000.00
  RH = 100.0
&END
&LPRNT
  FPRT = 1
  HPRT = 1

```

```

BPRT    = 1
TPRT    = 0
KWARN   = 1
&END
&RDATA
ISCOLH  = 2
JSCOLH  = 3
KSCOLH  = 3
SSLOPH  = 1.
SZEROH  = 1.02
&END
&BOOT
&END

```

Table 71: ICEG2D Input File for Example Case 23

```

$TITLES
title(1) = "NACA0012 - Run 405",
title(2) = "Mach = 0.3167, alpha = 3.5 deg.",
title(3) = "Single-block, C-grid.",
$END
$INPUTS
chord = 1.75,
mach = 0.3167,
alpha = 3.5,
pressureStatic = 14.7,
temperatureStatic = 471.67,
isothermalWall = .true.,
wallTemperature = 491.67,
turbulenceModel = "spalart allmaras",
order = 5,
cycles = 200,
iterationsPerCycle = 10,
cfl = 10.0
version = 1
$END

```

Table 72: ICEG2D Batch Script for Example Case 23

```

def mode batch
def case case23

```

```

def surf smooth
def solver wind
#
# initial clean solution
#
def iced_geom case23.cln
gen dist
gen grid
def info case23_start.inf
def output case23_start
gen input
gen solution
#
def info case23_restart.inf
def input_grid case23.grd
def input_q case23.q
#
# Preprocessing for ice accretion step #1
#
def lewice_input case23_1.inp
gen lewice_input
#
# Ice accretion step #1
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #1
#
gen grid_update
def iced_geom updated_airfoil
def dist 1-minute_shape.dst
gen dist
gen grid
def output case23_restart1.out
gen input
gen restart_update
#
# run wind for ice accretion step #1

```



```

#
gen solution
#
# Preprocessing for ice accretion step #2
#
def lewice_input case23_2.inp
gen lewice_input
#
# Ice accretion step #2
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #2
#
gen grid_update
def iced_geom updated_airfoil
def dist 2-minute_shape.dst
gen dist
gen grid
def output case23_restart2.out
gen input
gen restart_update
#
# run wind for ice accretion step #2
#
gen solution
#
# Preprocessing for ice accretion step #3
#
def lewice_input case23_3.inp
gen lewice_input
#
# Ice accretion step #3
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #3
#

```

```

gen grid_update
def iced_geom updated_airfoil
def dist 3-minute_shape.dst
gen dist
gen grid
def output case23_restart3.out
gen input
gen restart_update
#
# run wind for ice accretion step #3
#
gen solution
#
# Preprocessing for ice accretion step #4
#
def lewice_input case23_4.inp
gen lewice_input
#
# Ice accretion step #4
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #4
#
gen grid_update
def iced_geom updated_airfoil
def dist 4-minute_shape.dst
gen dist
gen grid
def output case23_restart4.out
gen input
gen restart_update
#
# run wind for ice accretion step #4
#
gen solution
#
# Preprocessing for ice accretion step #5

```

```

#
def lewice_input case23_5.inp
gen lewice_input
#
# Ice accretion step #5
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #5
#
gen grid_update
def iced_geom updated_airfoil
def dist 5-minute_shape.dst
gen dist
gen grid
def output case23_restart5.out
gen input
gen restart_update
#
# run wind for ice accretion step #5
#
gen solution
#
# Preprocessing for ice accretion step #6
#
def lewice_input case23_6.inp
gen lewice_input
#
# Ice accretion step #6
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #6
#
gen grid_update
def iced_geom updated_airfoil
def dist 6-minute_shape.dst
gen dist

```

```

gen grid
def output case23_restart6.out
gen input
gen restart_update
#
# run wind for ice accretion step #6
#
gen solution
#
# Preprocessing for ice accretion step #7
#
def lewice_input case23_7.inp
gen lewice_input
#
# Ice accretion step #7
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #7
#
gen grid_update
def iced_geom updated_airfoil
def dist 7-minute_shape.dst
gen dist
gen grid
def output case23_restart7.out
gen input
gen restart_update
#
# run wind for ice accretion step #7
#
gen solution
#
# Preprocessing for ice accretion step #8
#
def lewice_input case23_8.inp
gen lewice_input
#

```

```

# Ice accretion step #8
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #8
#
gen grid_update
def iced_geom updated_airfoil
def dist 8-minute_shape.dst
gen dist
gen grid
def output case23_restart8.out
gen input
gen restart_update
#
# run wind for ice accretion step #8
#
gen solution
#
# Preprocessing for ice accretion step #9
#
def lewice_input case23_9.inp
gen lewice_input
#
# Ice accretion step #9
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #9
#
gen grid_update
def iced_geom updated_airfoil
def dist 9-minute_shape.dst
gen dist
gen grid
def output case23_restart9.out
gen input
gen restart_update

```

```

#
# run wind for ice accretion step #9
#
gen solution
#
# Preprocessing for ice accretion step #10
#
def lewice_input case23_10.inp
gen lewice_input
#
# Ice accretion step #10
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #10
#
gen grid_update
def iced_geom updated_airfoil
def dist 10-minute_shape.dst
gen dist
gen grid
def output case23_restart10.out
gen input
gen restart_update
#
# run wind for ice accretion step #10
#
gen solution
#
# Preprocessing for ice accretion step #11
#
def lewice_input case23_11.inp
gen lewice_input
#
# Ice accretion step #11
#
gen lewice
#

```

```

# Preprocessing for flow solution using ice accretion #11
#
gen grid_update
def iced_geom updated_airfoil
def dist 11-minute_shape.dst
gen dist
gen grid
def output case23_restart11.out
gen input
gen restart_update
#
# run wind for ice accretion step #11
#
gen solution
#
# Preprocessing for ice accretion step #12
#
def lewice_input case23_12.inp
gen lewice_input
#
# Ice accretion step #12
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #12
#
gen grid_update
def iced_geom updated_airfoil
def dist 12-minute_shape.dst
gen dist
gen grid
def output case23_restart12.out
gen input
gen restart_update
#
# run wind for ice accretion step #12
#
gen solution

```

```

#
# Preprocessing for ice accretion step #13
#
def lewice_input case23_13.inp
gen lewice_input
#
# Ice accretion step #13
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #13
#
gen grid_update
def iced_geom updated_airfoil
def dist 13-minute_shape.dst
gen dist
gen grid
def output case23_restart13.out
gen input
gen restart_update
#
# run wind for ice accretion step #13
#
gen solution
#
# Preprocessing for ice accretion step #14
#
def lewice_input case23_14.inp
gen lewice_input
#
# Ice accretion step #14
#
gen lewice
#
# Preprocessing for flow solution using ice accretion #14
#
gen grid_update
def iced_geom updated_airfoil

```



```

def dist 14-minute_shape.dst
gen dist
gen grid
#
exit

```

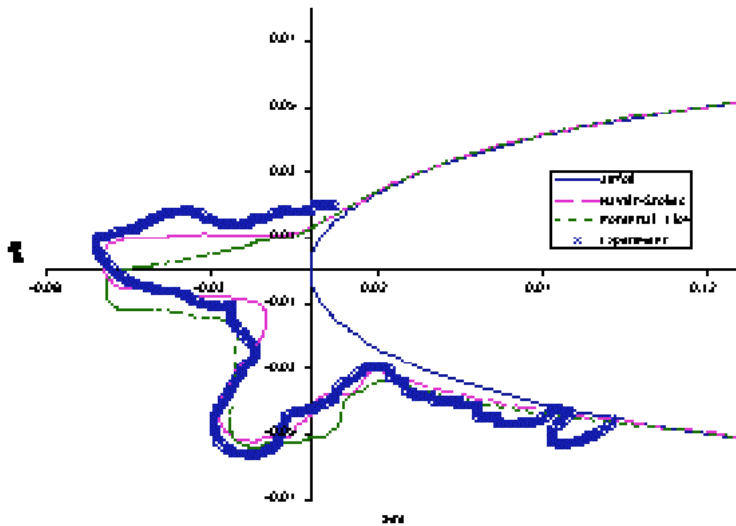


Figure 154: Ice Shape for Example Case 23

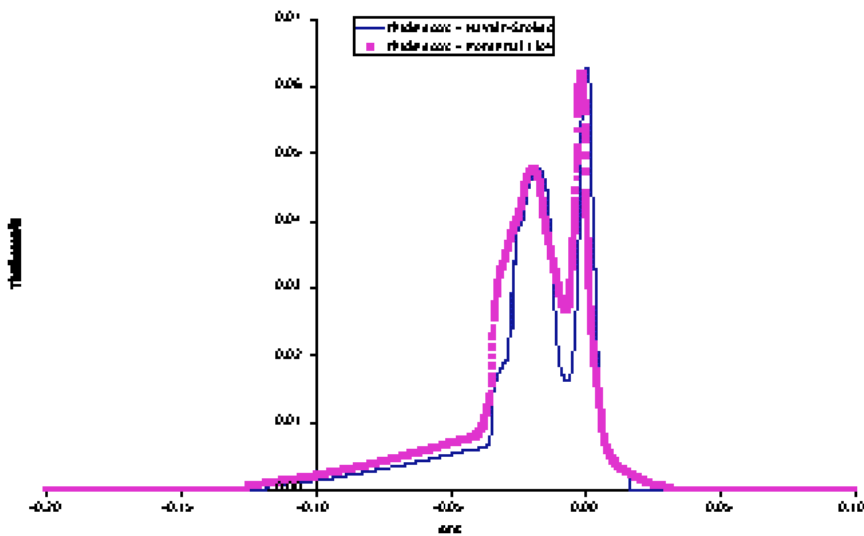


Figure 155: Ice Thickness for Example Case 23

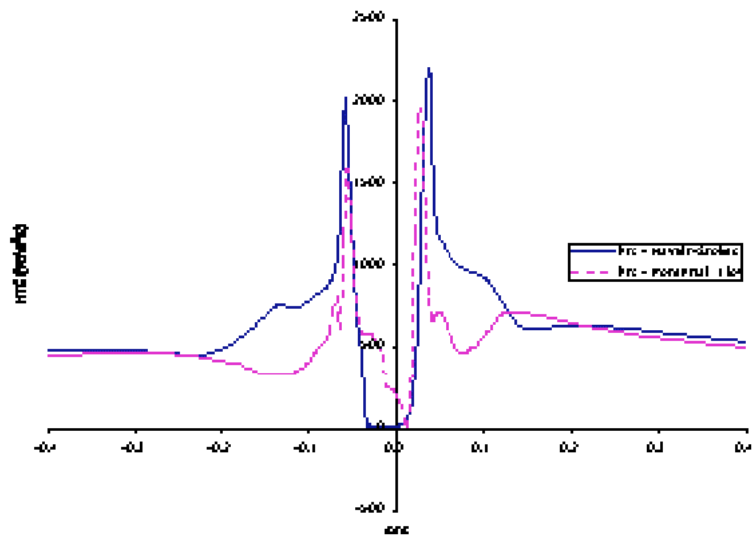


Figure 156: Heat Transfer Coefficient for Example Case 23

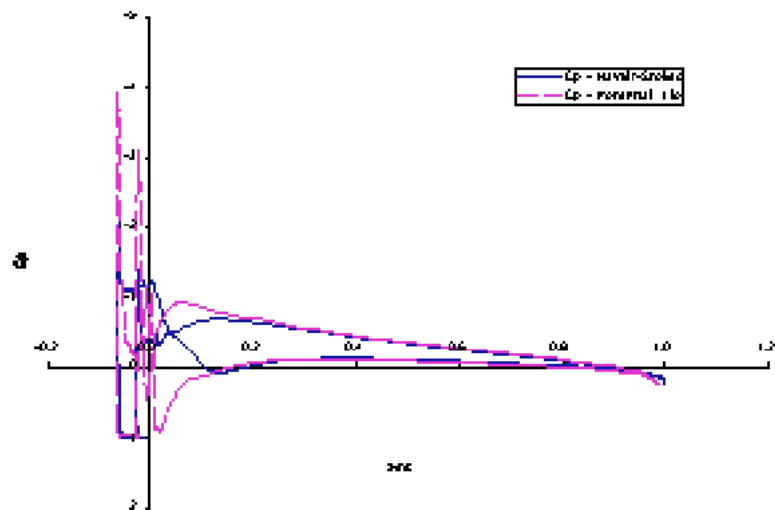


Figure 157: Pressure Coefficient for Example Case 23

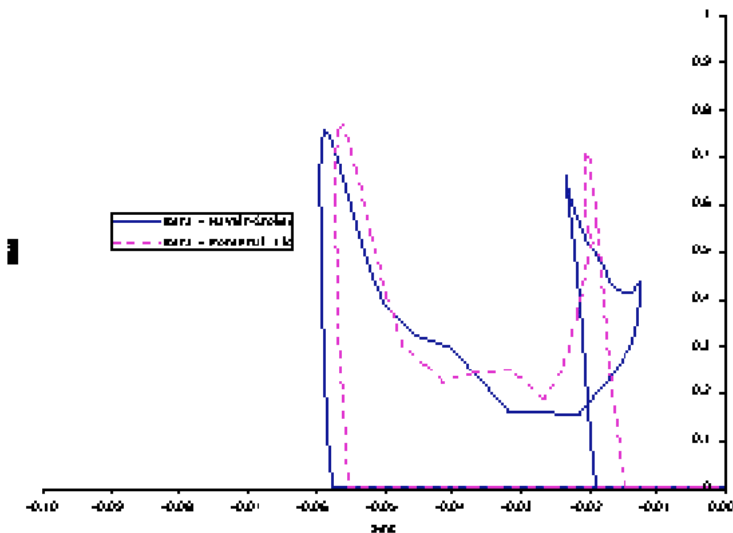


Figure 158: Collection Efficiency for Example Case 23

15.23.1. ICEG2D Process

This section provides a more detailed description of the ICEG2D process. The user is also strongly encouraged to read the ICEG2D documentation provided on the CD. ICEG2D has been modified since those documents were created however and discrepancies between the current process and the ICEG2D documentation will be noted.

```
def mode batch           Batch mode
def case case23         Defines case name
def surf smooth        Defines smooth surface
```

The user can choose “smooth” or “rough” as described in the ICEG2D manual.

```
def solver wind        Defines WIND as flow solver
#
# initial clean solution
#
def iced_geom case23.cln Defines initial geometry
gen dist              Generates point distribution
                    around airfoil

gen grid              Generates single block C-grid
def info case23_start.inf File name containing flow
                    input variables. (See note)
```

Note: The variable “VERSION” was added to the INPUTS Namelist of the ICEG2D input file. VERSION allows the user to provide the version of WIND being run. For example, an interim version of WIND (version 5.201) was used to generate the output for this example case. When WIND is run interactively, it prompts the user to select either option 0 (exit) or option 1 (WIND 5). Therefore, the user should set VERSION = 1 so that WIND is executed. Other WIND releases contain older versions of the software. Please make certain that you have selected variable VERSION so that WIND 5 is executed. Versions of WIND other than version 5 have not been tested with this procedure.

```
def output case23_start           Defines file name for output
gen input                         Generates WIND input files
gen solution                      Executes WIND
#
def info case23_restart.inf       Defines ICEG2D file name
                                   for next time step
def input_grid case23.grd         Defines grid file name
def input_q case23.q              Defines flow solution
                                   file name
#
# Preprocessing for ice accretion step #1
#
def lewice_input case23_1.inp     Defines LEWICE input file
name
```

Note: In addition, the user can define a de-icer input file name for ICEG2D. If the user does not specify a file name, the default de-icer file name is CaseName_dlew.inp. Since the user specified earlier that the case name is “case23” (by using “def case case23” earlier), the default de-icer file name is case23_dlew.inp. **A file with this name must exist or else ICEG2D will stop running** and produce the following error:

***** **ERROR** ***** (*filename*) is not a valid Lewice de-ice input file name.

```
gen lewice_input                 Generates LEWICE inputs
```

This command generates the files “xy.plt” and “q.plt” from the WIND output “case23.grd” and “case23.q”. It also creates a file containing responses to the interactive queries made by LEWICE.

```
#
# Ice accretion step #1
#
gen lewice                       Executes LEWICE
#
# Preprocessing for flow solution using ice accretion #1
#
gen grid_update                  Updates the current grid
```

def iced_geom updated_airfoil	Defines new (iced) geometry
def dist 1-minute_shape.dst	Defines file name for new point distribution
gen dist	Generates new point distribution
gen grid	Generates new grid
def output case23_restart1.out	Defines next output file name
gen input	Generates WIND inputs
gen restart_update	Generates information needed for WIND restart
#	
# run wind for ice accretion step #1	
#	
gen solution	Executes WIND for first ice shape
#	
# Preprocessing for ice accretion step #2	
#	
def lewice_input case23_2.inp	Defines LEWICE input file
gen lewice_input	Generates LEWICE inputs
#	
# Ice accretion step #2	
#	
gen lewice	Executes LEWICE using the first time step ice shape as the input geometry.
#	
# Preprocessing for flow solution using ice accretion #2	
#	
gen grid_update	Updates the old grid using the new ice shape.
def iced_geom updated_airfoil	Defines new iced geometry
def dist 2-minute_shape.dst	Defines file name for new distribution
gen dist	Generates new point distribution
gen grid	Generates grid for new iced geometry
iced geometry	

The process continues in this fashion for other time steps until the solution is completed. Note that this process requires multiple runs of both WIND and LEWICE. The output files which are created by these programs are saved to subdirectories within the directory structure where the user placed LEWICE. The ICEG2D process is quite complex and thus may not execute as smoothly as this example case. The following section will describe some of the pitfalls encountered in this process.

15.23.2. ICEG2D Input Problems

This section describes the problems encountered when trying to set up and run ICEG2D for Example Case 23 and other examples. 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 setting up the program.

Problem 1: Directory structure

The ICEG2D manual and README file instruct the user to modify their .cshrc file to specify the paths for ICEG2D, WIND and LEWICE. SGI systems in the icing branch had been set up to take this information from the .tcshrc file instead. You may need to contact your system administrator to help with the proper set up. Additionally, ICEG2D required that the user define a file structure for the flow solvers NPARC and HYBFL even though these flow solvers were not going to be used. The flow solvers do not have to be present in those directories, but the directory structure does need to be defined. Any valid directory path can be used if that flow solver is not going to be accessed. Finally, ICEG2D expects all of the WIND tools to be in the directory defined. Depending on the version of WIND you have, these tools may have been installed into other directories. For example, there exists WIND tools in directory *wind_tools/tools/bin* and *wind_tools/tools/SGI6.5/bin* as well as the directory *wind_tools/tools/SGI6.5/R12000/bin*. The WIND tools were all placed in *wind_tools/tools/SGI6.5/R12000/bin* as defined below. The user also need to make certain to define the appropriate executable for their system. For example, if you have an R10000 processor, then you need to specify the WIND executable in the R10000/bin path instead. The following table provides the path as they are set up in the user account "tobill".

Table 73: Path Definitions in .tcshrc File

```
setenv ICEG2D_DIR "/home/tobill/iceg2d"
set path = (. /home/tobill/iceg2d $path)
setenv LEWICE_DIR "/home/tobill/lewice"
setenv TRAP_FPE "ALL=COUNT; UNDERFL=ZERO; OVERFL=IEEE,TRACE(5),ABORT(100);
DIVZERO=ABORT; INVALID=TRACE(1),ABORT(1)"
setenv NPARC_DIR "/home/tobill/nparc"
setenv HYBFL_DIR "/home/tobill/nparc"
setenv WIND_DIR "/home/tobill/wind/wind/SGI6.5/R12000/bin"
setenv WIND_TOOLS_DIR "/home/tobill/wind_tools/tools/SGI6.5/R12000/bin"
```

Problem 2: Interactive responses

Both WIND and LEWICE are set up to take some inputs interactively. On UNIX systems, a batch process such as ICEG2D handles this by placing the interactive input in a file and then using this information by directing the contents into the executable as shown in the examples below.

Table 74: Example of batch structure

```
WIND < wind.inp
LEWICE < lewice.inp
```

In this example, the interactive information for WIND was placed in file “wind.inp” while the LEWICE interactive information was placed in the file “lewice.inp”. The following table shows the typical contents of the file “lewice.inp” while Table 77 shows an example of the “wind.inp” input file.

Table 75: lewice.inp Input File

```
case23_1.inp
case23.geom
Y
```

Table 76: wind.inp input File

```
1
case24
case24_restart6.out

1
n
n
```

ICEG2D will generate the files containing the interactive inputs for both WIND and LEWICE. A problem will occur when the interactive input required by WIND or LEWICE does not match the contents of this file. For example, if internal heat transfer coefficients for a piccolo tube system are read into LEWICE, LEWICE will generate an interactive warning about this use and will generate an additional warning message if the heat transfer coefficients are not aligned with the airfoil. ICEG2D will not provide those interactive inputs. When ICEG2D is ran in batch mode as in the example case, the process will crash because ICEG2D will invoke the next command “gen grid_update” whereas LEWICE is expecting a “Y” or “N” response to the interactive query. For LEWICE, this can be avoided by setting variable KWARN = 1 in the LPRNT namelist. LEWICE will still print the warning, but will automatically continue without requesting the interactive confirmation for reading external files. The user should use this function with caution however. The interactive warnings are for the user’s benefit and often highlight legitimate problems with the input.

The solution to this issue is more problematic with regards to WIND. If WIND requests additional interactive inputs other than those provided by ICEG2D, then changes must be made to the

ICEG2D source code to accommodate those inputs. Such was the case for the “VERSION” variable in the INPUTS namelist described earlier. Originally, ICEG2D set this value to “2” and it could not be changed by the user. When a version of WIND was used that accepted only values of “0” or “1” for this input, WIND would not run. Alternatively, the user could manually change the appropriate input file after invoking “gen input” and prior to invoking “gen solution” in the process. **It is highly recommended that the user run ICEG2D interactively until they are assured that the processes will work when invoked in batch mode.** By doing so, the user can manually check the input files generated by ICEG2D to ensure that they will work with the flow solver. Additionally, ICEG2D saves all user commands in a history file ICEG2D.HST. This history file can be used as a template for each cycle of a multiple ice accretion cycle run. The suggested procedure is to run ICEG2D interactively for one ice accretion cycle. Once it is verified that the process has worked correctly, the contents of the history file can be edited and a multiple-time step script created.

Problem 3: WIND (or other flow solver) and/or LEWICE does not run.

Many problems of this type can be alleviated or eliminated by running ICEG2D interactively. If WIND exits prematurely and does not produce the appropriate outputs, the user will recognize this and address the problem before continuing. If this occurs during a batch process, the script will simply move on to the next command. When this occurs, it is difficult if not impossible to determine when the error occurred. The same can be said if LEWICE exits prior to outputting the iced geometry. Usually these problems are simple to correct once they are spotted. For example, if the user sets up an example similar to case23 except using the 6” cylinder as the input geometry, WIND exits because the grid generated was too large. This was alleviated by increasing the values of **delta_s_min_nl**, **delta_s_max_nl**, and **delta_s_te_nl** in the file **dist_smooth.ini** (or **dist_rough.ini**). This file and other parameter files are given in directory **iceg2d/parameter_files**. Similarly, a case was attempted that used the case23 input except a de-icer was activated. During the third time step, LEWICE crashed while interpolating air velocities from the solution file. A review of the flow solution indicated that the grid resolution in the region with the residual ice shape might be causing a problem.

15.23.3. Using Other Flow Solvers in ICEG2D

It is feasible that the user could substitute other Navier-Stokes flow solvers and still use ICEG2D. This process would most likely require modification of the ICEG2D source code, however other options are available. The user could simply run ICEG2D to create the grid and then manually run their flow solver. Once the flow solution has been created, the user could invoke the “gen lewice_input” and “gen lewice” commands in ICEG2D or manually run LEWICE with the flow solver output. The user could attempt to use the “gen solution” command by renaming their flow solver to one that is supported. For this slight-of-hand to work, the flow solver being used must be set up to take the same inputs as the flow solver it is replacing. A batch process is considered unlikely to work with unsupported flow solvers though. NASA has plans to modify ICEG2D to more directly support other flow solvers.

Chapter 16: 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. This section differs from Chapter 6 which list user notes considered most important for accurate use of the software.

16.1. Old Input Files

Input files from previous versions of LEWICE will **not** work ‘as is’ with this version. Please follow the examples provided or see Chapter 5 for a summary of the differences.

16.2. 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.

16.3. 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.

16.4. 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 900 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.

All input and output file names have been presented in uppercase to make them stand out in the manual. The actual input and output files on the CD-ROM have lower case filenames and all output files from LEWICE 3.2 have lowercase filenames.

16.5. 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.

At a minimum, the user must add the following two lines to the end of the main input file for existing LEWICE 2.0 input files for them to work with LEWICE 3.2:

```
&RDATA  
&END
```

See Section 8.7 for details of the new namelist.

16.6. Multiple Bodies / Multi-Element Airfoils

LEWICE 3.2 can run multiple body simulations including multi-element airfoils. A report of its capabilities in this region shows very encouraging results¹⁵. However, much of the development effort for version 3.2 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 3.2 validated for multi-body flows.

16.7. 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 contain errors and is not recommended unless the user is willing to customize the software 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). Similarly, the variables in the user's PLOT3D file may not be the variables the program expects. Refer to Section 10.5 for the correct format.

16.8. 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 normalize the coordinates input. A typical value used for multi-element airfoils is the length of the airfoil in the stowed configuration.

16.9. 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.

16.10. Flow Module Limitation

For glaze ice shapes at high subsonic velocities, it is possible for the software 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/Navier-Stokes program 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 software 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 software 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 module and an estimate of the actual lift on the clean airfoil.

16.11. Ice Density

Ice density in LEWICE is fixed at a value of 917 kg/m^3 unless the $SLD = 1$ option is used. There exists two more correlations for ice density within the source code which are currently inactive due to undesirable numerical side effects. Developers who wish to activate these correlations can do so by changing an internal flag in the source code.

16.12. Icing Limit

The "**limit.dat**" output file was not generated in previous versions. The value for the final time step should agree with output from the utility program THICK although roundoff errors may cause the utility program to report an erroneous limit.

Chapter 17: Procedure for Acquiring LEWICE or other NASA Icing Software

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.

17.1. Current Address for Icing Software 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 18: Sample Software Request Letter

Dear Sir,

Our company/university would like to request the ice accretion software LEWICE 3.2 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 software will be run primarily on _____ (name the system) and it is preferred that the software be distributed on _____ (preferred media). Thank you.

Sincerely,

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

Chapter 19: Technical Support

Technical support is available from 9 AM to 5 PM Eastern time from the software developer(s). Questions on running the software 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) 977-7469

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. Software improvements including bug fixes which result from this correspondence will be incorporated into future versions which can be released by NASA.

Chapter 20: References

1. Wright, W. B., "User Manual for the NASA Glenn Ice Accretion Code LEWICE Version 2.0," NASA CR 209409, Sept. 1999.
2. Miller, D, et. al., "Validation of NASA Thermal Ice Protection Computer Codes Part 1 - Program Overview," NASA TM 107397, Jan. 1997.
3. Wright, W. B. and Rutkowski, A., "Validation Results for LEWICE 2.0," NASA CR 208690, Nov. 1998.
4. 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.
5. Frost, W., Chang, H., Shieh, C. and K. Kimble, "Two Dimensional Particle Trajectory Computer Program," Interim Report for Contract NAS3-224488, 1982.
6. 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.
7. Masiulaniec, K. C. and Wright, W. B., "User's Manual for the NASA Lewis Ice Accretion/ Heat Transfer Prediction Code with Electrothermal De-icer Input", NASA CR 4530, 1994.
8. Al-Khalil, K. M., "Numerical Simulation of an Aircraft Anti-Icing System Incorporating a Rivulet Model for the Runback Water," PhD Dissertation, University of Toledo, Toledo, Ohio, Jan. 1991.
9. Bush, R. H. , Power, G. D. and Towne, C. E., "WIND: The Production Flow Solver of the NPARC Alliance," AIAA 98-0935, Jan. 1998.
10. Thompson, D. S. and Soni, B. K., "ICEG2D: A Software Package for Ice Accretion Prediction," AIAA 2003-1070, Jan. 2003.
11. Wright, W.B., "Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 2.2.2," NASA CR 2002-211793, Aug. 2002.
12. Wright, W. B., "An Evaluation of Jet Impingement Heat Transfer Correlations for Piccolo Tube Applications," AIAA 2004-0062, Jan. 2004.
13. Wright, W. B., and Potapczuk, M. G., "Semi-Empirical Modeling of SLD Physics," AIAA 2004-0412, Jan. 2004.
14. Wright, W.B., "Users Manual for the Improved NASA Lewis Ice Accretion Code LEWICE 1.6," NASA CR 198355, June 1995.
15. Wright, W.B., "Capabilities of LEWICE 1.6 and Comparison with Experimental Data," AHS International Icing Symposium, Montreal, Canada, pp. 347-61, Sept. 1995.
16. 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.

17. Goldstein, R. J., Behbahni, A. I., and Heppelman, K. K., "Streamwise Distribution of the Recovery Factor and the Local Heat Transfer Coefficient to an Impinging Circular Air Jet," *Intl. J. of Heat and Mass Trans.*, Vol. 29, #8, pp. 1227-1235, 1986.
18. Brown, J., Raghunathan, S., Watterson, J. K., Linton, A. J., and Riordon, D., "Heat Transfer Correlation for De-Icing Systems," AIAA 2001-0837, Jan. 2001.
19. Tawfek, A. A., "Heat Transfer Studies of the Oblique Impingement of Round Jets upon a Curved Surface," *Heat and Mass Trans.*, Vol. 38, PP. 467-75, 2002.
20. Gau, C. and Chung, C. M., "Surface Curvature Effect on Slot-Air-Jet Impingement Cooling Flow and Heat Transfer Process," *Trans. of the ASME*, Vol. 113, pp. 858-864, Nov. 1991.
21. Huber, A. M. and Viskanta, R., "Effect of Jet-Jet Spacing on Convective Heat Transfer to Confined, Impinging Arrays of Axisymmetric Air Jets," *Int. J. Heat Mass Trans.*, vol. 37, #18, pp. 2859-2869, 1994.
22. Goldstein, R. J. and Sol, W. S., "Heat Transfer to a Row of Impinging Circular Air Jets Including the Effect of Entrainment," *Intl. J. of Heat and Mass Trans.*, Vol. 34, #8, pp. 2133-2147, 1991.
23. Hrycak, P., "Heat Transfer From a Row of Impinging Jets to Concave Cylindrical Surfaces," *Intl. J. Heat and Mass Trans.*, Vol. 24, pp. 407-419, 1981.
24. Papadakis, M., et. al., "Progress in Measuring Water Impingement Characteristics on Aircraft Surfaces," AIAA98-0488, Jan. 1998.
25. 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.
26. NPARC Version 3.0 User's Manual, NPARC Alliance, Sept. 1996
27. Walatka, P. P., et. al., "PLOT3D User Manual," NASA TM 101067, Mar. 1990.

Chapter 21: Appendix A

21.1. Functional Requirements for LEWICE version 3.2

This section has been included by the request of several users. It will define the equations and other functional requirements of LEWICE 3.2 so that users can increase their understanding of the underlying physics being solved. Note that the equations are simply defined, not derived. Users who wish to understand how these equations are arrived at are directed to the references to this report. The sections are organized by module, but the modules themselves are not presented in any particular order.

21.2. Geometry Smoothing

21.2.1. Description/Purpose

- 21.2.1.1. The purpose of geometry smoothing is to smooth the body geometry(s) supplied according to the requirements listed below.

21.2.2. Smoothing Requirements

- 21.2.2.1. The program shall modify the body geometry(s) such that the angle between any two adjoining segments is no greater than 0.001 degrees.
- 21.2.2.2. The program shall modify the body geometry(s) such that the size of any segment is no greater than or less than 5% of the size of an adjoining segment.
- 21.2.2.3. The program shall modify the body geometry(s) such that the minimum size of any segment is not less than the size criteria input by the user.
- 21.2.2.4. The maximum value which the user may input for the minimum segment size is 0.0008 times the chord length.
- 21.2.2.5. The program shall modify the body geometry(s) such that the maximum size of any segment is not greater than two times the minimum segment size input by the user.
- 21.2.2.6. Each 5% section (measured by wrap distance) of the body geometry(s) shall not have an overall curvature which is greater than the curvature of a circle whose radius is 0.2 times the geometry chord length.
- 21.2.2.7. The 5% sections of the body geometry(s) given in 1.2.6 shall overlap each other by 1% of the total surface wrap distance.
- 21.2.2.8. The modified body geometry(s) shall not differ from the original body geometry(s) by more than 5% based on the quantified ice measurement scale defined in NASA TM-208690.
- 21.2.2.9. The program shall transfer the variables for interim ice thickness and wrap distance to the modified body geometry(s).
- 21.2.2.10. The program shall store the modified values for body geometry(s), wrap distance, interim ice thickness and number of points into new variables and shall not overwrite the original values.

21.3. Point Generation

21.3.1. Description/Purpose

- 21.3.1.1. The purpose of the point generation function is to calculate surface points and the surface wrap distance of those points for a given body geometry(s) according to the requirements listed below.

21.3.2. Point Generation Requirements

- 21.3.2.1. The program shall modify the body geometry(s) such that the angle between any two adjoining segments is no greater than 0.001 degrees.
- 21.3.2.2. The program shall modify the body geometry(s) such that the size of any segment is no greater than 5% greater or less than a neighboring segment.
- 21.3.2.3. The program shall modify the body geometry(s) such that the minimum size of any segment is not less than the criteria input by the user.
- 21.3.2.4. The maximum value which the user may input for the minimum segment size is 0.0008 times the geometry chord length.
- 21.3.2.5. The program shall modify the body geometry(s) such that the maximum size of any segment is not greater than two times the minimum segment size input by the user.
- 21.3.2.6. The modified body geometry(s) for this step shall not differ from the original body geometry(s) by more than 1% based on the quantified ice measurement scale defined in NASA TM 208690.
- 21.3.2.7. The program shall transfer the variables for interim ice thickness and surface wrap distance to the modified body geometry(s).

21.4. Ice Addition

21.4.1. Description/Purpose

- 21.4.1.1. The purpose of ice addition is to add a previously calculated ice thickness distribution to the existing body geometry(s) according to the requirements listed below.

21.4.2. Ice Addition Requirements

- 21.4.2.1. The program shall add the ice thickness such that the area of ice added to each segment is within 0.1% of the calculated amount.
- 21.4.2.2. The program shall add the ice thickness such that the total area of ice added to the surface is within 0.1% of the calculated amount.
- 21.4.2.3. The program shall add the ice thickness in a direction methodology which is chosen by the user.*
- 21.4.2.4. The program shall allow the user to choose from the following three direction methodologies:*)
 - 21.4.2.4.1.
Normal to each segment
 - 21.4.2.4.2.
Normal to the flow direction
 - 21.4.2.4.3.
Normal to the trajectory impact direction
- 21.4.2.5. During the ice addition, the program shall not use those surface points for which no ice is to be added.
- 21.4.2.6. During the ice addition, the program shall add and remove surface points as necessary in accordance with the point generation requirements listed above.
- 21.4.2.7. The program shall output to file(s) the body geometry(s) after the new body geometry(s) have been calculated.

*Not active in LEWICE 3.2

21.5. Trajectory Integration

21.5.1. Description/Purpose

21.5.1.1. The purpose of trajectory integration is to numerically integrate the equations of motion of a water particle through a surrounding airstream.

21.5.2. Trajectory Integration Requirements

21.5.2.1. The program shall integrate the following equations:

21.5.2.1.1.

$$mx_p = -\vec{D} \cos \gamma - \vec{L} \sin \gamma + mg \sin \alpha$$

21.5.2.1.2.

$$my_p = -\vec{D} \sin \gamma + \vec{L} \cos \gamma - mg \cos \alpha$$

21.5.2.2. The terms in these equations have the following definitions:

21.5.2.2.1.

$$\gamma = \text{atan} \frac{y_p - V_y}{x_p - V_x}$$

21.5.2.2.2.

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

21.5.2.2.3.

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

21.5.2.2.4.

$$V = \sqrt{(x_p - V_x)^2 + (y_p - V_y)^2}$$

21.5.2.2.5.

$$c_d = \frac{24}{Re} + 0.4 + \frac{6}{1 + \sqrt{Re}} \text{ for } c_d \leq 100$$

21.5.2.2.6.

$$c_d = \frac{24}{Re} + 0.3 + \frac{6}{1 + \sqrt{Re}} \text{ for } c_d \geq 100$$

21.5.2.2.7.

$$Re = \frac{Vd_p}{\nu_a}$$

21.5.2.2.8.

$$A_p = \frac{\pi d_p^2}{4}$$

21.5.2.2.9.

m = particle mass (kg)

g = gravitational constant = 9.8 m/s²

x_p = x-location of water particle (m)

y_p = y-location of water particle (m)

\dot{x}_p = first derivative of x-location of water particle with respect to time (x-component of particle velocity) (m/s)

\dot{y}_p = first derivative of y-location of water particle with respect to time (y-component of particle velocity) (m/s)

\ddot{x}_p = second derivative of x-location of water particle with respect to time (x-component of particle acceleration) (m/s²)

\ddot{y}_p = second derivative of y-location of water particle with respect to time (y-component of particle acceleration) (m/s²)

V_x = x-component of local air velocity at particle location (m/s)

V_y = y-component of local air velocity at particle location (m/s)

ρ_a = local air density at particle location (kg/m³)

A_p = cross-sectional area of particle (m²)

d_p = particle diameter (m)

c_l = coefficient of lift = 0

c_d = coefficient of drag (dimensionless)

Re = Reynolds number (dimensionless)

α = angle of attack (degrees)

\bar{D} = drag force (kg*m/s²)

\vec{L} = lifting force (kg*m/s²)

γ = angle difference between particle velocity vector and air flow velocity vector (degrees)

ν = kinematic viscosity of air m²/s

π = 3.1415926536...

The integration shall start at a location specified in Section 8 and with an initial velocity as specified in Section 8.

21.5.2.3. The integration shall end when one of the following two conditions have been met:

21.5.2.4.1.

The particle path intersects one of the body geometry(s).

21.5.2.4.2.

The particle has traveled past the end of all of the body geometry(s) as measured in the x-direction.

21.5.2.4. The program shall model the breakup of water drops as defined in Section 5.*

21.5.2.5. The program shall determine if droplets whose path intersects the body geometry(s) will fully impinge, partially impinge or bounce in accordance with Section 6.*

21.5.2.6. The program shall output to file(s) the starting and ending location of the trajectory and the end condition met by Section 4.2.4.

21.5.2.7. The program shall output to file(s) the entire path of each trajectory or each trajectory whose path intersects the body geometry(s) if desired by the user.

*Not implemented in LEWICE 2.2 Beta Release

21.6. Droplet Breakup*

21.6.1. Description/Purpose

21.6.1.1. Droplet breakup occurs when fluid particles are moving at high velocities. The droplet will first deform from a spherical shape to an ellipsoid. This is modeled in the coefficient of drag in Sections 4.2.2.5 and 4.2.2.6. After this deformation, the droplet continues to deform until it breaks apart. The purpose of this section is to describe the requirements for modeling the physical processes of droplet breakup.

21.6.2. Droplet Breakup Requirements

21.6.2.1. The program shall determine if droplet breakup occurs after each step of the droplet trajectory integration as described in Section 4.

21.6.2.2. The program shall calculate the Weber number as defined by

$$We = \frac{\rho_a V^2 d_p}{\sigma}$$

where σ = surface tension between air and water (kg/s²).

21.6.2.3. Where the Weber number of Section 5.2.2 exceeds a value of 13 and the model has been invoked with the user flag SLD, the droplet shall be broken into equally sized smaller particles as described in the following equation:

$$d_a = \left(\frac{\rho_w}{\rho_a}\right)^{\frac{1}{4}} Re_e^{-\frac{1}{2}} d_o$$

21.6.2.4. The terms in this equation have the following definitions:

21.6.2.4.1.

d_o = original drop size (microns)

21.6.2.4.2.

d_a = drop size after breakup (microns)

21.6.2.4.3.

ρ_w = water density (kg/m³)

21.6.2.4.4.

ρ_a = air density (kg/m³)

21.6.2.4.5.

$$Re_w = \frac{\rho_w V d_o}{\mu_w}$$

where

21.6.2.4.6.

μ_w = water viscosity (kg/ms)

V is defined in Section 4.2.2.4

21.6.2.5. If droplet breakup has occurred, the program shall continue to integrate the trajectories of the smaller particles until one of the conditions in Section 4.2.4 has been met.

21.6.2.6. The droplet breakup routines shall not cause the execution time of the trajectory integration to increase by more than 25% over the execution time of LEWICE 3.2 without the SLD flag when the benchmark test cases are ran.

*Not implemented in LEWICE 2.2 Beta Release

21.7. Droplet Splashing

21.7.1. Description/Purpose

21.7.1.1. Droplets whose path intersects the body geometry(s) may fully impinge, partially impinge and partially bounce or may fully bounce off the body geometry(s). The purpose of this section is to describe requirements for modeling the physical phenomena involved.

21.7.2. Droplet Splashing Requirements

21.7.2.1. When the path of a droplet intersects a body geometry, the program shall determine the fraction of water which bounces or splashes from the following equations:

21.7.2.1.1.

$N_b = 0$ for $K_t \leq 200$ (no splashing or bouncing)

21.7.2.1.2.

$$N_b = 0.9(1 - \sin \alpha_i) \left[1 - \exp\left(\frac{-9.2(K_t - 200)}{1000}\right) \right]$$
(splashing and bouncing)
for $K_t > 200$ and $0 \leq \alpha_i \leq 90^\circ$

21.7.2.1.3.

The terms in these equations have the following definitions:

21.24.2.2.13.1.

N_b = mass fraction of water lost due to bouncing or splashing (dimensionless)

21.24.2.2.13.2.

$$K_{tn} = 0.859 \left(\frac{\rho_l}{LWC} \right)^{0.125} \sqrt{K}$$
(dimensionless)

21.24.2.2.13.3.

$$K_t = \frac{K_{tn}}{(\sin \alpha_i)^{1.25}}$$

21.24.2.2.13.4.

$$K = OhRe_p^{1.25}$$
(dimensionless)

21.24.2.2.13.5.

$$Oh = \frac{\mu_p}{\sqrt{\rho_p \sigma d_p}}$$

= Ohnesorge number (dimensionless)

21.24.2.2.13.6.

$$Re_p = \frac{\rho_p V d_p}{\mu_p}$$

= Droplet Reynolds number (dimensionless)

21.24.2.2.13.7.

μ_p = particle viscosity (kg/m/s)

ρ_p = particle density (kg/m³)

σ = surface tension between air and water (kg/s²)

d_p = particle diameter (m)

V = particle velocity (m/s) as given in 4.2.2.4

α_i = droplet impact angle (degrees)

21.7.2.2. The secondary droplet size is given by the following equations:

21.7.2.2.1.

For splashing (Section 6.2.1.2 criteria), the splashed droplet size is given by the following equations:

21.7.2.2.2.

$$d_s = 8.72 d_p \exp(-0.0281K)$$

for $77 < K < 183.68$

21.7.2.2.3.

$$d_s = 0.05 d_p \text{ for } K \geq 183.68$$

21.7.2.2.4.

$$d_s = d_p \text{ for } K \leq 77$$

21.7.2.2.5.

For incomplete bouncing (Section 6.2.1.3), the bounced droplet size is given by the following equation:

$$d_s = d_p N_b$$

21.7.2.2.6.

For complete bouncing (Section 6.2.1.4), the bounced droplet size is given by the following equation:

21.7.2.2.7.

$$d_s = d_p$$

21.7.2.3. The secondary droplet velocities are given by the following equations:

21.7.2.3.1.

$$V_{s,x} = V_{i,x}(1.075 - 0.0025\alpha_i)$$

21.7.2.3.2.

$$V_{s,y} = V_{i,y}(0.3 - 0.002\alpha_i)$$

21.7.2.3.3.

$$\alpha_s = \text{atan} \frac{V_{s,y}}{V_{s,x}}$$

21.7.2.3.4.

The terms in these equations have the following definitions:

$V_{s,x}$ = x-component of the splashed droplet (m/s)

$V_{s,y}$ = y-component of the splashed droplet (m/s)

$V_{i,x}$ = x-component of the incoming droplet (m/s)

$V_{i,y}$ = y-component of the incoming droplet (m/s)

α_s = splash angle (degrees)

For this function, the program shall subtract the local mass fraction of water lost due to splashing from the local collection efficiency defined in Section 8.

21.7.2.4. The program shall track the splashed droplets using the procedure in Section 4. If a particle is reimpinged, the fractional mass calculated in Section 6.2.1 from the previous splash event shall be added to the local collection efficiency defined in Section 8.

21.8. Impingement Limits

21.8.1. Description/Purpose

- 21.8.1.1. Impingement limits are the uppermost and lowermost locations on the body geometry(s) where water droplets impinge upon the surface. The purpose of this section is to provide the requirements for determining upper and lower impingement limits for the body geometry(s).

21.8.2. Impingement Limit Requirements

- 21.8.1.2. The program shall integrate trajectories as listed in Section 4 starting from the same x-location as determined in Section 9 but different y-locations until two trajectories meet the following criteria:

- 21.8.1.2.1.

- Their starting locations are within $5 \cdot 10^{-5}$ (dimensionless) distance.

- 21.8.1.2.2.

- One of the trajectories passes above the target body geometry while the other trajectory intersects the target body geometry.

- 21.8.1.2.3.

- The same criteria as listed in 7.2.1.1 and 7.2.1.2 are met except that the missed trajectory must pass below the target body geometry.

- 21.8.1.2.4.

- The program shall output to file(s) the starting and ending locations of the trajectories which intersect the target body geometry in the requirement above.

21.9. Collection Efficiency

21.9.1. Description/Purpose

- 21.9.1.1. The collection efficiency is the efficiency with which any location on the body geometry(s) will collect water droplets. The purpose of this section is to provide the requirements for calculating the collection efficiency on the body geometry(s).

21.9.2. Collection Efficiency Requirements

- 21.9.2.1. The program shall take as input from the user a number of trajectories for the collection efficiency calculation. This number shall be greater than or equal to 12 less than 50.
- 21.9.2.2. The program shall integrate the number of trajectories listed in 8.2.1 by the requirements listed in Section 4 starting from the same x-location as determined in Section 9 but different y-locations which are evenly spaced between the starting locations determined by the impingement limit requirements as given in Section 7.
- 21.9.2.3. The program shall determine the local collection efficiency at each point on the body geometry(s) by the following equation:

21.9.2.4.
$$\beta = \frac{dy_o}{ds}$$

- 21.9.2.5. The terms in the previous equation have the following definition:
 β = local collection efficiency (dimensionless)

y_0 = y-value of the starting locations of the collection efficiency trajectories (m)

s = surface wrap distance on target body geometry where the droplet intersects (m)

- 21.9.2.6. The local collection efficiency from the previous requirement shall be reduced due to splashing as described in Section 6.
- 21.9.2.7. The program shall output to file(s) the starting and ending locations of the trajectories which intersect the target body geometry in the requirement above.

21.10. Other Droplet Trajectory Requirements

21.10.1. Description/Purpose

21.10.1.1. This section shall describe requirements for droplet trajectory and collection efficiency calculation not covered by the previous sections.

21.10.2. Other Droplet Trajectory Requirements

21.10.2.1. The droplets shall start at an initial x-location and y-location where the local velocity is within 10^{-3} of the flight velocity input by the user.

21.10.2.2. The initial velocity of the droplet shall be the flight velocity input by the user plus the terminal velocity of the droplet as given by the equation:

$$\frac{4gd_p(\rho_w - \rho_a)}{3v_a^2\rho_a} = c_d Re_t^2$$

21.10.2.3.

21.10.2.4. The terms in the above equation have the following definitions:

ρ_w = density of water droplet (kg/m³)

$$Re_t = \frac{V_t d_p}{v_a}$$

= terminal velocity Reynolds number (dimensionless)

V_t = terminal droplet velocity (m/s)

other variables were defined in Section 4.2.2

21.10.2.5. The program shall calculate droplet trajectories, impingement limits and collection efficiencies for up to 5 body geometries and up to 10 drop sizes.

21.10.2.6. The program shall calculate an overall local collection efficiency from the individual collection efficiencies in Section 8 by the following equation:

$$\beta_o = \sum \beta_i N_i$$

21.10.2.8. The terms in the above equation have the following definitions:

β_o = overall local collection efficiency (dimensionless)

β_i = local collection efficiency for droplet size i (dimensionless)

N_i = mass fraction contained in droplet size i (dimensionless)

(N_i is input by the user)

21.10.2.9. The program shall output to file(s) the overall local collection efficiency.

21.11. Potential Flow

21.11.1. Description/Purpose

Potential flow describes a specific equation which provides flow variables such as velocity, pressure, and pressure coefficient on the surface of the body geometry(s) assuming that the flow field is inviscid, incompressible and irrotational. The purpose of this section is to provide the requirements for a potential flow solution.

21.11.2. Potential Flow Requirements

21.11.2.1. The program shall use potential flow as the flow solution when this input option is selected by the user.

21.11.2.2. The program shall solve the following equation:

21.11.2.3.
$$\Phi = \int \left[\frac{\xi}{r} + \frac{\partial}{\partial n} \left(\frac{\zeta}{r} \right) \right] dS$$

The terms in this equation have the following definitions:

21.11.2.3.1.

Φ = velocity potential (m²/s)

r = distance from an arbitrary point (m)

S = surface area (m²)

n = unit normal direction (dimensionless)

ξ = source strength per unit area (m/s)

ζ = dipole strength per unit area (m/s)

21.11.2.3.2.

The velocity potential is related to the velocity by the following equations:

$$V_x = \frac{\partial \Phi}{\partial x}$$

$$V_y = \frac{\partial \Phi}{\partial y}$$

21.11.2.4. The velocity at an arbitrary point in space shall be determined from the following equation:

21.11.2.4.1.

$$V = \xi_p \int_p \text{grad} \left(\frac{1}{r} \right) dS$$

21.11.2.4.2.

The subscript p in the previous equation refers to the panel (body segment)

21.11.2.5. The equation listed above shall be used to calculate air velocities during the droplet trajectory integration as listed in Section 4 when potential flow has been selected by the user.

21.11.2.6. The source strengths shall be calculated from the following equation:

21.11.2.4.3.

$$\frac{\xi_i}{2\pi} + \sum_{i \neq j}^n \frac{\xi_j}{2\pi} \int \frac{\partial}{\partial n_i} \left(\frac{1}{r} \right) dS = -V_\infty \cdot n_i$$

V_x = flight velocity (m/s)

ξ_i

= source strength (m/s)

21.12. Grid-based Flow

21.12.1. Description/Purpose

Grid-based flow describes flow equations which provides flow variables such as velocity, pressure, and pressure coefficient on the surface of the body geometry(s) and on predetermined locations on a structured grid. The purpose of this section is to provide the requirements for a grid-based flow solution.

21.12.2. Grid-based Flow Requirements

- 21.12.2.1. The program shall use grid-based flow as the flow solution when this input option is selected by the user.
- 21.12.2.2. The program shall read the grid-based flow solution from file(s) when this input option is selected by the user.
- 21.12.2.3. The two input options given above shall be considered separate inputs.
- 21.12.2.4. The format of the file(s) used shall be specified by the format used by the grid-based flow module which generated the flow solution.
- 21.12.2.5. When the user has selected a grid-based flow solution but did not select the read option, the program shall run a user-supplied script file which runs the grid-based flow solver.
- 21.12.2.6. The requirements for the grid-based flow solver shall be specified in the requirements document for the program being used and shall not be specified herein.
- 21.12.2.7. For either grid-based option, the program shall accept structured grids and flow solutions of up to 10 grid blocks.
- 21.12.2.8. When requested by the droplet trajectory requirements of Section 4, the program shall calculate velocities at locations which are not at grid points.

21.13. Compressibility Correction

21.13.1. Description/Purpose

21.13.1.1. A compressibility correction will adjust an incompressible flow solution for compressibility effects. The purpose of this section is to describe the requirements for this function.

21.13.2. Compressibility Requirements

21.13.2.1. The program shall apply standard corrections to the potential flow solution described in Section 10 and the grid-based flow solution of Section 11 to account for compressibility effects.

21.13.2.2. The following equation shall be used as the correction for the pressure coefficient:

21.13.2.2.1.

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

The terms in this equation have the following definition:

21.13.2.2.2.

$$c_{p_{inc}}$$

= incompressible pressure coefficient at a given location (dimensionless)

$$c_{p_{comp}}$$

= compressible pressure coefficient at the same location (dimensionless)

M_x = flight Mach number

21.13.2.3. The static pressure at a given location shall be calculated from the pressure coefficient using the equation:

$$c_p = \frac{p - p_{\infty}}{\frac{1}{2} \rho_{\infty} V_{\infty}^2}$$

21.13.2.3.1.

The terms in the above equation are defined as follows:

21.13.2.3.2.

p = static pressure at a given location (N/m²)

p_x = ambient static pressure (N/m²)

ρ_x = ambient air density (kg/m³)

V_x = flight velocity (m/s)

21.13.2.4. The Mach number at a given location shall be calculated using the equation:

21.13.2.4.1.

$$\frac{p_o}{p} = \left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}}$$

The terms in this equation have the following definitions:

21.13.2.4.2.

γ = specific heat ratio = 1.4 for air (dimensionless)

M = Mach number at a given location (dimensionless)

p = static pressure at a given location (N/m²)

p_o = ambient total pressure (N/m²)

21.13.2.5. The ambient total pressure in Section 12.2.4 is defined by the equation:

21.13.2.5.1.

$$p_o = p_\infty + \frac{1}{2} \rho_\infty V_\infty^2$$

The variables in this equation have been previously defined in this section.

21.13.2.6. The static air temperature at a given location shall be defined by the equation:

21.13.2.6.1.

$$\frac{T_o}{T} = \left(\frac{p_o}{p}\right)^{\frac{\gamma-1}{\gamma}}$$

The variables in this equation have the following definitions:

21.13.2.6.2.

T = static air temperature at a given location (°K)

T_o = ambient total temperature (°K) which is defined as

$$\frac{T_o}{T_\infty} = \left(\frac{p_o}{p_\infty}\right)^{\frac{\gamma-1}{\gamma}}$$

21.13.2.7. The corrected velocity at a given point shall be calculated from the following equation:

21.13.2.7.1.

$$V = M\sqrt{\gamma RT}$$

The terms in the above equation have the following definitions:

21.13.2.7.2.

V = corrected velocity at a given location

R = ideal gas constant = 287.5 m²/s²/°K

The implementation in the LEWICE 3.2 Beta Release defines the corrected velocity from the equation:

$$V = \frac{V_{inc}(1-\gamma)}{1-\gamma V_{inc}^2}$$

$$M = \frac{V}{\sqrt{RT - \frac{\gamma-1}{2} V^2}}$$

where V_{inc} = incompressible velocity (m/s). It then calculates the local Mach number from the equation:

The maximum local Mach number allowed is $M = 0.8$. The velocity is also adjusted downward if the equation above gives a Mach number greater than 0.8. Other quantities are then calculated from the local Mach number as given earlier. The discrepancy between this implementation and the isentropic equations given earlier is due to numerical error in the calculated incompressible pressure coefficient. For large negative pressure coefficients, the equation in 12.2.3.1 will give a negative static pressure, an impossible result. For smaller negative pressure coefficients or for positive pressure coefficients, the two methodologies give approximately equal answers.

(For the following inputs: $P_x = 100,000$ N/m²; $\rho_x = 1.3$ kg/m³; $V_x = 150$ m/s, corrected pressure coefficients less than [more negative than] $c_p = -6.84$ will give negative static pressures. This result is a consequence of the physical limitations of potential flow when applying the equation in 12.2.2.1)

21.14. Stagnation Point

21.14.1. Description/Purpose

- 21.14.1.1. The flow solution specified in Sections 11 and 12 will have one or more locations where the surface velocity changes sign (direction). The purpose of this section is to describe the requirements for this function.

21.14.2. Stagnation Point Requirements

- 21.14.2.1. The program shall use the minimum x-location on the body geometry as the initial stagnation point estimate.
- 21.14.2.2. The program shall determine the location(s) on the body geometry(s) where the surface velocity changes sign or is equal to zero.
- 21.14.2.3. If only one stagnation point occurs on a body geometry, it shall be designated as the main stagnation point.
- 21.14.2.4. If more than one stagnation point occurs on a body geometry, the program shall select the main stagnation point as the location nearest to the previous value for the main stagnation point.

21.15. Roughness Height

21.15.1. Description/Purpose

21.15.1.1. Ice growth will have a characteristic surface roughness which is based on the specific conditions input by the user. The purpose of this section is to describe the requirements for determining this height.

21.15.2. Roughness Height Requirements

21.15.2.1. The surface roughness height shall be determined from the following empirical correlation:

21.15.2.1.1.

*

$$x_k = \frac{1}{2} \sqrt{0.15 + \frac{0.3}{N_{f_{stag}}}}$$

The terms in this equation have the following definitions:

21.15.2.1.2.

x_k = surface roughness height (dimensionless); also known as equivalent sand-grain roughness height

$N_{f_{stag}}$

= freezing fraction at stagnation point (dimensionless)

21.15.2.1.3.

The freezing fraction at stagnation point shall be calculated in accordance with the requirements given in Section 17.

This empirical equation was determined from experimental measurements of roughness heights as a function of the calculated freezing fraction at the stagnation point. It was not reverse-engineered in order to match ice shape predictions.

21.16. Heat Transfer Coefficient

21.16.1. Description/Purpose

21.16.1.1. The convective heat loss from the surface is characterized by a heat transfer coefficient which is calculated from a flow solution. The purpose of this section is to provide requirements for the calculation of this parameter.

21.16.2. Heat Transfer Coefficient Requirements

21.16.2.1. The program shall calculate the heat transfer coefficient on the surface of the body geometry(s) using an integral boundary layer approach when it is not supplied from a grid-based flow solution.

The integral boundary layer approach involves the following steps:

21.16.2.1.1.

Calculation of critical Reynolds number for transition from the following empirical equation:

$Re_c = 10^{22}$ when at stagnation point

$$Re_c = 3834.2 - 1.9846 \times 10^5 |s| + 3.2812 \times 10^6 |s|^2 - 6.9994 \times 10^6 |s|^3$$

when

s

is 3.5% chord and $Re_c \leq 600$

$Re_c = 600$ when

s

is 3.5% chord or $Re_c < 600$ in the equation above.

21.16.2.1.2.

Calculation of shape factors Z , Λ and K from the equations:

21.24.2.2.13.8.

$$\frac{dZ}{ds} = \frac{F(K)}{V}$$

21.24.2.2.13.9.

$$K = Z \frac{dV}{ds}$$

The terms in these equations have the following definitions:

21.24.2.2.13.10.

$$Z = \frac{\delta_2^2}{\nu}$$

21.24.2.2.13.11.

$$F(K) = 2 \left(\frac{37}{315} - \frac{\Lambda}{945} - \frac{\Lambda^2}{9072} \right) \left[2 - \frac{116}{315} \Lambda + \left(\frac{2}{945} + \frac{1}{120} \right) \Lambda^2 + \frac{2}{9072} \Lambda^3 \right]$$

21.24.2.2.13.12.

$$\Lambda = \frac{\delta^2 dV}{\nu ds}$$

21.24.2.2.13.13.

δ = boundary layer thickness (m)

δ_2 = momentum thickness (m)

V = surface velocity at a given location (m/s)

s = surface distance from stagnation point (m)

ν = kinematic viscosity of air (m²/s)

21.24.2.2.13.14.

Calculation of the roughness Reynolds number from the equation:

21.24.2.2.13.15.

$$Re_k = \frac{V_k x_k}{\nu}$$

The terms in this equation have the following definition:

21.24.2.2.13.16.

x_k = roughness height (m) from Section 14

ν = kinematic viscosity of air (m²/s)

V_k = velocity at the roughness height as found by the equation:

21.24.2.2.13.17.

$$\frac{V_k}{V} = (2\eta_k - 2\eta_k^3 + \eta_k^4) + \frac{\Lambda}{6}(\eta_k - 3\eta_k^2 + 3\eta_k^3 - \eta_k^4)$$

The term η_k is defined as:

21.16.2.1.3.

$$\eta_k = \frac{x_k}{\delta}$$

if $x_k \leq \delta$ and $\eta_k = 1$ if $x_k > \delta$

21.16.2.1.4.

The solution procedure is as follows:

- 1) The flow function $U(s)$ together with its derivative dU/ds are known (Sections 10 and 11);
- 2) Integration of the above equations gives the shape factors $Z(s)$ and $K(s)$ so that displacement thickness δ_2 can be calculated;
- 3) The first shape factor $\Lambda(s)$ is found;
- 4) The boundary layer thickness $\delta(s)$ is found;
- 5) Finally, the velocity distribution is found.

21.16.2.1.5.

Assigning a transition location to the first wrap distance from the stagnation point where the roughness Reynolds number has exceeded the critical Reynolds number.

21.16.2.1.6.

Calculation of the laminar heat transfer coefficient prior to the transition location from the equation:

21.16.2.1.7.

$$h_l = \frac{2k}{\delta_T}$$

The terms in this equation have the following definitions:

21.16.2.1.8.

δ_T = thermal boundary thickness (m) as given by the equation:

$$\left(\frac{\delta_T}{c}\right)^2 \frac{V_\infty c}{\nu} = \frac{46.72}{\left(\frac{V}{V_\infty}\right)^{2.87}} \int_0^{\frac{s}{c}} \left(\frac{V}{V_\infty}\right)^{1.87} d\left(\frac{s}{c}\right)$$

21.16.2.1.9.

c = chord length of body geometry(s)

k = thermal conductivity of air (W/m²/°K)

h_l = laminar heat transfer coefficient (W/m²/°K)

V = velocity at a given surface location (m/s)

s = surface distance from stagnation point (m)

ν = kinematic viscosity of air (m²/s)

V_x = flight velocity (m/s)

21.16.2.1.10.

Calculation of the turbulent heat transfer coefficient after the transition location from the equation:

21.16.2.1.11.

$$h_t = \frac{\frac{1}{2}c_f \rho V C_p}{0.9 + \sqrt{\frac{c_f}{2} 0.52 Re_{k_t}^{0.45} Pr^{0.8}}}$$

The terms in this equation have the following definitions:

21.16.2.1.12.

h_t = turbulent heat transfer coefficient (W/m²/°K)

ρ = air density at a given surface location (kg/m³)

V = air velocity at a given surface location (m/s)

C_p = specific heat of air (J/kg/°K)

Pr = Prandtl number (dimensionless) =

$$\frac{C_p \mu}{k}$$

k = thermal conductivity of air (W/m²/°K)

μ = viscosity of air (kg/m/s)

$$Re_{k_t}$$

= turbulent roughness Reynolds number as given by

21.16.2.1.13.

$$Re_{k_t} = \frac{Vc}{\nu} \sqrt{\frac{c_f}{2}}$$

21.16.2.1.14.

c_f = skin friction coefficient (dimensionless) as given by the empirical expression:

$$c_f = \frac{0.3362}{\left(\log\left(\frac{864\theta_t}{x_k} + 2.568\right)\right)^2}$$

The terms in this equation have the following meaning

21.16.2.1.15.

x_k = roughness height (m) from Section 14

θ_t = turbulent momentum thickness (m) as given by the equation:

$$\theta_t = \frac{0.36\nu^{0.2}}{\nu^{3.29}} \left(\int_{s_t}^s V^{3.86} ds \right)^{0.8} + \delta_2|_{s_t}$$

Terms in this equation which have not been previously defined in this section are defined as follows:

21.16.2.1.16.

s_t = wrap distance from stagnation point to transition location (m)

$$\delta_2|_{s_t}$$

= laminar momentum thickness at transition location (m) as calculated in Section 15.2.1.4.5

21.16.2.2. When the program performs an integral boundary layer integration, a separate integration shall be made for each body geometry.

21.16.2.3. If the heat transfer coefficient is supplied by a grid-based flow solution, its definition and requirements are a function of that grid-based flow module and is not specified by this document.

21.17. Energy Balance Equation

21.17.1. Description/Purpose

21.17.1.1. The energy balance equation determines the amount of heat gain and loss from different factors and determines the energy available at each location for freezing the available water. The purpose of this section is to describe the requirements for this equation.

21.17.2. Energy Balance Equation Requirements

21.17.2.1. The program shall solve the following equation for the temperature distribution on the surface of the body geometry(s):

21.17.2.1.1.

$$-\left(k_{\varphi} \frac{\partial T}{\partial \varphi}\right)_{\varphi=0} = q''_{nc} + q''_{evap} - q''_{ke} - q''_{lat} \pm q''_{sens}$$

The terms in this equation have the following definitions:

21.17.2.1.2.

k_{φ} = thermal conductivity of body in direction normal to the surface (W/m/°K)

T = temperature at a given location on the body geometry(s) (°K)

φ = direction normal to body surface (m)

q''_{nc} = net convective heat loss from body (W/m²)

q''_{evap} = evaporative heat loss from surface water (W/m²)

q''_{ke} = kinetic heat gain to body from impinging droplets (W/m²)

q''_{lat} = latent heat gain to body from freezing (W/m²)

q''_{sens} = sensible heat loss (or gain) from surface water (W/m²)

21.17.2.2. The net convective heat loss shall be determined by the equation:

21.17.2.2.1.

$$q''_{nc} = (q''_{conv} - q''_{fh})A_{ratio}$$

The terms in this equation have the following definition:

21.17.2.2.2.

q''_{conv} = convective heat loss (W/m²) as defined by the equation:

$$q''_{conv} = h(T_s - T_{\infty})$$

The terms in this equation have the following definitions:

21.24.2.2.13.1.

h = convective heat transfer coefficient (W/m²/K) as determined from Section 15

T_s = temperature at a given surface location on a body geometry (°K).

T_x = ambient static temperature (°K)

21.17.2.2.3.

\tilde{q}_{fh} = heat gain on surface from frictional heating (W/m²) as defined by the equation:

$$\mathbf{q''}_{fh} = \mathbf{h}(T_{rec} - T_{\infty})$$

T_{rec} has the following definition:

21.24.2.2.13.1.

T_{rec} = recovery temperature (°K) at a given surface location on a body geometry as given by the equation:

$$T_{rec} = \frac{T_o \left(1 + r \frac{\gamma - 1}{2} M^2 \right)}{\left(1 + \frac{\gamma - 1}{2} M^2 \right)}$$

The terms in this equation have the following definitions:

21.24.2.2.13.2.

γ , M and T_o were previously defined in Section 12

r = recovery factor (dimensionless) = $Pr^{1/2}$ if the boundary layer is laminar as defined in Section 15 and $r = Pr^{1/3}$ if the boundary layer is turbulent as defined in Section 15.

Pr = Prandtl number (dimensionless) as given in 15.2.1.6.2

21.17.2.2.4.

The area ratio (A_{ratio}) is defined as the ratio of the surface area of a bead on the surface with respect to the surface area of the control volume and has the following definition:

$$A_{ratio} = \frac{2}{1 + \cos \theta_c} (1 - w_{fz})$$

The terms in this equation have the following definitions:

$$\theta_c = \text{contact angle (radians)} = \pi/90 * (T_{mp} - T_s)$$

$$T_{mp} = \text{melting point temperature for ice (}^\circ\text{K)}$$

w_f = wetness factor constant (dimensionless) = 1 in impingement region and = 0.3 elsewhere

z_f = wetness factor = 1 in impingement region and

$$z_f = \left(1 - \frac{\theta_c}{\sin \theta_c}\right)$$

elsewhere

21.17.2.3. The evaporative heat loss shall be defined by the equation:

21.17.2.3.1.

$$q''_{evap} = \frac{A_{ratio} L_v h_m (MW)_w}{\hat{R}} \left(\frac{e_s}{T_s} - \frac{e_e}{T_e} \right)$$

The terms in this equation have the following definitions:

21.17.2.3.2.

A_{ratio} = area ratio as defined above

L_v = latent heat of vaporization (J/kg)

h_m = mass transfer coefficient (m/s)

$(MW)_w$ = molecular weight of water = 18 kg/kg-mole

\hat{R}

= molar ideal gas constant = 8337.5 kg*m²/kg-mole/s²/°K

T_s = temperature at a given surface location on a body geometry (°K).

T_e = Temperature at the edge of boundary layer (°K) as given in Section 12.2.6.1

e_s = saturated evaporative pressure at a given surface location on a body geometry (N/m²)

e_e = saturated evaporative pressure at the edge of boundary layer (N/m²)

21.17.2.3.3.

The saturated evaporative pressure shall be defined by the following empirical equations:

$$e = 6894.7 \exp\left(20.15247167 - \frac{11097.16963}{1.8T}\right)$$

for $T < T_{mp} = 273.15^\circ\text{K}$

$$e = 6894.7 \exp\left(14.56594634 - \frac{7129.219482}{1.8T - 72}\right)$$

for $T \geq T_{mp}$

T_{mp} = melting point temperature for ice ($^\circ\text{K}$)

21.17.2.3.4.

The mass transfer coefficient shall be defined by the following empirical equation:

21.24.2.2.13.1.

$$h_m = \frac{h}{(\rho C_p)_{air} \mathcal{L}^{2/3}}$$

The terms in this equation have the following definitions:

21.24.2.2.13.2.

ρ_{air} = density of air (kg/m^3) at a given location as given in Section 12.

$C_{p,air}$ = specific heat of air ($\text{J}/\text{kg}/^\circ\text{K}$)

\mathcal{L} = Lewis number (dimensionless) which is defined by the equation:

$$\mathcal{L} = \frac{k_{air}}{\rho_{air} C_{p,air} D_{aw}}$$

The terms in this equation have the following definitions:

21.24.2.2.13.3.

ρ_{air} = density of air (kg/m^3) at a given location as given in Section 12.

$C_{p,air}$ = specific heat of air ($\text{J}/\text{kg}/^\circ\text{K}$)

k_{air} = thermal conductivity of air ($\text{W}/\text{m}/^\circ\text{K}$)

\mathcal{D}_{aw} = mass diffusivity between water and air (m²/s)

21.17.2.4. The kinetic heat gain from the impinging droplets shall be defined by the following equation:

21.17.2.4.1.

$$q''_{ke} = \frac{\beta_o LWC V_\infty^3}{2}$$

The terms in this equation have the following definitions:

21.17.2.4.2.

β_o = local collection efficiency as defined in Section 9.2.4

LWC = ambient liquid water content (kg/m³)

V_x = flight velocity (m/s)

21.17.2.5. The latent heat gain shall be defined by the following equation:

21.17.2.5.1.

$$q''_{lat} = N_f \beta_o LWC V_\infty L_f$$

The terms in this equation which have not been defined in this section are defined as follows:

21.17.2.5.2.

L_f = latent heat of fusion (J/kg)

N_f = mass fraction of incoming water which does not freeze [freezing fraction] (dimensionless) as defined in the following equation:

$$N_f = \frac{T_{mp} + \Delta T_r - T_s}{\Delta T_r}$$

ΔT_r = temperature range between water and ice phases (°K)

21.17.2.6. The sensible heat transfer shall be defined by the following equations:

21.17.2.6.1.

This equation shall be used when $T_s < T_{mp}$ (rime ice)

$$q_{sens} = -m_{im}(C_{p_w}(T_{mp} - T_\infty) + C_{p_i}(T_s - T_{mp})) - m_{rb_{in}}(C_{p_w}(T_{mp} - T_{rb}) + C_{p_i}(T_s - T_{mp}))$$

The terms which have not been defined in this section have the following definitions:

m_{im}

= mass flux of impinging water = $\beta_o L W C V_x$ (kg/m²s)

m_{rb_{in}}

= mass flux of runback water entering a given section (kg/m²s) as defined in Section 17

T_{rb} = temperature at previous surface location on a body geometry (runback temperature) (°K)

C_{p_w} = specific heat of water (J/kg/°K)

C_{p_i} = specific heat of ice (J/kg/°K)

21.17.2.6.2.

This equation shall be used when $T_{mp} < T_s < T_{mp} + yT_r$

$$q_{sens} = -m_{im}C_{p_w}(T_{mp} - T_{\infty}) - m_{rb_{in}}C_{p_w}(T_{mp} - T_{rb})$$

21.17.2.6.3.

This equation shall be used when $T_s > T_{mp} + yT_r$

$$q_{sens} = -m_{im}C_{p_w}(T_s - T_{\infty}) - m_{rb_{in}}C_{p_w}(T_s - T_{rb})$$

21.18. Mass Balance Equation

21.18.1. Description/Purpose

21.18.1.1. The mass balance equation determines the amount of mass gain and loss from different factors and determines the mass of water which freezes at each location. The purpose of this section is to describe the requirements for this equation.

21.18.2. Mass Balance Equation Requirements

21.18.2.1. The program shall solve the following equation for the mass flux of runback water leaving at each location on a body geometry:

21.18.2.1.1.

$$m_{im} + m_{rb_{in}} = m_e + m_f + m_{rb_{out}} \frac{\Delta s_{next}}{\Delta s_{current}} + m_{sh} + m_{st}$$

The terms in this equation have the following definitions:

21.18.2.1.2.

m_{im}

= mass flux of impinging water = $\beta_o L W C V_x$ (kg/m²s)

$m_{rb_{in}}$

= mass flux of runback water entering a given section (kg/m²s)

m_e

= mass flux of evaporating water (kg/m²s)

$m_{rb_{out}}$

= mass flux of runback water leaving a given section (kg/m²s)

m_f

= mass flux of water freezing in a given section (kg/m²s)

m_{sh}

= mass flux of water shedding (kg/m²s) in a given section. This physical phenomena is different from droplet splashing as given in Section 6

m_{st}

= mass flux of standing water (kg/m²s) due to surface tension effects

$$\Delta s_{next}$$

= surface distance spacing at next location (m)

$$\Delta s_{current}$$

= surface distance spacing at current location (m)

21.18.2.2. The mass flux of runback water entering the stagnation point shall be equal to 0 (no mass flow).

21.18.2.3. The direction of runback mass flow shall be in the surface direction away from the stagnation point. (Therefore the calculation of 17.2.1.1 is initiated at the stagnation point and is calculated going away from this point on either side so that

$$m_{rb_{in}}$$

is a known quantity at each location.)

21.18.2.4. The mass flux of evaporating water shall be defined from the following equations:

21.18.2.4.1.

$$m_e = \frac{q_{evap}}{L_v}$$

when

$$m_e < m_{im} + m_{rb_{in}}$$

21.18.2.4.2.

$$m_e = m_{im} + m_{rb_{in}}$$

when

$$m_e \geq (m_{im} + m_{rb_{in}})$$

in the previous equation

21.18.2.5. In the case cited by Section 17.2.4.2, the other mass fluxes on the right hand side of 17.2.1.1 are identically zero.

21.18.2.6. The mass flux lost by water shedding shall be determined by the empirical equations:*

21.18.2.6.1.

$$m_{sh} = m_{rb_{out}} \left(1 - \frac{We_c}{We} \right)$$

for $We \geq We_c$

21.18.2.6.2.

$$m_{sh} = 0$$

for $We < We_c$

The terms in these equations have the following definitions:

21.18.2.6.3.

We_c = critical bead Weber number (dimensionless) which shall be determined from the following empirical expression:

$$We_c = 200 + 50000 \frac{x_k}{c}$$

x_k = surface roughness (m) as defined in Section 14

c = chord length of the body geometry(s)

21.18.2.6.4.

We = bead Weber number (dimensionless) which shall be defined by the equation:

$$We = \frac{\rho V^2 d_b}{\sigma}$$

The terms in this equation have the following definitions:

21.18.2.6.5.

ρ = air density (kg/m^3) at a given location at the edge of the boundary layer as defined in Section 12

V = air velocity (m/s) at a given location at the edge of the boundary layer as defined in Section 12

σ = surface tension between water and air (kg/s^2)

d_b = diameter of a bead of surface water (m) which shall be defined from the equation:

$$d_b = \frac{(m_{st})_{t=t-\Delta t} + m_{im}\Delta t}{\rho_w}$$

The terms in this equation have the following definitions:

21.18.2.6.6.

$$(m_{st})_{t = t - \Delta t}$$

= standing water from previous time step (kg/m²)

yt = current time step (sec)

ρ_w = water density (kg/m³)

21.18.2.7. The mass flux of standing water shall be determined by the following equation:

21.18.2.7.1.

$$m_{st} = \frac{h_b \rho_w}{\Delta t} + (m_{st})_{t = t - \Delta t}$$

where h_b = height of water bead =

$$\frac{1}{2} d_b$$

21.18.2.8. The mass flux of water freezing shall be determined from the following equation:

21.18.2.8.1.

$$m_f = N_f(m_{im} + m_{rb_{in}})$$

The terms in this equation have been previously defined in this section.

*Not implemented in LEWICE 3.2 Release.

21.19. One Dimensional Anti-Icing

21.19.1. Description/Purpose

21.19.1.1. The purpose of anti-icing is to determine the amount of heat required to keep the surface of the body geometry(s) free of ice by supplying bleed air from a compressor. or by supplying heat from an electrothermal heater pad inside the body geometry using a one dimensional steady-state heat transfer analysis.

21.19.2. One Dimensional Anti-Icing Requirements

21.19.2.1. The program shall determine the heat requirements for either a bleed air system or an electrothermal system based on an input surface temperature or based on an assumption that all of the impinging water evaporates.

21.19.2.1.1.

The four options stated above (bleed air evaporative, bleed air running wet, electrothermal evaporative and electrothermal running wet) shall be determined by the user.

21.19.2.2. For an input surface temperature, the program shall solve for the surface heat flux from the equation:

21.19.2.2.1.

$$q''_{surf} = q''_{nc} + q''_{evap} - q''_{ke} \pm q''_{sens}$$

The terms in this equation are defined in Section 16 with the additional definition:

21.24.2.2.13.1.

$$q''_{surf}$$

= heat flux at the surface (W/m²)

21.19.2.2.2.

The mass fluxes needed for the previous equation shall be calculated from Section 17 with the stipulation that

$$m_f$$

$$= 0$$

21.19.2.3. For an evaporative system, the surface heat flux shall be solved for as stated in Section 18.2.2 with the following alterations:

21.19.2.3.1.

The mass flux equation of Section 17 reduces to the following equation:

$$m_{im} = m_e$$

21.19.2.3.2.

The surface temperature shall be calculated from the equation in 18.2.3.1 noting the definitions of these terms from Sections 17 and 16.

21.19.2.4. For a bleed air system, the required heat flux shall be equal to the surface heat flux as defined in Sections 18.2.2 and 18.2.3.

21.19.2.5. For an electrothermal system, the required heat flux shall be calculated from the following equation:

21.19.2.5.1.

$$q''_{required} = q''_{surf} - \frac{T_s - T_\infty - q''_{surf} \left(\frac{\Delta y_h}{2k_h} + \sum_{j=h+1}^n \frac{\Delta y_j}{k_j} \right)}{\frac{1}{h_{inside}} + \frac{\Delta y_h}{2k_h} + \sum_{j=1}^{h-1} \frac{\Delta y_j}{k_j}}$$

21.19.2.5.2.

The terms in this equation are defined as follows:

$q''_{required}$

= required heat flux (W/m²)

T_s = surface temperature (K)

T_∞ = ambient temperature (K)

q''_{surf}

= surface heat flux (W/m²)

Δy_h = thickness of heater (m)

k_h = thermal conductivity of heater material (W/mK)

Δy_j = thickness of layer j in body structure (m)

k_j = thermal conductivity of material in layer j (W/mK)

n = number of layers in body structure

h = index number of heater layer in body structure as counted from the inside surface of the body structure

h_{inside} = convective heat transfer coefficient on inside surface of body structure (W/m²K). For a bleed air system, this is the heat transfer coefficient supplied by the bleed air. For an electrothermal system, this is the natural convection heat transfer coefficient.

21.19.2.6. For a bleed air system, the program shall calculate the local bleed air temperature from the following equation:

21.19.2.6.1.

$$T_{air} = T_s + q''_{surf} \left[\frac{1}{h_{inside}} + \sum_{j=1}^n \frac{\Delta y_j}{k_j} \right]$$

21.19.2.6.2.

The terms in the previous equation have already been defined in this section with the addition:

T_{air} = local bleed air temperature (°K)

21.19.2.7. For an electrothermal system, the program shall calculate the temperature at the top of the heater layer from the following equation:

21.19.2.7.1.

$$T_{h_{top}} = T_s - q''_{surf} \sum_{j=h+1}^n \frac{\Delta y_j}{k_j}$$

21.19.2.7.2.

The terms in this equation have been previously defined in this section with the addition:

$T_{h_{top}}$

= temperature at top of the heater layer (°K)

21.19.2.8. For an electrothermal system, the program shall calculate the temperature at the bottom of the heater layer from the following equation:

21.19.2.8.1.

$$T_{h_{bot}} = T_{h_{top}} - \frac{(2q''_{surf} - q''_{required})\Delta y_h}{2k_h}$$

21.19.2.8.2.

The terms in the following equation have been defined previously in this section with the addition:

$$T_{h_{bot}}$$

= temperature at bottom of the heater layer (K)

- 21.19.2.9. For a one-dimensional electrothermal system which uses the heat fluxes specified for use in Section 21, the program shall instead calculate the unknowns

$$q''_{surf}$$

and T_s using the known heat fluxes

for the values of

$$q''_{required}$$

d in the above equations.

- 21.19.2.10. For a one-dimensional bleed air system which uses the heat fluxes or heat transfer coefficients specified for use in Section 21, the program shall instead calculate the unknowns

$$q''_{surf}$$

and T_s

using the known heat fluxes for the values of

$$q''_{required}$$

d in the

above equations.

- 21.19.2.11. For sections 18.2.9 and 18.2.10, the program should allow the use of the output from this section to override or replace the mass calculated in Section 17 for an unheated surface.

21.20. Effective Heat Transfer Coefficient

21.20.1. Description/Purpose

21.20.1.1. The purpose of this module is to calculate the effective heat transfer coefficient for use when interfacing LEWICE 3.2 with other software such as an internal flow solver for bleed air applications.

21.20.2. Effective Heat Transfer Requirements

21.20.2.1. The effective heat transfer coefficient is defined by the equation:

21.20.2.1.1.

$$q''_{surface} = h_{eff}(T_s - T_{rec})$$

21.20.2.1.2.

$q''_{surface}$ is determined from Section 18.

21.20.2.1.3.

T_{rec} is defined in Section 16.2.2.3.1

21.20.2.1.4.

First, a temporary value using $h_{eff,temp}$ and a reference temperature, T_{ref} , are obtained without evaporation. For a mixed phase, $h_{eff,temp}$ and T_{ref} are defined by the following equations:

21.20.2.1.5.

$$h_{eff,temp} = h + (m_{im} + m_{rb,in})C_{p_w}$$

21.20.2.1.6.

$$T_{ref} = \frac{hT_{rec} + m_{im}C_{p_w}T_{\infty} + m_{rb,in}C_{p_w}T_{rb}}{h_{eff,temp}}$$

21.20.2.1.7.

The effective heat transfer coefficient is then calculated from the following equation:

$$h_{eff} = h_{eff,temp} + N_f L_f (m_{im} + m_{rb,in}) - q''_{evap}$$

21.20.2.1.8.

The estimated freezing fraction at each location on the body geometry shall be calculated from the following equation:

21.20.2.1.9.

$$N_f = \frac{-q''_{surface} + h(T_{mp} - T_{rec}) + q''_{evap} + m_{im}C_{p_w}(T_{mp} - T_{\infty}) + m_{rb,in}C_{p_w}(T_{mp} - T_{rb})}{L_f(m_{im} + m_{rb,in})}$$

21.20.2.1.10.

The terms in this equation are defined in Sections 16 and 17

21.20.2.1.11.

The evaporative term shall be evaluated using $T_s = T_{mp}$

21.20.2.2. For a solid (rime ice) condition, $h_{eff,temp}$ and T_{ref} are defined by the following equations:

21.20.2.2.1.

$$h_{eff,temp} = h + (m_{im} + m_{rb,in})C_{p_i}$$

21.20.2.2.2.

$$T_{ref} = \frac{hT_{rec} + q''_{sens}}{h_{eff,temp}}$$

21.20.2.2.3.

The term

$$q''_{sens}$$

is determined by the following equation:

$$q''_{sens} = m_{im}C_{p_w}(T_{\infty} - T_{mp}) + m_{rb,in}C_{p_w}(T_{rb} - T_{mp}) + C_{p_i}T_{mp}(m_{im} + m_{rb,in})$$

21.20.2.2.4.

Other terms (including those in 19.2.3.2.1) are defined in Sections 16 and 17

21.20.2.2.5.

The effective heat transfer coefficient is then calculated from the following equation:

$$h_{eff} = h_{eff,temp} + L_f(m_{im} + m_{rb,in}) - q''_{evap}$$

21.20.2.2.6.

For a liquid phase condition (including fully evaporative cases), $h_{eff,temp}$ and T_{ref} are defined by the following equations:

21.20.2.2.7.

$$h_{eff,temp} = h + (m_{im} + m_{rb,in})C_{p_w}$$

21.20.2.2.8.

$$T_{ref} = \frac{hT_{rec} + m_{im}C_{p_w}T_{\infty} + m_{rb_{in}}C_{p_w}T_{rb}}{h_{eff}}$$

21.20.2.2.9.

For a fully evaporative condition,

$$q''_{evap}$$

is defined by the following equation:

21.20.2.2.10.

$$q''_{evap} = (m_{im} + m_{rb_{in}})L_v$$

21.20.2.2.11.

The effective heat transfer coefficient is then calculated from the following equation:

$$h_{eff} = h_{eff_{temp}} - q''_{evap}$$

21.21. Internal Heat Transfer Coefficient

21.21.1. Description/Purpose

21.21.1.1. The purpose of this module is to calculate the internal heat transfer coefficient for jet impingement applications such as a piccolo tube.

21.21.2. Internal Heat Transfer Requirements

21.21.2.1. The internal heat transfer coefficient is defined by one of the following equations:

21.21.2.1.1.

For the user flag ICORR = 1, the internal heat transfer coefficient is defined by

$$Nu = C(Re^{REP}) \frac{\left(24 - \left| \left(\frac{z}{d} \right)^{ZP} - 7.75 \right| \right)}{533 + 44 \left(\frac{r}{d} \right)^{RP}}$$

21.21.2.1.2.

For the user flag ICORR = 2, the internal heat transfer coefficient is defined by

$$Nu = C(Pr)^{\frac{1}{3}} (Re^{REP}) \left(\frac{z}{d} \right)^{ZP} \left(\frac{c_n}{d} \right)^{RRP} \left(\frac{r}{d} \right)^{RP} \left(\frac{D}{d} \right)^{DP}$$

21.21.2.1.3.

The terms in these equations have the following definition:

21.21.2.1.4.

$$Re = \frac{4mc_n}{\pi N \mu d}$$

21.21.2.1.5.

$$Pr = \frac{C_p \mu}{k}$$

21.21.2.1.6.

m

= air mass flow rate per unit span (kg/ms)

c_n = spanwise distance between holes (m)

$\pi = 3.1415926\dots$

N = number of holes

μ = air viscosity (kg/ms)

k = air conductivity (W/mK)

C_p = air heat capacity (J/kgK)

d = hole diameter (m)

r = radial distance from hole (m)

z = distance from hole to inside surface of the body (m)

D = piccolo tube diameter (m)

C, REP, ZP, RP, RRP, DP = correlation constants (dimensionless)

21.22. Two Dimensional De-icing/Anti-Icing

21.22.1. Description/Purpose

21.22.1.1. The purpose of this module is to calculate the temperatures on a two dimensional grid inside the body geometry(s) using a given amount of heat from bleed air or an electrothermal heater pad inside the body geometry. This module can also be used to determine the heat required to keep the surface of the body geometry(s) at a specified surface temperature using the same type of heat input as described earlier.

21.22.2. Two Dimensional De-Icing Requirements

21.22.2.1. The program shall solve the following equation within the airfoil geometry:

21.22.2.1.1.

$$(\rho C_p)_k \frac{\partial T}{\partial t} = k_{x,k} \frac{\partial^2 T}{\partial x^2} + k_{y,k} \frac{\partial^2 T}{\partial y^2} + q'''_k$$

21.22.2.1.2.

The terms in this equation are defined as follows:

ρ_k = density of material k (kg/m³)

$C_{p,k}$ = specific heat of material k (J/kg/°K)

T = temperature at each location in material k (°K)

k_x = thermal conductivity of material k in direction parallel to surface (W/m/K)

k_y = thermal conductivity of material k in direction normal to surface (W/m/K)

q'''_k

= volumetric heat source emanating from material k (W/m³)

t = time (s)

x = direction parallel to surface (m)

y = direction normal to surface (m)

21.22.2.1.3.

In the ice/water region, the above equation shall be replaced with the following expression:

$$\frac{\partial H}{\partial t} = k_{i,w} \frac{\partial^2 T}{\partial x^2} + k_{i,w} \frac{\partial^2 T}{\partial y^2}$$

21.22.2.1.4.

The additional terms are defined as follows:

H = enthalpy in ice/water region (J/m³)

k_{i,w} = thermal conductivity of ice or water (changed as needed)
(W/m/K)

21.22.2.2. The program shall solve the above equation for up to 29 different materials in the parallel direction and for up to 29 different materials in the direction normal to the surface.

21.22.2.3. The program shall use the following relationships at the interface between materials within the airfoil geometry:

21.22.2.3.1.

$$T_{material1} = T_{material2}$$

at every interface between different materials

21.22.2.3.2.

$$-k_x \left(\frac{\partial T}{\partial x} \right)_{material1} = -k_x \left(\frac{\partial T}{\partial x} \right)_{material2}$$

at interfaces normal to the surface

21.22.2.3.3.

$$-k_y \left(\frac{\partial T}{\partial y} \right)_{material1} = -k_y \left(\frac{\partial T}{\partial y} \right)_{material2}$$

at interfaces parallel to the surface

21.22.2.4. The program shall be able to use any of the following boundary conditions for each external interface of the airfoil geometry:

21.22.2.4.1.

$$T)_{surface} = T_{const}$$

21.22.2.4.2.

The temperature constant (T_{const}) can be different for each external boundary

21.22.2.4.3.

$$-k_x \left(\frac{\partial T}{\partial x} \right)_{surface} = h(T_{surface} - T_{const})$$

for an external boundary normal to the surface (at interior of de-icer or at airfoil (or ice) external surface).

21.22.2.4.4.

$$-k_y \left(\frac{\partial T}{\partial y} \right)_{surface} = h(T_{surface} - T_{const})$$

for an external boundary parallel to the surface (at ends of airfoil)

21.22.2.4.5.

The temperature constant (T_{const}) and heat transfer coefficient (h) can be different for each external boundary.

21.22.2.4.6.

$$-k_x \left(\frac{\partial T}{\partial x} \right)_{surface} = q''_{surface}$$

for an external boundary normal to the surface

21.22.2.4.7.

$$-k_y \left(\frac{\partial T}{\partial y} \right)_{surface} = q''_{surface}$$

for an external boundary parallel to the surface

21.22.2.4.8.

The surface heat flux ($q_{surface}$) can be different for each external boundary.

21.22.2.4.9.

In place of the boundary conditions specified above, the outer surface boundary of the airfoil geometry shall be also able to use the equations in Section 16 as the external boundary condition.

21.22.2.4.10.

In place of the boundary conditions specified above, the outer surface boundary of the airfoil geometry shall be also able to use the following boundary condition:

21.22.2.4.11.

$$-k_y \left(\frac{\partial T}{\partial y} \right)_{surface} = h(T_{surface} - T_{rec})$$

where T_{rec} is defined in Section 16.2.2.3.1

21.22.2.4.12.

The heat transfer coefficient for the external boundary parallel to the surface may be defined in three ways:

21.24.2.2.13.1.

A user-specified constant

21.24.2.2.13.2.

A user-specified array as a function of wrap distance

21.24.2.2.13.3.

From the equations listed in Section 15

21.24.2.2.13.4.

The heat transfer coefficient for the internal boundary parallel to the surface may be defined in two ways:

21.24.2.2.13.5.

A user-specified constant

21.24.2.2.13.6.

A user-specified array as a function of wrap distance

21.24.2.2.13.7.

The surface heat flux for the boundaries parallel to the surface may be defined in two ways:

21.24.2.2.13.8.

A user-specified constant

21.24.2.2.13.9.

A user-specified array as a function of wrap distance

21.22.2.5. The ice layer shall be allowed to shed (be removed from the computation) when the following condition is met:

21.22.2.5.1.

$$F_{ad} \leq \sqrt{(F_n^2 + F_r^2 - F_p^2)}$$

The terms in this equation have the following definitions:

F_{ad} = adhesion force (Pascals) as defined by the empirical equations*

$$F_{ad} = 14583(T_{mp} - T_s)$$

when 267.7K T_s T_{mp}

$$F_{ad} = 34475(1.5(267.3 - T_s) + 4)$$

when 258.15K T_s $267.7K$

$$F_{ad} = 34475(3.79(258.15 - T_s) + 12.5)$$

when T_s $258.15K$

T_{mp} = melting point of ice = 273.15K

T_s = surface temperature (K)

F_p = component of force parallel to the ice as provided by the flow solution in Sections 10 or 11

F_n = component of force normal to the ice as provided by the flow solution in Sections 10 or 11

F_r = component of force on the ice as provided by a rotating airfoil (rotor or propeller) as defined by

$$F_r = \frac{\rho_{air} V_r^2 r}{2(3600)}$$

where

ρ_{air} = air density (kg/m³)

V_r = rotor velocity (RPM) (=0 for fixed wing simulations)

r = distance of airfoil section from rotor hub (m)

*These empirical equations were derived from experimental data with a reported error rate of ±100%. *Caveat emptor.*

21.22.2.5.2.

Only ice is shed by these mechanisms. If the ice/water layer contains only water, it is shed by the equations in Section 17.2.6 or it is subjected to runback or standing water as defined in Section 17

21.22.2.5.3.

The shedding formula provided above can be applied by using one of three user-specified options:

21.24.2.2.13.1.

It is applied independently at each grid location

21.24.2.2.13.2.

It is applied as a summation of forces for each section in the direction parallel to the surface with each section shedding independent of the others. (default)

21.24.2.2.13.3.

It is applied as a summation of forces for the entire ice shape, meaning that the entire ice shape must shed as a whole unit or not at all.

- 21.22.2.6. The equations in this section shall be discretized into a two-dimensional grid with implicit central differencing (second order accuracy of Taylor series) in the spatial direction and first-order differencing in time.

21.23. Pneumatic Boot Requirements

21.23.1. Description/Purpose

21.23.1.1. The purpose of this module is to calculate the ice shape as previously defined except within the limits prescribed by the user. Within the user-defined limits, ice thickness is limited to the residual height which is also specified by the user.

21.23.2. Boot Requirements

21.23.2.1. The program shall read in the upper and lower boot limits and a residual ice height.

21.23.2.1.1.

Within these limits, ice accretion is limited by the residual height

21.23.2.1.2.

Outside these limits, the ice mass is calculated using the procedure in Section 17.

21.24. Error Checking

21.24.1. Description/Purpose

21.24.1.1. The variables input by the user must be checked for errors and for the ranges imposed by the available test data.

21.24.2. Error Checking Requirements

21.24.2.1. The program shall check the following input data and issue a warning message for the following inputs:

21.24.2.1.1.

The automated time step procedure was not selected.

21.24.2.1.2.

The simulation time is > 45 minutes (data limit).

21.24.2.1.3.

The number of body geometries is > 1 and d 5

21.24.2.1.4.

The number of flow solutions is < the number calculated for the automated time step procedure.

21.24.2.1.5.

The point spacing for any body geometry is $S 8 \times 10^{-4}$

21.24.2.1.6.

The number of trajectories is < 10 or > 50

21.24.2.2.13.1.

The warning message for number of trajectories < 10 shall indicate that this number is insufficient and the program shall set this variable equal to 24.

21.24.2.2.13.2.

The warning message for number of trajectories > 50 shall indicate that this number is unnecessarily high.

21.24.2.1.7.

The particle density is d 980 kg/m^3 or S 1020 kg/m^3 .

21.24.2.1.8.

A grid-based flow solution is selected.

21.24.2.1.9.

An additional warning if an interactive grid-based flow solution is selected.

21.24.2.1.10.

The anti-icing option is selected.

21.24.2.1.11.

The sum of the particle distribution is | 1.

21.24.2.2.13.1.

The program shall adjust the individual particle fractions such that their sum is =1 by the following equation:

$$\eta_{LWC_j} = \frac{\eta_{LWC_{inputj}}}{\sum_{j=1}^n \eta_{LWC_{inputj}}}$$

21.24.2.2.13.2.

The terms in this equation are defined as follows:

η_{LWC}

= fraction liquid water content (dimensionless)

$\eta_{LWC_{input}}$

= fraction liquid water content input

(dimensionless)

n = number of drop sizes input

j = index value for a given drop size in distribution

21.24.2.1.12.

The volume median drop size of the distribution is > 50 μm which acknowledges that the distribution is outside the FAA certification range.

21.24.2.1.13.

An additional warning if the volume median drop size of the distribution is > 270 μm (data limit).

21.24.2.1.14.

The angle of attack is < -6 or $> +6$ to indicate possible low separation.

21.24.2.1.15.

The ambient Mach number calculated in Section 23.2.2.13.1 is 0.6 to reflect limits of incompressible flow.

21.24.2.1.16.

The ambient liquid water content is $> 2 \text{ g/m}^3$ (data limit).

21.24.2.1.17.

The ambient temperature is $< 240 \text{ }^\circ\text{K}$ (outside meteorological probability) or $\leq 273.15 \text{ }^\circ\text{K}$ (no ice will form).

21.24.2.1.18.

Output files which are estimated $> 100 \text{ kB}$ will be generated.

21.24.2.1.19.

Blank lines are found in a body geometry input file.

21.24.2.2.13.1.

The program shall otherwise ignore the blank lines in a body geometry input file.

21.24.2.1.20.

A body geometry is not enclosed (first and last input data points are different).

21.24.2.1.21.

Two consecutive body geometry points are the same.

21.24.2.2.13.1.

The program shall remove one of the duplicate points from the body geometry.

21.24.2.1.22.

The body geometry points are input counter-clockwise.

21.24.2.2.13.1.

The program shall reverse the body geometry points so they are clockwise.

21.24.2.1.23.

The angle between two adjacent body segments is $> 45^\circ$.

21.24.2.1.24.

An additional warning is issued if the angle between two adjacent body segments is $> 145^\circ$.

21.24.2.1.25.

Number of thermal time steps must be > 0

21.24.2.1.26.

Number of thermal iterations must be $0 < N < 50$

21.24.2.1.27.

Time before spray cannot be < 0

21.24.2.1.28.

Time after spray cannot be < 0

21.24.2.1.29.

Time before spray is > 0

21.24.2.1.30.

Time after spray is > 0

21.24.2.1.31.

Number of thermal output files must be $0 \leq N < 50$

21.24.2.1.32.

Printout time number must be ≥ 0

21.24.2.1.33.

Printout type must be $1 \leq N \leq 5$

21.24.2.1.34.

Number of printouts must be between $0 \leq N < 50$

21.24.2.1.35.

Printout flag must be $1 \leq N \leq 3$

21.24.2.1.36.

Second printout flag must be $1 \leq N \leq 3$

21.24.2.1.37.

Printout layer must be 1 d N d number of layers.

21.24.2.1.38.

Printout section must be 1 d N d number of sections

21.24.2.1.39.

Printout time must be S 0

21.24.2.1.40.

Total number of columns for external data files must be S 2

21.24.2.1.41.

Column for 1st variable in external data files must be 1 d N d number of columns.

21.24.2.1.42.

Column for 2nd variable in external data files must be 1 d N d number of columns.

21.24.2.1.43.

Columns for 1st and 2nd variable in external data files cannot be the same.

21.24.2.1.44.

Offset may be past the trailing edge.

21.24.2.1.45.

Output file format flag must be 0 or 1

21.24.2.1.46.

The offset value may be past the trailing edge

21.24.2.1.47.

Boundary condition index must be $0 < N d 3$

21.24.2.1.48.

Constant temperature boundary condition is not normally used in icing calculations.

21.24.2.1.49.

Constant heat flux boundary condition is not normally used in icing calculations.

21.24.2.1.50.

Temperature at boundary is not equal to the ambient temperature input from the main LEWICE input file.

21.24.2.1.51.

Convective heat flux boundary condition is not normally used in icing calculations.

21.24.2.1.52.

Negative heat flux boundary condition is not normally used in icing calculations.

21.24.2.1.53.

Temperature for bleed air is less than freezing. This is not normally used in icing calculations. Mass flow rate for bleed air must be S_0

21.24.2.1.54.

Mass fraction going up must be $0 \leq N_d \leq 1$

21.24.2.1.55.

Mass flow rate input location is greater than one chord length.

21.24.2.1.56.

Heat flux flag must be 0 or 1

21.24.2.1.57.

Bleed air flag must be 0 or 1

21.24.2.1.58.

“Optimum” bleed air flag must be 0 or 1

21.24.2.1.59.

Heat flux flag and bleed air flag cannot both be on.

21.24.2.1.60.

“Optimum” bleed flag and bleed air flag cannot both be on.

21.24.2.1.61.

Heat flux flag and “optimum” bleed flag cannot both be on.

21.24.2.1.62.

Conduction flag must be 0 or 1

21.24.2.1.63.

Boundary flag must be 0 < N d 3

21.24.2.1.64.

Initial temperature flag must be 0 < N d 3

21.24.2.1.65.

Ice shed flag must be 0 or 1

21.24.2.1.66.

Shed type flag must be 0 < N d 2

21.24.2.1.67.

Phase change flag must be 0 or 1

21.24.2.1.68.

For anti-icing case, conduction flag must be = 1 since heaters will turn on immediately.

21.24.2.1.69.

Since conduction flag is off (ICOND = 0), then the boundary condition flag (IBOUND) must be = 3 and temperatures must be initialized to the accretion temperature (INIT = 3).

21.24.2.1.70.

Since phase change flag is off (IGDE = 0), then the boundary condition flag (IBOUND) must be = 2 and temperatures must be initialized to the recovery temperature (INIT = 2).

21.24.2.1.71.

Upper boundary is set to dry air. Ice will not form.

21.24.2.1.72.

Initial temperature selected is not normally used in icing simulations.

21.24.2.1.73.

Ice will not shed. This is usually too conservative.

21.24.2.1.74.

The preferred shedding mode is by sections (ISTD = 1).

21.24.2.1.75.

Phase change is not selected. This is not normally used in icing simulations.

21.24.2.1.76.

Ice cannot shed without phase change.

21.24.2.1.77.

3D streamline flag must be 0 or 1

21.24.2.1.78.

Desired surface temperature is less than freezing.

21.24.2.1.79.

Evaporation flag must be $0 < N d 2$

21.24.2.1.80.

De-ice flag must be 0 or 1

21.24.2.1.81.

Span location on rotor cannot be < 0

21.24.2.1.82.

Rotor speed cannot be < 0

21.24.2.1.83.

Rotor flag must be 0 or 1

21.24.2.1.84.

Thermal time steps < 0.0001 or > 1.0 should not be used.

21.24.2.2. The program shall check the following input data and issue an error message for the following inputs:

21.24.2.2.1.

The simulation time input is less than or equal to zero.

21.24.2.2.2.

The number of body geometries is < 1 or > 5 .

21.24.2.2.3.

The number of flow solutions is $d 0$.

21.24.2.2.4.

The point spacing for any body geometry is $d 0$.

21.24.2.2.5.

The particle density is $d 0$.

21.24.2.2.6.

Any drop size in the distribution is $d \geq 0$.

21.24.2.2.7.

The chord length is $d \geq 0$.

21.24.2.2.8.

The ambient velocity is $d \geq 0$.

21.24.2.2.9.

The ambient liquid water content is < 0 .

21.24.2.2.10.

The ambient static temperature is $d \geq 0$.

21.24.2.2.11.

The ambient static pressure is $d \geq 0$.

21.24.2.2.12.

The ambient relative humidity is $< 0\%$ or $> 100\%$.

21.24.2.2.13.

The ambient Mach number is $S \geq 1$

21.24.2.2.13.1.

The ambient Mach number is calculated from the equation:

$$M_{\infty} = \frac{V_{\infty}}{\sqrt{\gamma R T_{\infty}}}$$

21.24.2.2.13.2.

The terms in this equation are defined as follows:

M_x = ambient Mach number (dimensionless)

V_x = ambient velocity (m/s)

T_x = ambient temperature ($^{\circ}\text{K}$)

γ = heat capacity ratio = 1.4 (dimensionless)

R = ideal gas constant = $1430 \text{ m}^2/\text{s}^2/^{\circ}\text{K}$

21.24.2.2.14.

A given body geometry contains no data points.

21.24.2.2.15.

One body geometry lies completely inside any other body geometry.

21.24.2.2.16.

Any two body geometries intersect.

21.24.2.2.17.

Blank lines are read as data with a value of zero. Check input file for blank lines.

21.24.2.2.18.

Layer number must be $1 < N < 30$.

21.24.2.2.19.

Number of points in a layer must be $1 < N < 310$

21.24.2.2.20.

Length of a layer must be > 0 .

21.24.2.2.21.

Thermal conductivity of layer must be > 0 .

21.24.2.2.22.

Thermal diffusivity of layer must be > 0 .

21.24.2.2.23.

Anisotropic ratio of layer must be > 0 .

21.24.2.2.24.

Total number of points in normal direction is > 310 .

21.24.2.2.25.

Number of points in a section must be $1 < N < 310$

21.24.2.2.26.

Length of a section must be > 0

21.24.2.2.27.

Thermal conductivity of section must be > 0

21.24.2.2.28.

Thermal diffusivity of section must be > 0

21.24.2.2.29.

Anisotropic ratio of section must be > 0

21.24.2.2.30.

Additional length of a section must be ≥ 0

21.24.2.2.31.

Layer number for additional thickness must be $0 < N \leq$ number of layers.

21.24.2.2.32.

Total number of points in wrap direction < 310

21.24.2.2.33.

Heater number must be $0 < N \leq$ number of layers.

21.24.2.2.34.

Number of parameter studies must be $0 < N < 30$

21.24.2.2.35.

Heater wattage must be ≥ 0

21.24.2.2.36.

Heater on time must be ≥ 0

21.24.2.2.37.

Heater off time must be ≥ 0

21.24.2.2.38.

Heater lag time must be ≥ 0

21.24.2.2.39.

Temperature control flag must be $0 \leq N \leq$ number of layers

21.24.2.2.40.

Temperature control flag is on. ON temperature must be $>$ OFF temperature.

21.24.2.2.41.

Temperature for temperature controlled heater cannot be $<$ the ambient temperature.

21.24.2.2.42.

Slope cannot be zero.

21.24.2.2.43.

Number of beta values input is out of range

21.24.2.2.44.

Wrap distance value input is out of range.

21.24.2.2.45.

Number of hot air values input is out of range

21.24.2.2.46.

Number of streamline values input is out of range

21.24.2.2.47.

The ice layer cannot be the same as the heater layer.

21.24.2.2.48.

Heat transfer coefficient must be $S \geq 0$

References for Appendix A

- Abbott, I. H., and A. E. von DoenHoff, "Theory of Wing Sections," Dover Publications, New York, 1959.
- Al-Khalil, K. M., Keith, T. G., and K. J. De Witt, "Development of an Improved Model for Runback Water on Aircraft Surfaces",; AIAA 92-0042, Jan. 1992.
- Anderson, D. A., Tannehill, J. C., and R. H. Pletcher, "Computational Fluid Mechanics and Heat Transfer," McGraw-Hill Book Company, New York, 1984.
- Anderson, J. D., "Modern Compressible Flow with Historical Perspective," McGraw-Hill Book Company, New York, 1982.
- Bilanin, A. J. "Proposed Modifications to Ice Accretion/Icing Scaling Theory," AIAA88-0203, Jan. 1988.
- Bird, R.B., Stewart, W. E., and E. N. Lightfoot, "Transport Phenomena," John Wiley & Sons, Inc., New York, 1960.
- Cansdale, J. T. and R. W. Gent, "Ice Accretion on Aerofoils in Two-Dimensional Compressible Flow - A Theoretical Model," Royal Aircraft Establishment Technical Report 82128, 1983.
- Carlson, D.J. and Haglund, R.F. "Particle Drag and Heat Transfer in Rocket Nozzles." AIAA Journal, Vol. 2, No. 11, 1964, pp. 1980-1984.
- Chang, H., Frost, W., and Shaw, R.J. "Influence of Multidroplet Size Distribution on Icing Collection Efficiency." AIAA-83-0110, 1983.
- Clift, R., "Bubbles, Drops and Particles", Academic Press, New York, 1978.
- Crowe, C., Sommerfeld, M., and Y. Tsuji, "Multiphase Flows with Droplets and Particles," CRC Press 1998.
- Dipprey, D. F., and R. H. Sabersky, "Heat and Momentum Transfer in Smooth and Rough Tubes at Various Prandtl Numbers," Int. J. Heat and Mass Transfer vol. 6, pp. 329-353, 1963.
- Frost, W., Chang, H., Shieh, C. and K. Kimble, "Two Dimensional Particle Trajectory Computer Program," Interim Report for Contract NAS3-224488, 1982.
- Gent, R.W., and Cansdale, J.T.: "One-Dimensional Treatment of Thermal Transients in Electrically De-iced Helicopter Rotor Blades," Technical Report 80159, Royal Aircraft Establishment, Procurement Executive, Ministry of Defense, Farnborough, Hunts, England, 1980.
- Hansman, R. J., A. Reehorst, and J. Sims, "Analysis of Surface Roughness Generation in Aircraft Ice Accretion," AIAA-92-0298, Jan. 1992.
- 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), Pergmon Press, Elmsford, New York, 1967.

Gear, C.W. "The Automatic Integration of Ordinary Differential Equations." *Comm. ACM* 14, 1971, pp. 176-179.

Hobbs, P.V., "Ice Physics," Oxford University Press, Ely House, London, 1974.

Hollworth, B. R., and R. D. Berry, "Heat Transfer from Arrays of Impinging Jets with Large Jet-to-Jet Spacing," *J. of Heat Transfer*, v. 100, pp. 352-7, May 1978.

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

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

Kim, J. J., "Computational Particle Trajectory Analysis on a 3-Dimensional Engine Inlet," AIAA 85-0411, Jan. 1985.

Macklin, W. C., and G. S. Payne, "A Theoretical Study of the Ice Accretion Process," *Quarterly Journal of the Royal Meteorological Society*, 1967, Vol. 93, pp. 195-213.

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.

Miller, D., et. al, "Validation of NASA Thermal Ice Protection Computer Codes, Part 1: Program Overview," AIAA 97-0049, Jan. 1997

Mills, A. F., "Heat and Mass Transfer," ISBN 0-256-11443-9, 1995.

Olsen, W., Shaw, R., and Newton, J. "Ice Shapes and the Resulting Drag Increase for an NACA 0012 Airfoil." NASA Technical Memorandum 83556, 1984.

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

Olsen, W., "Close-up Movies of the Icing Process on the Leading Edge of an Airfoil." NASA Lewis Research Center Movie C-313, 1985.

Perry, R. H., Green, D. W., and J. O. Maloney, "Perry's Chemical Engineer's Handbook," McGraw-Hill Book Company, 1984.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and B. P. Flannery, "Numerical Recipes in Fortran - The Art of Scientific Computing," Second Edition, Cambridge University Press, New York, 1992.

Reehorst, A. L., Ratvasky, T. P. and J. Sims, "Close-Up Analysis of Inflight Ice Accretion," AIAA 94-0804, 1994.

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

Roache, P. J., "Verification and Validation in Computational Science and Engineering," ISBN 0-913478-08-3, 1998.

Scavuzzo, R. J. and Chu, M. L., "Structural Properties of Impact Ices Accreted on Aircraft Structures," NASA CR 179580, 1987.

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

Schneider, G. E. and Zedan, M. "A Modified Strongly Implicit Procedure for the Numerical Solution of Field Problems," Numer. Heat Transfer, vol. 4, 1981, pp. 1-19.

Schneider, G. E. and Raw, M. J. , "An Implicit Solution Procedure for Finite Difference Modeling of the Stefan Problem," AIAA J., vol. 22, Nov. 1984, pp. 1685-1690.

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

Shin, J., "Characteristics of Surface Roughness Associated With Leading Edge Ice Accretion," NASA TM106459 (AIAA-94-0799), Jan. 1994.

Steger, J.L., "Implicit Finite Difference of Flow About Arbitrary Geometries with Application to Airfoils," AIAA-77-665, 1977.

Tribus, M.V., et al. "Analysis of Heat Transfer Over a Small Cylinder in Icing Conditions on Mount Washington." American Society of Mechanical Engineers Transactions, Vol. 70, 1949, pp. 871-876.

Von Doenhoff, A. E. and E. A. Horton, "Low-Speed Experimental Investigation of the Effect of Sandpaper Type Roughness on Boundary-Layer Transition," NACA TN 3858, 1956.

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

Subject Index

A

adhesion, 142, 173, 402
AIRMD, 90, 91, 113, 123, 172
AIRUP, 90, 91, 113, 123
AK, 75, 76, 105
AKX, 78, 79
AL, 75, 76
ALP, 75, 76
ALPX, 78, 79
ALX, 78, 79
angle, 69, 70, 149, 150, 154, 354, 355
angle of attack, 45, 129
anisotropy, 75, 76, 77, 78, 80, 81, 120, 121, 416
ANTICE, 240
ANTICON, 159, 304
ANTICON2, 159, 291, 305
anti-ice, 1, 2, 9, 13, 14, 18, 19, 20, 21, 36, 37, 52, 54, 55, 57, 77, 86, 87, 88, 90, 91, 94, 97, 98, 100, 102, 105, 106, 108, 126, 127, 141, 169, 171, 172, 175, 226, 227, 240, 265, 276, 277, 287, 292, 303, 305, 340, 389, 399, 408, 413
AOA, 45, 157
array size, 31, 170
ASCII, 27, 73, 116, 151, 331
automatic, 9, 12, 29, 30, 311, 341

B

benchmark, 7, 194, 352
BETA.DAT, 15, 136, 159, 304
binary, 128, 158, 310, 332
bleed air, 1, 90, 98, 99, 100, 105, 110, 112, 132, 145, 172, 227, 240, 287, 288, 289, 290, 291, 292, 304, 389, 390, 391, 392, 393, 399, 412
body, 16, 19, 20, 21, 24, 25, 36, 38, 63, 69, 71, 73, 74, 79, 80, 90, 93, 97, 100, 101, 107, 120, 126, 127, 128, 130, 131, 132, 133, 134, 149, 216, 291, 311
boot, 2, 9, 21, 40, 41, 59, 222, 223, 224, 225
boundary condition, 20, 73, 86, 87, 88, 89, 101, 102, 103, 105, 122, 141, 227, 401, 411, 413
boundary layer, 39, 40, 101, 241, 304, 373, 374, 375, 377, 380, 381, 387
breakup, 2, 208, 213, 349, 351, 352

C

CD-ROM, 7, 133, 147, 151, 291, 311, 312, 325, 332
chord, 44, 53, 54, 55, 56, 57, 58, 59, 62, 63, 71, 82, 90, 111, 126, 127, 128, 130, 131, 132, 142, 145, 149
CLT, 157
collection efficiency, 15, 35, 38, 39, 50, 54, 130, 133, 136, 141, 179, 182, 186, 190, 194, 198, 203, 208, 213, 215, 222, 226, 250, 255, 266, 277, 287, 289, 291, 292, 302, 304, 305, 309, 311,

325, 355, 359, 361, 383, 419
computation time, 18, 37, 151, 175, 179, 182, 186, 190, 194, 199, 203, 208, 213, 216, 222, 226,
233, 240, 254, 265, 276, 287, 289, 291, 292, 305, 311
computational domain, 18, 19, 73, 79, 82, 90
conduction, 2, 100, 101, 102, 103, 105, 138, 141, 240, 412
control volume, 2
convergence, 107, 171
CONVERT, 157
cycle time, 81, 83
cylinder, 194, 421

D

debug, 21, 116, 143, 144, 172
default, 7, 12, 19, 29, 33, 34, 35, 37, 39, 42, 44, 45, 49, 79, 87, 90, 98, 99, 100, 126, 127, 128,
130, 131, 132
de-ice, 1, 2, 9, 13, 18, 20, 21, 24, 25, 30, 36, 40, 41, 73, 77, 80, 81, 82, 83, 84, 85, 86, 87, 89,
100, 101, 102, 105, 106, 107, 108, 109, 110, 114, 120, 126, 127, 131, 142, 143, 144, 157, 158,
160, 171, 172, 175, 227, 228, 233, 234, 235, 239, 240, 241, 242, 254, 256, 265, 266, 267, 276,
277, 278, 287, 293, 303, 312, 326, 330, 341, 399, 401, 414
density, 12, 18, 75, 85, 86, 121, 305, 334, 348, 354, 361, 368, 376, 382, 387, 388, 399, 403, 407,
414, 420
design, 88, 158, 159, 287, 305, 332, 370
DHOLE, 9, 91, 93, 113, 123
disk space, 175, 179, 182, 186, 190, 194, 199, 203, 208, 213, 216, 222, 226, 233, 240, 254, 265,
276, 287, 289, 291, 292, 305, 311
distribution, 2, 7, 14, 17, 20, 21, 42, 70, 101, 103, 106, 108, 109, 131, 141, 147, 175, 208, 222,
289, 291, 300, 304, 305, 310, 325, 327, 337, 341, 346, 375, 379, 408, 415, 419
distribution disk, 7, 133, 152, 223, 304
DP, 9, 93, 95, 113, 123
drop, 1, 2, 9, 14, 15, 35, 43, 48, 130, 138, 141
DSMN, 13
DTAUF, 107, 109, 114, 123, 172
DTAUI, 21, 107, 108, 109, 114, 123, 172
DTAUM, 107, 108, 109, 114, 123, 172
DTUBE, 9, 91, 92, 93, 113, 123
DUMMY.INP, 143
DYICE.DAT, 305

E

ELDE, 20, 75, 76, 102, 105
electrothermal, 1, 9, 19, 78, 82, 86, 98, 99, 103, 105, 106, 142, 233, 240, 265, 276, 287, 289,
304, 340, 389, 390, 391, 392, 399
ELX, 78, 79
ELXY, 78, 79
energy balance, 20, 50, 98, 107, 138, 168, 169, 171, 240, 379
EPRT, 50, 138

error, 16, 17, 19, 23, 24, 25, 27, 31, 33, 34, 42, 43, 44, 45, 46, 47, 48, 49, 51, 52, 53, 55, 56, 58, 59, 63, 67, 69, 70, 71, 74, 77, 78, 79, 80, 82, 83, 84, 87, 88, 89, 90, 91, 94, 95, 97, 100, 103, 114, 126, 127, 128, 129, 130, 131, 132, 134, 136, 142, 147, 148, 149, 152, 154, 155, 156, 157, 158, 165, 166, 167, 168, 169, 170, 171, 172, 331, 332, 333, 334, 339, 369, 403, 407, 414
Euler, 303, 334
evaporative, 14, 105, 106, 138, 141, 169, 226, 227, 232, 233, 276, 287, 379, 381, 389, 394
example, 9, 13, 14, 19, 20, 23, 27, 31, 32, 34, 41, 42, 43, 44, 47, 49, 51, 58, 63, 68, 70, 74, 76, 78, 79, 80, 81, 83, 85, 89, 91, 100, 103, 105, 106, 107, 109, 110, 111, 112, 113, 115, 116, 118, 120, 127, 133, 143, 144, 148, 150, 154, 159, 170, 254, 302, 304, 331, 332
example case, 6, 16, 21, 23, 60, 73, 129, 166, 175, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 201, 202, 203, 205, 207, 208, 210, 212, 213, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 232, 233, 234, 235, 239, 240, 241, 242, 248, 249, 250, 251, 252, 253, 254, 255, 256, 261, 262, 263, 264, 265, 266, 267, 272, 273, 274, 275, 276, 277, 278, 283, 284, 285, 286, 287, 289, 291, 292, 293, 299, 300, 301, 305, 306, 307, 308, 309, 311, 312, 313, 323, 324, 325, 328, 332
exceedence, 15, 43, 46, 199
EXPAND, 155
experiment, 1, 15, 20, 31, 33, 35, 43, 44, 45, 47, 62, 93, 94, 100, 101, 128, 134, 147, 148, 150, 151, 152, 175, 183, 186, 199, 213, 216, 227, 241, 265, 276, 287, 333, 334, 340, 371, 403, 420, 421
external heat flux, 40, 56, 131

F

file name, 5, 18, 23, 24, 25, 63, 126, 127, 128, 130, 131, 133, 145, 147, 148, 151, 155, 156, 157, 158, 159, 227, 234, 326, 327, 331, 332
flow, 1, 17, 20, 32, 33, 44, 45, 49, 68, 69, 70, 97, 128, 129, 130, 135, 137, 165, 166, 167, 168, 175, 216, 287, 291, 292, 303, 304, 305, 306, 309, 310, 311, 314, 315, 316, 317, 318, 319, 320, 321, 322, 325, 326, 327, 328, 330, 333, 346, 349, 365, 367, 370, 373, 375, 377, 386, 393, 403, 407, 414, 419, 421
FLOW.DAT, 159, 304
FLUENT, 303
FLWC, 141
FORCE.DAT, 142
FORCE2.DAT, 142
FORT.2, 159, 304, 305
FORT.26, 159, 304, 305
FORT.54, 144
FORT.80, 144
FORT.81, 144
FPRT, 49
freezing fraction, 135, 141, 371, 383, 393
freezing rain, 222

G

geometry, 2, 11, 14, 15, 16, 17, 18, 24, 30, 63, 73, 223, 303, 310, 311, 312, 325, 327
grid, 12, 18, 20, 30, 35, 38, 40, 73, 75, 77, 78, 102, 128, 129, 130, 142, 144, 158, 159, 166, 172,

303, 305, 306, 309, 310, 311, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 325, 326, 327,
329, 330, 333, 341, 365, 367, 373, 377, 399, 403, 407
GRIDGEN, 303

H

H1, 19, 88, 89, 90, 101, 122
H2, 88, 89, 122
H3, 88, 89, 122
H4, 88, 89, 122
HCONST, 9, 93, 94, 95, 113, 123
heat flux, 1, 14, 18, 19, 20, 37, 39, 40, 57, 83, 85, 88, 89, 94, 98, 99, 100, 105, 106, 111, 112,
122, 127, 131, 132, 142, 143, 240, 265, 276, 291, 304, 389, 390, 392, 401, 402, 411, 412
heat transfer coefficient, 2, 10, 18, 19, 20, 36, 37, 39, 40, 55, 83, 88, 89, 90, 94, 95, 97, 98, 101,
106, 110, 112, 122, 126, 131, 141, 142, 145, 178, 241, 249, 254, 261, 265, 266, 276, 277, 287,
288, 289, 290, 291, 292, 299, 301, 304, 305, 311, 329, 373, 375, 376, 377, 380, 391, 393, 394,
401, 402, 418
heater, 1, 2, 19, 20, 25, 36, 37, 73, 74, 76, 78, 79, 80, 81, 82, 83, 84, 85, 86, 99, 100, 101, 103,
109, 116, 120, 121, 122, 142, 143, 234, 239, 240, 241, 248, 249, 251, 252, 253, 254, 262, 263,
264, 265, 266, 272, 273, 274, 275, 276, 277, 283, 284, 285, 286, 389, 390, 391, 399, 413, 417
HLINP, 73, 98, 110, 126
hot air, 9, 19, 82, 83, 87, 88, 90, 98, 111, 114, 123, 124, 126, 127
HRES, 21, 59
HTC.DAT, 159, 304

I

I3D, 73, 97, 98, 100, 112, 113, 123, 128, 159
IBC1, 87, 88, 89, 122
IBC2, 20, 87, 89, 101, 122
IBC3, 87, 89, 122
IBC4, 87, 89, 122
IBETA, 38, 39, 54, 73, 130, 159, 305
IBLEED, 20, 73, 94, 97, 98, 99, 100, 106, 110, 113, 123, 126, 305
IBOOT, 40, 59, 222
IBOUND, 19, 20, 88, 100, 101, 102, 103, 105, 113, 123, 413
ICASE, 9, 113, 123
ice accretion, 1, 13, 18, 29, 34, 37, 100, 101, 102, 103, 199, 216, 240, 340, 419, 420, 421
ice growth, 108, 145
ice shape, 2, 9, 13, 35, 36, 300, 371, 404, 420
ice thickness, 18, 20, 77, 102, 301, 344, 345, 346
ICE1, 9
ice1.dat, 17, 40
ICEG2D, 2, 12, 18, 21, 30, 36, 51, 311, 312, 313, 325, 326, 328, 329, 330
ICFLAG, 83, 85, 86, 121
icing, 18, 30, 77, 108
ICOND, 20, 100, 101, 102, 103, 105, 113, 123, 413
ICORR, 91, 93, 94, 95, 106

ICP, 38, 52, 73, 130, 159, 305
IDEICE, 13, 18, 37, 105, 106, 141, 142, 143, 171, 254
IEVAP, 13, 14, 105, 106, 113, 123, 141, 265
IFLO, 29, 172
IGDE, 20, 100, 102, 103, 105, 113, 123
IGRID, 12, 30, 38, 141, 142
IHTC, 39, 40, 55, 73, 101, 131, 159, 305
IJDE, 82, 83, 85, 86, 98, 103, 105, 121
ILAYER, 116, 117, 118
ILINER, 10, 93, 96, 97
IMP.DAT, 133
impingement, 213
INIT, 20, 100, 101, 103, 105, 113, 123
input file, 5, 6, 9, 11, 12, 13, 14, 16, 17, 18, 19, 23, 24, 25, 28, 34, 36, 38, 52, 53, 55, 56, 58, 63,
67, 68, 73, 74, 77, 82, 87, 89, 90, 98, 99, 100, 105, 108, 109, 110, 111, 114, 120, 126, 127,
128, 129, 130, 131, 132, 133, 134, 135, 141, 142, 143, 147, 148, 155, 156, 157, 166, 167, 169,
172, 175, 179, 183, 186, 190, 194, 199, 203, 208, 213, 217, 223, 227, 228, 233, 234, 241, 242,
254, 256, 266, 267, 277, 278, 292, 293, 305, 310, 311, 331, 332, 333, 339, 409, 412, 416
intercycle, 21, 41
intercycle ice, 21
intercycle.dat, 40
interpolation, 155, 166, 170, 291
IOTYPE, 21, 116, 117, 118, 119, 124, 125, 143, 144, 160
IPAR, 19, 82, 83, 85, 86, 121
IQAIN, 20, 73, 94, 97, 98, 99, 100, 105, 106, 111, 113, 123, 127
IQEX, 39, 40, 56, 57, 73, 131
IQW, 20, 83, 94, 97, 98, 99, 100, 113, 123, 143
IRIX, 7, 23, 27, 28, 68, 147, 148, 155, 156, 157, 158, 159, 331
IROT, 20, 107, 113, 123
IRT, 47, 199
ISCOLA, 110, 111, 114, 123
ISCOLB, 54
ISCOLC, 52, 53
ISCOLH, 55, 56
ISCOLQ, 111, 112, 114, 124
ISCOLS, 112, 113, 114, 124
ISCOLT, 10, 112, 114, 124
ISCOLX, 56, 57
ISH, 100, 101, 102, 103, 105, 113, 123, 142
ISTD, 100, 101, 102, 103, 105, 113, 123
ITHERM, 50, 105, 106, 113, 123, 145
ITIME, 115, 124, 144
ITIMFL, 29, 162, 194
ITSUR, 10, 97, 98, 99, 100, 112, 113, 123, 131
IVXY, 78, 79
IWHERE, 116, 117

J

JCOUNT, 109
JCOUNTD, 107, 109, 114, 123, 172
JLAYER, 116, 117, 118
JSCOLA, 110, 111, 114, 123
JSCOLB, 54
JSCOLC, 52, 53
JSCOLH, 55, 56
JSCOLQ, 111, 112, 114, 124
JSCOLS, 112, 113, 114, 124
JSCOLT, 10, 112, 114, 124
JSCOLX, 56, 57
junk.dat, 16, 25, 171
JWHERE, 116, 117, 118

K

KSCOLA, 110, 111, 114, 123
KSCOLB, 54
KSCOLC, 52, 53
KSCOLH, 55, 56
KSCOLQ, 111, 112, 114, 124
KSCOLS, 112, 113, 114, 124
KSCOLT, 10, 112, 114, 124
KSCOLX, 56, 57
KWARN, 16, 21, 25, 51, 329

L

LAG time, 83, 84
laminar, 18, 36, 37, 254, 265, 276
layer, 1, 18, 20, 21, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 98, 102, 103, 108, 116, 117, 118, 119, 120, 121, 124, 125, 172, 390, 391, 402, 403, 411, 416, 417
LDE, 74, 75, 80, 103, 105, 117, 120
leading edge, 15, 18, 19, 37, 58, 59, 79, 90, 111
LewCon, 160
LEWI3DGR, 54, 55, 303, 304, 305
LEWICE/Thermal, 21, 240
LEWICE3D, 159, 291, 292
LIMIT.DAT, 334
liner, 93, 96, 97

M

mass balance, 135, 137
mass diffusivity, 383
MPRT, 50
MSIP, 109

N

namelist, 9, 58
Navier-Stokes, 2, 9, 10, 36, 52, 55, 57, 158, 159, 303, 304, 311, 330
NHOLE, 9, 91, 92, 93, 95, 113, 123
NISP, 107, 108, 109, 114, 123
NMSP, 107, 108, 109, 114, 123
NODE, 75, 76, 79, 80, 102, 108, 109, 116, 117, 120, 129, 144, 172, 173
NOICE.DAT, 18
normal direction, 117, 143
note, 9, 21, 30, 32, 35, 36, 41, 75, 76, 77, 79, 81, 82, 83, 85, 86, 87, 88, 89, 98, 99, 100, 101, 102, 103, 107, 108, 112, 114, 115, 116, 127, 131, 134, 135, 142, 143, 144, 156, 161, 162, 223, 234, 241, 254, 255, 266, 277, 287, 289, 292, 304, 312
NOX, 78, 79
NPARC, 311, 328
NPRF, 114, 115, 124, 145
NPRT, 114, 115, 124
NPTSD, 116, 117, 118, 119, 124, 125
NTYPE, 115, 116, 118, 124
NXDE, 74, 78, 80, 83, 117, 118, 120

O

OFF temperature, 19, 20, 83, 84, 85, 99, 417
OFF time, 83, 84, 417
OFFSET, 19, 78, 79, 82, 83, 85, 86, 121
ON temperature, 19, 20, 83, 84, 85, 99, 417
ON time, 83, 417

P

panel, 11, 12, 13, 16, 17, 35, 38, 50, 68, 69, 70, 135, 165, 203, 208, 306, 364
parting strip, 79, 81, 82, 85, 86, 121, 122, 240, 254
PC, 3, 5, 6, 7, 23, 24, 27, 115, 148, 155, 310, 331, 332
PEAKS.DAT, 150
phase change, 20, 102, 103, 109, 144, 171, 172, 173, 413
physical domain, 18, 19, 73, 79
physical property, 25, 73
piccolo, 2, 20, 91, 93, 94, 95, 96, 329
piccolo tube, 10
piccolo1.dat, 10, 50, 145
piccolo2.dat, 10, 50, 145
PLOT3D, 2, 115, 158, 159, 305, 333
PLOT3D.Q, 305
PLOT3D.XYZ, 305
potential flow, 2, 11, 20, 35, 36, 38, 128, 135, 216, 303, 306, 311, 334, 363, 364, 367, 369, 419
pressure, 15, 16, 47, 130, 137, 333, 363, 365, 367, 368, 381, 415
pressure coefficient, 38, 52, 130, 142, 182, 189, 193, 198, 207, 212, 216, 222, 226, 227, 250, 255, 266, 277, 287, 289, 291, 292, 302, 309, 311, 324, 333, 363, 365, 367, 369

Q

Q.PLT, 35, 128, 158
Q.TXT, 158, 159
QAIN.INP, 73, 98, 112, 127
QDE, 19, 83, 85, 86, 121
QEXTIN.INP, 40, 58, 73, 131
QOPT.DAT, 19, 20, 85, 99, 143
QW1, 19, 88, 89, 123
QW2, 88, 89, 123
QW3, 88, 89, 123
QW4, 88, 89, 123

R

RBETA.INP, 38, 55, 73, 130
RDATA, 9, 52, 58, 332
RDIST, 9, 91, 92, 93, 95, 113, 123
REDUCE, 160, 241
re-impinge, 208
relative humidity, 16, 48, 415
REP, 9, 93, 95, 113, 123
request, 337
residual, 21, 41, 59, 99, 223
Reynolds number, 129, 135, 217, 348, 354, 361, 373, 374, 377
RFLOW.INP, 38, 53, 73, 130
RHTC.INP, 39, 56, 73, 131
ridge, 223
rivulet, 291
rotational, 20, 107
roughness, 241, 371, 374, 377, 387, 419, 421
RP, 9, 93, 95, 113, 123
RPM, 20, 107, 113, 123, 403
RRP, 9, 93, 95, 113, 123
runback, 2, 98, 135, 137, 171, 241, 254, 340, 384, 385, 403, 419
running wet, 239, 265, 389
runtime error, 9, 165, 310

S

section, 2, 19, 74, 76, 78, 79, 80, 81, 82, 83, 84, 85, 86, 102, 116, 117, 118, 120, 121, 124, 126, 143, 241
SGI, 7, 23, 27, 155, 310, 311, 328, 332
shed, 2, 20, 21, 101, 102, 103, 107, 142, 173, 254, 385, 386, 402, 403, 413
SLD, 2, 9, 10, 14, 15, 35, 134, 141, 162, 334, 351, 352
slope, 53, 54, 56, 57, 75, 76, 78, 80, 81, 111, 120, 121, 417
SLOPEX, 78
SLOPX, 79
specific heat, 18, 75

splash, 2, 15, 35, 208, 213
SREF, 9
SSLOPA, 110, 111, 114, 123, 127
SSLOPB, 54
SSLOPC, 52, 53
SSLOPH, 55, 56
SSLOPQ, 111, 112, 114, 124
SSLOPS, 112, 113, 114, 124
SSLOPT, 10, 112, 114, 124
SSLOPX, 56, 57, 58
STA, 20, 107, 113, 123
stagnation, 38, 133, 134, 135, 136, 137, 138, 139, 141, 199, 333, 370, 371, 373, 374, 376, 377, 386
STREAM.DAT, 159, 304
STREAM.INP, 73, 98, 113, 127, 128
streamline, 52, 98, 112, 113, 114, 124, 127, 128, 159, 291, 304, 305, 414, 418
Sun, 7
supercooled large drop, 35
surface temperature, 145
system error, 23, 28, 67, 147, 155, 156, 157, 158, 159
SZEROA, 110, 111, 114, 123, 127
SZEROB, 54, 55
SZEROC, 52, 53
SZEROH, 55, 56
SZEROQ, 111, 112, 114, 124
SZEROS, 112, 113, 114, 124
SZEROT, 10, 112, 114, 124
SZEROX, 56, 58

T

TAFTER, 109, 110, 114, 123
TAIRH, 90, 91, 113, 123
technical support, 337, 339
TECPLOT, 2, 115, 145
TEMP.DAT, 305
temperature, 2, 16, 18, 19, 20, 21, 37, 40, 46, 47, 62, 75, 78, 81, 82, 83, 84, 86, 87, 88, 89, 90, 98, 99, 101, 103, 108, 109, 114, 116, 117, 118, 122, 131, 143, 144, 172, 213, 232, 233, 239, 240, 241, 251, 252, 253, 254, 255, 262, 263, 264, 265, 266, 272, 273, 274, 275, 276, 277, 283, 284, 285, 286, 287, 289, 291, 300, 305, 340, 368, 379, 380, 381, 383, 384, 389, 390, 391, 399, 400, 403, 409, 411, 413, 414, 415, 417, 420
TEMPS1.DAT, 143, 144, 241
TEMPS1A.DAT, 160
TEMPS2.DAT, 143, 144, 241
TEMPS2A.DAT, 160
TEMPS3.DAT, 143, 144, 241
TEMPS3A.DAT, 160

TEMPS4.DAT, 143, 144
TEMPS5.DAT, 143, 144
TEMPS6.DAT, 143
TEMPS7.DAT, 143
TG1, 88, 89, 122
TG2, 88, 89, 122
TG3, 88, 89, 122
TG4, 88, 89, 122
thermal, 1, 18, 20, 37, 77, 78, 82, 102, 234, 239, 240, 254, 265, 276, 287, 289, 292, 340, 375, 410, 414, 419, 420
thermal conductivity, 18, 75, 76, 77, 78, 80, 81, 83, 89, 98, 112, 120, 121, 131, 142, 376, 379, 382, 390, 399, 416
thermal diffusivity, 75, 76, 77, 78, 80, 81, 120, 121, 416
thermocouple, 116, 117, 143, 144, 160, 241, 254
THICK, 10
time step, 107, 108, 137
TINF, 88
TLAG, 83, 85, 86, 121
TOASCII, 159, 310
TOBINARY, 158, 310
TOFF, 19, 83, 85, 86, 121
TON, 83, 85, 86, 121
trailing edge, 19, 53, 55, 56, 58, 79, 91, 111, 126, 130, 131, 132
trajectory, 20, 21, 38, 39, 107, 130, 303
transient, 30, 172
TSPRAY, 109, 110, 114, 123
TSTART, 12, 30, 36
TSTOP, 19, 20, 30, 85, 99, 287
TSURF, 14, 105, 106, 113, 123
tsurf.inp, 99, 112, 131
turbulent, 241, 254, 265, 376, 377, 380

U

unix, 3, 7

V

validation, 16, 30, 35, 63
VSAERO, 303

W

warning, 20, 24, 51, 53, 54, 59, 87, 88, 90, 91, 92, 93, 94, 95, 98, 99, 100, 101, 102, 103, 106, 107, 108, 109, 110, 111, 112, 114, 115, 116, 117, 118, 126, 127, 128, 130, 131, 136, 157, 172
wattage, 81
WIND, 2, 52, 57, 311, 325, 326, 327, 328, 329, 330
wrap, 99

wrap direction, 73, 74, 75, 78, 101, 117, 120, 144, 173, 417
wrap distance, 19, 38, 39, 40, 52, 53, 54, 55, 56, 57, 58, 78, 79, 95, 98, 110, 111, 112, 116, 126,
127, 128, 130, 131, 132, 135, 143, 144, 149, 172, 173, 304, 344, 345, 359, 375, 377, 402, 418

X

XAIRMD, 90, 91, 113, 123
XBOOTLOW, 59
XBOOTUP, 59
XHOLE, 10, 95, 96, 113, 123
XLHND, 9, 91, 92, 113, 123
XLLINER, 10, 96, 97
XLLND, 10, 92, 96, 97
XLLST, 10, 97
XUHND, 9, 91, 92, 113, 123
XULINER, 10, 96, 97
XULND, 10, 92, 96, 97
XULST, 10, 96, 97
XY.PLT, 35, 128, 158
XY.TXT, 158, 159

Y

YCASE, 9, 91, 92, 93, 95, 113, 123
YHOLE, 10, 92, 95, 96, 113, 123

Z

ZDIST, 10, 95, 96, 113, 123
ZP, 9, 93, 95, 113, 123

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 01-11-2008		2. REPORT TYPE Final Contractor Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE User's Manual for LEWICE Version 3.2			5a. CONTRACT NUMBER NAS3-00145		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Wright, William			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER WBS 457280.02.07.03.02		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) QSS Group, Inc. 21000 Brookpark Road Cleveland, Ohio 44135			8. PERFORMING ORGANIZATION REPORT NUMBER E-15537		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001			10. SPONSORING/MONITORS ACRONYM(S) NASA		
			11. SPONSORING/MONITORING REPORT NUMBER NASA/CR-2008-214255		
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Categories: 01 and 02 Available electronically at http://gltrs.grc.nasa.gov This publication is available from the NASA Center for AeroSpace Information, 301-621-0390					
13. SUPPLEMENTARY NOTES					
14. 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 3.2 of this software, which is called LEWICE. This version differs from release 2.0 due to the addition of advanced thermal analysis capabilities for de-icing and anti-icing applications using electrothermal heaters or bleed air applications, the addition of automated Navier-Stokes analysis, an empirical model for supercooled large droplets (SLD) and a pneumatic boot option. An extensive effort was also undertaken to compare the results against the database of electrothermal results which have been generated in the NASA Glenn Icing Research Tunnel (IRT) as was performed for the validation effort for version 2.0. This report will primarily describe the features of the software related to the use of the program. Appendix A has been included to list some of the inner workings of the software or the physical models used. This information is also available in the form of several unpublished documents internal to NASA. This report is intended as a 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 software.					
15. SUBJECT TERMS Aircraft icing; Thermodynamics; Heat transfer					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 429	19a. NAME OF RESPONSIBLE PERSON STI Help Desk (email: help@sti.nasa.gov)
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code) 301-621-0390

