0-315-08903-2

National Library of Canada — Bibliothèque nationale du Canada

14

Canadian Theses Division — Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

56820

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

MYRON MORRIS OLESKIW

Date of Birth — Date de naissance	Country of Birth — Lieu de naissance
MAY 26, 1951	CANADA

Permanent Address — Résidence fixe

#3 3000 RICHTER STREET
KELOWNA, B.C.
V1Y 2M5

Title of Thesis — Titre de la thèse

A COMPUTER SIMULATION OF TIME-DEPENDENT RIME ICING ON AIRFOILS

University — Université

U. OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

Ph.D.

Year this degree conferred — Année d'obtention de ce grade	Name of Supervisor — Nom du directeur de thèse
1982	DR. E. P. LOZOWSKI

Date	Signature
January 28, 1982	Myron Oleskiw

NL-91 (4-77)

THE UNIVERSITY OF ALBERTA

A COMPUTER SIMULATION OF TIME-DEPENDENT RIME ICING ON AIRFOILS

by

MYRON MORRIS OLESKIW

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

IN

METEOROLOGY

DEPARTMENT OF GEOGRAPHY

EDMONTON, ALBERTA

SPRING, 1982

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR MYRON MORRIS OLESKIW

TITLE OF THESIS A COMPUTER SIMULATION OF TIME-DEPENDENT RIME ICING ON AIRFOILS

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED SPRING, 1982

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Myron Oleskiw

PERMANENT ADDRESS:

#3, 3000 Richter Street

Kelowna, British Columbia

V1Y 8M5

DATED December 17, 1981

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A COMPUTER SIMULATION OF TIME-DEPENDENT RIME ICING ON AIRFOILS submitted by MYRON MORRIS OLESKIW in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY in METEOROLOGY.

Supervisor

External Examiner

Date: December 17, 1981

DEDICATION

To Mom and Dad -

Their many years of love, support and encouragement

have created the foundation

upon which this work is based

and

To Gisèle -

Her unquestioning confidence and enthusiasm

have provided the inspiration to overcome

many of the difficulties along the road to completion

ABSTRACT

While atmospheric icing no longer poses a serious threat to the operation of large transport aircraft equipped with anti- or de-icing devices, the operating environment and structural characteristics of general aviation aircraft and helicopters have generally prevented manufacturers from installing equally effective devices in these craft. Icing is particularly troublesome for helicopters. Accretions upon the main rotor blade may increase the torque requirements to beyond the engine's capabilities, resulting in a forced landing. Asymmetrical shedding from the main and tail rotors may cause severe vibration and structural damage.

This dissertation describes a numerical model which has been developed to predict the characteristics of rime ice accretion on an airfoil in a steady, inviscid, irrotational, incompressible, two-dimensional flow. The airflow about an arbitrarily shaped airfoil is calculated by a surface vorticity substitution technique. The full set of equations describing the accelerated motion of supercooled cloud droplets are integrated with a variable time step to yield the trajectories. An automated routine determines the placement of trajectory starting points for the efficient calculation of the local collision efficiency curve. Several such curves may be combined to approximate the effects of a natural droplet size distribution. The thickness of the accretion (calculated under the assumption that all droplets freeze immediately upon impact) leads to a determination of the resulting profile after a limited accretion period. This new profile is used to recompute the airflow about the airfoil, the droplet trajectories, and the other steps above, to give a simulation of time-dependent accretion. Efforts are made to optimize the code's efficiency while maintaining a high level of precision. The simulations are compared with previous analytical and experimental results. The agreement is generally quite good, although a lack of precise experimental simulations prevents a complete verification of the model. Two of the model's applications are presented: to study the change in ice accretion as a result of a change in the airfoil profile; and to test an airfoil scaling theory for its accuracy. The thesis concludes by recommending a series of enhancements to the model, and points out the need for improving the experimental simulations which could be used to verify the model.

ACKNOWLEDGMENTS

I wish to express my appreciation for the support and cooperation which have been provided to me by individuals and institutions during the course of this work.

Heartfelt thanks go to my thesis supervisor, Dr. E.P. Lozowski, whose generous aid and enthusiastic and patient guidance have contributed so much to the completion of this work. I wish also to express my appreciation to Mr. J.R. Stallabrass and Drs. E.M. Gates, R.B. Charlton and E.R. Reinelt, for serving patiently on my examining committee.

Dr. D.J. Marsden of the Mechanical Engineering Department, and Mr. M. Bragg of Ohio State University have kindly provided me with research results which have been used in this dissertation. Discussions with my colleagues in the Meteorology Division have helped me solve a number of problems. In this regard, the assistance of Mr. D.S. Phillips has been particularly valuable.

The production of this dissertation has been advanced a great deal by the very capable and generous assistance of Mrs. Laura Smith. The entry of the text into the computer for formatting was performed by Miss Susan Preston. To both I wish to express my gratitude.

I would also like to thank the Atmospheric Environment Service, the Natural Sciences and Engineering Research Council (NSERC), and the U.S. Army Cold Regions Research and Engineering Laboratory for providing research funds for this project. I am deeply grateful to NSERC as well for making a fellowship available to me during part of this study.

Table of Contents

List of Tables

Table		Page

List of Figures

Figure | Page

Figure | Page

Figure | Page

Figure		Page

Figure | Page

List of Symbols

CEE	control element endpoint	199
D	a point	39
D_d	droplet diameter.	76
D_j	half length of control element j.	25
D_{mm}	mass median droplet diameter.	50
DD	droplet diameter.	199
DDISTN	number of droplet size categories in distribution.	199
DENSE	accretion density formulation indicator.	199
e	Joukowski airfoil eccentricity.	15
e_i	local truncation error.	186
E_m	total collision efficiency.	44
$\bar{E}_m$	average (or filtered) total collision efficiency.	81
E(k)	complete elliptic integral of the second kind.	197
$E(\zeta,k)$	incomplete elliptic integral of the second kind.	197
EPS	local truncation error tolerance.	199
EQN	equation formulation indicator for droplet trajectories.	199
f	generalized function.	29
F	maximum Boxcar filter length.	52
$F(\zeta,k)$	incomplete elliptic integral of the first kind.	197
$F_v(\ell)$	variable length Boxcar filter.	51
FILTER	length of Boxcar filter.	199
$\bar{g}$	ND gravitational acceleration.	29
g'	variable in Joukowski airfoil generation.	23
$\bar{G}$	gravitational acceleration vector.	27
h	ND airfoil thickness.	16
h'	variable in Joukowski airfoil generation.	23
h_o	maximum ND airfoil thickness.	16
H	airfoil thickness.	16
H_o	maximum airfoil thickness.	60
i	general index.	24
I_k	cross-sectional area under airfoil profile interpolator for layer k.	60

ICE	ND accretion parameter.	199
IFR	instrument flight rules.	1
j	general index.	25
j'	variable in Joukowski airfoil generation.	23
J	Jacobian.	33
k	general index or constant.	33
$\hat{k}$	a constant.	187
k_k	strength of line vortex.	22
k_m	source strength.	22
K	ND inertia parameter.	75
K_{ij}	influence coefficient of control element i on control point j.	25
$K(k)$	complete elliptic integral of the first kind.	197
ℓ	ND length along airfoil surface from the nose.	39
ℓ^+	value of ℓ for upper trajectory of pair.	47
ℓ^-	value of ℓ for lower trajectory of pair.	47
ℓ_G	length along airfoil surface to grazing trajectory impact point.	40
ℓ_{GL}	value of ℓ_G on lower airfoil surface.	45
ℓ_{GU}	value of ℓ_G on upper airfoil surface.	45
ℓ_o	value of ℓ at peak of β curve.	52
ℓ_R	range in ℓ.	46
L	ND length along airfoil surface.	47
LAYER	maximum number of accretion layers.	199
LWC	liquid water content.	3
m	ND accretion thickness on a curved surface.	54
m_{VD}	thickness of accretion with variable density.	56
M	ND length along airfoil surface.	55
MMD	mass median diameter.	3
n	general index.	24
ND	non-dimensional.	15
NEB	number of CEE's in back two-thirds of each airfoil surface.	199

NEF	number of CEE's in front third of each airfoil surface.	199
NIF	number of SSE's between CEE's on front third of each surface.	199
O	a point.	20
p	ND accretion thickness.	54
P	a point.	20
P_F	air pressure for full-scale model.	98
P_M	air pressure for scale model.	98
P_q	scaling pressure ratio.	98
q	a constant.	191
$\hat{q}$	model scaling ratio (ratio of chord lengths).	98
Q	a point.	55
Q_D	ratio of lengths along new and old airfoil surfaces.	61
Q_{DM}	maximum allowed value of Q_D.	61
Q_S	ratio of CEE to SSE indices.	62
r_c	ND radius of curvature.	54
r_d	ND droplet radius.	20
r_{1j}	a distance as defined in Fig. 6.	25
r_{2j}	a distance as defined in Fig. 6.	25
R_d	droplet radius.	27
R_{dF}	full-scale droplet radius.	99
R_{dM}	scale model droplet radius.	99
R_q	ratio of droplet radii.	99
Re_c	wing Reynolds number.	98
Re_d	droplet Reynolds number.	27
Re_∞	free-stream Reynolds number.	74
RHS	right hand side.	29
s	a constant.	191
s_{1j}	a distance as defined in Fig. 6.	25
s_{2j}	a distance as defined in Fig. 6.	25
S_I	slope of airfoil surface at SSE i in rotated coordinates.	57
S_L	approximate slope of airfoil surface in Fig. 9.	39

S_β	average slope of β curve segment	49
SSE	surface segment endpoint	26
t	ND time	29
t_i	ND time prior to time step i	37
T	time since beginning of droplet acceleration	27
T_A	accretion period	54
Δt	time step or interval	33
Δt_i	ith time step	186
$\underline{u}$	truncation error component	186
u^*	droplet impact velocity component	40
u_∞	ND free-stream air velocity component	21
u_a	ND air velocity component	20
u_d	ND droplet velocity component	29
$\hat{u}_d$	fourth-order estimate of the droplet velocity	186
$\hat{\hat{u}}_d$	fifth-order estimate of the droplet velocity	29
U_∞	free-stream air velocity	98
$U_{\infty F}$	full-scale free-stream air velocity	98
$U_{\infty M}$	scale model free-stream air velocity	98
$U_{\infty q}$	ratio of free-stream air velocities	98
$\underline{v}$	truncation error component	186
v^*	droplet impact velocity component	40
v_∞	ND free-stream air velocity component	21
v_a	ND air velocity component	20
v_d	ND droplet velocity component	29
$\hat{v}_d$	fourth-order estimate of the droplet velocity	186
$\hat{\hat{v}}_d$	fifth-order estimate of the droplet velocity	186
$\overline{V}_a$	air velocity vector	27
$\overline{V}_d$	droplet velocity vector	27
$\overline{V}_i^*$	droplet impact velocity vector	55
w_i	fraction of total liquid water content in category i	50
W	cloud liquid water content	54
x	general coordinate	16

$\underline{x}$	truncation error component	186
$\dot{\bar{x}}$	generalized velocity vector	30
$\ddot{x}$	generalized acceleration	30
x'	coordinate for Joukowski airfoil derivation	22
x''	coordinate for Joukowski airfoil derivation	22
x^*	droplet-airfoil impact coordinate	40
$\hat{x}$	fourth-order estimate of x	186
x_d	ND droplet position coordinate	30
x_D	coordinate of droplet surface closest to airfoil	40
x_I	approximation to desired airfoil abecissa	198
x_L	airfoil coordinate on lower surface	15
x_N	airfoil nose coordinate	39
x_{NR}	new nose coordinate in old rotated coordinate system	60
x_R	ND rotated airfoil coordinate	56
x_{TR}	airfoil tail coordinate in old rotated coordinate system	61
x_U	airfoil coordinate on upper surface	15
xo	starting point for droplet trajectory	36
X	generalized spline segment abscissa	195
$\bar{X}_d$	droplet position vector	28
y	general coordinate	16
$\underline{Y}$	truncation error component	186
y'	coordinate for Joukowski airfoil derivation	21
y''	coordinate for Joukowski airfoil derivation	22
y^*	droplet-airfoil impact coordinate	40
$\hat{y}$	fourth-order estimate of y	186
y_A	coordinate of airfoil closest to droplet	40
y_c	coordinate of NACA mean line	18
y_{CLAP}	closest vertical approach between airfoil and droplet surfaces	40
y_d	ND droplet position coordinate	31
y_D	coordinate of droplet surface closest to airfoil	40
y_G	ordinate of interpolator to estimate true airfoil surface	198

y_h	airfoil ordinate value.	18
y_L	airfoil coordinate on lower surface.	15
y_N	airfoil nose coordinate.	39
y_{NR}	new nose coordinate in old rotated coordinate system.	59
yo^+	starting point of upper droplet trajectory.	46
yo^-	starting point of lower droplet trajectory.	46
y_I	approximation to airfoil ordinate.	198
y_R	ND rotated airfoil coordinate.	54
y_{s1}	airfoil ordinate point in Fig. 9.	39
y_{s2}	airfoil ordinate point in Fig. 9.	39
y_U	airfoil coordinate on upper surface.	15
yo	starting point for droplet trajectory integration.	37
yo_G	value of yo for grazing trajectory.	42
yo_{GL}	value of yo_G for lower surface.	44
yo_{GU}	value of yo_G for upper surface.	44
Y'	generalized spline segment ordinate.	195
z	coordinate in complex plane.	16
Z	ND cross-sectional area of accreted ice.	87
α	angle of attack.	13
β	local collision efficiency.	43
$\bar{\beta}$	combined local collision efficiency from droplet distribution.	51
β_o	maximum value of local collision efficiency.	47
$\bar{\beta}_o$	combined maximum local collision efficiency from droplet distribution.	81
$\beta_F(\ell)$	filtered (averaged) local collision efficiency.	51
β_N	normalized local collision efficiency.	47
β_R	range of β values for an airfoil under given conditions.	47
γ	slope of NACA mean line.	18
γ	vorticity density.	24
γ_j	vorticity density along CEE j.	25
δ	spline segment abscissa.	46

$\hat{\delta}_i$	spline segment abscissa	60
δ_N	normalized spline segment abscissa	46
δ_R	rotated ND spline segment abscissa	55
Δt	time step or interval	33
Δt_i	ith time step.	186
$\Delta\theta$	angle interval.	25
ϵ	tolerance for local truncation error - RKF4 method	42
ζ	complex coordinate in transformed plane.	16
ζ'	complex coordinate in transformed plane.	23
η	imaginary part of ζ .	16
η'	imaginary part of ζ'.	23
θ	polar angle from x-axis.	14
θ^*	angle between droplet trajectory and airfoil surface normal.	41
θ_j	angle of CEE's in non-transformed plane.	25
θ_s	angle of normal to airfoil surface.	41
θ_T	angle of droplet trajectory at impact	40
θ	temperature.	56
θ_F	full-scale temperature.	99
θ_M	scale model temperature.	99
θ_q	ratio of air temperatures.	99
θ_s	accretion surface temperature.	57
θ_∞	free-stream air temperature.	58
$\Delta\theta$	angle interval.	25
κ	impact angle tolerance.	43
λ_k	eigenvalues of Jacobian.	33
μ	dynamic air viscosity.	75
ν	kinematic air viscosity.	27
ν_a	ND kinematic air viscosity.	29
ξ	real part of ζ .	16
ξ'	real part of ζ'.	23
ρ_a	air density.	28

ρ_d	water density.	28
ρ_i	ice density.	54
σ	ND accretion cross-sectional area.	55
τ	time.	27
ϕ	ND impingement parameter.	75
Φ	ND ice density.	56
χ_f	time-step size change factor.	187
ψ	ND streamfunction.	20
ψ_o	integration constant.	20
ω	ND accretion parameter.	53
Ω	accretion thickness (accretion parameter).	53

1. INTRODUCTION

1.1 Airfoil icing: the problem.

As air transport developed during the 1930's and 1940's, an increasingly greater emphasis was placed upon the need for all-weather operations. This had become possible with the advent of sufficiently advanced avionics so that pilots could fly in cloud and precipitation via IFR (instrument flight rules), that is, without the need for visual contact with the ground or horizon. It was soon discovered however, that flight through clouds which were composed of supercooled water droplets could lead to carburetor icing, and ice accretion on the propellors, struts, antennas, leading edges of the wings and tail, and even on the aircraft fuselage itself. At times such icing could cause a severe loss in performance, resulting in a forced landing or even a stall in mid-flight.

In an effort to find a solution to the icing problem, research began in earnest in several countries, virtually simultaneously during World War II. The U.S. National Advisory Committee on Aeronautics (NACA) undertook a number of theoretical and experimental studies into icing, some of which will be outlined below. This work led to an increased understanding of the icing problem, and it allowed engineers to design anti- or de-icing equipment for the larger transport aircraft where sufficient weight and power reserves permitted it. One popular solution was to use the hot engine bleed air to heat areas prone to icing. Another solution involved the use of pneumatic boots on the leading edges of the wings to break the ice away periodically.

With the advent of jet aircraft, the icing problem became less severe since these planes could rapidly climb through icing regions to the 30,000 or 40,000 foot levels where the problem essentially did not exist (Beheim, 1978a). The relatively short periods that such aircraft did spend in descent were not a significant problem either because the lighter fuel load gave the plane an even greater power reserve. As a result, the research effort in icing abated somewhat after the 1950's.

The early emphasis in icing research was directed toward large commercial and military transport aircraft. However, according to Beheim (1978a) "the icing protection requirements for . . small aircraft are so uniquely different from those for large transports that an extrapolation of the current base of icing technology is clearly inadequate. The components of these aircraft are smaller so proportionately heavier accretions of ice are more likely to occur. Consequently, their aerodynamic performance will deteriorate more drastically." The large power reserves and sufficient quantities of high pressure heated air which exist on the larger airplanes are not available [illegible]ht general aviation aircraft. A second class of aircraft, the rotorcraft, [illegible] also plagued by icing problems. For helicopters, ice accretions can be particularly dangerous. Ice forming on the main and tail rotors causes an increase in airfoil drag, thereby requiring an increase in engine power to maintain altitude (Lake & Bradley, 1976). If sufficient ice forms, it can lead to unexpected stall on the trailing rotor blade (Stallabrass, 1958a). Further, the centrifugal forces acting on the ice, combined with the rapidly varying blade pitch and blade flexing in forward flight, may cause portions of the accreted ice to be shed. If this shedding occurs asymmetrically, severe vibrations and structural damage to the helicopter can result (Lozowski *et al.*, 1979). Ice chunks leaving the tail rotor may hit the main fuselage causing damage there. Also, ingestion of ice chunks into the turbines may produce damage to the compressor blades causing a loss of power. Icing of the windshield can result in a loss of visibility, and if ice should form on critical control linkages in the rotor hub, violent loss of control may result (Stallabrass, 1958a). To date, only the French PUMA helicopter has been certified for unrestricted flight in icing conditions (Lecoutre, 1978).

1.2 The icing environment.

The operational environment of helicopters and light aircraft is such that icing conditions are much more likely to be encountered by these craft than by jet aircraft. Helicopters in particular are routinely required to supply oil rigs, to fly search and rescue missions, and to perform anti-submarine duties all over the ocean where the temperature and liquid water content of any clouds that are present could lead to hazardous icing in winter conditions (Ryder, 1978).

In the United States, the Federal Aviation Administration (FAA) has set down guidelines (FAR-25) regarding the conditions which aircraft must meet if they are to be certified for IFR operations through supercooled clouds. The aircraft must continue to operate safely through stratiform (continuous icing) and cumulus (intermittent icing) clouds where the combination of liquid water content (LWC), air temperature, and the droplet distribution representative mass median diameter (MMD) are defined by the solid lines of Fig. 1. The data which were used in drafting these regulations were obtained from measurements made by transport aircraft in the late 1940's and early 1950's (Lewis, 1947; Lewis & Bergrun, 1952). These regulations as they apply to helicopters have come under increasing attack in recent years (Werner, 1975; Rosen & Potash, 1981; Frost *et al.*, 1978) because they may be too stringent. They appear to be based on exceedance levels of 0.1%. Also these regulations may not be appropriate for the lower altitudes at which helicopters fly. Re-analyzing Lewis & Bergrun's data, Werner (1975) has concluded that the 1% exceedance probability curves for severe icing for three areas in the United States are as shown by the dotted and dashed lines in Fig. 1. Based upon their results, he has recommended a new set of atmospheric icing criteria for helicopters as set out in Figs. 2 and 3. The FAA has requested that the National Aeronautics and Space Administration (NASA) conduct research to update the data upon which FAR-25 is based. This work is proceeding (Jeck, 1981).

Frost *et al.* also express their frustration with the strict icing criteria applied to certifying helicopters for IFR operations. The FAA continues to require natural ice testing, a costly, time-consuming and uncertain means of achieving the desired goals. They claim that the upper limits of the meteorological design cirteria as defined in Fig. 1 are rarely encountered in natural testing. Helicopters are more limited in range than jet transports, and thus they are not able to seek out areas where icing conditions may be appropriate for testing unless such areas are near to their base. When conditions are not suitable, many man-hours can be wasted at great expense to the helicopter manufacturer. With these problems in mind, other routes have been taken to aid in finding a solution to helicopter and light aircraft icing.

1.3 Experimental icing investigations.

1.3.1 Natural icing tests.

Reports of natural helicopter icing tests are very rare. Where such tests have been carried out by helicopter manufacturers, the results have generally remained proprietary. Rosen & Potash (1981) describe one of the earliest experiments – that of placing a Sikorsky R-4 helicopter at the summit of Mt. Washington, New Hampshire in 1945. The results from these tests proved inconclusive because of a lack of appropriate conditions.

Stallabrass (1958a) detailed the results of a Sikorsky S-55 helicopter flight in a natural supercooled fog. This experiment was terminated when, after 40 minutes of flight, the increase of engine power required for hover was very slow. Although the LWC and MMD of the fog were not measured, they were estimated to be in the region which Werner (1975) would define as "Trace".

1.3.2 Testing in an artificial indoor environment.

Two laboratory facilities have been constructed for the investigation of icing on rotating helicopter rotor blades. One (described by Stallabrass, 1957) was built to test the effectiveness of de-icing via electro-thermal pads mounted on the leading edge of a shortened whirling rotor placed in a coldroom. The other, designed to test full-scale helicopters with the blades in motion, but with the helicopter remaining on the ground, was built within a refrigerated hanger at Eglin Air Force Base, Florida (Rosen & Potash, 1981). This icing spray rig was installed over the helicopter. "The testing was limited [during the 1949 – 1952 period] to a temperature range of 23°F to 28°F and was conducted with excessive LWC because of spray rig limitations" (Rosen & Potash, 1981).

In addition to these facilities, Ackley *et al.* (1979) built a small whirling cylinder device in order to study the thickness and nature of the resulting accretions when the device was operated in a supercooled cloud formed in a cold room. The results were compared with theoretical calculations to be described below.

1.3.3 Icing on airfoils in wind tunnels.

Among the earliest controlled experimental simulations of the icing process were those carried out in the NACA Lewis icing tunnel by Gelder *et al.* (1956). They tested the Joukowski 0015 and various NACA 6-series airfoils at angles of attack ranging from 0° to 12° with a dye tracer technique to find the local and total droplet impingement rates for a variety of droplet MMD's and airfoil chord lengths. They found the experimental impingement results to be within ±10% of the average of the results calculated from theoretical trajectories. Other tests were carried out (see for example Gray, 1957) upon other airfoils to determine the shape and aerodynamic effects of ice accretion.

More recently, icing simulations have been carried out upon a cylinder and a helicopter tail rotor section within the icing tunnel at the National Research Council of Canada Low Temperature Laboratories in Ottawa (Stallabrass & Lozowski, 1978). The cylinder was used to allow comparisons with a theoretical model (to be described below). Accretion on the airfoil was carried out at angles of attack between 0° and 12° at various speeds and air temperatures. Several cases will be described in detail in Chapters 4 and 5. The airfoil accretions resembled those achieved in spray-rig experiments (described in the next section). A novelty of these experiments was the introduction of mixed cloud conditions (ice crystals and supercooled water droplets). The results indicated that such clouds posed less of an icing threat than those composed only of liquid water.

The continuing need for a solution to light aircraft icing has led Bragg *et al.* (1981) to test a Hicks modified NACA 64-215 airfoil in the NACA Lewis icing tunnel. The results have been compared to a theoretical model of airfoil icing that they have developed.

All of these experimental investigations have revealed the strong dependence of the ice accretion upon the environmental conditions (LWC, air temperature, ambient pressure, droplet size spectrum) and also upon the flow conditions (air velocity, airfoil chord length, and angle of attack).

1.3.4 The NRC Spray Rig.

Stallabrass (1957 and 1958b) has described the development of a spray rig by the National Research Council of Canada which is capable of providing an icing environment within which helicopters may simulate hovering conditions in a natural icing cloud. The rig produces a cloud of supercooled droplets with a MMD of about 30 µm. The maximum theoretical LWC is about 2 g m^{-3}. The value of this device is evidenced by the extended period of use it has enjoyed. A number of trials of various helicopters have been performed: a Bell HTL-4 (Stallabrass, 1957); a Sikorsky S-55 (Stallabrass 1958a) and a Bell UH-1H (Cotton, 1976) to name a few. The purposes of the tests have ranged from gaining a fundamental understanding of the ice accretion process under realistic conditions, to checking out a de-icing system. One icing test simulation by Stallabrass (1958a) will be considered in more detail in Chapter 5.

1.3.5 The Helicopter Icing Spray System (HISS).

The development of a spray system attached to a CH-47C helicopter has been summarized by Belte (1981). The present version of this system can produce a cloud of water droplets with a LWC between 0.25 and 1.0 g m^{-3} and a MMD of about 25 to 35 µm. When a helicopter flies in the spray plume produced by the HISS at the appropriate air temperature, natural icing conditions may be simulated fairly well, although not all of the helicopter may be immersed in the plume at once. This allows testing of helicopters in forward flight, a feature unavailable in any other experimental simulation. Naturally the costs of this type of simulation are higher than for ground based simulators, although they are not as high as for natural icing testing because the icing clouds are produced artifically and only the appropriate temperatures need be ensured. Measurements of the droplet size spectrum produced by the HISS are shown in Fig. 4 as a set of points. The drop size spectrum of a natural cloud in Minnesota is displayed as a dashed line. For comparison, the Langmuir "D" distribution (Langmuir & Blodgett, 1946) used later in this dissertation is displayed as a solid line. This latter curve is calculated assuming a MMD of 20 µm and a LWC of 1 g m^{-3}.

1.4 Theoretical calculations of droplet impingement and ice accretion.

Calculations of the trajectories of water droplets in a flow about various airfoil shapes began in the 1940's, with the results from papers by Langmuir & Blodgett (1946), Guibert *et al.* (1949), and Brun *et al.* (1953) still widely used. These results were based upon the use of a differential analyzer, an analog device. A series of NACA Technical Notes followed outlining droplet impingement calculations for various airfoils under a variety of conditions. Some of these will be described in detail in Chapter 4.

Working in parallel were a number of investigators of the thermodynamics of the ice accretion process. Some papers were applied to the thermodynamics of the hail formation process (Ludlam, 1951 is one of the first of many in this field). Others were slanted more toward airfoil icing (Messinger, 1953)

The problem of icing on stationary structures (in its glaze, rime, and freezing rain forms) also began to receive attention (McKay & Thompson, 1969; Poots & Rodgers, 1976; List, 1977; Makkonen, 1981, and McComber & Touzot, 1981). These studies are important in airfoil icing as well because even though the icing conditions are somewhat different, many of the same techniques may be applied.

Work on the microstructure of accreted ice and its density has been carried on by Macklin (1962), Macklin & Payne (1968), and Buser & Aufdermaur (1973). These papers are significant to the present study because they can be used to provide a formulation for the density of accreted ice.

With the advent of large electronic computers, theoretical models of the ice accretion process have been given a big boost. The complex calculations of droplet trajectories and the subtleties of thermodynamic feedbacks may now begin to be investigated. Early endeavors in this field were those of Kloner (1970) and Werner (1973) at Lockheed California Company. Kloner developed a model of the ice accretion process on arbitrarily shaped airfoils where the accretion was treated as a steady-state process. Werner added to this model by incorporating a set of thermodynamic equations, and predicting the surface temperature of the deposit as well as the ice build-up rates and initial freezing rates on NACA airfoils suitable for helicopter main and tail rotors. His conclusion was that icing could pose at least as great a problem for the tail rotor as for the main rotor of a helicopter. No comparisons between his model and experimental results are made however.

Cansdale & McNaughtan (1977) have developed a theoretical scheme to be used for the prediction of the surface temperature and rate of ice accretion of an airfoil in a mixed water droplet/ice crystal cloud. They propose to subdivide the airfoil surface into a number of sectors and calculate the thermodynamic equations in each sector. This will allow them to model runback of water which has accreted but not frozen due to the surface of the deposit not being below 0°C. No results from this model have yet been published, although preliminary results are available (Cansdale, personal communication).

Lozowski *et al.* (1979) have developed the model proposed by Cansdale & McNaughtan for a non-rotating cylinder. Detailed calculations of the thermodynamics are made, and mixed icing conditions can be simulated. The model results are compared to experimental observations of icing upon a cylinder within the NRC icing tunnel. The agreement between model and experiment was good when the accretion was relatively dry, but it deteriorated when the conditions allowed significant runback.

The limitation of time-independent growth assumed in the two previous models was relaxed somewhat by Ackley & Templeton (1979). While their model incorporated the effects of a liquid water cloud only, and did not treat the detailed thermodynamics of the ice accretion, the time dependence of a rime accretion was simulated by accreting a series of thin layers. The actual shape of each layer was not simulated, but rather it was assumed that the cross-section of the accretion always remained elliptical. Their results were compared to the accretions observed on a whirling cylindrical bar (Ackley *et al.*, 1979). They attained reasonable agreement when rime icing was simulated.

Simultaneously with, but independently of the development of the model described in this thesis, Bragg *et al.* (1981) have developed a model to be used for the prediction of ice accretion shape and mass on arbitrarily shaped airfoils. They can simulate the time-dependence of the rime accretion process by discretizing the icing process into a set of layers, with the accretion process taking into account the change in shape of the airfoil profile as the accretion proceeds. Their preliminary comparisons with other theoretical and experimental results show reasonable agreement in most cases regarding the accreted ice profile, as well as agreement regarding the degradation in airfoil performance caused by the ice accretion.

1.5 Goals of the present study.

In this introduction we have outlined the continuing icing problems experienced by light aircraft and helicopters. The escalating costs of aircraft development imply that a renewed and coordinated icing research effort must be carried out (Beheim, 1978a). The Icing Research and Facilities Committee of NASA has recommended (Beheim, 1978b) that

> "The large aircraft companies have already developed sophisticated means of [icing] analysis, but their availability is not widespread, particularly for the general aviation industry. In view of recent progress achieved in computational fluid mechanics, even further improvements in analysis could be developed and the committee was enthusiastic that renewed efforts would have a good chance of success in providing more accurate predictive and design methods. Such an effort to improve existing methods and increase their availability was strongly endorsed."

This dissertation will describe the development of a numerical model which can predict the shape and mass of rime accretion on an arbitrarily shaped airfoil. The time dependence of the accretion process will be modelled by discretizing the accretion period, and allowing the ice to build up in a series of layers. The flowfield and droplet trajectories will be re-computed after each layer. The ice density will be specified according to the formula proposed by Macklin (1962). An attempt will be made to incorporate high accuracy in all calculations, and then to reduce the tolerances to determine if acceptable results can be achieved with a smaller computing effort. The model predictions will be compared to other theoretical and experimental ice accretion results to verify the soundness and reliability of the model. Finally, recommendations will be proposed for the improvement of the model, and also for the improvement of intercomparisons between the model and experimental observations.

2. METHODOLOGY

2.1 Introduction

The goal of this dissertation has been defined in Chapter 1, that is to develop a numerical model capable of predicting the shape and extent of rime ice accretion on a two dimensional airfoil of arbitrary shape in a 2-D steady, incompressible, irrotational, inviscid flow containing an ensemble of supercooled cloud droplets. The techniques which have been employed to develop this program are described in this chapter.

The modelling of the accretion process consists of three major steps, to be elaborated upon in the following three sections of this chapter. They are:

1. determination of the flowfield about an arbitrarily shaped two dimensional airfoil;
2. calculation of the trajectories of droplets embedded within the flow, and the rate at which they collide with the airfoil surface; and
3. computation of the thickness of the resulting ice accretion, together with the shape of the new airfoil surface following accretion.

Since we are dealing only with rime ice, no attempt is made to work out the thermodynamic processes which occur at the airfoil surface. This must be left for a subsequent study.

Before the flowfield may be calculated about an airfoil, the shape of the airfoil must be given. In general, the profile will not be specified by a set of analytic functions, but rather by a set of discrete coordinate values. Thus the program has been written to interpolate a smooth airfoil surface between the data points. When analytic forms exist for certain airfoils, these equations are used to genérate a set of data points, thereby maintaining a consistent approach.

The "airfoil" shapes that may be accommodated include:

1. the cylinder, which is included because of its use in many aspects of icing research (see for example Langmuir & Blodgett, 1946, or Stallabrass & Lozowski, 1978);
2. the Joukowski airfoil, which was the basis for early analytic flowfield calculations around airfoils;
3. NACA (US National Advisory Committee for Aeronautics) four and five digit

airfoils, the standard profiles for many helicopter rotor blades and general aviation airfoils; and

4. any airfoil defined at a series of points along its periphery.

Only the first two of these have analytic expressions available for the potential flowfield about them. In other cases, the flowfield is generated by the vorticity substitution method (Kennedy & Marsden, 1976). This method consists of solving for the vorticity density on a series of straight line segments approximating the airfoil surface, subject to the appropriate boundary conditions. The sum of the influence of the vorticity elements then yields the potential flowfield at any point outside the airfoil surface.

Having calculated the shape of the airfoil and the flow about it, the next step is to find the trajectories of the droplets making up the cloud. The equations of motion of such droplets are presented in Section 2.3.2. Their right hand sides are made up of the following terms: the acceleration of gravity, the decelerative drag due to the relative motion between the air and the droplets, and the deceleration produced by the finite rate at which vorticity may be shed from the fluid near the droplet. The most important factors affecting the trajectories are: the droplet inertia, which tends to make the droplet follow a straight line path; and the drag of the air, which tends to pull the droplet around the airfoil in much the same way as the air flows about the airfoil. Section 2.3.4 outlines the numerical algorithms which are used to integrate the differential equations of motion. The methods which have been used include the Runge Kutta 4th order, Runge-Kutta-Fehlberg 4th order and the 4th order Hamming Predictor-Corrector methods.

With the means of calculating the trajectories and the shape of the airfoil surface known, we then proceed to calculate which droplets strike the airfoil surface, and at what location. The uppermost and lowermost trajectories of droplets which collide with the airfoil are known as the grazing trajectories for a given airflow and droplet size. They define the total mass of impinging droplets over a given time interval. Other trajectories within the envelope will allow us to determine the fraction of the freestream mass flux of droplets which will be deposited at any point on the airfoil surface (*i.e.* the collision efficiency). We may then calculate the thickness of the ice layer which grows during a given accretion period. If there are several droplet size categories in the natural droplet size distribution, a combined or average collision efficiency may be used. The formulae for determining the accretion thickness take into account the curvature of the surface. All ice growth is assumed to take place normal to the underlying surface. If the droplets freeze rapidly as they impact, they tend to retain their shape, forming accretions of low density. Two formulations for the variation of the accretion density have been devised to calculate the accretion thickness. The positions of points defining a new airfoil surface may be computed based upon the accretion thickness at these points. The cross-sectional area of the accretion is determined and used to estimate the accuracy of the calculation of the new airfoil shape. A new set of vorticity elements may then be defined along the new airfoil surface. The stage is set for repeating this sequence of steps, thereby effecting the calculation of the time-dependent accretion on an airfoil.

Details of the steps outlined above follow in the remainder of this chapter. The techniques and formulae to be described are implemented in the program RIME. A listing of this program is given in Appendix G.

2.2 Airfoils and the airflow about them.

2.2.1 The flow regime about a helicopter rotor blade.

Lowry (1969) has described the airflow about a helicopter rotor blade as "an aerodynamic situation of exquisite intractability." Particularly during forward flight, many complex aerodynamic interactions occur between the rotor blade and other structural components. The rotor itself experiences a rapidly varying angle of attack, air velocity, and yaw (Reichert & Wagner, 1973). Since the blade is flexible, these fluctuations induce aeroelastic effects which further complicate the flow. As the tip of the advancing blade approaches the critical Mach number, compression of the air significantly alters the flow field (Hammond & Pierce, 1973). In addition, the retreating blade may approach the stall condition, where the lift cannot be maintained because of separation of the boundary layer (Reichert & Wagner, 1973).

Reichert & Wagner recommend that a complete aerodynamic model of the flow about a helicopter rotor blade should incorporate the effects of the boundary layer (including reverse flow and separation) and compressibility. In addition, the unsteadiness and the three-dimensional nature of the flow should be accommodated. However, Maskew & Dvorak (1978) conclude that "a thorough and exact calculation of the development of boundary layer separation is properly the domain of the time-dependent solution to the Navier-Stokes equations. Unfortunately, the computer does not yet exist which is capable of handling such a problem, and even if one did, the cost in computing time would be astronomical." Several approximations must thus be made to facilitate modelling of the airflow. These are:

1. Ignore the existence of the boundary layer. Except during a leading-edge stall, the thickness of the boundary layer along the leading half of a rotor blade is very small as compared to the blade chord length (Maskew & Dvorak, 1978). With the exception of very small droplets, it may be expected that the boundary layer influence upon the trajectories of impinging droplets will be short lived and thus minimal.

2. Avoid consideration of transonic flow regimes. Rotor blades may experience local transonic flow in two situations during forward flight. The first involves low angles of attack and Mach numbers of about 0.85 on the advancing blade. The other, high angles of attack (over 15°) and Mach numbers of about 0.5 on the retreating blade (Wortmann, 1973). If we avoid these conditions by restricting ourselves to moderate angles of attack ($\alpha \leq 10^{\circ}$) on the inner half-span of the blade, then compressibility effects will be minimal (Maskew & Dvorak, 1978). In addition, Brun *et al.* (1953) have determined that even at high subsonic local Mach numbers, the compressibility of the airflow has little effect upon most of the droplet trajectories. This is because the greatest effect of compressibility occurs very near the airfoil. In this small region, only those droplets moving slowly (that is the smallest ones) would be affected by the change in flow due to air compression or expansion. Larger droplets would cross the region too quickly for a significant change to occur in their trajectories. Brun *et al.* have found that the effect of compressibility upon the

total collision efficiency of a cylinder is less than three percent for all the cases they examined. They also claim that an extension of these results to airfoils is straightforward because compressibility alters the flowfield in much the same way as that about a cylinder. Furthermore, within the region where compressibility significantly alters the airflow, the sub-region of greatest change in the flow has been found to be further back along the airflow than the limits of impingement for all but the largest droplets.

3. Ignore three-dimensional and time dependent effects. Wortmann (1973) states that "the three-dimensionality and the unsteadiness of the flow over the blade airfoil are . . . of secondary importance . . . The flow on the blades is mostly two-dimensional."

With these restrictions and assumptions, we may treat helicopter rotor blade icing as a function of a steady incompressible, two-dimensional flow in a fluid without vorticity or viscosity. This allows us to consider potential flow fields about an airfoil, thereby keeping total modelling costs within reasonable limits.

2.2.2 Specification of the airfoil shape.

The first step in modelling icing is to specify the profile of a two-dimensional airfoil upon which we wish the accretion to occur. This will aid us in determining:

1. the flowfield about the airfoil;
2. the locations of droplet-airfoil collisions; and
3. the direction and thickness of ice accretion.

Only a few of the profiles may be defined analytically (the cylinder, the Joukowski airfoil, and some NACA airfoils are amongst this group). Others are specified at a limited set of points. Further, after the first layer of accretion, none of the resulting airfoils will have a shape easily defined analytically. For reasons of consistency, all airfoils are thus specified at a set of points, with the profile between such points being defined via cubic spline interpolation.

For all airfoil shapes, the coordinate system has been non-dimensionalized by the inital (before icing) airfoil chord length, C. In this non-dimensional coordinate system, the nose is at (0,0) and the tail is at (1,0). Let θ be defined as the polar angle measured from the negative x-axis. We will now define the profiles of a series of different airfoils.

2.2.2.1 The cylinder.

The cylinder has been included for validation purposes. Its upper and lower surfaces are defined by

$$x_U = (1 - \cos\theta)/2 \qquad (2.1)$$

$$x_L = x_U \qquad (2.2)$$

$$y_U = \sqrt{0.25 - (x_U - 0.5)^2} \qquad (2.3)$$

and

$$y_L = - y_U \qquad (2.4)$$

where x and y are the non-dimensional (ND) coordinates of points on the airfoil surface. The subscripts U and L refer to the upper and lower surfaces respectively (COORDS[65,70]).

2.2.2.2 The Joukowski airfoil.

This airfoil possesses the very useful attributes of having a profile very similar to that of certain helicopter rotor blades, and at the same time an analytical solution for the potential flow around it. It too has been included for validation purposes, and may be defined in the following way (after Houghton & Brock, 1970), by the transformation of the appropriate circle.

Let us start with a circle of radius a=b(1+e) shifted to the left of the origin by an amount be. Thus its coordinates are

$$x = - b(1 + e)\cos\theta - be \qquad (2.5)$$

and

$$y = b(1 + e) \sin \theta \quad (2.6)$$

Now if z and ζ are complex numbers such that

$$z = x + iy \quad (2.7)$$

and

$$\zeta' = \xi + i\eta \quad (2.8)$$

then the Joukowski transformation is:

$$\zeta = z + b^2/z \quad (2.9)$$

or

$$\xi = x[1 + b^2/(x^2 + y^2)] \quad (2.10)$$

and

$$\eta = y[1 - b^2/(x^2 + y^2)] \quad (2.11)$$

The airfoil length in the transformed coordinate system is thus $4b(1+2e+e^2)/(1+2e)$ which allows us to specify b to achieve an airfoil of unit length, viz:

$$b = 0.25 (1 + 2e)/(1 + 2e + e^2) \quad (2.12)$$

For any value of ξ the thickness of the airfoil is 2η, and thus the ratio of the airfoil thickness H to the chord length C is:

$$h = 2b(1 + e) \sin \theta\{1 - 1/[(1 + e)^2 + e^2 + 2e(1 + e) \cos \theta]\} \quad (2.13)$$

Since we wish h to have some predetermined maximum value h_o, we need to solve (2.13) for the appropriate values of e and θ. Because this equation would be difficult to solve analytically, it is solved numerically. The program (COORDS[71,103]) makes two initial approximations to e, finds the corresponding difference between h and h_o, and uses the Secant algorithm (see for example Burden *et al.*, 1978) to converge to a sufficiently accurate value of e. Within each step of the Secant algorithm, the program uses the Golden Section-search algorthm ZXGSN (IMSL, 1979) to find the value of θ in (2.13) which results in the maximum thickness for that value of e.

The above transformation places the endpoints of the airfoil at

$$\xi = -0.5(1 + 2e + 2e^2)/(1 + 2e + e^2) \quad (2.14)$$

and

$$\xi = 0.5(1 + 2e)/(1 + 2e + e^2) \quad (2.15)$$

for $\theta=0$ and $\theta=\pi$ respectively. One final transformation, a leftward shift of the origin, is made (COORDS[109,119]) to give the airfoil coordinates:

$$x_U = \xi + 0.5(1 + 2e + 2e^2)/(1 + 2e + e^2) \quad (2.16)$$

$$x_L = x_U \quad (2.17)$$

$$y_U = \eta \quad (2.18)$$

and

$$y_L = -y_U \quad (2.19)$$

The procedure of this section results in a Joukowski airfoil whose coordinates are much more accurate than those obtained by the classical formulae (see for example, Houghton & Brock, 1970).

2.2.2.3 NACA Four- and five-digit wing sections.

During the first half of this century, the United States National Advisory Committee on Aeronautics (NACA) designed a large number of airfoils. Of these, the four and five digit series of airfoils are of particular significance for this study because they have frequently been employed in helicopter main and tail rotors, and also in general aviation aircraft wings. The two series may be designated as

1. NACA $c_m c_p h_o$; where the first digit, c_m, gives the maximum camber as a

percentage of the chord; the second, c_p, gives the abscissa of this ordinate (in tenths of the chord), and the last two digits, h_o, specify the maximum airfoil thickness as a percentage of the chord length.

2. NACA $c_\ell c_p h_o$, where the design lift coefficient, in tenths, is three-halves the value of the first integer, c_ℓ, the second and third digits together, c_p, indicate twice the distance from the nose to the position of maximum camber in percent of the chord; and the last two digits, h_o, once again give the maximum airfoil thickness in percent of the chord.

Abbott and von Doenhoff (1959) have summarized the data for these two series of airfoils. They give the thickness distributions for the four and five digit airfoil series as:

$$y_h = 0.05\,h_o(0.2969\sqrt{x} - 0.126\,x - 0.3516\,x^2 + 0.2843\,x^3 - 0.1015\,x^4) \tag{2.20}$$

The expression in parentheses on the right hand side (RHS) of this equation has a value of 2.1×10^{-3} when x=1. Ideally this value should be exactly 0. Consequently, the amount 2.1×10^{-3} is removed in a linearly increasing fashion from x=0.3 to x=1.0, so as to give a razor sharp trailing edge to the airfoil profile. This refinement to the standard specification results in a more accurate flowfield generation by the method of Section 2.2.4.

The ND y coordinates and the angle of the slope of the mean line for the four digit series are given by the formulae:

for $x \le c_p$ $$y_c = c_m(2c_p x - x^2)/c_p^2 \tag{2.21}$$

$$\gamma = \tan^{-1}[2c_m(c_p - x)/c_p^2] \tag{2.22}$$

and

for $x > c_p$ $$y_c = c_m[(1 - 2c_p) + 2c_p x - x^2]/(1 - c_p)^2 \tag{2.23}$$

$$\gamma = \tan^{-1}[2c_m(c_p - x^2)/(1 - c_p)^2] \tag{2.24}$$

The mean line for the five digit family of airfoils is derived from the values of the two parameters c_k and c_m. These values, for several airfoils of interest, may be found in Table 1. The ND mean line ordinate and angle of slope from the x-axis are:

for $x \le c_m$

$$y_c = c_k[x^3 - 3c_m x^2 + c_m^2(3 - c_m)x]/6 \tag{2.25}$$

$$\gamma = \tan^{-1}\{c_k[3x^2 - 6c_m x + c_m^3(3 - c_m)]/6\} \tag{2.26}$$

and

for $x > c_m$

$$y_c = c_k c_m^3(1 - x)/6 \tag{2.27}$$

$$\gamma = \tan^{-1}[- c_k c_m^3/6] \tag{2.28}$$

Finally, the thickness distributions may be combined with the mean line (COORDS(12,64)) to obtain the coordinates of the upper and lower surfaces:

$$x_U = x - y_h \sin\gamma \tag{2.29}$$

$$x_L = x + y_h \sin\gamma \tag{2.30}$$

$$y_U = y_c + y_h \cos\gamma \tag{2.31}$$

and

$$y_L = y_c - y_h \cos\gamma \tag{2.32}$$

2.2.2.4 Special airfoils.

Recently, many modifications have been made to the standard sections described above. In order to permit the use of such airfoil sections, the program (MAIN[43,70]) will accept user input in the form of x and y coordinate values which define points along the upper and lower surfaces. Control element endpoints (CEE's), which are discussed in Section 2.2.4, may also be specified in this manner. The program documentation gives further details (refer to Appendix G).

2.2.3 Determining potential flow by analytical methods.

For an incompressible fluid in two-dimensional (2-D) motion, the continuity equation is

$$\frac{\partial u_a}{\partial x} + \frac{\partial v_a}{\partial y} = 0 \tag{2.33}$$

where u_a and v_a are the components of the ND air velocity in the x and y directions respectively. We define the ND streamfunction ψ to be

$$\psi = \psi_o + \int (u_a dy - v_a dx) \tag{2.34}$$

where ψ_o is a constant, and the line integral is taken along an arbitrary curve joining the reference point O to the point P with coordinates (x,y). After Batchelor (1970), we have

$$u_a = \frac{\partial \psi}{\partial y} \tag{2.35}$$

and

$$v_a = -\frac{\partial \psi}{\partial x} \tag{2.36}$$

The components of the air velocity are thus dependent upon the streamfunction at any point outside the airfoil. In order to evaluate equations (2.35) and (2.36), a finite difference scheme is employed, based upon the grid displayed in Fig. 5. These two equations become:

$$u_a = \frac{\psi_1 - \psi_2}{2r_d} \tag{2.37}$$

and

$$v_a = \frac{\psi_3 - \psi_4}{2r_d} \tag{2.38}$$

where r_d is the ND droplet radius.

There are two airfoils of interest in this study for which the streamfunction can be defined analytically: the cylinder, and the Joukowski airfoil. The cylinder is significant because it has been used in icing research as a stepping stone to more complex airfoil shapes. Papers exist which outline experimental and theoretical studies on cylinder icing with which we may make comparisons. Because the flow about a Joukowski airfoil may be determined analytically, this allows us to verify (using a realistic airfoil profile) the more general flowfield generating technique described in Section 2.2.4.

2.2.3.1 The cylinder.

The normalized or ND stream function, ψ, for a flow from left to right (with a unit ND velocity infinitely far from the cylinder) about a cylinder of unit diameter with center at (0.5,0.0) is (after Houghton & Brock, 1970):

$$\psi = y\{1 - 0.25/[(x - 0.5)^2 + y^2]\} \tag{2.39}$$

2.2.3.2 The Joukowski airfoil.

Let us begin with the ND streamfunction for a cylinder centered at the origin in a flow from left to right with a unit ND velocity infinitely far from the cylinder (U_∞=1). Thus (after Houghton & Brock, 1970):

$$\psi = y'' \left[1 - \frac{k_m}{2\pi(x''^2 + y''^2)} \right] \tag{2.40}$$

or in polar coordinates:

$$\psi = a \sin\theta - \frac{k_m \sin\theta}{2\pi a} \tag{2.41}$$

where k_m is the source strength, a is the radius of the cylinder, and x'' and y'' are the x and y coordinates. If we rotate the reference frame through an angle of α, we obtain:

$$x'' = x' \cos\alpha + y' \sin\alpha \tag{2.42}$$

and

$$y'' = y' \cos\alpha - x' \sin\alpha \tag{2.43}$$

The streamline ψ = 0 intersects with the cylinder at two points. Thus this streamline must also define the surface of the cylinder (if there is to be no flow through the surface), and we have that

$$\frac{1}{a}\frac{\partial\psi}{\partial\theta} = 0 \tag{2.44}$$

which implies that

$$k_m = 2\pi a^2 \tag{2.45}$$

Additionally, the ND streamfunction for a line vortex at the origin is

$$\psi = -k_k / \{2\pi \ln[\sqrt{x'^2 + y'^2} / a]\} \tag{2.46}$$

where

$$k_k = -4\pi a \sin\alpha \tag{2.47}$$

In order to produce a negative (clockwise) circulation and thus positve lift for this airfoil with its nose at the left, we have introduced a negative sign in (2.47). Finally, shifting the cylinder by be to the left of the origin gives:

$$x = x' - be \tag{2.48}$$

$$y = y' \tag{2.49}$$

and hence

$$\psi = y\cos\alpha - (x + be)\sin\alpha$$
$$- a^2[y\cos\alpha - (x + be)\sin\alpha]/[(x + be)^2 + y^2] \quad (2.50)$$
$$+ 2a\sin\alpha\,\ln[\sqrt{(x + be)^2 + y^2}\,/a]$$

This equation gives us the value of the ND streamfunction for a point in the z plane. Generally, we desire to know ψ for an arbitrary point (ξ', η') in the ζ' plane, where

$$\xi = \xi' - 0.5(1 + 2e + 2e^2)/(1 + 2e + e^2) \quad (2.51)$$

and

$$\eta = \eta' \quad (2.52)$$

Thus, we require the inverse Joukowski transformation. From (2.9) we have:

$$z^2 - \zeta z + b^2 = 0 \quad (2.53)$$

Finding the roots gives:

$$z = 0.5[\zeta \pm \sqrt{(\xi^2 - \eta^2 - 4b^2) + i(2\xi\eta)}] \quad (2.54)$$

If we set

$$g' = \xi^2 - \eta^2 - 4b^2 \quad (2.55)$$

$$h' = 2\xi\eta \quad (2.56)$$

and

$$j' = \sqrt{g'^2 + h'^2} \quad (2.57)$$

then

$$x = 0.5[\xi + \sqrt{(j' + g')/2}\ \mathrm{sgn}\ \xi] \quad (2.58)$$

and

$$y = 0.5[n + \sqrt{(j' - g')/2}\ \mathrm{sgn}\ n] \qquad (2.59)$$

Equations (2.50), (2.51), (2.52), and (2.55) through (2.59) may thus be used to obtain the ND streamfunction for flow about a Joukowski airfoil (PJK[1,22]).

2.2.4 Determining the potential flow for arbitrarily shaped airfoils.

The two previous sections have described how an analytical flow may be determined about a cylinder or a Joukowski airfoil. For all other airfoils used in this study (including these first two, following a layer of accretion), the flow must be found by a more general method.

Two classes of techniques exist for finding the potential flow about an arbitrarily shaped airfoil. The first involves the conformal transformation of a near circle to an airfoil, in a fashion similar to the one described in the previous section. Such transformations are iterated until the desired shape is achieved in the airfoil plane (see for example Theodorsen & Garrick, 1932).

The second class of techniques involves the use of a distributed set of singularities. The most referenced paper in this field (Hess & Smith, 1967) used a set of sources and sinks along the airfoil surface to estimate the flowfield. This basic method has been refined in a series of papers by various authors, leading to the method used in this study (Kennedy & Marsden, 1976). The primary difference between this method and most of the previous ones is that the boundary condition has been reformulated. It requires that the streamline which lies along the airfoil surface must pass through a point slightly aft of the trailing edge of the airfoil. Previous methods generally constrained the component of the velocity normal to the airfoil surface to be zero creating considerable computing difficulty in the region near the trailing edge where velocities are changing very rapidly with distance. Kennedy and Marsden claim that a program based upon their formulation will require less than one tenth the computing time required for a solution with similar accuracy using the Hess and Smith formulation. The Kennedy and Marsden method may be summarized as follows:

A set of points (CEE's) which lie on the airfoil surface are joined by straight line segments called control elements. Each element j has a constant vorticity density γ_j along it. At its center is a control point C_j. If we write an equation giving the total influence of the vorticity density for all control elements $j=1,\ldots,n;\ j\neq i$ on the flow at control point C_i, then the collection of such equations results in a system:

$$\psi + \sum_{j=1}^{n} \gamma_j K_{ij} = R_i \qquad i = 1,2,\ldots,n \tag{2.60}$$

where

$$R_i = y_i \cos\alpha - x_i \sin\alpha \tag{2.61}$$

$$K_{ij} = \left\{(s_{2j} + D_j)\ln(r_{1j}^2) - (s_{2j} - D_j)\ln(r_{2j}^2) + 2s_{1j}\tan^{-1}\left[\frac{2s_{1j}D_j}{s_{1j}^2 + s_{2j}^2 - D_j^2}\right] - 4D_j\right\}\Big/4\pi \tag{2.62}$$

and s_{1j}, s_{2j}, r_{1j}, r_{2j} and D_j are defined in Fig. 6. To close the system, we add one more equation similar to (2.60) applied at a control point just aft of the airfoil's trailing edge. This gives us n+1 equations involving the n unknown γ_j and ψ. The program (POTI(1,68)) solves this system using the IMSL subroutine LEQT1F. Additional details may be found in Kennedy & Marsden (1976).

After considerable testing, Kennedy and Marsden have found that at least 40 control elements are required over an airfoil surface in order to obtain good agreement with those cases where the streamfunction may be derived analytically. They distribute the control element endpoints by using a set of θ_j which are equally spaced about a circle in the airfoil plane. We instead space the θ_j equally about a circle in the circle plane. Thus

$$\theta_j = (j - 1)\Delta\theta \qquad j=1,\ldots,n+1 \tag{2.63}$$

where $\Delta\theta$ is a constant, or

$$x_j = (1 - \cos \theta_j)/2 \quad (2.64)$$

Equation (2.63) may be used in (2.1), or (2.5) and (2.6). Alternatively (2.64) may be substituted into (2.20) through (2.32).

The present study is especially dependent upon the accurate modelling of the flow in the vicinity of the airfoil nose. As a result, two modifications have been made to the procedures just described for those cases where the airfoil surface is defined analytically. First, the number of CEE's placed for $0 \le |\theta| < \pi/3$ ($0 \le |\theta| < \pi/2$ for the cylinder) may be specified independently of the number for $|\theta| \ge \pi/3$ ($|\theta| \ge \pi/2$ for the cylinder). Typically there will be an additional number of CEE's in the forward section. Also, in order to more accurately define the airfoil surface, additional surface segment endpoints (SSE's) may be placed between CEE's over the forward portion of the airfoil). This increases the total number of spline segments (to be discussed in Section 2.4.1) defining the airfoil surface, and results in a more accurate determination of the point of collision between a droplet and the airfoil. The distribution of CEE's and SSE's on a typical airfoil is displayed in Fig. 7.

2.3 Calculating the droplet trajectories.

2.3.1 Droplet-airfoil interaction.

This study involves the capture of cloud droplets by an arbitrarily shaped airfoil. The ratio of characteristic linear dimensions is about 1:10⁴ (O[10⁻⁵] for the droplets *vs.* O[10⁻¹] for the airfoil). Also the liquid water content of typical clouds was stated in Chapter 1 to be of the order of 1 part in 10⁶ by mass. It is assumed that these two factors combined remove the need for calculating the effect of the droplets upon the flowfield about the airfoil. This is in contrast to some previous work, such as that of Pitter & Pruppacher (1974) where cloud droplet – ice crystal interactions could only be modelled through the superpositioning of the flowfields about the droplets and crystals, because the linear dimensions were comparable.

With this complication removed, determining the droplet trajectories is accomplished by integrating the differential equations describing their accelerations in the undisturbed potential airflow about the airfoil.

With the trajectories of the droplets known, we may determine which droplets actually strike the airfoil surface, and where. This in turn will permit the calculation of the rate of accretion as a function of position along the airfoil surface, leading to the development of the thickness of the accretion.

2.3.2 The equations of motion.

We begin by assuming that the relative velocity between the droplets and the airflow is sufficiently low that we need not concern ourselves with deformation of the droplets. According to Pruppacher and Klett (1978), this assumption is valid (at least for droplets in free-fall) when the droplet Reynolds number:

$$Re_d = 2R_d|\bar{V}_d - \bar{V}_a|/\nu \tag{2.65}$$

is less then 260. In this equation, R_d is the droplet radius, $\bar{V}_d$ and $\bar{V}_a$ are the velocity vectors of the droplet and airflow respectively, and ν is the kinematic viscosity of the air. They state that circulations within the droplets do not have significant effect upon their drag. When $Re_d > 400$, periodic vortex shedding may induce oscillations in the droplets. These would affect the drag somewhat; however, it is clear in Appendix I that if such high Reynolds numbers are reached, it is only just prior to collision, and thus the time interval over which these secondary effects could influence the droplet motion is so small that they may be ignored.

The complete vector equations describing the accelerated motion of water droplets (having a fixed mass) in dry air are (following Pearcey & Hill, 1956, and Landau & Lifshitz, 1959):

$$\frac{d\bar{V}_d}{dT} = \frac{2(\rho_d - \rho_a)}{(2\rho_d + \rho_a)}\bar{G} - \frac{3C_D\rho_a}{4R_d(2\rho_d + \rho_a)}\left|\bar{V}_d - \bar{V}_a\right|(\bar{V}_d - \bar{V}_a)$$
$$- \frac{9\rho_a}{(2\rho_d + \rho_a)\,R_d}\sqrt{\frac{\nu}{\pi}}\int_{-\infty}^{T}\frac{d\bar{V}_d}{d\tau}\frac{d\tau}{\sqrt{T - \tau}} \tag{2.66}$$

and

$$\frac{d\bar{X}_d}{dT} = \bar{V}_d \tag{2.67}$$

where ρ_a and ρ_d are the density of air and water respectively, C_D is the droplet drag coefficient, $\bar{G}$ is the gravitational acceleration, and $\bar{X}_d$ is the droplet position vector.

Among the first to integrate (2.66) and (2.67) for the droplet trajectories were Langmuir & Blodgett (1946). They used a simplified version of (2.66) which may be written as:

$$\frac{d\bar{V}_d}{dT} = \frac{3}{8}\frac{\rho_a C_D}{\rho_d R_d}\left|\bar{V}_d - \bar{V}_a\right|(\bar{V}_d - \bar{V}_a) \tag{2.68}$$

Using an empirical fit between $C_D Re_d/24$ and Re_d, their formulation for the steady state drag term was:

$$C_D = 24/Re_d + 4.73/Re_d^{0.37} + 6.24 \times 10^{-3}\ Re_d^{0.38} \tag{2.69}$$

where the last two terms on the RHS of (2.69) account for the departure of the drag coefficient from the Stokes' value as Re_d increases.

More recent experimental work has led Sartor and Abbott (1975) to formulate a new expression for C_D which is claimed to be more accurate in the range $0.01 \le Re_d \le 5$, namely:

$$C_D = 24/Re_d + 2.2 \tag{2.70}$$

Using dimensional analysis and boundary layer theory, Abraham (1970) has derived the following formulation:

$$C_D = 0.2924(1 + 9.06/\sqrt{Re_d})^2 \tag{2.71}$$

which is valid for $Re_d \le 5000$. The program allows any of the droplet drag coefficient formulations to be chosen provided that they are within their range of applicability.

2.3.3 Non-dimensionalizing the equations.

For convenience in plotting the results, and for the sake of easy comparison between different test cases, the above equations have been put in a non-dimensional (ND) form. This form also allows us to condense a number of different combinations of conditions into a smaller number of non-dimensional cases. The normalizing parameters are the airfoil chord length C, and the freestream velocity U_∞. A list of correspondence between the standard and ND form of various quantities is given in Table 2. Since mass does not appear explicitly in these equations, a normalizing parameter was not chosen for this property.

The ND vectorial equation is thus:

$$\frac{d\bar{v}_d}{dt} = \frac{2(\rho_d - \rho_a)}{(2\rho_d + \rho_a)}\bar{g} - \frac{3\rho_a C_D}{4r_d(2\rho_d + \rho_a)}\left|\bar{v}_d - \bar{v}_a\right|(\bar{v}_d - \bar{v}_a) - \frac{9\rho_a}{(2\rho_d + \rho_a)r_d}\sqrt{\frac{\nu_a}{\pi}}\int_{-\infty}^{t}\frac{d\bar{v}_d}{d\tau}\frac{d\tau}{\sqrt{t-\tau}} \tag{2.72}$$

Bars over g, v_d, and v_a indicate a vector quantity. The first term on the RHS combines the buoyancy of the droplet in air and the gravitational acceleration. The second term is the steady viscous drag, and the third (referred to as the history term) is related to the finite rate of vorticity diffusion from the surface of the droplet in accelerated motion. The equation implicitly incorporates the droplet induced mass resulting from the momentum it imparts to the air as it accelerates.

Pearcey & Hill (1956) describe the basis for their inclusion of the third term on the RHS of (2.72). "A further effect occurs owing to the finite rate at which vorticity diffuses from the surface of the body. The distribution of vorticity throughout the medium depends upon the past velocity of the body and thus upon its history. The actual drag experienced at any particular time is more affected by the recent past history than by the distant past."

2.3.4 Integrating the equation with a steady drag.

2.3.4.1 The form of the equations.

The x component of (2.72) is of the form:

$$\ddot{x} = f_1[\bar{x}(t),\ \dot{\bar{x}}(t)] + \int_0^t f_2[t,\tau,\ddot{\bar{x}}(\tau)]d\tau \tag{2.73}$$

which is a second order Volterra integro-differential equation of the second kind. Finding suitable numerical methods for determining the solution of this type of equation is a topic of current research in numerical analysis (Makroglou, 1977, or Baker *et al.*, 1979). These state-of-the-art methods, as well as earlier methods (Pouzet, 1960) are very complex and difficult to implement. For this reason, we begin by adopting the approach of earlier investigators (Langmuir & Blodgett, 1946, and Sartor & Abbott, 1975); that is, to drop the history term as a first approximation.

In order to justify this approximation, we have estimated the potential importance of the history term in finding the correct solution to (2.72), by calculating an acceleration modulus as defined in Crowe *et al.* (1963):

$$A_m = 2r_d \left|\frac{d\bar{v}_d}{dt}\right| \Big/ (\bar{v}_d)^2 \tag{2.74}$$

The terms in (2.74) are estimated from the solution of (2.72) without the history term. According to Crowe *et al.*, when $A_m \gtrsim 10^{-2}$, the steady drag coefficients can no longer be used without appreciable error, and the history term should be included.

With the history term removed, we are left with a system of four first order differential equations:

$$\frac{dx_d}{dt} = f_3[t,x_d,y_d,u_d,v_d] = u_d \tag{2.75}$$

$$\frac{du_d}{dt} = f_4[t, x_d, y_d, u_d, v_d] = \frac{2(\rho_d - \rho_a)}{(2\rho_d + \rho_a)} g \sin \alpha$$

$$- \frac{3\rho_a}{4(2\rho_d + \rho_a)} \frac{C_D}{r_d} \left| \bar{v}_d - \bar{v}_a \right| (u_d - u_a) \quad (2.76)$$

with similar equations for y_d and v_d (ACCN[1,32]).

In general, the gravity term (the first term on the RHS of (2.76)) will be omitted in the results which follow because this term is much smaller than the other terms of this equation.

2.3.4.2 The integration of ordinary differential equations.

As summarized in Hamming (1973), there are three interrelated problems associated with the use of approximate numerical methods for finding the solution to an ordinary differential equation (ODE). They are:

1. "Amplifications of roundoff errors due to certain combinations of coefficients in the finite difference formulae,
2. "Truncation errors that arise from finite approximations for the derivatives, [and]
3. "Propagation errors (instability) that arise from solutions of the approximate difference equations that do not correspond to solutions of the differential equations."

In actual fact, the AMDAHL 470 V/8 used in this study (and the IBM System 370 upon which it is based) does not roundoff numbers to a specified number of hexadecimal digits, but rather truncates them. This tends to render less valuable the theories which have been developed for the propagation of round-off error because such theories are based upon a random process. This may be a good approximation for roundoff, but it is not for chopping, where the change is always in the same direction. Further, errors may increse linearly with the number of machine operations during chopping. This compares unfavorably with machines which roundoff, where errors generally increase as the square root of the number of operations. Consequently, in order to minimize the effects of roundoff or chopping, double-precision arithmetic has been used. This allows computations to proceed using fourteen hexadecimal or about 16.7 decimal digits, as opposed to 6 hexadecimal or 7.2 decimal digits for single-precision. In this way round-off errors are far less likely to contaminate the "significant" part of the final answer.

Truncation errors may be minimized by including as many terms as possible in the Taylor-like expansions of the finite difference approximations to the derivatives. Naturally, a trade-off is involved as greater accuracy is achieved by formulae of greater complexity. Generally, these formulae require an increased number of functions and derivative evaluations. It has been found (see for example Burden *et al.*, 1978) that methods having truncation errors of order four to six effect the best compromise between computing cost, accuracy, and ease of implementation.

In order to choose a numerical method to be used in solving a system of ODE's such as (2.75) and (2.76), we must test to see which of the available methods will be stable for this particular system. For an ODE such as

$$\dot{\bar{x}} = \bar{f}(t,\bar{x}) \tag{2.77}$$

instability may result if the problem is stiff. Wanner (1977) has defined stiffness as follows. "A differential equation problem is stiff, if some of the eigenvalues of $\partial\bar{f}/\partial\bar{x}$ have large negative real parts and if, at the same time, the interval of interest in the solution is relatively large."

The components of the Jacobian $J = \partial\bar{f}/\partial\bar{x}$ are found by the straightforward, if somewhat tedious, differentiation of f_3 through f_6 in (2.75) and (2.76) by x_d, u_d, y_d, and v_d respectively. All such derivatives have been evaluated analytically except $\partial u_a/\partial x_d$, $\partial u_a/\partial y_d$, $\partial v_a/\partial x_d$, and $\partial v_a/\partial y_d$. These are evaluated numerically (by a finite difference technique) using the grid shown in Fig. 5. The complex eigenvalues of the Jacobian are determined numerically using the IMSL subroutine EIGRF (STAB[1,46]). Further details will be found in Appendix A.

2.3.4.3 Methods for stiff problems.

In recent years considerable effort has been expended to develop programs designed specifically to handle arbitrarily stiff systems of ODE's. One of the best known programs is that of Gear (1971), known as DIFSUB. More recently better performing algorithms have been proposed by Liniger (1976), Cash (1980), and others. Papers have been written (see for example Enright *et al.*, 1975; or Hull, 1980) which compare the performance of various algorithms. Up to the present time, however, all programs designed to handle stiff systems have been substantially less efficient than those able to successfully integrate only mildly stiff or non-stiff systems. Thus in order to determine the degree of stiffness of the system (2.75) and (2.76), the relatively easy-to-implement Runge-Kutta algorithm was employed. This follows the advice of Shampine (1980). "It is obviously valuable to have programs for non-stiff problems which diagnose stiffness."

2.3.4.4 The Runge-Kutta fourth-order algorithm (RK4).

This traditional method (for a description of the algorithm see, for example, Burden *et al.* (1978), p. 244) is often used as a standard against which to compare other numerical techniques. It has a local truncation error of order four in the time step ($O[(\Delta t)^4]$), and requires four function evaluations per step. This compares favorably with the second-order Runge-Kutta technique, for example, which requires two evaluations per time step, but gives a greater truncation error for the same number of function evaluations ($O[(\Delta t/2)^2]$).

If we denote the eigenvalues of J by λ_k, $k=1,\ldots,4$, then Lambert (1980) shows that the RK4 method will be stable for weakly stiff equations provided that $Re(\lambda\Delta t) \geq -2.78$, where $\lambda = min(\lambda_k)$, $k=1,\ldots,4$.

2.3.4.5 The Hamming fourth-order predictor-corrector algorithm (PC4).

While RK4 gives useful results, it has the disadvantage of requiring a relatively large number of function evaluations per step. This is a problem in the present study because calculations of the air velocity tend to be the most expensive part of the integrating procedure. Thus minimizing the number of these calculations will result in greater efficiency. As a result, it was decided to evaluate a fourth-order predictor-corrector method designed by Hamming (1973) to have the greatest possible stability for a predictor-corrector method while at the same time minimizing the truncation error for this order of difference method. While the stability criterion is more stringent than for RK4 ($Re(\lambda\Delta t) \geq -1.4$), the method gives an improvement in efficiency over RK4 of about 21%. This method is somewhat more difficult to implement, however, and it also requires an explicit starting method (such as RK4) for the first three time steps.

The method is described on page 407 of Hamming (1973). Since the predictor and corrector are modified so as to reduce the truncation error, the method is actually of greater accuracy than $O[(\Delta t)^4]$.

2.3.4.6 The Runge-Kutta-Fehlberg fourth-order algorithm (RKF4).

While the PC4 algorithm is very efficient for mildly stiff systems with a given (and constant) step size Δt, the method is difficult to adapt to a frequently changing step size. Such a variable step size might be dictated by the requirements of stability, or in the interests of maintaining a constant local truncation error. The PC4 algorithm depends upon data from several time steps back, and thus the time step may be most easily changed by halving or doubling. Since each change requires a certain amount of overhead, to maintain efficiency, changes should not be made frequently

Fehlberg (1969) described a modification to the RK4 algorithm which allows an estimation of the local truncation error. This is accomplished by integrating the system of equations by both fourth- and fifth-order Runge-Kutta formulae, where the coefficients have been chosen to minimize the total number of function evaluations required. The local truncation error at each time step is then estimated via the difference between the answers provided by the fourth- and fifth-order formulae. If this error is less than a predefined tolerance, the integration of the ODE proceeds using a new time step. If not, the process is repeated using a smaller time step. In either case, the step size is adjusted so that the anticipated trucation error will be some specified fraction of the tolerance. Details of the RKF4 algorithm will be found in Appendix B.

The stability considerations for the routine RKF4 are identical to those given in Section 2.3.4.4 for RK4.

2.3.4.7 Estimating the global truncation error.

The program RKF4 described above estimates the local truncation error and adjusts the time step to maintain this estimate below a certain tolerance. It does not, however, provide an estimate of the global error in the solution of the problem. This latter quantity would be useful in assessing the confidence to place in the final answer.

A common, and perhaps almost traditional method for estimating the global truncation error in a given ODE integration, involves recomputing the solution using a smaller error tolerance (for methods based upon local error control). Equivalently, a smaller step size may be employed (for methods with a constant step size). The answers are then compared. Shampine (1980) emphasizes that some programs, when used to solve certain problems, may provide the same final answer even while the tolerance is reduced by five orders of magnitude! This might lead the user to the conclusion that the answer is as accurate as is possible on the machine being used. In actual fact, it may be nothing of the kind. The problem is that the above "procedure depends on a monotone behavior of the error with respect to the input tolerance" (Shampine, 1980). Since the programs were not designed to ensure this, one must be wary of estimating the global errors in this way.

Prothero (1980) has reviewed the state-of-the-art in algorithms designed for the efficient estimation of the solution of a system of ODE's and their associated global errors. He concludes that the best algorithm available is the one described by Shampine and Watts (1976), and implemented in their program GERK. A modified verson of this algorithm, as well as a similar one from the same paper, is implemented in the program.

When the order and step extrapolation algorithms were applied to determine the global errors in sample problems provided by Shampine and Watts, our results were similar to theirs. However, when exactly the same subroutine was given the task of estimating global errors in the droplet position and velocity components, these estimates consistently proved to be far smaller than the actual variations in these components when the local trucation error tolerance was changed. Thus, as the tolerance was tightened, the final droplet position and velocity at collision with the airfoil surface did not converge to an answer within the error range provided upon the answers using a less stringent tolerance. This leaves the value of these algorithms in doubt for this system of equations. Further work will be required to determine the cause for this failure.

2.3.5 Integrating the complete trajectory equations.

Throughout Section 2.3.4 we have concerned ourselves with the integration of the droplet equations of motion simplified by the use of a steady drag formulation. In this section we shall remove that restriction.

Section 2.3.4.1 described the difficulties involved in the integration of Volterra integro-differential equations of the second kind. Norment (1980), and others, have found that the acceleration modulus defined in (2.74) did not exceed 1/100, and thus that the history term need not be incorporated in the droplet equations of motion. Joe (1975) and List (1977), on the other hand, found the history term to significantly affect the trajectories of droplets which had bounced off hailstones. In several cases spread over a wide range of conditions, we have found A_m to exceed 1/100. Thus we decided to attempt to integrate the complete equations of motion (2.72). Appendix C describes the method which was used for this purpose.

2.3.6 The initial conditions.

Ideally the droplet trajectory integration should start infinitely far upstream from the airfoil, with the droplets having the same velocity as the air. For computational reasons, this is impractical. As a result, the program has been designed to allow a choice to be made as to how far-upstream from the airfoil nose, xo, to start the integration. The user may also choose the starting offset from the extended airfoil chord line, yo. These two parameters are illustrated in Fig. 8.

At the starting point, the program (TRAJEC[142,177]) calculates an initial droplet velocity which varies from the air velocity in such a way that the droplet Reynolds number, Re_d, is equal to 1/1000. This is necessary to prevent the initial Reynolds number from taking on a value of zero, leading to an infinite drag coefficient via (2.69), (2.70), or (2.71). It is possible to reformulate (2.72) so as to prevent this situation from occurring, but Chapter 3 will show that the Reynolds number increases rapidly within several time steps in any case, and that as a result this is a reasonable approximation in the circumstances. Furthermore, Chapter 3 presents the results of trajectories which begin at various values of xo. It will be shown that integrations beginning at least five chord lengths ahead of the airfoil nose produce trajectories which are sufficiently accurate that other approximations would mask the increased accuracy achieved by starting further back.

2.3.7 Integrating the equations just prior to collision.

Up to this point, Section 2.3 has described the search for high-order, high-accuracy solutions to the system of ODE's governing the droplet trajectories. This section will deal with the problems such methods encounter during the time step in which collision occurs between the droplets and the airfoils, and will describe the method used to circumvent the problem.

Let us imagine that during the time interval $(t_i, t_{i+1}]$ the droplet has collided with the airfoil surface. Reference to Fig. 5 will indicate the likelihood that at least one of the grid points 1 through 4 is then within the airfoil profile. The value of the streamfunction at this gridpoint will then be highly erroneous (since the streamfunction supplied by (2.39), (2.50), or (2.60) only applies outside the airfoil profile). This in turn will lead to an incorrect approximation for the air velocity at $(x_d, y_d)_{i+1}$, and possibly also at any other time during this interval after collision has occured. Appendix D shows that all of the integrators discussed above (RK4, RKF4 and PC4) use the value of the air velocity at some point within the time interval $(t_i, t_{i+1}]$ to find the position of the droplet at t_{i+1}. This leads us to question the accuracy of the droplet position $(x_d, y_d)_{i+1}$, and even as to whether or not the droplet really shoud have impacted in this time interval.

The above dilemma is solved by using a different type of ODE integrator after the droplet has passed the abscissa of the airfoil nose. A first approximation to the droplet velocity and position is made via a third-order Hermite extrapolation. The details of this formula are given in Appendix D. If extrapolation predicts a position which the methods of Section 2.4 indicate is within the airfoil profile, then we must find the collision location. If it predicts that the droplet has crossed a view window boundery (used for plotting purposes), then the location of this occurrence is estimated by the same method. If neither of these events has occurred, then the step is re-integrated using one of the higher order techniques mentioned above. If the latter integration predicts a collision (in contradiction to the extrapolation), then the contradiction is resolved by repeating the extrapolation using a time step one-half the size of the previous one (TRAJEC[253,266])

2.4 Accreting the ice.

Sections 2.2 and 2.3 have described how the profile of the airfoil of interest is generated; how the airflow surrounding the airfoil is determined; and the means by which we may calculate the trajectory of a water droplet within this flowfield. This section shall discuss the techniques which are employed to determine the point of collision between a droplet and the airfoil surface; the method of choosing the starting points for the trajectories; and the resulting calculations of the rate of ice accretion and its ultimate thickness after a given period of accretion. We shall also describe the means by which a new airfoil shape is determined, and the process of repeating the above steps for subsequent ice layers.

2.4.1 Specification of a continuous airfoil surface.

With the position of the droplet known at some time t_i, we need to know the distance between the droplet and the airfoil, that is, the closest approach. In order to accomplish this goal, the airfoil surface must first be defined in a continuous fashion. Section 2.2 dealt with the specification of the airfoil surface at a finite number of points (SSE's). An interpolation procedure is required to locate the airfoil surface between the specified points. The procedure chosen is the semi-clamped (or in case of the cylinder; clamped) cubic spline fitted independently to the upper and lower surfaces (front half surfaces for the cylinder).

Inspection of Fig. 7 will reveal that the slope of the airfoil surface at the nose is infinite. Attempts to fit the surfaces by a free spline, or by clamping the left end of the spline to a large positive or negative slope (depending upon the surface being fitted) have led to a poor interpolation. As a result, the upper and lower surfaces are rotated by plus and minus 30° respectively before fitting (see Kennedy and Marsden, 1976) (FIT[6,15]). This allows the angle of the slope at the nose to be clamped (in the new coordinate system) to plus or minus 60° respectively. The right end of each spline is left "free" in all cases except the cylinder. Here a problem exists once either surface is specified in the rotated coordinate system, because near the "tail" of the airfoil, the surface becomes double valued. The spline fitting routine employed will not interpolate in this situation. Since riming can occur only upon the front surface of the cylinder, and since the interpolated surface is only required where icing may occur, then the spline is fitted to only the front half of each cylinder surface, with the right end of the spline being clamped to a slope of $\mp\sqrt{3}$ (in the rotated reference frame) for the upper and lower surfaces respectively (FIT [16,30]).

The coefficients of the cubic polynomial interpolator between any two SSE's are determined via the IMSL (1979) subroutine ICSICU. With these coefficients known, the methods of Appendix E may be used to determine the distance ℓ from the nose along the airfoil surface to the point $(x,y)_a$.

Since the interpolation is performed upon points in a rotated coordinate system, an iterative approach must be employed to find the ordinate value of the airfoil surface corresponding to a given abscissa. The details of this approach will be found in Appendix F.

2.4.2 Finding the closest vertical approach between the droplet and the airfoil.

If we are to determine whether or not a particular droplet is to contribute to the accretion on the airfoil, we must be able to detect if and when it collides with the airfoil surface. With the position of the droplet specified at t_{i+1} by the Hermite extrapolating technique of Section 2.3.7, we need to know the closest vertical approach between the droplet and airfoil surface at that time. The closest vertical approach is defined as the distance AD in Fig. 9.

Collision cannot occur until the right edge of the droplet (x_d+r_d, yd) is to the right of the airfoil nose (x_N, y_N). Therefore, let y_{s1} and y_{s2} be the airfoil ordinates for the abscissae $\max(x_N, x_d)$ and x_d+r_d. The slope of the line joining these points is:

$$S_L = \sqrt{[x_d + r_d - \max(x_N,x_d)]^2 + [y_{s2} - y_{s1}]^2} \tag{2.78}$$

The coordinates along the droplet surface which are closest to the airfoil surface are thus:

$$y_D = y_d \pm r_d^2/S_L \tag{2.79}$$

and

$$x_D = x_d \mp r_d(y_{s1} - y_{s2})/S_L \tag{2.80}$$

where the first sign in each equation applies on the upper surface of the airfoil. The closest vertical approach is thus

$$y_{CLAP} = y_D - y_A \tag{2.81}$$

where y_A is the airfoil surface ordinate at xo (WHAMO [49,59]).

2.4.3 Determining the point of impact.

Section 2.4.2 discussed the method of determining the closest vertical approach. When y_{CLAP} is positive (for the upper surface) at t_i and negative (or zero) at t_{i+1} then we may conclude that a collision has occurred in the time interval (t_i, t_{i+1}). The situation is illustrated in Fig. 10. The problem is to find the point (x^*, y^*) where $|y_{CLAP}|$ is less than a predetermined tolerance. With the values of y_{CLAP} known at two different values of x_d, that is at time t_i and at time t_{i+1}, we may employ the Secant algorithm to iterate upon x_d and y_{CLAP} to find (x^*, y^*). At each value of x_d between $(x_d)_i$ and $(x_d)_{i+1}$, however, we must be able to determine y_d. The method for doing this involves finding the appropriate root of the cubic Hermite extrapolating function which fits x_d to t (WHAMO [1,46]). With this value of t, we may find the value of y_d from the Hermite cubic polynomial fitting y_d to t.

The components of the velocity of the droplet at the moment of impact, u^* and v^*, are found in a similar fashion, that is through the Hermite cubic polynomials extrapolating u_d and v_d as functions of t. The angle of the tangent to the trajectory at the instant of impact is given by:

$$\theta_T = u^*/v^* \tag{2.82}$$

The angle of the perpendicular to the airfoil surface from the tangent to the trajectory (at impact) is:

$$\theta^* = \theta_S - \theta_T \qquad (2.83)$$

where θ_S, the angle of the slope of the perpendicular to the surface, is determined by the methods of Section 2.4.7. By the definition of θ^* (see Fig. 10), we desire to have

$$-\pi/2 \leq \theta^* \leq \pi/2 \qquad (2.84)$$

The program ensures that when the calculation of (2.84) is made, the answer is translated into the appropriate quadrants so as to fall within this range (TRAJEC [375,386]).

2.4.4 Finding the grazing trajectories.

The methods of the previous section allow us to determine, at the moment of impact between a droplet and the airfoil, the droplet's position, its velocity, and the angle from the normal to the airfoil surface at which it impacted. Inspection of Fig. 8 reveals that there is only one trajectory on the upper (lower) surface where θ^* will be equal to $\pi/2$ ($-\pi/2$). Any trajectories above (below) this one will not collide with the upper (lower) airfoil surface. These two trajectories (one on each surface) are the grazing trajectories. Their significance will be explained in Section 2.4.5, where the local and total collision efficiencies will be defined. For the present, it suffices to emphasize that these trajectories should be determined accurately. Since ℓ changes rapidly for small changes in yo when yo is near its grazing value, determining the value of ℓ at grazing, ℓ_G, can be difficult. Langmuir & Blodgett (1946) were able to identify the grazing trajectories by calculating the paths of several droplets which impacted within the grazing trajectory envelope. They then employed a theorem (valid only for cylinders) which enabled an accurate estimate of the grazing trajectory collision point. Bragg *et al.* (1981) determined an interpolator between the droplet impingement angle (the angle between the tangent to the trajectory and the tangent to the airfoil slope) and the starting ordinate yo for several trajectories within the envelope. Extrapolation of this function was then used to approximate the value of yo and thus ℓ at grazing (yo_G and ℓ_G respectively).

Because the rate of change of yo with ℓ becomes very small as we approach the grazing trajectory, a small error in the estimate of yo_G will result in a large error in the estimate of ℓ_G. The method for finding the grazing trajectory outlined below allows us to approximate these values to within a specified tolerance, not by extrapolation, but rather by ensuring that the grazing trajectory does indeed fall within the tolerance we have set.

The program may perform the task of locating the grazing trajectories most efficiently, if reasonable estimates exist from which to begin the iterative procedure described below. Thus, the user may select the manual trajectory mode, and when prompted, input the droplet size, the trajectory integrating tolerance ϵ, and the droplet starting position (xo, yo). When the resulting trajectory results in a ND value of Y_{CLAP} of 0.01 or less, the appropriate value of yo may be entered into the input file, and the auto-trajectory mode selected. If the airflow or the airfoil is asymmetrical, then a grazing trajectory estimate will be required for the lower surface as well. In this case, a resulting value of Y_{CLAP} which is greater than −0.01 is required.

The technique which is employed to determine the grazing trajectories is similar for the upper and lower airfoil surfaces. Following the first trajectory, the starting ordinate for the second trajectory is given by

$$yo_{i+1} = yo_i - 0.95\, Y_{CLAP} \tag{2.85}$$

where the constant 0.95 has been determined by trial and error to lead to the best second estimate for yo, and i is a trajectory index. After two trajectories have been calculated, the third and subsequent trajectory starting ordinates may be determined by use of the modified Secant algorithm, viz:

$$yo_{i+1} = yo_i - kY_{CLAP_i}(yo_i - yo_{i-1})/(Y_{CLAP_i} - Y_{CLAP_{i-1}}) \tag{2.86}$$

where the constant k initially has the value 0.85. The Secant method is repeated until one of three cases arises.

1. The sum of two successive values of Y_{CLAP} is less than 0.00002. The

constant k is replaced by

$$k = k + 0.1 \tag{2.87}$$

and the Secant algorithm is continued.

2. $y_{CLAP_i} \geq y_{CLAP_{i-1}}$
In this case, (2.85) is used to find the next value of yo.

3. A collision occurs. A test is carried out to determine if $y_{CLAP_{i-1}} \leq 1.5 \times 10^{-5}$. If so, then the last trajectory is deemed to be the grazing one. If not, then we check to see if $\pi/2 - |\theta^*| \leq \kappa$, where initially κ has the value 0.2°. If this relation is true, then it indicates the grazing trajectory. If not, the next trajectory starting ordinate is chosen to be midway between the two previous values. Also the constant k is replaced by k−0.05, and κ is replaced by $\kappa + 0.1^{o}$. If this next trajectory hits the airfoil, the process is repeated. If it misses, then the Secant algorithm is recalled to find the next value of yo.

The above procedure is continued until a grazing trajectory is found for the upper surface. If necessary, the process is repeated for the lower surface. The advantage of this rather complex procedure is the knowledge that when the criteria are finally met, the true grazing trajectory must lie between the last trajectory to miss and the last trajectory to hit the airfoil. This provides an estimate of the error of yo_G. These procedures may be found in the subroutine TRAJEC[483,513].

2.4.5 Determining the collision efficiency.

2.4.5.1 Definitions of β and E_m.

Following Langmuir and Blodgett (1946), we define the local collision efficiency, β, as:

$$\beta = (dyo/d\ell) \cos \alpha \tag{2.88}$$

where yo is the y-coordinate of the trajectory starting point, and ℓ is the distance along the airfoil surface from the nose to the point of collision. The factor $\cos \alpha$, which does not appear in Langmuir and Blodgett, is necessary here because the x and y axes are fixed with respect to the original airfoil chord line, rather than to the flow at infinity.

Physically, β may be interpreted as the ratio of the mass flux impacting with the airfoil surface, to the freestream mass flux. This concept is made clearer by reference to Fig. 8. Following the central pair of trajectories in this figure, we see that the mass which flows through a plane perpendicular to the flow at infinity is deposited along the airfoil surface between ℓ_1 and ℓ_2. Because of the two-dimensional nature of the flow, the local collision efficiency is is simply

$$\lim_{(yo_2 - yo_1) \to 0} \left[\frac{(yo_2 - yo_1)}{(\ell_2 - \ell_1)} \right] \cos \alpha \qquad (2.89)$$

In a similar fashion, we note that using the grazing trajectories yo_{GU} and yo_{GL}, we may form a definition for the total collision efficiency:

$$E_m = (yo_{GU} - yo_{GL}) \cos \alpha / h_o \qquad (2.90)$$

In this equation, the subscripts GU and GL refer to the grazing trajectories on the upper and lower surfaces, respectively, and h_o is the ND maximum airfoil thickness. The two quantities β and E_m are related by:

$$E_m = \frac{1}{h_o} \int_{\ell_{GL}}^{\ell_{GU}} \beta d\ell \qquad (2.91)$$

Physically, E_m, may be interpreted as the ratio of the total mass flux impacting with the airfoil surface to the freestream mass flux passing through an "invisible plate" (that is, one that does not disturb the flow) of width h_o. From (2.91), we gain an appreciation of the need for determining the grazing trajectories accurately if we are to estimate the total mass of all the water droplets impacting with the airfoil surface.

2.4.5.2 Locating additional trajectories within the grazing trajectory envelope.

In order to calculate the thickness of the ice which accretes at any point on the airfoil surface, over a given time interval, it is necessary to know the mass flux of droplets colliding with the surface at that point. This flux is the product of the freestream mass flux and the local collision efficiency, β, at that point. We see that a knowledge of β is required along the entire airfoil surface.

A large number of calculations is required to determine a single droplet trajectory. In fact, such calculations contribute significantly to the overall cost of running the program. To prevent an excessive number of trajectories from being required, we must attempt to determine the β curve as accurately as possible using the smallest possible number of trajectories. This goal may be attained if we are able to meet three requirements:

1. a means of locating trajectory starting points which will lead to an accurate interpolating function for the β curve.
2. a way of deciding when to stop adding more trajectories; that is, when the β curve is sufficiently accurate.
3. a β curve interpolator which is smooth (that is, continuously differentiable), but not overly smoothed (to the point of masking relevant information).

It was discovered that requirements 1 and 3 are generally in conflict with one another. Thus, separate techniques have been developed to meet each of them. This section describes the methods used to satisfy requirements 1 and 2, while the next section will deal with 3.

Let us begin with the two grazing trajectories which were determined by the methods of the previous section. If we add several other trajectories within the grazing trajectory envelope (the means of accomplishing this are discussed below), then we have a sequence of n data points (for n trajectories): $(\ell, yo)_i$, $i = 1, \ldots, n$. We may interpolate upon this set of points to give us a yo *vs.* ℓ curve. Then, if we find the slope of this curve, we obtain an estimate of the β *vs.* ℓ curve. Since it is this latter curve in which we are most highly interested, we must choose the (ℓ, yo) points which will result in the most accurate interpolation for the β curve. Examples of these two curves are shown in Figs. 11 and 12.

Investigations into the nature of a yo *vs.* ℓ interpolator based upon cubic spline functions determined that the most stable curves resulted when the points (ℓ, yo) were as evenly spaced as possible, confirming the advice of Spath (1974). However, since the shape of the curve is not known in advance, it is difficult to accomplish this goal without wasting poorly placed trajectories. Furthermore, it was discovered that if two adjacent points were too closely spaced as compared to the others, wild oscillations often resulted in the β curve. These problems were solved by employing a cubic Hermite interpolator HERMIT [1,8]. The resulting cubic polynomials are not coupled between adjacent intervals (as cubic splines are via their second derivatives), and thus changes to the interpolating polynomials in the intervals adjoining a new point in the set do not propagate through the curve as they do for cubic splines. However, in order to fit cubic Hermite polynomials, the values of β must also be specified at the (ℓ, yo) datapoints. In order to accomplish this goal, pairs of trajectories are calculated for each point on the yo *vs.* ℓ curve (except at the ends, where we know that β must equal zero). These pairs of trajectories yield information which leads to mean values of yo, ℓ, and β at each point i, i=2, ..., n−1, where there are n datapoints (ℓ, yo, β). Thus if + and − signs may be used to distinguish the upper and lower trajectory of each pair, then

$$yo_i = (yo_i^+ + yo_i^-)/2 \tag{2.92}$$

$$\ell_i = (\ell_i^+ + \ell_i^-)/2 \tag{2.93}$$

and

$$\beta_i = (yo_i^+ - yo_i^-)/(\ell_i^+ - \ell_i^-) \tag{2.94}$$

Using Hermite interpolators, we have more freedom in our choice of spacing between the points defining the curve. Thus we may choose to attempt to space the points equally on a normalized β *vs.* ℓ curve, in order to maximize the accuracy of the resulting interpolator. The curve is normalized by the range in ℓ, that is by $\ell_R = \ell_{GU} - \ell_{GL}$, and by twice the range in β, that is $\beta_R = 2\beta_o$, where β_o is the maximum value of β on the curve. This is necessary because the length of the curve, between any two points would otherwise be a function of the scaling factor between β and ℓ. Thus if the Hermite cubic polynomial interpolator between any two points $(\ell, yo, \beta)_i$ and $(\ell, yo, \beta)_{i+1}$ is

$$yo = c_{3,i}\delta^3 + c_{2,i}\delta^2 + c_{1,i}\delta + yo_i \tag{2.95}$$

where

$$\delta = \ell - \ell_i \tag{2.96}$$

then the normalized form of the equations is:

$$yo_N = b_{3,i}\delta_N^3 + b_{2,i}\delta_N^2 + b_{1,i}\delta_N + yo_{Ni} \tag{2.97}$$

where

$$b_{3,i} = c_{3,i}\ell_R^3/\beta_R \tag{2.98}$$

$$b_{2,i} = c_{2,i}\ell_R^2/\beta_R \tag{2.99}$$

$$b_{1,i} = c_{1,i}\ell_R/\beta_R \tag{2.100}$$

$$yo_{Ni} = yo_i/\beta_R \tag{2.101}$$

and

$$\delta_N = \delta/\ell_R \tag{2.102}$$

From (2.94) we may derive an expression for β_N, viz:

$$\beta_N = 3b_{3,i}\delta_N^2 + 2b_{2,i}\delta_N + b_{1,i} \tag{2.103}$$

Armed with a means of interpolating the curves, we may return to our objective of locating data points to yield an accurate β curve for the smallest numbers of data points. The method of Appendix E may be used to find the lengths of the cubic polynomial segments given by (2.103). Once the longest segment has been found, we will attempt to locate a new datapoint (ℓ,β) midway between the two datapoints bounding the segment. This is accomplished by the Secant algorithm, which iterates upon δ_N and L (the length along the curve from the point $(\ell,\beta)_i$ to the point corresponding to δ_N) until L equals half the length of the curve segment. With the value of δ_N known, the corresponding value of yo can be found from

(2.102) and (2.95). We may then determine the starting positions for another pair of trajectories, again find the corresponding values of ℓ and β, and add one more point (ℓ, yo, β) to be interpolated (CE[116,186]).

When the newly interpolated curve is compared with the previous version, point by point, at say 200 points between ℓ_{GU} and ℓ_{GL}, then the maximum difference in β between the two curves may be determined. The process of adding datapoints through the calculation of trajectory pairs may be continued until the change between successive interpolated curves falls below a predetermined tolerance. We may also insist that a minimum number of datapoints exist to be interpolated (CE[99,115]).

2.4.5.3 Finding a smooth β v.s. ℓ interpolator.

The method of the previous section ensures that when a data point is added to the set of points to be interpolated, changes to the interpolated curve can only occur in the segments immediately adjoining the new point. Thus a point which is poorly placed because of insufficient accuracy in the trajectory calculations cannot influence the whole curve, causing wild fluctuations in the interpolator. Such fluctations act as a magnet for further datapoints since the lengths of curve segments which oscillate frequently will tend to be greater than curve lengths between other more accurate datapoints. A disadvantage of this method is that the second derivative of the yo *vs.* ℓ interpolator (that is, the slope of the β curve) need not be continuous, and the interpolated curve may, in some instances, take on a segmented look. With the number of datapoints to be interpolated fixed after completion of the procedure described in the last section, several alternatives are available to alleviate this problem.

The first option is to interpolate the (ℓ, yo) points by a cubic spline. This would result in a curve with a continuous second derivative, and thus would lead to a smooth β curve. However, the values of the datapoints would not be interpolated, and thus our goal of utilizing our available information most efficiently would not be attained.

The second option is to interpolate the (ℓ, β) points by a cubic spline. This will result in a smooth curve, but unfortunately the resulting curve may lack accuracy because, for example, the total collision efficiency would likely not equal the integral under the β curve, as it should according to (2.91).

The third option is to fit the (ℓ, yo, β) points with a quintic spline (Spath, 1974). This curve possesses all of the advantages of the first two options without their disadvantages. Further, the interpolator will utilize all the information in the data set. For example, the area under the β curve between any two data points will be equal to the total collision efficiency within the trajectory envelope corresponding to this pair of datapoints. A requirement of this method is the specification of the slope of the β curve at the ends of the curve. This requirement has been met by assuming that these end slopes are related to the average slopes of the β curve in the two nearest intervals, viz:

$$S_{\beta 1} = 0.5 \tan[2 \tan^{-1} (S_{1,2}) - \tan^{-1} (S_{2,3})] \tag{2.104}$$

and

$$S_{\beta n} = 0.5 \tan[2 \tan^{-1} (S_{n,n-1}) - \tan^{-1} (S_{n-1,n-2})] \tag{2.105}$$

where

$$S_{1,2} = (\beta_2 - \beta_1)/(\ell_2 - \ell_1) \tag{2.106}$$

and similarly for $S_{2,3}$, $S_{n,n-1}$, and $S_{n-1,n-2}$.

The program interpolates the dataset via the third option, that is the quintic Hermite spline. Then, it determines the difference between the interpolated β values of this curve and the cubic Hermite curve at 200 points along the curve's length. If the maximum difference is less than some predetermined tolerance, the quintic spline is adopted for further use. If the difference between the curves is too great, it implies that oscillations exist in the quintic spline because of poorly placed datapoints. In this case, the Hermite cubic spline of the last section becomes the interpolated curve (CE[187,200]).

2.4.5.4 The combined collision efficiency for droplet distribution.

Chapter 1 described the types of droplet distributions which may be encountered in an icing cloud. To this point, this section has dealt with methods of determining the collision efficiency of an airfoil for a monodisperse distribution of droplets. The challenge of extending this method to realistic droplet distributions remains.

The simplest approximation to the natural droplet size distribution is to assume that all droplets have the mass median diameter D_{mm} of the natural size spectrum. In this case, half of the liquid water content of the cloud will be composed of droplets with diameters less than D_{mm}, and half above this size.

Greater accuracy in determining the true airfoil collision efficiency when encountering a natural droplet distribution would be achieved by dividing the natural distribution into a set of categories, each catory being represented by droplets having the mass median diameter for that group. Fig. 13 shows the Langmuir "D" distribution, for example, divided into five categories, each representing 20% of the total liquid water content of the cloud. The associated representative droplet sizes are factors of 1.75, 1.27, 1.00, 0.77, and 0.50 times the mass median diameter of the entire distribution, D_{mm}. The methods described earlier in this section may be employed to determine the collision efficiency of each category separately. Then, for any point on the airfoil surface, the collision efficiency for the natural distribution of droplets may be approximated by the sum of the collision efficiency values for the separate categories β_i, each weighted by the fraction of the total liquid water content which that category contributes to the total w_i. Thus we have:

$$\overline{\beta}(\ell) = \sum_{i=1}^{n} w_i \beta_i(\ell) \qquad \text{for } n \leq 5 \qquad (2.107)$$

Naturally, the total LWC in any approximate distribution must equal the total LWC of the distribution being modelled. The only disadvantage of this method is its cost. The computing effort expended in running the entire program is approximately proportional to the number of categories chosen to model the natural droplet distribution.

A compromise between a single droplet size category, and a set of five categories (the maximum number permitted in the present program) is two categories. The results of such an experiment are described in Chapter 3, with a representative pair of collision efficiency curves displayed in Fig. 14. Inspection of this graph reveals the problem that exists in combining the two curves at the point where the inner curve (corresponding to the smaller droplets) falls to zero collision efficiency. The combined β curve typically has a kink in it at this grazing value of ℓ. In order to remove this kink, a variable length Boxcar filter (Jenkins and Watts, 1968) was applied to each collision efficiency curve prior to the curves being combined (CE[346,382]). A variable length filter was chosen because there is very little need to apply any smoothing near the peak of the β curve, but there exists a greater need near the ends of this curve. The method used in filtering may be described briefly as follows.

Smoothing the β curve with a Boxcar filter of length $F_v(\ell)$ simply consists of replacing the value of $\beta(\ell)$ with the average value of β between $\ell - F_v(\ell)/2$ and $\ell + F_v(\ell)/2$, viz:

$$\beta_F(\ell) = \frac{1}{F_v(\ell)} \int_{\ell - F_v(\ell)/2}^{\ell + F_v(\ell)/2} \beta d\ell \qquad (2.108)$$

or

$$\beta_F(\ell) = [yo(\ell + F_v(\ell)/2) - yo(\ell - F_v(\ell)/2)]/F_v(\ell) \qquad (2.109)$$

If $\ell - F_v(\ell)/2 < \ell_1$ or $\ell + F_v(\ell)/2 > \ell_n$, where ℓ_1 and ℓ_n are the limiting values of ℓ at the grazing trajectories, then we may replace the corresponding value of yo in (2.109) by either yo_1 or yo_n, as appropriate. The form of the variable length filter is:

$$\begin{aligned} F_v(\ell) &= F - 0.9\, F(\ell - \ell_1)/(\ell_o - \ell_1) \quad \text{for } \ell_1 \le \ell \le \ell_o \\ &= 0.1\, F + 0.9\, F(\ell - \ell_o)/(\ell_n - \ell_o) \quad \text{for } \ell_o \le \ell \le \ell_n \end{aligned} \qquad (2.110)$$

where ℓ_0 is the value of ℓ corresponding to the peak of the β curve (that is at β_0), and F is the maximum length of the Boxcar filter. The user may indirectly control the maximum length of the Boxcar filter by inputting the value of the ratio F/ℓ_R, which sets the maximum filter length for a given droplet size category as some fraction of the total length of accretion on the airfoil surface produced by droplets from that category. Further discussion on the effectiveness of this technique follows in Chapters 3, 4, and 5.

2.4.6 Finding the accretion thickness.

The previous sections of this chapter have outlined the methods used to predict the mass flux of water droplets impinging upon any point of the airfoil surface. This section shall be concerned with determining the thickness of the accretion which results.

Rime ice is formed when supercooled water droplets collide with a substrate under conditions in which the droplets freeze upon impact. The deposit temperature remains sufficiently below freezing so that no runback of liquid water occurs. The density of the deposit will depend upon the degree of deformation of the impacting droplets as they freeze, but by this definition, the density will be less than or equal to that of solid ice.

On a microscopic scale (that is, the scale of the individual droplets), the rime structure is influenced by the stochastic nature of the impaction process. One realization of a numerical simulation of this process is displayed in Fig. 15 (after Lozowski, 1981). Droplets from a monodisperse size distribution enter from the left with their ordinate value selected from a uniform random distribution. They continue to the right, flowing with the uniform airflow (which is assumed to be unaffected by the accretion which may have already taken place). The droplets continue this motion until either they pass through the right edge of the figure, or until they impact with another droplet forming part of the accretion. When collision occurs, it is assumed that the droplets freeze instantly, retaining their original shape. Inspection of this figure reveals considerable variations in the accretion density, depending upon the location and size of the sampling area. The existence of rime feathers (features characterized by their long, slender appearance) is also predicted by this simulation. The angle of growth predicted by this model averages about 20°.

On the macroscopic scale which is simulated in the present study, stochastic fluctuations in the rime density have not been modelled. Nor have we attempted to predict the formation of rime feathers. Both features are beyond the predictive capability of the deterministic formulations employed herein. Instead, we have assumed either a constant rime ice density, or one which is dependent upon the droplet diameter, the impact velocity, and the deposit temperature.

The mass of the accreted ice actually deposited upon the airfoil is the product of the collision efficiency and the coalescence efficiency. This latter term is simply the ratio of the mass of water droplets which stick to the airfoil, to the mass of water droplets impinging upon the airfoil. We have assumed the coalescence efficiency to be unity in this study. Hallett (1980) cautions, however, that at aircraft speeds, some splashing of impacting water droplets may occur if the droplets are sufficiently large. He has determined that when the ratio of droplet kinetic energy to surface energy exceeds about 20, some loss of mass may occur. Until this ratio exceeds 200, however, the loss of mass will not be important.

We assume all accretion to grow normal to the airfoil surface. Bragg *et al.* (1981) have investigated the result of allowing the accretion to grow in the direction from which the droplets have arrived at their impact location. The shape of the accretion formed in this way can be substanially different from that presented in the following chapters, but Bragg *et al.* were unable to conclude which formulation might better approximate experimental results. This remains a question for further investigation.

In the limit where only a small number of droplets impinge onto the airfoil surface, forming a layer about one droplet diameter thick, the growth will be approximately normal to the original airfoil surface. Thus if the accretion process is treated as a time dependent process, with a layer being formed during each of several discrete accretion periods (rather than during one extended interval), then the assumption of normal growth may better approximate the natural accretion process. The program has been written to facilitate time dependent, multi-layer accretion.

Details of this formulation are presented in the remainder of this chapter .

2.4.6.1 Accretion on a flat surface.

The thickness Ω of the accretion that would occur on a flat surface, oriented perpendicular to the airfoil which does not disturb the flow, is a function of the droplet mass flux through the plate, the period of the accretion T_A, and the ice density ρ_I, viz:

$$\Omega = U_\infty W T_A / \rho_i \qquad (2.111)$$

where W is the cloud liquid water content, and U_∞ is the freestream air velocity. This may be non-dimensionalized via the chord length C to give a ND accretion parameter:

$$\omega = \Omega / C \qquad (2.112)$$

With the mean collision efficiency $\overline{\beta}$ or $\overline{\beta}_F$ known for a point on the airfoil surface, we can estimate the ND local accretion thickness, if the surface were locally flat, to be:

$$\mu = \omega \, \overline{\beta}_F \qquad (2.113)$$

2.4.6.2 Accretion on a curved surface.

If the airfoil surface is not locally flat, as is the case for most parts of the airfoils studied, then the thickness of the ice accretion calculated by (2.113) may be in error, especially if the surface has a small radius of curvature. To correct for the effect of the surface curvature, a new approach is required.

Fig. 16 shows a curved section of an airfoil surface having on it a layer of ice accretion of thickness m. If the representative radius of curvature of the original surface at point O at time t_1 is r_1, and at t_2 is r_2, then the cross-sectional area of the accretion over the time interval $[t_1, t_2]$ is:

$$\sigma = \frac{M(r_2^2 - r_1^2)}{2r_1} \tag{2.114}$$

$$= Mm + 0.5\ Mm^2/r_1$$

where M is the length along the surface at t_1 from points P to Q. Point O is located at SSE i, point P is midway between SSE's i−1 and i, and point Q is midway between SEE's i and i+1. The ND cross-sectional area on a locally flat surface of length M and thickness p (where p is given by (2.113)) is:

$$\sigma = p\ M \tag{2.115}$$

Combining (2.114) and (2.115), and solving for m gives:

$$m = -r_1 + \sqrt{r_1^2 + 2r_1 p} \qquad \text{for } r_i > 0$$
$$m = -r_1 - \sqrt{r_1^2 + 2r_1 p} \qquad \text{for } r_i < 0 \tag{2.116}$$

These formulae are implemented in subroutine ICING [190] and [234].

The problem of determining a representative radius of curvature for the airfoil surface remains. The standard formula for the radius of curvature of a surface defined by:

$$y = f(x) \tag{2.117}$$

is

$$r_c = \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{1.5} \Big/ \frac{d^2y}{dx^2} \tag{2.118}$$

Now, the spline interpolator between SSE's i and i+1 is:

$$y_R = C_{3,i}\delta_R^3 + C_{2,i}\delta_R^2 + C_{1,i}\delta_R + y_{Ri} \tag{2.119}$$

where

$$\delta_R = x_R - x_{Ri} \tag{2.120}$$

and the coordinates of SSE i are (x_{Ri}, y_{Ri}). Combining (2.118) and the derivatives of (2.119) yields the radius of curvature at SSE i:

$$r_{ci} = \frac{(1 + C_{1,i}^2)^{1.5}}{2C_{2,i}} \tag{2.121}$$

If an averaged value of the radius of curvature is required at SSE i, we may use the arithmetic average of the values at SSE's i-1, i, and i+1. This averaged value has been found particularly useful in smoothing out small irregularities in the surface smoothness. Such irregularities, which might be caused by a lack of smoothness of the β curve for the previous accretion layer, can cause substantial fluctuations in the collision efficiency over relatively small surface distances. Such fluctuations lead to a rapid amplification of the original perturbation as the number of layers increases. For this reason the averaged radius of curvature is usually employed in the program (ICING [231] and [187]).

2.4.6.3 Accommodating a variable ice density.

The density of the rime accretion is influenced by stochastic fluctuations in the ice deposition, which can lead to rime feathers, for example. This cause for density variation will not be modelled. A second factor affecting the density is the degree of droplet deformation during the freezing period following the droplet's impact with the airfoil surface. This deformation is a function of the interval over which the freezing takes place, the droplet radius R_d, the impact speed V^*, the temperature of the accretion surface θ, the temperature of the droplets just prior to impact, and the rate at which the airfoil surface is being ventillated, among other factors.

Once we know the density of the ice, we may calculate the thickness of the accretion m_{VD}, viz:

$$m_{VD} = m/\phi \tag{2.122}$$

where ϕ is the ND ice density, that is the calculated density normalized by the density of pure ice $\rho_I = 917\ \mathrm{kg\ m^{-3}}$.

A series of experimental observations has led Macklin (1962) to derive an empirical formula for the accreted ice density when droplets of radius R_d (in μm) impact with a surface of temperature θ_s (in degrees Celcius) at a speed of V^* (in m s⁻¹) This equation may be normalized by the density of pure ice to give (ICING [170,176] and [219,224]):

$$\phi = 0.120(-R_d V^*/\theta_s)^{0.76} \tag{2.123}$$

where the following restrictions hold:

$$-20 \le \theta_s \le -5 \tag{2.124}$$

$$0.88 \le R_d V^*/\theta_s \le 16.29 \tag{2.125}$$

Since (2.123) was empirically derived, the droplet radius actually refers to the mass median radius of the droplet distribution extant in the measuring wind tunnel. The impact speed similarly refers to the speed of the these same droplets. The program calculates droplet impact velocities for each of the droplet size categories which are used to model the natural droplet distribution. Thus, if we wish to apply (2.122) and (2.123) at some point on the airfoil surface, we must determine the impact velocities for each of the droplets which are able to impact at that point, and find a combined velocity which is determined in a fashion similar to that used to calculate the combined collision efficiency at that point, viz:

$$\bar{V}^*(\ell) = \left[\sum_{i=1}^{n(\ell)} w_i \ V_i^*(\ell)\right] \Bigg/ \left[\sum_{i=1}^{n(\ell)} w_i\right] \tag{2.126}$$

In this equation, $n(\ell)$ refers to the total number of the droplet size categories whose droplets impact at ℓ, and as before, the w_i refer to the fraction of the total liquid water content of the natural distribution contributed by droplets from the category i. These calculations take place in subroutines ICING [92,94], ICING [150,152] and COLVEL [1,29].

Equation (2.123) requires a knowledge of θ_s, which has not been discussed previously. Since the present version of the program does not solve the thermodynamic aspects of the accretion problem, we must assume that the deposit temperature is the same as that of the airstream θ_∞, which is an input parameter.

The variables V^*, V_i^*, and $\overline{V}^*$ may be defined by either of two formulations when used in (2.123). The first is the total velocity of the impacting droplets. The second is the component of this velocity which is normal to the airfoil surface at the point of impact. In either case, the velocity is obtained by interpolating upon the datapoints (ℓ, v^*, θ^*) with cubic splines. The datapoints are obtained from each pair of impacting trajectories calculated by the methods of section 2.3. The variable θ^* is the angle from the normal to the airfoil surface to the tangent to the droplet trajectory at impact (ICING [166,169] and [215,218]).

2.4.7 The airfoil shape following a layer of accretion.

Section 2.4.6 has allowed us to determine the thickness of the ice accretion layer that forms at any point on the airfoil surface during a given time interval. If we are to repeat the process, that is to accrete subsequent layers, we must know the shape of the airfoil following each accretion interval.

The slope of the normal to the airfoil in the rotated reference frame at surface segment endpoint (SSE) i may be obtained from (2.119) as

$$S_i = -1/C_{1,i} \tag{2.127}$$

The equation of the normal to the surface is thus:

$$(y - y_i) = S_i(x - x_i) \tag{2.128}$$

However, we also know that the length of this line joining a SSE on the old surface to the corresponding SSE on the new surface must be m or m_{VD}. Thus:

$$(x - x_i)^2 + (y - y_i)^2 = m^2 \tag{2.129}$$

Combining (2.128) and (2.129), and solving for x and y yields:

$$x = x_i + \frac{S_i}{|S_i|}\sqrt{\frac{m^2}{1 + S_i^2}} \tag{2.130}$$

and

$$y = y_i + S_i (x - x_i) \qquad (2.131)$$

The point (x,y) becomes a SSE on the new airfoil surface, following accretion, with respect to the old rotated reference frame (ICING [191,192] and [235,236]).

Section 2.4.1 outlined the method used to interpolate cubic splines to the upper and lower airfoil surfaces. One requirement of the method was a set of rotated coordinates with origin at the airfoil nose. Since the position of the nose (which is defined as the point on the surface having the smallest abscissa value in the unrotated coordinate system) has likely changed following a layer of accretion, the first step is to locate its new position. Fig. 17 displays the two airfoil surfaces, at time t_k and at t_{k+1}. Also shown are rotated reference frames for the upper surface centered upon the new and old nose positions. Given an abscissa value for a point on the new airfoil surface in the old rotated frame, x_R, the subroutine NSURF calculates the abscissa of this point in the old unrotated frame. This subroutine is employed in turn by the IMSL (1979) subroutine ZXGSN to find the new nose position through the use of the Golden section-search algorithm for locating the minimum value of a function in a given interval (ICING [253,280]).

With the new nose position located, a new SSE is created at this point. All other SSE's on the new airfoil surface are then tagged as belonging to either the new upper or lower surface, depending upon their ordinate value as compared to that of the new nose. The old rotated coordinate system is then translated so that the new system has its origin co-located with the new nose position. With this accomplished, we are in a position to fit cubic splines on the new upper and lower surfaces, employing the new rotated coordinate systems (ICING [281,424]).

2.4.8 The cross-sectional area of the accreted layer.

A cross check upon the accurary of calculating the new airfoil shape is provided by comparing the mass of the accretion layer calculated in two different ways. The first method involves finding the product of the cross-sectional area of the accretion, A_T, and the ice density ρ_I. In this 2-D problem, the spanwise length is unity. The second way is to determine the product of: the accretion parameter $\dot{\Omega}$, the ice density ρ_i, the total collision efficiency E_m; and the maximum airfoil thickness H_o. In order to make this comparison, we need to find the cross-sectional area of the accretion layer.

The formula for the cubic spline interpolator of the airfoil surface in the rotated reference frame is given by (2.119). If we integrate this equation between SSE's i and i+1 with respect to the distance variable δ_R, we obtain the area under the surface spline segment with respect to the rotated reference coordinates. This integral is:

$$I_{k,i} = \int_0^{\hat{\delta}_i} y_R d\delta_R$$
$$= 0.25\ c_{3,i}\hat{\delta}_i^4 + 0.33\ c_{2,i}\hat{\delta}_i^3 + 0.5\ c_{1,i}\hat{\delta}_i^2 + y_{Ri}\hat{\delta}_i \tag{2.132}$$

where $\hat{\delta}_i = x_{R(i+1)} - x_{Ri}$, and $c_{3,i}$ through $c_{1,i}$ are the coefficients of the cubic polynomial spline segment between SSE's i and i+1. Maintaining our attention on the upper surface for the moment, we may then sum over all n_k segments making up the old surface to obtain the total area of this surface above the old rotated x-axis, viz:

$$I_k = \sum_{i=1}^{n_k} I_{k,i} \tag{2.133}$$

Repeating the process for the new airfoil (k+1) with respect to the new rotated coordinates (that is those centered on the new nose) gives

$$I_{k+1} = \sum_{i=1}^{n_{k+1}} I_{k+1,i} \tag{2.134}$$

Referring to Fig. 17, we see that the new nose position in the old rotated coordinate system is (x_{NR}, y_{NR}). Since the integrals I_k and I_{k+1} are with respect to different coordinate systems, we must make an adjustment to the difference between the integrals if we are to find the area between the two curves. This adjustment involves the rectangle with area $y_{NR}x_{TR}$, where x_{TR} is the abscissa of the airfoil tail in the old rotated system. It also involves the triangle with area $0.5\, x_{NR}y_{NR}$, whose hypotenuse joins the two nose positions. Combining these adjustments with (2.133) and (2.134) gives us the cross-sectional area of the accretion on the upper surface:

$$A_U = I_{k+1} - I_k - x_{NR}y_{NR}/2 + y_{NR}x_{TR} \qquad (2.135)$$

A similar expression may be derived for the area between the lower surfaces, A_L, and the two combined to give the total cross-sectional accretion area for the layer (FIT [33,66]):

$$A_T = A_U + A_L \qquad (2.136)$$

2.4.9 Placement of the control element endpoints on the new airfoil surface.

Section 2.4.7 outlined the method used to locate the SSE's on the k+1 airfoil surface, that is, after the accretion of layer k. If we are to solve for the airflow about the new airfoil shape (a step vital to the time-dependent modelling of the accretion process), the Kennedy & Marsden technique of section 2.2.4 requires that we locate a set of CEE's on the new airfoil surface. We have several requirements to satisfy in placing these CEE's:

1. We wish to retain the locations of the CEE's on those parts of the airfoil surface where no change has occurred, that is, where there has been no accretion.
2. A CEE should be located at the new airfoil nose position.
3. If we are to maintain a reasonable cost in computing the droplet trajectories, the total number of CEE's should not increase substantially as the number of accreted layers increases.
4. The CEE's on the newly accreted surface should be spaced apart in a fashion which is consistent with their spacing on the previous surface.
5. If an area of strong surface curvature exists, CEE's should be placed so that the straight-line elements joining them approximate the curved surface reasonably well.

Let us confine the remainder of this discussion to the upper surface – the procedures used for the lower surface are virtually identical. Our first step is to locate the CEE which lies immediately aft of the accretion region. Assume that the index of this CEE is j, and the index of the co-located SSE is i. This CEE and all those aft of it retain their previous positions (with appropriate changes for the translation of the rotated coordinate system). Let us define two ratios: that of CEE to SSE indices; and of lengths from the new and old nose positions along the new and old surfaces to CEE j; viz:

$$Q_S = (j - 1)/(i - 1) \tag{2.137}$$

$$Q_D = \ell_{k+1,j}/\ell_{k,j} \tag{2.138}$$

We desire to ensure that the lengths on the new surface between CEE's foward of CEE j remain approximately proportional to the corresponding lengths on the old surface. If the same number of CEE's were desired on both surfaces, then the ratio of these corresponding length intervals would simply be Q_D. But, the lengths between corresponding pairs of SSE's vary as the thickness of the accretion and the radius of curvature vary over the accretion region. Thus, exact correspondence is generally unattainable. Further, we position CEE's on the new surface only at positions where SSE's are located. Since there are $1/Q_S$ surface segments for each control element in this region, adjustments are necessary to effect a compromise between the conflicting goals of proportional spacing, and co-locating the CEE's with some SSE's. This is accomplished by ensuring that the distance between successive CEE's on the new surface divided by the corresponding distance on the old surface is:

$$\frac{\ell_{k+1,j} - \ell_{k+1,j-1}}{\ell_{k,j} - \ell_{k,j-1}} \geq \min\,[Q_D, Q_{DM}]\,(1 - 0.5\,Q_S) \tag{2.139}$$

The parameter Q_{DM} is chosen to be the maximum allowed increase in length between corresponding CEE's on the two surfaces. It must be sufficiently small so that the spacing between CEE's on the new surface is smaller near the nose (where typically the greatest curvatures occur) than farther back, along the non-accreted airfoil surface. This method of locating CEE's will tend to be most successful when the ratio Q_S is small, implying a large number of SSE's to choose from when locating a new CEE. The last of our goals above is satisfied by checking the ratio of the lengths between two CEE's along the airfoil surface, to that along the straight line joining them. If this ratio exceeds 1.2, a new CEE is placed midway between the other two, given a sufficient number of SSE's in the region. These formulae are implemented in subroutine ICING [425,524].

2.5 Time-dependent accretion modelling.

Section 2.2 has described the flow regime about a helicopter rotor blade and has detailed the methods used in generating the potential flowfield about an airfoil of arbitrary shape. It also has developed the techniques for defining the initial airfoil surface. Section 2.3 has outlined the calculation of the water droplet trajectories, which begin their paths several chord lengths ahead of the airfoil, moving in virtually the same fashion as the air surrounding them. Section 2.4 has described the means by which the impact locations of the droplets are found, the manner in which the collision efficiency is calculated, how the thickness of a layer of ice accretion is determined, and the method for finding the shape of the airfoil surface following this layer of accretion.

If the entire process is repeated, beginning this time with the new airfoil surface shape, then the revised flowfield will lead to a new collision efficiency curve(s), and a new distribution of ice for the following layer. Once the shape of the resulting airfoil surface has been found, the process may be repeated.

This sequence of steps forms the basis for the time-dependent modelling of the ice accretion which can form on an airfoil.

3. CODE OPTIMIZATION

3.1 Introduction

In the course of writing a lengthy and complex program, a number of occasions arise where small adjustments must be made to standard algorithms. There are usually even more locations where tolerances must be set for program branch points, that is places where branches may be taken depending upon whether or not a variable exceeds a given tolerance. In addition to these, programs may provide the user with a set of input options. Appendix H outlines the adjustments, options and tolerances embodied in the program described in Chapter 2. This chapter details the sequence of trials that were used to estimate the optimum values of the various tolerances and options.

3.2 Optimizing User options and Input values.

Appendix H describes the locations of adjustments and tolerances which are built into the program, and which should not require frequent user alteration. This section, on the other hand, will detail those options and adjustments which are required as input to the program each time it is run. There are also a number of input values which define the conditions of each simulation. These will generally not be described here. A complete list and description of user inputs is given at the beginning of the program (see Appendix G).

The discussion of this section will center around a sequence of trial simulations which were carried out to approximate optimal values for various options and tolerances, and also to test the sensitivity of the simulation results to changes in these values. The primary goal of this procedure was to increase the program's efficiency, that is to attain the smallest expenditure of computing effort for a given level of accuracy.

3.2.1 Control elements and velocity calculations.

One of the first steps that the program takes (after finding the airfoil shape) is to calculate the air velocity at points outside the airfoil profile. The accuracy with which such values are calculated (when the Kennedy and Marsden technique is employed) is dependent upon the number and spacing of the control element endpoints used. The greater the number of endpoints, and the shorter the straight line segments joining them, the better these segments approximate the true airfoil shape, and thus give the correct vorticity density along the segments. On the other hand, we anticipate that the cost of computing droplet trajectories will depend greatly upon the number and distribution of endpoints. Thus they must be located with care.

We shall begin by choosing an appropriate case for these optimization experiments. The program is more severely tested (particularly in determining the collision efficiency curve) when the angle of attack ALPHA is non-zero. Also, it is essential to choose an airfoil shape for which we know the solution for the velocity of the potential airflow at any point. This profile should also resemble a typical helicopter rotor blade cross-section so that the conclusions drawn from these tests will be applicable to the more general cases to be run later in subsequent chapters.

The conditions chosen conform to those used in a paper with which comparisons will be made in the next chapter (Werner, 1973). The angle of attack is ALPHA = 4.6°; the airfoil is an uncambered Joukowski profile of length C=0.711 meters, and thickness THICK=12.0%; the freestream airspeed is VINF= 128.6 m s^{-1}; the temperature of the freestream air is TINF=-20°C; and the freestream static pressure is PINF=101.3 kPa. Finally, in order to request that velocities be calculated at specified input points, we have set VINQ=1. The results of several runs are displayed in Table 3.

The errors in the calculation of the velocity at all points upstream of one chord length ahead of the airfoil are less then one percent for all combinations of NEF and NEB tested. For each of the upper and lower airfoil surfaces, these two parameters are the number of CEE's on the front third, and on the remainder of the airfoil surface respectively. Their method of placement was described in Section 2.2.4. Just ahead of the nose position, that is at (-0.01,0.001), the error in velocity shows an inverse dependence upon NEF. This relation also holds true farther back near the lower airfoil surface (0.12533,-0.05328). On average, the results for the two points just above the upper surface ((0.005,0.015) and (0.01,0.0185)) display this result again, although the accuracy at any given point appears to be related to its position with respect to the nearest CEE's. These results are to be expected – as the distance between CEE's decreases, the elements better approximate the airfoil profile, and thus lead to a better estimate of the air velocity.

Kennedy and Marsden (1976) recommend that a minimum of 41 CEE's be used to define the entire airfoil surface. The case just meeting this requirement (NEF=11, NEB=10) has a maximum error of less than 1.7% for all the points interrogated, and thus it was chosen as a suitable representative case for the experiments to follow in the next section.

3.2.2 Control elements and trajectory calculations.

The results of the previous section have provided some idea as to the number of CEE's required to model sufficiently accurately the potential airflow about an airfoil. But these runs are only the first step because we suspect that the accuracy of the accretion mass and shape is highly dependent upon the accuracy of the droplet trajectory calculations, and thus on the accuracy of the air and drop velocities at all points along the droplet path. To study the dependence of the droplet collision location upon the parameters NEF, NEB, and EPS (the truncation error tolerance), a series of experiements was conducted. The results are summarized in Table 4.

Column 5 gives the local collision efficiency (defined in Chapter 2) for a particular pair of trajectories impacting slightly back of the nose on the upper airfoil surface. Column 7 displays the distance of impact of the upper droplet trajectory from the nose along the airfoil surface. Columns 6 and 8 show the errors in these values as compared to the values for an analytical airflow. Column 9 compares the computing cost of calculating the pair of trajectories using the Kennedy and Marsden approach (rows 2 through 15) to the cost of the analytical approach (row 1).

In the first few rows (1 through 7) EPS is constant, while the ratios of NEF to NEB increases. As NEF increases, the relative cost increases, as does (in general) the accuracy of the simulations. An exception is row 5, where NEB=7. This implies that accuracy will be maintained provided that NEB ≥ 10. The cost also escalates as the sum NEF + NEB increases (see rows 6 and 7). Keeping NEF and NEB relatively constant for rows 8 through 15 allows us to vary EPS. One result is predictable: as EPS decreases, the relative cost increases. However, changes in the accuracy seem to be small and variable. The best overall compromise appears to be row 15, where the errors in β and in ℓ^* are both relatively small. Further, this is one of the least expensive of the runs presented, and at the same time it has a relatively small truncation error tolerance EPS. Based upon these results, the values of EPS, NEF, NEB and NIF chosen for the simulations of the next section are those in row 15.

An interesting sidelight may be provided by comparing the components of the costs associated with obtaining the results of this table. Using the case of row 15 as an example, the computing cost of loading the entire program and computing the control element vorticity density by the method of section 2.2.4 is about 51 cents. The additional cost involved in the computation of two trajectories is $2.84 when the RKF4 method (PC=2) and the full set of trajectory equations (EQN=2) are used. From this we can see that while the efficiency of the technique used in determining the vorticity density is high (as claimed by Kennedy and Marsden, 1976), the cost of computing air velocities at every time step (and sub-time-steps in the case of the RK4 and RKF4 algorithms) can be very substantial. This knowledge emphasizes the need to calculate the collision efficiency curve as accurately as possible. In this connection, we will now investigate the relative costs of the RKF4 and PC4 algorithms.

As was mentioned in Section 2.3, the Hamming PC4 method utilizes a minimal number of function evaluations (air velocity computations) per time step, but it is restricted to a constant time step. The algorithm RKF4 on the other hand uses more evaluations per step, but is is able to change the step size after each time step so as to maintain the largest time step consistent with the associated truncation error tolerance. For this comparison we have chosen the system of equations without the history term (EQN=1), because this term may interfere with the time-step selection method of RKF4 (described in Appendix B), and/or the error-mop-up process of PC4 (Hamming, 1973). The results of the comparison are given in Table 5. The columns have the same meaning as those in Table 4, except that column 1 now gives the type of ODE integrator used, and column 7 displays the ND step size in the step just prior to collision.

The table shows that the relative cost between the RKF4 and analytical solutions is approximately the same as that for the full set of equations (shown in Table 4). We assume that a fair comparison between the RKF4 and PC4 methods is made when the step size of the smallest time-step for the RKF4 method is similar to the constant step size for PC4 method. This assures similar truncation errors for the two techniques in the region just prior to collision where the air velocity is changing most rapidly. The result of this assumption is a computing cost for the PC4 method which is 5.7 times that of the RKF4 method. Clearly, any method using constant time steps, no matter how efficiently, is not suitable for the solution of this system of equations.

With the RKF4 algorithm having been singled out as the most appropriate of the ones attempted, and with suitable starting values of EPS, NEF, NEB and NIF having been determined, we shall now move to choose the most appropriate values of other constants and options.

3.2.3 Program sensitivity testing for monodisperse droplet distributions.

Let us retrict ourselves to results from modelling with a single droplet size. We must choose appropriate tolerances and options to obtain the collision efficiency curve (and thus the accretion shape and mass) which best approximate the true values for a given computing effort. The parameters we shall choose to vary are the method of finding the potential airflow (analytical (TYPE 0) *vs.* vorticity density (TYPE>0)), the number and location of CEE's and SSE's (given by NEF, NEB and NIF); the system of equations used (EQN=2 for the full set; EQN=1 without the history term); the drag coefficient formulation (CDS=1 for the hybrid Stokes-Sartor and Abbott-Abraham method; CDS=2 for the Langmuir and Blodgett method), the truncation error tolerance (EPS); the maximum permissible change in the β curve after incorporating the last trajectory pair (CEDEL); and the trajectory starting point abscissa (X0). The results for a series of experimental runs where these parameters were varied are displayed in Table 6. In column 12 we have the total accretion area. All other columns have meanings similar to columns in Tables 4 and 5.

We begin (in row 1) with an analytical run in which tolerances are set to the fine end of the range within which we wish to experiment. A plot of the flowfield about the Joukowski airfoil of this case is shown in Fig. 18. This figure also displays the trajectories used to calculate the collision efficiency curve of Fig. 19, and the ice accretion profile of Fig. 20. Changing the number of points specifying the airfoil surface, as in row 2, results in a small inaccuracy, and a small saving in expenditure. Changing the tolerance CEDEL (row 3) decreases the cost still further by requiring one fewer trajectory to be calculated. Since the grazing trajectories are unchanged, E_m remains constant, but the shape of the β curve changes somewhat. Changing the drag efficient formulation (row 4) increases the overall cost, and results in a significant change in E_m. Returning CDS to its normal value and relaxing the truncation error tolerance (row 5) yields values for β_o, ℓ_o and E_m almost identical to those in row 1, but at half the cost. Removing the history term from the droplet trajectory equations (row 6) effects a substantial improvement in relative cost (to 0.29), but incurs some errors in β_o, ℓ_o, E_m and A_T. A further change of X0 (row 7) from −10 to −5 has little effect upon either the simulation accuracy, or the cost. Comparing rows 8 to 5, we see that the decrease in X0 from −10 to −5 more than compensates for the increased cost in altering CEDEL from 4.0 to 1.0, and with virtually no degradation of accuracy. If we relax the tolerance EPS once again, and begin the trajectories only 2.5 chord-lengths ahead of the airfoil, the relative costs drop from 0.44 to 0.31.

Let us now confine our attention to the subset of the runs with a relative cost of less than 0.50. Of these, the ones in rows 5, 8 and 9 are the most accurate. The least expensive one is that in row 9. Thus a suitable compromise with which to begin the simulations of the next section will be a run employing the values EQN=2; CDS=1; EPS=1×10^{-4}; CEDEL= 1.0; and X0=−5 or −2.5 for a droplet of diameter 20 μm

3.2.4 Program sensistivity testing with a variable number of droplet size categories.

The results of the previous section were based upon simulations using a single droplet size. In actual fact, natural clouds have a distribution of droplet sizes within them. Let us assume that the Langmuir "D" distribution (see Fig. 9) fairly approximates a typical natural cloud droplet distribution. Then we may employ several droplet size categories (up to five in the present version of the program) to approximate airfoil icing. Simulations using only one droplet size (for example, that of the mass median diameter for the entire distribution) are predicted to produce less accuracy. The results for a series of such simulations are presented in Table 7.

Column 1 of this table indicates the length of the Boxcar filter used in smoothing the β curve (C denotes a constant length filter, V a variable length one). This length is the fraction of the total length (in ℓ) of the β curve. Column 2 gives the number of droplet size categories. Columns 3 through 8 display, respectively, the mass median droplet diameter, the fraction of the total LWC in that particular size category, the truncation error tolerance, the maximum value of the β curve, the location of this maximum, and the total collision efficiency for each of the categories. Columns 9 and 10 display the mean values of β_o and ℓ_o for the set of categories, or the corresponding values of the smoothed β curve, when filtering is applied. Column 11 gives the accretion cross-sectional area based on the mean and/or filtered β curve, while column 12 displays the relative cost as compared to the most accurate multi-category simulation, that of Case 1.

From Table 7 we may make the following observations. Case 1 displays the most accurate and comprehensive simulation we have made for these conditions (the conditions used for these simulations are the same as those in Section 3.3.1). A plot of the β curves for this case is given in Fig. 21, with the mean curve (as defined in Section 2.4.5.4) superimposed as a heavy solid line without symbols. The other curves are nested, with the curve for the smallest droplets having the smallest peak value β_o. We see that the all curves have peaks at approximately the same location. As the droplet size increases, the area under the β curve, corresponding to the total mass of ice accreted from that size category, increases as well. Further, the impingement limits (that is the locations of the grazing trajectories) increase in distance from the nose as the droplet diameter increases. Case 2 is virtually identical, except that the values chosen for EPS correspond more closely to the optimum value suggested from Section 3.2.3. There is a small change in the height of the β curve peak, but the overall accuracy is nearly as high at 79% of the computing cost. From this point, we continue to relax tolerances by reducing the number of droplet size categories. In Case 3, there are three categories. The relative cost of this simulation is 0.53, and the accuracy of the simulation appears to be very good indeed. Cases 4, 5 and 6 combine a two-category distribution with several different category weights (that is relative contributions to the total LWC) and representative droplet diameters. The trend in these simulations is toward greater accuracy (and only marginally increasing cost) as the weights approach an even split, that is 50% each. Cases 7 through 10 illustrate the results of simulations using constant and variable length filters on the β curves. The need for this smoothing is apparent if we examine Fig. 21 and compare the center β curve (with triangle symbols) to the heavy solid symbol-less mean curve $\bar{\beta}$. The central curve corresponds to a simulation where the natural droplet distribution is modelled by a monodisperse distribution with all droplets having the mass median diameter of the natural distribution. Such a simulation results in an overestimate of the value of β_0 (70.3% vs. 67.9%), and a substantial error in predicting the limits of the $\bar{\beta}$ curve (ℓ =−0.1368 *vs.* −0.2342 for the lower surface, and ℓ =0.0196 *vs.* 0.0367 for the upper surface. Further, if we compare the accretion outlines for these two simulations (the solid curve in Fig. 22 corresponds to the $\bar{\beta}$ curve), we see that the more serious departure from the shape of the mean curve occurs on the upper surface. In that region the predicted airfoil shape following the accretion has a prominent cusp whereas the mean curve joins the original airfoil surface smoothly. Case 7 combines two equally weighted size categories using a constant length filter F=0.10. Case 8 uses a variable length filter. While the variable length filter results in a slightly greater over-estimate of the cross-sectional accretion area, it improves considerably on the value of β_0 and ℓ_0. If we compare the $\bar{\beta}$ curves for Case 8 and Case 1 in detail (shown as solid and dashed symbol-less curves in Fig. 23), we see that Case 8 provides a very good approximation indeed to the composite β curve of the most accurate simulation, and at 37% of the cost of the high accuracy case. Cases 9 and 10 are similar to line 3 of Case 1, where the natural droplet distribution is modelled by a monodisperse distribution at the MMD. However, Case 9 employs a constant length filter F=0.20, while Case 10 uses a variable length filter of the same maximum length, to smooth the β curves. These β curves are displayed in Figs. 24 and 25 respectively. The $\overline{\beta}$ curve of Fig. 24 poorly estimates the height and location of the natural curve's peak – a situation similar to that of Case 7. The variable length filter applied in Fig. 25 improves the simulation considerably, however, with a smaller difference between the heavy solid and dashed $\overline{\beta}$ curves. The remarkable result is that the smoothed curve results in a good approximation to the $\overline{\beta}$ curve of Case 1 (dashed line), but at only 12% of the cost. Small adjustments in the manner in which the smoothing is applied might result in an even better fit. The corresponding accretion profiles are shown in Fig. 26. The cusp on the airfoil surface apparent just above the nose in Fig. 22 has been removed by applying the smoothing. Further, the lower accretion surfaces coincide more accurately as well in Fig. 26. The final two cases, numbers 11 and 12, represent the results of simulations made using the Kennedy & Marsden vorticity density method to compute the flowfield. We see that, particularly when the finer tolerance of Case 12 is employed, the accuracy of the simulation is comparable to that of Case 10. However, we also note that this pair of simulations confirms the results of our earlier experiments regarding the costs of this method compared with the analytical method. The vorticity density method costs almost 12 times as much for comparable accuracy.

3.3 Conclusions on the choice of parameters for further simulations.

Section 3.2 has dealt with a series of experiments to determine the best settings for the tolerances and options which the user must input to employ the program. Trials to determine the accuracy of the flowfield calculations by the vorticity density method, and the reliability of local collision efficiency calculations through droplet trajectory pair simulations, have led to a preliminary set of input parameters. These have been used to calculate the collision efficiency curve for an airfoil under conditions similar to those which have been measured during helicopter icing trials (see

Chapter 1). Such simulations, combined with others employing multi-droplet size categories have led to the following conclusion. Even simulations using only a monodisperse droplet size distribution can lead to an accretion profile very similar to a profile generated by the most natural simulation available within this program if the β curve is smoothed with a suitable variable length filter. Without smoothing, the profile is comparable only if two or more droplet size categories are used to model the natural droplet size distribution. Further comments on the desirability of smoothing the β curve will be made in Chapter 5.

The parameters effecting the best compromise between computing cost and the accuracy of the accretion profile area and shape are: NEF=11, NEB=10, NIF=6, EQN=2, CDS=1, EPS approximately equal to 1×10^{-4} depending upon the droplet size, CEDEL=1.0, X0=−5.0, and DDISTN=1 provided that FILTER ≠ 0.0

4. TESTING THE CODE FOR COLLISION EFFICIENCY ACCURACY

4.1 Introduction

The previous chapter has described the adjustment of user input options · tolerances so as to find a good compromise between computing cost and In order to prepare for applications, we now require a more general ve the program's accuracy. The predictive capability of the program enco the collision efficiency, and the accretion profile. Since these s have commonly been separated in past results, we shall first attempt pare program's predictions regarding the total collision efficiency, the impingement, the droplet velocity at impact, maximum local collisior and the overall shape of the β curve, with the predictions of other ailable In the next chapter, these intercomparisons will be carried one step verall accretion shape, accretion area, and indirectly, accretion density.

Within this chapter, comparisons of results will be made firs inders, because they were the first substrates used in early studies of ice accretion. We will then progress through Joukowski airfoils, NACA four digit airfoils, and finally on to a more modern airfoil. Most of the papers with which we may compare display theoretical results. One or two also outline experimental ones.

In order to ensure repeatability of the results presented herein, all simulations in the remainder of the dissertation shall be identifed by a unique Case number, continuing on from those of Section 3.2.4, with a listing of the input options and parameters for each case given in Appendix H. Additionally, for each case where an input parameter is changed so as to affect the conditions defining the simulation, up to four additional non-dimensional parameters will be given. These parameters are:

1. The non-dimensional free-stream Reynolds number. This is the value that Re_d would take on if the droplet were moving through the air with a relative velocity of U_∞, that is:

$$Re_\infty = 2R_d U_\infty \rho_a/\mu \tag{4.1}$$

In this equation R_d is the droplet radius, U_∞ is the free stream velocity, ρ_a is the air density, and μ is the dynamic air viscosity.

2. The ND inertia parameter. This was first defined by Langmuir & Blodgett (1946) for a cylinder. The definition used herein is:

$$K = \frac{2\rho_d R_d^2 U_\infty}{9\mu C} \tag{4.2}$$

It conforms to current common practice, and differs from Langmuir and Blodgett's definition in that the characteristic length C now refers to the airfoil chord. Thus in the case of the cylinder, C is the diameter, whereas Langmuir and Blodgett used the radius. In (4.2) ρ_d is the density of water.

3. The ND impingement parameter. This was also first defined by Langmuir and Blodgett, and again differs from current use by the same convention regarding C. Our definition shall be:

$$\phi = \frac{18\ \rho_a^2 U_\infty C}{\mu \rho_d} \tag{4.3}$$

We may note that the relation between Re_∞, K and ϕ is:

$$\phi = Re_\infty^2 / K \tag{4.4}$$

4. The ND accretion parameter. This was defined in (2.111) and (2.112).

Any two of the first three parameters are sufficient to uniquely define the conditions which should produce the same collision efficiency curve. The addition of the accretion parameter allows us to define the combination of conditions leading to the same accretion profile.

4.2 The collision efficiency of a cylinder.

Amongst the first to perform an in-depth analysis of the phenomenon of icing on cylinders were Langmuir & Blodgett (1946). Their calculations of supercooled water droplet trajectories were made on a differential analyzer. Following a relatively large number of such simulations, they prepared a series of tables and charts reducing the large numbers of cases through the use of the non-dimensional parameters Re_∞, K and ϕ. Their predictions included the total collision efficiency E_m, the local collision efficiency at the stagnation line (of a cylinder) β_o, and the maximum angle of impingement, θ_m which corresponds to our grazing trajectory length ℓ_G. In some instances, they also predicted the components of the velocity at impact, u^* and v^*.

Using the same techniques, Brun & Mergler (1953) repeated a number of experiments performed by Langmuir and Blodgett in the course of evaluating the multi-cylinder method for determining cloud properties, such as MMD, LWC, and the shape of the droplet size distribution. Table 1 of their paper presents a comparison between their results and those of Langmuir and Blodgett. From this table we have chosen three sample pairs with which to make comparisons. The values in two of the three display substantial disagreement in the value of E_m. The results of the third pair agree much more closely with each other.

In the six cases presented in Table 8, the even numbered ones incorporate the history term in the droplet equations of motion. The odd numbered ones do not. An inspection of this table reveals some interesting results. In the first set, the total collision efficiency values of Cases B and 13 agree well, as do the values of ℓ_G, u^* and v^*. In both cases the equations of droplet motion exclude the history term. There is less agreement with Case A.

In the second set, the greatest agreement is reached between Cases A and 15. This time, our predictions vary considerably from those of Brun and Mergler. As would be expected, the differences between 13 and 14 are greater than those between 15 and 16, because in the latter cases the trajectories are less curved, there is less acceleration, and thus the history term, which is a function of the strength of past accelerations, is smaller. The trajectories of Case 15 are shown in Fig. 27, with the corresponding β curve displayed in Fig. 28.

In the third set, where the trajectories are once again more curved (see Fig. 29), the agreement seems to be best between Cases B and 17. Thus, in the three sets examined, the present results agree best with those of Brun and Mergler twice, and with Langmuir and Blodgett once. More important, our results compare very well with at least one case of each set. Further, although the formulation excluding the history term generally provides a better comparison, as would be expected since the other papers did not include the history term in their calculations, the inclusion of the history term does not affect the values used for comparison by an amount much greater than the disagreement between the results of the previous papers.

The study of rime ice formation on stationary objects, such as power lines, has led to the use of modern, sophisticated techniques to solve for the collision efficiency. McComber & Touzot (1981), for example, have employed a finite-element grid, with a restructuring of the droplet equations of motion using an Eulerian reference frame, to solve for the velocity field of the droplets. This contrasts with the method of the present study, where we solve for the air velocity field first, and then calculate individual Lagrangian droplet trajectories. The droplet velocity field calculated for one size of droplet (usually a size for which K is small), then allows McComber and Touzot to iterate to the droplet velocity field for the next value of K and thus D_d. The local collision efficiency is determined from the velocity field, and is integrated numerically to yield the total collision efficiency.

The pressure and temperature values chosen for comparative simulations with McComber and Touzot were once again those used by Langmuir from experiments on Mt. Washington, New Hampshire. Table 9 details the results of series of simulations. As before, definitions of K and ϕ vary, and thus values using both defintions are given.

For the cases shown in Table 9 (Cases 19 through 24), the trajectory equation of motion was varied, as was the drag coefficient formulation. The reason for doing this was to study the effect such changes produce in the parameters used for comparison. As before, the inclusion of the history term has the most significant effect on the values of E_m, ℓ_G and β_o when the accelerations are strongest (E_m smallest). This is seen by comparing Cases 19 and 21 with Cases 22 and 24. Similarly, when the drag coefficient formulation is changed (Cases 20 and 23), the greatest effect upon the results occurs when the accelerations are strongest. When the history term is excluded (which makes a fairer comparison with the methods of the other two papers) we see that our results most closely match those of Langmuir and Blodgett for both sets of Re_∞ and K parameters. Once again the differences in the predictions are small, leading to an increased confidence in the accuracy of our results. The fact that McComber and Touzot's results deviate somewhat from the others suggests that the finite element grid which they employed may have been too coarse.

4.3 The collision efficiency of a 36.5 percent thick Joukowski airfoil.

Brun and Voyt (1957) studied the impingement of droplets upon a Joukowski airfoil in order to determine if such an airfoil (or set of airfoils) might be better suited for estimating the cloud LWC and MMD than the rotating cylinder method. Their method of solution for the droplet trajectories was the same as that used by Brun and Mergler (1953), that is the mechanical analog. We shall make comparisons with this paper because it allows us to take one step up the ladder of airfoil flowfield complexity. The analytical solution for the flowfield about this type of airfoil is known, just as it is for the cylinder, and thus it allows us to evaluate the ultimate effect of the accuracy with which we calculate the flowfield by the method of Kennedy and Marsden. Results for several simulations are found in Table 10. Comparisons of E_m and ℓ_G show that Case 25 approximates the results of Brun and Voyt to within the 1% error limit estimated by Brun and Voyt to be appropriate for their results. In this situation, inclusion of the history term has only a small effect upon E_m and β. If we compare the β curves for Case 25 (trajectories displayed in Fig. 30) and Case D, we can see (Fig. 31) that the agreement is excellent except perhaps right at the airfoil nose, where a small discrepancy exists between the solid line (present results) and the dashed line (results of Brun and Voyt).

4.4 The collision efficiency of uncambered four-digit NACA airfoils.

As explained in Chapter 2, no analytical solution exists for the flowfield about four and five-digit NACA airfoils. Thus we have chosen to compare the results of the present model with those of Werner (1973) and Bragg *et al.* (1981) to determine the program's accuracy in modelling these more-difficult-to-model airfoils.

Werner (1973) has attacked the problem of determining the flowfield about an arbitrary shaped airfoil in essentially the same way as we have in the present paper

He too uses a vorticity substitution technique for generating the flowfield at a distance from the airfoil. However, whereas we continue to use the flowfield provided by this method to calculate the droplet trajectories up to their point of airfoil collision, he uses another (unknown) technique near the airfoil surface, presumably because he feels his vorticity substitution method is not sufficiently accurate near the surface. We have shown that errors in the potential flow velocity quite near the airfoil surface remain small when the Kennedy and Marsden technique is used to generate the flowfield. Werner goes on to integrate the simplified system of equations describing the droplet's acceleration, the system designated here by EQN=0. He carries the process one step further by also incorporating a limited set of thermodynamic processes, which he uses to predict the initial freezing rate based upon the initial collision efficiency. The results of a comparison between the two programs is given in Table 11. From this table we may note that the best agreement is between Cases E and 27, where the equations of droplet motion are most similar. There is a greater discrepancy in E_m here than has been noted in earlier comparisons, although the values of β_0 agree reasonably well. However, we see that there is a considerable discrepancy in the limits of impingement. To study this problem more closely, we may turn our attention to a comparison of the β curves for the two cases. Fig. 32 displays the trajectories used in calculating our β curve, with the curve itself shown as a solid line in Fig. 33. This latter figure also shows Werner's result as a dashed line. We see that even though the peaks of the curves are aligned, there appears to be a general shift of the Werner curve to the right. This could occur if the angle of attack were in error. It may also be due to Werner beginning his trajectory calculations insufficiently far upstream from the airfoil.

Because of the rather large disagreement between these two curves, we decided to make a further comparison with Bragg *et al.* (1981) using a similar airfoil, the NACA 0015. Bragg *et al.* have written a program to accomplish many of the aims described in Chapter 1. They have employed a completely different approach toward the calculation of the airflow about arbitrarily shaped airfoils however. Whereas we employ a vorticity density technique to create the appropriate airflow subject to the boundary conditions of Kennedy and Marsden, the technique of Bragg *et al.* is based upon the Theodorsen & Garrick (1932) method of conformal transformation. This involves the fitting of a series of conformally transformed circles to provide a composite airfoil shape that matches the desired shape sufficiently well. The matching of airfoil shapes is accomplished by Fourier components. These same components may then be used to find the composite flowfield which matches the composite airfoil profile.

Two cases have been chosen for comparison. They are outlined in Table 12. The history term is excluded in order to make a fair comparison with the predictions of Bragg *et al*. Further, the effects of this term have been determined in the experiments above. The agreement between the results appears to be quite good, especially given that the results are obtained by substantially different methods. Inspection of Figs. 34 and 35, where a comparison is made between the collision efficiency curves of Bragg (1981) and the values predicted in this paper, shows that the curves match quite well. The greater extent of impingement predicted by the present results may be a function of the considerable care that has been taken to find these values directly, rather than by extrapolation. In both figures Bragg's β curve has a slightly lower value of ℓ_0, that is the peak is shifted to the left. This is the opposite shift to that of Fig. 33. The techniques used by Werner much more closely resemble those used here, than do those of Bragg. Since the agreement between Bragg and the present paper is much better than with Werner, we may suspect that Werner's results may be in error.

4.5 Comparison with experimental collision efficiency curves for several airfoil types.

All of the results with which we have been making comparisons up to this point have been based upon theoretical calculations of β curves and impingement characteristics. We now subject the program to a series of tests which will allow us to determine its accuracy as compared to experimental results obtained in a wind tunnel using a distribution of impinging droplet sizes. These results are found in a paper by Gelder *et al*. (1956). The local collision efficiency was determined by covering an airfoil surface with blotter paper, and then injecting a water soluble dye into the water used to produce the droplet spray. A colorimetric analysis of the blotter paper revealed the rate at which dye reached the airfoil surface and thus gave the value of β. In order to increase confidence in the conclusions which could be reached from this intercomparison, three sets of cases were run. In the first three cases the history term was included. In Case 34, the history term was dropped in order to study its significance upon the accuracy of the results.

4.5.1 The collision efficiency of 15% thick Joukowski airfoil at a zero attack angle.

The first comparison set is displayed in Table 13. The droplet size distribution of the tunnel spray resembled a Langmuir "D" distribution with a mass median diameter of 18.6 µm. All four cases produced from the program (31 through 34) show a value of $\overline{\beta}_o$ which is slightly higher than the measured value. The overall collision efficiencies $\overline{E}_m$ tend to be slightly lower than measured, and because we have not modelled the largest droplets in the spectrum, the impingement limits $\overline{\varepsilon}_G$ are significantly underestimated. Once again the monodisperse droplet size distribution of Case 33 provides good estimates of $\overline{\beta}_o$ and $\overline{E}_m$ as compared to those of Case G. The effect of dropping the history term (Case 34) is to further reduce both E_m and β_o. This is consistent with previous comparisons with and without the history term. We see from this comparison that including the history term does indeed result in a better simulation of the experiment results. We now turn our attention to Figs. 36 through 38 which display the results of Cases 31 through 33 respectively as solid lines, and experimental results as dashed lines. There has been no smoothing applied to the curves making up Case 31. For discussion of the need for smoothing, return to Section 3.2.4. The β curves for 25.5 and 13.2 µm diameter droplets are combined to give a mean curve (shown as a heavy solid line without symbols). We see that where the inner β curve (for the smaller droplets) terminates, a kink results in the $\overline{\beta}$ curve. If this case is repeated, but with the application of a variable length filter of maximum length 0.2 times the length of the total β curve, the result is the $\overline{\beta}$ curve of Fig. 37. The kink has been smoothed out and the limits of impingement have been extended. We see that the variable length filtering has two significant and desirable features:

1. cusps caused by simulating the natural droplet distribution by a small number of

monodisperse size categories are removed;

2. the limits of impingement are extended, and thus the effects of the larger droplets which are not explicitly modelled may be crudely accounted for.

Fig. 38 shows the filtered and unfiltered β curves for a simulation using a single droplet size of 18.6 µm. Once again the agreement with the experimental results is very good, although the effects of the largest droplets are not modelled correctly. The overall fit is not quite as good as that for Case 32, but if this level of error is acceptable, the results of Case 33 can be produced at approximately half the cost of Case 32.

4.5.2 The collision efficiency of a 15% thick Joukowski airfoil at a 4° angle of attack.

Table 14 displays the results of the second comparison set. The experimental results of Gelder *et al.* are designated as Case G. We have also inserted two other theoretical results in order to increase the pool of results available for comparison. Kloner (1970) (designated as Case H) has produced a numerical model which solves for the potential flow about an arbitrarily shaped airfoil and then calculates the droplet trajectories. His model is very similar to that of Werner (1973), which was described in Section 4.4. Kloner also cites the results of Guibert *et al.* (1949) (designated as Case I) which were obtained by the methods of Langmuir & Blodgett (1946), described in Section 4.2. In both cases, monodisperse droplet size distributions are used. The total collision efficiency predicted in Cases G, H and I is 39.2, 37, and 39% respectively. Gelder *et al.* measured a maximum local collision efficiency β_o, of 70%. If we turn to the predictions of the present program for ALPHA=4.0, we see E_m varying from 38.6% to 39.5% and β_o varying from 68.2% to 74.7% depending upon the number of size categories used. These results are displayed in greater detail in Figs. 39 through 41. When we compare the unsmoothed mean β curve (Fig. 39) or its smoothed counterpart (Fig. 40) to the experimental β curve (displayed as a dashed line) we immediately notice that our results seem to be shifted somewhat to the left of the experimental curves. Excepting this anomaly, all of our curves appear to match quite well, especially the smoothed versions. When only a single droplet size category is used, as in Case 37 (Fig. 41), the same misalignment is evident. Once again, as in Section 4.5.1, the smoothed monodisperse β curve approximates the experimental curve almost as well as the $\bar{\beta}$ curve from Case 36, but at a considerably reduced cost.

In order to investigate a possible explanation for the misalignment between our results and the experimental ones, we have run another simulation identical to Case 36 except that the angle of attack has been changed to 3°. The result (Case 38, Fig. 42) is a much better match between the two β curves. We may speculate that a one degree error in the experimental results might be possible, as this seems to be a relatively small error in the alignment of a wing section relative to the flow in the wind tunnel. These results point out a significant factor in these intercomparisons. The present program is able to predict the changes that will occur in the β curve as a result of a small change in the conditions defining the case. Thus if comparisons are to be fruitful, the experiments must be done with great care.

4.5.3 The collision efficiency of a NACA 65-212 airfoil at a 4° angle of attack.

This set of simulations is outlined in Table 15. The experimental results with which we are comparing are once again those of Gelder *et al.* (1956) designated as Case G. Two case sets are incorporated in the table. The first set is the more difficult to simulate because the collision efficiency is very low. This means that the trajectories are much more curved and thus that more computing effort is required to maintain sufficient accuracy during such calculations. Because of the computing effort required, monodisperse droplet size distributions are used in both of our cases. The values of E_m and β_o for Cases G and 39 are very similar, indicating that the program is performing well under these conditions. The impingement limits are in poorer agreement because the largest droplets in the spray droplet size distribution are not modelled in Case 38. If we shift our attention to the second set, we discover that there is a significant discrepancy between the experimental results (Case G) and those predicted by this program (Case 40). It is interesting to note that the theoretical predictions of Bragg *et al.* (Case F) are in better agreement with our results than are the experimental ones. This is even more evident when we look at the three β curves of Fig. 43. The peaks of the three curves align very well, contrary to the situation in the previous section. The greatest disagreement between Cases G and 40 appears to be along the lower surface. We see that the short-dash curve of Bragg *et al.* is also lower than the one predicted by our program, although our program predicts a greater extent of impingement along the lower surface than does that of Bragg *et al.* The considerable discrepancy between the two theoretical curves and the experimental curve remains unexplained. Figure 44 shows the impinging droplet trajectories used in determining the β curve of Case 38. It also displays the slender nature of this particular airfoil. It is the small radius of curvature of the airfoil nose which leads to the sharp peak in the collision efficiency curves.

4.6 The collision efficiency of a modern light aircraft wing.

This set of simulations is included to show that the methods presented may be applied to a variety of two-dimensional airfoil profiles under conditions appropriate for general aviation wings as well as helicopter main rotor blades. The case with which shall compare is described by Bragg *et al.* (1981). They used conditions based on experimental results obtained using a full-scale general aviation wing section in the NASA Lewis icing wind tunnel. A comparison between the results of our programs and the experimental impingement results for this Hicks modified NACA 64-215 airfoil at a 0.7° angle of attack is given in Table 16. Bragg *et al.* do not provide values of E_m for their experimental and theoretical results. However, they do provide estimates of the amount of ice accretion which forms for a given value of the non-dimensional accretion parameter. From these, E_m may be inferred as 5.3 and 6.2% respectively. The value obtained by the present program is 8.2%. The set of trajectories which were used to obtain this result are shown in Fig. 45. The airfoil shape was derived from data provided by Bragg (1981) and it was verified against the profile coordinates provided in the original paper (Szelazek and Hicks, 1979). A plot (provided by Bragg, 1981) of the experimental and theoretical accretion shape on the airfoil nose displays an airfoil profile in the nose region which departs substantially from the one provided for use in this dissertation. This discrepancy has not yet been resolved. The collision efficiency curves for the two theoretical results are displayed in Fig. 46 where the solid line represents the present results, and the dashed line those of Bragg *et al*.

4.7 A summary of the collision efficiency simulations.

In this chapter a series of intercomparisons has been made using a variety of airfoils to explore the degree of agreement between the collision efficiency predictions of the present program, and the impingement characteristics of other theoretical and experimental results. We have begun with a simple icing shape, the cylinder, and gradually moved up to recent airfoil designs, such as the Hicks modified NACA 64-215 airfoil. The agreement with previous results has been very good in many cases, with at least one of our simulations agreeing well with either a previous theoretical or experimental result for each airfoil tested, except perhaps the final one. In this last case, where a modern general aviation airfoil was used, the error may be caused by a discrepancy in the airfoil profile between our results and those with which we are comparing. However even in this case, the general features of the β curve are reproduced reasonably well. These results concerning β give us confidence to carry out still further comparisons, this time of accretion profiles, in the next chapter.

5. THE PREDICTION OF ICE ACCRETION AND OTHER APPLICATIONS.

5.1 Introduction

Chapter 4 has presented a series of intercomparisons between results of the model described in Chapter 2, and theoretical and experiemental results for various "airfoils". These comparisons were limited to several characteristics of the droplet impingement, such as the total collision efficiency E_m and the slope of the local collision efficiency (or β) curve. This restriction was intentional. It allowed us to make comparisons with those types of results for which numerous examples exist. Our desire to compare the present model's predictive capabilities regarding the area and shape of accreted ice profiles with other theoretical and experimental simulations is hampered by a distinct lack of carefully controlled results with which to compare. This fact will be a subject for discussion in Chapter 6.

Within this chapter, we shall make our first comparisons for accretion on a cylinder. This substrate has played a pivotal role in icing studies, and is one of the few for which previous theoretical predictions of accretion profiles exist. The next airfoil to be studied will be the NACA 0015 at 0° and 8° angles of attack. Following this, the NACA 0012 airfoil forming the main rotor of a Sikorsky S-55 will be examined.

Sections 5.5 and 5.6 will indicate some of the applications for the program described and tested within this dissertation. In Section 5.5, the predicted accretions on a NACA 0012 and a NPL 9615 airfoil are compared to see what effect a change in airfoil shape has upon its icing characteristics. This type of comparison is also applied to the Joukowski 0015 and NACA 0015 airfoils. Finally, in Section 5.6, an experiment is carried out to test a scaling theory, by comparing our collision efficiency results for Joukowski ~~0015 airfoils~~ at full and one-quarter scale.

5.2 Accretion on a cylinder.

A set of cylinder icing computer model simulations and wind tunnel experiments have been carried out at the National Research Council's facilities in Ottawa, Canada (Stallabrass & Lozowski, 1978; Lozowski *et al.*, 1979). The numerical simulations incorporate the collision efficiency results of Langmuir & Blodgett (1946) and a sector-by-sector calculation of the thermodynamics of the accretion. These calculations take into account the impingement of supercooled water droplets and ice crystals. When the deposit temperature is at freezing, the unfrozen deposit is allowed to run back along the cylinder surface, thereby altering the accretion profile. This model does not incorporate the time dependence of the accretion process. This will not cause difficulty in making comparisons, however, because the present model can be run in a single step fashion as well.

The cases with which we wish to make comparisons will be limited to those from the sets described by Lozowski *et al.* which occur in a cloud composed entirely of supercooled water droplets at temperatures at or below −15°C. These restrictions are necessary because the present program has not been designed to accommodate the calculations of ice crystal trajectories, and because it is restricted by design to simulating riming. Thus the accreting droplets must freeze upon impact, requiring a relatively low air temperature.

5.2.1 Accretion with a constant density.

The first seven cases presented in Table 17 are calculations of the accretion on a cylinder with an assumed constant density of 917 kg m^{-3}, as identified by DENSE=0. Cases 42 and 43 were run to investigate the importance of incorporating the airfoil surface curvature when calculating the thickness of the accretion. This process was discussed in Section 2.4.6.2. The non-dimensional cross-sectional area of the accreted ice is given by

$$Z = h_o \omega \bar{E}_m \qquad (5.1)$$

where h_o is the ND maximum airfoil thickness, ω is the ND accretion parameter, and $\bar{E}_m$ is the total collision efficiency. When we compare $\bar{E}_m$ given by (5.1) with the value obtained from (2.90) for Cases 42 and 43, we find that while the total collision efficiencies of Case 42 (ignoring airfoil curvature) show a relative disagreement of 8.3%, those of Case 43 are virtually identical (a difference of −0.1%). We may conclude from this experiment that due consideration of surface curvature is important for obtaining an accurate estimate of the accretion profile shape and area. The profiles for these two cases are displayed in Fig. 47. Also indicated is the experimentally derived profile of Lozowski *et al.* (1979).

Maintaining the same conditions as for Case 43, Case 44 incorporates two categories of droplet size (27.0 and 14.4 μm diameter) as compared to the monodisperse distribution of Case 43 (20 μm diameter). The resulting accretion area is identical, although the values of $\bar{\beta}_0$ and $\bar{E}_m$ have decreased slightly. When we compare the accretion profile to the profiles observed and predicted by Lozowski *et al.* (see Fig. 48), we see that the agreement with the experimental results is somewhat better for Case 44 than for Case 43, particularly in the region where the accretion is thinning rapidly, that is near ℓ_G. We also may note that the values of β_0 are virtually identical for the two theoretical results (Cases 44 and H), although the program of Lozowski *et al.* underpredicts the accretion thickness virtually everywhere as compared to the experimentally observed thickness. We may speculate that the agreement at the "nose" is due to their use of Langmuir & Blodgetts' (1946) values for

We have shown that the present program generates results that agree well with those of Langmuir & Blodgett. On the other hand, the formulation used by Lozowski *et al.* to specify the β curve does not conserve mass; that is they do not require the area under the β curve to equal the total collision efficiency as we do.

The accretion areas for the experimental and theoretical profiles of Lozowski *et al.* (1979) are 0.081 and 0.065 respectively as compared to 0.0864 and 0.0841 for Cases 43 and 44 respectively. The areas for the results of Lozowski *et al.* were determined by measuring (with a planimeter) their profiles drawn at the same scale as the original (full size) version of Fig. 47. Thus the relative difference in areas between Case G (experimental result) and Case 44 is only 4%. The remaining cases in Table 17 deal with simulations made with a LWC of 0.8 g m^{-3}. The collision efficiency curves for the 27.0 μm and 14.4 μm diameter droplets of Cases 44, 45 and 46 are shown as solid lines with symbols in Fig. 49. Also shown is the filtered β curve of Case 46. The effect of this filtering may be studied by comparing Cases 45 and 46 in Table 17 and Fig. 50. We note that while the total accreted area remains the same, the peak value of β is reduced slightly by filtering. Of greater significance, however, is the extension of the limit of impingement by 19% when filtering is applied. This is seen clearly in Fig. 50 where the dashed curve corresponds to the filtered case. The filtering also removes the cusp in the accretion profile of Case 45, caused by kinks in the unsmoothed $\bar{\beta}$ curve first noted in Chapter 3. In Fig. 51 we compare our results for Case 46 with the theoretical (dashed line) and experimental (solid symbol-less line) results of Lozowski *et al*. Once again, as in Fig. 48, the theoretical results match well at the "nose". In this figure, the departure from the experimental results of the model results of Lozowski *et al*. is greater than it was previously. Our simulation underpredicts the accretion thickness observed by Lozowski *et al*. virtually everywhere, although the departure is greatest where it appears that rime feathers may have begun to form at the outer edge of the accretion. The implications of the variation of ice density in such rime feathers will be investigated in the next section.

Table 17 compares the accretion areas for Cases G, H, 45 and 46. Cases H, 45 and 46 have accreted areas which are 40%, 16% and 16% less than the area of the experimentally determined accretion of Case G.

Cases 47 and 48 are the same as Cases 45 and 46 with regard to filtering, although in the new pair the natural droplet distribution is modelled by a monodisperse distribution of droplets having the mass median diameter of the natural distribution. The unfiltered and filtered collision efficiency curves of these two cases are shown in Fig. 52 as solid lines with and without symbols respectively. The area of the accretion is shown in Table 17 to remain unchanged by the filtering, though the value of $\bar{\beta}_0$ is reduced, and that of $\bar{\ell}_G$ increased. The two corresponding accretion profiles are given in Fig. 53, with the filtered version appearing as a dashed line.

The best results from Cases 45 through 48 (that is, those with the kinks filtered out) are displayed in Fig. 54. The difference between the profiles represented by a solid line with symbols (Case 48) and the long dashed line (Case 46) is small, indicating that by filtering the β curve, monodisperse droplet distribution simulations can provide comparable results to two-droplet simulations at approximately 50% of the computing cost.

The accreted area for Case 48 is in somewhat better agreement with the experimental results of Lozowski *et al.* (Case G) than is Case 46, but the improvement is small (14% *vs.* 16% error).

5.2.2 Varying the density of the accretion on a cylinder.

Section 5.2.1 drew attention to the discrepancy between the present results and the experimental observations of Lozowski *et al.* (1979) regarding the area and the shape of the accretion profile. This disagreement is most pronounced where it appears that rime feathers have formed.

The stochastic fluctuations in density which are an integral part of rime formation were discussed in Section 2.4.6. These are not explicitly modelled by the present program. On the other hand, variations of density caused by the degree of droplet distortion upon impact may be modelled inasmuch as these variations are a function of the impacting droplet's velocity, diameter and the temperature of the droplet prior to impact. A formula for varying the accretion density based upon these three variables was presented in Section 2.4.6.3. We have allowed for two interpretations of the manner in which this formula is to be applied. The first (denoted by DENSE = 2) uses the total droplet impact velocity in (2.123), whereas the second (DENSE = 1) uses only the component of the impact velocity normal to the airfoil surface at the point of impact. The results of three simulations with variable density are presented as Cases 49, 50 and 51 in Table 17.

Beginning with Case 49 (DENSE = 2), we see that an improvement in the total accreted area is made over previous cases (a difference of 6% *vs.* 14% for Case 47) but that this is at the expense of agreement in the accretion thickness at the "nose". Since the $\overline{\beta}$ curve has not changed from Case 47, we also note that the limits of impingement are identical in Cases 47 and 49. The accretion profile for this case is compared to the profiles of Lozowski *et al.* (1979) in Fig. 55. The conclusions reached from the table are verified: the areas of our results and the experimental ones are more similar than before, but the previous cases (Cases 46 and 48) seem to fit the experimental profile better over most of the layer's extent.

When the normal component of the velocity is used (DENSE = 1) the accretion thickness (Case 50) at the "nose" remains the same as for Case 49, but there is a substantial increase in the thickness as we approach the limits of impingement (see Fig. 56). In this region the component of the velocity perpendicular to the airfoil surface decreases rapidly, thereby producing a rapid decrease in the accretion density according to (2.123). The total area of the accretion increases considerably as well (from 86% of the experimental area for Case 48 to 136% for Case 50).

If conditions remain the same but the natural droplet distribution is modelled by two size categories instead of one, and if a variable length filter (F = 0.2) is applied, we find that the accreted area decreases somewhat from the previous case (from 136% for Case 50 to 128% of the experimental area for Case 51). Further, the profile shapes agree to a sightly greater extent, but generally agreement is still not good (see Fig. 57). This points out the need for better formulae to be used in estimating accretion density. Such formulae should be based on empirical studies of the microscopic processes of rime accretion. Judging from the poor performance of the present results, variable density simulations will not be pursued further.

There is also a need for better understanding of the growth angle of rime feathers. Lozowski (personal communication) has simulated the growth of rime feathers numerically (see Fig. 15). They display a total growth angle of about 35°. It is interesting to note that the angle between the edge of what appears to be a rime feather in Fig. 57 and the edge of the predicted accretion is approximately 15°.

5.2.3 Multi-layer (time-dependent) accretions on a cylinder.

All simulations carried out to this point have employed the airflow about the original airfoil profile to determine the collision efficiency and thus the accretion profile. We shall now move to time-dependent modelling, where the airflow is recalculated to account for the change in airfoil shape after each of a series of layers have been accreted. The first example of this method is Case 52 in Table 17. Here the accretion parameter ω has been reduced to one-third of its previous value, and three layers of ice have been simulated. In physical terms, this is equivalent to studying changes in the accretion after time periods that are one-third that of the original accretion period. From Table 17 we note that β_o increases slightly with time, while E_m and ℓ_G decrease. This can be seen as well in Fig. 58, which displays the filtered (solid line without symbols) β curve for layer 3 and a similar curve (dashed line) for layer 1. The accretion area decreases with time much as does the total collision efficiency. The total accreted area after three layers is 0.1677, about 84% of the experimental area, compared with 86% for the single step case (Case 48). The limits of impingement are essentially the same, the total collision efficiencies are similar (55% for Case 48 compared with an average of 53.4% for Case 52) but the effective combined value of $\bar{\beta}_o$ has increased to 80.1% from 70.9%. This combined value of $\bar{\beta}_o$ is derived from the thickness of the accretion at the nose, and thus incorporates the effect of the radius of curvature. It indicates the value $\bar{\beta}$ would need to have at the "nose" in order to achieve the same thickness with only one layer. The accretion profiles for Case 52 are shown in Fig. 59. Also shown are the experimental and theoretical profiles of Lozowski *et al.* (1979) in solid without symbols, and short dashes respectively, and the predicted profile for a single layer of accretion (Case 48) in long dashes. We note from these comparisons that the agreement between Case 48 and the experimental results is better than between Case 52 and the observed profile. Although we do not know why time-dependent modelling has resulted in poorer agreement rather than better, we suspect that an accurate formula for predicting the density of the deposit has not yet been employed. It is interesting to note that the multi-layer case does give better agreement with the angle of growth of rime feathers simulated by Lozowski.

The results of Case 52 incorporated the use of a variable length Boxcar filter (F=0.20) upon the β curve for each layer. A strong incentive for the development of such a smoothing operator is displayed as Case 53 in the next four figures. Fig. 60 shows the collision efficiency curves (solid lines with symbols) for the 27.0 and 14.4 μm diameter droplets used in the simulation of layer 1. The unsmoothed β curve lies between them. The collision efficiency curves for layers 2 and 3 are displayed in Figs. 61 and 62. Our attention is immediately drawn to the wavy nature of these curves near the limit of impingement. To study the cause for this, we must examine the accretion profiles for the situation, which are displayed in Fig. 63. The unsmoothed $\overline{\beta}$ curve for layer 1 has resulted in a slight trough and ridge in the accretion profile for this layer. The program is so sensitive to the profile shape, that when the curves are calculated for the second layer, there is an amplification of the waviness of the surface of the first layer. That is, the collision efficiency is predicted to decrease on the "windward" side of the trough, and to increase on the "windward" side of the ridge. Careful scrutiny reveals that this positive feedback process continues for the third layer as well. The net result is a "windward" shift of the trough as the number of layers increases, along with an amplification of its magnitude. Similar tendencies may be noted for the ridges. This type of feedback must be damped out if we are to successfully model multi-layer time-dependent accretion. Part of the problem lies within the interpolation scheme used for determining the shape of the β curve. The present scheme seems to amplify the 'waviness' which exists in the data points (β values). However, this case graphically displays an example of preferential riming upon small protrusions on a airfoil surface. It may well be that such protrusions play a significant role in the formation of rime feathers as well, such as that shown in the experimentally observed profile of Fig. 59.

5.3 Accretion on a NACA 0015 airfoil at 0° and 8° angle of attack.

Stallabrass and Lozowski (1978) have described a series of wind tunnel experiments which they carried out to study the icing of a section of a helicopter tail rotor. We have chosen two cases from these experiments with which to make comparisons. The conditions for these two cases are summarized in Appendix H and in Table 18.

Let us first confine out attention to accretion on a NACA 0012 airfoil at a 0° angle of attack. A single layer simulation of the accretion under these conditions is designated Case 54, which may be compared to the experimental results (Case I) in Table 18 or in Fig. 64. From this figure we see that the theoretical and experimental accretions nearly coincide in all regions except near the nose where a considerable difference exists. The area of the accretion predicted by the program is 6% less than that observed in the experiments.

In an attempt to improve upon the results of this first simulation, we have tried another, this time with a total of three layers. The non-dimensional accretion parameter ω has been reduced by a factor of three to give an equivalent total accretion period, with all other conditions remaining the same. Scrutiny of the results of Case 55 in Table 18 reveals the following facts. As the number of layers increases, the peak of the β curve retains its original value although the total collision efficiency decreases. There is a gradual increase in the limits of droplet impingement. The total accreted area decreases much as does E_m, with the final accreted area for all three layers being 7% less than the area of the observed accretion profile (as compared to 6% less for the single layer case above). These results may also be seen in Fig. 65, which shows the filtered and unfiltered β curves for the first (solid lines) and third (dashed lines) layers. The accretion profiles for this case are displayed in Fig. 66. The equivalent value of $\bar{\beta}_0$ for all three layers is 98.4% as compared to 80.4% for the single layer. Since the actual values of $\bar{\beta}_0$ do not exceed 80.4% for any of the layers of Case 55, we see that the layers increase in thickness with time because of the decreasing radius of curvature of the airfoil surface near the nose. This accounts for the higher equivalent value which is calculated by employing the ratio of accretion thicknesses at the nose between Cases 54 and 55. The result of this simulation is to alter the shape of the ice accretion, that is to make it generally more elongated than for the previous single layer simulation. The profile for Case 55 appears to agree better with the experimental profile than that for Case 54, for virtually its entire length. The inability of the single step method to take into account the changes in the radius of curvature is another of the weaknesses of the method.

We shall now turn to a set of comparisons for the same airfoil under almost the same conditions, with the exception of a change in angle of attack to 8°. The experimental results of Stallabrass and Lozowski are once again designated Case I in Table 18. When we compare Case 56 with Case I, we see that our single layer simulation overestimates the accreted area by 32%. Inspection of Fig. 67 reveals that the accretion near the nose is underpredicted, while that along the lower airfoil surface is overpredicted. Moving on to a three layer simulation (Case 57), we note that while $\bar{E}_m$ decreases consistently with time, $\bar{\ell}_G$ increases with time just as for

Case 55. The accreted area A_T decreases again along with $\bar{E}_m$. The total accreted area for the three layers increases very slightly, while the effective value of $\bar{\beta}_0$ over the three layers shows a small increase. The change in the β curves is illustrated in Fig. 68, while the accretion profile is displayed in Fig. 69. There is still lack of agreement between the profile for the triple layer case and the experimental profile. This time, however, the accretion at the nose is better simulated. The disagreement leads us to suspect that the angle of attack for the experiments may have been different than for the present simulation. There are also uncertainties caused during the measurement of the accretion profile.

5.4 Accretion on a NACA 0012 airfoil at a 5.7° angle of attack.

Stallabrass (1958) describes a series of icing experiments performed upon a Sikorsky H04S-2 helicopter in the icing spray rig of the National Research Council, Ottawa. The spray rig produces a cloud composed of supercooled water droplets which envelopes a portion of the helicopter hovering nearby. The accretion period is controlled by the time the helicopter remains within the cloud. Other conditions are clearly defined, except for the liquid water content of the cloud, and the size distribution of the droplets. Various factors contributed to the difficulty of determining the liquid water content accurately.

Stallabrass resolved this problem in determining the LWC by comparing the ice accretion thickness for a given airfoil with the accretion predicted at the stagnation line of a cylinder of radius equal to the airfoil radius of curvature at the nose. The LWC was estimated so that the two thicknesses would be identical.

The results of a numerical simulation of the icing in one experiment described by Stallabrass are given as Case 58 in Table 19. The predicted accretion area is 50% greater than the observed icing accretion area. The droplet trajectories used to calculate the collision efficiency curve upon which the accretion area is based are shown in Fig. 70. The two accreted profiles are displayed in Fig. 71. We note that the accretion at the nose is underpredicted, while the thickness on the lower and upper surfaces is highly overpredicted. This is similar to the results of Section 5.3. An attempt was made to improve the results of the comparison via a three layer time-dependent simulation. The results are found in Table 19 as Case 59. As time progressed, the values of β_o and E_m decreased, while the limits of impingement ℓ_{GU} and ℓ_{GL} generally increased. This can be seen also in Fig. 72 where the curves of the first and third layers are compared. The accretion area turned out to be the same as that predicted by the one layer simulation, although Fig. 73 reveals that the agreement in the shape has improved somewhat near the nose as well as along the upper and lower surfaces.

The disagreement between theoretical and experimental results is somewhat different here than between Cases 57 and I. In the earlier pair, the entire accretion was shifted upwards so as to imply that a different effective angle of attack might exist. Here the program overpredicts the accretion thickness on both the upper and lower airfoil surfaces. Since the droplet size distribution was not measured precisely, this effect could explain the disagreement evident in Fig. 73.

5.5 Predicting the effect upon icing of changes in airfoil shape.

A possible application of the program presented in this thesis is to study the effects of changing the airfoil profile upon the accreted ice. One pair of rotor blade profiles chosen for such a comparison is made up of the NACA 0012 airfoil and the NPL 9615 airfoil which is derived from it. The latter profile has a 6.2% longer chord which is developed by forming a drooped nose extension to the standard NACA airfoil. The primary purpose behind such a restructuring of the profile is to improve the stall characteristics of the blade when the angle of attack is great. However, it will be interesting to study what effect this change has upon the blade's icing properties.

The results of such a simulation are given in Table 19 (Case 60). All conditions were the same as for Case 58), except for the longer chord length. The trajectories used to calculate the collision efficiency curve for this airfoil are displayed in Fig. 74. The resulting filtered and unfiltered collision efficiency curves are shown in Fig. 75 as dashed lines. They are compared to the results for the NACA 0012 airfoil of Case 58 (shown as a solid line without symbols). We see that the primary difference occurs part of the way back along the lower surface, where the NPL airfoil has lower values of β. This curve also extends farther back along the length of the lower surface than does the corresponding curve for NACA 0012. Table 19 indicates that the change in E_m is relatively small; from 60.1% for Case 58 to 59.3% for Case 60. The accreted area computed in the coordinate system of Case 58 shows that the second airfoil accretes marginally less ice over the same time interval. The accretion profiles for both airfoils are shown in Fig. 76 where the NPL 9615 airfoil is shown in proper perspective relative to the NACA 0015 from which it is derived.

The slight indentation in the accretion profile near the nose for the NPL airfoil is an artifact produced by an error in the way the present version of the program calculates the accretion thickness of the highly curved surface when this surface is specified by too few points. A slightly more accurate prediction of the shape of the peak of the β curve when narrow peaks occur would help to alleviate the problem.

A second set of comparisons between two airfoils under identical conditions may be made by re-examining Cases 12 and 27. They are presented together in Table 20. Case 12 describes a Joukowski 0012 airfoil, while Case 27 is for a NACA 0012 airfoil. We see from Table 20 that there are only small differences between the values of β_o, ℓ_G and E_m. These differences may be studied in Fig. 77 where the two β curves are displayed. The difference in accretion areas is only 4%, and a plot of the accretion and airfoil profiles (Fig. 78) reveals only minor differences. This indicates that small changes in the airfoil shape will generally produce only very small changes in the characteristics of the accretion.

5.6 The scaling of airfoil models.

A problem which has plagued aeronautical engineers since the inception of manned flight has been to determine the aerodynamic characteristics of a newly designed airfoil without producing and flying a full scale prototype. One solution is to test the airfoil in a wind tunnel where near-realistic conditions are simulated. However, as aircraft have become larger, building wind tunnels capable of achieving aircraft flight speeds in test sections large enough to house aircraft prototypes has become impractical. A simple solution is to scale down the prototype, exactly reproducing the airfoil characteristics at a substantially reduced size. According to dimensional analysis, several dimensionless ratios, such as the wing Reynolds number, must remain constant when scaled experiments are run if the results are to be meaningful. A second significant factor is the maintenance of the correct Mach number, or flight speed as a fraction of the speed of sound. This number must remain constant between the model and the full scale to obtain similar effects of compressibility. Unfortunately, maintaining both a constant wing Reynolds number given by

$$Re_c = U_\infty c \rho_a / \mu \tag{5.2}$$

and a constant Mach number is impossible as the airfoil size is scaled down. Thus a compromise is required to ensure nearly identical conditions between the full and reduced scale airfoils.

Scaling theory, as it applies to aircraft, has been the subject of research of a series of investigators: Hauger *et al.* (1954), Brun (1957), Googan & Jackson (1967), and Googan & Hubbold (1968). Their results (as they apply to helicopters and aircraft in general) have been summarized by Armand *et al.* (1978). They have set down a number of conditions which must be met if the scale models are to lead to valid simulations of the full-scale conditions and results. Included in his summary are equations dealing with aerodynamic, thermodynamic, water droplet trajectory, and ice deposit similitude. Since we have chosen to treat the helicopter rotor blade as an airfoil in two dimensional flow, the aspects of similitude due to the rotary blade motion may be ignored here. Also we do not consider the thermodynamic aspects of the icing process, and we shall ignore the requirements for thermodynamic similitude, provided that we are careful to ensure that both full and scaled down versions of our simulations fall within the range of conditions where no runback can occur.

If the ratio between the model and full-scale airfoil chord lengths is

$$\hat{q} = c_M / c_F \tag{5.3}$$

the ratio of pressures is

$$P_q = P_M / P_F \tag{5.4}$$

the ratio of air temperatures (in °K) is

$$\theta_q = \theta_M/\theta_F \tag{5.5}$$

the ratio of air velocities is

$$U_{\infty q} = U_{\infty M}/U_{\infty F} \tag{5.6}$$

and the ratio of droplet radii is

$$R_q = R_{dM}/R_{dF} \tag{5.7}$$

then the equation relating all these ratios is given by Armand *et al.* (1978) as:

$$\hat{q} = \frac{R_q^{\,2-\hat{b}}\; U_{\infty q}^{\,1-\hat{b}}}{\left[P_q^{\,\hat{b}}\; \theta_q^{\,(3-5\hat{b})/2}\; (\theta_M + 117)/(\theta_F + 117)\right]^{\hat{b}-1}} \tag{5.8}$$

The value of $\hat{b}$ in (5.8) is that obtained from

$$C_D\, Re_d/24 = \hat{a}(Re_d)^{\hat{b}} \tag{5.9}$$

where this equation represents the least squares best fit to the actual droplet drag curve over the range of Reynolds numbers that the droplet experiences prior to colliding with the airfoil.

Equation (5.8) may be simplified considerably if we set some of the ratios equal to one. For example, to maintain a constant Mach number, set $U_{\infty q} = 1$ and $\theta_q = 1$. Let the model simulations occur at the same pressure as the full scale. Further, following the lead of Bragg *et al.* (1981) and conforming to the approach we have adopted in Chapter 2, let us rewrite (5.9) in the form

$$C_D = \tilde{a}(Re_d)^{\tilde{b}} \tag{5.10}$$

From these assumptions we have $P_q = 1$ and $\theta_q = 1$ with

$$\tilde{b} = \hat{b} - 1 \tag{5.11}$$

and

$$\tilde{a} = \hat{a}/24 \tag{5.12}$$

Equation (5.8) may thus be reduced to

$$R_q = \hat{q}^{\frac{1}{1-\tilde{b}}} \tag{5.13}$$

In an effort to verify the above analysis, we have run a simulation (designated Case 61) using a Joukowski 0015 airfoil at one-quarter the scale of the airfoil used in Case 32. From Table 21 we see that Case 32 used two droplet size categories with mass median diameters for the two categories of 25.5 and 13.2 μm. The detailed results of the trajectories of this case are displayed as a sample program output in Appendix I. From this output, we may note the range of Reynolds numbers that each droplet size experiences prior to grazing or colliding with the airfoil surface. Fig. 79 displays a log-log plot of the calculated drag coefficient as a function of Re_d, and also shows two straight line least-squares fits, one for each droplet size category. The values of $\tilde{b}$ for the two categories are −0.66 and −0.71 for the larger and smaller droplets respectively. When these values of $\tilde{b}$ and the values of R_{dF} are input into (5.13) and (5.7), we obtain the scaled values of the droplet diameter: 11.05 and 5.87 μm for the larger and small droplet size categories. Table 21 shows the results of the simulation using the reduced airfoil chord length and droplet diameters (Case 61). The values of $\overline{E}_m$ and $\overline{\ell}_G$ are identical to the full scale model. The relative errors in the values of $\overline{\beta}_o$ and A_T are less than 1%. The collision efficiency curves for these two cases are displayed in Fig. 80. Once again we may note the excellent agreement between the β curves. Further tests are required under other conditions to verify that (5.13) has general validity, but these results are encouraging.

This pair of simulations has provided another application of the present program. It may be used to check upon the validity of the assumptions leading to a particular version of a scaling theory by actually simulating the full-size and scaled down conditions and determing the degree of agreement between the results. The theory summarized by Armand verifies well with our simulations to the extent that we have tested the theory. A future version of the program which incorporates thermodynamic calculations, might be employed to verify the thermodynamic similitude conditions.

5.7 A summary of the accretion profile simulations.

This chapter has described a series of computer simulations of the ice accretions that would form on various airfoils under a diverse set of conditions. The agreement with the experimental observations of various researchers has been reasonably good, but certainly not as good as was experienced in comparisons with the experimental collision efficiency curves in Chapter 4. This lack of agreement could be the result of program errors or poor assumptions leading to the methods or equations employed within the program. However, the lack of experimental results with which we may compare, and the fact that most of these experiments were carried out when the droplet size distribution and liquid water content of the cloud could not be measured accurately, leads us to believe that the experimental conditions may not be sufficiently precisely defined to allow conclusive comparisons.

Two applications of the program were also presented. The first involves predicting the effects of changing airfoil shape upon the accretion shape and area. The second consists of simulating the results of varying the airfoil chord length and droplet diameters so as to obtain approximate aerodynamic and droplet trajectory similitude. The results of these simulations suggest that the theory presented by Armand (1978), to the extent that we have tested it, is correct.

6. CONCLUSIONS

6.1 Summary

In this dissertation, we have developed a numerical model for the prediction of rime or dry ice accretion on two-dimensional airfoils. The model is primarily intended for application to helicopter rotor blades, but the techniques employed are equally suitable for other 2-D airfoil shapes, such as those used on general aviation aircraft. A set of assumptions has been presented which restricts the validity of the simulations to cases where the Mach number is below about 0.5, and the viscous, three-dimensional and time-dependent features of the flow about a rotor blade are ignored.

The program incorporates the ability to model several airfoil shapes explicitly (the cylinder, the Joukowski airfoil, and several types of four- and five-digit NACA airfoils) and also any other profile whose surface can be specified by a series of (x,y) coordinates. The flow is calculated by analytical means when possible (for the cylinder and Joukowski airfoil), and by a vorticity substitution method otherwise. Since the ice accretion is caused by the impingement upon the airfoil surface of supercooled water droplets, the equations of motion for these droplets are integrated to yield the droplet trajectories. The integrations begin as the droplets move with the air several chord lengths upstream of the airfoil. The equations of motion employed incorporate all the accelerative terms (including the effects of the droplet inertia, the effects of the drag of the air upon the droplet, and the effects of the finite rate at which vorticity is shed by the droplet as it accelerates). The integrator employed is the Runge-Kutta-Fehlberg fourth-order variable time step algorithm with local truncation error estimation.

A series of colliding trajectories is calculated for a given droplet diameter under a specified set of ambient conditions. When the yo *vs.* ℓ values for these trajectories are fitted by a quintic Hermite spline, the rate of droplet impingement at any point on the airfoil surface within the grazing trajectory limits may be determined. This allows us to calculate the ice accretion thickness in the vicinity of that point. The thickness is influenced by the curvature of the underlying surface, by the density of the deposit, and by the accretion time. The latter quantity is kept small so that only a relatively thin accretion is normally considered. The accretion density may be considered to be a constant, or a deterministic function of the droplet diameter, the droplet impact speed, and the surface deposit temperature. The growth of the accreted ice is assumed to occur in a direction which is perpendicular to the underlying airfoil surface. The new airfoil surface which is calculated in this way allows us to return to the first step, that is to calculate the new flowfield about the iced airfoil, and to accrete another layer of ice in a time-dependent fashion.

An effort has been made to optimize the above procedure by varying a number of built-in and external (input) options and tolerances so as to achieve the greatest computing economy for a given level of accuracy in the accretion profile simulation. Up to five droplet size categories may be used to simulate a natural droplet size spectrum. The collision efficiency curve may be smoothed by a variable length Boxcar filter to better approximate the edge effects of a natural droplet distribution.

The results of a series of model simulations have been compared with the experimental and theoretical results of other researchers. Agreement of the collision efficiency curves with other work has been very good. Model simulations of accretion shapes observed in wind tunnel and *in-situ* experiments have shown less agreement however. Unfortunately, the difficulty of measuring experimental icing conditions accurately, and the limited number of experimental results available, preclude a complete verification of the methods employed in the model, and make it difficult to discern the precise reasons for the lack of agreement.

Two applications of the model have been presented. The first involves predicting the changes in ice accretion which will occur if modifications are made to the airfoil shape. Such modifications could be made to improve the aerodynamic properties of the airfoil, but could conceivably have a detrimental effect upon the airfoil's icing characteristics. The second application has been a limited verification of an airfoil scaling theory through the comparison of simulations of icing on full and one-fourth scale airfoils.

6.2 Conclusions

Two major sets of conclusions may be drawn from the present work. The first set deals with the effectiveness of the methods employed here as compared to those used by others in previous theoretical icing simulations. The second set is concerned with the results of the simulations, and their comparison with other experimental and theoretical results.

6.2.1 The simulation techniques.

1. The vorticity substitution method of Kennedy and Marsden (1976), which was used in the dissertation to model the potential flow about complex airfoil profiles, provided accurate results when compared with the exact analytical flowfield about the cylinder and the Joukowski airfoil.
2. The history term in the equations of droplet motion should be included if the goal of computer simulations is to achieve high accuracy. This is especially important under those conditions where the total collision efficiency is low, that is, when the droplets undergo rapid acceleration and their trajectories are highly curved.
3. The effects upon the ice accretion of a natural distribution of droplet sizes may be approximated by using either a set of droplet size categories which lead to expensive computations, or by a monodisperse distribution of droplets all having the mass median diameter of the natural distribution. If the latter method is adopted, then the filtering of the resulting collision efficiency curve by a variable length Boxcar filter improves the realism of the simulation near the limits of droplet impingement. That is, the effects of the impingement by very large droplets are approximated with only a small error in the total accreted area, and with a greatly improved correspondence to the natural accretion profile near the edge. The costs of simulating by this technique are much less than those associated with the multi-category approach.
4. The Runge-Kutta-Fehlberg algorithm has proven to be the most cost efficient of the ODE integration techniques which we have used. The nature of the calculations of the droplet trajectories implies that large changes (over several

orders of magnitude) are required in the time step size to maintain a constant local truncation error

5. Of the techniques used to interpolate the yo *vs.* ℓ curve (or alternately the β curve) the one with the greatest accuracy over the largest number of trials has been the quintic Hermite spline fitted to the yo *vs.* ℓ curve. It provides a smooth β curve when differentiated, and yet retains the important quality that the area under the curve equals the total collision efficiency E_m. It must be used with care however in cases where the slope of the β curve changes abruptly. In these cases this interpolator may create undesirable oscillations in the β curve.

6. The curvature of the underlying surface can be an important factor in calculating the thickness of a layer of accretion, especially when the radius of curvature is small. This factor should be included in all thickness calculations.

6.2.2 The comparisons with other results.

1. The methods used in developing the program appear to be based on reasonable assumptions judging from the agreement which has been achieved with previous theoretical and experimental results. In general, the agreement between the present results and others is best for the β curves. The comparisons with experimentally observed accretion profiles show greater disparity; however, even here the general appearance of the accretion is predicted reasonably well.

2. The comparisons which have been drawn between single-layer and multi-layer (time-dependent) simulations show that for the cases attempted, the β curve does not change substantially with time. However, considerable changes in the shape of the final accreted layer (in the multi-layer *vs.* the single layer cases) are the result of changes in the curvature of the accreted surface as accretion proceeds. The net result of time dependent modelling is to elongate the profile – that is, to increase the thickness of the accretion at the airfoil nose, and to decrease the thickness further back.

3. Comparisons between experimentally observed accretion profiles and the profiles predicted by the single and multi-layer approaches of this work verify that for all

cases except the cylinder, the agreement for profile shapes and cross-sectional areas is greater when the time-dependent method is used. For the cylinders, the greatest disparity between the observed and predicted accretion profiles occurs midway back along the surface where rime feathers appear to form. Since the present model cannot simulate rime feathers, this deficiency may explain the lack of agreement in this case.

4. The variable ice density formula proposed by Macklin (1962), when incorporated into the present model, did not improve the agreement with the experimentally observed results. The use, in Macklin's formula, of the total droplet collision velocity, and also the component of this velocity which is normal to the airfoil surface at the point of collision led to equally poor agreement.

5. We have verified the airfoil scaling theory summarized by Armand *et al.* (1978) over a very limited range of testing conditions. The collision efficiency curves of the one-quarter scale model match very well with those of the full-scale simulation.

6. The shape and the cross-sectional area of the ice accreted by the NPL 9615 and NACA 0012 airfoils under the same conditions are very similar. Thus if the NPL airfoil has better aerodynamic characteristics, this comparison persuades us to recommend the use of of the more advanced profile.

6.3 Recommendations

In the course of developing the ice accretion model which has been described within this dissertation, and during the comparisons which have been made with previous theoretical and experimental icing results, several recommendations have been formulated to either improve upon the present model, or to increase our confidence in the experimental results with which the model may be compared.

1. At present, the accretion thickness is calculated only at surface segment endpoints (SSE's). For most airfoils this causes no problem because we may specify the number and location of these points on the original surface of the airfoil. However, for those airfoils whose profile is specified by a set of discrete coordinates and for which only a limited number of (x,y) coordinates are

provided (TYPE = 4 or TYPE = 5), there may be a lack of points in the nose region (especially if it has a small radius of curvature). Furthermore, the points may be poorly placed, resulting in a poor interpolation of the surface by the cubic spline method. In such cases, a change should be made to the program to enable it to create intermediate SSE's by interpolation.

2. Careful scrutiny of the droplet trajectory detailed output (for an example, see Appendix I) shows that the RKF4 automatic step-size selection algorithm described in Appendix B typically encounters a situation once for each trajectory calculated, where it is unable to find a suitable step size with which to continue. This problem requires more investigation. Presently such problem areas are stepped-over and the integration continues. It may be that the tolerance which detects the problem has been set too fine, or that a minor adjustment is required in the algorithm which chooses the step-size.

3. When local collision efficiency values are calculated and the Hermite quintic spline is used to interpolate a β curve, there are occasions when oscillations occur in the curve near points which are unevenly spaced, or where the slope of the curve must change rapidly. At present, such situations are detected by the program and the quintic Hermite spline is replaced by cubic Hermite polynomical segments. Further research into spline interpolation might result in a better solution to this problem.

4. Related to the problem in 3 is the need to specify the β curve very accurately in regions where the radius of curvature of the airfoil is small. Such sharp-nosed airfoils affect the thickness of the accretion significantly when the curvature effect is incorporated into the thickness calculations. If the peak of the β curve is slightly shifted from its proper location, a significant error will result in the accretion profile. Therefore, a special effort must be made to ensure that the collision efficiency curve is particularly accurate in regions where the value of β is changing rapidly along the airfoil surface.

5. The variable-length Boxcar filter used to smooth the curve was incorporated late in the model develoment and thus the algorithm employed to effect the variation of filter length may not be optimally adjusted. This aspect requires further

investigation.

6. We have assumed in this model, that all accretion forms in a direction normal to the underlying accretion (or airfoil) surface. This assumption has greater validity for single droplet diameter thick layers, except near the limits of impingement. Variation of the growth direction from that used here to that from which the droplets have arrived might result in better agreement with experimental results. A rationale for the variation of the growth direction is required.

7. The variation of the accretion density according to the empirically derived formula of Macklin (1962) did not improve the agreement with experimental results. Further, experimental investigation of the accretion density variation is required, particularly for the cases with rime feather growth.

8. When the multi-layer (time-dependent) approach was used to model an accretion with a total thickness of over 7% of the chord length, a problem was encountered in maintaining a reasonable computing efficiency. It may be related to our lack of smoothing of the accreted airfoil profile and the resulting amplification of small perturbations on the airfoil surface, or to the creation of too many control elements. Further work should reveal the cause of this problem.

9. This dissertation has been restricted in scope to the prediction of the features of rime ice. The applicability of the model would be enhanced if the thermodynamic processes which occur during the accretion process could be incorporated. This would allow the program to handle accretion at warmer temperatures, and should lead to better agreement with observed accretion when runback of liquid water occurs on the airfoil surface. Further it would allow simulation of the effect of including heat sources within the airfoil for the purpose of thermal de-icing.

10. There is a distinct lack of experimental results with which we may make comparisons to verify the present model. Further, of the results which do exist, we know of none where the liquid water content and cloud droplet distribution were measured by state-of-the-art techniques. The methods used to display or measure the accretion profiles are also relatively crude. Improvements in these areas would greatly enhance the opportunity for refining the present model so as to improve its predictive capabilities regarding accretion profiles.

11. Only one set of experiments was performed to test the scaling theory summarized by Armand *et al.* (1978). In order to fully test this theory, other experiments should be carried out within the full range of conditions for which the theory applies.

12. The present program would require very little modification to allow a change in the angle of attack or air velocity after each accretion layer. Such a change would allow a better simulation of the cyclic varition of a helicopter rotor blade during forward flight.

13. The present model would be of greater benefit to airfoil design engineers if it incorporated an analysis of the aerodynamic effects of the accreted ice. The lift coefficient that it presently provides is based upon potential flow theory. This should be enhanced by the addition of an analysis of the airfoil drag.

TABLE 1. Parameters defining the mean line of a NACA five digit airfoil for a given mean line designation[1]

Mean line designation	Non-dimensional position of camber c_p	Parameter c_m	Parameter c_k
210	0.05	0.058	361.4
220	0.10	0.126	51.64
230	0.15	0.2025	15.957
240	0.20	0.29	6.643
250	0.25	0.391	3.320

[1]The values of c_m and c_k have been calculated to give the desired position of camber, and a design lift coefficient of 0.3.

TABLE 2. Derivation of non-dimensional quantities.

ND symbol	Code name	Meaning	Derivation from standard variables
x	XDS	Distance	X/C
y	YDS		Y/C
u_a	UAS	Velocity of air	U_a/U_∞
v_a	VAS		V_a/U_∞
u_d	VDS	Velocity of droplet	U_d/U_∞
v_d	VDS		V_d/U_∞
t	TS	Time	TU_∞/C
Δt	DTS	Time step	$\Delta TU_\infty/C$
r_d	RDS	Droplet radius	R_d/C
$\overline{g}$	GS	Gravitational acceleration	$\overline{G}C/U_\infty^2$
ν_a	NUS	Kinematic viscosity of air	$\mu\ /(\rho_a CU_\infty) = \nu$

TABLE 3. The dependence of the accuracy of the flow field calculation upon the number and location of the control element endpoints (CEE's).

NEF[1]		AS[3]	12	20	16	11	6
NEB[2]		AS[3]	28	12	15	10	14
x	y	$\bar{v}$	Percentage error				
-10	-1	0.99981	0.001	0.002	0.001	0.001	0.001
-5	-0.5	0.99945	0.0	0.001	0.0	0.001	0.001
-1	-0.1	0.99293	0.001	0.002	0.002	0.004	0.006
-0.01	0.001	0.83615	-0.209	-0.006	-0.081	-0.144	-0.538
0.005	0.015	1.56856	0.385	0.007	-1.204	-1.676	1.767
0.01	0.0185	1.64639	-2.331	0.201	-1.134	-1.467	-2.601
0.12553	-0.05328	0.97686	-0.308	-0.236	-0.277	0.504	-0.960

[1]No. of CEE's on front third of airfoil (per surface)

[2]No. of CEE's on remainder of airfoil (per surface)

[3]Analytical solution

TABLE 4 Comparing the accuracy of the local collision efficiency and impact location calculations against the relative computing cost as the number and position of CEE's and the truncation error tolerance are varied

	1	2	3	4	5	6	7	8	9
Row	EPS	NEF[1]	NEB[2]	NIF[3]	β(%)	Error (%)	ℓ	Error (%)	Relative Cost
1	1×10^{-4}	12[4]	9[4]	5[4]	35	0	0.00981	0	1.00
2	1×10^{-4}	6	14	11	46	29	0.00747	-24	7.7
3	1×10^{-4}	9	11	8	38	7	0.00886	-10	8.2
4	1×10^{-4}	11	9	6	34	-3	0.00964	-2	9.6
5	1×10^{-4}	13	7	5	31	-11	0.01065	9	9.5
6	1×10^{-4}	14	11	4	35	0	0.00961	-2	11.4
7	1×10^{-4}	17	13	3	34	-3	0.00957	-2	13.0
8	3×10^{-4}	12	9	5	33	-6	0.00990	1	8.7
9	6×10^{-5}	12	9	5	34	-3	0.00979	0	11.9
10	1×10^{-4}	12	10	5	35	0	0.00958	-2	11.0
11	1×10^{-4}	11	11	5	37	3	0.00930	-5	10.4
12	3×10^{-4}	11	11	5	35	0	0.00942	-4	9.3
13	6×10^{-4}	11	11	5	35	0	0.00927	-6	9.0
14	3×10^{-4}	12	10	5	34	-3	0.00970	-1	9.3
15	1×10^{-4}	11	10	6	35	0	0.00944	-4	10.2

[1] NEF: No. of CEE's on front third of airfoil (per surface)

[2] NEB: No. of CEE's on remainder of airfoil (per surface)

[3] NIF: No. of SSE's between adjacent CEE's on front third of airfoil

[4] Denotes analytical solution to potential flow about a Joukowski airfoil at 4.6° attack angle.

TABLE 5. Comparing the accuracy of the local collision efficiency and impact location calculations against the relative computing cost and final step size as a function of the type of differential equation solver used.

	1	2	3	4	5	6	7	8
Row #	Type	EPS	β (%)	Error (%)	ℓ	Error (%)	Final Δt	Relative Cost
1	RKF4[1]		44	0	0.00649	0	0.0060	1.00
2	RKF4		46	4	0.00647	-0.3	0.0059	11.9
3	PC4		46	4	0.00662	2	0.0150	68.4

[1]Denotes analytical solution to potential flow about a Joukowski airfoil.

TABLE 6. Studying changes in accuracy and cost when single droplet size simulations are carried out with varied user input options and tolerances.

Row	1 NEF	2 NEB	3 NIF	4 EQN	5 CDS	6 EPS	7 CEDEL (%)	8 XO	9 β_o (%)	10 t_o	11 E_m (%)	12 A_T	13 Relative Cost
1	11	23	5	2	1	5×10^{-6}	1.0	-10	70.2	-0.006	35.5	0.002133	1.00
2	5	11	2	2	1	5×10^{-6}	1.0	-10	70.0	-0.006	35.5	0.002151	0.95
3	11	23	5	2	1	5×10^{-6}	4.0	-10	71.1	-0.007	35.5	0.002133	0.88
4	11	23	5	2	2	5×10^{-6}	4.0	-10	69.5	-0.006	34.1	0.002047	0.93
5	11	23	5	2	1	5×10^{-5}	4.0	-10	70.5	-0.006	35.5	0.002133	0.48
6	11	23	5	1	1	5×10^{-5}	4.0	-10	70.7	-0.007	34.5	0.002070	0.29
7	11	23	5	1	1	5×10^{-5}	4.0	-5	70.3	-0.007	34.5	0.002069	0.26
8	11	23	5	2	1	5×10^{-5}	1.0	-5	70.3	-0.006	35.6	0.002135	0.44
9	11	23	5	2	1	1×10^{-4}	1.0	-2.5	70.5	-0.006	35.6	0.002135	0.31

TABLE 7 Studying changes in accuracy and cost when multi-droplet size simulations are carried out with various degrees of smoothing

	1	2	3	4	5 ζ_ℓ	6	7	8	9	10	11	12
CASE	FILTER	DDISTN	DD	W	EPS	β_o (%)	ℓ_o	E_m (%)	$\bar{\beta}_o$ (%)	$\bar{\ell}_o$	A_T	COST
1	-	5	35.0	0.20	1×20^{-5}	81.9	-0.004	57.8	67.9	-0.006	0.002144	1.00
			25.4	0.20	2×10^{-5}	76.1	-0.005	44.8				
			20.0	0.20	3×10^{-5}	70.3	-0.006	35.6				
			15.4	0.20	4×10^{-5}	63.2	-0.006	26.4				
			10.0	0.20	5×10^{-5}	48.2	-0.007	14.1				
2	-	5	35.0	0.20	1×10^{-5}	81.8	-0.004	57.8	67.4	-0.006	0.002144	0.79
			25.4	0.20	4×10^{-5}	76.0	-0.005	44.8				
			20.0	0.20	1×10^{-4}	70.4	-0.006	35.5				
			15.4	0.20	8×10^{-5}	62.6	-0.007	26.4				
			10.0	0.20	6×10^{-5}	47.7	-0.009	14.1				
3	-	3	35.0	0.20	1×10^{-5}	81.8	-0.004	57.8	67.8	-0.006	0.002142	0.53
			20.0	0.60	1×10^{-4}	70.4	-0.006	35.5				
			10.0	0.20	6×10^{-5}	47.7	-0.009	14.1				
4	-	2	35.0	0.20	1×10^{-5}	81.8	-0.004	57.8	70.1	-0.006	0.002217	0.24
			18.0	0.80	8×10^{-5}	67.2	-0.006	31.7				
5	-	2	32.0	0.30	4×10^{-5}	80.5	-0.005	54.2	69.6	-0.006	0.002190	0.26
			16.6	0.70	4×10^{-5}	65.1	-0.006	28.9				
6	-	2	28.8	0.40	4×10^{-5}	78.5	-0.005	49.9	69.1	-0.006	0.002147	0.26
			15.4	0.60	4×10^{-5}	62.9	-0.006	26.4				
7	0.10c[1]	2	27.4	0.50	1×10^{-5}	77.7	-0.005	47.9	64.3	-0.008	0.002150	0.35
			14.2	0.50	4×10^{-5}	60.5	-0.006	23.8				

continued...

	1	2	3	4	5	6	7	8	9	10	11	12
CASE	FILTER	DDISTN	DD	W	EPS	β_o (%)	ℓ_o	E_m (%)	$\bar{\beta}_o$ (%)	$\bar{\ell}_o$	A_T	COST
8	$0.10V^2$	2	27.4 14.2	0.50 0.50	1×10^{-5} 4×10^{-5}	77.7 60.5	-0.005 -0.006	47.9 23.8	69.0	-0.006	0.002168	0.37
9	$0.20c^1$	1	20.0	1.00	1×10^{-4}	70.6	-0.006	35.5	61.3	-0.011	0.002132	0.12
10	$0.20V^2$	1	20.0	1.00	1×10^{-4}	70.6	-0.006	35.5	70.4	-0.005	0.002198	0.12
11[3]	-	1	20.0	1.00	6×10^{-4}	72.7	-0.006	35.9	72.7	-0.006	0.002154	1.14
12[4]	-	1	20.0	1.00	1×10^{-4}	69.9	-0.007	36.1	69.9	-0.007	0.002164	1.43

[1] constant length Boxcar filter on β curve

[2] variable length filter

[3] NEF = 14 NEB = 11 NIF = 4 TYPE = 2 (vorticity density)

[4] NEF = 11 NEB = 10 NIF = 5 TYPE = 2 (vorticity density)

TABLE 8. Intercomparisons of the characteristics of droplet impingement upon cylinders.

Case #	EQN[3]	Re_∞	K^1	K^2	ϕ^1	ϕ^2	E_m (%)	ℓ_G	β_0 (%)	u*	v*
A	0	7.071	0.50	0.25	100	200	12.7	0.283	32.0		
B	0						15.7	0.300		0.445	0.650
13	1						15.7	0.318	35.0	0.459	0.633
14	2						18.2	0.348	37.5	0.535	0.654
A	0	894.4	16	8	50,000	100,000	61.5	0.585	76.0		
B	0						68.2	0.629		1.009	0.327
15	1						60.7	0.584	77.5	1.002	0.432
16	2						61.1	0.586	77.7	1.006	0.428
A	0	100.0	1	0.5	10,000	20,000	14.8	0.299	34.3	0.494	0.725
B	0						15.7	0.298		0.441	0.650
17	1						16.3	0.311	37.1	0.451	0.639
18	2						17.9	0.330	38.7	0.502	0.656

[1]Value according to definition of Langmuir and Blodgett

[2]Value according to present definition

[3]EQN = 0: basic trajectory equations
= 1: as above with addition of induced droplet mass during acceleration
= 2: as for EQN = 1 but with the addition of the history term

A: Results of Langmuir and Blodgett (1946)

B: Results of Brun and Mergler (1953)

TABLE 9. Intercomparisons of the characteristics of droplet impingement upon cylinders.

Case #	EQN	CDS	Re_∞	K^1	K^2	ϕ^1	ϕ^2	E_m (%)	ℓ_G	β_0 (%)
A	0	2	14.0	0.196	0.098	1000	2000	0.9	0.080	9.5
C	0	-						2.46	0.221	12.2
19	1	1						0.8	0.068	7.6
20	1	2						0.5	0.052	5.9
21	2	1						2.8	0.148	12.7
A	0	2	180.0	32.4	16.2	1000	2000	88.0	0.728	93.2
C	0	1						85.7	0.731	91.4
22	1	1						88.3	0.725	93.6
23	1	2						87.7	0.722	93.3
24	2	1						88.4	0.730	93.6

[1]Value according to definition of Langmuir and Blodgett

[2]Value according to present definition

A: Results of Langmuir and Blodgett (1946)

C: Results of McComber and Touzot (1981)

TABLE 10. Intercomparisons of the characteristics of droplet impingement on a Joukowski airfoil of 36.5% thickness.

Case #	EQN	Re_∞	K	ϕ	E_m (%)	ℓ_G	β_o (%)
D	0	16	0.3214	796.5	41.0	0.184	68.0
25	1				39.9	0.186	64.9
26	2				41.3	0.190	65.8

D: Results of Brun and Voyt (1957)

TABLE 11 Intercomparisons of the characteristics of droplet impingement on a NACA 0012 airfoil

Case #	EQN	Re_∞	K	ϕ	E_m	ℓ_{GU}	ℓ_{GL}	β_o	ℓ_o
E	0	202.2	0.238	1.718×10^5	32.5	0.039	-0.110	71.5	-0.006
27	1				34.5	0.018	-0.147	70.2	-0.006
28	2				35.6	0.018	-0.153	71.1	-0.006

E: Results of Werner (1973)

TABLE 12. Intercomparisons of the characteristics of droplet impingement on a NACA 0015 airfoil.

Case #	EQN	Re_∞	K	ϕ	E_m (%)	ℓ_{GU}	ℓ_{GL}	β_o (%)	ℓ_o
F	0	55	0.257	1.18×10^4	47.3	0.020	-0.234	73.8	-0.013
29	1				49.9	0.021	-0.260	75.5	-0.011
F	0	109	0.407	2.92×10^4		0.018	-0.285	76.4	-0.019
30	1				58.2	0.024	-0.321	78.8	-0.011

F: Results of Bragg (1981)

TABLE 13. Intercomparisons of the characteristics of droplet impingement on a Joukowski airfoil of 15% thickness at 0° angle of attack.

Case #	Filter	EQN	DDISTN	DD	W	β_o (%)	E_m (%)	ℓ_G	$\bar{\beta}_o$	$\bar{E}_m$	ℓ_G
G	-			18.6					68.8	37.8	0.189
31	0.0	2	2	25.5 13.2	0.50 0.50	81.3 62.7	47.8 24.4	0.115 0.057	72.0	36.1	0.115
32	0.20	2	2	25.5 13.2	0.50 0.50	81.3 62.7	47.8 24.4	0.115 0.057	71.9	36.6	0.138
33	0.30	2	1	18.6	1.00	73.4	36.1	0.085	73.2	37.1	0.111
34	0.20	1	2	25.5 13.2	0.50 0.50	80.9 61.7	47.0 23.3	0.109 0.054	71.2	35.7	0.130

G: Results of Gelder et al. (1956)

TABLE 14. Intercomparisons of the characteristics of droplet impingement on a Joukowski airfoil of 15% thickness at 4° angle of attack.

CASE #	ALPHA	FILTER	DDISTN	DD	W	β_o (%)	E_m (%)	ℓ_{GU}	ℓ_{GL}	$\bar{\beta}_o$ (%)	$\bar{E}_m$ (%)	$\bar{\ell}_{GU}$	$\bar{\ell}_{GL}$
G	4.0	-	-							-70.0	39.2	0.117	-0.292
H	4.0	-									37.0		
I	4.0	-	-								39.0		
35	4.0	-	2	25.1 13.4	0.50 0.50	74.7 62.1	50.8 26.4	0.058 0.029	-0.202 -0.108	74.7	38.6	0.061	-0.206
36	4.0	0.10	2	25.5 13.2	0.50 0.50	75.0 61.5	51.5 25.9	0.058 0.029	-0.209 -0.108	68.2	38.9	0.072	-0.226
37	4.0	0.20	1	18.6	1.00	72.2	38.7	0.043	-0.155	72.1	39.5	0.063	-0.177
38	3.0	0.10	2	25.5 13.2	0.50 0.50	80.3 62.0	49.9 25.3	0.068 0.034	-0.169 -0.090	71.2	37.7	0.081	-0.182

G: Results of Gelder et al. (1956)

H: Results of Kloner (1970)

I: Results of Guibert et al. (1949)

TABLE 15. Intercomparisons of the characteristics of droplet impingement on a NACA 65-212 airfoil at 4° angle of attack.

Case #	Re_∞	K	ϕ	E_m (%)	β_o (%)	ℓ_{GU}	ℓ_{GL}
G	86.4	0.0374	2.00×10^5	9.2	52.0	0.02	-0.13
39				9.6	49.9	0.005	-0.060
G	96.2	0.257	3.60×10^4	32.7	72.0	0.109	-0.460
F					78.0	0.017	-0.208
40				43.9	82.0	0.018	-0.279

F: Results of Bragg et al. (1981)

G: Results of Gelder et al. (1956)

TABLE 16. Intercomparisons of the characteristics of droplet impingement on a NACA 64-215 Hicks modified airfoil at 0.7° angle of attack.

Case #	Re_∞	K	ϕ	E_m (%)	β_o (%)	ℓ_{GU}	ℓ_{GL}
F^1	113.9	0.0436	2.976×10^5	5.3			
F^2				6.2			
41				8.2	36.7	0.032	-0.018

F^1: Experimental results of Bragg et al. (1981)

F^2: Theoretical results of Bragg et al (1981)

TABLE 17. Intercomparison of the characteristics of droplet impingement on a cylinder.

Case #	Filter	Athick	Ice	Dense	Layer	DDISTN	DD	w	β_o	E_m	ℓ_G	$\bar{\beta}_o$	$\bar{E}_m$	$\bar{\ell}_G$	A_T
G															0.081
H															0.065
42	0.0	0	0.157	0	1	1	20.0	1.00	71.8	55.1	0.575	71.8	55.1	0.575	0.0942
43	0.0	1	0.157	0	1	1	20.0	1.00	71.8	55.1	0.574	71.8	55.1	0.574	0.0864
44	0.0	1	0.157	0	1	2	27.0	0.50	80.5	67.4	0.634	70.2	53.6	0.634	0.0841
							14.4	0.50	59.7	39.7	0.481				
G															0.20
H															0.12
45	0.0	1	0.314	0	1	2	27.0	0.50	80.5	67.4	0.634	69.9	53.5	0.643	0.1681
							14.4	0.50	59.7	39.7	0.481				
46	0.20	1	0.314	0	1	2	27.0	0.50	80.5	67.4	0.634	68.9	53.5	0.766	0.1681
							14.4	0.50	59.7	39.7	0.481				
47	0.0	1	0.314	0	1	1	20.0	1.00	71.8	55.1	0.575	71.8	55.1	0.575	0.1728
48	0.20	1	0.314	0	1	1	20.0	1.00	71.8	55.1	0.575	70.9	55.0	0.694	0.1728
49	0.0	1	0.314	2	1	1	20.0	1.00	71.8	55.1	0.575	71.8	55.1	0.575	0.1879
50	0.0	1	0.314	1	1	1	20.0	1.00	71.8	55.1	0.575	71.8	55.1	0.575	0.2716
51	0.20	1	0.314	1	1	2	27.0	0.50	80.5	67.4	0.634	68.9	53.5	0.766	0.2565
							14.4	0.50	59.7	39.7	0.481				
52	0.20	1	0.1047	0	1	1	20.0	0.50	71.9	55.2	0.574	71.9	55.9	0.695	0.0585
					2				72.2	52.7	0.537	72.2	53.2	0.653	0.0557
					3				72.5	50.5	0.510	72.5	51.0	0.620	0.0534
												80.1	53.4	0.695	0.1677

G: Experimental results of Lozowski et al. (1979) as measured by the present author.
H: Theoretical results of Lozowski et al. (1979) as measured by the present author.

TABLE 18. Intercomparison of the characteristcis of droplet impingement on a NACA 0015 airfoil at 0° and 8° angle of attack.

Case #	Alpha	Filter	Layer	β_o	E_m	ℓ_{GU}	ℓ_{GL}	$\overline{\beta}_o$	$\overline{E}_m$	$\overline{\ell}_{GU}$	$\overline{\ell}_{GL}$	A_T
I												0.0025
54	0.0	0.20	1	80.5	43.4	0.108	0.108	80.4	44.0	0.130	0.130	0.002347
55	0.0	0.30	1	80.5	43.4	0.108	0.108	80.4	44.7	0.130	0.130	0.000798
			2	80.5	41.9	0.111	0.111	80.3	43.2	0.146	0.146	0.000772
			3	80.4	40.4	0.115	0.115	80.3	41.8	0.150	0.150	0.000746
			TOTAL					98.4	43.2	0.150	0.150	0.002316
I												0.0025
56	8.0	0.10	1	79.5	59.2	0.025	-0.310	79.4	60.1	0.042	-0.380	0.003293
57	8.0	0.15	1	79.5	59.2	0.025	-0.310	79.3	61.2	0.052	-0.380	0.001117
			2	78.4	58.4	0.026	-0.335	78.3	60.3	0.056	-0.387	0.001101
			3	79.2	57.5	0.028	-0.344	79.0	59.5	0.057	-0.394	0.001086
			TOTAL					93.0	60.4	0.057	-0.394	0.003305

I: Experimental results of Stallabrass and Lozowski (1978) as determined by a planimeter.

TABLE 19. Intercomparisons of the characteristics of droplet impingement on a NACA 0012 airfoil and a NPL 9815 airfoil at a 5.7° angle of attack.

Case #	Type	Ice	Layer	β_o (%)	E_m (%)	ℓ_{GU}	ℓ_{GL}	$\bar{\beta}_o$ (%)	$\bar{E}_m$ (%)	$\bar{\ell}_{GU}$	$\bar{\ell}_{GL}$	A_T
J												0.0014
58	0	0.0296	1	83.3	59.6	0.026	-0.281	83.2	60.1	0.042	-0.373	0.002142
59	0	0.0099	1	83.3	59.6	0.026	-0.281	83.2	60.3	0.042	-0.373	0.000716
			2	82.7	58.5	0.021	-0.303	82.6	59.7	0.037	-0.380	0.000709
			3	80.9	58.0	0.023	-0.316	80.7	60.4	0.041	-0.385	0.000717
			TOTAL					108.4	60.1	0.042	-0.385	0.002142
60	5	0.0279	1	81.1	58.4	0.033	-0.357	80.9	59.3	0.058	-0.493	0.001869
60[1]	5	0.0296	1	81.1	58.4	0.035	-0.379	80.9	59.3	0.062	-0.524	0.002108

[1]The results of the previous row have been adjusted to reflect the shorter chord length of the NACA 0012 airfoil.

J: Experimental results of Stallabrass (1958) as determined by a planimeter.

TABLE 20. Intercomparisons of the characteristics of droplet impingement on a Joukowski 0012 airfoil and on a NACA 0012 airfoil at a 4° angle of attack.

Case #	Type	β_o	E_m	ℓ_{GU}	ℓ_{GL}	A_T
12	2	69.9	36.1	0.021	-0.141	0.002164
27	0	70.2	34.5	0.018	-0.147	0.002073

TABLE 21. Intercomparisons of the characteristics of droplet impingement on a Joukowski 0015 airfoil at full and one-quarter scale.

Case #	C	DD	W	β_o	E_m	ℓ_G	$\overline{\beta}_o$	$\overline{E}_m$	$\overline{\ell}_G$	A_T
32	0.330	25.5	0.50	81.3	47.8	0.115	71.9	36.6	0.138	0.002743
		13.2	0.50	62.7	24.4	0.057				
61	0.0825	11.05	0.50	81.0	47.8	0.112	71.6	36.6	0.138	0.002742
		5.87	0.50	62.3	24.3	0.058				

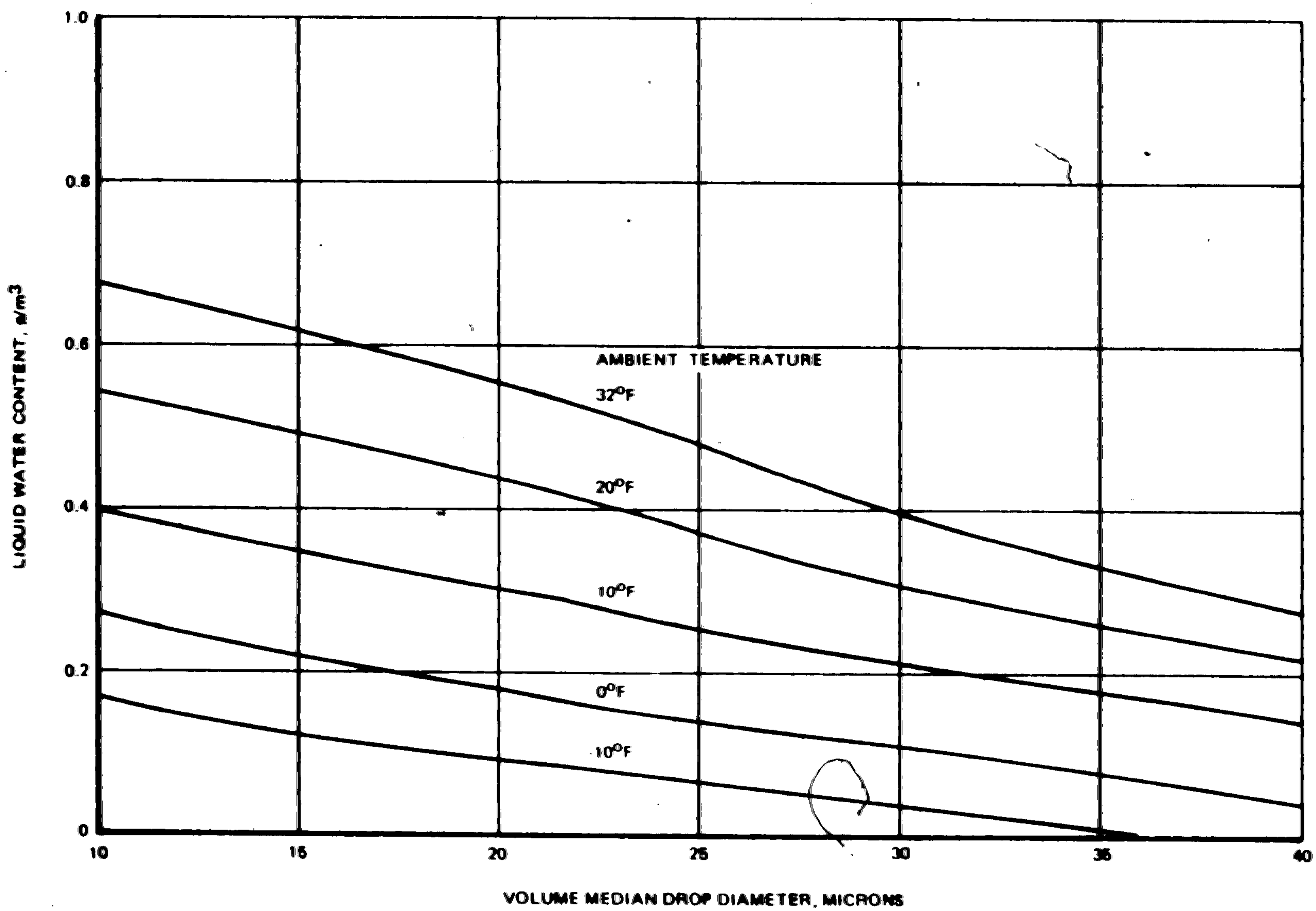

FIG. 1 Icing severity levels for a probability of exceedance equal to 0.01 for stratiform clouds (from Werner, 1975)

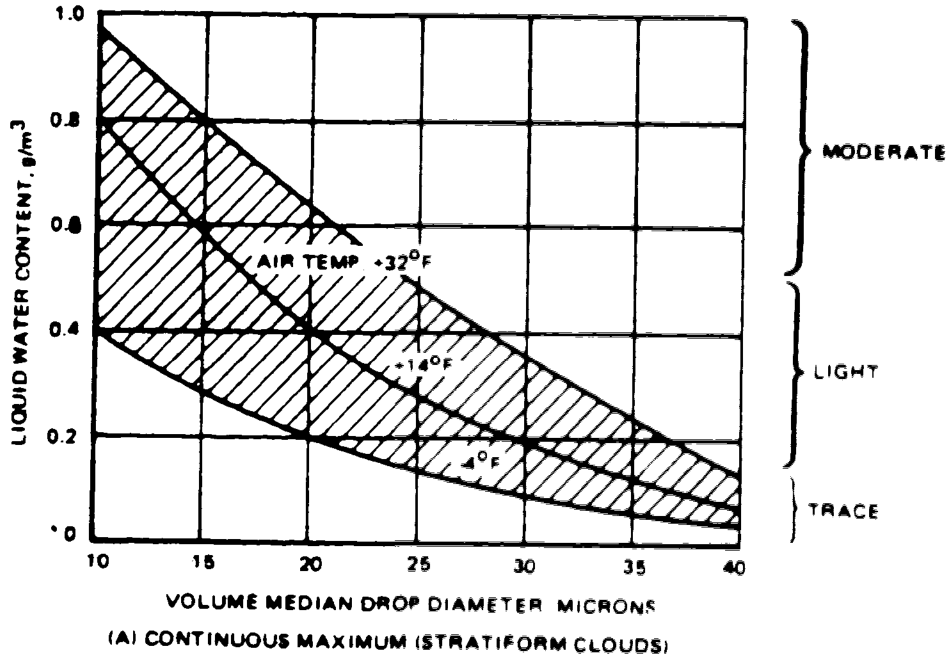

FIG. 2. Recommended atmospheric icing criteria for stratiform clouds (from Werner, 1975).

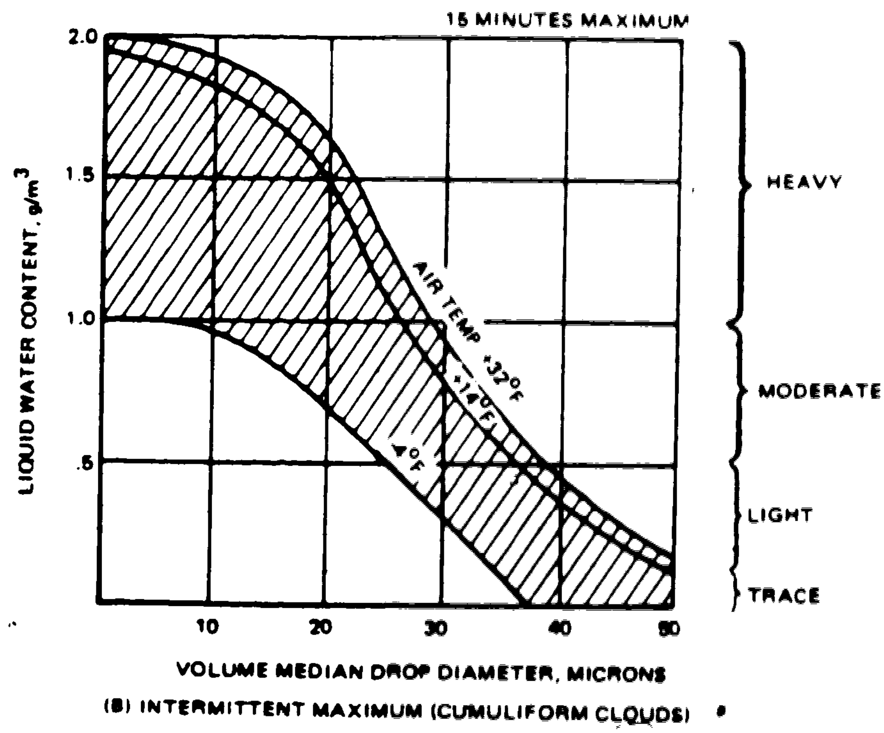

FIG. 3. Recommended atmospheric icing criteria for cumuliform clouds (from Werner, 1975).

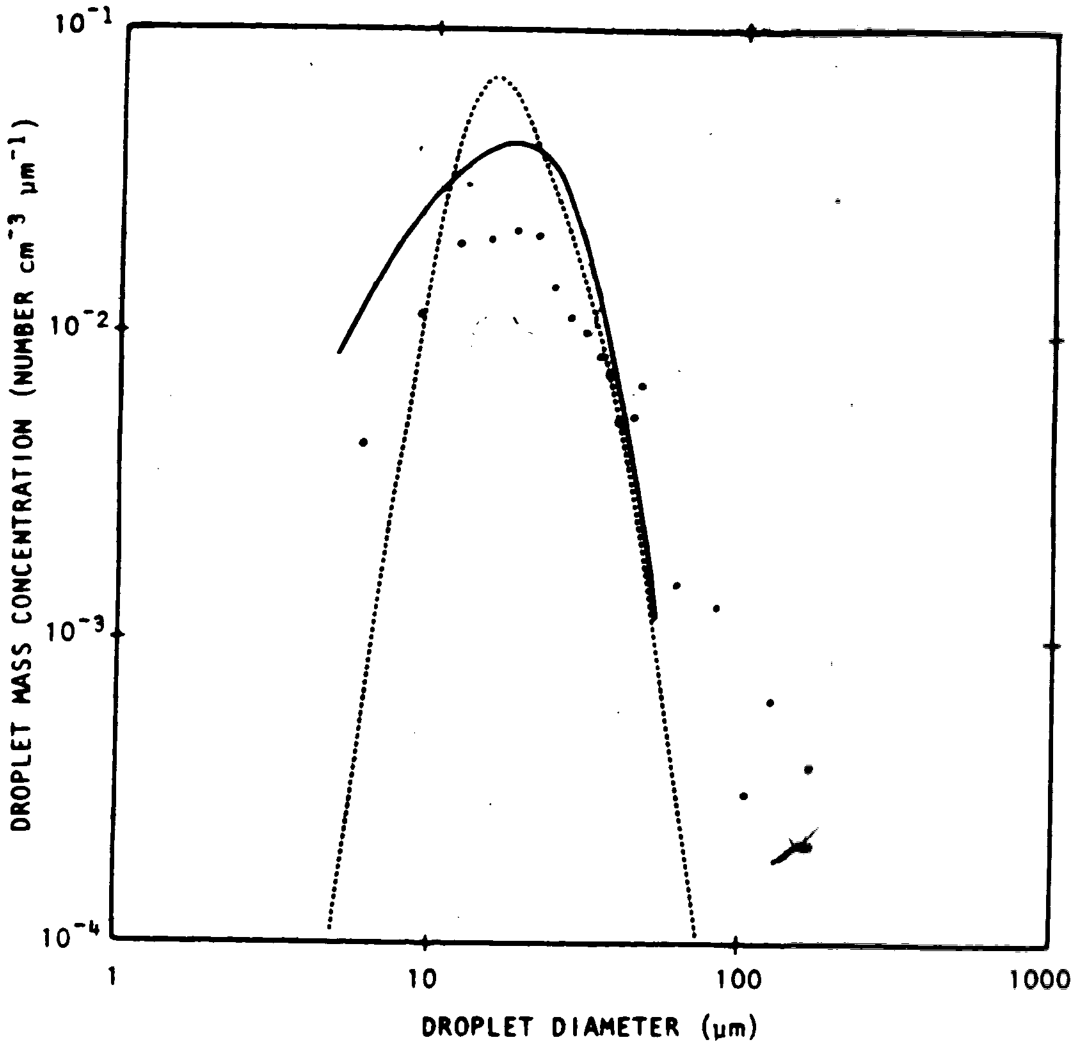

FIG. 4. A comparison of drop size mass distribution for a natural Minnesota cloud (dashed line), the spray from HISS (symbols), and from the Langmuir "D" distribution (solid line).

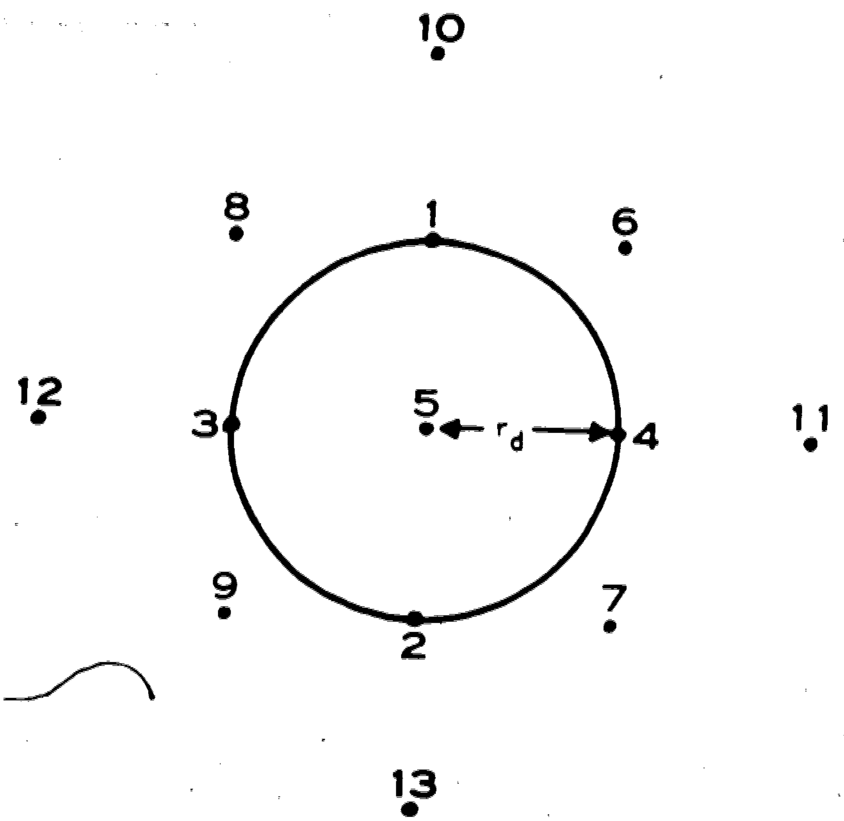

FIG. 5. Gridpoint notation for the grid, centered upon and moving with the droplet, which is used to calculate air velocities and accelerations. The grid length is equal to the radius of the droplet.

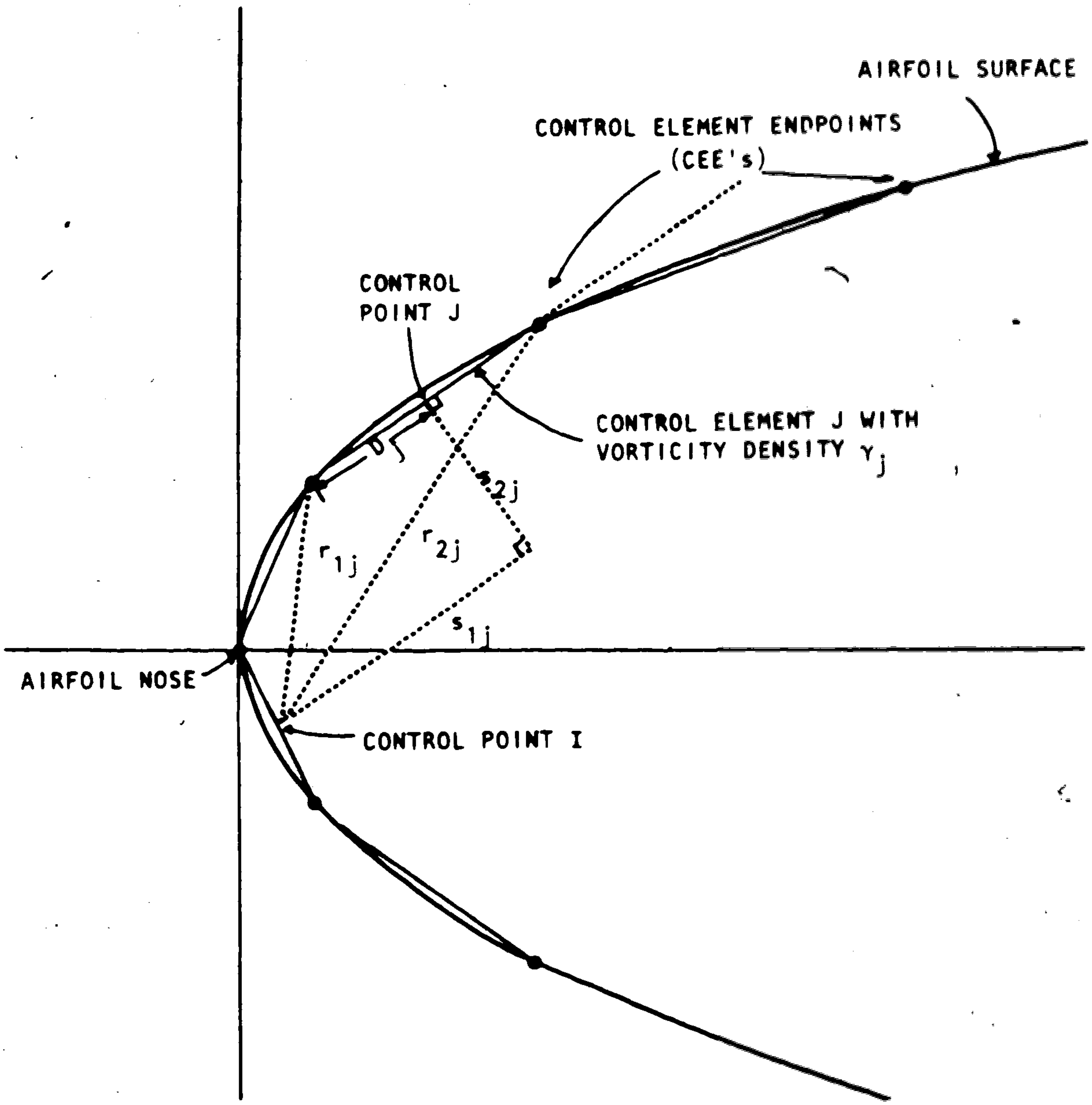

FIG 6. Notation used to calculate influence coefficients (after Kennedy & Marsden, 1976).

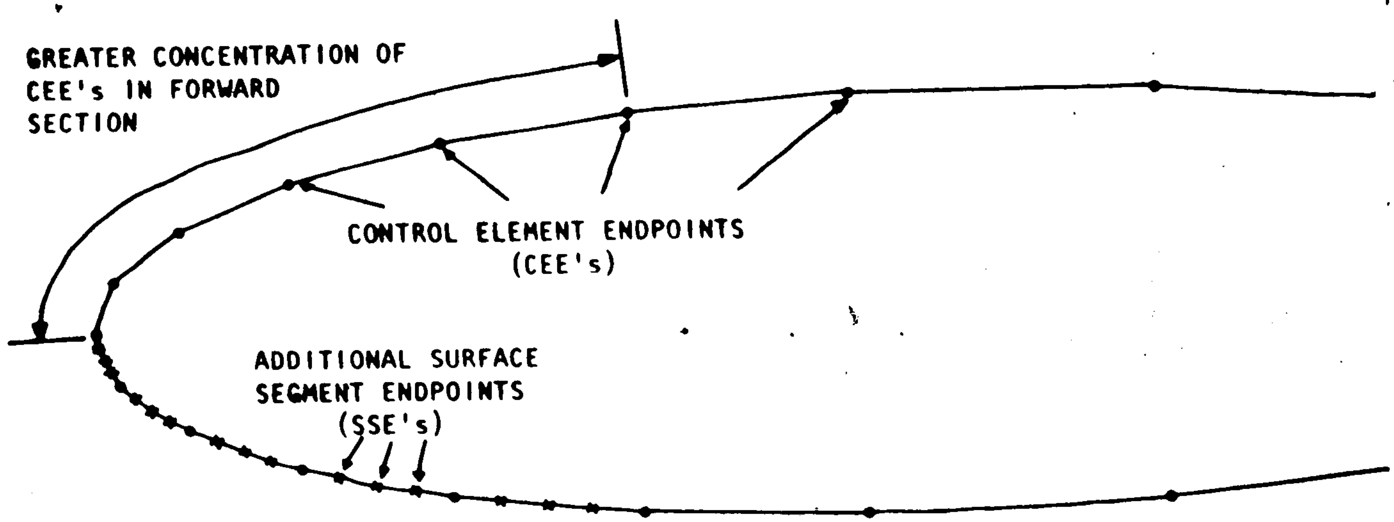

FIG. 7. A typical airfoil as defined by a series of control element endpoints and surface segment endpoints. The former also define control segments used to model the potential flow about the airfoil. A greater concentration of CEE's in the forward section improves the flow accuracy in the icing region. Additional SSE's provide greater definition and accuracy for the icing surface of the airfoil.

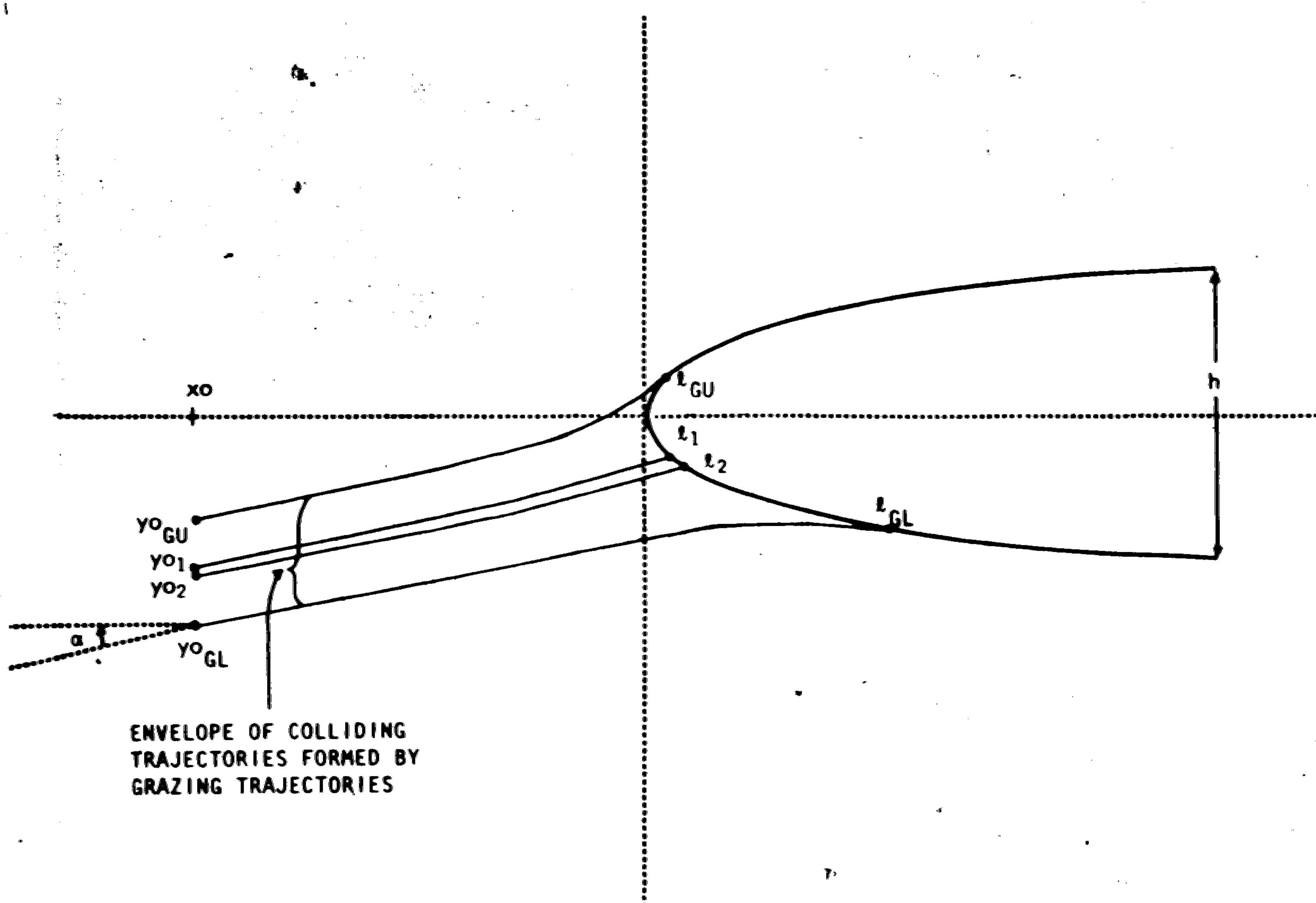

FIG. 8. Droplet trajectories which define the local and total collision efficiency.

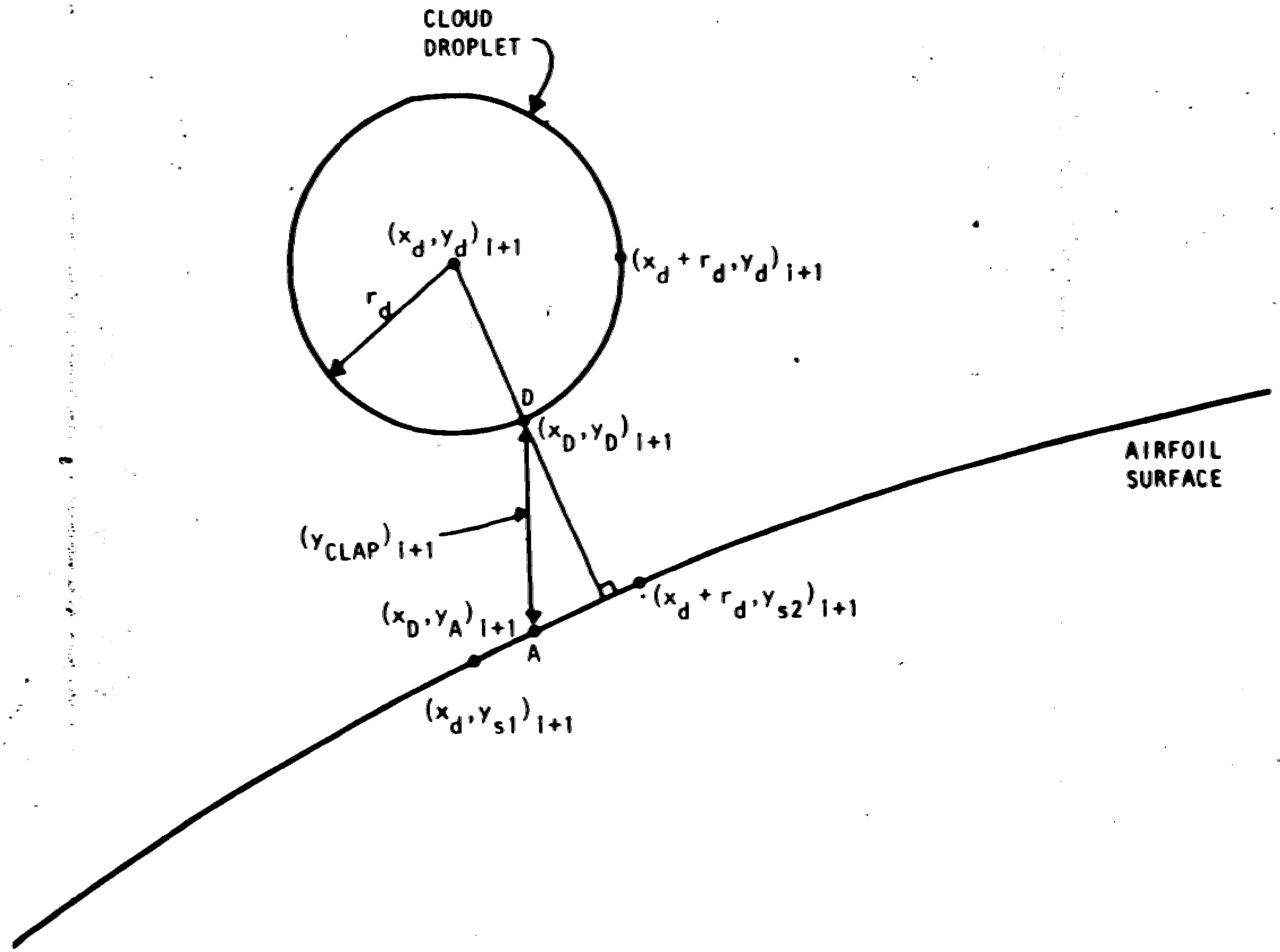

FIG. 9. Finding the closest vertical approach (distance AD) between the droplet and airfoil surfaces at time t_{i+1}

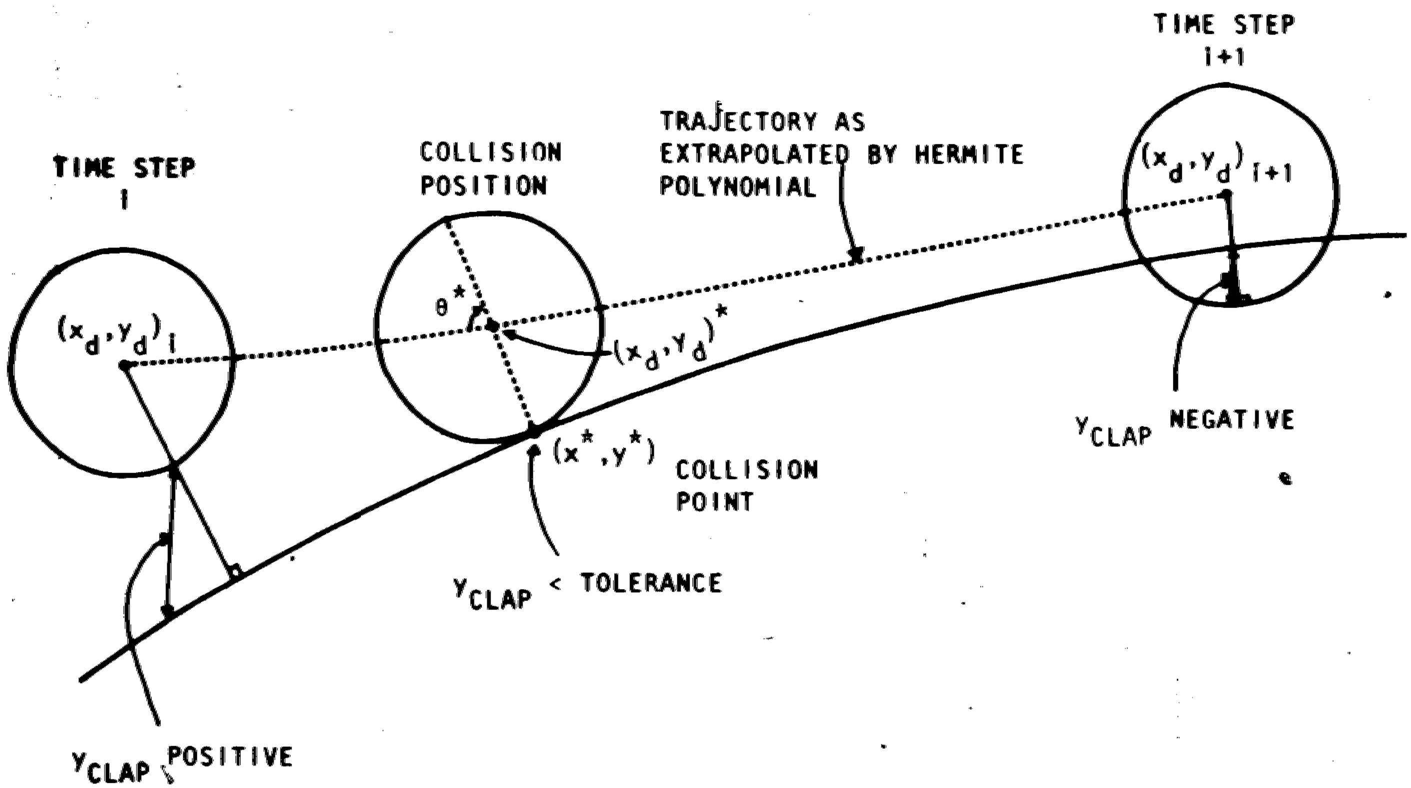

FIG. 10. The droplet position at collision is illustrated as lying along the trajectory predicted by Hermite extrapolation between the positions at time t_i (when Y_{CLAP} is positive and time t_{i+1} (when Y_{CLAP} is negative).

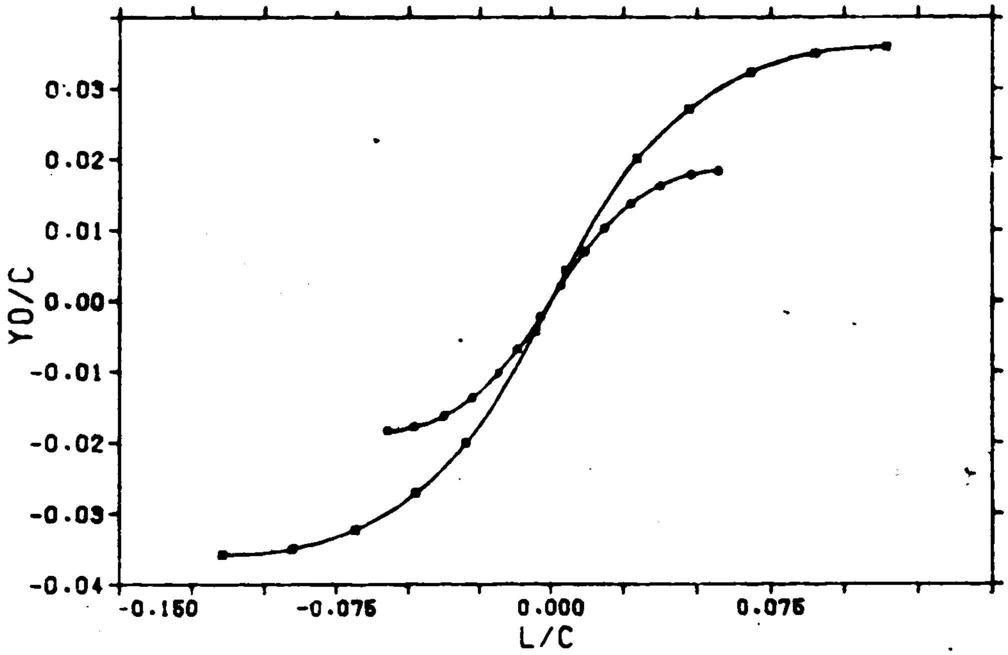

FIG. 11. A sample yo *vs.* ℓ curve.

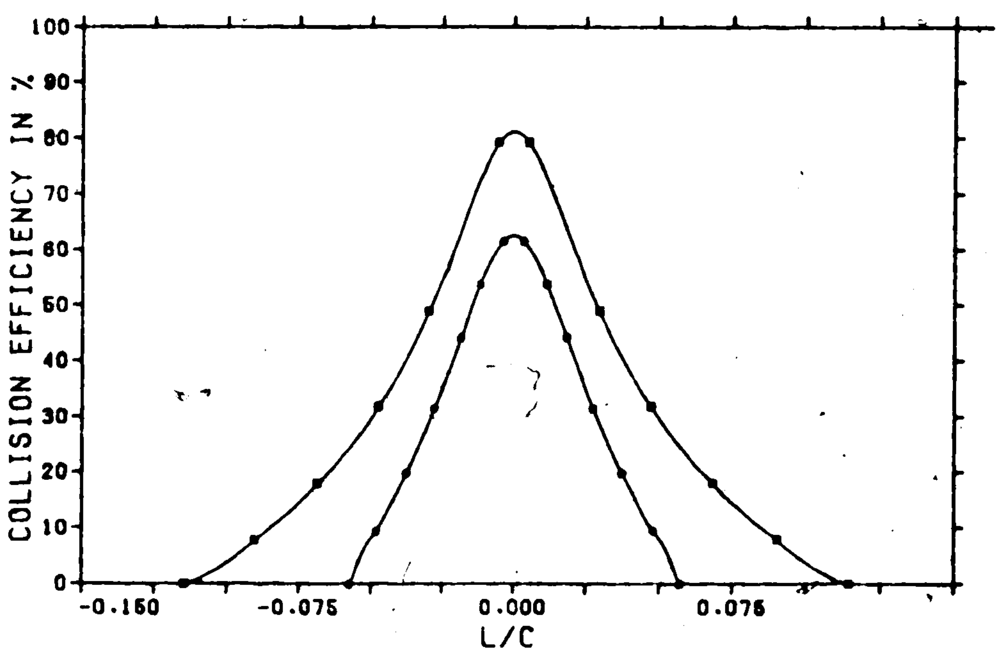

FIG. 12. A sample β *vs.* ℓ curve.

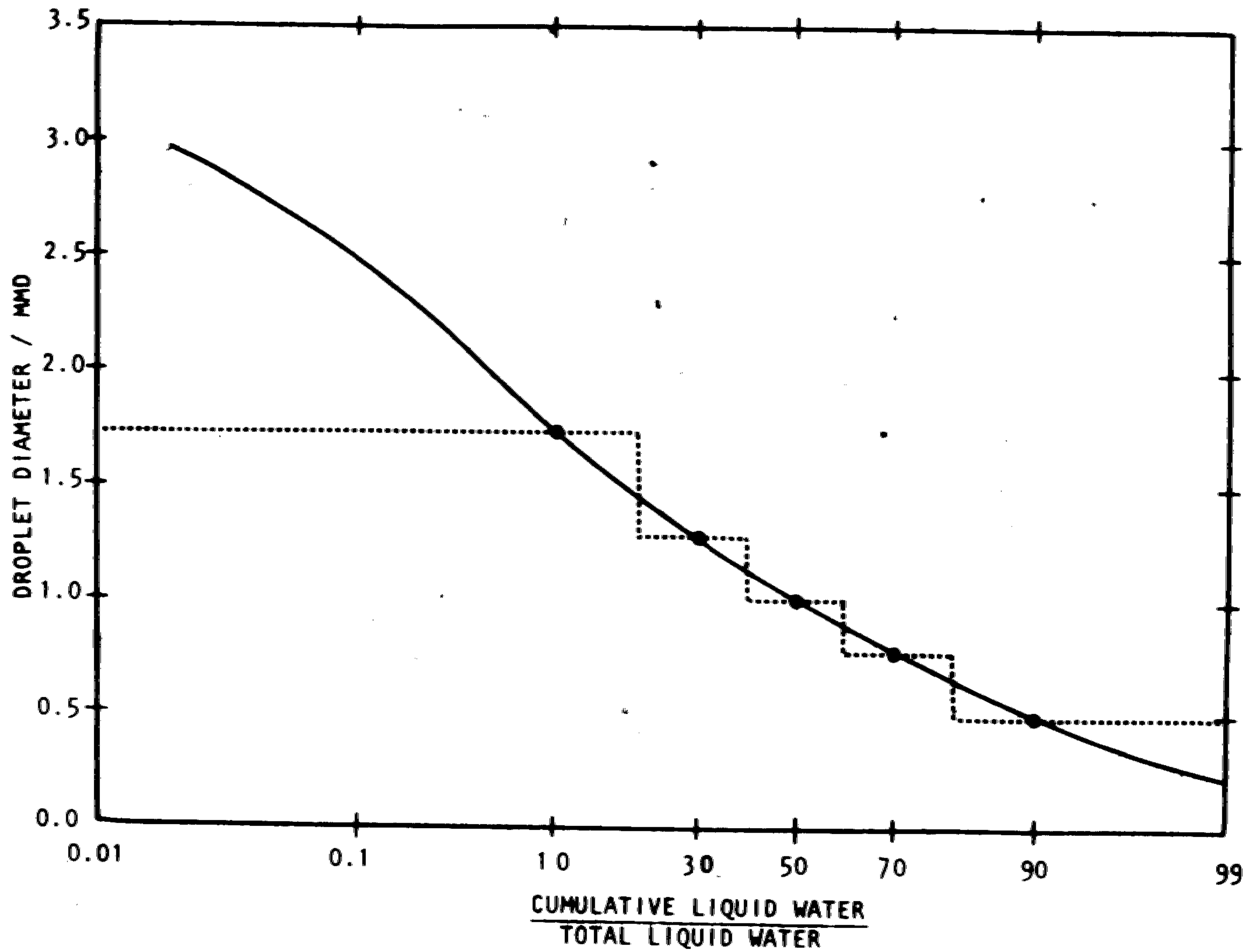

FIG. 13. The Langmuir "D" distribution of droplet sizes (as a solid line) and its approximation by a set of five droplet size categories (shown by dashed lines).

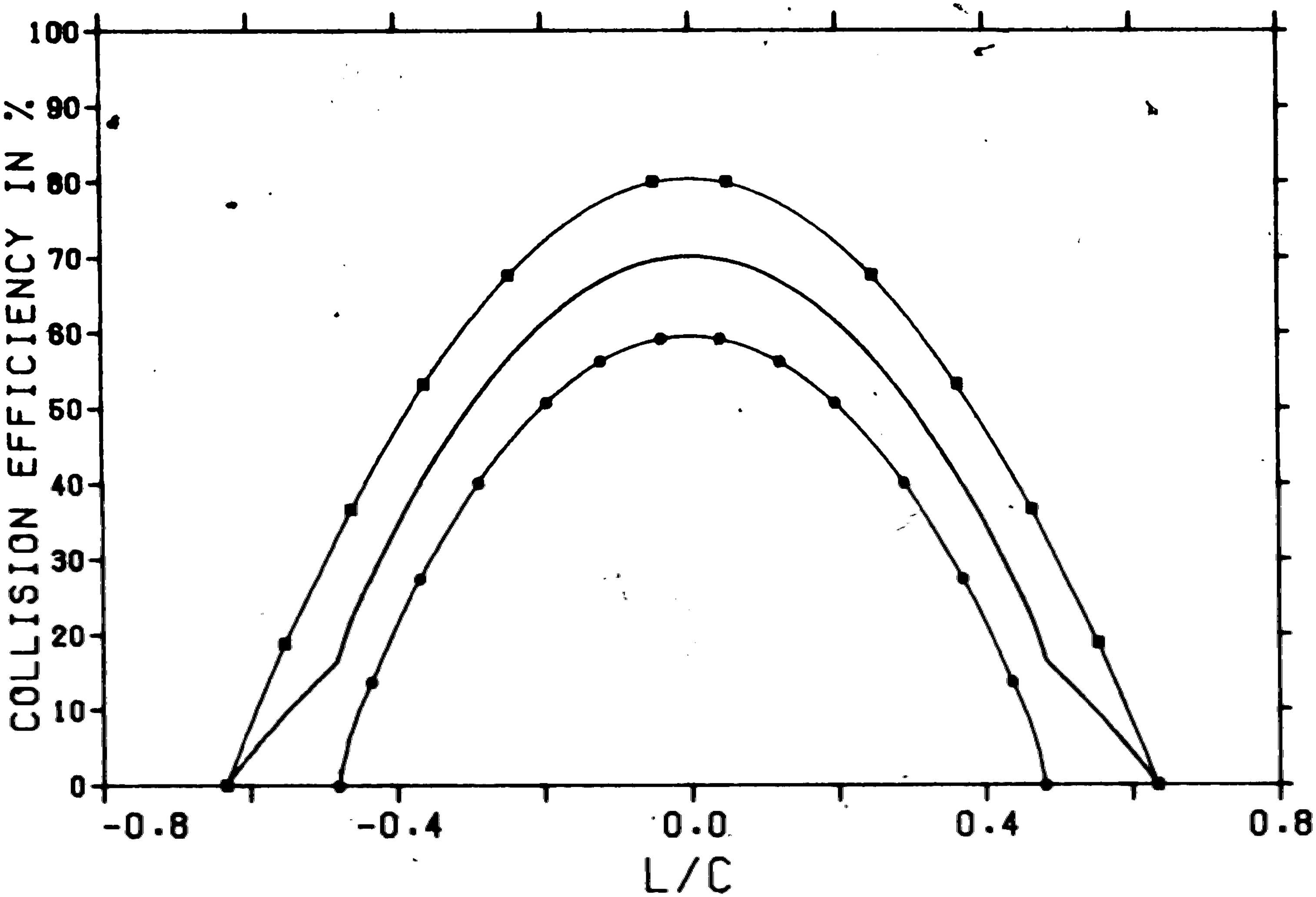

FIG 14 A sample collision efficiency curve for a two droplet size category distribution

FIG. 15. The characteristics of rime growth on a microscopic scale (after Lozowski (1981)).

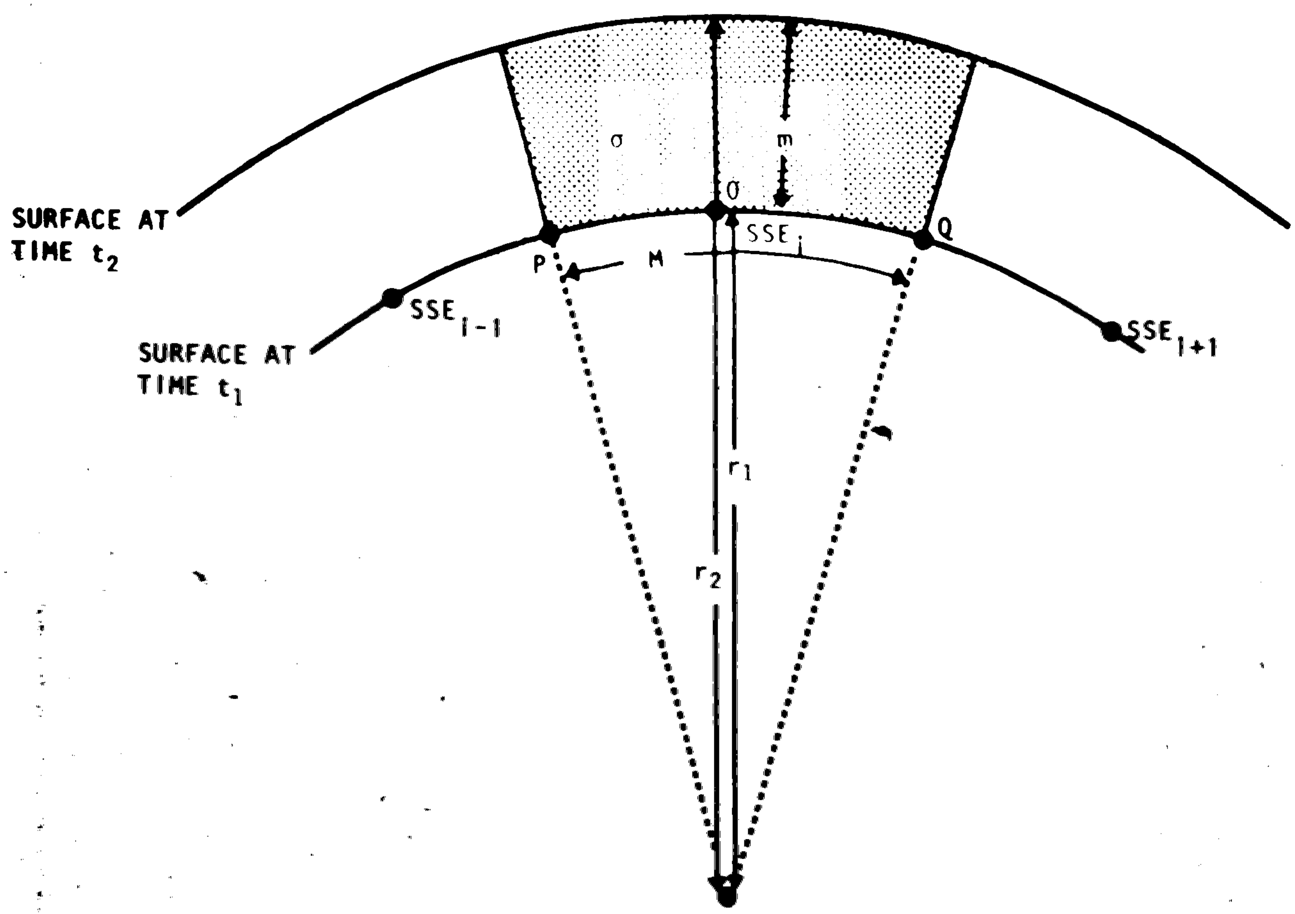

FIG. 16. The cross-sectional area and thickness of accretion on a curved 2-D surface.

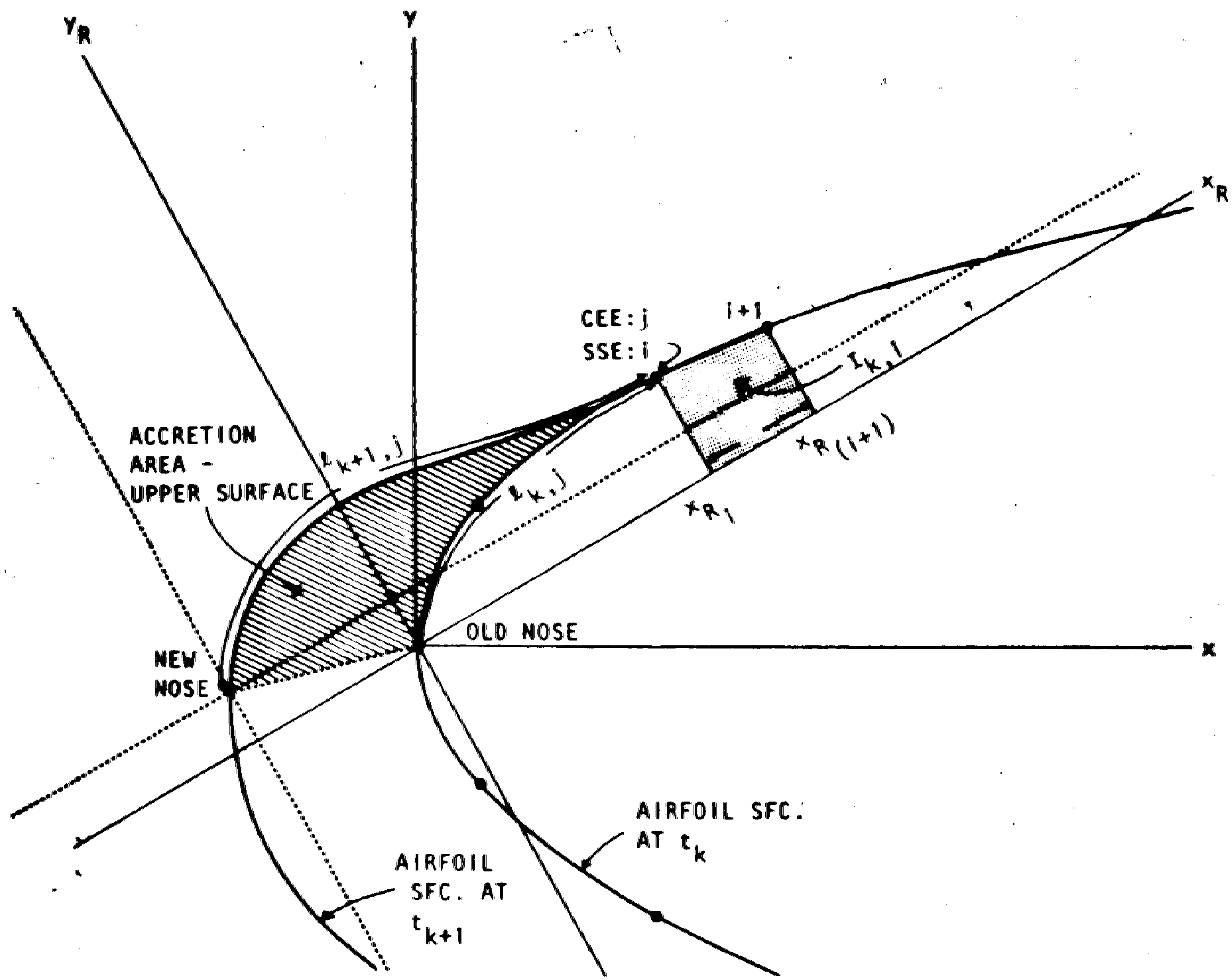

FIG. 17. Determining the area of the accretion layer, and placing CEE's on the new airfoil surface

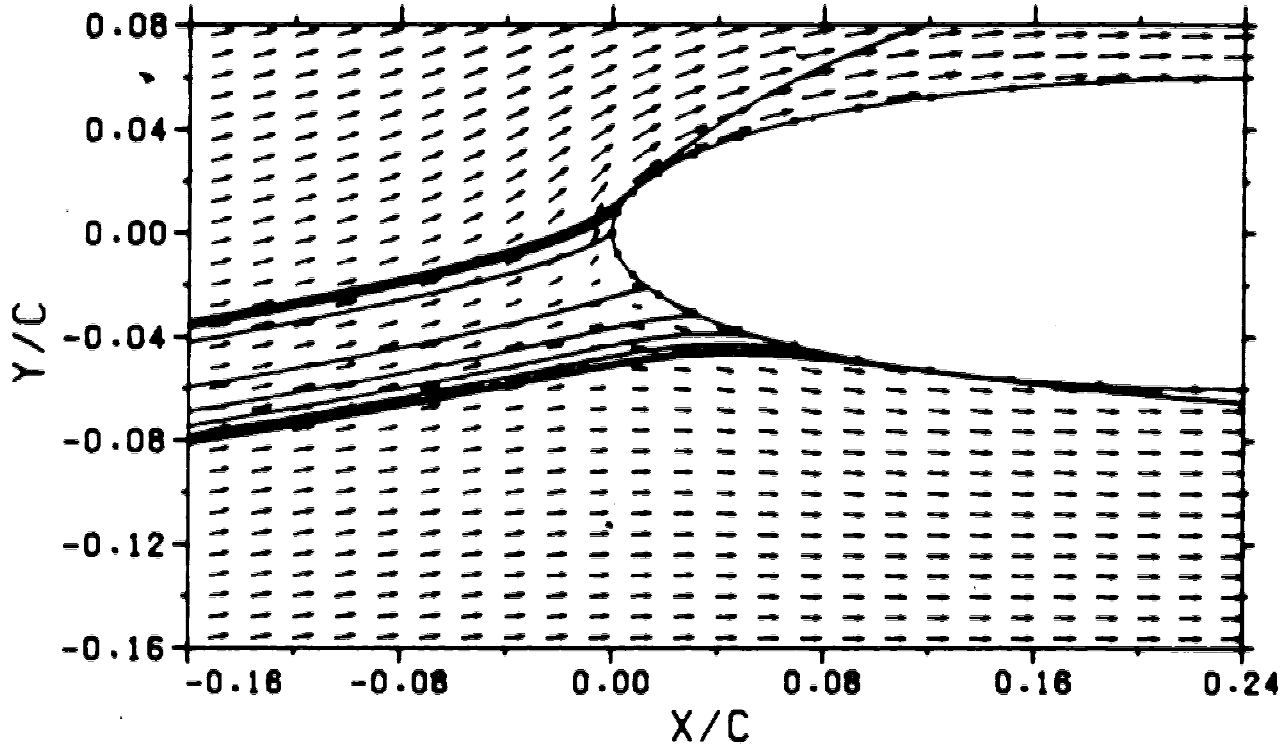

FIG. 18. The potential flow velocity vectors and a series of trajectories for a Joukowski 0012 airfoil at 4.6° attack angle. Non-dimensional parameters are K=0.249 and Re_∞ =221.9

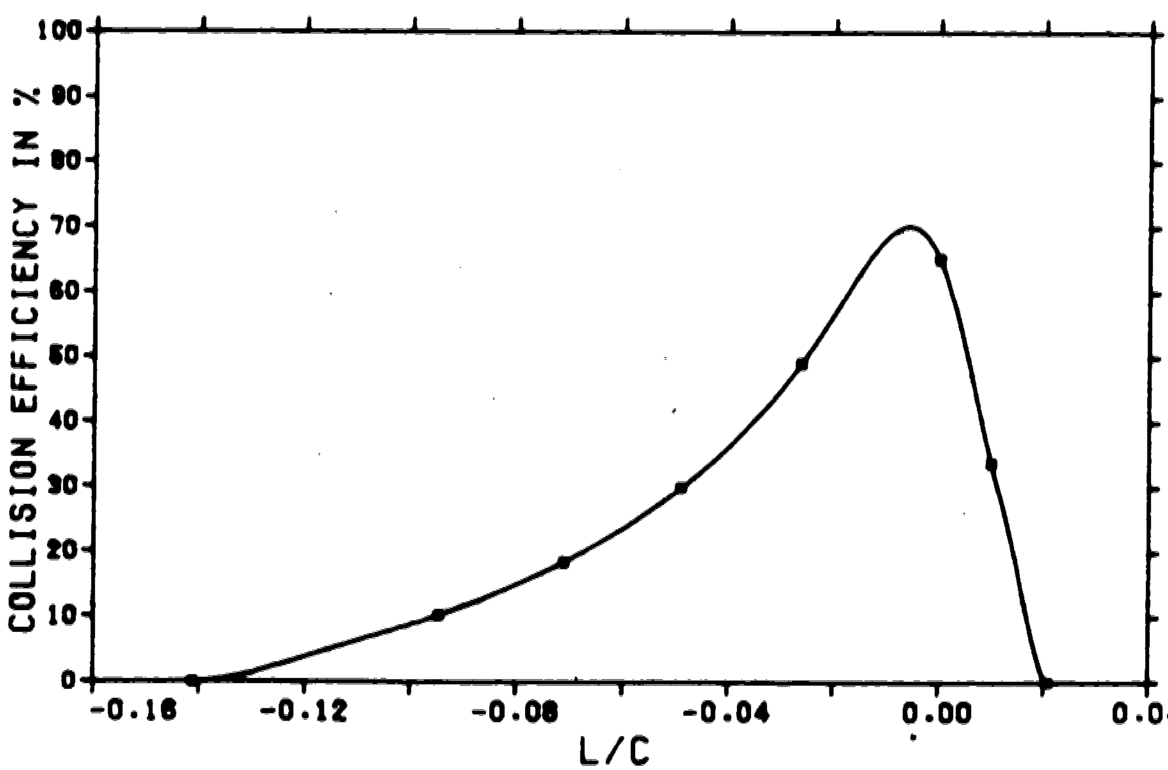

FIG. 19. The β curve for Case 1 of Table 6, corresponding to the trajectories plotted in Fig. 18

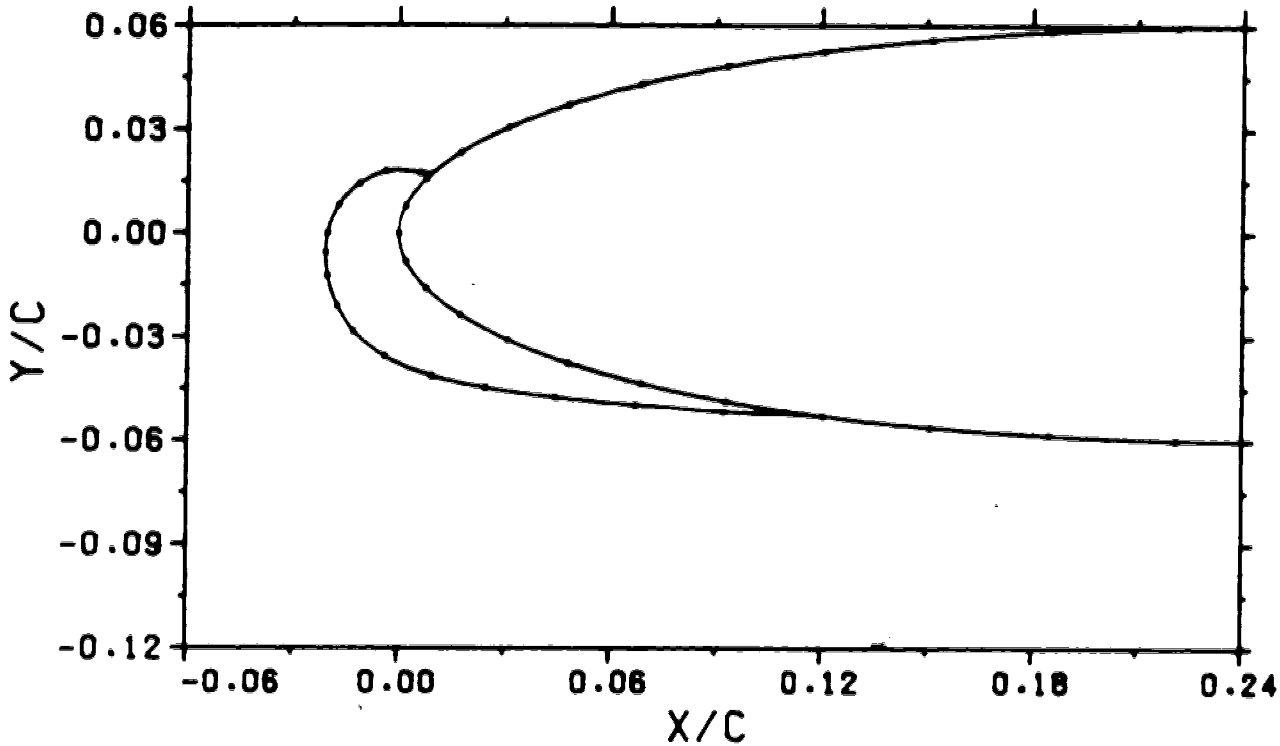

FIG. 20. The predicted ice accretion for Case 1 of Table 6 when the ND-accretion parameter ω =0.050, and surface curvature is incorporated in calculating the ND accretion thickness m (ATHICK=1). K=0.249 and Re_∞ =221.9

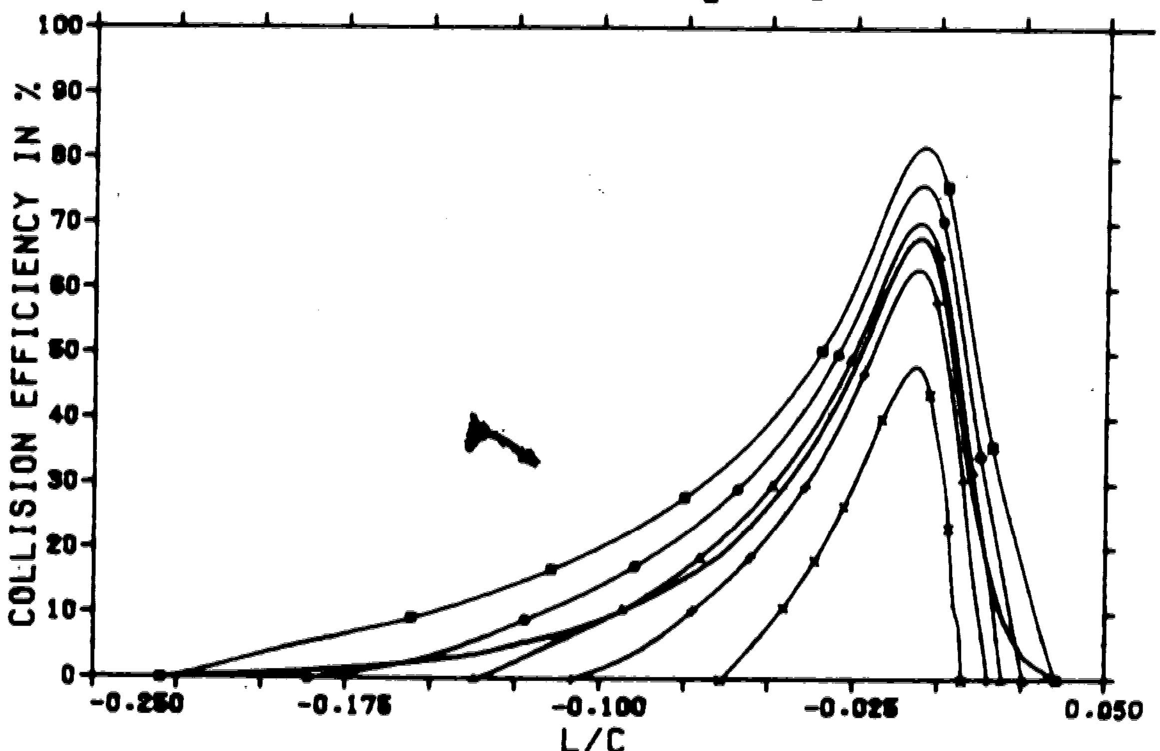

FIG. 21. The set of β curves for Case 1 of Table 7. The curves with symbols are for droplet diameters 35.0, 25.4, 20.0, 15.4 and 10.0 μm, nested in that order. The heavier line without symbols is the mean curve for the distribution $\bar{\beta}$.

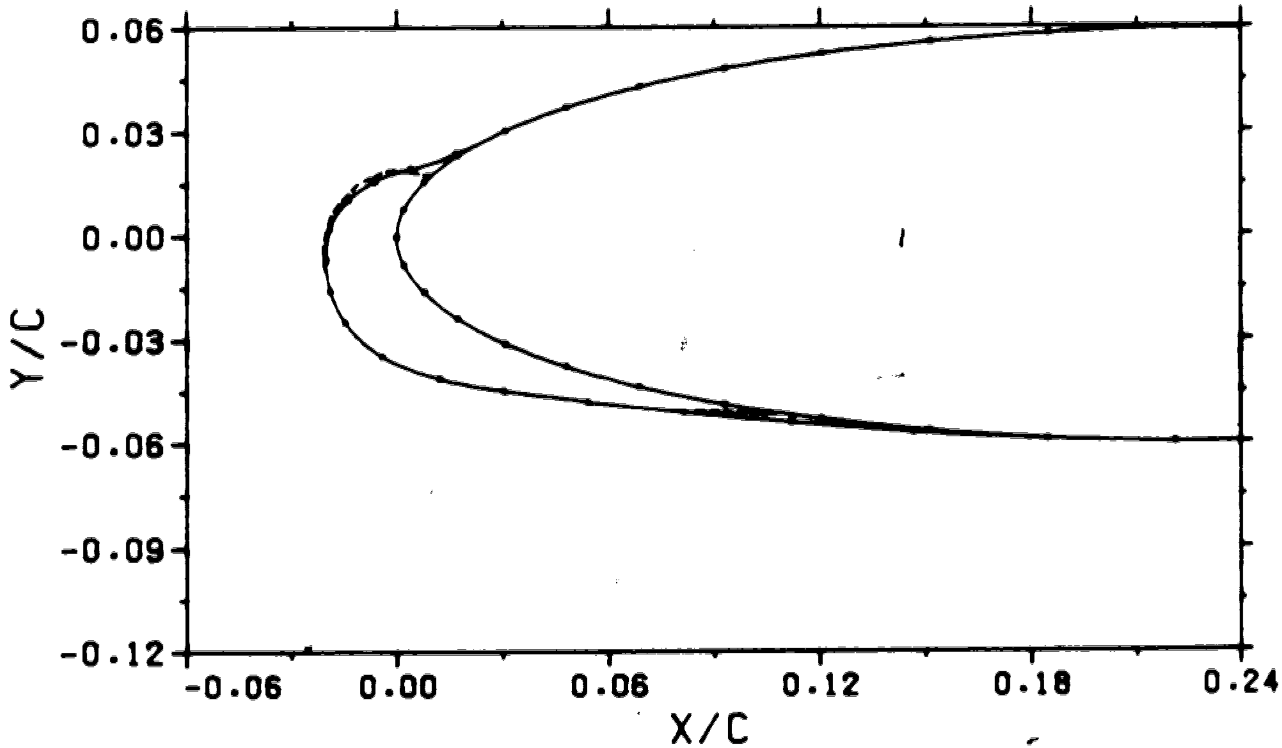

FIG. 22. The predicted ice accretion for Case 1 of Table 7 (in solid) compared to that for a monodisperse droplet distribution with all droplets having the mass median diameter of the distribution used in Case 1.

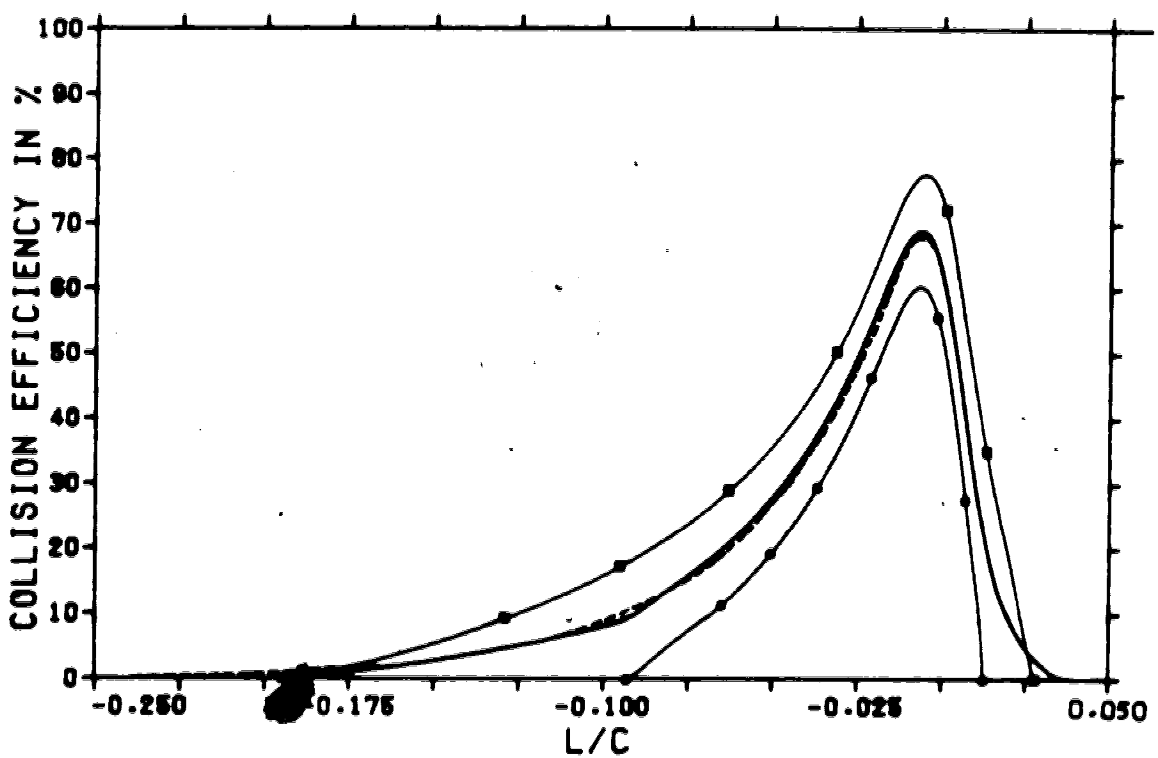

FIG. 23. The set of β and $\bar{\beta}$ curves for Case 8 of Table 7 in solid lines with symbols and a heavy solid line without symbols, respectively. Superimposed is a dashed $\bar{\beta}$ curve corresponding to the 5 category simulation of Case 1 of Table 7.

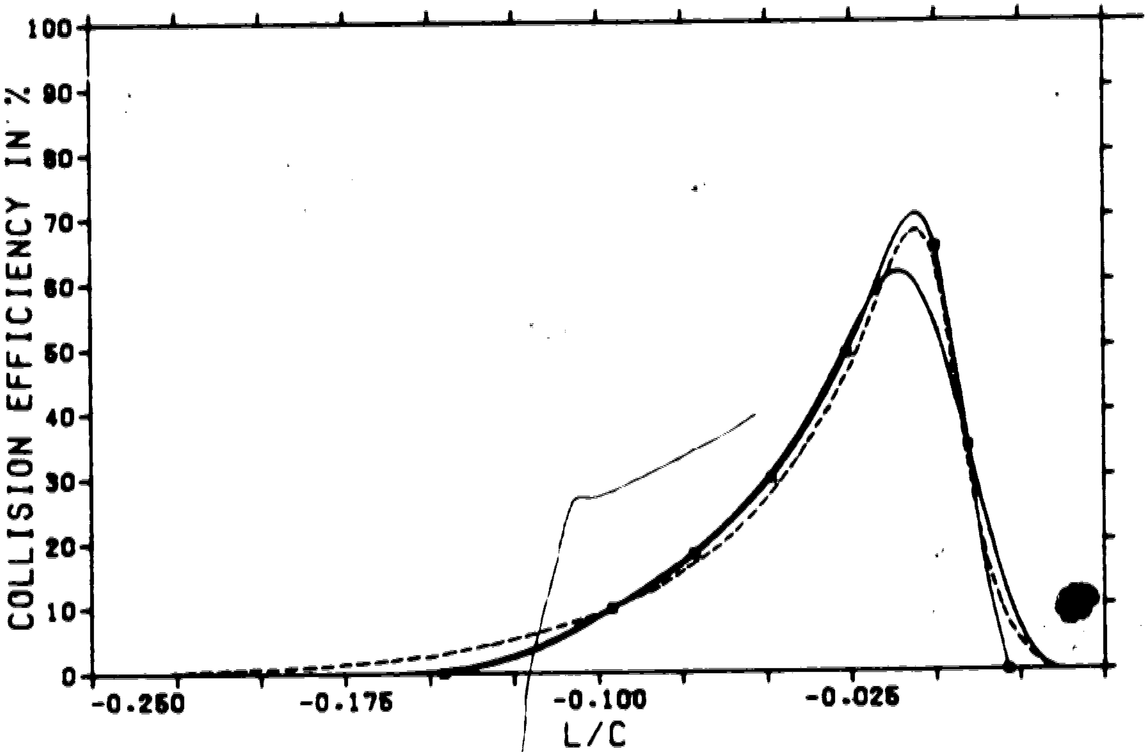

FIG. 24. As for Fig 23, except that Case 9 of Table 7 is shown.

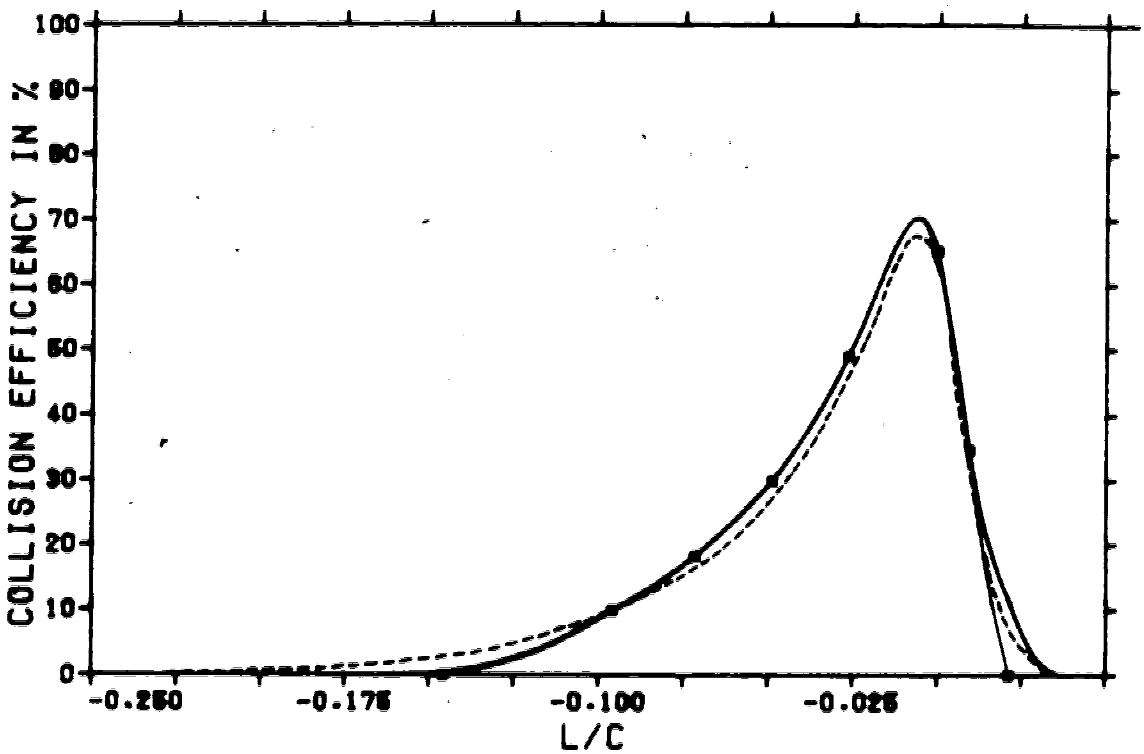

FIG. 25. As for Fig. 23, except that Case 10 of Table 7 is shown.

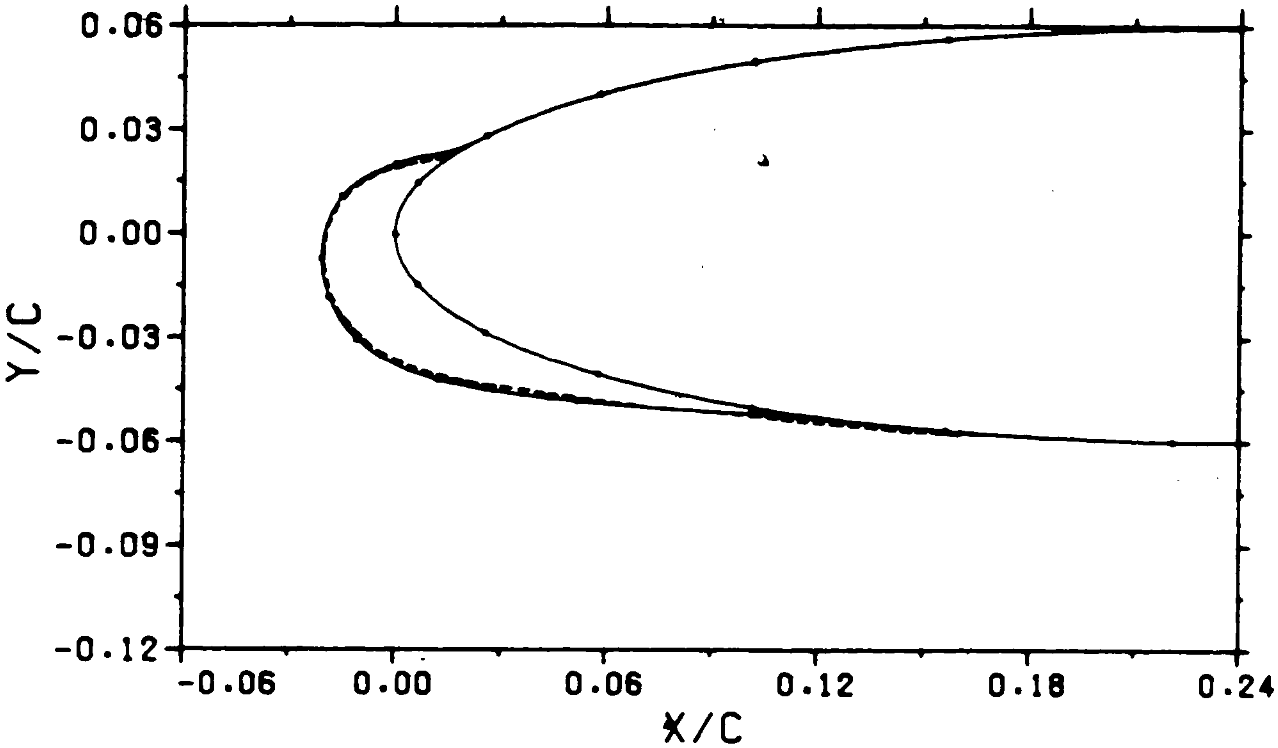

FIG. 26 The accretion profiles of Case 10 (solid line) and Case 1 (dashed line)

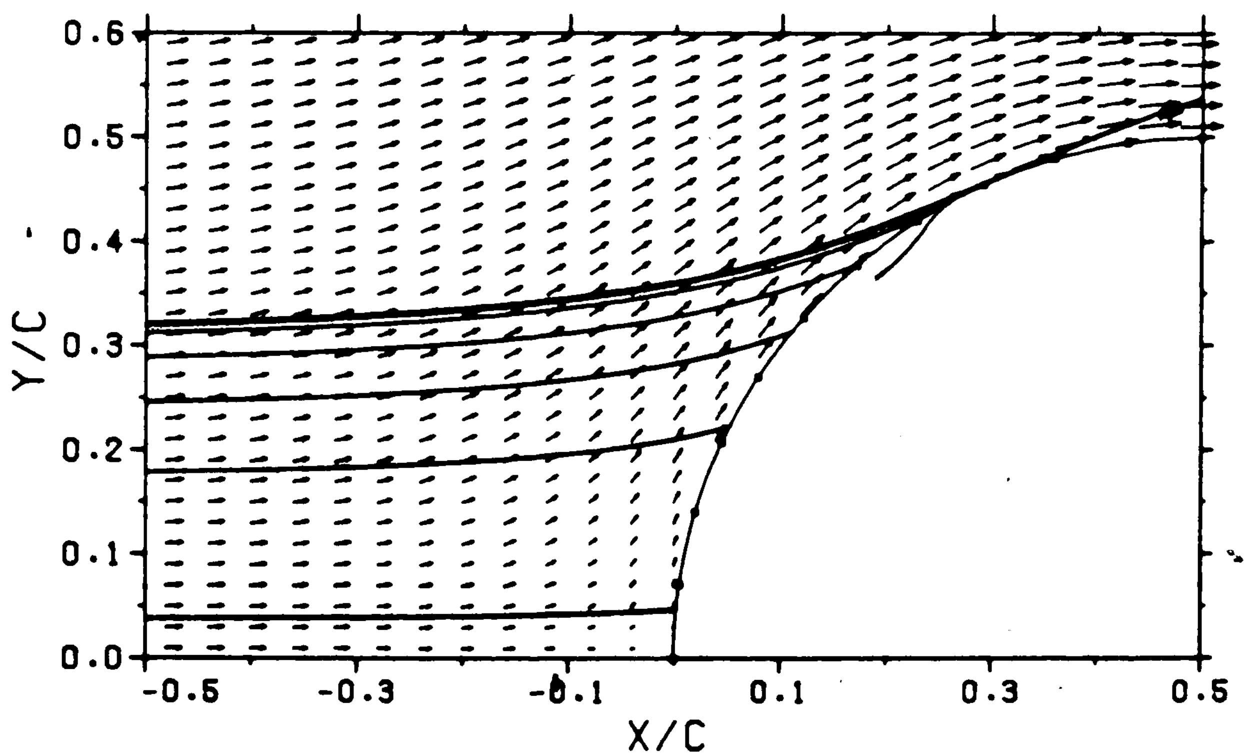

FIG. 27. The trajectories of droplets in a flow about a cylinder with the conditions of Case 15. Re_∞ =894.4 K=8

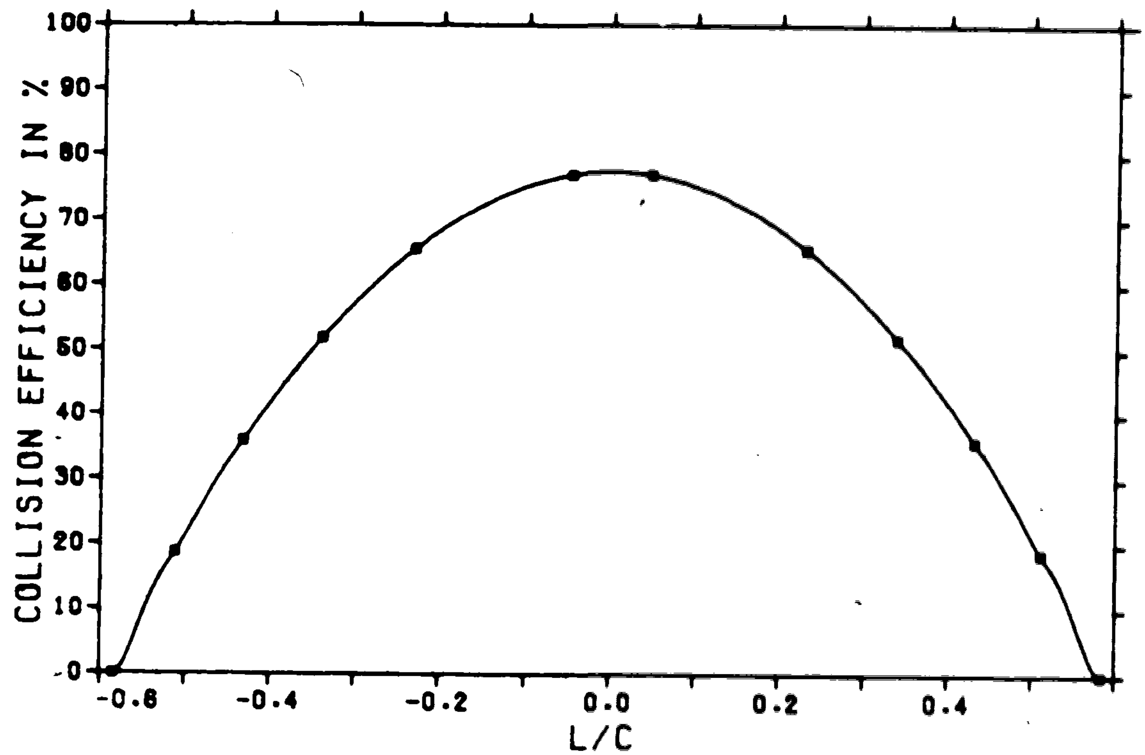

FIG. 28 The collision efficiency curve corresponding to the trajectories and conditions of Fig. 27 (Case 15).

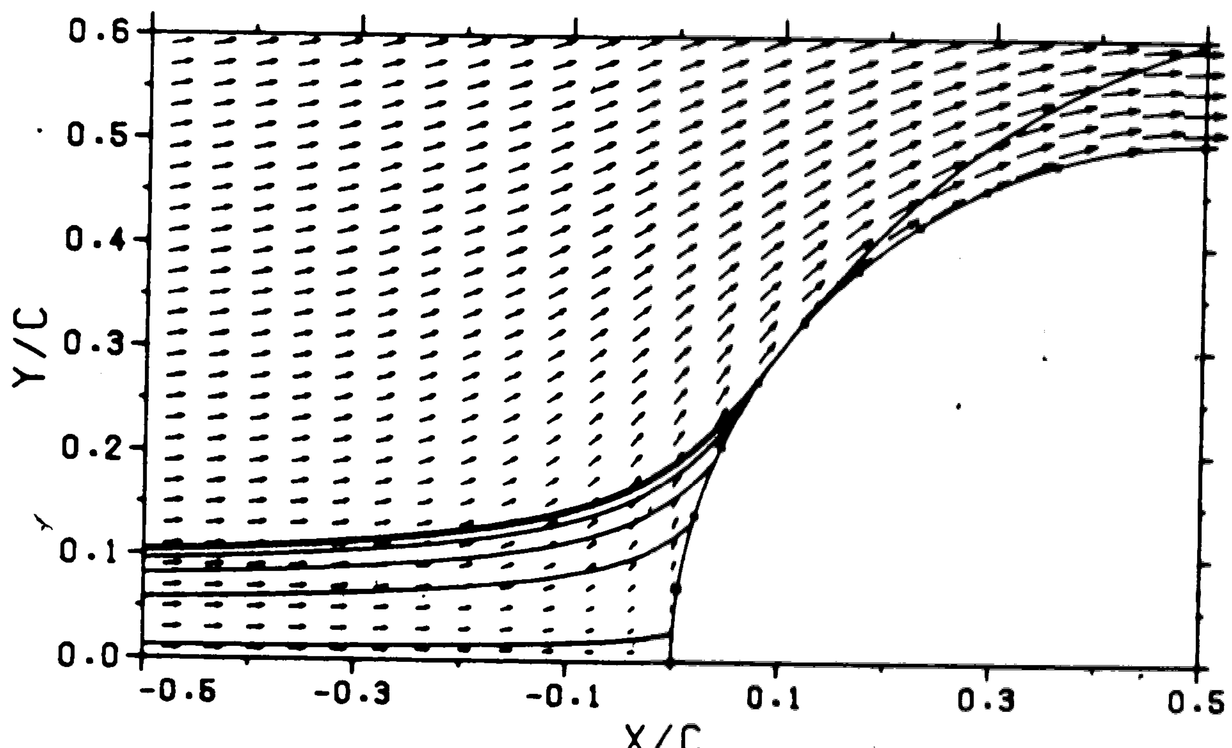

FIG 29 As for Fig. 28, but for Case 18 with $Re_\infty = 16$ K=0.3214

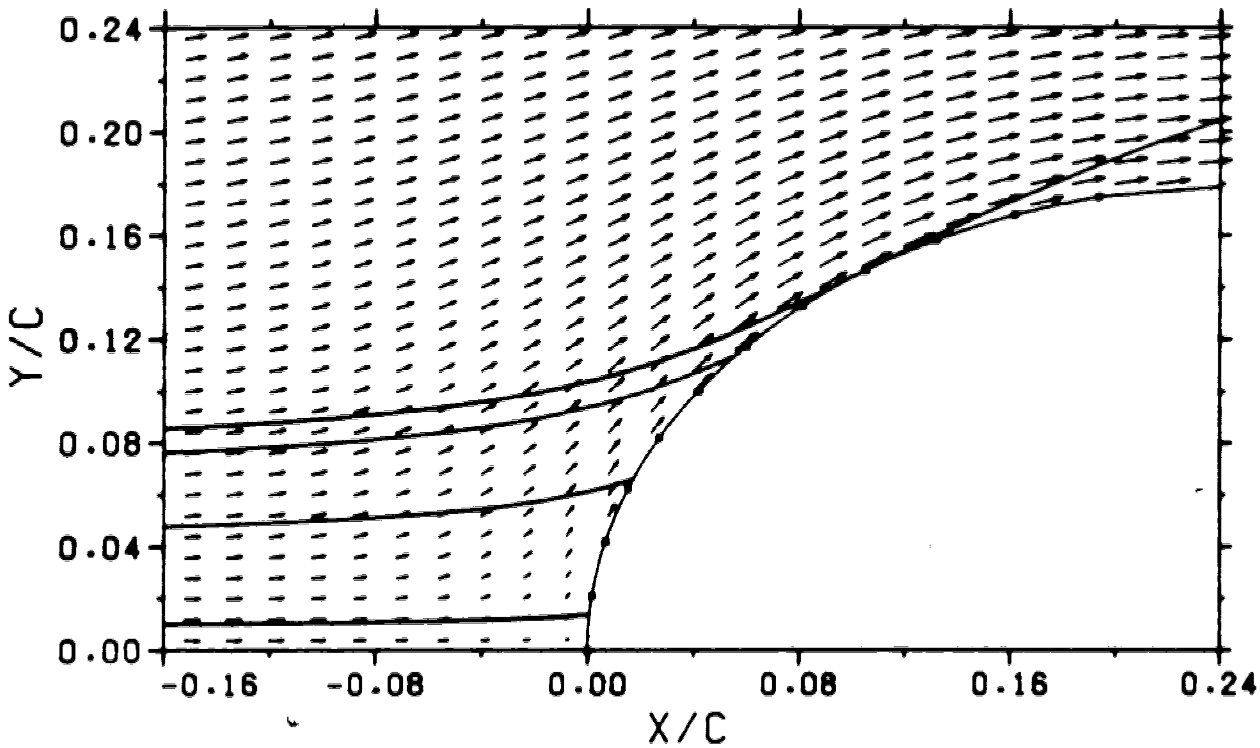

FIG. 30 The trajectories of droplets in a flow about a 36.5% thick Joukowski airfoil. The conditions are those of Case 25: $Re_\infty = 16$ $K=0.3214$

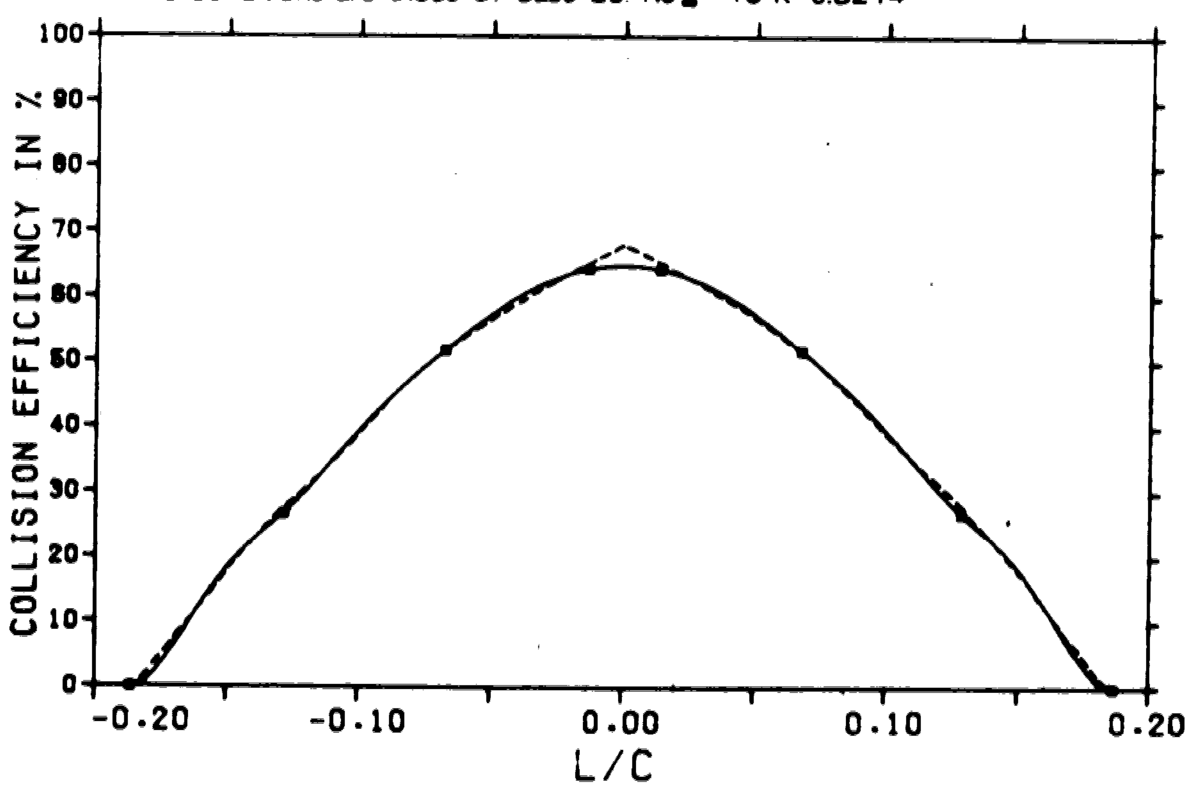

FIG. 31 The collision efficiency curve corresponding to the trajectories of Fig. 30 (Case 25) in solid. The dashed line is from the results of Brun & Voyt (1957).

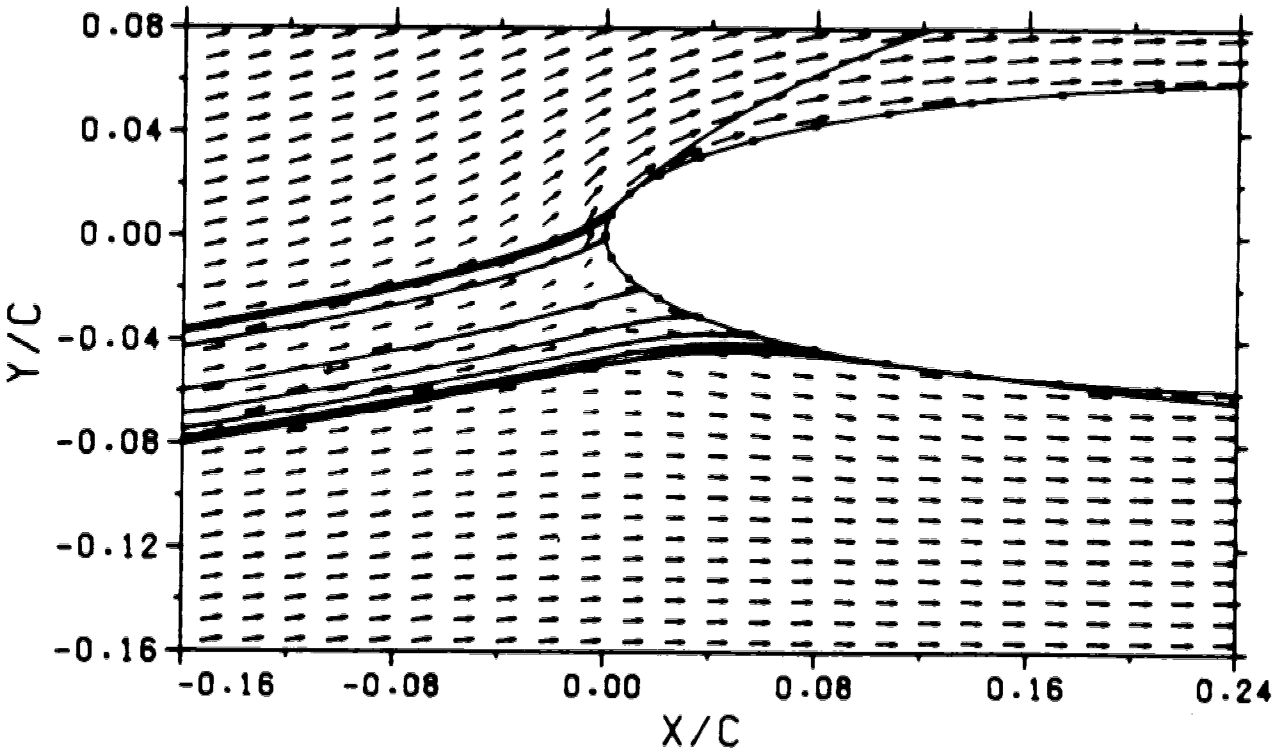

FIG. 32. The trajectories of droplets in a flow about a NACA 0015 airfoil. The conditions are those of Case 27. Re_∞ =202.2 K=0.238

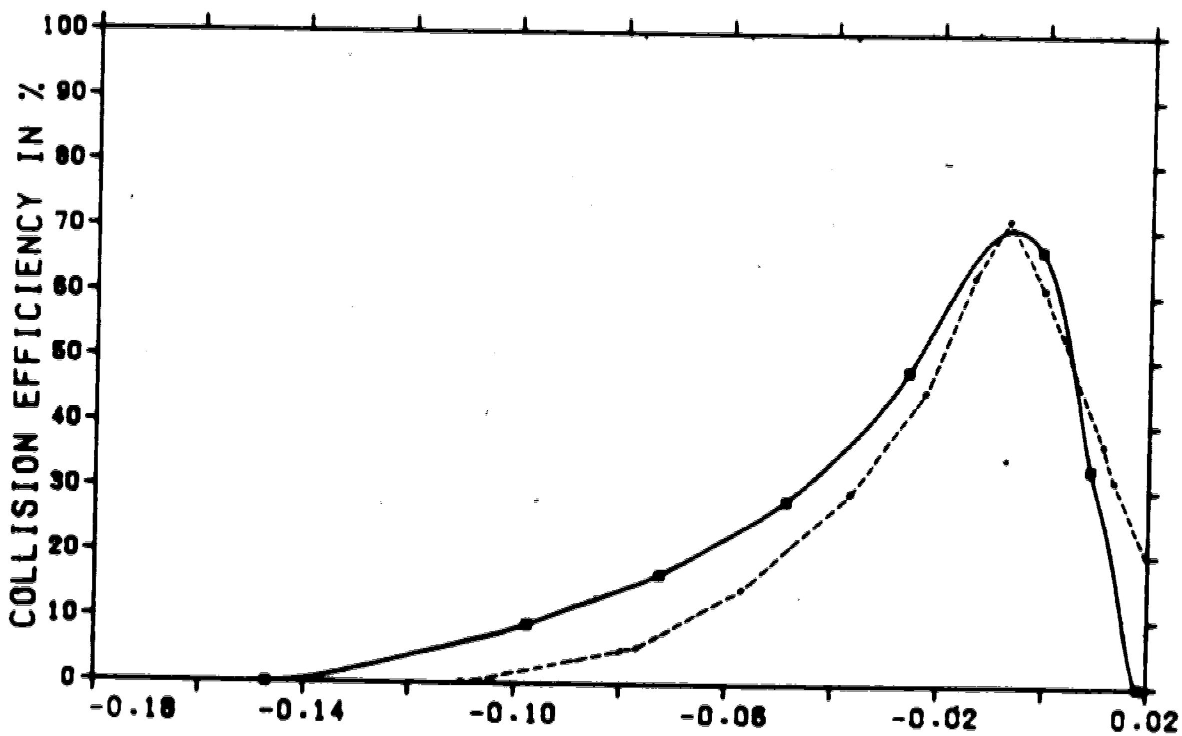

FIG. 33. The collision efficiency curve corresponding to the trajectories of Fig. 32 (Case 27) as a solid line. The dashed line displays the curve of Werner (1973).

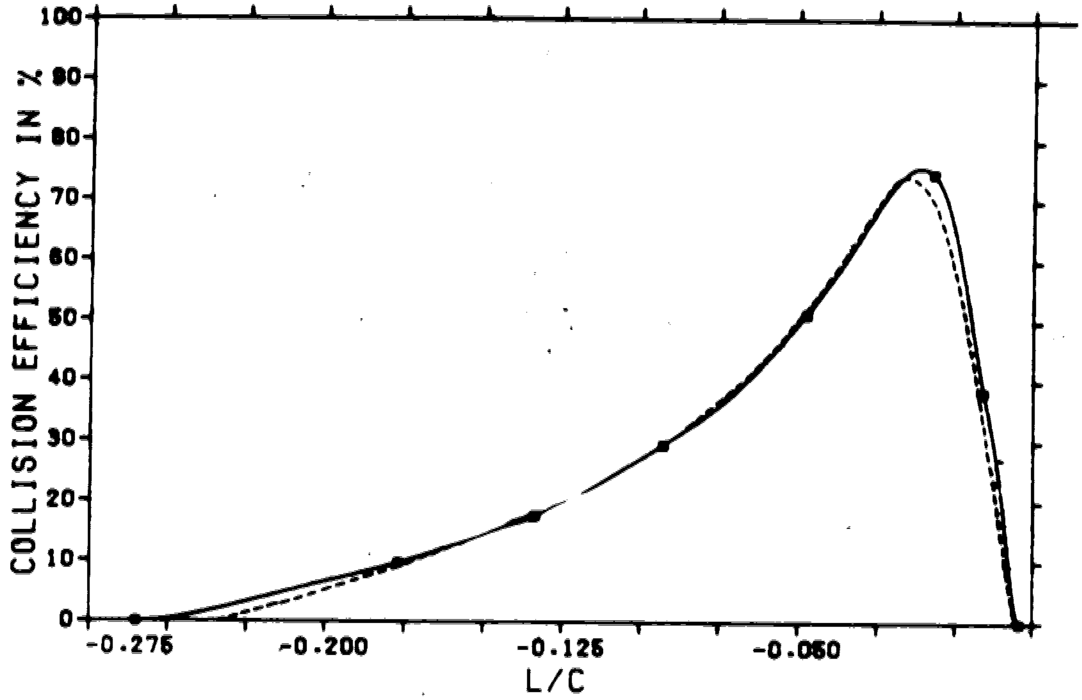

FIG. 34. The collision efficiency curve of Case 29 as a solid line. The dashed line corresponds to the results of Bragg (1981). Re_∞ =55 and K=0.257

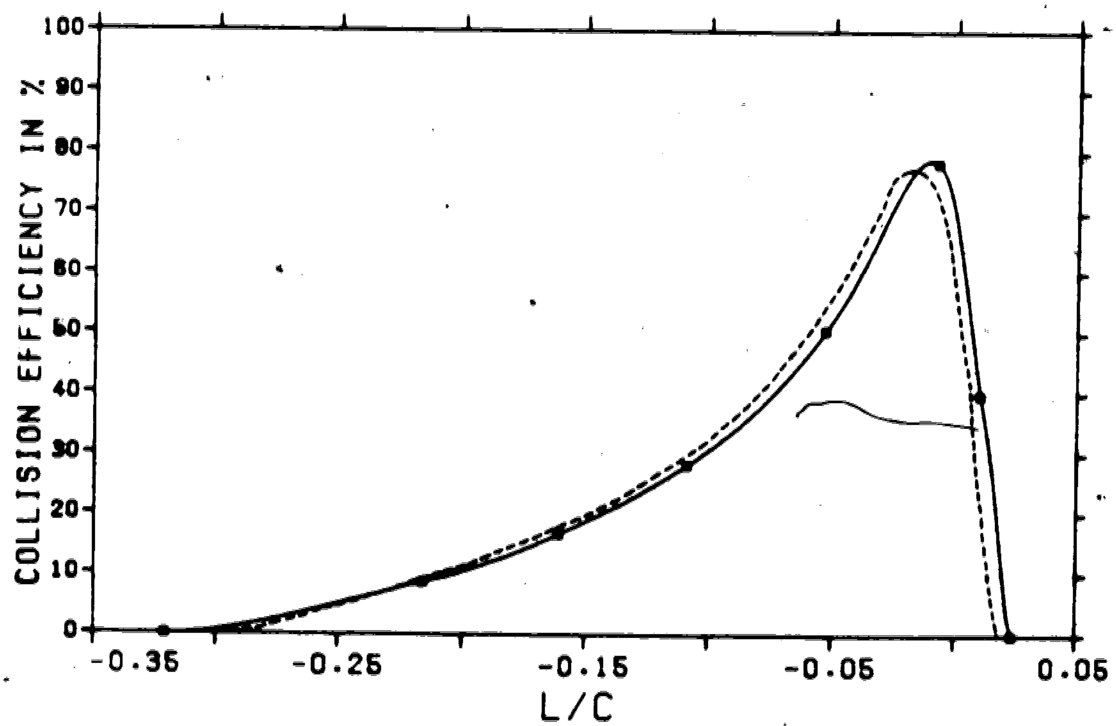

FIG. 35. As for Fig. 34, but with Re_∞ = 109 and K=0.407 (Case 30).

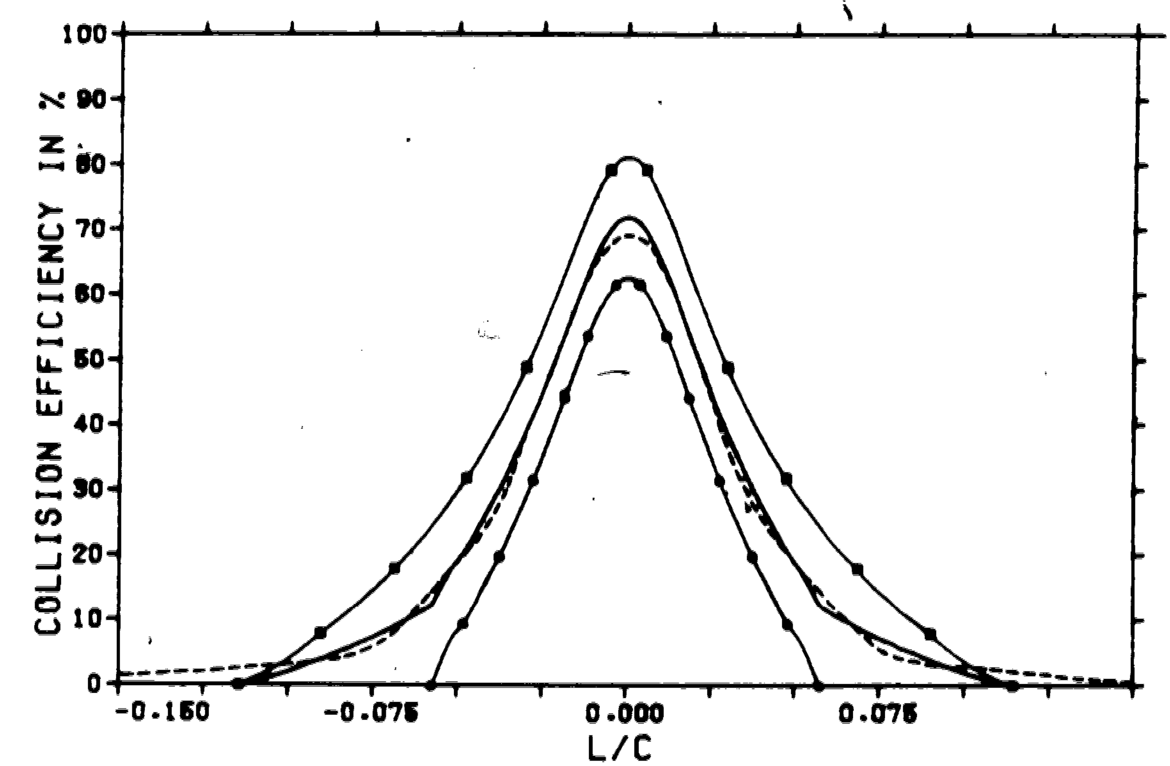

FIG. 36 The solid lines represent the collision efficiency curves for Case 31. The droplet diameters are 25.5 and 13.2 μm. The non-dimensional parameters for the MMD droplet (18.6 μm) are Re_∞=96.2 and K=0.257. The dashed line is the experimental result of Gelder *et al.* (1956).

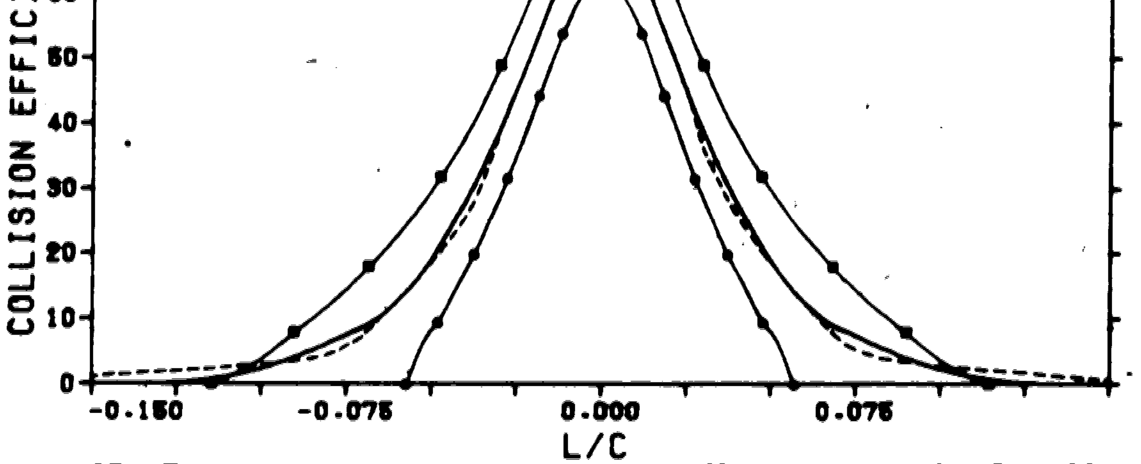

FIG. 37 The solid lines represent the collision efficiency curves for Case 32. All parameters remain the same as in Fig. 36, except that a variable length filter has been applied to smooth the mean curve. The dashed line gives the comparable result from Gelder *et al.* (1956).

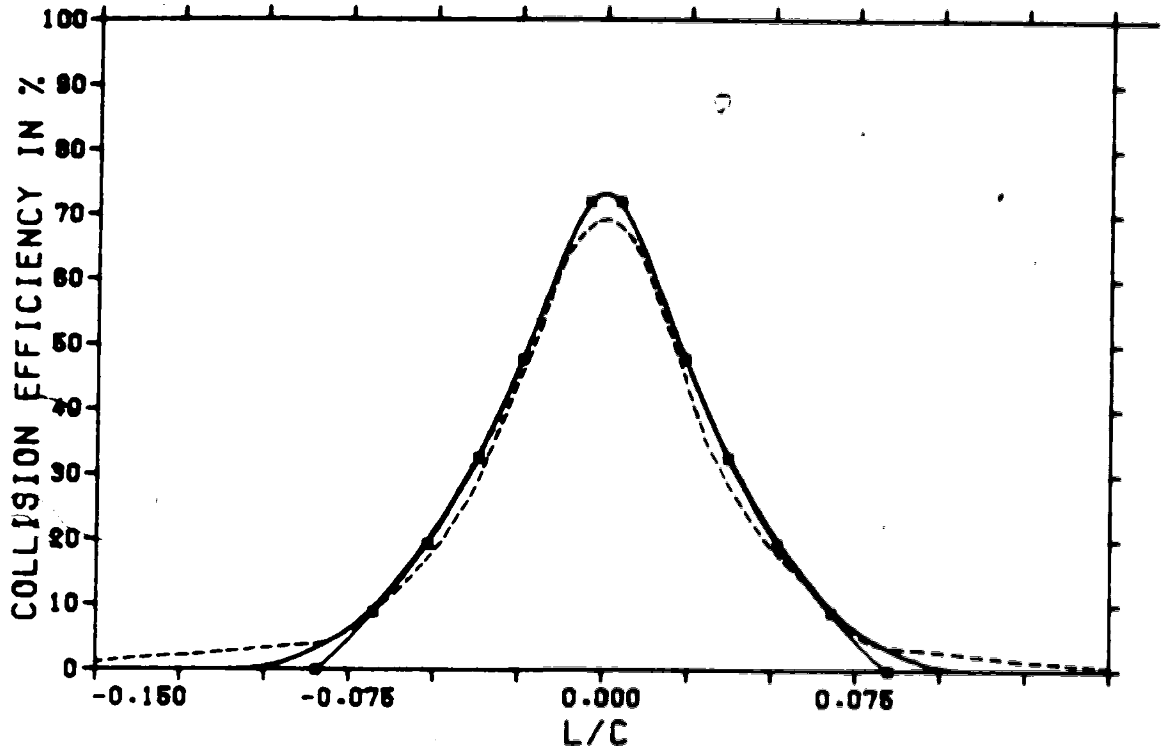

FIG. 38. The solid lines represent the collision efficiency curves for Case 33. The heavier line without symbols is once again the smoothed β curve. The dashed line is from Gelder *et al.* (1956).

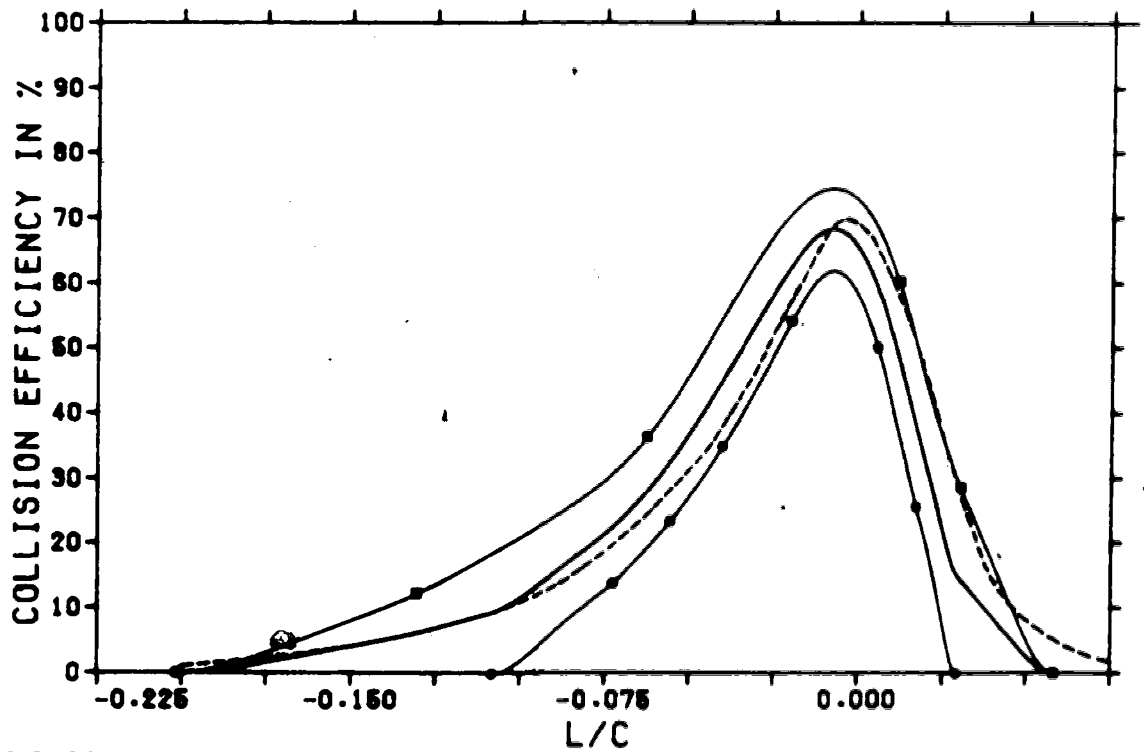

FIG. 39. The collision efficiency curves of Case 35 as solid lines. The heaviest line without symbols is the β curve for the droplet distribution used. The dashed line represents the results of Gelder *et al.* (1956). Re_{∞} =96.2 K=0.257

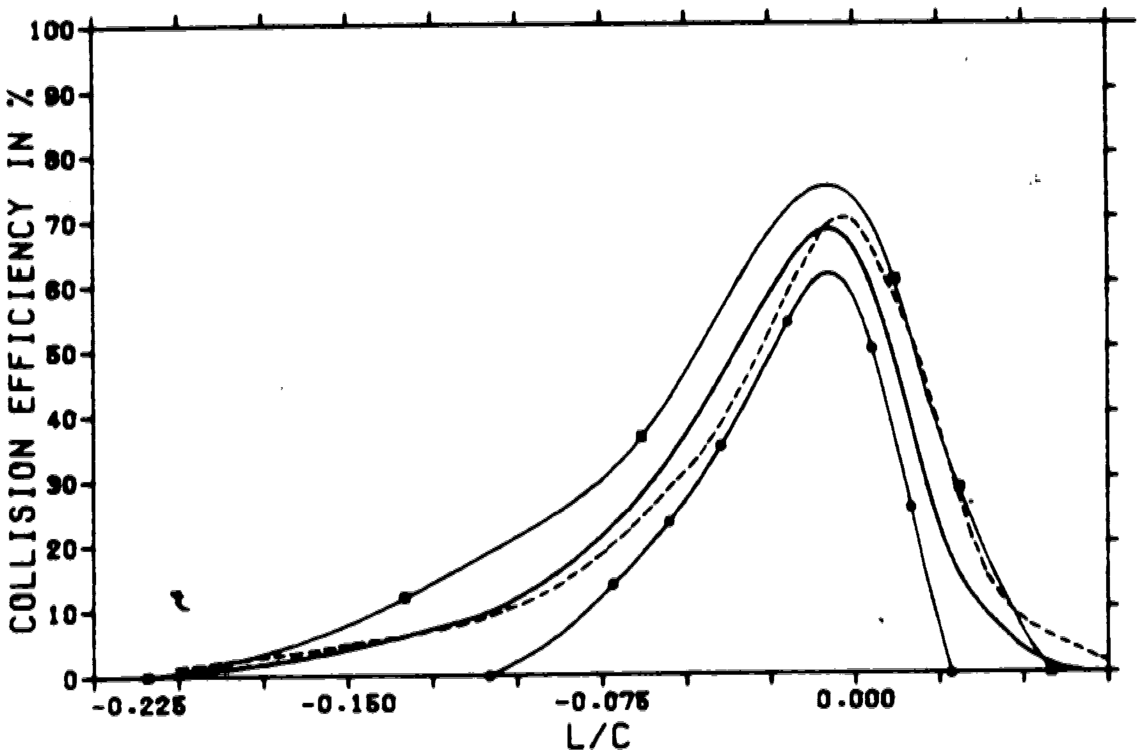

FIG 40 As in Fig. 39 except for Case 36.

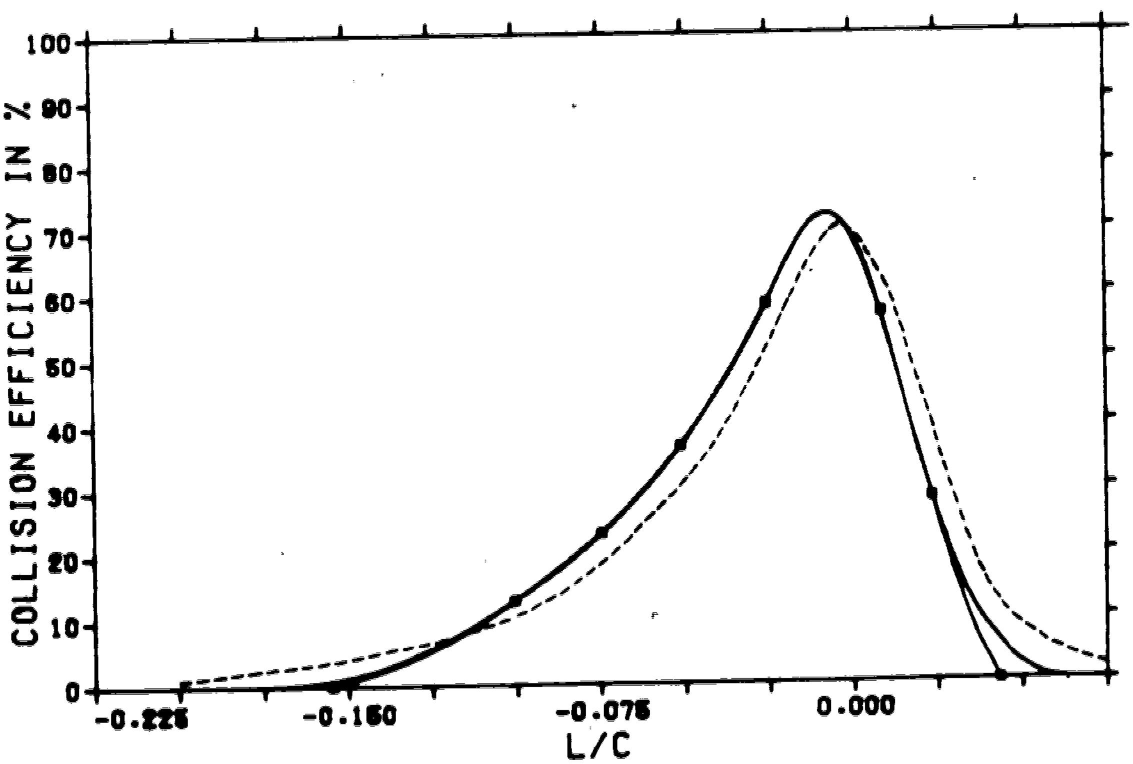

FIG 41. As in Fig. 39 except for Case 37.

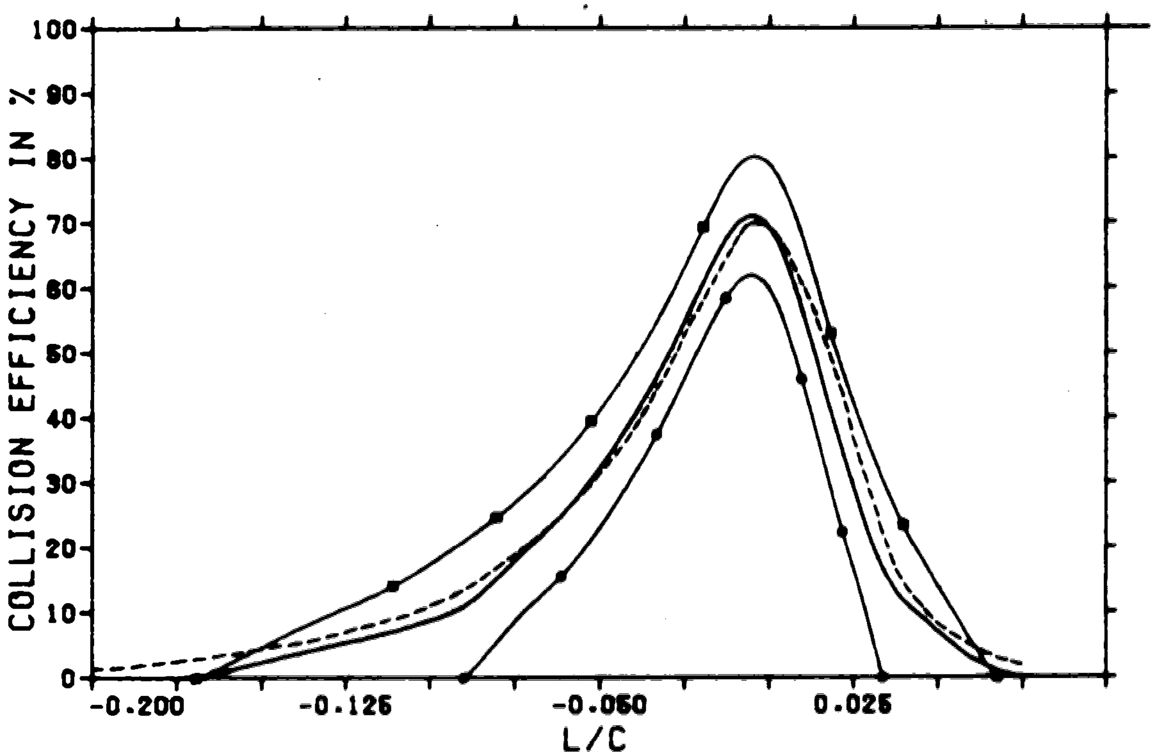

FIG. 42. As in Fig. 39 except for Case 38.

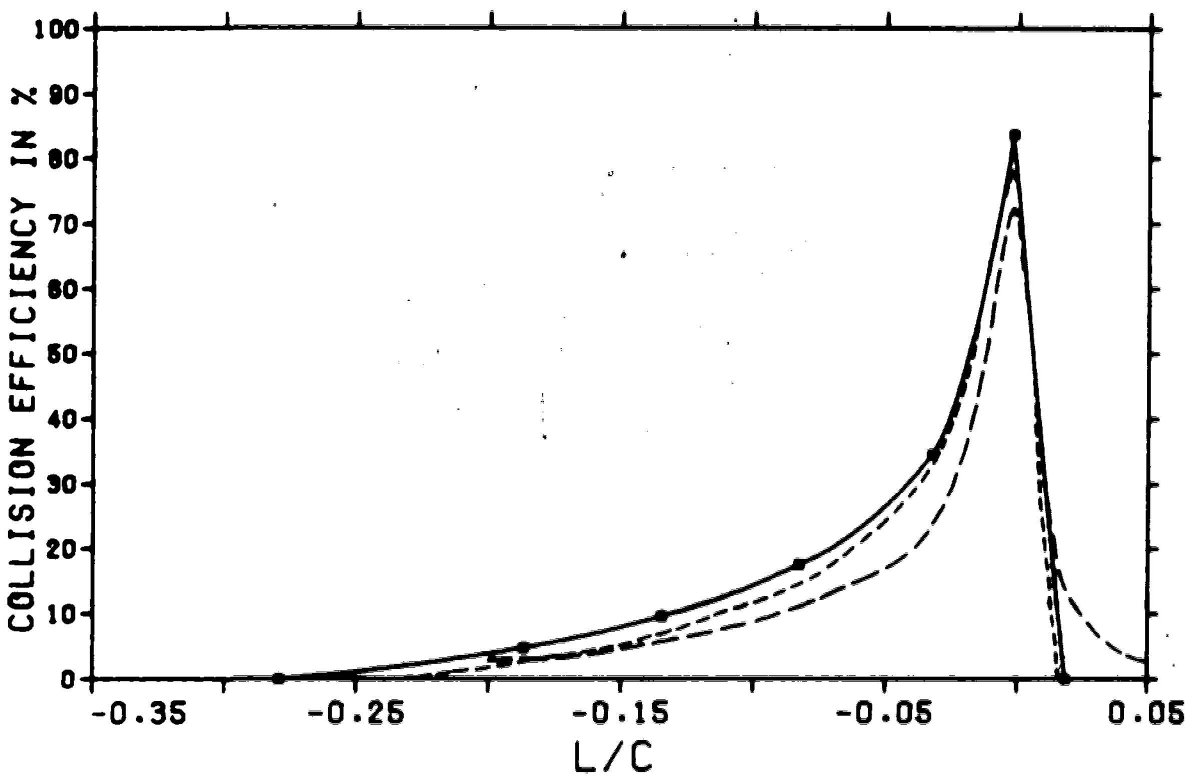

FIG. 43 The collision efficiency curves of Cases F (short dashes), G (long dashes), and 40 (solid line). Re_∞ =96.2 K=0.257

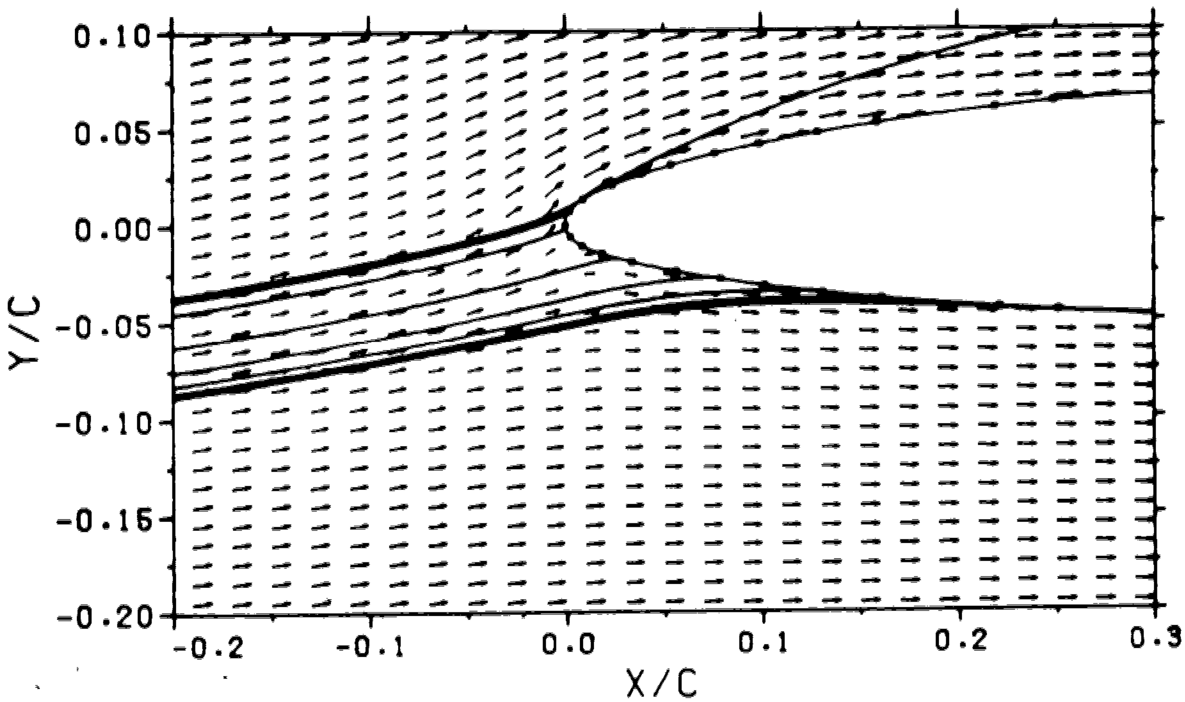

FIG 44 The trajectories of droplets in a flow about a NACA 65-212 airfoil. The conditions are those of Case 40.

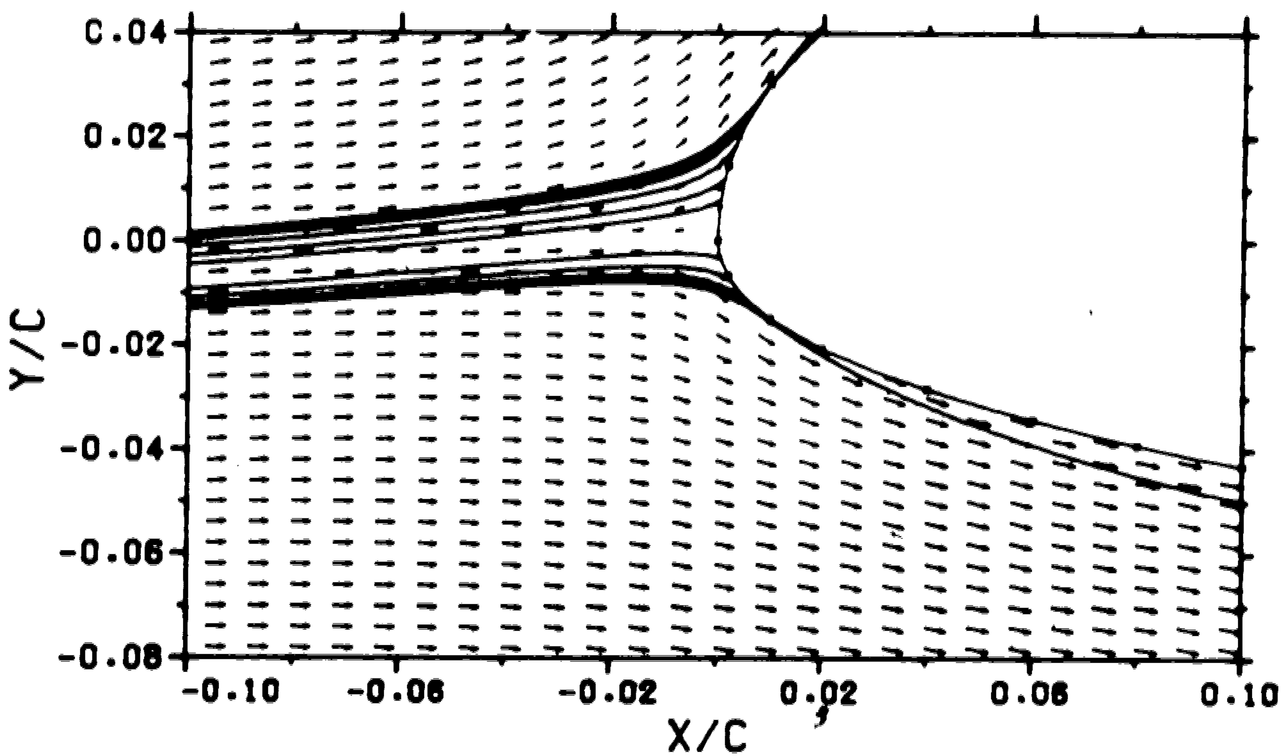

FIG 45 The trajectories of droplets in a flow about a NACA 64-215 Hick's modified airfoil. The conditions are those of Case 41. $Re_\infty = 113.9$ $K = 0.0436$

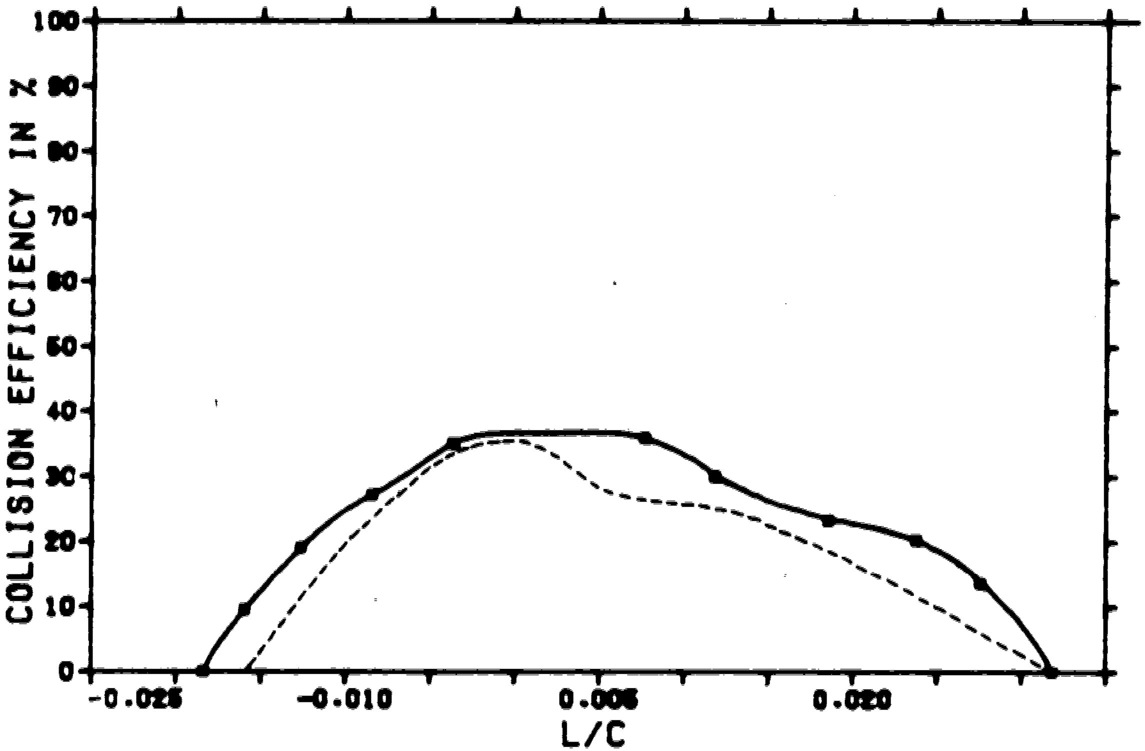

FIG. 46. The solid line represents the collision efficiency curve for Case 41. The dashed line is from the results of Bragg *et al.* (1981).

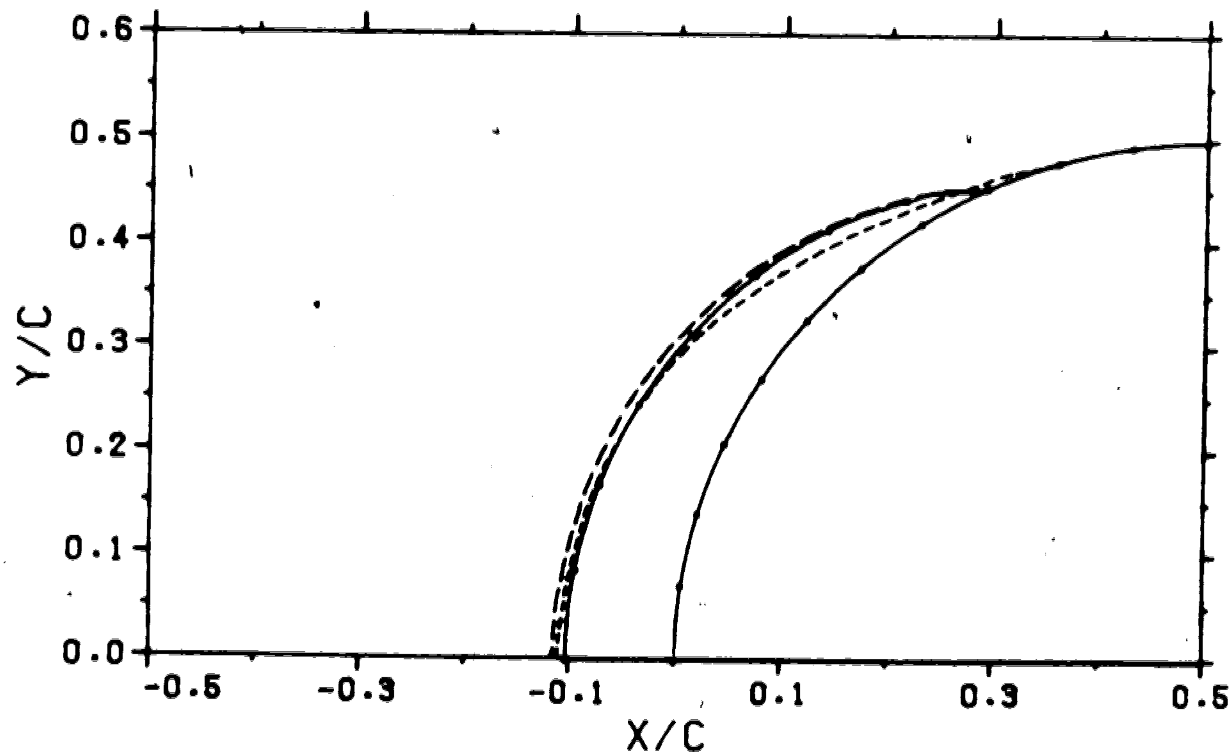

FIG. 47. The profile of an accreted layer on a cylinder. The solid line corresponds to Case 43 where surface curvature has been taken into account. The long dashed line shows Case 42 with the thickness calculated as if the substrate were locally flat. The short dashed line displays the experimental results of Lozowski *et al.* (1979). $Re_\infty = 49.0$ $K = 1.624$ $\omega = 0.157$

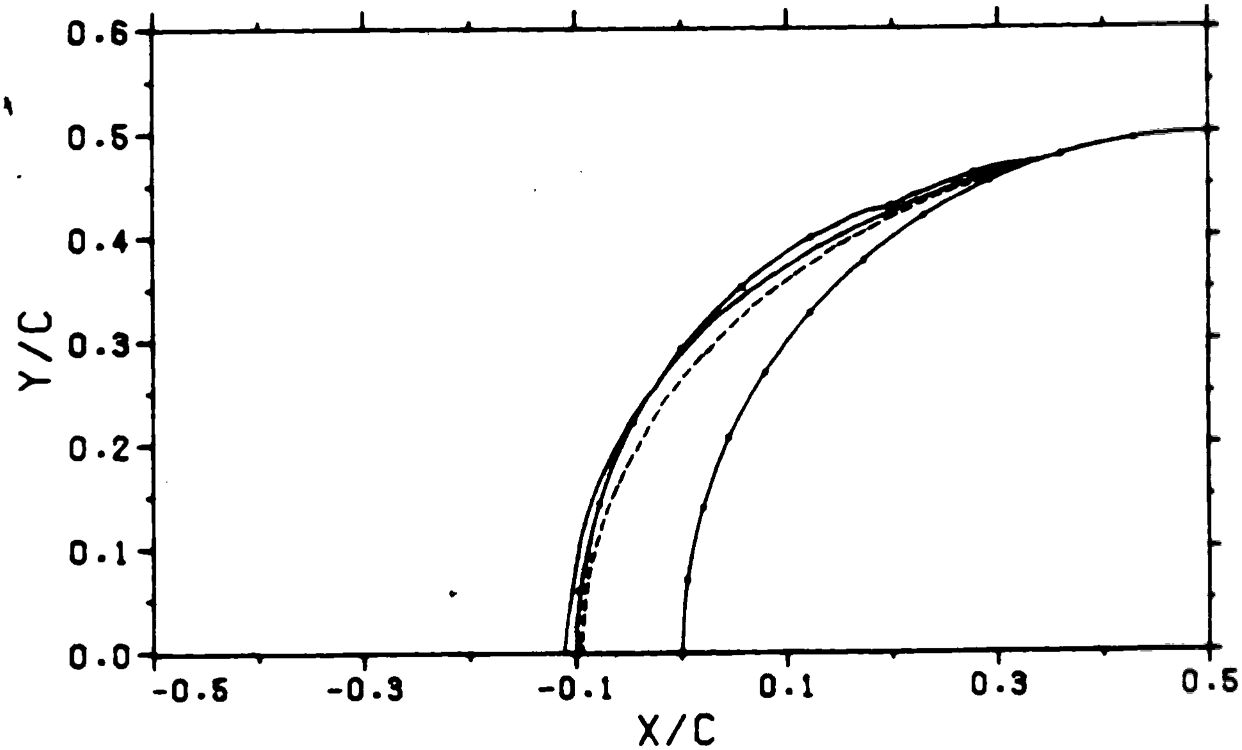

FIG. 48. The profile of an accreted layer on a cylinder. The solid line with symbols is for Case 44. The solid symbol-less line shows the profile of the experimental results of Lozowski *et al.* (1979). The dashed line is their theoretical prediction for the same conditions. Re_∞ =49.0 K=1.624 ω =0.157

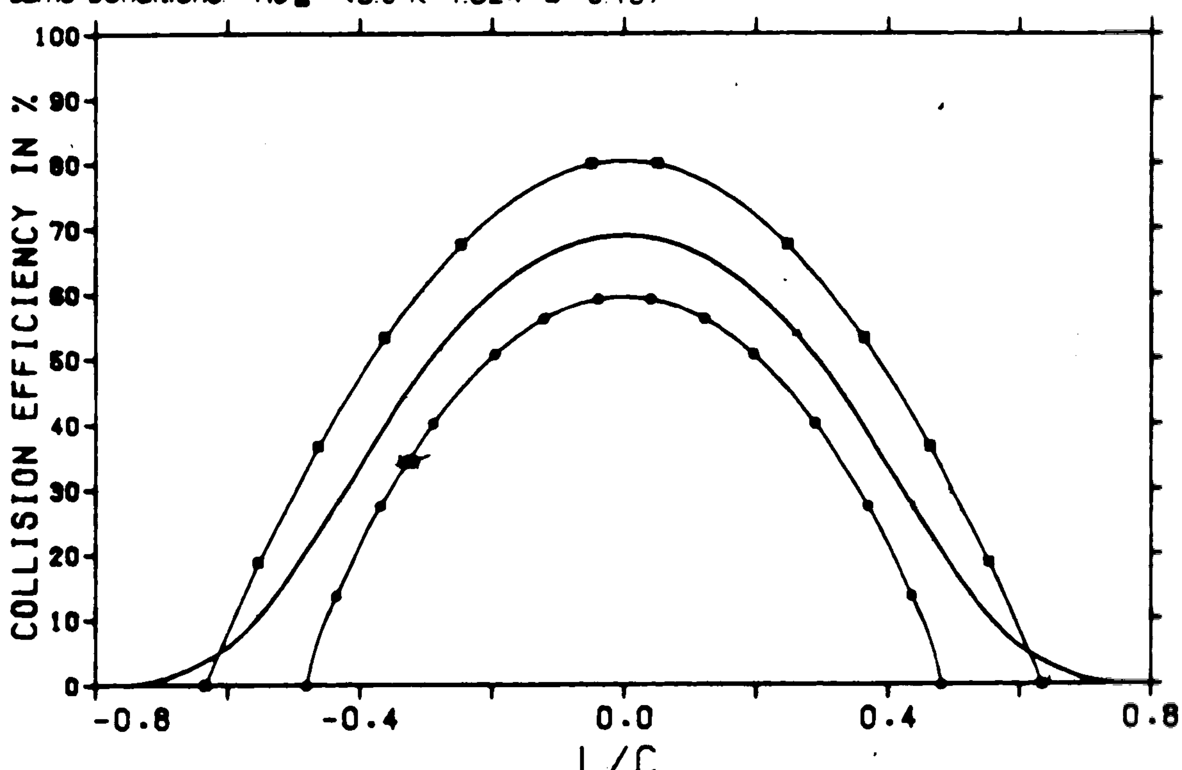

FIG. 49. The collision efficiency curves of Case 46 are displayed as solid lines with symbols (droplet diameters are 27.0 and 14.4 μm for the inner curve). The heavy solid line is the smoothed β curve.

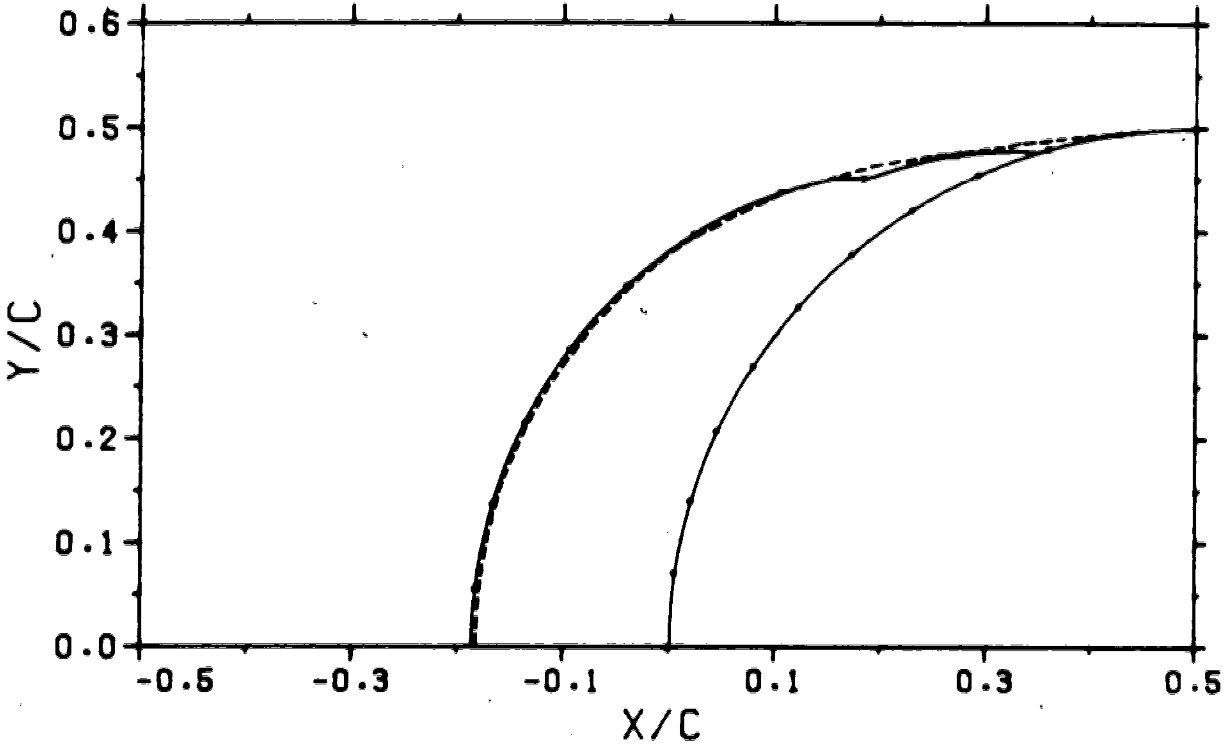

FIG. 50. Accretion on a cylinder. The accretion profile of Case 45 is shown as a solid line; the profile of Case 46 is dashed. LWC=0.8 g m⁻³ Re_∞ =49.0 K=1.624 ω =0.314

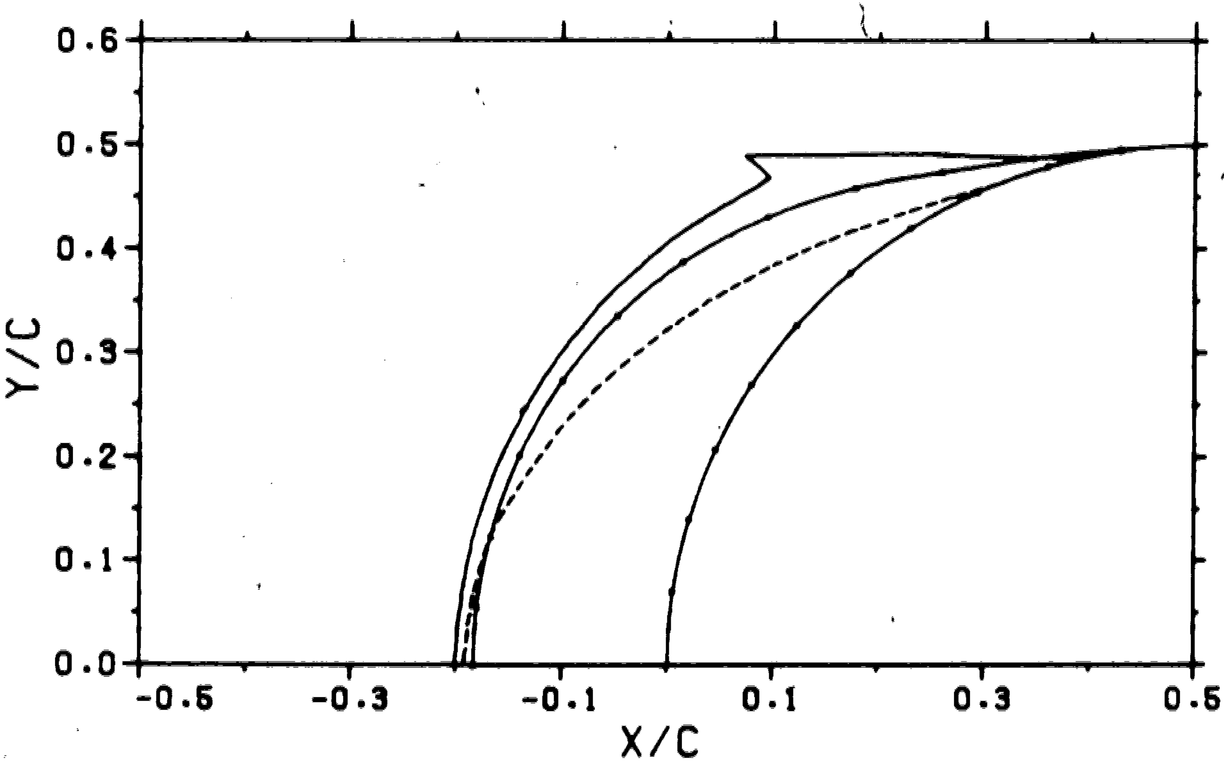

FIG. 51. The profile of an accreted layer on a cylinder. The solid line with symbols is for Case 46. The solid symbol-less line shows the profile of the experimental results of Lozowski *et al.* The dashed line is their theoretical prediction for the same conditions. LWC=0.8 g m^{-3} Re_∞ =49.0 K=1.624 ω =0.314

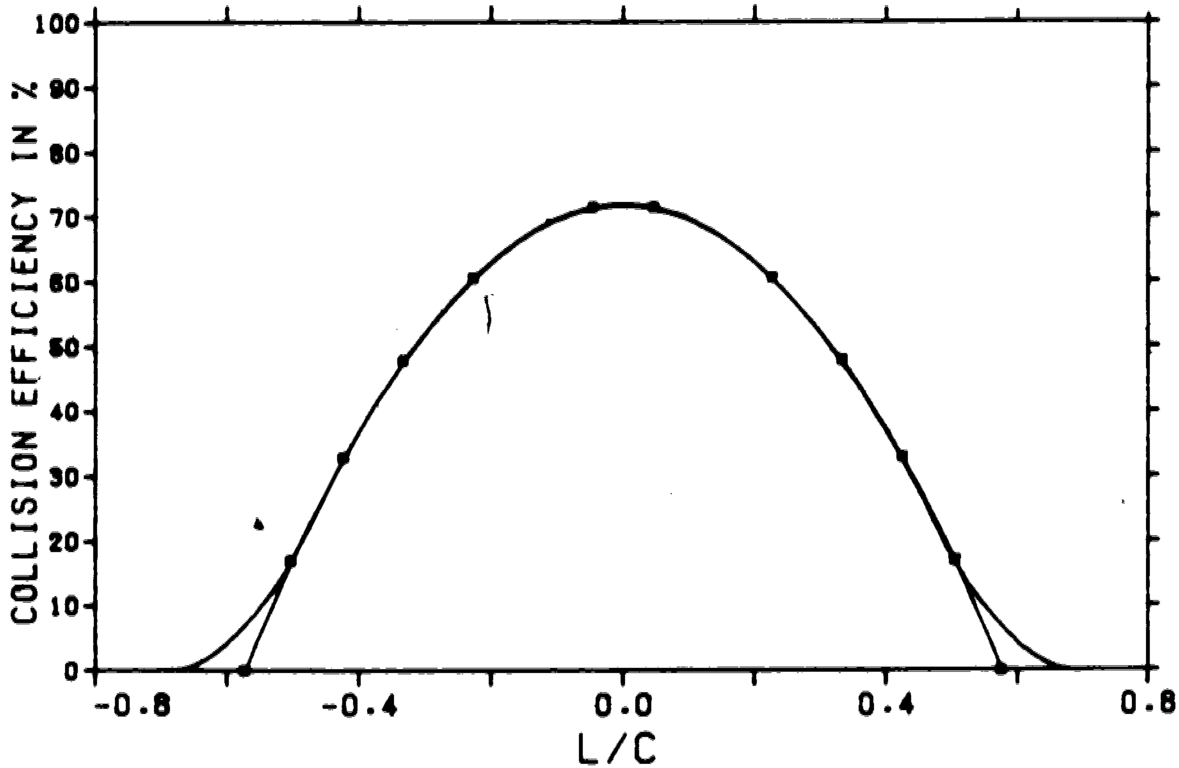

FIG. 52. The collision efficiency curves of Case 48. The heavy solid line without symbols is the filtered $\bar{\beta}$ curve for this case. Re_{∞} =49.0 K=1.624

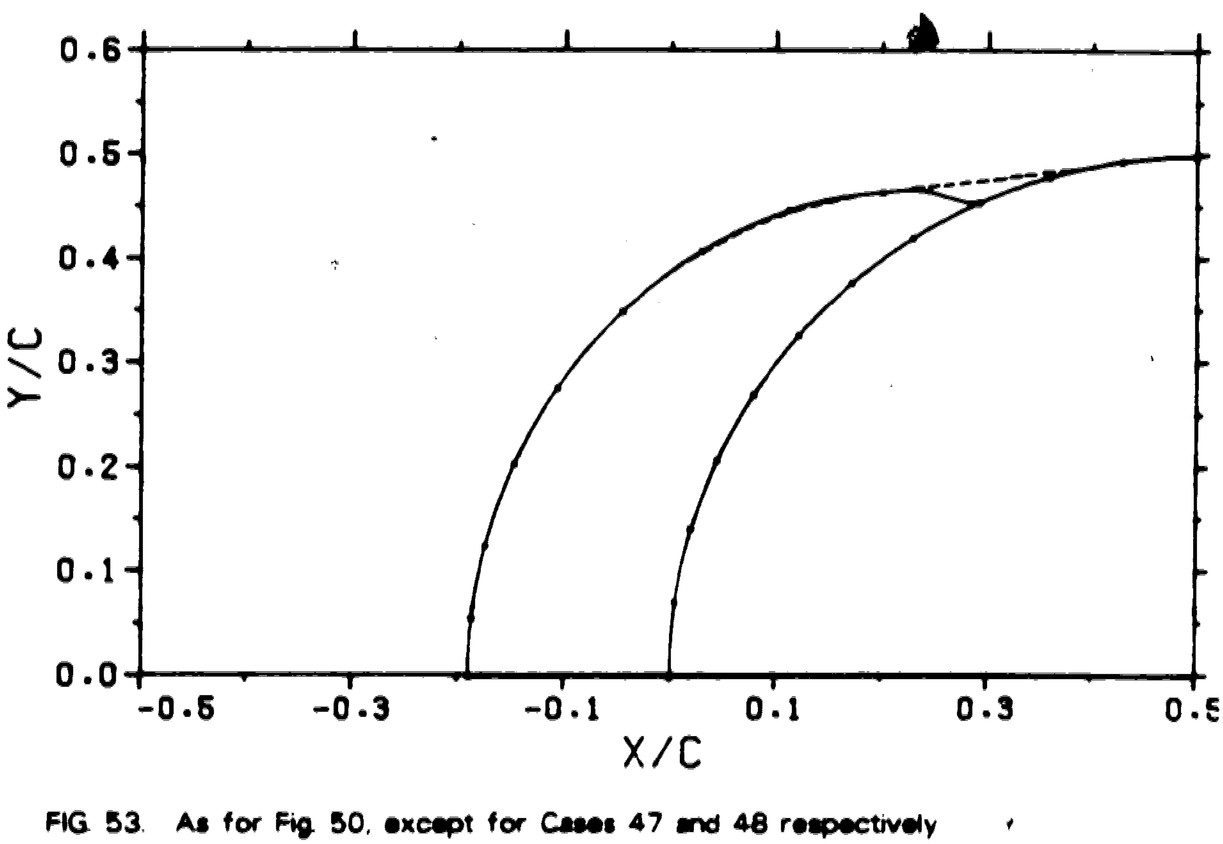

FIG. 53. As for Fig. 50, except for Cases 47 and 48 respectively

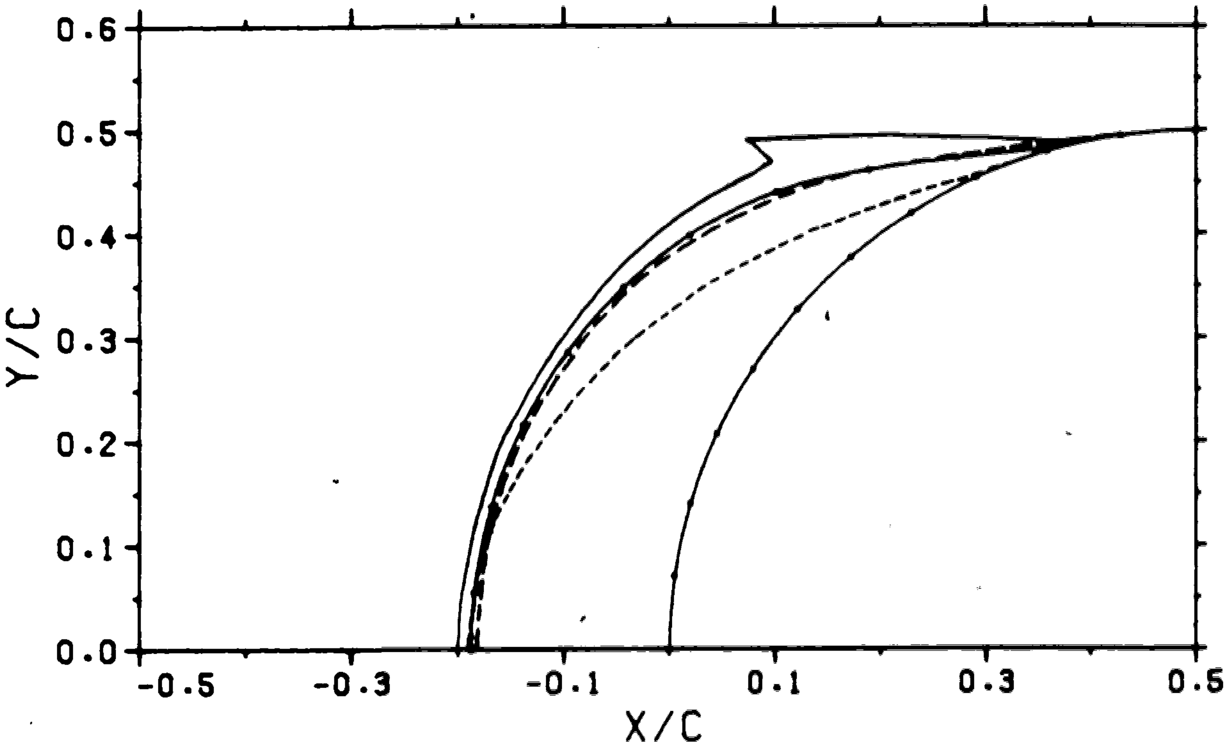

FIG. 54 The profiles of accreted layers on a cylinder. The solid line with symbols is for Case 48. The line of long dashes corresponds to Case 46 for two categories of droplet sizes. The solid symbol-less line is the experimental result of Lozowski *et al.* (1979). The short dashed line is their corresponding theoretical curve.

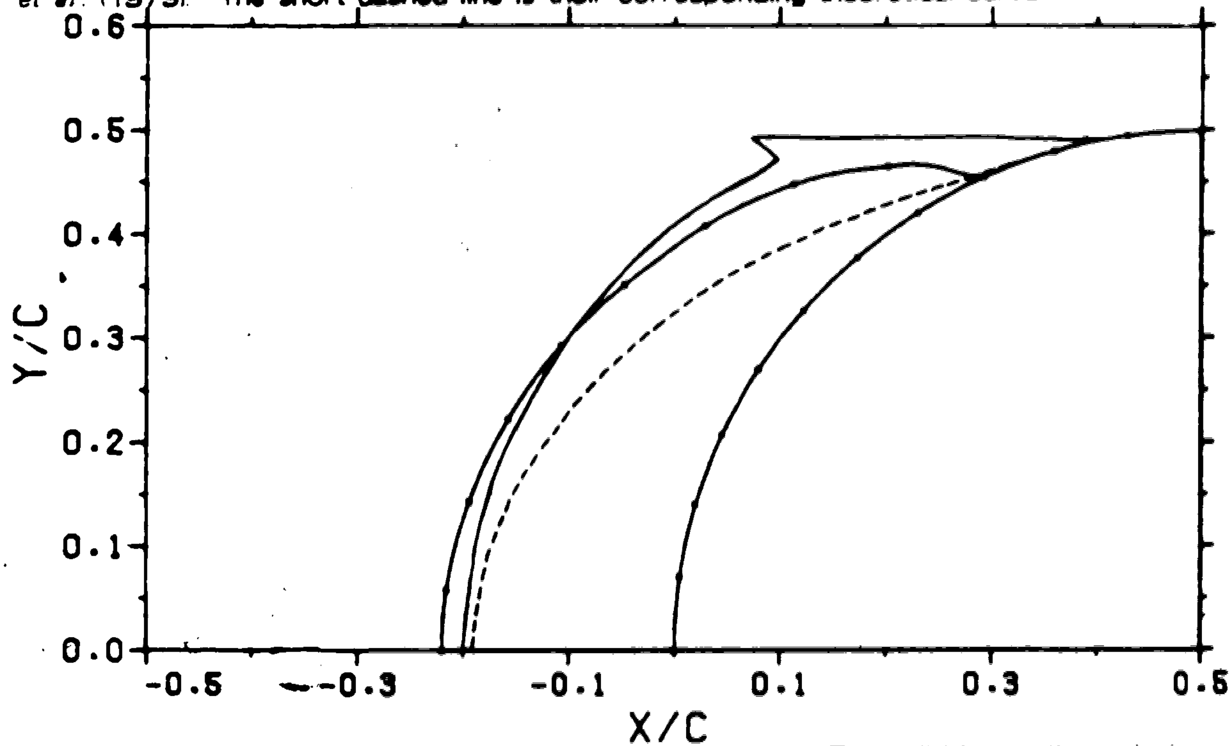

FIG. 55 The profile of an accreted layer on a cylinder. The solid line with symbols represents Case 49. The solid symbol-less line is for the experimental results of Lozowski *et al.* (1979). The dashed line is their theoretical prediction for the same conditions.

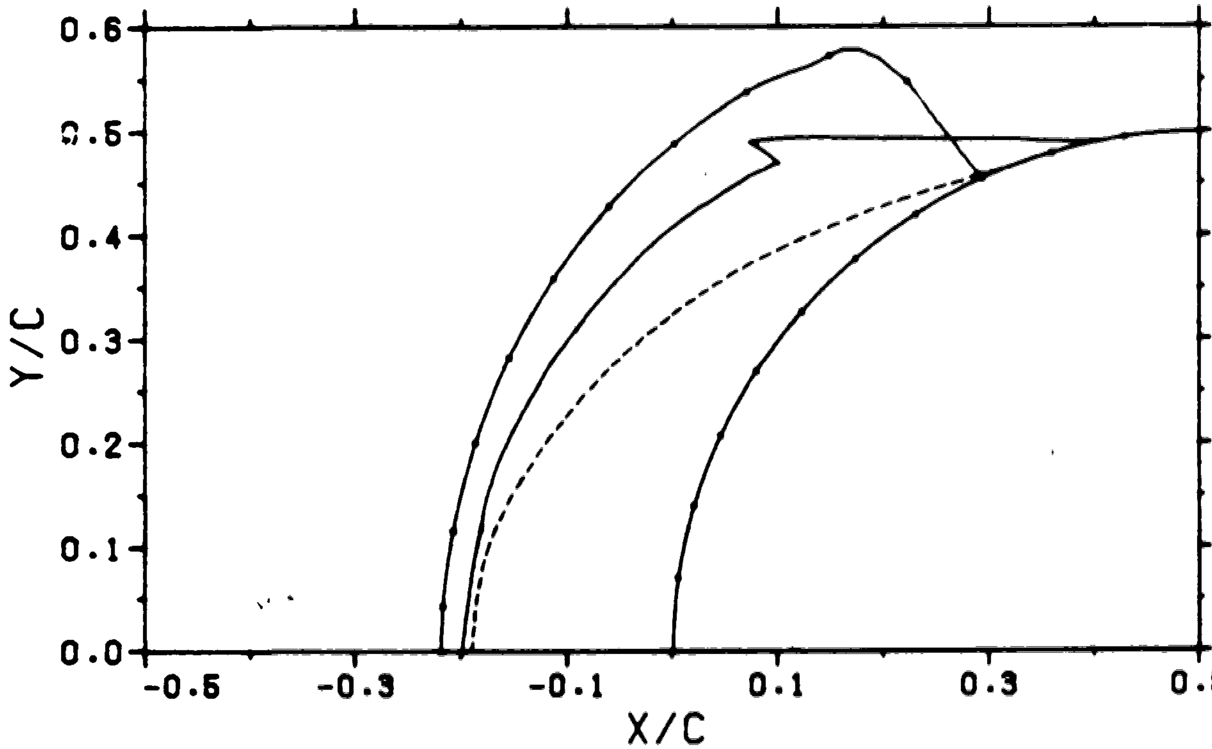

FIG. 56 As in Fig. 55, but for Case 50

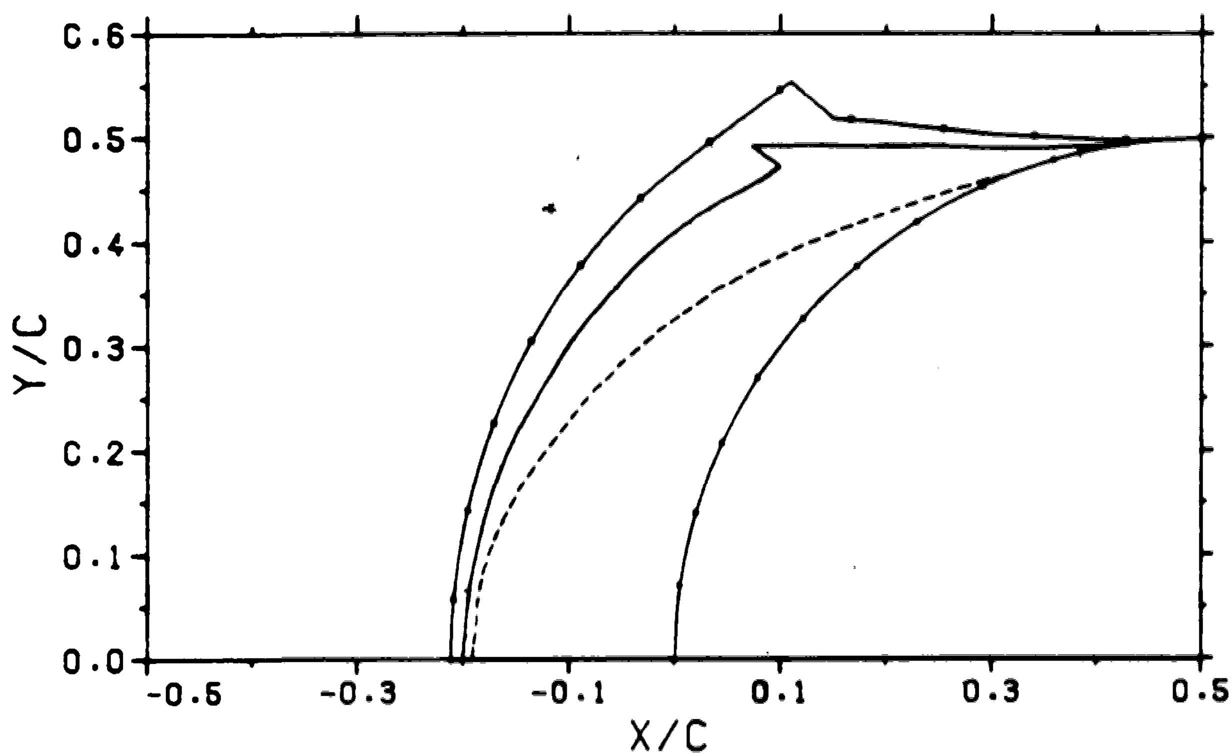

FIG. 57. As in Fig. 55, but for Case 51

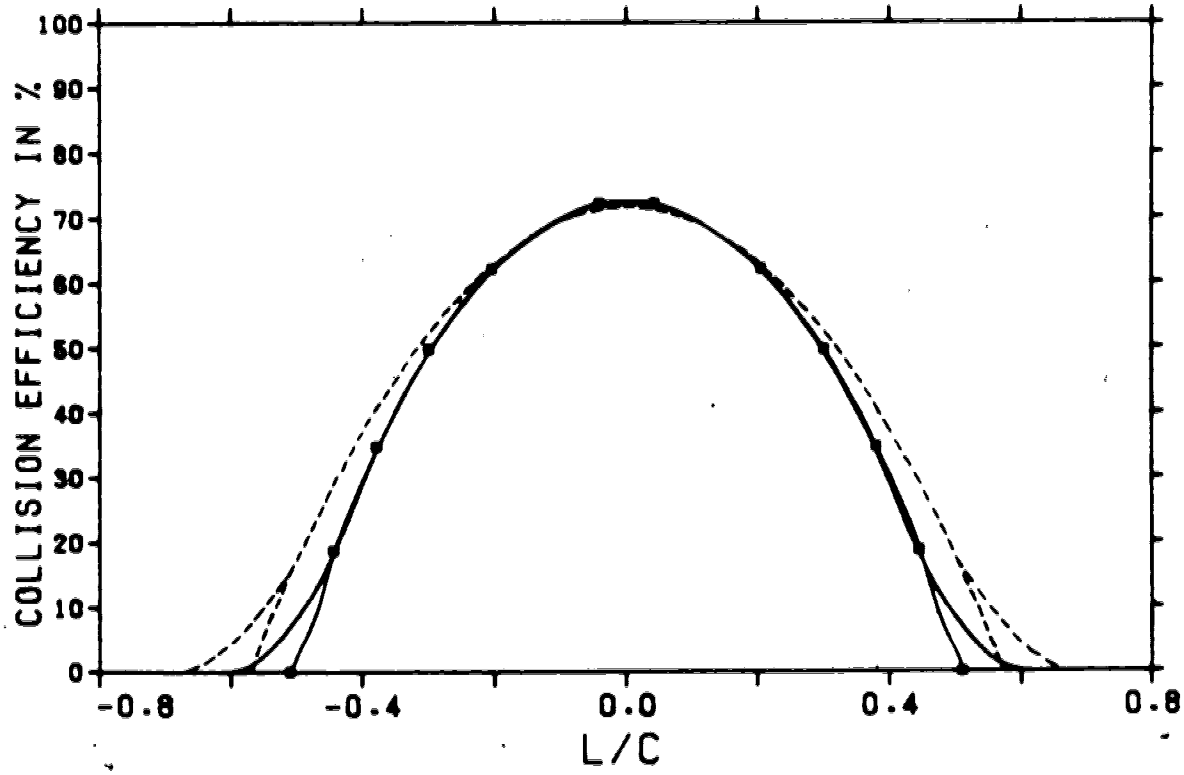

FIG 58. The collision efficiency curves for Case 52. The solid lines represent layer 3 - unfiltered (with symbols) and filtered (without symbols). The two dashed lines are the unfiltered and filtered curves for layer 1

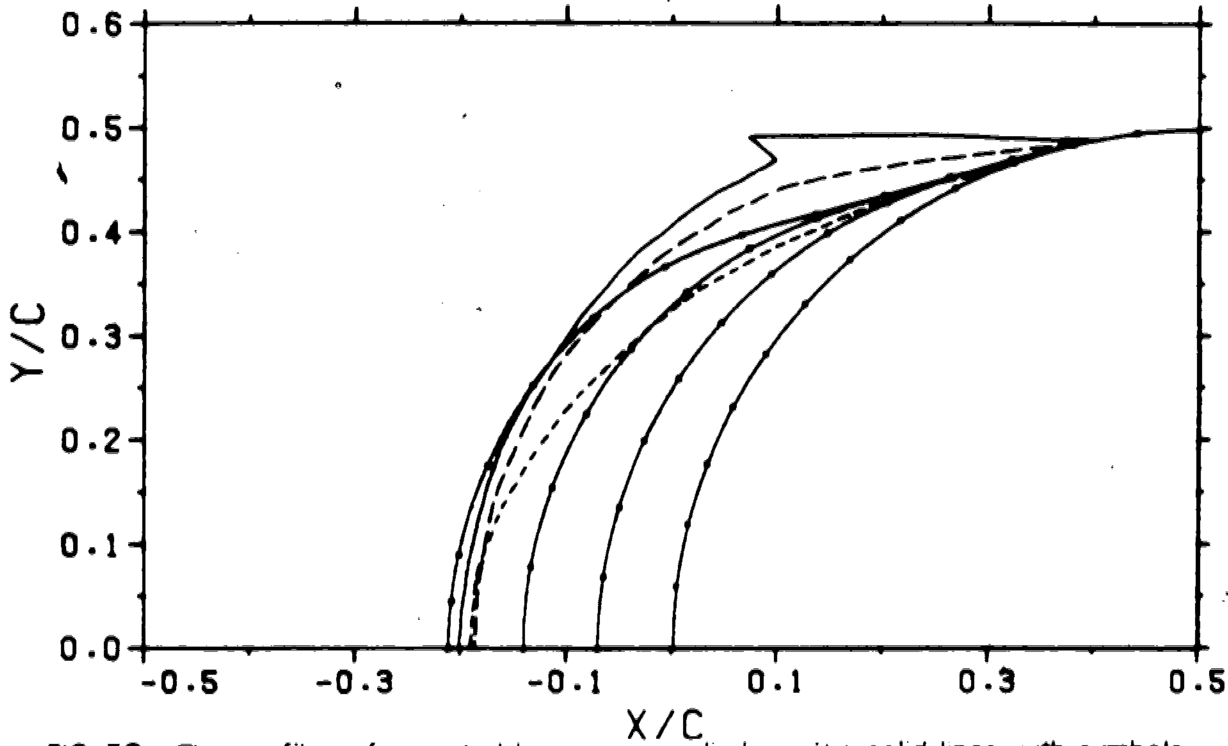

FIG 59. The profiles of accreted layers on a cylinder. The solid lines with symbols display the profiles of the three layers of Case 52. The solid symbol-less line is the experimental result, and the short dashed line, the theoretical result of Lozowski *et al.* (1979) for the same conditions. The long dashed line corresponds to Case 48, that is, for a single layer. Re_∞ =49.0 K= 1.624 ω =0.1047

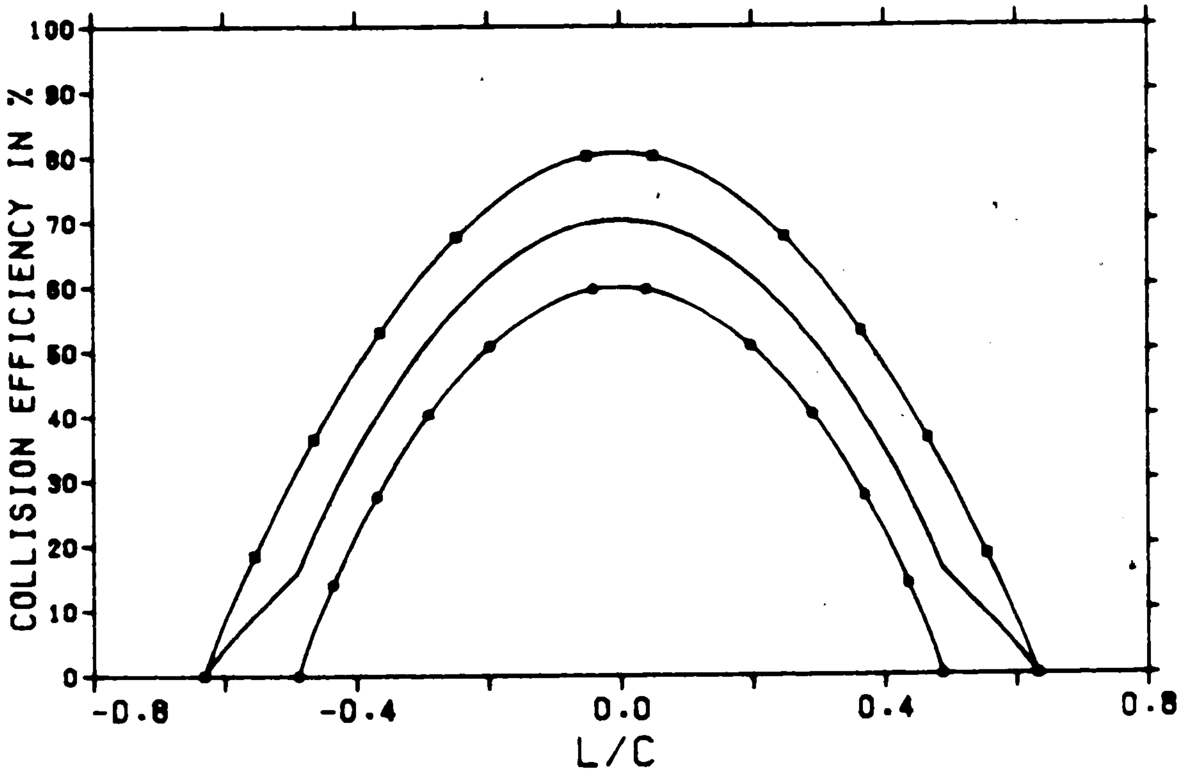

FIG. 60 The collision efficiency curves for layer 1 of Case 53. The outer and inner solid lines with symbols are the β curves for the 27.0 and 14.4 μm droplets respectively. The solid symbol-less line is the unsmoothed β curve. Re_∞ =49.0 K=1.624 ω=0.1047.

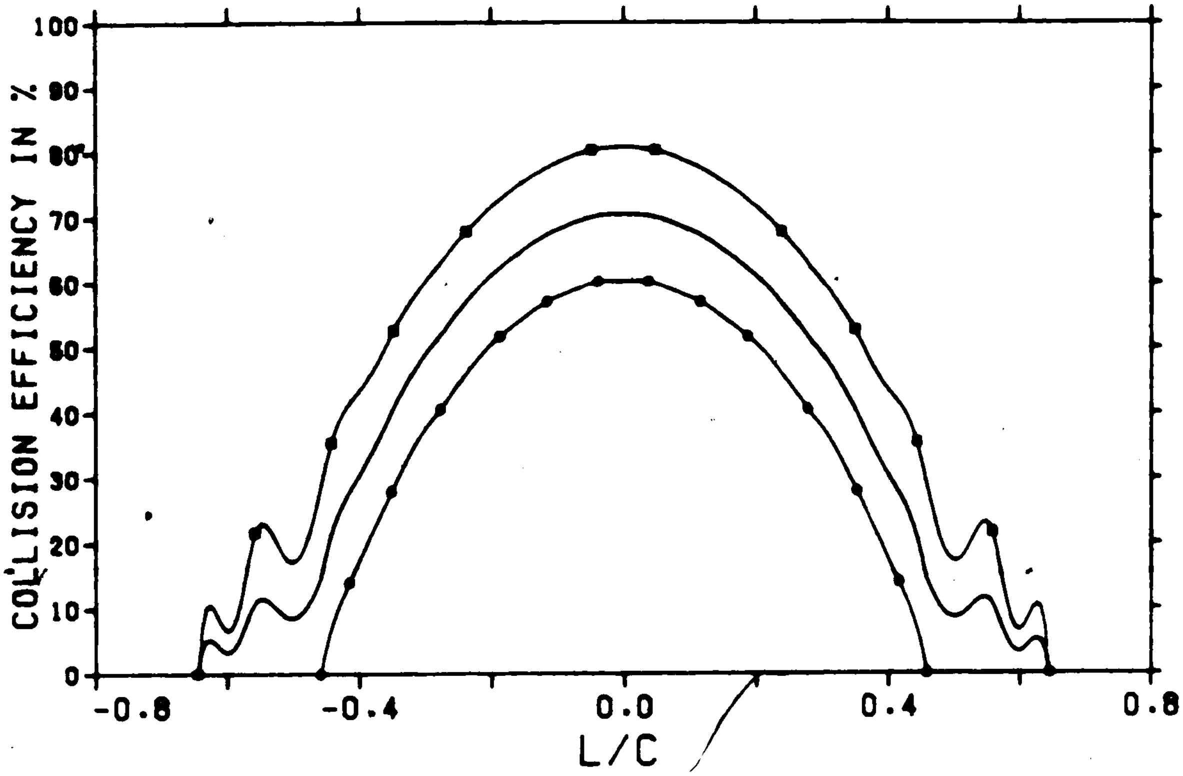

FIG. 61 As for Fig. 60, but for layer 2.

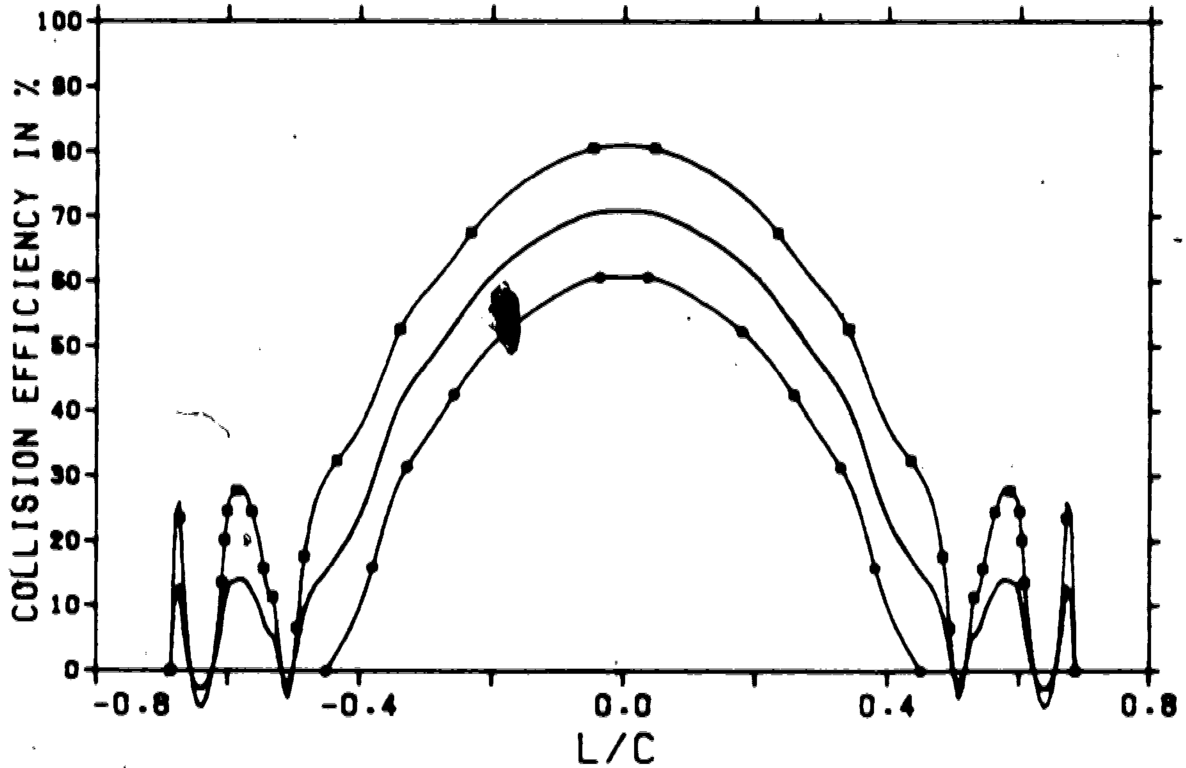

FIG. 62. As for Fig. 60, but for layer 3.

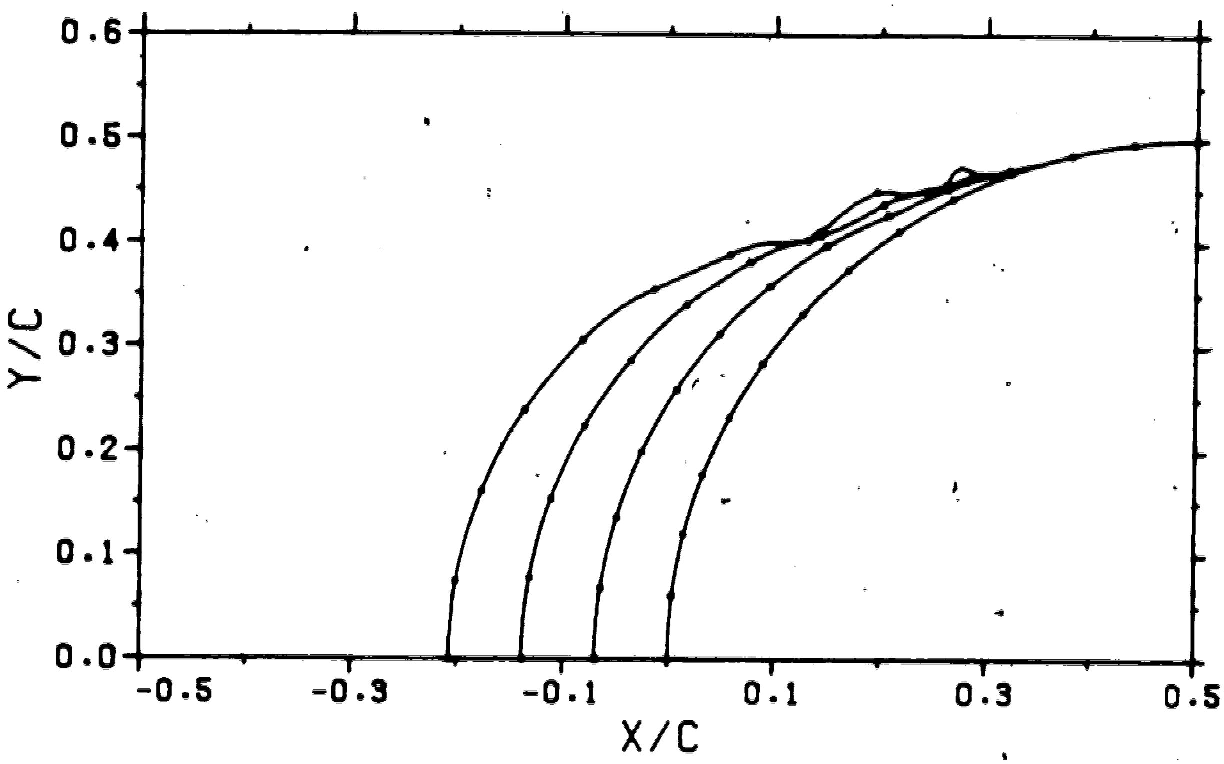

FIG. 63. The profiles of the three layers of accretion on a cylinder in Case 53. Re_{∞} =49.0 K=1.624 ω =0.1047

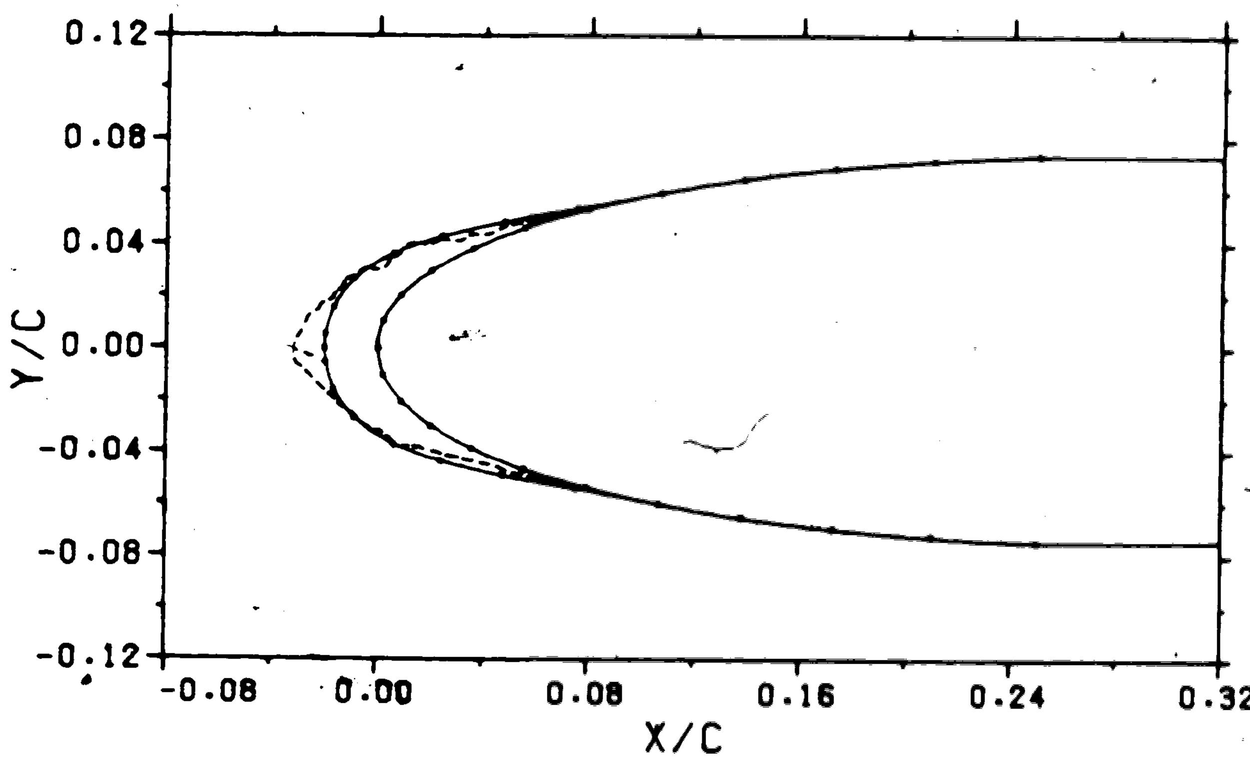

FIG 64 The profile of an accreted layer on a NACA 0015 airfoil at 0° angle of attack. The solid curve with symbols represents the results of Case 54. The dashed line shows the experimental results of Stallabrass & Lozowski (1978). Re_∞ = 98.7 K=0.387 ω = 0.0356

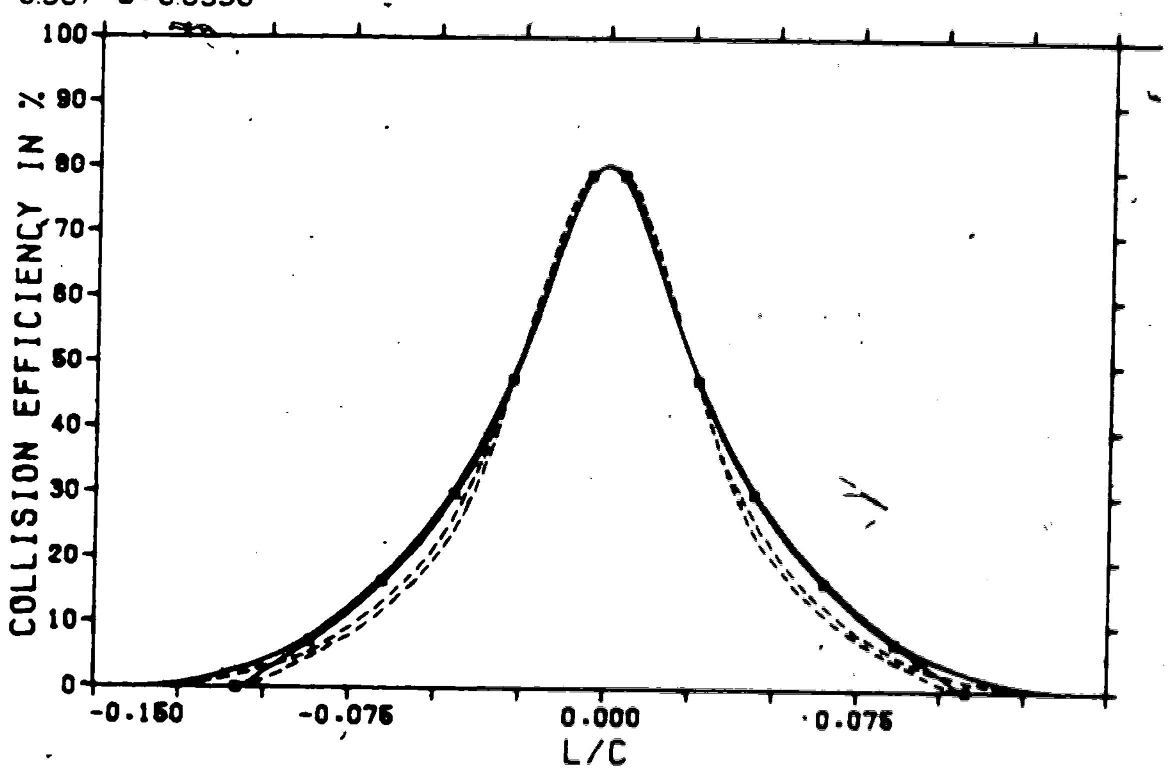

FIG 65 The collision efficiency curves for Case 55. The solid lines represent layer 1 - unfiltered (with symbols) and filtered (without symbols). The two dashed lines are the unfiltered and filtered curves for layer 3.

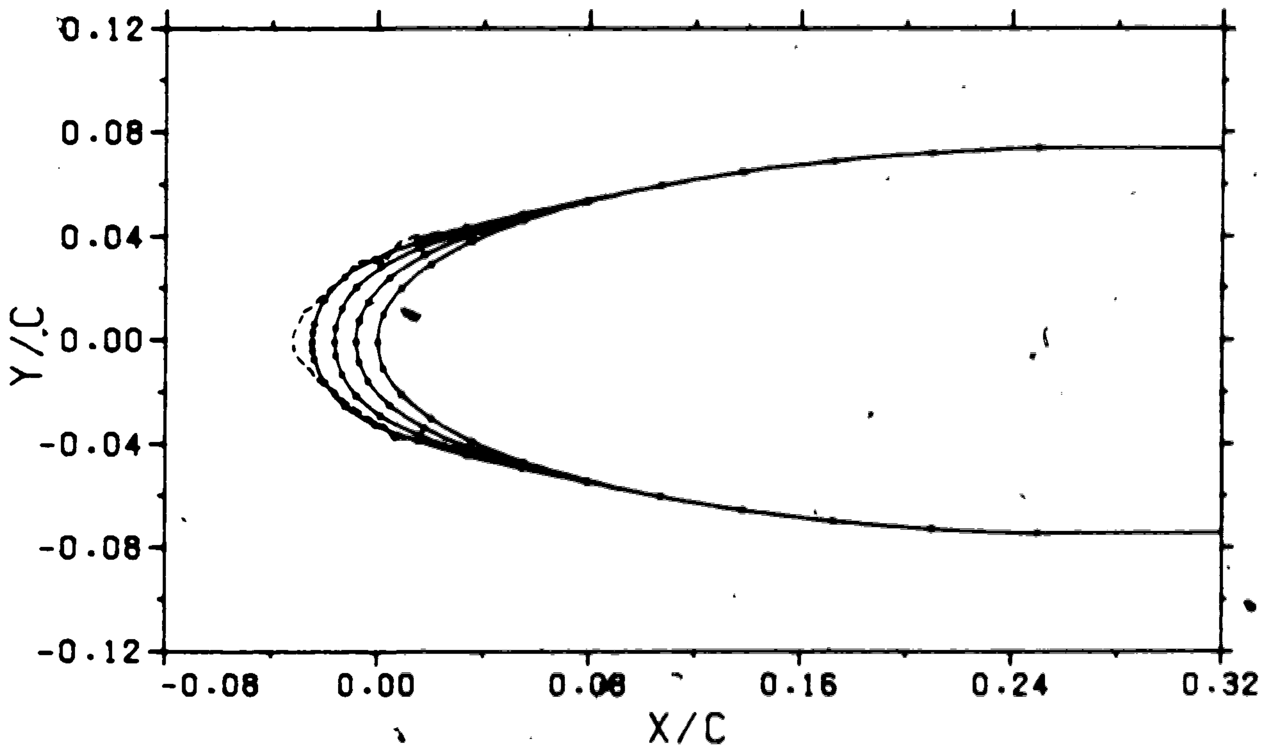

FIG. 66 As in Fig. 64, except for the three layer example of Case 55. $Re_\infty = 98.7$ $K = 0.387$ $\omega = 0.0119$

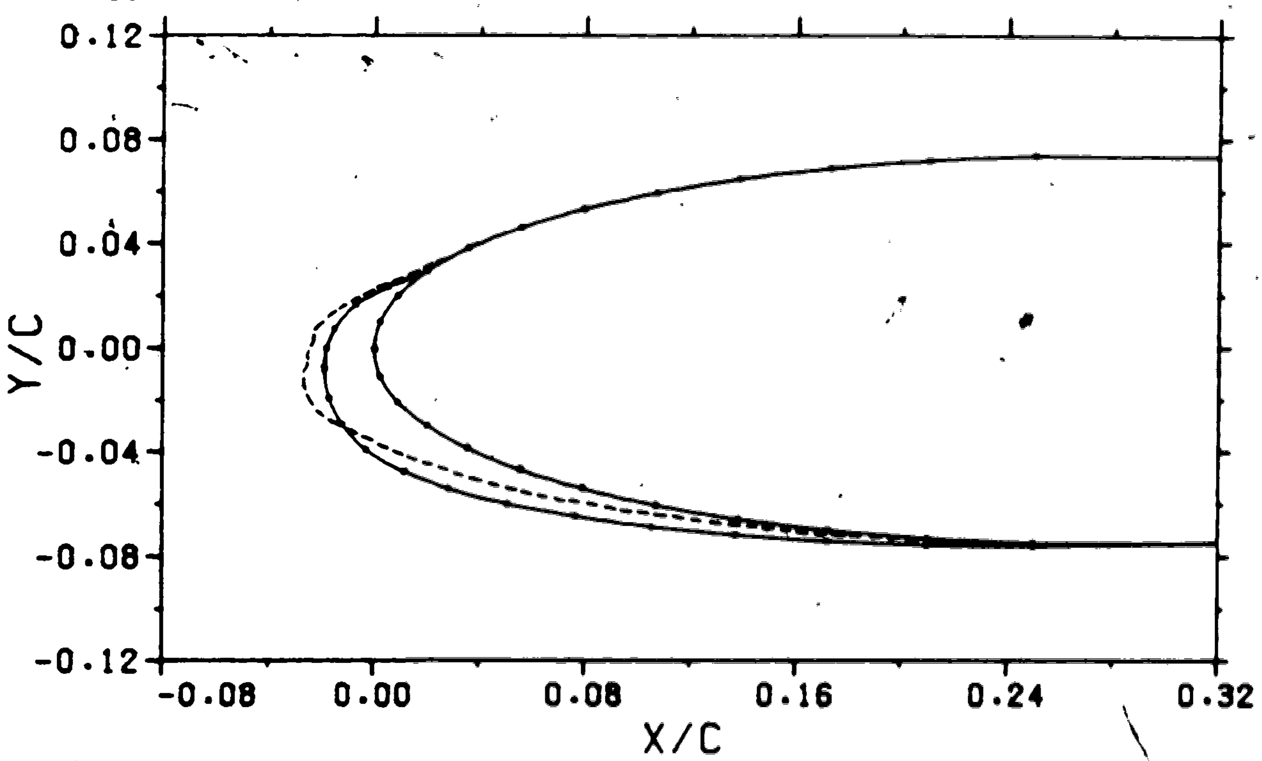

FIG. 67 The profile of an accreted layer on a NACA 0015 airfoil at 8° angle of attack. The solid curve with symbols represents the results of Case 56. The dashed line shows the experimental results of Stallabrass & Lozowski (1978). $Re_\infty = 98.0$ $K = 0.387$ $\omega = 0.0365$

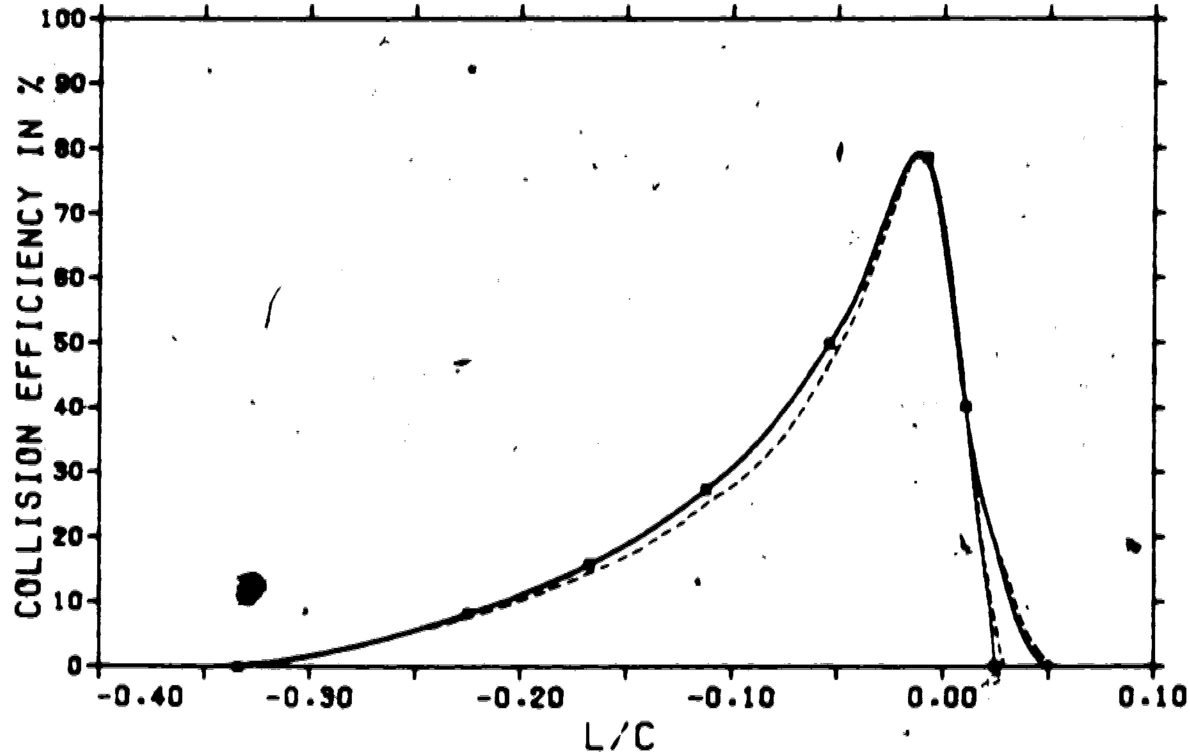

FIG 68 As in Fig 65, but for Case 57 (angle of attack is 8°).

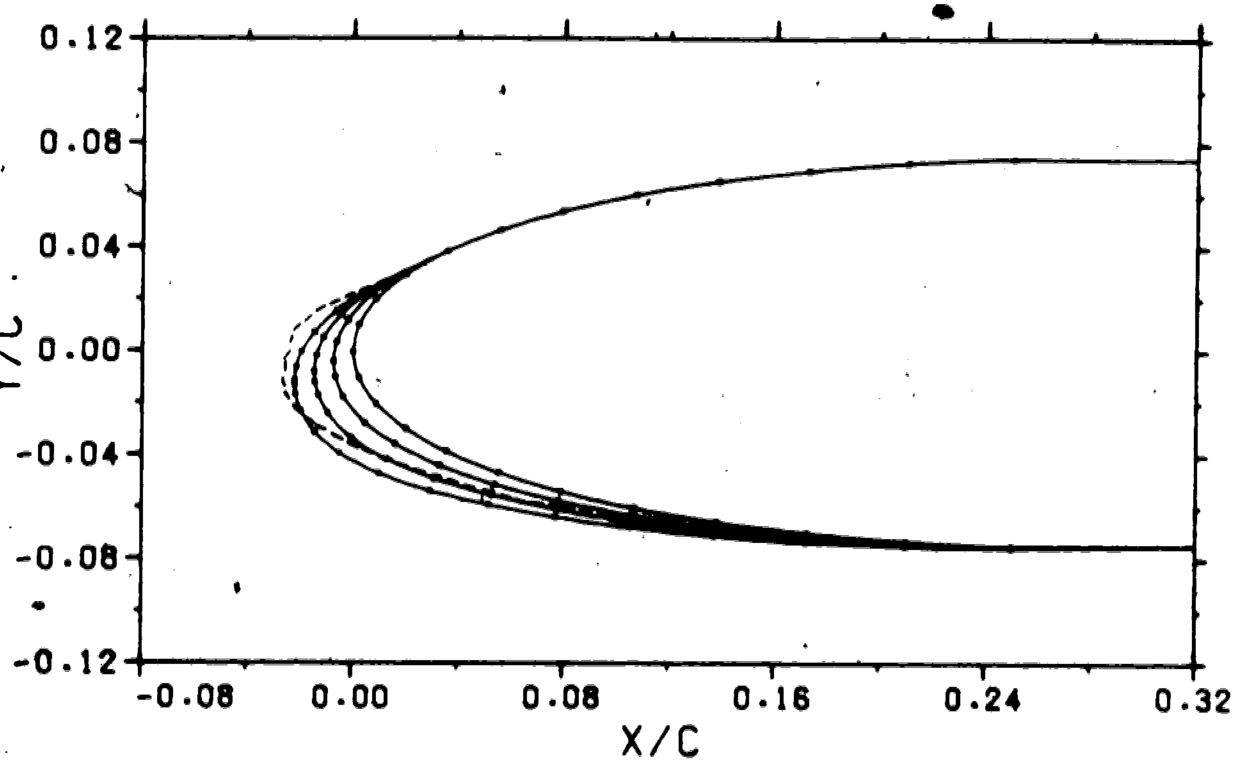

FIG 69 As in Fig 67 except for the three layer example of Case 57

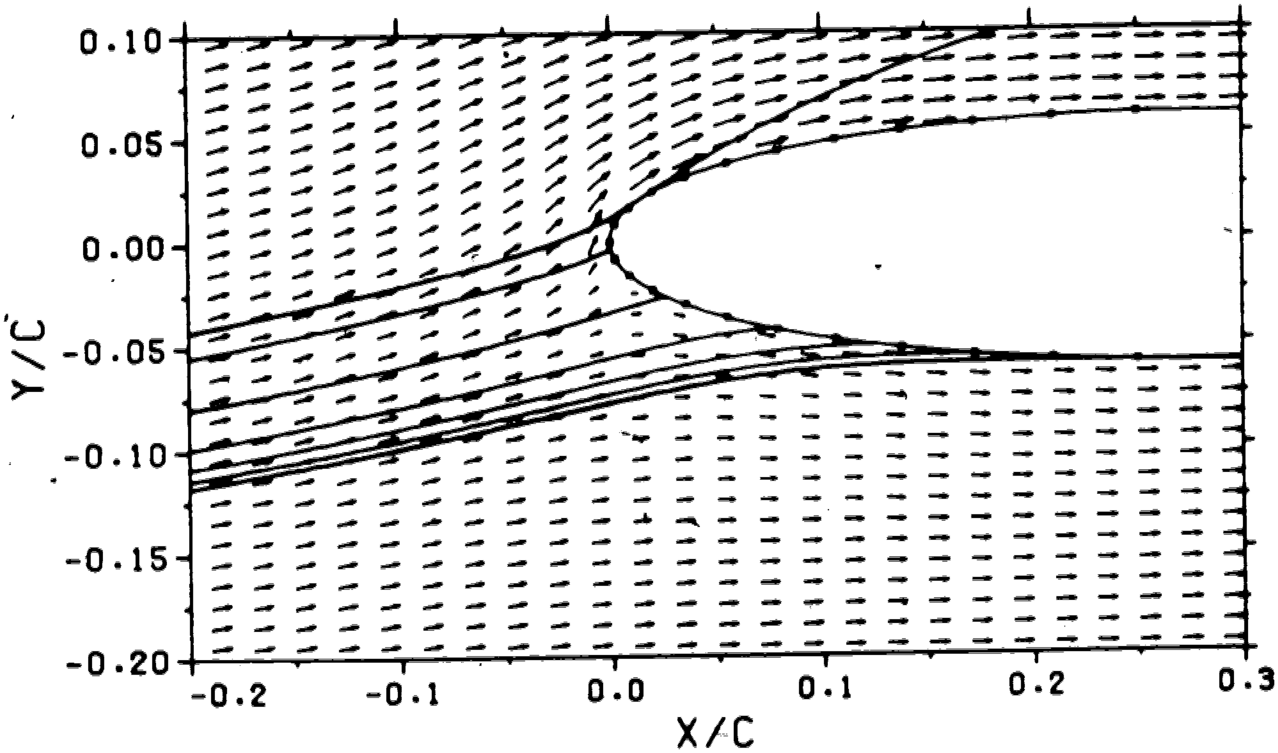

FIG. 70 The trajectories of droplets in a flow about a NACA 0012 airfoil at a 5.7° angle of attack. $Re_\infty = 144$ $K = 0.436$ The conditions are those of Case 58.

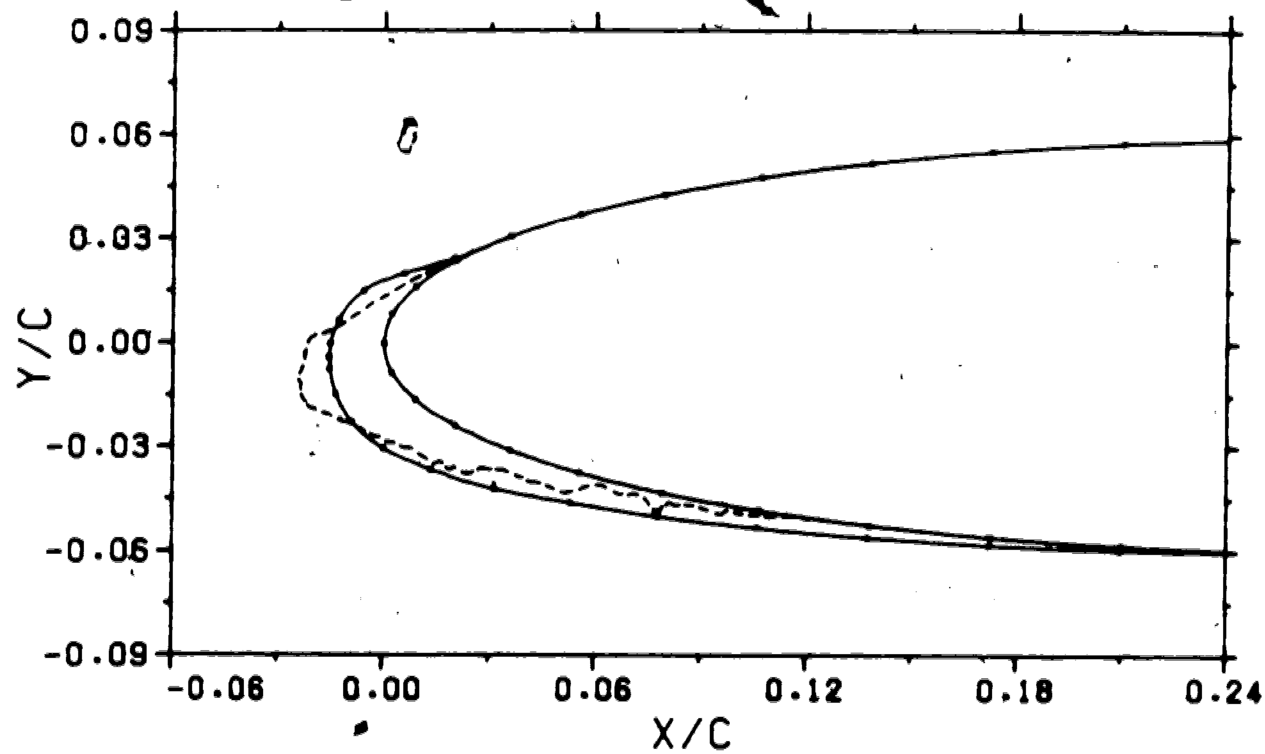

FIG. 71 The profile of an accreted layer on a NACA 0012 airfoil at a 5.7° angle of attack. The solid curve with symbols represents the results of Case 58. The dashed line shows the experimental results of Stallabrass (1958). $Re_\infty = 144$ $K = 0.436$ $\omega = 0.0296$

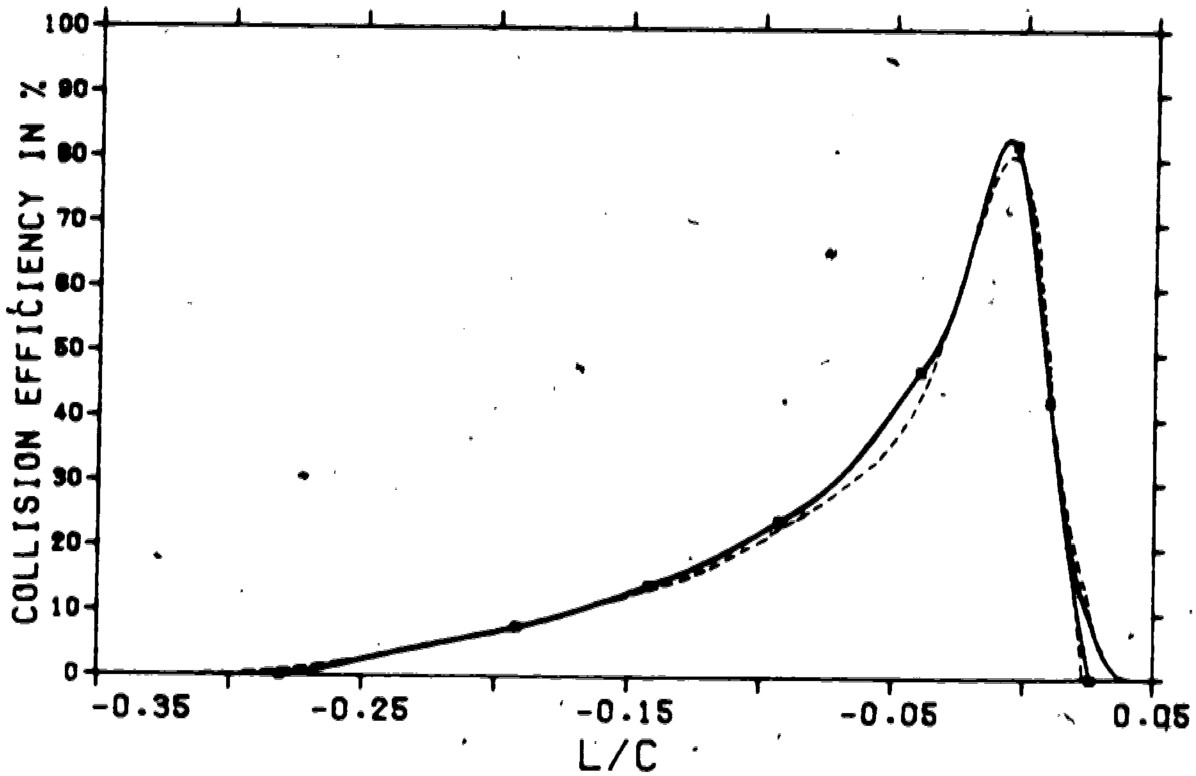

FIG. 72. The collision efficiency curves for Cases 58 and 59. The solid lines represent Case 58 or equivalently layer 1 of Case 59 unfiltered (with symbols) and filtered (without symbols). The two dashed lines are the unfiltered and filtered curves for layer 3 of Case 59.

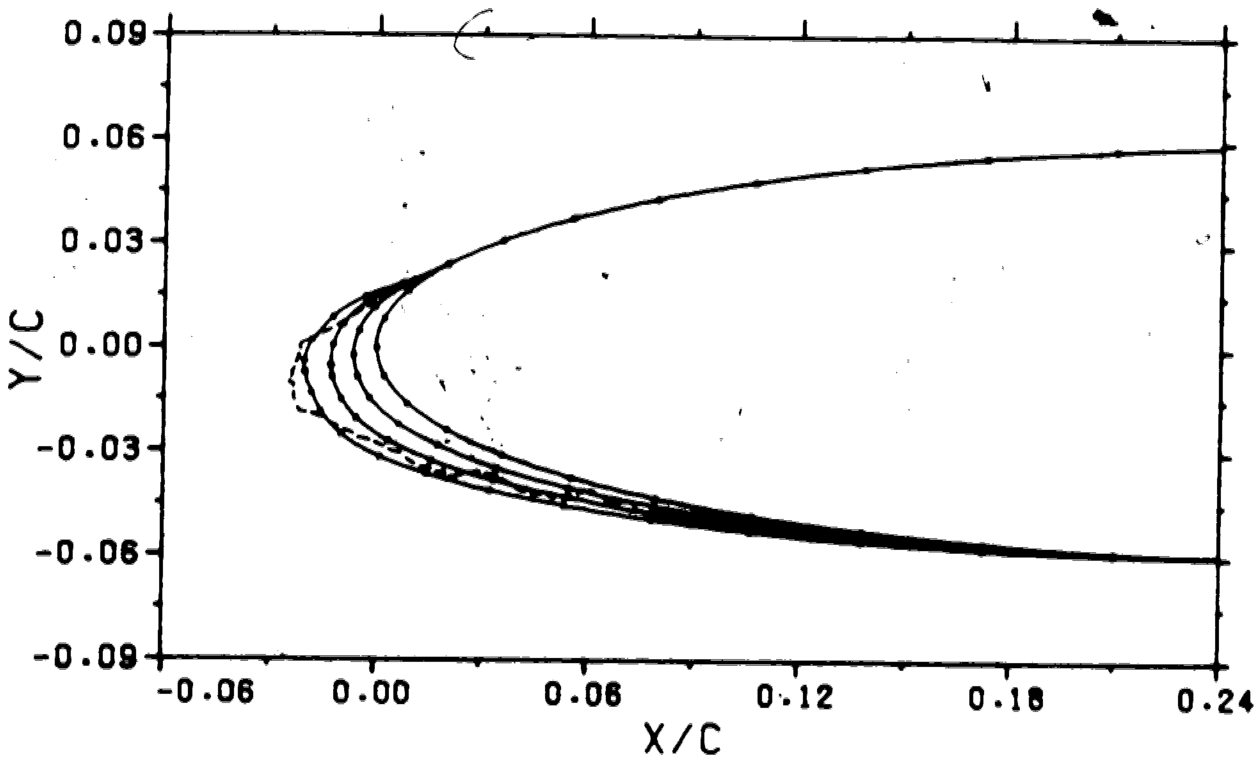

FIG. 73. As in Fig. 71 except for the three layer example of Case 59. $Re_\infty = 144$, $K = 0.436$, $\omega = 0.0099$

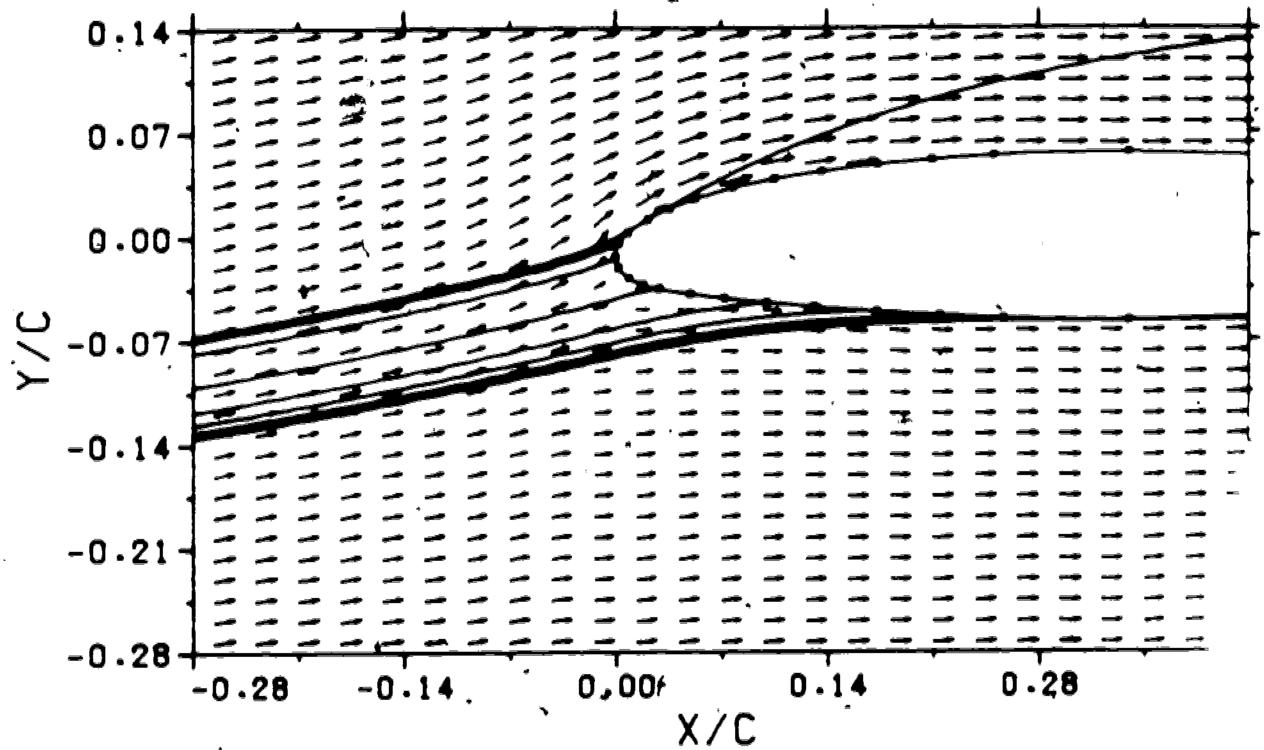

FIG. 74 The trajectories of droplets in a flow about a NPL 9615 airfoil at a 5° angle of attack. The conditions are those of Case 60. $Re_\infty = 144$ $K = 0.41$ $\omega = 0.0279$

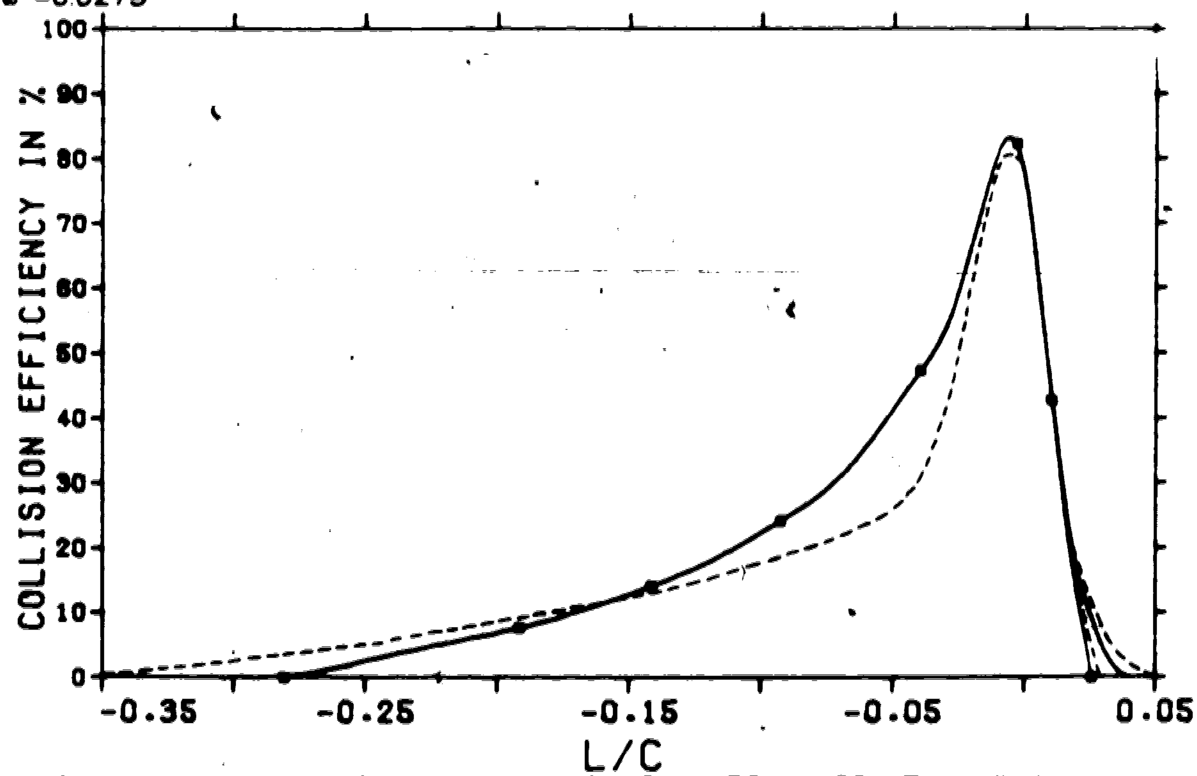

FIG. 75 The collision efficiency curves for Cases 58 and 60. The solid lines represent Case 58 - unfiltered (with symbols) and filtered (without symbols). The two dashed lines are the unfiltered and filtered curves for Case 60.

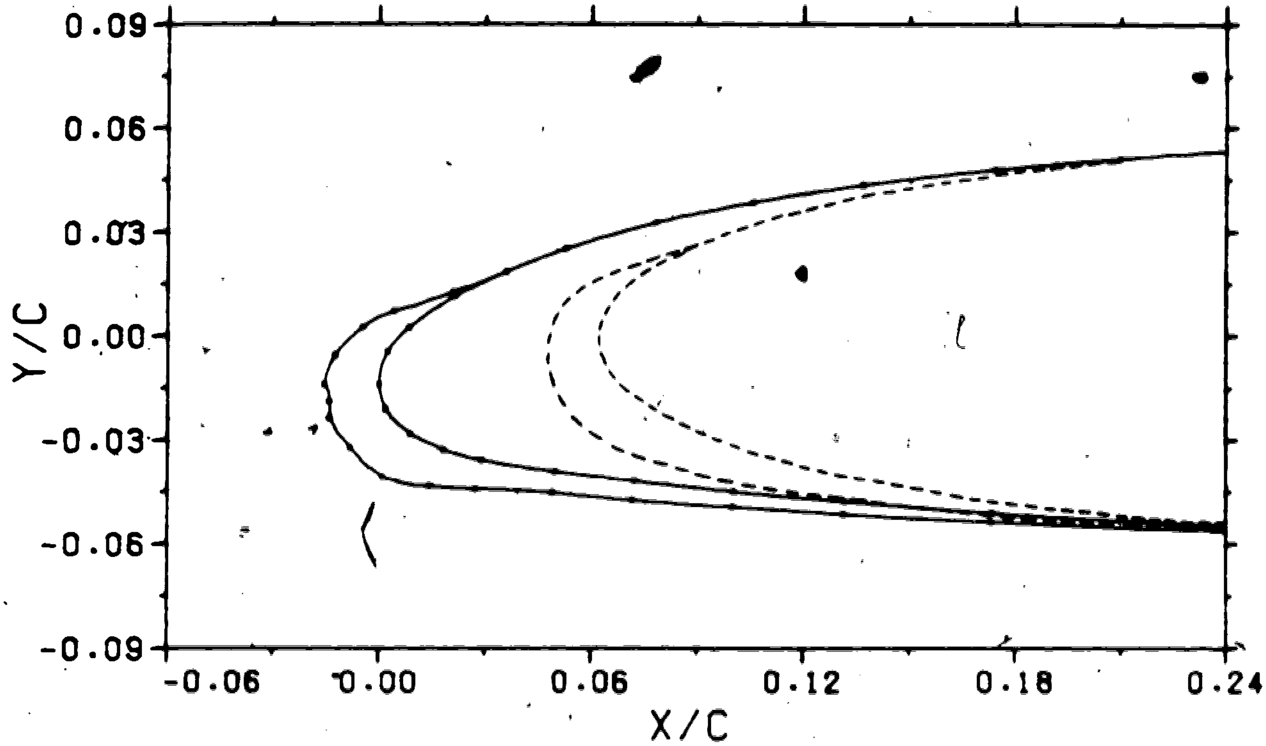

FIG. 76 The solid lines represent the profile of an accreted layer on a NPL 9615 airfoil at 5.7° angle of attack (Case 60). The dashed lines are for a NACA 0012 airfoil under the same conditions (Case 58). The two airfoils are similar except that the NPL 9615 has a drooped-nose extension to the NACA 0012. The NPL airfoil's chord is 6.2% longer.

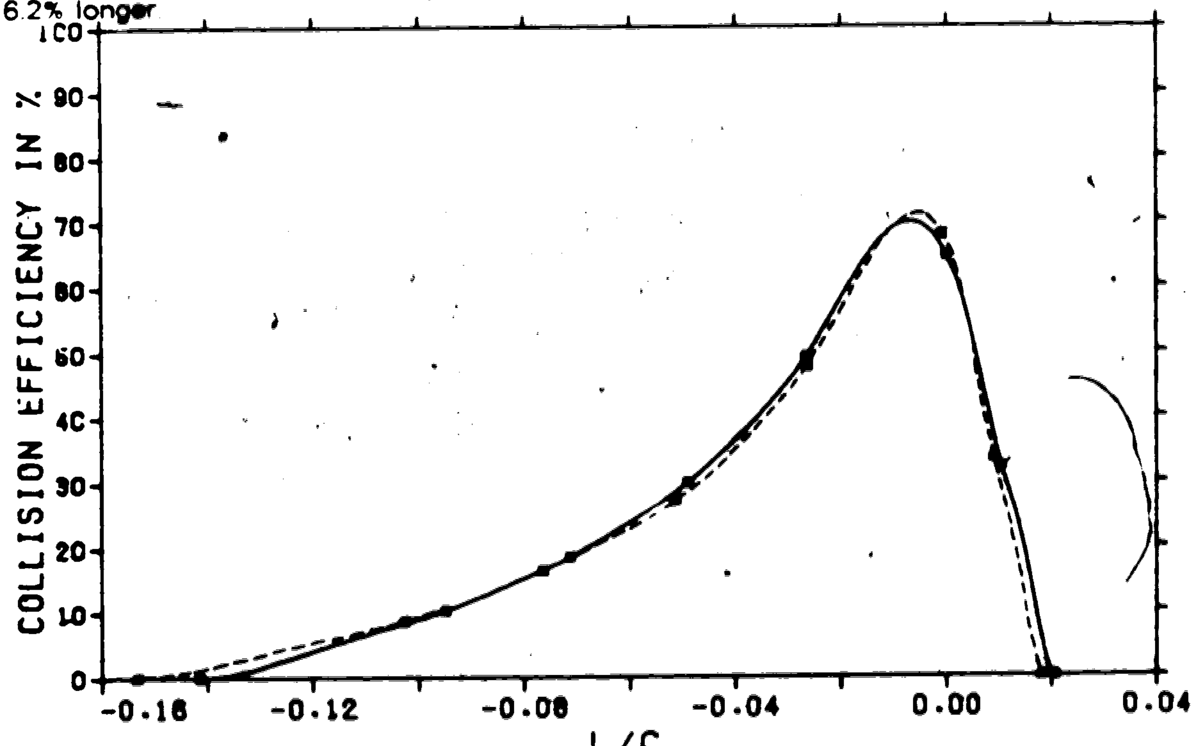

FIG. 77. The collision efficiency curve for Case 12 as a solid line with symbols, and for Case 27 as a dashed line. Case 12 represents a Joukowski 0015 airfoil, and Case 27 represents a NACA 0015 airfoil under the same conditions.

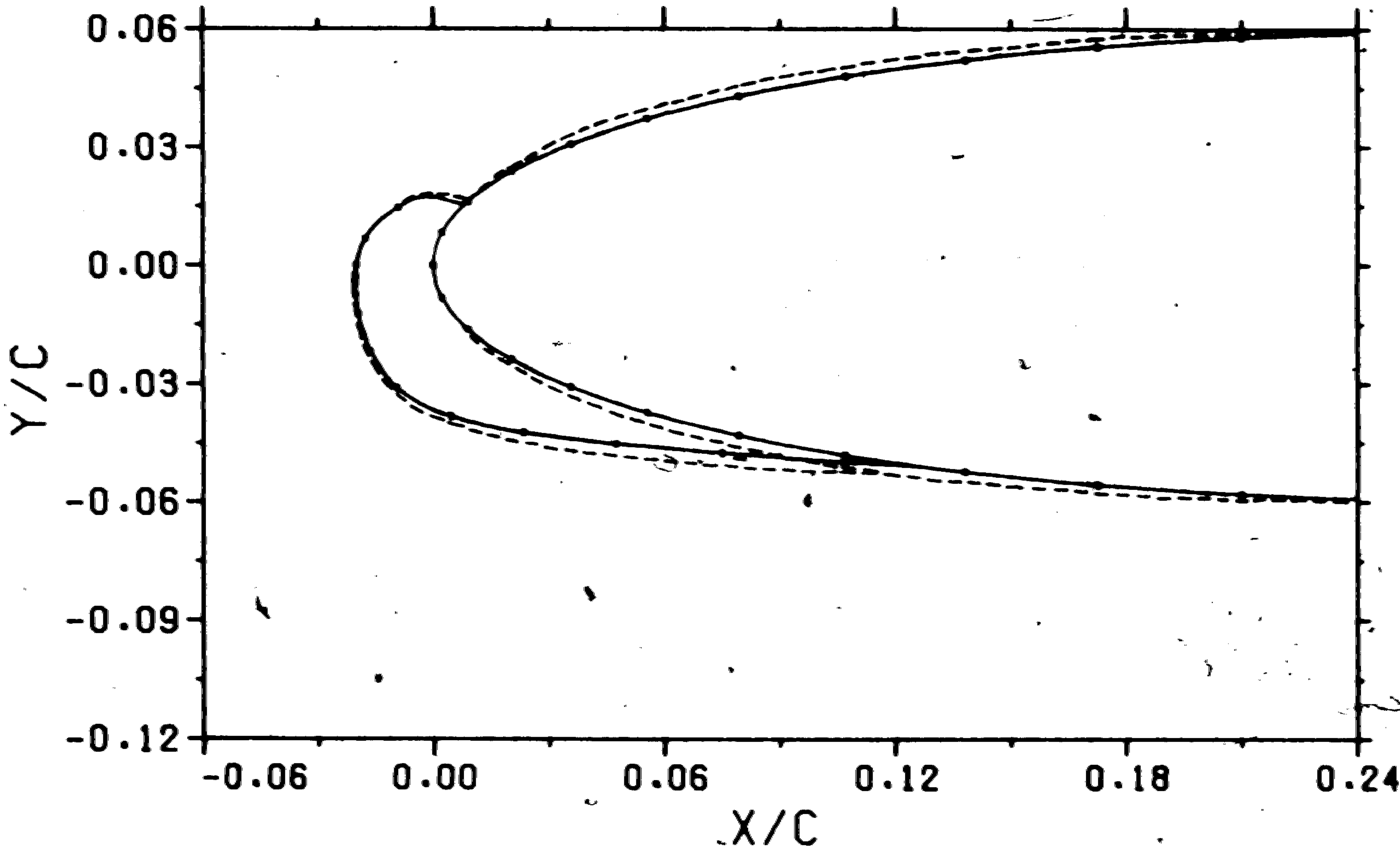

FIG. 78. The accreted layer profiles corresponding to the collision efficiency curves of Fig. 77. The dashed line is for the Joukowski 0015 airfoil. The solid line is for the NACA 0015 airfoil.

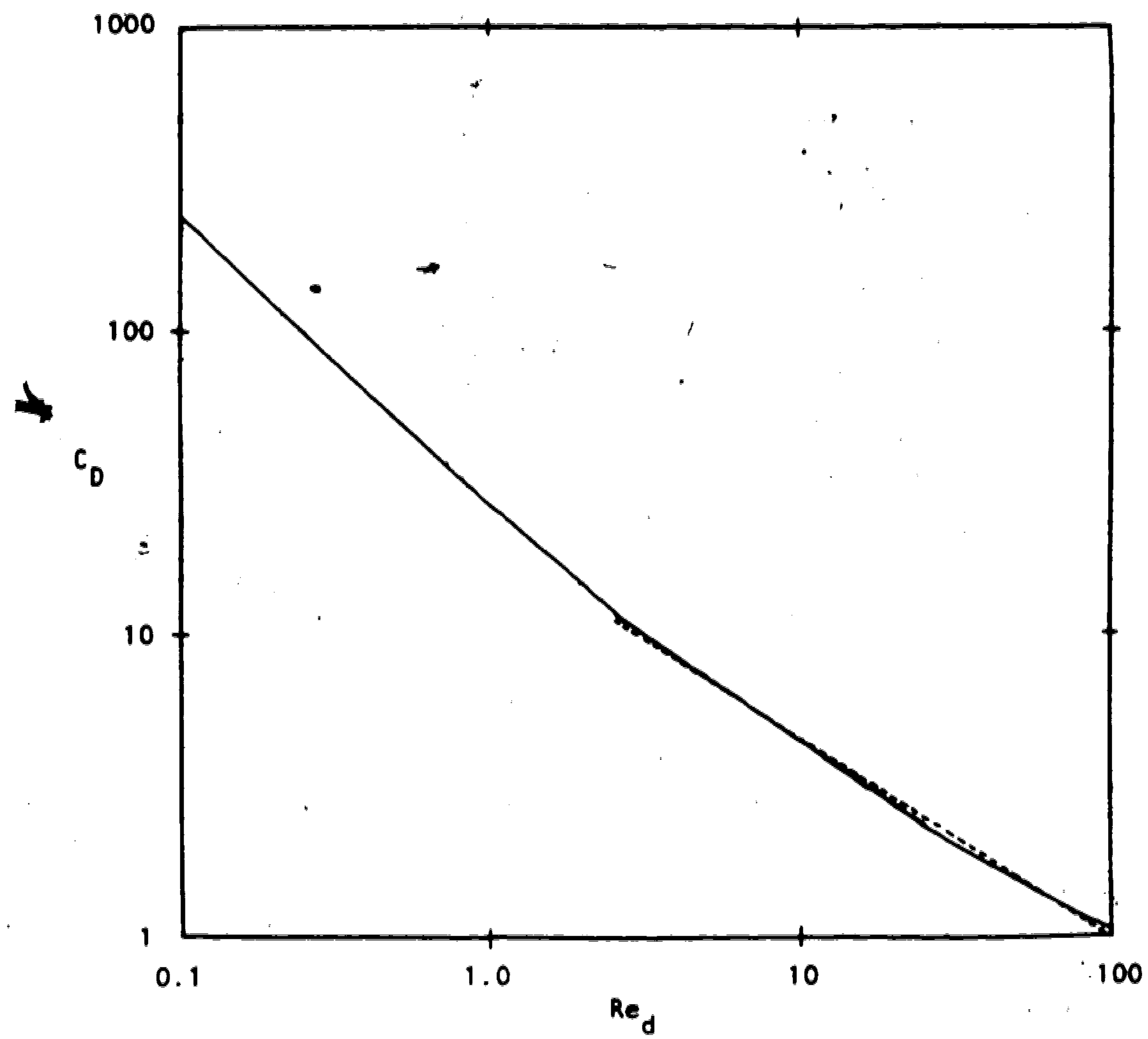

FIG. 79 The functional dependence of the drag coefficient C_D upon the Reynolds number Re_d. The short dashed line is the log-log least squares fit for 25.5 μm droplets in Case 32; it has a slope of −0.66. The long dashed line is the fit for 13.2 μm droplets in Case 32; it has a slope of −0.71.

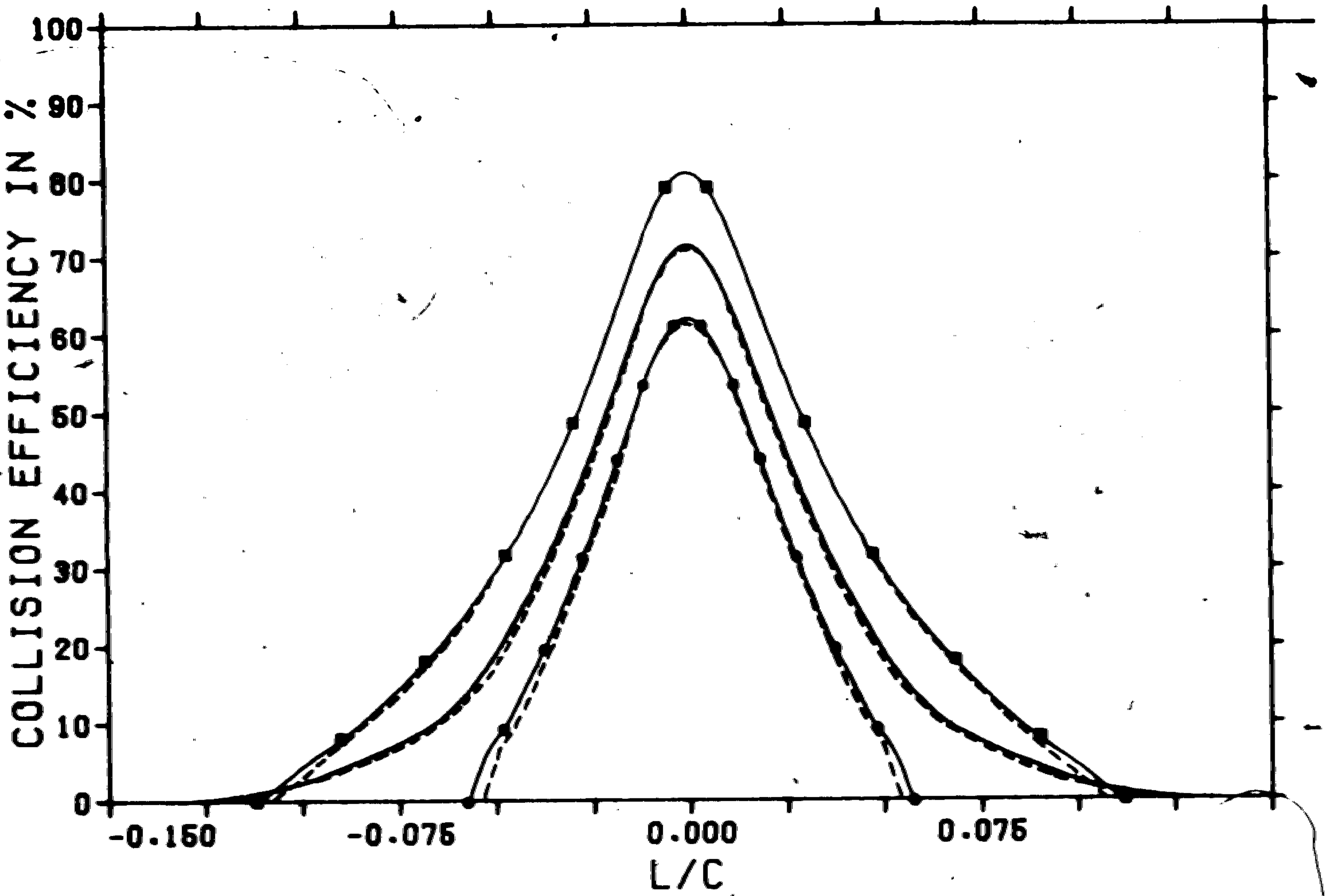

FIG 80 A comparison of the collision efficiency curves for Case 32 (dashed lines) at full scale, and Case 61 (solid lines) at one-quarter scale.

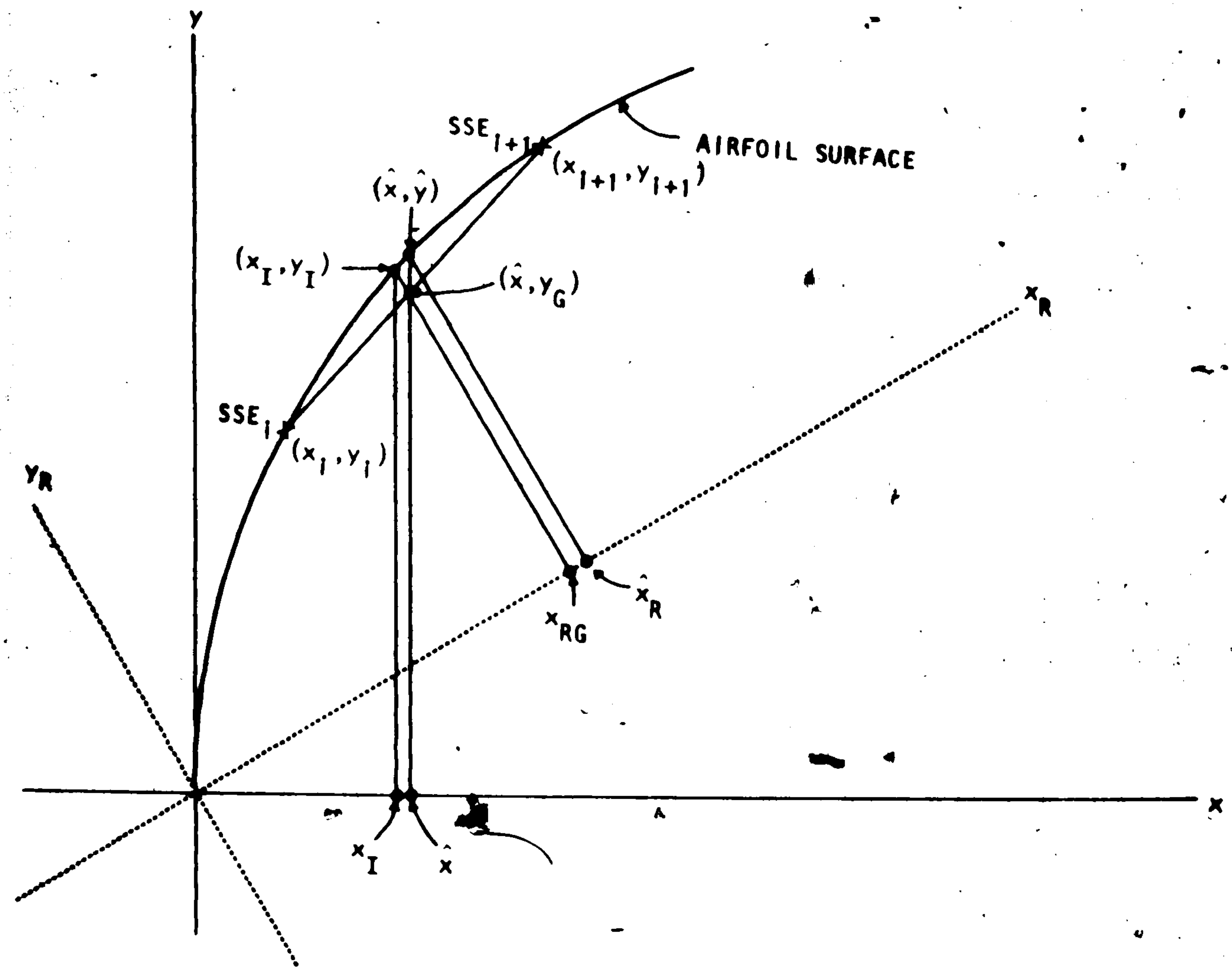

FIG. 81 Notation used to locate the ordinate value for a given airfoil abscissa

BIBLIOGRAPHY

Abbott, I.H. & A.E. von Doenhoff, 1959: *Theory of Wing Sections Including a Summary of Airfoil Data.* Dover, 693 pp.

Abraham, F.F., 1970: Functional dependence of drag coefficient of a sphere on Reynolds number. *Phys. Fluids*, 13, 2194-95.

Ackley, S.F., G.E. Lemieux, K. Itagaki and J. O'Keeke, 1979: Laboratory experiments on icing of rotating blades. In *Snow Removal and Ice Control Research*. National Academy of Sciences, 355 pp.

____ and M.K. Templeton, 1979: Computer modeling of atmospheric ice accretion. U.S. Army Cold Regions Research and Engineering Lab Report CRREL 79-4, 36 pp.

Armand, C., F. Charpin, G. Fasso and G. Leclere, 1978: Techniques and facilities used at the ONERA Modane Centre for icing tests. In *Aircraft Icing*. NATO Advisory Group for Aerospace Research and Development, Advisory Report# AGARD-AR-127, A6-1 to A6-23.

Baker, C.T.H., A. Makroglou and E. Short, 1979: Regions of stability in the numerical treatment of Volterra integro-differential equation. *Siam Jour. Numer. Anal.*, 16, 890-910.

Batchelor, G.K., 1970: *An Introduction to Fluid Dynamics.* Cambridge University Press, 615 pp.

Belheim, M.A., 1978a: Executive summary of aircraft icing specialists workshop. In *Aircraft Icing*. NASA Conf. Pub. 2086, FAA-RD-78-109, 1-16.

____, 1978b: Summary report - Icing research and facilities committee. In *Aircraft Icing*. NASA Conf. Pub. 2086, FAA-RD-78-109, 121-128.

Belte, D., 1981: Helicopter icing spray system improvements and flight experience. *Can. Aeronautics & Space Jour.*, 27, 93-106.

Bragg, M.B., G.M. Gregorek, and R.J. Shaw, 1981: An analytical approach to airfoil icing. Amer. Inst. Aeronautics Astronautics, Paper #AIAA-81-0403, 17 pp.

Brun, E.A., 1957: *Icing problems and recommended solutions.* NATO Advisory Group for Aerospace Research and Development, AGARDograph #16.

____ and H.W. Mergler, 1953: Impingement of water droplets on a cylinder in an incompressible flow field and evaluation of rotating multicylinder method for measurement of droplet-size distribution, volume-median droplet size, and liquid-water content in clouds. NACA Tech. Note 2904.

____, J.S. Serafini, and H.M. Gallagher, 1953: Impingement of cloud droplets on aerodynamic bodies as affected by compressibility of air flow around the body. NACA Tech. Note 2903.

____ and D.E. Vogt, 1957: Impingement of cloud droplets on 36.5-percent thick Joukowski airfoil at zero angle of attack and discussion of use of cloud measuring instrument in dye-tracer technique. NACA Tech. Note 4035.

Burden, R.L., J.D. Faires and A.C. Reynolds, 1978: *Numerical Analysis.* Prindle, Weber & Schmidt, 579 pp.

Buser, O. and A.N. Aufdermaur, 1973: The density of rime on cylinders. *Quart. Jour.*

Roy. Meteor. Soc., **99**, 388–391.

Cansdale, J.T. and I.I. McNaughtan, 1977: Calculation of surface temperature and ice accretion rate in a mixed water droplet/ice crystal cloud. Royal Aircraft Estab. Tech. Report 77090, 24 pp.

Cash, J.R., 1980: On the integration of stiff systems of O.D.E.'s using extended backward differentiation formulae. *Numer. Math.*, **34**, 235–246.

Cotton, R.H., 1976: Ottawa Spray Rig tests of an ice protection system applied to the UH-1H helicopter. U.S. Army Air Mobility Research and Development Lab., Tech. Report USAAMRDL-TR-76-32, 95 pp.

Crowe, C.T., J.A. Nicholls and R.B. Morrison, 1963: Drag coefficients of inert and burning particles accelerating in gas streams. *Ninth Int'l. Symp. on Combustion*, Academic Press, 395–405.

Enright, W.H., T.E. Hull, and B. Lindberg, 1975: Comparing numerical methods for stiff systems of ODEs. BIT, **15**, 10–48.

Fehlberg, E., 1969: Low order classical Runge-Kutta formulas with step-size control and their applications to some heat transfer problems. NASA Tech. Report #TR-R-315.

Fredenburgh, E.A., 1979: Aerodynamic design of the Sikorsky S-76 Spirit helicopter. *Jour. Amer. Helicopter Soc.*, **24**, #3, 11–19.

Frost, W., D.W. Camp, J.W. Connolly, J.H. Enders, J.F. Sowar & H.L. Burton, 1978: Second annual workshop on meteorological and environmental inputs to aviation systems. *Bull. Amer. Meteor. Soc.*, **60**, 38–45.

Gear, C.W., 1971: *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Inc., 253 pp.

Gelder, T.F., W.H. Smyers, Jr., and V. von Glahn, 1956: Experimental droplet impingement on several two-dimensional airfoils with thickness ratios of 6 to 16 percent. NACA Tech. Note 3839.

Googan, R. and E.T. Jackson, 1967: The use of scale models in an icing tunnel to determine the ice catch of a prototype aircraft, with particular reference to the Concorde. British Aircraft Corp. (Operating) Ltd. Filton Div. SST/B75T/RMMoK/242 Issue #1.

Googan, R. and J.A. Hubbold, 1968: Icing tests on a 1/6th scale model (G14) at Modane. British Aircraft Corp. (Operating) Ltd. Filton Div. SST/B72T-51/5027 Issue #1.

Gray, J.H., 1957: Correlations among ice measurements, impingement rates, icing conditions, and drag coefficients for unswept NACA 65A004 airfoil. NACA Tech. Note 4151.

Guibert, A.G., E. Janssen, and W.M. Robbins, 1949: Determination of rate, area and distribution of impingement of waterdrops on various airfoils from trajectories on the differential analyzer. NACA Research Memo 9A05.

Hallett, J., 1980: Characteristics of atmospheric ice particles: A survey of techniques. U.S. Air Force AFGL-TR-80-0308 (AD-A093 927).

Hamming, R.W., 1973: *Numerical Methods for Scientists and Engineers, 2nd Ed.* McGraw-Hill, Inc., 721 pp.

Hammond, C.E. and G.A. Pierce, 1973: A compressible unsteady aerodynamic theory for helicopter rotors. *Aerodynamics of Rotary Wings*. AGARD Conf. Proceedings

#111, 13.1-13.15

Hauger, H.H., K.G. Engler and W.W. Reaser, 1954: Analysis of model testing in an icing wind tunnel. Douglas Aircraft Company Inc., Report #SM 14993.

Hess, J.L. and A.M.O. Smith, 1967: Calculation of potential flow about arbitrary bodies. *Progress in Aeronautical Sciences*, **8**, Pergamon Press, 1-138.

Houghton, E.L. and A.E. Brock, 1970: *Aerodynamics for Engineering Students, 2nd Edition*. Edward Arnold, 458 pp.

Hull T.E., 1980: Comparison of algorithms for initial value problems. In Gladwell, I. and D.K. Sayers, Eds. *Computation Techniques for Ordinary Differential Equations*. Academic Press, 303 pp.

IMSL, Inc., (1979): *IMSL Library Reference Manual, 7th Ed*. IMSL Inc., 3 volumes.

Jeck, R.K., 1980: Icing characteristics of low altitude, supercooled layer clouds. U.S. Federal Aviation Administration FAA-RD-80-24 (Revised) (AD-A088 892).

Joe, P.I., 1975: Investigation of the bouncing of supercooled water droplets from an artificially growing hailstone. M. Sc. Thesis, Dept. of Physics, Univ. of Toronto, 254 pp.

Jenkins, G.M., and D.W. Watts, 1968: *Spectral Analysis and Its Applications*. Holden-Day, 525 pp.

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.

Kloner, M.O., 1970: A method for calculating ice catch on airfoils and inlets. Lockheed California Co. Report #LR 23373, 54 pp.

Lake, J.B. and J. Bradley, 1976: The problem of certifying helicopters for flight in icing conditions. *Aeronautical Jour.*, 419-433.

Lambert, J.D., 1980: Stiffness. In Gladwell, I. and D.K. Sayers, Eds., *Computational Techniques for Ordinary Differential Equations*. Academic Press, 303 pp.

Landau, L. and E.M. Lifshitz, 1959: *Fluid Mechanics*. Pergamon Press, 536 pp.

Langmuir, I. and K.B. Blodgett, 1946: A mathematical investigation of water droplet trajectories. In *Collected Works of I. Langmuir*. Pergamon Press, **10**, 348-393.

Lecoutre, J.C., 1978: Protection systems against icing on the Puma. Presented at NATO Panel X Helicopter Icing Symposium, London.

Lewis, W., 1947: A flight investigation of the meteorological conditions conducive to the formation of ice on airplanes. NACA Tech. Note 1393.

____, and R. Bergrun, 1952: A probability analysis of the meteorological factors conducive to aircraft icing in the U.S. NACA Tech Note 2738.

Liniger, W., 1976: High-order A-stable averaging algorithms for stiff differential systems. In Schiesser, W.E. and L. Lapidus, Eds., *Numerical Methods for Differential Systems*. Academic Press, 291 pp.

List, R., 1977: Ice accretions on structures. *Journal of Glaciology*, **19**, 451-465.

Loewy, R.G., 1969: Review of rotary-wing V/STOL dynamic and aeroelastic problems. *J. Amer. Helicopter Soc.*, **14**, #3, 3-23.

Lozowski, E.P., 1981 Personal communication.

____, and M.M. Oleskiw, 1981: Computer simulation of airfoil icing without runback. Amer. Inst. Aeronautics Astronautics, Paper AIAA-81-0402, 8 pp.

____, J.R. Stallabrass and P.F. Hearty, 1979: The icing of an unheated non-rotating cylinder in liquid water droplet-ice crystal clouds. Nat. Research Coun. Canada, Lab. Report #LTR-LT-96, 61 pp.

Makkonen, L., 1981: Estimating intensity of atmospheric ice accretion on stationary structures. *Jour. Appl. Meteor.*, **20**, 595-600.

Ludlam, F.H., 1951: The heat economy of a rimed cylinder. *Quart. Jour. Roy. Meteor. Soc.*, **77**, 663-666.

Macklin, W.C., 1962. The density and structure of ice formed by accretion. *Quart. Jour. Roy. Meteor. Soc.*, **88**, 30-50.

____, and G.S. Payne, 1968: Some aspects of the accretion process. *Quart. Jour. Roy. Meteor. Soc.*, **94**, 167-175.

Makroglou, A., 1977: Numerical solution of Volterra integro-differential equations. Ph. D. Thesis, U. of Manchester, U.K.

Maskew, B. and F.A. Dvorak, 1978: The prediction of Clmax using a separated flow model. *Jour. Amer. Helicopter Soc.*, **23**, 2-8.

McComber, P. and G. Touzot, 1981: Calculation of the impingement of cloud droplets in a cylinder by the finite-element method. *Jour. Atmos. Sci.*, **38**, 1027-1036.

McKay, G.A. and H.A. Thompson, 1969: Estimating the hazard of ice accretion in Canada from climatological data. *Jour. Appl. Meteor.*, **8**, 927-935.

Mesinger, B.L., 1953: Equilibrium temperature of an unheated icing surface as a function of airspeed. *Jour. Aeronautical Sci.*, **20**, 29-41.

Norment, H.G., 1980: Calculation of water drop trajectories to and about arbitrary three-dimensional bodies in potential airflow. NASA Contractor Report 3291, 82 pp.

Oleskiw, M.M. and E.P. Lozowski, 1980: Helicopter rotor blade icing: A numerical simulation. 8th Int'l. Conf. on Cloud Phys., Clermont-Ferrand, 281-284.

Pearcey, T. and G.W. Hill, 1956: The accelerated motion of droplets and bubbles. *Aust. Jour. Phys.*, **9**, 19-30.

Phillips, D.S., 1980: Personal communication.

Pitter, R.L. and H.R. Pruppacher, 1974: A numerical investigation of collision efficiencies of simple ice plates colliding with supercooled water drops. *Jour. Atmos. Sci.*, **31**, 551-559.

Poots, G. and G.G. Rodgers, 1976: The icing of a cable. *Inst. Maths. Applics.*, **18**, 203-217.

Pouzet, P., 1960: Methode d'integration numerique des equations integrales et integro-differentielles du type Volterra de seconde espece formules de Runge-Kutta. *Symposium on the Numerical Treatment of Ordinary Differential Equations, Integral Equations, and Integro-differential Equations*. Birkhauser Verlag, 679 pp.

Prothero, A., 1980: Estimating the accuracy of numerical solutions to ordinary differential equations. In Gladwell, I. and D.K. Sayers, Eds., *Computational*

Techniques for Ordinary Differential Equations. Academic Press. 303 pp.

Pruppacher, H.R. and J.D. Klett, 1978: *Microphysics of Clouds and Precipitation.* D. Reidel, 714 pp.

Reichert, G. and S.N. Wagner, 1973: Some aspects of the design of rotor-airfoil shapes. *Aerodynamics of Rotary Wings,* AGARD Conf. Proceedings #111, 14.1-14.20.

Riegels, F.W., 1961: *Aerofoil Sections. Results From Wind-Tunnel Investigations. Theoretical Foundations.* Butterworths, 281 pp.

Rosen, K.M. and M.L. Potash, 1981: 40 years of helicopter ice protection experience at Sikorsky Aircraft. *Jour. Amer. Helicopter Soc.,* **26**, 5-19.

Ryder, P., 1978: The role of meteorology in helicopter icing problems. *Met. Mag.,* **107**, 140-147.

Sartor, J.D. and C.E. Abbott, 1975: Prediction and measurement of the accelerated motion of water drops in air. *Jour. Appl. Meteor.,* **14**, 232-239.

Szelazel, C.A. and R.M. Hicks, 1979: Upper-surface modifications for Clmax improvement of selected NACA 6-series airfoils. NASA Tech. Memorandum 78603, 79 pp.

Shampine, L.F., 1980: What everyone solving differential equations should know. In Gladwell, I. and D.K. Sayers, Eds., *Computational Techniques for Ordinary Differential Equations.* Academic Press, 303 pp.

_____, and H.A. Watts, 1976: Global error estimation for ordinary differential equations. *ACM Trans. Math. Software,* **2**, 172-186.

Spath, H., 1974: *Spline Algorithms for Curves and Surfaces.* Utilitas Mathematica Pub. Inc., 198 pp.

Stallabrass, J.R., 1957: Some aspects of helicopter icing. *Canadian Aeronautical Jour.,* **8**, 273-283.

_____, 1958a: Icing flight trials of a Sikorsky H04S-2 helicopter. Nat. Aeronautical Estab. Canada, Lab. Report #LR-219, 25 pp.

_____, 1958b: Canadian research in the field of helicopter icing. *Jour. Helicopter Assoc. Great Britain,* **12**, 3-40.

_____, and E.P. Lozowski, 1978: Ice shapes on cylinders and rotor blades. Presented at NATO Panel X Symposium on Helicopter Icing, London.

Theodorsen, T. and I.E. Garrick, 1932: General potential theory of arbitrary wing sections. NACA Report 452.

Wanner, G., 1977: On the integration of stiff differential equations. In Descloux, J. and J. Marti, Eds.: *Numerical Analysis, Proceedings of the Colloquium on Numerical Analysis,* Lausanne, Oct. 11-13, 1976.

Werner, J.B., 1973: Ice protection investigation for advanced rotary-wing aircraft. US Army Air Mobility Research and Development Lab., Tech. Report USAAMRDL-TR-73-38.

_____, 1975: The development of an advanced anti-icing / de-icing capability for U.S. Army helicopters. Vol 1., Design criteria and technology considerations. U.S. Army Air Mobility Research and Development Lab., Tech. Report USAAMRDL-TR-75-34A., 253 pp.

Wortmann, F.X., 1973: Prepared comment on "Some aspects of the design of rotor-airfoil shapes." *Aerodynamics of Rotary Wings* AGARD Conf. Proceedings #111, 14.21-14.22.

APPENDIX A. Finding the eigenvalues of the Jacobian of the system of droplet trajectory equations.

Section 2.3.4.2 has stated the need for developing an indicator of the stability of the ordinary differential equation solver used to determine the droplet trajectories. This indicator is based upon the complex eigenvalues of the Jacobian $\partial\bar{f}/\partial\bar{x}$. This appendix will outline the means of finding those eigenvalues.

If we ignore the gravity term, and the history term in (2.66), the vector equations of motion become:

$$\frac{d\bar{V}_d}{d\bar{T}} = -\frac{3C_D\rho_a}{4r_d(2\rho_d + \rho_a)} |\bar{V}_d - \bar{V}_a| (\bar{V}_d - \bar{V}_a) \quad (A.1)$$

and

$$\frac{d\bar{X}_d}{d\bar{T}} = \bar{V}_d \quad (A.2)$$

If these equations are broken into their components and non-dimensionalized, the resulting set of first-order equations is:

$$\frac{dx_d}{dt} = u_d = f_1(u_d) \quad (A.3)$$

$$\frac{du_d}{dt} = -K_3C_D|\bar{v}_d - \bar{v}_a|(u_d - u_a) = f_2(x_d, u_d, y_d, v_d) \quad (A.4)$$

$$\frac{dy_d}{dt} = f_3(v_d) \quad (A.5)$$

and

$$\frac{dv_d}{dt} = -K_3C_D|\bar{v}_d - \bar{v}_a|(v_d - v_a) = f_4(x_d, u_d, y_d, v_d) \quad (A.6)$$

where

$$K_3 = \frac{3\rho_a}{4r_d(2\rho_d + \rho_a)} \quad (A.7)$$

The Jacobian $\partial f/\partial y$ is thus

$$\begin{vmatrix} 0 & 1 & 0 & 0 \\ \partial f_2/\partial x_d & \partial f_2/\partial u_d & \partial f_2/\partial y_d & \partial f_2/\partial v_d \\ 0 & 0 & 0 & 1 \\ \partial f_4/\partial x_d & \partial f_4/\partial u_d & \partial f_4/\partial y_d & \partial f_4/\partial v_d \end{vmatrix} \tag{A.8}$$

If we now set

$$z_1 = \partial u_a/\partial x_d \tag{A.9}$$

$$z_2 = \partial u_a/\partial y_d \tag{A.10}$$

$$z_3 = \partial v_a/\partial x_d \tag{A.11}$$

$$z_4 = \partial v_a/\partial y_d \tag{A.12}$$

$$g_1 = (u_d - u_a) \tag{A.13}$$

$$g_2 = (v_d - v_a) \tag{A.14}$$

$$g_3 = |\vec{v}_d - \vec{v}_a| = \sqrt{g_1^2 + g_2^2} \tag{A.15}$$

and

$$K_2 = 2r_d/\nu_a \tag{A.16}$$

then algebraic manipulation will yield

$$\frac{\partial f_2}{\partial x_d} = -K_3\left[K_2 \frac{\partial C_D}{\partial Re_d}\frac{\partial g_3}{\partial x_d} g_3 g_1 + \frac{\partial g_3}{\partial x_d} C_D g_1 + \frac{\partial g_1}{\partial x_d} C_D g_3\right] \tag{A.17}$$

$$\frac{\partial f_2}{\partial u_d} = -K_3 \left[K_2 \frac{\partial C_D}{\partial Re_d} \frac{\partial g_3}{\partial u_d} g_3 g_1 + \frac{\partial g_3}{\partial u_d} C_D g_1 + \frac{\partial g_1}{\partial u_d} C_D g_3 \right] \qquad (A.18)$$

and similarly for $\partial f_2/\partial y_d$ through $\partial f_4/\partial v_d$. When these values are inserted into (A.18) and the determinant calculated, the complex eigenvalues of the system of equations will have been determined. These eigenvalues shall be complex if the solution to the system of ordinary differential equations contains both an oscillatory and a decaying or growing part. The eigenvalues cannot be stated here precisely, because as was mentioned in Section 2.3.4.2, the derivatives denoted z_1 through z_4 must be determined numerically as the integration proceeds.

APPENDIX B. A modified Runge-Kutta-Fehlberg (RKF4) algorithm.

The standard RKF4 algorithm may be found in several textbooks (see for example page 254 of Burden *et al.*, 1978). If we have a system of first order equations to solve, we must apply each section of the algorithm to each equation before moving on to the next section. Briefly the algorithm may be summarized as follows.

Beginning at time t_i, we wish to solve for x_d, u_d, y_d, and v_d at t_{i+1}. Using (A.3) through (A.6), we find the values of

$$k_1 = \Delta t_i f_1(t_i, x_{d_i}, u_{d_i}, y_{d_i}, v_{d_i}) \quad \text{(B.1)}$$

$$\vdots$$

$$n_1 = \Delta t_i f_4(t_i, x_{d_i}, u_{d_i}, y_{d_i}, v_{d_i}) \quad \text{(B.4)}$$

$$k_2 = \Delta t_i f_1(t_i + \Delta t_i/4, x_{d_i} + k_1/4, u_{d_i} + \ell_1/4, y_{d_i} + m_1/4, v_{d_i} + n_1/4) \quad \text{(B.5)}$$

$$\vdots$$

$$n_6 = \Delta t_i f_4(t_i + \Delta t_i/2, x_{d_i} - 8/27 n_1 + 2n_2 - 3544/2565 n_3 \ldots) \quad \text{(B.24)}$$

These values may then be combined to give fourth and fifth order estimates for x_d, u_d, y_d, and v_d, denoted: $\hat{x}_{d(i+1)}, \hat{\hat{x}}_{d(i+1)}, \hat{u}_{d(i+1)}$ and so on to $\hat{\hat{v}}_{d(i+1)}$. If we set

$$\underline{x}_i = (\hat{x}_{d_{i+1}} - \hat{\hat{x}}_{d_{i+1}})/\Delta t_i \quad \text{(B.25)}$$

and similarly for $\underline{u}_i$, $\underline{y}_i$ and $\underline{v}_i$, then the local truncation error at time t_i may be estimated as:

$$e_i = \max\{|\underline{x}_i|, |\underline{u}_i|, |\underline{y}_i|, |\underline{v}_i|\} \quad \text{(B.26)}$$

The input parameter EPS sets the local error tolerance ε. Thus, if $e_i \le \varepsilon$, we accept the values $\hat{x}_{d(i+1)}, \hat{u}_{d(i+1)}, \hat{y}_{d(i+1)}, \hat{v}_{d(i+1)}$ and prepare for the next time step. If $e_i > \varepsilon$, then we repeat the integration at t_i with a smaller time step.

The choice of an appropriate time step for the RKF4 algorithm has proven to be somewhat problematical. Burden *et al.* suggest setting the new time step according to the formula

$$\Delta t_{i+1} = \hat{k}\, \chi_i \Delta t_i \tag{B.27}$$

where

$$\chi_i = [\varepsilon/\max\{|\underline{x}|, |\underline{u}|, |\underline{y}|, |\underline{v}|\}]^{0.25} \tag{B.28}$$

and

$$\hat{k} = (0.5)^{0.25} \tag{B.29}$$

Equation (B.27) is also used when a step is to be repeated. In this case Δt_{i+1} is replaced by Δt_i. Experience with this formulation led to serious reservations about its applicability to the system of equations we were attempting to solve. Following the lead of Burden *et al.* (1978), we have found it necessary to "eliminate large changes in step size to avoid spending too much time with very small step sizes in regions with irregularities in derivatives of [x_d, y_d, u_d, and v_d], and to avoid spending too little time [in a region] with large step sizes, which may result in skipping sensitive regions nearby." Another problem we encountered with (B.27) was an oscillation ("chattering") in the size of the time step about a gradually varying average value. In order to prevent these two problems from occurring, we have replaced (B.27) with a more complex set of equations. The rationale for this set is as follows. Limits placed upon the rate at which the time step can grow or shrink will tend to prevent the problem discussed by Burden *et al.* Eliminating the "chattering" is more difficult. The most effective procedure we have devised to date sets limits upon the value of χ_i depending upon its value at the two previous time steps χ_{i-1}, and χ_{i-2}. Thus for example, if the two previous values of χ were less than 1, then the new time step will be specified by (B.27) provided that $0.2 \le \chi_i \le 1$. If $\chi_i \le 0.2$, then we replace it with

0.2 in (B.27). This will ensure that time steps do not become too small, too rapidly. If on the other hand (B.28) indicates that a value greater than 1 is required, the following equation is used for finding the new time step:

$$\Delta t_{i+1} = \hat{k}\Delta t_i[(\chi_i - 1)/10 + 1] \qquad (B.30)$$

This dampens the rapid growth of the time step immediately after two time steps which have decreased in size. The complete set of equations for finding the new time step is: For $\chi_{i-1} \geq 1$, and $\chi_{i-2} \geq 1$:

$$\text{If } \chi_{i-1} < 1, \quad \Delta t_{i+1} = k_1\chi_i\Delta t_i \qquad (B.31)$$

$$\text{If } 1 \leq \chi_i < 9, \quad \Delta t_{i+1} = k_1\Delta t_i[(\chi_i - 1)/4 + 1] \qquad (B.32)$$

$$\text{If } \chi_i \geq 9, \quad \Delta t_{i+1} = 3k_1\Delta t_i \qquad (B.33)$$

For $\chi_{i-1} \geq 1$, and $\chi_{i-2} < 1$:

$$\text{If } \chi_i < 1, \quad \Delta t_{i+1} = k_1\chi_i\Delta t_i \qquad (B.34)$$

$$\text{If } 1 \leq \chi_i < 11, \quad \Delta t_{i+1} = k_1\Delta t_i[(\chi_i - 1)/10 + 1] \qquad (B.35)$$

$$\text{If } \chi_i \geq 11, \quad \Delta t_{i+1} = 2k_1\Delta t_i \qquad (B.36)$$

For $\chi_{i-1} < 1$, and $\chi_{i-2} \geq 1$:

$$\text{If } \chi_i \leq 0.8, \quad \Delta t_{i+1} = 0.8\,k_1\Delta t_i \qquad (B.37)$$

$$\text{If } 0.8 < \chi_i \leq 1, \quad \Delta t_{i+1} = k_1\chi_i\Delta t_i \qquad (B.38)$$

$$\text{If } \chi_i > 1, \quad \Delta t_{i+1} = k_1\Delta t_i[(\chi_i - 1)/10 + 1] \qquad (B.39)$$

For $\chi_{i-1} < 1$, and $\chi_{i-2} < 1$:

$$\text{if} \quad \chi_i \le 0.2, \quad \Delta t_{i+1} = 0.2\, k_1 \Delta t_i \tag{B.40}$$

$$\text{if} \quad 0.2 < \chi_i \le 1, \quad \Delta t_{i+1} = k_1 \Delta t_i \chi_i \tag{B.41}$$

$$\text{if} \quad \chi_i > 1, \quad \Delta t_{i+1} = k_1 \Delta t_i [(\chi_i - 1)/10 + 1)] \tag{B.42}$$

When $e_i > \varepsilon$, $i+1$ is replaced by i in (B.23) through (B.34), and the step is re-integrated (GLERK5[86,111]).

It was discovered during the testing of this modified algorithm, that occasionally the automatic step-size routine would hang up at one point in time, unable to find an appropriate step size to continue. This seemed to occur when the air velocity components changed very rapidly. To prevent this problem from terminating execution, the following two modifcations were added. First, the step size was not allowed to decrease below $\Delta t_{i+1} = 5 \times 10^{-4}$. This limit was determined on the basis of trial and error as a reasonable compromise between efficiency and accuracy. Second, if $e_i > \varepsilon$, and yet

$$e_i > 4 e_{i-1} \frac{\min \{|\underline{x}_i|, |\underline{u}_i|, |\underline{y}_i|, |\underline{v}_i|\}}{\min \{|\underline{x}_{i-1}|, |\underline{u}_{i-1}|, |\underline{y}_{i-1}|, |\underline{v}_{i-1}|\}} \tag{B.43}$$

then rather than re-integrating, we set

$$\Delta t_{i+1} = (\Delta t_i + \Delta t_{i+1})/2 \tag{B.44}$$

where Δt_{i+1} on the R.H.S. of (B.44) is taken from (B.31) to (B.42). We then return to the use of (B.31) through (B.42) if possible. If (B.44) must be used to step over a region of difficulty, a "*" is placed in the first column of the output (see Appendix I) (GLERK5[112,128]).

In one last effort to improve the efficiency of the above procedures, it was determined through experimentation that an extrapolated value of Δt_{i+1} given by:

$$\Delta t_{i+1} = \min[\Delta t_{i+1}, 2\Delta t_{i+1} - \Delta t_i] \qquad (B.45)$$

prevented excessive re-evaluation during a continual reduction in step size while maintaining a constant local truncation error estimate (GLERK5[129]). Thus a value of Δt_{i+1} is determined from (B.31) through (B.42). If no difficulty is encountered (that is $e_i > \varepsilon$) then this value is substituted into the RHS of (B.45) and a final value of Δt_{i+1} is obtained.

APPENDIX C. Integrating the history term.

Sections 2.3.4.2 and 2.3.5 outlined the difficulty in integrating the complete integro-differential equations which describe the droplet trajectories. We have developed a technique which may yield a somewhat less accurate value for the contribution of the history term, but which is not as difficult to implement as the classical methods of solving such problems. We justify this approximation on the basis that the history term becomes a significant factor in the droplet acceleration only just prior to a droplet airfoil collision, or around the point of closest approach. Except in these circumsances, the history term has only a minor effect upon the solution to the system in (2.69).

The history term in (2.69) is of the form:

$$\text{const.} \int_0^t \frac{d\bar{v}_d}{d\tau} \frac{d\tau}{\sqrt{t - \tau}} \tag{C.1}$$

We have changed the lower limit of integration from $-\infty$ to 0 because before t=0, the droplet is assumed to be travelling in a constant uniform airflow where there are no accelerations.

Experiments involving the numerical evaluation of the history term integral, using various Newton-Cotes formulae showed that such formulae provided accurate estimates of the integral for all but the portion of the interval where τ approached t. In this interval, substantial errors could result.

As a result of the above discovery, a semi-analytical technique for finding the solution was adopted. It was noted that if the accelerative part of the kernel, that is $d\bar{v}_d/dt$ could be interpolated by a Lagrange polynomial of degree less than or equal to three, then the value of the history term could be approximated from a combination of the following formulae:

$$\int_q^s \frac{\tau^3 d\tau}{\sqrt{t - \tau}} = \frac{2}{35}\Big[(5q^3 + 6q^2 t + 8qt^2 + 16t^3)\sqrt{t - q} - (5s^3 + 6s^2 t + 8st^2 + 16t^3)\sqrt{t - s}\Big] \tag{C.2}$$

$$\int_q^s \frac{\tau^2 d\tau}{\sqrt{t-\tau}} = \frac{2}{15}\left[(3q^2 + 4qt + 8t^2)\sqrt{t-q} - (3s^2 + 4st + 8t^2)\sqrt{t-s}\right] \quad (C.3)$$

$$\int_q^s \frac{\tau d\tau}{\sqrt{t-\tau}} = \frac{2}{3}\left[(2t + q)\sqrt{t-q} - (2t + s)\sqrt{t-s}\right] \quad (C.4)$$

and

$$\int_q^s \frac{d\tau}{\sqrt{t-\tau}} = 2[\sqrt{t-q} - \sqrt{t-s}] \quad (C.5)$$

The formulae necessary for obtaining the coefficients of the Lagrange polynomial are given by Burden *et al*. (1978), for example.

The value of the acceleration at the time step t_{i+1} is obtained by a two-part iterative process. First, for some algorithms (RK4, RKF4) values of the history term are required for times intermediate between t_i and t_{i+1}. These are obtained by extrapolation using a third order Lagrange polynomial fitted to the history term values for the time steps $t_{i-1,i-2,i-3,i-4}$ (except for the first few time steps, when a lower order Lagrange polynomial is used). Then, extrapolation from the previously fitted polynomial is used to predict the value of the history term at t_{i+1}. This allows the calculation of the accelerations at t_{i+1}. With an estimate of $\partial\bar{v}_d/\partial t$ at t_{i+1} known, we may interpolate $\partial\bar{v}_d/\partial t$ between t=0 to $t = t_{i+1}$. If i is odd, this is accomplished by a sequence of second degree Lagrange polynomials over successive triplets of time steps. If i is even, the procedure is the same, but with a third degree Lagrange polynomial used over the last four points. In this way we retain the greatest interpolation accuracy in the time steps just past, that is, those which contribute the most to the history term.

APPENDIX D. Integrating ordinary differential equations by a Hermite extrapolation technique.

Section 2.3.7 pointed out the problem that occurs when a droplet approaches the airfoil surface and it becomes necessary to determine whether or not a collision has actually occurred, and if so at what point. In that section it was stated that the higher order integrations employed in this thesis all used air velocity values within the time interval $(t_i, t_{i+1}]$ to determine the position and velocity of the droplet at t_{i+1}. Careful scrutiny of these integrators reveals:

1. The RK4 algorithm must calculate the air velocity at points approximately midway between $(x_d, y_d)_i$ and $(x_d, y_d)_{i+1}$, as well as near $(x_d, y_d)_{i+1}$ in order to determine the value of (x_d, y_d) at t_{i+1}.
2. The PC4 algorithm due to Hamming (1973) uses the modified estimate from the predictor to calculate the acceleration at t_{i+1}, for use in the corrector. This acceleration is based upon knowledge of the air velocity near $(x_d, y_d)_{i+1}$.
3. The RKF4 algorithm calculates the air velocity at points approximately 0.25, 0.375, 0.5 and 0.923 of the distance between $(x_d, y_d)_i$ and $(x_d, y_d)_{i+1}$ as well as near $(x_d, y_d)_{i+1}$.

If any of the gridpoints used to find the air velocity (see Fig. 5) in the interval $(t_i, t_{i+1}]$ lie within the airfoil profile, the corresponding streamfunction value will be meaningless. This will lead to an incorrect value for the air velocity, and thus will adversely affect the accuracy of the droplet position and velocity at time t_{i+1}.

The problem is resolved by extrapolating forward from the position and velocity of the droplet at t_i and t_{i-1} instead of using values in the interval $(t_i, t_{i+1}]$. For an equation of the form of (2.77), the Hermite extrapolator may be expressed as:

$$\bar{x}_{i+1} = -4\bar{x}_i + 5\bar{x}_{i-1} + \Delta t_i (4\dot{\bar{x}}_i + 2\dot{\bar{x}}_{i-1}) \qquad (D.1)$$

This formula has a lower order truncation error than do the other integrators (third-order *vs.* fourth-order for RK4, RKF4, and PC4) and so it is used only to test whether or not the collision has occurred by the time the droplet reaches its position at t_{i+1}. If it has, Section 2.4.3 describes the methods used to find the collision location. If not, the step is re-integrated by one of the higher order methods.

APPENDIX E. Finding the length of a portion of a cubic spline curve.

The following set of solutions has been derived by Phillips (1980). Let us begin with the general cubic polynomial defining the spline segment between X_o and X_1:

$$Y = Yo + a_1\delta^3 + a_2\delta^2 + a_3\delta \tag{E.1}$$

where δ is given by

$$\delta = X - Xo \tag{E.2}$$

Then we have

$$Y' = 3a_1\delta^2 + 2a_2\delta + a_3 \tag{E.3}$$

The length of a curve between X_o and X may be expressed as:

$$L(Xo,X) = \int_{Xo}^{X} \sqrt{1 + (Y')^2}\, dX \tag{E.4}$$

or

$$L(\delta) = \int_{0}^{\delta} \sqrt{1 + (3a_1\delta^2 + 2a_2\delta + a_3)^2\, d\delta} \tag{E.5}$$

We must test now for the values of a_1 and a_2. This will lead to three separate solutions:

1. If $a_1 = a_2 = 0$, then

$$L(\delta) = \delta\sqrt{1 + a_3^2} \tag{E.6}$$

2. If $a_1 = 0$ but $a_2 \neq 0$, then (E.5) may be rewritten as:

$$L(\delta) = \int_{0}^{\delta} \sqrt{1 + (2a_2\delta + a_3)^2\, d\delta} \tag{E.7}$$

The solution to the integral is given by:

$$L(\delta) = \{(2a_2\delta + a_3)\sqrt{1 + (2a_2\delta + a_3)^2} - a_3\sqrt{1 + a_3^2}$$
$$+ \ln[(2a_2\delta + a_3) + \sqrt{1 + (2a_2\delta + a_3)^2}] - \ln[a_3 + \sqrt{1 + a_3^2}]\}/4a_2 \qquad (E.8)$$

3. If $a_1 \neq 0$, then let

$$V = \sqrt{3|a_1|}\ (\delta + a_2/3a_1) \qquad (E.9)$$

$$\nu_1 = \sqrt{3|a_1|}\ (\delta + a_2/3a_1) \qquad (E.10)$$

and

$$\nu_o = a_2\sqrt{3|a_1|}\ /3a_1 \qquad (E.11)$$

where

$$\delta_1 = X_1 - X_o \qquad (E.12)$$

A change of variable allows us to write (E.5) in the form:

$$L = \frac{1}{\sqrt{3|a_1|}} \int_{\nu_o}^{\nu_1} \sqrt{1 + (V^2 + \Delta)^2\ dV} \qquad (E.13)$$

$$= \frac{1}{\sqrt{3|a_1|}}\ [I(\nu_1) - I(\nu_o)] \qquad (E.14)$$

where

$$\Delta = (a_3 - a_2^2/3a_1)\,\mathrm{sgn}(a_1) \qquad (E.15)$$

The integral $I(\nu)$ in (E.14) is given by

$$I(\nu) = \int_0^{\nu} \sqrt{1 + (V^2 + \Delta)^2\ dV}$$
$$= \frac{\nu}{3}\sqrt{1 + (V^2 + \Delta)^2}\left(1 + \frac{2\Delta G^2}{1 + V^2G^2}\right) + \frac{1}{3G^3}\left[(1 + \Delta G^2)F(\zeta,k) - 2\Delta G^2 E(\zeta,k)\right] \qquad (E.16)$$

where

$$G = (1 + \Delta^2)^{-0.25} \tag{E.17}$$

$$k = \sqrt{(1 - \Delta G^2)/2} \tag{E.18}$$

and

$$\zeta = \tan^{-1}\left[\frac{2G\nu}{1 - \nu^2 G^2}\right] \tag{E.19}$$

The functions $F(\zeta,k)$ and $E(\zeta,k)$ in (E.16) are the incomplete elliptic integrals of the first and second kind respectively. If

$$\nu = 1/G \tag{E.20}$$

then

$$\zeta = \pi/2 \tag{E.21}$$

which allows us to replace $F(\zeta,k)$ and $E(\zeta,k)$ in (E.16) by $K(k)$ and $E(k)$, the complete elliptic integrals of the first and second kind respectively. If $0 \leq \nu \leq 1/G$, we may use (E.16) directly. If $1/G < \nu < \infty$, then we replace $F(\zeta,k)$ and $E(\zeta,k)$ in (E.16) by

$$F(\zeta,k) = 2K(k) - F(\pi - \zeta,k) \tag{E.22}$$

and

$$E(\zeta,k) = 2E(k) - E(\pi - \zeta,k) \tag{E.23}$$

In the present program, these elliptic integrals are evaluated using the subroutines DELI1, DELI2, DCEL1, and DCEL2 from the SSPLIB subroutine package provided by IBM.

APPENDIX F. Locating points on the interpolated airfoil surface.

In Section 2.4.1 mention was made of the need for an iterative process to determine the ordinate value $\hat{y}$ of an interpolated point on the airfoil surface when the abscissa is given as $\hat{x}$. If the rotated coordinate system is denoted by the subscript R, then the equations relating a point in the two coordinate systems are:

$$x_R = x \cos 30^\circ + y \sin 30^\circ \tag{F.1}$$

and

$$y_R = y \cos 30^\circ - x \sin 30^\circ \tag{F.2}$$

The interpolation equations were formulated on the rotated coordinate system for reasons outlined in Section 2.4.1. From (F.1) and (F.2) it is apparent that if only a value for x is known ($\hat{x}$), we cannot interpolate for $\hat{y}$ until we are able to determine $\hat{x}_R$.

To overcome this problem, we begin by fitting a straight line between the surface segment endpoints SSE i and SSE i+1 with coordinates (x_i, y_i) and (x_{i+1}, y_{i+1}) respectively (see Fig. 8.1). The ordinate value on this line for the abscissa $\hat{x}$ is

$$y_G = y_i + (\hat{x} - x_i)(y_{i+1} - y_i)/(x_{i+1} - x_i) \tag{F.3}$$

Now that y_G is known, we may substitute into (F.1) and (F.2) to find the point on the airfoil surface (x_I, y_I) for the rotated x value x_{RG}. This is our first approximation to $(\hat{x}, \hat{y})$. Since the cubic spline interpolator may be differentiated with respect to x_R, we may use the Newton–Raphson algorithm to iterate on successive values of x_{RG} until $x_I - \hat{x}$ becomes sufficiently small. The inverse pair of equations from (F.1) and (F.2) may then be used to give this approximated value for $\hat{y}$.

APPENDIX G. The program listing.

```
C
C WRITTEN BY: M  OLESKIW  ON 790526  LAST MODIFIED:811024
C
C CALCULATE POTENTIAL FLOW ABOUT AN ARBITRARILY SHAPED AEROFOIL:
C   CALCULATE A SERIES OF DROPLET TRAJECTORIES AND
C   DETERMINE THE COLLISION LOCATIONS: FIND THE RESULTING COLLISION
C   EFFICIENCY AND ACCRETE A LAYER OF ICE
C   REPEAT THE PROCESS FOR A PREDETERMINED NUMBER OF STEPS.
C
C INTERNAL SUBROUTINES AND FUNCTIONS:
C     ACCN: CALCULATES RHS OF NON-DIMENSIONAL EQNS  OF MOTION
C     AIRPLT: PLOTS OUTLINE OF AIRFOIL WITHIN VIEW WINDOW.
C     AIRVEL: CALCULATES THE AIR VELOCITY COMPONENTS AT A
C       GIVEN LOCATION
C     CE: CALCULATES AND PLOTS COLLISION EFFICIENCY CURVES OF
C       ARBITRARY AIRFOILS BY DETERMINING A SET OF
C       IMPACTING TRAJECTORIES.
C     COLVEL: INTERPOLATES DROPLET IMPACT VELOCITY
C       ALONG AIRFOIL SFC
C     COORDS: CALCULATES A SET OF POINTS DEFINING THE AIRFOIL SFC.
C     DRAG: CALCULATES THE REYNOLDS NUMBER AND
C       DRAG COEFFICIENT OF THE DROPLET
C     FIT: ROTATES UPPER AND LOWER SFCS.  IF REQUIRED, FIT
C       CALCULATES CUBIC SPLINES AND DETERMINES LENGTHS
C       ALONG THE AIRFOIL SFC  TO EACH ENDPOINT.
C       CALCULATES THICKNESS OF ACCRETION.
C     GLERK5: INTEGRATES THE DROPLET EQNS. OF MOTION
C       (IN X AND Y ) USING
C       1: A 4TH ORDER RUNGE-KUTTA-FEHLBERG TECHNIQUE
C       2: ORDER EXTRAPOLATION OF THE ABOVE.
C       3: STEP EXTRAPOLATION OF THE ABOVE (5TH ORDER ACCURACY).
C     GROWTH: PLOTS SUCCESSIVE AIRFOIL OUTLIENS WITHIN VIEW WINDOW
C     HERM5: CALCULATES COEFFICIENTS FOR HERMITE QUINTIC SPLINES.
C     HERMIT: CALCULATES THE HERMITE CUBIC POLYNOMIAL
C       INTERPOLATOR GIVEN THE FUNCTION AND ITS DERIVATIVES
C       AT THE ENDPTS  OF THE INTERVAL.
C     HIST: DETERMINES VALUE OF INTEGRAL IN HISTORY TERM
C     ICING: CALCULATES AMOUNT OF ACCRETION AND DETERMINES A NEW
C       SET OF AIRFOIL SURFACE ELEMENT ENDPOINTS AFTER DETERMINING
C       THE AIRFOIL NOSE LOCATION
C     JTHICK: CALCULATES THE NEGATIVE OF THE THICKNESS OF THE
C       JOUKOWSKI AIRFOIL AS A FUNCTION OF THETA AND E.
C     NSURF: CALCULATES THE UNROTATED X VALUE OF A POINT ON THE
C       ACCRETED AIRFOIL SFC. BASED UPON THE COLLISION EFFICIENCY,
C       DIRECTION OF GROWTH, AND OLD AIRFOIL (ROTATED) SFC. POSITION.
C     PC4: INTEGRATES THE EQNS. OF MOTION USING THE 4TH ORDER
C       PREDICTOR-CORRECTOR METHOD OF HAMMING.
C     PSK: CALCULATES ANALYTICAL VALUE OF STREAMFN. AT TRANSFORMED
C       COORDS. X,Y USING THE EXACT AIRFOIL GENERATION METHOD.
C     PLTSZ: DETERMINES PARAMETERS NECESSARY FOR SCALING OF A
C       PLOT AND ITS AXES.
C     POT1: SOLVES FOR SURFACE VORTEX DENSITY ON A ONE-ELEMENT
C       AIRFOIL IN POTENTIAL FLOW, GIVEN THE COORDS. OF THE
C       AIRFOIL SFC.
C     RK4: INTEGRATES THE DROPLET EQNS. OF MOTION (IN X AND Y)
C       USING THE 4TH ORDER RUNGE-KUTTA TECHNIQUE.
C     SFC: CALCULATES Y VALUES AND THE LENGTH FROM THE NOSE ON THE
C       SFC. OF THE AIRFOIL BY A CUBIC SPLINE INTERPOLATION.
C     SFCLEN: CALCULATES THE LENGTH ALONG A SEGMENT OF THE CUBIC
C       SPLINE FIT OF THE AIRFOIL SFC.
C     STAB: FINDS THE JACOBIAN (DF/DY), ITS EIGENVALUES AND
C       DETERMINES SUITABILITY OF ODE INTEGRATING TECHNIQUES.
C     STRMFN: CALCULATES THE STREAMFN. ON A GRID ABOUT AN AIRFOIL
```

```
C       SECTION GIVEN THE SFC. VORTICITY DENSITY ON THE AIRFOIL
C       AND PLOTS THE FLOW USING VELOCITY VECTORS.
C     TRAJEC: CALCULATES TRAJECTORIES OF DROPLETS IN POTENTIAL FLOW
C       ABOUT AN AIRFOIL.
C     WHAMO: DETERMINES CLOSEST APPROACH BETWEEN DROPLET AND
C       AIRFOIL SFC.
C
C EXTERNAL SUBROUTINES:
C   IN IMSL  (INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARY)
C     LEQT1F: SOLVES SYSTEM OF EQNS
C     ICSICU: CUBIC SPLINE INTERPOLATION
C     ZXGSN: GOLDEN SECTION SEARCH METHOD FOR FINDING FN. MINIMUM.
C     VSRTRD: SORT A VECTOR SO THAT ELEMENTS ARE IN INCREASING ORDER.
C     EIGRF: FIND THE COMPLEX EIGENVALUES OF A MATRIX.
C     IQHSCU. CALCULATE COEFFICIENTS OF A QUASI-HERMITE
C       INTERPOLATING POLYNOMIAL
C
C   IN SSPLIB  (SCIENTIFIC SUBROUTINE LIBRARY - SUPPLIED BY IBM)
C     DELI1: INCOMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND.
C     DELI2  INCOMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND.
C     DCEL1  COMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND.
C     DCEL2: COMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND
C
C INPUT/OUTPUT DEVICE ASSIGNMENTS
C     2  AIRFOIL INPUT COORDINATES (IF TYPE=4 OR TYPE=5)
C     3: DATA READ BY SUBPROGRAM PLTSZ TO SCALE PLOTS.
C     4: PROGRAM INPUT PARAMETERS AND OPTIONS (DESCRIBED BELOW).
C     5: INPUT CRT DEVICE FOR CONTROL OF PROGRAM
C     6: OUTPUT CRT DEVICE FOR MONITORING OF PROGRAM.
C     7: OUTPUT HARDCOPY DEVICE FOR PRINTED OUTPUT.
C     8: OUTPUT OF SURFACE SEGMENT ENDPOINTS FOR EACH ACCRETED
C        SURFACE IN FORMAT SUITABLE FOR SUBSEQUENT
C        INPUT INTO DEVICE 2
C     9: OUTPUT FILE FOR STORAGE OF PLOT DESCRIPTION
C        (CALCOMP FORMAT).
C
C PROGRAM INPUT PARAMETERS AND OPTIONS:
C     TO BE READ IN FROM INPUT DEVICE 4.  EACH GROUP OF PARAMETERS
C     IS TO BE READ FROM THE SAME LINE (CARD) USING THE SPECIFIED
C     FORMAT.  EACH DATA LINE PRECEDED BY A DESCRIPTIVE REMINDER
C     LINE. SEE APPENDIX I FOR DETAILS.
C
C
C OPTIONS AND DATA  (SEPARATED BY COMMAS) (SYMBOLS IN BRACKETS
C AT ENDS OF LINES REFER TO FORTRAN FORMAT TYPE)
C    ALPHA=ANGLE OF ATTACK IN DEGREES(F)
C    TYPE=AIRFOIL TYPE(I)
C        -11:ANALYTICAL PARABOLA AS A 3-D BODY OF REVOLUTION ABOUT
C            THE CHORD.  NOTE: STRMFN UNDEFINED AT CHORD LINE.
C        -10:ANALYTICAL FLYING CIGAR AS A 3-D BODY OF REVOLUTION
C            ABOUT THE CHORD
C         -3:ANALYTICAL JOUKOWSKI AEROFOIL (APPROXIMATE)
C         -2:ANALYTICAL JOUKOWSKI AEROFOIL (EXACT)
C         -1:ANALYTICAL CYLINDER
C          0:NACA RAZOR
C          1:CYLINDER (VORTEX SHEETS)
C          2:JOUKOWSKI (VORTEX) (EXACT)
C          3:JOUKOWSKI (VORTEX) (APPROXIMATE)
C          4:INPUT X AND Y COORDS FOR UPPER SFC. OF SYMMETRICAL
C            AEROFOIL
C          5:INPUT X AND Y COORDS FOR BOTH SFCS OF ASYMMETRICAL
C            AEROFOIL
C    THICK=THICKNESS OF AIRFOIL IN PERCENT (F)
C    MEAN=NACA DESIGNATION FOR MEAN LINE IN 4 & 5 DIGIT AEROFOILS (I)
C    NEF=NO. OF CEE'S ON FRONT THIRD OF AEROFOIL (I)
C    NEB=NO  OF CEE'S ON BACK TWO-THIRDS OF AEROFOIL (I)
```

```
C        (INCLUDES THE ENDPT  AT THETA=90 DEG )
C     NIF=NO. OF SSE'S BETWEEN CEE'S ON FRONT THIRD (I)
C     ANAL=0:ESTIMATE SEGMENT LENGTH NUMERICALLY
C          1:DETERMINE SEGMENT LENGTH BY ANALYTICAL METHOD (APPENDIX E).
C     PLTFAC=PLOT REDUCTION OR EXPANSION FACTOR FOR ALL PLOTS (F)
C
C     UINF=FREESTREAM VELOCITY [M/S] (F)
C     C=CHORD LENGTH [M] (F)
C     TINF=FREESTREAM TEMPERATURE [C] (F)
C     PINF=FREESTREAM PRESSURE [KPA] (F)
C     VINQ=DETERMINE AIR VELOCITY COMPONENTS AT INPUT COORDS X & Y
C          (0 OR 1)
C
C     TRJPLA=0 NO TRAJECTORY PLOTS
C            1 PLOT TRAJECTORIES ONLY FOR FIRST LAYER
C            2 PLOT TRAJECTORIES FOR ALL LAYERS
C     XMIN=
C     XMAX= TRAJECTORY VIEWPORT SIZE IN X (F)
C     YMIN=
C     YMAX= TRAJECTORY VIEWPORT SIZE IN Y (F)
C     XZ= GRID SIZE IN X (I)
C     YZ= GRID SIZE IN Y (I)
C     XMINI=
C     XMAXI=ICE ACCRETION VIEWPORT SIZE IN X (F)
C     YMINI=
C     YMAXI=ICE ACCRETION VIEWPORT SIZE IN Y (F)
C
C     EQN=0: EQN. OF MOTION INCLUDES TERMS A AND B (NO INDUCED
C            MASS OR BUOYANCY)
C         1: EQN. OF MOTION INCLUDES TERMS APRIME AND BPRIME
C         2: EQN. OF MOTION INCLUDES TERMS APRIME, BPRIME, AND
C            CPRIME (HISTORY TERM)
C     PC=0:INTEGRATE BY RUNGE-KUTTA
C        1:INTEGRATE BY PREDICTOR-CORRECTOR (AFTER FIRST 3 INTERVALS)
C        2:INTEGRATE BY RUNGE-KUTTA-FEHLBERG
C     ACN=0:INITIAL DROPLET VELOCITY GREATER THAN THAT OF AIR
C           BUT IN THE SAME DIRECTION
C         1 INITIAL DROPLET VELOCITY DIFFERS FROM THAT OF AIR
C           AS PER LOCAL AIR ACCN
C     GRAV=INCLUDE GRAVITATIONAL ACCN (0 OR 1)
C     CDS=0:ABRAHAM (1970) CD
C         1:SARTOR & ABBOTT (1975) CD FOR 0.01<RED<5
C           STOKES CD FOR RED<0.01
C         2 LANGMUIR & BLODGETT (1945) CD
C     TRJPRA=PRINT TRAJCTORY INFO (0 OR 1)
C     PRINTO=NO. OF PRINT POINTS IN VIEWPORT DIAGONAL LENGTH
C           (OUTSIDE VIEWPORT (I))
C     PRINTI=NO. OF PRINT POINTS IN VIEWPORT DIAGONAL LENGTH
C           (WITHIN VIEWPORT (I))
C
C     DDISTN=NUMBER OF DROPLET SIZES IN DROPLET DISTRIBUTION (I)
C     DD & W=DROPLET DIAMETERS (IN MICROMETERS) AND FRACTIONAL
C            WEIGHTS FOR DROPLET DISTN. (ALTERNATELY) (F,F)
C
C     EPS= LOCAL ERROR IN ODE INTEGRATION DIVIDED BY STEP SIZE
C          FOR EACH DROPLET SIZE IN DISTRIBUTION  (D)
C
C     AT=0:START TRAJECTORIES AS SPECIFIED BY DD, EPS, XO, YO
C        1:AUTOMATICALLY DETERMINE TRAJECTORY STARTING POINTS
C          AFTER FIRST ONE FOR EACH SFC
C     CEDEL=CRITERION FOR MAX. % DIFFERENCE BETWEEN TWO REALIZATIONS
C           OF CE VS L CURVE (F)
C     EMDEL=CRITERION FOR MAX. % DIFFERENCE BETWEEN E MAX AS PER
C           INTEGRATION OF BETA, AND DISTANCE BETWEEN GRAZING TRAJ. (F)
C     HS=0:YO VS L CURVE INTERPOLATED BY HERMITE CUBIC POLYNOMIALS.
C        1:YO VS L CURVE INTERPOLATED BY HERMITE QUINTIC SPLINE.
```

```
      C     YOL=PLOT THE YO VS L GRAPH (0, 1, OR 2) (2 PLOTS AT HALF PAGE SIZE)
      C     CEL=PLOT THE CE VS L GRAPH (0,1,2,3,OR 4)
      C          (2 AND 4 PLOT AT HALF PAGE SIZE; 3 AND 4 ALSO PLOT MEAN
      C          CE VS L CURVE WHEN THERE IS A DROPLET DISTRIBUTION,
      C          OR IF SMOOTHING IS PERFORMED)
      C     CEX=PLOT THE CE VS X GRAPH (0, 1, OR 2) (2 PLOTS AT HALF PAGE SIZE)
      C     FILTER=LENGTH OF BOXCAR FILTER(AS A FRACTION OF L RANGE OF
      C             LARGEST DROPLET SIZE) TO BE APPLIED TO SMOOTH CE VS L
      C             CURVE(S)   IF 0, THEN DON'T FILTER  (F)
      C     LLEFT=LEFTMOST POINT TO BE PLOTTED IN YO VS L AND CE VS L
      C           CURVES   IF 0, DETERMINE AUTOMATICALLY  (F)
      C     LRIGHT=RIGHTMOST POINT AS ABOVE  (F)
      C
      C     ICEPLA=0  NO PLOT
      C             1  PLOT AEROFOIL & ICE LAYERS
      C     LYRMAX=MAX NUMBER OF LAYERS TO ACCRETE (I)
      C     ICE=FRACTION OF CHORD LENGTH TO BE ACCRETED PER LAYER ASSUMING
      C          A COLLISION EFFICIENCY OF 100% (F)
      C     LTOL=MAX  INCREASE IN LENGTH ALLOWED BETWEEN CEE'S
      C           BETWEEN SUCCESSIVE AIRFOIL SURFACES (F)
      C     ATHICK=0 CALCULATE ACCRETION THICKNESS ASSUMING FLAT SFC
      C               LOCALLY
      C             1 ACCOUNT FOR SFC  CURVATURE IN CALCULATING ACCRETION
      C               THICKNESS (IF ATHICK=-1, CALCULATE RADIUS OF
      C               CURVATURE FROM SPLINE FIT AT THAT POINT ONLY.)
      C     DENSE=0 CONSTANT ICE DENSITY
      C            1 VARY ICE DENSITY ACCORDING TO NORMAL COMPONENT
      C              OF DROPLET IMPACT VELOCITY
      C            2 VARY ICE DENSITY ACCORDING TO TOTAL DROPLET IMPACT VEL.
      C
      C     XO=X (UPSTREAM) COORD  FOR TRAJECTORY STARTING PTS. (F)
      C     YO=Y (OFF AXIS) COORDS FOR TRAJECTORY STARTING POINTS. (F)
      C        INPUT ONE SET FOR EACH SFC., IF BOTH EQUALS 1.
      C
 1    5     FORMAT(/,F6.0,I5,F6.0,I5,3I4,I5,F7.0)
 2    10    FORMAT(/,F7.0,F6.0,F7.0,F6.0,I5)
 3    15    FORMAT(/,I7,4F5.0,2I3,4F6.0)
 4    20    FORMAT(/,I4,I3,I4,I5,I4,3I7)
 5    25    FORMAT(/,I7,5(F6.0,F5.0))
 6    26    FORMAT(/,5D10.0)
 7    30    FORMAT(/,I3,2F6.0,I3,3I4,F7.0,F6.0,F7.0)
 8    35    FORMAT(/,2I7,F6.0,F5.0,I7,I6)
 9    40    FORMAT(I4)
10    50    FORMAT(I2,2F19.16)
11    55    FORMAT('0ENTER X & Y ')
12    60    FORMAT(2F10.2)
13    70    FORMAT(' VELOCITY COMPONENTS: U=',F9.5,'   V=',
           .F9.5,'   TOTAL VELOCITY  ',F9.5)
14    80    FORMAT('1',T26,'DISTANCE',T54,'DISTANCE',/,
           .'  END',T28,'FROM',T56,'FROM',/,
           .' POINT   X COORD   Y COORD    NOSE      X COORD   Y COORD    NOSE',/)
15    85    FORMAT('  ',I4,2(F10.5,2F9.5))
      C
16          DOUBLE PRECISION ALPHA,XE(101),YE(101),UINF,C,TINF,PINF,
           .PI,X,DFLOAT,LTOL,ICE,ACCRT,EPS(5),DD(5),W(5),LU(101),LL(101),
           .XU(101),YU(101),XL(101),YL(101),THICK,U,V,VV,TH,FS,
           .XN,YN,ALPHAR,THETA,XMINI,XMAXI,YMINI,YMAXI,DSORT,DABS,FILTER
      C
17          REAL XMAX,XMIN,YMIN,YMAX,PLTFAC,CEDEL,EMDEL,LLEFT,LRIGHT
      C
18          INTEGER I,J,TYPE,XZ,YZ,TRJPLA,NCOU,NCOL,EON,PC,ACN,TRJPRA,
           .PLT,LAYER,LYRMAX,NCOL1,CEL,YOL,ICEPLA,AT,BOTH,FAIL,ANAL,
           .ATYPE,IABS,IU(51),IL(51),NEB,NEF,NIF,NIFP1,CEX,II,IJ,NEU,NEL,
           .IXU(101),IXL(101),PRINTO,PRINTI,DDISTN,CDS,AMAXO,IK,
           .ATHICK,DENSE,VINO,JZ,GRAV,MEAN,H5,NEUU,NELL,H5D(5)
      C
```

```
      COMMON ALPHAR,PI/AERO1/XE,YE/NOSE/XN,YN
     ./LA/ANAL/AERO3/NCOU,NCOL/HERMT5/H5,H50/LG/LU,LL
     ./GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ/SFCS/XU,YU,XL,YL
     ./AERO4/NEU,NEL,NEUU,NELL/ENDS/IU,IL,IXU,IXL
     ./LLR/ACCRT,LAYER,ATYPE/TRANS1/UINF,PINF,TINF,EPS,DENSE/NACA/TH
     ./TRANS2/CDS,TRJPRA,PRINTI,PRINTO,EON,
     .PC,ACN,GRAV/WTS/W/TRANS3/DO,C,TYPE,JZ/CRITS/CEDEL,EMDEL
C
C INPUT PARAMETERS
      READ(4,5)ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC
      READ(4,10)UINF,C,TINF,PINF,VINO
      READ(4,15)TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,
     .YMINI,YMAXI
      READ(4,20)EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI
      READ(4,25)DDISTN,(DO(I),W(I),I=1,DDISTN)
      READ(4,26)(EPS(I),I=1,DDISTN)
      READ(4,30)AT,CEDEL,EMDEL,H5,VOL,CEL,CEX,FILTER,LLEFT,LRIGHT
      READ(4,35)ICEPLA,LYRMAX,ICE,LTOL,ATHICK,DENSE
C
      PI=3.141592653589793
      ALPHAR=ALPHA*PI/1.8D2
      BOTH=0
      IF(TYPE.EQ.5.OR.MEAN.NE.0
     ..OR.DABS(ALPHAR).GT.1.D-5)BOTH=1
      ACCRT=0.D0
      TH=THICK
      ATYPE=IABS(TYPE)
C DETERMINE PARAMETERS FOR JOUKOWSKI AEROFOILS
      IF(ATYPE.EQ.2)CALL JOUKEX(THICK)
      IF(ATYPE.EQ.3)CALL JOUKAP(THICK)
C DETERMINE PARAMETERS FOR NACA MEAN LINE.
      IF(TYPE.EQ.0)CALL KOORDS(MEAN)
      IF(TYPE.NE.4.AND.TYPE.NE.5)GOTO 200
C
C READ IN X AND Y COORDS. DEFINING THE AEROFOIL SFC.
      NCOU=0
      READ(2,40)NEU
      NEUU=NEU
      NELL=NEUU
      IJ=1
        DO 300 I=1,NEU
        READ(2,50)IXU(I),XU(I),YU(I)
        IF(IXU(I).EQ.0)GOTO 220
        NCOU=NCOU+1
        IU(IJ)=I
        IL(IJ)=I
        IJ=IJ+1
220     IF(TYPE.EQ.5)GOTO 300
        IXL(I)=IXU(I)
        XL(I)=XU(I)
        YL(I)=YU(I)
300     CONTINUE
        NCOL=NCOU
      IF(TYPE.EQ.4)GOTO 210
      IJ=1
      NCOL=0
      READ(2,40)NEL
      NELL=NEL
        DO 310 I=1,NEL
        READ(2,50)IXL(I),XL(I),YL(I)
        IF(IXL(I).EQ.0)GOTO 310
        NCOL=NCOL+1
        IL(IJ)=I
        IJ=IJ+1
310     CONTINUE
      GOTO 210
```

```
C
C CALCULATE AEROFOIL COORDS.
200    IF(ATYPE.EQ.1)FS=PI/2.DO
       IF(ATYPE.NE.1)FS=PI/3.DO
       NIFP1=NIF+1
       IJ=1
       NCOU=NEF+NEB
       NCOL=NCOU
         DO 110 I=1,NEF
          IU(I)=IJ
          IL(I)=IJ
            DO 140 J=1,NIFP1
            THETA=FS*DFLOAT((I-1)*NIFP1+J-1)/DFLOAT(NEF*NIFP1)
            CALL COORDS(TYPE,THICK,THETA,XU(IJ),XL(IJ),YU(IJ),YL(IJ))
            IJ=IJ+1
140         CONTINUE
110       CONTINUE
          IF(ATYPE.EQ.1)NEUU=IJ
          DO 150 I=1,NEB
          THETA=FS+(PI-FS)*DFLOAT(I-1)/DFLOAT(NEB-1)
          CALL COORDS(TYPE,THICK,THETA,XU(IJ),XL(IJ),YU(IJ),YL(IJ))
          IU(NEF+I)=IJ
          IL(NEF+I)=IJ
          IJ=IJ+1
150       CONTINUE
       NEU=IJ-1
       NEL=NEU
       IF(ATYPE.NE.1)NEUU=NEU
       NELL=NEUU
210    LAYER=1
       XN=XU(1)
       YN=YU(1)
C
       PLT=TRJPLA+YOL+CEL+CEX+ICEPLA
C TRANSFORM THESE COORDS  TO ONE VECTOR OF LENGTH NCOU+NCOL-1
C    IN CLOCKWISE ORDER, WITH XE(1)=XE(NCOL+NCOU-1) - THE LEADING PT.
100       DO 102 I=1,NCOU
          II=IU(I)
          XE(I)=XU(II)
          YE(I)=YU(II)
102       CONTINUE
       NCOL1=NCOL-1
          DO 104 I=1,NCOL1
          J=NCOU+NCOL-I
          II=IL(I)
          XE(J)=XL(II)
          YE(J)=YL(II)
104       CONTINUE
C
C SAVE COORDS OF LATEST LAYER.
       IF(LAYER.LE.1)GOTO 106
       WRITE(8,40)NEU
          DO 380 I=1,NEU
          WRITE(8,50)IXU(I),XU(I),YU(I)
380       CONTINUE
       WRITE(8,40)NEL
          DO 390 I=1,NEL
          WRITE(8,50)IXL(I),XL(I),YL(I)
390       CONTINUE
C
C FIT SPLINES TO UPPER & LOWER SFCS.
106    IF(LAYER.EQ.1)CALL FIT(BOTH)
       IF(LAYER.GT.LYRMAX)GOTO 370
C DETERMINE VORTICIES TO GENERATE FLOWFIELD.
       IF(TYPE.GE.O)CALL POT1
       IF(PLT.EQ.O)GOTO 121
```

```
          IF(LAYER GT 1)GOTO 125
C
C OPEN PLOTTING
          CALL PLOTS
          CALL METRIC(1)
          CALL ORGEP(5 O.5 O.5 O)
          CALL FACTOR(PLTFAC)
C
 125      IF(TRJPLA EQ O OR (TRJPLA EQ 1 AND.LAYER GT 1))GOTO 121
C PLOT VELOCITY VECTORS AND AEROFOIL SHAPE
          CALL STRMFN(TYPE)
          CALL AIRPLT(XMIN,XMAX,YMIN,YMAX,LAYER,O)
 121      IF(VINO EQ 1)GOTO 350
          IF(AT EQ 1)GOTO 130
          CALL TRAJEC(TRJPLA,THICK,AT,BOTH,DDISTN,LAYER,O)
          GOTO 360
C STORE COORDS  OF ICING SHAPE
 130      IF(ICEPLA EQ 1)CALL AIRPLT(XMINI,XMAXI,YMINI,YMAXI,LAYER,1)
C DETERMINE COLLISION EFFICIENCIES
          CALL CE(YOL,CEL,CEX,PLTFAC,THICK,LAYER,DDISTN,BOTH,AT,TRJPLA,
     .    FILTER,LLEFT,LRIGHT)
C DETERMINE COLLISION IMPACT VELOCITIES
          IF(DENSE NE O)CALL COLVEL(DDISTN)
C ACCRETE ICE LAYERS
          LAYER=LAYER+1
          CALL ICING(LTOL,ICE,BOTH,FAIL,DDISTN,ATHICK,FILTER)
          IF(LAYER GT LYRMAX AND ICEPLA EQ O)GOTO 360
          IF(FAIL EQ 1)GOTO 360
          GOTO 100
C
C FIND VELOCITY COMPONENTS AT ARBITRARY X & Y
 350      JZ=1
 355      WRITE(6,55)
          READ(5,60)X,Y
          IF(DABS(X) LT 1 D-10 AND.DABS(Y).LT.1.D-10)STOP
          CALL AIRVEL(X,Y,U,V,4)
          VV=DSQRT(U*U+V*V)
          WRITE(6,70)U,V,VV
          GOTO 355
C
 370      CALL AIRPLT(XMINI,XMAXI,YMINI,YMAXI,LAYER,1)
C PLOT THE ICING LAYERS
          CALL GROWTH(XMINI,XMAXI,YMINI,YMAXI,LYRMAX,PLTFAC)
 360      IF(PLT NE O)CALL PLOT(O.,O.,999)
C WRITE OUT THE NEW AIRFOIL COORDS.
          NEU=AMAXO(NEU,NEL)
          WRITE(7,80)
            DO 400 IK=1,NEU
            WRITE(7,85)IK,XU(IK),YU(IK),LU(IK),XL(IK),YL(IK),LL(IK)
 400        CONTINUE
          STOP
          END

C
C ***********************************************************************
C
          SUBROUTINE ACCN(UD,VD,UA,VA,RED,CD,EON,T,G)
C
C WRITTEN BY: M  OLESKIW  ON: 801216  LAST MODIFIED:810626
C
C CALCULATES RHS OF NON-DIMENSIONAL EQNS. OF MOTION
C
          DOUBLE PRECISION RED,NUS,RDS,APU,APV,BPU,BPV
     .    ,AN(2,6,2),HF,HX,HY,HT(2,6,2),DSQRT,AU,AV,BU,BV,RHOA,
     .    RHOD,GS,ALPHAR,PI,CD,UD,VD,UA,VA,TS(500,2),DTS(6,2),T,
     .    DCOS,DSIN,K2,K3,K4
```

```
C
      INTEGER EQN,G,I(2),IM4(2),IM3(2),IM2(2),IM1(2),IO(2),IP1(2),
     .FNCALL,MM
C
      COMMON ALPHAR,PI/EQNMN/GS,RHOA,RHOD,RDS,NUS,HF
     ./INTEG/AN,HT/LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM
     ./FC/FNCALL/STAB1/K2,K3,K4
C
C IN  UD=
C IN  VD=DROPLET VELOCITY COMPONENTS.
C IN  UA=
C IN  VA=AIR VELOCITY COMPONENTS
C IN  RED=RELATIVE MOTION REYNOLDS NO.
C IN  CD=DRAG COEFFICIENT
C IN  EQN=PARAMETER TO DETERMINE TERMS USED IN EQN. OF MOTION.
C IN  T=TIME AT THIS TIME STEP.
C IN  G=0:EXTRAPOLATE HISTORY TERM SEQUENCE.
C IN    1:CALCULATE NEW HISTORY TERM VALUE.
C
      FNCALL=FNCALL+1
      IF(EQN.EQ.0)GOTO 100
C
C FIRST TWO TERMS IN EQN. OF MOTION INCLUDING GRAVITATION AND
C   STEADY STATE DRAG  (INCLUDES BUOYANCY AND INDUCED MASS EFFECTS)
      APU=K4*GS*DSIN(ALPHAR)
      APV=K4*GS*DCOS(ALPHAR)
      BPU=CD*K3*(UD-UA)*RED/K2
      BPV=CD*K3*(VD-VA)*RED/K2
      AN(1,IP1(MM),MM)=APU-BPU
      AN(2,IP1(MM),MM)=-APV-BPV
      IF(EQN.EQ.2)GOTO 300
      HF=0.D0
      RETURN
C
C THIRD (HISTORY) TERM FOR SHEDDING OF VORTICITY
300   CALL HIST(T,G)
      HX=-9.D0*K3/0.75D0*DSQRT(NUS/PI)*HT(1,IP1(MM),MM)
      HY=-9.D0*K3/0.75D0*DSQRT(NUS/PI)*HT(2,IP1(MM),MM)
      AN(1,IP1(MM),MM)=AN(1,IP1(MM),MM)+HX
      AN(2,IP1(MM),MM)=AN(2,IP1(MM),MM)+HY
      IF(G.EQ.0)RETURN
      HF=DSQRT((HX*HX+HY*HY)/((APU-BPU)**2+(APV+BPV)**2))
      RETURN
C
C FIRST TWO TERMS IN EQN. OF MOTION WITHOUT BUOYANCY AND INDUCED MASS
100   AU=GS*DSIN(ALPHAR)
      AV=GS*DCOS(ALPHAR)
      BU=0.375D0*RHOA/RHOD*CD/RDS*(UD-UA)*RED/K2
      BV=0.375D0*RHOA/RHOD*CD/RDS*(VD-VA)*RED/K2
      AN(1,IP1(MM),MM)=AU-BU
      AN(2,IP1(MM),MM)=-AV-BV
      HF=0.D0
      RETURN
      END

C
C ********************************************************************
C
      SUBROUTINE AIRPLT(XMIN,XMAX,YMIN,YMAX,LAYER,PT)
C
C WRITTEN BY: M. OLESKIW  ON:800607  LAST MODIFIED: 810918
C
C PLOTS OUTLINE OF AEROFOIL WITHIN VIEW WINDOW
C
      DOUBLE PRECISION XU(101),YU(101),XL(101),YL(101),
     .XE(101),YE(101)
```

```
      C
              REAL XMIN,XMAX,YMIN,YMAX,SNGL,XP,YP,XPT(204),
             .YPT(204),XPE(203),YPE(203),XGR(204,10),YGR(204,10),
             .XGRE(203,10),YGRE(203,10),XPP,YPP
      C
              INTEGER NCOU,NCOL,NCOB,IE,IP,J,NCOB1,I,
             .IT(10),LAYER,ITT,IPB,ITE(10),ITTE,
              NEL,NEU,NELM2,PT,NEUU,NELL
      C
              COMMON /GROW/XGR,YGR,
             .XGRE,YGRE,ITE,IT/AERO1/XE,YE/AERO3/NCOU,NCOL
             ./SFCS/XU,YU,XL,YL/AERO4/NEU,NEL,NEUU,NELL
      C
      C IN   XMIN=
      C IN   XMAX=
      C IN   YMIN=
      C IN   YMAX=PLOT WINDOW BOUNDARIES.
      C IN   LAYER=LAYER NO
      C IN   PT=0:CALCULATE PLOTTING SHAPE AND PLOT IT.
      C IN      1:CALCULATE PLOTTING SHAPE ONLY.
      C
             NELM2=NEL-2
             NCOB=NCOU+NCOL-1
             NCOB1=NCOB-1
             IP=0
             IE=0
      C
      C FOR THE UPPER SFC.:
                DO 700 J=1,NEU
                XP=SNGL(XU(J))
                YP=SNGL(YU(J))
                IF(YP.GE.YMAX)GOTO 720
                IF(XP.GE.XMAX)GOTO 730
                IP=IP+1
                XPT(IP)=XP
                YPT(IP)=YP
      700       CONTINUE
             GOTO 740
      720    IF(IP.GT.0)GOTO 750
             XPT(IP+1)=XP
             YPT(IP+1)=YMAX
             GOTO 760
      C OUT ALONG THE TOP EDGE
      750    XPT(IP+1)=(XP-XPT(IP))/(YP-YPT(IP))*(YMAX-YPT(IP))+XPT(IP)
             YPT(IP+1)=YMAX
      C UPPER RIGHT CORNER
      760    IP=IP+2
             XPT(IP)=XMAX
             YPT(IP)=YMAX
             GOTO 740
      C OUT ALONG THE RIGHT EDGE
      730    XPT(IP+1)=XMAX
             YPT(IP+1)=(YP-YPT(IP))/(XP-XPT(IP))*(XMAX-XPT(IP))+YPT(IP)
             IP=IP+1
      C
      C FOR THE LOWER SFC.:
      740    IPB=IP
               DO 800 J=1,NELM2
               XP=SNGL(XL(NEL-J))
               YP=SNGL(YL(NEL-J))
               IF(XP.GE.XMAX.OR.YP.LE.YMIN)GOTO 820
               IF(J.EQ.1)GOTO 830
               IF(XPP.LE.XMAX.AND.YPP.GE.YMIN)GOTO 830
               IF(YPP.LE.YMIN)GOTO 840
      C IN ON THE RIGHT EDGE
               IP=IP+1
```

```
              XPT(IP)=XMAX
              YPT(IP)=(YP-YPP)/(XP-XPP)*(XMAX-XPP)+YPP
              GOTO 830
C IN ON THE BOTTOM EDGE
840           XPT(IP+1)=XMAX
              YPT(IP+1)=YMIN
              IP=IP+2
              XPT(IP)=(XP-XPP)/(YP-YPP)*(YMIN-YPP)+XPP
              YPT(IP)=YMIN
C ADD ANOTHER POINT WITHIN WINDOW.
830           IP=IP+1
              XPT(IP)=XP
              YPT(IP)=YP
820           XPP=XP
              YPP=YP
800           CONTINUE
           IF(IP.NE.IPB)GOTO 850
           IP=IP+1
           XPT(IP)=XMAX
           YPT(IP)=YMIN
C
C ADD PARAMETERS NECESSARY FOR PLOTTING
850        XPT(IP+1)=XPT(1)
           YPT(IP+1)=YPT(1)
           XPT(IP+2)=XMIN
           YPT(IP+2)=YMIN
             DO 200 I=1,NCOB1
             XP=SNGL(XE(I))
             YP=SNGL(YE(I))
             IF(XP.GT.XMAX)GOTO 200
             IF(YP.GT.YMAX.OR.YP.LT.YMIN)GOTO 200
             IE=IE+1
             XPE(IE)=XP
             YPE(IE)=YP
200          CONTINUE
           XPE(IE+1)=XMIN
           YPE(IE+1)=YMIN
           XPT(IP+3)=(XMAX-XMIN)/20.0
           XPE(IE+2)=(XMAX-XMIN)/20.0
           YPT(IP+3)=(YMAX-YMIN)/12.0
           YPE(IE+2)=(YMAX-YMIN)/12.0
           IT(LAYER)=IP+3
           ITT=IP+3
           ITE(LAYER)=IE+2
           ITTE=IE+2
           IF(PT.EQ.0)GOTO 460
C
C THESE ARE THE AEROFOIL OUTLINE LINE SEGMENTS
C   TO BE PLOTTED WITHIN THE WINDOW
             DO 400 I=1,ITT
             XGR(I,LAYER)=XPT(I)
             YGR(I,LAYER)=YPT(I)
400          CONTINUE
C THESE ARE THE AEROFOIL ELEMENT ENDPTS. WITHIN THE WINDOW.
             DO 450 I=1,ITTE
             XGRE(I,LAYER)=XPE(I)
             YGRE(I,LAYER)=YPE(I)
450          CONTINUE
           IF(PT.EQ.1)RETURN
C
           ENTRY ERRPLT
C PLOT THE AEROFOIL OUTLINE
460        CALL NEWPEN(3)
           CALL LINE(XPT,YPT,IP+1,1,0,0)
           CALL LINEP(0.1)
           CALL LINE(XPE,YPE,IE,1,-1,0)
```

```
      RETURN
      END
C
C ************************************************************************
C
      SUBROUTINE AIRVEL(X,Y,UAS,VAS,NP)
C
C WRITTEN BY  M  OLESKIW  ON 800222  LAST MODIFIED 810608
C
C CALCULATES THE AIR VELOCITY COMPONENTS AT A GIVEN LOCATION
C
      DOUBLE PRECISION X,Y,UAS,VAS,XP(13),YP(13),XC(101),YC(101),
     RDS,GAMMA(101),D(100),K(101),PI,PJK,DD(5),C,AA,MM,SIGMA,
     SI(100),CO(100),PSI(13),DXC,DYC,DELTA,A,B,R1S,R2S,TH,DSQRT,DSQRT,
     R3S,DATAN,T3,DABS,DSIGN,ALPHAR,T1,T2,DLOG,R,DCOS,DSIN
C
      INTEGER L,NP,J,NCOU,NCOL,N,TYPE,JJ
C
      COMMON ALPHAR,PI/AERO3/NCOU,NCOL/AERO2/XC,YC,GAMMA,D,SI,CO
     /AIR/XP,YP,PSI/TRANS3/DD,C,TYPE,JJ/NACA/TH
C
C IN  X=
C IN  Y=COORDS. AT WHICH AIR VELOCITY IS TO BE DETERMINED.
C OUT UAS=
C OUT VAS=COMPONENTS OF AIR VELOCITY.
C IN  NP=NUMBER OF POINTS AT WHICH TO CALCULATE PSI.
C
      N=NCOU+NCOL-2
      SIGMA=1.D0
C SET GRID FOR AIR VELOCITY CALCULATIONS
      RDS=DD(JJ)/2.D6/C
      XP(1)=X
      XP(2)=X
      XP(3)=X-RDS
      XP(4)=X+RDS
      XP(5)=X
      YP(1)=Y+RDS
      YP(2)=Y-RDS
      YP(3)=Y
      YP(4)=Y
      YP(5)=Y
      IF(NP.NE.13)GOTO 100
C GRID FOR JACOBIAN CALCULATIONS
      XP(6)=XP(4)
      XP(7)=XP(4)
      XP(8)=XP(3)
      XP(9)=XP(3)
      XP(10)=X
      XP(11)=X+2.D0*RDS
      XP(12)=X-2.D0*RDS
      XP(13)=X
      YP(6)=YP(1)
      YP(7)=YP(2)
      YP(8)=YP(1)
      YP(9)=YP(2)
      YP(10)=Y+2.D0*RDS
      YP(11)=Y
      YP(12)=Y
      YP(13)=Y-2.D0*RDS
100     DO 110 J=1,NP
        IF(TYPE.EQ.-1)GOTO 115
        IF(TYPE.EQ.-2.OR.TYPE.EQ.-3)GOTO 200
        IF(TYPE.EQ.-10)GOTO 400
        PSI(J)=0.0
          DO 120 L=1,N
```

```
C FIND DISTANCE BETWEEN CONTROL PT. L AND GRID PT. I,J.
               DXC=XP(J)-XC(L)
               DYC=YP(J)-YC(L)
C CALCULATE COMPONENTS OF EQN. 9 AND FIG. 2
               DELTA=D(L)/2.DO
               B=DXC*CO(L)+DYC*SI(L)
               A=DYC*CO(L)-DXC*SI(L)
               R1S=A*A+(B+DELTA)*(B+DELTA)
               R2S=A*A+(B-DELTA)*(B-DELTA)
               R3S=A*A+B*B-DELTA*DELTA
               IF(R3S.LT.1.D-30)GO TO 130
               T3=DATAN(2.DO*A*DELTA/R3S)
               GO TO 140
  130          IF(DABS(A).LT.1.D-30)GO TO 150
               T3=DATAN((B+DELTA)/A)-DATAN((B-DELTA)/A)
               GO TO 140
  150          T3=DSIGN(PI,A)
  140          T1=(B+DELTA)*DLOG(R1S)
               T2=(B-DELTA)*DLOG(R2S)
               K(L)=(T1-T2+2.DO*A*T3-4.DO*DELTA)/4.DO/PI
               PSI(J)=PSI(J)-GAMMA(L)*K(L)
  120          CONTINUE
             R=YP(J)*DCOS(ALPHAR)-XP(J)*DSIN(ALPHAR)
C ASSURE THAT PSI ON AEROFOIL = 0.
             PSI(J)=PSI(J)+R-GAMMA(N+1)
             GOTO 110
C
C STREAMFN. FOR A CYLINDER.
  115        PSI(J)=YP(J)-YP(J)/4.DO/((XP(J)-5.D-1)**2+YP(J)*YP(J))
             GOTO 110
C
C STREAMFN. FOR A JOUKOWSKI AEROFOIL
  200        PSI(J)=PJK(XP(J),YP(J))
             GOTO 110
C
C STREAMFN. FOR A FLYING CIGAR.
  400        AA=TH/4.D2
             MM=AA*AA
             IF(YP(J).LT.0.DO)GOTO 410
             PSI(J)=MM*(AA-XP(J))/DSQRT((AA-XP(J))**2+
             YP(J)*YP(J))+YP(J)*YP(J)/2.DO
             GOTO 110
  410        PSI(J)=2.DO*MM-MM*(AA-XP(J))/DSQRT((AA-XP(J))**2+
             YP(J)*YP(J))-YP(J)*YP(J)/2.DO
  110        CONTINUE
C
C FOR BODIES OF REVOLUTION IN 3-D, CHANGE THE VELOCITY
C    FROM STRMFN. FORMULA.
          IF(TYPE.LE.-10)SIGMA=DABS(Y)
C
C CALCULATE AIRSPEED FROM STREAMFN.
          UAS=(PSI(1)-PSI(2))/2.DO/RDS/SIGMA
          VAS=(PSI(3)-PSI(4))/2.DO/RDS/SIGMA
          RETURN
          END

C
C ***********************************************************************
C
          SUBROUTINE CE(VOL,CEL,CEX,PLTFAC,THICK,LAYER,DDISTN,BOTH,AT,
         .TRUPLA,FILTER,LLEFT,LRIGHT)
C
C WRITTEN BY: M. OLESKIW  ON:800622  LAST MODIFIED:811024
C
C CALCULATE AND PLOT COLLISION EFFICIENCY OF ARBITRARY AEROFOIL
C    BY DETERMINING A SET OF IMPACTING TRAJECTORIES.
```

```
C
      DOUBLE PRECISION D,L(31),YO(31),CEE(5,30),THICK,FILTER,
     .PN,P,DIST,SLP,SSLP,DABS,ALPHAR,CEMAX,ZZ,DCOS,C,DO(5),
     .LU(101),LL(101),XU(101),XL(101),YU(101),YL(101),Y,DBLE,
     .CEED(5,30,5),LD(31,5),YOD(31,5),LEST,DOD,BETA,VTW(2),ACW(2),
     .LSPLM,AA,AAM,BB,BBM,CC,CCM,LDRG,CEDRG,DE,LSPL,ELM,LMXCE(6),
     .ELPP,ELP,SPLPP,SPLP,FELPP,FELP,EL,DMAX1,DB,LW(2),YOW(2),
     .DYDL(31),YOM,DL,YPL,DYDLD(31,5),VTOT(31,5),ACOL(31,5),W(5),
     .CEF(5,30)
C
      REAL LPMIN,YOPMIN,LRG,SNGL,FACT(4),LP(203),FLOAT,E(2),LDL,
     .YOP(203),CEP(203),XPAR(4,24),YPAR(4,24),LS(33),YOS(33),LDR,
     .PLTFAC,XP(203),XPMIN,CEPMIN,X,XLF,XRG,COS,CEV(201,31),LMX,
     .CET,CALPH,CEDEL,CEMAXE,CETOT,EMDEL,CES(33),LBRG,FL(5),YOE(2),
     .ABS,AMAX1,CEDIF,CEB(203),LPB(203),LLEFT,LRIGHT,AMIN1,FLV
C
      INTEGER CEL,F,I,ICT,IRX,IRY,PX,PY,YOL,ICUD(5),ICLD(5),IJ,
     .KK,KL,KU,LAYER,NEU,NEL,CO,IIU,IIL,J,CEX,JJ,ICTD(5),II,DDISTN,
     .K,BOTH,GRAZE,KI,M,MI,MIM,MJ,NCH,ICU,ICL,AT,NCHA,TRJPLA,TYPE,
     .NDCPX,NDCPY,INSRT2,MP,H5,KE,NEUU,NELL,H5D(5),J1,J2,FF
C
      COMMON ALPHAR/COL/LD,YOD,ICTD,ICUD,ICLD/EFF/CEED/PLTPRM/XPAR,YPAR
     ./CEM/LMXCE/LG/LU,LL/SFCS/XU,YU,XL,YL/SRCH/D,IIU,IIL/WTS/W
     ./AERO4/NEU,NEL,NEUU,NELL/COLS/L,LW,YO,YOW,VTW,ACW
     ./CRITS/CEDEL,EMDEL/HERMT5/H5,H5D/TRANS3/DO,C,TYPE,J/CV/
     .VTOT,ACOL
C
C IN  YOL=PLOT THE YO VS L GRAPH (O, 1, OR 2)
C          (2 PLOTS AT HALF PAGE SIZE)
C IN  CEL=PLOT THE CE VS L GRAPH (O,1,2,3,OR 4)
C          (2 AND 4 PLOT AT HALF PAGE SIZE; 3 AND 4 ALSO PLOT MEAN
C          CE VS L CURVE WHEN THERE IS A DROPLET DISTRIBUTION,
C          OR IF SMOOTHING HAS BEEN PERFORMED
C IN  CEX=PLOT THE CE VS X GRAPH (O, 1, OR 2)
C          (2 PLOTS AT HALF PAGE SIZE)
C IN  PLTFAC=FACTOR FOR SCALING ALL PLOTS
C IN  THICK=AEROFOIL THICKNESS IN %
C IN  LAYER=LAYER OF ACCRETION
C IN  DDISTN=NO. OF SIZES IN DROPLET DISTN.
C IN  BOTH=TRAJECTORIES FOR BOTH SFCS. (O OR 1)
C IN  AT=AUTO-TRAJECTORY MODE (O OR 1)
C IN  TRJPLA=PLOT TRAJECTORIES (O OR 1)
C IN  FILTER=LENGTH OF BOXCAR FILTER(AS A FRACTION OF L RANGE OF
C               LARGEST DROPLET SIZE) TO BE APPLIED TO SMOOTH CE VS L
C               CURVE(S).  IF O, THEN DON'T FILTER. (F)
C IN  LLEFT=LEFTMOST POINT TO BE PLOTTED IN YO VS L AND CE VS L
C             CURVES.  IF O, DETERMINE AUTOMATICALLY. (F)
C IN  LRIGHT=RIGHTMOST POINT AS ABOVE. (F)
 10   FORMAT('1BETAO (MAX LOCAL CE) IS',F5.1,'%  AT A DISTANCE OF',
     .F7.3,' FROM THE NOSE ',/,'OTHE TOTAL COLLISION EFFICIENCY IS',
     .F5.1,'%')
 20   FORMAT('OFAILURE TO CONVERGE UPON MAX CE')
 30   FORMAT('OLOCAL BETA:',F5.1,
     .'%  EST. MAX BETA:',F5.1,'%  MAX BETA CHANGE:',F6.1,'%')
C
      FACT(1)=1.0
      FACT(2)=0.7
      FACT(3)=1.0
      FACT(4)=0.7
      CALPH=COS(SNGL(ALPHAR))
      LMXCE(6)=0.D0
C DO FOR EACH DROPLET SIZE
        DO 700 J=1,DDISTN
        ICT=2
        GRAZE=1
C FIND DROPLET GRAZING TRAJECTORIES
```

```
            IF(J.EQ.1.AND.LAYER.EQ.1)
           .CALL TRAJEC(TRUPLA,THICK,AT,BOTH,DDISTN,LAYER,GRAZE)
            IF(J.NE.1.OR.LAYER.NE.1)CALL TRAJEK(LAYER,GRAZE,1)
            GRAZE=0
            IF(BOTH.EQ.1)GOTO 130
C FOR SYMMETRICAL CASE, CREATE SYMMETRICAL VECTORS
            L(1)=-L(2)
            YO(1)=-YO(2)
            VTOT(1,J)=VTOT(2,J)
            ACOL(1,J)=-ACOL(2,J)
 130        LDRG=L(2)-L(1)
            LS(1)=SNGL(L(1))
            LS(2)=SNGL(L(2))
            LRG=LS(2)-LS(1)
            FL(J)=SNGL(FILTER)*LRG/2.0
C TOTAL COLLISION EFFICIENCY
            CET=SNGL((YO(2)-YO(1))/THICK)*CALPH*1.E4
C PLOTTING POINTS IN L
              DO 710 KI=1,201
              LP(KI)=LS(1)+FLOAT(KI-1)/200.0*LRG
 710        CONTINUE
C TARGET  DISTANCE IN L BETWEEN COLLISION PTS. OF
C  PAIRS OF TRAJECTORIES
C ******************************************************************
            DL=LDRG/5.D2
C ******************************************************************
            IF(DL.LT.2.D-4)DL=2.D-4
C ******************************************************************
C FIT HERMITE CUBIC POLYNOMIAL TO YO AND L
            CALL HERMIT(L(1),L(2),YO(1),YO(2),0.D0,0.D0,CEE(3,1),
           .CEE(2,1),CEE(1,1))
C FIND YO'S FOR FIRST TRAJECTORY PAIR
C ******************************************************************
            IF(BOTH.EQ.0)DDD=0.56D0*LDRG
            IF(BOTH.EQ.1)DDD=0.25D0*LDRG
C ******************************************************************
            DYDL(2)=(3.D0*CEE(3,1)*DDD+2.D0*CEE(2,1))*DDD+CEE(1,1)
            IF(BOTH.EQ.0)YOM=0.44D0*YO(1)+0.56D0*YO(2)
            IF(BOTH.EQ.1)YOM=0.75D0*YO(1)+0.25D0*YO(2)
C DISTANCE IN YO BETWEEN PAIR OF TRAJECTORIES
            YPL=DYDL(2)*DL
C ******************************************************************
            IF(YPL.LT.4.D-5)YPL=4.D-5
C ******************************************************************
            NCH=1
C SHIFT TO MAKE ROOM FOR 1ST TRAJECTORY
            IF(BOTH.EQ.0)GOTO 197
            L(3)=L(2)
            LS(3)=LS(2)
            YO(3)=YO(2)
            VTOT(3,J)=VTOT(2,J)
            ACOL(3,J)=ACOL(2,J)
            INSRT2=0
            M=0
            ICT=3
            MIM=1
            DYDL(1)=0.D0
            DYDL(3)=0.D0
            GOTO 190
C SHIFT TO MAKE ROOM FOR 1ST TWO TRAJECTORIES.
 197        L(4)=L(2)
            LS(4)=LS(2)
            YO(4)=YO(2)
            VTOT(4,J)=VTOT(2,J)
            ACOL(4,J)=ACOL(2,J)
            INSRT2=1
```

```
            M=0
            ICT=4
            MIM=2
            DYDL(1)=0.D0
            DYDL(4)=0.D0
C TRAJECTORY PAIR - YO VALUES:
 190        YOW(1)=YOM-YPL/2.D0
            YOW(2)=YOM+YPL/2.D0
            YO(MIM+1)=YOM
            CALL TRAJEK(LAYER,GRAZE,1)
            CALL TRAJEK(LAYER,GRAZE,2)
            L(MIM+1)=(LW(1)+LW(2))/2.D0
            VTOT(MIM+1,J)=(VTW(1)+VTW(2))/2.D0
            ACOL(MIM+1,J)=(ACW(1)+ACW(2))/2.D0
C CALCULATE COLLISION EFFICIENCY FOR TRAJECTORY PAIR.
            DYDL(MIM+1)=(YOW(2)-YOW(1))/(LW(2)-LW(1))
            IF(DYDL(MIM+1).LT.0.D0)RETURN
C FIT NEW HERMITE CUBIC POLYNOMIAL TO FIRST & SECOND
C   INTERVALS CREATED IN YO AND L VECTORS
            IF(BOTH.EQ.0.AND.M.EQ.0)GOTO 280
            CALL HERMIT(L(MIM),L(MIM+1),YO(MIM),YO(MIM+1),
            DYDL(MIM),DYDL(MIM+1),CEE(3,MIM),CEE(2,MIM),CEE(1,MIM))
 280        CALL HERMIT(L(MIM+1),L(MIM+2),YO(MIM+1),YO(MIM+2),
            DYDL(MIM+1),DYDL(MIM+2),CEE(3,MIM+1),
            CEE(2,MIM+1),CEE(1,MIM+1))
            LS(MIM+1)=SNGL(L(MIM+1))
            IF(INSRT2.EQ.0)GOTO 290
C CREATE SYMMETRICAL VECTORS FOR SYMMETRICAL SITUATION.
            L(NCH+1)=-L(MIM+1)
            LS(NCH+1)=-LS(MIM+1)
            YO(NCH+1)=-YO(MIM+1)
            VTOT(NCH+1,J)=VTOT(MIM+1,J)
            ACOL(NCH+1,J)=-ACOL(MIM+1,J)
            DYDL(NCH+1)=DYDL(MIM+1)
            CALL HERMIT(L(NCH),L(NCH+1),YO(NCH),YO(NCH+1),
            DYDL(NCH),DYDL(NCH+1),CEE(3,NCH),CEE(2,NCH),CEE(1,NCH))
            CALL HERMIT(L(NCH+1),L(NCH+2),YO(NCH+1),YO(NCH+2),
            DYDL(NCH+1),DYDL(NCH+2),CEE(3,NCH+1),CEE(2,NCH+1),CEE(1,NCH+1))
C
 290        MP=M
            M=ICT-1
            CEMAXE=0.0
            CEDIF=0.0
            CETOT=0.0
            I=1
            F=1
C
C FIND CE CURVE,TOTAL CE, AND MAX. VALUE OF CE
              DO 715 KI=1,201
 720          IF(LP(KI).LE.LS(F+1))GOTO 730
              F=F+1
              GOTO 720
 730          DB=DBLE(LP(KI)-LS(F))
              CEV(KI,M)=SNGL((3.D0*CEE(3,F)*DB
              +2.D0*CEE(2,F))*DB+CEE(1,F))*100.0*CALPH
              CETOT=CETOT+CEV(KI,M)
              CEMAXE=AMAX1(CEMAXE,CEV(KI,M))
              IF(CEMAXE.EQ.CEV(KI,M))PN=DBLE(LP(KI))
              IF(MP.NE.0)CEDIF=AMAX1(CEDIF,ABS(CEV(KI,M)-CEV(KI,MP)))
 715          CONTINUE
            CETOT=CETOT/SNGL(THICK)/2.0*LRG
C ******************************************************************
            BETA=DYDL(MIM+1)*1.D2*CALPH
C ******************************************************************
            WRITE(6,30)BETA,CEMAXE,CEDIF
            WRITE(7,30)BETA,CEMAXE,CEDIF
```

```
      CEDRG=2.DO*DBLE(CEMAXE/100.0/CALPH)
      IF(ICT.GT.5*(2-BOTH).AND.CEDIF/CEMAXE.LT.CEDEL/100.0
     .AND.ABS(CETOT-CET)/CET.LT.EMDEL/100.0)GOTO 180
C
      LSPLM=O.DO
C FIND FARTHEST APART PTS. ON CE VS L CURVE.
      DO 800 MI=1,M
        IF(BOTH.EQ.O.AND.L(MI+1).LT.O.DO)GOTO 800
C CREATE NORMALIZED CUBIC HERMITE POLYNOMIAL COEFFICIENTS
C   FOR SLOPE OF CURVE.
        AA=O.DO
        BB=3.DO*CEE(3,MI)*LDRG*LDRG/CEDRG
        CC=2.DO*CEE(2,MI)*LDRG/CEDRG
C FIND LENGTH OF CUBIC POLYNOMIAL SEGMENT.
        DE=(L(MI+1)-L(MI))/LDRG
        IF(DE.LT.1.D-2)GOTO 800
        IF((L(MI)-L(1))/(L(ICT)-L(1)).GT.O.9500
     .   AND.L(MI+1).GT.PN)GOTO 800
        CALL SFCLEN(DE,LSPL,AA,BB,CC)
C LOCATE LONGEST SEGMENT ON CE VS L CURVE
        LSPLM=DMAX1(LSPLM,LSPL)
        IF(LSPL.LT.LSPLM)GOTO 800
        ELM=DE
        MIM=MI
        AAM=AA
        BBM=BB
        CCM=CC
 800  CONTINUE
C FIND MIDPOINT TRAJECTORY.
      ELPP=ELM/3.DO
      ELP=ELM*2.DO/3.DO
      CALL SFCLEN(ELPP,SPLPP,AAM,BBM,CCM)
      FELPP=LSPLM/2.DO-SPLPP
 880  CALL SFCLEN(ELP,SPLP,AAM,BBM,CCM)
      FELP=LSPLM/2.DO-SPLP
      IF(DABS(FELP).LT.LSPLM/20.DO)GOTO 870
      EL=ELP-FELP*(ELP-ELPP)/(FELP-FELPP)
      ELPP=ELP
      ELP=EL
      FELPP=FELP
      GOTO 880
 870  DDD=ELP*LDRG
C ESTIMATED NEW VALUE OF L FOR INSERTION.
      LEST=DDD+L(MIM)
C SHIFT VECTORS TO MAKE ROOM FOR NEW TRAJECTORIES.
      NCH=ICT-MIM
        DO 810 MI=1,NCH
        MJ=ICT+1-MI
        L(MJ+1)=L(MJ)
        YO(MJ+1)=YO(MJ)
        DYDL(MJ+1)=DYDL(MJ)
        LS(MJ+1)=LS(MJ)
        VTOT(MJ+1,J)=VTOT(MJ,J)
        ACOL(MJ+1,J)=ACOL(MJ,J)
        IF(MI.EQ.1)GOTO 810
          DO 815 I=1,3
          CEE(I,MJ+1)=CEE(I,MJ)
 815      CONTINUE
 810    CONTINUE
      YOM=((CEE(3,MIM)*DDD+CEE(2,MIM))*DDD+CEE(1,MIM))*DDD+YO(MIM)
      IF(BOTH.EQ.O.AND.M.EQ.3)YOM=(YO(3)+YO(4))/2.DO
      ICT=ICT+1
      INSRT2=1
      IF(BOTH.EQ.1.OR.DABS(LEST).LT.LDRG/2.D2)INSRT2=O
      IF(INSRT2.EQ.O)GOTO 830
C INSERT ANOTHER TRAJECTORY PAIR FOR SYMMETRICAL CASES.
```

```
          NCHA=MIM+1
            DO 820 MI=1,NCHA
            MJ=ICT+1-MI
            L(MJ+1)=L(MJ)
            YO(MJ+1)=YO(MJ)
            DYDL(MJ+1)=DYDL(MJ)
            LS(MJ+1)=LS(MJ)
            VTOT(MJ+1,J)=VTOT(MJ,J)
            ACOL(MJ+1,J)=ACOL(MJ,J)
            IF(MI.EQ.1)GOTO 820
              DO 825 I=1,3
              CEE(I,MJ+1)=CEE(I,MJ)
825           CONTINUE
820         CONTINUE
          MIM=MIM+1
          ICT=ICT+1
830       DYDL(MIM+1)=(3.DO*CEE(3,MIM)*DDD
          +2.DO*CEE(2,MIM))*DDD+CEE(1,MIM)
          YPL=DYDL(MIM+1)*DL
          IF(YPL.LT.4.D-5)YPL=4.D-5
          GOTO 190
C
C FIND BETAO (MAX VALUE OF LOCAL CE)
C  USING THE NEWTON-RAPHSON ALGORITHM
180       H5D(J)=H5
          IF(H5.EQ.0)GOTO 181
          CEDIF=0.0
          CALL HERM5(L,YO,DYDL,ICT,CEF)
          FF=1
            DO 182 KI=1,201
183         IF(LP(KI).LE.LS(FF+1))GOTO 184
            FF=FF+1
            GOTO 183
184         DB=DBLE(LP(KI)-LS(FF))
            CEDIF=AMAX1(CEDIF,ABS(CEV(KI,M)-
            SNGL((((5.DO*CEF(5,FF)*DB+4.DO*CEF(4,FF))*DB
            +3.DO*CEF(3,FF))*DB+2.DO*CEF(2,FF))*DB
            +CEF(1,FF))*100.0*CALPH))
182         CONTINUE
          IF(CEDIF/CEMAXE.LT.0.08)GOTO 185
          H5D(J)=0
          GOTO 181
185         DO 186 J1=1,30
              DO 187 J2=1,5
              CEE(J2,J1)=CEF(J2,J1)
187           CONTINUE
186         CONTINUE
181       JJ=0
520       P=PN
C FIND CE VS L SLOPE AND ITS SLOPE
505       IF(P.GT.L(I))GOTO 500
          I=I-1
          IF(I.GE.1)GOTO 505
          P=L(1)
          I=1
          GOTO 510
500       IF(P.LE.L(I+1))GOTO 510
          I=I+1
          IF(I.LT.ICT)GOTO 500
          I=ICT-1
          P=L(ICT)
510       DIST=P-L(I)
          IF(H5D(J).EQ.1)GOTO 515
C SLOPE OF CE CURVE AND ITS SLOPE FOR HERMITE CUBIC POLYNOMIAL
          SSLP=6.DO*CEE(3,I)
          SLP=6.DO*CEE(3,I)*DIST+2.DO*CEE(2,I)
```

```
226          GOTO 517
      C SLOPE OF CE CURVE AND ITS SLOPE FOR HERMITE QUINTIC SPLINE.
227   515    SLP=((20 DO*CEE(5,I)*DIST+12.DO*CEE(4,I))*DIST+6 DO*CEE(3,I)
             )*DIST+2 DO*CEE(2,I)
228          SSLP=(60 DO*CEE(5,I)*DIST+24 DO*CEE(4,I))*DIST+6.DO*CEE(3,I)
229   517    PN=P-SLP/SSLP
230          IF(DABS(P-PN) LT.LDRG/5.D2)GOTO 512
231          JJ=JJ+1
232          IF(JJ LT 100)GOTO 520
233          WRITE(6,20)
234          WRITE(7,20)
235          GOTO 560
236   512    IF(HSD(J) EQ O)CEMAX=((3 DO*CEE(3,I)*DIST+2.DO*CEE(2,I))*
             DIST+CEE(1,I))*1 D2*DCOS(ALPHAR)
237          IF(HSD(J) EQ 1)CEMAX=((((5 DO*CEE(5,I)*DIST+4.DO*CEE(4,I))*DIST
             +3.DO*CEE(3,I))*DIST+2 DO*CEE(2,I))*DIST+CEE(1,I))*1.D2
             *DCOS(ALPHAR)
238          LMXCE(J)=P
239          WRITE(6,10)CEMAX,P,CET
240          WRITE(7,10)CEMAX,P,CET
241   560    ICU=O
242          ICL=O
      C
      C CREATE DISTRIBUTED SPLINE COEFF. MATRIX.
243          KE=3+2*HSD(J)
244            DO 570 I=1,ICT
245              IF(I EQ.ICT)GOTO 581
246              DO 580 K=1,KE
247              CEED(K,I,J)=CEE(K,I)
248   580        CONTINUE
249   581      IF(L(I) LT.O DO)ICL=ICL+1
250            IF(L(I) GE O DO)ICU=ICU+1
251            LD(I,J)=L(I)
252            YOD(I,J)=YO(I)
253            YOS(I)=SNGL(YO(I))
254            DYDLD(I,J)=DYDL(I)
255   570      CONTINUE
256          ICTD(J)=ICT
257          ICUD(J)=ICU
258          ICLD(J)=ICL
      C FIND PROBABLE LOCATION OF PEAK OF MEAN CE VS L CURVE.
259          LMXCE(6)=LMXCE(6)+LMXCE(J)*W(J)
260          IF(YOL EQ.O AND.CEL EQ O)GOTO 700
261          IF(J GT. 1)GOTO 170
      C
      C DETERMINE PLOTTING PARAMETERS
262          LRIGHT=AMAX1(LRIGHT,LS(ICT)+FL(1))
263          LLEFT=AMIN1(LLEFT,LS(1)-FL(1))
264          IF(LAYER EQ 1)CALL PLTSZ(LLEFT,LRIGHT,YOS(1),YOS(ICT),
             LPMIN,YOPMIN,PX,PY,IRX,IRY,NDCPX,NDCPY)
265          IF(LAYER GT 1)CALL PLTSZE(LLEFT,LRIGHT,YOS(1),YOS(ICT),
             LPMIN,YOPMIN,PX,PY,IRX,IRY,NDCPX,NDCPY)
266          LP(202)=LPMIN
267          LP(203)=XPAR(4,IRX)/10.O**PX
268          CALL NEWPEN(1)
269          IF(YOL EQ.O)GOTO 700
270          YOP(202)=YOPMIN
271          YOP(203)=YPAR(4,IRY)/10.O**PY
      C PLOT YO VS L AXES
272          CALL FACTOR(FACT(YOL)*PLTFAC)
273          CALL ORIGIN(999,20.0,13.0,5.0,5.0)
274          CALL AX2EP(XPAR(3,IRX),3,NDCPX,O,1.O)
275          CALL AXIS2(O.O,O.O,'L/C',-3,XPAR(2,IRX),O.O,LPMIN,XPAR(4,IRX
             )/10.O**PX,XPAR(3,IRX))
276          CALL AXIS2(XPAR(2,IRX),O.O,' ',-1,-YPAR(2,IRY),90.O,1.O,1.O,YPAR
             (3,IRY))
```

```
              CALL AX2EP(YPAR(3,IRY),3,NDCPY,0,1,1)
              CALL AXIS2(0.0,0.0,'YO/C',4,YPAR(2,IRY),90.0,YOPMIN,YPAR(4,IRY)/
              10.0**PY,-YPAR(3,IRY))
              CALL AXIS2(0.0,YPAR(2,IRY),' ',1,-XPAR(2,IRX),0.0,1.,1.,XPAR
             (3,IRX))
 170          IF(YOL.EQ.0)GOTO 700
C
C PLOT THE YO VS L POINTS
              LS(ICT+1)=LP(202)
              LS(ICT+2)=LP(203)
              YOS(ICT+1)=YOP(202)
              YOS(ICT+2)=YOP(203)
              CALL LINEP(0.15)
              CALL LINE(LS,YOS,ICT,1,-F,J-1)
              F=1
                DO 100 I=1,201
 120            IF(LP(I).LE.LS(F+1))GOTO 110
                F=F+1
                GOTO 120
 110            DB=DBLE(LP(I)-LS(F))
                IF(H5D(J).EQ.0)YOP(I)=SNGL(((CEED(3,F,J)*DB+CEED(2,F,J))*DB
                +CEED(1,F,J))*DB)+YOS(F)
                IF(H5D(J).EQ.1)YOP(I)=SNGL((((CEED(5,F,J)*DB+CEED(4,F,J))*DB
                +CEED(3,F,J))*DB+CEED(2,F,J))*DB+CEED(1,F,J))*DB)+YOS(F)
 100            CONTINUE
C PLOT THE YO VS L LINE
              CALL LINE(LP,YOP,201,1,0,1)
 700          CONTINUE
C
            IF(CEL.EQ.0)GOTO 300
            J=1
C PLOT THE CE VS L AXES
            CALL FACTOR(FACT(CEL)*PLTFAC)
            CALL ORIGIN(999,20.0,13.0,5.0,5.0)
            CALL AX2EP(XPAR(3,IRX),3,NDCPX,0,1,0)
            CALL AXIS2(0.0,0.0,'L/C',-3,XPAR(2,IRX),0.0,LPMIN,XPAR(4,IRX)/
            10.0**PX,XPAR(3,IRX))
            CALL AXIS2(XPAR(2,IRX),0.0,' ',-1,-YPAR(2,10),90.0,0.0,1.0,YPAR(3,
            10))
            CALL AX2EP(YPAR(3,10),3,0,0,1,1)
            CALL AXIS2(0.0,0.0,'COLLISION EFFICIENCY IN %',25,YPAR(2,10),
           90.0,0.0,YPAR(4,10)*10.0,-YPAR(3,10))
            CALL AXIS2(0.0,YPAR(2,10),' ',1,-20.0,0.0,1.,1.,XPAR(3,IRX))
            CEP(202)=0.0
            CEP(203)=YPAR(4,10)*10.0
            CEB(202)=0.0
            CEB(203)=CEP(203)
            LPB(202)=LP(202)
            LPB(203)=LP(203)
 250        ICT=ICTD(J)
              DO 240 I=1,ICT
              LS(I)=SNGL(LD(I,J))
              CES(I)=SNGL(DYDLD(I,J)*1.D2)*CALPH
 240          CONTINUE
            CES(ICT+1)=CEP(202)
            CES(ICT+2)=CEP(203)
            LS(ICT+1)=LP(202)
            LS(ICT+2)=LP(203)
            LRG=LS(ICT)-LS(1)
C FIND THE PLOTTING POINTS FOR THE MEAN AND/OR SMOOTHED CURVE.
            IF(J.NE.1.OR.(FILTER.EQ.0.D0.AND.DDISTN.EQ.1))GOTO 225
            LBRG=LRG+FL(J)*2.0
            LPB(1)=LS(1)-FL(J)
              DO 235 I=1,201
              LPB(I)=LPB(1)+FLOAT(I-1)/200.0*LBRG
 235          CONTINUE
```

```
225   F=1
C PLOT THE CE VS L POINTS.
      CALL LINEP(0.15)
      CALL LINE(LS,CES,ICT,1,-1,J-1)
        DO 210 I=1,201
        LP(I)=LS(1)+FLOAT(I-1)/200.0*LRG
230     IF(LP(I).LE.LS(F+1))GOTO 220
        F=F+1
        GOTO 230
220     DB=DBLE(LP(I)-LS(F))
        IF(H5D(J).EQ.0)CEP(I)=SNGL((3.D0*CEED(3,F,J)*DB
     +2.D0*CEED(2,F,J))*DB+CEED(1,F,J))*100.0*CALPH
        IF(H5D(J).EQ.1)CEP(I)=SNGL((((5.D0*CEED(5,F,J)*DB
     +4.D0*CEED(4,F,J))*DB+3.D0*CEED(3,F,J))*DB
     +2.D0*CEED(2,F,J))*DB+CEED(1,F,J))*100.0*CALPH
210     CONTINUE
C PLOT THE CE VS L LINE.
      IF((FILTER.NE.0.D0.OR.DDISTN.GT.1).AND.(CEL.EQ.3.OR.CEL.EQ.4))
      CALL NEWPEN(3)
      CALL LINE(LP,CEP,201,1,0,1)
      J=J+1
      IF(J.LE.DDISTN)GOTO 250
      IF(CEL.NE.3.AND.CEL.NE.4)GOTO 300
C
C PLOT THE MEAN AND/OR SMOOTHED CE VS L CURVE.
        DO 990 I=1,201
        CEB(I)=0.0
990     CONTINUE
      IF(FILTER.NE.0.D0)GOTO 1000
      IF(DDISTN.LE.1)GOTO 300
        DO 900 J=1,DDISTN
        F=1
        ICT=ICTD(J)
          DO 910 I=1,201
          IF(LPB(I).LT.SNGL(LD(1,J)))GOTO 910
          IF(LPB(I).GT.SNGL(LD(ICT,J)))GOTO 910
930       IF(LPB(I).LE.SNGL(LD(F+1,J)))GOTO 920
          F=F+1
          GOTO 930
920       DB=DBLE(LPB(I))-LD(F,J)
          IF(H5D(J).EQ.0)CEB(I)=CEB(I)+SNGL(W(J)*((3.D0*CEED(3,F,J)*DB
     +2.D0*CEED(2,F,J))*DB+CEED(1,F,J)))*100.0*CALPH
          IF(H5D(J).EQ.1)CEB(I)=CEB(I)+SNGL(W(J)*((((5.D0*CEED(5,F,J)*DB
     +4.D0*CEED(4,F,J))*DB+3.D0*CEED(3,F,J))*DB
     +2.D0*CEED(2,F,J))*DB+CEED(1,F,J)))*100.0*CALPH
910       CONTINUE
900     CONTINUE
      GOTO 1100
1000    DO 1010 J=1,DDISTN
        ICT=ICTD(J)
        F=1
        LDL=SNGL(LD(1,J))
        LDR=SNGL(LD(ICT,J))
        LMX=SNGL(LMXCE(J))
          DO 1020 IJ=1,201
          IF(LPB(IJ).GE.LMX)GOTO 1110
          IF(LPB(IJ).LE.LDL)GOTO 1120
C *****************************************************************
          FLV=FL(J)-0.9*FL(J)/(LMX-LDL)*(LPB(IJ)-LDL)
C *****************************************************************
          GOTO 1200
1110      IF(LPB(IJ).GE.LDR)GOTO 1120
C *****************************************************************
          FLV=0.1*FL(J)+0.9*FL(J)/(LDR-LMX)*(LPB(IJ)-LMX)
C *****************************************************************
          GOTO 1200
```

```
 1120       FLV=FL(J)
 1200       E(1)=LPB(IJ)-FLV
            E(2)=LPB(IJ)+FLV
              DO 1030 I=1,2
              IF(E(I).GT.SNGL(LD(1,J)))GOTO 1040
              YOE(I)=SNGL(YOD(1,J))
              GOTO 1030
 1040         IF(E(I).LT.SNGL(LD(ICT,J)))GOTO 1050
              YOE(I)=SNGL(YOD(ICT,J))
              GOTO 1030
 1050         IF(E(I).GT.SNGL(LD(F,J)))GOTO 1060
              F=F-1
              GOTO 1050
 1060         IF(E(I).LE.SNGL(LD(F+1,J)))GOTO 1070
              F=F+1
              GOTO 1060
 1070         DB=DBLE(E(I))-LD(F,J)
              IF(H5D(J).EQ.0)YOE(I)=SNGL(((CEED(3,F,J)*DB+CEED(2,F,J))*DB
              +CEED(1,F,J))*DB+YOD(F,J))
              IF(H5D(J).EQ.1)YOE(I)=SNGL(((((CEED(5,F,J)*DB+
              CEED(4,F,J))*DB+CEED(3,F,J))*DB+CEED(2,F,J))*DB
              +CEED(1,F,J))*DB+YOD(F,J))
 1030         CONTINUE
            CEB(IJ)=CEB(IJ)+(YOE(2)-YOE(1))/FLV*50.0*CALPH*SNGL(W(J))
 1020       CONTINUE
 1010     CONTINUE
 1100   CALL NEWPEN(1)
        CALL LINE(LPB,CEB,201,1,0,1)
C
 300    IF(CEX.EQ.O.OR.LAYER.GT.1)RETURN
        J=1
 600    ICT=ICTD(J)
C FIND RANGE OF X
          DO 610 I=1,ICT
          L(I)=LD(I,J)
 610      CONTINUE
          DO 310 KL=1,NEL
          IF(LL(KL).LE.L(1))GOTO 320
 310      CONTINUE
 320      DO 330 KU=1,NEU
          IF(LU(KU).GT.L(ICT))GOTO 340
 330      CONTINUE
 340    XRG=SNGL(XL(KL)+XU(KU))
        XLF=SNGL(-XU(KU))
        CO=0
        II=ICT-1
        IIL=1
        IIU=NEU
          DO 350 KK=1,201
          X=XLF+XRG/200.*FLOAT(KK-1)
          XP(KK)=X
          IF(X.GT.0.)GOTO 360
          CALL SFC(DBLE(-X),Y,1,1,ZZ)
          GOTO 370
 360      CALL SFC(DBLE(X),Y,0,1,ZZ)
 370      IF(CO.EQ.1)GOTO 380
          IF(ZZ.GT.L(ICT))GOTO 380
          IF(ZZ.GT.L(II))GOTO 410
          II=II-1
          IF(II.EQ.0)GOTO 390
          GOTO 370
 390      CO=1
 380      CEP(KK)=0.0
          GOTO 350
 410      DB=ZZ-L(II)
          IF(H5D(J).EQ.0)CEP(KK)=SNGL((3.DO*CEED(3,II,J)*DB
```

```
     +2.DO*CEED(2,II,J))*DB+CEED(1,II,J))*100.0*CALPH
      IF(H5D(J).EQ.1)CEP(KK)=SNGL((((5.DO*CEED(5,II,J)*DB
     +4.DO*CEED(4,II,J))*DB+3.DO*CEED(3,II,J))*DB
     +2.DO*CEED(2,II,J))*DB+CEED(1,II,J))*100.0*CALPH
350   CONTINUE
      IF(J.GT.1)GOTO 620
C DETERMINE PLOTTING PARAMETERS
      CALL PLTSZE(XP(1),XP(201),0.0,99.9,XPMIN,CEPMIN,PX,PY,IRX,IRY,
     NDCPX,NDCPY)
      XP(202)=XPMIN
      XP(203)=XPAR(4,IRX)/10.0**PX
      CEP(202)=0.0
      CEP(203)=YPAR(4,10)*10.0
C PLOT CE VS X AXES.
      CALL FACTOR(FACT(CEX)*PLTFAC)
      CALL ORIGIN(999,20.0,13.0,5.0,5.0)
      CALL AX2EP(XPAR(3,IRX),3,NDCPX,0.1.0)
      CALL AXIS2(0.0,0.0,'X/C',-3,XPAR(2,IRX),0.0,XPMIN,XPAR(4,IRX)
     ./10.0**PX,XPAR(3,IRX))
      CALL AXIS2(XPAR(2,IRX),0.0,' ',-1,-YPAR(2,10),90.0,0.0,1.0,YPAR(3,
     10))
      CALL AX2EP(YPAR(3,10),3,0,0,1.1)
      CALL AXIS2(0.0,0.0,'COLLISION EFFICIENCY IN %',25,YPAR(2,10),
     .90.0,0.0,YPAR(4,10)*10.0,-YPAR(3,10))
      CALL AXIS2(0.0,YPAR(2,10),' ',1,-20.0,0.0,1.,1.,XPAR(3,IRX))
C PLOT THE CE VS X LINE.
620   CALL LINE(XP,CEP,201,1,0,1)
      J=J+1
      IF(J.LE.DDISTN)GOTO 600
C
      RETURN
      END
C
C ******************************************************************
C
      SUBROUTINE COLVEL(DDISTN)
C
C WRITTEN BY  M. OLESKIW  ON:810225  LAST MODIFIED:810506
C
C INTERPOLATE DROPLET IMPACT VELOCITY ALONG AEROFOIL SFC.
C
      DOUBLE PRECISION BPAR(4),LD(31,5),YOD(31,5),
      A(31),V(31),L(31),COEFA(30,3),COEFV(30,3),CFA(3,30,5),
     .CFV(3,30,5),VTOT(31,5),ACOL(31,5)
C
      INTEGER I,J,DDISTN,ICT,ICTD(5),ICUD(5),ICLD(5),IER,K
C
      COMMON /CV/VTOT,ACOL/COL/LD,YOD,ICTD,ICUD,ICLD/CEV/CFV,CFA
C
C  IN DDISTN=NO. OF SIZES IN DROPLET DISTRIBUTION.
C
C CUBIC SPLINE END PARAMETERS (FREE SPLINE)
        DO 100 I=1,4
        BPAR(I)=0.DO
100     CONTINUE
        DO 200 J=1,DDISTN
        ICT=ICTD(J)
C CREATE SINGLE VECTORS FOR U AND V COMPONENTS,
C   AND IMPACT LOCATION LENGTHS.
          DO 210 I=1,ICT
          V(I)=VTOT(I,J)
          A(I)=ACOL(I,J)
          L(I)=LD(I,J)
210       CONTINUE
        IF(ICT.GE.4)GOTO 300
```

```
C
C FIT CUBIC SPLINES FOR IMPACT VELOCITY AND ANGLE.
            CALL ICSICU(L,V,ICT,BPAR,COEFV,30,IER)
            CALL ICSICU(L,A,ICT,BPAR,COEFA,30,IER)
            GOTO 310
C
C CALCULATE QUASI-HERMITE CUBIC POLYNOMIALS FOR IMPACT
C   VELOCITY AND ANGLE.
300         CALL IQHSCU(L,V,ICT,COEFV,30,IER)
            CALL IQHSCU(L,A,ICT,COEFA,30,IER)
C
C CREATE DISTRIBUTED COEFFICIENT MATRICES.
310           DO 220 I=1,ICT
                DO 230 K=1,3
                CFA(K,I,J)=COEFA(I,K)
                CFV(K,I,J)=COEFV(I,K)
230             CONTINUE
220           CONTINUE
200         CONTINUE
          RETURN
          END

C
C ****************************************************************
C
          SUBROUTINE COORDS(TYPE,T,THETA,XU,XL,YU,YL)
C
C WRITTEN BY: M. OLESKIW  ON:790928  LAST MODIFIED:810725
C
          DOUBLE PRECISION X,YU,YL,DSQRT,C,T,THETA,DCOS,EIM2,EIM1,
         .EI,DABS,A,B,DSIN,XI,ETA,ALPHAR,PI,TOL,LE,RE,THETAM,YC,
         .JTIM1,JTIM2,M,P,DFLOAT,K1,XU,XL,YT,DATAN,PHI,YURR,DTAN,
         .JTHICK
C
          INTEGER TYPE,ATYPE,IABS,I,IER,MEAN,MOD,MM
C
          COMMON ALPHAR,PI/JOUK1/A,B,EI
C
          EXTERNAL JTHICK
C
10        FORMAT('OFAILURE TO FIND CORRECT PARAMETERS FOR AEROFOIL')
C
C IN   TYPE=AEROFOIL TYPE
C IN   T=AEROFOIL THICKNESS IN PERCENT
C IN   THETA=ANGLE FROM NEGATIVE X AXIS
C OUT  X=X-COORD. OF AEROFOIL SFC.
C OUT  YU=
C OUT  YL= UPPER & LOWER Y-COORDS. OF AEROFOIL SFC.
C
          ATYPE=IABS(TYPE)
          IF(ATYPE.EQ.1)GOTO 101
          IF(ATYPE.EQ.2.OR.ATYPE.EQ.3)GOTO 102
          IF(TYPE.EQ.-10)GOTO 103
          GOTO 100
C
          ENTRY KOORDS(MEAN)
C
C IN   MEAN=DESIGNATION FOR NACA MEAN LINE.
C
C DETERMINE PARAMETERS FOR MEAN LINES OF NACA AEROFOILS.
          IF(MEAN.GE.100)GOTO 200
C
C FOUR DIGIT FAMILY OF NACA AEROFOILS.
          M=DFLOAT(MEAN/10)/1.D2
          P=DFLOAT(MOD(MEAN,10))/1.D1
          RETURN
```

```
C
C FIVE DIGIT FAMILY OF NACA AEROFOILS.
  200   MM=(MEAN-200)/10
        GOTO(210,220,230,240,250),MM
  210   M=0.058D0
        K1=361.4D0
        RETURN
  220   M=0.126D0
        K1=51.64D0
        RETURN
  230   M=0.2025D0
        K1=15.957D0
        RETURN
  240   M=0.29D0
        K1=6.643D0
        RETURN
  250   M=0.391D0
        K1=3.23D0
        RETURN
C
C CALCULATE THE THICKNESS DIST. OF A NACA AEROFOIL
C MODIFIED TO HAVE A RAZOR-LIKE TRAILING EDGE BY REMOVING
C A LINEARLY INCREASING AMOUNT FROM X=0.3 TO X=1.0
C REF: GREGORY, N. & P.G. WILBY (1973), A.R.C. PAPER #1261
C       ABBOTT, I.H. & A.E. VON DOENHOFF (1959), THEORY OF WING SECTIONS,
C       TL 672 A12 1959, P113 & 321
C
C CALCULATE AEROFOIL X & Y COORDS. FOR EACH SFC.
  100   X=(1.D0-DCOS(THETA))/2.D0
        B=0.2969D0*DSQRT(X)-0.126D0*X-0.3516D0*X*X
        C=0.2843D0*X**3-0.1015D0*X**4
        YT=T/0.2D2*(B+C)
        IF(X.GT.0.3D0)YT=YT-(X-0.3D0)*2.1D-3*T/0.7D0/0.2D2
        IF(X-1.D0.GT.-1.D-8)YT=0.D0
        IF(MEAN.NE.0)GOTO 520
C SYMMETRICAL NACA AEROFOIL
        XU=X
        XL=X
        YU=YT
        YL=-YT
        RETURN
C
  520   IF(MEAN.GE.100)GOTO 530
C
C FOUR DIGIT FAMILY OF MEAN LINES.
        IF(X.GT.P)GOTO 540
        YC=M/P/P*(2.D0*P*X-X*X)
        PHI=DATAN(2.D0*M/P/P*(P-X))
        GOTO 550
  540   YC=M/(1.D0-P)**2*(1.D0-2.D0*P+2.D0*P*X-X*X)
        PHI=DATAN(2.D0*M/(1.D0-P)**2*(P-X))
        GOTO 550
C
C FIVE DIGIT FAMILY OF MEAN LINES.
  530   IF(X.GT.M)GOTO 560
        YC=K1/6.D0*(X**3-3.D0*M*X*X+M*M*(3.D0-M)*X)
        PHI=DATAN(K1/6.D0*(3.D0*X*X-6.D0*M*X+M*M*(3.D0-M)))
        GOTO 550
  560   YC=K1/6.D0*M**3*(1.D0-X)
        PHI=DATAN(-K1/6.D0*M**3)
  550   XU=X-YT*DSIN(PHI)
        YU=YC+YT*DCOS(PHI)
        XL=X+YT*DSIN(PHI)
        YL=YC-YT*DCOS(PHI)
        RETURN
C
```

```
C CALCULATE THE X & Y COORDS. OF A CYLINDER
101     XU=(1.DO-DCOS(THETA))/2.DO
        XL=XU
        YU=DSQRT(O.25DO-(XU-O.5DO)*(XU-O.5DO)).
        IF(XU-1.DO.GT.-1.D-8)YU=O.DO
        YL=-YU
        RETURN
C
        ENTRY JOUKEX(T)
C DETERMINE VALUE OF E TO HAVE APPROPRIATELY THICK AEROFOIL
C    FOR EXACT JOUKOWSKI AEROFOIL GENERATION USING SECANT METHOD.
        I=1
C ******************************************************************
        TOL=PI/1.8D2
C ******************************************************************
C INITIAL GUESS AT E.
        EI=4.DO/3.DO/DSQRT(3.DO)*T/1.D2
        EIM2=EI
330     LE=PI/3.15DO
        RE=PI/2.6DO
C FIND MAX THICKNESS OF AEROFOIL FOR THIS VALUE OF E.
        B=(1.DO+2.DO*EI)/4.DO/(1.DO+2.DO*EI+EI*EI)
        A=B*(1.DO+EI)
340     CALL ZXGSN(JTHICK,LE,RE,TOL,THETAM,IER)
        IF(IER.LT.129.OR.IER.GT.132)GOTO 300
        WRITE(6,10)
        WRITE(7,10)
        GOTO 510
300     IF(I.GE.2)GOTO 320
        JTIM2=-JTHICK(THETAM)-T/1.D2
C SECOND GUESS AT E
        EI=T/O.66D2/DSQRT(3.DO)
        EIM1=EI
        I=2
C ******************************************************************
        TOL=PI/1.8D4
C ******************************************************************
        GOTO 330
320     JTIM1=-JTHICK(THETAM)-T/1.D2
        I=I+1
C SUCCESSIVELY BETTER APROXIMATIONS FOR E TO GIVE DESIRED THICKNESS.
        EI=EIM1-JTIM1*(EIM1-EIM2)/(JTIM1-JTIM2)
C ******************************************************************
        IF(DABS(JTIM1).LT.1.D-8)GOTO 500
C ******************************************************************
        EIM2=EIM1
        EIM1=EI
        JTIM2=JTIM1
        LE=THETAM-PI/1.D2
        RE=THETAM+PI/1.D2
        B=(1.DO+2.DO*EI)/4.DO/(1.DO+2.DO*EI+EI*EI)
        A=B*(1.DO+EI)
        GOTO 340
C
        ENTRY JOUKAP(T)
C DETERMINES VALUE OF E TO HAVE APPROPRIATELY THICK AEROFOIL FOR
C    APPROXIMATE JOUKOWSKI AEROFOIL GENERATION
C
        EI=4.DO/9.DO*DSQRT(3.DO)*T/1.D2
C
C DETERMINE A AND B
500     B=(1.DO+2.DO*EI)/4.DO/(1.DO+2.DO*EI+EI*EI)
        A=B*(1.DO+EI)
510     RETURN
C
C CALCULATE THE SHAPE OF A JOUKOWSKI AEROFOIL USING THE FULL (EXACT)
```

```
C    TRANSFORMATION AND SHIFTING FORMULAE.
C    REF: HOUGHTON, E.L. & A.E. BROCK (1970) AERODYNAMICS FOR ENGINEERING
C         STUDENTS (2ND EDITION) EDWARD ARNOLD LTD., LONDON, 458PP.
C
 102   X=-B*(1.DO+EI)*DCOS(THETA)-B*EI
       YU=B*(1.DO+EI)*DSIN(THETA)
       XI=X*(1.DO+B*B/(X*X+YU*YU))
       ETA=YU*(1.DO-B*B/(X*X+YU*YU))
       XU=(1.DO+2.DO*EI+2.DO*EI*EI)/2.DO/(1.DO+2.DO*EI+EI*EI)+XI
       IF(XU.LT.O.DO)XU=O.DO
       XL=XU
       YU=ETA
       IF(XU-1.DO.GT.-1.D-8)YU=O.DO
       YL=-YU
       RETURN
C
C CALCULATE THE SHAPE OF A 3-D CIGAR.
C    REF: MILNE-THOMSON (3RD. ED.)
C
 103   A=T/4.D2
       IF((PI-THETA)/PI.LT.1.D-5)GOTO 400
C CHECK FOR FRONT OR REAR SECTION OF CIGAR.
       YU=A*DSIN(THETA)/DCOS(THETA/2.DO)
       IF(THETA.LT.PI/2.DO)GOTO 410
C REAR SFC. IS 45 DEGREE SLOPING LINE.
       YURR=(A-1.DO)*DTAN(THETA)/(1.DO-DTAN(THETA))
       IF(YU.LT.YURR)GOTO 410
       YU=YURR
       YL=-YU
       XU=1.DO-A-YU
       XL=XU
       RETURN
C
C CIGAR SHAPED FRONT SECTION
 410   YL=-YU
       XU=A*(1.DO-DCOS(THETA)/DCOS(THETA/2.DO))
       XL=XU
       RETURN
C
 400   XU=1.DO
       XL=1.DO
       YU=O.DO
       YL=O.DO
       RETURN
       END

C
C ****************************************************************
C
       SUBROUTINE DRAG(UDS,VDS,UAS,VAS,CDS,RED,CD)
C
C WRITTEN BY: M. OLESKIW  ON:800222  LAST MODIFIED:810606
C
C CALCULATES THE REYNOLDS NUMBER AND DRAG COEFFICIENT OF THE DROPLET
C
       DOUBLE PRECISION DSORT,UDS,VDS,UAS,VAS,RED,CD,
      .K2,K3,K4
C
       INTEGER CDS
C
       COMMON /STAB1/K2,K3,K4
C
C IN  UDS=
C IN  VDS=DROPLET VELOCITY COMPONENTS.
C IN  UAS=
C IN  VAS=AIR VELOCITY COMPONENTS.
```

```
      C IN  CDS=PARAMETER TO DETERMINE DRAG COEFFICIENT FORMULATION.
      C OUT RED=RELATIVE MOTION REYNOLDS NO
      C OUT CD=DRAG COEFFICIENT.
      C
 5          RED=DSQRT((UDS-UAS)**2+(VDS-VAS)**2)*K2
 6          IF(CDS.EQ.2)GOTO 300
 7          IF(CDS.EQ.1.AND.RED.LE.5.DO)GOTO 100
      C
      C STEADY STATE DRAG COEFFICIENT OF DROPLET FOR RED < 5000
      C   ABRAHAM (1970)
 8          CD=0.2924DO*(1.DO+9.06DO/DSQRT(RED))**2
 9          RETURN
10    100   IF(RED.GE.1.D-2)GOTO 200
      C
      C STEADY STATE STOKE'S DRAG FOR RED < 0.01
11          CD=24.DO/RED
12          RETURN
      C
      C STEADY STATE DRAG COEFFICIENT FOR 0.01 < RED < 5 - SARTOR
      C   AND ABBOTT (1975)
13    200   CD=24.DO/RED+2.2DO
14          RETURN
      C
      C STEADY STATE DRAG COEFFICIENT - LANGMUIR & BLODGETT (1945)
15    300   CD=24.DO/RED+4.73DO/RED**0.37DO+6.24D-3*RED**0.38DO
16          RETURN
17          END

      C
      C ***************************************************************
      C
 1          SUBROUTINE FIT(BOTH)
      C
      C WRITTEN BY: M. OLESKIW  ON:810201  LAST MODIFIED:810923
      C
      C ROTATE UPPER AND LOWER SFCS. IF REQUIRED TO FIT CUBIC SPLINES
      C . AND DETERMINE LENGTHS ALONG SFC. TO EACH ENDPT.  CALCULATE
      C   THICKNESS OF ACCRETION.
      C
 2    30    FORMAT('-THE ACCRETED AREA FOR LAYER',I3,'  IS',F9.6,/,
           .'0THE ACCUMULATED ACCRETED AREA IS',F9.6)
      C
 3          DOUBLE PRECISION S30,C30,DSQRT,XUR(101),YUR(101),XU(101),YU(101),
           .XLR(101),YLR(101),XL(101),YL(101),BPARU(4),BPARL(4),CU(100,3),
           .CL(100,3),LU(101),LL(101),XS,LEN,INTU,INTL,XNNUR,YNNUR,XNP,YNP,
           .INTUP,INTLP,YNNLR,XNNLR,XURTLP,XLRTLR,ACCR,ACCRU,ACCRL,ACCRT,
           .XN,YN,XUXR,XLXR
      C
 4          INTEGER ATYPE,NEU,NEL,IERU,IERL,LAYER,BOTH,I,LYRM1,NEUU,NELL
      C
 5          COMMON /FOIL/XUR,YUR,XLR,YLR/LG/LU,LL/ROTP/C30,S30
           ./SPLINE/CU,CL/AERO4/NEU,NEL,NEUU,NELL/NNOSE/XNP,YNP/LLR/
           .ACCRT,LAYER,ATYPE/SFCS/XU,YU,XL,YL/NOSE/XN,YN/XXR/XUXR,XLXR
      C
      C IN  BOTH=TRAJECTORIES TO COLLIDE ON BOTH SFCS. (0 OR 1)
      C
      C ROTATE UPPER & LOWER SFCS. BY 30 DEG. ABOUT NOSE IN ORDER
      C   TO FIT CUBIC SPLINES
      C   - SEE KENNEDY & MARSDEN (1976)
 6          S30=5.D-1
 7          C30=DSQRT(3.DO)/2.DO
 8            DO 320 I=1,NEU
 9            XUR(I)=(XU(I)-XU(1))*C30+(YU(I)-YU(1))*S30
10            YUR(I)=(YU(I)-YU(1))*C30-(XU(I)-XU(1))*S30
11    320     CONTINUE
12            DO 330 I=1,NEL
```

```
              XLR(I)=(XL(I)-XL(1))*C30-(YL(I)-YL(1))*S30
              YLR(I)=(YL(I)-YL(1))*C30+(XL(I)-XL(1))*S30
  330         CONTINUE
C
C SET PARAMETERS FOR SPLINE FITTING
           BPARU(1)=1.DO
           BPARU(2)=6.DO/(XUR(2)-XUR(1))*((YUR(2)-YUR(1))/(XUR(2)-XUR
          .(1))-DSQRT(3.DO))
           BPARU(3)=0.DO
           BPARU(4)=0.DO
           BPARL(1)=1.DO
           BPARL(2)=6.DO/(XLR(2)-XLR(1))*((YLR(2)-YLR(1))/(XLR(2)-
          .XLR(1))+DSQRT(3.DO))
           BPARL(3)=0.DO
           BPARL(4)=0.DO
           IF(ATYPE.NE.1)GOTO 230
           BPARU(3)=1.DO
           BPARU(4)=6.DO/(XUR(NEUU)-XUR(NEUU-1))*
          .(-DSQRT(3.DO)/3.DO-(YUR(NEUU)-YUR(NEUU-1))/
          .(XUR(NEUU)-XUR(NEUU-1)))
           BPARL(3)=1.DO
           BPARL(4)=6.DO/(XLR(NELL)-XLR(NELL-1))*
          .(DSQRT(3.DO)/3.DO-(YLR(NELL)-YLR(NELL-1))/
          .(XLR(NELL)-XLR(NELL-1)))
C FIT CUBIC SPLINES TO EACH SFC.
  230      CALL ICSICU(XUR,YUR,NEUU,BPARU,CU,100,IERU)
           CALL ICSICU(XLR,YLR,NELL,BPARL,CL,100,IERL)
C
C CALCULATE INTEGRAL OF UPPER AND LOWER SFC. PROFILES.
C FIND THE LENGTHS FROM THE NOSE TO VARIOUS ENDPTS.
           LU(1)=0.DO
           LL(1)=0.DO
           INTU=0.DO
           INTL=0.DO
             DO 340 I=2,NEU
             IF(I.LE.NEUU)GOTO 360
             LU(I)=0.DO
             GOTO 340
  360        XS=XUR(I)-XUR(I-1)
             CALL SFCLEN(XS,LEN,CU(I-1,3),CU(I-1,2),CU(I-1,1))
             LU(I)=LU(I-1)+LEN
             INTU=INTU+(((CU(I-1,3)*XS/4.DO+CU(I-1,2)/3.DO)*XS
          .  +CU(I-1,1)/2.DO)*XS+YUR(I-1))*XS
  340        CONTINUE
             DO 350 I=2,NEL
             IF(I.LE.NELL)GOTO 370
             LL(I)=0.DO
             GOTO 350
  370        XS=XLR(I)-XLR(I-1)
             CALL SFCLEN(XS,LEN,CL(I-1,3),CL(I-1,2),CL(I-1,1))
             LL(I)=LL(I-1)-LEN
             INTL=INTL+(((CL(I-1,3)*XS/4.DO+CL(I-1,2)/3.DO)*XS
          .  +CL(I-1,1)/2.DO)*XS+YLR(I-1))*XS
  350        CONTINUE
          XUXR=XU(NEUU)
          XLXR=XL(NELL)
          IF(LAYER.EQ.1)GOTO 400
          XNNUR=(XN-XNP)*C30+(YN-YNP)*S30
          YNNUR=(YN-YNP)*C30-(XN-XNP)*S30
          ACCRU=INTU-INTUP+YNNUR*XURTLP-XNNUR*YNNUR/2.DO
          IF(BOTH.EQ.1)GOTO 410
          ACCR=2.DO*ACCRU
          GOTO 420
  410     XNNLR=(XN-XNP)*C30-(YN-YNP)*S30
          YNNLR=(YN-YNP)*C30+(XN-XNP)*S30
          ACCRL=INTLP-INTL-YNNLR*XLRTLP+XNNLR*YNNLR/2.DO
```

```
      ACCR=ACCRU+ACCRL
420   ACCRT=ACCRT+ACCR
      LYRM1=LAYER-1
      WRITE(6,30)LYRM1,ACCR,ACCRT
      WRITE(7,30)LYRM1,ACCR,ACCRT
400   INTUP=INTU
      INTLP=INTL
      XURTLP=XUR(NEUU)
      XLRTLP=XLR(NELL)
      RETURN
      END
C
C ******************************************************************
C
      SUBROUTINE GLERK5(EQN,CDS,EPS,LAMBH,WARN,SHORT,GLOBAL,GER)
C
C WRITTEN BY: M. OLESKIW  ON  810626  LAST MODIFIED: 810722
C
C INTEGRATE THE DROPLET EQNS. OF MOTION (IN X AND Y) USING:
C   1:A 4TH ORDER RUNGE-KUTTA-FEHLBERG TECHNIQUE.
C   2:ORDER EXTRAPOLATION OF THE ABOVE
C   3:STEP EXTRAPOLATION OF THE ABOVE (5TH ORDER ACCURACY).
C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978),
C       NUMERICAL ANALYSIS, P.254,  QA 297 B84
C       PROTHERO,A., 1980:  ESTIMATING THE ACCURACY OF NUMERICAL SOLNS.
C       TO ODE'S.  IN GLADWELL,I. AND D.K. SAYERS, EDS.  COMPUTATIONAL
C       TECHNIQUES FOR ODE'S.  ACADEMIC PRESS, 303 PP.. QA 370 C74 1978
C       AND  SHAMPINE, L.F. AND H.A. WATTS, 1976:  GLOBAL ERROR
C       ESTIMATION FOR ODE'S.  ACM TRANS. MATH. SOFTWARE, 2, #2, 172-186.
C
      DOUBLE PRECISION EPS,XDS(6,2),UDS(6,2),AN(2,6,2),YDS(6,2),
     .VDS(6,2),HT(2,6,2),DTS(6,2),UAS(6,2),VAS(6,2),RED(6,2),CD,RE,
     .K1,K2,K3,K4,K5,K6,L1,L2,L3,L4,L5,L6,M1,M2,M3,M4,M5,M6,
     .N1,N2,N3,N4,N5,N6,UA,VA,RMAX,DMAX1,DMIN,DMIN1,
     .DABS,XR,YR,UR,VR,XT,YT,UT,VT,RMINP(2),RMAXP(2),RMIN,
     .XD,YD,UD,VD,CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,
     .C14,C15,C16,C17,C18,C19,C20,C21,C22,C23,C24,TS(500,2)
      DOUBLE PRECISION DMINP(2),DMINPP(2),LAMBH,EIGMX,XDSO,YDSO,
     .UDSO,VDSO,TSO,DTSO,DTSK
C
      INTEGER EQN,CDS,I(2),IM4(2),IM3(2),IM2(2),IM1(2),IO(2),
     .IP1(2),MM,SHORT,WARN,GLOBAL,GER
C
      COMMON /PV/XDS,YDS,UDS,VDS/INTEG/AN,HT
     ./LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM
     ./REL/UAS,VAS,RED,CD
     ./RKFM/CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,
     .C15,C16,C17,C18,C19,C20,C21,C22,C23,C24
C
C IN   EQN=DENOTES PORTION OF TOTAL SYSTEM OF EQUATIONS TO BE SOLVED.
C IN   CDS=TYPE OF DRAG COEFFICIENT TO BE USED.
C IN   EPS=LOCAL ERROR PARAMETER.
C OUT  LAMBH=STABILITY PARAMETER.
C OUT  WARN=WARNING OF INSTABILITY (0 OR 1).
C OUT  SHORT=INDICATOR FOR NECESSITY OF SHORTING THE AUTO-
C              STEP-SIZE ALGORITHM.
C IN   GLOBAL=0 RKF4 INTEGRATING METHOD.
C              1 AS ABOVE BUT WITH ORDER EXTRAPOLATION TO FIND GLOBAL
C                ERROR.
C              2 AS ABOVE BUT USING STEP EXTRAPOLATION.
C IN   GER=INDICATOR THAT COLLISION HAS OCCURRED, AND THUS THAT
C          GLOBAL EXTRAPOLATION CANNOT BE CONTINUED.
C
      SHORT=0
      XDSO=XDS(IO(MM),MM)
```

```
      YDSO=YDS(IO(MM),MM)
      UDSO=UDS(IO(MM),MM)
      VDSO=VDS(IO(MM),MM)
      TSO=TS(I(MM),MM)
      DTSO=DTS(IO(MM),MM)
      IF(I(MM).GT.1)GOTO 100
      DMINPP(MM)=1.01D0
      DMINP(MM)=1.01D0
  100 TS(I(MM)+1,MM)=TSO+DTSO
      K1=DTSO*UDSO
      L1=DTSO*VDSO
      M1=DTSO*AN(1,IO(MM),MM)
      N1=DTSO*AN(2,IO(MM),MM)
      XD=XDSO+CC1*K1
      YD=YDSO+CC1*L1
      UD=UDSO+CC1*M1
      VD=VDSO+CC1*N1
      CALL AIRVEL(XD,YD,UA,VA,4)
      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
C
      K2=DTSO*UD
      L2=DTSO*VD
      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TSO+DTSO/4.D0,0)
      M2=DTSO*AN(1,IP1(MM),MM)
      N2=DTSO*AN(2,IP1(MM),MM)
      XD=XDSO+CC2*K1+C3*K2
      YD=YDSO+CC2*L1+C3*L2
      UD=UDSO+CC2*M1+C3*M2
      VD=VDSO+CC2*N1+C3*N2
      CALL AIRVEL(XD,YD,UA,VA,4)
      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
C
      K3=DTSO*UD
      L3=DTSO*VD
      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TSO+DTSO*3.75D-1,0)
      M3=DTSO*AN(1,IP1(MM),MM)
      N3=DTSO*AN(2,IP1(MM),MM)
      XD=XDSO+C4*K1-C5*K2+C6*K3
      YD=YDSO+C4*L1-C5*L2+C6*L3
      UD=UDSO+C4*M1-C5*M2+C6*M3
      VD=VDSO+C4*N1-C5*N2+C6*N3
      CALL AIRVEL(XD,YD,UA,VA,4)
      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
C
      K4=DTSO*UD
      L4=DTSO*VD
      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TSO+12.D0/13.D0
     .*DTSO,0)
      M4=DTSO*AN(1,IP1(MM),MM)
      N4=DTSO*AN(2,IP1(MM),MM)
      XD=XDSO+C7*K1-C8*K2+C9*K3-C10*K4
      YD=YDSO+C7*L1-C8*L2+C9*L3-C10*L4
      UD=UDSO+C7*M1-C8*M2+C9*M3-C10*M4
      VD=VDSO+C7*N1-C8*N2+C9*N3-C10*N4
      CALL AIRVEL(XD,YD,UA,VA,4)
      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
C
      K5=DTSO*UD
      L5=DTSO*VD
      CALL ACCN(UD,VD,UA,VA,RE,CD,EQN,TS(I(MM)+1,MM),0)
      M5=DTSO*AN(1,IP1(MM),MM)
      N5=DTSO*AN(2,IP1(MM),MM)
      XD=XDSO-C11*K1+C12*K2-C13*K3+C14*K4-C15*K5
      YD=YDSO-C11*L1+C12*L2-C13*L3+C14*L4-C15*L5
      UD=UDSO-C11*M1+C12*M2-C13*M3+C14*M4-C15*M5
      VD=VDSO-C11*N1+C12*N2-C13*N3+C14*N4-C15*N5
```

```
      CALL AIRVEL(XD,YD,UA,VA,4)
      CALL DRAG(UD,VD,UA,VA,CDS,RE,CD)
C
      K6=DTSO*UD
      L6=DTSO*VD
      CALL ACCN(UD,VD,UA,VA,RE,CD,EON,TSO+DTSO/2.D0,0)
      M6=DTSO*AN(1,IP1(MM),MM)
      N6=DTSO*AN(2,IP1(MM),MM)
C
      IF(MM.EQ.2.AND.GER.EQ.0)GOTO 110
C 4TH ORDER ESTIMATE AT TS(I(MM)+1,MM)
      XT=XDSO+C16*K1+C17*K3+C18*K4-C19*K5
      YT=YDSO+C16*L1+C17*L3+C18*L4-C19*L5
      UT=UDSO+C16*M1+C17*M3+C18*M4-C19*M5
      VT=VDSO+C16*N1+C17*N3+C18*N4-C19*N5
C
C NEW POSITION AND VELOCITY AT TS(I(MM)+1,MM)
110   XDS(IP1(MM),MM)=XDSO+C20*K1+C21*K3+C22*K4-C23*K5+C24*K6
      YDS(IP1(MM),MM)=YDSO+C20*L1+C21*L3+C22*L4-C23*L5+C24*L6
      UDS(IP1(MM),MM)=UDSO+C20*M1+C21*M3+C22*M4-C23*M5+C24*M6
      VDS(IP1(MM),MM)=VDSO+C20*N1+C21*N3+C22*N4-C23*N5+C24*N6
      IF(MM.EQ.2.AND.GER.EQ.0)GOTO 130
C
C DETERMINE DIFFERENCES IN 4TH AND 5TH ORDER ESTIMATES.
      XR=DABS((XT-XDS(IP1(MM),MM))/DTSO)
      YR=DABS((YT-YDS(IP1(MM),MM))/DTSO)
      UR=DABS((UT-UDS(IP1(MM),MM))/DTSO)
      VR=DABS((VT-VDS(IP1(MM),MM))/DTSO)
      RMAX=DMAX1(XR,YR,UR,VR)
      RMIN=DMIN1(XR,YR,UR,VR)
      IF(GLOBAL.LT.2)DMIN=(EPS/RMAX)**0.25D0
      IF(GLOBAL.EQ.2)DMIN=(EPS/RMAX)**0.20D0
C ADJUST NEXT STEP SIZE. TRY TO MINIMIZE OSCILLATIONS.
C ***************************************************************
      IF(DMINP(MM).LT.1.D0)GOTO 200
      IF(DMINPP(MM).LT.1.D0)GOTO 230
      IF(DMIN.LT.1.D0)DTSK=0.84D0*DMIN*DTSO
      IF(DMIN.GE.9.D0)DTSK=2.52D0*DTSO
      IF(DMIN.GE.1.D0.AND.DMIN.LT.9.D0)DTSK=((DMIN-1.D0)/4.D0+1.D0)
     .*DTSO*0.84D0
      GOTO 210
230   IF(DMIN.LT.1.D0)DTSK=0.84D0*DMIN*DTSO
      IF(DMIN.GE.11.D0)DTSK=1.68D0*DTSO
      IF(DMIN.GE.1.D0.AND.DMIN.LT.11.D0)DTSK=((DMIN-1.D0)/10.D0+1.D0)
     *DTSO*0.84D0
      GOTO 210
200   IF(DMINPP(MM).LT.1.D0)GOTO 220
      IF(DMIN.LE.0.8D0)DTSK=0.672D0*DTSO
      IF(DMIN.GT.1.D0)DTSK=((DMIN-1.D0)/10.D0+1.D0)*0.84D0*DTSO
      IF(DMIN.GT.0.8D0.AND.DMIN.LE.1.D0)DTSK=DMIN*0.84D0*DTSO
      GOTO 210
220   IF(DMIN.LE.0.2D0)DTSK=0.168D0*DTSO
      IF(DMIN.GT.1.D0)DTSK=((DMIN-1.D0)/10.D0+1.D0)*0.84D0*DTSO
      IF(DMIN.GT.0.2D0.AND.DMIN.LE.1.D0)DTSK=DMIN*0.84D0*DTSO
C ***************************************************************
C
C  CHECK FOR SUFFICIENT ACCURACY.
210   DMINPP(MM)=DMINP(MM)
      DMINP(MM)=DMIN
      IF(DTSK.LT.5.D-4)DTSK=5.D-4
      IF(RMAX.GT.EPS.AND.DTSK.GT.5.D-4)GOTO 170
C DO NOT ALLOW TIME STEP TO INCREASE INTO INSTABILITY.
      IF(I(MM).GT.1.AND.DTSK/DTSO*LAMBH.LT.-2.2D0)
     .DTSK=-2.2D0/LAMBH*DTSO
      DTS(IP1(MM),MM)=DTSK
      GOTO 300
```

```
170    IF(I(MM).NE.1)GOTO 140
       GOTO 160
C ****************************************************************
140    IF(RMAX/RMAXP(MM).GT.4.DO*RMIN/RMINP(MM))GOTO 150
C ****************************************************************
160    DTSO=DTSK
       DTS(IO(MM),MM)=DTSK
       GOTO 100
C STEP OVER EXTREMELY RAPIDLY CHANGING AREAS
150    DTS(IP1(MM),MM)=(DTSO+DTSK)/2.DO
       SHORT=1
300    RMINP(MM)=RMIN
       RMAXP(MM)=RMAX
       IF(I(MM).GT.1)DTS(IP1(MM),MM)=DMIN1(DTS(IP1(MM),MM),
      .2.DO*DTS(IP1(MM),MM)-DTSO)
C
       IF(GLOBAL.EQ.2)GOTO 130
C NEW POSITION AND VELOCITY ARE 4TH ORDER ESTIMATES
       XDS(IP1(MM),MM)=XT
       YDS(IP1(MM),MM)=YT
       UDS(IP1(MM),MM)=UT
       VDS(IP1(MM),MM)=VT
C
C NEW ACCELERATIONS AT I+1
130    CALL AIRVEL(XDS(IP1(MM),MM),YDS(IP1(MM),MM),
      .UAS(IP1(MM),MM),VAS(IP1(MM),MM),5+8*(2-MM))
       CALL DRAG(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
      .VAS(IP1(MM),MM),CDS,RED(IP1(MM),MM),CD)
       CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
      .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EON,TS(I(MM)+1,MM),0)
       IF(MM.GT.1)RETURN
C SKIP STABILITY CALCULATION IF TIME STEP IS DECREASING
C AND IS FAR FROM STABILITY LIMIT
       IF(I(1).GT.1.AND.DTSK/DTSO.LT.1.AND.LAMBH.GT.-1.8DO)
      .GOTO 120
       CALL STAB(RED(IP1(MM),MM),CD,UDS(IP1(MM),MM),VDS(IP1(MM),MM),
      .UAS(IP1(MM),MM),VAS(IP1(MM),MM),CDS,EIGMX)
       LAMBH=EIGMX*DTSO
       IF(LAMBH.LT.-2.7DO)WARN=1
120    IF(EON.NE.2)RETURN
       CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
      .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EON,TS(I(MM)+1,MM),1)
       RETURN
       END

C
C ****************************************************************
C
       SUBROUTINE GROWTH(XMIN,XMAX,YMIN,YMAX,LYRMAX,PLTFAC)
C
C WRITTEN BY: M. OLESKIW  ON 800713  LAST MODIFIED 810422
C
C PLOTS SUCCESSIVE AEROFOIL OUTLINES WITHIN VIEW WINDOW
C
       REAL XGR(204,10),YGR(204,10),PLTFAC,XMIN,XMAX,YMIN,YMAX,
      .XPLT(204),YPLT(204),XGRE(203,10),YGRE(203,10),XPLTE(101),
      .YPLTE(101),DX,DY,DDX,DDY,ABS,AINT
C
       INTEGER IT(10),LYRMAX,ITT,I,J,LYRM1,
      .ITE(10),ITTE,NDCPX,NDCPY
C
       COMMON/GROW/XGR,YGR,XGRE,YGRE,ITE,IT
C
C IN  XMIN=
C IN  XMAX=
C IN  YMIN=
```

```
C IN  YMAX=X AND Y LIMITS OF ICE ACCRETION PLOT WINDOW.
C IN  LYRMAX=NO. OF LAYERS TO BE ACCRETED.
C IN  PLTFAC=PLOT EXPANSION/REDUCTION FACTOR.
C
      NDCPX=0
      NDCPY=0
      DX=(XMAX-XMIN)/20.0
      DY=(YMAX-YMIN)/12.0
      DDX=ABS(4.0*DX)+1.E-6
      DDY=ABS(2.0*DY)+1.E-6
500   IF(DDX-AINT(DDX).LT.2./10.**(6-NDCPX))GOTO 510
      NDCPX=NDCPX+1
      DDX=DDX*10.0
      GOTO 500
510   IF(DDY-AINT(DDY).LT.2./10.**(6-NDCPY))GOTO 520
      NDCPY=NDCPY+1
      DDY=DDY*10.0
      GOTO 510
C DRAW AXES FOR ICE ACCRETION PLOT.
520   CALL NEWPEN(1)
      CALL FACTOR(PLTFAC)
      CALL ORIGIN(999.20.0,12.0,5.0,5.0)
      CALL AX2EP(4.0,3,NDCPX,1,0.9)
      CALL AXIS2(0.,0.,'X/C',-3,20.,0.,XMIN,DX,4.0)
      CALL AXIS2(20.,0.,' ',-1,-12.0,90.,0.,0.,2.0)
      CALL AX2EP(2.0,3,NDCPY,1,1.2)
      CALL AXIS2(0.,0.,'Y/C',3,12.0,90.,YMIN,DY,-2.0)
      CALL AXIS2(0.,12.0,' ',1,-20.,0.,0.,0.,4.0)
      LYRM1=LYRMAX+1
        DO 100 I=1,LYRM1
        ITT=IT(I)
        ITTE=ITE(I)
          DO 110 J=1,ITT
          XPLT(J)=XGR(J,I)
          YPLT(J)=YGR(J,I)
110       CONTINUE
          DO 210 J=1,ITTE
          XPLTE(J)=XGRE(J,I)
          YPLTE(J)=YGRE(J,I)
210       CONTINUE
        CALL NEWPEN(3)
C PLOT ACCRETION OUTLINES.
        CALL LINE(XPLT,YPLT,IT(I)-2,1,0,0)
        CALL LINEP(0.07)
C PLOT CONTROL SEGMENT ENDPTS.
        CALL LINE(XPLTE,YPLTE,ITE(I)-2,1,-1,0)
100     CONTINUE
      RETURN
      END

C
C ******************************************************************
C
      SUBROUTINE HERM5(L,YO,DYDL,N,CEE)
C
C WRITTEN BY: M. OLESKIW  ON: 810721  LAST MODIFIED: 811002
C
C CALCULATE COEFFICIENTS FOR HERMITE QUINTIC SPLINE.
C
      DOUBLE PRECISION L(31),YO(31),DYDL(31),CEE(5,30),U2,V2,S2,T2,
     .Z,R,U1,V1,S1,T1,DTAN,DATAN,SS2,SS3,SN1,SN2
C
      INTEGER N,N1,K,J2,J1,N2
C
C IN  L=LENGTH ALONG AEROFOIL SFC.
C IN  YO=TRAJECTORY STARTING VALUE.
```

```
C IN  DYDL=SLOPE OF YO VS L CURVE.
C IN  N=NUMBER OF DATA PTS. TO BE FITTED.
C OUT CEE=VECTOR OF COEFFICIENTS FOR QUINTIC HERMITE POLYNOMIAL SPLINE.
C
      N1=N-1
      N2=N-2
      SS2=(DYDL(2)-DYDL(1))/(L(2)-L(1))
      SS3=(DYDL(3)-DYDL(2))/(L(3)-L(2))
      SN1=(DYDL(N)-DYDL(N1))/(L(N)-L(N1))
      SN2=(DYDL(N1)-DYDL(N2))/(L(N1)-L(N2))
      CEE(2,1)=0.5D0*DTAN(2.D0*DATAN(SS2)-DATAN(SS3))
      CEE(2,N)=0.5D0*DTAN(2.D0*DATAN(SN1)-DATAN(SN2))
      CEE(4,1)=0.D0
      CEE(3,1)=0.D0
        DO 200 K=1,N1
        J2=K+1
        U2=1.D0/(L(J2)-L(K))
        CEE(5,K)=U2
        V2=U2*U2
        S2=10.D0*V2*U2*(YO(J2)-YO(K))
        T2=4.D0*V2*(DYDL(J2)+DYDL(K))
        IF(K.EQ.1)GOTO 100
        Z=1.D0/(3.D0*(U1+U2)+U1*CEE(4,J1))
        CEE(4,K)=-U2*Z
        R=S2-S1-T2+T1+2.D0*(V1-V2)*DYDL(K)
        IF(K.EQ.2)R=R+U1*CEE(2,1)
        IF(K.EQ.N1)R=R+U2*CEE(2,N)
        CEE(3,K)=Z*(R+U1*CEE(3,J1))
 100    J1=K
        U1=U2
        V1=V2
        S1=S2
        T1=T2
 200    CONTINUE
      CEE(2,N1)=CEE(3,N1)
      IF(N1.LE.2)GOTO 400
        DO 300 J1=2,N2
        K=N-J1
        CEE(2,K)=CEE(3,K)-CEE(4,K)*CEE(2,K+1)
 300    CONTINUE
 400    DO 500 K=1,N1
        J2=K+1
        R=YO(J2)-YO(K)
        Z=CEE(5,K)
        CEE(3,K)=Z*(CEE(2,J2)-3.D0*CEE(2,K)-Z*(6.D0*DYDL(K)
     +4.D0*DYDL(J2)-10.D0*Z*R))
        CEE(4,K)=Z*Z*(3.D0*CEE(2,K)-2.D0*CEE(2,J2)
     +Z*(8.D0*DYDL(K)+7.D0*DYDL(J2)-15.D0*Z*R))
        CEE(5,K)=Z*Z*Z*(CEE(2,J2)-CEE(2,K)
     -3.D0*Z*(DYDL(K)+DYDL(J2)-2.D0*Z*R))
        CEE(1,K)=DYDL(K)
 500    CONTINUE
      RETURN
      END
C
C ****************************************************************
C
      SUBROUTINE HERMIT(XO,X1,YO,Y1,YPO,YP1,A,B,C)
C
C WRITTEN BY: M. OLESKIW  ON 810414  LAST MODIFIED:
C
C CALCULATE THE HERMITE CUBIC POLYNOMIAL INTERPOLATOR
C   GIVEN THE FUNCTION AND ITS DERIVATIVES AT THE
C   ENDPTS. OF THE INTERVAL.
C REF: BURDEN,R.L. ET AL.(1978), BUMERICAL ANALYSIS,
```

```
C       PRINDLE,WEBER, & SCHMIDT, BOSTON, QA 297 B84, P 109.
C
      DOUBLE PRECISION XO,X1,YO,Y1,YPO,YP1,A,B,C,Z
C
C IN  XO=
C IN  X1=LEFT AND RIGHT BOUNDS OF THE INTERVAL
C IN  YO=
C IN  Y1=FN. VALUES AT ENDS OF INTERVAL.
C IN  YPO=
C IN  YP1=DERIVATIVES OF FN. AT ENDS OF INTERVAL.
C OUT A=
C OUT B=
C OUT C=CUBIC POLYNOMIAL COEFFICIENTS.
C
      Z=X1-XO
      A=(2.DO*YO/Z-2.DO*Y1/Z+YPO+YP1)/Z/Z
      B=(3.DO*Y1/Z-3.DO*YO/Z-2.DO*YPO-YP1)/Z
      C=YPO
      RETURN
      END

C
C ****************************************************************
C
      SUBROUTINE HIST(T,G)
C
C WRITTEN BY: M. OLESKIW  ON:801216  LAST MODIFIED:810626
C
C DETERMINES VALUE OF INTEGRAL IN HISTORY TERM,FOR U COMPONENT EQN.
C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978)
C       NUMERICAL ANALYSIS  P. 90   QA 297.B84
C
      DOUBLE PRECISION TAU3,TAU2,TAU1,TAUO,P11,P10,P21,P22,P20,
     .P33,P32,P31,P30,TO,T1,T2,T3,TS(500,2),FO,F1,F2,F3,DSQRT,DTS(6,2),
     .HT(2,6,2),T,A,B,C,D,F,AN(2,6,2),P(2,745,2),Z2,Z33,Z32,Z31,Z30,
     .AA,BB
C
      INTEGER J,L,FF,E,MOD,JI,JJ,G,I(2),IM4(2),IM3(2),IM2(2),IM1(2),
     .IO(2),IP1(2),MM,II
C
      COMMON /LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM/INTEG/AN,HT
C
C IN  T=TIME AT PRESENT TIME STEP.
C IN  G=O:EXTRAPOLATE HISTORY TERM SEQUENCE.
C IN     1:CALCULATE NEW HISTORY TERM.
C
      TAU3(A,B)=((5.DO*A**3+6.DO*A*A*T+8.DO*A*T*T+16.DO*T**3)
     .*DSQRT(T-A)-(5.DO*B**3+6.DO*B*B*T+8.DO*B*T*T+16.DO*T**3)
     .*DSQRT(T-B))*2.DO/35.DO
      TAU2(A,B)=((3.DO*A*A+4.DO*A*T+8.DO*T*T)*DSQRT(T-A)
     .-(3.DO*B*B+4.DO*B*T+8.DO*T*T)*DSQRT(T-B))*2.DO/15.DO
      TAU1(A,B)=((2.DO*T+A)*DSQRT(T-A)-(2.DO*T+B)*DSQRT(T-B))*2.DO/3.DO
      TAUO(A,B)=2.DO*(DSQRT(T-A)-DSQRT(T-B))
C
C STATEMENT FNS. TO FIND THE TERMS OF THE LAGRANGE POLY. FIT.
      P11(TO)=(F1-FO)/(T1-TO)
      P10(TO)=(FO*T1-F1*TO)/(T1-TO)
      Z2(A,B,C,F)=F/(A-B)/(A-C)
      P22(TO)=Z2(TO,T1,T2,FO)+Z2(T1,TO,T2,F1)+Z2(T2,TO,T1,F2)
      P21(TO)=-(T1+T2)*Z2(TO,T1,T2,FO)-(TO+T2)*Z2(T1,TO,T2,F1)
     .-(TO+T1)*Z2(T2,TO,T1,F2)
      P20(TO)=T1*T2*Z2(TO,T1,T2,FO)+TO*T2*Z2(T1,TO,T2,F1)
     .+TO*T1*Z2(T2,TO,T1,F2)
      Z33(A,B,C,D,F)=F/(A-B)/(A-C)/(A-D)
      P33(TO)=Z33(TO,T1,T2,T3,FO)+Z33(T1,TO,T2,T3,F1)
     .+Z33(T2,TO,T1,T3,F2)+Z33(T3,TO,T1,T2,F3)
```

```
          Z32(A,B,C,D,F)=-(B+C+D)*F/(A-B)/(A-C)/(A-D)
          P32(TO)=Z32(TO,T1,T2,T3,FO)+Z32(T1,TO,T2,T3,F1)
          +Z32(T2,TO,T1,T3,F2)+Z32(T3,TO,T1,T2,F3)
          Z31(A,B,C,D,F)=(B*C+B*D+C*D)*F/(A-B)/(A-C)/(A-D)
          P31(TO)=Z31(TO,T1,T2,T3,FO)+Z31(T1,TO,T2,T3,F1)
          +Z31(T2,TO,T1,T3,F2)+Z31(T3,TO,T1,T2,F3)
          Z30(A,B,C,D,F)=-B*C*D*F/(A-B)/(A-C)/(A-D)
          P30(TO)=Z30(TO,T1,T2,T3,FO)+Z30(T1,TO,T2,T3,F1)
          +Z30(T2,TO,T1,T3,F2)+Z30(T3,TO,T1,T2,F3)
C
          II=I(MM)
          IF(G.EQ.1)GOTO 200
C EXTRAPOLATION OF HISTORY TERM SEQUENCE
          GOTO(140,120,100),II
          TO=TS(II-3,MM)
          T1=TS(II-2,MM)
          T2=TS(II-1,MM)
          T3=TS(II,MM)
            DO 110 J=1,2
            FO=HT(J,IM3(MM),MM)
            F1=HT(J,IM2(MM),MM)
            F2=HT(J,IM1(MM),MM)
            F3=HT(J,IO(MM),MM)
            HT(J,IP1(MM),MM)=P33(TO)*T**3+P32(TO)*T*T+P31(TO)*T+P30(TO)
  110       CONTINUE
          RETURN
C
  100     TO=TS(1,MM)
          T1=TS(2,MM)
          T2=TS(3,MM)
            DO 130 J=1,2
            FO=HT(J,IM2(MM),MM)
            F1=HT(J,IM1(MM),MM)
            F2=HT(J,IO(MM),MM)
            HT(J,IP1(MM),MM)=P22(TO)*T*T+P21(TO)*T+P20(TO)
  130       CONTINUE
          RETURN
C
  120     TO=TS(1,MM)
          T1=TS(2,MM)
            DO 150 J=1,2
            FO=HT(J,IM1(MM),MM)
            F1=HT(J,IO(MM),MM)
            HT(J,IP1(MM),MM)=P11(TO)*T+P10(TO)
  150       CONTINUE
          RETURN
C
  140     HT(1,IP1(MM),MM)=0.DO
          HT(2,IP1(MM),MM)=0.DO
          RETURN
C
  200     L=(I(MM)-4)/2*3+1
          HT(1,IP1(MM),MM)=0.DO
          HT(2,IP1(MM),MM)=0.DO
          GOTO(400,500,600,700),II
          FF=MOD(I(MM),2)
          E=II-5+FF
C EVALUATE INTEGRAL UP TO LAST SEVERAL INTERVALS
            DO 210 J=1,E,2
            AA=TS(J,MM)
            BB=TS(J+2,MM)
            JI=(J-1)/2*3+1
              DO 220 JJ=1,2
              HT(JJ,IP1(MM),MM)=HT(JJ,IP1(MM),MM)+P(JJ,JI,MM)*TAU2(AA,BB)
              +P(JJ,JI+1,MM)*TAU1(AA,BB)+P(JJ,JI+2,MM)*TAUO(AA,BB)
  220         CONTINUE
```

```
      210     CONTINUE
            IF(FF.EQ.1)GOTO 600
C EVALUATE INTEGRAL FOR LAST 4 INTERVALS
C   USING TWO INTERVAL PAIRS (FOR I EVEN)
      700   TO=TS(II-3,MM)
            T1=TS(II-2,MM)
            T2=TS(II-1,MM)
              DO 710 J=1,2
              FO=AN(J,IM3(MM),MM)
              F1=AN(J,IM2(MM),MM)
              F2=AN(J,IM1(MM),MM)
C FIT A 2ND ORDER LAGRANGE POLYNOMIAL
              P(J,L,MM)=P22(TO)
              P(J,L+1,MM)=P21(TO)
              P(J,L+2,MM)=P20(TO)
              HT(J,IP1(MM),MM)=HT(J,IP1(MM),MM)+P(J,L,MM)*TAU2(TO,T2)+
     .        P(J,L+1,MM)*TAU1(TO,T2)+
     .        P(J,L+2,MM)*TAUO(TO,T2)
      710     CONTINUE
C FOR THE SECOND PAIR OF THE SET
C   (OR FOR THE VERY FIRST PAIR OF INTERVALS)
      500   TO=TS(II-1,MM)
            T1=TS(II,MM)
            T2=TS(II+1,MM)
              DO 720 J=1,2
              FO=AN(J,IM1(MM),MM)
              F1=AN(J,IO(MM),MM)
              F2=AN(J,IP1(MM),MM)
              HT(J,IP1(MM),MM)=HT(J,IP1(MM),MM)+P22(TO)*TAU2(TO,T2)+
     .        P21(TO)*TAU1(TO,T2)+
     .        P20(TO)*TAUO(TO,T2)
      720     CONTINUE
            RETURN
C
C EVALUATE INTEGRAL FOR LAST 3 INTERVALS (FOR I ODD)
      600   TO=TS(II-2,MM)
            T1=TS(II-1,MM)
            T2=TS(II,MM)
            T3=TS(II+1,MM)
              DO 610 J=1,2
              FO=AN(J,IM2(MM),MM)
              F1=AN(J,IM1(MM),MM)
              F2=AN(J,IO(MM),MM)
              F3=AN(J,IP1(MM),MM)
              HT(J,IP1(MM),MM)=HT(J,IP1(MM),MM)+P33(TO)*TAU3(TO,T3)+
     .        P32(TO)*TAU2(TO,T3)+
     .        P31(TO)*TAU1(TO,T3)+P30(TO)*TAUO(TO,T3)
      610     CONTINUE
            RETURN
C
C EVALUATE INTEGRAL FOR THE FIRST INTERVAL
      400   TO=TS(1,MM)
            T1=TS(2,MM)
              DO 410 J=1,2
              FO=AN(J,IO(MM),MM)
              F1=AN(J,IP1(MM),MM)
              HT(J,IP1(MM),MM)=HT(J,IP1(MM),MM)+P11(TO)*TAU1(TO,T1)+
     .        P10(TO)*TAUO(TO,T1)
      410     CONTINUE
            RETURN
            END

C
C ---------------------------------------------------------------------
C
            SUBROUTINE ICING(LTOL,ICE,BOTH,FAIL,DDISTN,ATHICK,FILTER)
```

```
      C
      C WRITTEN BY: M. OLESKIW  ON:800713  LAST MODIFIED:811024
      C
      C CALCULATE AMOUNT OF ACCRETION AND DETERMINE A NEW SET OF AEROFOIL
      C    SURFACE ELEMENT ENDPOINTS AFTER DETERMINING THE AEROFOIL
      C    NOSE LOCATION.
      C
2           DOUBLE PRECISION XN,YN,XNN,YNN,XUR(101),YUR(101),BU(101),
           .CU(100,3),CL(100,3),XLR(101),YLR(101),L(31,5),YO(31,5),
           .D,CEED(5,30,5),LTOL,DSIGN,XLRN(101),YLRN(101),XNM,BL(101),
           .S30,C30,NSURF,XURN(101),YURN(101),CEU(101,5),CEL(101,5),
           .XUT(101),XLT(101),XU(101),YU(101),XL(101),YL(101),E(2),
           .YUT(101),YLT(101),DABS,DSORT,LU(101),LL(101),ICE,MDU(101),
           .TOL,LE,RE,ICEE,ALPHAR,DCOS,LUP(101),LLP(101),SL,MDL(101)
3           DOUBLE PRECISION NSURFY,XRMIN,XNP,YNP,DDD,DSIN,FXU,FXL,
           .ACU(101),ACL(101),KU(101),KL(101),W(5),XI,YI,C,DD(5),DFLOAT,
           .THKU(101),THKL(101),UNEU,VNEU,UNEL,BETA,FLV,LUM,LLM,ACCRT,
           .VNEL,CFV(3,30,5),CFA(3,30,5),MMD,VTOT(31,5),ACOL(31,5),
           .RU(101),RL(101),RHOSU(101),RHOSL(101),UINF,PINF,TINF,
           .VTIMP,TI,PAR,DMIN1,DMAX1,EPS(5),KLF,LRU,LRL,TOLL,R1,R2,R3,
           .VTL(101),VTU(101),YOE(2),FL(5),FILTER,LDL,LDR,LMXCE(6),LMX
      C
4           INTEGER BOTH,J,NCOU,NCOL,ICT(5),ICU(5),ICL(5),I,IER,NOAC,
           .IM,IUS,ILS,IK,FAIL,RUN,NEU,NEL,IU(51),IL(51),ATH,DDSTN,
           .KK,IXU(101),IXL(101),IUU,ILL,NEUP,NELP,KJP,KJ,NCOUM1,II,JK,
           .NCOLM1,DDISTN,JJ,ATHICK,TYPE,DENSE,H5,N,ICTT,NELLP,NEUUP,
           .IMAXU,IMAXL,IMXU,IMXL,M,NEUUP1,NELLP1,KI,NEUU,NELL,JI,IJ,
           .LAYER,ATYPE,H5D(5)
      C
5           COMMON ALPHAR/AERO3/NCOU,NCOL/NOSE/XN,YN/FOIL/XUR,YUR,
           .XLR,YLR/ROTP/C30,S30/CEM/LMXCE/IND/NSURFY,FL,ICEE,I,JJ,RUN
           ./AERO4/NEU,NEL,NEUU,NELL/CEV/CFV,CFA/RC/RU,RL,ATH,DDSTN
           ./SPLINE/CU,CL/ENDS/IU,IL,IXU,IXL/NNOSE/XNP,YNP/HERMT5/H5,H5D
           ./WTS/W/TRANS3/DD,C,TYPE,J/CV/VTOT,ACOL
           ./TRANS1/UINF,PINF,TINF,EPS,DENSE/LLR/ACCRT,LAYER,ATYPE
           ./COL/L,YO,ICT,ICU,ICL/EFF/CEED/SFCS/XU,YU,XL,YL/LG/LU,LL
      C
6           EXTERNAL NSURF
      C
      C IN  LTOL=MAX. INCREASE ALLOWED IN LENGTH BETWEEN CEE'S
      C          BETWEEN SUCCESSIVE AIRFOIL SURFACES.
      C IN  ICE=THICKNESS OF ICE ACCRETION ASSUMING CE=100%.
      C IN  BOTH=TRAJECTORIES FOR BOTH SFCS. (0 OR 1)
      C OUT FAIL=FAILURE INDICATOR.
      C IN  DDISTN=NO. OF SIZES IN DROPLET DISTN.
      C IN  ATHICK=INCORPORATE SFC. CURVATURE IN ACCRETION THICKNESS
      C             CALCULATION.
      C IN  DENSE=VARY DENSITY OF ICE ACCRETION (0 OR 1)
      C IN  FILTER=LENGTH OF BOXCAR FILTER(AS A FRACTION OF L RANGE OF
      C             LARGEST DROPLET SIZE) TO BE APPLIED TO SMOOTH CE VS L
      C             CURVE(S).  IF 0, THEN DON'T FILTER. (F)
      C
7      10   FORMAT('0FAILURE TO CONVERGE TO NEW NOSE POSITION')
8      15   FORMAT('-MASS MEAN DIAMETER:',F7.1,' MICROMETERS.')
9      20   FORMAT('0',T64,5('  DROPLET    '),/,' ',T64,5('  DIAMETER   '),/,
           .' ',T64,5(F6.1,' UM   '))
10     25   FORMAT(' ',T26,'DISTANCE    LAYER     NORM.    AVERAGE',/,
           .'  END',T28,'FROM    ACCRETION   ICE    ',6('COLLISION  '),
           ./,' POINT   X COORD   Y COORD    NOSE    THICKNESS DENSITY ',
           .6('EFFICIENCY '),/,' ')
11     30   FORMAT(' ',I4,F10.5,3F9.5,F8.3,2PF9.2,5(2PF11.2))
12     40   FORMAT(' ')
      C
13          DDSTN=DDISTN
14          ATH=ATHICK
15            DO 100 I=1,NEU
```

```
           ACU(I)=0.D0
           VTU(I)=0.D0
           MDU(I)=0.D0
           BU(I)=0.D0
           RHOSU(I)=1.D0
           IF(I.GT.NEUU)GOTO 100
C FIND PERPENDICULAR SLOPES FOR UPPER SFC. POINTS.
           KU(I)=-1.D0/DSIGN(DMAX1(DABS(CU(I,1)),1.D-10),CU(I,1))
C FIND RADUIS OF CURVATURE FOR UPPER SFC. POINTS
           RU(I)=((1.D0+CU(I,1)**2)**1.5D0)/2.D0/
           DSIGN(DMAX1(DABS(CU(I,2)),1.D-10),-CU(I,2))
 100       CONTINUE
         IF(BOTH.EQ.0)GOTO 180
           DO 110 I=1,NEL
           ACL(I)=0.D0
           VTL(I)=0.D0
           MDL(I)=0.D0
           BL(I)=0.D0
           RHOSL(I)=1.D0
           IF(I.GT.NELL)GOTO 110
C FIND PERPENDICULAR SLOPES FOR LOWER SFC. POINTS.
           KL(I)=-1.D0/DSIGN(DMAX1(DABS(CL(I,1)),1.D-10),CL(I,1))
C FIND RADIUS OF CURVATURE FOR LOWER SFC. POINTS.
           RL(I)=((1.D0+CL(I,1)**2)**1.5D0)/2.D0/
           DSIGN(DMAX1(DABS(CL(I,2)),1.D-10),CL(I,2))
 110       CONTINUE
 180     MMD=0.D0
C
C FIND WEIGHTED AND/OR FILTERED COLLISION EFFICIENCY
C FOR UPPER SFC.:
           DO 205 J=1,DDISTN
           JJ=ICL(J)
           JI=ICL(J)
           ICTT=ICT(J)
           FL(J)=FILTER*(L(ICTT,J)-L(1,J))/2.D0
           LDL=L(1,J)
           LDR=L(ICTT,J)
           LMX=LMXCE(J)
C FIND VOLUME MEAN DIAMETER.
         MMD=MMD+W(J)*DD(J)
            DO 200 I=1,NEU
            IF(I.GT.NEUU)GOTO 215
 220        IF(LU(I).LE.L(JI+1,J))GOTO 240
            JI=JI+1
            IF(JI.LT.ICT(J))GOTO 220
C NO ACCRETION REGION
 215        CEU(I,J)=0.D0
            IF(FILTER.EQ.0.D0)GOTO 1375
            GOTO 1400
 240        D=LU(I)-L(JI,J)
C CALCULATE INTERPOLATED CE.
            IF(H5D(J).EQ.0)CEU(I,J)=((3.D0*CEED(3,JI,J)*D
            +2.D0*CEED(2,JI,J))*D+CEED(1,JI,J))*DCOS(ALPHAR)
            IF(H5D(J).EQ.1)CEU(I,J)=((((5.D0*CEED(5,JI,J)*D
            +4.D0*CEED(4,JI,J))*D+3.D0*CEED(3,JI,J))*D
            +2.D0*CEED(2,JI,J))*D+CEED(1,JI,J))*DCOS(ALPHAR)
            IF(FILTER.NE.0.D0)GOTO 1400
            BU(I)=BU(I)+CEU(I,J)*W(J)
            GOTO 1375
 1400       IF(LU(I).GE.LMX)GOTO 1610
            IF(LU(I).LE.LDL)GOTO 1620
            FLV=FL(J)-0.9D0*FL(J)/(LMX-LDL)*(LU(I)-LDL)
            GOTO 1600
 1610       IF(LU(I).GE.LDR)GOTO 1620
            FLV=0.1D0*FL(J)+0.9D0*FL(J)/(LDR-LMX)*(LU(I)-LMX)
            GOTO 1600
```

```
1620       FLV=FL(J)
1600       E(1)=LU(I)-FLV
           E(2)=LU(I)+FLV
             DO 1430 IJ=1,2
              IF(E(IJ).GT.L(1,J))GOTO 1440
              YOE(IJ)=YO(1,J)
              GOTO 1430
1440          IF(E(IJ).LT.L(ICTT,J))GOTO 1450
              YOE(IJ)=YO(ICTT,J)
              GOTO 1430
1450          IF(E(IJ).GT.L(JJ,J))GOTO 1460
              JJ=JJ-1
              GOTO 1450
1460          IF(E(IJ).LE.L(JJ+1,J))GOTO 1470
              JJ=JJ+1
              GOTO 1460
1470          D=E(IJ)-L(JJ,J)
              IF(H5D(J).EQ.0)YOE(IJ)=((CEED(3,JJ,J)*D
              +CEED(2,JJ,J))*D+CEED(1,JJ,J))*D+YO(JJ,J)
              IF(H5D(J).EQ.1)YOE(IJ)=((((CEED(5,JJ,J)*D
              +CEED(4,JJ,J))*D+CEED(3,JJ,J))*D
              +CEED(2,JJ,J))*D+CEED(1,JJ,J))*D+YO(JJ,J)
1430          CONTINUE
           BETA=(YOE(2)-YOE(1))/2.DO/FLV*DCOS(ALPHAR)
           BU(I)=BU(I)+W(J)*BETA
1375       IF(DENSE.EQ.0)GOTO 200
C CALCULATE AVERAGE DROPLET IMPACT VELOCITY COMPONENTS.
           IF(LU(I).LE.L(ICTT,J))GOTO 1380
           IF(J.GT.1)GOTO 200
           VTU(I)=VTU(I)+W(1)*VTOT(ICTT,1)
           ACU(I)=ACU(I)+W(1)*DABS(ACOL(ICTT,1))
           MDU(I)=MDU(I)+W(1)*DD(1)
           GOTO 200
1380       D=LU(I)-L(JI,J)
           VTU(I)=VTU(I)+W(J)*(((CFV(3,JI,J)*D+CFV(2,JI,J))*D
           +CFV(1,JI,J))*D+VTOT(JI,J))
           ACU(I)=ACU(I)+W(J)*DABS(((CFA(3,JI,J)*D+CFA(2,JI,J))*D
           +CFA(1,JI,J))*D+ACOL(JI,J))
           MDU(I)=MDU(I)+W(J)*DD(J)
200        CONTINUE
         IF(BOTH.EQ.0)GOTO 205
C
C FOR LOWER SFC:
         JJ=ICL(J)+1
         JI=ICL(J)+1
           DO 210 I=1,NEL
           IF(I.GT.NELL)GOTO 225
230        IF(LL(I).GT.L(JI,J))GOTO 250
           JI=JI-1
           IF(JI.GT.0)GOTO 230
C NO ACCRETION REGION.
225        CEL(I,J)=0.DO
           IF(FILTER.EQ.0.DO)GOTO 1475
           GOTO 1500
250        D=LL(I)-L(JI,J)
C CALCULATE INTERPOLATED CE.
           IF(H5D(J).EQ.0)CEL(I,J)=((3.DO*CEED(3,JI,J)*D
           +2.DO*CEED(2,JI,J))*D+CEED(1,JI,J))*DCOS(ALPHAR)
           IF(H5D(J).EQ.1)CEL(I,J)=((((5.DO*CEED(5,JI,J)*D
           +4.DO*CEED(4,JI,J))*D+3.DO*CEED(3,JI,J))*D
           +2.DO*CEED(2,JI,J))*D+CEED(1,JI,J))*DCOS(ALPHAR)
           IF(FILTER.NE.0.DO)GOTO 1500
           BL(I)=BL(I)+CEL(I,J)*W(J)
           GOTO 1475
1500       IF(LL(I).GE.LMX)GOTO 1710
           IF(LL(I).LE.LDL)GOTO 1720
```

```
                FLV=FL(J)-0.9D0*FL(J)/(LMX-LDL)*(LL(I)-LDL)
                GOTO 1700
      1710      IF(LL(I).GE.LDR)GOTO 1720
                FLV=0.1D0*FL(J)+0.9D0*FL(J)/(LDR-LMX)*(LL(I)-LMX)
                GOTO 1700
      1720      FLV=FL(J)
      1700      E(1)=LL(I)-FLV
                E(2)=LL(I)+FLV
                  DO 1530 IJ=1,2
                  IF(E(IJ).GT.L(1,J))GOTO 1540
                  YOE(IJ)=YO(1,J)
                  GOTO 1530
      1540        IF(E(IJ).LT.L(ICTT,J))GOTO 1550
                  YOE(IJ)=YO(ICTT,J)
                  GOTO 1530
      1550        IF(E(IJ).GT.L(JJ,J))GOTO 1560
                  JJ=JJ-1
                  GOTO 1550
      1560        IF(E(IJ).LE.L(JJ+1,J))GOTO 1570
                  JJ=JJ+1
                  GOTO 1560
      1570        D=E(IJ)-L(JJ,J)
                  IF(H5D(J).EQ.0)YOE(IJ)=((CEED(3,JJ,J)*D
                  +CEED(2,JJ,J))*D+CEED(1,JJ,J))*D+YO(JJ,J)
                  IF(H5D(J).EQ.1)YOE(IJ)=((((CEED(5,JJ,J)*D
                  +CEED(4,JJ,J))*D+CEED(3,JJ,J))*D
                  +CEED(2,JJ,J))*D+CEED(1,JJ,J))*D+YO(JJ,J)
      1530        CONTINUE
                BETA=(YOE(2)-YOE(1))/2.D0/FLV*DCOS(ALPHAR)
                BL(I)=BL(I)+W(J)*BETA
      1475      IF(DENSE.EQ.0)GOTO 210
      C CALCULATE AVERAGE DROPLET IMPACT VELOCITY COMPONENTS.
                IF(LL(I).GE.L(1,J))GOTO 1480
                IF(J.GT.1)GOTO 210
                VTL(I)=VTL(I)+W(1)*VTOT(1,1)
                ACL(I)=ACL(I)+W(1)*DABS(ACOL(1,1))
                MDL(I)=MDL(I)+W(1)*DD(1)
                GOTO 210
      1480      D=LL(I)-L(JI,J)
                VTL(I)=VTL(I)+W(J)*(((CFV(3,JI,J)*D+CFV(2,JI,J))*D
                +CFV(1,JI,J))*D+VTOT(JI,J))
                ACL(I)=ACL(I)+W(J)*DABS(((CFA(3,JI,J)*D+CFA(2,JI,J))*D
                +CFA(1,JI,J))*D+ACOL(JI,J))
                MDL(I)=MDL(I)+W(J)*DD(J)
      210       CONTINUE
      205     CONTINUE
      C
      C ACCRETE ICE ON EACH SFC (TAKING INTO ACCOUNT THE
      C    SFC CURVATURE IF ATHICK IS 1)
      C FOR THE UPPER SFC.
              NOAC=0
                DO 300 I=1,NEU
                IF(NOAC.EQ.1)GOTO 320
                IF(BU(I).EQ.0.D0)NOAC=1
      C INDEX OF BEGINNING OF NO ACCRETION REGION.
                IF(NOAC.EQ.1)IMAXU=I
                IF(DENSE.EQ.0)GOTO 355
      C CALCULATE MEAN ICE DENSITY
                UNEU=VTU(I)*DSIN(ACU(I))
                VNEU=VTU(I)*DCOS(ACU(I))
                IF(DENSE.EQ.1)VTIMP=VNEU*UINF
                IF(DENSE.EQ.2)VTIMP=DSQRT(UNEU**2+VNEU**2)*UINF
      C ICE DENSITY ACCORDING TO MACKLIN (1962)
                TI=DMAX1(TINF,-20.D0)
                TI=DMIN1(TI,-5.D0)
                PAR=-MDU(I)/2.D0*VTIMP/TI
```

```
          PAR=DMIN1(PAR,16.29D0)
          PAR=DMAX1(PAR,0.88D0)
          RHOSU(I)=1.1D2*(PAR**0.76D0)/9.17D2
355       THKU(I)=ICE*BU(I)/RHOSU(I)
          IF(ATHICK.EQ.0)GOTO 330
          IF(ATHICK.EQ.-1)GOTO 370
          IF(I.NE.1)GOTO 340
          IF(BOTH.EQ.1)GOTO 365
          R1=DSIGN(2.D0*DABS(RU(2))+DABS(RU(1)),RU(1))/3.D0
          GOTO 380
365       R1=DSIGN(DABS(RL(2))+DABS(RU(1))+DABS(RU(2)),RU(1))/3.D0
          GOTO 380
C UNSMOOTHED RADIUS OF CURVATURE.
370       R1=RU(I)
          GOTO 380
340       R1=DSIGN(DABS(RU(I-1))+DABS(RU(I))+DABS(RU(I+1)),RU(I))/3.D0
C ICE ACCRETION THICKNESS INCORPORATING CURVED SFC.
380       IF(I.EQ.1)R2=R1
          IF(I.EQ.2)R3=R1
          THKU(I)=-R1+DSIGN(DSQRT(R1**2+2.D0*R1*THKU(I)),R1)
C COORDS. FOR NEW SFC.
330       XURN(I)=XUR(I)+DSIGN(DSQRT(THKU(I)**2/
         .(1.D0+KU(I)**2)),KU(I))
          YURN(I)=YUR(I)+KU(I)*(XURN(I)-XUR(I))
          GOTO 300
C NO ACCRETION
320       XURN(I)=XUR(I)
          YURN(I)=YUR(I)
300       CONTINUE
        IF(R2.GE.R3)GOTO 306
C ICE ACCRETION THICKNESS INCORPORATING CURVED SFC.
        THKU(1)=-R3+DSIGN(DSQRT(R3**2+2.D0*R3*ICE*BU(1)
        /RHOSU(1)),R3)
        XURN(1)=XUR(1)+DSIGN(DSQRT(THKU(1)**2/
       .(1.D0+KU(1)**2)),KU(1))
        YURN(1)=YUR(1)+KU(1)*(XURN(1)-XUR(1))
306     WRITE(7,15)MMD
        WRITE(7,20)(DD(N),N=1,DDISTN)
        WRITE(7,25)
          DO 190 I=1,NEUU
          WRITE(7,30)I,XU(I),YU(I),LU(I),THKU(I),RHOSU(I),BU(I),
         .(CEU(I,N),N=1,DDISTN)
          IF(BU(I).EQ.0.D0)GOTO 406
190       CONTINUE
406     IF(BOTH.EQ.0)GOTO 590
C
C FOR THE LOWER SFC.
        NOAC=0
          DO 400 I=1,NEL
          IF(NOAC.EQ.1)GOTO 420
          IF(BL(I).EQ.0.D0)NOAC=1
C INDEX OF BEGINNING OF NO ACCRETION REGION.
          IF(NOAC.EQ.1)IMAXL=I
C CALCULATE MEAN ICE DENSITY
          IF(DENSE.EQ.0)GOTO 455
          UNEL=VTL(I)*DSIN(ACL(I))
          VNEL=VTL(I)*DCOS(ACL(I))
          IF(DENSE.EQ.1)VTIMP=VNEL*UINF
          IF(DENSE.EQ.2)VTIMP=DSQRT(UNEL**2+VNEL**2)*UINF
C NORMALIZED ICE DENSITY ACCORDING TO MACKLIN (1962).
          TI=DMAX1(TINF,-20.D0)
          TI=DMIN1(TI,-5.D0)
          PAR=-MDL(I)/2.D0*VTIMP/TI
          PAR=DMIN1(PAR,16.29D0)
          PAR=DMAX1(PAR,0.88D0)
          RHOSL(I)=1.1D2*(PAR**0.76D0)/9.17D2
```

```
      455      THKL(I)=ICE*BL(I)/RHOSL(I)
               IF(ATHICK EQ.0)GOTO 430
               IF(I NE 1)GOTO 440
               THKL(1)=THKU(1)
               GOTO 430
      440      IF(ATHICK EQ.-1)GOTO 470
C FIND RADIUS OF CURVATURE
               R1=DSIGN(DABS(RL(I-1))+DABS(RL(I))+DABS(RL(I+1)),RL(I))/3.DO
               GOTO 480
      470      R1=RL(I)
C ICE ACCRETION THICKNESS INCORPORATING CURVED SFC.
      480      THKL(I)=-R1+DSIGN(DSQRT(R1**2+2.DO*R1*THKL(I)),R1)
C COORDS. FOR NEW SFC
      430      XLRN(I)=XLR(I)-DSIGN(DSQRT(THKL(I)**2/
               (1.DO+KL(I)**2)),KL(I))
               YLRN(I)=YLR(I)+KL(I)*(XLRN(I)-XLR(I))
               GOTO 400
C NO ACCRETION
      420      XLRN(I)=XLR(I)
               YLRN(I)=YLR(I)
      400      CONTINUE
             WRITE(7,40)
               DO 235 I=1,NELL
               WRITE(7,30)I,XL(I),YL(I),LL(I),THKL(I),RHOSL(I),BL(I),
               (CEL(I,N),N=1,DDISTN)
               IF(BL(I) EQ.O.DO)GOTO 900
      235      CONTINUE
             GOTO 900
C
C UPPER & LOWER SFCS. MIRROR IMAGES; NOSE STAYS ON THE X-AXIS.
      590      DO 595 I=1,NEU
               XLRN(I)=XURN(I)
               YLRN(I)=-YURN(I)
      595      CONTINUE
             IMAXL=IMAXU
             GOTO 930
C
C FIND NEW NOSE LOCATION USING THE GOLDEN SECTION SEARCH METHOD
C   OF DETERMINING THE MIN. VALUE OF THE NEW SURFACE X-COORD.
      900    ICEE=ICE
             RUN=O
             I=1
             JJ=1
             IF(LMXCE(6) LT.O.DO)GOTO 905
             XNM=XN-THKU(1)
               DO 1010 KK=1,NCOU
               IF(XU(KK)-THKU(KK).GT.XNM)GOTO 1025
      1010     CONTINUE
      1025   LE=1.D-10
             RE=XUR(KK)
             GOTO 920
      905    XNM=XN-THKL(1)
               DO 910 KK=1,NCOL
               IF(XL(KK)-THKL(KK) GT.XNM)GOTO 925
      910      CONTINUE
      925    LE=1.D-10
             RE=XLR(KK)
C ***************************************************************
      920    TOL=1.D-8
C ***************************************************************
             FAIL=O
C LIMITS OF SEARCH
             CALL ZXGSN(NSURF,LE,RE,TOL,XRMIN,IER)
             IF(IER LT 129 OR IER GT 132)GOTO 950
             FAIL=1
             WRITE(6,10)
```

```
      WRITE(7,10)
      GOTO 720
C NEW NOSE COORDS
950   YNN=NSURFY
      XNN=NSURF(XRMIN)
C
C DE-ROTATE NEW UPPER & LOWER SFCS. ABOUT PREVIOUS NOSE POSITION
930     DO 500 I=1,NEU
        XUT(I)=XURN(I)*C30-YURN(I)*S30+XN
        YUT(I)=XURN(I)*S30+YURN(I)*C30+YN
500     CONTINUE
        DO 510 I=1,NEL
        XLT(I)=XLRN(I)*C30+YLRN(I)*S30+XN
        YLT(I)=-XLRN(I)*S30+YLRN(I)*C30+YN
510     CONTINUE
      IF(BOTH.EQ.1)GOTO 520
      XNN=XUT(1)
      YNN=YUT(1)
520   XU(1)=XNN
      XL(1)=XNN
      YU(1)=YNN
      YL(1)=YNN
      IUU=1
      ILL=1
      IF(BOTH.EQ.0)GOTO 625
      IF(LMXCE(6).LT.0.D0)GOTO 605
C
C SEE IF ANY UPPER SFC. ENDPTS. ARE BELOW THE NEW NOSE POSITION
C   & THUS BELONG ON THE LOWER SFC.
        DO 1110 IM=1,NEU
        IF(DABS(YUT(IM)-YNN).LT.0.2D0*(YUT(2)-YUT(1)))GOTO 1120
        IF(YUT(IM).GT.YNN)GOTO 1130
1110    CONTINUE
1120  IF(IM.GT.2)GOTO 1140
      IF(IM.EQ.2)GOTO 1150
C SAME NOSE INDEX
      IUS=2
      ILS=2
      GOTO 660
C NEW NOSE IS NEAR FIRST ENDPT. ABOVE PREVIOUS NOSE.
1150  IUS=3
      ILS=1
      GOTO 660
C NEW NOSE IS NEAR SECOND OR GREATER ENDPT. ABOVE PREVIOUS NOSE
1140  IK=IM-2
        DO 1170 I=1,IK
        ILL=ILL+1
        XL(ILL)=XUT(IM-I)
        YL(ILL)=YUT(IM-I)
1170    CONTINUE
      IUS=IM+1
      ILS=1
      GOTO 660
1130  IF(IM.GT.2)GOTO 1180
C NEW NOSE IS BETWEEN FIRST & SECOND ENDPTS. ON UPPER SFC.
      IUS=2
      ILS=1
      GOTO 660
C NEW NOSE IS ABOVE SECOND ENDPT. ON UPPER SFC.
1180  IK=IM-2
        DO 1190 I=1,IK
        ILL=ILL+1
        XL(ILL)=XUT(IM-I)
        YL(ILL)=YUT(IM-I)
1190    CONTINUE
      IUS=IM
```

```
          ILS=1
          GOTO 660
C
C SEE IF ANY LOWER SFC. ENDPTS. ARE ABOVE THE NEW NOSE POSITION
C   & THUS BELONG ON THE UPPER SFC.
605         DO 610 IM=1,NEL
            IF(DABS(YLT(IM)-YNN).LT.0.2D0*(YLT(1)-YLT(2)))GOTO 620
            IF(YLT(IM).LT.YNN)GOTO 630
610         CONTINUE
620       IF(IM.GT.2)GOTO 640
          IF(IM.EQ.2)GOTO 650
C SAME NOSE INDEX
625       IUS=2
          ILS=2
          GOTO 660
C NEW NOSE IS NEAR FIRST ENDPT. BELOW PREVIOUS NOSE
650       IUS=1
          ILS=3
          GOTO 660
C NEW NOSE IS NEAR SECOND OR GREATER ENDPT. BELOW PREVIOUS NOSE
640       IK=IM-2
            DO 670 I=1,IK
            IUU=IUU+1
            XU(IUU)=XLT(IM-I)
            YU(IUU)=YLT(IM-I)
670         CONTINUE
          IUS=1
          ILS=IM+1
          GOTO 660
630       IF(IM.GT.2)GOTO 680
C NEW NOSE IS BETWEEN FIRST & SECOND ENDPTS. ON LOWER SFC.
          IUS=1
          ILS=2
          GOTO 660
C NEW NOSE IS BELOW SECOND ENDPT. ON LOWER SFC.
680       IK=IM-2
            DO 690 I=1,IK
            IUU=IUU+1
            XU(IUU)=XLT(IM-I)
            YU(IUU)=YLT(IM-I)
690         CONTINUE
          IUS=1
          ILS=IM
660         DO 700 I=IUS,NEU
            IUU=IUU+1
            XU(IUU)=XUT(I)
            YU(IUU)=YUT(I)
700         CONTINUE
            DO 710 I=ILS,NEL
            ILL=ILL+1
            XL(ILL)=XLT(I)
            YL(ILL)=YLT(I)
710         CONTINUE
          NEUP=NEU
          NELP=NEL
          NEU=IUU
          NEL=ILL
          NEUUP=NEUU
          NELLP=NELL
          NEUU=NEUU+NEU-NEUP
          NELL=NELL+NEL-NELP
          XNP=XN
          YNP=YN
          XN=XNN
          YN=YNN
          IF(LAYER.GT.2)GOTO 750
```

```
      LUM=LU(2)
      LLM=LL(2)
750     DO 800 I=1,NEUUP
        LUP(I)=LU(I)
800     CONTINUE
        DO 810 I=1,NELLP
        LLP(I)=LL(I)
810     CONTINUE
        DO 840 I=1,NEU
        IXU(I)=0
840     CONTINUE
        DO 845 I=1,NEL
        IXL(I)=0
845     CONTINUE
      IXU(NEU)=1
      IXU(1)=1
      IXL(NEL)=1
      IXL(1)=1
      KJP=NEU
      KJ=KJP
C SET VALUES TO 0 FOR PROPER OUTPUT.
      IF(NEU-NEL GT 0)GOTO 960
      IF(NEU-NEL EQ 0)GOTO 970
      NEUUP1=NEUU+1
        DO 980 KI=NEUUP1,NELL
        XU(KI)=0 D0
        YU(KI)=0.D0
        LU(KI)=0 D0
980     CONTINUE
      GOTO 970
960   NELLP1=NELL+1
        DO 990 KI=NELLP1,NEUU
        XL(KI)=0 D0
        YL(KI)=0.D0
        LL(KI)=0.D0
990     CONTINUE
C
C FIND INDICES ON NEW SFCS OF BEGINNING OF NO ACCRETION REGION.
970   IMXU=NEU-NEUP+IMAXU
      IMXL=NEL-NELP+IMAXL
C
C FIT CUBIC SPLINES TO THE SFCS. AND CHECK IF THERE ARE
C   ENOUGH VORTICITY SEGMENTS.
      CALL FIT(BOTH)
C
C FIND RATIO OF LENGTHS ALONG SFC. TO BEGINNING OF NO ACCRETION REGION.
      LRU=LU(IMXU)/LUP(IMAXU)
      LRL=LL(IMXL)/LLP(IMAXL)
C ****************************************************************
      TOLL=0.99D0
C ****************************************************************
C FIND RATIO OF SFC ENDPTS TO CONTROL ENDPTS. IN ACCRETION REGION.
        DO 1200 KI=1,NCOU
        IF(IU(KI) GE.IMAXU)GOTO 1250
1200    CONTINUE
C ****************************************************************
1250  FXU=1.D0-DFLOAT(KI-1)/2.D0/DFLOAT(IMAXU-1)
C ****************************************************************
      IF(BOTH.EQ.1)GOTO 1210
      FXL=FXU
      GOTO 1300
1210    DO 1220 KI=1,NCOL
        IF(IL(KI) GE IMAXL)GOTO 1260
1220    CONTINUE
C ****************************************************************
1260  FXL=1.D0-DFLOAT(KI-1)/2 D0/DFLOAT(IMAXL-1)
```

```
C ******************************************************************
C
C FOR THE UPPER SFC.
 1300  NCOUM1=NCOU-1
       M=O
         DO 820 I=1,NCOUM1
         II=NCOU+1-I
C REDEFINE CEE'S IN NO-ACCRETION REGION.
 834     IF(KJP.LE.IMXU)GOTO 833
         KJ=KJ-1
         IF(NEU-KJ.LT.NEUP-IU(II-1))GOTO 834
         GOTO 880
C DISTANCE BETWEEN CONTROL ENDPTS. ON PREVIOUS SFC.
 833     IF(M.EQ.1)GOTO 830
         SL=DMAX1(LUP(IU(II))-LUP(IU(II-1)),LUM)
         TOLL=DMIN1(LRU,LTOL)*FXU
C ******************************************************************
 831     IF(LU(KJP).GT.2.DO*TOLL*SL)GOTO 830
         IF(LU(KJP).LT.1.2DO*TOLL*SL)GOTO 821
         SL=LU(KJP)/2.DO/DMAX1(TOLL,1.DO)
C ******************************************************************
         M=1
 830     KJ=KJ-1
         IF(KJ.EQ.1)GOTO 880
         IF(LU(KJP)-LU(KJ).LT.TOLL*SL)GOTO 830
C DISTANCE EXCEEDS THAT OF PREVIOUS CONTROL SEGMENT.
 880     IXU(KJ)=1
         IF(KJ.EQ.1)GOTO 821
         IF(KJP-KJ.EQ.1)GOTO 870
         DDD=DSQRT((XU(KJ)-XU(KJP))**2+(YU(KJ)-YU(KJP))**2)
C ******************************************************************
         IF((LU(KJP)-LU(KJ))/DDD.LE.1.3DO)GOTO 870
C ******************************************************************
C CONSIDERABLE SFC  CURVATURE, CREATE NEW CONTROL PT.
C    MIDWAY BETWEEN PREVIOUS ONES.
         JK=KJP-1
 860     IF(LU(KJP)-LU(JK).GT.O.45DO*TOLL*SL)GOTO 850
         JK=JK-1
         GOTO 860
 850     IXU(JK)=1
 870     KJP=KJ
         IF(M.EQ.1)GOTO 821
         IF(II.EQ.2)GOTO 831
 820     CONTINUE
C
C FOR THE LOWER SFC.
 821   KJP=NEL
       KJ=KJP
       NCOLM1=NCOL-1
       TOLL=O.99DO
       M=O
         DO 825 I=1,NCOLM1
         II=NCOL+1-I
C REDEFINE CEE'S IN NO-ACCRETION REGION.
 837     IF(KJP.LE.IMXL)GOTO 838
         KJ=KJ-1
         IF(NEL-KJ.LT.NELP-IL(II-1))GOTO 837
         GOTO 885
C DISTANCE BETWEEN CONTROL ENDPTS. ON PREVIOUS SFC.
 838     IF(M.EQ.1)GOTO 835
         SL=DMAX1(LLP(IL(II-1))-LLP(IL(II)),LLM)
         TOLL=DMIN1(LRL,LTOL)*FXL
C ******************************************************************
 836     IF(-LL(KJP).GT.2.DO*TOLL*SL)GOTO 835
         IF(-LL(KJP).LT.1.2DO*TOLL*SL)GOTO 826
         SL=-LL(KJP)/2.DO/DMAX1(TOLL,1.DO)
```

```
C *****************************************************************
        M=1
  835   KJ=KJ-1
        IF(KJ.EQ.1)GOTO 885
        IF(LL(KJ)-LL(KJP).LT.TOLL*SL)GOTO 835
C DISTANCE EXCEEDS THAT OF PREVIOUS CONTROL SEGMENT.
  885   IXL(KJ)=1
        IF(KJ.EQ.1)GOTO 826
        IF(KJP-KJ.EQ.1)GOTO 875
        DDO=DSQRT((XL(KJ)-XL(KJP))**2+(YL(KJ)-YL(KJP))**2)
C *****************************************************************
        IF((LL(KJ)-LL(KJP))/DDO.LE.1.3DO)GOTO 875
C *****************************************************************
C CONSIDERABLE SFC. CURVATURE. CREATE NEW CONTROL PT.
C   MIDWAY BETWEEN PREVIOUS ONES.
        JK=KJP-1
  865   IF(LL(KJ)-LL(KJP).GT.0.45DO*TOLL*SL)GOTO 855
        JK=JK-1
        GOTO 865
  855   IXL(JK)=1
  875   KJP=KJ
        IF(M.EQ.1)GOTO 826
        IF(II.EQ.2)GOTO 836
  825   CONTINUE
  826  IUU=1
        DO 730 I=1,NEU
        IF(IXU(I).EQ.0)GOTO 730
        IU(IUU)=I
        IUU=IUU+1
  730   CONTINUE
       ILL=1
        DO 740 I=1,NEL
        IF(IXL(I).EQ.0)GOTO 740
        IL(ILL)=I
        ILL=ILL+1
  740   CONTINUE
       NCOU=IUU-1
       NCOL=ILL-1
  720  RETURN
       END

C
C *****************************************************************
C
       DOUBLE PRECISION FUNCTION JTHICK(THETA)
C
C WRITTEN BY: M. OLESKIW  ON:810212  LAST MODIFIED:810315
C
C CALCULATES THE NEGATIVE OF THE THICKNESS OF THE JOUKOWSKI AEROFOIL
C   AS A FN. OF THETA AND E.
C
       DOUBLE PRECISION E,DSIN,DCOS,THETA,A,B,X,Y
C
       COMMON /JOUK1/A,B,E
C
C IN  THETA=ANGLE FROM NEGATIVE X-AXIS.
C
       X=-B*(1.DO+E)*DCOS(THETA)-B*E
       Y=B*(1.DO+E)*DSIN(THETA)
       JTHICK=-2.DO*Y*(1.DO-B*B/(X*X+Y*Y))
       RETURN
       END

C
C *****************************************************************
C
```

```
          DOUBLE PRECISION FUNCTION NSURF(XROT)
     C
     C WRITTEN BY: M. OLESKIW  ON: 800805  LAST MODIFIED: 811024
     C
     C CALCULATES THE UNROTATED X VALUE OF A POINT ON THE ACCRETED AEROFOIL
     C    SURFACE BASED UPON THE COLLISION EFFICIENCY, DIRECTION OF
     C    GROWTH, AND OLD AEROFOIL (ROTATED) SFC. POSITION.
     C
2         DOUBLE PRECISION XUR(101),YUR(101),CU(100,3),XLR(101),YLR(101),
         .CL(100,3),C30,S30,XROT,D,LENG,LEN,LU(101),LL(101),DABS,DCOS,
         .L(31,5),YO(31,5),XLRT,YLRT,XN,YN,DDD,XLRN,YLRN,UINF,PINF,
         .DSIGN,DSQRT,ICE,NSURFY,CE,CEED(5,30,5),RL(101),CLL,KLL,RB,THK,
         .ALPHAR,W(5),RU(101),CUU,KUU,XURT,YURT,XURN,YURN,TINF,EPS(5),
         .CFV(3,30,5),CFA(3,30,5),ACN,VTOT(31,5),ACOL(31,5),VTIMP,UIMP,
         .VIMP,DMIN1,DMAX1,TI,PAR,RHOS,DO(5),C,VT,AC,DSIN,E(2),YOE(2)
3         DOUBLE PRECISION MO,LMXCE(6),LDL,LDR,LMX,FL(5),FLV
     C
4         INTEGER JK,RUN,I,ICT(5),ICU(5),ICL(5),NEU,NEL,NEL1,J,H5D(5),
         .DDISTN,ATHICK,NEU1,DENSE,TYPE,JZ,H5,NEUU,NELL,IJ,II
     C
5         COMMON ALPHAR/FOIL/XUR,YUR,XLR,YLR/SPLINE/CU,CL/ROTP/C30,S30
         ./IND/NSURFY,FL,ICE,I,JK,RUN/LG/LU,LL/COL/L,YO,ICT,ICU,ICL
         ./NOSE/XN,YN/AERO4/NEU,NEL,NEUU,NELL/RC/RU,RL,ATHICK,DDISTN
         ./WTS/W/CEM/LMXCE/CV/VTOT,ACOL/TRANS1/UINF,PINF,TINF,EPS,DENSE
         ./TRANS3/DD,C,TYPE,JZ/HERMT5/H5,H5D/CEV/CFV,CFA/EFF/CEED
     C
     C   IN XROT=ROTATED X POSITION ON LOWER AEROFOIL SFC.
     C
6    10   FORMAT('OOUT OF BOUNDS IN SEARCHING FOR AEROFOIL',
          ' OR CE SPLINES IN NSURF')
     C
7         IF(JK.LT.1)JK=1
8         RUN=RUN+1
9         NEL1=NELL-1
10        NEU1=NEUU-1
11        II=I
12        IF(LMXCE(6).LT.0.D0)GOTO 120
     C
     C FOR THE UPPER SFC:
     C FIND THE APPROPRIATE AEROFOIL SPLINE SEGMENT
13   320  IF(XROT.GT.XUR(JK))GOTO 305
14        JK=JK-1
15        IF(JK.EQ.0)GOTO 600
16        GOTO 320
17   305  IF(XROT.LE.XUR(JK+1))GOTO 310
18        JK=JK+1
19        IF(JK.LE.NEU1)GOTO 305
20        GOTO 600
21   310  D=XROT-XUR(JK)
     C FIND LENGTH ALONG SFC. FROM NOSE TO THIS POINT.
22        CALL SFCLEN(D,LENG,CU(JK,3),CU(JK,2),CU(JK,1))
23        LEN=LU(JK)+LENG
     C ROTATED COORDS.
24        XURT=XROT
25        YURT=YUR(JK)+((CU(JK,3)*D+CU(JK,2))*D+CU(JK,1))*D
     C TANGENT SLOPE.
26        CUU=((3.D0*CU(JK,3)*D+2.D0*CU(JK,2))*D+CU(JK,1))
27        IF(DABS(CUU).LT.1.D-10)GOTO 360
     C PERPENDICULAR SLOPE.
28        KUU=-1.D0/CUU
29        GOTO 330
30   360  KUU=DSIGN(1.D10,-CUU)
     C BLENDED RADIUS OF CURVATURE.
31   330  RB=RU(JK)+(RU(JK+1)-RU(JK))/(LU(JK+1)-LU(JK))*LENG
32        GOTO 230
     C
```

```
C FOR THE LOWER SFC
C FIND THE APPROPRIATE AEROFOIL SPLINE SEGMENT
120   IF(XROT.GT.XLR(JK))GOTO 105
      JK=JK-1
      IF(JK.EQ.0)GOTO 600
      GOTO 120
105   IF(XROT.LE.XLR(JK+1))GOTO 110
      JK=JK+1
      IF(JK.LE.NEL1)GOTO 105
      GOTO 600
110   D=XROT-XLR(JK)
C FIND LENGTH ALONG SFC. FROM NOSE TO THIS POINT.
      CALL SFCLEN(D,LENG,CL(JK,3),CL(JK,2),CL(JK,1))
      LEN=LL(JK)-LENG
C ROTATED COORDS
      XLRT=XROT
      YLRT=YLR(JK)+((CL(JK,3)*D+CL(JK,2))*D+CL(JK,1))*D
C TANGENT SLOPE
      CLL=((3.D0*CL(JK,3)*D+2.D0*CL(JK,2))*D+CL(JK,1))
      IF(DABS(CLL).LT.1.D-10)GOTO 160
C PERPENDICULAR SLOPE
      KLL=-1.D0/CLL
      GOTO 130
160   KLL=DSIGN(1.D10,-CLL)
C BLENDED RADIUS OF CURVATURE
130   RB=RL(JK)-(RL(JK+1)-RL(JK))/(LL(JK+1)-LL(JK))*LENG
C
C DETERMINE THE WEIGHTED COLLISION EFFICIENCY.
230   ACN=0.D0
      VT=0.D0
      AC=0.D0
      MD=0.D0
      RHOS=1.D0
        DO 200 J=1,DDISTN
        LDL=L(1,J)
        LDR=L(ICT(J),J)
        LMX=LMXCE(J)
C FIND THE APPROPRIATE CE VS L SPLINE SEGMENT
        IF(FL(J).NE.0.D0)GOTO 1400
        IF(I.LT.1)I=1
220     IF(LEN.GT.L(I,J))GOTO 205
        I=I-1
        IF(I.EQ.0)GOTO 600
        GOTO 220
205     IF(LEN.LE.L(I+1,J))GOTO 210
        I=I+1
        IF(I.LE.ICT(J))GOTO 205
        GOTO 600
210     DDD=LEN-L(I,J)
        IF(HSD(J).EQ.0)CE=((3.D0*CEED(3,I,J)*DDD+2.D0*CEED(2,I,J))*DDD
     +CEED(1,I,J))*DCOS(ALPHAR)
        IF(HSD(J).EQ.1)CE=((((5.D0*CEED(5,I,J)*DDD+4.D0*CEED(4,I,J))*DDD
     +3.D0*CEED(3,I,J))*DDD+2.D0*CEED(2,I,J))*DDD+CEED(1,I,J))
     *DCOS(ALPHAR)
        GOTO 1390
1400    IF(LEN.GE.LMX)GOTO 1110
        IF(LEN.LE.LDL)GOTO 1120
        FLV=FL(J)-0.9D0*FL(J)/(LMX-LDL)*(LEN-LDL)
        GOTO 1200
1110    IF(LEN.GE.LDR)GOTO 1120
        FLV=0.1D0*FL(J)+0.9D0*FL(J)/(LDR-LMX)*(LEN-LMX)
        GOTO 1200
1120    FLV=FL(J)
1200    E(1)=LEN-FLV
        E(2)=LEN+FLV
          DO 1430 IJ=1,2
```

```
               IF(E(IJ).GT.L(1,J))GOTO 1440
               YOE(IJ)=YO(1,J)
               GOTO 1430
      1440     IF(E(IJ).LT.L(ICT(J),J))GOTO 1450
               YOE(IJ)=YO(ICT(J),J)
               GOTO 1430
      1450     IF(E(IJ).GT.L(II,J))GOTO 1460
               II=II-1
               GOTO 1450
      1460     IF(E(IJ).LE.L(II+1,J))GOTO 1470
               II=II+1
               GOTO 1460
      1470     DDD=E(IJ)-L(II,J)
               IF(H5D(J).EQ.0)YOE(IJ)=((CEED(3,II,J)*DDD
               +CEED(2,II,J))*DDD+CEED(1,II,J))*DDD+YO(II,J)
               IF(H5D(J).EQ.1)YOE(IJ)=((((CEED(5,II,J)*DDD
               +CEED(4,II,J))*DDD+CEED(3,II,J))*DDD
               +CEED(2,II,J))*DDD+CEED(1,II,J))*DDD+YO(II,J)
      1430     CONTINUE
             CE=(YOE(2)-YOE(1))/2.DO/FLV*DCOS(ALPHAR)
      1390   ACN=ACN+CE*W(J)
             IF(DENSE.EQ.0)GOTO 200
             IF(LEN.LE.L(ICT(J),J))GOTO 1385
             IF(J.GT.1)GOTO 200
             VT=VT+W(1)*VTOT(ICT(1),1)
             AC=AC+W(1)*DABS(ACOL(ICT(1),1))
             MD=MD+W(1)*DD(1)
             GOTO 200
      1385   IF(LEN.GE.L(1,J))GOTO 1380
             IF(J.GT.1)GOTO 200
             VT=VT+W(1)*VTOT(1,1)
             AC=AC+W(1)*DABS(ACOL(1,1))
             MD=MD+W(1)*DD(1)
             GOTO 200
      1380   DDD=LEN-L(I,J)
             VT=VT+W(J)*(((CFV(3,I,J)*DDD+CFV(2,I,J))*DDD+CFV(1,I,J))
             *DDD+VTOT(I,J))
             AC=AC+W(J)*DABS(((CFA(3,I,J)*DDD+CFA(2,I,J))*DDD+CFA(1,I,J))
             *DDD+ACOL(I,J))
             MD=MD+W(J)*DD(J)
      200    CONTINUE
            IF(DENSE.EQ.0)GOTO 440
            UIMP=VT*DSIN(AC)
            VIMP=VT*DCOS(AC)
            IF(DENSE.EQ.1)VTIMP=DABS(VIMP)*UINF
            IF(DENSE.EQ.2)VTIMP=DSQRT(UIMP*UIMP+VIMP*VIMP)*UINF
            TI=DMAX1(TINF,-20.DO)
            TI=DMIN1(TI,-5.DO)
            PAR=-MD/2.DO*VTIMP/TI
            PAR=DMAX1(PAR,0.88DO)
            PAR=DMIN1(PAR,16.29DO)
            RHOS=1.1D2*(PAR**0.76DO)/9.17D2
      C CALCULATE THICKNESS FOR ASSUMED FLAT SFC.
      440   THK=ICE*ACN/RHOS
            IF(ATHICK.EQ.0)GOTO 430
      C CALCULATE THICKNESS FOR CURVED SFC.
            THK=-RB+DSIGN(DSQRT(RB*RB+2.DO*RB*THK),RB)
      C NEW SFC. COORDS
      430   IF(LMXCE(6).LT.0.DO)GOTO 420
            XURN=XURT+DSIGN(DSQRT(THK*THK/(1.DO+KUU*KUU)),KUU)
            YURN=YURT+KUU*(XURN-XURT)
            NSURF=XURN*C30-YURN*S30+XN
            NSURFY=XURN*S30+YURN*C30+YN
            RETURN
      420   XLRN=XLRT-DSIGN(DSQRT(THK*THK/(1.DO+KLL*KLL)),KLL)
            YLRN=YLRT+KLL*(XLRN-XLRT)
```

```
      NSURF=XLRN*C30+YLRN*S30+XN
      NSURFY=-XLRN*S30+YLRN*C30+YN
      RETURN
600   WRITE(6,10)
      WRITE(7,10)
      RETURN
      END
C
C *****************************************************************
C
      SUBROUTINE PC4(EQN,CDS,LAMBH,WARN)
C
C WRITTEN BY: M. OLESKIW  ON: 800122 LAST MODIFIED: 810626
C
C INTEGRATE EQNS. OF MOTION USING THE 4TH ORDER PREDICTOR-
C   CORRECTOR METHOD OF HAMMING.
C REF: BURDEN, R.L., J.D. FAIRES, & A.C. REYNOLDS (1978),
C       NUMERICAL ANALYSIS   QA 297.B84   P.266
C       HAMMING, R.W. (1973), NUMERICAL METHODS FOR SCIENTISTS &
C       ENGINEERS, 2ND ED.,   QA 297 H28   CHAPS. 22 & 23
C
      DOUBLE PRECISION XDS(6,2),UDS(6,2),AN(2,6,2),HT(2,6,2),YDS(6,2),
     .VDS(6,2),AO,A1,A2,BO,B1,B2,B3,LAMBH,EIGMX,DTSO,
     .CO,C1,C2,DM1,DO,D1,D2,UPI,UCI,VPI,VCI,MUAS,MVAS,
     .PUDS,DTS(6,2),PVDS,MUDS,MVDS,CUDS,CVDS,UDSP1,VDSP1
     .,FMU,FMV,UST,VST,ER1,ER2,PXDS,PYDS,MXDS,MYDS,CXDS,CYDS
     .,UAS(6,2),VAS(6,2),RED(6,2),XPI,XCI,YPI,YCI,RE,CD,TS(500,2)
C
      INTEGER I(2),EQN,IM4(2),IM3(2),IM2(2),IM1(2),IO(2),IP1(2),
     .CDS,WARN,MM
C
      COMMON/INTEG/AN,HT/PV/XDS,YDS,UDS,VDS
     ./PCM/AO,A1,A2,BO,B1,B2,B3,CO,C1,C2,DM1,DO,D1,D2,
     .UPI,UCI,VPI,VCI,ER1,ER2,XPI,XCI,YPI,YCI,UST,VST
     ./LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM
     ./REL/UAS,VAS,RED,CD ,
C
C IN   EQN=DENOTES PORTION OF TOTAL SYSTEM OF EQUATIONS TO BE SOLVED.
C IN   CDS=TYPE OF DRAG COEFFICIENT TO BE USED.
C OUT  LAMBH=STABILITY PARAMETER.
C OUT  WARN=WARNING OF INSTABILITY (O OR 1).
C
      DTSO=DTS(IO(MM),MM)
      TS(I(MM)+1,MM)=TS(I(MM),MM)+DTSO
C
C THE PREDICTOR
      PXDS=AO*XDS(IO(MM),MM)+A1*XDS(IM1(MM),MM)+A2*XDS(IM2(MM),MM)
     .+DTSO*(BO*UDS(IO(MM),MM)+B1*UDS(IM1(MM),MM)+
     .B2*UDS(IM2(MM),MM)+B3*UDS(IM3(MM),MM))
      PYDS=AO*YDS(IO(MM),MM)+A1*YDS(IM1(MM),MM)+A2*YDS(IM2(MM),MM)
     .+DTSO*(BO*VDS(IO(MM),MM)+B1*VDS(IM1(MM),MM)+
     .B2*VDS(IM2(MM),MM)+B3*VDS(IM3(MM),MM))
      PUDS=AO*UDS(IO(MM),MM)+A1*UDS(IM1(MM),MM)+A2*UDS(IM2(MM),MM)
     .+DTSO*(BO*AN(1,IO(MM),MM)+B1*AN(1,IM1(MM),MM)+
     .B2*AN(1,IM2(MM),MM)+B3*AN(1,IM3(MM),MM))
      PVDS=AO*VDS(IO(MM),MM)+A1*VDS(IM1(MM),MM)+A2*VDS(IM2(MM),MM)
     .+DTSO*(BO*AN(2,IO(MM),MM)+B1*AN(2,IM1(MM),MM)+
     .B2*AN(2,IM2(MM),MM)+B3*AN(2,IM3(MM),MM))
C
C MODIFICATION OF THE PREDICTOR
      MXDS=PXDS-ER1*(XPI-XCI)
      MYDS=PYDS-ER1*(YPI-YCI)
      MUDS=PUDS-ER1*(UPI-UCI)
      MVDS=PVDS-ER1*(VPI-VCI)
      CALL AIRVEL(MXDS,MYDS,MUAS,MVAS,4)
```

```
          CALL DRAG(MUDS,MVDS,MUAS,MVAS,CDS,RE,CD)
          CALL ACCN(MUDS,MVDS,MUAS,MVAS,RE,CD,EQN,TS(I(MM)+1,MM),O)
          FMU=AN(1,IP1(MM),MM)
          FMV=AN(2,IP1(MM),MM)
C
C THE CORRECTOR
          CXDS=CO*XDS(IO(MM),MM)+C1*XDS(IM1(MM),MM)+C2*XDS(IM2(MM),MM)
         .+DTSO*(DM1*MUDS+DO*UDS(IO(MM),MM)+D1*UDS(IM1(MM),MM)+D2*
         .UDS(IM2(MM),MM))
          CYDS=CO*YDS(IO(MM),MM)+C1*YDS(IM1(MM),MM)+C2*YDS(IM2(MM),MM)
         .+DTSO*(DM1*MVDS+DO*VDS(IO(MM),MM)+D1*VDS(IM1(MM),MM)+D2*
         .VDS(IM2(MM),MM))
          CUDS=CO*UDS(IO(MM),MM)+C1*UDS(IM1(MM),MM)+C2*UDS(IM2(MM),MM)
         .+DTSO*(DM1*FMU+DO*AN(1,IO(MM),MM)+D1*AN(1,IM1(MM),MM)+
         .D2*AN(1,IM2(MM),MM))
          CVDS=CO*VDS(IO(MM),MM)+C1*VDS(IM1(MM),MM)+C2*VDS(IM2(MM),MM)
         .+DTSO*(DM1*FMV+DO*AN(2,IO(MM),MM)+D1*AN(2,IM1(MM),MM)+
         .D2*AN(2,IM2(MM),MM))
C
C FINAL VALUES
          XDS(IP1(MM),MM)=CXDS+ER2*(PXDS-CXDS)
          YDS(IP1(MM),MM)=CYDS+ER2*(PYDS-CYDS)
          UDS(IP1(MM),MM)=CUDS+ER2*(PUDS-CUDS)
          VDS(IP1(MM),MM)=CVDS+ER2*(PVDS-CVDS)
C NEW VALUES FOR ACCELERATION AT I+1
          CALL AIRVEL(XDS(IP1(MM),MM),YDS(IP1(MM),MM),UAS(IP1(MM),MM),
          VAS(IP1(MM),MM),13)
          CALL DRAG(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
         .VAS(IP1(MM),MM),CDS,RED(IP1(MM),MM),CD)
          CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
         .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EQN,
         .TS(I(MM)+1,MM),O)
          CALL STAB(RED(IP1(MM),MM),CD,UDS(IP1(MM),MM),VDS(IP1(MM),MM),
         .UAS(IP1(MM),MM),VAS(IP1(MM),MM),CDS,EIGMX)
          LAMBH=EIGMX*DTSO
          IF(LAMBH.LT.-1.3DO)WARN=1
          UDSP1=AN(1,IP1(MM),MM)
          VDSP1=AN(2,IP1(MM),MM)
          IF(EQN.NE.2)GOTO 100
          CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
         .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EQN,
         .TS(I(MM)+1,MM),1)
C
C CALCULATE STABILITY INDICES
  100     UST=(FMU-UDSP1)/(MUDS-UDS(IP1(MM),MM))
          VST=(FMV-VDSP1)/(MVDS-VDS(IP1(MM),MM))
          XPI=PXDS
          XCI=CXDS
          YPI=PYDS
          YCI=CYDS
          UPI=PUDS
          UCI=CUDS
          VPI=PVDS
          VCI=CVDS
          DTS(IP1(MM),MM)=DTSO
          RETURN
          END

C
C ********************************************************************
C
          DOUBLE PRECISION FUNCTION PJK(X,Y)
C
C WRITTEN BY: M. OLESKIW  ON: 801001  LAST MODIFIED: 810728
C
C CALCULATES ANALYTICAL VALUE OF STREAMFN. AT TRANSFORMED COORDS X,Y
```

```
C    USING THE EXACT AEROFOIL GENERATION METHOD.
C
      DOUBLE PRECISION ALPHAR,A,B,E,XI,ETA,X,Y,G,H,J,
     .DSQRT,XX,DSIGN,YY,T2,T1,T3,DSIN,DLOG,DCOS,DABS
C
      COMMON ALPHAR/JOUK1/A,B,E
C
C IN   X=
C IN   Y=COORDS. IN TRANSFORMED COORDINATE SYSTEM AT WHICH PSI IS
C        TO BE FOUND
C
      XI=X-(1.D0+2.D0*E+2.D0*E*E)/2.D0/(1.D0+2.D0*E+E*E)
      ETA=Y
      G=XI*XI-ETA*ETA-4.D0*B*B
      H=2.D0*XI*ETA
      J=DSQRT(G*G+H*H)
      IF(J+G.GE.0.D0)GOTO 100
      XX=XI/2.D0
      GOTO 200
100   XX=(XI+DSIGN(DSQRT((J+G)/2.D0),XI))/2.D0
200   IF(DABS(Y).GT.1.D-60)GOTO 210
      YY=0.D0
      GOTO 220
210   YY=(ETA+DSIGN(DSQRT((J-G)/2.D0),ETA))/2.D0
220   T1=YY*DCOS(ALPHAR)-(XX+B*E)*DSIN(ALPHAR)
      T2=A*A*T1/((XX+B*E)**2+YY*YY)
      T3=2.D0*A*DSIN(ALPHAR)*DLOG(DSQRT((XX+B*E)**2+YY*YY)/A)
      PJK=T1-T2+T3
      RETURN
      END
C
C ******************************************************************
C
      SUBROUTINE PLTSZ(XMIN,XMAX,YMIN,YMAX,XL,YB,PX,PY,IRX,IRY,
     .NDCPX,NDCPY)
C
C WRITTEN BY: M. OLESKIW   ON:800627   LAST MODIFIED:810420
C
C DETERMINE PARAMETERS NECESSARY FOR SCALING OF A PLOT AND ITS AXES
C
      REAL XPAR(4,24),YPAR(4,24),XD,FLOAT,AINT,XMIN,XMAX,
     .XL,YD,YMIN,YMAX,YB,DX,DY,XR,YT,DDX,DDY,ABS
C
      INTEGER PX,PNX,PY,PNY,I,J,IX,IRX,INT,IY,IRY,IFIX,NDCPX,NDCPY
C
      COMMON/PLTPRM/XPAR,YPAR
C
C IN   XMIN=
C IN   XMAX=
C IN   YMIN=
C IN   YMAX=
C OUT  XL=LEFT EDGE OF PLOT
C OUT  YB=BOTTOM EDGE OF PLOT
C OUT  PX=POWER OF TEN IN X-AXIS RANGE
C OUT  PY=POWER OF TEN IN Y-AXIS RANGE
C OUT  IRX=MIN. LENGTH OF X AXIS.
C OUT  IRY=MIN. LENGTH OF Y AXIS.
C OUT  NDCPX=NO. OF DECIMAL PLACES IN X-AXIS SCALES.
C OUT  NDCPY=NO. OF DECIMAL PLACES IN Y-AXIS SCALES.
C
10    FORMAT(8F10.0)
C
C READ IN PLOTTING PARAMETERS
        DO 101 I=3,24
        READ(3,10)(XPAR(J,I),J=1,4),(YPAR(J,I),J=1,4)
```

```
101      CONTINUE
C
         ENTRY PLTSZE(XMIN,XMAX,YMIN,YMAX,XL,YB,PX,PY,IRX,IRY,
        .NDCPX,NDCPY)
         PNX=0
         PNY=0
C
C DETERMINE THE PLOTTING RANGE OF THE X VARIABLE
100      PX=PNX
         XD=(XMAX-XMIN)*10.0**PX
         IF(XD.GT.22.0)PNX=PNX-1
         IF(XD.LT.2.20001)PNX=PNX+1
         IF(PNX.NE.PX)GOTO 100
C PX GIVES 1/(POWER OF TEN) OF THE X VARIABLE PLOTTING RANGE
         IX=1
120      IRX=INT(XD)+IX
         IF(IRX.NE.7.AND.IRX.NE.9.AND.IRX.NE.11.AND.IRX.LT.13)GOTO 140
         IF(IRX.EQ.16.OR.IRX.EQ.20.OR.IRX.EQ.24)GOTO 140
         IX=IX+1
         GOTO 120
140      DX=FLOAT(IRX)/10.0**PX/XPAR(1,IRX)
C SET THE X VALUE AT THE LEFT GRAPH EDGE
         IF(XMIN.LT.0.0)XL=AINT(XMIN/DX-1.0)*DX
         IF(XMIN.GE.0.0)XL=AINT(XMIN/DX)*DX
C SET X VALUE AT RIGHT GRAPH EDGE
         XR=XL+XPAR(1,IRX)*DX
         IF(XR.GE.XMAX)GOTO 105
         IX=IX+1
         GOTO 120
C DETERMINE CORRECT NUMBER OF DECIMAL PLACES ON AXIS SCALES.
105      NDCPX=0
         DDX=ABS(DX)+1.E-6
160      IF(DDX-AINT(DDX).LT.2./10.**(6-NDCPX))GOTO 150
         NDCPX=NDCPX+1
         DDX=DDX*10.0
         GOTO 160
150      IF(IFIX((XR-XMAX)/DX).LE.IFIX((XMIN-XL)/DX))GOTO 110
C CENTRE THE PLOT
         XL=XL-DX
         XR=XR-DX
         GOTO 150
C
C DETERMINE THE PLOTTING RANGE OF THE Y VARIABLE
110      PY=PNY
         YD=(YMAX-YMIN)*10.0**PY
         IF(YD.GT.22.0)PNY=PNY-1
         IF(YD.LT.2.20001)PNY=PNY+1
         IF(PNY.NE.PY)GOTO 110
C PY GIVES 1/(POWER OF TEN) OF THE Y VARIABLE PLOTTING RANGE
         IY=1
130      IRY=INT(YD)+IY
         IF(IRY.NE.13.AND.IRY.NE.15.AND.IRY.NE.17
        ..AND.IRY.NE.19.AND.IRY.NE.21.AND.IRY.NE.23)GOTO 170
         IY=IY+1
         GOTO 130
170      DY=FLOAT(IRY)/10.0**PY/YPAR(1,IRY)
C SET THE Y VALUE AT THE BOTTOM OF THE GRAPH
         IF(YMIN.LT.0.0)YB=AINT(YMIN/DY-1.0)*DY
         IF(YMIN.GE.0.0)YB=AINT(YMIN/DY)*DY
C SET THE Y VALUE AT THE TOP OF THE GRAPH.
         YT=YB+YPAR(1,IRY)*DY
         IF(YT.GE.YMAX)GOTO 135
         IY=IY+1
         GOTO 130
C DETERMINE CORRECT NUMBER OF DECIMAL PLACES ON AXIS SCALES.
135      NDCPY=0
```

```
      DDY=ABS(DY)+1.E-6
  190 IF(DDY-AINT(DDY).LT.2./10.**(6-NDCPY))GOTO 180
      NDCPY=NDCPY+1
      DDY=DDY*10.0
      GOTO 190
  180 IF(IFIX((YT-YMAX)/DY).LE.IFIX((YMIN-YB)/DY))RETURN
C CENTRE THE PLOT.
      YB=YB-DY
      YT=YT-DY
      GOTO 180
      END

C
C ********************************************************************
C
      SUBROUTINE POT1
C
C WRITTEN BY: M. OLESKIW ON:781129 LAST MODIFIED:810726
C
C SOLVE FOR SURFACE VORTEX DENSITY ON 1 ELEMENT AEROFOIL IN POTENTIAL
C FLOW, GIVEN COORDS. OF AEROFOIL SURFACE.
C REF: KENNEDY, J.L. & D.J. MARSDEN (1976), CAN. AERO. & SPACE JOUR.,
C      V22, #5, P243-256
C ADAPTED FROM KENNEDY'S PROGRAM IN SCSS:LIB
C SUBROUTINE: LEQT1F OF *IMSLDPLIB: LINEAR EQN. SOLN., FULL STORAGE
C MODE, SPACE ECONOMIZER SOLN.
C
      DOUBLE PRECISION XE(101),YE(101),XC(101),YC(101),R(101),
     .DATAN,DABS,DSIGN,DLOG,SI(100),CO(100),PI,CL,
     .K(101,101),WKAREA(101),D(100),XT,YT,DE,DELTA,
     .DXC,DYC,B,A,R1S,R2S,R3S,T3,T1,T2,ALPHAR,DCOS,DSIN,DSQRT
C
      INTEGER N,N1,J,J1,IDGT,IER,I,NCOU,NCOU1,NCOL,JJ
C
      COMMON ALPHAR,PI/AERO1/XE,YE/AERO3/NCOU,NCOL/AERO2/XC,YC,R,D,SI,CO
C
   10 FORMAT('OFOR EQN. SOLN. IER=',I3)
   15 FORMAT('OTHE POTENTIAL FLOW LIFT COEFFICIENT IS',F9.5)
   16 FORMAT('1THE POTENTIAL FLOW LIFT COEFFICIENT IS',F9.5)
   20 FORMAT('-CONTROL PT.   X COORD.   Y COORD.   SFC. AIR VEL.')
   30 FORMAT(' ',I6,5X,2F10.5,F11.5)
C
      NCOU1=NCOU-1
      N=NCOU1+NCOL-1
      N1=N+1
C
C CALC. ELEMENT LENGTHS (D) AND CONTROL POINTS (XC,YC)
C XE(1)=XE(2*NCO-1)=XE(N1)=LEADING PT. X COORD.
        DO 110 J=1,N
        J1=J+1
        XC(J)=(XE(J)+XE(J1))*0.5D0
        YC(J)=(YE(J)+YE(J1))*0.5D0
        D(J)=DSQRT((XE(J1)-XE(J))**2+(YE(J1)-YE(J))**2)
  110   CONTINUE
C FIND TRAILING POINT COORDS. XC(N1),YC(N1): FIG.5
      XT=XE(NCOU)-(XC(NCOU1)+XC(NCOU))*0.5D0
      YT=YE(NCOU)-(YC(NCOU1)+YC(NCOU))*0.5D0
      XC(N1)=XE(NCOU)+1.D-2*XT
      YC(N1)=YE(NCOU)+1.D-2*YT
C FORM MATRICES K AND R: EQNS. 9 & 10
C DO FOR EACH SFC. ELEMENT J (COLUMN OF K) AND ROW OF R
        DO 120 J=1,N1
        R(J)=YC(J)*DCOS(ALPHAR)-XC(J)*DSIN(ALPHAR)
        IF(J.EQ.N1)GO TO 140
        J1=J+1
        DE=D(J)
```

```
C CALCULATE ANGLE OF ELEMENT TO X-AXIS.
         CO(J)=(XE(J1)-XE(J))/DE
         SI(J)=(YE(J1)-YE(J))/DE
         DELTA=DE/2.D0
 140       DO 130 I=1,N1
           IF(J.EQ.N1)GO TO 150
C FIND DISTANCE BETWEEN CONTROL PTS. I AND J.
           DXC=XC(I)-XC(J)
           DYC=YC(I)-YC(J)
C CALCULATE COMPONENTS OF EQN  9 AND FIG 2
           B=DXC*CO(J)+DYC*SI(J)
           A=DYC*CO(J)-DXC*SI(J)
           R1S=A*A+(B+DELTA)*(B+DELTA)
           R2S=A*A+(B-DELTA)*(B-DELTA)
           R3S=A*A+B*B-DELTA*DELTA
           IF(R3S.LT.1.D-30)GO TO 160
           T3=DATAN(2.D0*A*DELTA/R3S)
           GO TO 170
 160       IF(DABS(A).LT.1.D-30)GO TO 180
           T3=DATAN((B+DELTA)/A)-DATAN((B-DELTA)/A)
           GO TO 170
 180       T3=DSIGN(PI,A)
 170       T1=(B+DELTA)*DLOG(R1S)
           T2=(B-DELTA)*DLOG(R2S)
           K(I,J)=(T1-T2+2.D0*A*T3-4.D0*DELTA)/4.D0/PI
           GO TO 130
C FOR LAST COLUMN OF K
 150       K(I,J)=1.D0
 130       CONTINUE
 120     CONTINUE
C ***********************************************************************
       IDGT=8
C ***********************************************************************
       CALL LEQT1F(K,1,N1,101,R,IDGT,WKAREA,IER)
C ON OUTPUT, THE SOLN. IS IN R
C
C CALCULATE THE LIFT COEFFICIENT.
       CL=0.D0
         DO 200 JJ=1,N
         CL=CL-2.D0*R(JJ)*D(JJ)
 200     CONTINUE
       WRITE(6,10) IER
       WRITE(6,15)CL
       WRITE(7,16) CL
       WRITE(7,20)
C OUTPUT AEROFOIL COORDS. AND SFC. VELOCITY.
         DO 210 JJ=1,N1
         WRITE(7,30)JJ,XC(JJ),YC(JJ),R(JJ)
 210     CONTINUE
       RETURN
       END
C
C ***********************************************************************
C
       SUBROUTINE RK4(EQN,CDS,LAMBH,WARN)
C
C WRITTEN BY  M. OLESKIW  ON: 790926  LAST MODIFIED: 810703
C
C INTEGRATE THE DROPLET EQNS. OF MOTION (IN X AND Y) USING THE 4TH
C   ORDER RUNGE-KUTTA TECHNIQUE.
C REF: BURDEN,R.L., J.D  FAIRES, & A.C. REYNOLDS (1978), NUMERICAL
C      ANALYSIS  P. 281   QA 297.B84
C
       DOUBLE PRECISION K1,L1,K2,L2,K3,L3,K4,L4,DTS(6,2),XDS(6,2),
      .UDS(6,2),YDS(6,2),VDS(6,2),AN(2,6,2),HT(2,6,2),EIGMX,LAMBH,
```

```
     .M1,M2,M3,M4,N1,N2,N3,N4,U1,U2,U3,V1,V2,V3,CD,RE,RED(6,2),
     .VAS(6,2),UAS(6,2),TS(500,2),DTSO,TSO,XDSO,YDSO,UDSO,VDSO
C
      INTEGER I(2),EQN,IM4(2),IM3(2),IM2(2),IM1(2),IO(2),IP1(2),
     .MM,CDS,WARN
C
      COMMON /INTEG/AN,HT/PV/XDS,YDS,UDS,VDS
     ./LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM
     ./REL/UAS,VAS,RED,CD
C
C IN  EQN=DENOTES PORTION OF TOTAL SYSTEM OF EQUATIONS TO BE SOLVED.
C IN  CDS=TYPE OF DRAG COEFFICIENT TO BE USED.
C OUT LAMBH=STABILITY PARAMETER.
C OUT WARN=WARNING OF INSTABILITY (0 OR 1).
C
      TSO=TS(I(MM),MM)
      DTSO=DTS(IO(MM),MM)
      XDSO=XDS(IO(MM),MM)
      YDSO=YDS(IO(MM),MM)
      UDSO=UDS(IO(MM),MM)
      VDSO=VDS(IO(MM),MM)
      TS(I(MM)+1,MM)=TSO+DTSO
      K1=DTSO*UDSO
      L1=DTSO*VDSO
      M1=DTSO*AN(1,IO(MM),MM)
      N1=DTSO*AN(2,IO(MM),MM)
      CALL AIRVEL(XDSO+K1/2.D0,YDSO+L1/2.D0,U1,V1,4)
      CALL DRAG(UDSO+M1/2.D0,VDSO+N1/2.D0,U1,V1,CDS,RE,CD)
C
      K2=DTSO*(UDSO+M1/2.D0)
      L2=DTSO*(VDSO+N1/2.D0)
      CALL ACCN(UDSO+M1/2.D0,VDSO+N1/2.D0,U1,V1,RE,CD,EQN,
     .TSO,0)
      M2=DTSO*AN(1,IP1(MM),MM)
      N2=DTSO*AN(2,IP1(MM),MM)
      CALL AIRVEL(XDSO+K2/2.D0,YDSO+L2/2.D0,U2,V2,4)
      CALL DRAG(UDSO+M1/2.D0,VDSO+N1/2.D0,U2,V2,CDS,RE,CD)
C
      K3=DTSO*(UDSO+M2/2.D0)
      L3=DTSO*(VDSO+N2/2.D0)
      CALL ACCN(UDSO+M2/2.D0,VDSO+N2/2.D0,U2,V2,RE,CD,EQN,
     .TSO+DTSO/2.D0,0)
      M3=DTSO*AN(1,IP1(MM),MM)
      N3=DTSO*AN(2,IP1(MM),MM)
      CALL AIRVEL(XDSO+K3,YDSO+L3,U3,V3,4)
      CALL DRAG(UDSO+M3,VDSO+N3,U3,V3,CDS,RE,CD)
C
      K4=DTSO*(UDSO+M3)
      L4=DTSO*(VDSO+N3)
      CALL ACCN(UDSO+M3,VDSO+N3,U3,V3,RE,CD,EQN,
     .TSO+DTSO/2.D0,0)
      M4=DTSO*AN(1,IP1(MM),MM)
      N4=DTSO*AN(2,IP1(MM),MM)
C
C NEW DROPLET POSITION AT I+1
      XDS(IP1(MM),MM)=XDSO+(K1+2.D0*K2+2.D0*K3+K4)/6.D0
      YDS(IP1(MM),MM)=YDSO+(L1+2.D0*L2+2.D0*L3+L4)/6.D0
C NEW VELOCITIES AT I+1
      UDS(IP1(MM),MM)=UDSO+(M1+2.D0*M2+2.D0*M3+M4)/6.D0
      VDS(IP1(MM),MM)=VDSO+(N1+2.D0*N2+2.D0*N3+N4)/6.D0
C NEW ACCELERATIONS AT I+1
      CALL AIRVEL(XDS(IP1(MM),MM),YDS(IP1(MM),MM),
     .UAS(IP1(MM),MM),VAS(IP1(MM),MM),13)
      CALL DRAG(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
     .VAS(IP1(MM),MM),CDS,RED(IP1(MM),MM),CD)
      CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
```

```
     .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EQN,
     .TS(I(MM)+1,MM),0)
      CALL STAB(RED(IP1(MM),MM),CD,UDS(IP1(MM),MM),VDS(IP1(MM),MM),
     .UAS(IP1(MM),MM),VAS(IP1(MM),MM),CDS,EIGMX)
      LAMBH=EIGMX*DTSO
      IF(LAMBH.LT.-2.7DO)WARN=1
      DTS(IP1(MM),MM)=DTSO
      IF(EQN.NE.2)RETURN
C
      CALL ACCN(UDS(IP1(MM),MM),VDS(IP1(MM),MM),UAS(IP1(MM),MM),
     .VAS(IP1(MM),MM),RED(IP1(MM),MM),CD,EQN,TS(I(MM)+1,MM),1)
      RETURN
      END
C
C ****************************************************************
C
      SUBROUTINE SFC(X,Y,S,L,LEN)
C
C WRITTEN BY: M. OLESKIW  ON:800623  LAST MODIFIED:811018
C
C CALCULATES Y VALUES AND THE LENGTH FROM THE NOSE
C   ON THE SFC. OF THE AEROFOIL BY A CUBIC SPLINE INTERPOLATION
C
      DOUBLE PRECISION XN,YN,XUR(101),YUR(101),CU(100,3),CL(100,3),
     .XLR(101),YLR(101),XB,DELTA,DELTAP,DABS,DSIGN,DATAN,DATAN2,
     .S30,C30,XR,YR,X,Y,LU(101),LL(101),LEN,LENG,D,AS,RS,ALPHAR,PI,
     .XU(101),YU(101),XL(101),YL(101),YG,DSQRT,DFLOAT
C
      INTEGER S,L,JU,JL,NEU1,NEU,NEL1,NEL,IU,IL,NEUU,NELL,NEUU1,NELL1
C
      COMMON ALPHAR,PI/NOSE/XN,YN/LG/LU,LL/FOIL/XUR,YUR,XLR,YLR
     ./ROTP/C30,S30/AERO4/NEU,NEL,NEUU,NELL
     ./SRCH/D,IU,IL/SA/AS/SPLINE/CU,CL/SFCS/XU,YU,XL,YL
C
C IN  X=POINT AT WHICH Y VALUE IS TO BE CALCULATED
C OUT Y=SFC. POSITION ON SPLINE
C IN  S=0:LOWER SFC
C       1:UPPER SFC.
C IN  L=1:FIND LENGTH ALONG AEROFOIL SFC. FROM NOSE TO (X,Y)
C OUT LEN=LENGTH ALONG AEROFOIL SFC. FROM NOSE TO (X,Y)
C
   10 FORMAT('0OUT OF BOUNDS ON SEARCHING FOR SFC. POSITION ',
     .'IN ROUTINE SFC')
C
      JU=1
      JL=1
C ROTATED X COORD.
      IF(S.LE.0)GOTO 150
C
C FOR THE UPPER SFC.
      NEUU1=NEUU-1
      NEU1=NEU-1
      IF(X.GT.XN)GOTO 121
      IF(X.LT.XN)GOTO 600
      Y=YN
      LEN=0.DO
      AS=PI/2.DO
      RETURN
C
C FIND THE APPROPRIATE SPLINE SEGMENT
  121 IF(X.GT.XU(IU))GOTO 106
      IU=IU-1
      IF(IU.EQ.0)GOTO 600
      GOTO 121
  106 IF(X.LE.XU(IU+1))GOTO 111
```

```
      IU=IU+1
      IF(IU.LE.NEU1)GOTO 106
      GOTO 600
111   IF(IU.GT.NEUU1)GOTO 700
      YG=YU(IU)+(YU(IU+1)-YU(IU))/(XU(IU+1)-XU(IU))
     *(X-XU(IU))
      XR=(X-XN)*C30+(YG-YN)*S30
120   IF(XR.GT.XUR(IU))GOTO 105
      IU=IU-1
      IF(IU.EQ.0)GOTO 600
      GOTO 120
105   IF(XR.LE.XUR(IU+1))GOTO 110
      IU=IU+1
      IF(IU.LE.NEU1)GOTO 105
      GOTO 600
110   D=XR-XUR(IU)
C ROTATED Y COORD
      YR=((CU(IU,3)*D+CU(IU,2))*D+CU(IU,1))*D+YUR(IU)
C TANGENT LINE SLOPE
      RS=(3.DO*CU(IU,3)*D+2.DO*CU(IU,2))*D+CU(IU,1)
      IF(DABS(RS).GT.1.D20)RS=DSIGN(1.D20,RS)
      XB=XR*C30-YR*S30+XN
      DELTA=X-XB
C ***************************************************************
      IF(DABS(DELTA).LE.1.D-10)GOTO 400
C ***************************************************************
C USE NEWTON-RAPHSON METHOD TO CONVERGE TO CORRECT XR,YR.
      DELTAP=-C30+S30*RS
      XR=XR-DELTA/DELTAP
      IF(XR.LE.0.DO)XR=1.D-7*DFLOAT(JU)
      JU=JU+1
      GOTO 120
C
C UNROTATED Y COORD
400   Y=YR*C30+YN+XR*S30
C ANGLE OF TANGENT LINE FROM X AXIS
      AS=DATAN(RS)+DATAN2(S30,C30)
      IF(L.EQ.0)RETURN
      CALL SFCLEN(D,LENG,CU(IU,3),CU(IU,2),CU(IU,1))
      LEN=LU(IU)+LENG
      RETURN
C
700   Y=DSQRT(0.25DO-(X-0.5DO)**2)
      LEN=0.DO
      RETURN
C
C FOR THE LOWER SFC.
150   NEL1=NEL-1
      NELL1=NELL-1
C FIND THE APPROPRIATE SFC. SPLINE SEGMENT
221   IF(X.GT.XL(IL))GOTO 206
      IL=IL-1
      IF(IL.EQ.0)GOTO 600
      GOTO 221
206   IF(X.LE.XL(IL+1))GOTO 211
      IL=IL+1
      IF(IL.LE.NEL1)GOTO 206
      GOTO 600
211   IF(IL.GT.NELL1)GOTO 800
      YG=YL(IL)+(YL(IL+1)-YL(IL))/(XL(IL+1)-XL(IL))
     *(X-XL(IL))
      XR=(X-XN)*C30-(YG-YN)*S30
220   IF(XR.GT.XLR(IL))GOTO 205
      IL=IL-1
      IF(IL.EQ.0)GOTO 600
      GOTO 220
```

```
 205    IF(XR.LE.XLR(IL+1))GOTO 210
        IL=IL+1
        IF(IL.LE.NEL1)GOTO 205
        GOTO 600
 210    D=XR-XLR(IL)
C ROTATED Y COORD.
        YR=((CL(IL,3)*D+CL(IL,2))*D+CL(IL,1))*D+YLR(IL)
C TANGENT LINE SLOPE
        RS=(3.DO*CL(IL,3)*D+2.DO*CL(IL,2))*D+CL(IL,1)
        IF(DABS(RS).GT.1.D20)RS=DSIGN(1.D20,RS)
        XB=XR*C30+YR*S30+XN
        DELTA=X-XB
C ****************************************************************
        IF(DABS(DELTA).LE.1.D-10)GOTO 500
C ****************************************************************
C USE NEWTON-RAPHSON METHOD TO CONVERGE TO CORRECT XR,YR.
        DELTAP=-C30-S30*RS
       .XR=XR-DELTA/DELTAP
        IF(XR.LE.O.DO)XR=1.D-7*DFLOAT(JL)
        JL=JL+1
        GOTO 220
C
C UNROTATED Y COORD.
 500    Y=-XR*S30+YR*C30+YN
C ANGLE OF TANGENT LINE FROM X AXIS.
        AS=DATAN(RS)-DATAN2(S30,C30)
        IF(L.EQ.O)RETURN
        CALL SFCLEN(D,LENG,CL(IL,3),CL(IL,2),CL(IL,1))
        LEN=LL(IL)-LENG
        RETURN
C
 800    Y=-DSQRT(O.25DO-(X-O.5DO)**2)
        LEN=O.DO
        RETURN
C
 600    WRITE(6,10)
        WRITE(7,10)
        RETURN
        END

C
C ****************************************************************
C
        SUBROUTINE SFCLEN(D,L,A,B,C)
C
C WRITTEN BY: M. OLESKIW  ON:800525  LAST MODIFIED:800802
C
C CALCULATES THE LENGTH ALONG A SEGMENT OF THE CUBIC SPLINE FIT OF THE
C   AEROFOIL SFC.
C
C REF:DOUG S. PHILLIPS (1980)
C
        DOUBLE PRECISION II,NU,E,F,DSQRT,DELTA,Q,A,B,C,D,L,
       .T1,T2,T3,T4,NU1,ANU1,DABS,NUO,ANUO,K,E2,F2,E3,F3,EO2,FO2,
       .EO3,FO3,XO,X1,CK,FO,EO,F1,E1,YP,DISTP,DIST,DFLOAT,Y,
       .DLOG,DSIGN
C
        INTEGER IER,I,ANAL
C
        COMMON /LA/ANAL
C
C IN  D=ROTATED X COORDINATE OF POINT FROM BEGINNING OF SEGMENT
C         OF INTEREST TO WHICH THE LENGTH IS TO BE FOUND.
C OUT L=SEGMENT LENGTH
C IN  A=
C IN  B=
```

```
C IN  C= SPLINE PARAMETERS FOR SECTION OF INTEREST
C
      II(NU,E,F)=NU/3.DO*DSQRT(1.DO+(DELTA+NU*NU)**2)*
     .(1.DO+2.DO*DELTA*G*G/(1.DO+NU*NU*G*G))
     .+((1.DO+DELTA*G*G)*F-2.DO*DELTA*G*G*E)/3.DO/G**3
C
      IF(ANAL.EQ.O)GOTO 200
      IF(A.NE.O.DO)GOTO 100
      IF(B.NE.O.DO)GOTO 110
C
C A AND B EQUAL TO O
      L=D*DSQRT(1.DO+C*C)
      RETURN
C
C A EQUAL O, B NOT EQUAL O
110   T1=(2.DO*B*D+C)*DSQRT(1.DO+(2.DO*B*D+C)**2)
      T2=C*DSQRT(1.DO+C*C)
      T3=DLOG((2.DO*B*D+C)+DSQRT(1.DO+(2.DO*B*D+C)**2))
      T4=DLOG(C+DSQRT(1.DO+C*C))
      L=(T1-T2+T3-T4)/4.DO/B
      RETURN
C
C A NOT EQUAL O
100   NU1=DSQRT(3.DO*DABS(A))*(D+B/3.DO/A)
      ANU1=DABS(NU1)
      NUO=B/3.DO/A*DSQRT(3.DO*DABS(A))
      ANUO=DABS(NUO)
      DELTA=(C-B*B/3.DO/A)*DSIGN(1.DO,A)
      G=1.DO/(1.DO+DELTA*DELTA)**0.25DO
      K=DSQRT(5.D-1-DELTA*G*G/2.DO)
      E2=O.DO
      F2=O.DO
      EO2=O.DO
      FO2=O.DO
      XO=2.DO*G*ANUO/(1.DO-ANUO*ANUO*G*G)
      X1=2.DO*G*ANU1/(1.DO-ANU1*ANU1*G*G)
      CK=DSQRT(1.DO-K*K)
      IF(ANU1.EQ.1.DO/G)GOTO 120
      IF(ANU1.GT.1.DO/G)GOTO 130
C
C ZETA LESS THAN PI/2
      CALL DELI1(F1,X1,CK)
      CALL DELI2(E1,X1,CK,1.DO,CK*CK)
      GOTO 140
C ZETA GREATER THAN PI/2
130   CALL DELI1(F2,-X1,CK)
      CALL DELI2(E2,-X1,CK,1.DO,CK*CK)
C ZETA EQUALS PI/2
120   CALL DCEL1(F3,K,IER)
      CALL DCEL2(E3,K,1.DO,CK*CK,IER)
      F1=2.DO*F3-F2
      E1=2.DO*E3-E2
140   IF(ANUO.EQ.1.DO/G)GOTO 150
      IF(ANUO.GT.1.DO/G)GOTO 160
C ZETA LESS THAN PI/2
      CALL DELI1(FO,XO,CK)
      CALL DELI2(EO,XO,CK,1.DO,CK*CK)
      GOTO 170
C ZETA GREATER THAN PI/2
160   CALL DELI1(FO2,-XO,CK)
      CALL DELI2(EO2,-XO,CK,1.DO,CK*CK)
C ZETA EQUALS PI/2
150   CALL DCEL1(FO3,K,IER)
      CALL DCEL2(EO3,K,1.DO,CK*CK,IER)
      FO=2.DO*FO3-FO2
      EO=2.DO*EO3-EO2
```

```
170   L=(DSIGN(1.DO,NU1)*II(ANU1,E1,F1)-DSIGN(1.DO,NUO)*II(ANUO,EO,FO))
     ./DSQRT(3.DO*DABS(A))
      RETURN
C
C NON-ANALYTICAL (APPROXIMATE) SFC. LENGTH DETERMINATION.
200   L=0.DO
      YP=0.DO
      DISTP=0.DO
        DO 210 I=1,25
        DIST=D*DFLOAT(I)/25.DO
        Y=((A*DIST+B)*DIST+C)*DIST
        L=L+DSQRT((DIST-DISTP)**2+(Y-YP)**2)
        YP=Y
        DISTP=DIST
210     CONTINUE
      RETURN
      END

C
C ******************************************************************
C
      SUBROUTINE STAB(RED,CD,UD,VD,UA,VA,CDS,EIGMX)
C
C WRITTEN BY: M. OLESKIW  ON: 810608  LAST MODIFIED:810610
C
C FINDS THE JACOBIAN (DF/DY), ITS EIGENVALUES AND DETERMINES
C   SUITABILITY OF ODE INTEGRATING TECHNIQUE.
C
      DOUBLE PRECISION XP(13),YP(13),PSI(13),Z1,Z2,Z3,Z4,K2,K3,K4,
     .RED,CD,DCD,UD,VD,UA,VA,DSQRT,G1,G2,G3,DG3X,DG3Y,DG3U,DG3V,
     .J(4,4),EIG(8),ZZ(32),WK(25),DD(5),C,RDS,EIGMX,DMIN1
C
      INTEGER CDS,N,IA,IZ,IER,JJ,TYPE
C
      COMMON /AIR/XP,YP,PSI/STAB1/K2,K3,K4/TRANS3/DD,C,TYPE,JJ
C
C IN  RED=RELATIVE MOTION REYNOLDS NO.
C IN  CD=DRAG COEFFICIENT
C IN  UD=
C IN  VD=DROPLET VELOCITY COMPONENTS.
C IN  UA=  "
C IN  VA=AIR VELOCITY COMPONENTS.
C IN  CDS=PARAMETER TO DETERMINE DRAG COEFFICIENT FORMULATION.
C OUT EIGMX=LARGEST NEGATIVE REAL PART OF EIGENVALUES OF JACOBIAN MATRIX
C
      RDS=DD(JJ)/2.D6/C
C DUAS/DXDS
      Z1=(PSI(6)-PSI(7)-PSI(8)+PSI(9))/4.DO/RDS/RDS
C DUAS/DYDS
      Z2=(PSI(10)-2.DO*PSI(5)+PSI(13))/4.DO/RDS/RDS
C DVAS/DXDS
      Z3=(2.DO*PSI(5)-PSI(11)-PSI(12))/4.DO/RDS/RDS
C DVAS/DYDS
      Z4=(PSI(8)-PSI(6)-PSI(9)+PSI(7))/4.DO/RDS/RDS
C
      G1=UD-UA
      G2=VD-VA
C RELVEL
      G3=RED/K2
      DG3X=-(G1*Z1+G2*Z3)/G3
      DG3Y=-(G1*Z2+G2*Z4)/G3
      DG3U=G1/G3
      DG3V=G2/G3
C FIND DCD/DRED
      IF(CDS.EQ.2)GOTO 300
      IF(CDS.EQ.1.AND.RED.LE.5.DO)GOTO 100
```

```
      DCD=-0.5848D0*(1.D0+9.06D0/DSQRT(RED))*4.53D0/RED**1.5D0
      GOTO 400
 100  DCD=-24.D0/RED/RED
      GOTO 400
 300  DCD=-24.D0/RED/RED-1.75D0/RED**1.37D0+2.37D-3/RED**0.62D0
C
C FILL JACOBIAN MATRIX
 400  J(1,1)=0.D0
      J(2,1)=-K3*(DCD*K2*DG3X*G3*G1+DG3X*CD*G1-Z1*CD*G3)
      J(3,1)=0.D0
      J(4,1)=-K3*(DCD*K2*DG3X*G3*G2+DG3X*CD*G2-Z3*CD*G3)
      J(1,2)=1.D0
      J(2,2)=-K3*(DCD*K2*DG3U*G3*G1+DG3U*CD*G1+CD*G3)
      J(3,2)=0.D0
      J(4,2)=-K3*(DCD*K2*DG3U*G3*G2+DG3U*CD*G2)
      J(1,3)=0.D0
      J(2,3)=-K3*(DCD*K2*DG3Y*G3*G1+DG3Y*CD*G1-Z2*CD*G3)
      J(3,3)=0.D0
      J(4,3)=-K3*(DCD*K2*DG3Y*G3*G2+DG3Y*CD*G2-Z4*CD*G3)
      J(1,4)=0.D0
      J(2,4)=-K3*(DCD*K2*DG3V*G3*G1+DG3V*CD*G1)
      J(3,4)=1.D0
      J(4,4)=-K3*(DCD*K2*DG3V*G3*G2+DG3V*CD*G2+CD*G3)
C FIND EIGENVALUES OF THE JACOBIAN.
      N=4
      IA=4
      IZ=4
      CALL EIGRF(J,N,IA,0,EIG,ZZ,IZ,WK,IER)
      EIGMX=DMIN1(EIG(1),EIG(3),EIG(5),EIG(7))
      RETURN
      END

C
C ******************************************************************
C
      SUBROUTINE STRMFN(TYPE)
C
C WRITTEN BY: M. OLESKIW  ON: 800222  LAST MODIFIED: 810823
C
C CALCULATE STREAMFUNCTION ON A GRID ABOUT AN AEROFOIL SECTION
C   GIVEN THE SFC. VORTICITY DENSITY ON THE AEROFOIL AND PLOT THE
C   FLOW USING VELOCITY VECTORS.
C REF: KENNEDY, J.L. & D.F. MARSDEN (1976), CAN. AERO. & SPACE JOUR.
C       V 22, #5, PP 243-256
C
      DOUBLE PRECISION ALPHAR,XE(101),YE(101),XC(101),YC(101),GAMMA(101)
     .,D(100),SI(100),CO(100),DBLE,YUP1,YLP1,YU,YL,ZZ,PJK,DD,
     .PID,YUM1,YLM1,TH
C
      REAL PSI(3721),K(101),DELTA,PI,ALPHAS,SNGL,SCO,SSI,X,Y,DXC,DYC,
     .XMIN,XMAX,YMIN,YMAX,B,A,R1S,R2S,T3,ATAN,SIGN,T1,T2,DEN,AA,MM,
     .R,ABS,LOG,FLOAT,SIN,COS,R3S,DX,DY,DPX,DPY,XPAGE,YPAGE,AINT,
     .XTIP,YTIP,XP1,YP1,YM1,U,V,AHL,AHLEN,SQRT,XM1,SIGMA,DXX,
     .DYY,DDXX,DDYY
C
      INTEGER XZ,YZ,TYPE,J,I,M,XZ1,YZ1,F,N,NCOU,NCOL,L,IIU,IIL,MOD,
     .INC,NDCPX,NDCPY
C
      COMMON ALPHAR,PID/AERO1/XE,YE/AERO3/NCOU,NCOL/AERO2/XC,YC,GAMMA,D,
     .SI,CO/NACA/TH
     ./GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ/SRCH/DD,IIU,IIL
C
C IN  TYPE=AEROFOIL TYPE.
C
      N=NCOU+NCOL-2
      PI=SNGL(PID)
```

```
C ALPHAR=ANGLE OF ATTACK IN RADIANS
        ALPHAS=SNGL(ALPHAR)
          DO 120 J=1,XZ
          X=XMIN+FLOAT(J-1)/FLOAT(XZ-1)*(XMAX-XMIN)
            IF(MOD(J,2).EQ.0)GOTO 121
            I=2
            INC=2
            GOTO 123
121         I=1
            INC=1
C PSI IS STORED IN VECTOR FORM BY COLUMNS.
123         M=(J-1)*YZ+I
            Y=YMAX-FLOAT(I-1)/FLOAT(YZ-1)*(YMAX-YMIN)
            PSI(M)=0.0
            IF(TYPE.EQ.-1)GOTO 135
            IF(TYPE.EQ.-2.OR.TYPE.EQ.-3)GOTO 300
            IF(TYPE.EQ.-10)GOTO 400
              DO 140 L=1,N
C FIND DISTANCE BETWEEN CONTROL PT. L AND GRID PT. I,J.
              DXC=X-SNGL(XC(L))
              DYC=Y-SNGL(YC(L))
C CALCULATE COMPONENTS OF EQN. 9 AND FIG. 2
              DELTA=SNGL(D(L))/2.0
              SCO=SNGL(CO(L))
              SSI=SNGL(SI(L))
              B=DXC*SCO+DYC*SSI
              A=DYC*SCO-DXC*SSI
              R1S=A*A+(B+DELTA)*(B+DELTA)
              R2S=A*A+(B-DELTA)*(B-DELTA)
              R3S=A*A+B*B-DELTA*DELTA
              IF(R3S.LT.1.E-30)GO TO 160
              T3=ATAN(2.0*A*DELTA/R3S)
              GO TO 170
160           IF(ABS(A).LT.1.E-30)GO TO 180
              T3=ATAN((B+DELTA)/A)-ATAN((B-DELTA)/A)
              GO TO 170
180           T3=SIGN(PI,A)
170           T1=(B+DELTA)*LOG(R1S)
              T2=(B-DELTA)*LOG(R2S)
              K(L)=(T1-T2+2.0*A*T3-4.0*DELTA)/4.0/PI
              PSI(M)=PSI(M)-SNGL(GAMMA(L))*K(L)
140           CONTINUE
           R=Y*COS(ALPHAS)-X*SIN(ALPHAS)
C ASSURE THAT PSI ON AEROFOIL = 0.
           PSI(M)=PSI(M)+R-SNGL(GAMMA(N+1))
           GOTO 130
C
C STREAMFN. FOR A CYLINDER.
135        DEN=(X-0.5)**2+Y*Y
           IF(DEN.LT.1.E-70)GOTO 136
           PSI(M)=Y-Y/4.D0/DEN
           GOTO 130
C
C STREAMFN FOR A JOUKOWSKI AEROFOIL
300        PSI(M)=SNGL(PJK(DBLE(X),DBLE(Y)))
           GOTO 130
C
C STREAMFN. FOR A FLYING CIGAR.
400        AA=SNGL(TH)/400.0
           MM=AA*AA
           IF(Y.LT.0.0)GOTO 410
           PSI(M)=MM*(AA-X)/SQRT((AA-X)**2+Y*Y)+Y*Y/2.0
           GOTO 130
410        PSI(M)=2.0*MM-MM*(AA-X)/SQRT((AA-X)**2+Y*Y)-Y*Y/2.0
           GOTO 130
C
```

```
136         PSI(M)=0.0
130         I=I+INC
            IF(I.LE.YZ)GOTO 123
120       CONTINUE
        XZ1=XZ-1
        YZ1=YZ-1
        DPX=20./FLOAT(XZ1)
        DX=(XMAX-XMIN)/FLOAT(XZ1)
        DPY=12./FLOAT(YZ1)
        DY=(YMAX-YMIN)/FLOAT(YZ1)
C
        ENTRY STMFCN
C PLOT BOUNDARIES
        NDCPX=0
        NDCPY=0
        DXX=(XMAX-XMIN)/20.0
        DYY=(YMAX-YMIN)/12.0
        DDXX=ABS(4.0*DXX)+1.E-6
        DDYY=ABS(2.0*DYY)+1.E-6
500     IF(DDXX-AINT(DDXX).LT.2./10.**(6-NDCPX))GOTO 510
        NDCPX=NDCPX+1
        DDXX=DDXX*10.0
        GOTO 500
510     IF(DDYY-AINT(DDYY).LT.2./10.**(6-NDCPY))GOTO 520
        NDCPY=NDCPY+1
        DDYY=DDYY*10.0
        GOTO 510
C DRAW AXES FOR ICE ACCRETION PLOT.
520     CALL NEWPEN(1)
        CALL ORIGIN(999,20.0,12.0,5\0.5.0)
        CALL AX2EP(4.0,3,NDCPX,1,0,9)
        CALL AXIS2(0..0.,'X/C',-3,20.,0.,XMIN,DXX,4.0)
        CALL AXIS2(20..0.,' ',-1,-12.0,90.,0.,0.,2.0)
        CALL AX2EP(2.0,3,NDCPY,1,1,2)
        CALL AXIS2(0..0.,'Y/C',3,12.0,90.,YMIN,DYY,-2.0)
        CALL AXIS2(0.,12.0,' ',1,-20.,0.,0.,0.,4.0)
C CHANGE TO SECOND PEN
        CALL NEWPEN(2)
        IIU=1
        IIL=1
        SIGMA=1.0
        YUP1=-1.D-10
        YLP1=1.D-10
          DO 200 J=2,XZ1,2
          F=0
          X=XMIN+FLOAT(J-1)*DX
C ARROWHEAD TAIL IN FRAME COORDS.
          XPAGE=FLOAT(J-1)*DPX
          XP1=X+DX
          XM1=X-DX
C CHECK IF CENTERED DIFFERENCING IS OK
          IF(XP1.LE.SNGL(XE(1)).OR.XM1.GE.SNGL(XE(NCOU)))GOTO 220
          YUM1=YUP1
          YLM1=YLP1
          IF(X.GT.SNGL(XE(1)).AND.X.LT.SNGL(XE(NCOU)))GOTO 320
          YU=-1.D-10
          YL=1.D-10
          GOTO 330
320       F=1
          CALL SFC(DBLE(X),YU,1,0,ZZ)
          CALL SFC(DBLE(X),YL,0,0,ZZ)
330       IF(XP1.GT.SNGL(XE(NCOU)))GOTO 280
          CALL SFC(DBLE(XP1),YUP1,1,0,ZZ)
          CALL SFC(DBLE(XP1),YLP1,0,0,ZZ)
          GOTO 290
280       YUP1=YL
```

```
            YLP1=YU
      290   F=F+1
C DO FOR EACH COLUMN OF ARROWHEAD TAILS
      220      DO 210 I=2,YZ1,2
               Y=YMAX-FLOAT(I-1)*DY
C ARROWHEAD TAIL IN FRAME COORDS
               YPAGE=12.-FLOAT(I-1)*DPY
               M=(J-1)*YZ+I
               IF(F.NE.2)GOTO 230
               YP1=Y-DY
               YM1=Y+DY
C IS CENTERED DIFFERENCING IN Y OK?
               IF(YP1.GE.SNGL(YU).OR.YM1.LE.SNGL(YL))GOTO 230
               IF(Y.GE.SNGL(YU))GOTO 250
C CHECK FOR LOCATION WITHIN AEROFOIL
               IF(Y.GT.SNGL(YL))GOTO 210
C FORWARD DIFFERENCING IN Y
               IF(TYPE.LE.-10)SIGMA=ABS((Y+YP1)/2.0)
               U=(PSI(M)-PSI(M+1))/DY/SIGMA
               GOTO 240
C BACKWARD DIFFERENCING IN Y
      250      IF(TYPE.LE.-10)SIGMA=ABS((Y+YM1)/2.0)
               U=(PSI(M-1)-PSI(M))/DY/SIGMA
               GOTO 240
C CENTERED DIFFERENCING IN Y
      230      IF(TYPE.LE.-10)SIGMA=ABS(Y)
               U=(PSI(M-1)-PSI(M+1))/2.0/DY/SIGMA
C
C IS CENTERED DIFFERENCING IN X OK?
      240      IF(TYPE.LE.-10)SIGMA=ABS(Y)
               IF(F.EQ.0)GOTO 260
               IF(Y.GE.SNGL(YUP1).AND.Y.GE.SNGL(YUM1))GOTO 260
               IF(Y.LE.SNGL(YLP1).AND.Y.LE.SNGL(YLM1))GOTO 260
C IS FORWARD DIFFERENCING OK?
               IF(Y.GE.SNGL(YUP1).OR.Y.LE.SNGL(YLP1))GOTO 310
C BACKWARD DIFFERENCING IN X
               V=(PSI((J-2)*YZ+I)-PSI((J-1)*YZ+I))/DX/SIGMA
               GOTO 270
C FORWARD DIFFERENCING IN X
      310      V=(PSI((J-1)*YZ+I)-PSI(J*YZ+I))/DX/SIGMA
               GOTO 270
C CENTERED DIFFERENCING IN X
      260      V=(PSI((J-2)*YZ+I)-PSI(J*YZ+I))/2.0/DX/SIGMA
C ARROWHEAD TIP
      270      XTIP=XPAGE+U*DPX
               YTIP=YPAGE+V*DPX
               AHL=SQRT(U*U+V*V)
C ARROWHEAD LENGTH
               AHLEN=0.25*AHL*DPX
               CALL AROHD(XPAGE,YPAGE,XTIP,YTIP,AHLEN,0,16)
      210      CONTINUE
      200   CONTINUE
         RETURN
         END

C
C ********************************************************************
C
         SUBROUTINE TRAJEC(TRJPLA,THICK,AT,BOTH,DDISTN,LAYER,GRAZE)
C
C WRITTEN BY: M. OLESKIW   ON:790526   LAST MODIFIED:811018
C
C CALCULATE TRAJECTORIES OF DROPLETS IN POTENTIAL FLOW
C ABOUT AN AEROFOIL
C
         DOUBLE PRECISION DFLOAT,UINF,C,DD(5),CD,G5,RDS,RHOA,RHOD,MUS,
```

```
          .MU,DTS(6,2),XP(13),YP(13),WDSREL,DBLE,HF,UST,VST,EPS(5),PI,
          .CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,
          .C17,C18,C19,C20,C21,C22,C23,C24,HFP,AS,DATAN2,CLAPN,CX,CY,
          .CLAP,ACOLLD,TIM1,TIM2,TST,XCD,XIM1,XIM2,XPL1,XXI,YCA,YOG,KLAP,
          .XCOLL,YCOLL,DABS,DSIGN,ACOLL,ACOL(31,5),YIM1,YIM2,TCOLL,
          .CLAPP,K,LTH,XN,YN,ALPHAR,D,LW(2),YOW(2),VTW(2),ACW(2)
 3         DOUBLE PRECISION PSI(13),DUADX,DVADY,DMIN1,L(31),YO(31),XUXR,
          .UAS(6,2),VAS(6,2),RED(6,2),AU,BU,CU,AV,BV,CV,UCOLL,VCOLL,XLXR,
          .LEN,PRDSTI,PRDSTO,DIST,TS(500,2),UVAT,XXO,YYO,XX,YY,TTO,TT1,
          .DSORT,PINF,TINF,XO(5),YOT(10,2),YOI(2,5),DFDY,GEXX,GEX,GEY
 4         DOUBLE PRECISION XDS(6,2),UDS(6,2),AN(2,6,2),YDS(6,2),TTLACN,VPSQ,
          .VDS(6,2),HT(2,6,2),AO,A1,A2,BO,B1,B2,B3,E5B,CO,C1,C2,ATJ,
          .DM1,DO,D1,D2,E5,UPI,UCI,VPI,VCI,XPI,XCI,YPI,YCI,ER1,ER2,
          .PRD,THICK,SLP,XCLAP,LAMBH,MLAMBH,GEU,GEV,DMAX1,NA,ZZ,TD,PSIP,PSIN,
          .VTOT(31,5),VTTL,K2,K3,K4,AX,AY,BX,BY,CIM2,CIM1,XL,XR,HCLAP,HCLAPN
      C
 5         REAL XMIN,XMAX,YMIN,YMAX,SNGL,X,Y,XDSP(250),YDSP(250),YPREV,
          .XPREV
      C
 6         INTEGER I(2),CDS,XZ,YZ,IJ,IK,TRJEND,SMASH,AT,BOTH,ACN,
          .GRAZE,IG,J,IU,IL,N,DENSE,FNCALL,SHORT,WARN,WARNP,IABS,
          .TRJPRA,TRJPLA,PRINTI,PRINTO,TYPE,GLOBAL,CPRED,S,LL,FR,GER,
          .IM4(2),IM3(2),IM2(2),IM1(2),IO(2),IP1(2),MM,ITEMP,EON,PPC,PC,
          .LAYER,IMN1,DDISTN,EQ,GRAV,ITP,SHORTP
      C
 7         COMMON ALPHAR,PI/EONMN/GS,RHOA,RHOD,RDS,NUS,HF
          ./AIR/XP,YP,PSI/REL/UAS,VAS,RED,CD/STAB1/K2,K3,K4
          ./GRID/XMIN,XMAX,YMIN,YMAX,XZ,YZ/XXR/XUXR,XLXR
          ./PV/XDS,YDS,UDS,VDS/INTEG/AN,HT/SA/AS
          ./PCM/AO,A1,A2,BO,B1,B2,B3,CO,C1,C2,DM1,DO,D1,D2,
          .UPI,UCI,VPI,VCI,ER1,ER2,XPI,XCI,YPI,YCI,UST,VST
          ./LOC/TS,DTS,I,IM4,IM3,IM2,IM1,IO,IP1,MM
 8         COMMON /RKFM/CC1,CC2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,
          .C15,C16,C17,C18,C19,C20,C21,C22,C23,C24
          ./COLS/L,LW,YO,YOW,VTW,ACW/NOSE/XN,YN/FC/FNCALL
          ./TRANS1/UINF,PINF,TINF,EPS,DENSE/SRCH/D,IU,IL
          ./TRANS2/CDS,TRJPRA,PRINTI,PRINTO,EON,
          .PPC,ACN,GRAV/TRANS3/DD,C,TYPE,J/CV/VTOT,ACOL
          ./WH/AX,BX,CX,AY,BY,CY,XXO,YYO,TTO,TT1
      C
      C IN  TRJPLA=PLOT TRAJECTORIES (O OR 1)
      C IN  THICK=AEROFOIL THICKNESS IN %
      C IN  AT=AUTO-TRAJECTORY MODE (O OR 1)
      C IN  BOTH=TRAJECTORIES TO COLLIDE ON BOTH SFCS  (O OR 1)
      C IN  DDISTN=NO  OF SIZES IN DROPLET DISTN.
      C IN  LAYER=LAYER NO
      C IN  GRAZE=FIND GRAZING TRAJECTORY MODE (O OR 1)
      C
 9    10   FORMAT('1ACCRETION OF LAYER',I3,'  DROPLET DIAMETER:',F7.1)
10    11   FORMAT('1ACCRETION OF LAYER',I3,'  DROPLET DIAMETER:',F7.1,
          .'   EPS=',1PE10.3)
11    12   FORMAT('-ACCRETION OF LAYER',I3,'  DROPLET DIAMETER:',F7.1)
12    13   FORMAT('-ACCRETION OF LAYER',I3,'  DROPLET DIAMETER:',F7.1,
          .'   EPS=',1PE10.3)
13    15   FORMAT('1')
14    20   FORMAT(' C *WARNING*** STABILITY PARAMETER INDICATES POSSIBLE',
          .' INSTABILITY')
15    30   FORMAT(/5F10.0)
16    40   FORMAT('OSTEP',T8,'TIME',T15,'DTS',T23,'XDS',T32,'YDS',T41,'PSI',
          .T50,'UAS',T59,'UDS',T68,'VAS',T77,'VDS',T88,'RED',T94,
          .'ACCN/MOD HIST/RHS',T114,'USTAB',T123,'VSTAB')
17    50   FORMAT(' ',I4,F6.2,F7.4,7F9.5,F11.3,4F9.5)
18    55   FORMAT(' *',I3,F6.2,F7.4,7F9.5,F11.3,4F9.5)
19    60   FORMAT(' STABILITY INDEX:',F8.3,'  AT X=',F8.5,
          .'   CLOSEST APPROACH IS Y=',F8.5,
          ./,' TIME STEPS=',I3,'  FN. EVALUATIONS=',I4,'  FINAL Y=',F8.5)
```

```
65     FORMAT('OSTABILITY INDEX:',F8.3,'  AT X=',F8.5,
      .'  CLOSEST APPROACH IS Y=',F8.5,
      ./,'OTIME STEPS=',I3,'  FN. EVALUATIONS=',I4,'  FINAL Y=',F8.5)
70     FORMAT('OTRAJECTORY STARTING POSITION IS X=',
      .F6.2,'  YO=',F8.5)
80     FORMAT(' COLLISION COORDS: X=',F8.5,'  Y=',F8.5,'  L=',F8.5,/,
      .' TIME STEPS=',I3,'  FN. EVALUATIONS=',I4,
      /,' STABILITY INDEX=',F8.3,'  COLLISION VELOCITY:',F8.5,'  AT',
      F6.1,' DEG ')
85     FORMAT('OCOLLISION COORDS  X=',F8.5,'  Y=',F8.5,'  L=',F8.5,/,
      'OTIME STEPS=',I3,'  FN  EVALUATIONS=',I4,
      /,'OSTABILITY INDEX=',F8.3,'  COLLISION VELOCITY:',F8.5,'  AT',
      F6.1,' DEG ')
90     FORMAT('OFIRST TRAJECTORY HIT AEROFOIL')
95     FORMAT('OUNEXPECTED AEROFOIL MISS')
96     FORMAT('OOD,EPS,XO,YO,GLOBAL?')
97     FORMAT(F10.0,D10.0,2F10.0,I2)
98     FORMAT(' GLOBAL ERRORS AT X=',F8.5,' ARE:  IN X:',F8.5,'  IN Y:',
      F8.5,'  IN U:',F8.5,'  IN V:',F8.5)
99     FORMAT('OGLOBAL ERRORS AT X=',F8.5,' ARE:  IN X:',F8.5,'  IN Y:',
      F8.5,'  IN U:',F8.5,'  IN V:',F8.5)
C
       IU=1
       IL=1
       IF(AT.EQ.0)GOTO 710
C INPUT PARAMETERS FOR AUTO-TRAJECTORY MODE.
       READ(4,30)(XO(J),J=1,DDISTN)
       READ(4,30)(YOI(2,J),J=1,DDISTN)
       IF(BOTH.EQ.1)READ(4,30)(YOI(1,J),J=1,DDISTN)
C FIND GRAZING TRAJECTORIES FIRST.
       GRAZE=1
       IF(PPC.LE.2)GLOBAL=0
C SET FOR STEP EXTRAPOLATION
       IF(PPC.EQ.3.AND.EQN.NE.2)GLOBAL=2
C FIRST CATEGORY IN DROPLET DISTN.
       J=1
C
C NON-DIMENSIONAL VIEWPORT DIAGONAL LENGTH
710    LEN=DSQRT(DBLE((XMAX-XMIN)**2+(YMAX-YMIN)**2))
C PRINT LENGTH INTERVAL WITHIN VIEWPORT
       PRDSTI=LEN/DFLOAT(PRINTI)
C PRINT LENGTH INTERVAL TO LEFT OF VIEWPORT
       PRDSTO=LEN/DFLOAT(PRINTO)
C NON-DIMENSIONAL ACCN. OF GRAVITY
       GS=DFLOAT(GRAV)*9.81D0*C/UINF/UINF
C AIR DENSITY
       RHOA=PINF*1.D3/287.04D0/(TINF+273.16D0)
C WATER DENSITY  REF:  LIST - SMT
       RHOD=999.15D0
C DYNAMIC VISCOSITY OF AIR  REF: LOZOWSKI ET AL. (1979)
       MU=1.718D-5+5.1D-8*TINF
C NON-DIMENSIONAL KINEMATIC VISCOSITY OF AIR
       NUS=MU/RHOA/C/UINF
       IF(PPC.LT.2)GOTO 420
C
C DETERMINE PARAMETERS FOR RUNGE-KUTTA-FEHLBERG METHOD.
       CC1=.25D0
       CC2=3.D0/32.D0
       C3=9.D0/32.D0
       C4=1932.D0/2197.D0
       C5=72.D2/2197.D0
       C6=7296.D0/2197.D0
       C7=439.D0/216.D0
       C8=8.D0
       C9=3680.D0/513.D0
       C10=845.D0/4104.D0
```

```
      C11=8.DO/27.DO
      C12=2.DO
      C13=3544.DO/2565.DO
      C14=1859.DO/4104.DO
      C15=11.DO/40.DO
      C16=25.DO/216.DO
      C17=1408.DO/2565.DO
      C18=2197.DO/4104.DO
      C19=.2DO
      C20=16.DO/135.DO
      C21=6656.DO/12825.DO
      C22=28561.DO/56430.DO
      C23=9.DO/50.DO
      C24=2.DO/55.DO
      GOTO 400
420   IF(PPC.NE.1)GOTO 400
C
C DETERMINE PARAMETERS FOR PREDICTOR-CORRECTOR METHOD.
      A1=0.DO
      A2=6.25D-2
      AO=1.DO-A1-A2
      BO=(55.DO+9.DO*A1+8.DO*A2)/24.DO
      B1=(-59.DO+19.DO*A1+32.DO*A2)/24.DO
      B2=(37.DO-5.DO*A1+8.DO*A2)/24.DO
      B3=(-9.DO+A1)/24.DO
      E5B=(251.DO-19.DO*A1-8.DO*A2)/6.DO
      C1=A1
      C2=A2
      CO=1.DO-C1-C2
      DM1=(9.DO-C1)/24.DO
      DO=(19.DO+13.DO*C1+8.DO*C2)/24.DO
      D1=(-5.DO+13.DO*C1+32.DO*C2)/24.DO
      D2=(1.DO-C1+8.DO*C2)/24.DO
      E5=(-19.DO+11.DO*C1-8.DO*C2)/6.DO
      ER1=E5B/(E5B-E5)
      ER2=E5/(E5B-E5)
C
400   IF(AT.EQ.1)GOTO 470
        J=1
C READ IN VALUES FOR INDIVIDUAL TRAJECTORY MODE.
490     WRITE(6,96)
        READ(5,97)DD(J),EPS(J),XDS(1,1),YDS(1,1),GLOBAL
        IF(DD(J).EQ.0.DO)RETURN
        IF(PPC.LE.1.OR.EQN.EQ.2)GLOBAL=0
        GOTO 480
C
C BEGINNING OF AUTO-TRAJECTORY MODE.
        ENTRY TRAJEK(LAYER,GRAZE,N)
C
C IN  N=INDEX OF TRAJECTORY PAIR.
C
470     IJ=3
        IF(GRAZE.EQ.1)GOTO 460
C TRAJECTORY SPECIFIED BY CE SUBROUTINE.
        YDS(1,1)=YOW(N)
        GOTO 405
C TRAJECTORY DETERMINED TO FIND GRAZING TRAJECTORY.
460     IJ=IJ-1
        IF(IJ.EQ.2.AND.TRJPRA.EQ.O)WRITE(7,15)
        IG=1
C SLOPES FOR SECANT METHOD.
C ***************************************************************
        K=0.85DO
C ***************************************************************
        TD=0.2DO
C INITIAL DROPLET POSITION
```

```
              YDS(1,1)=YOI(IJ,J)
      405     XDS(1,1)=XO(J)
      C
      C PARAMETERS FOR CALCULATING THE JACOBIAN (DF/DY).
      480     RDS=DD(J)/C/2.D6
              K2=2.DO*RDS/NUS
              K3=0.75DO*RHOA/RDS/(2.DO*RHOD+RHOA)
              K4=2.DO*(RHOD-RHOA)/(2.DO*RHOD+RHOA)
      C
      C SET COUNTERS
                DO 485 MM=1,2
                IM4(MM)=2
                IM3(MM)=3
                IM2(MM)=4
                IM1(MM)=5
                IO(MM)=6
                IP1(MM)=1
                I(MM)=0
      485       CONTINUE
              IK=0
              CPRED=0
              GER=0
              WARN=0
              MLAMBH=0.DO
              FNCALL=0
              MM=1
              XCLAP=XDS(1,1)
      C DROPLET AT INITIAL POSITION
              IF(TRJPRA.EQ.1)GOTO 406
              IF(PPC.LT.2)WRITE(7,12)LAYER,DD(J)
              IF(PPC.GE.2)WRITE(7,13)LAYER,DD(J),EPS(J)
              GOTO 407
      406     IF(PPC.LT.2)WRITE(7,10)LAYER,DD(J)
              IF(PPC.GE.2)WRITE(7,11)LAYER,DD(J),EPS(J)
      407     WRITE(6,70) XDS(1,1),YDS(1,1)
              WRITE(7,70) XDS(1,1),YDS(1,1)
              IF(PPC.NE.1)GOTO 410
      C
      C SET PREVIOUS PREDICTOR-CORRECTOR VALUES TO 0.
              XPI=0.DO
              XCI=0.DO
              YPI=0.DO
              YCI=0.DO
              UPI=0.DO
              UCI=0.DO
              VPI=0.DO
              VCI=0.DO
      410     IF(ACN.EQ.1)GOTO 415
      C
      C SET DROPLET TRAVELLING WITH JUST SLIGHTLY GREATER VELOCITY
      C  THAN AIR (RED=0.001)
              CALL AIRVEL(XDS(1,1),YDS(1,1),UAS(1,1),VAS(1,1),5)
      C CALCULATE TOTAL AIR VELOCITY
              UVAT=DSQRT(UAS(1,1)*UAS(1,1)+VAS(1,1)*VAS(1,1))
      C CALCULATE TOTAL STARTING RELATIVE VELOCITY.
              WDSREL=1.D-3*NUS/2.DO/RDS
      C CALCULATE INITIAL DROPLET VELOCITY
              UDS(1,1)=UAS(1,1)*(1.DO+WDSREL/UVAT)
              VDS(1,1)=VAS(1,1)*(1.DO+WDSREL/UVAT)
              GOTO 416
      C
      C ASSURE STARTING RED=0.001 WEIGHTED BY POTENTIAL FLOW
      C   ACCELERATIVE COMPONENTS.
      C SET GRID FOR INITIAL DROPLET VELOCITY CALCULATIONS
      415     XP(6)=XDS(1,1)+RDS
              XP(7)=XP(6)
```

```
               XP(8)=XDS(1,1)-RDS
               XP(9)=XP(8)
               YP(6)=YDS(1,1)+RDS
               YP(7)=YDS(1,1)-RDS
               YP(8)=YP(6)
               YP(9)=YP(7)
C FIND AIR VELOCITY
               CALL AIRVEL(XDS(1,1),YDS(1,1),UAS(1,1),VAS(1,1),9)
C CALCULATE DUA/DX
               DUADX=(PSI(6)-PSI(7)-PSI(8)+PSI(9))/4.D0/RDS/RDS
C CALCULATE DVA/DY
               DVADY=(PSI(8)-PSI(6)-PSI(9)+PSI(7))/4.D0/RDS/RDS
C TOTAL POTENTIAL FLOW ACCELERATIVE TERM
               UVAT=DSQRT(DUADX*DUADX+DVADY*DVADY)
C CALCULATE TOTAL STARTING RELATIVE VELOCITY
               WDSREL=1.D-3*NUS/2.D0/RDS
               UDS(1,1)=UAS(1,1)-DUADX/UVAT*WDSREL
               VDS(1,1)=VAS(1,1)-DVADY/UVAT*WDSREL
C
416            CALL DRAG(UDS(1,1),VDS(1,1),UAS(1,1),VAS(1,1),CDS,RED(1,1),CD)
C CALCULATE STARTING ACCELERATIONS
               EQ=EON
               IF(EON.EQ.2)EQ=1
               CALL ACCN(UDS(1,1),VDS(1,1),UAS(1,1),VAS(1,1),
               RED(1,1),CD,EQ,0.D0,0)
               IF(TRJPRA.EQ.1)WRITE(7,40)
               IF(AT.EQ.0)WRITE(6,40)
               TRJEND=0
               HT(1,1,1)=0.D0
               HT(2,1,1)=0.D0
               TS(1,1)=0.D0
               CLAP=1.D0
               PSIN=PSI(5)
               SHORT=0
               PC=PPC
               SMASH=0
               IF(PC.LT.2)GOTO 103
C FIND INITIAL STEP SIZE FOR RKF4 & GLERK5.
               DFDY=DMAX1(DABS(UDS(1,1)),DABS(VDS(1,1)),DABS(AN(1,1,1)),
               DABS(AN(2,1,1)))
               IF(GLOBAL.LE.1)DTS(1,1)=0.5D0*(EPS(J)/DFDY)**0.25D0
               IF(GLOBAL.EQ.0)GOTO 100
               IF(GLOBAL.EQ.2)DTS(1,1)=0.33D0*(EPS(J)/DFDY)**0.2D0
C FOR GLOBAL EXTRAPOLATION, INITIALIZE.
               HT(1,1,2)=0.D0
               HT(2,1,2)=0.D0
               TS(1,2)=0.D0
               XDS(1,2)=XDS(1,1)
               YDS(1,2)=YDS(1,1)
               UDS(1,2)=UDS(1,1)
               VDS(1,2)=VDS(1,1)
               UAS(1,2)=UAS(1,1)
               VAS(1,2)=VAS(1,1)
               RED(1,2)=RED(1,1)
               DTS(1,2)=DTS(1,1)
               AN(1,1,2)=AN(1,1,1)
               AN(2,1,2)=AN(2,1,1)
               MM=2
               LL=2
               GOTO 100
C INITIAL STEP SIZE FOR RK4 & PC4.
103            DTS(1,1)=EPS(J)**0.25D0
C
C REINITIALIZE DISTANCE BETWEEN PRINT POSITIONS.
100            PRD=0.D0
C
```

```
      105     IF(GLOBAL.GT.0)GOTO 104
C FOR RK4,PC4,AND RKF4 METHODS.
              FR=1
              MM=1
              GOTO 106
C
      104     IF(GLOBAL.EQ.2)GOTO 107
C FOR ORDER EXTRAPOLATION:
              IF(MM.EQ.1)GOTO 108
              IF(SMASH.EQ.0)GOTO 908
C FIND GLOBAL ERRORS, SINCE TRAJECTORY HAS ENDED UPON
C   SECOND STEP OF PAIR
              GEXX=XDS(IO(1),1)
              GEX=XDS(IO(1),1)-XDS(IO(2),2)
              GEY=YDS(IO(1),1)-YDS(IO(2),2)
              GEU=UDS(IO(1),1)-UDS(IO(2),2)
              GEV=VDS(IO(1),1)-VDS(IO(2),2)
C CONTINUE FIRST STEPS TO END OF TRAJECTORY.
              GER=1
C BEGIN FIRST STEP OF PAIR
      908     MM=1
              FR=1
              GOTO 106
C
      108     IF(GER.EQ.1)GOTO 106
              MM=2
C SECOND STEP IN PAIR OF SAME SIZE.
              DTS(IP1(2),2)=DTS(IO(1),1)
              FR=0
              GOTO 106
C
C FOR STEP EXTRAPOLATION:
      107     IF(GER.EQ.1)GOTO 106
              IF(LL.EQ.2)GOTO 909
              IF(SMASH.EQ.0)GOTO 109
C FIND GLOBAL ERRORS, SINCE TRAJECTORY HAS ENDED
C   UPON FIRST STEP OF TRIPLET.
              GEXX=XDS(IP1(2),2)
              GEX=(XDS(IO(1),1)-XDS(IP1(2),2))/31.D0
              GEY=(YDS(IO(1),1)-YDS(IP1(2),2))/31.D0
              GEU=(UDS(IO(1),1)-UDS(IP1(2),2))/31.D0
              GEV=(VDS(IO(1),1)-VDS(IP1(2),2))/31.D0
C CONTINUE HALF-STEPS TO END OF TRAJECTORY.
              GER=1
              MM=2
              FR=1
              GOTO 106
C BEGIN NEXT STEP OF TRIPLET.
      909     MM=1
              LL=0
              FR=0
              GOTO 106
C
C BEGIN FIRST OR SECOND HALF-STEP.
      109     MM=2
              LL=LL+1
              FR=1
              DTS(IP1(2),2)=DTS(IO(1),1)/2.D0
C
C INCREMENT INDICES
      106     ITEMP=IM4(MM)
              IM4(MM)=IM3(MM)
              IM3(MM)=IM2(MM)
              IM2(MM)=IM1(MM)
              IM1(MM)=IO(MM)
              IO(MM)=IP1(MM)
```

```
                 IP1(MM)=ITEMP
                 I(MM)=I(MM)+1
                 IF(FR.EQ.1)HFP=HF
                 IF(FR.EQ.1)WARNP=WARN
                 XPL1=XDS(IO(MM),MM)+1.2D0*DTS(IO(MM),MM)*UDS(IO(MM),MM)
                 IF(XPL1.LT.XN-RDS+1.D-9.OR.XPL1.GT.
                 DMIN1(XUXR,XLXR)-RDS/5.D0)GOTO 120
C
C HERMITE EXTRAPOLATION TO CHECK FOR COLLISION
                 CALL HERMIT(TS(I(MM)-1,MM),TS(I(MM),MM),
                 XDS(IM1(MM),MM),XDS(IO(MM),MM),
                 UDS(IM1(MM),MM),UDS(IO(MM),MM),AX,BX,CX)
                 CALL HERMIT(TS(I(MM)-1,MM),TS(I(MM),MM),
                 YDS(IM1(MM),MM),YDS(IO(MM),MM),
                 VDS(IM1(MM),MM),VDS(IO(MM),MM),AY,BY,CY)
                 TTO=TS(I(MM)-1,MM)
                 TT1=TS(I(MM),MM)
                 XXO=XDS(IM1(MM),MM)
                 YYO=YDS(IM1(MM),MM)
      130        TS(I(MM)+1,MM)=TS(I(MM),MM)+DTS(IO(MM),MM)
                 TST=TS(I(MM)+1,MM)-TS(I(MM)-1,MM)
                 XDS(IP1(MM),MM)=((AX*TST+BX)*TST+CX)*TST+XDS(IM1(MM),MM)
                 YDS(IP1(MM),MM)=((AY*TST+BY)*TST+CY)*TST+YDS(IM1(MM),MM)
                 CPRED=1
                 X=SNGL(XDS(IP1(MM),MM))
                 XPREV=SNGL(XDS(IO(MM),MM))
                 GOTO 190
C
C INTEGRATE EQNS  OF MOTION VIA HIGHER ORDER TECHNIQUE
C    (RK4,PC4,RKF4 OR GLERK5)
      120        HCLAPN=CLAPN
                 IF(FR.EQ.1)PSIP=PSIN
                 IF(FR.EQ.1)SHORTP=SHORT
                 IF(PC.GE.2)CALL GLERK5(EQN,CDS,EPS(J),LAMBH,WARN,SHORT,
                 GLOBAL,GER)
                 IF(I(MM).GE.4.AND.PC.EQ.1)CALL PC4(EQN,CDS,LAMBH,WARN)
                 IF(I(MM).LT.4.AND.PC.EQ.1.OR.PC.EQ.0)
                 CALL RK4(EQN,CDS,LAMBH,WARN)
C
                 IF(FR.EQ.1)PSIN=PSI(5)
                 CPRED=O
C STABILITY PARAMETER
                 IF(MM.EQ.1)MLAMBH=DMIN1(MLAMBH,LAMBH)
                 IF(FR.EQ.O)GOTO 192
                 IF(WARN.EQ.O.OR.WARNP.EQ.1)GOTO 175
                 WRITE(6,20)
                 IF(TRJPRA.EQ.1)WRITE(7,20)
C CALCULATE DISTANCE SINCE LAST PRINT OF DROPLET POSITION
      175        DIST=DSQRT((XDS(IP1(MM),MM)-XDS(IO(MM),MM))**2+
                 (YDS(IP1(MM),MM)-YDS(IO(MM),MM))**2)
                 PRD=PRD+DIST
                 X=SNGL(XDS(IP1(MM),MM))
                 XPREV=SNGL(XDS(IO(MM),MM))
C CHECK IF DROPLET HAS ENTERED VIEW WINDOW.
                 IF(X.GT.XMIN)GOTO 190
                 IF(TRJPRA.EQ.O)GOTO 105
                 IF(PRD.GE.PRDSTO.OR.SHORTP.EQ.1)GOTO 231
                 GOTO 105
      190        Y=SNGL(YDS(IP1(MM),MM))
                 YPREV=SNGL(YDS(IO(MM),MM))
C CHECK FOR OUT-OF-BOUNDS.
                 IF(Y.GE.YMAX)GOTO 211
                 IF(BOTH.EQ.O)GOTO 191
                 IF(Y.LT.YMIN.AND.YPREV.GT.YMIN)GOTO 212
      191        IF(X.GE.XMAX)GOTO 213
C CHECK IF COLLISION IS POSSIBLE.
```

```
192     IF(XDS(IP1(MM),MM).GE.XN-RDS+1.D-9.AND.
           XDS(IP1(MM),MM).LE.DMIN1(XUXR,XLXR)-RDS/5.DO)GOTO 245
C FIRST POINT TO BE PLOTTED?
        IF(TRJPLA.EQ.O.OR.(TRJPLA.EQ.1.AND.LAYER.GT.1))GOTO 222
        IF(IK.EQ.O.AND.FR.EQ.1.AND.CPRED.EQ.O)GOTO 226
C STORE POINT FOR PLOTTING?
222     IF(CPRED.EQ.1)GOTO 120
        GOTO 221
245     IF(CPRED.EQ.1)GOTO 140
C
C HIGH ORDER INTEGRATING TECHNIQUE (CPRED=O)
        S=1
        CALL WHAMO(XDS(IP1(MM),MM),YDS(IP1(MM),MM),TS(I(MM)+1,MM),
           S,CLAPN,XCD,YCA)
C HAS THERE BEEN A COLLISION?
        IF(CLAPN*DFLOAT(S).LT.O.DO)GOTO 250
        IF(FR.EQ.O)GOTO 105
C IS THIS THE CLOSEST APPROACH?
        IF(CLAPN/CLAP.GT.1.DO)GOTO 221
C STORE CLOSEST APPROACH VALUE AND LOCATION
        CLAP=CLAPN
        HCLAP=HCLAPN
        XCLAP=XCD
        GOTO 221
C
C HERMITE EXTRAPOLATION & HIGHER ORDER INTEGRATING
C    METHOD DON'T AGREE.  LATTER PREDICTS COLLISION.
C    TRY AGAIN USING HALF-SIZE DT (AND RK4 IF USING PC4).
250     IF(PC.EQ.1)PC=O
        DTS(IO(1),1)=DTS(IO(1),1)/2.DO
        IF(PC.GE.2.AND.GLOBAL.NE.O)GOTO 260
        MM=1
260     IF(MM.EQ.1)GOTO 130
C RETURN TO FIRST STEP OF PAIR OR TRIPLET.
        MM=1
        LL=O
        IF(LL.EQ.2)GOTO 270
C DECREMENT INDICES FOR LL=1
        I(2)=I(2)-1
        ITEMP=IP1(2)
        IP1(2)=IO(2)
        IO(2)=IM1(2)
        IM1(2)=IM2(2)
        IM2(2)=IM3(2)
        IM3(2)=IM4(2)
        IM4(2)=ITEMP
        FR=1
        GOTO 130
C DECREMENT INDICES FOR LL=2
270     I(2)=I(2)-2
        ITP=IP1(2)
        ITEMP=IO(2)
        IP1(2)=IM1(2)
        IO(2)=IM2(2)
        IM1(2)=IM3(2)
        IM2(2)=IM4(2)
        IM3(2)=ITP
        IM4(2)=ITEMP
        FR=O
        GOTO 130
C
C HERMITE EXTRAPOLATION TECHNIQUE (CPRED=1)
140     S=1
        CALL WHAMO(XDS(IP1(MM),MM),YDS(IP1(MM),MM),TS(I(MM)+1,MM),
           S,CLAPN,XCD,YCA)
C HAS THERE BEEN A COLLISION?
```

```
C IF NOT, TRY AGAIN WITH A HIGHER ORDER METHOD
C ****************************************************************
        IF(CLAPN/DSIGN(RDS,DFLOAT(S)).GT.1.D-3)GOTO 120
C ****************************************************************
C COLLISION BY HERMITE EXTRAPOLATION
        SMASH=1
        IF(FR.EQ.0)GOTO 105
C IS THIS AN 'ALMOST' COLLISION?
        IF(CLAPN*DFLOAT(S).LT.0.D0)GOTO 320
        XCOLL=XCD
        TCOLL=TS(I(MM)+1,MM)
        CALL SFC(XCOLL,YCOLL,S,1,LTH)
        GOTO 210
C A TRUE COLLISION - FIND COLLISION LOCATION:
C SET UP ITERATIVE PROCEDURE
320     XIM2=DMAX1(XN-RDS+1.D-10,XDS(IO(MM),MM))
        XL=XIM2
        S=2
        CALL WHAMO(XIM2,YIM2,TIM2,S,CIM2,XCD,YCA)
C DOES THE TRAJECTORY CROSS THE YN LINE?
        IF(YIM2*YDS(IP1(MM),MM).GT.0.D0)GOTO 330
        YIM1=DSIGN(1.D-10,YIM2)
        S=3
        CALL WHAMO(XIM1,YIM1,TIM1,S,CIM1,XCD,YCA)
        GOTO 510
330     XIM1=XDS(IP1(MM),MM)
        XR=XIM1
        CIM1=CLAPN
C ITERATE USING SECANT METHOD
510     XXI=XIM1-CIM1*(XIM1-XIM2)/(CIM1-CIM2)
        IF(XXI.GE.XL)GOTO 511
        XXI=XL
511     IF(XXI.LE.XR)GOTO 512
        XXI=XR
512     XIM2=XIM1
        CIM2=CIM1
        XIM1=XXI
        S=2
        CALL WHAMO(XIM1,YIM1,TIM1,S,CIM1,XCD,YCA)
C ****************************************************************
        IF(DABS(XIM1-XIM2).GT.1.D-9)GOTO 510
C ****************************************************************
C COLLISION LOCATION
        XCOLL=XCD
        TCOLL=TIM1
        CALL SFC(XCOLL,YCOLL,S,1,LTH)
C
C END OF TRAJECTORY FLAGGED: COLLISION
210     TRJEND=1
C
C VELOCITY AT COLLISION
        CALL HERMIT(TS(I(MM)-1,MM),TS(I(MM),MM),UDS(IM1(MM),MM),
        UDS(IO(MM),MM),AN(1,IM1(MM),MM),AN(1,IO(MM),MM),AU,BU,CU)
        CALL HERMIT(TS(I(MM)-1,MM),TS(I(MM),MM),VDS(IM1(MM),MM),
        VDS(IO(MM),MM),AN(2,IM1(MM),MM),AN(2,IO(MM),MM),AV,BV,CV)
        TST=TCOLL-TS(I(MM)-1,MM)
        UCOLL=((AU*TST+BU)*TST+CU)*TST+UDS(IM1(MM),MM)
        VCOLL=((AV*TST+BV)*TST+CV)*TST+VDS(IM1(MM),MM)
C TOTAL VELOCITY
        VPSQ=UCOLL*UCOLL+VCOLL*VCOLL
C ANGLE OF TRAJECTORY INCLINATION AT COLLISION
        ATJ=DATAN2(VCOLL,UCOLL)
C ANGLE OF TRAJECTORY FROM PERPENDICULAR TO THE SFC.
        ACOLL=AS-ATJ
        ACOLL=DSIGN(PI/2.D0,ACOLL)-ACOLL
        IF(GRAZE.EQ.1)ACOLL=DSIGN(ACOLL,DFLOAT(2*IJ-3))
```

```
            ACOLLD=ACOLL/PI*1.8D2
            VTTL=DSQRT(VPSQ)
            IK=IK+1
            XDSP(IK)=SNGL(XCOLL)
            YDSP(IK)=SNGL(YCOLL)
            GOTO 232
C END OF TRAJECTORY FLAGGED: EXCEEDED YMAX
 211        TRJEND=1
            IK=IK+1
            YDSP(IK)=YMAX
            IF(CPRED.EQ.O)GOTO 215
            GOTO 219
C END OF TRAJECTORY FLAGGED: EXCEEDED YMIN
 212        TRJEND=1
            IK=IK+1
            YDSP(IK)=YMIN
            IF(CPRED.EQ.O)GOTO 215
 219        YY=DBLE(YDSP(IK))
            S=5
            GOTO 233
C FIND X FOR HIGHER ORDER METHOD.
 215        XDSP(IK)=(X-XPREV)/(Y-YPREV)*(YDSP(IK)-YPREV)+XPREV
            GOTO 232
C END OF TRAJECTORY FLAGGED: EXCEEDED XMAX
 213        TRJEND=1
            IK=IK+1
            XDSP(IK)=XMAX
            IF(CPRED.EQ.O)GOTO 216
            XX=DBLE(XDSP(IK))
            S=4
            GOTO 233
C FIND Y FOR HIGHER ORDER METHOD.
 216        YDSP(IK)=(Y-YPREV)/(X-XPREV)*(XMAX-XPREV)+YPREV
 232        IF(TRJPRA.EQ.O.AND.AT.EQ.1)GOTO 234
            GOTO 231
C FIND X & Y FOR HERMITE EXTRAPOLATION.
 233        CALL WHAMO(XX,YY,TCOLL,S,ZZ,XCD,YCA)
            IF(IABS(S).EQ.4)YDSP(IK)=SNGL(YY)
            IF(IABS(S).EQ.5)XDSP(IK)=SNGL(XX)
            IF(TRJPRA.EQ.O.AND.AT.EQ.1)GOTO 180
            GOTO 231
C
C STORE PLOT COORDINATES FOR FIRST POINT WITHIN WINDOW
 226        IK=1
            XDSP(IK)=XMIN
            YDSP(IK)=(Y-YPREV)/(X-XPREV)*(XMIN-XPREV)+YPREV
            IF(CPRED.EQ.O)GOTO 230
            GOTO 120
C
C STORE COORDS FOR LATER PLOTTING
 221        IF(TRJPLA.EQ.O)GOTO 230
            IF(TRJPLA.EQ.1.AND.LAYER.GT.1)GOTO 230
            IK=IK+1
            XDSP(IK)=SNGL(XDS(IO(MM),MM))
            YDSP(IK)=SNGL(YDS(IO(MM),MM))
 230        IF(TRJPRA.EQ.O)GOTO 105
            IF(PRD.LT.PRDSTI.AND.SHORTP.EQ.O)GOTO 105
C
C PRINT INTERVAL EXCEEDED
 231        TTLACN=DSQRT(AN(1,IO(MM),MM)*AN(1,IO(MM),MM)+
            AN(2,IO(MM),MM)*AN(2,IO(MM),MM))
            VPSQ=UDS(IO(MM),MM)*UDS(IO(MM),MM)+VDS(IO(MM),MM)*VDS(IO(MM),MM)
            NA=RDS*TTLACN/DTS(IO(MM),MM)/VPSQ
C
C WRITE TRAJECTORY INFO INTO STORAGE FILE.
            IMN1=I(MM)-1
```

```
          IF(PC.EQ.1.AND.I(MM).GT.4)GOTO 235
C FOR RK4,RKF4 & GLERK5
          IF(SHORTP.EQ.O)WRITE(7,50)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),UDS
          (IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,HFP
          IF(SHORTP.EQ.1)WRITE(7,55)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),UDS
          (IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,HFP
          IF(TRJEND.EQ.O)GOTO 100
          GOTO 225
C FOR PC4
235       IF(SHORTP.EQ.O)WRITE(7,50)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),
          UDS(IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,
          HFP,UST,VST
          IF(SHORTP.EQ.1)WRITE(7,55)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),
          UDS(IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,
          HFP,UST,VST
          IF(TRJEND.EQ.O)GOTO 100
C END OF TRAJECTORY INFO
225       WRITE(7,50)I(MM),TS(I(MM)+1,MM),DTS(IP1(MM),MM),X,Y
          IF(AT.EQ.1)GOTO 181
C WRITE END OF TRAJECTORY INFO ONTO TERMINAL
          IF(SHORTP.EQ.O)WRITE(6,50)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),
          UDS(IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,
          HFP
          IF(SHORTP.EQ.1)WRITE(6,55)IMN1,TS(I(MM),MM),DTS(IO(MM),MM),
          XDS(IO(MM),MM),YDS(IO(MM),MM),PSIP,UAS(IO(MM),MM),
          UDS(IO(MM),MM),VAS(IO(MM),MM),VDS(IO(MM),MM),RED(IO(MM),MM),NA,
          HFP
          WRITE(6,50)I(MM),TS(I(MM)+1,MM),DTS(IP1(MM),MM),X,Y
181       IF(TRJPLA.EQ.O)GOTO 180
          IF(TRJPLA.EQ.1.AND.LAYER.GT.1)GOTO 180
C
C PLOT TRAJECTORIES
234       XDSP(IK+1)=XMIN
          XDSP(IK+2)=(XMAX-XMIN)/20.0
          YDSP(IK+1)=YMIN
          YDSP(IK+2)=(YMAX-YMIN)/12.0
          CALL LINE(XDSP,YDSP,IK,1,0,0)
180       IF(SMASH.EQ.1)GOTO 195
          IF(IK.NE.O)GOTO 170
C WRITE CLOSEST APPROACH INFO.
          WRITE(6,60)MLAMBH,XCLAP,CLAP,I(MM),FNCALL,YDS(IP1(MM),MM)
          WRITE(7,65)MLAMBH,XCLAP,CLAP,I(MM),FNCALL,YDS(IP1(MM),MM)
          GOTO 196
170       WRITE(6,60)MLAMBH,XCLAP,CLAP,I(MM),FNCALL,YDSP(IK)
          WRITE(7,65)MLAMBH,XCLAP,CLAP,I(MM),FNCALL,YDSP(IK)
          GOTO 196
C WRITE COLLISION INFO
195       WRITE(6,80)XCOLL,YCOLL,LTH,I(MM),FNCALL,MLAMBH,VTTL,ACOLLD
          WRITE(7,85)XCOLL,YCOLL,LTH,I(MM),FNCALL,MLAMBH,VTTL,ACOLLD
196       IF(GLOBAL.EQ.O)GOTO 197
          IF(GER.EQ.1)GOTO 199
          IF(GLOBAL.EQ.2)GOTO 198
C CALCULATE ORDER EXTRAPOLATION GLOBAL ERROR.
          GEXX=XDS(IO(1),1)
          GEX=XDS(IO(1),1)-XDS(IP1(2),2)
          GEY=YDS(IO(1),1)-YDS(IP1(2),2)
          GEU=UDS(IO(1),1)-UDS(IP1(2),2)
          GEV=VDS(IO(1),1)-VDS(IP1(2),2)
          GOTO 199
C CALCULATE STEP EXTRAPOLATION GLOBAL ERROR.
198       GEXX=XDS(IM1(2),2)
```

```
              GEX=(XDS(IO(1),1)-XDS(IM1(2),2))/31.DO
              GEY=(YDS(IO(1),1)-YDS(IM1(2),2))/31.DO
              GEU=(UDS(IO(1),1)-UDS(IM1(2),2))/31.DO
              GEV=(VDS(IO(1),1)-VDS(IM1(2),2))/31.DO
      199     IF(AT.EQ.O)WRITE(6,98)GEXX,GEX,GEY,GEU,GEV
              WRITE(7,99)GEXX,GEX,GEY,GEU,GEV
      197     IF(AT.EQ.O)GOTO 490
              IF(GRAZE.EQ.O)GOTO 630
              IF(SMASH.EQ.1)GOTO 610
C
C ITERATE TOWARD THE GRAZING TRAJECTORY
              IF(IG.EQ.1)GOTO 600
              IF(CLAP/CLAPP.GE.1.DO)GOTO 605
C ***************************************************************
              IF(DFLOAT(2*IJ-3)*(CLAP+CLAPP).LE.2.D-5)K=K+O.1DO
C ***************************************************************
C FIND NEW YO POSITION BY USING THE SECANT METHOD TO ESTIMATE
C    THE LOCATION OF YO AT GRAZING
              SLP=(YOT(IG,IJ)-YOT(IG-1,IJ))/(CLAP-CLAPP)
              YOT(IG+1,IJ)=YOT(IG,IJ)-K*CLAP*SLP
              GOTO 606
C AFTER FIRST MISSING TRAJECTORY, ESTIMATE NEW YO VIA CLAP.
      600     YOT(1,IJ)=YOI(IJ,J)
C ESTIMATE NEW YO VIA CLAP ALONE.
C ***************************************************************
      605     YOT(IG+1,IJ)=YOT(IG,IJ)-O.95DO*CLAP
C ***************************************************************
      606     YOG=YOT(IG,IJ)
              CLAPP=CLAP
              KLAP=CLAP
              IG=IG+1
              YDS(1,1)=YOT(IG,IJ)
              GOTO 405
C
C THESE ARE COLLIDING TRAJECTORIES.  ARE THEY THE GRAZING ONE?
      610     IF(IG.GT.1)GOTO 620
              WRITE(6,90)
C ADJUST FIRST TRAJECTORY TO BE A NEAR MISS
              YOI(IJ,J)=YOI(IJ,J)+DSIGN(5.D-4,DFLOAT(2*IJ-3))
              YDS(1,1)=YOI(IJ,J)
              GOTO 405
C WAS LAST TRAJECTORY ALMOST GRAZING?
C ***************************************************************
      620     IF(DABS(KLAP).LT.1.5D-5)GOTO 625
C ***************************************************************
C IS ANGLE OF COLLISION CLOSE TO 90 DEG.?
C ***************************************************************
              IF(90.DO-DABS(ACOLL)/PI*1.8D2.LE.TD)GOTO 625
C ***************************************************************
              TD=TD+O.1DO
C ***************************************************************
C THE ANGLE OF COLLISION ISN'T CLOSE ENOUGH TO 90 DEG.
C    TRY AGAIN MIDWAY BETWEEN PREVIOUS TWO TRAJECTORIES.
C ***************************************************************
              K=K-O.05DO
C ***************************************************************
              YOT(IG,IJ)=(YOT(IG,IJ)+YOG)/2.DO
              YDS(1,1)=YOT(IG,IJ)
              GOTO 405
C THIS IS THE GRAZING TRAJECTORY.
      625     YO(IJ)=YOT(IG,IJ)
              L(IJ)=LTH
              VTOT(IJ,J)=VTTL
              ACOL(IJ,J)=ACOLL
              IF(BOTH.EQ.1.AND.IJ.EQ.2)GOTO 460
              RETURN
```

```
C
C THESE ARE COLLIDING TRAJECTORIES
630    IF(SMASH EQ 1)GOTO 635
       WRITE(6,95)
       RETURN
C
635    LW(N)=LTH
       VTW(N)=VTTL
       ACW(N)=ACOLL
       RETURN
       END
C
C ****************************************************************
C
       SUBROUTINE WHAMO(X,Y,T,S,CLAP,XCD,YCA)
C
C WRITTEN BY: M. OLESKIW  ON: 810623  LAST MODIFIED: 811018
C
C DETERMINE CLOSEST APPROACH BETWEEN DROPLET AND AIRFOIL SFC.
C
      . DOUBLE PRECISION DSIGN,DSQRT,AX,AY,A1,A2,A3,BX,BY,TST,
      .CLAP,CX,CY,DSQ,Q,R,RDS,DD(5),C,SLOPE,SS,T,TT,DFLOAT,
      .X,XCD,XN,YN,XO,YO,Y,YCA,YCD,YS1,YS2,ZZ,D,THETA3,DARCOS,
      .SQ,RT(4),ALPHAR,PI,T1,DCOS,TO,A31,DABS,DMIN1,DMAX1,
      .XL,XR,XUXR,XLXR
C
       INTEGER IABS,ISIGN,TYPE,J,S,AS,IR(4),LA,I
C
       COMMON ALPHAR,PI/WH/AX,BX,CX,AY,BY,CY,XO,YO,TO,T1
      ./TRANS3/DD,C,TYPE,J/NOSE/XN,YN/XXR/XUXR,XLXR
C
C IN/OUT X=DROPLET X COORD
C IN/OUT Y=DROPLET Y COORD.
C IN/OUT T=TIME AT ABOVE POSITION.
C IN/OUT S=1:GIVEN X, Y & T, FIND CLAP
C           2:GIVEN X, FIND T, Y & CLAP.
C           3:GIVEN Y, FIND T, X & CLAP.
C           4:GIVEN X, FIND T & Y.
C           5:GIVEN Y, FIND T & X.
C           +VE:DROPLET IS ABOVE NOSE
C           -VE:DROPLET IS BELOW NOSE
C OUT    CLAP=CLOSEST APPROACH BETWEEN DROP & AIRFOIL SFC.
C OUT    XCD=
C OUT    YCA=X & Y COORDS OF AIRFOIL AT CLOSEST APPROACH.
C
       RDS=DD(J)/C/2.D6
       AS=IABS(S)
       IF(AS.EQ.1)GOTO 200
       IF(AS.EQ.2.OR.AS.EQ.4)GOTO 150
C
C FIND COEFFICIENTS FOR Y EQN.
       A1=BY/AY
       A2=CY/AY
       A3=(YO-Y)/AY
       GOTO 110
C FIND COEFFICIENTS FOR X EQN.
150    A1=BX/AX
       A2=CX/AX
       A3=(XO-X)/AX
C FIND TIME TST
110    Q=(3.DO*A2-A1*A1)/9.DO
       R=(9.DO*A1*A2-27.DO*A3-2.DO*A1**3)/54.DO
       D=Q**3+R*R
       A31=A1/3.DO
       IF(D.GT.O)GOTO 310
```

```
C FOR A NEGATIVE DISCRIMINANT:
      THETA3=DARCOS(R/DSQRT(-Q**3))/3.DO
      SQ=2.DO*DSQRT(-Q)
      RT(1)=0.99DO*(T1-TO)
      RT(2)=SQ*DCOS(THETA3)-A31
      RT(3)=SQ*DCOS(THETA3+2.DO/3.DO*PI)-A31
      RT(4)=SQ*DCOS(THETA3+4.DO/3.DO*PI)-A31
        DO 315 I=1,4
        IR(I)=I
315     CONTINUE
C SORT FOR LEAST ROOT GREATER THAN TS(I(MM),MM).
      LA=4
      CALL VSRTRD(RT,LA,IR)
      I=1
330   IF(IR(I).EQ.1)GOTO 320
      I=I+1
      GOTO 330
320   TST=RT(I+1)
      GOTO 340
C FOR A POSITIVE DISCRIMINANT:
310   DSQ=DSQRT(D)
      SS=DSIGN((DABS(R+DSQ))**(1.DO/3.DO),R+DSQ)
      TT=DSIGN((DABS(R-DSQ))**(1.DO/3.DO),R-DSQ)
      TST=SS+TT-A31
340   T=TST+TO
      IF(AS.EQ.2.OR.AS.EQ.4)GOTO 160
C
C POSITION OF DROP CENTRE AT TIME T
      X=((AX*TST+BX)*TST+CX)*TST+XO
      GOTO 200
160   Y=((AY*TST+BY)*TST+CY)*TST+YO
200   IF(AS.GE.4)RETURN
C SET S NEGATIVE FOR BELOW NOSE.
      IF(Y.LT.YN)S=-IABS(S)
C
300   XL=DMAX1(X,XN+1.D-10)
      XR=DMIN1(X+RDS,XUXR,XLXR)
      CALL SFC(XL,YS1,S,O,ZZ)
      CALL SFC(XR,YS2,S,O,ZZ)
      SLOPE=DSQRT((XR-XL)**2+(YS2-YS1)**2)
C FIND DROPLET X & Y COORDS. OF CLOSEST APPROACH.
      YCD=Y-DSIGN((XR-XL)*RDS/SLOPE,DFLOAT(S))
      XCD=DMAX1(XN,X-ISIGN(1,S)*RDS*(YS1-YS2)/SLOPE)
C FIND AIRFOIL Y COORD. AT CLOSEST APPROACH.
      XCD=DMAX1(XCD,XN+1.D-10)
      XCD=DMIN1(XCD,XUXR,XLXR)
      CALL SFC(XCD,YCA,S,O,ZZ)
C CLOSEST APPROACH
      CLAP=YCD-YCA
      RETURN
      END
```

APPENDIX H. Program tolerances, adjustments and options.

Chapter 3 described a sequence of trials that were used to estimate the optimum values of the subset of tolerances and options which the program requires as user input. It also mentioned that another set of tolerances and adjustments are built into the program because they should not require frequent modification. The first section of this appendix describes these built-in tolerances and adjustments. The second section is a listing of the complete set of input parameters in the exact format required by the program for each of the cases mentioned in Chapters 3, 4, and 5.

All of the locations where adjustments may be made to tolerances and to algorithms in the program have been indicated by a row of '*'s before and after the line of interest. The references to program location are by the convention adopted in the previous chapter, that is by the internal statement number of a given subroutine as listed in Appendix G. The important adjustment options are outlined below, in approximately the same order as they would be encountered during a routine execution of the program. Their present values have been chosen through a process of trial and error.

1. Finding appropriate values for e and θ in (2.13) to produce a Joukowski airfoil of desired thickness (see Section 2.2.2.2). Tolerances for the subroutine ZXGSN are set at COORD[73] and [90] for θ. A tolerance for ending the Secant algorithm used to find the appropriate value of e is set at COORDS[95].
2. Inverting the system of equations to give the vorticity density in the Kennedy and Marsden technique (see Section 2.2.4). The inverting subroutine LEQT1F tests for the accuracy of the solution to $A\bar{x} = \bar{b}$ by changing elements of A after IDGT decimal places and determining if the resulting solution is near the original. IDGT is set at POT1[54].
3. Finding ordinate values of interpolated points along the airfoil surface. A Newton-Raphson algorithm is used to iterate to the correct value of y_R according to formulae of Appendix F. The tolerance for deciding when to stop the iterations is found at SFC [43] and [86].
4. Determining the grazing trajectories (see Section 2.4.4). The secant algorithm is used to iterate to the grazing trajectory. The algorithm is modified by the

parameter k in (2.86), which is set at TRAJEC[104], [485] and [505]. When a droplet passes to within 10^{-5} chord lengths from the airfoil surface on two consecutive trajectories, the rate of convergence is accelerated. This tolerance is set at TRAJEC[485]. The rate of convergence after the first trajectory (before the Secant algorithm may be employed) is set at TRAJEC[490]. If the previous trajectory passed within 1.5×10^{-5} chord lengths of the airfoil surface and the present one ends in a collision, this is considered to be the grazing trajectory. This tolerance is set at TRAJEC[502]. On the other hand, if the angle between the tangent to the trajectory and the normal to the airfoil surface is close enough to 90°, this is the grazing trajectory. This tolerance is set at TRAJEC[503] and [504].

5. Proximity of approach which is defined as a collision (see Section 2.4.3): If the droplet surface approaches the airfoil to within 0.1% its radius, it is deemed to have collided. This tolerance is set at TRAJEC[340].

6. Finding the collision location (see Section 2.4.3): The Secant algorithm is used to iterate toward the collision location as in Fig. 6 until the difference in x between two iterations is less than 10^{-4} (TRAJEC[370]).

7. Adjusting the time step in RKF4 conservatively (see Appendix B). This is done to reduce the likelihood that the step size chosen will be too large, thereby causing the estimated trucation error to exceed the tolerance (GLERK[94, 111]).

8. Adjustment of the next time step depending upon the previous and current step sizes (see Appendix B): This is required to damp the undesirable oscillations in the step which are chosen automatically. This adjustment also occurs in the interval GLERK5[94, 111].

9. Stepping over difficult integrating regions (see Appendix B). Occasionally, the air velocity will change so rapidly in a short distance that the current time step cannot be made small enough for all component equations to satisfy the truncation error tolerance because of small errors in calculating the air velocity. When such situations occur, the tolerance is bypassed. This situation is detected in GLERK5[121].

10. The distance between trajectory pairs for calculating β (see Section 2.4.5.1): The

greater the distance between the pair of trajectories, the greater is the computational accuracy. However, at the same time the β value derived becomes averaged over a greater interval, thereby losing accuracy if the slope of the β curve in that region is changing rapidly. The distance between the pairs is set at CE[35] and [44].

11. Locating the first trajectory pair within the grazing trajectory envelope (see Section 2.4.5.2). Experiments have shown that the position of the first point on the β curve can greatly affect the rapidity with which the procedure used to determine the β curve converges to a consistent shape. These locations are calculated at CE[38,39].

12. Adjusting the variable filter length algorithm (see Section 2.4.5.4). The constants used in these formulae are adjustable to give the "best" agreement between the averaged and smoothed β curves. The filter length is calculated at CE[355] and [358].

13. The position of the new nose after accretion (see Section 2.4.7). The Golden Section search algorithm ZXGSN needs to know when to stop searching for the new nose position. This criterion is located at ICING[271].

14. Finding CEE's on the new airfoil surface (see Section 2.4.9). The parameters for determining which SSE's also become CEE's are located at ICING[430], [434], [441], [453,455], [464], [488,490] and [499].

Case 1.

```
ALPHA,TYPE,THICK,MEAN,NE .NEB,NIF,ANAL,PLTFAC,
 4 60.  -2.12 00. 000. 11. 10.  6.   1.1 0000.
  UINF.    C.  TINF. PINF,VINO.
128.60.0 711,-20.00,101 3.    0.
TRJPLA.XMIN.XMAX.YMIN.YMAX.XZ.YZ.XMINI.XMAXI.YMINI.YMAXI.
     0.- 16.0.24.- 16.0.08.51.61.-0.06. 0 24.-0 12. 0 06.
EON.PC.ACN.GRAV.CDS.TRJPRA.PRINTO.PRINTI.
  2. 2.  0.   0.  1.     1.    2.    30.
DDISTN.DD(1).W(1).DD(2).W(2).DD(3).W(3).DD(4).W(4).DD(5).W(5).
     5. 35.0.0 20. 25 4.0 20. 20.0.0 20. 15 4.0 20. 10.0.0.20.
EPS(1).EPS(2).EPS(3).EPS(4).EPS(5).
 1.D-5. 2.D-5. 3 D-5. 4.D-5. 5 D-5.
AT.CEDEL.EMDEL.HS.YOL.CEL.CEX.FILTER.LLEFT.LRIGHT.
 1.  1.0.  1.0. 1.  0.  3.  0.   0.0.   0.0.    0.0.
ICEPLA.LYRMAX.   ICE.LTOL.ATHICK.DENSE.
     1.      1.0.050.1 15.      1.    0.
XO(1).   XO(5).
-5 .-5 .-5 .-5 .-5 .
YO(1). . .YO(5).   (UPPER SFC )
-0.49550.-0.51200.-0.52350.-0.53440.-0.54900.
YO(1). . .YO(5).   (LOWER SFC.)
-0.56650.-0.56720.-0.56720.-0.56720.-0.56700.
END OF FILE
```

Case 2.

```
ALPHA.TYPE.THICK.MEAN.NEF.NEB.NIF.ANAL.PLTFAC.
 4 60.  -2.12 00. 000.  6. 14. 10.   1.1.0000.
  UINF.    C.  TINF. PINF.VINO.
128 60.0 711.-20 00.101 3.    0.
TRJPLA.XMIN.XMAX.YMIN.YMAX.XZ.YZ.XMINI.XMAXI.YMINI.YMAXI.
    -1.- 16.0 24.  16.0 08.51.61.-0.06. 0 24.-0.12. 0.06.
EON.PC.ACN.GRAV CDS.TRJPRA.PRINTO.PRINTI.
  2. 2   0.   0.  1.     1.    2.    30.
DDISTN.DD(1).W(1).DD(2).W(2).DD(3).W(3).DD(4).W(4).DD(5).W(5).
     5. 35 0.0 20. 25 4.0 20. 20 0.0.20. 15.4.0.20. 10.0.0.20.
EPS(1).EPS(2).EPS(3).EPS(4).EPS(5).
 1 0-5. 4 D-5. 1 D-4. 8 D-5. 6 D-5.
AT.CEDEL.EMDEL.HS.YOL.CEL.CEX.
 1.  1 0.  1 0. 1.  0.  1.  0.
ICEPLA.LYRMAX.   ICE.LTOL.ATHICK.DENSE.
     1.      1.0 050.1 01.      1.    0.
XO(1).   XO(5).
-5 .-5 . 5 .-5 .-5 .
YO(1).   .YO(5).   (UPPER SFC )
-0 49550.-0 51200. 0 52350.-0 53440.-0 54900.
YO(1).   .YO(5).   (LOWER SFC )
-0 56650.-0 56720. 0 56720.-0 56720.-0.56700.
END OF FILE
```

Case 3.

```
ALPHA.TYPE.THICK.MEAN.NEF.NEB.NIF.ANAL.PLTFAC.
 4 60.  -2.12 00. 000.  6. 14. 10.   1.1.0000.
  UINF.    C.  TINF. PINF.VINO.
128 60.0.711.-20.00.101.3.    0.
TRJPLA.XMIN.XMAX.YMIN.YMAX.XZ.YZ.XMINI.XMAXI.YMINI.YMAXI.
     0.- 16.0 24.- 16.0 08.51.61.-0 06. 0.24.-0.12. 0 06.
EON.PC.ACN.GRAV.CDS.TRJPRA.PRINTO.PRINTI.
  2. 2.  0.   0.  1.     1.    2.    30.
DDISTN.DD(1).W(1).DD(2).W(2).DD(3).W(3).DD(4).W(4).DD(5).W(5).
     3. 35.0.0 20. 20.0.0.60. 10.0.0.20. 15 4.0.20. 10.0.0.20.
EPS(1).EPS(2).EPS(3).EPS(4).EPS(5).
 1 0-5. 1.0-4. 6.0-5. 8 0-5. 6.0-5.
AT.CEDEL.EMDEL.HS.YOL.CEL.CEX.
 1.  1.0.  1.0. 1.  0.  1.  0.
ICEPLA.LYRMAX.   ICE.LTOL.ATHICK.DENSE.
     1.      1.0 050.1 01.      1.    0.
XO(1).   XO(5).
-5 .-5 .-5 .-5 .-5 .
YO(1).   .YO(5).   (UPPER SFC.)
-0 49550.-0 52350.-0 54900.-0.53440.-0.54900.
YO(1).   .YO(5).   (LOWER SFC.)
-0 56650.-0 56720.-0 56700.-0.56720.-0.56700.
END OF FILE
```

Case 4.

```
ALPHA.TYPE.THICK.MEAN.NEF.NEB.NIF.ANAL.PLTFAC.
 4 60.  -2.12 00. 000.  6. 14. 10.   1.1 0000.
  UINF.    C.  TINF. PINF.VINO.
128 60.0 711. 20 00.101 3.    0.
TRJPLA.XMIN.XMAX.YMIN.YMAX.XZ.YZ.XMINI.XMAXI.YMINI.YMAXI.
     0.- 16.0 24.  16.0 08.51.61.-0 06. 0.24.-0 12. 0.06.
EON.PC.ACN.GRAV.CDS.TRJPRA.PRINTO.PRINTI.
  2. 2.  0.   0.  1.     1.    2.    30.
DDISTN.DD(1).W(1).DD(2).W(2).DD(3).W(3).DD(4).W(4).DD(5).W(5).
     2. 35 0.0 20. 18 0.0 80.
EPS(1).EPS(2).EPS(3).EPS(4).EPS(5).
 1 D-5. 8 D 5.
AT.CEDEL.EMDEL.HS.YOL.CEL.CEX.
 1.  1 0.  1 0. 1.  0   1.  0.
ICEPLA.LYRMAX.   ICE.LTOL.ATHICK.DENSE.
     1.      1.0 050.1.20.      1.    -0.
XO(1).   XO(5).
-5 . 5 .
YO(1).   .YO(5).   (UPPER SFC )
-0 49550.-0 52800.
YO(1).   .YO(5).   (LOWER SFC )
-0 56650.-0 56720.
END OF FILE
```

Case 5.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,  -2,12.00, 000,  6, 14, 10,   1,1.0000,
  UINF,     C,   TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 16,0.24,- 16,0.08,51,61,-0.06, 0.24,-0.12, 0.06,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 32.0,0.30, 16.6,0.70,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 4.D-5, 4.D-5,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,
 1,  1.0,   1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
    1,       1,0.050,1.20,       1,     0,
XO(1),    XO(5),
-5.,-5.,
YO(1),    ,YO(5),   (UPPER SFC.)
-0.50030,-0.53140,
YO(1),    ,YO(5),   (LOWER SFC.)
-0.56670,-0.56720,
END OF FILE
```

Case 6.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,  -2,12.00, 000,  6, 14, 10,   1,1.0000,
  UINF,     C,   TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,YMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,  16,0.24,- 16,0.08,51,61, 0.06, 0.24,-0.12, 0.06,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 28.8,0.40, 15.4,0.60,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 4.D-5, 4.D-5,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,
 1,  1.0,   1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
    1,       1,0.050,1.20,       1,     0,
XO(1),    XO(5),
-5.,-5.,
YO(1),    ,YO(5),   (UPPER SFC.)
-0.50580,-0.53440,
YO(1),    ,YO(5),   (LOWER SFC.)
-0.56700,-0.56720,
END OF FILE
```

Case 7.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,  -2,12.00, 000, 11, 10,  6,   1,1.0000,
  UINF,     C,   TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 16,0.24,- 16,0.08,51,61,-0.06, 0.24,-0.12, 0.06,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 27.4,0.50, 14.2,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1.D-5, 4.D-5,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,   1.0, 1,  0,  3,  0,  0.10,-0.23, 0.035,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
    1,       1,0.050,1.15,       1,     0,
XO(1),    ,XO(5),
-5.,-5.,
YO(1),    ,YO(5),   (UPPER SFC.)
-0.50820,-0.53750,
YO(1),    ,YO(5),   (LOWER SFC.)
-0.56700,-0.56700,
END OF FILE
```

Case 8.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,  -2,12.00, 000, 11, 10,  6,   1,1.0000,
  UINF,     C,   TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,  16,0.24,- 16,0.08,51,61,-0.06, 0.24, 0.12, 0.06,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      0,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 27.1,0.50, 14.2,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1.D-5, 4.D-5,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,   1.0, 1,  0,  3,  0,  0.10,-0.23, 0.035,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
    1,       1,0.050,1.15,       1,     0,
XO(1),    XO(5),
5.,  5.,
YO(1),    ,YO(5),   (UPPER SFC.)
-0.50820,-0.53750,
YO(1),    ,YO(5),   (LOWER SFC.)
-0.56700,-0.56700,
END OF FILE
```

Case 9.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,   -2.12 00, 000,  6, 14, 10,    1,1.0000,
  UINF,     C,  TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 16.0 24.- 16.0 08,51,61,-0.06, 0.24,-0.12, 0.06,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      0,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1.D-4,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  3,  0,  0.20,-0.23, 0.025,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.15,      1,     0,
XO(1),...XO(5),
-5.,
YO(1),...,YO(5),  (UPPER SFC.)
-0.52350,
YO(1),....,YO(5),  (LOWER SFC.)
-0.56720,
END OF FILE
```

Case 10.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 1.60,   2.12 00, 000,  6, 14, 10,    1,1.0000,
  UINF,     C,  TINF, PINF,VINO,
128.60,0.711, 20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,  16,0.24,  16,0.08,51,61,-0.06, 0.24,-0.12, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      0,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1.D-1,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  3,  0,  0.20,-0.23, 0.035,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.18,      1,     0,
XO(1),   XO(5),
 5.,
YO(1),    YO(5),  (UPPER SFC.)
-0.52350,
YO(1),  ,YO(5),  (LOWER SFC.)
 0.56720,
END OF FILE
```

Case 11.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,   2,12.00, 000, 14, 11,  4,    1,1.0000,
  UINF,     C,  TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 16.0.24,- 16.0.08,51,61,-0.06, 0.24,-0.12, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
6.D-4,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.20,      1,     0,
XO(1),...XO(5),
-5.,-5.,
YO(1),....,YO(5),  (UPPER SFC.)
-0.52350,
YO(1),...,YO(5),  (LOWER SFC.)
-0.56720,
END OF FILE
```

Case 12.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,   2.12.00, 000, 14, 10, 05,    1,1.0000,
  UINF,     C,  TINF, PINF,VINO,
128.60,0.711,-20.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 16.0 24.- 16.0 08,51,61,-0.08, 0.24,-0.12, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,      1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-4,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.20,      1,     0,
XO(1),   XO(5),
-5.,-5.,
YO(1),   ,YO(5),  (UPPER SFC.)
-0.52340,
YO(1),  ,YO(5),  (LOWER SFC.)
-0.56735,
END OF FILE
```

Case 13.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINO,
   5.67, .0302,-10.00, 78.5,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0.50.0.00.0 60.51.61,-0.50, 0.50, 0.00, 0.60.
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
   1, 2,  0,   0,  1,     1,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       1,0.050,1.20,      1,    0,
XO(1),   XO(5),
-10.,
YO(1),   ..YO(5),  (UPPER SFC )
0.09130,
YO(1),   ..YO(5),  (LOWER SFC )

END OF FILE
```

Case 14.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,    1,100 0, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,   TINF, PINF,VINO,
   5.67, .0302, 10.00, 78.5,    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1, 50.0.50.0.00.0.60.51.61,-0.50, 0.50, 0.00, 0.60.
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
   2, 2,  0,   0,  1,     1,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       1,0.050,1.20,      1,    0,
XO(1),   XO(5),
-10.,
YO(1),   ..YO(5),  (UPPER SFC )
0.09130,
YO(1),   ..YO(5),  (LOWER SFC )

END OF FILE
```

Case 15.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINO,
   71.7,1.194,-10.00, 78.5,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0.50.0.00.0.60.51.61,-0.50, 0.50, 0.00, 0.60.
FQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
   1, 2,  0,   0,  1,     1,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1,200.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-7,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       1,0.050,1.20,      1,    0,
XO(1),   XO(5),
-10.,
YO(1),   ..YO(5),  (UPPER SFC )
0.30650,
YO(1),   ..YO(5),  (LOWER SFC )

END OF FILE
```

Case 16.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,    1,1.0000,
  UINF,    C,   TINF, PINF,VINO,
   71.7,1.191,-10.00, 78.5,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1.- 50.0.50.0.00.0.60.51.61, 0.50, 0.50, 0.00, 0.60.
FQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
   2, 2,  0,   0,  1,     1,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1,200.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-7,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       1,0.050,1.20,      1,    0,
XO(1),   XO(5),
-10.,
YO(1),   ..YO(5),  (UPPER SFC )
0.30650,
YO(1),   ..YO(5),  (LOWER SFC )

END OF FILE
```

Case 17.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1.100 0.000, 11, 10, 06.    1,1 0000.
  UINF.     C.   TINF, PINF,VINQ,
  80.2 0.214,-10.00, 78.5,    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0.50.0 00.0 60.51.61, 0.50, 0.50, 0.00, 0.60.
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2.  0.    0.  1.     1.     2.     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1. 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0.  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1.        1,0 050.1.20.      1.     0.
XO(1),   .XO(5).
-10 .
YO(1),  ..YO(5).   (UPPER SFC.)
0.09,
YO(1),   ..YO(5).   (LOWER SFC.)

END OF FILE
```

Case 18.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1.100 0.000, 11, 10, 06.    1,1 0000,
  UINF.     C.   TINF, PINF,VINQ,
  80.2.0.214,-10.00, 78.5,    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
     1.- 50.0.50.0 00.0 60.51.61, 0.50, 0.50, 0.00, 0.60,
EQN,FC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2.  0.    0.  1.     1.     2.     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1. 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0.  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1.        1,0 050.1 20.      1.     0.
XO(1),   XO(5).
-10 .
YO(1),   ..YO(5).   (UPPER SFC )
0.09,
YO(1),   ..YO(5).   (LOWER SFC )

END OF FILE
```

Case 19.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1.100 0.000, 11, 10, 06.    1,1 0000.
  UINF.     C.   TINF, PINF,VINQ,
  11.23, 1527,-10.00, 78.5,    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0 00.0 60.51.61,-0.50, 0.50, 0.00, 0.60.
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2.  0.    0.  1.     1.     2.     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1. 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
8.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0.  1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1.        1,0 050.1 20.      1.     0.
XO(1),   XO(5).
-10 .
YO(1),  ..YO(5).   (UPPER SFC.)
0.00403,
YO(1),....YO(5).   (LOWER SFC )

END OF FILE
```

Case 20.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1.100 0.000, 11, 10, 06.    1,1 0000.
  UINF.     C.   TINF, PINF,VINQ,
  11.23, 1527, 10.00, 78.5,    0.
TRJPLA,YMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1.  50.0 50.0 00.0 60.51.61,-0.50, 0.50, 0.00, 0.60.
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2.  0.    0.  2.     1.     2.     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1. 20.0.1.00,
EPS(1),EPS(2),EPS(3) EPS(4),EPS(5),
8.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  1,  0.
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1.        1,0 050.1 20.      1.     0.
XO(1),   XO(5).
-10..
YO(1),   .YO(5).   (UPPER SFC )
0.00240.
YO(1),   .YO(5).   (LOWER SFC.)

END OF FILE
```

Case 21.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1.100 0. 000. 11. 10. 06.    1.1 0000.
  UINF.     C.   TINF.  PINF,VINO.
  11 23.  1527.-10 00.  78.5.    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0 00.0 60.51.61.-0.50.  0.50.  0.00.  0.60.
EON.PC.ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI.
  2. 2.   0.    0.   1.    1.     2.    30.
DDISIN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5).
      1. 20 0.1 00.
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5).
8 D-6.
AT.CEDEL,FMDEL,HS,YOL,CEL,CEX.
 1.  1 0.   1.0.  1.   0.   1.   0.
ICEPLA,LYRMAX.   ICE,LTOL,ATHICK,DENSE.
     1.       1.0 050.1 20.      1.      0.
XO(1),.   XO(5).
-10.,
YO(1).      ,YO(5).   (UPPER SFC )
0 014D0.
YO(1).     ..YO(5).   (LOWER SFC )

END OF FILE
```

Case 22.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC.
 0 00.   -1.100 0. 000. 11. 10. 06.    1.1 0000.
  UINF.     C.   TINF.  PINF,VINO.
  14 44.  1187.  10 00.  78 5.    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0 00.0 60.51.61.-0.50.  0 50.  0.00.  0.60.
EON.PC.ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI.
  1. 2.   0.    0.   1.    1.     2.    30.
DDISIN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5).
      1.200 0.1 00.
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5).
8 D-9.
AT.CEDEL,FMDEL,HS,YOL,CEL,CEX.
 1.  1 0.   1 0.  1.   0.   1.   0.
ICEPLA,LYRMAX.   ICE,LTOL,ATHICK,DENSE.
     1.       1.0 050.1 20.      1.      0.
XO(1).    XO(5).
-10 .
YO(1).     ,YO(5).   (UPPER SFC )
0 44250.
YO(1).     ,YO(5).   (LOWER SFC )

END OF FILE
```

Case 23.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC.
 0.00,  -1.100 0. 000. 11. 10. 06.    1.1 0000.
  UINF.     C.   TINF.  PINF,VINO.
  14 44.  1187.-10 00.  78.5.    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0 00.0 60.51.61.-0.50.  0.50.  0.00.  0.60.
EON.PC.ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI.
  1. 2.   0.    0.   2.    1.     2.    30.
DDISIN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5).
      1.200 0.1 00.
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5).
8-D-9.
AT.CEDEL,EMDEL,HS,YOL,CEL,CEX.
 1.  1 0.   1 0.  1.   0.   1.   0.
ICEPLA,LYRMAX.   ICE,LTOL,ATHICK,DENSE.
     1.       1.0 050.1 20.      1.      0.
XO(1),.   XO(5).
-10.,
YO(1).     ,YO(5).   (UPPER SFC )
D 44250.
YO(1).     ,YO(5).   (LOWER SFC )

END OF FILE
```

Case 24.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC.
 0 00.   -1.100 0. 000. 11. 10. 06.    1.1 0000.
  UINF.     C.   TINF.  PINF,VINO.
  14 44.  1187.  10 00.  78 5.    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0 00.0 60.51.61.-0 50.  0 50.  0.00.  0.60.
EON.PC.ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI.
  2. 2.   0.    0.   1.    1.     2.    30.
DDISIN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5).
      1.200 0.1 00.
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5).
8 D-9.
AT.CEDEL,EMDEL,HS,YOL,CEL,CEX.
 1.  1 0.   1 0.  1.   0.   1.   0.
ICEPLA,LYRMAX.   ICE,LTOL,ATHICK,DENSE.
     1.       1.0 050.1 20.      1.      0.
XO(1).    XO(5).
-10 .
YO(1).     ,YO(5).   (UPPER SFC )
0 44250.
YO(1).     ,YO(5).   (LOWER SFC )

END OF FILE
```

Case 25.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   -2,36 50, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,  TINF, PINF,VINQ,
 12 84, .0532,-10 00, 78 5,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.,0.16,0.24,0.00,0.24,51,61,-0 06, 0.24, 0.00, 0.18,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1 0,   1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,        1,0.050,1.20,       1,     0,
XO(1),...XO(5),
-5.,
YO(1),...,YO(5),  (UPPER SFC.)
0.07350,
YO(1),...,YO(5),  (LOWER SFC.)
-0.56955,
END OF FILE
```

Case 26.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   -2.36 50, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,  TINF, PINF,VINQ,
 12 84, .0532, 10 00, 78 5,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 16.0 24,0 00,0 24,51,61,-0 06, 0 24, 0.00, 0.18,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,   1 0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,        1,0 050,1 20,       1,     0,
XO(1),...XO(5),
-5 ,
YO(1),...,YO(5),  (UPPER SFC )
0 07580,
YO(1),...,YO(5),  (LOWER SFC )

END OF FILE
```

Case 27.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.60,    0.12 00, 000, 11, 10, 07,    1,1.0000,
  UINF,     C,  TINF, PINF,VINQ,
128.60,0.711,-[illegible].70,101 3,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 16,0 24,-.16,0.08,51,61,-0.06, 0.24,-0 12, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-4,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1 0,   1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,        1,0.050,1 20,       1,     0,
XO(1),...XO(5),
-5.,
YO(1),....,YO(5),  (UPPER SFC.)
-0.52715,
YO(1),....,YO(5),  (LOWER SFC.)
-0.56955,
END OF FILE
```

Case 28.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4 60,    0.12 00, 000, 11, 10, 07,    1,1.0000,
  UINF,     C,  TINF, PINF,VINQ,
128 60,0.711, 08 70,101.3,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 16,0 24,- 16,0 08,51,61,-0 06, 0 24,-0.12, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20 0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D 4,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1 0,   1.0, 1,  0,  1,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,        1,0.050,1.20,       1,     0,
XO(1),...XO(5),
-5.,
YO(1),....,YO(5),  (UPPER SFC.)
-0.52490,
YO(1),....,YO(5),  (LOWER SFC.)
-0 56850,
END OF FILE
```

Case 29.

```
ALPHA,TYPE,THICK,MEAN,NEF,NFB,NIF,ANAL,PLTFAC,
 8.00,   0, 15.0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C.  TINF, PINF,VINQ,
 30.50,0 159,-31 00, 97 7,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,*
     1,- 10.0 30,-.16,0.08,51,61,-0 10, 0 30, -.16, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2,  0,   0,  1,    1,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1,20.00,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
7 D-5,
AT,CFDEL,FMDEL,HS,YOL,CEL,CEX,
 1,  1 0,  1.0, 1,  0,  1,  0.
ICEPLA,LYRMAX   ICE,LTOL,ATHICK,DENSE,
     1,     1,0 020,1 20,     1,    0.
XO(1),.   XO(5),
-5 ,
YO(1),...,YO(5),  (UPPER SFC.)
-0 92135,
YO(1),...,YO(5),  (LOWER SFC )
-0 99780,
END OF FILE
```

Case 30.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 8 00,   0, 15 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,  TINF, PINF,VINQ,
 61.00,0.213,-31.00, 97 0,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
     1 -.10,0 40,-.20.0 10,51,61,-0 08, 0 32, 0 12, 0 12,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2,  0,   0,  1,    1,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1,20.00,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-4,
AT,CEDEL,FMDEL,HS,YOL,CFL,CEX,
 1,  1 0,  1 0, 1,  0,  3,  1.
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1,     1,0 050,1 20,     1,    0.
XO(1),   XO(5),
-5 ,
YO(1),...,YO(5),  (UPPER SFC )
-0.90500,
YO(1),...,YO(5),  (LOWER SFC )
-0.99550,
END OF FILE
```

Case 31.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0 00,  -2, 15 0  000, 11, 10, 06,   1,1.0000,
  UINF,    C.  TINF, PINF,VINQ,
 78.20,0.330,  10 0, 95 1,   0.
TRJPLA,YMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
     0,-.20,0 30,-.20.0 10,51,61,-0.06, 0.24,-0 09, 0 09,
EQN PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    1,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     2, 25 5,0.50, 13 2,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
3 D-6,3 D-5,
AT,CFDEL,FMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,   0 0,-.149, 0.149,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1,     1,0 050,1.15,     1,    0.
XO(1),   XO(5),
-5 ,.5 ,
YO(1),...,YO(5),  (UPPER SFC.)
0 03600,0 01880,
YO(1),   ,YO(5),  (LOWER SFC.)

END OF FILE
```

Case 32.

```
ALPHA TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0 00,    2  15 0, 000, 11, 10, 06,   1,1 0000,
  UIN ,     C   TINF, PINF,VINQ,
 78 20 0 330    10 0, 95 1,   0,
TRJPLA XMIN XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
     0   20,0.30,- 20.0 10,51,61,-0 06, 0 24,-0 09, 0 09,
EQN PC ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     2  25 5,0 50, 13 2,0 50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
3 D-6,3 D-5
AT,CFDEL FMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,  0 20,- 149, 0.149,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
     1,     1,0.050,1 15,     1,    0,
XO(1),   XO(5),
-5 , 5 ,
YO(1),...,YO(5),  (UPPER SFC.)
0 01600,0 01880,
YO(1),...,YO(5),  (LOWER SFC.)
```

Case 33.

```
ALPHA,TYPE,THICK,MEAN,NEF,NFB,NIF,ANAL,PLTFAC,
 0.00,  -2. 15.0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,  TINF, PINF,VINQ,
 78 20.0 330,  10.0, 95.1,   0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,-.20,0.30,- 20.0 10.51,61,-0.06, 0.24,-0.09, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0.   0,  1,    0,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
        1, 18.6,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5.D-6,
AT,CEDEL,EMDEL,H5,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
  1,  1.0,   1.0, 1,  0,  3,  0,  0.30,-.149,  .149,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 050,1 15,       1,     0,
XO(1),    XO(5),
-5 ,
YO(1),. . ,YO(5),  (UPPER SFC.)
0.02790,
YO(1),  ..,YO(5),  (LOWER SFC.)

END OF FILE
```

Case 34.

```
ALPHA,TYPE,THICK,MEAN,NEF,NFB,NIF,ANAL,PLTFAC,
 0.00,  -2.15.00, 000, 11, 10,  6,    1,1.0000,
  UINF,    C,  TINF, PINF,VINQ,
 78 20.0 330, 10.00, 95.1,    0.
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.-.20,0.30,- 20.0 10.51,61,-0.06, 0.24,-0.09, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  1, 2,  0.   0.  1,     0,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       2, 25 5,0.50, 13 2,0 50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 3 D-6,3.D-5,
AT,CEDEL,EMDEL,H5,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
  1,  1.0,   1.0, 1,  0,  3,  0,  0.20,-.149, 0.149,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 050,1 15,       1,     0,
XO(1),   XO(5),
-5 ,-5 ,
YO(1),   ,YO(5)   (UPPER SFC.)
0.03500,0.01800,
YO(1),   ,YO(5),  (LOWER SFC.)

END OF FILE
```

Case 35.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.00,  -2. 15.0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,  TINF, PINF,VINQ,
  78.2,0.330, 10.00, 95.1,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,-.20,0.30,- 20.0,10.51,61,-0.06, 0.24,-0.09, 0.09,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    1,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       2, 25.1,0.50, 13.4,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5.D-6,9.D-5,
AT,CEDEL,EMDEL,H5,YOL,CEL,CEX,
  1,  1.0,   1.0, 1,  0,  3,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 050,1.20,       1,     0,
XO(1),...XO(5),
-5.,-5.,
YO(1),....YO(5),  (UPPER SFC.)
-0.42800,-0.46000,
YO(1),....YO(5),  (LOWER SFC.)
-0.50550,-0.50100,
END OF FILE
```

Case 36.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 1.00,  -2. 15.0, 000, 11, 10, 06,    1,1.0000,
  UINF,    C,  TINF, PINF,VINQ,
 78 20.0 330,  10.0, 95.1,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.-.20,0.30,- 20.0 10.51,61,-0.06, 0.24,-0.09, 0.09,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,    2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       2, 25 5,0.50, 13.2,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5.D-6,1.D-5,
AT,CEDEL,EMDEL,H5,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
  1,  1.0,   1.0, 1,  0,  3,  0,  0.10,-.224,  .074,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 050,1 15,       1,     0,
XO(1),   XO(5),
5 , 5 ,
YO(1),   ,YO(5),  (UPPER SFC.)
0.12720, 0.16080,
YO(1),...,YO(5),  (LOWER SFC.)
0.50550, 0.50100,
END OF FILE
```

Case 37.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.00,  -2.15.00, 000, 11, 10,  6,    1,1.0000,
  UINF,    C,   TINF,  PINF,VINQ,
 78.20.0 330,  10.00, 95.1,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 20.0.30,- 20.0.10.51,61,-0.06, 0.24,-0.08, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     0,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 18.6,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 5.D-6,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
  1,  1.0,  1.0, 1,  0,  3,  0,  0.20,-.224, 0.074,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.15,       1,     0,
XO(1),...XO(5),
-5.,
YO(1),...,YO(5),  (UPPER SFC.)
-0.44450,
YO(1),...,YO(5),  (LOWER SFC.)
-0.50360,
END OF FILE
```

Case 38.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 3.00,  -2, 15.0, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,   TINF,  PINF,VINQ,
 78.20.0 330,  10.0, 95.1,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 20.0 30,- 20.0.10.51,61,-0.06, 0.24,-0.08, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     1,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 25 5,0 50, 13 2,0 50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5.D-6,1.D-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  3,  0,  0.10,- 189,  .089,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.15,       1,     0,
XO(1),   XO(5),
-5.,-5.,
YO(1),...,YO(5),  (UPPER SFC.)
-0.31120,-0.34070,
YO(1),...,YO(5),  (LOWER SFC.)
-0.38720,-0.38000,
END OF FILE
```

Case 39.

Case 40.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 4.00,    5, 12.0, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,   TINF,  PINF,VINQ,
 78.20.0 330,   10.0, 95.1,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 20.0 30,  20.0 10,51,61,-0.06, 0.24,-0.08, 0.08,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0, 1,      0,      2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 18.6,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
4.D-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1   1.0,  1.0, 1,  0,  3,  0,   0.0,  0.0,   0.0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0.050,1.15,      -1,     0,
XO(1),...XO(5),
-5.,
YO(1),...,YO(5),  (UPPER SFC.)
-0.48100,
YO(1),...,YO(5),  (LOWER SFC.)
-0.53135,
END OF FILE
```

Case 41.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.70,    5,15.00, 000, 11, 10,  6,    1,1.0000,
  UINF,     C,   TINF,  PINF,VINQ,
 90.00,1.613,-15.00,101.3,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 10.0.10,-.08,0.04,51,61,-0.05, 0.05,-0.03, 0.03,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 15.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1 D-4,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,   1.0, 1,  0,  3,  0,  0.00,  0.0,    0.0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1, .0438,1.10,       1,      0,
XO(1),  . XO(5),
-2 ,
YO(1),. . .YO(5),  (UPPER SFC.)
 0.06400,
YO(1), . . .YO(5),  (LOWER SFC.)
-0.07700,
END OF FILE
```

Case 42.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   -1,100 0, 000,-11,  10, 06,    1,1.0000,
  UINF,      C,   TINF,  PINF,VINQ,
 30.50, .0254,  -15.0,  97.7,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 50.0.50.0.00,0.60,51,61,-0.50, 0.50, 0.00, 0.60,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1 D-7,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,
 1,  1.0,   1.0, 1,  0,  3,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 157,1.20,       0,
XO(1),   XO(5),
-10 ,
YO(1),. . .YO(5),  (UPPER SFC.)
0.27550,
YO(1), . . .YO(5),  (LOWER SFC.)
-0.53125,
END OF FILE
```

Case 43.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   -1,100.0, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,   TINF,  PINF,VINQ,
 30.50, .0254,  -15.0,  97.7,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,-.50,0.50.0.00,0.60,51,61,-0.50, 0.50, 0.00, 0.60,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     0,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 1 D-7,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,
 1,  1.0,   1.0, 1,  0,  3,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 157,1 20,      1,      0,
XO(1),  . XO(5),
-10 ,
YO(1),. .  .YO(5),  (UPPER SFC.)
0.27550,
YO(1), . . .YO(5),  (LOWER SFC.)
-0.53125,
END OF FILE
```

Case 44.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   -1,100 0, 000, 11, 10, 06,    1,1.0000,
  UINF,     C,   TINF,  PINF,VINQ,
 30.50, .0254,  -15.0,  97.7,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 50,0.50.0.00,0.60,51,61,-0.50, 0.50, 0.00, 0.60,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 27.0,0.50, 14.4,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5 D-8,5 D-7,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,
 1,  1.0,   1.0, 1,  0,  3,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,      1,0 157,1.20,       1,      0,
XO(1),   XO(5),
-10 ,-10 ,
YO(1),  . .YO(5),  (UPPER SFC.)
0.33720,0.19900,
YO(1),   .YO(5),  (LOWER SFC.)

END OF FILE
```

Case 45.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINQ,
 30.50, 0254, -15 0, 97 7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,-.50,0 50,0 00,0.60,51,61,-0.50, 0.50, 0.00, 0.60,,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       2, 27.0,0 50, 14.4,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5 D-8,5 D-7,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,
 1,  1 0,  1 0, 1,  0,  3,  0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0 314,1.20,      1,     0,
XO(1),...XO(5),
-10 ,-10.,
YO(1),....YO(5),  (UPPER SFC )
0 33720,0.19900,
YO(1),....YO(5),  (LOWER SFC )
END OF FILE
```

Case 46.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0 00,  -1,100 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINQ,
 30 50, 0254, -15 0, 97 7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 50,0 50,0 00,0 60,51,61,-0 50, 0 50, 0 00, 0 60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       2, 27 0,0 50, 14 4,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5 D-8,5 D-7,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,  0.20,  0.0,   0.0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0 314,1 15,      1,     0,
XO(1),...XO(5),
-10 ,-10.,
YO(1),....YO(5),  (UPPER SFC )
0 33720,0 19900,
YO(1),....YO(5),  (LOWER SFC )
END OF FILE
```

Case 47.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100.0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINQ,
 30.50, 0254, -15 0, 97 7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 50,0 50,0.00,0.60,51,61,-0.50, 0 50, 0.00, 0.60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-7,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  0,  0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0.314,1.20,      1,     0,
XO(1),...XO(5),
-10.,
YO(1),....YO(5),  (UPPER SFC.)
0.27550,
YO(1),....YO(5),  (LOWER SFC.)
-0.53125,
END OF FILE
```

Case 48.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0 00,  -1,100 0, 000, 11, 10, 06,   1,1.0000,
  UINF,    C,   TINF, PINF,VINQ,
 30 50, 0254, -15 0, 97 7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 50,0 50,0 00,0 60,51,61,-0 50, 0 50, 0.00, 0 60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,    0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 20.0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 D-7,
AT,CEDEL,EMDEL,HS,VOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,  0 20,  0 0,   0 0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
      1,      1,0 314,1 15,      1,     0,
XO(1),...XO(5),
-10 ,
YO(1),....YO(5),  (UPPER SFC )
0 27550,
YO(1),....YO(5),  (LOWER SFC )
-0 53125,
END OF FILE
```

Case 49.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1 0000,
  UINF,    C,  TINF, PINF,VINQ,
 30.50, 0254, -15.0, 97.7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 50.0 50.0 00,0 60.51,61,-0.50, 0.50, 0.00, 0.60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-7,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  0,  0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
     1,      1,0.314,1.20,      1,    2,
XO(1),  ..XO(5),
-10.,
YO(1),  ...YO(5),  (UPPER SFC.)
0.27550,
YO(1),  ..,YO(5),  (LOWER SFC.)
-0.53125,
END OF FILE
```

Case 50.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1 0000,
  UINF,    C,  TINF, PINF,VINQ,
 30.50, 0254, -15.0, 97.7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 50.0 50.0 00,0 60.51,61,-0.50, 0.50, 0.00, 0.60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1.D-7,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,
 1,  1.0,  1.0, 1,  0,  0,  0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
     1,      1,0.314,1.20,      1,    1,
XO(1),  XO(5),
-10.,
YO(1),..  ,YO(5),  (UPPER SFC.)
0.27550,
YO(1),..  ,YO(5),  (LOWER SFC.)
-0.53125,
END OF FILE
```

Case 51.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,  -1,100 0, 000, 11, 10, 06,   1,1 0000,
  UINF,    C,  TINF, PINF,VINQ,
 30.50, 0254, -15.0, 97.7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 50,0 50,0.00,0 60.51,61,-0.50, 0.50, 0.00, 0.60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     0,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DU(5),W(5),
     2, 27 0.0 50, 14 4,0 50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
5.0-8,5.D-7,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  0,  0,  0.20,  0.0,    0.0,
ICEPLA,LYRMAX,  ICE,LTOL ATHICK,DENSE,
     1,      1,0.314,1.15,      1,    1,
XO(1),  ..XO(5),
-10.,-10.,
YO(1),  ..,YO(5),  (UPPER SFC.)
0.33720,0.19900,
YO(1),  ..,YO(5),  (LOWER SFC.)

END OF FILE
```

Case 52.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,   1,100 0, 000, 13, 10, 05,   1,1 0000,
  UINF,    C,  TINF, PINF,VINQ,
 30.50, 0254,  15.0, 97.7,   0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0.- 50.0 50.0.00,0 60.51,61,-0.50, 0.50, 0.00, 0.60,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,   0,  1,     1,     2,    30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
3.D-6,
AT,CEDEL,EMDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  3,  0,  0.20,  0.0,    0.0,
ICEPLA,LYRMAX,  ICE,LTOL,ATHICK,DENSE,
     1,      3, 1047,1.05,      1,    0,
XO(1),  XO(5),
-10.,
YO(1),   ,YO(5),  (UPPER SFC.)
0.27700,
YO(1),   ,YO(5),  (LOWER SFC.)

END OF FILE
```

Case 53.

```
ALPHA,TYPE,THICK,MEAN,NEF,NFB,NIF,ANAL,PLTFAC,
 0.00,    1,100 0, 000, 13, 10, 05,    1,1 0000,
  UINF,      C,   TINF, PINF,VINO,
 30 50, 0254, -15.0, 97 7,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      1,- 50,0 50,0.00,0 60,51,61,-0.50, 0.50, 0.00, 0.90,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
      2, 27.0,0.50, 14.4,0.50,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
1 0-6,5.0-6,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,
 1,  1 0,   1.0, 1,  0,   3,  0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       3,.1047,1.15,       1,     0,
XO(1),  ..XO(5),
-10.,-10.,
YO(1),...,YO(5),   (UPPER SFC.)
0 33765,0 20020,
YO(1),....,YO(5),  (LOWER SFC )
END OF FILE
```

Case 54.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0.00,    0.15 00, 000, 11, 10,  6,    1,1 0000,
  UINF,      C,   TINF, PINF,VINO,
 61 00,0 213, 15 00, 98 4,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 12,0 18,0 00,0 18,51,61,-0 08, 0 32,-0 12, 0.12,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 3 0-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,   1 0, 1   0,   3,   0    0 20,   0 0,    0 0,
ICEPLA,LYRMAX,    ICE,LTOL,ATHICK,DENSE,
      1,       1, 0356,1 15,        1,     0.
XO(1),    XO(5),
-5.,
YO(1),..  ,YO(5),   (UPPER SFC )
0 03300,
YO(1),...,YO(5),   (LOWER SFC )
END OF FILE
```

Case 55.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 0 00,    0,15.00, 000, 11, 10,   6,    1,1 0000,
  UINF,      C,   TINF, PINF,VINO,
 61 00,0 213, 15 00, 98 4,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,YMINI,XMAXI,YMINI,YMAXI,
      0,- 12,0 18,0 00,0 18,51,61,-0.08, 0 32,-0 12, 0 12,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,      30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 3 0-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,   1.0, 1,  0,   3,   0,   0.30,  0.0,    0 0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       3, 0119,1 05,       1,     0,
XO(1),    XO(5),
-5 ,
YO(1),..  ,YO(5),   (UPPER SFC )
0 03300,
YO(1),    ,YO(5),   (LOWER SFC )
END OF FILE
```

Case 56.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 8 00,    0 15 00, 000, 11, 10,   6,    1,1 0000,
  UINF,      C,   TINF, PINF,VINO,
 61 00,0 213,-15 00, 97 7,    0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
      0,- 20,0 30,   20,0 10,51,61,-0 08, 0 32,-0.12, 0 12,
EQN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,      30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
     1, 20 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 3 0 5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,   1 0, 1,  0,   3,   0,   0.10,  0 0,    0 0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
      1,       1, 0365,1.15,       1,     0,
XO(1),...YO(5),
-5 ,
YO(1),..  ,YO(5),   (UPPER SFC )
-0 90400,
YO(1),...,YO(5),   (LOWER SFC )
-0 98470,
END OF FILE
```

Case 57.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 8 00,    0.15 00, 000, 11, 10,  6.    1,1.0000,
  UINF,     C.   TINF, PINF,VINQ,
 61.00.0 213,-15 00, 97 7,     0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 24,0.36,- 24,0.12,51,61,-0 08, 0.32,-0 12, 0.12,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 20 0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 3 D-5,
AT,CEDFL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1.0, 1,  0,  3,  0,  0.15,  0.0,   0.0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
       1,       3, 0122,1 05,       1,     0,
XO(1),.  .XO(5),
-5 .
YO(1),..  .YO(5),   (UPPER SFC.)
-0 90400,
.YO(1),   ..YO(5),   (LOWER SFC.)
-0 99470,
END OF FILE
```

Case 58.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 5 70,    0.12 00, 000, 11, 10,  6,    1,1.0000,
  UINF,     C.   TINF, PINF,VINQ,
 60 00,0 417,-14.00, 98 0,     0,
TR JPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 20.0 30,-.20.0 10,51,61,-0 08, 0 24,-0.08, 0 08,
ECN,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     0,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 30 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 2 D-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,  0.10,  0.0,   0.0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
       1,       1, 0296,1 15,      -1,     0,
XO(1),    XO(5),
-5 .
YO(1),   ..YO(5),   (UPPER SFC.)
-0 63250,
YO(1),..  .YO(5),   (LOWER SFC.)
-0.70530,
END OF FILE
```

Case 59.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 5 70,    0.12.00, 000, 11, 10,  6.    1,1.0000,
  UINF,     C.   TINF, PINF,VINQ,
 60.00,0.417,-14 00, 98.0,     0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 20.0.30,-.20.0 10 51,61,-0.08, 0.24,-0.08, 0.08,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     1,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 30.0,1.00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 2 D-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1.0,  1.0, 1,  0,  3,  0,  0.10,  0.0,   0.0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
       1,       3, 0099,1.05,       1,     0,
XO(1)....XO(5),
-5 .
YO(1)...  .YO(5),   (UPPER SFC.)
-0 63250,
YO(1), ...YO(5),   (LOWER SFC.)
 0 70530,
END OF FILE
```

Case 60.

```
ALPHA,TYPE,THICK,MEAN,NEF,NEB,NIF,ANAL,PLTFAC,
 5 70,    5,11 30, 000, 11, 10,  6.    1,1.0000,
  UINF,     C.   TINF, PINF,VINQ,
 60 00,0 443,-14 00, 98 0,     0,
TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
       0,- 28,0 42,- 28,0 14,51,61,-0 08, 0 24,-0 08, 0.08,
EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
  2, 2,  0,    0,  1,     0,     2,     30,
DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
       1, 30 0,1 00,
EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
 2 D-5,
AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
 1,  1 0,  1 0, 1,  0,  3,  0,  0.10,  0.0,   0.0,
ICEPLA,LYRMAX,   ICE,LTOL,ATHICK,DENSE,
       1,       1, 0279,1 15,       1,     0,
XO(1),    XO(5),
-5 .
YO(1),.   .YO(5),   (UPPER SFC.)
-0.63950,
YO(1),...YO(5),   (LOWER SFC.)
-0 70650,
END OF FILE
```

Case 61.

```
      ALPHA,TYPE,THICK,MEAN,NFF,NEB,NIF,ANAL,PLTFAC,
       0.00,  -2.15 00, 000, 11, 10,  6.    1,1.0000,
        UINF.      C.   TINF, PINF,VINQ,
       78.20,.0825,-10.00, 95 1.    0.
      TRJPLA,XMIN,XMAX,YMIN,YMAX,XZ,YZ,XMINI,XMAXI,YMINI,YMAXI,
            0,-.20,0.30,-.20,0.10,51,61,-0.06, 0.24,-0.09, 0.08,
      EON,PC,ACN,GRAV,CDS,TRJPRA,PRINTO,PRINTI,
        2, 2.  0.    0,  1.       1.       2.     30.
      DDISTN,DD(1),W(1),DD(2),W(2),DD(3),W(3),DD(4),W(4),DD(5),W(5),
            2,11.05,0.50, 5.87,0.50,
      EPS(1),EPS(2),EPS(3),EPS(4),EPS(5),
       3.0-6,2.0-5,
      AT,CEDEL,ENDEL,HS,YOL,CEL,CEX,FILTER,LLEFT,LRIGHT,
       1,  1.0,  1.0, 1,  0,  3,  0,  0.20,-.149, 0.149,
      ICEPLA,LYMMAX,  ICE,LTOL,ATHICK,DENSE,
            1,       1,0.050,1.15,      1,     0,
      XO(1),  ..XO(5),
      -5.,-5 ,
      YO(1),   ..,YO(5),   (UPPER SFC.)
      0.03600,0.01880,
      YO(1),....,YO(5),   (LOWER SFC.)

END OF FILE
```

APPENDIX I. Sample program output.

The output which follows is a portion of the output from Case 32. It is displayed here because the trajectory data for this case were used in determining the correct values of the scaling parameters for Case 61. Included are sample pages of output prior to the grazing trajectory, as well as for a trajectory pair well within the grazing trajectory envelope. Also displayed are pages giving the layer thickness and collision efficiency values, the total accreted area, and the coordinates of the final airfoil surface.

On the first page, STEP refers to the time step number, DTS to its size, and TIME to the sum of DTS. XDS and YDS give the droplet position, PSI the streamfunction value, and UAS, UDS, VAS, VDS the x and y components of the air and droplet velocities. RED displays the droplet Reynolds number, ACCN/MOD the acceleration modulus, and HIST/RHS the proportion of the total acceleration contributed by the history term.

ACCRETION OF LAYER 1 DROPLET DIAMETER 25 5 EPS= 3 000E-06

TRAJECTORY STARTING POSITION IS X= -5 00 YO= 0 03587

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RHS
3	0 21	0 2715	-4 79470	0 03587	0 03585	0 99935	0 99940	0 00001	0 00001	0 006	0 00000	0 03690
4	0 48	0 3728	-4 52330	0 03588	0 03585	0 99928	0 99937	0 00001	0 00001	0 011	0 00000	0 05767
5	0 85	0 4349	-4 18083	0 03588	0 03585	0 99916	0 99929	0 00001	0 00001	0 017	0 00000	0 07589
6	1 20	0 4735	-3 71626	0 03588	0 03585	0 99897	0 99916	0 00002	0 00001	0 029	0 00000	0 08866
7	1 76	0 4896	-3 24320	0 03589	0 03584	0 99868	0 99895	0 00003	0 00002	0 036	0 00000	0 09760
8	2 24	0 4616	-2 79817	0 03590	0 03584	0 99823	0 99865	0 00004	0 00003	0 059	0 00000	0 10110
9	2 71	0 4121	-2 29727	0 03592	0 03583	0 99755	0 99820	0 00007	0 00004	0 089	0 00000	0 10000
10	3 12	0 3571	-1 88600	0 03594	0 03582	0 99655	0 99756	0 00011	0 00007	0 134	0 00000	0 09638
11	3 47	0 3036	-1 52895	0 03597	0 03580	0 99507	0 99649	0 00018	0 00011	0 213	0 00000	0 09126
13	4 03	0 2085	-0 97483	0 03607	0 03571	0 98978	0 99306	0 00058	0 00028	0 539	0 00000	0 07908
15	4 41	0 1321	-0 60181	0 03625	0 03549	0 97902	0 98887	0 00181	0 00075	1 306	0 00001	0 05672
19	4 76	0 0227	-0 23654	0 03688	0 03432	0 93161	0 97031	0 01229	0 00383	5 224	0 00022	0 04530
20	4 80	0 0228	-0 21450	0 03698	0 03411	0 92398	0 96740	0 01472	0 00449	5 883	0 00026	0 04427
21	4 83	0 0222	-0 19248	0 03709	0 03389	0 91480	0 96399	0 01793	0 00534	6 687	0 00031	0 04396
22	4 85	0 0260	-0 17108	0 03722	0 03355	0 90409	0 96012	0 02208	0 00639	7 674	0 00032	0 04209
23	4 87	0 0271	-0 14616	0 03741	0 03308	0 88861	0 95465	0 02880	0 00803	9 139	0 00038	0 04039
24	4 90	0 0252	-0 12034	0 03766	0 03243	0 86790	0 94752	0 03966	0 01048	11 185	0 00054	0 03833
25	4 93	0 0212	-0 09699	0 03796	0 03157	0 84298	0 93907	0 05547	0 01389	13 809	[illegible]	0 03612
27	4 97	0 0149	-0 06048	0 03865	0 02940	0 79019	0 92072	0 10348	0 02349	20 191	[illegible]10	0 03224
29	4 99	0 0108	-0 03903	0 03949	0 02696	0 74909	0 90169	0 17056	0 03779	27 775	0 [illegible]77	0 02826
32	5 02	0 0070	-0 01086	0 04086	0 02174	0 73352	0 87903	0 31587	0 06589	36 190	0 01173	0 02686
36	5 05	0 0071	0 01201	0 04318	0 01449	0 84461	0 86488	0 46679	0 11454	46 536	0 01867	0 02716
40	5 08	0 0069	0 03706	0 04739	0 00627	1 04886	0 88014	0 48081	0 17580	46 044	0 01901	0 03104
43	5 10	0 0081	0 05622	0 05153	0 00224	1 15244	0 90836	0 41468	0 20856	42 136	0 01047	0 03462
46	5 12	0 0102	0 08040	0 05729	0 00018	1 21666	0 94826	0 32353	0 22989	37 494	0 00841	0 03802
48	5 14	0 0124	0 10104	0 06228	0 00075	1 24088	0 98072	0 29722	0 23961	34 473	0 00436	0 03975
50	5 17	0 0156	0 12728	0 06846	0 00223	1 25118	1 01984	0 19923	0 23264	31 629	0 00281	0 04081
52	5 20	0 0202	0 16157	0 07586	0 00635	1 24963	1 05234	12137	0 21879	28 948	0 00187	0 04122
53	5 22	0 0220	0 18299	0 08026	0 00946	1 24312	1 07080	0 08776	0 20683	27 622	0 00157	0 04126
54	5 25	0 0231	0 20669	0 08464	0 01321	1 23441	1 08780	0 05645	0 19223	26 346	0 00136	0 04130
55	5 27	0 0239	0 23199	0 08888	0 01737	1 22380	1 10281	0 02840	0 17986	25 136	0 00121	0 04140
56	5 29	0 0248	0 25851	0 09286	0 02181	1 21132	1 11448	0 00371	0 15764	23 886	0 00107	0 04164
57	5 32	0 0257	0 28628	0 09659	0 02181	1 19794	1 12387	-0 01794	0 13884	22 862	0 00096	0 04302
58	5 34	0 0202	0 31524	0 09986								

STABILITY INDEX -0 995 AT X= 0 08015 CLOSEST APPROACH IS Y= 0 00002

TIME STEPS= 58 FN EVALUATIONS= 400 FINAL Y= 0 08810

ACCRETION OF LAYER 1 DROPLET DIAMETER 25 5 EPS= 3 000E-06

TRAJECTORY STARTING POSITION IS X= -5 00 YO= 0 00420

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RMS
3	0 21	0 2672	-4 79470	0 00420	0 00419	0 99936	0 99940	0 00000	0 00000	0 006	0 00000	0 03880
4	0 47	0 3702	-4 52762	0 00420	0 00419	0 99928	0 99937	0 00000	0 00000	0 011	0 00000	0 08742
5	0 84	0 4332	-4 15766	0 00420	0 00419	0 99916	0 99929	0 00000	0 00000	0 017	0 00000	0 07980
6	1 28	0 4726	-3 72480	0 00420	0 00419	0 99887	0 99916	0 00000	0 00000	0 025	0 00000	0 08844
7	1 75	0 4853	-3 25268	0 00420	0 00419	0 99868	0 99896	0 00000	0 00000	0 036	0 00000	0 08748
8	2 23	0 4623	-2 76790	0 00420	0 00419	0 99824	0 99865	0 00000	0 00000	0 064	0 00000	0 10108
9	2 70	0 4131	-2 30628	0 00420	0 00419	0 99757	0 99821	0 00001	0 00001	0 085	0 00000	0 10004
10	3 11	0 3580	-1 89404	0 00420	0 00419	0 99657	0 99758	0 00001	0 00001	0 133	0 00000	0 08647
11	3 47	0 3045	-1 53701	0 00421	0 00419	0 99511	0 99671	0 00002	0 00001	0 211	0 00000	0 08137
13	4 03	0 2081	-0 98030	0 00422	0 00418	0 98984	0 99388	0 00007	0 00003	0 534	0 00000	0 07821
15	4 40	0 1324	-0 60803	0 00424	0 00415	0 97811	0 98892	0 00021	0 00008	1 283	0 00001	0 06685
* 19	4 78	0 0224	-0 24068	0 00431	0 00402	0 93122	0 97036	0 00142	0 00044	5 163	0 00022	0 04837
20	4 80	0 0225	-0 21887	0 00432	0 00398	0 92342	0 96746	0 00170	0 00052	5 808	0 00026	0 04434
21	4 82	0 0217	-0 19720	0 00434	0 00396	0 91401	0 96405	0 00206	0 00061	6 602	0 00031	0 04367
22	4 84	0 0251	-0 17636	0 00435	0 00393	0 90306	0 96022	0 00253	0 00073	7 544	0 00032	0 04219
23	4 87	0 0258	-0 15237	0 00437	0 00388	0 88718	0 95486	0 00329	0 00091	8 932	0 00039	0 04082
24	4 89	0 0239	-0 12783	0 00440	0 00381	0 86577	0 94794	0 00446	0 00117	10 846	0 00095	0 03851
26	4 94	0 0174	-0 08598	0 00447	0 00361	0 80796	0 93041	0 00855	0 00200	16 225	0 00131	0 03418
28	4 97	0 0123	-0 05645	0 00455	0 00332	0 73108	0 90836	0 01608	0 00332	23 573	0 00320	0 03023
31	5 00	0 0077	-0 02817	0 00470	0 00271	0 57816	0 87178	0 03951	0 00676	38 837	0 01115	0 02523
36	5 03	0 0032	-0 00427	0 00503	0 00098	0 21342	0 79307	0 18013	0 03027	78 647	0 08164	0 01882
38	5 04	0 0021	0 00019	0 00517	0 00082	0 08476	0 78251	0 24800	0 02878	98 804	0 20776	0 01732
39	5104	0 0086	0 00175	0 00523								

COLLISION COORDS X= 0 00062 Y= 0 00517 L= 0 00520

TIME STEPS= 39 FN EVALUATIONS= 267

STABILITY INDEX= -0 883 COLLISION VELOCITY 0 76075 AT 13 6 DEG

ACCRETION OF LAYER 1 DROPLET DIAMETER 25 5 EPS= 3 000E-06

TRAJECTORY STARTING POSITION IS X= -5 00 YO= 0 00441

STEP	TIME	DTS	XDS	YDS	PSI	UAS	UDS	VAS	VDS	RED	ACCN/MOD	HIST/RMS
3	0 21	0 2705	-4 79470	0 00441	0 [illegible]	0 99936	0 99940	0 00000	0 00000	0 006	0 00000	0 03880
4	0 48	0 3720	-4 52437	0 00441	0 00440	0 99928	0 99937	0 00000	0 00000	0 011	0 00000	0 08761
5	0 85	0 4345	-4 15258	0 00441	0 00440	0 99916	0 99929	0 00000	0 00000	0 017	0 00000	0 07581
6	1 28	0 4732	-3 71846	0 00441	0 00440	0 99897	0 99916	0 00000	0 00000	0 025	0 00000	0 08860
7	1 76	0 4895	-3 24572	0 00441	0 00440	0 99868	0 99896	0 00000	0 00000	0 036	0 00000	0 08787
8	2 24	0 4617	-2 76080	0 00441	0 00440	0 99823	0 99865	0 00001	0 00000	0 055	0 00000	0 10109
9	2 70	0 4123	-2 29878	0 00441	0 00440	0 99755	0 99820	0 00001	0 00001	0 089	0 00000	0 10000
10	3 12	0 3572	-1 88833	0 00442	0 00440	0 99655	0 99757	0 00001	0 00001	0 134	0 00000	0 08640
11	3 47	0 3036	-1 53211	0 00442	0 00440	0 99508	0 99669	0 00002	0 00001	0 213	0 00000	0 08129
13	4 03	0 2085	-0 97689	0 00443	0 00439	0 98979	0 99386	0 00007	0 00004	0 538	0 00000	0 07911
15	4 41	0 1319	-0 60364	0 00445	0 00436	0 97800	0 98887	0 00022	0 00008	1 302	0 00001	0 08676
* 19	4 78	0 0223	-0 23984	0 00453	0 00422	0 93087	0 97027	0 00150	0 00047	5 184	0 00022	0 04533
20	4 80	0 0225	-0 21831	0 00454	0 00419	0 92316	0 96736	0 00179	0 00055	5 832	0 00026	0 04430
21	4 82	0 0216	-0 19862	0 00456	0 00416	0 91373	0 96396	0 00218	0 00065	6 627	0 00032	0 04362
22	4 84	0 0290	-0 17584	0 00457	0 00413	0 90275	0 96012	0 00267	0 00077	7 571	0 00032	0 04215
23	4 87	0 0287	-0 15192	0 00459	0 00407	0 88685	0 95476	0 00347	0 00096	8 962	0 00038	0 04048
24	4 89	0 0238	-0 12746	0 00462	0 00400	0 86541	0 94783	0 00471	0 00124	10 881	0 00066	0 03847
26	4 94	0 0174	-0 08572	0 00469	0 00379	0 80710	0 93027	0 00903	0 00210	16 271	0 00132	0 03415
28	4 97	0 0123	-0 05625	0 00478	0 00349	0 73081	0 90820	0 01696	0 00350	23 636	0 00322	0 03030
31	5 00	0 0077	-0 02802	0 00494	0 00284	0 57834	0 87184	0 04173	0 00713	38 962	0 01118	0 02521
36	5 03	0 0032	-0 00427	0 00528	0 00101	0 21676	0 79239	0 16739	0 02124	78 486	0 08136	0 01896
38	5 04	0 0020	0 00018	0 00542	0 00088	0 07221	0 76326	0 25875	0 03009	96 007	0 20726	0 01738
39	5 04	0 0086	0 00173	0 00540								

COLLISION COORDS X= 0 00087 Y= 0 00843 L= 0 00847

TIME STEPS= 39 FN EVALUATIONS= 267

STABILITY INDEX= -0 884 COLLISION VELOCITY 0 76102 AT 14 3 DEG

LOCAL BETA: 79 2% EST MAX BETA 81 3% MAX BETA CHANGE 0 0%

BETAO (MAX LOCAL CE) IS 62 7% AT A DISTANCE OF -0 000 FROM THE NOSE

THE TOTAL COLLISION EFFICIENCY IS 24 4%

MASS MEAN DIAMETER 19 3 MICROMETERS

END POINT	X COORD	Y COORD	DISTANCE FROM NOSE	LAYER ACCRETION THICKNESS	NORM ICE DENSITY	AVERAGE COLLISION EFFICIENCY	DROPLET DIAMETER 25 5 UM COLLISION EFFICIENCY	DROPLET DIAMETER 13 2 UM COLLISION EFFICIENCY
1	0 00000	0 0	0 0	0 02482	1 000	71 87	81 27	62 67
2	0 00004	0 00141	0 00141	0 02447	1 000	71 80	81 13	62 48
3	0 00015	0 00282	0 00283	0 02441	1 000	71 31	80 69	61 93
4	0 00035	0 00423	0 00425	0 02431	1 000	70 96	79 97	61 15
5	0 00062	0 00564	0 00569	0.02420	1 000	69 96	78 98	60 17
6	0 00097	0 00705	0 00714	0 02404	1 000	68 35	77 73	58 96
7	0 00139	0 00845	0 00860	0.02384	1 000	66 87	76 26	57 48
8	0 00189	0 00986	0 01008	0 02360	1 000	65 17	74 59	55 79
9	0 00247	0 01125	0 01161	0 02330	1 000	63 28	72 75	53 81
10	0 00313	0 01265	0 01315	0 02295	1 000	61 26	70 76	51 76
11	0 00386	0 01404	0 01472	0 02256	1 000	59 12	68 64	49 59
12	0 00467	0 01542	0 01632	0 02212	1 000	56 88	66 45	47.31
13	0 00556	0 01680	0 01796	0 02162	1 000	54 57	64 18	44 95
14	0 00652	0 01817	0 01964	0 02107	1 000	52 21	61 86	42 56
15	0 00756	0 01954	0 02135	0 02047	1 000	49 81	59 51	40 12
16	0 00867	0 02090	0 02311	0 01982	1 000	47 41	57 16	37 65
17	0 00987	0 02225	0 02491	0 01912	1 000	44 99	54 82	35 16
18	0 01113	0 02359	0 02676	0 01839	1 000	42 59	52 50	32 68
19	0 01248	0 02493	0 02866	0 01763	1 000	40 23	50 20	30 25
20	0 01390	0 02625	0 03060	0 01684	1 000	37 90	47 94	27 86
21	0 01539	0 02757	0 03259	0 01601	1.000	35 58	45 70	25 47
22	0 01696	0 02888	0 03464	0 01516	1 000	33 28	43 49	23.06
23	0 01861	0 03017	0 03673	0 01428	1 000	31 02	41 33	20 70
24	0 02033	0 03146	0 03888	0 01341	1 000	28 85	39 22	18 48
25	0 02213	0 03273	0 04108	0 01254	1 000	26 72	37 16	16 29
26	0 02400	0 03400	0 04334	0 01162	1 000	24 56	35 15	13 97
27	0 02594	0 03525	0 04565	0 01068	1 000	22 [illegible]	33 20	11 90
28	0 02796	0 03648	0 04802	0 00979	1.000	20 41	31 29	9 53
29	0 03006	0 03771	0 05045	0 00903	1 000	18 71	29 43	8 00
30	0 03222	0 03892	0 05293	0 00810	1.000	16 69	27 61	5 77
31	0 03447	0 04012	0 05547	0 00680	1 000	14.13	25 83	2 42
32	0 03678	0 04130	0 05807	0 00581	1 000	12 08	24 11	0 0
33	0 03917	0 04247	0 06073	0 00551	1 000	11 22	22 43	0 0
34	0 04163	0 04362	0 06344	0 00512	1 000	10 40	20 80	0 0
35	0 04416	0 04476	0 06622	0 00474	1 000	9 61	19 21	0 0
36	0 04676	0 04588	0 06906	0 00436	1 000	8.83	17 65	0 0
37	0 04944	0 04699	0 07196	0 00399	1 000	8 06	16 12	0 0
38	0 05219	0 04808	0 07491	0 00363	1 000	7 32	14 63	0 0
39	0 05500	0 04915	0.07792	0 00328	1 000	6 60	13.20	0 0
40	0 05789	0 05020	0 08100	0 00294	1 000	5 91	11 81	0 0
41	0.06085	0 05124	0 08413	0 00260	1 000	5 23	10 47	0 0
42	0 06388	0 05226	0 08733	0 00227	1 000	4 57	9 14	0 0
43	0 06698	0 05325	0 09059	0 00194	1 000	3 88	7 77	0 0
44	0.07015	0 05423	0 09390	0.00158	1 000	3 17	6 34	0 0
45	0 07339	0 05519	0 09728	0 00124	1 000	2 48	4 95	0 0
46	0 07670	0 05613	0 10072	0 00092	1.000	1.84	3 68	0.0
47	0 08007	0 05706	0 10422	0 00064	1.000	1 28	2 56	0 0
48	0 08352	0 05795	0 10778	0 00040	1 000	0 80	1 60	0 0
49	0.08703	0.05883	0.11138	0 00018	1 000	0 37	0.74	0 0
50	0 09060	0 05968	0.11507	-0 00000	1.000	0.0	0 0	0.0

THE ACCRETED AREA FOR LAYER 1 IS 0.002710

THE ACCUMULATED ACCRETED AREA IS 0.002710

END POINT	X COORD	Y COORD	DISTANCE FROM NOSE	X COORD	Y COORD	DISTANCE FROM NOSE
1	-0 02452	0 00000	0 0	-0 02452	0 00000	0 0
2	-0 02440	0 00275	0 00275	-0 02440	-0 00275	-0 00275
3	-0 02411	0 00548	0 00550	-0 02411	-0 00548	-0 00550
4	-0 02364	0 00818	0 00824	-0 02364	-0 00818	-0 00824
5	-0 02302	0 01083	0 01086	-0 02302	-0 01083	-0 01086
6	-0 02222	0 01342	0 01367	-0 02222	-0 01342	-0 01367
7	-0 02125	0 01593	0 01636	-0 02125	-0 01593	-0 01636
8	-0 02012	0 01835	0 01904	-0 02012	-0 01835	-0 01904
9	-0 01884	0 02068	0 02168	-0 01884	-0 02068	-0 02168
10	-0 01741	0 02289	0 02433	-0 01741	-0 02289	-0 02433
11	-0 01586	0 02500	0 02695	-0 01586	-0 02500	-0 02695
12	-0 01418	0 02699	0 02955	-0 01418	-0 02699	-0 02955
13	-0 01239	0 02886	0 03214	-0 01239	-0 02886	-0 03214
14	-0 01049	0 03060	0 03472	-0 01049	-0 03060	-0 03472
15	-0 00850	0 03223	0 03729	-0 00850	-0 03223	-0 03729
16	-0 00642	0 03374	0 03986	-0 00642	-0 03374	-0 03986
17	-0 00426	0 03514	0 04243	-0 00426	-0 03514	-0 04243
18	-0 00203	0 03643	0 04501	-0 00203	-0 03643	-0 04501
19	0 00025	0 03763	0 04758	0 00025	-0 03763	-0 04758
20	0 00259	0 03873	0 05017	0 00259	-0 03873	-0 05017
21	0 00498	0 03974	0 05277	0 00498	-0 03974	-0 05277
22	0 00743	0 04066	0 05538	0 00743	-0 04066	-0 05538
23	0 00992	0 04151	0 05801	0 00992	-0 04151	-0 05801
24	0 01244	0 04231	0 06066	0 01244	-0 04231	-0 06066
25	0 01499	0 04304	0 06332	0 01499	-0 04304	-0 06332
26	0 01761	0 04370	0 06601	0 01761	-0 04370	-0 06601
27	0 02027	0 04429	0 06874	0 02027	-0 04429	-0 06874
28	0 02293	0 04489	0 07147	0 02293	-0 04489	-0 07147
29	0 02558	0 04555	0 07419	0 02558	-0 04555	-0 07419
30	0 02834	0 04603	0 07700	0 02834	-0 04603	-0 07700
31	0 03127	0 04623	0 07994	0 03127	-0 04623	-0 07994
32	0 03414	0 04659	0 08283	0 03414	-0 04659	-0 08283
33	0 03678	0 04744	0 08561	0 03678	-0 04744	-0 08561
34	0 03949	0 04828	0 08845	0 03949	-0 04828	-0 08845
35	0 04225	0 04910	0 09132	0 04225	-0 04910	-0 09132
36	0 04507	0 04990	0 09425	0 04507	-0 04990	-0 09425
37	0 04794	0 05069	0 09723	0 04794	-0 05069	-0 09723
38	0 05087	0 05146	0 10027	0 05087	-0 05146	-0 10027
39	0 05386	0 05222	0 10335	0 05386	-0 05222	-0 10335
40	0 05691	0 05296	0 10648	0 05691	-0 05296	-0 10648
41	0 06001	0 05370	0 10967	0 06001	-0 05370	-0 10967
42	0 06317	0 05441	0 11292	0 06317	-0 05441	-0 11292
43	0 06640	0 05510	0 11622	0 06640	-0 05510	-0 11622
44	0 06969	0 05575	0 11957	0 06969	-0 05575	-0 11957
45	0 07306	0 05638	0 12298	0 07306	-0 05638	-0 12298
46	0 07645	0 05702	0 12645	0 07645	-0 05702	-0 12645
47	0 07991	0 05767	0 12997	0 07991	-0 05767	-0 12997
48	0 08342	0 05834	0 13354	0 08342	-0 05834	-0 13354
49	0 08696	0 05901	0 13717	0 08696	-0 05901	-0 13717
50	0 09060	0 05969	0 14085	0 09060	-0 05969	-0 14085
51	0 09425	0 06053	0 14459	0 09425	-0 06053	-0 14459
52	0 09796	0 06135	0 14839	0 09796	-0 06135	-0 14839
53	0 10173	0 06214	0 15225	0 10173	-0 06214	-0 15225
54	0 10557	0 06292	0 15616	0 10557	-0 06292	-0 15616
55	0 10948	0 06367	0 16014	0 10948	-0 06367	-0 16014
56	0 11345	0 06439	0 16418	0 11345	-0 06439	-0 16418
57	0 11748	0 06510	0 16827	0 11748	-0 06510	-0 16827
58	0 12158	0 06578	0 17243	0 12158	-0 06578	-0 17243
59	0 12574	0 06644	0 17664	0 12574	-0 06644	-0 17664
60	0 12996	0 06708	0 18091	0 12996	-0 06708	-0 18091
61	0 13425	0 06769	0 18524	0 13425	-0 06769	-0 18524
62	0 13859	0 06828	0 18962	0 13859	-0 06828	-0 18962
63	0 14300	0 06884	0 19406	0 14300	-0 06884	-0 19406
64	0 14746	0 06938	0 19856	0 14746	-0 06938	-0 19856
65	0 15199	0 06990	0 20312	0 15199	-0 06990	-0 20312
66	0 15657	0 07039	0 20773	0 15657	-0 07039	-0 20773
67	0 16122	0 07086	0 21239	0 16122	-0 07086	-0 21239
68	0 16592	0 07130	0 21712	0 16592	-0 07130	-0 21712
69	0 17068	0 07172	0 22189	0 17068	-0 07172	-0 22189
70	0 17549	0 07211	0 22672	0 17549	-0 07211	-0 22672
71	0 18036	0 07248	0 23161	0 18036	-0 07248	-0 23161
72	0 18529	0 07283	0 23655	0 18529	-0 07283	-0 23655
73	0 19027	0 07314	0 24154	0 19027	-0 07314	-0 24154
74	0 19531	0 07344	0 24659	0 19531	-0 07344	-0 24659
75	0 20040	0 07371	0 25168	0 20040	-0 07371	-0 25168
76	0 20554	0 07396	0 25683	0 20554	-0 07396	-0 25683
77	0 21074	0 07417	0 26203	0 21074	-0 07417	-0 26203
78	0 21598	0 07436	0 26728	0 21598	-0 07436	-0 26728
79	0 31307	0 07379	0 36439	0 31307	-0 07379	-0 36439
80	0 42122	0 06690	0 47280	0 42122	-0 06690	-0 47280
81	0 53578	0 05405	0 58804	0 53578	-0 05405	-0 58804
82	0 65119	0 03689	0 70444	0 65119	-0 03689	-0 70444
83	0 76092	0 02367	0 81519	0 76092	-0 02367	-0 81519
84	0 85771	0 01163	0 91275	0 85771	-0 01163	-0 91275
85	0 93402	0 00383	0 98947	0 93402	-0 00383	-0 98947
86	0 98307	0 00061	1 03863	0 98307	-0 00061	-1 03863
87	1 00000	0 0	1 05567	1 00000	0 0	-1 05567