

AD-A126 484

COMPUTER MODELING OF TIME-DEPENDENT RIME ICING IN THE
ATMOSPHERE(U) COLD REGIONS RESEARCH AND ENGINEERING LAB
HANOVER NH E P LOZOWSKI ET AL. JAN 83 CRREL-83-2

1/1

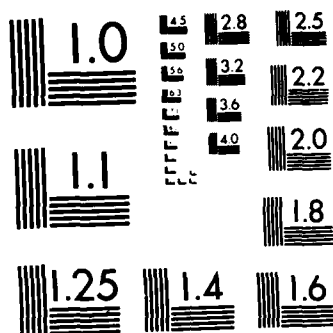
UNCLASSIFIED

F/G 8/12

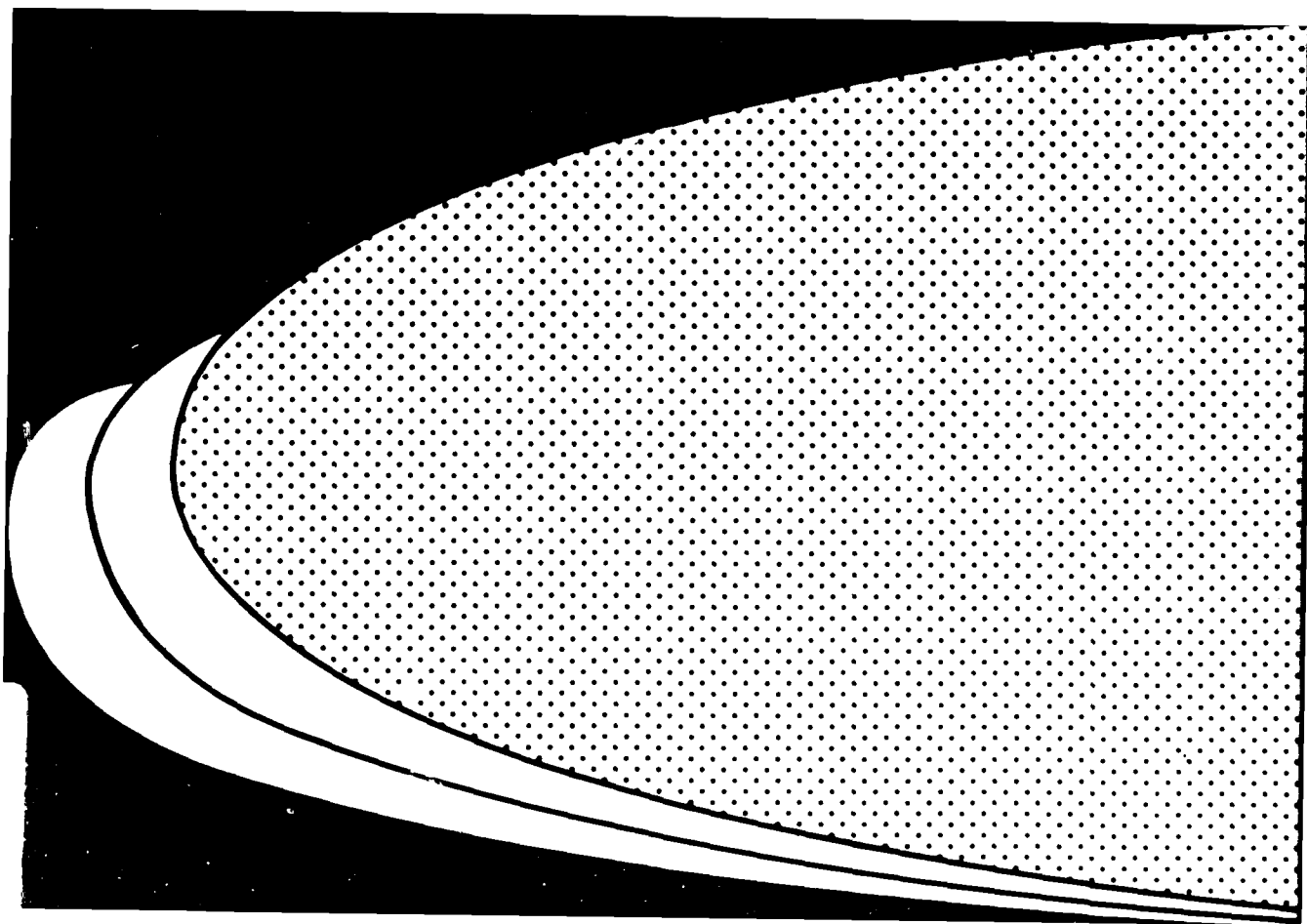
NL

END

FILED
JAN 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A





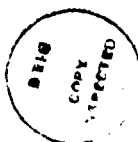
CRREL Report 83-2

January 1983

Computer modeling of time-dependent rime icing in the atmosphere

Edward P. Lozowski and Myron M. Oleskiw

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Approved for release by	
Dist. Special	
A	



Prepared for
HEADQUARTERS SPACE DIVISION
UNITED STATES AIR FORCE
Approved for public release. distribution unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CRREL Report 83-2	2. GOVT ACCESSION NO. AD-A126404	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER MODELING OF TIME-DEPENDENT RIME ICING IN THE ATMOSPHERE		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Edward P. Lozowski and Myron M. Oleskiw		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Alberta Edmonton, Alberta, Canada		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project 4A161102AT24, Task C/E1, Work Unit 002
11. CONTROLLING OFFICE NAME AND ADDRESS HQ Space Division, USAF Box 92960, Worldwide Postal Ctr. Los Angeles, California 90009		12. REPORT DATE January 1983
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) U.S. Army Cold Regions Research and Engineering Laboratory 72 Lyme Rd. Hanover, New Hampshire 03755		13. NUMBER OF PAGES 81
		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Airfoils Computerized simulation Ice formation Rime ice		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A numerical model of rime ice accretion on an arbitrary two-dimensional airfoil is presented. The physics of the model are described and results are presented that demonstrate, by comparison with other theoretical data and experimental data, that the model predictions are believable. Results are also presented that illustrate the capability of the model to handle time-dependent rime ice accretion, taking into account the feedback between the ice accretion and the airflow and droplet trajectory fields.		

DD FORM 1473

1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This report was prepared by Dr. Edward P. Lozowski, Professor of Meteorology, and Myron M. Oleskiw, Ph.D. Candidate in Meteorology, of the University of Alberta in Edmonton, Alberta, Canada. The project was contracted under DA Project 4A161102AT24, *Adhesion and Physics of Ice*, Task C/E1, Work Unit 002.

The report was technically reviewed by Michael G. Ferrick and Jacqueline A. Richter of CRREL. Several individuals have contributed directly or indirectly to the success of the research. In particular, the authors wish to thank J. Stallabrass for his advice, assistance and inspiration; Professor D.J. Marsden for his assistance with the potential flow calculations; S.F. Ackley, who monitored the work and saw that it was brought to fruition; T. Cansdale and M. Bragg, whose work along parallel lines has stimulated the authors and helped them to gain confidence in the validity of the present model; L. Smith, who persevered in typing through several versions of the manuscript; and G. Lester and his staff, who capably prepared the illustrations.

CONTENTS

	Page
Abstract	i
Preface	ii
Nomenclature	v
Introduction	1
Methodology	2
Potential flow around an arbitrary airfoil	2
Incompressible velocity field	3
Droplet trajectory equation	3
Computational procedure for trajectories	4
Determining the point of impact	5
Calculation of collision efficiencies	5
Accreting an ice layer	5
Determining the accuracy of the flow field	6
Determining the accuracy of the trajectories	6
Results and discussion	11
Comparing results with and without the history term	11
Collision efficiency of NACA 0015 airfoil at 8° attack angle	11
Time-dependent accretion on NACA 0015 airfoil at 8° attack angle	11
Time-dependent accretion on NACA 0015 airfoil at 0° attack angle	16
Conclusions and recommendations	19
Literature cited	20
Appendix A: Sample input	21
Appendix B: Sample output	23
Appendix C: Program listing	31

ILLUSTRATIONS

Figure	
1. Trajectories about a cylinder	8
2. Collision efficiency vs length along cylinder surface	9
3. Collision efficiency vs distance along chord	10
4. Trajectories about a NACA 0015 airfoil at 8° attack angle. $C = 16.9$ cm; $V_\infty = 30.5 \text{ m s}^{-1}$; $r_d = 10 \mu\text{m}$	12
5. Collision efficiency vs length along airfoil surface. $C = 16.9$ cm; $V_\infty = 30.5 \text{ m s}^{-1}$; $r_d = 10 \mu\text{m}$	12
6. Trajectories about a NACA 0015 airfoil at 8° attack angle. $C = 21.3$ cm; $V_\infty = 61 \text{ m s}^{-1}$; $r_d = 10 \mu\text{m}$	13
7. Collision efficiency vs length along airfoil surface. $C = 21.3$ cm; $V_\infty = 61 \text{ m s}^{-1}$; $r_d = 10 \mu\text{m}$	14
8. Accretion after 2.5 min and 5 min on NACA 0015 airfoil at 8° attack angle	15
9. Accretion after 5 min in a single step, two steps and experimentally	15

Figure	Page
10. Trajectories about a NACA 0015 airfoil at 0° attack angle.	16
11. Collision efficiency vs length along airfoil surface. $C = 21.3$ cm; V_∞ $= 91.5$ m s $^{-1}$; $r_d = 10$ μ m; $LWC = 0.4$ g m $^{-3}$	17
12. Accretion after 1 through 5 min on a NACA 0015 airfoil at 0° attack angle.	18
13. Accretion after 5 min in a single step, five steps and experimentally. . . .	18

TABLES

Table	
1. Icing on a cylinder, present calculations vs Langmuir and Blodgett. . . .	7
2. Collision efficiencies as a function of time for the case of Figure 10. . . .	17

NOMENCLATURE

C	cylinder diameter (m)
C_i	the control point on the i th line segment (control element), approximating the airfoil surface
C_D	drag coefficient (dimensionless)
E	total collision efficiency (%)
E_m	maximum local collision efficiency (%)
$\bar{g}$	acceleration due to gravity (m s^{-2})
K	Langmuir inertia parameter (dimensionless)
L	nondimensional distance along the surface of the accretion, starting at the nose (dimensionless)
N	number of line segments (control elements) approximating the airfoil surface
P	any point in the airstream
P	air pressure (Pa)
r	distance between a point on a control element and any point in the airstream (m)
$r(L)$	local radius of curvature of the accretion or substrate at distance L from the nose (m)
r_d	droplet radius (m)
$R(L)$	icing flux at distance L from the nose ($\text{kg m}^{-2} \text{s}^{-1}$)
Re	Reynolds number of the droplet (dimensionless)
S_j	any point on the j th control element
T	air temperature (K)
t	time (s)
t_A	total accretion time for a layer (s)
u	x component of airspeed (m s^{-1})
v	y component of airspeed (m s^{-1})
v_x	x component of droplet impact speed (m s^{-1})
v_y	y component of droplet impact speed (m s^{-1})
V_∞	freestream airspeed (m s^{-1})
$\bar{V}_a$	vector air velocity (m s^{-1})
$\bar{V}_d$	vector droplet velocity (m s^{-1})
w	liquid water content of cloud (kg m^{-3})
x	x -coordinate (m)
X	nondimensional x -coordinate = x/C (dimensionless)
$\bar{X}_d$	nondimensional droplet position vector (dimensionless)
y	y -coordinate (m)
Y	nondimensional y -coordinate = y/C (dimensionless)

α	angle of attack of airfoil chord relative to freestream direction (radians)
β	local collision efficiency (%)
β_0	maximum local collision efficiency (%)
γ	vorticity density along a control element ($\text{s}^{-1} \text{m}^{-1}$)
ϕ	nondimensional impingement parameter (dimensionless)
μ_a	dynamic viscosity of airstream ($\text{kg m}^{-1} \text{s}^{-1}$)
ν_a	kinematic viscosity of airstream ($\text{m}^2 \text{s}^{-1}$)
ρ_a	density of airstream (kg m^{-3})
ρ_d	density of a water droplet (kg m^{-3})
ρ_i	density of accreted ice (kg m^{-3})
θ_m	maximum angle of impingement on cylinder (i.e. maximum accretion extent) (radians)
τ	time (s)
ψ	stream function ($\text{m}^2 \text{s}^{-1}$)

COMPUTER MODELING OF TIME-DEPENDENT RIME ICING IN THE ATMOSPHERE

Edward P. Lozowski and Myron M. Oleskiw

INTRODUCTION

The literature on the subject of icing is very extensive, and we do not intend to review it here. Instead, we will mention simply that the present work arose chiefly as a result of two earlier investigations into icing, one at the U.S. Army Cold Regions Research and Engineering Laboratory and the other at the National Research Council of Canada in Ottawa. The first of these studies was reported by Ackley and Templeton (1979), while the second was described by Lozowski et al. (1979). Both were computer-simulated models of ice accretion on a cylinder. The first included time-dependent effects but ignored runback, and the second ignored time dependence but allowed for the thermodynamics of runback.

Although cylinder icing models are of intrinsic importance for understanding powerline icing, for example, their geometry is not appropriate for the study of airfoil icing. Airfoil icing has been a subject of renewed interest in recent years, in part because of a need to certify helicopters and general aviation aircraft for flight in IFR (instrument flight rules) icing conditions. The limited power available on such aircraft and the new materials used in airfoil construction demand that deicing or anti-icing equipment be carefully designed for maximum efficiency. Although the design of such equipment requires wind-tunnel and ultimately field testing, computer simulation models are considered to be an important tool in the design process (Rosen and Potash 1981).

The objective of the present work is therefore to develop and test a computer simulation model for airfoil icing. This report describes a model that permits simulation of the time-dependent growth of ice without runback on an arbitrary, two-dimensional airfoil. In developing the model, a great deal of effort has gone into carefully specifying the assumptions made and into testing the individual components of the model. We are confident that within the framework of assumptions of the model, the icing accretions that it predicts are believable. Because of the effort required to develop and test the present model, it has not been possible in the time available to make the model completely general. Consequently, we have not, for example, incorporated accretion thermodynamics into the model nor taken into account rotation effects, such as could be found on helicopter rotor blades. These are developments for the future. Nevertheless, the model as it stands should be very useful for estimating the icing rate and shape on airfoils when the accretion is dry (i.e. no runback) and when rotation effects (or any other three-dimensional effects) may be ignored. During the course of this work, two opportunities arose to make presentations of the progress to date to audiences of cloud physicists and aerodynamicists. These presentations are summarized in Oleskiw and Lozowski (1980) and Lozowski and Oleskiw (1981).

METHODOLOGY

The modeling of airfoil icing may be separated into two distinct aspects. The first is the impingement of supercooled water droplets on the airfoil surface, and the second is the mechanics and thermodynamics of the resulting accretion. The present study deals exclusively with the first aspect. This is sufficient for the investigation of the dry accretion process, in which the heat transfer is great enough that all of the impinging water droplets freeze at their point of impact. This restriction is analogous to that made by Ackley and Templeton (1979) in their model of icing cylinders. Cansdale and McNaughtan (1977) and Lozowski et al. (1979) also considered the case of wet accretion on cylinders, in which some impacting water remains unfrozen and is blown back along the icing surface. The extension of these wet accretion models to airfoils requires an ability to calculate the heat transfer of iced airfoils. We have not had the time or funds to do this, either theoretically or by experiment, under the present contract.

The computer algorithm for simulating the dry accretion process may be broken down into the following steps:

1. Determining the potential flow stream function field around an arbitrary two-dimensional airfoil in crossflow.
2. Determining the incompressible velocity field around the airfoil.
3. Calculating droplet trajectories and points of impact.
4. Determining the airfoil collision efficiency as a function of surface position for specified values of freestream airspeed, droplet size, and airfoil angle of attack.
5. Calculating the spatial distribution of icing during a short time interval, under the dry accretion assumption.
6. Determining the accretion shape and mass.
7. Calculating the new airfoil shape as modified by the ice accretion.
8. Repeating steps one to seven as often as desired to obtain the growth of the accretion as a function of time.

In the detailed descriptions that follow, we deal with each of these steps in turn.

Potential flow around an arbitrary airfoil

There are numerous potential flow codes available that permit the determination of the stream function field around an arbitrary two-dimensional airfoil. These can be broadly classified into two groups: a) conformal transformation techniques, e.g. Theodorson and Garrick (1932), and b) surface singularity methods, e.g. Hess and Smith (1967). The particular technique chosen to address the icing problem should have the following characteristics. First, it should be particularly efficient in terms of computer time, because of the large number of air velocity calculations required to determine droplet trajectories. Secondly, it should be capable of handling the changes to the airfoil profile due to the ice accretion. In this latter connection, the computer code must be capable of accepting a specification of the airfoil in terms of surface coordinates and, moreover, it should not be too sensitive to small errors in the specified airfoil coordinates.

In keeping with these considerations, we chose the method described by Kennedy and Marsden (1976). This is one of the so-called "surface singularity" or "panel" methods. It is thought to be the simplest available and provides exceptional accuracy for little computing effort. For the purpose of calculating the potential flow, the airfoil surface is approximated by N straight-line segments or "panels," labeled S_j , $j = 1, 2, \dots, N$. A constant, but unknown, vorticity density $\gamma(S_j)$ is distributed along each panel or control element. If this airfoil model is immersed in a uniform stream of unit velocity (nondimensionalized), at an angle of attack α , the stream function at any point external to the airfoil $P(x,y)$ is, according to elementary potential flow theory, given by:

$$\psi(x,y) = y \cos \alpha - x \sin \alpha - \frac{1}{2\pi} \sum_{j=1}^N \int_{S_j} \gamma(S_j) \ln r(P, S_j) dS_j \quad (1)$$

where $r(P, S_j)$ is the distance between point P and any point on the element S_j .

To solve for the unknown $\gamma(S_j)$ for each panel, eq 1 is applied at a control point, C_i , $i = 1, 2, \dots, N$, on each panel. Imposing the boundary condition, that the stream function be a constant along the airfoil surface (i.e. at each control point), and the Kutta condition, that the surface streamline leave the trailing edge smoothly, leads to a set of linear algebraic equations for the unknown vorticity densities. These matrix equations are solved in the usual way, and eq 1 then allows the determination of the stream function anywhere in the potential flow. The airspeed at any point can then be determined by differentiation of eq 1.

Other investigators (Bragg and Gregorek 1981) have adopted the conformal transformation approach, while still others (McComber and Touzot 1981) have solved Poisson's equation for the stream function using finite element methods. We believe, however, that the present method yields greater accuracy and spatial resolution for a similar computing effort.

Incompressible velocity field

The air velocity components may be calculated at any desired point in the airstream by differentiating the stream function; that is, by approximating the equations:

$$u = + \frac{\partial \psi}{\partial y} \quad v = - \frac{\partial \psi}{\partial x} \quad (2)$$

with finite differences. This is done with a space increment, Δx or Δy , equal to the diameter of a cloud droplet. Thus it is possible to obtain a very accurate estimate of the airspeed at the position of the droplet's center of mass, wherever that happens to be. We believe that this approach is more accurate than that used by some other investigators (e.g. Cansdale 1980, private communication), who determine the airstream velocity field initially at a fixed array of grid points. When the air velocity at the droplet position is desired, an interpolation procedure among the grid point values is applied. Our approach of evaluating the air velocity as needed at precise points along the droplet trajectory, rather than by interpolation, is made possible by the economy with which ψ can be calculated using the Kennedy-Marsden approach.

Droplet trajectory equation

Pearcey and Hill (1956) have expressed the equation of motion of a spherical droplet of fixed mass in an accelerated air flow as:

$$\frac{d\bar{X}_d}{dt} = \bar{V}_d \quad (3)$$

$$\frac{d\bar{V}_d}{dt} = \frac{2(\rho_d - \rho_a)}{(2\rho_d + \rho_a)} \bar{g} \quad (\text{Buoyancy term})$$

$$- \frac{3C_D \rho_a}{4r_d(2\rho_d + \rho_a)} |\bar{V}_d - \bar{V}_a| (\bar{V}_d - \bar{V}_a) \quad (\text{Drag term})$$

$$- \frac{9\rho_a}{(2\rho_d + \rho_a)r_d} \sqrt{\frac{\rho_a}{\pi}} \int_{-\infty}^t \frac{d\bar{V}_d}{d\tau} \frac{d\tau}{\sqrt{t-\tau}} \quad (\text{History term}) \quad (4)$$

where $\bar{X}_d(x_d, y_d)$ = droplet position vector
 $\bar{V}_d(u_d, v_d)$ = droplet velocity vector
 $\bar{V}_a(u_a, v_a)$ = air velocity vector
 $\bar{g}$ = gravitational acceleration
 C_D = steady-state droplet drag coefficient
 ρ_a = air density
 ρ_d = droplet density
 r_d = droplet radius
 ν_a = kinematic viscosity of the air
 t = time.

The first term on the righthand side of eq 4 is the net buoyancy of the droplet in air. The second term is the steady drag, and the third is known as the history term (because of the time integral over the entire droplet history). The first two terms are probably in need of no explanation, although the gravitational term is frequently ignored in icing calculations (e.g. Langmuir and Blodgett 1946). The significance of the history term, however, may not be so apparent. It is essentially a correction to the drag term, which is necessary when the drag coefficient used in the second term is the steady-state value, appropriate for nonaccelerating droplets. For a given relative velocity between the droplet and the airstream, the true value of C_D is smaller for a drop that is accelerating with respect to the flow than for one that is not accelerating (i.e. one that is in equilibrium or steady state). This may be thought of as a phase lag effect, due to the finite rate of vorticity diffusion, which requires a certain time for the droplet to reach equilibrium with the airstream. Because of the large droplet acceleration that occurs in certain icing situations, we felt it important to examine the effects of the history term on the calculation of the droplet trajectories. Consequently, comparisons have been made between results calculated without the history term (referred to as the steady-state drag formulation) and those calculated with the history term included (referred to as the non-steady-state formulation). It should be noted that eq 4 also incorporates the effects of the droplet's induced mass resulting from the momentum it imparts to the air as it accelerates.

The formulation used to determine the steady-state drag coefficient as a function of droplet Reynolds number is given below:

1. $Re < 0.01$ $C_D = 24/Re_d$
2. $0.01 \leq Re \leq 5$ $C_D = 24/Re_d + 2.2$ (5)
3. $5 < Re < 5000$ $C_D = 0.2924(1 + 9.06 Re_d^{-0.5})^2$.

The droplet Reynolds number is defined by:

$$Re_d = \frac{2r_d\rho_a}{\mu_a} |\bar{V}_d - \bar{V}_a|$$

where $\bar{V}_d$ and $\bar{V}_a$ are respectively the droplet and air velocity vectors, and μ_a is the dynamic viscosity of the air. The second formulation is from Sartor and Abbott (1975), while the third is given by Abraham (1970).

Computational procedure for trajectories

Equations 3 and 4 in component form yield four equations that, for the steady-state formulation, are numerically integrated using a fourth-order Runge-Kutta-Fehlberg method (Lapidus and Seinfeld 1971, Burden et al. 1978). This procedure permits the time step to be

adjusted continuously for optimum speed of computation given a specified degree of accuracy required.

When the history term is incorporated into eq 4, it becomes a Volterra integro-differential equation of the second kind. The method of solution we used in this case is essentially the same as that used for the steady-state case, with the additional provision that the history term is approximated by a combined numerical and analytical technique. With this scheme, the integral is approximated by a finite sum at full time steps of the Runge-Kutta-Fehlberg method (for example, at τ and $\tau + \Delta\tau$). At intermediate time steps, however (between τ and $\tau + \Delta\tau$), the value of the integral is approximated by the extrapolation of a Legendre polynomial fitted to the previous values of the integral at full time steps.

Determining the point of impact

The droplet is assumed to have impinged upon the airfoil if any part of it contacts the airfoil surface. Thus, close to the airfoil, the finite size of the droplet is taken into account. This is particularly important for those trajectories just within the envelope of colliding trajectories, where the angle of incidence from the normal to the airfoil surface is close to 90° .

Calculation of collision efficiencies

To determine the local collision efficiency, β , at any point on the airfoil surface, use is made of the relation:

$$\beta(L) = \frac{dY}{dL} \cos \alpha$$

where Y is the ordinate at the starting point of a particular trajectory, L is the distance along the airfoil surface between the nose and the impact point of the same trajectory, and α is the angle of attack. By calculating the trajectories for between 10 and 20 droplets, Y may be plotted as a function of L , and the derivative taken to obtain β . These latter operations are in fact performed numerically using a cubic spline fit to the point values on the graph of Y vs L . The resulting local collision efficiencies may be plotted as a function of L (e.g. Fig. 2b) or of the corresponding abscissa X (e.g. Fig. 3a).

Accreting an ice layer

In the model, it is assumed that the ice growth on a particular small segment of the airfoil surface is oriented perpendicular to the surface. According to Lozowski et al. (1979), the accretion thickness is then given by the equation:

$$h(L) = \frac{2R(L)t_A}{\rho_i} \left/ \left(1 + \sqrt{1 + \frac{2R(L)t_A}{\rho_i r(L)}} \right) \right. \quad (7)$$

where

$$R(L) = V_\infty w \beta(L) \quad (8)$$

is the icing flux with V_∞ the freestream velocity and w the liquid water content of the airstream. t_A is the period of accretion, ρ_i the assumed ice density (890 kg m^{-3}), and r the radius of curvature of the airfoil surface.

In the results presented here, we assume that time interval t_A is sufficiently small that the second term under the root in the denominator may be ignored.

By plotting the accretion thickness as a function of distance along the airfoil surface from the nose, it is possible to determine a new airfoil surface shape after it has iced for the speci-

fied time interval. The entire procedure can now be repeated, using the new iced airfoil surface to determine a new stream function, new droplet trajectories, and ultimately a second accretion layer. By continuing in this manner, it is possible to build up a substantial ice accretion on the airfoil.

Determining the accuracy of the flow field

The accuracy of the Kennedy-Marsden technique was tested by comparing its predicted stream function for potential flow around a cylinder with the known analytic solution. Using 50 control elements, the error in ψ is at most 0.1% near the cylinder. It falls to below 0.01% at distances from the surface exceeding about four cylinder diameters.

We have also made a comparison between the corresponding velocity fields. In one such test for example, using a cylinder diameter of 0.15 m, an air pressure of 78.5 kPa, an air temperature of -10°C , and a freestream velocity of 114.3 m s^{-1} , the velocity field of the analytic solution was compared with that provided by the model using 38 control elements. At a distance of five diameters upstream, the air velocities differed by less than 10⁻³%. Very close to the cylinder they were as high as 1 to 2%. However, the effect of these airstream velocity errors on the computed droplet collision efficiencies was found to be much less than 1%.

Determining the accuracy of the trajectories

To establish some confidence in the trajectories themselves, it was decided to make a comparison with two cases considered by Langmuir and Blodgett (1946). The two cases were chosen to check our method of trajectory calculation for both high and low collision efficiencies. For both cases, Langmuir's ϕ parameter was chosen to be 10^4 . This is given by:

$$\phi = 9 \frac{\rho_a^2 C V_{\infty}}{\mu_a \rho_d} \quad (9)$$

where ρ_d = droplet density

ρ_a = air density

μ_a = dynamic viscosity of air

C = cylinder diameter

V_{∞} = freestream speed.

ϕ is a nondimensional impingement parameter. Large values of ϕ imply a large radius of curvature of the streamlines, and vice versa. Langmuir's K parameter was 36.0 in the first case and 1.0 in the second. K is given by the expression:

$$K = \frac{4 \rho_d r_d^2 V_{\infty}}{9 \mu_a C} \quad (10)$$

where r_d is the droplet radius. K is the nondimensional inertia parameter. It is the ratio of the droplet's projectile range under Stokes' law to the radius of the cylinder. If K is small, the droplets tend to follow the streamlines, and, hence, the collision efficiency tends to be low.

In these cases, as in all the experiments considered in this report, the droplets were introduced into the airstream five chord lengths upstream of the nose of the target with a Reynolds number Re_d of 0.001. Ideally, the droplet trajectory integration should begin infinitely far upstream with the droplets having the same velocity as the air (i.e. $\text{Re}_d = 0$), but for computational reasons this is impractical. Tests indicate that the trajectory errors caused by this imperfect initial condition are smaller than the numerical integration errors.

The parameters chosen for the two cases considered are given in Table I. Table I also presents a comparison between our results and those of Langmuir and Blodgett (1946). The

Table 1. Icing on a cylinder, present calculations (rows 2, 4, 5) vs Langmuir and Blodgett (rows 1, 3).

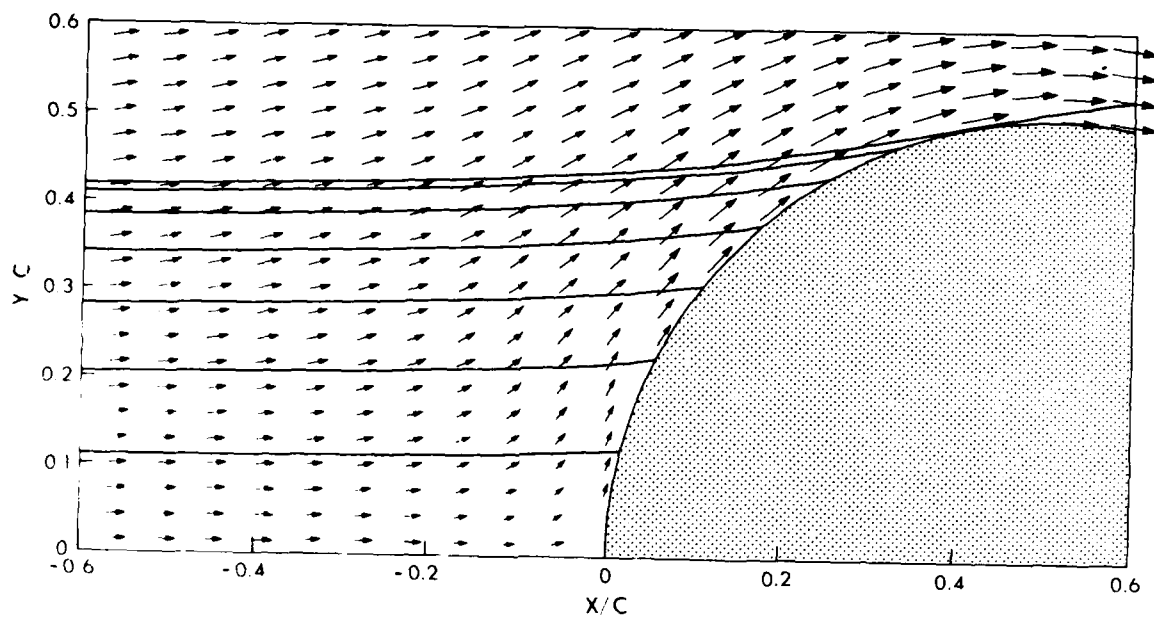
ϕ	K	T (°C)	P (kPa)	ρ_d (kg m ⁻³)	C (m)	V_∞ (m s ⁻¹)	r_d (μ m)	v_x	v_y	E_m (%)	β_0 (%)	θ_m (deg)	History term
10,000	36	—	—	—	—	—	—	1.056	0.193	81.9	88.5	79.8	No (L & B)
10,000	36	-10	78.5	999.15	0.15	114.3	42.1	1.056	0.196	81.4	89.8	79.5	No (Model)
10,000	1	—	—	—	—	—	—	0.494	0.725	15.6	34.8	34.2	No (L & B)
10,000	1	-10	78.5	999.15	0.15	114.3	7.02	0.477	0.650	17.0	37.6	35.6	No (Model)
10,000	1	-10	78.5	999.15	0.15	114.3	7.02	0.527	0.662	18.7	39.1	38.9	Yes (Model)

symbols v_x and v_y denote the droplet impact velocity components in the x - and y -directions, respectively. θ_m denotes the maximum angle of droplet impingement from the forward stagnation point. Our model results for v_x , v_y , E_m , β_0 and θ_m (given in rows 2 and 4) compare favorably with those of Langmuir and Blodgett (given in rows 1 and 3). The discrepancies, which are larger for the case with the smaller inertia parameter K , are quite acceptable, if one recognizes that the Langmuir and Blodgett data should not be viewed as an absolute standard.

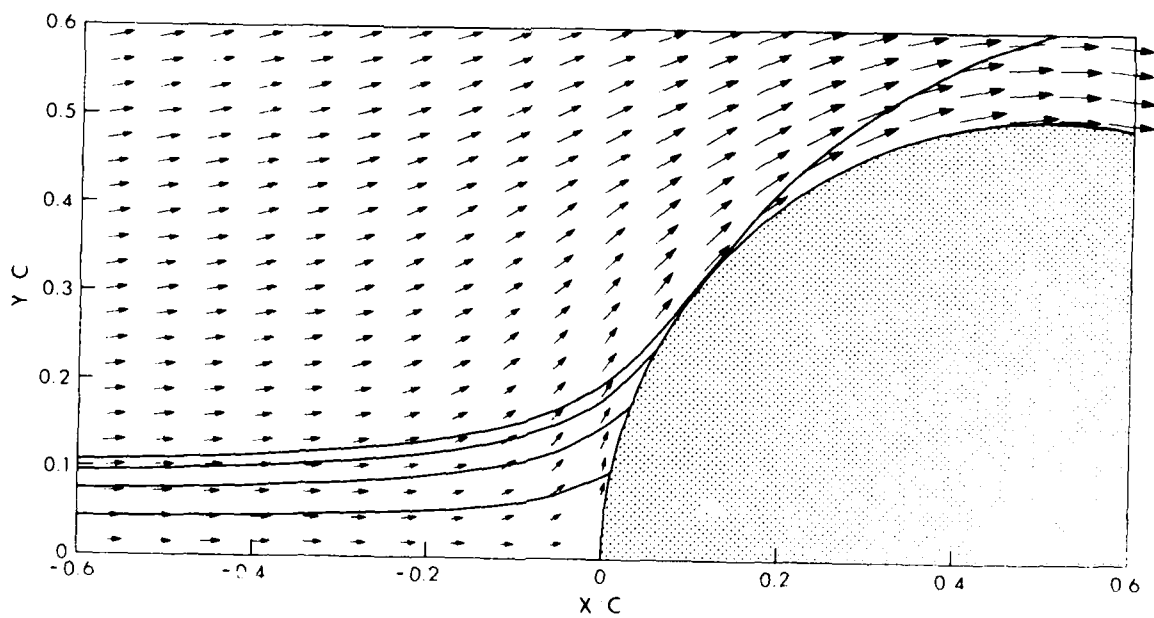
Figure 1 shows the flow field (indicated by velocity vectors) and the droplet trajectories (indicated by solid curves) for flow about the cylinder in the two cases just considered. It is interesting to note the much higher curvature of the trajectories for the less massive drops and the larger shadow zone between the grazing trajectory and the cylinder. One might also speculate on the collisions that could occur between the small and large drops, because of the way the smaller ones track across the trajectories of the large ones. It is hard to see, however, how such collisions might have any significant effect on the icing.

Figure 2 displays the collision efficiency for the two cases as a function of the nondimensional distances along the cylinder surface from the stagnation point (that is, actual distance L divided by cylinder diameter C). Negative values of L/C lie below the stagnation point, while positive values lie above it. Figure 2a is almost a cosine curve, a result that would occur if the droplet trajectories were straight lines. The slight "kinks" in Figure 2b are artificial and arise from the numerical spline fitting procedure. The overall collision efficiency is equal to the total area under the curves.

Figure 3 is similar to Figure 2, except that the abscissa is now X/C , where X is the projection of the arc length L onto the x -axis. The effect of this change in abscissa is to "squeeze" the curves in towards the origin. This squeezing is greatest near the origin, so that the most apparent effect is to sharpen the peak in the curves. Although it is somewhat redundant to present collision efficiencies as functions of X/C and L/C for a cylinder, the difference is more meaningful for airfoils, as some of the historical papers prefer X/C and others prefer L/C . For airfoils, a plot of collision efficiency vs L/C provides more resolution near the nose.

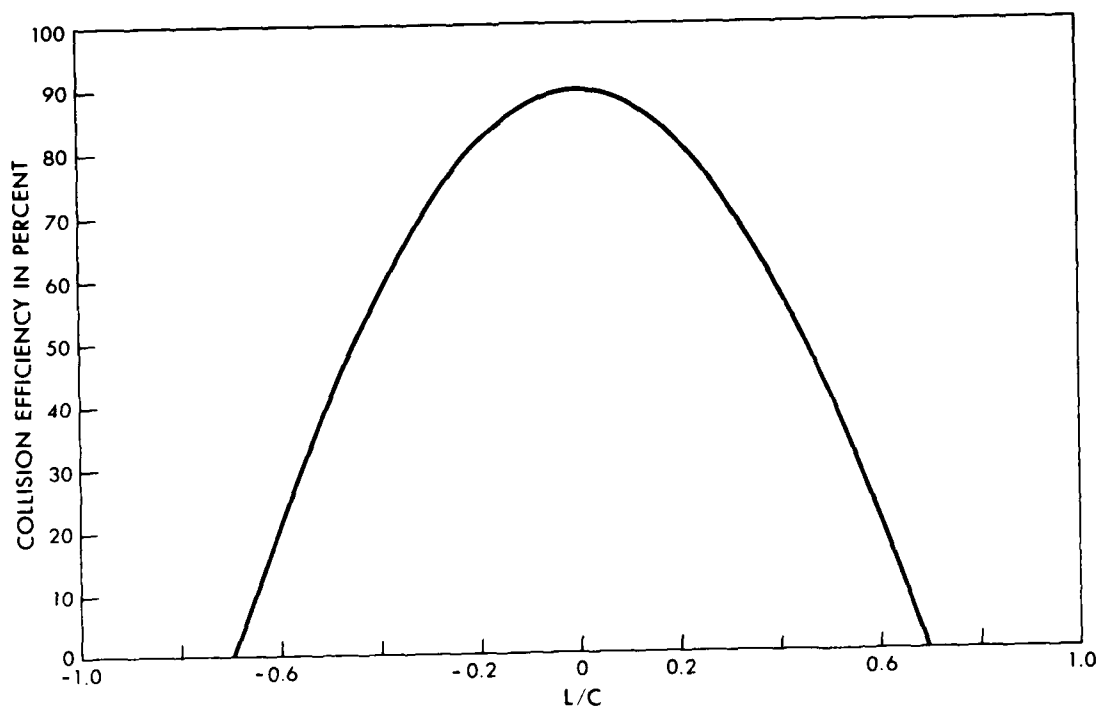


a. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 42.1 \mu\text{m}$.

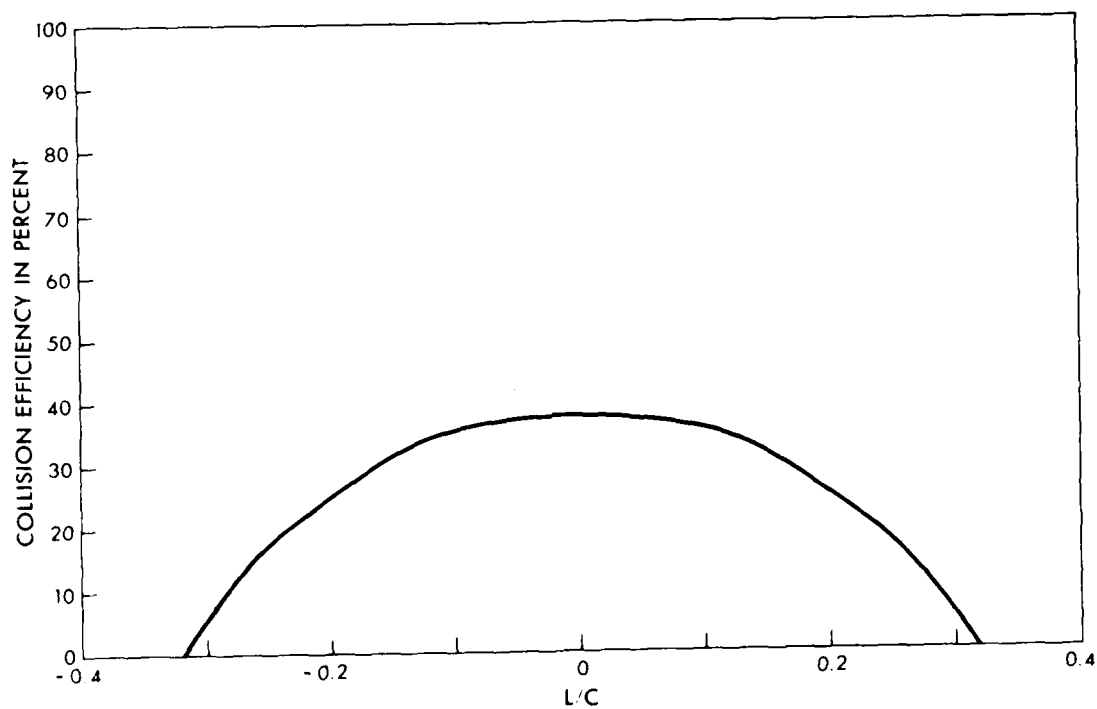


b. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 7.0 \mu\text{m}$.

Figure 1. Trajectories about a cylinder.

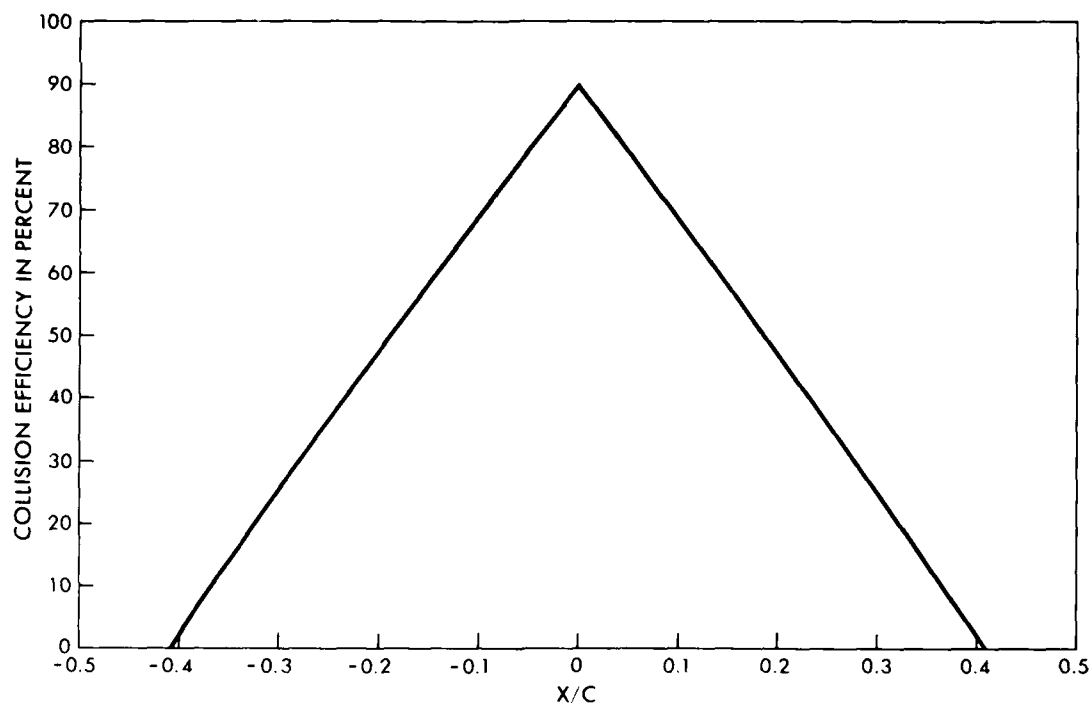


a. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 42.1 \text{ }\mu\text{m}$.

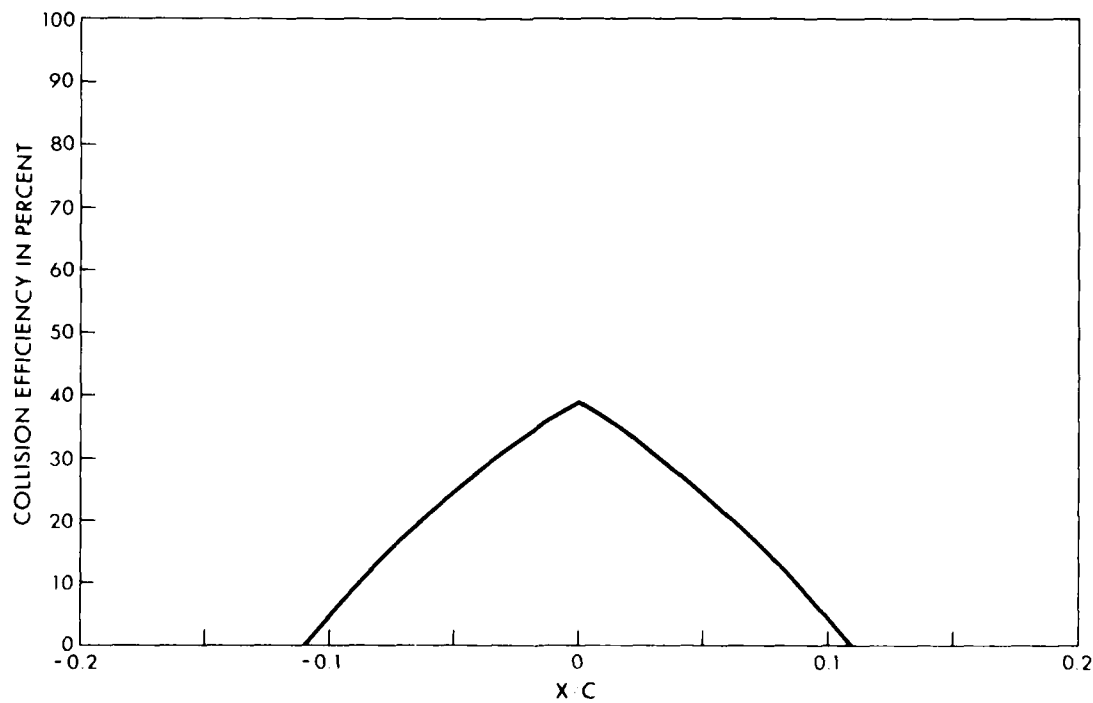


b. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 7.0 \text{ }\mu\text{m}$.

Figure 2. Collision efficiency vs length along cylinder surface.



a. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 42.1 \mu\text{m}$.



b. $C = 15 \text{ cm}$; $V_{\infty} = 114.3 \text{ m s}^{-1}$; $r_d = 7.0 \mu\text{m}$.

Figure 3. Collision efficiency vs distance along chord.

RESULTS AND DISCUSSION

Comparing results with and without the history term

Rows 4 and 5 of Table 1 present results of numerical experiments run respectively without and including the history term. All the other conditions of the experiment are identical. The droplet trajectories calculated with the history term are less influenced by the rapid changes in the airflow just before the collision, and so they tend to travel along straighter paths than those whose trajectories ignore the history term. This is explained by the fact that the history term acts to reduce the droplet acceleration, because it takes into account the finite rate at which vorticity is shed from the accelerating droplet. The net result is that in this case, ignoring the history term reduces the maximum impingement angle, θ_m , by 3.3° , reduces the stagnation line collection efficiency, β_0 , by 1.5%, and reduces the overall collection efficiency, E_m , by 1.7% (about 10% of the actual value). The particular case used to study the influence of the history term was chosen to give a large effect. In most cases, the effects would probably be less than those indicated here, suggesting that the term may be ignored without severely affecting the accuracy of the calculations.

Collision efficiency of NACA 0015 airfoil at 8° attack angle

Thus far, the computational icing experiments have been limited to cylinders. Let us now consider the case of a NACA 0015 airfoil at an attack angle of 8° . The chord length is 16.9 cm, the freestream speed 30.5 m s^{-1} , and the droplet diameter $20 \mu\text{m}$. The history term is not included in the computation. Figure 4 illustrates the resulting airflow (indicated by velocity vectors) and droplet trajectories (indicated by solid curves) for this case. It should be kept in mind that the flow region depicted in the figure is only a small portion of the total flow considered. In addition, the coordinate system is fixed to the airfoil so that the flow appears to be impinging upwards in a horizontally oriented airfoil. In fact, the entire figure should be rotated clockwise by 8° .

The droplet trajectories clearly indicate the asymmetry of the impingement above and below the stagnation point, when the airfoil is not at 0° attack. This is generally reflected in different icing characteristics above and below the stagnation line. Figure 5, which depicts the local collision efficiency, also illustrates this asymmetry. Negative values of L/C lie below the airfoil nose and positive values above it. The collision efficiency is a maximum close to (though not necessarily at) the stagnation line. The overall collision efficiency for this case is 50.1% and the maximum is 74.4% at a distance $L/C = -0.009$ below the nose. A comparison of Figure 5 has been made with the results of Bragg (private communication), who has also recently investigated this problem (see, for example, Bragg and Gregorek 1981). The differences between the two sets of computed results are generally negligible, in the sense that experiments would not likely be of sufficient accuracy to allow one to choose between the two.

Time-dependent accretion on NACA 0015 airfoil at 8° attack angle

The collision efficiencies plotted in Figure 5 are those for the airfoil surface itself at the onset of icing. Once a significant accretion has built up on the airfoil, the collision efficiencies change, and this change affects the subsequent development of the accretion. This feedback process between the accretion and the airflow and droplet trajectories goes on continuously in nature. We decided to simulate the continuous process in a step-wise fashion. Thus, using the computed initial collision efficiencies, we estimate the profile of the ice accretion after a finite, but small, time interval. We then use this new airfoil profile (including the already accreted ice layer) to determine a new flow field, droplet trajectories, and collision efficiencies. After that, a new increment to the accretion is once again calculated, and the entire process is repeated for as long a total period as desired. Other authors (e.g. Lozowski et

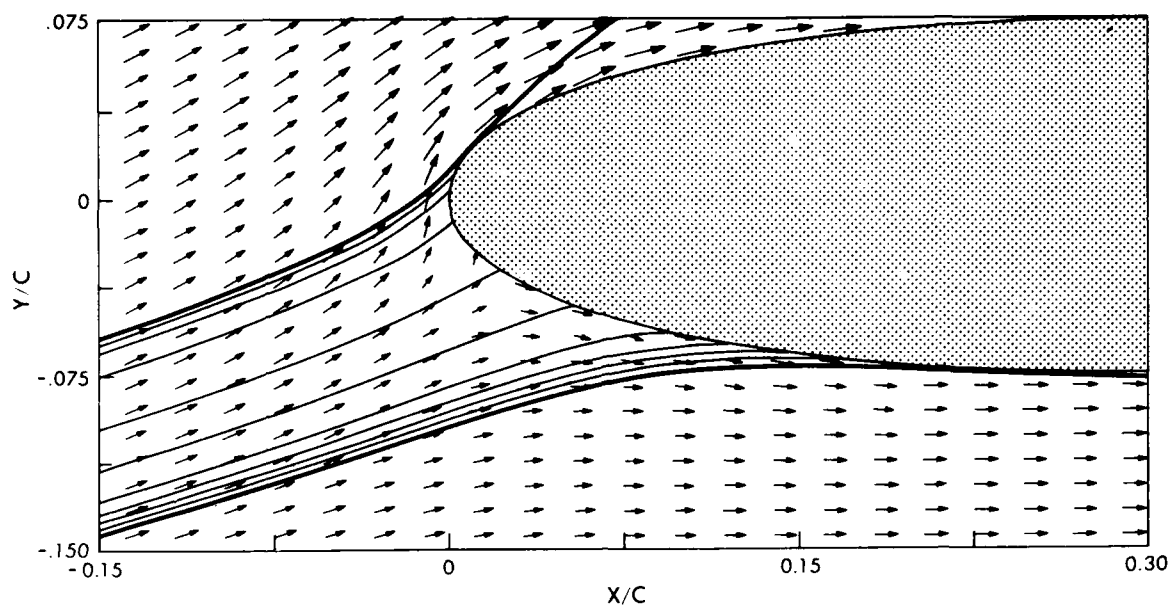


Figure 4. Trajectories about a NACA 0015 airfoil at 8° attack angle. $C = 16.9$ cm; $V_\infty = 30.5$ m s^{-1} ; $r_d = 10$ μ m.

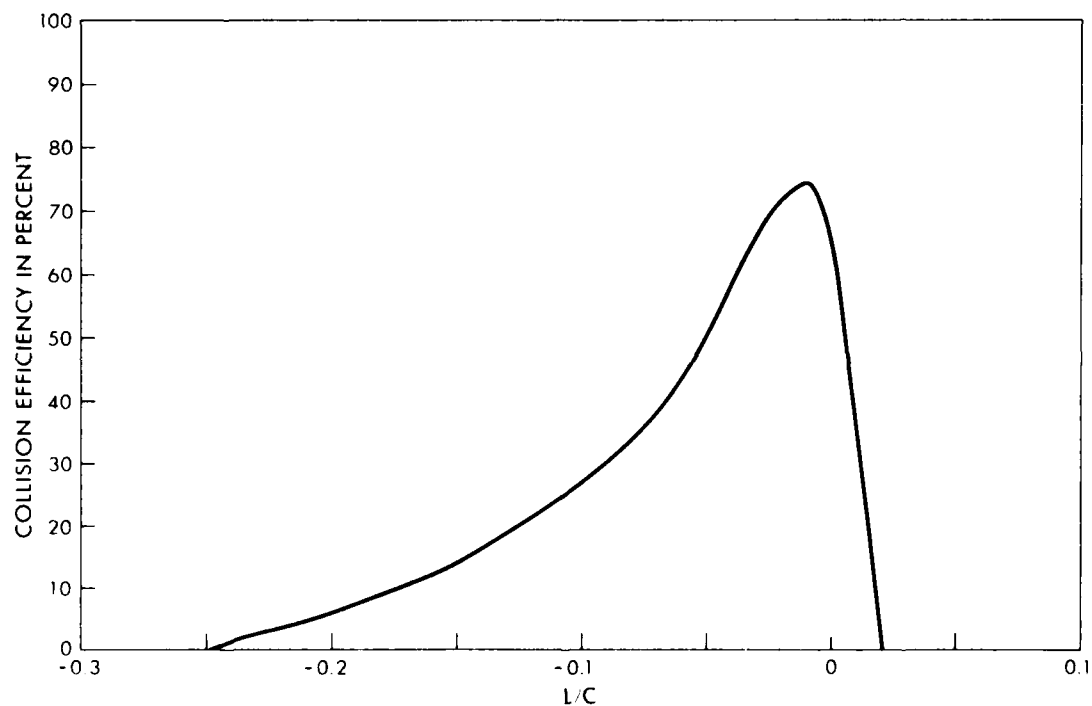


Figure 5. Collision efficiency vs length along airfoil surface (NACA 0015 airfoil at 8° attack angle). $C = 16.9$ cm; $V_\infty = 30.5$ m s^{-1} ; $r_d = 10$ μ m.

al. 1979) have not taken this feedback process into account, but instead have used the initial collision efficiencies to try to extrapolate the growth of the accretion over substantial periods of time. In this section we compare these two procedures: viz. single-step accretion vs multi-step with feedback. We also make a comparison with an experimental accretion grown in the NRC high-speed icing tunnel.

Figure 6 shows the airflow and the droplet trajectories for a NACA 0015 airfoil at an 8° angle of attack, but the conditions are somewhat different from those of Figure 4. In the present case, the chord length is 21.3 cm, the freestream speed 61 m s^{-1} , and the droplet diameter $20 \text{ }\mu\text{m}$. The history term is not included in this simulation. The solid line in Figure 7 shows the initial local collision efficiency β vs the nondimensional length L/C along the airfoil surface. Based on these values of β , and assuming a cloud liquid water content of 0.4 g m^{-3} and an accretion density of 890 kg m^{-3} , the accretion growth in a single step over 2.5 minutes was calculated. The modified airfoil profile, with the ice accretion attached, was then used as a basis for calculating a new airflow and new droplet trajectories. From these, a new determination was made of the local collision efficiency after 2.5 minutes of icing. This is indicated as the dashed curve in Figure 7 (L now being measured from the nose of the accretion rather than from the nose of the airfoil). Although the differences between the solid and dashed curves are not striking, it is clear that there are some. In particular, E_m falls with time from 58.2% to 56.5% while β_0 , the maximum collision efficiency, actually rises from 75.5% to 78.9%. Although the comparison is difficult to interpret because L/C has a slightly different meaning in each case (although C itself remains the chord length of the basic airfoil), it is apparent that the collision efficiency distribution has narrowed and become more peaked as a result of the change in the airflow caused by the first 2.5 minutes of accretion.

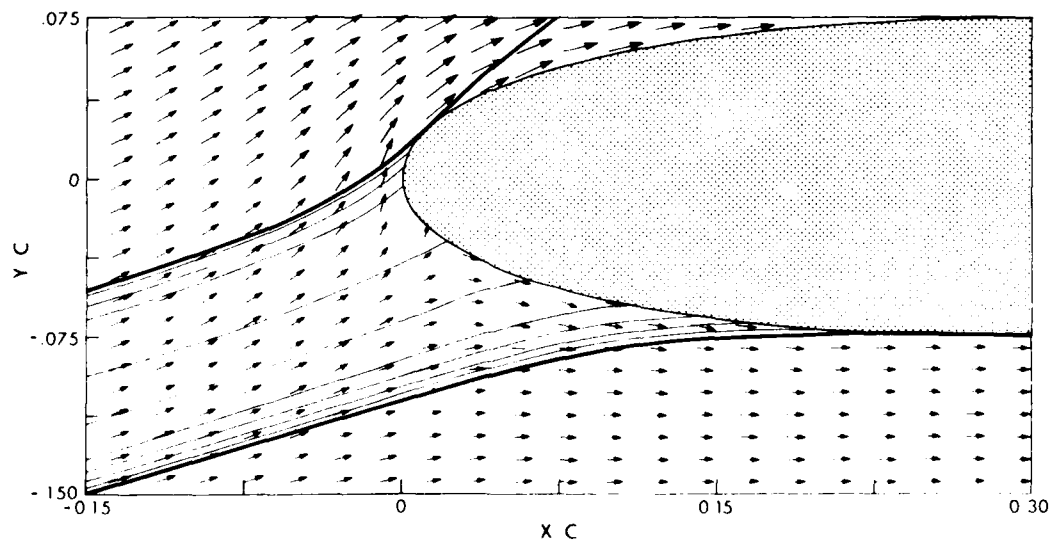


Figure 6. Trajectories about a NACA 0015 airfoil at 8° attack angle, $C = 21.3 \text{ cm}$; $V_\infty = 61 \text{ m s}^{-1}$; $r_d = 10 \text{ }\mu\text{m}$.

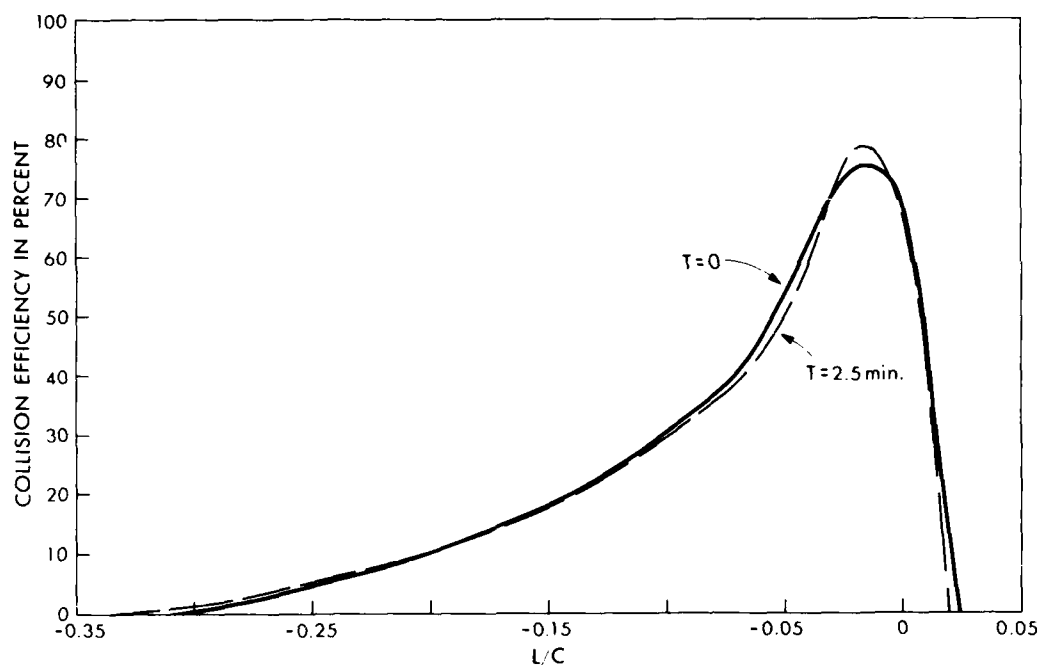


Figure 7. Collision efficiency vs length along airfoil surface (NACA 0015 airfoil at 8° attack angle). $C = 21.3$ cm; $V_\infty = 61$ m/s; $v_d = 10$ μ m.

The decrease in the collision efficiency that occurs with time on the lower surface near the nose is illustrated in Figure 8, which shows the accretion profiles after the addition of two successive 2.5-minute accretions. This two-step accretion shape is compared in Figure 9 with a single-step 5-minute accretion, calculated using only the initial collision efficiencies. The single-step accretion model overestimates the growth above and below the nose and underestimates it at the nose itself. Although the differences between the single and multistep models may seem relatively minor over this period of time, they will be much more significant over longer periods, the multistep method giving much more realistic results.

The shape of an experimental accretion profile made under similar conditions in the NRC high-speed icing tunnel (Stallabrass and Lozowski 1978) is also shown in Figure 9. The experimental and theoretical results are not perfectly comparable because a droplet size (approximately equal to the medium volume diameter of the tunnel droplet spectrum) was employed in the model. The general agreement is quite encouraging, though one gets the impression that the model accretion occurs too low on the airfoil relative to the experimental one. This discrepancy may be the result of a bias error in the model. On the other hand, it may have to do with the way the experimental profiles were measured. The experimental profiles were obtained by making an impression in plasticine and then photographing their outline against the outline of the airfoil. Inaccurate registration of the airfoil outline and that of the plasticine mold may have displaced the experimental profile upwards from where it should be. Only further experiments, with an improved technique for measuring the experimental profile, can resolve which is the correct explanation.

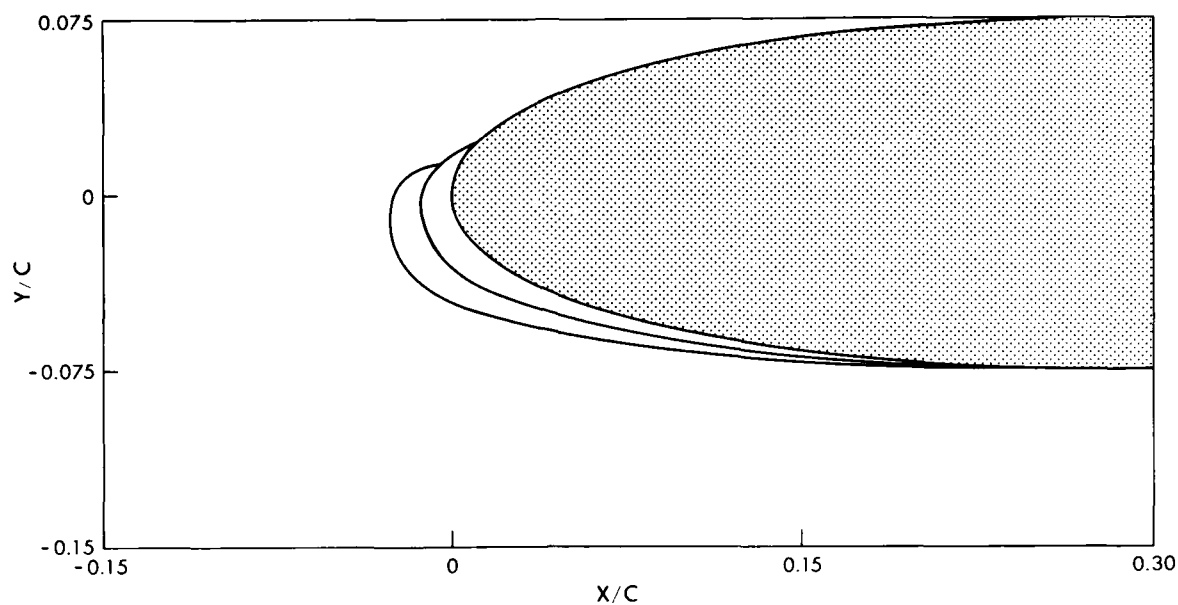


Figure 8. Accretion after 2.5 min and 5 min on NACA 0015 airfoil at 8° attack angle. $C = 21.3$ cm; $V_\infty = 61$ m s $^{-1}$; $r_d = 10$ μ m; LWC = 0.4 g m $^{-3}$.

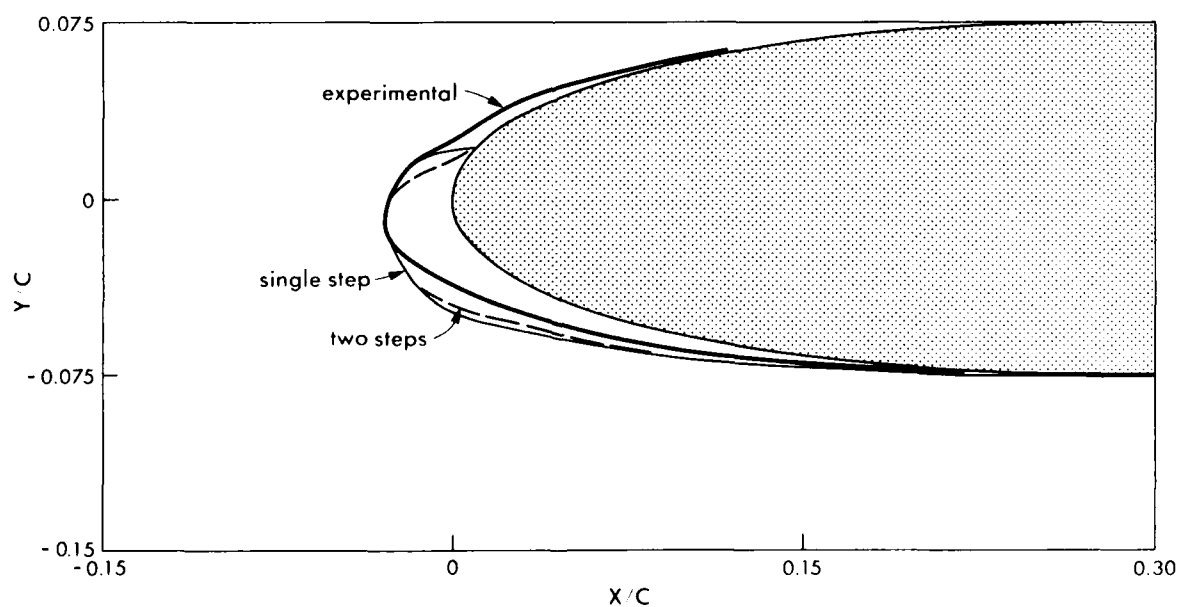


Figure 9. Accretion after 5 min in a single step (solid), two steps (dashed), and experimentally (bold) (NACA 0015 airfoil at 8° attack angle). $C = 21.3$ cm; $V_\infty = 61$ m s $^{-1}$; $r_d = 10$ μ m; LWC = 0.4 g m $^{-3}$.

Time-dependent accretion on NACA 0015 airfoil at 0° attack angle

In this section we demonstrate that the multistep accretion process can be continued for as many as five steps, after each of which a new potential flow and new droplet trajectories are calculated. We have not as yet attempted to increase the number of steps beyond five, although we see no reason why, in principle, this could not be done. Figure 10 illustrates the airflow and droplet trajectories at the initial time before ice accretion begins. The airfoil chord length is 21.3 cm, the freestream velocity 91.5 m s^{-1} , and the droplet diameter $20 \mu\text{m}$. To be rigorous, the history term was included in the trajectory calculations for this simulation, although generally speaking its effect may not be large.

Figure 11 shows the collision efficiency at the initial time and after 3 minutes of ice accretion. The local collision efficiency increases with time near the nose and diminishes with time farther back along the airfoil surface. Table 2 shows that the overall collision efficiency decreases with time, while the collision efficiency at the nose increases with time.

The result of this effect on the accretion itself is shown in Figure 12, where we see that the accretion tends to become more "pointed" with time, and that the growth rate at the nose in the model increases with time. This result seems reasonable inasmuch as the effect of the accretion is to decrease the local radius of curvature at the nose, thereby enhancing the collision efficiency and increasing the growth rate. Unfortunately, no time-dependent experimental growth rate measurements are available to confirm this result. Figure 13 compares the resulting accretion for the multistep approach with that obtained using a single 5-minute step. The single-step accretion slightly underestimates the growth at the nose and greatly overestimates it farther back.

An experimental ice accretion profile grown under similar conditions in the NRC high-speed icing tunnel is included in Figure 13 for comparison. Although the agreement is good at the nose, a substantial difference occurs farther back. We suspect that this is due to the growth of low-density feathery rime in the experimental case. Because we assume an ice density of 890 kg m^{-3} in the model, feathery rime growth is not taken into account.

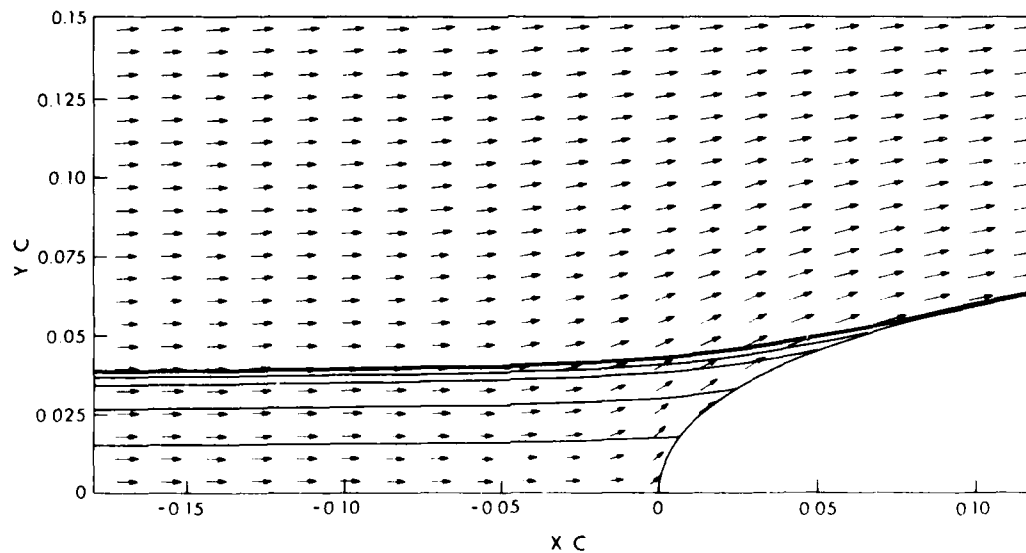


Figure 10. Trajectories about a NACA 0015 airfoil at 0° attack angle. $C = 21.3 \text{ cm}$; $V_\infty = 91.5 \text{ m s}^{-1}$; $r_d = 10 \mu\text{m}$; $\text{LWC} = 0.4 \text{ g m}^{-3}$.

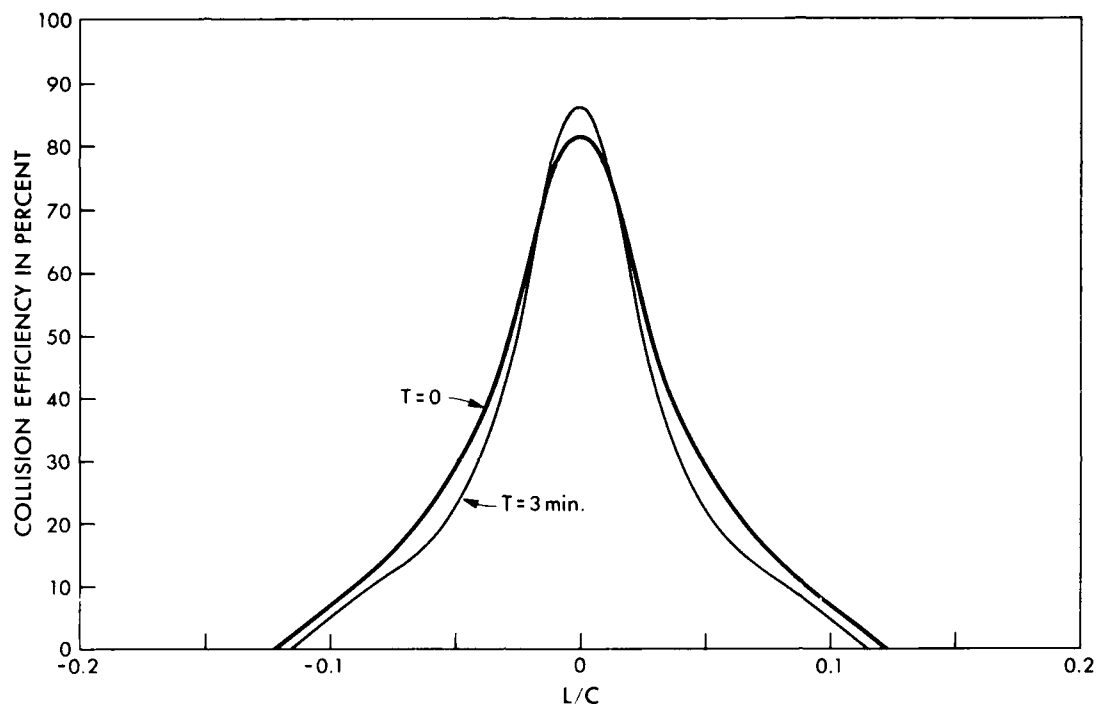


Figure 11. Collision efficiency vs length along airfoil surface (NACA 0015 airfoil at 0° attack angle). $C = 21.3$ cm; $V_\infty = 91.5$ m s $^{-1}$; $r_d = 10$ μ m; LWC = 0.4 g m $^{-1}$.

Table 2. Collision efficiencies as a function of time for the case of Figure 10.

Time (min)	E_m (%)	β_0 (%)
0	49.1	81.8
1	47.7	83.8
2	46.1	85.5
3	44.7	86.3
4	43.3	87.1

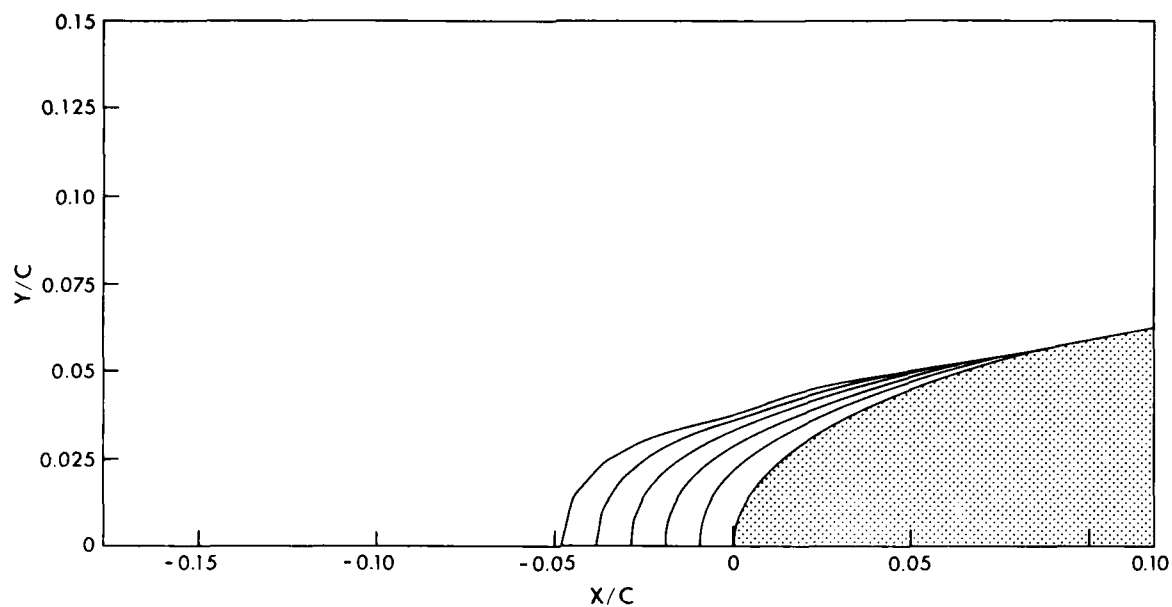


Figure 12. Accretion after 1 through 5 min on a NACA 0015 airfoil at 0° attack angle. $C = 21.3$ cm; $V_\infty = 91.5$ m s $^{-1}$; $r_d = 10$ μ m; LWC = 0.4 g m $^{-3}$.

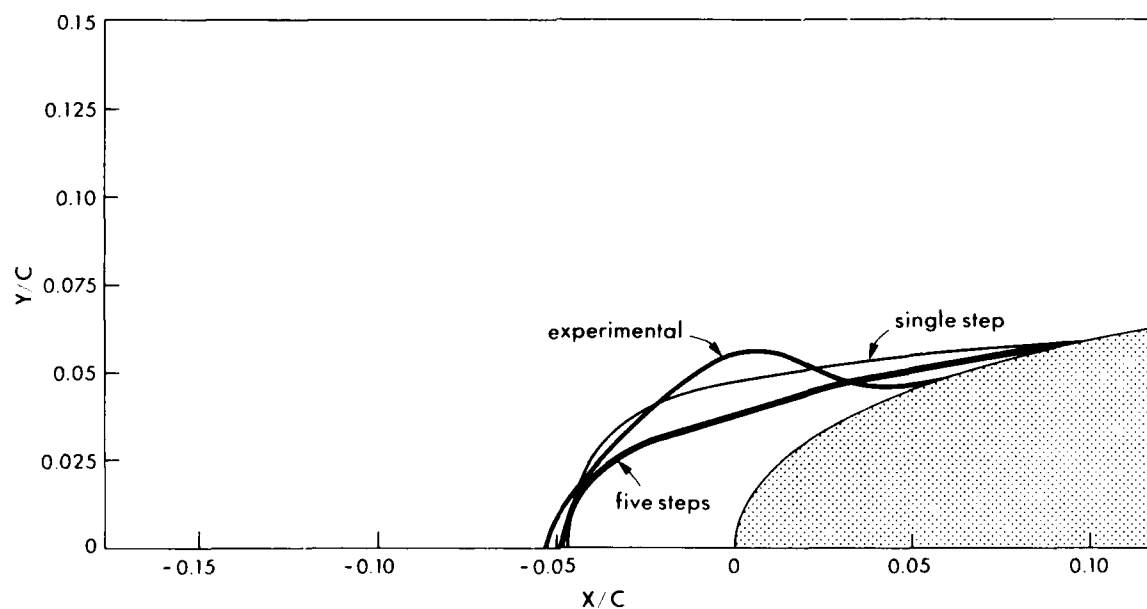


Figure 13. Accretion after 5 min in a single step (solid), five steps (boldest), and experimentally (bold) (NACA 0015 airfoil at 0° attack angle). $C = 21.3$ cm; $V_\infty = 91.5$ m s $^{-1}$; $r_d = 10$ μ m; LWC = 0.4 g m $^{-3}$.

CONCLUSIONS AND RECOMMENDATIONS

1. The principal accomplishment of the work performed under the present contract has been the development and testing of a computational simulation model of rime icing on arbitrary two-dimensional airfoils. The computer code for the model is presented in Appendix C. The program is annotated so that it should be possible for scientists elsewhere to run the program, check the present results, and develop new results for their own applications. Should any difficulties be encountered in the implementation of the model program, the authors will be pleased to offer their advice and assistance.

2. Most of the model runs described in the present report have been performed to test the accuracy of the components of the model. The potential flow field was tested against the known analytic solution for a circular cylinder, and was found to behave acceptably (stream function errors of 0.1% or less, which give rise to collision efficiency errors of less than 0.5%). The accuracy of the droplet trajectories was determined by comparing the model-predicted collision efficiencies with those computed by Langmuir and Blodgett (1946). Relative agreement to better than 10% was found even in the worst case. Finally, the ice accretion profiles themselves were tested against two experimental accretions, and, while the agreement was not exact, it was encouraging as to the model's simulation capabilities.

3. The other model runs presented in this report were performed either to test the importance of the history term in the droplet equations of motion or to compare a single-step vs a multistep accretion process. Although these tests were not exhaustive, they did indicate that omitting the history term did not have a dramatic effect on the results. The biggest effect of the history term occurred in cases with low collision efficiencies. The tests also showed that the accretion profiles predicted by the single-step and multistep processes are different, and that the difference increases with the duration of the accretion. As a result of these tests, and because in principal the multistep accretion procedure better simulates what is happening in nature, we recommend that the single-step approach be used only for brief accretion durations. Thus, for example, a single-step model might be quite useful for helicopter deicing calculations. On the other hand, the multistep method would be preferable for simulations of powerline icing where the duration may be hours or days.

4. Within the scope of the present contract, it has not been possible to use the model to investigate the effects of various parameters on the shape and development of the accretion. We recommend, however, that such studies be undertaken with the model. Although the model is presently limited to simulating rime accretions, there are many questions that it can be used to investigate. What is the effect of airfoil size and shape on the accretion? What would be the effect on the accretion of using a realistic cloud droplet distribution (see, for example, Ackley and Templeton 1979) rather than a single droplet size? What parameters should be simulated to properly model ice accretions at a reduced scale? How is the accretion changed if the airfoil attack angle and the airstream speed oscillate as they would for a helicopter rotor blade? Such questions and many others could and should be profitably considered using the present icing model.

LITERATURE CITED

- Abraham, F.F.** (1970) Functional dependence of the drag coefficient of a sphere on Reynolds number. *Physics of Fluids*, **13**: 2194-2195.
- Ackley, S.F. and M.K. Templeton** (1979) Computer modeling of atmospheric ice accretion. U.S. Army Cold Regions Research and Engineering Laboratory, CRREL Report 79-4. ADA 068582.
- Bragg, M.B. and G.M. Gregorek** (1981) An analytical approach to airfoil icing. AIAA 19th Aerospace Sciences Meeting, St. Louis, 12-15 January 1981. New York: American Institute of Aeronautics and Astronautics, AIAA-81-0403.
- Burden, R.L., J.D. Faires and A.C. Reynolds** (1978) *Numerical Analysis*. Boston: Prindle, Weber & Schmidt, 579 pp.
- Cansdale, J.T. and I.I. McNaughtan** (1977) Calculation of surface temperature accretion rate in a mixed water droplet/ice crystal cloud. Royal Aircraft Establishment, Technical Report 77090, 24 pp.
- Crowe, C.T., J.A. Nicholls and R.B. Morrison** (1963) Drag coefficients of inert and burning particles accelerating in gas streams. In *Ninth Symposium on Combustion*. New York: Academic Press, pp. 395-405.
- Hess, J.L. and A.M.O. Smith** (1967) Calculation of potential flow about arbitrary bodies. *Progress in Aeronautical Science*, **8**.
- Kennedy, J.L. and D.J. Marsden** (1976) Potential flow velocity distributions on multi-component airfoil sections. *Canadian Aeronautics and Space Journal*, **22**: 243-256.
- Langmuir, I. and K.B. Blodgett** (1946) A mathematical investigation of water droplet trajectories. In *Collected Works of Irving Langmuir*. New York: Pergamon Press, **10**: 348-393.
- Lapidus, L. and J.H. Seinfeld** (1971) *Numerical Solutions of Ordinary Differential Equations*. New York: Academic Press, 299 pp.
- Lozowski, E.P. and M.M. Oleskiw** (1981) Computer simulation of airfoil icing without run-back. AIAA 19th Aerospace Sciences Meeting, St. Louis, 12-15 January 1981. New York: American Institute of Aeronautics and Astronautics, AIAA-81-0402.
- Lozowski, E.P., J.R. Stallabrass and P.F. Hearty** (1979) The icing of an unheated non-rotating cylinder in liquid water droplet-ice crystal clouds. National Research Council of Canada, Report LTR-LT-96, 109 pp.
- McComber, P. and G. Touzot** (1981) Calculation of the impingement of cloud droplets on a cylinder by the finite-element method. *Journal of the Atmospheric Sciences*, **38**: 1027-1036.
- Oleskiw, M.M. and E.P. Lozowski** (1980) Helicopter rotor blade icing: A numerical simulation. In *Proceedings of the 8th International Conference on Cloud Physics, Clermont-Ferrand, France, 15-19 July 1980*. International Commission on Cloud Physics.
- Pearcey, T. and G.W. Hill** (1956) The accelerated motion of droplets and bubbles. *Australian Journal of Physics*, **9**: 19-30.
- Rosen, K.M. and M.L. Potash** (1981) 40 years of helicopter ice protection experience at Sikorsky Aircraft. AIAA 19th Aerospace Sciences Meeting, St. Louis, 12-15 January 1981. New York: American Institute of Aeronautics and Astronautics, AIAA-81-0407.
- Sartor, J.D. and C.E. Abbott** (1975) Prediction and measurement of the accelerated motion of water drops in air. *Journal of Applied Meteorology*, **14**: 232-239.
- Stallabrass, J.R. and E.P. Lozowski** (1978) Ice shapes on cylinders and rotor blades. Paper presented at NATO Panel X Symposium on Helicopter Icing, November 1978, London.
- Theodorson, T. and I.E. Garrick** (1932) General potential theory of arbitrary wing sections. NACA Report 452. Washington, D.C.: National Advisory Committee on Aeronautics.

APPENDIX A: SAMPLE INPUT

This appendix contains a sample of the parameters that must be input to the program along with examples of their typical values. These parameters are read by the program from input device 4 (see, for example, program line 186).

```

1  NEF,NEB,NIF,
2  9, 11, 3,
3  ALPHA,TYPE,THICK,XMIN,XMAX,YMIN,YMAX,XZ,YZ,ANAL,
4  0.0, 1,100.0,-0.6, 0.6, 0.0, 0.6,43,43, 1,
5  PLTFAC,TRJPLA,YOL,CEL,CEX,ICEPLA,LYRMAX,CETOL, ICE,
6  1.0, 1, 1, 1, 1, 1, 1, 0.3, 0.05,
7  UINF, C, PINF, TINF, RD, A1, A2,
8  114.3, 0.15, 78.5,-10.0, 42.1, 0.00, 6.25D-2,
9  CDS,TRJPRA,PRINTI,PLOTI,PRINTO,CRIT,BETAO,
10 1, 1, 25, 50, 10, 1.0, 0.89,
11 NTRAJU,NTRAJL,AT,BOTH,EQN,PC, DTS, EPS,ACN,
12 6, 0, 1, 0, 1, 2,.06D0, 1.D-6, 0.
13 XO,
14 -5.0,
15 YO,... FIRST LAYER
16 0.4065,
17 YO,... SECOND LAYER
18 0.0,
19 YO,... THIRD LAYER
20 0.0,
21 YO,... FOURTH LAYER
22 0.0,
23 YO,... FIFTH LAYER
24 0.0,
END OF FILE

```

APPENDIX B: SAMPLE OUTPUT

This appendix contains a sample of the printed output produced by the program starting with the sample input values given in Appendix A. This output is written to output device 7. The trajectories calculated correspond to those of Figure 1. The collision efficiency data (on the last page) correspond to Figures 3 and 5.

FOR EQN. SOLN. IER= 0

THE POTENTIAL FLOW LIFT COEFFICIENT IS 0.00000

CONTROL PT.	X COORD.	Y COORD.	SFC. VEL.
1	0.00380	0.04341	-0.17408
2	0.01887	0.12892	-0.51694
3	0.04857	0.21050	-0.84409
4	0.09198	0.28570	-1.14558
5	0.14779	0.35221	-1.41223
6	0.21430	0.40802	-1.63591
7	0.28949	0.45143	-1.80976
8	0.37108	0.48113	-1.92846
9	0.45659	0.49620	-1.98401
10	0.53911	0.49692	-1.99819
11	0.61636	0.48469	-1.94391
12	0.69075	0.46052	-1.84660
13	0.76044	0.42501	-1.70398
14	0.82372	0.37903	-1.51956
15	0.87903	0.32372	-1.29778
16	0.92501	0.26044	-1.04407
17	0.96052	0.19075	-0.76468
18	0.98469	0.11636	-0.46647
19	0.99692	0.03911	-0.15678
20	0.99692	-0.03911	0.15678
21	0.98469	-0.11636	0.46647
22	0.96052	-0.19075	0.76468
23	0.92501	-0.26044	1.04407
24	0.87903	-0.32372	1.29778
25	0.82372	-0.37903	1.51956
26	0.76044	-0.42501	1.70398
27	0.69075	-0.46052	1.84660
28	0.61636	-0.48469	1.94391
29	0.53911	-0.49692	1.99819
30	0.45659	-0.49620	1.98401
31	0.37108	-0.48113	1.92846
32	0.28949	-0.45143	1.80976
33	0.21430	-0.40802	1.63591
34	0.14779	-0.35221	1.41223
35	0.09198	-0.28570	1.14558
36	0.04857	-0.21050	0.84409
37	0.01887	-0.12892	0.51694
38	0.00380	-0.04341	0.17408
39	1.00003	0.0	-0.00000

TRAJECTORY STARTING POSITION IS X= -5.00 Y0= 0.40700

STEP	TIME	DTS	YDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94048	0.40707	0.40366	0.99174	0.99192	0.00124	0.00120	0.10834	0.00000	0.0		
3	0.17	0.1944	-4.83336	0.40720	0.40363	0.99141	0.99192	0.00132	0.00120	0.30974	0.00000	0.0		
4	0.36	0.3499	-4.64053	0.40744	0.40348	0.99077	0.99191	0.00147	0.00121	0.70110	0.00000	0.0		
5	0.71	0.5161	-4.29345	0.40786	0.40300	0.98941	0.99187	0.00181	0.00121	1.51575	0.00000	0.0		
6	1.23	0.6160	-3.78156	0.40849	0.40183	0.98681	0.99174	0.00254	0.00125	3.06155	0.00000	0.0		
7	1.84	0.5682	-3.17072	0.40929	0.39972	0.98223	0.99140	0.00401	0.00134	5.73218	0.00000	0.0		
8	2.41	0.5974	-2.60755	0.41010	0.39542	0.97557	0.99076	0.00656	0.00154	9.59751	0.00000	0.0		
9	3.01	0.5889	-2.01605	0.41114	0.38640	0.96375	0.98946	0.01217	0.00201	16.58722	0.00000	0.0		
10	3.60	0.4623	-1.43403	0.41262	0.37090	0.94199	0.98684	0.02593	0.00320	30.17147	0.00000	0.0		
11	4.06	0.2998	-0.97867	0.41459	0.35116	0.90399	0.98256	0.05477	0.00573	52.54998	0.00002	0.0		
12	4.36	0.2366	-0.68478	0.41682	0.32385	0.87717	0.97743	0.09862	0.00973	80.39044	0.00004	0.0		
13	4.60	0.1772	-0.45424	0.41983	0.29071	0.84551	0.97091	0.16853	0.01651	118.23268	0.00009	0.0		
14	4.77	0.1310	-0.28281	0.42355	0.25485	0.82837	0.96402	0.26251	0.02651	163.31494	0.00034	0.0		
15	4.91	0.1196	-0.15692	0.42780	0.21027	0.83644	0.95804	0.36970	0.03938	211.18507	0.00070	0.0		
16	5.03	0.0846	-0.04263	0.43355	0.17077	0.88622	0.95325	0.50224	0.05820	269.43021	0.00072	0.0		
17	5.11	0.0772	0.03789	0.43924	0.12934	0.96888	0.95214	0.61034	0.07743	319.88825	0.00104	0.0		
18	5.19	0.0575	0.11153	0.44607	0.09650	1.09732	0.95518	0.70329	0.10023	371.73509	0.00178	0.0		
20	5.28	0.0246	0.20179	0.45703	0.06219	1.33382	0.93938	0.96871	0.13400	436.57149	0.00527	0.0		
23	5.34	0.0166	0.26306	0.46619	0.03511	1.53938	0.98686	0.72567	0.15802	475.26932	0.00862	0.0		
29	5.41	0.0044	0.32795	0.47719	0.01462	1.78278	1.04124	0.63857	0.18036	536.82450	0.03760	0.0		
39	5.46	0.0079	0.37972	0.48659	0.00501	1.83917	1.04124	0.46137	0.19194	505.28922	0.01786	0.0		
78	5.52	0.0040	0.43886	0.49756	0.00659	1.94067	1.07571	0.20471	0.19835	518.96558	0.03485	0.0		
86	5.56	0.0051	0.49214	0.50723	0.01875	1.98619	1.10427	0.06653	0.19658	534.84820	0.02742	0.0		
96	5.61	0.0083	0.54719	0.51670	0.04404	1.90441	1.13258	-0.16026	0.18761	507.93519	0.01477	0.0		
101	5.65	0.0084	0.59496	0.52431	0.07079	1.82285	1.15305	-0.31023	0.17586	496.53490	0.01354	0.0		
102	5.66	0.0087	0.60464	0.52577										

CLOSEST APPROACH IS Y= 0.00008 NO. OF STEPS REQUIRED=102 PSI= 0.071

TRAJECTORY STARTING POSITION IS X= -5.00 Y0= 0.40692

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94048	0.40699	0.40358	0.99174	0.99192	0.00124	0.00120	0.10834	0.00000	0.0		
3	0.17	0.1944	-4.83336	0.40712	0.40355	0.99141	0.99192	0.00132	0.00120	0.30974	0.00000	0.0		
4	0.36	0.3499	-4.64053	0.40735	0.40340	0.99077	0.99191	0.00147	0.00121	0.70110	0.00000	0.0		
5	0.71	0.5161	-4.29345	0.40778	0.40292	0.98941	0.99187	0.00181	0.00121	1.51576	0.00000	0.0		
6	1.23	0.6160	-3.78156	0.40841	0.40175	0.98681	0.99174	0.00254	0.00125	3.06156	0.00000	0.0		
7	1.84	0.5682	-3.17073	0.40920	0.39964	0.98223	0.99140	0.00401	0.00134	5.73218	0.00000	0.0		
8	2.41	0.5974	-2.60756	0.41002	0.39534	0.97557	0.99076	0.00656	0.00154	9.59750	0.00000	0.0		
9	3.01	0.5889	-2.01606	0.41106	0.38632	0.96375	0.98946	0.01217	0.00201	16.58716	0.00000	0.0		
10	3.60	0.4623	-1.43405	0.41253	0.37083	0.94198	0.98684	0.02593	0.00320	30.17128	0.00000	0.0		
11	4.06	0.2998	-0.97869	0.41450	0.35109	0.90998	0.98256	0.05476	0.00573	52.55051	0.00002	0.0		
12	4.36	0.2366	-0.68481	0.41674	0.32378	0.87715	0.97743	0.09860	0.00973	80.39062	0.00004	0.0		
13	4.60	0.1772	-0.45428	0.41975	0.29064	0.84548	0.97091	0.16850	0.01651	118.23119	0.00009	0.0		
14	4.77	0.1310	-0.28283	0.42347	0.25479	0.82832	0.96402	0.26249	0.02651	163.32065	0.00020	0.0		
15	4.91	0.1196	-0.15696	0.42771	0.21021	0.83635	0.95803	0.36968	0.03937	211.19303	0.00034	0.0		
16	5.03	0.0845	-0.04268	0.43346	0.17072	0.88608	0.95324	0.50224	0.05819	269.44352	0.00072	0.0		
17	5.11	0.0772	0.03784	0.43916	0.12928	0.96869	0.95212	0.61038	0.07743	319.90951	0.00104	0.0		

18	5.19	0.0576	0.11147	0.44598	0.09641	1.09709	0.95515	0.70340	0.10023	371.76847	0.00178	0.0
20	5.28	0.0246	0.20178	0.45595	0.06212	1.33375	0.96869	0.76365	0.13402	436.66320	0.00528	0.0
23	5.34	0.0167	0.26304	0.46611	0.03500	1.53940	0.98683	0.72594	0.15806	475.38563	0.00862	0.0
29	5.41	0.0044	0.32792	0.47711	0.01450	1.78308	1.01425	0.63926	0.18041	537.18119	0.03760	0.0
39	5.46	0.0079	0.37959	0.48650	0.00489	1.83901	1.04118	0.46170	0.19198	505.29246	0.01788	0.0
82	5.52	0.0041	0.43959	0.49762	0.00660	1.94009	1.07614	0.20299	0.19842	518.35470	0.03386	0.0
90	5.56	0.0054	0.49215	0.50716	0.01873	1.98654	1.10431	0.06672	0.19665	535.01908	0.02598	0.0
100	5.62	0.0084	0.55014	0.51713	0.04547	1.90153	1.13398	-0.16891	0.18706	507.62302	0.01453	0.0
105	5.66	0.0084	0.59813	0.52474	0.07265	1.81539	1.15431	-0.31826	0.17502	494.87569	0.01336	0.0
106	5.67	0.0089	0.60788	0.52620								

CLOSEST APPROACH IS Y= 0.00001 NO. OF STEPS REQUIRED=106 PSI= 0.073

TRAJECTORY STARTING POSITION IS X= -5.00 YO= 0.40690

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94048	0.40698	0.40357	0.99174	0.99192	0.00124	0.00120	0.10834	0.00000	0.0		
3	0.17	0.1944	-4.83336	0.40711	0.40353	0.99141	0.99192	0.00132	0.00120	0.30974	0.00000	0.0		
4	0.36	0.3499	-4.64053	0.40734	0.40338	0.99077	0.99191	0.00147	0.00121	0.70110	0.00000	0.0		
5	0.71	0.5161	-4.29345	0.40776	0.40291	0.98941	0.99187	0.00181	0.00121	1.51576	0.00000	0.0		
6	1.23	0.6160	-3.78156	0.40840	0.40173	0.98681	0.99174	0.00254	0.00125	3.06156	0.00000	0.0		
7	1.84	0.5682	-3.17073	0.40919	0.39963	0.98223	0.99140	0.00401	0.00134	5.73218	0.00000	0.0		
8	2.41	0.5974	-2.60756	0.41000	0.39533	0.97557	0.99076	0.00656	0.00154	9.59750	0.00000	0.0		
9	3.01	0.5889	-2.01607	0.41104	0.38630	0.96375	0.98946	0.01217	0.00201	16.58715	0.00000	0.0		
10	3.60	0.4623	-1.43405	0.41252	0.37081	0.94198	0.98684	0.02593	0.00320	30.17124	0.00000	0.0		
11	4.06	0.2998	-0.97869	0.41449	0.35108	0.90398	0.98256	0.05476	0.00573	52.55060	0.00002	0.0		
12	4.36	0.2365	-0.68481	0.41673	0.32377	0.87715	0.97743	0.09860	0.00973	80.39064	0.00004	0.0		
13	4.60	0.1772	-0.45429	0.41973	0.29063	0.84548	0.97091	0.16849	0.01651	118.23090	0.00009	0.0		
14	4.77	0.1310	-0.28284	0.42345	0.25478	0.82830	0.96402	0.26248	0.02651	163.32170	0.00020	0.0		
15	4.91	0.1196	-0.15696	0.42770	0.21020	0.83633	0.95803	0.36967	0.03937	211.19449	0.00034	0.0		
16	5.03	0.0845	-0.04269	0.43345	0.17071	0.88605	0.95324	0.50223	0.05819	269.44597	0.00072	0.0		
17	5.11	0.0772	0.03783	0.43914	0.12927	0.96866	0.95211	0.61038	0.07743	319.91342	0.00104	0.0		
18	5.19	0.0576	0.11146	0.44597	0.09639	1.09705	0.95515	0.70341	0.10023	371.77462	0.00178	0.0		
20	5.28	0.0246	0.20178	0.45694	0.06210	1.33374	0.96868	0.76370	0.13403	436.68007	0.00528	0.0		
23	5.34	0.0167	0.26303	0.46610	0.03498	1.53940	0.98683	0.72599	0.15807	475.40724	0.00862	0.0		
29	5.41	0.0044	0.32792	0.47709	0.01447	1.78314	1.01424	0.63938	0.18042	537.24778	0.03760	0.0		
39	5.46	0.0079	0.37956	0.48648	0.00487	1.83899	1.04117	0.46176	0.19199	505.29383	0.01789	0.0		
43	5.48	0.0024	0.40621	0.49142	0.00294	1.93025	1.05555	0.44102	0.19640	544.93309	0.06463	0.0		
44	5.49	0.0022	0.40878	0.49189										

COLLISION COORDS: X= 0.4087804 Y= 0.4916086 L= 0.6936905 NO. OF STEPS REQUIRED= 44

TRAJECTORY STARTING POSITION IS X= -5.00 YO= 0.40650

STEP TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94048	0.40557	0.40317	0.99174	0.99192	0.00124	0.00120	0.00000	0.0	0.00000	0.0
3	0.17	0.1944	-4.83336	0.40670	0.40313	0.99141	0.99192	0.00132	0.00120	0.00000	0.0	0.00000	0.0
4	0.36	0.3499	-4.64053	0.40694	0.40298	0.99077	0.99191	0.00147	0.00120	0.00000	0.0	0.00000	0.0
5	0.71	0.5161	-4.29345	0.40736	0.40251	0.98941	0.99187	0.00181	0.00121	0.00000	0.0	0.00000	0.0
6	1.23	0.6160	-3.78157	0.40799	0.40134	0.98681	0.99174	0.00254	0.00125	0.00000	0.0	0.00000	0.0
7	1.84	0.5682	-3.17075	0.40878	0.39923	0.98223	0.99140	0.00401	0.00134	0.00000	0.0	0.00000	0.0
8	2.41	0.5973	-2.60761	0.40959	0.39493	0.97557	0.99076	0.00655	0.00154	0.00000	0.0	0.00000	0.0
9	3.01	0.5889	-2.01614	0.41063	0.38592	0.96375	0.98946	0.01216	0.00201	0.00000	0.0	0.00000	0.0
10	3.60	0.4624	-1.43416	0.41211	0.37044	0.94197	0.98684	0.02590	0.00319	0.00000	0.0	0.00000	0.0
11	4.06	0.2997	-0.97876	0.41408	0.35073	0.90994	0.98255	0.05472	0.00573	0.00002	0.0	0.00000	0.0
12	4.36	0.2365	-0.68494	0.41631	0.32345	0.87708	0.97742	0.09852	0.00972	0.00004	0.0	0.00000	0.0
13	4.60	0.1773	-0.45450	0.41931	0.29032	0.84533	0.97091	0.16834	0.01649	0.00009	0.0	0.00000	0.0
14	4.77	0.1309	-0.28297	0.42303	0.25449	0.82802	0.96400	0.26237	0.02649	0.00020	0.0	0.00000	0.0
15	4.91	0.1196	-0.15714	0.42727	0.20994	0.83587	0.95800	0.36956	0.03935	0.00034	0.0	0.00000	0.0
16	5.03	0.0845	-0.04291	0.43302	0.17044	0.88534	0.95318	0.50222	0.05816	0.00072	0.0	0.00000	0.0
17	5.11	0.0772	0.03758	0.43871	0.12899	0.96771	0.95202	0.61056	0.07739	0.00104	0.0	0.00000	0.0
18	5.19	0.0578	0.11118	0.44553	0.09592	1.09590	0.95502	0.70392	0.10021	0.00177	0.0	0.00000	0.0
20	5.28	0.0243	0.20173	0.45654	0.06174	1.33335	0.96856	0.76485	0.13417	0.00536	0.0	0.00000	0.0
23	5.35	0.0181	0.26655	0.46628	0.03219	1.54963	0.98802	0.72176	0.15959	0.00793	0.0	0.00000	0.0
31	5.42	0.0034	0.33606	0.47819	0.01198	1.80826	1.01861	0.58066	0.18313	0.04695	0.0	0.00000	0.0
40	5.46	0.0079	0.38339	0.48684	0.00383	1.84628	1.04317	0.45726	0.19297	0.01799	0.0	0.00000	0.0
41	5.47	0.0077	0.39166	0.48837									

COLLISION COORDS: X= 0.3906382 Y= 0.4878934 L= 0.6751708 NO. OF STEPS REQUIRED= 41

TRAJECTORY STARTING POSITION IS X= -5.00 YO= 0.39860

STEP TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.39867	0.39533	0.99174	0.99191	0.00122	0.00118	0.00000	0.0	0.00000	0.0
3	0.17	0.1944	-4.83336	0.39880	0.39530	0.99141	0.99191	0.00129	0.00118	0.00000	0.0	0.00000	0.0
4	0.36	0.3499	-4.64053	0.39903	0.39515	0.99076	0.99190	0.00144	0.00118	0.00000	0.0	0.00000	0.0
5	0.71	0.5159	-4.29345	0.39944	0.39469	0.98941	0.99186	0.00178	0.00119	0.00000	0.0	0.00000	0.0
6	1.23	0.6157	-3.78177	0.40006	0.39353	0.98679	0.99174	0.00249	0.00122	0.00000	0.0	0.00000	0.0
7	1.84	0.5679	-3.17129	0.40084	0.39147	0.98221	0.99140	0.00393	0.00131	0.00000	0.0	0.00000	0.0
8	2.41	0.5968	-2.60848	0.40163	0.38726	0.97553	0.99075	0.00643	0.00151	0.00000	0.0	0.00000	0.0
9	3.01	0.5882	-2.01751	0.40265	0.37843	0.96367	0.98946	0.01192	0.00197	0.00000	0.0	0.00000	0.0
10	3.60	0.4633	-1.43618	0.40410	0.36318	0.94178	0.98683	0.02541	0.00313	0.00000	0.0	0.00000	0.0
11	4.06	0.2985	-0.97984	0.40603	0.34387	0.90924	0.98250	0.05385	0.00563	0.00002	0.0	0.00000	0.0
12	4.36	0.2350	-0.68730	0.40822	0.31720	0.87568	0.97734	0.09593	0.00954	0.00004	0.0	0.00000	0.0
13	4.59	0.1789	-0.45837	0.41114	0.28415	0.84253	0.97075	0.16555	0.01616	0.00009	0.0	0.00000	0.0
14	4.77	0.1300	-0.28527	0.41483	0.24880	0.82241	0.96357	0.26018	0.02614	0.00021	0.0	0.00000	0.0
15	4.90	0.1188	-0.16040	0.41899	0.20464	0.82691	0.95730	0.36744	0.03886	0.00034	0.0	0.00000	0.0
16	5.02	0.0840	-0.04703	0.42463	0.16522	0.87143	0.95199	0.50184	0.05755	0.00073	0.0	0.00000	0.0
17	5.11	0.0768	0.03284	0.43023	0.12349	0.94927	0.95027	0.61380	0.07682	0.00107	0.0	0.00000	0.0
19	5.23	0.0358	0.15352	0.44241	0.07321	1.18894	0.95745	0.76378	0.11772	0.00337	0.0	0.00000	0.0
22	5.31	0.0153	0.23201	0.45328	0.03874	1.43228	0.97438	0.79274	0.15001	0.00957	0.0	0.00000	0.0
27	5.38	0.0125	0.29798	0.46420	0.00945	1.63349	0.99896	0.70691	0.17571	0.01200	0.0	0.00000	0.0
32	5.41	0.0010	0.32607	0.46925	0.00260	1.76769	1.01193	0.75359	0.18592	0.01784	0.0	0.00000	0.0
33	5.41	0.0010	0.32713	0.46944									

COLLISION COORDS: X= 0.3271172 Y= 0.4691605 L= 0.6088966 NO. OF STEPS REQUIRED= 33

TRAJECTORY STARTING POSITION IS X= -5.00 Y= 0.37369

STEP	TIME	DTS	XDS	VDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.37375	0.37062	0.99172	0.99190	0.00114	0.00111	0.10848	0.00000	0.0		
3	0.17	0.1944	-4.83336	0.37387	0.37059	0.99139	0.99190	0.00121	0.00111	0.31015	0.00000	0.0		
4	0.36	0.3499	-4.64054	0.37409	0.37045	0.99074	0.99189	0.00135	0.00111	0.70205	0.00000	0.0		
5	0.71	0.5153	-4.29346	0.37448	0.37002	0.98938	0.99185	0.00167	0.00112	1.51798	0.00000	0.0		
6	1.23	0.6146	-3.78236	0.37506	0.36894	0.98675	0.99172	0.00234	0.00115	3.06404	0.00000	0.0		
7	1.84	0.5668	-3.17290	0.37579	0.36700	0.98214	0.99138	0.00369	0.00123	5.73272	0.00000	0.0		
8	2.41	0.5953	-2.61111	0.37653	0.36306	0.97542	0.99073	0.00604	0.00142	9.59375	0.00000	0.0		
9	3.00	0.5863	-2.02164	0.37749	0.35479	0.96345	0.98943	0.01119	0.00185	16.56669	0.00000	0.0		
10	3.59	0.4669	-1.44225	0.37884	0.34028	0.94119	0.98679	0.02385	0.00293	30.09834	0.00000	0.0		
11	4.06	0.2946	-0.98239	0.38067	0.32218	0.90704	0.98234	0.05111	0.00531	52.87910	0.00002	0.0		
12	4.35	0.2309	-0.69367	0.38270	0.29732	0.87133	0.97706	0.09194	0.00898	80.62649	0.00004	0.0		
13	4.58	0.1851	-0.46883	0.38540	0.26437	0.83388	0.97026	0.15697	0.01516	118.04472	0.00009	0.0		
14	4.77	0.1270	-0.28996	0.38903	0.23049	0.80462	0.96214	0.25397	0.02346	166.66587	0.00022	0.0		
15	4.90	0.1039	-0.16825	0.39294	0.19277	0.78996	0.95502	0.36123	0.03746	215.64175	0.00041	0.0		
16	5.00	0.0947	-0.06936	0.39760	0.14852	0.82251	0.94890	0.48360	0.05345	268.98880	0.00065	0.0		
17	5.09	0.0856	0.02024	0.40363	0.09994	0.89175	0.94473	0.62263	0.07517	329.99604	0.00101	0.0		
19	5.21	0.0306	0.13153	0.41469	0.06031	1.09396	0.94715	0.80219	0.11523	421.46840	0.00420	0.0		
22	5.28	0.0189	0.20099	0.42424	0.02130	1.30710	0.95839	0.85747	0.14733	474.66329	0.00804	0.0		
29	5.33	0.0004	0.24719	0.43184	0.00255	1.46229	0.97154	0.97069	0.17051	563.18460	0.053418	0.0		
30	5.33	0.0005	0.24755	0.43190										

COLLISION COORDS: X= 0.2476601 Y= 0.4316533 L= 0.5209184 NO. OF STEPS REQUIRED= 30

TRAJECTORY STARTING POSITION IS X= -5.00 Y= 0.33217

STEP	TIME	DTS	XDS	VDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.33223	0.32944	0.99170	0.99187	0.00102	0.00099	0.10864	0.00000	0.0		
3	0.17	0.1944	-4.83337	0.33233	0.32941	0.99136	0.99187	0.00108	0.00099	0.31061	0.00000	0.0		
4	0.36	0.3499	-4.64055	0.33252	0.32929	0.99071	0.99186	0.00121	0.00099	0.70313	0.00000	0.0		
5	0.71	0.5145	-4.29348	0.33287	0.32890	0.98934	0.99182	0.00149	0.00099	1.52050	0.00000	0.0		
6	1.23	0.6131	-3.78325	0.33339	0.32794	0.98670	0.99170	0.00208	0.00102	3.06686	0.00000	0.0		
7	1.84	0.5653	-3.17535	0.33403	0.32622	0.98205	0.99135	0.00329	0.00110	5.73343	0.00000	0.0		
8	2.41	0.5931	-2.61510	0.33470	0.32272	0.97526	0.99070	0.00538	0.00126	9.58974	0.00000	0.0		
9	3.00	0.5834	-2.02787	0.33554	0.31539	0.96310	0.98939	0.00997	0.00165	16.54437	0.00000	0.0		
10	3.58	0.4750	-1.45139	0.33674	0.30209	0.94029	0.98673	0.02124	0.00261	30.01918	0.00000	0.0		
11	4.06	0.2887	-0.98370	0.33840	0.28593	0.90331	0.98205	0.04652	0.00479	53.46506	0.00002	0.0		
12	4.35	0.2259	-0.70087	0.34020	0.26374	0.86419	0.97657	0.08365	0.00808	81.25274	0.00004	0.0		
13	4.57	0.1768	-0.48100	0.34257	0.23502	0.81989	0.96937	0.14324	0.01361	118.71162	0.00009	0.0		
14	4.75	0.1311	-0.31038	0.34566	0.20250	0.77842	0.96066	0.23101	0.02233	166.22120	0.00021	0.0		
15	4.88	0.1002	-0.18497	0.34930	0.16777	0.75317	0.95174	0.34011	0.03418	218.82518	0.00043	0.0		
16	4.98	0.0915	-0.08998	0.35341	0.12592	0.75209	0.94357	0.46307	0.04903	273.69081	0.00070	0.0		
17	5.07	0.0603	-0.00404	0.35878	0.09199	0.78741	0.93602	0.61316	0.06982	337.96534	0.00152	0.0		
19	5.17	0.0289	0.08889	0.36733	0.04743	0.90933	0.93090	0.81276	0.10429	425.25790	0.00467	0.0		
23	5.25	0.0068	0.16403	0.37725	0.00500	1.07507	0.93419	0.97649	0.14335	506.95171	0.02650	0.0		
26	5.26	0.0014	0.17197	0.37849	0.00249	1.08304	0.93504	1.01900	0.14830	529.88968	0.13443	0.0		
27	5.26	0.0016	0.17331	0.37870										

COLLISION COORDS: X= 0.1732322 Y= 0.3784479 L= 0.4293008 NO. OF STEPS REQUIRED= 27

TRAJECTORY STARTING POSITION IS X= -5.00 YO= 0.27404

STEP TIME	DTS	XDS	VDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.27409	0.27178	0.99167	0.99185	0.00084	0.00081	0.00000	0.0	0.00000	0.0
3	0.17	0.1944	-4.83337	0.27417	0.27176	0.99133	0.99184	0.00089	0.00082	0.00000	0.0	0.00000	0.0
4	0.36	0.3499	-4.64056	0.27433	0.27166	0.99067	0.99183	0.00100	0.00082	0.00000	0.0	0.00000	0.0
5	0.71	0.5134	-4.29350	0.27462	0.27134	0.98929	0.99180	0.00123	0.00082	0.00000	0.0	0.00000	0.0
6	1.23	0.6112	-3.78431	0.27505	0.27055	0.98663	0.99167	0.00172	0.00084	0.00000	0.0	0.00000	0.0
7	1.84	0.5635	-3.17826	0.27558	0.26911	0.98194	0.99132	0.00272	0.00091	0.00000	0.0	0.00000	0.0
8	2.40	0.5904	-2.61985	0.27612	0.26624	0.97506	0.99067	0.00445	0.00104	0.00000	0.0	0.00000	0.0
9	2.99	0.5799	-2.03528	0.27682	0.26022	0.96269	0.98935	0.00824	0.00136	0.00000	0.0	0.00000	0.0
10	3.57	0.4840	-1.46220	0.27780	0.24895	0.93921	0.98665	0.01756	0.00153	0.00000	0.0	0.00000	0.0
11	4.05	0.2856	-0.98565	0.27921	0.23521	0.89867	0.98169	0.03948	0.00402	0.00002	0.0	0.00002	0.0
12	4.34	0.2218	-0.70600	0.28071	0.22646	0.85459	0.97587	0.07156	0.00682	0.00004	0.0	0.00004	0.0
13	4.56	0.1673	-0.49033	0.28268	0.19239	0.80103	0.96806	0.12364	0.01153	0.00010	0.0	0.00010	0.0
14	4.73	0.1528	-0.32914	0.28514	0.15855	0.74515	0.95857	0.19875	0.01870	0.00018	0.0	0.00018	0.0
15	4.88	0.0917	-0.18358	0.28887	0.12813	0.68555	0.94540	0.32347	0.03158	0.00054	0.0	0.00054	0.0
16	4.97	0.0841	-0.09735	0.29234	0.09072	0.65532	0.93468	0.44351	0.04515	0.00086	0.0	0.00086	0.0
18	5.10	0.0411	0.02149	0.30005	0.04045	0.66113	0.91646	0.70127	0.07869	0.00312	0.0	0.00312	0.0
25	5.20	0.0013	0.10637	0.30922	0.00211	0.71762	0.90413	0.94595	0.12084	0.00000	0.0	0.00000	0.0
26	5.20	0.0020	0.10754	0.30937					0.508	0.11663	0.0	0.0	0.0

COLLISION COORDS: X= 0.1069864 Y= 0.3090961 L= 0.3332461 NO. OF STEPS REQUIRED= 26

TRAJECTORY STARTING POSITION IS X= -5.00 YO= 0.19930

STEP TIME	DTS	XDS	VDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.19934	0.19766	0.99164	0.99182	0.00061	0.00059	0.00000	0.0	0.00000	0.0
3	0.17	0.1944	-4.83337	0.19940	0.19764	0.99130	0.99181	0.00065	0.00059	0.00000	0.0	0.00000	0.0
4	0.36	0.3499	-4.64057	0.19952	0.19757	0.99064	0.99181	0.00073	0.00060	0.00000	0.0	0.00000	0.0
5	0.71	0.5124	-4.29352	0.19972	0.19733	0.98924	0.99177	0.00090	0.00060	0.00000	0.0	0.00000	0.0
6	1.22	0.6094	-3.78537	0.20004	0.19676	0.98655	0.99164	0.00126	0.00062	0.00000	0.0	0.00000	0.0
7	1.83	0.5616	-3.18115	0.20042	0.19573	0.98182	0.99129	0.00199	0.00066	0.00000	0.0	0.00000	0.0
8	2.40	0.5878	-2.62457	0.20082	0.19363	0.97486	0.99063	0.00325	0.00076	0.00000	0.0	0.00000	0.0
9	2.98	0.5766	-2.04263	0.20132	0.18927	0.96227	0.98930	0.00601	0.00099	0.00000	0.0	0.00000	0.0
10	3.56	0.4673	-1.47287	0.20203	0.18136	0.93813	0.98658	0.01281	0.00156	0.00000	0.0	0.00000	0.0
11	4.03	0.3061	-1.01284	0.20301	0.17066	0.89712	0.98172	0.02811	0.00285	0.00002	0.0	0.00002	0.0
12	4.33	0.2212	-0.71325	0.20417	0.15646	0.84488	0.97521	0.05372	0.00505	0.00004	0.0	0.00004	0.0
13	4.55	0.1539	-0.49842	0.20564	0.13873	0.78034	0.96661	0.09447	0.00864	0.00010	0.0	0.00010	0.0
14	4.72	0.1298	-0.34083	0.20745	0.11651	0.70674	0.95589	0.15406	0.01413	0.00022	0.0	0.00022	0.0
15	4.85	0.1185	-0.21752	0.20977	0.08517	0.62541	0.94256	0.23945	0.02243	0.00041	0.0	0.00041	0.0
16	4.97	0.0598	-0.10687	0.21316	0.06313	0.53141	0.92413	0.37548	0.03666	0.00137	0.0	0.00137	0.0
18	5.07	0.0343	-0.01071	0.21817	0.02415	0.44399	0.90017	0.58226	0.06053	0.00409	0.0	0.00409	0.0
22	5.14	0.0025	0.05207	0.22331	0.00084	0.40174	0.87958	0.75434	0.08705	0.00000	0.0	0.00000	0.0
23	5.15	0.0040	0.05427	0.22353					0.492	0.11797	0.0	0.0	0.0

COLLISION COORDS: X= 0.0525900 Y= 0.2232139 L= 0.2314103 NO. OF STEPS REQUIRED= 23

TRAJECTORY STARTING POSITION IS X= -5.00 Y0= 0.10795

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MDD	HIST/RHS	USTAB	VSTAB
2	0.06	0.1080	-4.94049	0.10797	0.10706	0.99161	0.99179	0.00033	0.00032	0.10918	0.00000	0.0		
3	0.17	0.1944	-4.83338	0.10801	0.10705	0.99127	0.99179	0.00035	0.00032	0.31216	0.00000	0.0		
4	0.36	0.3499	-4.64058	0.10807	0.10701	0.99061	0.99178	0.00040	0.00032	0.70678	0.00000	0.0		
5	0.71	0.5116	-4.29354	0.10818	0.10689	0.98920	0.99174	0.00049	0.00033	1.52904	0.00000	0.0		
6	1.22	0.6080	-3.78620	0.10835	0.10658	0.98650	0.99161	0.00068	0.00033	3.07658	0.00000	0.0		
7	1.83	0.5602	-3.18343	0.10856	0.10602	0.98173	0.99126	0.00108	0.00036	5.73652	0.00000	0.0		
8	2.39	0.5857	-2.62828	0.10878	0.10488	0.97470	0.99060	0.00176	0.00041	9.57799	0.00000	0.0		
9	2.98	0.5740	-2.04839	0.10905	0.10252	0.96194	0.98927	0.00326	0.00054	16.47521	0.00000	0.0		
10	3.55	0.4524	-1.48121	0.10943	0.09842	0.93726	0.98652	0.00695	0.00085	29.77600	0.00000	0.0		
11	4.00	0.2940	-1.03589	0.10994	0.09303	0.89626	0.98178	0.01493	0.00151	51.93415	0.00002	0.0		
12	4.30	0.2259	-0.74806	0.11052	0.08546	0.84418	0.97556	0.02781	0.00260	80.26203	0.00004	0.0		
13	4.52	0.1754	-0.52863	0.11129	0.07522	0.77326	0.96674	0.04964	0.00448	119.20374	0.00009	0.0		
14	4.70	0.1369	-0.35998	0.11232	0.06197	0.68076	0.95459	0.08483	0.00762	170.69586	0.00021	0.0		
15	4.84	0.1073	-0.23031	0.11367	0.04563	0.56671	0.93855	0.13820	0.01260	235.47969	0.00047	0.0		
16	4.94	0.0661	-0.13059	0.11538	0.03126	0.43462	0.91836	0.21404	0.02007	312.69240	0.00127	0.0		
18	5.04	0.0278	-0.04022	0.11797	0.01338	0.26180	0.88879	0.33976	0.03322	418.72403	0.00526	0.0		
23	5.11	0.0028	0.01358	0.12041	-0.00002	0.15997	0.86289	0.46208	0.04713	489.73179	0.07123	0.0		
24	5.11	0.0045	0.01603	0.12054										

COLLISION COORDS: X= 0.0146779 Y= 0.1203817 L= 0.1215945 NO. OF STEPS REQUIRED= 24

BETA0 (MAX LOCAL CE) IS 89.8% AT A DISTANCE OF 0.0 FROM THE NOSE
THE TOTAL COLLISION EFFICIENCY IS 81.4%

ENDPT.	X COORD.	Y COORD.	DIST.	FROM NOSE	COLL. EFF.	ENDPT.	X COORD.	Y COORD.	DIST.	FROM NOSE	COLL. EFF.
0	0	0	0	0	0.8980	0	0	0	0	0.37091	0.6215
0	0.0048	0.02181	0.02181	0.02181	0.8970	0	0.13136	0.33780	0.33780	0.39272	0.5894
0	0.0190	0.04358	0.04358	0.04358	0.8941	0	0.14645	0.35355	0.35355	0.41454	0.5557
0	0.0428	0.06526	0.06526	0.06526	0.8891	0	0.16220	0.36864	0.36864	0.43636	0.5204
0	0.0760	0.08682	0.08682	0.08682	0.8823	0	0.17861	0.38302	0.38302	0.45817	0.4837
0	0.1185	0.10822	0.10822	0.10822	0.8734	0	0.19562	0.39668	0.39668	0.47999	0.4456
0	0.1704	0.12941	0.12941	0.12941	0.8626	0	0.21321	0.40958	0.40958	0.50181	0.4061
0	0.2314	0.15035	0.15035	0.15035	0.8498	0	0.23135	0.42170	0.42170	0.52362	0.3652
0	0.3015	0.17101	0.17101	0.17101	0.8352	0	0.25000	0.43301	0.43301	0.54544	0.3229
0	0.3806	0.19134	0.19134	0.19134	0.8187	0	0.26913	0.44351	0.44351	0.56726	0.2793
0	0.4685	0.21131	0.21131	0.21131	0.8002	0	0.28869	0.45315	0.45315	0.58907	0.2344
0	0.5649	0.23087	0.23087	0.23087	0.7799	0	0.30866	0.46194	0.46194	0.61089	0.1881
0	0.6699	0.25000	0.25000	0.25000	0.7578	0	0.32899	0.46985	0.46985	0.63271	0.1405
0	0.7830	0.26865	0.26865	0.26865	0.7340	0	0.34965	0.47686	0.47686	0.65452	0.0915
0	0.9042	0.28679	0.28679	0.28679	0.7084	0	0.37059	0.48296	0.48296	0.67634	0.0411
0	1.0332	0.30438	0.30438	0.30438	0.6811	0	0.39178	0.48815	0.48815	0.69816	0.0
0	1.1698	0.32139	0.32139	0.32139	0.6521	0	0.41318	0.49240	0.49240		

THE ACCRETED AREA FOR LAYER 1 IS 0.09271
THE ACCUMULATED ACCRETED AREA IS 0.09271

APPENDIX C: PROGRAM LISTING

This appendix contains the program listing as written in Fortran. The program listing has been carefully annotated. However, should difficulties be encountered in attempting to run the program as listed, the authors are prepared to offer advice and assistance.

```

1      C
2      C WRITTEN BY: M. OLESKIW  ON:790526  LAST MODIFIED:801228
3      C
4      C CALCULATE POTENTIAL FLOW ABOUT AN ARBITRARILY SHAPED AEROFOIL;
5      C CALCULATE A SERIES OF DROPLET TRAJECTORIES AND
6      C DETERMINE THE COLLISION LOCATIONS; FIND THE RESULTING COLLISION
7      C EFFICIENCY AND ACCRETE A LAYER OF ICE.
8      C REPEAT THE PROCESS FOR A PREDETERMINED NUMBER OF STEPS.
9      C
10     C INTERNAL SUBROUTINES:
11     C   COORDS: CALCULATE THE UPPER AND LOWER SFC. COORDINATES
12     C     OF THE AEROFOIL.
13     C   POT1: SOLVE FOR SFC. VORTEX DENSITY ON 1 ELEMENT AEROFOIL
14     C     IN POTENTIAL FLOW, GIVEN COORDINATES OF AEROFOIL SFC.
15     C   STRMFN: CALCULATE STREAMFUNCTION ON A GRID ABOUT AN AEROFOIL
16     C     SECTION GIVEN THE SFC. VORTICITY DENSITY ON THE AEROFOIL
17     C     AND PLOT THE FLOW USING VELOCITY VECTORS.
18     C   AIRPLT: PLOTS AEROFOIL OUTLINE WITHIN WINDOW
19     C   SFC: CALCULATE Y VALUES AND THE LENGTH FROM THE NOSE ON THE
20     C     SFC. OF THE AEROFOIL BY A CUBIC SPLINE INTERPOLATION.
21     C   SFCLEN: CALCULATES THE LENGTH ALONG A SEGMENT OF A CUBIC SPLINE.
22     C   CE: CALCULATE AND PLOT COLLISION EFFICIENCY OF ARBITRARY
23     C     AEROFOIL BY DETERMINING A SET OF IMPACTING TRAJECTORIES.
24     C   PLTSZ: DETERMINE PARAMETERS NECESSARY FOR SCALING OF A PLOT
25     C     AND ITS AXES.
26     C   ICING: CALCULATE AMOUNT OF ACCRETION AND DETERMINE A NEW SET
27     C     OF AEROFOIL SFC. ELEMENT ENDPOINTS AFTER DETERMINING THE
28     C     AEROFOIL NOSE LOCATION.
29     C   GROWTH: PLOTS SUCCESSIVE AEROFOIL OUTLINES WITHIN VIEW WINDOW.
30     C   TRAJEC: CALCULATES TRAJECTORIES OF DROPLETS IN POTENTIAL FLOW
31     C     ABOUT AN AEROFOIL.
32     C   ACCN: CALCULATES RHS OF NON-DIMENSIONAL EQNS. OF MOTION.
33     C   AIRVEL: CALCULATES THE AIR VELOCITY COMPONENTS AT A GIVEN
34     C     LOCATION.
35     C   DRAG: CALCULATES THE REYNOLDS NUMBER AND DRAG COEFFICIENT
36     C     OF THE DROPLET AT ANY STEP ALONG ITS TRAJECTORY.
37     C   HIST: DETERMINES VALUE OF INTEGRAL IN HISTORY TERM.
38     C   RK4: THE RUNGE-KUTTA-FEHLBERG 4TH ORDER ODE INTEGRATION TECHNIQUE
39     C   RK4: THE RUNGE-KUTTA 4TH ORDER ODE INTEGRATION TECHNIQUE.
40     C   PC4: THE PREDICTOR-CORRECTOR 4TH ORDER ODE INTEGRATION TECHNIQUE
41     C INTERNAL FUNCTIONS:
42     C   NSURF: CALCULATES THE UNROTATED X VALUE OF A POINT ON THE
43     C     ACCRETED AEROFOIL SFC. BASED UPON THE COLLISION EFFICIENCY,
44     C     DIRECTION OF GROWTH, AND OLD AEROFOIL (ROTATED) SFC. POSITION
45     C
46     C EXTERNAL SUBROUTINES:
47     C   IMSL: (INTERNATIONAL MATHEMATICAL AND SCIENTIFIC LIBRARY)
48     C   LEQ1F: SOLVES SYSTEM OF EQNS.
49     C   ICSICU: CUBIC SPLINE INTERPOLATION
50     C   ZXGSN: GOLDEN SECTION SEARCH METHOD FOR FINDING FN. MINIMUM.
51     C
52     C   SSPLIB: (IBM SUPPLIED SCIENTIFIC SUBROUTINE LIBRARY)
53     C   DELI1: INCOMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND.
54     C   DELI2: INCOMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND.
55     C   DCEL1: COMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND.
56     C   DCEL2: COMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND.
57     C
58     C INPUT/OUTPUT DEVICE ASSIGNMENTS:
59     C   3: DATA READ BY SUBPROGRAM PLTSZ TO SCALE PLOTS.

```

```

60 C 4 PROGRAM INPUT PARAMETERS (DESCRIBED BELOW).
61 C 5 INPUT CRT DEVICE FOR CONTROL OF PROGRAM.
62 C 6 OUTPUT CRT DEVICE FOR MONITORING OF PROGRAM.
63 C 7 OUTPUT HARDCOPY DEVICE FOR PRINTED OUTPUT.
64 C 9 OUTPUT FILE FOR STORAGE OF PLOT DESCRIPTION (CALCOMP FORMAT).
65 C
66 C PROGRAM INPUT PARAMETERS:
67 C TO BE READ IN FROM INPUT DEVICE 4. EACH GROUP OF PARAMETERS
68 C IS TO BE READ FROM THE SAME LINE (CARD) USING THE SPECIFIED
69 C FORMAT. EACH DATA LINE PRECEDED BY A DESCRIPTIVE REMINDER LINE.
70 C SEE EXAMPLE FOR DETAILS.
71 C
72 C NEF=NO. OF ELEMENT ENDPTS. ON FRONT HALF OF AEROFOIL (I4)
73 C NEB=NO. OF ELEMENT ENDPTS. ON BACK HALF OF AEROFOIL
74 C (INCLUDES THE MIDPOINT ENDPT. (AT THETA=90)) (I4)
75 C NIF=NO. OF SPLINE ENDPTS/ELEMENT ENDPT. (FRONT HALF) (I4)
76 C
77 C ALPHA=ANGLE OF ATTACK IN DEGREES (F6.0)
78 C TYPE=AEROFOIL TYPE (I5)
79 C -1 ANALYTICAL CYLINDER
80 C 0 NACA RAZOR
81 C 1 CYLINDER (VORTEX SHEETS)
82 C THICK=THICKNESS OF AEROFOIL IN PERCENT (F6.0)
83 C XMIN=
84 C XMAX= VIEWPORT SIZE IN X (2F5.0)
85 C YMIN=
86 C YMAX= VIEWPORT SIZE IN Y (2F5.0)
87 C XZ= VELOCITY VECTOR GRID SIZE IN X (I3)
88 C YZ= VELOCITY VECTOR GRID SIZE IN Y (I3)
89 C ANAL=0 ESTIMATE SEGMENT LENGTH BY NUMERICAL APPROXIMATION. (I5)
90 C 1 DETERMINE SEGMENT LENGTH BY ANALYTICAL METHOD.
91 C
92 C PLTFAC= PLOT REDUCTION OR EXPANSION FACTOR FOR ALL PLOTS (F7.2)
93 C TRJPLA=PLOT TRAJECTORIES (0 OR 1) (I7)
94 C YOL=PLOT THE YO VS L GRAPH (0, 1, OR 2) (2 PLOTS AT HALF PAGE SIZE) (I4)
95 C CEL=PLOT THE CE VS L GRAPH (0, 1, OR 2) (2 PLOTS AT HALF PAGE SIZE) (I4)
96 C CEX=PLOT THE CE VS X GRAPH (0, 1, OR 2) (2 PLOTS AT HALF PAGE SIZE) (I4)
97 C ICEPLA=PLOT AEROFOIL AND ICE LAYERS (0 OR 1) (I7)
98 C LYRMAX=MAX. NUMBER OF LAYERS TO ACCRETE (I7)
99 C CETOL=CRITERION (FOR CHANGE IN CE BETWEEN ENDPTS.) TO DETERMINE
100 C WHETHER OR NOT TO CREATE NEW ENDPTS. (F6.2)
101 C ICE=FRACTION OF CHORD LENGTH TO BE ACCRETED PER LAYER ASSUMING
102 C A COLLISION EFFICIENCY OF 100% (F7.2)
103 C
104 C UINF=FREESTREAM VELOCITY (M/S) (F6.0)
105 C C=CHORD LENGTH (M) (F6.0)
106 C PINF=FREESTREAM PRESSURE (KPA) (F6.0)
107 C TINF=FREESTREAM TEMPERATURE (C) (F6.0)
108 C RD=DROPLET RADIUS (MICROMETERS) (F6.0)
109 C A1=
110 C B1=PARAMETERS FOR PREDICTOR-CORRECTOR FORMULAE (2D10.0)
111 C
112 C CDS:DRAG COEFFICIENT FORMULATION: (I4)
113 C =0 ABRAHAM (1970)
114 C =1: RE < 0.01: STOKES DRAG
115 C 0.01 < RE < 5: SARTOR AND ABBOTT (1975)
116 C RE > 5: ABRAHAM (1970)
117 C =2: LANGMUIR AND BLODGETT (1945)
118 C TRJPRA=PRINT TRAJECTORY INFO (0 OR 1) (I7)
119 C PRINTI=NO. OF STEPS AT WHICH TO PRINT TRAJECTORY INFO
120 C WITHIN VIEWPORT. (I7)
121 C PLOTI=NO. OF STEPS AT WHICH TO PLOT TRAJECTORY WITHIN VIEWPORT.
122 C (I6)
123 C PRINTO=NO. OF STEPS AT WHICH TO PRINT TRAJECTORY INFO
124 C OUTSIDE VIEWPORT. (I7)
125 C CRIT=CRITERION (EXPRESSED AS % OF DROPLET RADIUS) USED
126 C TO INDICATE SUFFICIENTLY CLOSE DROPLET APPROACH
127 C TO DENOTE COLLISION (F5.0)
128 C BETAO=ESTIMATED LOCAL COLLISION EFFICIENCY AT STAGNATION PT.
129 C (F6.0)
130 C
131 C NTRAJU=MANUAL MODE: NO. OF TRAJECTORIES PRINTED/PLOTTED (I7)
132 C =AUTO MODE: NO. OF TRAJECTORIES DESIRED ON UPPER SFC.
133 C NTRAJL=AUTO MODE: NO. OF TRAJECTORIES DESIRED ON LOWER SFC. (I7)

```

```

134 C AT=0: START TRAJECTORIES AS SPECIFIED BY INPUT TERMINAL. (I3)
135 C 1: AUTOMATICALLY DETERMINE TRAJECTORY STARTING POINTS
136 C AFTER FIRST ONE FOR EACH SFC.
137 C BOTH=0: SYMMETRICAL AEROFOIL AT 0 DEGREES ATTACK -
138 C CALCULATE TRAJECTORIES FOR UPPER SFC. ONLY.
139 C 1: CALCULATE TRAJECTORIES FOR BOTH SFCS. (I5)
140 C EQN=0: EQN. OF MOTION INCLUDES TERMS A AND B (NO INDUCED
141 C MASS OR BUOYANCY) (I4)
142 C 1: EQN. OF MOTION INCLUDES TERMS APRIME AND BPRIME
143 C 2: EQN. OF MOTION INCLUDES TERMS APRIME, BPRIME, AND
144 C CPRIME (HISTORY TERM)
145 C PC=INTEGRATE BY RUNGE-KUTTA (0) OR PREDICTOR-CORRECTOR (1)
146 C (AFTER FIRST 3 INTERVALS) OR RUNGE-KUTTA-FEHLBERG (2) (I3)
147 C DTS=NON-DIM. INITIAL TIME STEP (F6.0)
148 C EPS= FOR ODE INTEGRATION TECHNIQUE RUNGE-KUTTA-FEHLBERG:
149 C LOCAL ERROR DIVIDED BY LOCAL STEP SIZE. (D8.0)
150 C ACN=0: DROPLET INITIAL VELOCITY VECTOR SLIGHTLY GREATER THAN
151 C THAT OF THE AIR AT THAT POINT. (I4)
152 C 1: DROPLET INITIAL ACCELERATION WEIGHTED BY CHANGE IN
153 C POTENTIAL FLOW FIELD.
154 C
155 C XO=X (UPSTREAM) COORD. FOR TRAJECTORY STARTING PTS. (F10.0)
156 C
157 C YO=Y (OFF AXIS) COORDS FOR TRAJECTORY STARTING POINTS. (F10.0)
158 C INPUT ONE FOR EACH SFC. (AUTO-TRAJECTORY MODE), OR FOR ALL
159 C THE TRAJECTORIES DESIRED OTHERWISE.
160 C
161 5 FORMAT(/,3I4)
162 10 FORMAT(/,F6.0,I5,F6.0,4F5.0,2I3,I5)
163 20 FORMAT(/,F7.2,I7,3I4,2I7,F6.2,F7.2)
164 30 FORMAT('OTHE ACCRETED AREA FOR LAYER',I3,' IS',F10.5,/,
165 ' THE ACCUMULATED ACCRETED AREA IS',F10.5)
166 C
167 DOUBLE PRECISION ALPHA, XE(101), YE(101), LEN, YNNUR, XNNLR,
168 PI, X, DFLOAT, LU(101), LL(101), XS, CETOL, ICE, YNNLR, ACCRU, ACCRL,
169 XNP, YNP, XURTL, XLRTL, ACCR, ACCRT, INTU, INTL, XNNUR,
170 XU(101), YU(101), XL(101), YL(101), THICK, S30, C30,
171 XLR(101), YLR(101), DSORT, XN, YN, BPARU(4), BPARL(4), CU(100,3),
172 CL(100,3), XUR(101), YUR(101), ALPHAR, THETA, INTUP, INTLP
173 C
174 REAL XMAX, XMIN, YMIN, YMAX, PLTFAC
175 INTEGER I, J, TYPE, XZ, YZ, TRUPLA, NCOU, NCOL, IERU, IERL, LYRM1,
176 PLT, LAYER, LYRMAX, NCOL1, L, YOL, ICEPLA, AT, BOTH, FAIL, ANAL,
177 ATYPE, IABS, IU(51), IL(51), NEB, NEF, NIF, NIFP1, CEX, II, IJ, NEU, NEL
178 C
179 COMMON ALPHAR, PI/AERO1/XE, YE/NOSE/XN, YN/FOIL/XUR, YUR,
180 XLR, YLR/LG/LU, LL/LA/ANAL/AERO3/NCOU, NCOL/ROTP/C30, S30
181 /GRID/XMIN, XMAX, YMIN, YMAX, XZ, YZ/SFCS/XU, YU, XL, YL
182 /SPLINE/CU, CL/AERO4/NEU, NEL/ENDS/IU, IL
183 /NNOSE/XNP, YNP, XURTL, XLRTL
184 C
185 C INPUT PARAMETERS:
186 READ(4,5) NEF, NEB, NIF
187 READ(4,10) ALPHA, TYPE, THICK, XMIN, XMAX, YMIN, YMAX, XZ, YZ, ANAL
188 READ(4,20) PLTFAC, TRUPLA, YOL, CEL, CEX, ICEPLA, LYRMAX, CETOL, ICE
189 PI=3.14159265358979324
190 C INITIALIZE PARAMETERS
191 ALPHAR=ALPHA*PI/1.802
192 NCOU=NEF+NEB
193 NCOL=NCOU
194 YN=0.0D0
195 YN=0.0D0
196 ACCRT=0.0D0
197 NIFP1=NIF+1
198 IU=1
199 ATYPE=IABS(TYPE)
200 C
201 C CALCULATE AEROFOIL COORDS
202 C UPPER AND LOWER COORDS FOR LEFT HALF OF AEROFOIL
203 DO 110 I=1, NEF
204 IU(I)=IU
205 IL(I)=IU
206 DO 140 J=1, NIFP1
207 THETA=PI/2.0D0*DFLOAT((I-1)*NIFP1+J-1)/DFLOAT(NEF*NIFP1)

```

```

208          CALL COORDS(TYPE,THICK,THETA,X,YU(IJ),YL(IJ))
209          XU(IJ)=X
210          XL(IJ)=X
211          IJ=IJ+1
212      140      CONTINUE
213      110      CONTINUE
214      C UPPER AND LOWER COORDS. FOR RIGHT HALF OF AEROFOIL.
215          DO 150 I=1,NEB
216          THETA=PI/2.DO*(1.DO+DFLOAT(I-1)/DFLOAT(NEB-1))
217          CALL COORDS(TYPE,THICK,THETA,X,YU(IJ),YL(IJ))
218          XU(IJ)=X
219          XL(IJ)=X
220          IU(NEF+I)=IJ
221          IL(NEF+I)=IJ
222          IJ=IJ+1
223      150      CONTINUE
224          NEU=IJ-1
225          NEL=NEU
226          LAYER=1
227      C
228      C TRANSFORM THESE COORDS. TO ONE VECTOR OF LENGTH NCOU+NCOL-1
229      C IN CLOCKWISE ORDER. WITH XE(1)=XE(NCOL+NCOU-1) - THE LEADING PT.
230      100      DO 102 I=1,NCOU
231          II=IU(I)
232          XE(I)=XU(II)
233          YE(I)=YU(II)
234      102      CONTINUE
235          NCOL1=NCOL-1
236          DO 104 I=1,NCOL1
237          J=NCOU+NCOL-I
238          II=IL(I)
239          XE(J)=XL(II)
240          YE(J)=YL(II)
241      104      CONTINUE
242      C
243      C ROTATE UPPER & LOWER SFCS. BY 30 DEG. ABOUT NOSE IN ORDER
244      C TO FIT CUBIC SPLINES
245      C - SEE KENNEDY & MARSDEN (1976)
246      C DO NOT ROTATE IF AEROFOIL IS A CYLINDER.
247          IF(ATYPE.EQ.1)GOTO 200
248          S30=5.D-1
249          C30=DSQRT(3.DO)/2.DO
250          GOTO 210
251      200      S30=0.DO
252          C30=1.DO
253      210      DO 320 I=1,NEU
254          XUR(I)=(XU(I)-XU(1))*C30+(YU(I)-YU(1))*S30
255          YUR(I)=(YU(I)-YU(1))*C30-(XU(I)-XU(1))*S30
256      320      CONTINUE
257          DO 330 I=1,NEL
258          XLR(I)=(XL(I)-XL(1))*C30-(YL(I)-YL(1))*S30
259          YLR(I)=(YL(I)-YL(1))*C30+(XL(I)-XL(1))*S30
260      330      CONTINUE
261      C
262      C SET PARAMETERS FOR SPLINE FITTING
263          IF(ATYPE.EQ.1)GOTO 220
264          BPARU(1)=1.DO
265          BPARU(2)=6.DO/(XUR(2)-XUR(1))*((YUR(2)-YUR(1))/(XUR(2)-XUR(1))-DSQRT(3.DO))
266          BPARU(3)=0.DO
267          BPARU(4)=0.DO
268          BPARL(1)=1.DO
269          BPARL(2)=6.DO/(XLR(2)-XLR(1))*((YLR(2)-YLR(1))/(XLR(2)-XLR(1))-DSQRT(3.DO))
270          BPARL(3)=0.DO
271          BPARL(4)=0.DO
272          GOTQ 230
273      220      BPARU(1)=0.DO
274          BPARU(2)=0.DO
275          BPARU(3)=0.DO
276          BPARU(4)=0.DO
277          BPARL(1)=0.DO
278          BPARL(2)=0.DO
279          BPARL(3)=0.DO
280          BPARL(4)=0.DO
281

```

```

282      BPARL(4)=0.DO
283      C FIT CUBIC SPLINES TO EACH SFC.
284      230  CALL ICSICU(XUR,YUR,NEU,BPARU,CU,100,IERU)
285          CALL ICSICU(XLR,YLR,NEL,BPARL,CL,100,IERL)
286      C
287      C CALCULATE INTEGRAL OF UPPER AND LOWER SFC. PROFILES.
288      C FIND THE LENGTHS FROM THE NOSE TO VARIOUS ENDPNTS.
289          LU(1)=0.DO
290          LL(1)=0.DO
291          INTU=0.DO
292          INTL=0.DO
293          DO 340 I=2,NEU
294              XS=XUR(I)-XUR(I-1)
295              CALL SFCLN(XS,LEN,CU(I-1,3),CU(I-1,2),CU(I-1,1))
296              LU(I)=LU(I-1)+LEN
297              INTU=INTU+(((CU(I-1,3)*XS/4.DO+CU(I-1,2)/3.DO)*XS
298                  +CU(I-1,1)/2.DO)*XS+YUR(I-1))*XS
299      340  CONTINUE
300          DO 350 I=2,NEL
301              XS=XLR(I)-XLR(I-1)
302              CALL SFCLN(XS,LEN,CL(I-1,3),CL(I-1,2),CL(I-1,1))
303              LL(I)=LL(I-1)+LEN
304              INTL=INTL+(((CL(I-1,3)*XS/4.DO+CL(I-1,2)/3.DO)*XS
305                  +CL(I-1,1)/2.DO)*XS+YLR(I-1))*XS
306      350  CONTINUE
307      IF(LAYER.EQ.1)GOTO 400
308      XNNUR=(XN-XNP)*C30+(YN-YNP)*S30
309      YNNUR=(YN-YNP)*C30-(XN-XNP)*S30
310      C ACCRETION AREA FOR UPPER LAYER.
311      ACCRU=INTU-INTUP+YNNUR*XURTL-P-XNNUR*YNNUR/2.DO
312      IF(BOTH.EQ.1)GOTO 410
313      ACCR=2.DO*ACCRU
314      GOTO 420
315      410  XNNLR=(XN-XNP)*C30-(YN-YNP)*S30
316          YNNLR=(YN-YNP)*C30+(XN-XNP)*S30
317      C ACCRETION AREA FOR LOWER LAYER
318      ACCRL=INTLP-INTL-YNNLR*XLRTL-P+XNNLR*YNNLR/2.DO
319      ACCR=ACCRU+ACCR
320      420  ACCRT=ACCR+ACCR
321          LYRM1=LAYER-1
322          WRITE(6,30)LYRM1,ACCR,ACCR
323          WRITE(7,30)LYRM1,ACCR,ACCR
324      400  INTUP=INTU
325          INTLP=INTL
326      C
327          IF(LAYER.GT.LYRMAX AND ICEPLA.EQ.1)GOTO 121
328          IF(LAYER.GT.LYRMAX AND ICEPLA.EQ.2)GOTO 130
329          IF(TYPE.EQ.0)CALL POT1
330          PLT=TRUPLA*YOL*CEL*CFX+ICEPLA
331          IF(PLT.EQ.0)GOTO 120
332          IF(LAYER.GT.1)GOTO 125
333      C
334      C OPEN PLOTTING
335          CALL PLOTS
336          CALL METRIC(1)
337          CALL ORGE(5,0,5,0,5,0)
338          CALL FACTOR(PLTFAC)
339      C
340          IF(TRUPLA.EQ.0)GOTO 121
341      C PLOT AEROFOIL OUTLINE AND VELOCITY VECTORS
342          130  CALL SIRMN(TYPE)
343          121  CALL AIRPLT(LAYER,TRUPLA,LYRMAX)
344              IF(LAYER.GT.LYRMAX)GOTO 370
345      C
346      C CALCULATE DROPLET TRAJECTORIES
347          120  IF(LAYER.EQ.1)CALL TRAJECTYPE,TRUPLA,THICK,AT,POT1
348              IF(LAYER.GT.1)CALL TRAJEK
349              IF(AT.EQ.0)GOTO 360
350      C
351      C CALCULATE COLLISION EFFICIENCY
352          CALL GEEO(CEL,CFX,PLTFAC,THICK,LAYER)
353      C
354      C ACCRET ICE AND FIND NEW AEROFOIL SHAPE
355          CALL ICING(ETOL,ICE,BOTH,FAT1)

```

```

356      IF(LAYER.EQ.LYRMAX.AND.ICEPLA.EQ.O)GOTO 360
357      IF(FAIL.EQ.1)GOTO 360
358      LAYER=LAYER+1
359      GOTO 100
360      C
361      C PLOT SUCCESSIVE AEROFOIL SHAPES ON ONE PLOT.
362      370 CALL GROWTH(ICEPLA,LYRMAX,PLTFAC,TRUPLA)
363      360 IF(PLT.NE.O)CALL PLOT(O.,O.,999)
364      STOP
365      END
366      C
367      C
368      SUBROUTINE COORDS(TYPE,T,THETA,X,YU,YL)
369      C
370      C WRITTEN BY: M. OLESKIW ON:790928 LAST MODIFIED:801022
371      C
372      C CALCULATE THE UPPER AND LOWER SFC. COORDINATES OF THE AEROFOIL.
373      C
374      DOUBLE PRECISION X,YU,YL,DSQRT,B,C,T,THETA,DCOS,EIM2,EIM1,
375      EI,DABS,A,B,DSIN,XI,ETA,E,TC
376      C
377      INTEGER TYPE,ATYPE,IABS
378      C
379      COMMON /JOUK1/A,B,EI
380      C
381      C IN TYPE=AEROFOIL TYPE
382      C IN T=AEROFOIL THICKNESS IN PERCENT
383      C IN THETA=ANGLE FROM X AXIS
384      C OUT X=X-COORD. OF AEROFOIL SFC.
385      C OUT YU=
386      C OUT YL= UPPER & LOWER Y-COORDS. OF AEROFOIL SFC.
387      C
388      C
389      ATYPE=IABS(TYPE)
390      IF(ATYPE.EQ.1)GOTO 101
391      C
392      C CALCULATE THE SHAPE OF A NACA AEROFOIL MODIFIED TO HAVE A RAZOR-LIKE
393      C TRAILING EDGE BY REMOVING A LINEARLY INCREASING AMOUNT
394      C FROM X=0.3 TO X=1.0
395      C REF GREGORY, N. & P.G. WILBY (1973), A.R.C. PAPER #1261
396      C ABBOTT, I.H. & A.E. VON DOENHOFF (1959), THEORY OF WING SECTIONS,
397      C TL 672 A12 1959, P113 & 321
398      C
399      C CALCULATE AEROFOIL X & Y COORDS. FOR EACH SFC.
400      X=(1.DO-DCOS(THETA))/2.DO
401      B=0.2969DO*DSQRT(X)-0.126DO*X-0.3516DO*X*X
402      C=0.2843DO*X**3-0.1015DO*X**4
403      YU=T/O.2D2*(B+C)
404      IF(X.GT.O.3DO)YU=YU-(X-O.3DO)*2.1D-3*T/O.7DO/O.2D2
405      IF(X.GT.O.9999999999)YU=O.DO
406      YL=-YU
407      RETURN
408      C
409      C CALCULATE THE X & Y COORDS. OF A CYLINDER
410      101 X=(1.DO-DCOS(THETA))/2.DO
411      YU=DSQRT(O.25DO*(X-O.5DO)*(X-O.5DO))
412      IF(X.GT.O.9999999999)YU=O.DO
413      YL=-YU
414      RETURN
415      C
416      C
417      C
418      SUBROUTINE PLOT1
419      C
420      C WRITTEN BY: M. OLESKIW ON:781129 LAST MODIFIED:801227
421      C
422      C SOLVE FOR SURFACE VORTEX DENSITY ON 1 ELEMENT AEROFOIL IN POTENTIAL
423      C FLOW, GIVEN COORDS. OF AEROFOIL SURFACE
424      C REF KENNEDY, J.L. & D.J. MARSDEN (1976), CAN. AERO. & SPACE JOUR.,
425      C V22, #5, P243-256
426      C SUBROUTINE LEQ11F OF *IMSLOP1B LINEAR EQN. SOLN., FULL STORAGE
427      C MODE, SPACE ECONOMIZER SOLN
428      C
429      DOUBLE PRECISION XE(101),YE(101),XC(101),YC(101),R(101),

```

```

430      DATAN,DABS,DSIGN,DLOG,SI(100),CO(100),PI,CL,
431      K(101,101),WKAREA(101),D(100),XT,YT,DE,DELTA,
432      DXC,DYC,B,A,R1S,R2S,R3S,T3,T1,T2,ALPHAR,DCOS,DSIN,DSQRT
433      C
434      INTEGER N,N1,J,J1,IDGT,IER,I,NCOU,NCOU1,NCOL,JJ
435      C
436      COMMON ALPHAR,PI/AERO1/XE,YE/AERO3/NCOU,NCOL/AERO2/XC,YC,R,D,SI,CO
437      C
438      10  FORMAT('OFOR EQN. SOLN. IER=',I3,/,
439      'OTHE POTENTIAL FLOW LIFT COEFFICIENT IS',F9.5)
440      20  FORMAT('OCONTROL PT.  X COORD.  Y COORD.  SFC. VEL.  ')
441      30  FORMAT(' ',I6,5X,2F10.5,F11.5)
442      C
443      NCOU1=NCOU-1
444      N=NCOU1+NCOL-1
445      N1=N+1
446      C
447      C CALC. ELEMENT LENGTHS (D) AND CONTROL POINTS (XC,YC)
448      C XE(1)=XE(2*NCO-1)=XE(N1)=LEADING PT. X COORD.
449      DO 110 J=1,N
450      J1=J+1
451      XC(J)=(XE(J)+XE(J1))*0.5DO
452      YC(J)=(YE(J)+YE(J1))*0.5DO
453      D(J)=DSQRT((XE(J1)-XE(J))**2+(YE(J1)-YE(J))**2)
454      110  CONTINUE
455      C
456      C FIND TRAILING POINT COORDS. XC(N1),YC(N1): FIG.5
457      XT=XE(NCOU)-(XC(NCOU1)+XC(NCOU))*0.5DO
458      YT=YE(NCOU)-(YC(NCOU1)+YC(NCOU))*0.5DO
459      XC(N1)=XE(NCOU)+1.D-2*XT
460      YC(N1)=YE(NCOU)+1.D-2*YT
461      C
462      C FORM MATRICES K AND R: EQNS. 9 & 10
463      C DO FOR EACH SFC. ELEMENT J (COLUMN OF K) AND ROW OF R
464      DO 120 J=1,N1
465      R(J)=YC(J)*DCOS(ALPHAR)-XC(J)*DSIN(ALPHAR)
466      IF(J.EQ.N1)GO TO 140
467      J1=J+1
468      DE=D(J)
469      C CALCULATE ANGLE OF ELEMENT TO X-AXIS.
470      CO(J)=(XE(J1)-XE(J))/DE
471      SI(J)=(YE(J1)-YE(J))/DE
472      DELTA=DE/2.DO
473      140  DO 130 I=1,N1
474      IF(J.EQ.N1)GO TO 150
475      C FIND DISTANCE BETWEEN CONTROL PTS. I AND J.
476      DXC=XC(I)-XC(J)
477      DYC=YC(I)-YC(J)
478      C CALCULATE COMPONENTS OF EQN. 9 AND FIG 2
479      B=DXC*CO(J)+DYC*SI(J)
480      A=DYC*CO(J)-DXC*SI(J)
481      R1S=A*A+(B+DELTA)*(B+DELTA)
482      R2S=A*A+(B-DELTA)*(B-DELTA)
483      R3S=A*A+B*B-DELTA*DELTA
484      IF(R3S.LT.1.D-30)GO TO 160
485      T3=DATAN(2.DO*A*DELTA/R3S)
486      GO TO 170
487      160  IF(DABS(A).LT.1.D-30)GO TO 180
488      T3=DATAN((B+DELTA)/A)-DATAN((B-DELTA)/A)
489      GO TO 170
490      180  T3=DSIGN(PI,A)
491      170  T1=(B+DELTA)*DLOG(R1S)
492      T2=(B-DELTA)*DLOG(R2S)
493      K(I,J)=(T1-T2+2.DO*A*T3-4.DO*DELTA)/4.DO/PI
494      GO TO 130
495      C FOR LAST COLUMN OF K
496      150  K(I,J)=1.DO
497      130  CONTINUE
498      120  CONTINUE
499      IDGT=8
500      CALL LEQ1F(K,1,N1,101,R,IDGT,WKAREA,IER)
501      C ON OUTPUT, THE SOLN IS IN R
502      C
503      C CALCULATE THE LIFT COEFFICIENT.

```

```

504      CL=0.DO
505      DO 200 JJ=1,N
506      CL=CL-2.DO*R(JJ)*D(JJ)
507  200    CONTINUE
508      WRITE(6,10) IER,CL
509      WRITE(7,10) IER,CL
510      WRITE(7,20)
511  C OUTPUT AEROFOIL COORDS. AND SFC. VELOCITY.
512      DO 210 JJ=1,N1
513      WRITE(7,30)JJ,XC(JJ),YC(JJ),R(JJ)
514  210    CONTINUE
515      RETURN
516      END
517  C
518  C
519      SUBROUTINE STRMFN(TYPE)
520  C
521  C WRITTEN BY: M. OLESKIW ON: 800222 LAST MODIFIED: 801229
522  C
523  C CALCULATE STREAMFUNCTION ON A GRID ABOUT AN AEROFOIL SECTION
524  C GIVEN THE SFC. VORTICITY DENSITY ON THE AEROFOIL AND PLOT THE
525  C FLOW USING VELOCITY VECTORS.
526  C REF: KENNEDY, J.L. & D.F. MARSDEN (1976), CAN. AERO. & SPACE JOUR.
527  C V 22, #5, PP 243-256
528  C
529      DOUBLE PRECISION ALPHAR,XE(101),YE(101),XC(101),YC(101),GAMMA(101)
530      ,D(100),SI(100),CO(100),DBLE,YUP1,YLP1,YU,YL,ZZ,DEN,PJKE,PJKA,DD,
531      ,PID
532  C
533      REAL PSI(3721),K(101),DELTA,PI,ALPHAS,SNGL,SCO,SSI,X,Y,DXC,DYC,
534      ,XMIN,XMAX,YMIN,YMAX,B,A,R1S,R2S,T3,ATAN,SIGN,T1,T2,
535      ,R,ABS,LOG,FLOAT,SIN,COS,R3S,DX,DY,DPX,DPY,XPAGE,YPAGE,
536      ,XTIP,YTIP,XP1,YP1,YM1,U,V,AHL,AHLEN,SQRT
537  C
538      INTEGER XZ,YZ,TYPE,J,I,M,XZ1,YZ1,F,N,NCOU,NCOL,L,II
539  C
540      COMMON ALPHAR,PID/AERO1/XE,YE/AERO3/NCOU,NCOL/AERO2/XC,YC,GAMMA,D,
541      ,SI,CO
542      ,/GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ/SRCH/DD,II
543  C
544  C IN TYPE=AEROFOIL TYPE.
545  C
546  C PLOT BOUNDARIES
547      CALL NEWPEN(1)
548      CALL ORIGIN(999,21.0,10.5,5.0,5.0)
549      CALL AX2EP(3.5,3.2,0.0,9)
550      CALL AXIS2(0.0,0.0,'X/C',-3.21,0.0,XMIN,(XMAX-XMIN)/21.,3.5)
551      CALL AXIS2(21.0,0.0,'Y/C',-1,-10.5,90.0,0.0,1.75)
552      CALL AX2EP(1.75,3.3,0.0,11)
553      CALL AXIS2(0.0,0.0,'Y/C',3,10.5,90.0,YMIN,(YMAX-YMIN)/10.5,-1.75)
554      CALL AXIS2(0.0,10.5,'X/C',1,-21.0,XMIN,(XMAX-XMIN)/21.,3.5)
555  C CHANGE TO SECOND PEN
556      CALL NEWPEN(2)
557      N=NCOU+NCOL-2
558      PI=SNGL(PID)
559  C ALPHAR=ANGLE OF ATTACK IN RADIANS
560      ALPHAS=SNGL(ALPHAR)
561  C
562  C CALCULATE STRMFN. ON GRID.
563      DO 120 J=1,XZ
564      X=XMIN+FLOAT(J-1)/FLOAT(XZ-1)*(XMAX-XMIN)
565      DO 130 I=1,YZ
566  C PSI IS STORED IN VECTOR FORM BY COLUMNS.
567      M=(J-1)*YZ+I
568      Y=YMAX-FLOAT(I-1)/FLOAT(YZ-1)*(YMAX-YMIN)
569      PSI(M)=0.0
570      IF(TYPE.EQ.-1)GOTO 135
571      DO 140 L=1,N
572  C FIND DISTANCE BETWEEN CONTROL PT. L AND GRID PT. I,J.
573      DXC=X-SNGL(XC(L))
574      DYC=Y-SNGL(YC(L))
575  C CALCULATE COMPONENTS OF EQN. 9 AND FIG. 2
576      DELTA=SNGL(D(L))/2.0
577      SCO=SNGL(CO(L))

```

```

578      SSI=SNGL(SI(L))
579      B=DXC*SCO+DYC*SSI
580      A=DYC*SCO-DXC*SSI
581      R1S=A*A+(B+DELTA)*(B+DELTA)
582      R2S=A*A+(B-DELTA)*(B-DELTA)
583      R3S=A*A+B*B-DELTA*DELTA
584      IF(R3S.LT.1.E-30)GO TO 160
585      T3=ATAN(2.0*A*DELTA/R3S)
586      GO TO 170
587      160      IF(ABS(A).LT.1.E-30)GO TO 180
588      T3=ATAN((B+DELTA)/A)-ATAN((B-DELTA)/A)
589      GO TO 170
590      180      T3=SIGN(PI,A)
591      170      T1=(B+DELTA)*LOG(R1S)
592      T2=(B-DELTA)*LOG(R2S)
593      K(L)=(T1-T2+2.0*A*T3-4.0*DELTA)/4.0/PI
594      PSI(M)=PSI(M)-SNGL(GAMMA(L))*K(L)
595      140      CONTINUE
596      R=Y*COS(ALPHAS)-X*SIN(ALPHAS)
597      C ASSURE THAT PSI ON AEROFOIL = 0.
598      PSI(M)=PSI(M)+R-SNGL(GAMMA(N+1))
599      GOTO 130
600      C
601      C STREAMFN. FOR A CYLINDER.
602      135      DEN=(X-5.D-1)**2+Y*Y
603      IF(DEN.LT.1.D-70)GOTO 136
604      PSI(M)=Y-Y/4.DO/((X-5.D-1)**2+Y*Y)
605      136      PSI(M)=0.DO
606      130      CONTINUE
607      120      CONTINUE
608      C
609      XZ1=XZ-1
610      YZ1=YZ-1
611      F=0
612      II=1
613      C
614      DO 200 J=2,XZ1,2
615      DX=(XMAX-XMIN)/FLOAT(XZ1)
616      X=XMIN+FLOAT(J-1)*DX
617      DPX=21./FLOAT(XZ1)
618      C ARROWHEAD TAIL IN FRAME COORDS.
619      XPAGE=FLOAT(J-1)*DPX
620      XP1=X+DX
621      C CHECK IF CENTERED DIFFERENCING IS OK
622      IF(XP1.LE.SNGL(XE(1)))GOTO 220
623      CALL SFC(DBLE(XP1),YUP1,1,0,ZZ)
624      CALL SFC(DBLE(XP1),YLP1,0,0,ZZ)
625      F=F+1
626      IF(X.LE.SNGL(XE(1)))GOTO 220
627      CALL SFC(DBLE(X),YU,1,0,ZZ)
628      CALL SFC(DBLE(X),YL,0,0,ZZ)
629      F=F+1
630      C
631      C DO FOR EACH COLUMN OF ARROWHEAD TAILS
632      220      DO 210 I=2,YZ1,2
633      DPY=10.5/FLOAT(YZ1)
634      DY=(YMAX-YMIN)/FLOAT(YZ1)
635      Y=YMAX-FLOAT(I-1)*DY
636      C ARROWHEAD TAIL IN FRAME COORDS.
637      YPAGE=10.5-FLOAT(I-1)*DPY
638      M=(J-1)*YZ+I
639      IF(F.LE.1)GOTO 230
640      YP1=Y-DY
641      YM1=Y+DY
642      C IS CENTERED DIFFERENCING OK?
643      IF(YP1.GE.SNGL(YU).OR.YM1.LE.SNGL(YL))GOTO 230
644      IF(Y.GE.SNGL(YU))GOTO 250
645      C CHECK FOR LOCATION WITHIN AEROFOIL
646      IF(Y.GT.SNGL(YL))GOTO 210
647      C FORWARD DIFFERENCING IN Y
648      U=(PSI(M)-PSI(M+1))/DY
649      GOTO 240
650      C BACKWARD DIFFERENCING IN Y
651      250      U=(PSI(M-1)-PSI(M))/DY

```

```

652          GOTO 240
653      C CENTERED DIFFERENCING IN Y
654      230      U=(PSI(M-1)-PSI(M+1))/2.O/DY
655      240      IF(F.EQ.O)GOTO 260
656      C IS CENTERED DIFFERENCING OK?
657          IF(Y.GE.SNGL(YUP1).OR.Y.LE.SNGL(YLP1))GOTO 260
658      C BACKWARD DIFFERENCING IN X
659          V=(PSI((J-2)*YZ+1)-PSI((J-1)*YZ+1))/DX
660          GOTO 270
661      C CENTERED DIFFERENCING IN X
662      260      V=(PSI((J-2)*YZ+1)-PSI(J*YZ+1))/2.O/DX
663      C ARROWHEAD TIP
664      270      XTIP=XPAGE+U*DPX
665          YTIP=YPAGE+V*DPX
666          AHL=SQRT(U*U+V*V)
667      C ARROWHEAD LENGTH
668          AHLEN=O.25*AHL*DPX
669          CALL AROHD(XPAGE,YPAGE,XTIP,YTIP,AHLEN,O,16)
670      210      CONTINUE
671      200      CONTINUE
672      RETURN
673      END
674      C
675      C
676      SUBROUTINE AIRPLT(LAYER,TRJPLA,LYRMAX)
677      C
678      C WRITTEN BY: M. OLESKIW ON:800607 LAST MODIFIED: 801022
679      C
680      C PLOTS OUTLINE OF AEROFOIL WITHIN VIEW WINDOW
681      C
682          DOUBLE PRECISION XU(101),YU(101),DD,XL(101),YL(101),
683          .XE(101),YE(101)
684      C
685          REAL XMIN,XMAX,YMIN,YMAX,SNGL,XP,YP,XPT(104),
686          .YPT(104),XPE(103),YPE(103),XGR(104,10),YGR(104,10),
687          .XGRE(103,10),YGRE(103,10),XPP,YPP
688      C
689          INTEGER NCOU,NCOL,XZ,YZ,NCOB,IE,IP,I,J,NCOB1,I,II,
690          .IT(10),LAYER,ITT,TRJPLA,IPB,LYRMAX,ITE(10),ITTE,
691          .IEL,NEL,NEU,NELM2
692      C
693          COMMON /GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ/GROW/XGR,YGR,
694          .XGRE,YGRE,ITE,IT/AERO1/XE,YE/AERO3/NCOU,NCOL/SRCH/DD,II
695          ./SFCS/XU,YU,XL,YL/AERO4/NEU,NEL
696      C
697      C IN LAYER=INDEX OF ACCRETION LAYER
698      C IN TRJPLA=PLOT TRAJECTORIES (0 OR 1)
699      C IN LYRMAX=INDEX OF FINAL ACCRETION LAYER
700      C
701          NELM2=NEL-2
702          NCOB=NCOU+NCOL-1
703          NCOB1=NCOB-1
704          IP=O
705          IE=O
706      C
707      C FOR THE UPPER SFC.:
708          DO 700 J=1,NEU
709              XP=SNGL(XU(J))
710              YP=SNGL(YU(J))
711              IF(YP.GE.YMAX)GOTO 720
712              IF(XP.GE.XMAX)GOTO 730
713              IP=IP+1
714              XPT(IP)=XP
715              YPT(IP)=YP
716      700      CONTINUE
717          GOTO 740
718      720      IF(IP.GT.O)GOTO 750
719          XPT(IP+1)=XP
720          YPT(IP+1)=YMAX
721          GOTO 760
722      C OUT ALONG THE TOP EDGE
723      750      XPT(IP+1)=(XP-XPT(IP))/(YP-YPT(IP))*(YMAX-YPT(IP))+XPT(IP)
724          YPT(IP+1)=YMAX
725      C UPPER RIGHT CORNER

```

```

726      760      IP=IP+2
727              XPT(IP)=XMAX
728              YPT(IP)=YMAX
729              GOTO 740
730      C OUT ALONG THE RIGHT EDGE
731      730      XPT(IP+1)=XMAX
732              YPT(IP+1)=(YP-YPT(IP))/(XP-XPT(IP))*(XMAX-XPT(IP))+YPT(IP)
733              IP=IP+1
734      C
735      C FOR THE LOWER SFC.:
736      740      IPB=IP
737              IEL=O
738              DO 800 J=1,NELM2
739                  XP=SNGL(XL(NEL-J))
740                  YP=SNGL(YL(NEL-J))
741                  IF(XP.GE.XMAX.OR.YP.LE.YMIN)GOTO 820
742                  IF(J.EQ.1)GOTO 830
743                  IF(XPP.LE.XMAX.AND.YPP.GE.YMIN)GOTO 830
744                  IF(YPP.LE.YMIN)GOTO 840
745      C IN ON THE RIGHT EDGE
746              IP=IP+1
747              XPT(IP)=XMAX
748              YPT(IP)=(YP-YPP)/(XP-XPP)*(XMAX-XPP)+YPP
749              GOTO 820
750      C IN ON THE BOTTOM EDGE
751      840      XPT(IP+1)=XMAX
752              YPT(IP+1)=YMIN
753              IP=IP+2
754              XPT(IP)=(XP-XPP)/(YP-YPP)*(YMIN-YPP)+XPP
755              YPT(IP)=YMIN
756              GOTO 820
757      C ADD ANOTHER POINT WITHIN WINDOW.
758      830      IP=IP+1
759              XPT(IP)=XP
760              YPT(IP)=YP
761      820      XPP=XP
762              YPP=YP
763      800      CONTINUE
764              IF(IP.NE.IPB)GOTO 850
765              IP=IP+1
766              XPT(IP)=XMAX
767              YPT(IP)=YMIN
768      C
769      C ADD PARAMETERS NECESSARY FOR PLOTTING
770      850      XPT(IP+1)=XPT(1)
771              YPT(IP+1)=YPT(1)
772              XPT(IP+2)=XMIN
773              YPT(IP+2)=YMIN
774              DO 200 I=1,NCOB1
775                  XP=SNGL(XE(I))
776                  YP=SNGL(YE(I))
777                  IF(XP.GT.XMAX)GOTO 200
778                  IF(YP.GT.YMAX.OR.YP.LT.YMIN)GOTO 200
779                  IE=IE+1
780                  XPE(IE)=XP
781                  YPE(IE)=YP
782      200      CONTINUE
783              XPE(IE+1)=XMIN
784              YPE(IE+1)=YMIN
785              XPT(IP+3)=(XMAX-XMIN)/21.0
786              XPE(IE+2)=(XMAX-XMIN)/21.0
787              YPT(IP+3)=(YMAX-YMIN)/10.5
788              YPE(IE+2)=(YMAX-YMIN)/10.5
789              IT(LAYER)=IP+3
790              ITT=IP+3
791      C
792      C THESE ARE THE AEROFOIL OUTLINE POINTS TO BE PLOTTED WITHIN THE WINDOW
793              DO 400 I=1,ITT
794                  XGR(I,LAYER)=XPT(I)
795                  YGR(I,LAYER)=YPT(I)
796      400      CONTINUE
797              ITE(LAYER)=IE+2
798              ITTE=IE+2
799      C THESE ARE THE AEROFOIL ELEMENT ENDPNTS WITHIN THE WINDOW

```

```

800          DO 450 I=1,ITTE
801          XGRE(I,LAYER)=XPE(I)
802          YGRE(I,LAYER)=YPE(I)
803          450 CONTINUE
804          IF(TRJPLA.EQ.O.OR.LAYER.GT.LYRMAX)GOTO 500
805          CALL NEWPEN(3)
806          CALL LINE(XPT,YPT,IP+1,1,0,0)
807          CALL LINEP(0,1)
808          CALL LINE(XPE,YPE,IE,1,-1,0)
809          500 RETURN
810          END
811          C
812          C
813          DOUBLE PRECISION FUNCTION NSURF(XROT)
814          C
815          C WRITTEN BY: M. OLESKIW ON. 800905 LAST MODIFIED: 801022
816          C
817          C CALCULATES THE UNROTATED X VALUE OF A POINT ON THE ACCRETED AEROFOIL
818          C SURFACE BASED UPON THE COLLISION EFFICIENCY, DIRECTION OF
819          C GROWTH, AND OLD AEROFOIL (ROTATED) SFC. POSITION.
820          C
821          DOUBLE PRECISION XUR(101),YUR(101),CU(100,3),XLR(101),YLR(101),
822          CL(100,3),C30,S30,XROT,D,LENG,LEN,LU(101),LL(101),
823          L(51),YO(51),CEE(50,3),XLRT,YLRT,XN,YN,DD,SLP,K,XLRN,YLRN,
824          DSIGN,DSORT,ICE,NSURFY,CE
825          C
826          INTEGER J,RUN,I,ICT,ICU,ICL,NEU,NEL,NEL1
827          C
828          COMMON /FOIL/XUR,YUR,XLR,YLR/SPLINE/CU,CL/ROTP/C30,S30
829          /IND/NSURFY,ICE,I,J,RUN/LG/LU,LL/COL/L,YO,ICT,ICU,ICL/EFF/CEE
830          /NOSE/XN,YN/AERO4/NEU,NEL
831          C
832          C IN XROT=ROTATED X POSITION ON LOWER AEROFOIL SFC.
833          C
834          10 FORMAT('OOUT OF BOUNDS IN SEARCHING FOR AEROFOIL',
835          'OR CE SPLINES IN NSURF')
836          C
837          IF(J.LT.1)J=1
838          RUN=RUN+1
839          NEL1=NEL-1
840          C
841          C FIND THE APPROPRIATE AEROFOIL SPLINE SEGMENT
842          120 IF(XROT.GT.XLR(J))GOTO 105
843          J=J-1
844          IF(J.EQ.O)GOTO 600
845          GOTO 120
846          105 IF(XROT.LE.XLR(J+1))GOTO 110
847          J=J+1
848          IF(J.LE.NEL1)GOTO 105
849          GOTO 600
850          110 D=XROT-XLR(J)
851          C FIND LENGTH ALONG SFC. FROM NOSE TO THIS POINT.
852          CALL SFLEN(D,LENG,CL(J,3),CL(J,2),CL(J,1))
853          LEN=LL(J)+LENG
854          C ROTATED COORDS.
855          XLRT=XROT
856          YLRT=YLR(J)+((CL(J,3)*D+CL(J,2))*D+CL(J,1))*D
857          C
858          C FIND THE APPROPRIATE CE VS L SPLINE SEGMENT
859          IF(I.LT.1)I=1
860          220 IF(-LEN.GT.L(I))GOTO 205
861          I=I-1
862          IF(I.EQ.O)GOTO 200
863          GOTO 220
864          205 IF(-LEN.LE.L(I+1))GOTO 210
865          I=I+1
866          IF(I.LE.ICL)GOTO 205
867          GOTO 600
868          C CE EQUALS 0 - NEW AND OLD SFCS. THE SAME.
869          200 NSURF=XLRT*C30+YLRT*S30+XN
870          NSURFY=-XLRT*S30+YLRT*C30+YN
871          RETURN
872          C
873          C CALCULATE THE COLLISION EFFICIENCY.

```

```

874      210 DD=-LEN-L(I)
875          CE=(3.DO*CEE(I,3)*DD+2.DO*CEE(I,2))*DD+CEE(I,1)
876      C FIND AEROFOIL SLOPE
877          SLP=(3.DO*CL(J,3)*D+2.DO*CL(J,2))*D+CL(J,1)
878          K=-1.DO/SLP
879      C
880      C NEW SURFACE COORDS:
881          XLRN=XLRT-DSIGN(DSQRT(ICE*ICE*CE*CE/(1.DO+K*K)),K)
882          YLRN=YLRT+K*(XLRN-XLRT)
883          NSURF=XLRN*C30+YLRN*S30+XN
884          NSURFY=-XLRN*S30+YLRN*C30+YN
885          RETURN
886      600 WRITE(6,10)
887          WRITE(7,10)
888          RETURN
889      END
890      C
891      C
892          SUBROUTINE SFC(X,Y,S,L,LEN)
893      C
894      C WRITTEN BY: M. OLESKIW ON:800623 LAST MODIFIED:801102
895      C
896      C CALCULATES Y VALUES AND THE LENGTH FROM THE NOSE
897      C ON THE SFC. OF THE AEROFOIL BY A CUBIC SPLINE INTERPOLATION
898      C
899          DOUBLE PRECISION XN,YN,XUR(101),YUR(101),CU(100,3),CL(100,3),
900          XLR(101),YLR(101),XB,DELTA,DELTAP,DABS
901          S30,C30,XR,YR,X,Y,LU(101),LL(101),LEN,LENG,D
902      C
903          INTEGER S,L,I,JU,JL,NEU1,NEU,NEL1,NEL
904      C
905          COMMON /NOSE/XN,YN/LG/LU,LL/FOIL/XUR,YUR,XLR,YLR/SPLINE/CU,CL
906          /ROTP/C30,S30/AERO4/NEU,NEL/SRCH/D,I
907      C
908      C IN X=POINT AT WHICH Y VALUE IS TO BE CALCULATED
909      C OUT Y=SFC. POSITION ON SPLINE
910      C IN S=0: LOWER SFC.
911      C      1: UPPER SFC.
912      C IN L=1: FIND LENGTH ALONG AEROFOIL SFC. FROM NOSE TO (X,Y)
913      C OUT LEN=LENGTH ALONG AEROFOIL SFC. FROM NOSE TO (X, )
914      C
915      10 FORMAT('OUT OF BOUNDS ON SEARCHING FOR SFC. POSITION ',
916          'IN ROUTINE SFC')
917      C
918          JU=1
919          JL=1
920      C ROTATED X COORD.
921          XR=(X-XN)*C30
922          IF(S.EQ.0)GOTO 150
923      C
924      C FOR THE UPPER SFC.
925          NEU1=NEU-1
926          IF(XR.GT.0.DO)GOTO 120
927          IF(XR.LT.0.DO)GOTO 600
928          Y=YN
929          LEN=0.DO
930          RETURN
931      C FIND THE APPROPRIATE SPLINE SEGMENT.
932      120 IF(XR.GT.XUR(I))GOTO 105
933          I=I-1
934          IF(I.EQ.0)GOTO 600
935          GOTO 120
936      105 IF(XR.LE.XUR(I+1))GOTO 110
937          I=I+1
938          IF(I.LE.NEU1)GOTO 105
939          GOTO 600
940      110 D=XR-XUR(I)
941      C ROTATED Y COORD.
942          YR=((CU(I,3)*D+CU(I,2))*D+CU(I,1))*D+YUR(I)
943          XB=XR*C30-YR*S30+XN
944          DELTA=X-XB
945          IF(DABS(DELTA).LE.1.D-10)GOTO 400
946          DELTAP=-C30+S30*((3.DO*CU(I,3)*D+2.DO*CU(I,2))*D+CU(I,1))
947          XR=XR-DELTAP/DELTAP

```

```

948      JU=JU+1
949      GOTO 120
950      C
951      C UNROTATED Y COORD.
952      400  Y=YR*C30+YN+XR*S30
953           IF(L.EQ.O)GOTO 300
954      C FIND THE SEGMENT LENGTH.
955      CALL SFCLN(D,LENG,CU(I,3),CU(I,2),CU(I,1))
956      LEN=LU(I)+LENG
957      GOTO 300
958      C
959      C FOR THE LOWER SFC.:
960      150  NEL1=NEL-1
961      C FIND THE APPROPRIATE SFC. SPLINE SEGMENT.
962      220  IF(XR.GT.XLR(I))GOTO 205
963           I=I-1
964           IF(I.EQ.O)GOTO 600
965           GOTO 220
966      205  IF(XR.LE.XLR(I+1))GOTO 210
967           I=I+1
968           IF(I.LE.NEL1)GOTO 205
969           GOTO 600
970      210  D=XR-XLR(I)
971      C ROTATED Y COORD.
972      YR=((CL(I,3)*D+CL(I,2))*D+CL(I,1))*D+YLR(I)
973      XB=XR*C30+YR*S30+XN
974      DELTA=X-XB
975      IF(DABS(DELTA).LE.1.D-10)GOTO 500
976      DELTAP=-C30-S30*((3.DO*CL(I,3)*D+2.DO*CL(I,2))*D+CL(I,1))
977      XR=XR-DELTA/DELTAP
978      JL=JL+1
979      GOTO 220
980      C
981      C UNROTATED Y COORD.
982      500  Y=-XR*S30+YR*C30+YN
983           IF(L.EQ.O)GOTO 300
984      C FIND THE SEGMENT LENGTH.
985      CALL SFCLN(D,LENG,CL(I,3),CL(I,2),CL(I,1))
986      LEN=LL(I)+LENG
987      300  RETURN
988      C
989      600  WRITE(6,10)
990           WRITE(7,10)
991           RETURN
992      END
993      C
994      C
995      SUBROUTINE SFCLN(D,L,A,B,C)
996      C
997      C WRITTEN BY: M. OLESKIW ON:800525 LAST MODIFIED:800902
998      C
999      C CALCULATES THE LENGTH ALONG A SEGMENT OF THE CUBIC SPLINE FIT OF THE
1000     C AEROFOIL SFC.
1001     C
1002     C REF:DOUG S. PHILLIPS (1980)
1003     C
1004     DOUBLE PRECISION II,NU,E,F,DSQRT,DELTA,G,A,B,C,D,L,
1005     .T1,T2,T3,T4,NU1,ANU1,DABS,NUO,ANUO,K,E2,F2,E3,F3,E02,FO2,
1006     .E03,FO3,XO,X1,CK,FO,E0,F1,E1,YP,DISTP,DIST,DFLOAT,Y,
1007     .DLOG,DSIGN
1008     C
1009     INTEGER IER,I,ANAL
1010     C
1011     COMMON /LA/ANAL
1012     C
1013     C IN D=ROTATED X COORDINATE OF POINT FROM BEGINNING OF SEGMENT
1014     C OF INTEREST TO WHICH THE LENGTH IS TO BE FOUND.
1015     C OUT L=SEGMENT LENGTH
1016     C IN A=
1017     C IN B=
1018     C IN C= SPLINE PARAMETERS FOR SECTION OF INTEREST
1019     C

```

```

1020      II(NU,E,F)=NU/3.DO*DSQRT(1.DO+(DELTA+NU*NU)**2)*
1021      .(1.DO+2.DO*DELTA*G*G/(1.DO+NU*NU*G*G))
1022      .+((1.DO+DELTA*G*G)*F-2.DO*DELTA*G*G*E)/3.DO/G**3
1023      C
1024      IF(ANAL.EQ.O)GOTO 200
1025      IF(A.NE.O.DO)GOTO 100
1026      IF(B.NE.O.DO)GOTO 110
1027      C
1028      C A AND B EQUAL TO O
1029      L=D*DSQRT(1.DO+C*C)
1030      RETURN
1031      C
1032      C A EQUAL O, B NOT EQUAL O
1033      110 T1=(2.DO*B*D+C)*DSQRT(1.DO+(2.DO*B*D+C)**2)
1034      T2=C*DSQRT(1.DO+C*C)
1035      T3=DLOG((2.DO*B*D+C)+DSQRT(1.DO+(2.DO*B*D+C)**2))
1036      T4=DLOG(C+DSQRT(1.DO+C*C))
1037      L=(T1-T2+T3-T4)/4.DO/B
1038      RETURN
1039      C
1040      C A NOT EQUAL O
1041      100 NU1=DSQRT(3.DO*DABS(A))*(D+B/3.DO/A)
1042      ANU1=DABS(NU1)
1043      NUO=B/3.DO/A*DSQRT(3.DO*DABS(A))
1044      ANUO=DABS(NUO)
1045      DELTA=(C-B*B/3.DO/A)*DSIGN(1.DO,A)
1046      G=1.DO/(1.DO+DELTA*DELTA)**0.25DO
1047      K=DSQRT(5.D-1-DELTA*G*G/2.DO)
1048      E2=O.DO
1049      F2=O.DO
1050      EO2=O.DO
1051      FO2=O.DO
1052      XO=2.DO*G*ANUO/(1.DO-ANUO*ANUO*G*G)
1053      X1=2.DO*G*ANU1/(1.DO-ANU1*ANU1*G*G)
1054      CK=DSQRT(1.DO-K*K)
1055      IF(ANU1.EQ.1.DO/G)GOTO 120
1056      IF(ANU1.GT.1.DO/G)GOTO 130
1057      C
1058      C ZETA LESS THAN PI/2
1059      CALL DELI1(F1,X1,CK)
1060      CALL DELI2(E1,X1,CK,1.DO,CK*CK)
1061      GOTO 140
1062      C
1063      C ZETA GREATER THAN PI/2
1064      130 CALL DELI1(F2,-X1,CK)
1065      CALL DELI2(E2,-X1,CK,1.DO,CK*CK)
1066      C
1067      C ZETA EQUALS PI/2
1068      120 CALL DCEL1(F3,K,IER)
1069      CALL DCEL2(E3,K,1.DO,CK*CK,IER)
1070      F1=2.DO*F3-F2
1071      E1=2.DO*E3-E2
1072      140 IF(ANUO.EQ.1.DO/G)GOTO 150
1073      IF(ANUO.GT.1.DO/G)GOTO 160
1074      C
1075      C ZETA LESS THAN PI/2
1076      CALL DELI1(FO,XO,CK)
1077      CALL DELI2(EO,XO,CK,1.DO,CK*CK)
1078      GOTO 170
1079      C
1080      C ZETA GREATER THAN PI/2
1081      160 CALL DELI1(FO2,-XO,CK)
1082      CALL DELI2(EO2,-XO,CK,1.DO,CK*CK)
1083      C
1084      C ZETA EQUALS PI/2
1085      150 CALL DCEL1(FO3,K,IER)
1086      CALL DCEL2(EO3,K,1.DO,CK*CK,IER)
1087      FO=2.DO*FO3-FO2
1088      EO=2.DO*EO3-EO2
1089      170 L=(DSIGN(1.DO,NU1)*II(ANU1,E1,F1)-DSIGN(1.DO,NUO)*II(ANUO,EO,FO))
1090      ./DSQRT(3.DO*DABS(A))
1091      RETURN
1092      C
1093      C NON-ANALYTICAL (APPROXIMATE) SFC LENGTH DETERMINATION.

```

```

1094      200      L=0.DO
1095              YP=0.DO
1096              DISTP=0.DO
1097                  DO 210 I=1,25
1098                      DIST=D*DFLOAT(I)/25.DO
1099                      Y=((A*DIST+B)*DIST+C)*DIST
1100                      L=L+DSQRT((DIST-DISTP)**2+(Y-YP)**2)
1101                      YP=Y
1102                      DISTP=DIST
1103      210      CONTINUE
1104              RETURN
1105              END
1106      C
1107      C
1108              SUBROUTINE CE(YOL,CEL,CEX,PLTFAC,THICK,LAYER)
1109      C
1110      C WRITTEN BY: M. OLESKIW  ON:800622  LAST MODIFIED:801227
1111      C
1112      C CALCULATE AND PLOT COLLISION EFFICIENCY OF ARBITRARY AEROFOIL
1113      C GIVEN A SET OF IMPACTING TRAJECTORIES
1114      C
1115              DOUBLE PRECISION D,L(51),YO(51),BPAR(4),CEE(50,3),THICK,
1116              .PN,P,DIST,SLP,SSLP,DABS,CET,ALPHAR,DCOS,CEMAX,PNI,ZZ,DCOS,
1117              .LU(101),LL(101),XU(101),XL(101),YU(101),YL(101),Y,DBLE
1118      C
1119              REAL LPMIN,YOPMIN,LRG,YORG,SNGL,FACT(2),LP(103),FLOAT,
1120              .YOP(103),CEP(203),XPAR(4,10),YPAR(4,10),LS(53),YOS(53),
1121              .PLTFAC,XP(203),XPMIN,CEPMIN,X,XLF,XRG,COS
1122      C
1123              INTEGER CEL,F,I,ICT,IER,IRX,IRY,PX,PY,YOL,ICU,ICL,J,CEX,II,
1124              .KK,KL,KU,LAYER,NEU,NEL,CO,IJ
1125      C
1126              COMMON ALPHAR/COL/L,YO,ICT,ICU,ICL/EFF/CEE/PLTPRM/XPAR,YPAR
1127              ./CEM/P/LG/LU,LL/SFCS/XU,YU,XL,YL/SRCH/D,IJ/AERO4/NEU,NEL
1128      C
1129      C IN YOL=PLOT YO VS L GRAPH(O OR 1)
1130      C IN CEL=PLOT CE VS L GRAPH(O OR 1).
1131      C IN CEX=PLOT CE VS X GRAPH(O OR 1).
1132      C IN PLTFAC=FACTOR FOR SCALING ALL PLOTS.
1133      C IN THICK=AEROFOIL THICKNESS IN %.
1134      C IN LAYER=INDEX OF ACCRETION LAYER.
1135      C
1136      10      FORMAT(' -BETAO (MAX LOCAL CE) IS',F7.1,'% AT A DISTANCE OF',
1137      .F10.3,' FROM THE NOSE',/, ' THE TOTAL COLLISION EFFICIENCY IS',
1138      .F7.1,'%')
1139      20      FORMAT(' -FAILURE TO CONVERGE UPON MAX CE')
1140      C
1141              FACT(1)=1.0
1142              FACT(2)=0.7
1143      C CUBIC SPLINE END PARAMETERS
1144              BPAR(1)=1.DO
1145              BPAR(2)=6.DO*(YO(2)-YO(1))/(L(2)-L(1))**2
1146              BPAR(3)=1.DO
1147              BPAR(4)=-6.DO*(YO(1CT)-YO(1CT-1))/(L(1CT)-L(1CT-1))**2
1148      C CREATE SINGLE PRECISION VERSIONS OF L AND YO IN LS AND YOS
1149              DO 130 I=1,ICT
1150                  LS(I)=SNGL(L(I))
1151      130      CONTINUE
1152              DO 140 I=1,ICT
1153                  YOS(I)=SNGL(YO(I))
1154      140      CONTINUE
1155      C FIT CUBIC SPLINE TO YO VS L CURVE
1156              CALL ICSICU(L,YO,ICT,BPAR,CEE,50,IER)
1157      C
1158      C FIND BETAO (MAX VALUE OF LOCAL CE)
1159              PNI=0.DO
1160      540      PN=PNI
1161              J=0
1162      520      P=PN
1163      C FIND YO VS L SLOPE AND CE VS L SLOPE
1164              I=1
1165              IF(P.LT.L(1))P=L(1)
1166      500      IF(P.LT.L(I+1))GOTO 510
1167              I=I+1

```

```

1168         IF(I.LT.ICT)GOTO 500
1169         P=L(ICT)
1170     510     DIST=P-L(I)
1171         SSLP=6.DO*CEE(I,3)
1172         SLP=6.DO*CEE(I,3)*DIST+2.DO*CEE(I,2)
1173         IF(DABS(SSLP-O.DO).LT.1.D-10)GOTO 512
1174     C THE NEWTON-RAPHSON METHOD
1175         PN=P-SLP/SSLP
1176         J=J+1
1177         IF(J.LT.100)GOTO 530
1178         IF(PNI.LE.-4.D-2)GOTO 550
1179         PNI=PNI-1.D-2
1180         GOTO 540
1181     550     WRITE(6,20)
1182         WRITE(7,20)
1183         GOTO 560
1184     530     IF(DABS(PN-P).GT.1.D-5)GOTO 520
1185     C
1186     C FIND THE TOTAL AND MAX. COLLISION EFFICIENCY.
1187     512     CET=(YO(ICT)-YO(1))*DCOS(ALPHAR)/THICK*1.D4
1188         CEMAX=((3.DO*CEE(I,3)*DIST+2.DO*CEE(I,2))*DIST+CEE(I,1))*1.D2
1189         *DCOS(ALPHAR)
1190         WRITE(6,10)CEMAX,P,CET
1191         WRITE(7,10)CEMAX,P,CET
1192     C
1193     C DETERMINE PLOTTING PARAMETERS.
1194     560     IF(LAYER.EQ.1)CALL PLTSZ(LS(1),LS(ICT),YOS(1),YOS(ICT),
1195         .LPMIN,YOPMIN,PX,PY,IRX,IRY)
1196         IF(LAYER.GT.1)CALL PLTSZ(LS(1),LS(ICT),YOS(1),YOS(ICT),
1197         .LPMIN,YOPMIN,PX,PY,IRX,IRY)
1198         LS(ICT+1)=LPMIN
1199         LS(ICT+2)=XPAR(4,IRX)/10.O**PX
1200         CALL NEWPEN(1)
1201         IF(YOL.EQ.O)GOTO 200
1202     C
1203     C PLOT THE YO VS L GRAPH.
1204         YOS(ICT+1)=YOPMIN
1205         YOS(ICT+2)=YPAR(4,IRY)/10.O**PY
1206         CALL FACTOR(FACT(YOL)*PLTFAC)
1207         CALL ORIGIN(999,20.O,13.O,5.O,5.O)
1208         CALL AX2EP(XPAR(3,IRX),3,1+PX,0,1,0)
1209         CALL AXIS2(0.O,0.O,'L/C',-3,XPAR(2,IRX),0.O,LPMIN,XPAR(4,IRX)/
1210         10.O**PX,XPAR(3,IRX))
1211         CALL AXIS2(XPAR(2,IRX),0.O,' ',-1,-YPAR(2,IRY),90.O,1.O,1.O,YPAR(3
1212         ,IRY))
1213         CALL AX2EP(YPAR(3,IRY),3,1+PY,0,1,1)
1214         CALL AXIS2(0.O,0.O,'YO/C',4,YPAR(2,IRY),90.O,YOPMIN,YPAR(4,IRY)/
1215         10.O**PY,-YPAR(3,IRY))
1216         CALL AXIS2(0.O,YPAR(2,IRY),' ',1,-XPAR(2,IRX),0.O,1.,1.,XPAR(3,IRX
1217         ))
1218     C PLOT THE YO VS L POINTS
1219         CALL LINEP(0.15)
1220         CALL LINE(LS,YOS,ICT,1,-1,0)
1221         F=1
1222         LRG=LS(ICT)-LS(1)
1223         YORG=YOS(ICT)-YOS(1)
1224         DO 100 I=1,101
1225             LP(I)=LS(1)+FLOAT(I-1)/100.O*LRG
1226     120     IF(LP(I).LE.LS(F+1))GOTO 110
1227             F=F+1
1228             GOTO 120
1229     110     D=LP(I)-LS(F)
1230             YOP(I)=SNGL(((CEE(F,3)*D+CEE(F,2))*D+CEE(F,1))*D)+YOS(F)
1231     100     CONTINUE
1232             YOP(102)=YOS(ICT+1)
1233             YOP(103)=YOS(ICT+2)
1234             LP(102)=LS(ICT+1)
1235             LP(103)=LS(ICT+2)
1236     C PLOT THE YO VS L LINE
1237         CALL LINE(LP,YOP,101,1,0,1)
1238     C
1239     C PLOT THE CE VS L GRAPH.
1240     200     IF(CEL.EQ.O)GOTO 300
1241         CALL FACTOR(FACT(CEL)*PLTFAC)

```

```

1242      CALL ORIGIN(999,20.0,13.0,5.0,5.0)
1243      CALL AX2EP(XPAR(3,IRX),3,1+PX,0,1.0)
1244      CALL AXIS2(0.0,0.0,0.0,'L/C',-3,XPAR(2,IRX),0.0,LPMIN,XPAR(4,IRX)/
1245      10.0*PX,XPAR(3,IRX))
1246      CALL AXIS2(XPAR(2,IRX),0.0,' ',-1,-YPAR(2,10),90.0,0.0,1.0,YPAR(3,
1247      10))
1248      CALL AX2EP(YPAR(3,10),3,0.0,1.1)
1249      CALL AXIS2(0.0,0.0,0.0,'COLLISION EFFICIENCY IN %',.25,YPAR(2,10),
1250      90.0,0.0,YPAR(4,10)*10.0,-YPAR(3,10))
1251      CALL AXIS2(0.0,YPAR(2,10),' ',1,-20.0,0.0,1.1,XPAR(3,IRX))
1252      LRG=LS(1CT)-LS(1)
1253      F=1
1254      C DETERMINE PLOTTING VALUES OF CE.
1255      DO 210 I=1,101
1256      LP(I)=LS(1)+FLOAT(I-1)/100.0*LRG
1257      230      IF(LP(I).LE.LS(F+1))GOTO 220
1258      F=F+1
1259      GOTO 230
1260      220      D=LP(I)-LS(F)
1261      CEP(I)=SNGL((3.DO*CEE(F,3)*D+2.DO*CEE(F,2))*D+CEE(F,1))*100.0
1262      *COS(SNGL(ALPHAR))
1263      210      CONTINUE
1264      LP(102)=LS(1CT+1)
1265      LP(103)=LS(1CT+2)
1266      CEP(102)=0.0
1267      CEP(103)=YPAR(4,10)*10.0
1268      C PLOT THE CE VS L LINE.
1269      CALL LINE(LP,CEP,101,1,0,1)
1270      300      IF(CEX.EQ.0.OR.LAYER.GT.1)GOTO 400
1271      DO 310 KL=1,NEL
1272      C
1273      C PLOT THE CE VS X GRAPH.
1274      IF(-LL(KL).LE.L(1))GOTO 320
1275      310      CONTINUE
1276      320      DO 330 KU=1,NEU
1277      IF(LU(KU).GT.L(1CT))GOTO 340
1278      330      CONTINUE
1279      340      XRG=SNGL(XL(KL)+XU(KU))
1280      XLF=SNGL(-XU(KU))
1281      CO=0
1282      II=1CT-1
1283      DO 350 KK=1,201
1284      X=XLF+XRG/200.*FLOAT(KK-1)
1285      XP(KK)=X
1286      C DETERMINE VALUE OF L FOR EACH X.
1287      IF(X.GT.0.)GOTO 360
1288      CALL SFC(DBLE(-X),Y,1,1,ZZ)
1289      GOTO 370
1290      360      CALL SFC(DBLE(X),Y,0,1,ZZ)
1291      ZZ=-ZZ
1292      370      IF(CO.EQ.1)GOTO 380
1293      IF(ZZ.GT.L(1CT))GOTO 380
1294      IF(ZZ.GT.L(II))GOTO 410
1295      II=II-1
1296      IF(II.EQ.0)GOTO 390
1297      GOTO 370
1298      390      CO=1
1299      380      CEP(KK)=0.0
1300      GOTO 350
1301      410      D=ZZ-L(II)
1302      CEP(KK)=SNGL((3.DO*CEE(II,3)*D+2.DO*CEE(II,2))*D+CEE(II,1))*100.
1303      *COS(SNGL(ALPHAR))
1304      350      CONTINUE
1305      C DETERMINE THE PLOTTING PARAMETERS.
1306      CALL PLT5ZE(XP(1),XP(201),0.DO,99.9,XPMIN,CEPMIN,PX,PY,IRX,IRY)
1307      XP(202)=XPMIN
1308      XP(203)=XPAR(4,IRX)/10.0*PX
1309      CEP(202)=0.0
1310      CEP(203)=YPAR(4,10)*10.0
1311      C PLOT CE VS X AXES
1312      CALL FACTOR(FACT(CEX)*PLTFAC)
1313      CALL ORIGIN(999,20.0,13.0,5.0,5.0)
1314      CALL AX2EP(XPAR(3,IRX),3,1+PX,0,1.0)
1315      CALL AXIS2(0.0,0.0,0.0,'X/C',-3,XPAR(2,IRX),0.0,XPMIN,XPAR(4,IRX)

```

```

1316      /10.0*PX, XPAR(3, IRX))
1317      CALL AXIS2(XPAR(2, IRX), 0.0, ' ', -1, -YPAR(2, 10), 90.0, 0.0, 1.0, YPAR(3,
1318      10))
1319      CALL AX2EP(YPAR(3, 10), 3, 0.0, 1.1)
1320      CALL AXIS2(0.0, 0.0, 'COLLISION EFFICIENCY IN %', 25, YPAR(2, 10),
1321      90.0, 0.0, YPAR(4, 10)*10.0, -YPAR(3, 10))
1322      CALL AXIS2(0.0, YPAR(2, 10), ' ', 1, -20.0, 0.0, 1.1, XPAR(3, IRX))
1323      C PLOT THE CE VS X LINE.
1324      CALL LINE(XP, CEP, 201, 1.0, 1)
1325      400      RETURN
1326      END
1327      C
1328      C
1329      SUBROUTINE PLTSZ(XMIN, XMAX, YMIN, YMAX, XL, YB, PX, PY, IRX, IRY)
1330      C
1331      C WRITTEN BY: M. OLESKIW ON: 800627 LAST MODIFIED: 801018
1332      C
1333      C DETERMINE PARAMETERS NECESSARY FOR SCALING OF A PLOT AND ITS AXES
1334      C
1335      REAL XPAR(4, 10), YPAR(4, 10), XD, FLOAT, AINT, XMIN, XMAX,
1336      XL, YD, YMIN, YMAX, YB, DX, DY, XR, YT
1337      C
1338      INTEGER PX, PN, PY, PNY, I, J, IX, IRX, INT, IY, IRY, IFIX
1339      C
1340      COMMON/PLTPRM/XPAR, YPAR
1341      C
1342      C IN XMIN=
1343      C IN XMAX=
1344      C IN YMIN=
1345      C IN YMAX=
1346      C OUT XL=LEFT EDGE OF PLOT
1347      C OUT YB=BOTTOM EDGE OF PLOT
1348      C OUT PX=POWER OF TEN IN X-AXIS RANGE
1349      C OUT PY=POWER OF TEN IN Y-AXIS RANGE
1350      C OUT IRX=MIN. LENGTH OF X AXIS.
1351      C OUT IRY=MIN. LENGTH OF Y AXIS.
1352      C
1353      10      FORMAT(8F10.0)
1354      C
1355      C READ IN PLOTTING PARAMETERS
1356      DO 101 I=2, 10
1357      READ(3, 10)(XPAR(J, I), J=1, 4), (YPAR(J, I), J=1, 4)
1358      101      CONTINUE
1359      C
1360      ENTRY PLTSZ(XMIN, XMAX, YMIN, YMAX, XL, YB, PX, PY, IRX, IRY)
1361      PN=0
1362      PNY=0
1363      C
1364      C DETERMINE THE PLOTTING RANGE OF THE X VARIABLE
1365      100      PX=PN
1366      XD=(XMAX-XMIN)*10.0*PX
1367      IF(XD.GT.10.0)PNX=PNX-1
1368      IF(XD.LT.1.00001)PNX=PNX+1
1369      IF(PNX.NE.PX)GOTO 100
1370      C PX GIVES 1/(POWER OF TEN) OF THE X VARIABLE PLOTTING RANGE
1371      IX=1
1372      120      IRX=INT(XD)+IX
1373      DX=FLOAT(IRX)/10.0*PX/XPAR(1, IRX)
1374      C SET THE X VALUE AT THE LEFT GRAPH EDGE
1375      IF(XMIN.LT.0)XL=AINT(XMIN/DX-1.0)*DX
1376      IF(XMIN.GE.0)XL=AINT(XMIN/DX)*DX
1377      XR=XL+XPAR(1, IRX)*DX
1378      IF(XR.GE.XMAX.AND.IRX.NE.3.AND
1379      IRX.NE.6.AND.IRX.NE.7.AND.IRX.NE.9)GOTO 105
1380      IX=IX+1
1381      GOTO 120
1382      105      IF(IFIX((XR-XMAX)/DX) LE IFIX((XMIN-XL)/DX))GOTO 110
1383      C CENTRE THE PLOT.
1384      XL=XL-DX
1385      XR=XR-DX
1386      GOTO 105
1387      C
1388      C DETERMINE THE PLOTTING RANGE OF THE Y VARIABLE
1389      110      PY=PNY

```

```

1390      YD=(YMAX-YMIN)*10.0**PY
1391      IF(YD.GT.9.99999)PNY=PNY-1
1392      IF(YD.LT.1.0)PNY=PNY+1
1393      IF(PNY.NE.PY)GOTO 110
1394      C PY GIVES 1/(POWER OF TEN) OF THE Y VARIABLE PLOTTING RANGE
1395      IY=1
1396      130      IRY=INT(YD)+IY
1397      DY=FLOAT(IRY)/10.0**PY/YPAR(1,IRY)
1398      C SET THE Y VALUE AT THE BOTTOM OF THE GRAPH
1399      IF(YMIN.LT.0.0)YB=AINT(YMIN/DY-1.0)*DY
1400      IF(YMIN.GE.0.0)YB=AINT(YMIN/DY)*DY
1401      YT=YB+YPAR(1,IRY)*DY
1402      IF(YT.GE.YMAX)GOTO 135
1403      IY=2
1404      GOTO 130
1405      135      IF(IFIX((YT-YMAX)/DY).LE.IFIX((YMIN-YB)/DY))GOTO 140
1406      C CENTRE THE PLOT
1407      YB=YB-DY
1408      YT=YT-DY
1409      GOTO 135
1410      140      RETURN
1411      END
1412      C
1413      C
1414      SUBROUTINE ICING(CETOL,ICE,BOTH,FAIL)
1415      C
1416      C WRITTEN BY: M. OLESKIW ON:800713 LAST MODIFIED:801227
1417      C
1418      C CALCULATE AMOUNT OF ACCRETION AND DETERMINE A NEW SET OF AEROFOIL
1419      C SURFACE ELEMENT ENDPOINTS AFTER DETERMINING THE AEROFOIL
1420      C NOSE LOCATION.
1421      C
1422      DOUBLE PRECISION XN,YN,XNN,YNN,XUR(101),YUR(101),
1423      CU(100,3),CL(100,3),XLR(101),YLR(101),L(51),YO(51),
1424      D,CEE(50,3),CETOL,K,DSIGN,XLRN(101),YLRN(101),
1425      S30,C30,NSURF,XURN(101),YURN(101),CEU(101),CEL(101),D1,D2,
1426      XUT(101),XLT(101),XU(101),YU(101),XL(101),YL(101),
1427      YUT(101),YLT(101),DABS,DSORT,LU(101),LL(101),ICE,
1428      PP,TOL,LE,RE,ICEE,ALPHAR,DCOS
1429      DOUBLE PRECISION NSURFY,XRMIN,XNP,YNP,XURTLP,XLRTLP
1430      INTEGER BOTH,J,NCOU,NCOL,NCOUN,NCOLN,ICT,ICU,ICL,I,IER,NOAC,ONCE,
1431      IM,IUS,ILS,IK,FAIL,RUN,NEU,NEL,IU(51),IL(51),IUN(51),ILN(51),
1432      I1,I2,J1,J2,KK,KL,LLL,IXU(101),IXL(101),IZU(101),IZL(101),IUU,ILL
1433      C
1434      COMMON ALPHAR/AERO3/NCOU,NCGL/NOSE/XN,YN/FOIL/XUR,YUR,
1435      XLR,YLR/ROTP/C30,S30/CEM/PP/IND/NSURFY,ICEE,I,J,RUN/AERO4/NEU,
1436      NEL/COL/L,YO,ICT,ICU,ICL/EFF/CEE/SFCS/XU,YU,XL,YL/LG/LU,LL
1437      /SPLINE/CU,CL/ENDS/IU,IL/NNOSE/XNP,YNP,XURTLP,XLRTLP
1438      C
1439      EXTERNAL NSURF
1440      C
1441      C IN CETOL=CRITERION FOR DETERMINING THE NEED FOR NEW CONTROL
1442      C SEGMENT ENDPOINTS.
1443      C IN ICE=MAX. THICKNESS OF ICE ACCRETION (ASSUMING CE=100%).
1444      C IN BOTH=TRAJECTORIES FOR BOTH SFCS (0 OR 1)
1445      C OUT FAIL=FAILURE INDICATOR.
1446      C
1447      10      FORMAT(' -FAILURE TO CONVERGE TO NEW NOSE POSITION')
1448      20      FORMAT(' 1ENDPT. X COORD. Y COORD. DIST. FROM NOSE COLL. EFF. ')
1449      30      FORMAT(' ',F14.5,F10.5,F17.5,F12.4)
1450      40      FORMAT(' ')
1451      C
1452      XURTLP=XUR(NEU)
1453      XLRTLP=XLR(NEL)
1454      J=ICL
1455      NOAC=0
1456      ONCE=0
1457      C
1458      C FOR THE UPPER SFC
1459      DO 100 I=1,NEU
1460      IF(NOAC.EQ.1)GOTO 115
1461      C DETERMINE THE APPROPRIATE CE VS L SEGMENT
1462      110      IF(LU(I).LE.L(J+1))GOTO 120
1463      J=J+1

```

```

1464         IF(J.LT.ICT)GOTO 110
1465         NOAC=1
1466 C NO ACCRETION REGION ON TOP SFC.
1467 115      CEU(I)=0.DO
1468          XURN(I)=XUR(I)
1469          YURN(I)=YUR(I)
1470          GOTO 100
1471 120      D=LU(I)-L(J)
1472          CEU(I)=((3.DO*CEE(J,3)*D+2.DO*CEE(J,2))*D+CEE(J,1))*DCOS(ALPHAR)
1473          IF(DABS(CU(I,1)).LT.1.D-20)GOTO 150
1474          K=-1.DO/CU(I,1)
1475 C
1476 C NEW ENDPPTS.:
1477          XURN(I)=XUR(I)+DSIGN(DSQR(ICE*ICE*CEU(I)*CEU(I)/(1.DO+K*K)),K)
1478          YURN(I)=YUR(I)+K*(XURN(I)-XUR(I))
1479          GOTO 100
1480 C GROWTH IN Y AXIS DIRECTION
1481 150      XURN(I)=XUR(I)
1482          YURN(I)=YUR(I)+CEU(I)*ICE
1483 100      CONTINUE
1484          DO 160 I=1,NCOU
1485          IUN(I)=IU(I)
1486 160      CONTINUE
1487          NCOUN=NCOU
1488 C
1489 C CHECK FOR NEED OF CREATING NEW CONTROL ENDPPTS. ON UPPER SFC.
1490          DO 300 I=2,NCOU
1491          IF(ONCE.EQ.O)GOTO 335
1492          ONCE=O
1493          GOTO 300
1494 335      I1=IUN(I)
1495          I2=IUN(I-1)
1496          IF(I1.EQ.I2+1)GOTO 300
1497          IF(CEU(I2).EQ.O.DO)GOTO 390
1498 C CHECK FOR ZERO CE BETWEEN CONTROL ENDPPTS.
1499          IF(CEU(I1).EQ.O.DO)GOTO 330
1500 C CHECK FOR RAPID CHANGE IN CE
1501 325      IF(DABS(CEU(I1)-CEU(I2)).LT.CETOL)GOTO 315
1502          J1=I2+1
1503          J2=I1
1504          DO 320 J=J1,J2
1505          IF(DABS(CEU(J)-CEU(J-1)).GE.CETOL/1.2DO)GOTO 350
1506 320      CONTINUE
1507          GOTO 360
1508 330      J1=I2+1
1509          J2=I1
1510          DO 340 J=J1,J2
1511          IF(CEU(J).EQ.O.DO.AND.CEU(J-1).GE.CETOL/2.DO)GOTO 350
1512 340      CONTINUE
1513          GOTO 325
1514 350      KK=J-1
1515          IF(J.EQ.J1)KK=J
1516          GOTO 370
1517 C CHECK IF DISTANCE BETWEEN CONTROL ENDPPTS. IS INCREASING SUBSTANTIALLY
1518 315      D1=DSQR((XUR(I1)-XUR(I2))**2+(YUR(I1)-YUR(I2))**2)
1519          D2=DSQR((XURN(I1)-XURN(I2))**2+(YURN(I1)-YURN(I2))**2)
1520          IF(D2.LT.1.25DO*D1)GOTO 300
1521 360      KK=(I1+I2)/2
1522          ONCE=1
1523 370      KL=NCOUN-I+1
1524 C
1525 C SHIFT INDICES OF CONTROL ENDPPTS. TO MAKE ROOM FOR A NEW ONE.
1526          DO 380 LLL=1,KL
1527          IUN(NCOUN+2-LLL)=IUN(NCOUN+1-LLL)
1528 380      CONTINUE
1529          NCOUN=NCOUN+1
1530          IUN(I)=KK
1531 300      CONTINUE
1532 390      J=1
1533          DO 170 I=1,NEU
1534          IF(IUN(J).EQ.I)GOTO 180
1535          IXU(I)=O
1536          GOTO 170
1537 180      IXU(I)=1

```

```

1538      J=J+1
1539      CONTINUE
1540      WRITE(7,20)
1541      DO 190 I=1,NEU
1542      WRITE(7,30)XU(I),YU(I),LU(I),CEU(I)
1543      IF(CEU(I).EQ.O.DO)GOTO 195
1544      190      CONTINUE
1545      195      IF(BOTH.EQ.O)GOTO 590
1546      J=ICL+1
1547      NOAC=0
1548      C
1549      C FOR THE LOWER SFC.:
1550      DO 200 I=1,NEL
1551      IF(NOAC.EQ.1)GOTO 215
1552      C DETERMINE THE APPROPRIATE CE VS L SEGMENT.
1553      210      IF(-LL(I).GT.L(J))GOTO 220
1554      J=J-1
1555      IF(J.GT.O)GOTO 210
1556      NOAC=1
1557      C NO ACCRETION REGION ON LOWER SFC.
1558      215      CEL(I)=O.DO
1559      XLRN(I)=XLR(I)
1560      YLRN(I)=YLR(I)
1561      GOTO 200
1562      220      D=-LL(I)-L(J)
1563      CEL(I)=((3.DO*CEE(J,3)*D+2.DO*CEE(J,2))*D+CEE(J,1))*DCOS(ALPHAR)
1564      IF(DABS(CL(I,1)).LT.1.D-20)GOTO 250
1565      K=-1.DO/CL(I,1)
1566      C
1567      C NEW ENDPTS.:
1568      XLRN(I)=XLR(I)-DSIGN(DSQRT(ICE*ICE*CEL(I)*CEL(I)/(1.DO+K*K)),K)
1569      YLRN(I)=YLR(I)+K*(XLRN(I)-XLR(I))
1570      GOTO 200
1571      C GROWTH IN Y AXIS DIRECTION
1572      250      XLRN(I)=XLR(I)
1573      YLRN(I)=YLR(I)-CEL(I)*ICE
1574      200      CONTINUE
1575      DO 260 I=1,NCOL
1576      ILN(I)=IL(I)
1577      260      CONTINUE
1578      NCOLN=NCOL
1579      ONCE=0
1580      C
1581      C CHECK FOR NEED OF CREATING NEW CONTROL ENDPTS. ON LOWER SFC.
1582      DO 400 I=2,NCOL
1583      IF(ONCE.EQ.O)GOTO 435
1584      ONCE=0
1585      GOTO 400
1586      435      I1=ILN(I)
1587      I2=ILN(I-1)
1588      IF(I1.EQ.I2+1)GOTO 400
1589      IF(CEL(I2).EQ.O.DO)GOTO 905
1590      C CHECK FOR ZERO CE BETWEEN CONTROL ENDPTS.
1591      IF(CEL(I1).EQ.O.DO)GOTO 430
1592      C CHECK FOR RAPID CHANGE IN CE.
1593      425      IF(DABS(CEL(I1)-CEL(I2)).LT.CETOL)GOTO 415
1594      J1=I2+1
1595      J2=I1
1596      DO 420 J=J1,J2
1597      IF(DABS(CEL(J)-CEL(J-1)).GE.CETOL/1.200)GOTO 450
1598      420      CONTINUE
1599      GOTO 460
1600      430      J1=I2+1,
1601      J2=I1
1602      DO 440 J=J1,J2
1603      IF(CEL(J).EQ.O.DO.AND.CEL(J-1).GE.CETOL/2.DO)GOTO 450
1604      440      CONTINUE
1605      GOTO 425
1606      450      KK=J-1
1607      IF(J.EQ.J1)KK=J
1608      GOTO 470
1609      C CHECK IF DISTANCE BETWEEN CONTROL ENDPTS. IS INCREASING SUBSTANTIALLY.
1610      415      D1=DSQRT((XLR(I1)-XLR(I2))**2+(YLR(I1)-YLR(I2))**2)
1611      D2=DSQRT((XLRN(I1)-XLRN(I2))**2+(YLRN(I1)-YLRN(I2))**2)

```

```

1612      IF(D2.LT.1.25D0*D1)GOTO 400
1613 460      KK=(I1+I2)/2
1614      ONCE=1
1615 470      KL=NCOLN-I+1
1616      C
1617      C SHIFT INDICES OF CONTROL ENDPNTS. TO MAKE ROOM FOR A NEW ONE.
1618      DO 480 LLL=1,KL
1619      ILN(NCOLN+2-LLL)=ILN(NCOLN+1-LLL)
1620 480      CONTINUE
1621      NCOLN=NCOLN+1
1622      ILN(I)=KK
1623 400      CONTINUE
1624 905      J=1
1625      DO 270 I=1,NEL
1626      IF(ILN(I).EQ.I)GOTO 280
1627      IXL(I)=0
1628      GOTO 270
1629 280      IXL(I)=1
1630      J=J+1
1631 270      CONTINUE
1632      WRITE(7,40)
1633      DO 230 I=1,NEL
1634      WRITE(7,30)XL(I),YL(I),LL(I),CEL(I)
1635      IF(CEL(I).EQ.O.DO)GOTO 900
1636 230      CONTINUE
1637      GOTO 900
1638      C
1639      C UPPER & LOWER SFCS. MIRROR IMAGES; NOSE STAYS ON THE X-AXIS.
1640 590      DO 595 I=1,NEU
1641      XLRN(I)=XURN(I)
1642      YLRN(I)=-YURN(I)
1643      IXL(I)=IXU(I)
1644 595      CONTINUE
1645      GOTO 930
1646      C
1647      C FIND NEW NOSE LOCATION USING THE GOLDEN SECTION SEARCH METHOD
1648      C OF DETERMINING THE MIN. VALUE OF THE NEW SURFACE X-COORD.
1649 900      ICEE=ICE
1650      RUN=0
1651      J=1
1652      I=1
1653      DO 910 KK=1,NCOL
1654      IF(LL(KK).GE.-PP)GOTO 920
1655 910      CONTINUE
1656 920      TOL=1.D-5
1657      FAIL=0
1658      LE=1.D-10
1659      RE=XL(R(KK))
1660      CALL ZXGSN(NSURF,LE,RE,TOL,XRMIN,IER)
1661      IF(IER.LT.129.OR.IER.GT.132)GOTO 950
1662      FAIL=1
1663      WRITE(6,10)
1664      WRITE(7,10)
1665      GOTO 720
1666      C NEW NOSE COORDS.:
1667 950      YNN=NSURFY
1668      XNN=NSURF(XRMIN)
1669      C
1670      C DE-ROTATE NEW UPPER & LOWER SFCS. ABOUT PREVIOUS NOSE POSITION
1671 930      DO 500 I=1,NEU
1672      XUT(I)=XURN(I)*C30-YURN(I)*S30+XN
1673      YUT(I)=XURN(I)*S30+YURN(I)*C30+YN
1674 500      CONTINUE
1675      DO 510 I=1,NEL
1676      XLT(I)=XLRN(I)*C30+YLRN(I)*S30+XN
1677      YLT(I)=-XLRN(I)*S30+YLRN(I)*C30+YN
1678 510      CONTINUE
1679      IF(BOTH.EQ.1)GOTO 520
1680      XNN=XUT(1)
1681      YNN=YUT(1)
1682      IM=1
1683 520      XU(1)=XNN
1684      XL(1)=XNN
1685      YU(1)=YNN

```

```

1686      YL(1)=YNN
1687      IUU=1
1688      ILL=1
1689      IF(BOTH.EQ.O)GOTO 625
1690      C
1691      C SEE IF ANY LOWER SFC. ENDPNTS. ARE ABOVE THE NEW NOSE POSITION
1692      C & THUS BELONG ON THE UPPER SFC.
1693      DO 610 IM=1,NEL
1694      IF(DABS(YLT(IM)-YNN).LT.1.D-4)GOTO 620
1695      IF(YLT(IM).LT.YNN)GOTO 630
1696      610      CONTINUE
1697      620      IF(IM.GT.2)GOTO 640
1698      IF(IM.EQ.2)GOTO 650
1699      C SAME NOSE INDEX
1700      625      IUS=2
1701      ILS=2
1702      GOTO 665
1703      C NEW NOSE IS NEAR FIRST ENDPNT. BELOW PREVIOUS NOSE
1704      650      IUS=1
1705      ILS=3
1706      GOTO 665
1707      C NEW NOSE IS NEAR SECOND OR GREATER ENDPNT. BELOW PREVIOUS NOSE
1708      640      IK=IM-2
1709      DO 670 I=1,IK
1710      IUU=IUU+1
1711      XU(IUU)=XLT(IM-I)
1712      YU(IUU)=YLT(IM-I)
1713      IZU(IUU)=IXL(IM-I)
1714      670      CONTINUE
1715      IUS=1
1716      ILS=IM+1
1717      665      IZU(1)=1
1718      IZL(1)=1
1719      GOTO 660
1720      630      IF(IM.GT.2)GOTO 680
1721      C NEW NOSE IS BETWEEN FIRST & SECOND ENDPNTS. ON LOWER SFC.
1722      IUS=1
1723      ILS=2
1724      GOTO 666
1725      C NEW NOSE IS BELOW SECOND ENDPNT. ON LOWER SFC.
1726      680      IK=IM-2
1727      DO 690 I=1,IK
1728      IUU=IUU+1
1729      XU(IUU)=XLT(IM-I)
1730      YU(IUU)=YLT(IM-I)
1731      IZU(IUU)=IXL(IM-I)
1732      690      CONTINUE
1733      IUS=1
1734      ILS=IM
1735      666      IZU(1)=1
1736      IZL(1)=1
1737      660      DO 700 I=IUS,NEU
1738      IUU=IUU+1
1739      XU(IUU)=XUT(I)
1740      YU(IUU)=YUT(I)
1741      IZU(IUU)=IXU(I)
1742      IF(I.EQ.IUS.AND.IUU.LT.3)IZU(IUU)=0
1743      700      CONTINUE
1744      DO 710 I=ILS,NEL
1745      ILL=ILL+1
1746      XL(ILL)=XLT(I)
1747      YL(ILL)=YLT(I)
1748      IZL(ILL)=IXL(I)
1749      710      CONTINUE
1750      NEU=IUU
1751      NEL=ILL
1752      XNP=XN
1753      YNP=YN
1754      XN=XNN
1755      YN=YNN
1756      IUU=1
1757      DO 730 I=1,NEU
1758      IF(IZU(I).EQ.O)GOTO 730
1759      IU(IUU)=I

```

```

1760      IUU=IUU+1
1761      730      CONTINUE
1762      ILL=1
1763      DO 740 I=1,NEL
1764      IF (IZL(I).EQ.O)GOTO 740
1765      IL(ILL)=I
1766      ILL=ILL+1
1767      740      CONTINUE
1768      NCOU=IUU-1
1769      NCOL=ILL-1
1770      720      RETURN
1771      END
1772      C
1773      C
1774      SUBROUTINE GROWTH(ICEPLA,LYRMAX,PLTFAC,TRJPLA)
1775      C
1776      C WRITTEN BY: M. OLESKIW ON:800713 LAST MODIFIED:801022
1777      C
1778      C PLOTS SUCCESSIVE AEROFOIL OUTLINES WITHIN VIEW WINDOW
1779      C
1780      REAL XGR(104,10),YGR(104,10),PLTFAC,XMIN,XMAX,YMIN,YMAX,
1781      .XPLT(104),YPLT(104),XGRE(103,10),YGRE(103,10),XPLTE(101),
1782      .YPLTE(101)
1783      C
1784      INTEGER IT(10),XZ,YZ,LYRMAX,ICEPLA,ITT,I,J,TRJPLA,LYRM1,
1785      .ITE(10),ITTE
1786      C
1787      COMMON/GROW/XGR,YGR,XGRE,YGRE,ITE,IT/GRID/XMIN,XMAX,YMIN,YMAX,XZ,
1788      .YZ
1789      C
1790      C IN ICEPLA=PLOT ACCRETION OUTLINE. (0 OR 1)
1791      C IN LYRMAX=NO. OF LAYERS TO BE ACCRETED.
1792      C IN PLTFAC=PLOT EXPANSION/REDUCTION FACTOR.
1793      C IN TRJPLA=PLOT TRAJECTORIES. (0 OR 1)
1794      C
1795      IF(ICEPLA.EQ.2)GOTO 120
1796      C DRAW AXES.
1797      CALL NEWPEN(1)
1798      CALL ORIGIN(999,21.0,10.5,5.0,5.0)
1799      CALL AX2EP(3.5,3,2,0,0.9)
1800      CALL AXIS2(0.,0., 'X/C',-3,21.,0.,XMIN,(XMAX-XMIN)/21.,3.5)
1801      CALL AXIS2(21.,0., ' ', -1,-10.5,90.,0.,0.,1.75)
1802      CALL AX2EP(1.75,3,3,0,1.1)
1803      CALL AXIS2(0.,0., 'Y/C',3,10.5,90.,YMIN,(YMAX-YMIN)/10.5,-1.75)
1804      CALL AXIS2(0.,10.5, ' ', 1,-21.,0.,XMIN,(XMAX-XMIN)/21.,3.5)
1805      LYRM1=LYRMAX+1
1806      120      DO 100 I=1,LYRM1
1807      ITT=IT(I)
1808      ITTE=ITE(I)
1809      DO 110 J=1,ITT
1810      XPLT(J)=XGR(J,I)
1811      YPLT(J)=YGR(J,I)
1812      110      CONTINUE
1813      DO 210 J=1,ITTE
1814      XPLTE(J)=XGRE(J,I)
1815      YPLTE(J)=YGRE(J,I)
1816      210      CONTINUE
1817      CALL NEWPEN(3)
1818      C DRAW ACCRETION OUTLINES.
1819      CALL LINE(XPLT,YPLT,IT(I)-2,1,0,0)
1820      CALL LINEP(0,1)
1821      C PLOT CONTROL SEGMENT ENDPIS.
1822      CALL LINE(XPLTE,YPLTE,ITE(I)-2,1,-1,0)
1823      100      CONTINUE
1824      RETURN
1825      END
1826      C
1827      C
1828      SUBROUTINE TRAJEC(TYPE,TRJPLA,THICK,AT,BOTH)
1829      C
1830      C WRITTEN BY: M. OLESKIW ON:790526 LAST MODIFIED:801227
1831      C
1832      C CALCULATE TRAJECTORIES OF DROPLETS IN POTENTIAL FLOW
1833      C ABOUT AN AEROFOIL

```

```

1834 C
1835 DOUBLE PRECISION DFLOAT,UINF,C,RD,CD,GS,RDS,RHOA,RHOD,NUS,
1836 MU,DTS(6),DEL,XP(7),YP(7),WDSREL,DBLE,HF,UST,VST,EPS,
1837 CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,
1838 C17,C18,C19,C20,C21,C22,C23,C24,F,TSLOPE,HFP,ADD,
1839 XL,YL,COORD,CLAP,XCDUPR,YCDUPR,YCAUPR,XCDLPR,YCDLPR,
1840 YCALPR,PIM1,PIM2,FPIM1,FPIM2,XCOLL,YCOLL,DABS,DSIGN,
1841 LT(51,2),CLAPP,K,K1,LG,LP1,LTH,TOL,XN,YN,YOG,XI
1842 DOUBLE PRECISION PSI(7),DUADX,DVADY,DMIN1,DTSS,L(51),YO(51),
1843 YCDU,YCDL,XCDU,XCDL,ZZ,YCAU,YCAL,UAS(6),VAS(6),RED(6),
1844 LEN,PRDST1,PRDST0,PLDST1,DIST,TS(500),YUS1,YLS1,YUS2,YLS2,UVAT,
1845 DSQRT,PINF,TINF,CRIT,XO,USLOPE,LSLOPE,YOT(25,2),DDD
1846 DOUBLE PRECISION XDS(6),UDS(6),AN(2,6),YDS(6),TTLACN,VPSQ,NA,
1847 VDS(6),HT(2,6),AO,A1,A2,B0,B1,B2,B3,E5B,CO,C1,C2,
1848 DM1,DO,D1,D2,E5,UPI,UCI,VPI,VC1,XPI,XCI,YPI,YCI,ER1,ER2,
1849 PRD,PLD,BETAO,YCG,THICK,CLAPPP,SLP,YOTUX,YOTLX,LINT
1850
1851 C
1852 REAL XMIN,XMAX,YMIN,YMAX,SNGL,X,Y,XDSP(150),YDSP(150),YPREV,
1853 XPREV
1854 C
1855 INTEGER I,CDS,XZ,YZ,IJ,IK,TRJEND,SMASH,ALMOST,AT,BOTH,ACN,
1856 GRAZE,IC,ICL,ICT,ICU,IG,IU,NT,PLOTI,UX,LX,II,ICLL,III,
1857 TRJPR,TRJPLA,PRINTI,PRINTO,NTRAJU,NTRAJL,TYPE,TYPE2,
1858 IM4,IM3,IM2,IM1,IO,IP1,ITEMP,EQN,PC,INT,EQ
1859 C
1860 COMMON /EQNMN/GS,RHOA,RHOD,RDS,NUS,HF
1861 /AIR/XP,YP,DEL,PSI,TYPE2/REL/UAS,VAS,RED,CD
1862 /GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ
1863 /PV/XDS,YDS,UDS,VDS/INTEG/AN,HT
1864 /PCM/AO,A1,A2,B0,B1,B2,B3,CO,C1,C2,DM1,DO,D1,D2,
1865 UPI,UCI,VPI,VC1,ER1,ER2,XPI,XCI,YPI,YCI,UST,VST
1866 /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1
1867 COMMON /RKFM/CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,
1868 C15,C16,C17,C18,C19,C20,C21,C22,C23,C24
1869 /COL/L,YO,ICT,ICU,ICL,NOSE/XN,YN/SRCH/DDD,III
1870 C
1871 C IN TYPE=AEROFOIL TYPE.
1872 C IN TRJPLA=PLOT DROPLET TRAJECTORIES. (0 OR 1)
1873 C IN THICK=AEROFOIL THICKNESS IN %.
1874 C IN AT=AUTO TRAJECTORY MODE (0 OR 1)
1875 C IN BOTH=CALCULATE TRAJECTORIES TO COLLIDE ON BOTH SFCS. (0 OR 1)
1876 C
1877 10 FORMAT(/5F6.0,2D10.2)
1878 20 FORMAT(/14,2I7,16,I7,F5.0,F6.0)
1879 25 FORMAT(/2I7,I3,I5,I4,I3,F6.0,D8.0,I4)
1880 30 FORMAT(/30F10.0)
1881 40 FORMAT('OSTEP',T7,'TIME',T15,'DTS',T22,'XDS',T31,'YDS',T40,'PSI',
1882 T49,'UAS',T58,'UDS',T67,'VAS',T76,'VDS',T86,'RED',T94,
1883 'ACCN/MOD HIST/RHS',T114,'USTAB',T123,'VSTAB')
1884 50 FORMAT(' ',I4,F6.2,F7.4,7F9.5,F10.5,4F9.5)
1885 60 FORMAT('OCLOSEST APPROACH IS Y=',F10.5,' NO. OF STEPS REQUIRED=',
1886 I3,' PSI=',F8.3)
1887 70 FORMAT('1TRAJECTORY STARTING POSITION IS X=',
1888 F6.2,' YO=',F9.5)
1889 75 FORMAT('-TRAJECTORY STARTING POSITION IS X=',
1890 F6.2,' YO=',F9.5)
1891 80 FORMAT('OCOLLISION COORDS: X=',F10.7,' Y=',F10.7,' L=',F10.7,
1892 ' NO. OF STEPS REQUIRED=',I3)
1893 90 FORMAT('OFIRST TRAJECTORY HIT AEROFOIL')
1894 95 FORMAT('OUNEXPECTED AEROFOIL MISS')
1895 96 FORMAT('OYO?')
1896 97 FORMAT(F10.0)
1897 C
1898 C STATEMENT FUNCTION TO CALCULATE DISTANCE BETWEEN
1899 C AEROFOIL SLOPE AND TRAJECTORY.
1900 F(X)=TSLOPE*(X-XL)+YL-COORD
1901 C
1902 C INPUT PARAMETERS
1903 READ(4,10)UINF,C,PINF,TINF,RD,A1,A2
1904 READ(4,20)CDS,TRJPR,PRINTI,PLOTI,PRINTO,CRIT,BETAO
1905 READ(4,25)NTRAJU,NTRAJL,AT,BOTH,EQN,PC,DTSS,EPS,ACN
1906 READ(4,30)XO
1907 C CHECK FOR AUTO-TRAJECTORY MODE
1908 IF(AT.EQ.1)GOTO 700

```

```

1908      NT=51
1909      GOTO 710
1910      C CHECK TO SEE IF COLLISION EFFICIENCIES ARE TO BE CALCULATED
1911      C   FOR BOTH SFCS.
1912      700  NT=BOTH+1
1913      C
1914      C NON-DIMENSIONAL VIEWPORT DIAGONAL LENGTH
1915      710  LEN=DSQRT(DBLE((XMAX-XMIN)**2+(YMAX-YMIN)**2))
1916      C PRINT LENGTH INTERVAL WITHIN VIEWPORT
1917      PRDSTI=LEN/DFLOAT(PRINTI)
1918      C PLOT LENGTH INTERVAL WITHIN VIEWPORT
1919      PLDSTI=LEN/DFLOAT(PLOTI)
1920      C PRINT LENGTH INTERVAL TO LEFT OF VIEWPORT
1921      PRDSTO=LEN/DFLOAT(PRINTO)
1922      C NON-DIMENSIONAL ACCN. OF GRAVITY
1923      GS=O.DO*C/UINF/UINF
1924      C NON-DIMENSIONAL DROPLET RADIUS
1925      RDS=RD*1.D-6/C
1926      DEL=RDS
1927      C AIR DENSITY
1928      RHOA=PINF*1.D3/287.O4DO/(TINF+273.16DO)
1929      C WATER DENSITY REF: LIST - SMT
1930      RHOD=999.15DO
1931      C DYNAMIC VISCOSITY OF AIR REF: LOZOWSKI ET AL. (1979)
1932      MU=1.718D-5+5.1D-8*TINF
1933      C NON-DIMENSIONAL KINEMATIC VISCOSITY OF AIR:
1934      NUS=MU/RHOA/C/UINF
1935      TOL=1.D-5*THICK
1936      TYPE2=TYPE
1937      IF(PC.NE.2)GOTO 420
1938      C
1939      C DETERMINE PARAMETERS FOR RUNGE-KUTTA-FEHLBERG METHOD.
1940      CC1=.25DO
1941      CC2=3.DO/32.DO
1942      C3=9.DO/32.DO
1943      C4=1932.DO/2197.DO
1944      C5=72.D2/2197.DO
1945      C6=7296.DO/2197.DO
1946      C7=439.DO/216.DO
1947      C8=8.DO
1948      C9=3680.DO/513.DO
1949      C10=845.DO/4104.DO
1950      C11=8.DO/27.DO
1951      C12=2.DO
1952      C13=3544.DO/2565.DO
1953      C14=1859.DO/4104.DO
1954      C15=11.DO/40.DO
1955      C16=25.DO/216.DO
1956      C17=1408.DO/2565.DO
1957      C18=2197.DO/4104.DO
1958      C19=.2DO
1959      C20=16.DO/135.DO
1960      C21=6656.DO/12825.DO
1961      C22=28561.DO/56430.DO
1962      C23=9.DO/50.DO
1963      C24=2.DO/55.DO
1964      GOTO 400
1965      420  IF(PC.NE.1)GOTO 400
1966      C
1967      C DETERMINE PARAMETERS FOR PREDICTOR-CORRECTOR METHOD.
1968      AO=1.DO-A1-A2
1969      BO=(55.DO+9.DO*A1+8.DO*A2)/24.DO
1970      B1=(-59.DO+19.DO*A1+32.DO*A2)/24.DO
1971      B2=(37.DO-5.DO*A1+8.DO*A2)/24.DO
1972      B3=(-9.DO+A1)/24.DO
1973      E5B=(251.DO-19.DO*A1-8.DO*A2)/6.DO
1974      C1=A1
1975      C2=A2
1976      CO=1.DO-C1-C2
1977      DM1=(9.DO-C1)/24.DO
1978      DO=(19.DO+13.DO*C1+8.DO*C2)/24.DO
1979      D1=(-5.DO+13.DO*C1+32.DO*C2)/24.DO
1980      D2=(1.DO-C1+8.DO*C2)/24.DO
1981      E5=(-19.DO+11.DO*C1-8.DO*C2)/6.DO

```

```

1982      ER1=E5B/(E5B-E5)
1983      ER2=E5/(E5B-E5)
1984      C
1985      C FOR EACH TRAJECTORY (OR TRAJECTORY SET):
1986      ENTRY TRAJEK
1987      400  IF(AT.EQ.0)GOTO 390
1988      READ(4,30)(YO(I),I=1,NT)
1989      390  DO 200 IJ=1,NT
1990          IF(AT.EQ.1)GOTO 395
1991          WRITE(6,96)
1992          READ(5,97)YO(IJ)
1993          IF(DABS(YO(IJ)).LT.1.D-10)GOTO 690
1994      395  IG=1
1995          GRAZE=1
1996          IC=1
1997          INT=0
1998          K1=1.DO
1999          K=0.85DO
2000          YDS(1)=YO(IJ)
2001      C SET COUNTERS
2002      405  IM4=2
2003          IM3=3
2004          IM2=4
2005          IM1=5
2006          IO=6
2007          IP1=1
2008      C
2009      C DROPLET AT INITIAL POSITION
2010          XDS(1)=XO
2011          WRITE(6,75) XDS(1),YDS(1)
2012          WRITE(7,70) XDS(1),YDS(1)
2013          IF(PC.NE.1)GOTO 410
2014      C
2015      C SET PREVIOUS PREDICTOR-CORRECTOR VALUES TO 0.
2016          XPI=0.DO
2017          XCI=0.DO
2018          YPI=0.DO
2019          YCI=0.DO
2020          UPI=0.DO
2021          UCI=0.DO
2022          VPI=0.DO
2023          VCI=0.DO
2024      410  IF(ACN.EQ.1)GOTO 415
2025      C
2026      C SET DROPLET TRAVELLING WITH JUST SLIGHTLY GREATER VELOCITY
2027      C THAN AIR (RED=0.001)
2028          CALL AIRVEL(XDS(1),YDS(1),UAS(1),VAS(1),4)
2029      C CALCULATE TOTAL AIR VELOCITY.
2030          UVAT=DSQRT(UAS(1)*UAS(1)+VAS(1)*VAS(1))
2031      C CALCULATE TOTAL STARTING RELATIVE VELOCITY.
2032          WDSREL=1.D-3*NUS/2.DO/RDS
2033      C CALCULATE INITIAL DROPLET VELOCITY
2034          UDS(1)=UAS(1)*(1.DO+WDSREL/UVAT)
2035          VDS(1)=VAS(1)*(1.DO+WDSREL/UVAT)
2036          GOTO 416
2037      C SET GRID FOR INITIAL DROPLET VELOCITY CALCULATIONS
2038      C
2039      415  XP(6)=XDS(1)+2.DO*RDS
2040          XP(7)=XDS(1)+2.DO*RDS
2041          YP(6)=YDS(1)+RDS
2042          YP(7)=YDS(1)-RDS
2043          CALL AIRVEL(XDS(1),YDS(1),UAS(1),VAS(1),7)
2044      C CALCULATE DUA/DX
2045          DUADX=(PSI(6)+PSI(4)-PSI(7)-PSI(3))/4.DO/RDS/RDS
2046      C CALCULATE DVA/DY
2047          DVADY=(PSI(3)+PSI(7)-PSI(6)-PSI(4))/4.DO/RDS/RDS
2048      C TOTAL POTENTIAL FLOW ACCELERATIVE TERM
2049          UVAT=DSQRT(DUADX*DUADX+DVADY*DVADY)
2050      C CALCULATE TOTAL STARTING RELATIVE VELOCITY
2051          WDSREL=1.D-3*NUS/2.DO/RDS
2052      C ASSURE STARTING RED=0.001 WEIGHTED BY POTENTIAL FLOW
2053      C ACCELERATIVE COMPONENTS.
2054          UDS(1)=UAS(1)-DUADX/UVAT*WDSREL
2055          VDS(1)=VAS(1)-DVADY/UVAT*WDSREL

```

```

2056      416      CALL DRAG(UDS(1),VDS(1),UAS(1),VAS(1),CDS,RED(1),CD)
2057      HT(1,1)=O.DO
2058      HT(2,1)=O.DO
2059      C CALCULATE STARTING ACCELERATIONS:
2060      IF(EQN.EQ.O)GOTO 417
2061      EQ=1
2062      GOTO 418
2063      417      EQ=O
2064      418      CALL ACCN(UDS(1),VDS(1),UAS(1),VAS(1),RED(1),CD,EQ,O.DO,O)
2065      IF(TRJPRA.EQ.1)WRITE(7,40)
2066      III=1
2067      I=O
2068      IK=O
2069      TRJEND=O
2070      ALMOST=O
2071      DT(1)=DTSS
2072      CLAP=1.D1
2073      XCDL=O.DO
2074      XCDU=O.DO
2075      YCAL=O.DO
2076      YCAU=O.DO
2077      YCDL=O.DO
2078      YCDU=O.DO
2079      SMASH=O
2080      TS(1)=O.DO
2081      PLD=O.DO
2082      C
2083      100      PRD=O.DO
2084      IF(PLD.LT.PLDSTI)GOTO 105
2085      102      PLD=O.DO
2086      C
2087      C INCREMENT INDICES
2088      105      ITEMP=IM4
2089      IM4=IM3
2090      IM3=IM2
2091      IM2=IM1
2092      IM1=IO
2093      IO=IP1
2094      IP1=ITEMP
2095      I=I+1
2096      HFP=HF
2097      C
2098      C INTEGRATE EQNS. OF MOTION
2099      IF(PC.EQ.2)CALL RKF4(EQN,CDS,EPS)
2100      IF(I.GE.4.AND.PC.EQ.1)CALL PC4(EQN,CDS)
2101      IF(I.LT.4.AND.PC.EQ.1.OR.PC.EQ.O)CALL RK4(EQN,CDS)
2102      C
2103      C CALCULATE DISTANCE SINCE LAST PRINT/PLOT OF DROPLET POSITION
2104      DIST=DSQRT((XDS(IP1)-XDS(IO))**2+(YDS(IP1)-YDS(IO))**2)
2105      PRD=PRD+DIST
2106      X=SNGL(XDS(IP1))
2107      XPREV=SNGL(XDS(IO))
2108      IF(X.GT.XMIN)GOTO 190
2109      IF(PRD.GE.PRDSTO)GOTO 230
2110      GOTO 105
2111      190      Y=SNGL(YDS(IP1))
2112      YPREV=SNGL(YDS(IO))
2113      C CHECK FOR OUT-OF-BOUNDS.
2114      IF(Y.GE.YMAX)GOTO 211
2115      IF(BOTH.EQ.O)GOTO 191
2116      IF(Y.LT.YMIN.AND.YPREV.GT.YMIN)GOTO 212
2117      191      IF(X.GE.XMAX)GOTO 213
2118      PLD=PLD+DIST
2119      IF(X.GE.SNGL(XN).AND.X.LE.1.O)GOTO 240
2120      IF(IK.EQ.O.AND.TRJPLA.EQ.1)GOTO 226
2121      IF(PLD.GE.PLDSTI)GOTO 220
2122      IF(PRD.GE.PRDSTI)GOTO 230
2123      GOTO 105
2124      C
2125      C HOW CLOSE IS DROPLET TO AEROFOIL?
2126      C COUNT NUMBER OF STEPS PAST NOSE.
2127      240      ALMOST=ALMOST+1
2128      IF(YDS(IP1).LT.YN)GOTO 310
2129      IF(YDS(IO).GT.YN)GOTO 320

```

```

2130      TSLOPE=(YDS(IP1)-YDS(IO))/(XDS(IP1)-XDS(IO))
2131      IF(DABS(TSLOPE-O.DO).LT.1.D-70)GOTO 320
2132      XI=(YN-YDS(IO))/TSLOPE+XDS(IO)+RDS
2133      IF(XI.GT.XN.AND.XI.LT.XDS(IP1)+RDS/1.D6)GOTO 330
2134      C
2135      C FOR UPPER SFC.:
2136      320      CALL SFC(XDS(IP1),YUS1,1,0,ZZ)
2137      CALL SFC(XDS(IP1)+RDS,YUS2,1,0,ZZ)
2138      USLOPE=DSQRT(RDS*RDS+(YUS2-YUS1)**2)
2139      C PREVIOUS CLOSEST APPROACH X AND Y COORDS.
2140      XCDUPR=XCDU
2141      YCDUPR=YCDU
2142      YCAUPR=YCAU
2143      C CALCULATE DROPLET Y COORD. OF CLOSEST APPROACH
2144      YCDU=YDS(IP1)-RDS*RDS/USLOPE
2145      C CALCULATE DROPLET X COORD. OF CLOSEST APPROACH
2146      XCDU=XDS(IP1)+RDS*(YUS2-YUS1)/USLOPE
2147      C CALCULATE AEROFOIL X AND Y COORDS. OF CLOSEST APPROACH
2148      CALL SFC(XCDU,YCAU,1,0,ZZ)
2149      C STORE CLOSEST APPROACH VALUE
2150      CLAP=DMIN1(CLAP,(YCDU-YCAU))
2151      C CHECK FOR DROPLET-AEROFOIL 'COLLISION'
2152      IF((YCDU-YCAU).LE.RDS*CRIT/1.D2)GOTO 505
2153      IF(PLD.GE.PLDSTI)GOTO 220
2154      IF(PRD.GE.PRDSTI)GOTO 230
2155      GOTO 105
2156      C
2157      C COLLISION FLAGGED:
2158      505      SMASH=1
2159      IF(YCDU.GT.YCAU)GOTO 520
2160      IF(ALMOST.EQ.1)GOTO 500
2161      XL=XCDUPR
2162      YL=YCDUPR
2163      TSLOPE=(YCDU-YCDUPR)/(XCDU-XCDUPR)
2164      PIM2=XDS(IO)
2165      CALL SFC(PIM2,COORD,1,0,ZZ)
2166      FPIM2=F(PIM2)
2167      PIM1=XDS(IO)+RDS
2168      CALL SFC(PIM1,COORD,1,0,ZZ)
2169      FPIM1=F(PIM1)
2170      GOTO 510
2171      C NEAR NOSE COLLISION
2172      500      XL=XDS(IO)+RDS
2173      YL=YDS(IO)
2174      TSLOPE=(YDS(IP1)-YL)/(XDS(IP1)+RDS-XL)
2175      IF(DABS(TSLOPE-O.DO).LT.1.D-70)GOTO 507
2176      PIM2=XN
2177      COORD=YN
2178      FPIM2=F(PIM2)
2179      PIM1=XDS(IP1)
2180      COORD=YUS1
2181      FPIM1=F(PIM1)
2182      GOTO 510
2183      507      XCOLL=XN
2184      YCOLL=YN
2185      LTH=O.DO
2186      GOTO 210
2187      C AN 'ALMOST' COLLISION
2188      520      XCOLL=XDS(IP1)
2189      CALL SFC(XCOLL,YCOLL,1,1,LTH)
2190      GOTO 210
2191      C
2192      C ITERATE TO COLLISION LOCATION USING SECANT METHOD.
2193      510      XCOLL=PIM1-FPIM1*(PIM1-PIM2)/(FPIM1-FPIM2)
2194      IF(XCOLL.GT.XN)GOTO 511
2195      XCOLL=XN
2196      COORD=YN
2197      GOTO 512
2198      511      CALL SFC(XCOLL,COORD,1,0,ZZ)
2199      512      PIM2=PIM1
2200      FPIM2=FPIM1
2201      PIM1=XCOLL
2202      FPIM1=F(XCOLL)
2203      IF(DABS(FPIM1).GT.RDS*CRIT/1.D2)GOTO 510

```

```

2204      CALL SFC(XCOLL,YCOLL,1,1,LTH)
2205      GOTO 210
2206      310  IF(YDS(IO).LT.YN)GOTO 330
2207          TSLOPE=(YDS(IP1)-YDS(IO))/(XDS(IP1)-XDS(IO))
2208          IF(DABS(TSLOPE-O.DO).LT.1.D-70)GOTO 330
2209          XI=(YN-YDS(IO))/TSLOPE+XDS(IO)+RDS
2210          IF(XI.GT.XN.AND.XI.LT.XDS(IP1)+RDS/1.D6)GOTO 320
2211      C
2212      C FOR LOWER SFC.:
2213      330  CALL SFC(XDS(IP1),YLS1,O.O,ZZ)
2214          CALL SFC(XDS(IP1)+RDS,YLS2,O.O,ZZ)
2215          LSLOPE=DSQRT(RDS*RDS+(YLS2-YLS1)**2)
2216      C PREVIOUS CLOSEST APPROACH X AND Y COORDS
2217          XCDLPR=XCDL
2218          YCDLPR=YCDL
2219          YCALPR=YCAL
2220      C CALCULATE DROPLET Y COORD. OF CLOSEST APPROACH
2221          YCDL=YDS(IP1)+RDS*RDS/LSLOPE
2222      C CALCULATE DROPLET X COORD. OF CLOSEST APPROACH
2223          XCDL=XDS(IP1)+RDS*(YLS1-YLS2)/LSLOPE
2224      C CALCULATE AEROFOIL X AND Y COORDS. OF CLOSEST APPROACH
2225          CALL SFC(XCDL,YCAL,O.O,ZZ)
2226      C STORE CLOSEST APPROACH VALUE
2227          CLAP=DSIGN(CLAP,-1.DO)
2228          CLAP=DMAX1(CLAP,(YCDL-YCAL))
2229      C CHECK FOR DROPLET-AEROFOIL 'COLLISION'
2230          IF((YCAL-YCDL).LE.RDS*CRIT/1.D2)GOTO 504
2231          IF(PLD.GE.PLDSTI)GOTO 220
2232          IF(PRD.GE.PRDSTI)GOTO 230
2233          GOTO 105
2234      C
2235      C COLLISION FLAGGED
2236      504  SMASH=1
2237          IF(YCAL.GT.YCDL)GOTO 570
2238          IF(ALMOST.EQ.1)GOTO 550
2239          XL=XCDLPR
2240          YL=YCDLPR
2241          TSLOPE=(YCDL-YCDLPR)/(XCDL-XCDLPR)
2242          PIM2=XDS(IO)
2243          CALL SFC(PIM2,COORD,O.O,ZZ)
2244          FPIM2=F(PIM2)
2245          PIM1=XDS(IO)+RDS
2246          CALL SFC(PIM1,COORD,O.O,ZZ)
2247          FPIM1=F(PIM1)
2248          GOTO 560
2249      C NEAR NOSE COLLISION
2250      550  XL=XDS(IO)+RDS
2251          YL=YDS(IO)
2252          TSLOPE=(YDS(IP1)-YL)/(XDS(IP1)+RDS-XL)
2253          IF(DABS(TSLOPE-O.DO).LT.1.D-70)GOTO 556
2254          PIM2=XN
2255          COORD=YN
2256          FPIM2=F(PIM2)
2257          PIM1=XDS(IP1)
2258          COORD=YLS1
2259          FPIM1=F(PIM1)
2260          GOTO 560
2261      556  XCOLL=XN
2262          YCOLL=YN
2263          LTH=O.DO
2264          GOTO 210
2265      C AN 'ALMOST' COLLISION
2266      570  XCOLL=XDS(IP1)
2267          CALL SFC(XCOLL,YCOLL,O.1,LTH)
2268          GOTO 210
2269      C
2270      C ITERATE TO COLLISION LOCATION USING SECANT METHOD.
2271      560  XCOLL=PIM1-FPIM1*(PIM1-PIM2)/(FPIM1-FPIM2)
2272          IF(XCOLL.GT.XN)GOTO 561
2273          XCOLL=XN
2274          COORD=YN
2275          GOTO 562
2276      561  CALL SFC(XCOLL,COORD,O.O,ZZ)
2277      562  PIM2=PIM1

```

```

2278          FPIM2=FPIM1
2279          PIM1=XCOLL
2280          FPIM1=F(XCOLL)
2281          IF(DABS(FPIM1).GT.RDS*CRIT/1.02)GOTO 560
2282          CALL SFC(XCOLL,YCOLL,0,1,LTH)
2283      C
2284      C END OF TRAJECTORY FLAGGED: COLLISION
2285      210      TRJEND=1
2286              IF(IK.EQ.0.OR.TRJPLA.EQ.0)GOTO 230
2287              IK=IK+1
2288              XDSP(IK)=SNGL(XCOLL)
2289              YDSP(IK)=SNGL(YCOLL)
2290              GOTO 230
2291      C END OF TRAJECTORY FLAGGED: EXCEEDED YMAX
2292      211      TRJEND=1
2293              IF(IK.EQ.0.OR.TRJPLA.EQ.0)GOTO 230
2294              IK=IK+1
2295              XDSP(IK)=(X-XPREV)/(Y-YPREV)*(YMAX-YPREV)+XPREV
2296              YDSP(IK)=YMAX
2297              GOTO 230
2298      C END OF TRAJECTORY FLAGGED: EXCEEDED YMIN
2299      212      TRJEND=1
2300              IF(IK.EQ.0.OR.TRJPLA.EQ.0)GOTO 230
2301              IK=IK+1
2302              XDSP(IK)=(X-XPREV)/(Y-YPREV)*(YMIN-YPREV)+XPREV
2303              YDSP(IK)=YMIN
2304              GOTO 230
2305      C END OF TRAJECTORY FLAGGED: EXCEEDED XMAX
2306      213      TRJEND=1
2307              IF(IK.EQ.0.OR.TRJPLA.EQ.0)GOTO 230
2308              IK=IK+1
2309              XDSP(IK)=XMAX
2310              YDSP(IK)=(Y-YPREV)/(X-XPREV)*(XMAX-XPREV)+YPREV
2311              GOTO 230
2312      C
2313      C STORE PLOT COORDINATES FOR FIRST POINT WITHIN WINDOW
2314      226      IK=IK+1
2315              XDSP(IK)=XMIN
2316              YDSP(IK)=(Y-YPREV)/(X-XPREV)*(XMIN-XPREV)+YPREV
2317              GOTO 230
2318      C STORE COORDS FOR LATER PLOTTING
2319      220      IF(TRJPLA.EQ.0)GOTO 230
2320              IK=IK+1
2321              XDSP(IK)=SNGL(XDS(IO))
2322              YDSP(IK)=SNGL(YDS(IO))
2323      230      IF(TRJPRA.EQ.0.AND.TRJEND.EQ.0)GOTO 100
2324              IF(PRD.LT.PRDSTI.AND.TRJEND.EQ.0)GOTO 102
2325      C
2326      C PRINT INTERVAL EXCEEDED
2327              TTLACN=DSQRT(AN(1,IO)*AN(1,IO)+AN(2,IO)*AN(2,IO))
2328              VPSQ=UDS(IO)*UDS(IO)+VDS(IO)*VDS(IO)
2329              NA=RDS*TTLACN/DTS(IO)/VPSQ
2330              IF(TRJPRA.EQ.0)GOTO 181
2331      C
2332      C PRINT TRAJECTORY INFO
2333              IF(PC.EQ.1.AND.I.GT.4)GOTO 235
2334              WRITE(7,50)I,TS(I),DTS(IO),XDS(IO),YDS(IO),PSI(5),UAS(IO),
2335              UDS(IO),VAS(IO),VDS(IO),RED(IO),NA,HFP
2336              IF(TRJEND.EQ.0)GOTO 100
2337              GOTO 225
2338      235      WRITE(7,50)I,TS(I),DTS(IO),XDS(IO),YDS(IO),PSI(5),UAS(IO),
2339              UDS(IO),VAS(IO),VDS(IO),RED(IO),NA,HFP,UST,VST
2340              IF(TRJEND.EQ.0)GOTO 100
2341      225      I=I+1
2342              WRITE(7,50)I,TS(I),DTS(IP1),X,Y
2343      181      IF(TRJPLA.EQ.0)GOTO 180
2344      C
2345      C PLOT TRAJECTORIES:
2346              XDSP(IK+1)=XMIN
2347              XDSP(IK+2)=(XMAX-XMIN)/21.0
2348              YDSP(IK+1)=YMIN
2349              YDSP(IK+2)=(YMAX-YMIN)/10.5
2350              CALL LINE(XDSP,YDSP,IK,1,0,0)
2351      180      IF(SMASH.EQ.1)GOTO 195

```

```

2352      WRITE(6,60)CLAP,I,PSI(5)
2353      WRITE(7,60)CLAP,I,PSI(5)
2354      GOTO 196
2355 195    WRITE(6,80)XCOLL,YCOLL,LTH,I
2356      WRITE(7,80)XCOLL,YCOLL,LTH,I
2357 196    IF(AT.EQ.0)GOTO 200
2358      IF(GRAZE.EQ.0)GOTO 630
2359      IF(SMASH.EQ.1)GOTO 610
2360      C
2361      C ITERATE TOWARD THE GRAZING TRAJECTORY
2362          IF(IG.EQ.1)GOTO 600
2363          IF(DABS(CLAP).LE.TOL)K=K1
2364      C FIND NEW YO POSITION BY USING THE SECANT METHOD TO ESTIMATE THE
2365      C LOCATION OF YO AT GRAZING
2366          SLP=(YOT(IG,IJ)-YOT(IG-1,IJ))/(CLAP-CLAPP)
2367          IF(DABS(SLP).LT.1.2DO.OR.IG.LT.3)GOTO 340
2368          SLP=(YOT(IG,IJ)-YOT(IG-2,IJ))/(CLAP-CLAPP)
2369          K=K1
2370 340    YOT(IG+1,IJ)=YOT(IG,IJ)-K*CLAP*SLP
2371      C SET PREVIOUS CLOSEST APPROACH
2372          CLAPP=CLAP
2373          CLAP=CLAP
2374          IG=IG+1
2375          YDS(1)=YOT(IG,IJ)
2376          GOTO 405
2377      C AFTER FIRST MISSING TRAJECTORY, ESTIMATE NEW YO VIA CLAP
2378 600    YOT(1,IJ)=YO(IJ)
2379          IF(DABS(CLAP).LE.TOL)K=K1
2380          YOT(2,IJ)=YOT(1,IJ)-K*CLAP
2381          CLAP=CLAP
2382          IG=2
2383          YDS(1)=YOT(2,IJ)
2384          GOTO 405
2385      C
2386      C THIS IS THE GRAZING TRAJECTORY
2387 610    IF(IG.GT.1)GOTO 625
2388          WRITE(8,90)
2389      C ADJUST FIRST TRAJECTORY TO BE A NEAR MISS
2390          IF(IJ.EQ.2)GOTO 605
2391          YO(1)=YO(1)+5.D-4
2392          GOTO 606
2393 605    YO(1)=YO(1)-5.D-4
2394 606    YDS(1)=YO(1)
2395          GOTO 405
2396 625    GRAZE=0
2397          YOG=YOT(IG,IJ)
2398          YOT(1,IJ)=YOG
2399          YCG=YCOLL
2400          LG=LTH
2401          LP1=LG
2402      C
2403      C THESE ARE COLLIDING TRAJECTORIES
2404 630    IF(SMASH.EQ.1)GOTO 635
2405          WRITE(6,95)
2406          IC=IC-1
2407          GOTO 640
2408 635    IF(IC.EQ.1)GOTO 800
2409          IF(DABS(DSIGN(YCOLL-YN,YCG-YN)-YCOLL+YN).GT.1.D-10)GOTO 645
2410          IF(INT.EQ.0)GOTO 810
2411          INT=0
2412          LT(IC,IJ)=LTH
2413          IC=IC+1
2414          GOTO 820
2415 810    IF(LT(IC-1,IJ)-LTH.LE.1.35DO*LINT)GOTO 800
2416          LT(IC+1,IJ)=LTH
2417          YOT(IC+1,IJ)=YOT(IC,IJ)
2418          INT=1
2419          YOT(IC,IJ)=0.6DO*YOT(IC-1,IJ)+0.4DO*YOT(IC,IJ)
2420          YDS(1)=YOT(IC,IJ)
2421          GOTO 405
2422 800    LT(IC,IJ)=LTH
2423 820    IF(IC.GT.1)GOTO 633
2424          IF(BOTH.EQ.0)GOTO 631
2425          IF(IJ.EQ.2)GOTO 632

```

```

2426 C ESTIMATED INTERVAL IN L BETWEEN COLLISIONS.
2427 LINT=LG/(DFLOAT(NTRAJU)+1.DO)
2428 ADD=-0.5DO
2429 GOTO 633
2430 632 LINT=LG/(DFLOAT(NTRAJL)+1.DO)
2431 ADD=-0.5DO
2432 GOTO 633
2433 631 LINT=LG/(DFLOAT(NTRAJU)+0.5DO)
2434 ADD=0.5DO
2435 633 LP1=LP1-LINT
2436 IF(IC.EQ.1)LP1=LP1-ADD*LINT
2437 IF(DABS(DSIGN(LP1,LTH)-LP1).GT.1.D-10.OR.DABS(LP1).LT.1.D-10)
2438 GOTO 640
2439 IC=IC+1
2440 C
2441 C ESTIMATE NEW YO TO SPREAD POINTS EVENLY ALONG CE VS L CURVE
2442 IF (BOTH.EQ.1)GOTO 620
2443 YOT(IC,IJ)=2.DO*YOG/LG*LP1*(1.DO-LP1/2.DO/LG)
2444 YDS(1)=YOT(IC,IJ)
2445 GOTO 405
2446 620 IF(IJ.EQ.2)GOTO 850
2447 YOT(IC,1)=BETAO*LP1*(1.DO-LP1/2.DO/LG)
2448 +YOG-BETAO*LG/2.DO
2449 GOTO 860
2450 850 YOT(IC,2)=-BETAO*LP1*(1.DO-LP1/2.DO/LG)
2451 +YOG+BETAO*LG/2.DO
2452 IF(YOT(IC,2).LT.YOT(ICU,1))GOTO 860
2453 IC=IC-1
2454 GOTO 640
2455 860 YDS(1)=YOT(IC,IJ)
2456 GOTO 405
2457 640 IF(IJ.EQ.1)ICU=IC
2458 IF(IJ.EQ.2)ICL=IC
2459 GOTO 200
2460 645 LT(IC,IJ)=LTH
2461 IF(IJ.EQ.2)GOTO 646
2462 ICU=IC-1
2463 UX=1
2464 GOTO 200
2465 646 ICL=IC-1
2466 LX=1
2467 200 CONTINUE
2468 C
2469 C TRANSFER COLLISION INFO TO SINGLE MONOTONICALLY INCREASING
2470 C (IN L) VECTORS
2471 IF(BOTH.EQ.1)GOTO 660
2472 IF(DABS(LT(ICU,1)-0.DO).GT.1.D-4)GOTO 651
2473 ICU=ICU-1
2474 651 YO(ICU+1)=0.DO
2475 L(ICU+1)=0.DO
2476 DO 650 I=1,ICU
2477 IU=2*ICU+2-I
2478 YO(I)=-YOT(I,1)
2479 YO(IU)=YOT(I,1)
2480 L(I)=-LT(I,1)
2481 L(IU)=LT(I,1)
2482 650 CONTINUE
2483 ICT=2*ICU+1
2484 ICL=ICU+1
2485 GOTO 690
2486 660 IF(UX.EQ.1)YOTUX=YOT(ICU+1,1)
2487 IF(LX.EQ.1)YOTLX=YOT(ICL+1,2)
2488 II=0
2489 DO 670 I=1,ICL
2490 IF(UX.NE.1)GOTO 665
2491 IF(YOTUX.GE.YOT(I,2))GOTO 665
2492 IF(DABS(YOTUX-YOT(I,2)).LT.1.D-5)GOTO 666
2493 IF(DABS(YOTUX-YOT(I-1,2)).LT.1.D-5)GOTO 666
2494 II=II+1
2495 YO(II)=YOTUX
2496 L(II)=-LT(ICU+1,1)
2497 666 UX=0
2498 665 II=II+1
2499 YO(II)=YOT(I,2)

```

```

2500      L(II)=-LT(I,2)
2501 670      CONTINUE
2502      IF(UX.NE.1)GOTO 667
2503      IF(DABS(YOTUX-YOT(ICL,2)).LT.1.D-5)GOTO 667
2504      II=II+1
2505      YO(II)=YOTUX
2506      L(II)=-LT(ICU+1,1)
2507 667      ICLL=ICL
2508      ICL=II
2509      DO 680 I=1,ICU
2510      IU=ICU+1-I
2511      IF(LX.NE.1)GOTO 675
2512      IF(YOTLX.GE.YOT(IU,1))GOTO 675
2513      IF(DABS(YOTLX-YOT(IU,1)).LT.1.D-5)GOTO 676
2514      IF(DABS(YOTLX-YOT(IU+1,1)).LT.1.D-5)GOTO 676
2515      II=II+1
2516      YO(II)=YOTLX
2517      L(II)=LT(ICLL+1,2)
2518 676      LX=0
2519 675      II=II+1
2520      YO(II)=YOT(IU,1)
2521      L(II)=LT(IU,1)
2522 680      CONTINUE
2523      ICT=II
2524      ICU=ICT-ICL
2525 690      RETURN
2526      END
2527      C
2528      C
2529      SUBROUTINE ACCN(UD,VD,UA,VA,RED,CD,EQN,T,G)
2530      C
2531      C WRITTEN BY: M. OLESKIW ON: 801216 LAST MODIFIED:801223
2532      C
2533      C CALCULATES RHS OF NON-DIMENSIONAL EQNS. OF MOTION
2534      C
2535      DOUBLE PRECISION RELVEL,RED,NUS,RDS,APU,APV,BPU,BPV
2536      ,AN(2,6),HF,HX,HY,HT(2,6),DSQRT,AU,AV,BU,BV,RHOA,
2537      ,RHOD,GS,ALPHAR,PI,CD,UD,VD,UA,VA,TS(500),DTS(6),T
2538      C
2539      INTEGER EQN,G,I,IM4,IM3,IM2,IM1,IO,IP1
2540      C
2541      COMMON ALPHAR,PI/EQNMN/GS,RHOA,RHOD,RDS,NUS,HF
2542      ,/INTEG/AN,HT/LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1
2543      C
2544      C IN UD=
2545      C IN VD=DROPLET VELOCITY COMPONENTS.
2546      C IN UA=
2547      C IN VA=AIR VELOCITY COMPONENTS.
2548      C IN RED=RELATIVE MOTION REYNOLDS NO.
2549      C IN CD=DRAG COEFFICIENT.
2550      C IN EQN=PARAMETER TO DETERMINE TERMS USED IN EQN. OF MOTION.
2551      C IN T=TIME AT THIS TIME STEP.
2552      C IN G=0:EXTRAPOLATE HISTORY TERM SEQUENCE.
2553      C IN 1:CALCULATE NEW HISTORY TERM VALUE.
2554      C
2555      RELVEL=RED*NUS/RDS/2.DO
2556      IF(EQN.EQ.0)GOTO 100
2557      C
2558      C FIRST TWO TERMS IN EQN. OF MOTION INCLUDING GRAVITATION AND
2559      C STEADY STATE DRAG. (INCLUDES BUOYANCY AND INDUCED MASS EFFECTS)
2560      APU=2.DO*(RHOD-RHOA)/(2.DO*RHOD+RHOA)*GS*DSIN(ALPHAR)
2561      APV=2.DO*(RHOD-RHOA)/(2.DO*RHOD+RHOA)*GS*DCOS(ALPHAR)
2562      BPU=0.75DO*CD*RHOA/RDS/(2.DO*RHOD+RHOA)
2563      ,*(UD-UA)*RELVEL
2564      BPV=0.75DO*CD*RHOA/RDS/(2.DO*RHOD+RHOA)
2565      ,*(VD-VA)*RELVEL
2566      AN(1,IP1)=APU-BPU
2567      AN(2,IP1)=-APV-BPV
2568      IF(EQN.EQ.2)GOTO 300
2569      HF=0.DO
2570      RETURN
2571      C
2572      C THIRD (HISTORY) TERM FOR SHEDDING OF VORTICITY
2573      300 CALL HIST(T,G)

```

```

2574      HX=-9.DO*RHOA/(2.DO*RHOD+RHOA)/RDS*DSQRT(NUS/PI)*HT(1,IP1)
2575      HY=-9.DO*RHOA/(2.DO*RHOD+RHOA)/RDS*DSQRT(NUS/PI)*HT(2,IP1)
2576      AN(1,IP1)=AN(1,IP1)+HX
2577      AN(2,IP1)=AN(2,IP1)+HY
2578      IF(G.EQ.O)RETURN
2579      HF=DSQRT((HX*HX+HY*HY)/((APU-BPU)**2+(APV+BPV)**2))
2580      RETURN
2581      C
2582      C FIRST TWO TERMS IN EQN. OF MOTION WITHOUT BUOYANCY AND INDUCED MASS
2583      100      AU=GS*DSIN(ALPHAR)
2584      AV=GS*DCOS(ALPHAR)
2585      BU=O.375DO*RHOA/RHOD*CD/RDS*(UD-UA)*RELVEL
2586      BV=O.375DO*RHOA/RHOD*CD/RDS*(VD-VA)*RELVEL
2587      AN(1,IP1)=AU-BU
2588      AN(2,IP1)=-AV-BV
2589      HF=O.DO
2590      RETURN
2591      END
2592      C
2593      C
2594      SUBROUTINE AIRVEL(X,Y,UAS,VAS,NP)
2595      C
2596      C WRITTEN BY: M. OLESKIW ON:80Q222 LAST MODIFIED:801216
2597      C
2598      C CALCULATES THE AIR VELOCITY COMPONENTS AT A GIVEN LOCATION
2599      C
2600      DOUBLE PRECISION X,Y,UAS,VAS,XP(7),YP(7),XC(101),YC(101)
2601      ,DEL,GAMMA(101),D(100),K(101),PI,PJKA,PJKE,
2602      ,SI(100),CO(100),PSI(7),DXC,DYC,DELTA,A,B,R1S,R2S,
2603      ,R3S,DATAN,T3,DABS,DSIGN,ALPHAR,T1,T2,DLOG,R,DCOS,DSIN
2604      C
2605      INTEGER L,NP,J,NCOU,NCOL,N,TYPE
2606      C
2607      COMMON ALPHAR,PI/AERO3/NCOU,NCOL/AERO2/XC,YC,GAMMA,D,SI,CO
2608      ,/AIR/XP,YP,DEL,PSI,TYPE
2609      C
2610      C IN X=
2611      C IN Y=COORDS. AT WHICH AIR VELOCITY IS TO BE DETERMINED.
2612      C OUT UAS=
2613      C OUT VAS=COMPONENTS OF AIR VELOCITY.
2614      C IN NP=NUMBER OF POINTS AT WHICH TO CALCULATE PSI.
2615      C
2616      N=NCOU+NCOL-2
2617      C SET GRID FOR AIR VELOCITY CALCULATIONS
2618      XP(1)=X+DEL
2619      XP(2)=X-DEL
2620      XP(3)=X
2621      XP(4)=X
2622      XP(5)=X
2623      YP(1)=Y
2624      YP(2)=Y
2625      YP(3)=Y+DEL
2626      YP(4)=Y-DEL
2627      YP(5)=Y
2628      DO 110 J=1,NP
2629      IF(TYPE.EQ.-1)GOTO 115
2630      PSI(J)=O.O
2631      DO 120 L=1,N
2632      C FIND DISTANCE BETWEEN CONTROL PT. L AND GRID PT. I,J.
2633      DXC=XP(J)-XC(L)
2634      DYC=YP(J)-YC(L)
2635      C CALCULATE COMPONENTS OF EQN. 9 AND FIG. 2
2636      DELTA=D(L)/2.DO
2637      B=DXC*CO(L)+DYC*SI(L)
2638      A=DYC*CO(L)-DXC*SI(L)
2639      R1S=A*A+(B+DELTA)*(B+DELTA)
2640      R2S=A*A+(B-DELTA)*(B-DELTA)
2641      R3S=A*A+B*B-DELTA*DELTA
2642      IF(R3S.LT.1.D-30)GO TO 130
2643      T3=DATAN(2.DO*A*DELTA/R3S)
2644      GO TO 140
2645      130      IF(DABS(A).LT.1.D-30)GO TO 150
2646      T3=DATAN((B+DELTA)/A)-DATAN((B-DELTA)/A)
2647      GO TO 140

```

```

2648      150      T3=DSIGN(PI,A)
2649      140      T1=(B+DELTA)*DLOG(R1S)
2650      T2=(B-DELTA)*DLOG(R2S)
2651      K(L)=(T1-T2+2.DO*A*T3-4.DO*DELTA)/4.DO/PI
2652      PSI(J)=PSI(J)-GAMMA(L)*K(L)
2653      120      CONTINUE
2654      R=YP(J)*DCOS(ALPHAR)-XP(J)*DSIN(ALPHAR)
2655      C ASSURE THAT PSI ON AEROFOIL = 0.
2656      PSI(J)=PSI(J)+R-GAMMA(N+1)
2657      GOTO 110
2658      115      PSI(J)=YP(J)-YP(J)/4.DO/((XP(J)-5.D-1)**2+YP(J)*YP(J))
2659      110      CONTINUE
2660      C
2661      C CALCULATE AIRSPEED FROM STREAMFN.
2662      UAS=(PSI(3)-PSI(4))/2.DO/DEL
2663      VAS=(PSI(2)-PSI(1))/2.DO/DEL
2664      RETURN
2665      END
2666      C
2667      C
2668      SUBROUTINE DRAG(UDS,VDS,UAS,VAS,CDS,RED,CD)
2669      C
2670      C WRITTEN BY: M. OLESKIW ON:800222 LAST MODIFIED:801216
2671      C
2672      C CALCULATES THE REYNOLDS NUMBER AND DRAG COEFFICIENT OF THE DROPLET
2673      C
2674      DOUBLE PRECISION DSQRT,UDS,VDS,UAS,VAS,RED,CD,
2675      .GS,RHOA,RHOD,RDS,NUS,HF
2676      C
2677      INTEGER CDS
2678      C
2679      COMMON /EQNMN/GS,RHOA,RHOD,RDS,NUS,HF
2680      C
2681      C IN UDS=
2682      C IN VDS=DROPLET VELOCITY COMPONENTS.
2683      C IN UAS=
2684      C IN VAS=AIR VELOCITY COMPONENTS.
2685      C IN CDS=PARAMETER TO DETERMINE DRAG COEFFICIENT FORMULATION.
2686      C OUT RED=RELATIVE MOTION REYNOLDS NO.
2687      C OUT CD=DRAG COEFFICIENT.
2688      C
2689      RED=DSQRT((UDS-UAS)**2+(VDS-VAS)**2)*2.DO*RDS/NUS
2690      IF(CDS.EQ.2)GOTO 300
2691      IF(CDS.EQ.1.AND.RED.LE.5.DO)GOTO 100
2692      C
2693      C STEADY STATE DRAG COEFFICIENT OF DROPLET FOR RED < 5000
2694      C ABRAHAM (1970)
2695      CD=0.2924DO*(1+9.06DO/DSQRT(RED))**2
2696      RETURN
2697      100 IF(RED.GE.1.D-2)GOTO 200
2698      C
2699      C STEADY STATE STOKES DRAG FOR RED < 0.01
2700      CD=24.DO/RED
2701      RETURN
2702      C
2703      C STEADY STATE DRAG COEFFICIENT FOR 0.01 < RED < 5 - SARTOR
2704      C AND ABBOTT (1975)
2705      200 CD=24.DO/RED+2.2DO
2706      RETURN
2707      C
2708      C STEADY STATE DRAG COEFFICIENT - LANGMUIR & BLODGETT (1945)
2709      300 CD=24.DO/RED+4.73DO/RED**0.37DO+6.24D-3*RED**0.38DO
2710      RETURN
2711      END
2712      C
2713      C
2714      SUBROUTINE HIST(T,G)
2715      C
2716      C WRITTEN BY: M. OLESKIW ON:801216 LAST MODIFIED:801222
2717      C
2718      C DETERMINES VALUE OF INTEGRAL IN HISTORY TERM FOR U COMPONENT EQN.
2719      C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978)
2720      C NUMERICAL ANALYSIS P. 90 QA 297.B84
2721      C

```

```

2722      DOUBLE PRECISION TAU3,TAU2,TAU1,TAUO,P11,P10,P21,P22,P20,
2723      P33,P32,P31,P30,TO,T1,T2,T3,TS(500),FO,F1,F2,F3,DSQRT,DTS(6),
2724      HT(2,6),T,A,B,C,D,F,AN(2,6),P(2,745),Z2,Z33,Z32,Z31,Z30,AA,BB
2725      C
2726      INTEGER J,I,L,FF,E,MOD,JI,UJ,G,I,IM4,IM3,IM2,IM1,IO,IP1
2727      C
2728      COMMON /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1/INTEG/AN,HT
2729      C
2730      C IN  T=TIME AT PRESENT TIME STEP.
2731      C IN  G=0:EXTRAPOLATE HISTORY TERM SEQUENCE.
2732      C IN  1:CALCULATE NEW HISTORY TERM.
2733      C
2734      C STATEMENT FNS. TO EVALUATE PORTIONS OF THE INTEGRAL.
2735      TAU3(A,B)=((5.DO*A**3+6.DO*A*A*T+8.DO*A*T*T+16.DO*T**3)
2736      *DSQRT(T-A)-(5.DO*B**3+6.DO*B*B*T+8.DO*B*T*T+16.DO*T**3)
2737      *DSQRT(T-B))*2.DO/35.DO
2738      TAU2(A,B)=((3.DO*A*A+4.DO*A*T+8.DO*T*T)*DSQRT(T-A)
2739      -(3.DO*B*B+4.DO*B*T+8.DO*T*T)*DSQRT(T-B))*2.DO/15.DO
2740      TAU1(A,B)=((2.DO*T+A)*DSQRT(T-A)-(2.DO*T+B)*DSQRT(T-B))*2.DO/3.DO
2741      TAUO(A,B)=2.DO*(DSQRT(T-A)-DSQRT(T-B))
2742      C
2743      C STATEMENT FNS. TO FIND THE TERMS OF THE LAGRANGE POLY. FIT.
2744      P11(TO)=(F1-FO)/(T1-TO)
2745      P10(TO)=(FO*T1-F1*TO)/(T1-TO)
2746      Z2(A,B,C,F)=F/(A-B)/(A-C)
2747      P22(TO)=Z2(TO,T1,T2,FO)+Z2(T1,TO,T2,F1)+Z2(T2,TO,T1,F2)
2748      P21(TO)=-(T1+T2)*Z2(TO,T1,T2,FO)-(TO+T2)*Z2(T1,TO,T2,F1)
2749      -(TO+T1)*Z2(T2,TO,T1,F2)
2750      P20(TO)=T1*T2*Z2(TO,T1,T2,FO)+TO*T2*Z2(T1,TO,T2,F1)
2751      +TO*T1*Z2(T2,TO,T1,F2)
2752      Z33(A,B,C,D,F)=F/(A-B)/(A-C)/(A-D)
2753      P33(TO)=Z33(TO,T1,T2,T3,FO)+Z33(T1,TO,T2,T3,F1)
2754      +Z33(T2,TO,T1,T3,F2)+Z33(T3,TO,T1,T2,F3)
2755      Z32(A,B,C,D,F)=-(B+C*D)*F/(A-B)/(A-C)/(A-D)
2756      P32(TO)=Z32(TO,T1,T2,T3,FO)+Z32(T1,TO,T2,T3,F1)
2757      +Z32(T2,TO,T1,T3,F2)+Z32(T3,TO,T1,T2,F3)
2758      Z31(A,B,C,D,F)=(B*C+B*D+C*D)*F/(A-B)/(A-C)/(A-D)
2759      P31(TO)=Z31(TO,T1,T2,T3,FO)+Z31(T1,TO,T2,T3,F1)
2760      +Z31(T2,TO,T1,T3,F2)+Z31(T3,TO,T1,T2,F3)
2761      Z30(A,B,C,D,F)=-B*C*D*F/(A-B)/(A-C)/(A-D)
2762      P30(TO)=Z30(TO,T1,T2,T3,FO)+Z30(T1,TO,T2,T3,F1)
2763      +Z30(T2,TO,T1,T3,F2)+Z30(T3,TO,T1,T2,F3)
2764      C
2765      IF(G.EQ.1)GOTO 200
2766      C EXTRAPOLATION OF HISTORY TERM SEQUENCE
2767      GOTO(140,120,100),I
2768      TO=TS(I-3)
2769      T1=TS(I-2)
2770      T2=TS(I-1)
2771      T3=TS(I)
2772      DO 110 J=1,2
2773      FO=HT(J,IM3)
2774      F1=HT(J,IM2)
2775      F2=HT(J,IM1)
2776      F3=HT(J,IO)
2777      HT(J,IP1)=P33(TO)*T**3+P32(TO)*T*T+P31(TO)*T+P30(TO)
2778      110  CONTINUE
2779      RETURN
2780      C
2781      100  TO=TS(1)
2782      T1=TS(2)
2783      T2=TS(3)
2784      DO 130 J=1,2
2785      FO=HT(J,IM2)
2786      F1=HT(J,IM1)
2787      F2=HT(J,IO)
2788      HT(J,IP1)=P22(TO)*T*T+P21(TO)*T+P20(TO)
2789      130  CONTINUE
2790      RETURN
2791      C
2792      120  TO=TS(1)
2793      T1=TS(2)
2794      DO 150 J=1,2
2795      FO=HT(J,IM1)

```

```

2796          F1=HT(J,IO)
2797          HT(J,IP1)=P11(TO)*T+P10(TO)
2798          150    CONTINUE
2799          RETURN
2800      C
2801      140    HT(1,IP1)=0.DO
2802          HT(2,IP1)=0.DO
2803          RETURN
2804      C
2805      200    L=(I-4)/2*3+1
2806          HT(1,IP1)=0.DO
2807          HT(2,IP1)=0.DO
2808          GOTO(400,500,600,700),I
2809          FF=MOD(I,2)
2810          E=I-5+FF
2811      C EVALUATE INTEGRAL UP TO LAST SEVERAL INTERVALS
2812          DO 210 J=1,E,2
2813              AA=TS(J)
2814              BB=TS(J+2)
2815              JI=(J-1)/2*3+1
2816              DO 220 JJ=1,2
2817                  HT(JJ,IP1)=HT(JJ,IP1)+P(JJ,JI)*TAU2(AA,BB)
2818                  +P(JJ,JI+1)*TAU1(AA,BB)+P(JJ,JI+2)*TAU0(AA,BB)
2819          220    CONTINUE
2820          210    CONTINUE
2821          IF(FF.EQ.1)GOTO 600
2822      C EVALUATE INTEGRAL FOR LAST 4 INTERVALS
2823      C    USING TWO INTERVAL PAIRS (FOR I EVEN)
2824          700    TO=TS(I-3)
2825          T1=TS(I-2)
2826          T2=TS(I-1)
2827          DO 710 J=1,2
2828              FO=AN(J,IM3)
2829              F1=AN(J,IM2)
2830              F2=AN(J,IM1)
2831      C FIT A 2ND ORDER LAGRANGE POLYNOMIAL
2832          P(J,L)=P22(TO)
2833          P(J,L+1)=P21(TO)
2834          P(J,L+2)=P20(TO)
2835          HT(J,IP1)=HT(J,IP1)+P(J,L)*TAU2(TO,T2)+P(J,L+1)*TAU1(TO,T2)
2836          +P(J,L+2)*TAU0(TO,T2)
2837          710    CONTINUE
2838      C FOR THE SECOND PAIR OF THE SET
2839      C    (OR FOR THE VERY FIRST PAIR OF INTERVALS)
2840          500    TO=TS(I-1)
2841          T1=TS(I)
2842          T2=TS(I+1)
2843          DO 720 J=1,2
2844              FO=AN(J,IM1)
2845              F1=AN(J,IO)
2846              F2=AN(J,IP1)
2847          HT(J,IP1)=HT(J,IP1)+P22(TO)*TAU2(TO,T2)+P21(TO)*TAU1(TO,T2)
2848          +P20(TO)*TAU0(TO,T2)
2849          720    CONTINUE
2850          RETURN
2851      C
2852      C EVALUATE INTEGRAL FOR LAST 3 INTERVALS (FOR I ODD)
2853          600    TO=TS(I-2)
2854          T1=TS(I-1)
2855          T2=TS(I)
2856          T3=TS(I+1)
2857          DO 610 J=1,2
2858              FO=AN(J,IM2)
2859              F1=AN(J,IM1)
2860              F2=AN(J,IO)
2861              F3=AN(J,IP1)
2862          HT(J,IP1)=HT(J,IP1)+P33(TO)*TAU3(TO,T3)+P32(TO)*TAU2(TO,T3)
2863          +P31(TO)*TAU1(TO,T3)+P30(TO)*TAU0(TO,T3)
2864          610    CONTINUE
2865          RETURN
2866      C
2867      C EVALUATE INTEGRAL FOR THE FIRST INTERVAL
2868          400    TO=TS(1)
2869          T1=TS(2)

```

```

2870      DD 410 J=1,2
2871      FO=AN(J,IO)
2872      F1=AN(J,IP1)
2873      HT(J,IP1)=HT(J,IP1)+P11(TO)*TAU1(TO,T1)+P10(TO)*TAUO(TO,T1)
2874 410      CONTINUE
2875      RETURN
2876      END
2877      C
2878      C
2879      SUBROUTINE RK4(EQN,CDS,EPS)
2880      C
2881      C WRITTEN BY: M. OLESKIW ON: 800227 LAST MODIFIED:801227
2882      C
2883      C INTEGRATE THE DROPLET EQNS. OF MOTION (IN X AND Y) USING
2884      C THE 4TH ORDER RUNGE-KUTTA-FEHLBERG TECHNIQUE.
2885      C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978),
2886      C NUMERICAL ANALYSIS, P.254, QA 297.B84
2887      C
2888      DOUBLE PRECISION EPS,XDS(6),UDS(6),AN(2,6),YDS(6),
2889      .VDS(6),HT(2,6),DTS(6),UAS(6),VAS(6),RED(6),CD,RE,
2890      .K1,K2,K3,K4,K5,K6,L1,L2,L3,L4,L5,L6,M1,M2,M3,M4,M5,M6,
2891      .N1,N2,N3,N4,N5,N6,UA,VA,RMAX,DMAX1,DMIN,DMIN1,XDEL,YDEL,
2892      .UDEL,VDEL,DABS,XR,YR,UR,VR,XT,YT,UT,VT,
2893      .XD,YD,UD,VD,CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,
2894      .C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,TS(500)
2895      DOUBLE PRECISION DMINP
2896      C
2897      INTEGER EQN,CDS,I,IM4,IM3,IM2,IM1,IO,IP1,K
2898      C
2899      COMMON /PV/XDS,YDS,UDS,VDS/INTEG/AN,HT
2900      . /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1
2901      . /REL/UAS,VAS,RED,CD
2902      . /RKFM/CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,
2903      . C15,C16,C17,C18,C19,C20,C21,C22,C23,C24
2904      C
2905      C IN EQN=DENOTES PORTION OF TOTAL SYSTEM OF EQUATIONS TO BE SOLVED.
2906      C IN CDS=TYPE OF DRAG COEFFICIENT TO BE USED.
2907      C IN EPS=LOCAL ERROR PARAMETER.
2908      C
2909      IF(DABS(DMINP).LT.1.D-70)DMINP=1.01D0
2910 100      TS(I+1)=TS(I)+DTS(IO)
2911      K1=DTS(IO)*UDS(IO)
2912      L1=DTS(IO)*VDS(IO)
2913      M1=DTS(IO)*AN(1,IO)
2914      N1=DTS(IO)*AN(2,IO)
2915      XD=XDS(IO)+CC1*K1
2916      YD=YDS(IO)+CC1*L1
2917      UD=UDS(IO)+CC1*M1
2918      VD=VDS(IO)+CC1*N1
2919      CALL AIRVEL(XD,YD,UA,VA,4)
2920      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
2921      C
2922      K2=DTS(IO)*UD
2923      L2=DTS(IO)*VD
2924      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I)+DTS(IO)/4.DO,0)
2925      M2=DTS(IO)*AN(1,IP1)
2926      N2=DTS(IO)*AN(2,IP1)
2927      XD=XDS(IO)+CC2*K1+C3*K2
2928      YD=YDS(IO)+CC2*L1+C3*L2
2929      UD=UDS(IO)+CC2*M1+C3*M2
2930      VD=VDS(IO)+CC2*N1+C3*N2
2931      CALL AIRVEL(XD,YD,UA,VA,4)
2932      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
2933      C
2934      K3=DTS(IO)*UD
2935      L3=DTS(IO)*VD
2936      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I)+DTS(IO)*3.75D-1,0)
2937      M3=DTS(IO)*AN(1,IP1)
2938      N3=DTS(IO)*AN(2,IP1)
2939      XD=XDS(IO)+C4*K1-C5*K2+C6*K3
2940      YD=YDS(IO)+C4*L1-C5*L2+C6*L3
2941      UD=UDS(IO)+C4*M1-C5*M2+C6*M3
2942      VD=VDS(IO)+C4*N1-C5*N2+C6*N3
2943      CALL AIRVEL(XD,YD,UA,VA,4)

```

```

2944      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
2945      C
2946      K4=DTS(IO)*UD
2947      L4=DTS(IO)*VD
2948      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I)+12.DO/13.DO
2949      *DTS(IO),O)
2950      M4=DTS(IO)*AN(1,IP1)
2951      N4=DTS(IO)*AN(2,IP1)
2952      XD=XDS(IO)+C7*K1-C8*K2+C9*K3-C10*K4
2953      YD=YDS(IO)+C7*L1-C8*L2+C9*L3-C10*L4
2954      UD=UDS(IO)+C7*M1-C8*M2+C9*M3-C10*M4
2955      VD=VDS(IO)+C7*N1-C8*N2+C9*N3-C10*N4
2956      CALL AIRVEL(XD,YD,UA,VA,4)
2957      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
2958      C
2959      K5=DTS(IO)*UD
2960      L5=DTS(IO)*VD
2961      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I+1),O)
2962      M5=DTS(IO)*AN(1,IP1)
2963      N5=DTS(IO)*AN(2,IP1)
2964      XD=XDS(IO)-C11*K1+C12*K2-C13*K3+C14*K4-C15*K5
2965      YD=YDS(IO)-C11*L1+C12*L2-C13*L3+C14*L4-C15*L5
2966      UD=UDS(IO)-C11*M1+C12*M2-C13*M3+C14*M4-C15*M5
2967      VD=VDS(IO)-C11*N1+C12*N2-C13*N3+C14*N4-C15*N5
2968      CALL AIRVEL(XD,YD,UA,VA,4)
2969      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
2970      C
2971      K6=DTS(IO)*UD
2972      L6=DTS(IO)*VD
2973      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I)+DTS(IO)/2.DO,O)
2974      M6=DTS(IO)*AN(1,IP1)
2975      N6=DTS(IO)*AN(2,IP1)
2976      C
2977      C NEW DROPLET POSITION AT I+1
2978      XDS(IP1)=XDS(IO)+C16*K1+C17*K3+C18*K4-C19*K5
2979      YDS(IP1)=YDS(IO)+C16*L1+C17*L3+C18*L4-C19*L5
2980      C NEW DROPLET VELOCITY AT I+1
2981      UDS(IP1)=UDS(IO)+C16*M1+C17*M3+C18*M4-C19*M5
2982      VDS(IP1)=VDS(IO)+C16*N1+C17*N3+C18*N4-C19*N5
2983      C
2984      C 5TH ORDER ESTIMATE OF POSITION AND VELOCITY
2985      XT=XDS(IO)+C20*K1+C21*K3+C22*K4-C23*K5+C24*K6
2986      YT=YDS(IO)+C20*L1+C21*L3+C22*L4-C23*L5+C24*L6
2987      UT=UDS(IO)+C20*M1+C21*M3+C22*M4-C23*M5+C24*M6
2988      VT=VDS(IO)+C20*N1+C21*N3+C22*N4-C23*N5+C24*N6
2989      C
2990      C DETERMINE DIFFERENCES IN 4TH AND 5TH ORDER ESTIMATES.
2991      XR=(XT-XDS(IP1))/DTS(IO)
2992      IF(DABS(XR).LT.1.D-70)XR=1.D-70
2993      YR=(YT-YDS(IP1))/DTS(IO)
2994      IF(DABS(YR).LT.1.D-70)YR=1.D-70
2995      UR=(UT-UDS(IP1))/DTS(IO)
2996      IF(DABS(UR).LT.1.D-70)UR=1.D-70
2997      VR=(VT-VDS(IP1))/DTS(IO)
2998      IF(DABS(VR).LT.1.D-70)VR=1.D-70
2999      C CALCULATE STEP SIZE ADJUSTING FACTORS.
3000      XDEL=(EPS/DABS(XR))**.25DO
3001      YDEL=(EPS/DABS(YR))**.25DO
3002      UDEL=(EPS/DABS(UR))**.25DO
3003      VDEL=(EPS/DABS(VR))**.25DO
3004      C ADJUST FOR LEAST PRECISE EQN.
3005      DMIN=DMIN1(XDEL,YDEL,UDEL,VDEL)
3006      RMAX=DMAX1(DABS(XR),DABS(YR),DABS(UR),DABS(VR))
3007      K=IO
3008      IF(RMAX.LE.EPS)K=IP1
3009      IF(DMINP.LT.1.DO)GOTO 200
3010      IF(DMIN.LT.1.DO)DTS(K)=0.9DO*DMIN*DTS(IO)
3011      IF(DMIN.GE.11.DO)DTS(K)=1.8DO*DTS(IO)
3012      IF(DMIN.GE.1.DO.AND.DMIN.LT.11.DO)DTS(K)=((DMIN-1.DO)/10.DO+1.DO)
3013      *DTS(IO)*0.9DO
3014      GOTO 210
3015      200 IF(DMIN.LE.0.5DO)DTS(K)=0.5DO*DTS(IO)
3016      IF(DMIN.GT.1.DO)DTS(K)=((DMIN-1.DO)/10.DO+1.DO)*0.9DO*DTS(IO)
3017      IF(DMIN.GT.0.5DO.AND.DMIN.LE.11.DO)DTS(K)=DMIN*0.9DO*DTS(IO)

```

```

3018 210 DMINP=DMIN
3019 IF(RMAX.GT.EPS)GOTO 100
3020 C
3021 C NEW ACCELERATIONS AT I+1
3022 CALL AIRVEL(XDS(IP1),YDS(IP1),UAS(IP1),VAS(IP1),5)
3023 CALL DRAG(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),CDS,RED(IP1),CD)
3024 CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),
3025 CD,EQN,TS(I+1),0)
3026 IF(EQN.NE.2)RETURN
3027 CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),
3028 CD,EQN,TS(I+1),1)
3029 RETURN
3030 END
3031 C
3032 C
3033 SUBROUTINE RK4(EQN,CDS)
3034 C
3035 C WRITTEN BY: M. OLESKIW ON: 790926 LAST MODIFIED:801223
3036 C
3037 C INTEGRATE THE DROPLET EQNS OF MOTION (IN X AND Y) USING THE 4TH
3038 C ORDER RUNGE-KUTTA TECHNIQUE.
3039 C REF: BURDEN,R.L., J.D. FAIRES, & A.C. REYNOLDS (1978), NUMERICAL
3040 C ANALYSIS P. 281 QA 297.B84
3041 C
3042 DOUBLE PRECISION K1,L1,K2,L2,K3,L3,K4,L4,DTS(6),XDS(6),UDS(6),
3043 YDS(6),VDS(6),AN(2,6),HT(2,6),
3044 M1,M2,M3,M4,N1,N2,N3,N4,U1,U2,U3,V1,V2,V3,CD,RE,RED(6),
3045 VAS(6),UAS(6),TS(500)
3046 C
3047 INTEGER I,EQN,IM4,IM3,IM2,IM1,IO,IP1,CDS
3048 C
3049 COMMON /INTEG/AN,HT/PV/XDS,YDS,UDS,VDS
3050 /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1
3051 /REL/UAS,VAS,RED,CD
3052 C
3053 C IN DTS=NON-DIMENSIONAL TIME STEP
3054 C IN I= PRESENT INDEX OF VECTORS XDS,UDS,...
3055 C IN EQN= CHOICE OF TERMS USED IN RHS OF ODE
3056 C
3057 TS(I+1)=TS(I)+DTS(IO)
3058 K1=DTS(IO)*UDS(IO)
3059 L1=DTS(IO)*VDS(IO)
3060 M1=DTS(IO)*AN(1,IO)
3061 N1=DTS(IO)*AN(2,IO)
3062 CALL AIRVEL(XDS(IO)+K1/2.DO,YDS(IO)+L1/2.DO,U1,V1,4)
3063 CALL DRAG(UDS(IO)+M1/2.DO,VDS(IO)+N1/2.DO,U1,V1,CDS,RE,CD)
3064 C
3065 K2=DTS(IO)*(UDS(IO)+M1/2.DO)
3066 L2=DTS(IO)*(VDS(IO)+N1/2.DO)
3067 CALL ACCN(UDS(IO)+M1/2.DO,VDS(IO)+N1/2.DO,U1,V1,RE,CD,EQN,
3068 TS(I),0)
3069 M2=DTS(IO)*AN(1,IP1)
3070 N2=DTS(IO)*AN(2,IP1)
3071 CALL AIRVEL(XDS(IO)+K2/2.DO,YDS(IO)+L2/2.DO,U2,V2,4)
3072 CALL DRAG(UDS(IO)+M2/2.DO,VDS(IO)+N2/2.DO,U2,V2,CDS,RE,CD)
3073 C
3074 K3=DTS(IO)*(UDS(IO)+M2/2.DO)
3075 L3=DTS(IO)*(VDS(IO)+N2/2.DO)
3076 CALL ACCN(UDS(IO)+M2/2.DO,VDS(IO)+N2/2.DO,U2,V2,RE,CD,EQN,
3077 TS(I)+DTS(IO)/2.DO,0)
3078 M3=DTS(IO)*AN(1,IP1)
3079 N3=DTS(IO)*AN(2,IP1)
3080 CALL AIRVEL(XDS(IO)+K3,YDS(IO)+L3,U3,V3,4)
3081 CALL DRAG(UDS(IO)+M3,VDS(IO)+N3,U3,V3,CDS,RE,CD)
3082 C
3083 K4=DTS(IO)*(UDS(IO)+M3)
3084 L4=DTS(IO)*(VDS(IO)+N3)
3085 CALL ACCN(UDS(IO)+M3,VDS(IO)+N3,U3,V3,RE,CD,EQN,
3086 TS(I)+DTS(IO)/2.DO,0)
3087 M4=DTS(IO)*AN(1,IP1)
3088 N4=DTS(IO)*AN(2,IP1)
3089 C
3090 C NEW DROPLET POSITION AT I+1
3091 XDS(IP1)=XDS(IO)+(K1+2.DO*K2+2.DO*K3+K4)/6.DO

```

```

3092      YDS(IP1)=YDS(IO)+(L1+2.DO*L2+2.DO*L3+L4)/6.DO
3093      C NEW VELOCITIES AT I+1
3094      UDS(IP1)=UDS(IO)+(M1+2.DO*M2+2.DO*M3+M4)/6.DO
3095      VDS(IP1)=VDS(IO)+(N1+2.DO*N2+2.DO*N3+N4)/6.DO
3096      C NEW ACCELERATIONS AT I+1
3097      CALL AIRVEL(XDS(IP1),YDS(IP1),UAS(IP1),VAS(IP1),5)
3098      CALL DRAG(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),CDS,RED(IP1),CD)
3099      CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),CD,EQN,
3100      TS(I+1),O)
3101      DTS(IP1)=DTS(IO)
3102      IF(EQN.NE.2)RETURN
3103      C
3104      CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),CD,EQN,
3105      TS(I+1),1)
3106      RETURN
3107      END
3108      C
3109      C
3110      SUBROUTINE PC4(EQN,CDS)
3111      C
3112      C WRITTEN BY: M. OLESKIW ON: 800122 LAST MODIFIED: 801223
3113      C
3114      C INTEGRATE EQNS. OF MOTION USING THE 4TH ORDER PREDICTOR-
3115      C CORRECTOR METHOD
3116      C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978),
3117      C NUMERICAL ANALYSIS QA 297.B84 P.266
3118      C HAMMING, R.W. (1973), NUMERICAL METHODS FOR SCIENTISTS &
3119      C ENGINEERS, 2ND ED., QA 297.H28 CHAPS. 22 & 23
3120      C
3121      DOUBLE PRECISION XDS(6),UDS(6),AN(2,6),HT(2,6),YDS(6),
3122      VDS(6),AO,A1,A2,BO,B1,B2,B3,
3123      CO,C1,C2,DM1,DO,D1,D2,UPI,UCI,VPI,VCI,MUAS,MVAS,
3124      PUDS,DTS(6),PVDS,MUDS,MVDS,CUDS,CVDS,UDSP1,VDSP1
3125      FMU,FMV,UST,VST,ER1,ER2,PXDS,PYDS,MXDS,MYDS,CXDS,CYDS
3126      UAS(6),VAS(6),RED(6),XPI,XCI,YPI,YCI,RE,CD,TS(500)
3127      C
3128      INTEGER I,EQN,IM4,IM3,IM2,IM1,IO,IP1,CDS
3129      C
3130      COMMON/INTEG/AN,HT/PV/XDS,YDS,UDS,VDS
3131      /PCM/AO,A1,A2,BO,B1,B2,B3,CO,C1,C2,DM1,DO,D1,D2,
3132      UPI,UCI,VPI,VCI,ER1,ER2,XPI,XCI,YPI,YCI,UST,VST
3133      /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1
3134      /REL/UAS,VAS,RED,CD
3135      C
3136      C IN EQN= CHOICE OF TERMS USED IN RHS OF ODE
3137      C IN CDS=TYPE OF DRAG COEFFICIENT TO BE USED.
3138      C
3139      TS(I+1)=TS(I)+DTS(IO)
3140      C
3141      C THE PREDICTOR
3142      PXDS=AO*XDS(IO)+A1*XDS(IM1)+A2*XDS(IM2)
3143      +DTS(IO)*(BO*UDS(IO)+B1*UDS(IM1)+B2*UDS(IM2)+B3*UDS(IM3))
3144      PYDS=AO*YDS(IO)+A1*YDS(IM1)+A2*YDS(IM2)
3145      +DTS(IO)*(BO*VDS(IO)+B1*VDS(IM1)+B2*VDS(IM2)+B3*VDS(IM3))
3146      PUDS=AO*UDS(IO)+A1*UDS(IM1)+A2*UDS(IM2)
3147      +DTS(IO)*(BO*AN(1,IO)+B1*AN(1,IM1)+B2*AN(1,IM2)+B3*AN(1,IM3))
3148      PVDS=AO*VDS(IO)+A1*VDS(IM1)+A2*VDS(IM2)
3149      +DTS(IO)*(BO*AN(2,IO)+B1*AN(2,IM1)+B2*AN(2,IM2)+B3*AN(2,IM3))
3150      C
3151      C MODIFICATION OF THE PREDICTOR
3152      MXDS=PXDS-ER1*(XPI-XCI)
3153      MYDS=PYDS-ER1*(YPI-YCI)
3154      MUDS=PUDS-ER1*(UPI-UCI)
3155      MVDS=PVDS-ER1*(VPI-VCI)
3156      CALL AIRVEL(MXDS,MYDS,MUAS,MVAS,4)
3157      CALL DRAG(MUDS,MVDS,MUAS,MVAS,CDS,RE,CD)
3158      CALL ACCN(MUDS,MVDS,MUAS,MVAS,RE,CD,EQN,TS(I+1),O)
3159      FMU=AN(1,IP1)
3160      FMV=AN(2,IP1)
3161      C
3162      C THE CORRECTOR
3163      CXDS=CO*XDS(IO)+C1*XDS(IM1)+C2*XDS(IM2)
3164      +DTS(IO)*(DM1*MUDS+DO*UDS(IO)+D1*UDS(IM1)+D2*UDS(IM2))
3165      CYDS=CO*YDS(IO)+C1*YDS(IM1)+C2*YDS(IM2)

```

```

3166      .+DTS(IO)*(DM1*MVDS+DO*VDS(IO)+D1*VDS(IM1)+D2*VDS(IM2))
3167      CUUS=CO*UDS(IO)+C1*UDS(IM1)+C2*UDS(IM2)
3168      .+DTS(IO)*(DM1*FMU+DO*AN(1,IO)+D1*AN(1,IM1)+D2*AN(1,IM2))
3169      CVDS=CO*VDS(IO)+C1*VDS(IM1)+C2*VDS(IM2)
3170      .+DTS(IO)*(DM1*FMV+DO*AN(2,IO)+D1*AN(2,IM1)+D2*AN(2,IM2))
3171      C
3172      C FINAL VALUES
3173      XDS(IP1)=CXDS+ER2*(PXDS-CXDS)
3174      YDS(IP1)=CYDS+ER2*(PYDS-CYDS)
3175      UDS(IP1)=CUUS+ER2*(PUUS-CUUS)
3176      VDS(IP1)=CVDS+ER2*(PVDS-CVDS)
3177      C
3178      C NEW VALUES FOR ACCELERATION AT I+1
3179      CALL AIRVEL(XDS(IP1),YDS(IP1),UAS(IP1),VAS(IP1),5)
3180      CALL DRAG(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),CDS,RED(IP1),CD)
3181      CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),CD,EQN,
3182      .TS(I+1),0)
3183      UDSP1=AN(1,IP1)
3184      VDSP1=AN(2,IP1)
3185      IF(EQN.NE.2)GOTO 100
3186      CALL ACCN(UDS(IP1),VDS(IP1),UAS(IP1),VAS(IP1),RED(IP1),CD,EQN,
3187      .TS(I+1),1)
3188      C
3189      C CALCULATE STABILITY INDICES
3190      100  UST=(FMU-UDSP1)/(MUUS-UDS(IP1))
3191          VST=(FMV-VDSP1)/(MVDS-VDS(IP1))
3192          XPI=PXDS
3193          XCI=CXDS
3194          YPI=PYDS
3195          YCI=CYDS
3196          UPI=PUUS
3197          UCI=CUUS
3198          VPI=PVDS
3199          VCI=CVDS
3200      DTS(IP1)=DTS(IO)
3201      RETURN
3202      END
END OF FILE

```

A facsimile catalog card in Library of Congress MARC format is reproduced below.

Lozowski, Edward F.

Computer modeling of time-dependent rime icing in the atmosphere / by Edward F. Lozowski and Myron M. Oleskiw. Hanover, N.H.: U.S. Cold Regions Research and Engineering Laboratory; Springfield, Va.: available from National Technical Information Service, 1983.

vi, 31 p., illus.; 28 cm. (CRREL Report 83-2.)

Bibliography: p. 20.

1. AirSails. 2. Computerized simulation. 3. Ice formation. 4. Rime ice. I. Oleskiw, Myron M.

II. United States. Army. Corps of Engineers.

III. Cold Regions Research and Engineering Laboratory, Hanover, New Hampshire. IV. Series: CRREL Report 83-2.

4-
DT