THE UNIVERSITY OF ALBERTA

NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS

by

© KAREN JOAN FINSTAD

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

Fall, 1986

ISBN 0-315-32478-3

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR KAREN JOAN FINSTAD

TITLE OF THESIS NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED Fall, 1986

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) Karen J. Finstad

PERMANENT ADDRESS:

#901 - 10149 Saskatchewan Drive

Edmonton, Alberta

T6E 6B6

DATED July 9 1986

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 NUMERICAL AND EXPERIMENTAL STUDIES OF RIME ICE ACCRETION ON CYLINDERS AND AIRFOILS submitted by KAREN JOAN FINSTAD in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY in METEOROLOGY.

Edward Lozowski

Supervisor

....

....

....

....

External Examiner

Date July 9, 1986

Abstract

Four separate studies describe improvements to numerical techniques used in the modelling of ice accretion on circular cylinders and on airfoils.

1. For collision efficiencies, accretion limits and impact velocities on circular cylinders, tables of numerical data and analytical approximating functions are presented. These data replace those of Langmuir and Blodgett (1946).
2. Parameterizations of the local collision efficiency distribution on circular cylinders are derived from numerical data, and applied in an 'operational' style computer model of rime ice accretion. The model's performance is compared with experimental accretions grown in the University of Alberta FROST tunnel.
3. The distribution of local ice density in rime ice is investigated in experimental wind-tunnel accretions made on circular cylinders. A form of the distribution is derived for application to the numerical model described in the previous study.
4. A second model, for rime ice accretion on airfoils or other arbitrary shapes, combines droplet trajectory integrations and parameterizations for local collision efficiency and density distribution. By use of these parameterizations, and of simple numerical methods throughout, the model is made significantly faster than previous models have been. Verification of the model is

provided by comparisons with several wind-tunnel experiments.

Acknowledgements

A most sincere thank-you is due to my thesis supervisor, Professor Edward Lozowski, for his expert guidance and perceptive advice, given throughout the course of my studies, and also for arranging generous research and conference travel funds . Profes[illegible]tes, as co-leader of the Icing Group, was an equally valuable source [illegible] and expertise. To both of them, many thanks for always lending the time, and a good-natured ear, to my questions.

I would also like to express my appreciation to the members of my examination committee for their careful reading of the manuscript, and pertinent comments: Professor Erhard Reinelt, Assistant Professor John Wilson, and especially to Professor Gordon Rostoker for taking time from his busy schedule to serve as Chairman, and External Examiner Dr. Lasse Makkonen, for overcoming the difficulties presented by some 15,000 km and a Finnish postal strike.

Thanks also, to the other student members of the Icing Group, and to the students and staff of the Meteorology Division, for support and friendship.

Financial support in the form of Postgraduate Fellowships from the Atmospheric Environment Service, Canada, and a Graduate Faculty Fellowship from The University of Alberta, is gratefully acknowledged.

Finally, to J.A.H., my chief editor, programming advisor, and billiards marker - thanks for everything.

Table of Contents

List of Tables

List of Figures

List of Symbols

Symbol	Definition
C_d	- droplet drag coefficient
$C_{X,n}$	- constant coefficients for Equation II.11
D_c	- circular cylinder diameter
D_d	- droplet diameter
d42	- diameter of droplet at the 42nd volume percentile of the droplet spectrum
E	- overall collision efficiency
h_0	- airfoil width as a fraction of chord lengt
K	- Langmuir non-dimensional mass parameter
K_0	- a function of K and Re, used in analytical fits to the data of Langmuir and Blodgett
lwc	- liquid water content
mvd	- median volume diameter droplet
R	- Macklin correlation parameter for rotating cylinder ice density
R'	- revised Macklin parameter for correlation of ρ_0
R_c	- circular cylinder diameter
Re	- droplet Reynolds number based on the free-stream velocity
Re_d	- droplet Reynolds number based on the relative velocity
r_d	- droplet radius
T	- temperature

Th	- thickness of ice layer
Tl	- tail length, added to α_{max}
T_s	- mean surface temperature of ice layer
$T_{s,0}$	- surface temperature of ice at the stagnation line
Δt	- integration time step for droplet trajectories
U	- free-stream air speed
u_x, u_y	- air speed coordinates
V	- relative speed of droplet with respect to the air
V_r	- impact speed of droplet in the direction normal to the cylinder surface
V_0	- droplet impact speed at the stagnation line
v	- absolute velocity of droplet
v_x, v_y	- droplet speed coordinates
x_0, y_0	- initial droplet coordinates
Δy	- initial separation of trajectory pair
$\Delta \ell$	- final separation in arc length of the trajectory pair
α	- angle between surface normal and the free-stream
α'	- scaled surface angle for Equation V.7
α_0	- angular position of maximum β
α_{max}	- angular position of accretion limit
α_m	- accretion limit plus tail length, Tl

β - local collision efficiency

β_0 - local collision efficiency at the stagnation line

ζ - Stallabrass correlation parameter

θ - position angle on circular cylinder

μ - dynamic viscosity of air

ρ - local ice density

ρ_a - density of air

ρ_d - density of the water droplet

ρ_0 - local ice density at the stagnation line

τ - ice layer time step

Φ - angle of attack of the free-stream

ϕ - non-dimensional Langmuir flow parameter, or Stokes number

ψ - potential flow streamfunction

I. Introduction

'Icing' presents a problem occasionally to pastry cooks, sometimes to hockey players, and always to those concerned with the natural phenomenon of ice accretion.

In cold climates, ice accretion may result from atmospheric cloud droplets, rain, snow, or sea spray. It may cause serious hazards for aircraft, power lines and transmission towers, ships, and off-shore oil rigs. All of these represent human enterprises that involve many lives and dollars, which may be jeopardized during icing events. So, the primary aim in studying icing is the prediction of its occurrence, whether of a particular event or of the probability of extreme events, and ultimately the prevention of costly and often disastrous accidents.

The icing phenomenon may appear straightforward at first glance: water is collected on a cold surface and freezes there. However, a deeper look reveals many layers of complexity in the physical processes involved, and serious practical difficulties in the observational, experimental, and theoretical aspects of the problem. The development of a predictive model is therefore not an easy task, but it is nonetheless one of some importance.

As with other physical problems, there are two approaches that may be taken in model building. The first is a statistical approach, in which a very large number of events (predictands) are observed, as well as their antecedent or concurrent conditions (predictors). A statistical analysis will then look for correlations between the predictors and predictands. Considerable computational effort may be required to establish a 'best guess' for the most significant predictors, and for the form of the predictive relation.

The reliability of the outcome is highly dependent on the number and quality of the observations. The advantage of this method is that a predictive relation may be derived and applied in the absence of any further understanding of the physical processes involved, which may be complex. But if the observations are adequate, and a good correlation can be found, it is the quickest way of achieving practical forecasting results.

However, the predictive skill of this method is limited by the strength of the discovered correlation, and its applicability is limited to the range of conditions defined by the original observations. For icing studies, this of course includes the geographical range, since the character of icing problems has a large regional variability.

The other approach to modelling is the physical one. This usually means a mathematical model, solved numerically using a digital computer, in which an attempt is made to simulate the intermediate steps, and to understand the causal relationships between the input conditions and the output predictions. Producing such a model does take considerably more time and effort than producing a statistical correlation, and the requirements for the observational data are just as stringent. Furthermore, no mathematical model can perfectly reproduce a given physical process or situation - there must always be simplifying assumptions.

But the knowledge gained from a physical model is less restricted in its application, within those assumptions, than is any purely statistical relationship, and it can provide a useful testing ground for theory. Even the simplest physical model might at least suggest what the most effective predictors will be for a statistical study; thus if it is found that liquid water content and cloud droplet size are of prime importance in determining model ice accretion rates, then collectors of field data should take note, even if these things are very difficult to observe.

A. The Icing Problem

A model of the complete ice accretion process involves the solution of a series of problems. First, an approximation to the airflow around the accreting object must be found. If potential flow is assumed, and there is no analytical solution (as there is for circular cylinders or Joukowski airfoils) then the partial differential equation for the velocity potential must be solved numerically, as in Hess and Smith (1967), or Kennedy and Marsden (1976), for example.

Once it is possible to calculate the air velocity at any point, the trajectories of water droplets carried by the airstream toward the object may be determined. Again, simplifying assumptions are applied to the droplet's equation of motion in order to reduce the computational time needed to integrate it numerically.

An ensemble of droplet trajectories is calculated, and their points of impact on the accretion surface determined. This information is necessary to derive the impingement characteristics of the accretion shape. These are: the distribution of the local collision efficiency, the overall collision efficiency, the maximum impingement angle, and impact speed of the droplets. The collection efficiency is the product of the collision efficiency and the sticking efficiency. Since the latter is unknown, but usually assumed to be unity, only the collision efficiency is dealt with. These quantities and their symbols are defined for a circular cylinder in Figure II.1.

Langmuir and Blodgett (1946) authored the classical solution to this problem, for collision efficiencies of cylinders, spheres, and ribbons in a potential flow field. More recent, but less extensive calculations have been made by McComber and Touzot (1981), and by Egelhofer, et al. (1984). The experimental work of Ranz and Wong (1952) qualitatively validates the theory of Langmuir and Blodgett, if not their numerical calculations.

The impingement characteristics depend on the cylinder size, droplet size, the free-stream air speed, and the air temperature and pressure. If more than one droplet size is present, the impingement characteristics of each size, or size interval, must be determined. A common simplification to this requirement for droplet spectra is made through the substitution of the impingement characteristics of a single droplet size, that of the median volume droplet of the size spectrum. The usefulness of this simplification is examined in Appendix VI, although it is used throughout the thesis.

Only when all of these calculations are completed can ice begin to accrete in the model. For obvious reasons then, most modelers have turned to standard accretion shapes for which these calculations are simpler, for which much data already exist, and for ease of comparison between models. Thus we have the ubiquitous horizontal circular cylinder in a uniform cross-flow, which appears in the models of Macklin and Payne (1967), McComber and Touzot (1981), Lozowski, Stallabrass and Hearty (1983), Horjen (1983), Egelhofer, et al. (1984) and Makkonen (1984). The more difficult problem of accretion on airfoils has been treated by Oleskiw (1982), Gent and Cansdale (1985), and MacArthur, et al. (1982).

The cloud water is usually assumed to be at the temperature of the air stream, and to arrive at the accreting surface in a uniform flux determined by the liquid water content, or lwc, (defined as the mass of water contained in a unit volume of air), the air speed and the object's collision efficiency. The density of the accreted ice also determines the thickness and shape of the deposit.

On rotating cylinders, accreted rime ice densities have been examined by Macklin (1962), and by Makkonen and Stallabrass (1985). On non-rotating objects, the shape of the deposit is also partly dependent on the variation of the local ice density along its surface. Bain and Gayet (1982) have suggested one form for this variation.

If the flux of super-cooled water is large enough so that the latent heat of freezing cannot be radiated, conducted or convected away from the surface fast enough, then the surface temperature may rise to the melting point. When this happens the ice may become 'wet', rather than 'dry', meaning that all of the impinging water is no longer able to freeze on impact. Now the problem becomes even more complex since a detailed heat balance must be calculated for the accretion surface, in order to determine how much of the impinging water is able to freeze, and how much is shed, or remains in the accretion as liquid water. Calculation of the heat balance is made especially difficult since the heat transfer properties of the ice surfaces and even of rough circular cylinders are very poorly known (Makkonen, 1985).

Models in which a computation of the heat balance on cylinders is attempted using empirical heat transfer data include those of Lozowski, Stallabrass and Hearty (1984), Macklin and Payne (1967), Egelhofer, et al. (1984), and Makkonen (1984). However, such data are scarce, especially for non-cylindrical shapes, and for rough surfaces. Some modelers have thus used numerically calculated heat transfer for cylinders (Makkonen, 1985) or for airfoils (Cansdale and McNaughton, 1977, Cansdale and Gent 1983).

Models developed specifically for marine applications must contend with the further complications of salinity, droplet distributions and liquid water contents that vary with height, and intermittent sprays. Models including these effects are currently under development by various researchers.

The models developed for this thesis are intended as 'operational' icing models, which would be efficient and inexpensive enough to be used regularly in the laboratory for predictive purposes, or for numerical experiments into specific aspects of icing. Previous models have often been much too costly to be used in such a way. To reduce computational time several 'shortcuts', or parameterizations, have been introduced which help to simplify some of the calculations needed in this problem. Specifically, the local collision efficiency and local ice density are approximated by analytical functions of the surface slope. Such an approach naturally involves some loss of accuracy, but this loss must be balanced against unavoidable uncertainties in the input data and the required precision of output predictions.

Two models are described, for circular cylinders in Chapter III, and in Chapter V for airfoils, that could also be applied to other arbitrary shapes. Both models are limited to rime ice accretion, i.e. there is no heat balance calculation, and a uniform air speed and flux of pure water droplets are assumed. This is so that the performance of the parameterizations can be easily compared with real ice accretions whose shape and mass are not dependent on the poorly known heat transfer distribution.

In the course of model development, significant improvements were made in two specific aspects, the theoretical calculation of collision efficiencies on smooth circular cylinders, (Chapter II) and the distribution of local density in non-rotating circular cylinder rime accretions (Chapter IV).

Experiments for model verification were carried out in the University of Alberta's original icing wind-tunnel, known as the Facility for Research On Solidification and Thawing, or FROST tunnel. A full description of the facility may be found in Gates (1981), or Lozowski and Gates (1984). All wind-tunnel experiments were performed under the direction of Professor Gates.

References

Bain, M. and Gayet, J-F. 1982: Contribution to the Modelling of the Ice Accretion Process: Ice Density Variation with the Impacted Surface Angle. Proceedings, First International Workshop on Atmospheric Icing of Structures, Hanover, 13-20.

Cansdale, J.T. and Gent, R.W. 1983: Ice Accretion on Aerofoils in Two-Dimensional Compressible Flow - A Theoretical Model. British Royal Aircraft Establishment Technical Report 82128, 31 pp. + 31 fig.s.

Cansdale, J.T. and McNaughtan, I.I. 1977: Calculation of Surface Temperature and Ice Accretion Rate in a Mixed Water Droplet / Ice Crystal Cloud. British Royal Aircraft Establishment, Technical Report 77090, 24 pp. + 6 fig.s.

Egelhofer, K.Z., Ackley, S.F. and Lynch, D.R. 1984: Computer Modelling of Atmospheric Ice Accretion and Aerodynamic Ice Loading of Transmission Lines. Proceedings, Second International Workshop on Atmospheric Icing of Structures, Trondheim, (in press).

Gates, E.M. 1981: FROST Tunnel. Departmental Report No. 26, Department of Mechanical Engineering, The University of

Alberta. 10 pp.

Gent, R.W. and Cansdale, J.T. 1985: The Development of Mathematical Modelling Techniques for Helicopter Rotor Icing. AIAA Publication 85-0336, 10 pp.

Hess, J.L. and Smith, A.M.O. 1967: Calculation of Potential Flow About Arbitrary Bodies, *Progress in Aeronautical Sciences, Vol. 8*, Pergamon Press, 1-138.

Horjen, I. 1983: Icing on Offshore Structures Atmospheric Icing. *Norwegian Maritime Research*, II, No. 3, 9-21.

Kennedy, J.L. and Marsden D.J. 1976: Potential Flow Velocity Distributions on Multi-Component Airfoil Sections. *Can.Aero. and Space Jour.*, 22, 243-256.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. Collected Works of Irving Langmuir, Vol.10, Pergamon Press, 348-393.

Lozowski, E.P. and Gates, E.M. 1984: Ice Accretion in a Marine Environment. Proceedings, 3rd International Symposium on Offshore Mechanics and Arctic Engineering, New Orleans, 8 pp.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

MacArthur, C.D. Keller, J.L. and Luers, J.K. 1982: Mathematical Modelling of Ice Accretion on Airfoils. AIAA Publication No. 82-0284.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.*, 38, 1027-1036.

Macklin, W.C. 1962: The Density and Structure of Ice Formed by Accretion. *Quart. Jour. Roy. Meteor. Soc.* 88, 30-50.

Macklin, W.C. and Payne, G.S. 1967: A Theoretical Study of the Ice Accretion Process. *Quart. Jour. Roy. Meteor. Soc.*, 93, 195-213.

Makkonen, L. 1984: Modelling of Ice Accretion on Wires. *Jour. Climate and App. Meteor.*, 23, 929-939.

Makkonen, L. 1985: Heat Transfer and Icing of a Rough Cylinder. *Cold Regions Science and Technology*, 10, 105-116.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Mechanical Engineering Division, Technical Report TR-LT-005, 50 pp.

Poots, G. 1984: Theoretical Solutions of Ice Accretion on Cables. Proceedings, Second International Workshop on Atmospheric Icing of Structures, (in press).

Ranz, W.E. and Wong, J.B. 1952: Impaction of Dust and Smoke Particles. *Industrial Engineering Chemistry*, 44, 1371-1381.

II. Cloud Droplet Trajectories Updated

A. Introduction

For many years the standard reference work for cloud droplet trajectories in an airstream flowing around an obstacle has been Langmuir and Blodgett (1946), hereafter LB. This was one of the first published treatments of the problem which used an analog computer to integrate the equations of motion for water droplets in a steady potential flow about an infinitely long circular cylinder.

These calculations are of interest in studies of atmospheric and marine ice accretion, in which the characteristics of droplet impingement on cylinders and other objects are important factors in modelling the icing process. However, since the calculations were done so long ago, on what would now be considered a primitive machine, and because the original reference can be difficult to obtain, a need was perceived to recalculate the droplet trajectories and to publish the results in the open literature. The effect of replacing the drag coefficient formulation used by LB with a more recent one, that of Beard and Pruppacher (1969), has also been investigated.

Functional fits have been devised by several authors to the original tabulated data of LB. A number of these are compared to show the need for improving and standardizing them. Toward this end, a non-linear regression fit to the new data has been found for the range of conditions pertaining to airborne, land structure and marine icing.

B. Impingement Parameters and Droplet Trajectories

From trajectory calculations may be derived the overall collision efficiency, E, the local collision efficiency at a point on the surface represented by its local surface slope, $\beta(\alpha)$, the local collision efficiency at the stagnation line, β_0, the maximum impingement angle on the cylinder surface, α_{max}, and the non-dimensional impact velocity at the stagnation line, V_0. V_0 is normalized by the free stream speed, U. The local surface angle, α, is defined as the angle between the local surface normal and the free stream direction. The other quantities are defined in Figure II.1.

Values of E, β_0, α_{max} and V_0 are given by LB for a large number of cases which vary in droplet size, cylinder size, and air speed. These results are tabulated in terms of the non-dimensional parameters, K:

$$K = (\rho_d D_d^2 U) / (9 \mu D_c), \qquad \text{II.1}$$

and ϕ:

$$\phi = Re^2 / K, \qquad \text{II.2}$$

with the droplet Reynolds number based on the free-stream velocity, Re, given by:

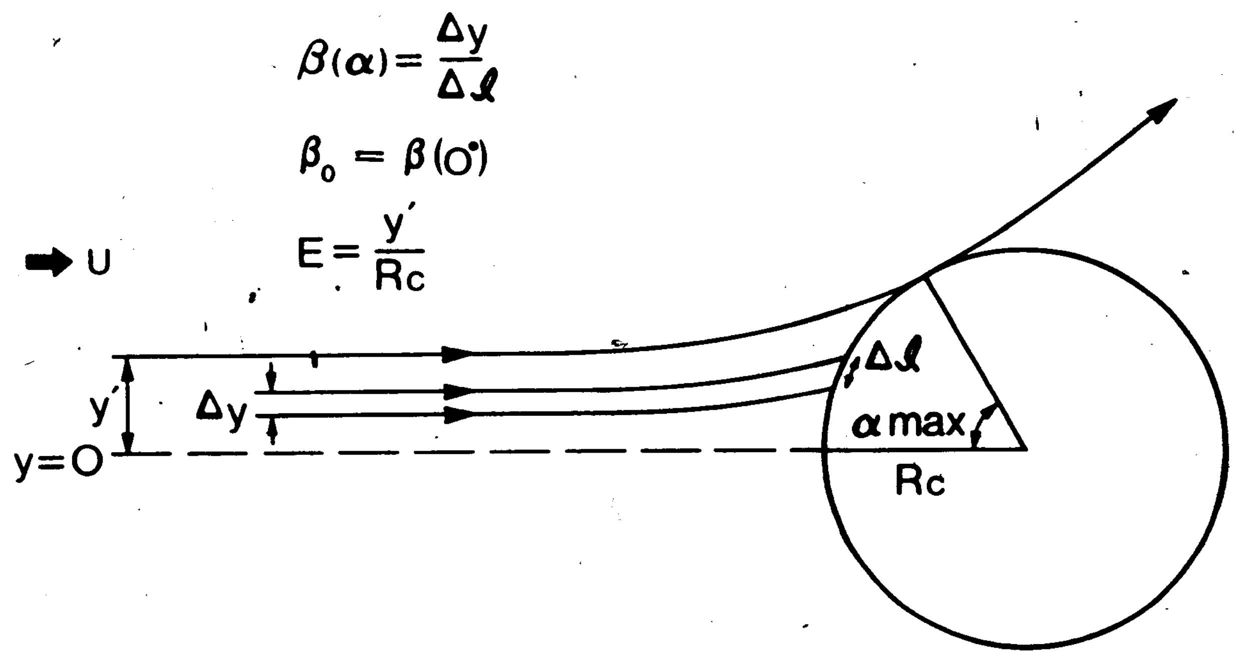


Figure II.1 - Droplet trajectories impinging on a circular cylinder, giving the definitions of $\beta(\alpha)$, β_0, E and α_{max}. On the accretion surface α is the angle between the local surface normal and the free stream direction.

$$Re = U D_d \rho_a / \mu. \qquad \text{II.3}$$

Here ρ_d is the density of water, ρ_a the density of air, μ the dynamic viscosity of air, D_d the droplet diameter, D_c the cylinder diameter, and U the free stream air speed.

The K parameter measures the inertia of the droplet, and is also known as the Stokes number. To aid in solving problems in which the droplet size is the unknown quantity, LB presented their data in terms of a second non-dimensional parameter , ϕ, that is independent of D_d.

The case $\phi = 0$ corresponds to Stokes, or creeping flow:

$$C_d \, Re_d / 24 = 1, \qquad \text{II.4}$$

where C_d is the droplet drag coefficient, and Re_d is the droplet Reynolds number based on the relative velocity with respect to the air, V, so that:

$$Re_d = |V| Re / U. \qquad \text{II.5}$$

In component form, the droplet equations of mot

$$dv_x / dt = -(v_x - u_x) (U C_d Re_d) / (12 D_c K)$$

$$dv_y / dt = -(v_y - u_y) (U C_d Re_d) / (12 D_c K),$$

where t is time, v_x, v_y and u_x, u_y are the droplet and air speed components, respectively. This set of equations is in dimensional form, with the origin at the cylinder axis and U in the positive x direction. Gravity and the effects of droplet acceleration on the drag have been neglected, as in the original theory of LB.

The air speed components for potential flow around a cylinder (Green, 1937) are:

$$u_x(x,y) = U\ (1 + (R_c^{\,2}\ (y^2 - x^2)\ /\ (x^2 + y^2)^2))$$

$$u_y(x,y) = -(2\ U\ R_c^{\,2})\ (x\ y)\ /\ (x^2 + y^2)^2. \qquad \text{II.7}$$

In order to permit a direct comparison with the original solutions by LB some trajectories have been integrated using their empirical formulation for the steady-state drag coefficient:

$$C_d\ Re_d\ /\ 24 = 1 + 0.197\ Re_d^{\,0.63} + 2.6*10^{-4}\ Re_d^{\,1.38} \qquad \text{II.8}$$

More recent studies have improved the accuracy of C_d approximations at low Reynolds numbers, however, so in further calculations over the complete range of conditions the formulation of Beard and Pruppacher (1969) has been substituted when Re_d is less than 200. This formulation is:

$$C_d \, Re_d/24 = 1 + 0.102 \, Re_d^{0.955}, \text{ for } 0.2 \le Re_d \le 2.0$$

$$C_d \, Re_d/24 = 1 + 0.115 \, Re_d^{0.802}, \text{ for } 2.0 \le Re_d \le 21.0 \qquad \text{II.9}$$

$$C_d \, Re_d/24 = 1 + 0.189 \, Re_d^{0.632}, \text{ for } 21.0 \le Re_d \le 200.0$$

A comparison of a variety of theoretical and experimental results for the drag coefficients of spheres is given by Beard (1976), in which the above formulation is shown to be an accurate analytical approximation.

The integration is carried out by the Heun method as described in Mesinger and Arakawa (1976). The droplet trajectories are begun at ten cylinder radii upstream of the origin with speeds at the initial position (x_0, y_0) assumed to be:

$$v_x(x_0, y_0) = u_x(x_0, y_0) \qquad \text{II.10}$$

$$v_y(x_0, y_0) = 0.5 \, u_y(x_0, y_0)$$

At this initial upstream distance, the final results are not greatly influenced by the choice of initial droplet speed as long as it lies in between the local air speed and the undisturbed upstream air speed. Tests indicate differences in collision efficiency of less than 0.5 percent, when the initial droplet speeds are varied within this range.

Other factors which influence the accuracy of the integration are the time step, Δt, and the initial upstream distance, x_0. A value of $x_0 = 10$ cylinder radii has been chosen, which, when compared to the results with $x_0 = 20$ cylinder radii, agrees to within 0.7 percent. At this initial distance, the size of the time step made less than a 0.5 percent difference in the results, as long as $\Delta t < 0.0033\ (D_c/U)$. The value used is $0.0025\ (D_c/U)$. Double precision is used throughout the program.

Values of the air viscosity and air density used are based on an assumed air temperature of -10° C. Within the temperature range of interest for most dry icing, i.e. -20 to 0° C, the collision efficiencies vary by a maximum of ±1 percent.

An air pressure of 100 kPa was assumed. The calculations of LB were carried out for an atmospheric pressure of 78.5 kPa, for application to Mt. Washington, or to airborne icing in general. At this pressure value, the present results differed from the 100 kPa results by less than 0.5 percent. The new results should therefore be applicable to both ground-level and airborne icing.

Two complete trajectories must be integrated in order to calculate one value of the local collision efficiency $\beta(\alpha)$, where α is the mean angle of the two impact points. As illustrated in Figure II.1, $\beta(\alpha)$ is defined as the ratio of the initial vertical separation of the two trajectories, to the final separation in arc length of the two impact points. In the present model, the initial y separation of the two trajectories is always 10^{-8} metres, an arbitrarily small distance for use in the finite difference approximation to $dy/d\ell$, the definition of local collision efficiency. The stagnation line value, β_0, is calculated from the case where the first trajectory's initial position $y_0 = 0$.

The maximum impingement angle, α_{max}, is found from the droplet trajectory which is tangent to the cylinder surface at its point of impact. This 'grazing' trajectory is found by increasing the initial position y_0 by successively smaller amounts until the tangent trajectory is reached.

The total collision efficiency E may then be calculated from the ratio of the initial vertical distance y_0(graze) to the cylinder radius.

The Fortran77 code written for these calculations is included in Appendix I.

C. Results

In Figure II.2 three curves of β_0 vs. K are shown for the value $\phi = 10^3$, to illustrate the differences between LB's original calculations, the re-calculation of their results, and the re-calculation using Beard and Pruppacher's formulation for drag coefficient. The data for this Figure are listed in Table II.1.

A comparison between the first two curves shows that the LB results are too large for small K ($K \leq 0.4$), by up to

K	β_o (LB), old C_d	β_o (new), old C_d	β_o (new), new C_d
.144	0.048	0.007	0.011
.196	0.095	0.058	0.073
.256	0.146	0.116	0.137
.4	0.233	0.224	0.247
.625	0.323	0.335	0.359
.9	0.398	0.425	0.447
1.6	0.513	0.558	0.575
3.6	0.680	0.714	0.724
6.4	0.778	0.798	0.806
10.	0.837	0.848	0.854
19.6	0.901	0.902	0.906
32.4	0.932	0.929	0.932
62.5	0.959	0.953	0.955
90.	0.969	0.962	0.963
160.	0.980	0.972	0.973
360.	0.989	0.981	0.981
640.	0.993	0.984	0.984
1000.	0.995	0.986	0.986

Table II.1 - $\beta_0(K)$ for $\phi = 10^3$. A comparison between Langmuir and Blodgett's results and the new results.

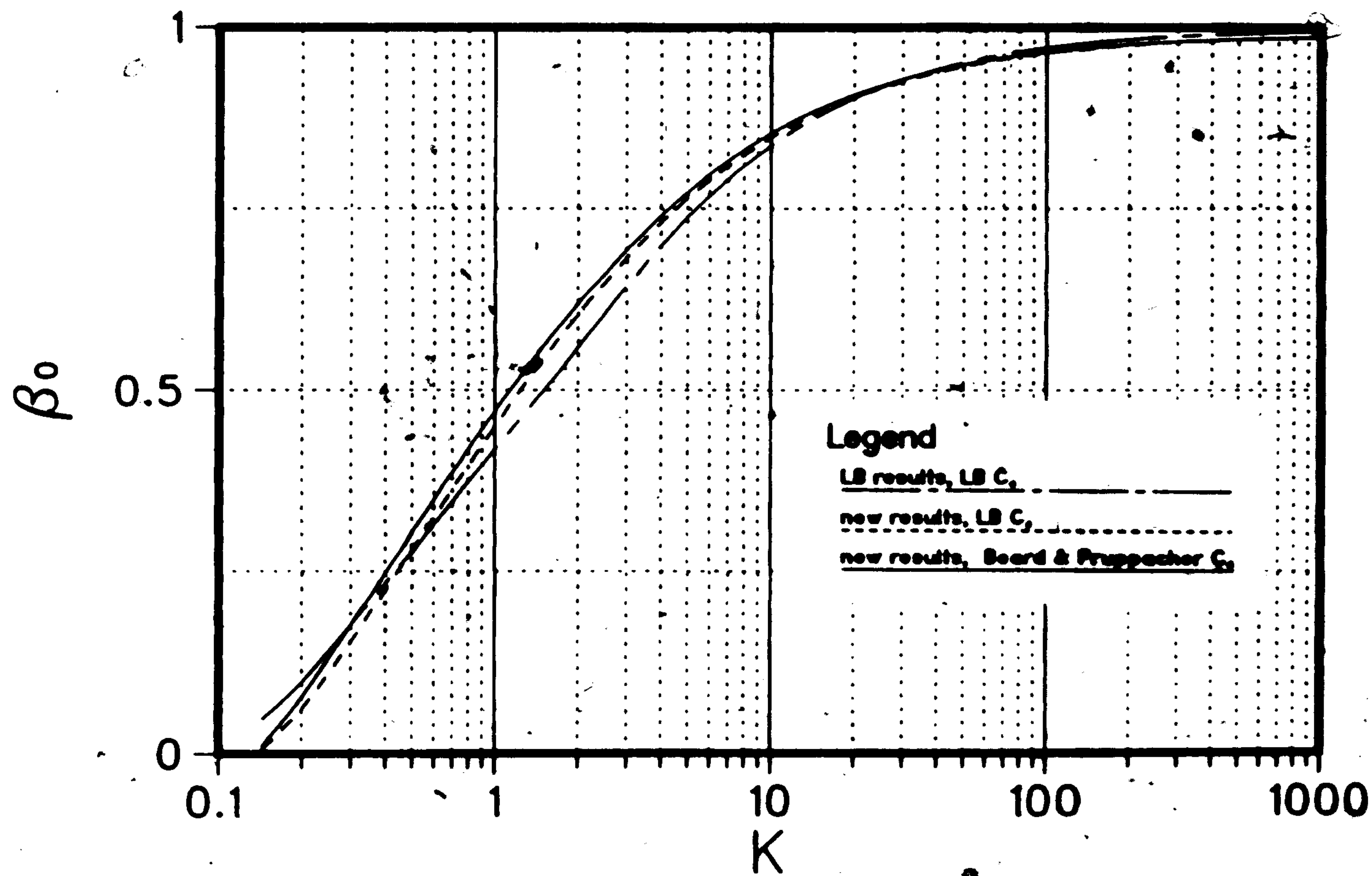


Figure II.2 - Comparison of β_0 vs. K for one value of $\phi = 10^3$.

several hundred percent. In the range $0.4 < K < 20.0$ their results are too small, by amounts near 10 percent in the middle of the range. For larger K, their values are again larger than the new results, but only by a few percent. The use of the revised drag coefficients (dotted curve, Figure II.2) gives another small increase over the LB results for all $K < 200.0$. A noticeable difference is that the new results for β_0 reach an effective value of zero for $K \simeq 0.14$, rather than at LB's theoretical value of 0.125. This difference is as yet unexplained. Results for other ϕ values, and for α_{max}, E and V_0 are quantitatively similar in differences between the old and new results.

It is concluded from the comparisons in Figure II.2 and Table II.1, that the original LB calculations are accurate only for very large K, but that when K is less than about 200 they may be significantly in error. The new results are suggested for future use. These are shown for β_0, α_{max}, E and V_0 for a complete range of ϕ and K in Tables II.2 to II.3, and in Figures II.3 through II.6, where the tabulated data have been fit by spline curves.

D. Curve Fitting

LB also drew smooth curves through their data, and derived analytical expressions in K and ϕ to fit them. Subsequently other authors have made attempts to improve on or to simplify these relations, and have then applied them

K	φ = 0			φ = 10			φ = 10^3			φ = 10^5		
	β_o	α_{max}	E	β_o	α_{max}	E	β_o	α_{max}	E	β_o	α_{max}	E
.144	0.014	0.018	0.0002	0.013	0.009	0.0001	0.011	0.015	0.0001	0.004	0.007	0.00002
.196	0.094	0.170	0.012	0.091	0.159	0.011	0.073	0.131	0.007	0.030	0.054	0.001
.256	0.175	0.328	0.043	0.170	0.310	0.040	0.137	0.247	0.025	0.058	0.105	0.004
.4	0.313	0.580	0.131	0.304	0.553	0.122	0.247	0.436	0.079	0.113	0.195	0.016
.625	0.446	0.799	0.249	0.432	0.760	0.233	0.359	0.612	0.157	0.179	0.302	0.039
.9	0.545	0.953	[illegible]	0.530	0.905	0.333	0.447	0.742	0.234	0.244	0.400	0.070
1.6	0.680	1.150	[illegible]	0.663	1.095	0.492	0.575	0.921	0.365	0.353	0.559	0.142
3.6	0.821	1.345	0.718	0.805	1.298	0.686	0.725	1.136	0.553	0.508	0.764	0.274
6.4	0.886	1.433	0.818	0.872	1.383	0.789	0.806	1.255	0.670	0.601	0.898	0.376
10.	0.921	1.476	0.873	0.909	1.439	0.848	0.854	1.329	0.745	0.670	0.997	0.459
19.6	0.954	1.521	0.927	0.945	1.489	0.908	0.906	1.413	0.832	0.759	1.130	0.580
32.4	0.968	1.538	0.951	0.961	1.513	0.936	0.932	1.457	0.878	0.812	1.214	0.661
62.5	0.979	1.552	0.970	0.974	1.532	0.959	0.955	1.496	0.920	0.866	1.303	0.749
90.	0.983	1.556	0.976	0.979	1.539	0.967	0.963	1.512	0.935	0.889	1.347	0.789
160.	0.986	1.563	0.982	0.984	1.545	0.976	0.973	1.530	0.953	0.917	1.400	0.840
360.	0.989	1.567	0.987	0.987	1.553	0.983	0.981	1.546	0.969	0.945	1.454	0.892
640.	0.990	1.568	0.988	0.989	1.559	0.986	0.984	1.553	0.976	0.957	1.481	0.917
1000.	0.990	1.568	0.989	0.989	1.557	0.987	0.986	1.557	0.980	0.965	1.499	0.932

Table II.2 - Numerically calculated values of β_0, α_{max} and E as a function of K and ϕ.

K	$\varphi = 10^2$			$\varphi = 10^4$			$\varphi = 10^6$		
	β_o	α_{max}	E	β_o	α_{max}	E	β_o	α_{max}	E
.16	0.034	0.054	0.001	0.021	0.035	0.0006	0.005	0.011	0.00004
.25	0.153	0.282	0.032	0.094	0.168	0.012	0.026	0.051	0.001
.36	0.257	0.466	0.088	0.164	0.287	0.034	0.050	0.089	0.003
.64	0.415	0.720	0.212	0.281	0.468	0.095	0.103	0.176	0.013
1.	0.528	0.889	0.325	0.374	0.609	0.163	0.157	0.260	0.029
1.96	0.676	1.097	0.500	0.518	0.817	0.296	0.251	0.380	0.067
3.24	0.763	1.219	0.617	0.617	0.958	0.406	0.319	0.488	0.112
6.25	0.848	1.336	0.743	0.726	1.115	0.547	0.424	0.633	0.191
9.	0.882	1.387	0.797	0.777	1.187	0.618	0.482	0.712	0.243
16.	0.921	1.446	0.863	0.838	1.278	0.711	0.570	0.832	0.333
36.	0.955	1.501	0.922	0.899	1.379	0.812	0.680	0.982	0.463
64.	0.968	1.523	0.946	0.927	1.434	0.864	0.745	1.080	0.550
100.	0.975	1.536	0.959	0.944	1.465	0.894	0.787	1.143	0.612
196.	0.982	1.548	0.972	0.960	1.497	0.927	0.839	1.227	0.693
324.	0.985	1.554	0.978	0.969	1.517	0.944	0.869	1.282	0.744
625.	0.987	1.558	0.983	0.977	1.533	0.959	0.900	1.337	0.799
900.	0.988	1.559	0.985	0.980	1.538	0.966	0.913	1.366	0.824

Table II.2 - continued.

K	V_0, $\varphi = 0$	V_0, $\varphi = 10$	V_0, $\varphi = 10^3$	V_0, $\varphi = 10^5$	K	V_0, $\varphi = 10^2$	V_0, $\varphi = 10^4$	V_0, $\varphi = 10^6$
.144	0.0002	0.0002	0.0002	0.0001	.16	0.002	0.0009	0.0001
.196	0.018	0.017	0.012	0.003	.25	0.051	0.021	0.003
.256	0.065	0.061	0.041	0.010	.36	0.133	0.059	0.010
.4	0.195	0.183	0.122	0.033	.64	0.297	0.144	0.029
.625	0.348	0.327	0.227	0.068	1.	0.428	0.228	0.055
.9	0.471	0.443	0.317	0.111	1.96	0.604	0.379	0.116
1.6	0.638	0.604	0.459	0.2[illegible]	3.24	0.708	0.491	0.162
3.6	0.804	0.772	0.640	0.346	6.25	0.810	0.625	0.251
6.4	0.879	0.852	0.742	0.447	9.	0.853	0.690	0.306
10.	0.918	0.896	0.806	0.528	16.	0,904	0.768	0.396
19.6	0.956	0.940	0.876	0.642	36.	0.948	0.854	0.520
32.4	0.973	0.960	0.913	0.715	64.	0.967	0.897	0.603
62.5	0.986	0.977	0.945	0.793	100.	0.977	0.922	0.659
90.	0.990	0.983	0.957	0.828	196.	0.986	0.948	0.732
160.	0.994	0.989	0.971	0.872	324.	0.990	0.962	0.777
360.	0.997	0.995	0.983	0.91[illegible]	625.	0.994	0.975	0.826
640.	0.999	0.997	0.989	0.936	900.	0.996	0.980	0.849
1000.	0.999	0.998	0.992	0.950				

Table II.3 - Numerically calculated values of V_0 as a function of K and ϕ.

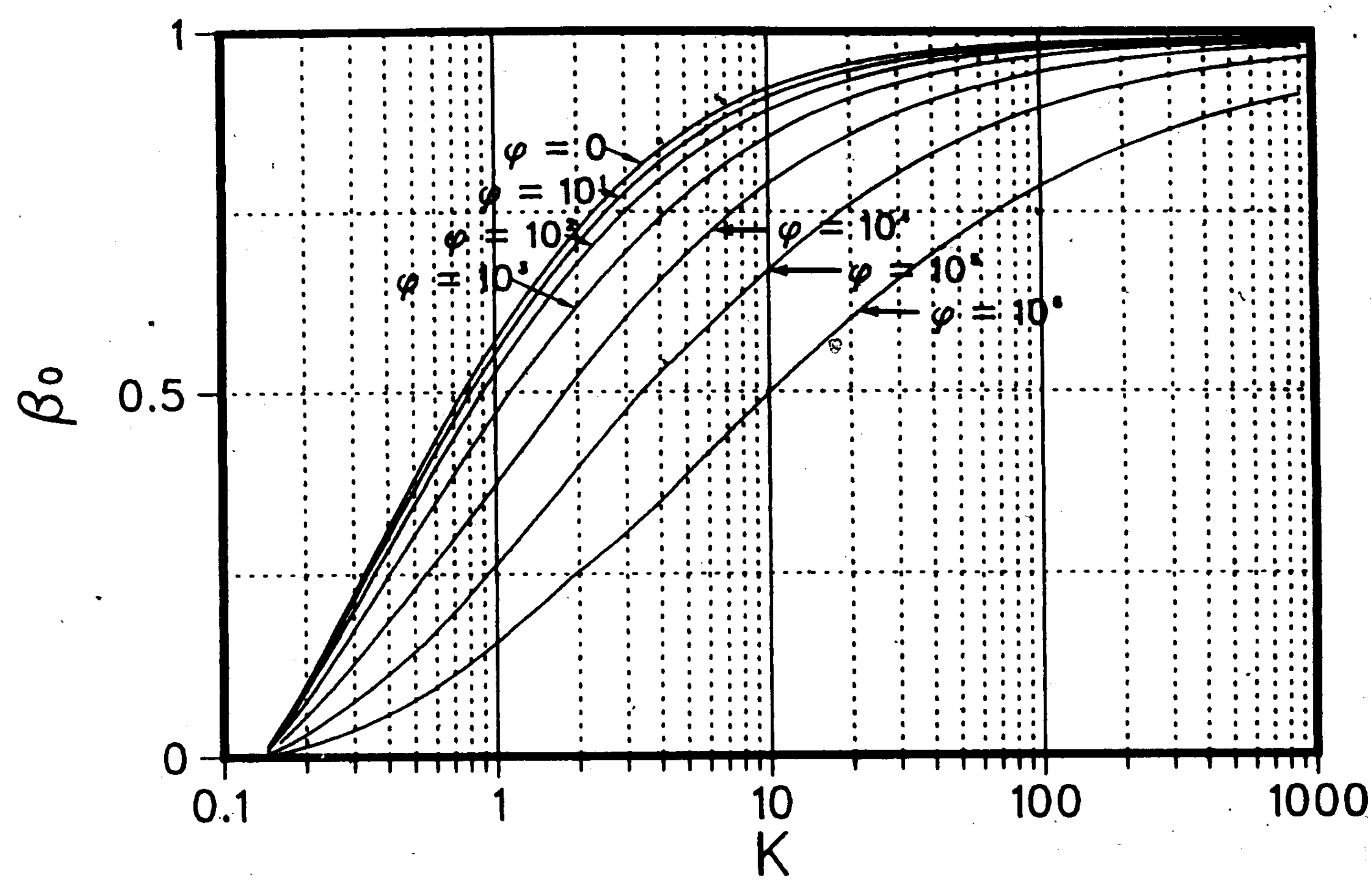


Figure II.3 - Smooth curves drawn for the numerical data of β_0 vs. K and ϕ.

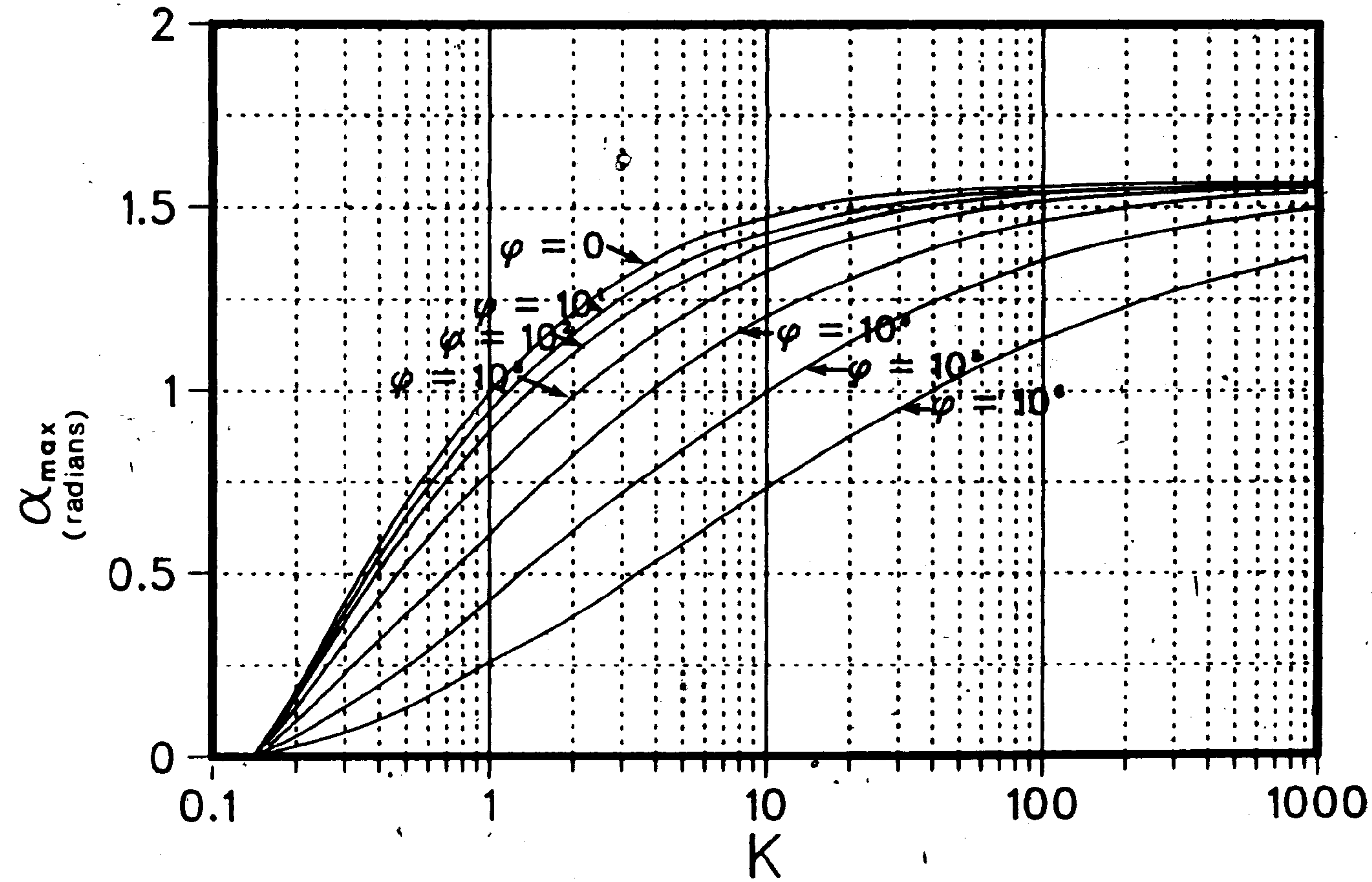


Figure II.4 - Smooth curves drawn for the numerical data of α_{max} vs. K and ϕ.

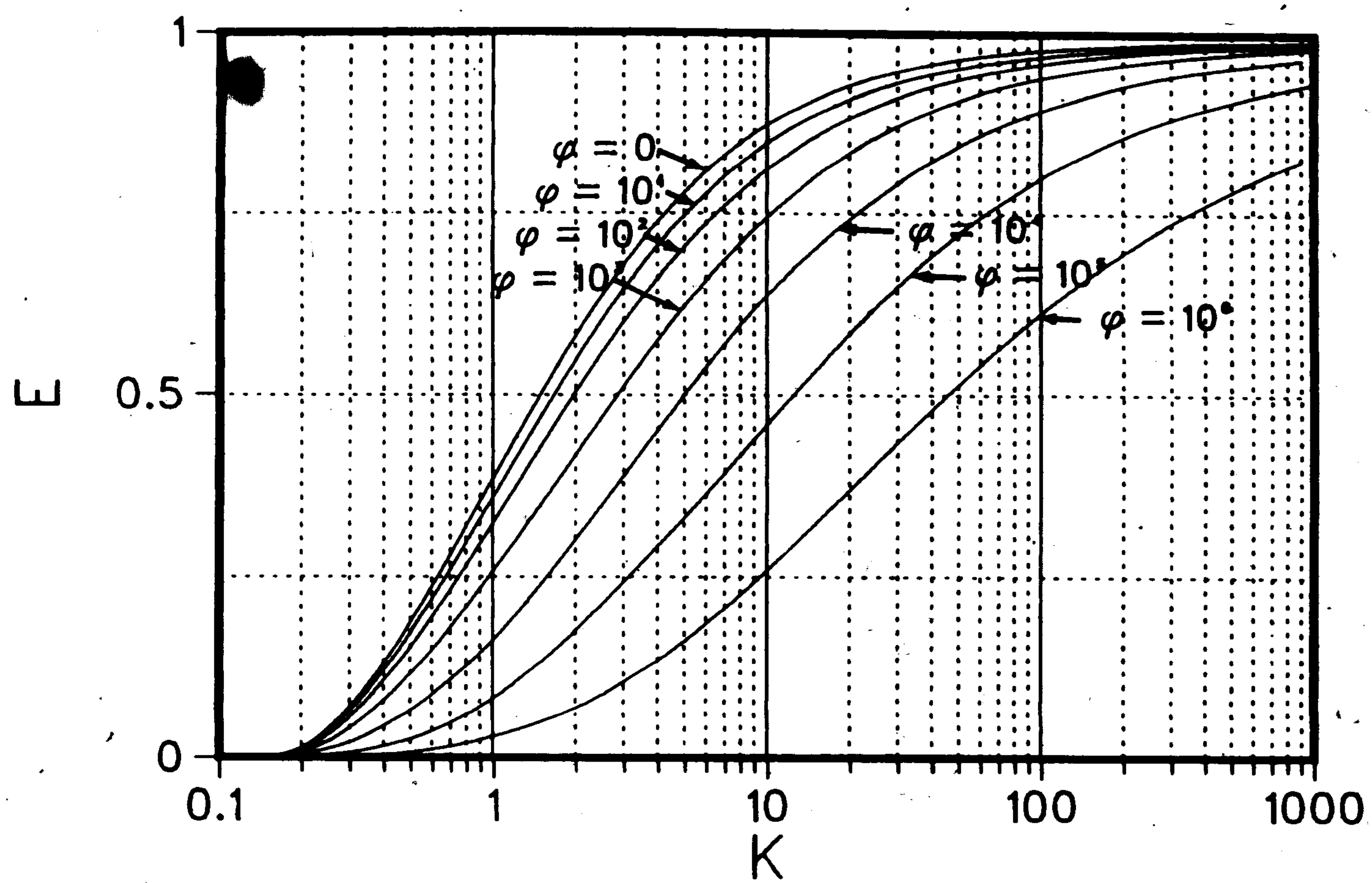


Figure II.5 - Smooth curves drawn for the numerical data of E vs. K and ϕ.

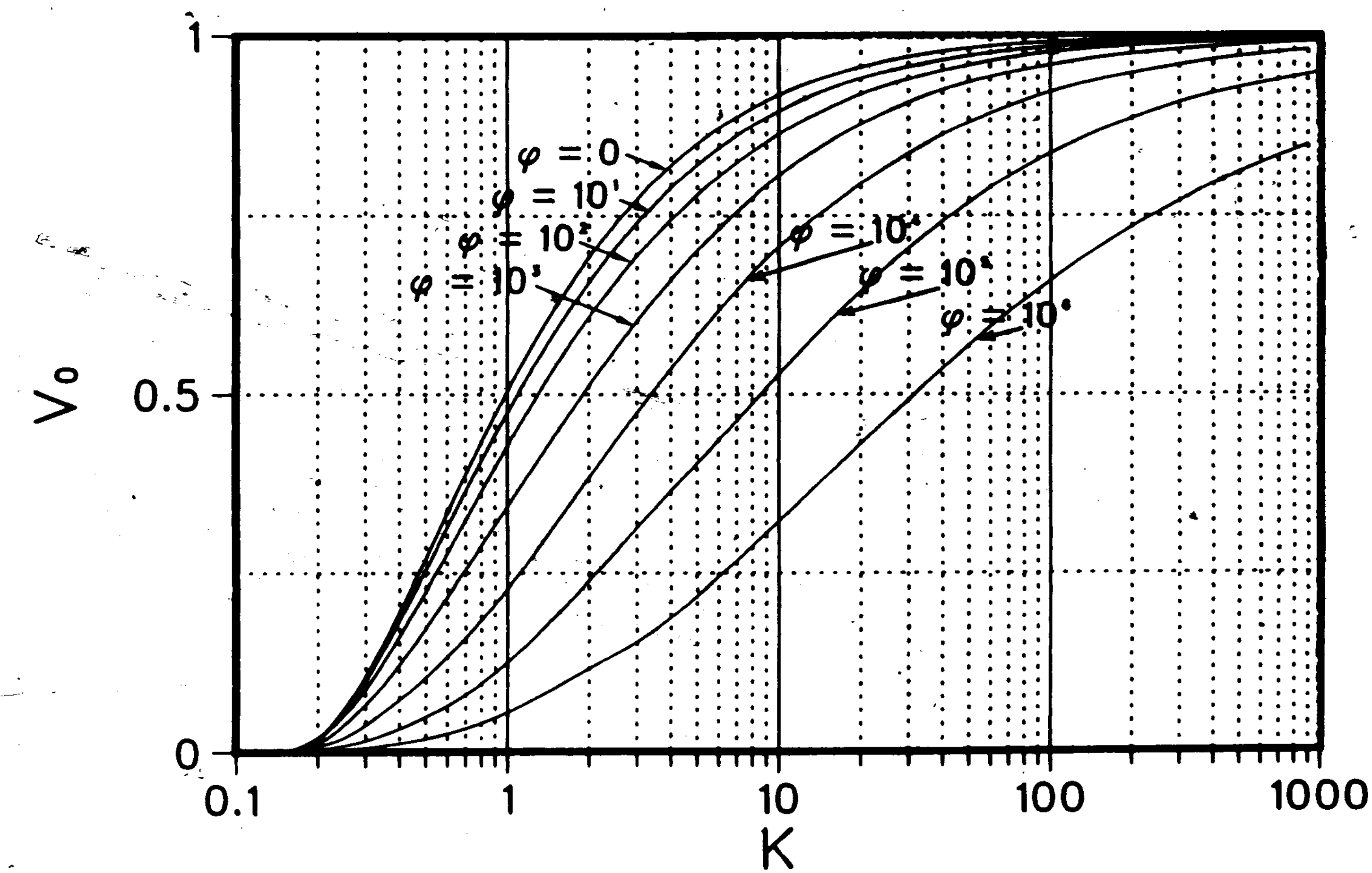


Figure II.6 - Smooth curves drawn for the numerical data of V_0 vs. K and ϕ.

Author(s)	$K_o(K,Re)$, $\zeta(U,D_d,D_c)$	$E(K_o)$, $E(K,Re)$, $E(\delta)$	Range
Langmuir and Blodgett	$K_o = .125 + \dfrac{(K - .125)}{1 + .0967\ Re^{.6367}}$	$E = .466\ (\log_{10}(8K_o))^2$ $E = \dfrac{K_o}{K_o + \frac{\pi}{2}}$ $E = \dfrac{K}{K + \frac{\pi}{2}\ .112\ Re^{.63} + .75\ 10^{-4}\ Re^{1.38}}$	$.125 < K_o < 1.1$ $K_o > 1.1$ $E > 0.5$
Lozowski, Stallabrass and Hearty	$K_o = .125 + \dfrac{(K - .125)}{1 + .0967\ Re^{.6367}}$	$E = .489\ (\log_{10}(8K_o))^{1.978}$ $E = \dfrac{K_o}{K_o + \frac{\pi}{2}}$	$.125 < K_o < 0.9$ $K_o > 0.9$
Cansdale and McNaughton	$K_0 = \dfrac{K}{1 + .087\ Re^{(.76\ Re^{-.027})}}$	$E = .53\ (\log_{10}(8K_o))^{1.8}$ $E = \dfrac{K_o^{1.1}}{K_o^{1.1} + 1.223}$	$.125 < K_o < 1.1$ $K_o > 1.1$
Makkonen	$K_o = \dfrac{K}{1 + .087\ Re^{(.76\ Re^{-.027})}}$	$E = .5\ (\log_{10}(8K_o))^{1.6}$ $E = \dfrac{K_o^{1.1}}{K_o^{1.1} + 1.426}$	$.125 < K_o < 0.8$ $K_0 > 0.8$
Stallabrass	$\zeta = \dfrac{U^{.6}\ D_d^{\ 1.6}}{D_c}$	$E = \dfrac{\zeta - 3200}{\zeta + 27000}$ $E = 0$	$\zeta > 3200$ $\zeta \leq 3200$

Table II.4 - Approximations to the numerical data of Langmuir and Blodgett derived by various authors.

to various numerical icing models (Cansdale and McNaughtan, 1977, Stallabrass, 1980, Lozowski, Stallabrass and Hearty, 1983, Makkonen, 1984). A number of these fitted relations for $E(K,\phi)$ are listed in Table II.4.

In order to compare these relations, values computed from each are listed with a selection of the original data in Table II.5. The data all fall within a range of ϕ between 10^2 and 10^4, since some of the fitting functions were specifically designed for similar ranges, and because this range should apply to most physical applications, corresponding to to $10 \le D_d \le 4000$ microns, $0.01 \le D_c \le 1.0$ metre, $5 \le U \le 200$ metres per second and $-20°C \le T \le 0°C$. It is worth noting that the natural range is actually narrower than this, since the largest drops do not occur together with the highest wind speeds (e.g. marine icing), nor do the smallest droplets occur at the lowest speeds (e.g. aircraft icing).

Langmuir and Blodgett's own fit is the most accurate, since it includes a correction table for $E < 0.5$, and for larger values of K_ϕ (giving E values > 0.5), they give theoretically derived functions for β_0, α_{max}, and E, which assume essentially straight-line trajectories. For this more elaborate scheme, the standard error (computed with respect to the numerical data) is generally an order of magnitude smaller than it is for the other fitting schemes. Even more important, however, is the fact that the variation among the values calculated by the various empirical formulae for a given case can exceed 10 percent. These discrepancies will add to the difficulty of making comparisons among the icing predictions made by the various models which use these relations.

To reduce any further confusion, the new data have been fitted with approximating functions of improved form and greater accuracy. These functions are described below.

The approach taken by other authors to derive the formulae of Table II.4 has been to collapse a family of curves, such as that in Figure II.3, into a single curve via the ratio K_0/K, calculated as a function of Re, or in the case of Stallabrass (1980), via the parameter ζ, a function of D_d, U, and D_c (see Table II.4). K_0 is the K value read from the $\phi = 0$ curve which yields the same value of β_0, α_{max}, E or V_0 as do the original K and ϕ. Extensive manipulation of the new data has revealed major problems with this approach. Specifically, the three families of curves for β, α_{max} and E are not collapsible by the same ratio K_0/K, but require three different ratios. Only LB take this into account for small values of E, by using their correction table, and theoretical approximations for larger E. Furthermore, the K_0/K's are *not* single-valued functions of Re (see Figure II.7), as has usually been assumed.

In the Stallabrass (1980) method, although ζ is used instead of K and ϕ, a similar problem arises for his curves, since they collapse to a single curve only for very large values of ζ.

φ	K	LB (numerical)	LB (approx.)	Lozowski, et al.	Cansdale	Makkonen	Stallabrass
10^2	.16	0.004	0.004	0.004	0.002	0.005	0.056
	.64	0.181	0.183	0.181	0.191	0.202	0.284
	3.24	0.591	0.588	0.568	0.630	0.594	0.634
	16.	0.855	0.854	0.836	0.884	0.867	0.867
	100.	0.966	0.965	0.958	[illegible]	0.972	0.966
	625.	0.992	0.992	0.989	0.995	0.994	0.992
10^3	.196	0.009	0.009	0.010	0.0	0.0	0.017
	.625	0.127	0.129	0.132	0.120	0.134	0.170
	3.6	0.542	0.540	0.509	0.560	0.522	0.536
	19.6	0.830	0.825	0.796	0.852	0.831	0.827
	90.	0.942	0.939	0.925	0.956	0.949	0.943
	640.	0.986	0.985	0.942	0.992	0.990	0.988
10^4	.25	0.012	0.013	0.014	0.0	0.0	0.0
	1.	0.148	0.142	0.146	0.120	0.134	0.152
	6.25	0.546	0.539	0.499	0.557	0.518	0.531
	36.	0.820	0.811	0.778	0.850	0.829	0.831
	196.	0.937	0.934	0.921	0.960	0.954	0.951
	900.	0.974	0.974	0.972	0.988	0.987	0.985
	standard error:		0.003	0.018	0.017	0.010	0.019

Table II.5 - Comparison of some original E values calculated numerically by Langmuir and Blodgett, and those derived from the approximations of Table II.4.

For these reasons, an accurate fit to the entire range of data given is difficult to find and awkward to use. Consequently, an approximating function is given here only for ϕ between 10^2 and 10^4, which is the range applicable to most atmospheric, land-based and marine icing problems.

A non-linear regression analysis on this range gives the following result for $0.17 \leq K \leq 10^3$:

$$X(K,\phi) = (C_{X,1}\, K^{C_{X,2}} \exp(C_{X,3}\, K^{C_{X,4}} + C_{X,5})$$

$$- (C_{X,6}(\phi - 100)^{C_{X,7}}) \qquad \text{II.11}$$

$$(C_{X,8}\, K^{C_{X,9}} \exp(C_{X,10}\, K^{C_{X,11}}) + C_{X,12}),$$

where X is either β_0, α_{max}, E or V_0, and the constants $C_{X,n}$ are as listed in Table II.6. The basic form of this function is a standard fitting function supplied by the BMDP non-linear regression package (Dixon, et al., 1981).

Values for $K < 0.17$ and $K > 10^3$ may be read from the curves or *roughly* approximated as follows:

$$\beta_0 = \alpha_{max} = E = V_0 = 0.[illegible], \text{ for } K < 0.17, \qquad \text{II.12}$$

and

$$\beta_0 = E = V_0 = 0.99, \text{ for } K > 10^3,$$

$$\qquad \text{II.13}$$

$$\alpha_{max} = 1.56, \text{ for } K > 10^3.$$

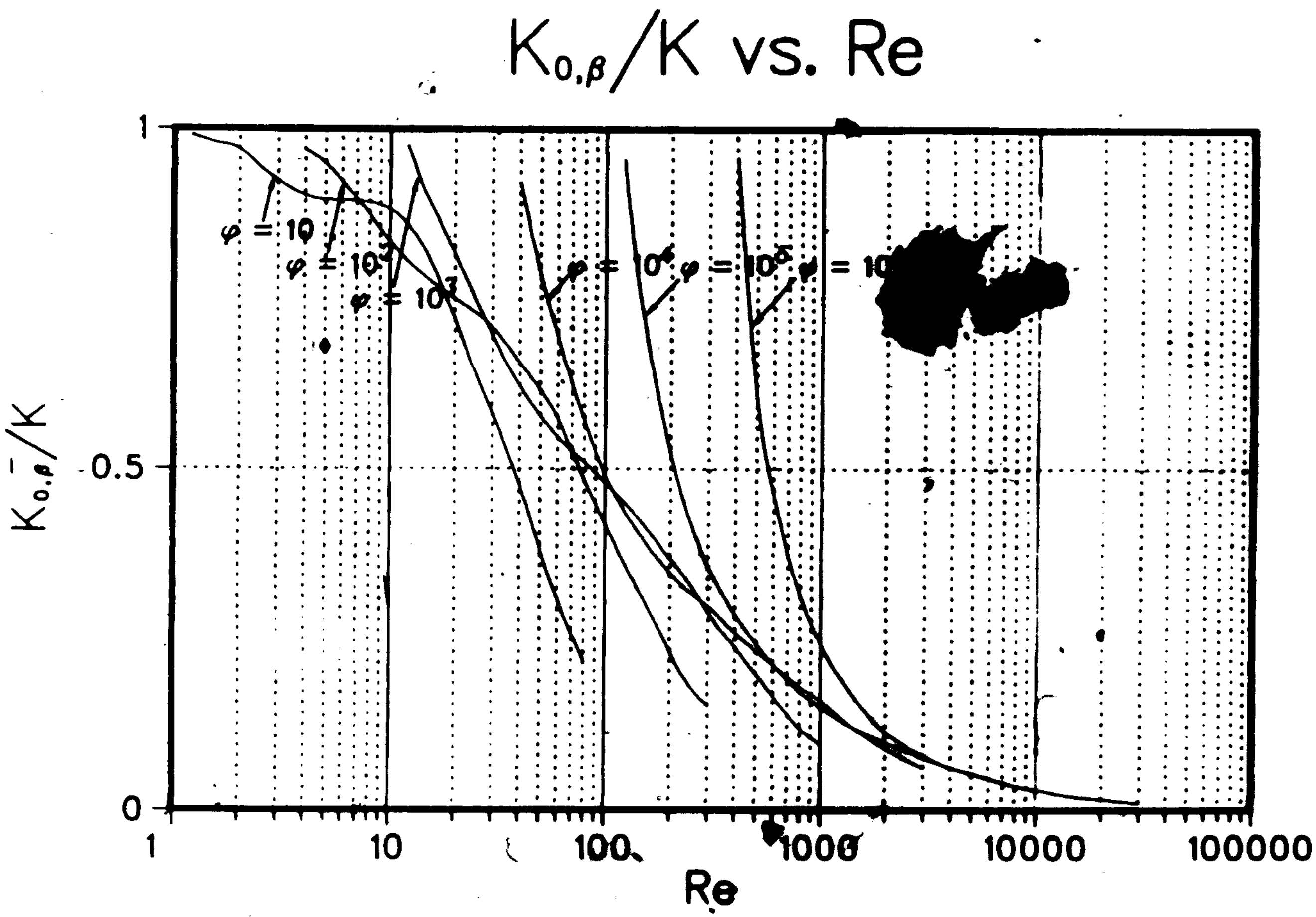


Figure Ik7 The dependence of $\frac{K_{0,\beta}}{K}$ on Reynolds Number Re. $K_{0,\beta}$ is defined by: $\beta_0(K,\varphi) = \beta_0(K_{0,\beta},0)$

	$X = \beta_o$	$X = a_{max}$	$X = E$	$X = V_o$
$c_{X,1}$	1.218	2.433	1.066	1.030
$c_{X,2}$	$-6.70*10^{-3}$	$-4.70*10^{-3}$	$-6.16*10^{-3}$	$1.68*10^{-3}$
$c_{X,3}$	-0.551	-0.375	-1.103	-0.796
$c_{X,4}$	-0.643	-0.576	-0.688	-0.780
$c_{X,5}$	-0.170	-0.781	-0.028	-0.040
$c_{X,6}$	$3.05*10^{-3}$	$8.50*10^{-3}$	$6.37*10^{-3}$	$9.44*10^{-3}$
$c_{X,7}$	0.430	0.383	0.381	0.344
$c_{X,8}$	2.220	1.757	3.641	2.657
$c_{X,9}$	-0.450	-0.298	-0.498	-0.519
$c_{X,10}$	-0.767	-0.420	-1.497	-1.060
$c_{X,11}$	-0.806	-0.960	-0.694	-0.842
$c_{X,12}$	-0.068	-0.179	-0.045	-0.029

Table II.6 Coefficients for approximating impingement parameters by the expression:

$$X(K,\varphi) = \left(c_{X,1}\, K^{c_{X,2}} \exp(c_{X,3}\, K^{c_{X,4}}) + c_{X,5}\right) - \left(c_{X,6}(\varphi - 100)^{c_{X,7}}\right) \left(c_{X,8}\, K^{c_{X,9}} \exp(c_{X,10}\, K^{c_{X,11}}) + c_{X,12}\right)$$

Within the range of $0.17 < K < 10^3$ the standard error of the fit with respect to the numerical data is given in Table II.7. It compares very favorably to the errors given in Table II.5 for the fitting schemes previously discussed.

E. Limitations of LB Theory

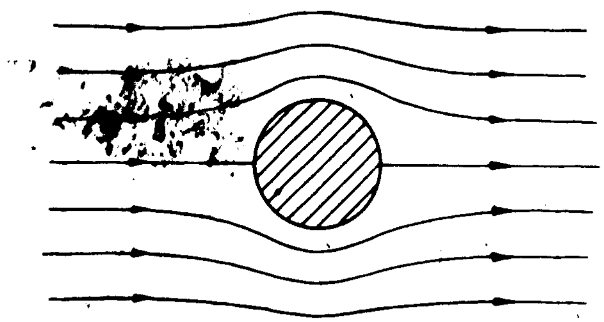
The collision efficiency theory of Langmuir and Blodgett assumes that the air is in potential flow about the collecting cylinder, that only steady-state drag forces are exerted on the water droplets by the flow, and that the flow itself remains unaffected by the presence of the droplets.

The potential flow field about an infinitely long, circular cylinder is as described by Equations II.7, and illustrated in Figure II.8(a). Figure II.8(b) illustrates the analogous situation for viscous flow at a cylinder Reynolds number of $\simeq 10^4$, corresponding to $D_c = 0.0254$ m, U = 10 m/sec, and T = -10° C. The upstream flow in Figure II.8(b), which determines the accretion, deviates slightly from the potential flow case due to the presence of a sizable wake on the downstream side. The cylinder plus wake will be felt by the flow further upstream, decreasing the streamline curvature, and thus increasing the collision efficiency. In most situations, however, the effect will be small.

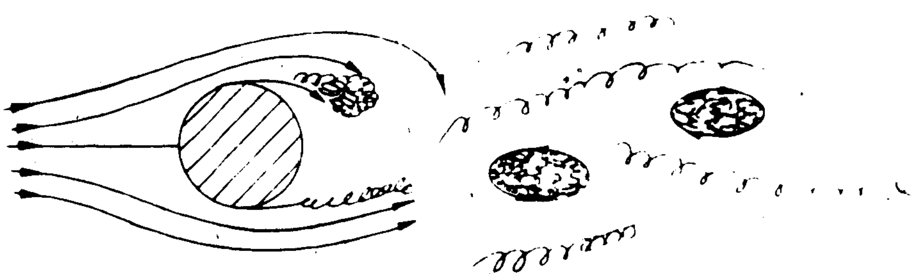
The potential flow assumption also ignores the cylinder boundary layer, which will be of negligible depth for a

		numerical				approximated			
φ	K	β_0	α_{max}	E	V_0	β_0	α_{max}	E	V_0
10^2	.16	0.034	0.054	0.001	0.002	0.036	0.054	0.0	0.0
	.64	0.415	0.720	0.212	0.297	0.416	0.720	0.210	0.293
	3.24	0.763	1.219	0.617	0.708	0.763	1.219	0.619	0.710
	16.	0.921	1.446	0.863	0.904	0.920	1.445	0.861	0.904
	100.	0.975	1.536	0.959	0.977	0.978	1.538	0.961	0.975
	625.	0.987	1.558	0.983	0.994	0.986	1.558	0.983	0.995
10^3	.196	0.073	0.131	0.007	0.012	0.075	0.135	0.005	0.013
	.625	0.359	0.612	0.157	0.227	0.363	0.611	0.159	0.220
	3.6	0.725	1.136	0.553	0.640	0.728	1.139	0.557	0.639
	19.6	0.906	1.413	0.832	0.876	0.904	1.401	0.825	0.869
	90.	0.963	1.512	0.935	0.957	0.964	1.503	0.931	0.951
	640.	0.984	1.553	0.976	0.989	0.983	1.549	0.974	0.989
10^4	.25	0.094	0.168	0.012	0.015	0.098	0.179	0.012	0.024
	1.	0.374	0.609	0.163	0.228	0.379	0.610	0.162	0.225
	6.25	0.726	1.115	0.547	0.625	0.726	1.115	0.548	0.637
	36.	0.899	1.379	0.812	0.854	0.899	1.380	0.816	0.865
	196.	0.960	1.497	0.927	0.948	0.962	1.497	0.930	0.954
	900.	0.980	1.538	0.966	0.980	0.980	1.543	0.967	0.986
					standard error:	0.002	0.005	0.003	0.006

Table II.7 - Comparisons between a selection of numerically calculated values, and approximated values calculated according to Table II.6.



(a)



(b)

Figure II.8 - (a):Potential flow streamlines about a cylinder in cross-section. (b):Viscous flow streamlines about a cylinder at a Reynolds number of 10,000.

smooth cylinder, but may become significant for rough surfaces. The effect, if any, of a deep, possibly turbulent boundary layer on the collision efficiency is unknown. In particular, the onset of accretion may be affected if the roughness elements are large enough to act as individual collectors.

The assumption of the airflow being undisturbed by the droplets carried in it, is a reasonable one given typical cloud liquid water contents.

In determining the deceleration of the drop by the flow, LB's steady-state viscous drag coefficient ignores buoyancy and gravitational forces, the influence of vorticity diffusion from the accelerating droplet surface, and the induced mass increase of the drop as a result of accelerating the air immediately surrounding it.

A 20 micron droplet falls in air with a terminal speed of 0.012 m/sec (Mason, 1971). Thus the vertical deviation of such a droplet approaching a 0.0254 m diameter cylinder from five diameters upstream, at a free-stream speed of 10 m/sec, is about 0.15 mm. This is only about one percent of the maximum vertical deviation due to the flow, under those conditions. At some point on the cylinder surface, the deviation due to the flow will be comparable to the gravitational drift, but since the relative difference in two adjacent trajectories will be much smaller, the collision efficiencies should not be affected.

Much larger drops, such as occur in sea spray icing, will have a correspondingly larger gravitational drift. Preliminary numerical calculations by Szilder (personal communication, 1986) indicate that there is still a very small effect on the collision efficiency, but this is a question which should be investigated further.

The other two effects on the drag are incorporated into the 'history' term, which, expressed as an acceleration, is:

$$-\frac{18\rho_a}{(2\rho_d+\rho_a)\,D_d}\sqrt{\frac{\mu}{\pi\rho_a}}\int_{-\infty}^{t}\frac{d\vec{v}}{d\tau}\frac{d\tau}{\sqrt{t-\tau}} \qquad \text{II.14}$$

The relative importance of the history term is often measured by the acceleration modulus (Norment, 1980):

$$N_A = D_d \left|\frac{dv}{dt}\right| / v^2, \qquad \text{II.15}$$

where v is the droplet's absolute velocity. If this quantity is < 0.01, the history term may be safely omitted from the droplet's equation of motion.

Numerical results from the trajectory integration model presented above show that N_A is well within this limit under most conditions of accretion on cylinders. However, for $K \leq 0.5$, N_A may reach values ≥ 0.01 just before impact, and for $K \leq 0.20$, the limit is exceeded as much as a few millimetres in front of the cylinder surface.

For these small K values then, the effect of the history term is to decrease the droplet's deceleration and increase both its total velocity near the surface, and the resulting collision efficiency. Numerical integrations by Oleskiw (1982) for $K = 0.196$, $\phi = 1000$ show an increase in E of 350%, and in β_0 of 167% when the history term is included.

Thus the 'cut-off' K value, for zero collision efficiency, which arises from LB theory, is not valid for real flow. In light of these results, it may be advisable to consider Equations II.11, II.12, and II.13 as valid only for $K > 0.25$. Below this limit, the trajectories should be recalculated using the complete drag terms in the droplet equation of motion.

The last assumption listed, that of an airflow undisturbed by the presence of water droplets, is probably a reasonable one for th small liquid water contents typical of clouds, $< 5\ g/m^3$. The case for heavy spray icing may, however, be very different, and this is another question which warrants further study.

F. Summary

Earlier results for water droplet trajectories at freezing temperatures have been re-examined and improved upon. The new results differ from LB by as much as 10 percent. Tabulated data are provided to replace those of Langmuir and Blodgett for the collisional parameters of water droplets on cylinders, and a good approximating function to the data has been found which will be of use to modellers. These results should only be applied, however, to real situations when $K > 0.25$, below which the non-steady-state drag terms have a significant effect on collision efficiencies.

References

Beard, K.V. 1976: Terminal Velocity and Shape of Cloud and Precipitation Drops Aloft. *J. Atmos. Sci.* , 33, 851-864. *J. Atmos. Sci.*, 33, 851-864.

Beard, K.V. and Pruppacher, H.R. 1969: A Determination of the Terminal Velocity and Drag of Small Water Droplets by Means of a Wind Tunnel. *J. Atmos. Sci.*, 26, 1066-1072.

Cansdale, J.T. and McNaughton, I.I. 1977: Calculation of Surface Temperature and Ice Accretion Rate in a Mixed Water Droplet / Ice Crystal Cloud. Royal Aircraft Establishment Technical Report 77090, 29 pp.

Dixon, W.J., Chief Editor, et. al. 1980: Chapter 14 - Nonlinear Regression. In: BMDP Statistical Software. University of California Press, 289-309.

Green, S. L. 1937: Hydro-and-Aero-Dynamics. Pitman and Sons, p. 26.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories.

Collected Works of Irving Langmuir Vol. 10, Pergamon Press, 348-393.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-Rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Mesinger, F. and Arakawa, A. 1976: Numerical Methods used in Atmospheric Models, Vol. 1. W.M.O. GARP Publications Series No. 17, 64 pp.

Mason, B.J. 1971: The Physics of Clouds, 2nd Edition. Clarendon Press, p. 594.

Makkonen, L. 1984: Modelling of Ice Accretion on Wires. *J. Climate and App. Meteor.*, 23, 929-939.

Norment, H.G. 1980: Calculation of Water Drop Trajectories to and About Three-Dimensional Bodies in Potential Airflow. NASA Contractor Report 3291, 82 pp.

Oleskiw, M.M. 1982: A Computer Simulation of Time-Dependent Rime Icing on Airfoils. Ph.D. thesis, The University of Alberta, 302 pp.

Stallabrass, J.R. 1980: Trawler Icing: A Compilation of Work

Done at NRC. NRC Report MD-56, NRC No. 19372, 103 pp.

III. An Operational Model for Rime Ice Accretion

A. Introduction

Modelling of the many physical processes involved in ice accretion can be a complex and time-consuming exercise. The cost of running such models tends to limit their use as investigative tools, or for predictive purposes of an operational character. In an attempt to alleviate part of this problem, simple parameterization methods have been developed which make possible a more practical, inexpensive and realistic model. In this chapter these methods, and their application in a computer simulation of rime icing on circular cylinders will be described.

The computer model was developed with several specific purposes in mind. The first was to reduce the computational time necessary, in particular, for the calculation of impingement parameters. The quantities E, β_0 and α_{max} for circular cylinders are available from the tabulated data and analytical fitting functions of Chapter II. Values of local collision efficiency between the stagnation line and maximum impingement angle usually have to be determined from a series of droplet trajectory pairs.[1] Oleskiw (1982) has calculated the distributions of local collision efficiency

---

[1] It has been suggested (McComber and Touzot, 1981) that the variation of the local radial velocity of impact be used as an approximation to the local collision efficiency. It has been previously shown (Lozowski, Finstad and Gates, 1985) that for realistic cases this can be a poor approximation at the stagnation line (compare Figures II.3 and II.6), and that the relative error increases dramatically with distance toward the accretion limit.

for several cases, from numerically integrated trajectories. From these, it was possible to derive a parameterized general form of the $\beta(\alpha)$ curve which depends only on the parameters β_0 and α_{max}.

Use of this parameterization is the major time-saving device in the model. Several other options are provided for simulating the effect on the $\beta(\alpha)$ distribution of a droplet size spectrum which cannot be adequately modelled by the median volume size. This effect is principally in the extension of the maximum accretion angle.

A second major purpose in developing this model was to incorporate the local ice density variations into the shape of the rime ice accretions. An empirical, parameterized function is used to describe this variation, and it is shown to be reasonably successful in reproducing observed accretion shapes on non-rotating cylinders under a variety of conditions.

A third purpose was to attempt to simulate a slow rotation of the collecting cylinder about its long axis. This is meant to imitate the effects of a gravitational torque on an ice-loaded cable. Although this capability has now been built into the model, it has not yet been been possible to test it against laboratory or field data.

In order to effectively examine the validity of these methods, only dry, rime ice accretions are considered. This is so that any comparisons with experimental results are not dependent on poorly known quantities such as heat transfer, which affect wet icing processes.

In the experimental accretion experiments used to aid in model development and verification, a crucial measurement is the droplet size spectrum, and its median volume diameter. All droplet spectra were measured by the oil slide method, an outline of which is given in Mason (1971). However, the mvd's derived from such measurements have been found to show a systematic error by Makkonen and Stallabrass (1984). We therefore apply their empirical correction formula to our data in all but a few cases, which fall outside the range of validity for the correction. The procedure followed in these cases is described in Appendix III.

To ensure rime icing under the conditions typical of the wind-tunnel experiments in which collectors with diameters of 2 to 5 cm were exposed to typical atmospheric liquid water contents and wind speeds, the air temperature in the model is assumed to be always at or below -10° C. For application to general conditions, the model should include a calculation of the critical liquid water content, or 'Ludlam limit', (Ludlam, 1951) in order to determine whether the ice will be wet or dry.

B. Parameterization for Local Collision Efficiency

For the idealized situation of straight-line droplet trajectories the local collision efficiency is given simply by cos α, where α is the angle of the local surface normal with respect to the free stream direction. The collision efficiency has a maximum value of unity at the stagnation line, and a minimum of zero at $\alpha = \pi/2$.

In a more realistic situation, the trajectories curve away from the free stream direction as they approach the cylinder. The stagnation line collision efficiency is reduced from 1, and the local value reaches zero for some $\alpha = \alpha_{max} < \pi/2$. This situation can be well approximated (especially for large droplets whose trajectory curvature is small) by directly scaling the cos α distribution to these two extreme values, as follows:

$$\beta(\alpha) = \beta_0 \cos((\alpha/\alpha_{max})(\pi/2)), \qquad \text{III.1}$$

where the two parameters β_0 and α_{max} are determined from tabulated values or approximating functions given in Chapter II.

A form of time dependence is built into this method since the collision efficiency at a point depends on its local surface angle, which changes with time as the ice layers are accreted on top of one another. The parameters β_0 and α_{max} do not change with time, an approximation which becomes worse as the accretion shape grows, and departs from a circular cross-section.

For smaller droplets the simple approximation of Equation III.1 fails, since the trajectories are curved by much larger amounts. It has been found that, for a β_0 <

0.65, the $\beta(\alpha)$ curve is more closely approximated by:

$$\beta(\alpha) = \beta_0 - (\beta_0 (1 - \cos^{0.5}\alpha) \text{[illegible]} \cos^{0.5}\alpha_{max})). \quad \text{III.2}$$

For a 0.0254 m diameter cylinder at typical ground level wind speeds of a few tens of metres per second, this corresponds to water droplets of less than 20 microns diameter.

Figure III.1 illustrates the goodness of fit of these functions against local collision efficiency curves calculated numerically by Oleskiw (1982), and approximated by Lozowski, Stallabrass and Hearty (1983).

In both Equations III.1 and III.2, a monodisperse droplet size population has been assumed, or at least a distribution for which the collision efficiency may be reasonably approximated by the collision efficiency of the median volume droplet, or mvd. The results of Appendix VI suggest that, for typical ground-level wind speeds, the single droplet size approximation holds well for $\beta_0 \geq 0.5$.

Otherwise, the effect of the size spectrum, and particularly its larger drops, should be accounted for. Droplet size distributions for icing purposes are commonly described by volume percentages in several size bins of finite width (see, for example, Figure III.2). Applying either Equation III.1 or III.2 to the median volume droplet of *each size bin*, a volume-weighted average distribution may be calculated. Such a weighted curve then illustrates the

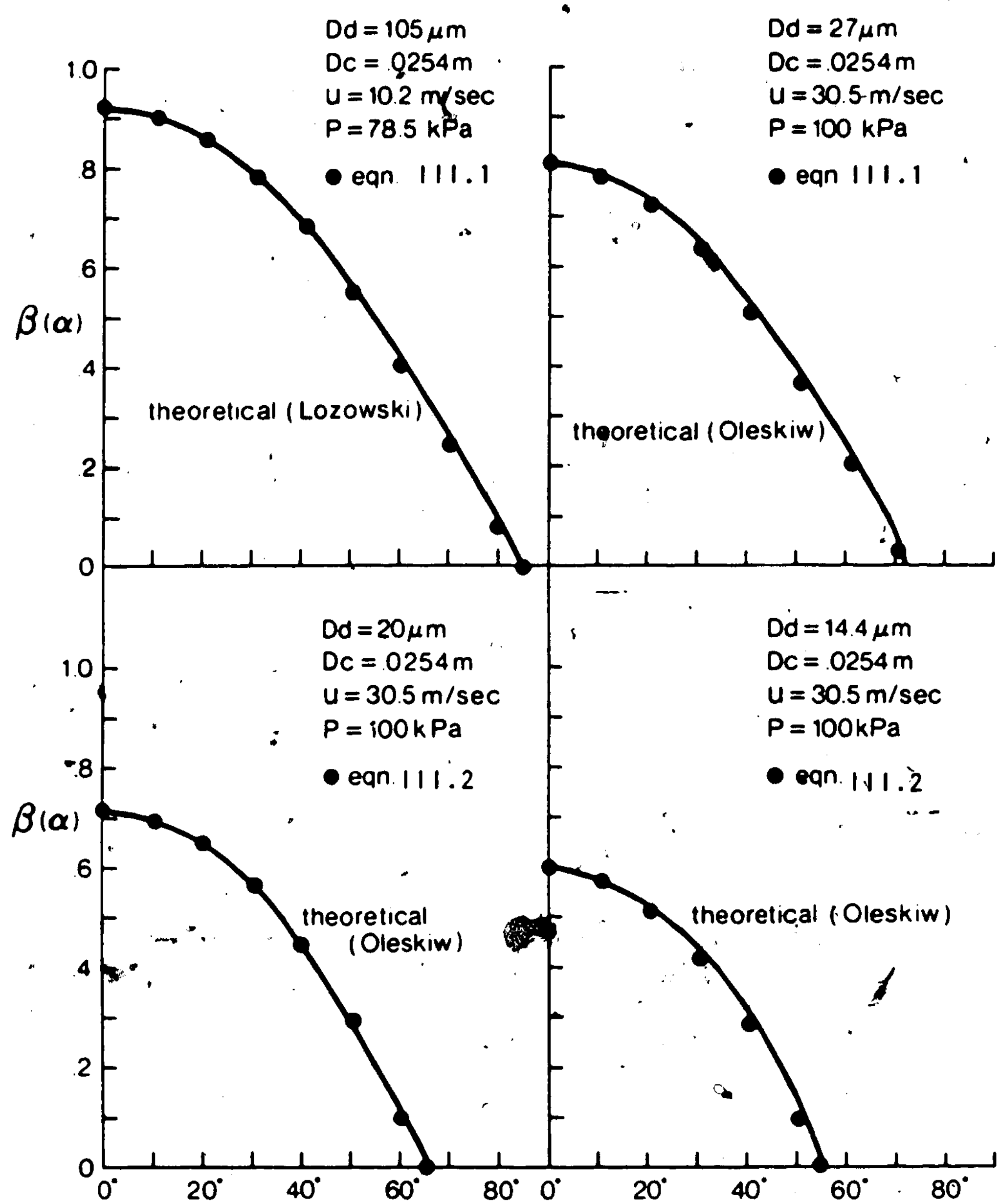


Figure III.1 - Parameterized $\beta(\alpha)$ values from Equations III.1 and III.2, compared to numerically calculated curves of Oleskiw (1982), and parameterized curve after Lozowski, Stallabrass, and Hearty (1983).

effect of the smaller droplets in extending the maximum accretion limit.

This weighting procedure, for a given input size spectrum, may be carried out within the model. This is the "weighted spectrum" option. However, the nature of the averaging procedure eventually gives rise to first-order discontinuities in the $\beta(\alpha)$ curve, and thus on the accretion surface. For this reason, the weighted spectrum option is not recommended when more than three ice layers are accreted, no matter how thin these layers are.

An alternative option is included to avoid this problem, which has been called the "simulated spectrum" option. It is another form of the $\beta(\alpha)$ curve, this time based on weighted average curves (from laboratory measured spectra) which have been scaled and fitted with polynomial functions.

for $\beta_0 < 0.65$:

$$\beta(\alpha/\alpha_m) = \beta_0 \left(1 + 0.22(\alpha/\alpha_m) - 3.08(\alpha/\alpha_m)^2 + 1.86(\alpha/\alpha_m)^3\right), \qquad \text{III.3}$$

and for $\beta_0 > 0.65$:

$$\beta(\alpha/\alpha_m) = \beta_0 \left(1 + 0.029(\alpha/\alpha_m) - 1.94(\alpha/\alpha_m)^2 + 2.48(\alpha/\alpha_m)^3 - 4.11(\alpha/\alpha_m)^4 + 2.54(\alpha/\alpha_m)^5\right), \qquad \text{III.4}$$

where $\alpha_m = \alpha_{max} + Tl$, and Tl is the "tail length", an extension to α_{max}, derived from numerical results and correlated to the mvd. For $\beta_0 < 0.65$:

$$Tl = 0.52 \exp(-mvd / 70) \qquad III.5$$

and for $\beta_0 > 0.65$:

$$Tl = 0.37 \exp(-mvd / 70). \qquad III.6$$

Here mvd is in μm and Tl is in radians. Equations III.3 and III.4 are compared in Figure III.2 with two weighted average curves for the cloud droplet spectra shown.

Although these methods may seem to have become very superficial, they do retain the desired economy of computation with an acceptably small loss of accuracy.

C. Variation of Local Density

A study of the local density variations in circular cylinder rime accretions will be discussed in some detail in Chapter IV. Here the result is presented briefly; an empirical curve describing local density as a function of surface slope.

Based on cross-sectional thickness measurements of several shallow rime layers grown in the laboratory under known conditions, the following function was derived for $\alpha/\alpha_m < 1$:

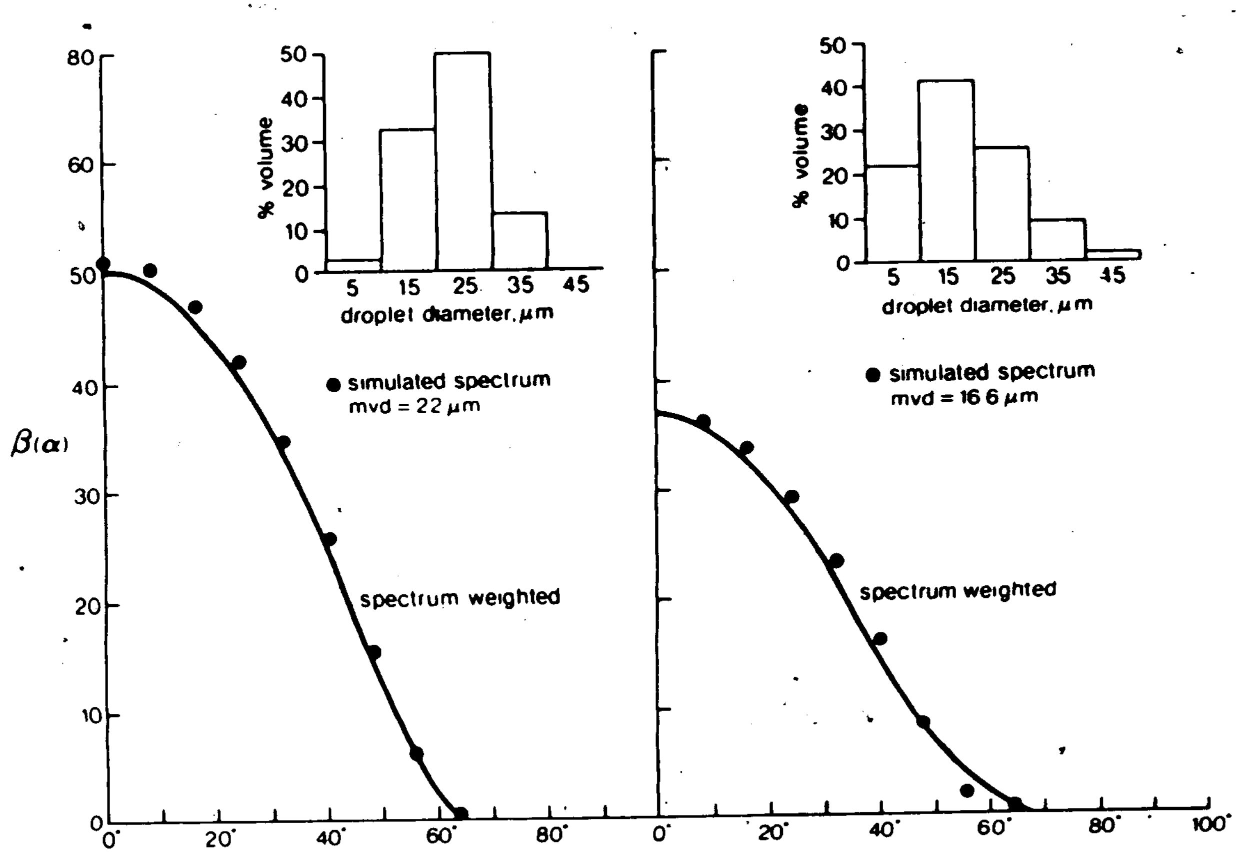


Figure III.[illegible] - Parameterized $\beta(\alpha)$ values from Equations III.3 and III.4, compared to weighted average curves for the cloud droplet spectra shown. These spectra are derived from the distribution of Best (1951).

$$\rho/\rho_0 = (1 - 0.143(\alpha/\alpha_m) - 0.246(\alpha/\alpha_m)^2 - 0.309(\alpha/\alpha_m)^3). \qquad \text{III.7}$$

Here α_m is defined as for Equations III.3 and III.4, and in this version of the model, $\rho_0 = 890$ kg / m^3. This assumption is made for simplicity; in reality the rime densities may be much lower, and this will affect the accretion thickness and overall shape. Chapter IV will discuss methods for estimating more realistic values of ρ_0.

The remainder of this section will examine the predictive success of the model under a variety of input conditions. Figure III.3 compares accretion profiles and masses from the model against ice accretions grown in wind tunnel experiments carried out in the University of Alberta's FROST tunnel (this facility and our experimental methods are described in Gates, 1981 and Lozowski and Gates, 1984). For the ranges shown of mvd, liquid water content and free-stream speed the modelled mass is always within 10% of the observed mass, and the major features of the experimental shapes are reproduced in the profiles.

However, it is not expected that the model will reproduce such features as ridges or troughs near the [illegible] stagnation line. These are most likely due to chang[illegible] the flow pattern as the accretion grows, and the res[illegible]g.

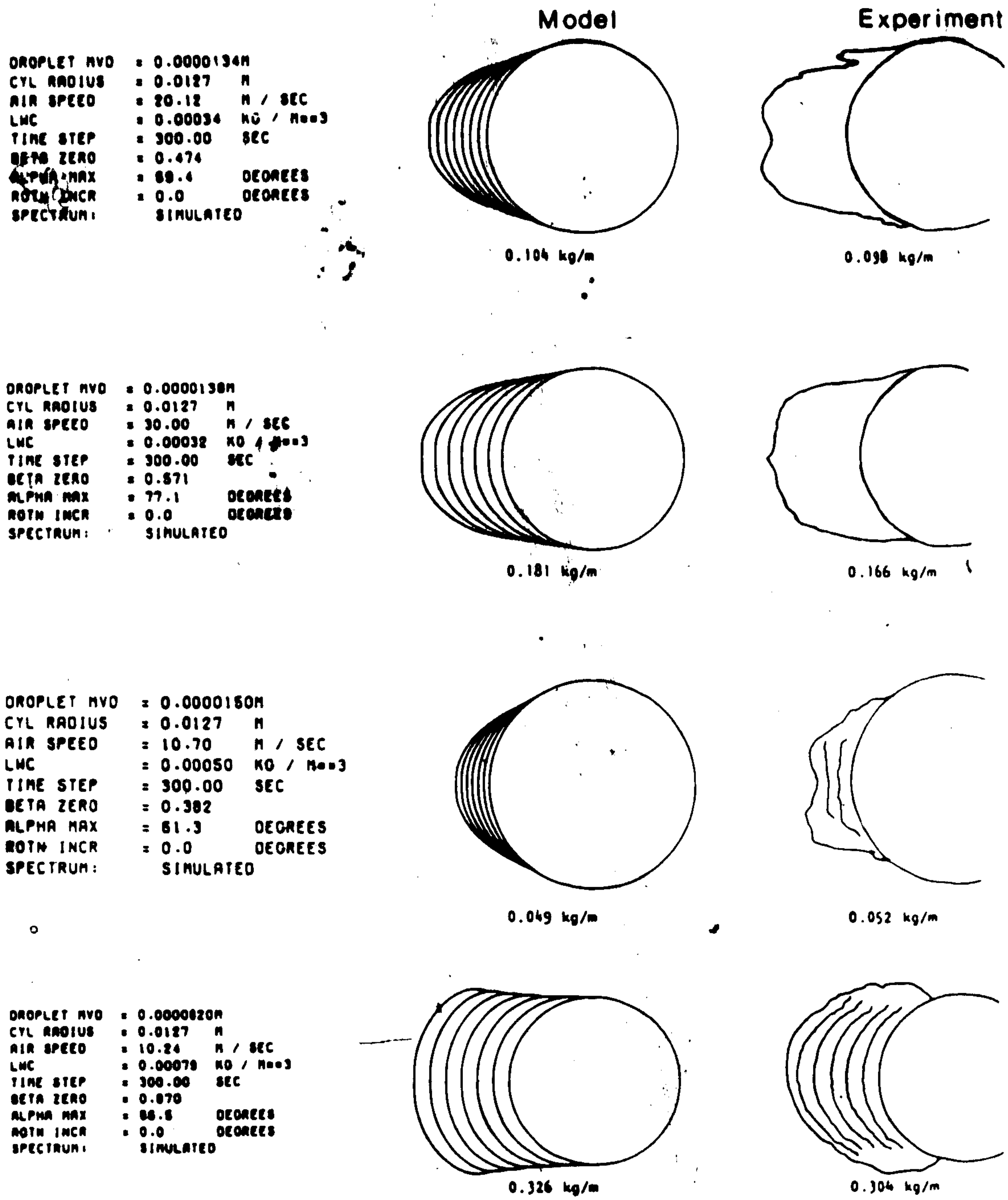


Figure III.3 - Comparisons between modelled and experimental profiles and masses per unit length for the conditions listed at left. The lower two experimental profiles have layer-outlines which were marked by spray paint.

changes in the impingement characteristics.

In particular, the value of β_0 will change with time, while the model assumes a constant value. Even with this assumption, however, it is able to distinguish between widening and narrowing profiles for small differences in growth conditions, and can produce the flat front surface which often precedes the formation of ridges and troughs in the experiments.

D. Rotation

An option has been included in the model which will allow the cylinder plus accretion to be rotated with respect to the free stream by a small amount after each layer is accreted. No attempt is made to calculate the correct rotation rate due to the gravitational and aerodynamic torque; the rotation speed in degrees per layer must be specified as one of the input parameters.

In order to achieve this effect within the model algorithm, the local collision efficiency and local density must be calculated as functions of the position angle rather than the surface angle on the current surface. (The position angle at a surface point is the angle between a line joining the point to the centre of the cylinder, and the free-stream direction.) This somewhat less accurate method was adopted in order to avoid problems when accreting over top of the previous layers' accretion limit.

There are presently no laboratory or field data available with which to test this part of the model. It is presented here only as an option for future study. An example of a rotating case is shown in Figure III.4.

The Fortran77 code written for the model is included in Appendix II. The code will also run on an IBM or compatible personal computer fitted with a math co-processor and graphics capability; a six layer rotated accretion takes about 90 seconds to compute on a regular PC.

E. Model Applications

This model is intended for use in the laboratory, in conjunction with wind-tunnel experiments. There the required input data are readily available, as accurately as present equipment and methods can determine them.

However, if this model or some future version of it were to be applied in the field, the situation becomes very different. Good estimates of the input data (duration, wind speed, liquid water content, droplet mvd and type of spectrum) must somehow be obtained. In addition, a field version should be able to predict at least the transition to wet icing, and so the air temperature will also be required as input.

These data are rarely available for a remote site. Local meteorological services may provide estimates or forecasts for duration, wind speed, air temperature, and precipitation, but usually not liquid water content or

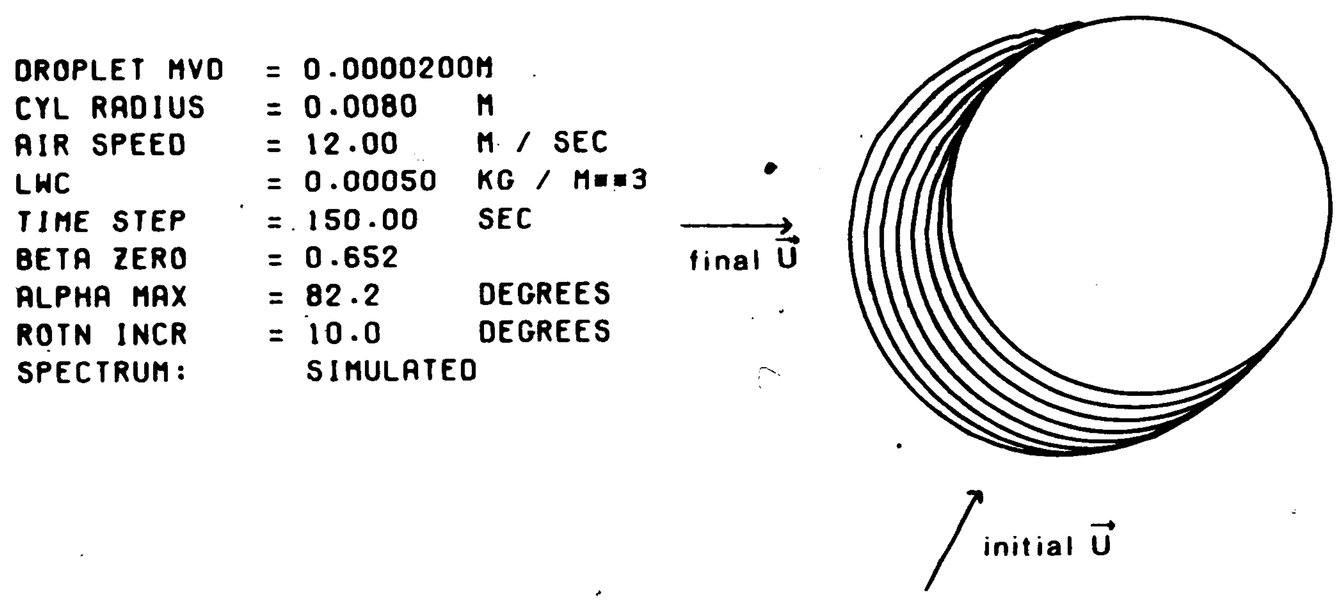


Figure III.4 - An example of the model output for a rotating cylinder.

cloud droplet size.

Clearly, real-time icing forecasting with such a model is a long way off. However, the model could conceivably be used to determine icing climatologies for different regions where adequate meteorological data have been obtained. This would mean sufficiently long-term observations of liquid water contents and droplet sizes (i.e. cloud type) to give an idea of the typical values of these quantities under the typical meteorological situations which lead to icing events in a given region.

However, given the current state of knowledge of field conditions, the best application for this model remains in furthering the understanding of what is happening under controlled rime icing conditions in the laboratory, which it is capable of doing much more efficiently and cheaply than previous models.

References

Gates, E.M. 1981: FROST Tunnel. Departmental Report No. 26, Department of Mechanical Engineering, The University of Alberta. 10 pp.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. In: Collected Works of Irving Langmuir, Vol. 10. Pergamon Press, 348-393.

Lozowski, E.P., Finstad, K.J. and Gates, E.M. 1985: Comments on "Calculation of the Impingement of Cloud Droplets by the Finite-Element Method". *J. Atmos. Sci.*, 42, 306-307.

Lozowski, E.P. and Gates, E.M. 1984: Ice Accretion on Structures in a Marine Environment. Proceedings, Third International Symposium on Offshore Mechanics and Arctic Engineering, New Orleans, 8 pp.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Non-Rotating Cylinder Part I: A Simulation Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Ludlam, F.H. 1951: The Heat Economy of a Rimed Cylinder. *Quart. Jour. Roy. Met. Soc.*, 77, 663-667.

Mason, B.J. 1971: The Physics of Clouds, 2nd Edition. Clarendon Press, p. 93.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.*, 38, 1027-1036.

Oleskiw, M. 1982: A Computer Simulation of Icing on Airfoils. Ph.D. Thesis, The University of Alberta, 302 pp.

IV. Local Densities of Circular Cylinder Rime Accretions

A. Introduction

A situation very commonly described for the study of rime ice accretion is that of a non-rotating, circular cylinder, enveloped in a steady cross-flow of cold, cloudy air. Rime ice is formed when all of the impinging cloud droplets are able to freeze completely at the point of impact, and before the arrival of another droplet. Rime ice is always 'dry', which means there is no liquid water on the surface or trapped within the ice. The conditions necessary for dry ice to form are described by the so-called "Ludlam limit" (Ludlam, 1951), a maximum value of cloud liquid water content computed as a function of air temperature, cylinder size and air speed. Above this limiting value, the accretion will be 'wet', or glaze ice, where the droplets may run along the surface before freezing, or are incorporated into pockets of liquid water within the ice. This results in a much higher and more uniform density of ice. This study deals only with rime ice.

Examination of a rime accretion in cross-section (Figure IV.1) reveals details of the internal structure which govern the local density. Air bubbles and channels are apparent, usually increasing in number and size toward the lateral edges, away from the stagnation line. At the extreme edges of the accretion are often found the individual elongated or fan-shaped structures known as

Figure IV.1 - A 1.5 cm thick cross-section of a circular cylinder rime accretion, viewed in transmitted light. The accretion conditions are those of Sample 1 in Table IV.1.

'rime feathers', which seem to merge into the accretion from the sides. These are seen more easily in a lateral view (Figure IV.2).

It is apparent even from such a cursory examination that there should be a trend in the local ice density, decreasing from the stagnation line out toward the edges. When averaged over intervals larger than the typical air channel size, a smooth distribution should emerge. The actual shape of this distribution and how it arises are of interest to modellers of the icing process.

Numerical models have been devised which predict the results of accretion experiments performed in wind tunnels under controlled conditions, e.g. Oleskiw (1982), Lozowski, Stallabrass and Hearty (1983), McComber and Touzot (1981). The parameters which determine the rate of ice growth in such models are the accretion object's size, the air temperature and pressure, air speed, liquid water content and the sizes and relative number distribution of the water droplets in the air stream. These quantities are needed to calculate the local and overall collision efficiencies on the collecting object, which together with the local ice density determines the mass and the shape of the modelled ice deposit.

Although collision efficiency and its local distribution have been well studied, at least for circular cylinders (see for example, Oleskiw, 1982, McComber and Touzot, 1981, Langmuir and Blodgett, 1946), the density of

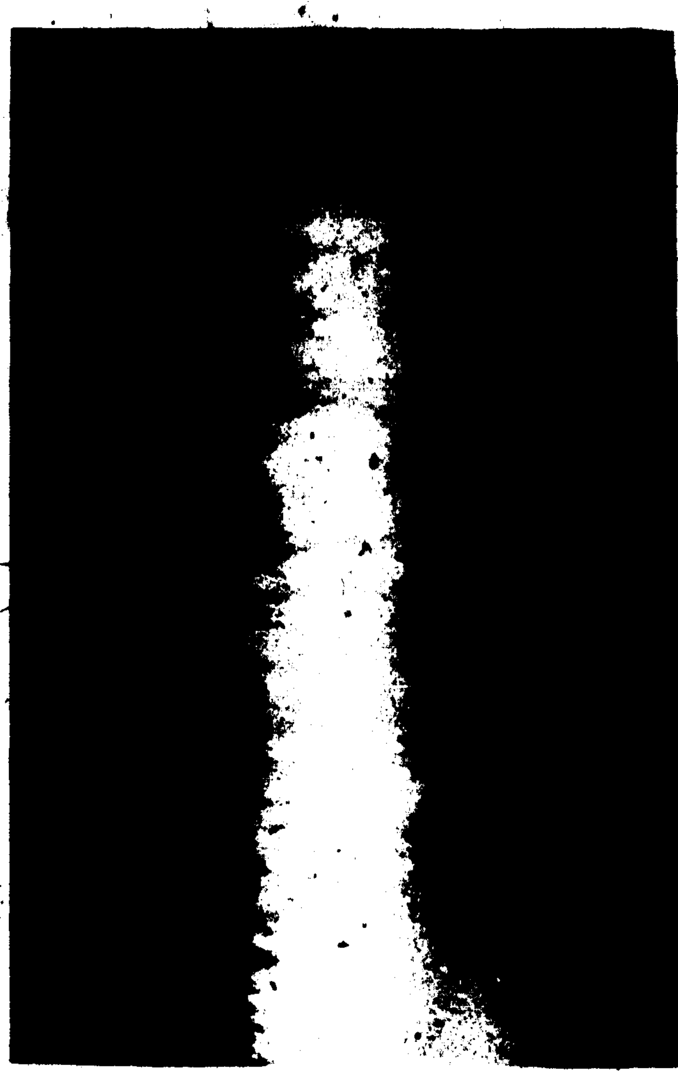


Figure IV.2 - A typical circular cylinder accretion showing rime feathers.

rime has not received as much attention. Some studies have dealt with the densities of ice accretions on rotating cylinders or hailstones (e.g. Macklin, 1962, List, Cantin and Ferland, 1970, Makkonen and Stallabrass, 1984).

Bain and Gayet (1983) have attempted to adapt these results to the fixed cylinder case; their method will be covered in the next section. Problems with this method, very similar to those outlined by Rasmussen and Heymsfield (1985), are also discussed, and a new density distribution is derived directly from experimental accretion shapes. Both distributions are tested in the model, and also against direct laboratory measurements of local density.

B. Rotating Cylinder Densities Applied to the Non-rotating Case

Macklin (1962) made extensive measurements of the bulk densities of rotating cylinder ice deposits. The results were found to correlate best with the parameter, called R, given by:

$$R = r_d V_0 / T_s \qquad \text{IV.1}$$

where r_d is the radius of the median volume droplet, V_0 is the impact speed of that droplet at the stagnation line, and T_s is the mean surface temperature of the accretion. Similar correlations were found for the experimental results of Bain and Gayet (1983) and Makkonen and Stallabrass (1984) on rotating cylinders, although the latter found significantly larger densities in the lower range of R.

Makkonen and Stallabrass attribute this difference to their method of measuring droplet sizes with optical probes. Macklin's droplets were measured using the oil slide method, which according to Makkonen and Stalla[illegible]ves underestimated sizes, in compari[illegible] optical probe method. Makkonen and Stallabrass' [illegible] the correlation for the [illegible] 0.2 < R < 1[illegible]:

$$\rho(R) = 0.378 + 0.4[illegible](\log_{10}(R)) - 0.0823(\log_{10}(R))^2. \qquad \text{IV.2}$$

Bain and Gayet (1983) made an attempt to apply the earlier correlation of Macklin to densities on a fixed cylinder accretion, by introdu[illegible] local variations into the R parameter as follows:

$$R(\theta) = r\, V_r(\theta) \,/\, T_s(\theta), \qquad \text{IV.3}$$

where now $V_r(\theta)$ is the radial component of droplet impact velocity as a function of the radius angle θ (measured with respect to the stagnation line). Bain and Gayet derive the total droplet speed at impact from an assumed cosine distribution of speed with cylinder radius angle, scaled to the stagnation line value as calculated by Langmuir and Blodgett. $T_s(\theta)$ is the variation of local surface temperature calculated from the model of Lozowski, Stallabrass and Hearty (1983).

An initial objection to this approach is philosophical. Macklin's correlation is a purely empirical one; it is not a physical model and therefore should not be extended to a physical situation so different from that which provided the original data.

On the fixed cylinder, $V_r(\theta)$ is not even analogous to V_0 on the rotating cylinder, which is the maximum impact speed felt at any one point during one rotation. Furthermore, on the rotating cylinder, the droplets arrive at a given point from a large range of angles, which should perhaps result in closer packing of the droplets and a higher density. More importantly, the shadowing effects of feather structures and the formation of air channels between them is not accounted for.

However, a more concrete objection is found by comparing this type of distribution to that found by direct measurement of the density, and by comparing the model accretion shapes and masses it produces to experimental profiles. The first comparison will be discussed in Section D of this Chapter. For the model comparison the simple rime accretion model described in Chapter III is used.

The experimental accretion profile shown on the left in Figure IV.3 was grown, under the conditions listed, in the University of Alberta FROST tunnel. Also shown are model profiles produced by three different distributions of local

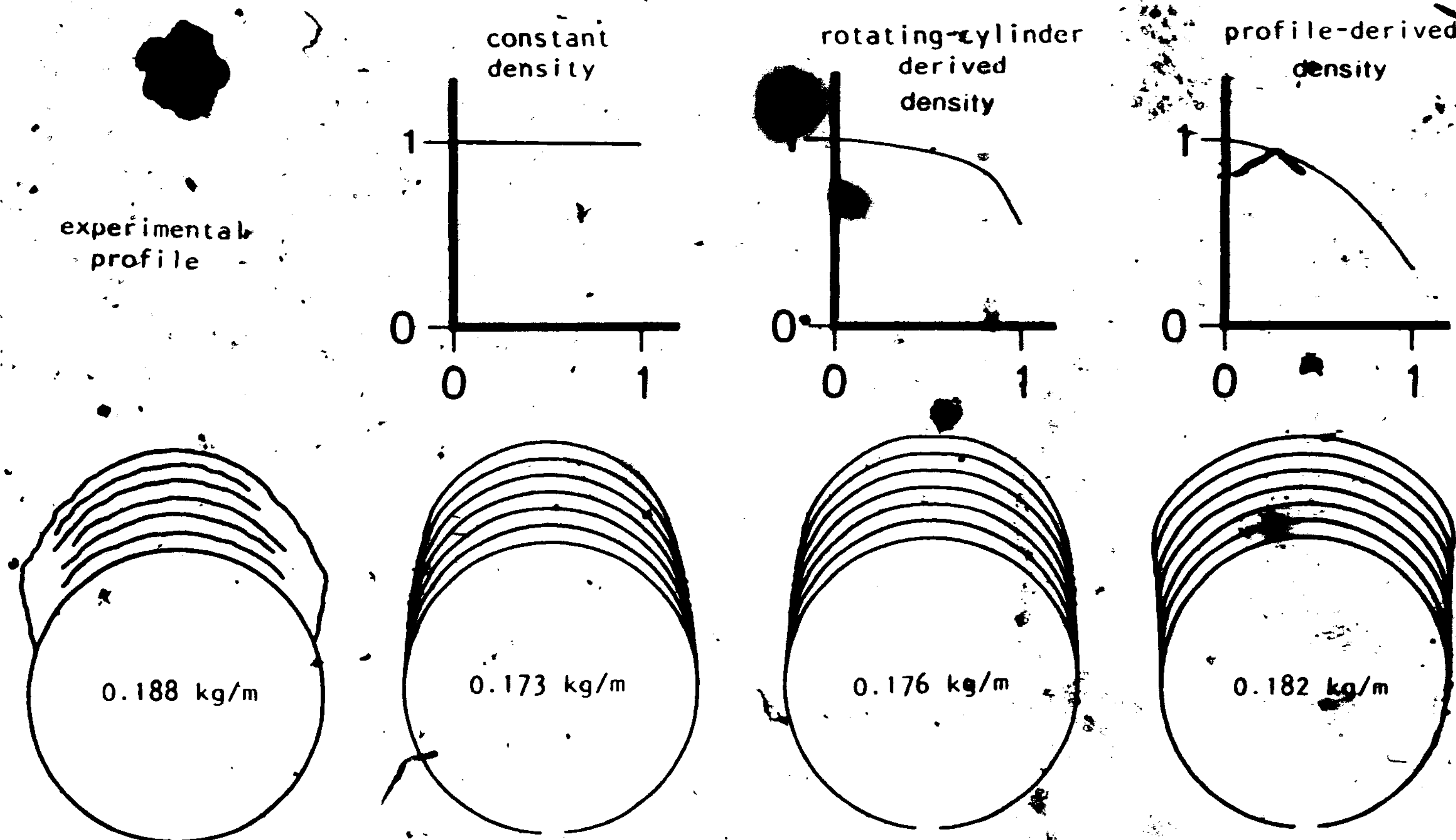


Figure IV.3 - Experimental and three model accretions showing the effects of three different density distributions (plotted above as ρ/ρ_0 vs. α/α_{max}). Conditions are: D_d = 51 μm, D_c = 0.0254 m, U = 10.7 m/sec, lwc = 0.49 g/m^3, τ = 300 sec.

density. The profile at centre left results from a constant density, that at centre right from a distribution that is derived following Bain and Gayet. The profile at the far right uses a third distribution which is described below.

The three density distributions are illustrated above their respective profiles. For easy comparison, each is plotted as ρ/ρ_0 vs. α/α_m, where ρ_0 is the local density at the stagnation line, α is the local angle of the surface normal with respect to the free stream direction, and α_m is defined in Chapter III. On the bare cylinder, α is identical with θ, but not for subsequent ice layers. The angle α is used rather than θ, so that the density distribution will be time-dependent, changing as the ice shape changes.

The middle distribution is derived from Equation IV.1, using $V_r(\alpha)$ as calculated by Bain and Gayet, and $T_s(\alpha)$ calculated by the model of Lozowski, Stallabrass and Hearty. The resulting curve is fitted with a third-order polynomial:

$$\rho(\alpha/\alpha_m) / \rho_0 = (1 - 0.303 (\alpha/\alpha_m) + 0.936 (\alpha/\alpha_m)^2 - 1.074 (\alpha/\alpha_m)^3). \quad \text{IV.4}$$

The second modelled profile is very similar to the first, constant density profile, which was too narrow in comparison with the experimental shape. Both of the modelled masses are less than the observed mass, by 8 and 6 percent, respectively. The result using the third distribution shown is in better agreement with the experimental shape, by having a widening profile, and a mass within 3 percent of observation. This distribution and its derivation are described next.

C. Profile-Derived Density Distribution

The thickness (Th) of ice deposited in time τ at a given point on the bare cylinder surface, which we describe by its surface angle α, is

$$Th(\alpha) = \beta(\alpha)\ U\ lwc\ \tau\ /\ \rho(\alpha), \qquad IV.5$$

where U is the free-stream speed, and lwc is the liquid water content. This version of the dry-growth equation ignores curvature effects, e.g. as described by Lozowski, Stallabrass and Hearty (1983). This is a reasonable simplification given the very small spacing of the points which define the icing surface in the model.

Given IV.5 then, if both the local thickness and the local collision efficiency $\beta(\alpha)$ are known, and if the interval τ is assumed to be small enough that $\beta(\alpha)$ and the density $\rho(\alpha)$ remain essentially constant, IV.5 may be solved for $\rho(\alpha)$.

Figure IV.4 shows another experimental accretion grown in the FROST tunnel, and its growth conditions. The total accretion time was 30 minutes. Every five minutes, a layer

Figure IV.4 - A circular cylinder rime deposit in profile, with paint layers marking the surface at five minute intervals. Conditions are those of Case B in Table III.1.

of paint was sprayed onto the surface, allowing changes in the shape of the ice surface to be followed with time. It also allowed local thickness measurements of thin layers to be obtained, which are unaffected by the melting or sublimation losses that must occur on the top surface, between the time of the end of the test and photography of the profile.

The distribution of local collision efficiency was calculated from the median volume diameter, and an assumed droplet size distribution, scaled to the mvd, based on measurements taken in the same wind tunnel. The mvd was originally measured by the oil slide method, which is known from the work of Makkonen and Stallabrass (1984) to be an overestimate; but for Case B of Table IV.1 the oil slide estimate falls outside of the size range for which Makkonen and Stallabrass' empirical correction formula is valid. Therefore, an mvd has been adopted which was derived from the stagnation line thickness measurement, the liquid water content measurement, and an assumed local density for the stagnation line of 890 kg / m^3 (a reasonable density in this case based on visual inspection of the accretion). The method used to derive this estimate of the mvd is described in Appendix III.

For Case A of the same table it was possible to make use of the correction formula mentioned above to correct the oil slide measured mvd.

In this way the distribution $\rho(\alpha)$ has been found for several layers of two different accretions. The data are listed in Table IV.1 and plotted in Figure IV.5. A cubic polynomial fit (solid curve, Figure IV.5) gives:

$$\rho(\alpha/\alpha_m) / \rho_0 = (1 - 0.143\ (\alpha/\alpha_m) - 0.246\ (\alpha/\alpha_m)^2 - 0.309\ (\alpha/\alpha_m)^3) \qquad \text{IV.6}$$

The relative distribution is thus assumed independent of lwc and air temperature; this assumption implies that these factors only come into the determination of the scaling parameter ρ_0. Normalization of the distribution by the two parameters ρ_0 and α_m makes the expression above easy to use in an icing model, assuming that it is generally applicable to a range of accretion conditions. Evidence for this was presented in Figure III.3 of the last Chapter, where the model results are compared to several independent wind tunnel experiments.

D. Laboratory Measurements of Local Density

For further evidence to support our proposed distribution of local density, a series of direct measurements were made from wind-tunnel accretions. Following the techniques described by List, Cantin and Ferland (1970), local densities were measured by drilling small holes (2.5 - 3.0mm diameter, less than 1 cm depth) in

	α (degrees)	Th (mm)	β(α)	ρ (kg/m³)
Case A	0	3.34	0.407	767
	10	3.12	0.393	792
	20	3.25	0.334	646
	30	2.9	0.249	540
	40	1.7	0.155	574
	50	0.9	0.069	482
Case B	0	1.48	0.842	890
	10	1.48	0.825	871
	20	1.48	0.776	820
	30	1.40	0.699	780
	40	1.33	0.592	696
	50	1.03	0.458	695
	60	0.81	0.302	583

Table IV.1 - Local densities as a function of surface slope derived from thickness measurements of shallow ice layers accreted at -10°C under the following conditions:

	D_d (µm)	D_c (m)	lwc (g/m³)	U (m/sec)	τ (sec)
Case A	15.5	0.0254	0.98	10.7	600
Case B	51.0	0.0254	0.49	10.7	300

β(α) is calculated from Equations II.11, III.3 and III.4.

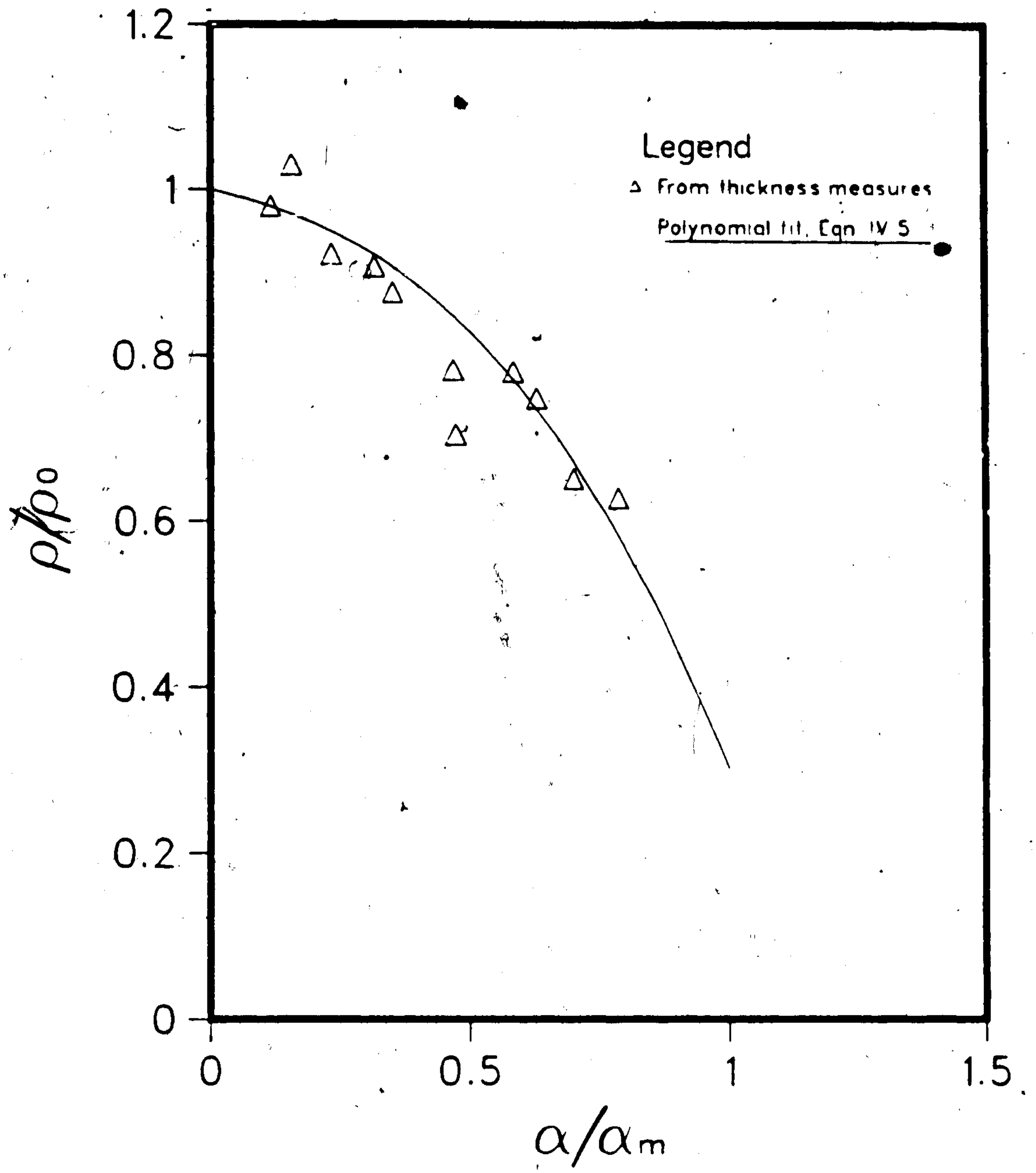


Figure IV.5 - Local densities derived from thickness measurements of shallow ice layers from Table IV.1. The solid curve is a polynomial fit to the data (Equation IV.3).

cross sectional cuts from two different rime accretions (see Figure IV.6). The location, volume and mass loss were recorded for each hole. The mass was measured to within 0.00005 g with a Mettler model H20 balance. For the data of Sample 1, Table IV.2, the hole depths and diameters from three sections of a single accretion were measured to within .05 mm with a Vernier caliper. For Sample 2 of Table IV.2, holes from two sections of accretions grown under nearly identical conditions were measured to 0.01mm in depth and 0.001mm in diameter, with the same bore and depth gauges used by List et al. (1970) to determine local densities in hailstones. The surface angles were approximated by the more easily measured radius angles, since paint layers to define the intermediate surfaces would have altered the densities. This is a good assumption for shallow layers, at any rate, and measurements have been used only from the layer immediately above the cylinder surface. The error in using the radius angle approximation is estimated to be ±five degrees.

Another source of error is mass loss from the entire sample through sublimation and handling during the hole drilling procedure. Several experimental estimates were made of the size of this error, and the average (0.001 g) was subtracted from the measured mass losses from the holes.

The resulting density measurements are shown in Table IV.2 and in Figures IV.7 and IV.8. The solid and dashed curves shown in each of these figures are from Equations

Figure IV.6 - An accretion section of about 1.5 cm. thickness, showing the holes drilled for the local density measurements. The condition are those of Sample 1 in Table IV.2

Sample 1 α (±5°)	Sample 1 ρ (±60 kg/m³)	Sample 2 α (±5°)	Sample 2 ρ (±30 kg/m³)
2	786	2	907
2	667	3	902
2	660	4	723
6	719	6	747
8	620	7	770
8	792	8	909
10	714	13	788
12	645	15	815
12	688	17	766
12	685	19	724
15	629	23	823
18	519	24	683
20	715		
21	630		
27	675		
28	427		
30	681		
30	733		
30	701		
35	539		
38	580		
42	749		
50	504		

Table IV.2 - Experimentally measured local densities as a function of surface slope taken from accretions grown under the following conditions:

D_d(m)	D_c(m)	lwc(g/m³)	U(m/sec)	T(°C)
16.4	0.046	0.53	14.9	-14
14.0	0.0254	0.36	30.0	-10

IV.2 and IV.4, respectively, where the choice of ρ_0 has been based on the density measurements, and α_{max} is calculated from the median volume diameter droplet, following the methods of Chapters II and III.

The scatter of points in both figures is considerable, and is much larger than the measurement errors represented by the error bars will allow. The most likely cause of the large scatter is in the nature of the rime itself. The internal air channels are scattered at random, and since they are often of the same size as the holes, this will introduce a natural scatter. Also, in the very fragile rime, the drilling process may cause walls between adjacent air channels to collapse, resulting in additional mass loss. This problem is especially bothersome near the edges of the accretion, where the density is lowest. Here, walls between the holes themselves will sometimes collapse, making it very difficult to obtain density values at large surface angles. Part of the problem might be alleviated by using accretions samples grown on a larger diameter cylinder, so that the holes do not have to be drilled so close together to get the same interval of surface angle. Unfortunately, the only accretions available were grown on cylinders of one and two inches in diameter.

For whatever reason, the results are too scattered to allow a clear choice between one parameterized distribution and the other. This is due, in part, to the lack of data at

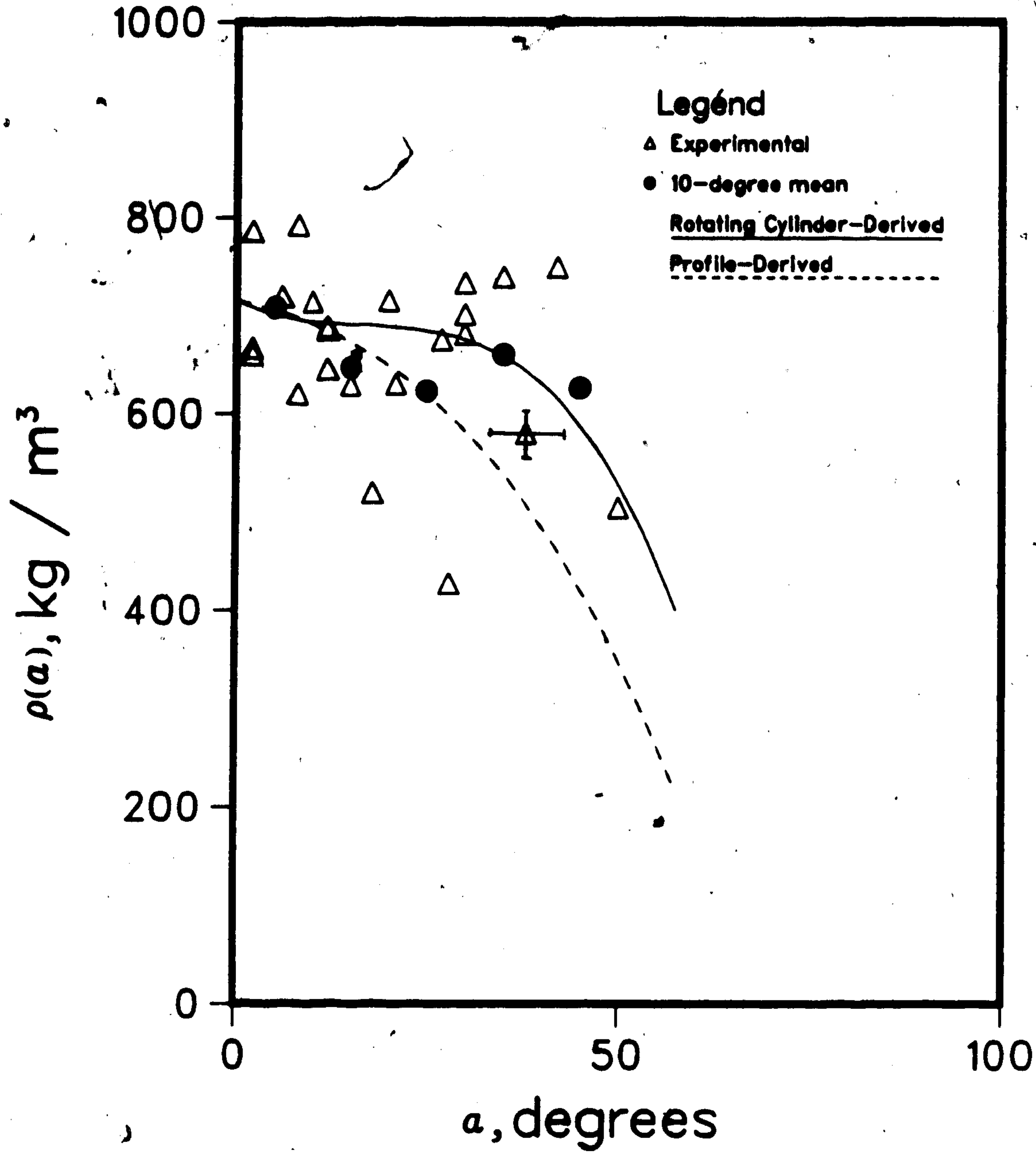


Figure IV.7 - Local density vs. surface angle; experimental measures and parameterized curves for the conditions of Sample 1 in Table IV.2. The filled symbols represent mean values for 10° intervals in surface slope.

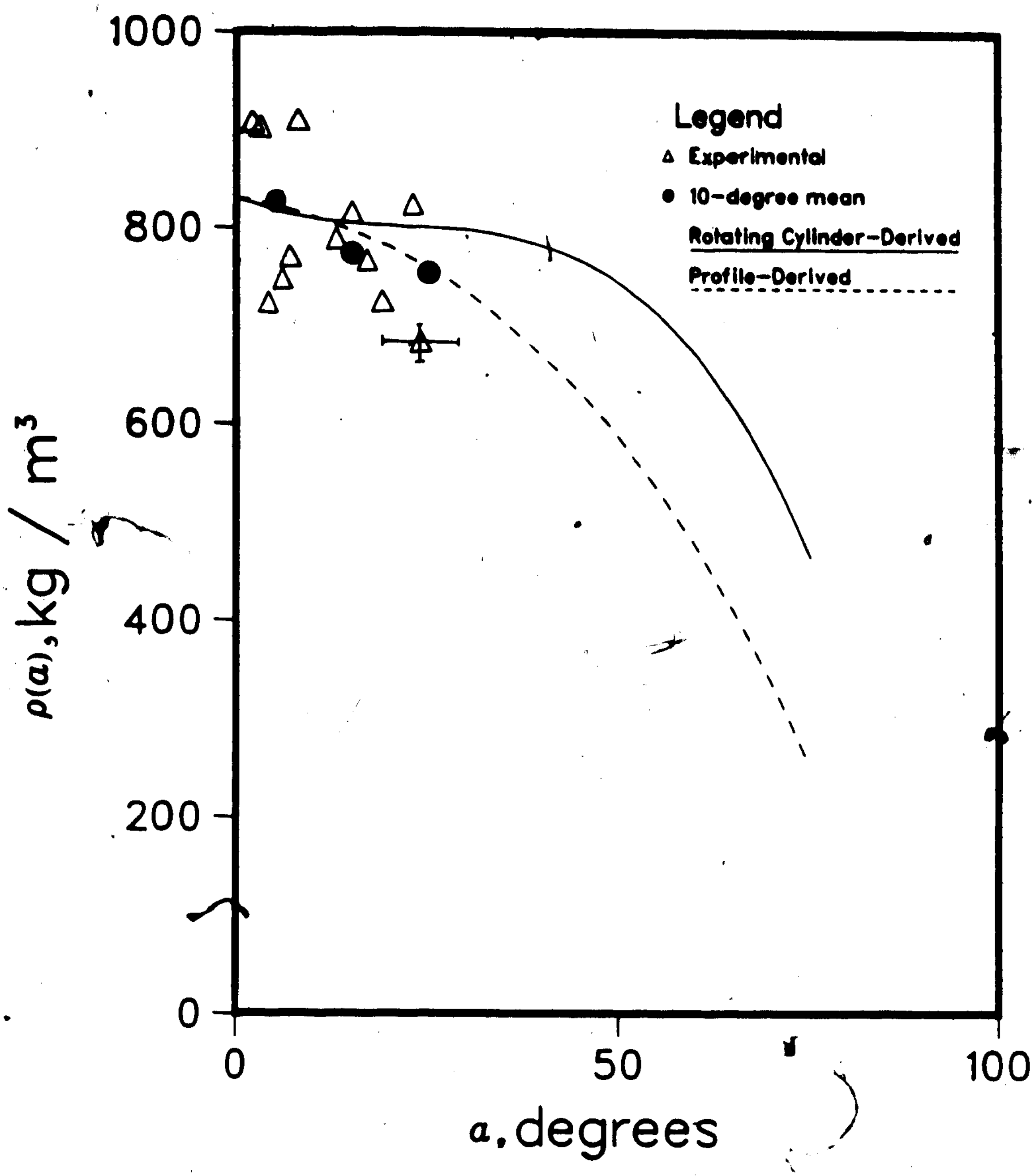


Figure IV.8 - Local density vs. surface angle; experimental measures and parameterized curves for the conditions of Sample 2 in Table IV.2. The filled symbols represent mean values for 10° intervals in surface angle.

large surface angles. When averaged over 10 degree intervals (filled symbols) the experimental data in both figures favour the profile-derived curve up to a surface angle of 30°, but for larger [illegible] Figure IV.7 they lie closer to the rotating cylinder-[illegible]d curve. [illegible] there are far fewer points defining the averages f[illegible] larger angles, and none at all for Figure IV.8.

Although these direct density measurements have not been conclusive, the model results and the practical advantages of the profile-derived distribution justify its adoption over the other. The application of this method is efficient and inexpensive, assuming that the scaling parameters may be easily determined. One of these parameters, the maximum accretion angle, α_{max}, may be quickly approximated according to the analytical function given in Chapter II, but ρ_0 remains to be determined.

E. Local Density at the Stagnation Line

Qualitative observations of many rime accretions show that the stagnation line density also varies with the environmental conditions. Since there are, at present, insufficient quantitative data on this variation, the correlation for rotating cylinder densities, Equation IV.1, may perhaps be applied. The justification for doing so here, but not for the entire density distribution, is that there is no shadowing at the stagnation line, and the conditions are more nearly analogous to the rotating cylinder case.

However, some important differences must be noted between the two situations. In the rotating cylinder correlation, the parameter V_0 is the impact speed of the median volume droplet at the stagnation line only. Over the remainder of the accretion surface, droplets impact at speeds ranging between zero and V_0, so the average impact speed, at least for cases with large α_{max}, is roughly one half V_0, and perhaps somewhat less if α_{max} is small. Thus, V_0 for the rotating cylinder is ≤ twice the average speed of impact. On the stagnation line of the non-rotating cylinder, the impact speed is constant and equal to V_0, so in order to substitute into the R parameter the correct analogy for twice the average speed, we use twice V_0. V_0 is estimated from Equation II.11.

The R parameter is further altered by substituting for the mean surface temperature, the value at the stagnation line ($T_{s,0}$) which is determined using the model of Lozowski, Stallabrass and Hearty (1983). The new correlation parameter is then:

$$R' = 2\, r_d\, V_0 \,/\, T_{s,0} \qquad \text{IV.7}$$

The results are shown in Figure IV.9, for five accretions for which ρ_0 has been experimentally measured by the methods described in the last section. The fit to Equation IV.1 is quite good, considering the size of the measurement errors in ρ_0 (i.e., the same size as for Figure IV.8). Until more experimental data are available, this is an acceptable method for estimating ρ_0.

F. Summary

Previous methods for estimating the local variation of rime ice density are examined, and found to be inadequate. A new method is derived empirically, and tested in model accretions, and against direct laboratory measurements of local density. The laboratory measured distribution is inconclusive, but model results show the new method to be more successful than the old for predicting the shapes and masses of circular cylinder rime ice accretions.

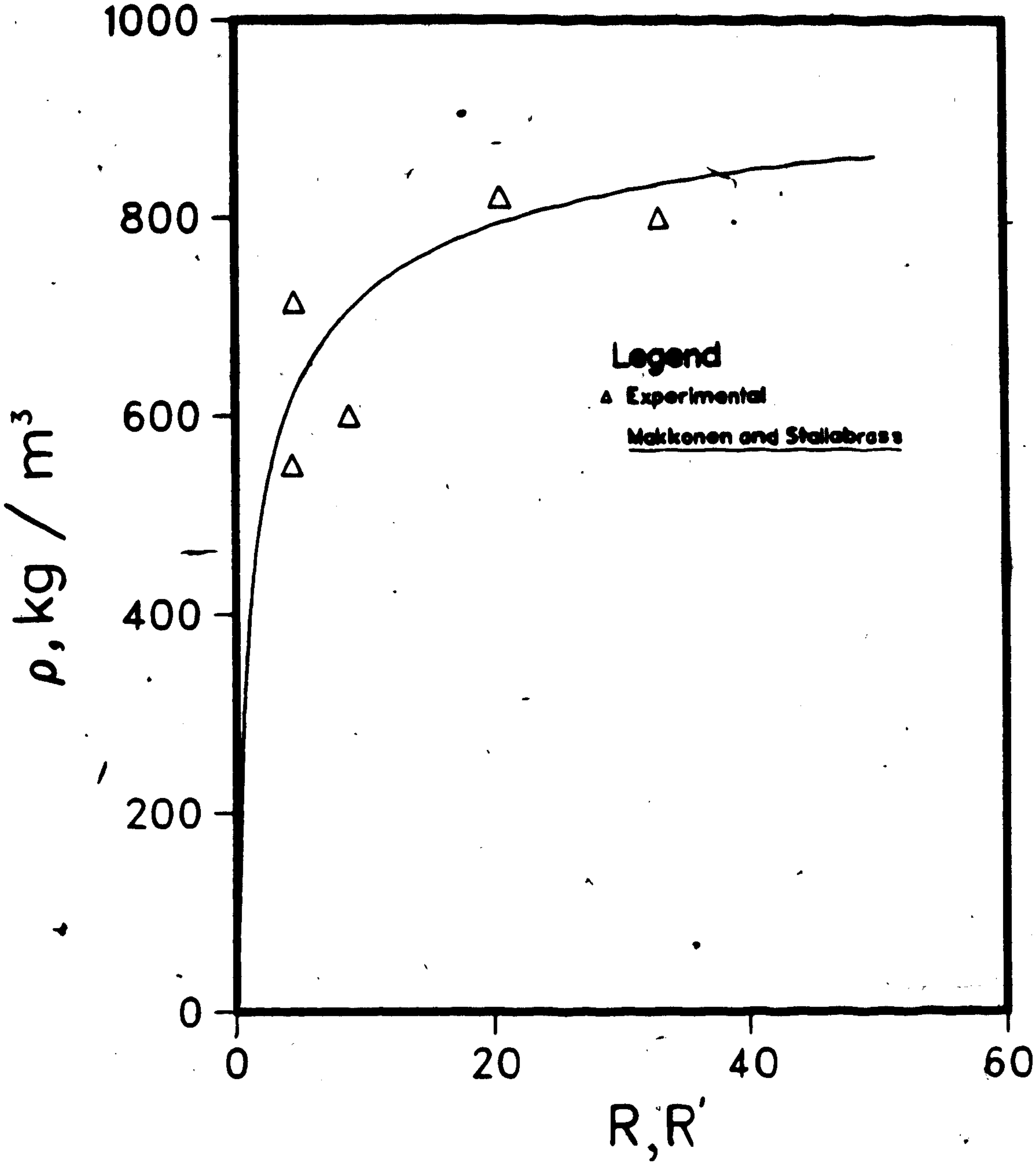


Figure IV.9 - Stagnation line density measurements vs. R' (symbols), and the rotating cylinder density correlation of Makkonen and Stallabrass, Equation IV.2, vs. R (solid line). R and R' are defined in Equations IV.1 and IV.7, respectively.

References

Bain, M. and Gayet, J.F. 1983: Etude de L'accretion de la Glace Atmospherique II. Modelisation Numerique avec Parametrisation de la Variation de la Mass Specifique du Givre en Fonction de la Position Angulaire sur un Cylindre. *J. Rech. Atmos.*, 17, 299-311.

Langmuir, I. and Blodgett, K. 1946: A Mathematical Investigation of Water Droplet Trajectories. Collected Works of Irving Langmuir, Vol 10. Pergamon Press, 348-393.

List, R. Cantin, J.-G. and Ferland, M. 1970: Structural Properties of Two Hailstone Samples. *J. Atmos. Sci.*, 27, 1080-1090.

Lozowski, E.P., Stallabrass, J.R. and Hearty, P.F. 1983: The Icing of an Unheated, Rotating Cylinder Part I: A Simulation Mod Model. *J. Climate and App. Meteor.*, 22, 2053-2074.

Ludlam, F. H. 1951: The Heat Economy of a Rimed Cylinder. *Quart. J. Roy. Meteor. Soc.*, 77, 663-666.

Macklin, W.C. 1962: The Density and Structure of Ice Formed by Accretion. *Quart. J. Roy. Meteor. Soc.*, **88**, 30-50.

Makkonen, L. and Stallabrass, J. 1984: Ice Accretion on Cylinders and Wires. National Research Council Technical Report TR-LT-005, NRC No. 23649, 43 pp.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Cylinder by the Finite Element Method. *J. Atmos. Sci.* 38, 1027-1036.

Rasmussen, R.M. and Heymsfield, A.J. 1985: A Generalized Form for Impact Velocities Used to Determine Graupel Accretional Densities. *J. Atmos. Sci.* 42, 2275-2279.

V. Rime Icing on Airfoils

A. Introduction

A numerical simulation of ice accretion on an arbitrary airfoil section involves greater computational difficulty than does the corresponding problem of icing on circular cylinders. The major reason is the requirement for numerical integration of the impinging droplet trajectories in order to determine collision efficiencies. Unlike the cylinder case, there is (except for the Joukowski airfoil) no analytical solution of the potential flow to simplify the task, nor are there convenient tables of previously calculated collision efficiencies as functions of the flow and inertia parameters.

This requirement adds considerably to the model's consumption of computing time and resources. Previous airfoil accretion models (e.g. Oleskiw, 1982, MacArthur et al., 1982, Cansdale and Gent, 1983) have thus tended to be expensive to run, and may be prone to numerical instabilities.

However, from the results of one such model (Oleskiw 1982), simple parameterizations have been derived for the local collision efficiency distribution with surface slope on NACA 0015 and 0012 airfoils. Three separate analytical functions approximate this distribution fr monodisperse or average droplet spectra, or for any spectrum when the angle of attack is non-zero. Use of these functions requires the determination of only one value of local collision efficiency, that at the stagnation line; thus considerably reducing the computation time. This single value may then be re-determined at intervals during the accretion to give the model a true time-dependence.

These parameterizations have been incorporated into a new, more efficient numerical icing model. Other computational simplifications include the representation of the airfoil and ice surfaces by straight line segments rather than by a spline curve, and a straightforward algebraic method to determine the droplet's point of collision with the surface. An approximate distribution function for the local density has also been introduced. Again, rime icing only is considered, i.e. the air temperature is assumed to be below -10° C, and the impinging water is assumed to freeze immediately and completely. Most of the parameterizations and simplifications could also be used in a model for wet accretion.

The model results are presented in comparison with those of Oleskiw (1982) and with real accretions grown in wind tunnel experiments.

B. Model Outline

A brief outline of the model algorithm is as follows:

Step 1. Specify the x, y coordinates of the original airfoil surface. The coordinates are dimensional, with the origin at the nose of the airfoil.

Step 2. Find the local collision efficiency at the stagnation line, β_0, by the following procedure:

1. Solve for the potential flow field about the airfoil or accretion shape by the method of Kennedy and Marsden (1976).
2. Integrate two droplet trajectories near the stagnation line and find their collision points.
3. Calculate β_0 from $\Delta y / \Delta \ell$, the ratio of initial separation normal to the free-stream of the trajectories to their final separation in arc length on the surface.

Step 3. For each surface point, calculate the thickness accreted perpendicular to the surface during one layer time step τ. The local collision efficiency at each point is approximated from one of three parameterized functions of β_0 and the local surface slope, α, depending on the type of droplet size spectrum and the angle of attack, Φ (defined as the angle between the free-stream direction and the airfoil chord). The local density distribution is also approximated from a parameterized function of surface slope, α.

Step 4. Calculate the mass and mean density of the accreted ice layer.

Step 5. Repeat Steps 3 and 4 for the number of layers requested. Repeat Step 2 only for every fifth layer. The layer time step, τ, should be chosen so that the combined stagnation line thickness of five layers is not more than 1/10 of the original chord length.

Step 7. Plot the original airfoil and all accreted layers.

The sections below will discuss each of Steps 1 through 4 in detail.

C. Specifying the Airfoil Shape

This version of the model employs the standard airfoil section known as the NACA (United States National Advisory Committee on Aeronautics) four-digit wing section, for which Abbott and von Doenhoff (1959) give the thickness distribution in non-dimensional x,y-coordinates normalized to the chord length, as:

$$y = 0.05\, h_0\, (0.2969\, x^{0.5} - 0.126\, x - 0.3516\, x^2 + 0.2843\, x^3 - 0.1015\, x^4) \qquad \text{V.1}$$

The number h_0 gives the maximum airfoil thickness as a percentage of the chord length, as shown in Figure V.1, and also makes up the last two digits of the "four-digit" designation. The first two digits represent c_m, the maximum camber as a percentage of the chord, and c_p, the abscissa of the camber ordinate in tenths of the chord. All of the paramaterizations used in the model are derived assuming that both c_m and c_p are zero. The maximum thickness h_0 is entered as model input, but expressed as a fraction of the chord, rather than a percentage.

Although defined above for non-dimensional coordinates, the model carries the airfoil and accretion profiles in dimensional coordinates throughout most of the calculations, because the chord length (initial chord plus accumulated ice thickness) is changing with every layer step. Where a non-dimensional profile is required for the potential flow calculation, the coordinates are normalized using the current chord length.

Since the profile is always represented by a vector of coordinate points, the points must be of sufficiently high density to ensure that the curved surfaces are well approximated by straight line segments joining them. This avoids the expensive computational process of fitting a few surface points with a spline curve, which may be prone to instabilities. It also simplifies the determination of trajectory impact points, but it it does require more data storage. However, even this can be minimized by using a point distribution scheme which concentrates points in the

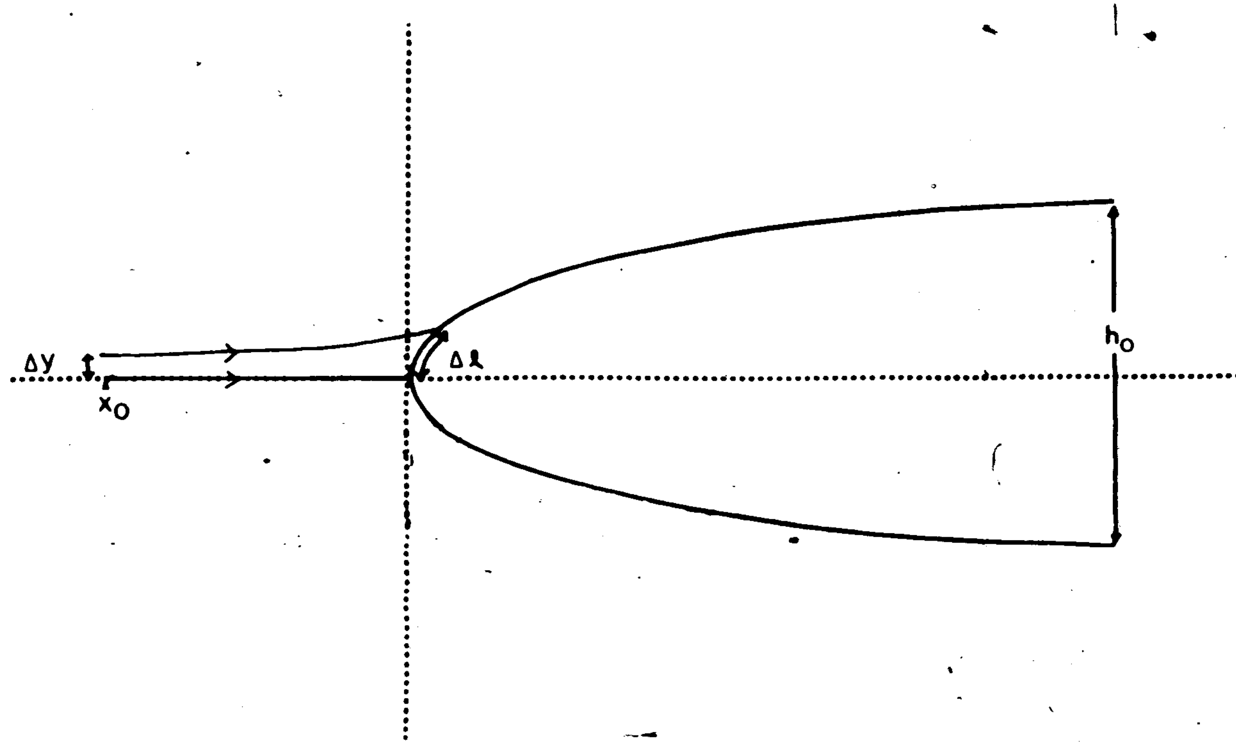


Figure V.1 - Determination of collision efficiency on an airfoil. $\beta = dy/d\ell \simeq \Delta y/\Delta \ell$. Also illustrated is the definition of Φ, the angle of atack.

regions of strong curvature at the nose and tail.

One such scheme is that suggested by Kennedy and Marsden (1976), in which the distribution of non-dimensional x-coördinates is given by:

$$x_i = 0.5\ (1 - \cos(2\pi i\ /\ N)) \qquad \text{V.2}$$

where i lies between 1 and N, an even integer. This scheme was found to give good results with N = 600 to specify the complete airfoil section for plotting purposes, although only half this number of points, defining the front portion of the airfoil, is allowed to accrete ice.

In the potential flow solution, a profile consisting of a much smaller number of points is desired to reduce the computational requirements; Kennedy and Marsden suggest that 40 points is an optimum number. This can be simply constructed from either the original airfoil profile, or from subsequent accretion shapes by taking every 15th point of the denser profile. The new sparse profile will also retain the desired distribution pattern of points.

Accretion thicknesses are calculated for each surface point in the direction perpendicular to the local surface slope. This causes the point separation to increase near the nose with each succeeding layer. After several layers, the separation may have become too large for a reasonable representation of the surface by straight line segments, and has destroyed the original point distribution.

A way to remedy the situation is to shift each coordinate point, after each layer is accreted, closer to the nose along the straight line segment joining it and its nearest neighbour (whose position has already been shifted). The amount of the shift must be carefully chosen to preserve the desired distribution of points, and an accurate profile. Because the original distribution concentrated points near the nose, and because points near the nose diverge faster than elsewhere, those points must be shifted by larger amounts (relative to the segment length) than points further away.

A shift of half the segment length at the nose, decreasing linearly to one eighth at about half the chord length gives the desired results. To test that profiles produced by this scheme are accurate, model profiles were also made with four times the optimum point density. There was no noticeable difference in the resulting plotted profiles.

D. Local Collision Efficiency

Parameterization schemes were derived in Chapter III for local collision efficiency β on circular cylinders as a scaled cosine function of local surface slope α. If the numerical results of Oleskiw (1982) for local collision efficiency on a NACA airfoil at 0° angle of attack are plotted against surface slope, they also closely follow a cosine curve, but one which is displaced negatively along the abscissa. The form holds for several cases computed by Oleskiw, as seen in Figures V.2, V.3 and V.4, and is described by:

$$\beta(\alpha) = \cos(\alpha) - (1 - \beta_0), \qquad \text{V.3}$$

where β_0 is the value of β at the stagnation line. This is the only parameter required to approximate the entire distribution, since the maximum accretion angle α_{max}, in radians, follows from:

$$\alpha_{max} = \arccos(1 - \beta_0). \qquad \text{V.4}$$

Appendix IV gives a geometrical derivation of Equation V.3.

So far, a monodisperse droplet population has been assumed. In practice, of course, a spectrum of droplet sizes is present. Often the collision efficiency of the entire population is modelled by a single droplet size at the median volume diameter of the spectrum. This is generally a good approximation to the weighted average $\beta(\alpha)$ distribution calculated according to the relative volumes of the droplet sizes in the spectrum, as discussed in Appendix VI.

In real populations, when relatively large numbers of very small droplets are present in the population, they can change the shape of the β distribution, as well as force the mvd-based maximum accretion limit to a smaller value

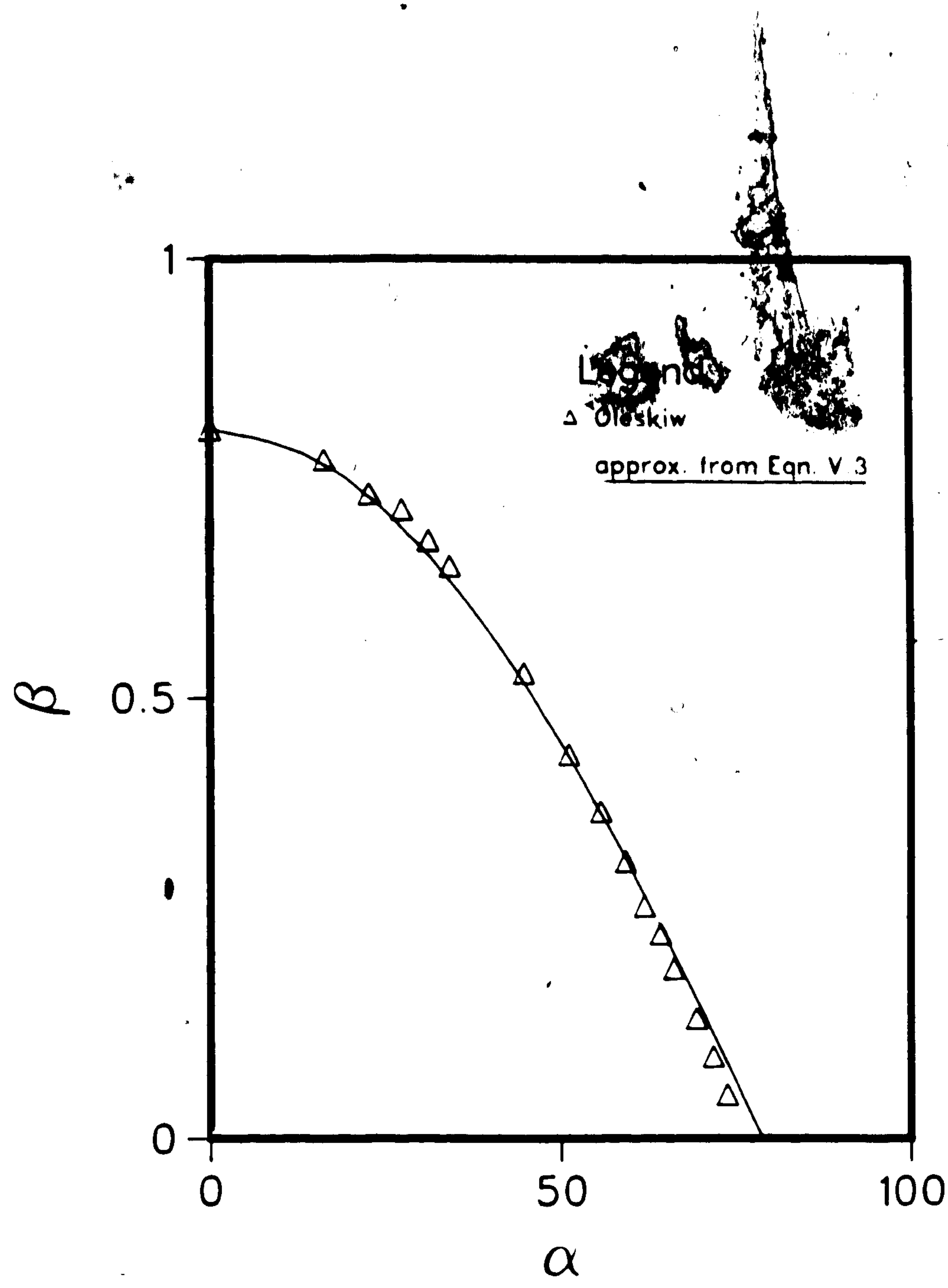


Figure V.2 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a NACA 0015, U = 61 m/sec, Chord = 0.213 m, D_d = 20 μm.

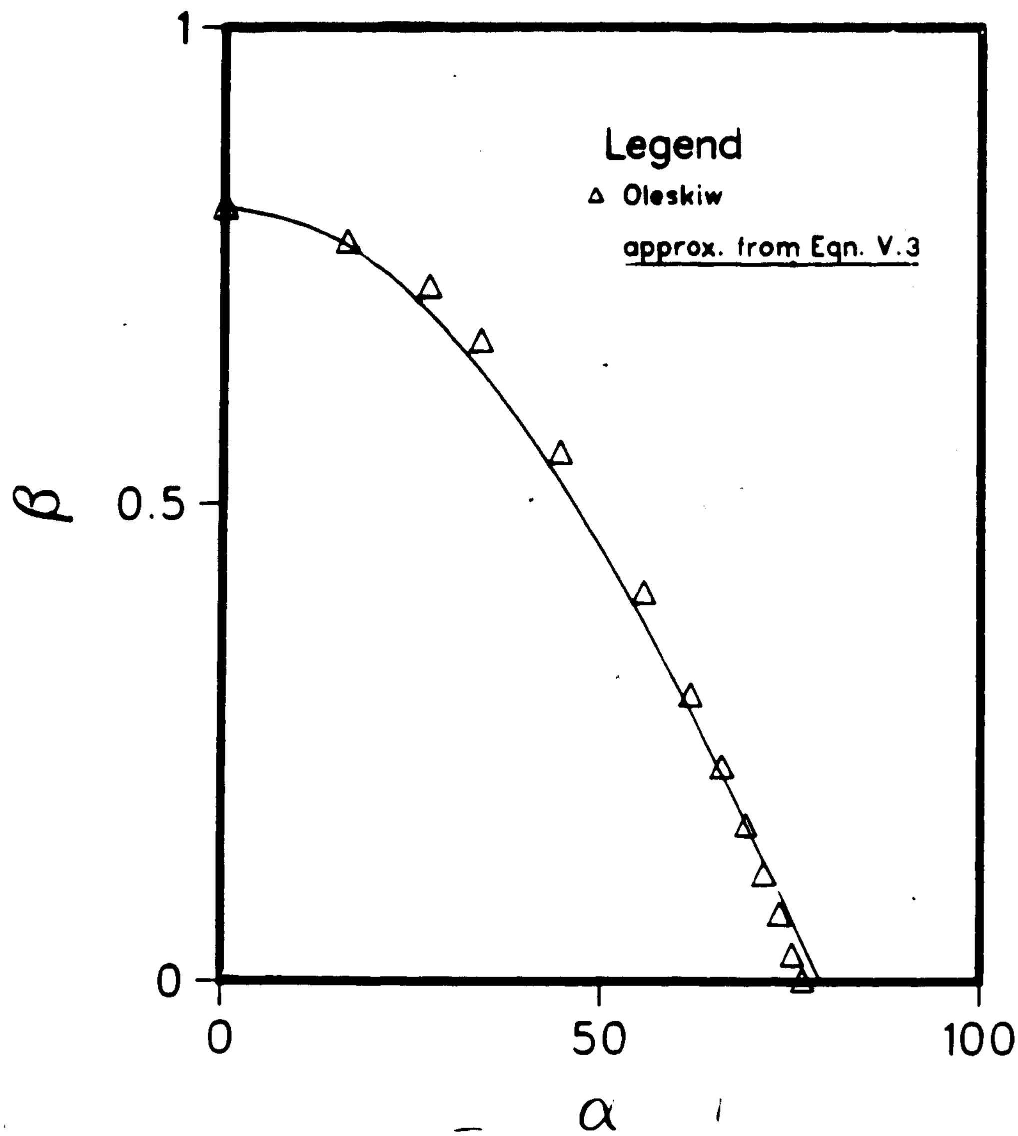


Figure V.3 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a Joukowski 15%, U = 78.2 m/sec, Chord = 0.33 m, D_d = 25.5 μm.

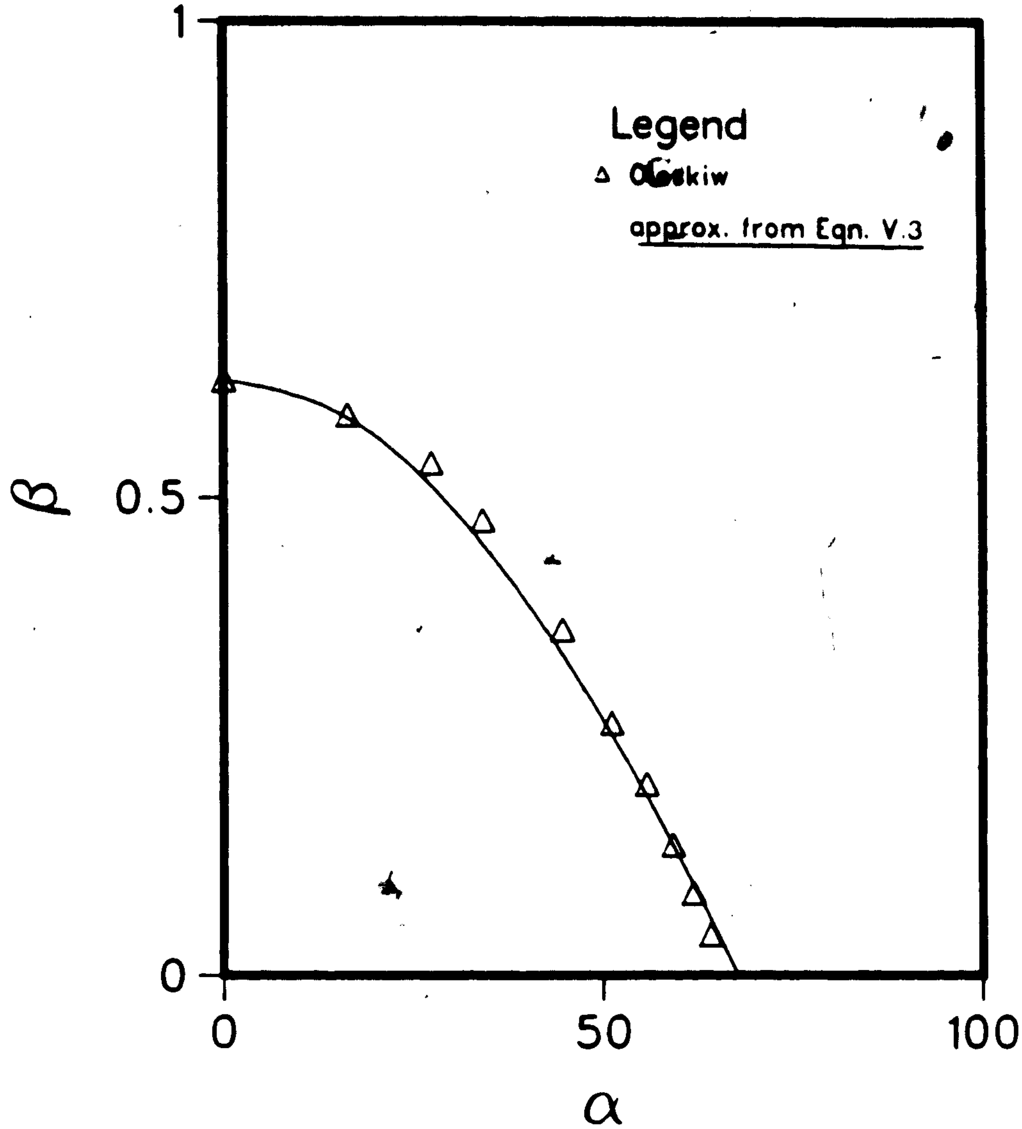


Figure V.4 - Comparison of Oleskiw's numerical results and Equation V.3, for collision efficiency of a single drop size. Airfoil is a Joukowski 15%, U = 78.2 m/sec, Chord = 0.33 m, D_d = 13.2 μm.

than actually occurs. For airfoils, the effect becomes important for β_0 less than about 0.8. For these cases, an alternative parameterization is proposed. It is derived from a polynomial fit to an average of several spectrum-weighted $\beta(\alpha)$ curves calculated by the author for droplet spectra observed in the FROST tunnel at the University of Alberta. The form of the parameterization is:

$$\beta(\alpha/\alpha_m) = \beta_0 \, (1 + 0.039(\alpha/\alpha_m) - 1.842(\alpha/\alpha_m)^2 - 0.543(\alpha/\alpha_m)^3 + 1.792(\alpha/\alpha_m)^4 - 0.444(\alpha/\alpha_m)^5, \qquad \text{V.5}$$

where $\beta_0 = \beta_{0,mvd}$, and

$$\alpha_m = \alpha_{max,mvd} + 0.13 \exp(-mvd \,/\, 70). \qquad \text{V.6}$$

Here α_m is again in radians, and mvd is in μm. In Figures V.5 and V.6, the approximation of Equation V.5 is compared with spectrum-weighted distributions for two other observed droplet spectra, which appear in Table A.1 as "Battan and Reitan stratus" (Battan and Reitan, 1957), and "FROST 3", a spectrum measured in the FROST tunnel. These figures also show the difference in β_0 between the spectrum-weighted value and that derived from the mvd for these cases.

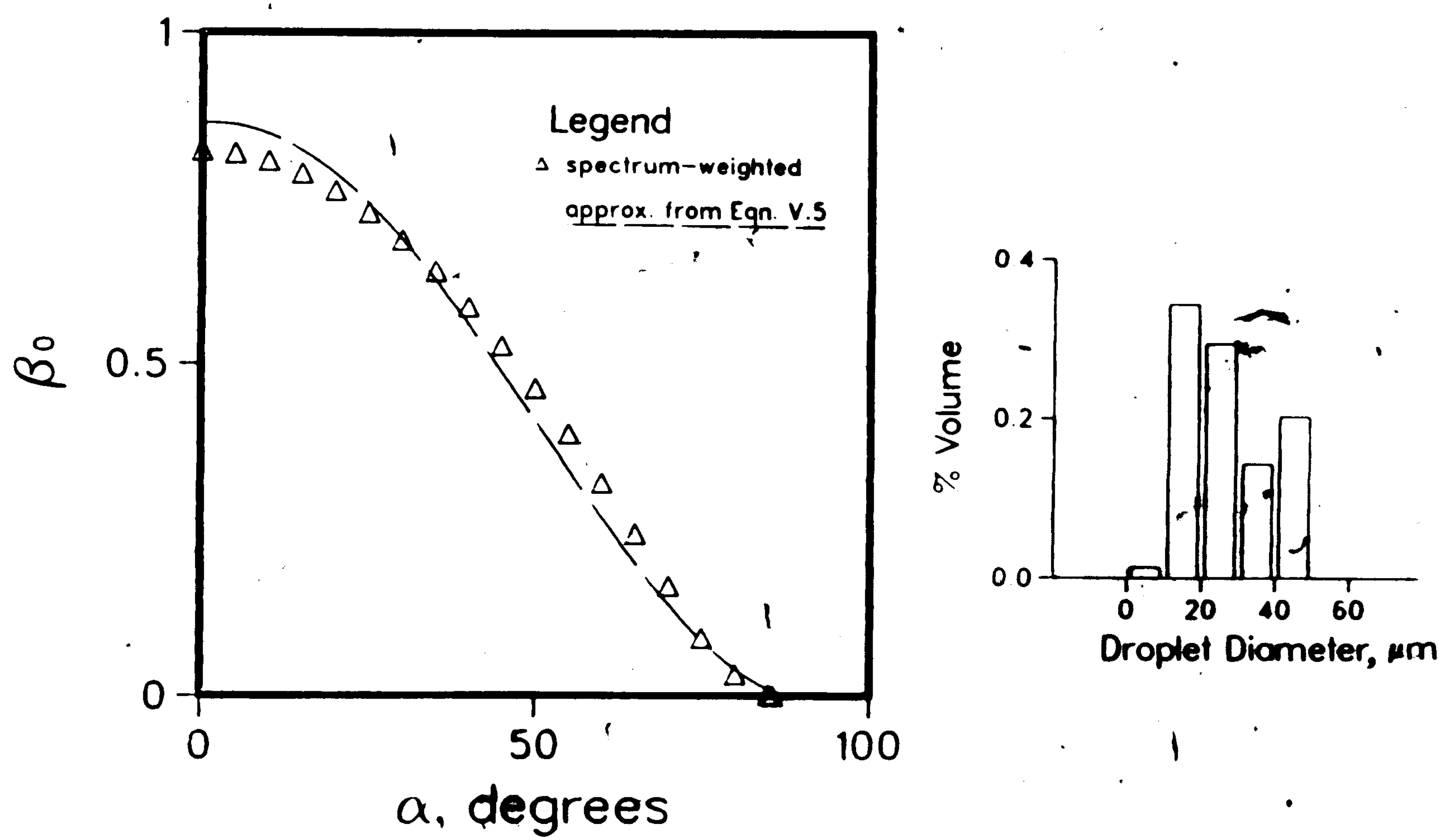
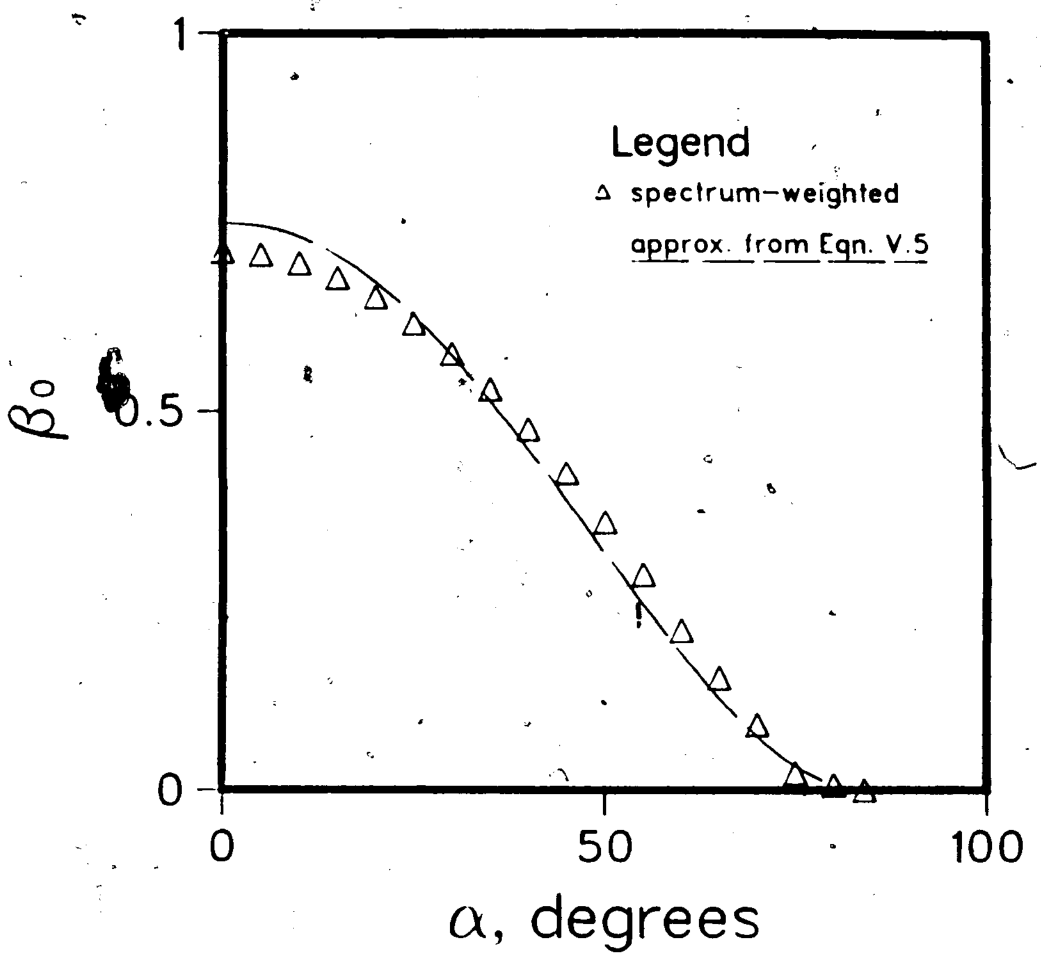


Figure V.5 - Comparison of spectrum-weighted collision efficiencies and Equation V.5 for the spectrum shown (mvd = 26.2 µm) on a NACA 0015, U = 50 m/sec, chord = 0.2 m.



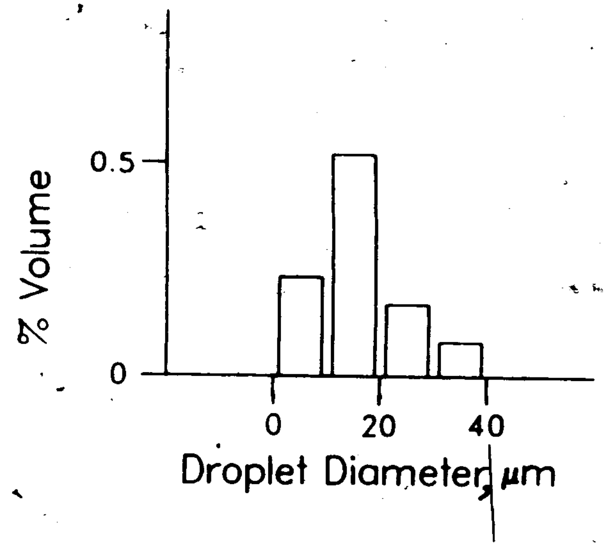


Figure V.6 - Comparison of spectrum-weighted collision efficiencies and Equation V.5 for the spectrum shown (mvd = 17.2 μm) on a NACA 0015, U = 50 m/sec, chord = 0.2 m.

When the angle of attack (Φ) is non-zero, Oleskiw's numerical results have been fitted with the following function:

$$\beta(\alpha') = 0.95\ \beta_0 \cos(\alpha'), \qquad \text{V.7}$$

where β_0 is calculated as for the $\Phi = 0$ case, and

$$\alpha' = ((\pi/2)\ (\alpha_0 - \alpha)\ /\ (\pi/2 - \alpha_0)). \qquad \text{V.8}$$

The position of maximum collision efficiency has been designated by α_0, and may be approximated by:

$$\alpha_0 = \pi \log_{10} ((180\ \Phi/\pi) + 1)\ /\ (0.180 \exp(30.5\ h_0)), \qquad \text{V.9}$$

where α_0 and Φ are in radians.

Unfortunately, there are only two numerical cases currently available (from Oleskiw, 1982) to test this fit, as illustrated in Figures V.7 and V.8. Further calculations at some time in the future may help to refine the approximation. In the meantime, it does give a rough estimate for the $\beta(\alpha)$ distribution when Φ is non-zero, which is otherwise very costly to compute.

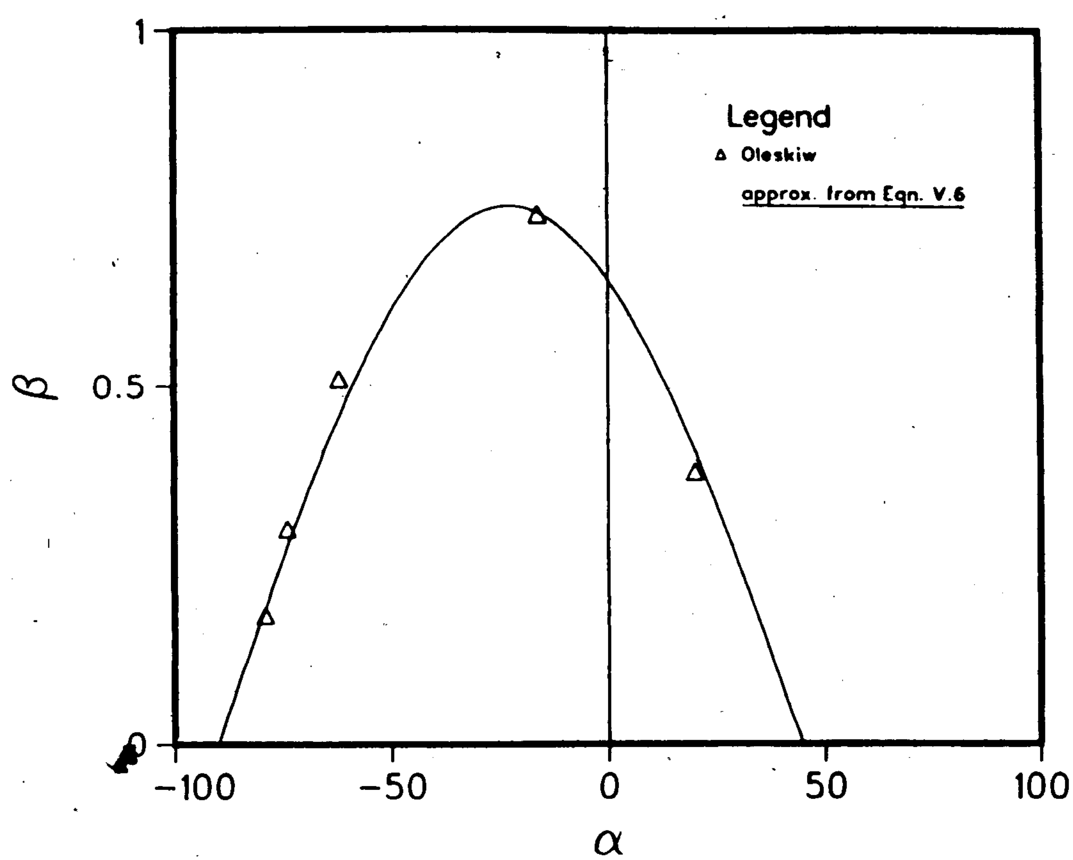


Figure V.7 - Comparison of Oleskiw's numerical results and Equation V.7, for collision efficiency on a NACA 0015 at an angle of attack = 8°, U = 30.5 m/sec, Chord = 0.169 m, D_d = 20 μm.

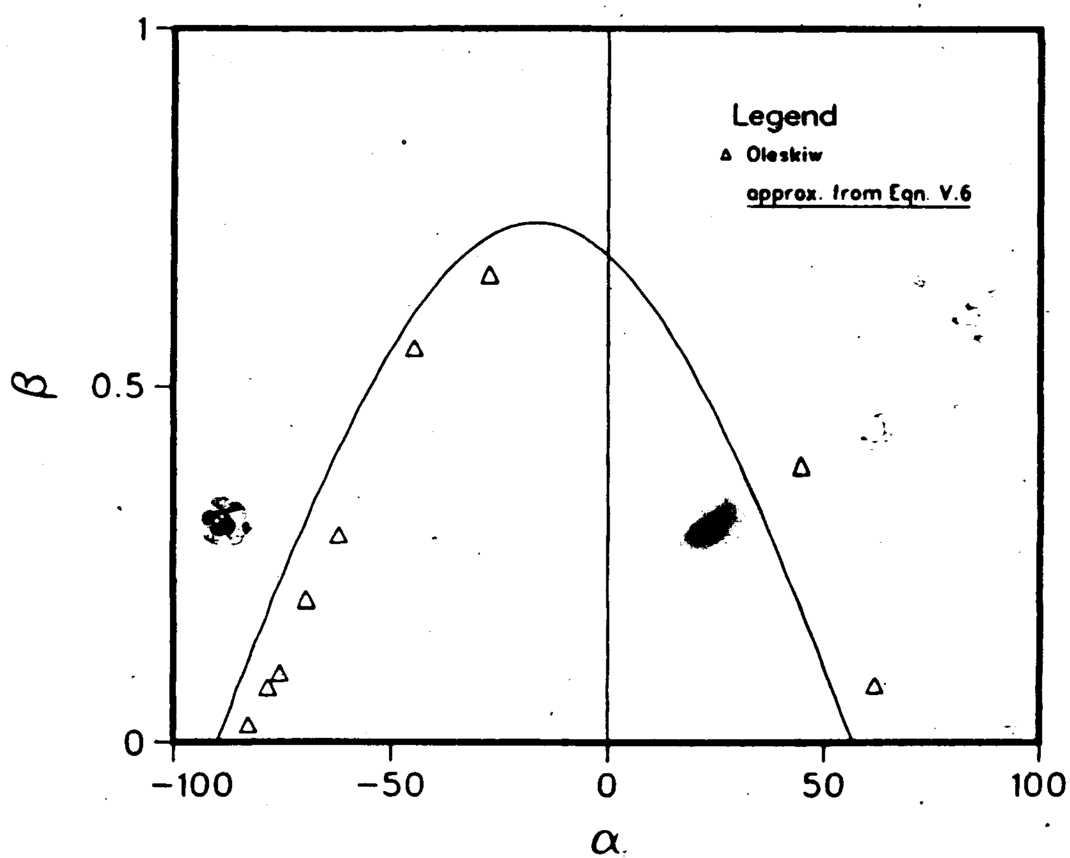


Figure V.8 - Comparison of Oleskiw's numerical results and Equation V.7, for collision efficiency on a Joukowski 15% at an angle of attack = 4°, U_∞ = 78.2 m/sec, Chord = 0.33 m, D_d = 18.6 μm.

E. Integration of Droplet Trajectories

The equations of motion of a cloud droplet carried by the air stream, in a more general form than for the cylinder problem (Equation II.6), are:

$$dv_x/dt = -3/4\ (\rho_a/\ \rho_d)\ (C_d/D_d)\ V\ (v_x\ -\ u_x)$$

V.10

$$dv_y/dt = -3/4\ (\rho_a/\ \rho_d)\ (C_d/D_d)\ V\ (v_y\ -\ u_y)$$

In this steady-state formulation the additional term due to gravity has been neglected, as has the "history" term, arising from the droplet's acceleration with respect to the flow. Neglect of these terms is justified by the results of Oleskiw (1982), and also by examination of the "acceleration modulus", (Norment, 1980):

$$N_A = D_d\ \left|\frac{dv}{dt}\right|\ /\ v^2 \qquad \text{V.11}$$

If this quantity is less than about 0.01, the history term may be safely neglected. The results of Oleskiw and of Norment showed that this limit is not exceeded in most aeronautical modelling situations.

The drag coefficient is again estimated from the the expressions of Beard and Pruppacher (1969), Equation II.9. The droplet trajectories are integrated by the Heun method (Mesinger and Arakawa, 1976), similar to the calculations in

Chapter II, except for the determination of the air speeds u_x and u_y.

The streamfunction ψ of the potential flow field about the current accretion shape is numerically calculated following the method of Kennedy and Marsden (1976). This is a version of the surface singularity method originated by Hess and Smith (1967), in which the airfoil surface is represented by a set of point vortices. Kennedy and Marsden have slightly altered, at the trailing edge, the usual boundary condition of a constant streamfunction along the airfoil surface. They claim this gives a very accurate result with a minimum of computing time. Full details are available in their paper. The air speeds may then be approximated from a finite-difference calculation of the streamfunction derivatives, with the spatial differences equal to the droplet diameter.

The droplet's initial x-coordinate is five chord lengths upstream of the nose, and the initial separation of the two trajectories, Δy, is 10^{-8} metres. An optimum time step was found to be $\Delta t = 0.0025$ (U/chord). The justification for these choices is given by Table V.1, which examines the sensitivity of the model results to changes in these parameters.

As the droplet approaches the airfoil, the model begins testing for its position with respect to the current accretion surface. At each time step Δt, the array of accreting surface points is searched for the pair of

Δt (chord/U)	β_0 (x_0=10 • chord)	β_0 (x_0= 5 • chord)
0.005	0.787	0.812
0.0033	0.794	0.804
0.0025	0.796	0.801
0.002	0.798	0.799

Table V.1 - Sensitivity of the integration results to the time step size Δt, and the initial x-position x_0. Trajectory is for a 20 micron drop impinging on a NACA 0015 airfoil of 0.213 m chord, in a 61 m/sec airstream.

adjacent points which bracket the current y-coordinate of the droplet. The droplet's x-coordinate is then compared to the x-coordinates of these two points. If either or both of them are greater than the droplet's x-coordinate, the current droplet position lies inside the accretion surface.

The collision point is then estimated by solving for the intersection of the straight line segments, one joining the two bracketing surface points, and one joining the current and immediately previous droplet positions (see Figure V.9). The arc length $\Delta \ell$ separating the two collision points on the surface is also approximated as a straight line. β_0 is then just $\Delta y/\Delta \ell$.

All of these straight line approximations, and the simple integration method, are sufficiently accurate because of the very small distances between surface points near the nose, the small Δt and initial Δy of the trajectories.

A comparison of the integration results with those of Oleskiw for his Case 55 shows very good agreement. Oleskiw calculates $\beta_0 = 0.805$ for a 20 μm drop impinging on [illegible] m chord NACA 0015, at 0° angle of attack and 61 m/s. [illegible] present model result is 0.801 for the same initial conditions. The small difference is easily account[illegible] the use of a slightly different formulation for drag coefficient and by Oleskiw's method of approximating β_0. This is from a spline fit to several $\beta(\alpha)$ points that are distributed along the surface, away from the stagnation line.

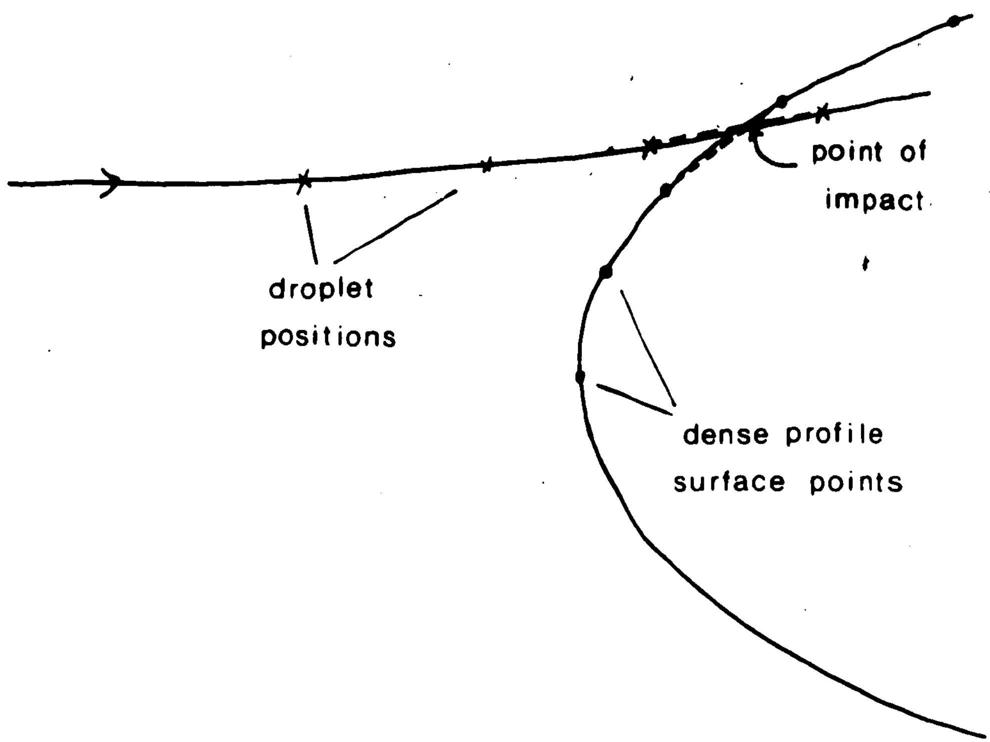


Figure V.9 - Determination of droplet impact point on the surface.

F. Local Density Variations

Once β_0 has been determined, α_{max} and the distribution of local collision efficiency may be approximated from Equations V.3, V.5 or V.7. The local ice thickness (Th) accreted over a layer time step τ at each surface point is then given by the dry growth equation:

$$Th = \beta(\alpha)\ U\ lwc\ \tau\ /\ \rho(\alpha), \qquad V.12$$

where now a suitable distribution for $\rho(\alpha)$ must be found.

This would ideally be done using the same techniques that were used in Chapter IV, to investigate circular cylinder accretion densities by direct measurement of local densities and of layer thicknesses. However, the wind tunnel experiments on airfoils which were performed in the FROST tunnel for this study, were grown on airfoils with thicknesses less than one inch. This is too small to allow effective measurements by the hole drilling technique, and the layers outlined by paint spray for some of the experiments were too thick to allow good results using the layer method.

Therefore, an approximate distribution was derived by plugging the above-mentioned circular cylinder curve into the model, and adjusting it until it could produce reasonable profiles. The result works best when used with the 'simulated spectrum' parameterization for local collision efficiency. We emphasize that this distribution is not yet supported by any empirical evidence, and that it should be adjusted further when such evidence is available. For the present however, a good estimate for the density distribution is:

$$\rho(\alpha/\alpha_m) = \rho_0\ (1 - 0.163\ (\alpha/\alpha_m) + 0.235(\alpha/\alpha_m)^2 - 2.357(\alpha/\alpha_m)^3 + 1.374(\alpha/\alpha_m)^4 + 0.019(\alpha/\alpha_m)^5), \quad \text{V.13}$$

where ρ_0 is assumed, for simplicity, to be always 890 kg/m^3. In certain conditions, the maximum density may be less than this value, but the model has no capability to determine this, at present.

The mass and mean density of each layer are calculated in the same way as for the circular cylinder described in Chapter III.

G. Model Results

Available data for model validation include the numerical results of Oleskiw, and a number of wind tunnel experiments carried out in the University of Alberta FROST tunnel, at NASA/Lewis and described in Olsekiw (1982b), at the National Aeronautical Establishment by Stallabrass (1958), and at the National Research Council Low Temperature Laboratory by Lozowski and Stallabrass (1978). In all of the experimental cases, the droplet sizes were determined by the oil slide method, and have been corrected by us according to the calibration of Makkonen and Stallabrass (1984) discussed in Appendix III. An exception is the case taken from Stallabrass (1958), for which the liquid water content given by Stallabrass was derived from the observed ice thickness and the oil slide measurement of mvd. Stallabrass' numbers have thus been retained for both the droplet size and lwc. In all other cases, however, the accretion conditions were measured independently of the resulting thicknesses.

Figure V.10 illustrates the modelled profiles of Oleskiw and of the present model for identical accretion conditions. Since a constant local density was used, the agreement in shape of the layers between the two models reflects the accuracy of the parameterized collision efficiency distribution, for this case Equation V.3. The present model also lacks the surface instabilities which are already appearing in the Oleskiw model profile, arising from the spline curve approximation to the layer surface. Most importantly, the bottom profile of Figure V.10 represents a reduction in computational costs of a factor of ten over the Oleskiw model.

The conditions of Figure V.10 were chosen to agree with those of an experimental case grown in the NASA/Lewis tunnel. This experimental profile appears in Figure V.11, together with a corresponding model simulation using a constant local density, and one using the 'simulated spectrum' version of the local collision efficiency

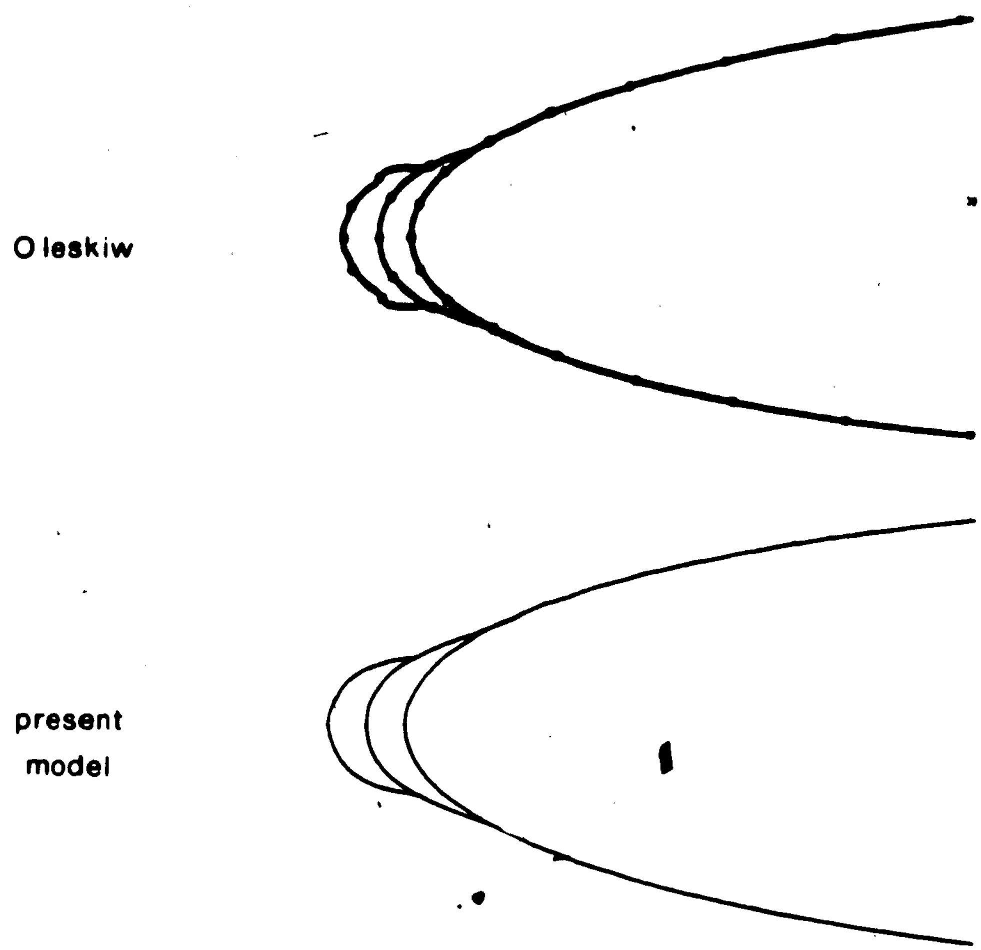


Figure V.10 - Comparison of present model results with those of Oleskiw (1982b) for a constant density accretion on a NACA 0015 at $\Phi = 0°$, $U = 58$ m/sec, lwc = 1.02 g/m^3, chord = 0.533 m, t = 5 min. A monodisperse droplet population at 12 μm diameter is assumed, and an air temperature of -15° C.

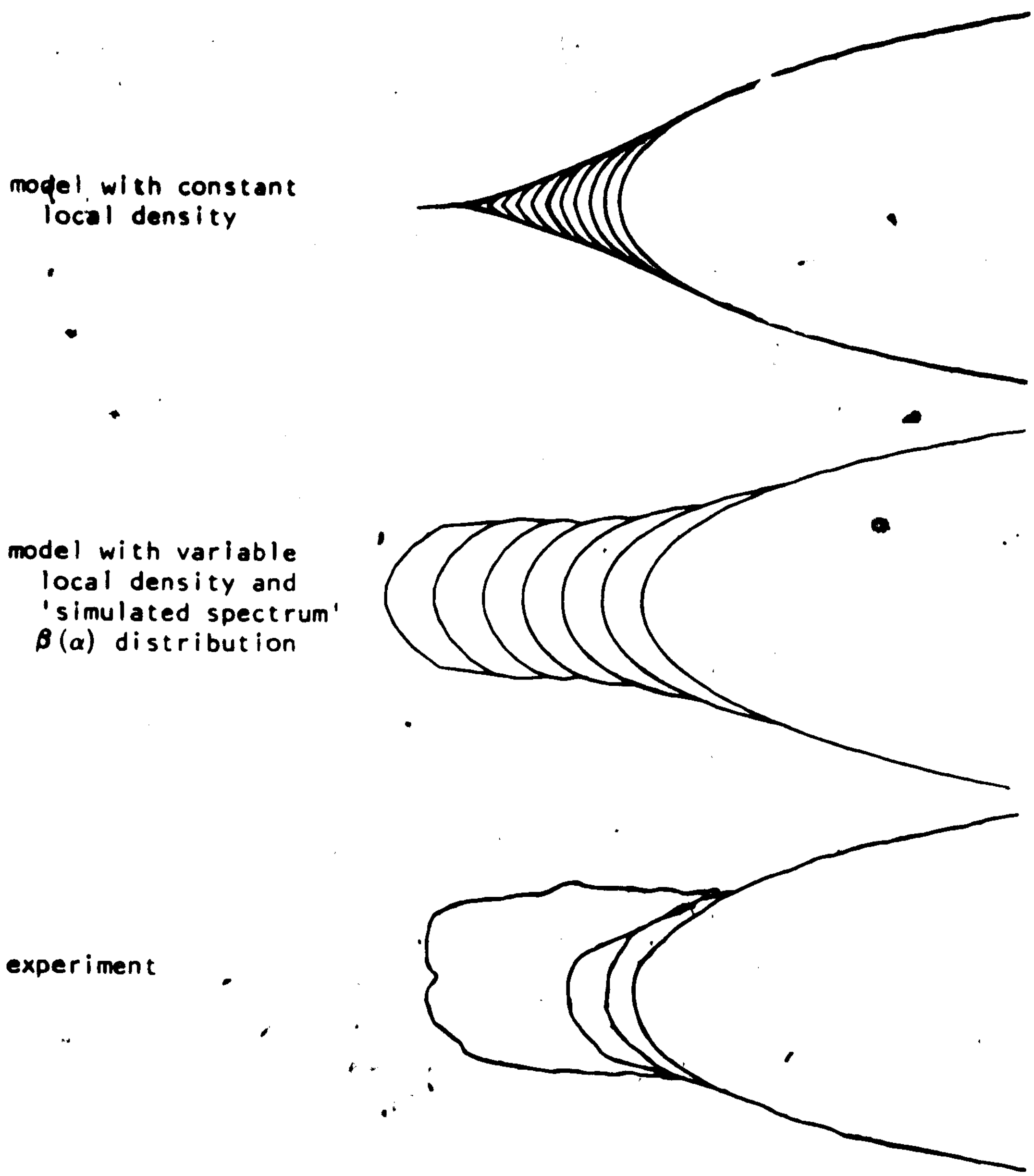


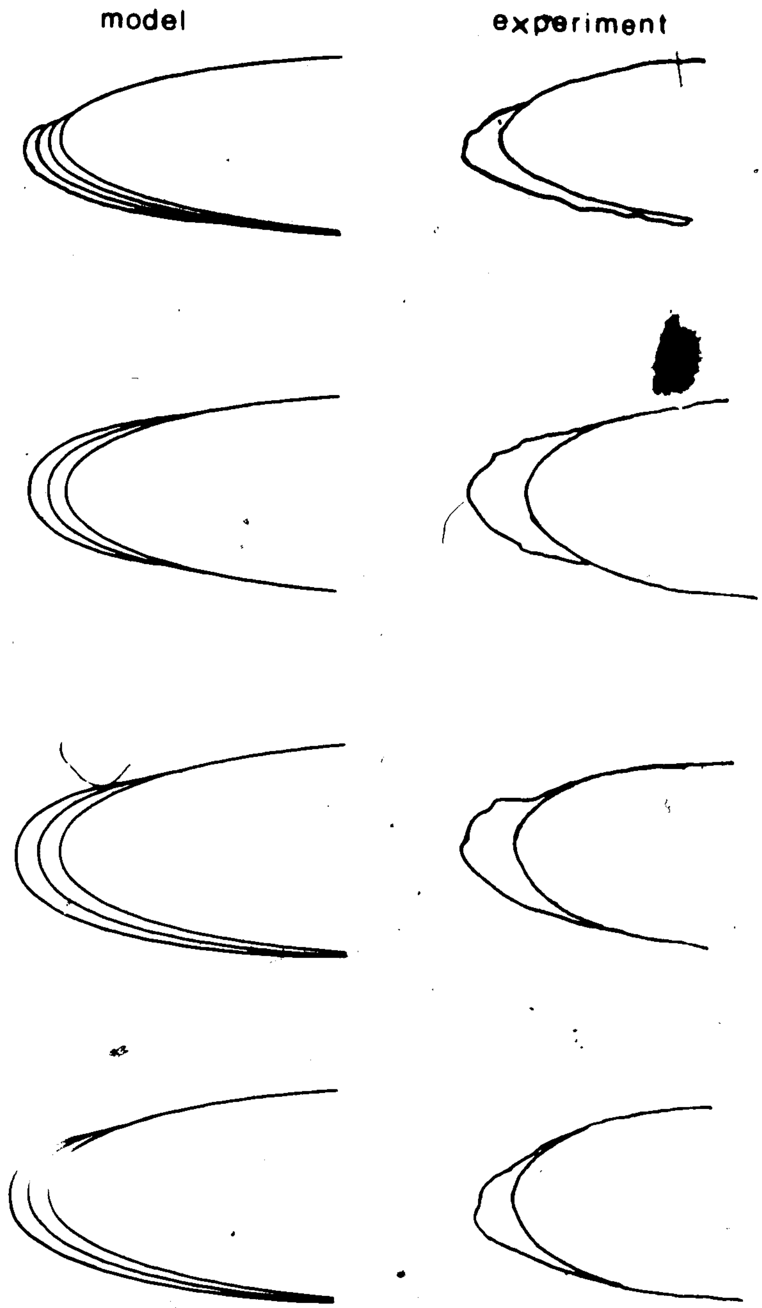
Figure V.11 - Comparison of present model results with an experimental accretion grown in the NASA/Lewis tunnel and described in Oleskiw (1982b), for the same conditions as in Figure V.10, but for a NACA 0012 airfoil, t = 15 min. Intermediate profiles at t = 2 and 5 minutes are also marked.

distribution, Equation V.5, together with the local density distribution of Equation V.12.

The two modelled profiles demonstrate a great improvement, in terms of a realistic accretion shape, in using the new local density and 'simulated spectrum' distributions over the old constant density and monodisperse spectrum versions. The model profiles are also seen to be free of surface instabilities and thus capable of accreting many more layers than the Oleskiw model.

Except for the forward surface of the experimental accretion, which seems to contain a small central trough, the agreement in shape is very good. Central troughs and ridges are features which are often seen to form on wind-tunnel accretions. They most probably arise from disturbances in the laminar flow pattern near the nose, as the ice layer alters its shape. The simple, potential flow model does not include such effects, and such features are not expected to show up in these simulations.

The experimental results of Lozowski and Stallabrass (1978) and of Stallabrass (1958) allow a test of Equation V.7, for local collision efficiency at non-zero angles of attack. Figure V.12 shows four experimental accretions and the corresponding model profiles, all including density variation and local collision efficiencies estimated for the $\Phi \neq 0$ case from Equation V.7. Since the method of approximating accretion limits for $\Phi \neq 0$ is somewhat crude,



chord = 0.417 m
$\Phi = 5.7°$
$D_d = 30\ \mu m$
lwc = 0.22 g/m^3
U = 60 m/sec
t = 14.3 min

chord = 0.213 m
$\Phi = 0°$
$D_d = 14.2\ \mu m$
lwc = 0.40 g/m^3
U = 61 m/sec
t = 5 min

chord = 0.213 m
$\Phi = 4°$
$D_d = 14.2\ \mu m$
lwc = 0.36 g/m^3
U = 91.5 m/sec
t = 4 min

chord = 0.213 m
$\Phi = 8°$
$D_d = 14.2\ \mu m$
lwc = 0.41 g/m^3
U = 61 m/sec, t = 5 min

Figure V.12 - Comparison of present model results with experimental profiles from Stallabrass (1958, top profile, NACA 0012) and Stallabrass and Lozowski (1978, lower three profiles, Air temperature is -15° C.

comparison in this region shows only rough agreement. However, the general agreement in shape and stagnation line thicknesses is encouraging. These cases at least illustrate that the approach taken in Equation V.9, to approximate the position of maximum collision efficiency α_0, is worth pursuing and refining, especially given the large computational costs of determining this position from a series of droplet trajectory integrations.

For all of the cases mentioned so far, accretion masses have not been available. However, a series of wind-tunnel experiments were carried out in the University of Alberta FROST tunnel, for which was measured the mass of ice accreted on a 15 cm length of a NACA 0012 airfoil under various conditions. These results, with intermediate layers as marked by paint spray, are shown in Figure V.13 together with the modelled masses and profiles.

The differences between the model masses and the experimental masses for the two cases are 11 and 16 percent. This is of the same order as typical measurement errors for input parameters such as droplet size and liquid water content.

These examples have shown the usefulness of a simple approach toward numerical methods for ice accretion modelling. The present results are comparable to and much less costly than those of the Oleskiw model, and are of

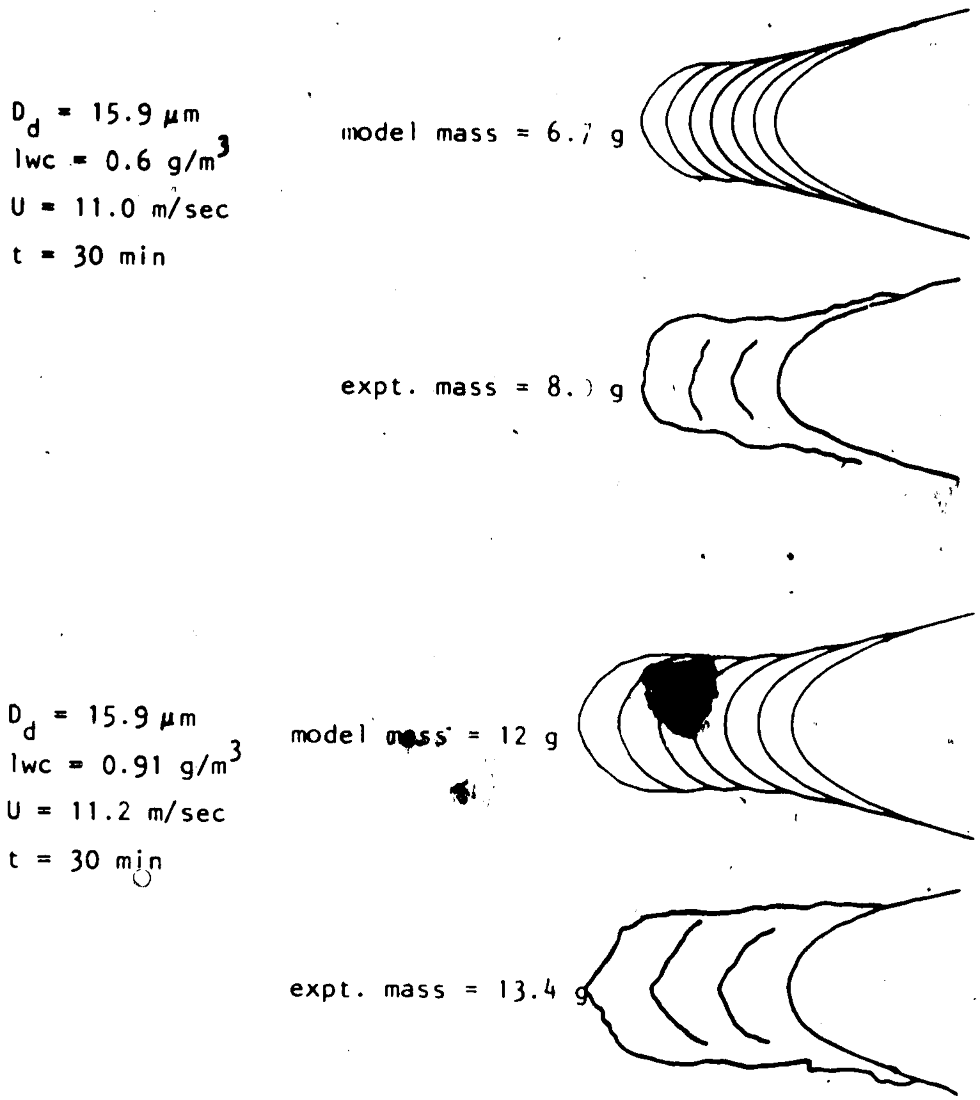


Figure V.13 - Comparison of present model results with experimental profiles and masses of accretions grown in The University of Alberta FROST tunnel on a NACA 0012 at $\Phi = 0°$, chord = 0.176 m, air temperature = -10° C.

sufficient accuracy to allow practical and realistic comparisons with rime experiments. Such methods will also help to make the modelling problem for wet icing more tractable.

References

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

Beard, K.V. and Pruppach, H.R. 1969: A Determination of the Terminal Velocity and Drag of Small Water Drops by Means of a Wind Tunnel. *J. Atmos. Sci.*, 26, 1066-1072.

Cansdale, J.T. and Gent, R.W. 1983: Ice Accretion on Airfoils in Two-Dimensional Compressible Flow - A Theoretical Model. British Royal Aircraft Establishment Technical Report No. 82128, 31 pp. plus 33 fig.s.

Hess, J.L. and Smith, A.M.O. 1967: Calculation of Potential Flow about Arbitrary Bodies. *Progress in Aeronautical Sciences, Vol. 8.* Pergamon Press, 1-138.

Kennedy, J.L. and Marsden, D.J. 1976: Potential Flow Velocity Distributions on Multi-Component Airfoil Sections. *Can. Aero. and Space Jour.*, 22, 243-256.

MacArthur, C.D., Keller, J.L. and Luers, J.K. 1982: Mathematical Modelling of Ice Accretion on Airfoils. AIAA Publication No. 82-0284.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-005, 50 pp.

Mesinger, F. and Arakawa, A. 1976: Numerical Methods used in Atmospheric Models, Vol. 1. W.M.O.GARP Publications Series No. 17, 64 pp.

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. 1982: Numerical Simulation of Icing on Airfoils. Ph.D. Thesis, The University of Alberta, 302 pp.

Oleskiw, M.M. 1982b: Contribution to NASA/Lewis Icing Workshop, unpublished.

Stallabrass, J.R. 1958: Icing Flight Trials of a Sikorsky H04S-2 Helicopter. National Aeronautical

Establishment, Laboratory Report LR-219, 25 pp.

Stallabrass, J.R. and Lozowski, E.P. 1978: Ice Shapes on Cylinders and Rotor Blades. Presented to NATO Panel X Symposium on Helicopter Icing, London.

VI. Concluding Remarks

In the four separate, but interrelated studies of this thesis several improvements have been put forward for physical models of ice accretion. Since conclusions were given within each chapter, these remarks will summarise the results, and give suggestions for future research.

Chapter II presented a re-calculation of the long-established work of Langmuir and Blodgett on collision efficiencies and other impingement parameters of circular cylinders. High-speed digital computer results for droplet trajectory integrations have provided a small improvement over the old analog computer results, but a larger improvement has been made in the application of a non-linear regression fit to the tabulated numerical values of collision efficiency, impact speed and accretion limit. This new analytical approximation could be very useful to modellers, who presently use a variety of functions to approximate the same thing, none of which is as accurate as the new expression.

These results are equally applicable to all types of icing, wet or dry, from transmission lines to marine structures, as long as it occurs on circular cylinders. A possibility for future work here is to extend the re-calculations to the spheres and flat ribbons also considered by Langmuir and Blodgett.

The circular cylinder rime model of Chapter III makes use of these new collision efficiencies, as well as of new parameterizations for the distribution of local collision efficiency under various conditions. The efficiency and accuracy of these methods was well demonstrated in that chapter, by model results which reproduced realistic profiles, and accreted masses to within 10 percent of observation. The speed of the model; a few seconds on the University of Alberta Amdahl 580, and about 90 seconds on an IBM PC; make it highly practical.

The cylinder model also represents an improvement on older models in its version of local ice density distribution, derived in Chapter IV. Based on empirical data, this distribution is more realistic than previous versions, based on rotating cylinder correlations, have been. Since an attempt to validate the density distribution by direct measurement was inconclusive, further work, perhaps using different measurement techniques, would be valuable.

A logical extensions to the circular cylinder model would be the addition of gravitational and aerodynamic torque calculations in order to determine the correct rate of rotation. Another would be the determination of stagnation line density, following the method described in Chapter IV. The stagnation line density is required as a scaling parameter of the empirical density distribution. This would require a simple heat balance calculation at the stagnation line in order to determine the surface temperature there.

The airfoil model of Chapter V shows that parameterization methods similar to those of Chapter III may also be found for more difficult shapes, where the potential saving in computational costs is much larger. Here the parameterization for $\rho(\alpha)$ is in particular need of empirical validation, and the collision efficiency approximations would benefit from further verification and refinement using numerical results for a wider range of conditions.

The airfoil model has the capability to update the potential flow approximation, from which the β_0 parameter is derived, at intervals throughout the accretion period. Although a suggestion was made for the length of this interval, it was only an estimate. More numerical experiments should be carried out to discover the optimum interval, since this procedure is an expensive one.

Both models fail in the prediction of the troughs and ridges which sometimes form along the stagnation line of experimental accretions, often after the formation of a flat forward surface. An interesting exercise would be to investigate the flow patterns and droplet trajectories about these flat surfaces, in an attempt to understand how such features are formed. Experimental data should be gathered on the conditions under which troughs and ridges form, since they are not always present. Such studies may discover ways to model their formation.

Appendix I

Program Listing: GRAZE

```
        *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
      C ********************** GRAZE **********************
      C
      C    Karen J. Finstad      June, 1986
      C
      C    This program calculates trajectory pairs for water
      C droplets in an air flow field around a circular cylinder,
      C and determines the stagnation line collision efficiency,
      C maximum accretion angle and overall collision efficiency.
      C The air is assumed to be at a temperature of -10 degrees
      C C, and a pressure of 100 kPa.
      C
      C    The equation of motion of the droplet is integrated
      C using the Heun scheme (Mesinger, F. and Arakawa, A. 1976.
      C Numerical Methods Used in Atmospheric Models, Vol. 1.
      C W.M.O. GARP Publications Series No. 17, 64pp.) The time
      C step is set to 1/400 of the time to travel one cylinder
      C radius at the free stream speed. A maximum of 7000 steps
      C are allowed, by which time the droplet is assumed to have
      C collided with the cylinder or to have passed it.
      C
      C    Note that the droplet's initial x-coordinate is set
      C to be 10 cylinder radii upstream of the cylinder axis.
      C The initial x-component speed is set equal to the air's
      C x-component, and the y-component is set to half the
      C air's y-component.
      C
      C    A first [illegible]ectory is integrated beginning at the
      C input initi[illegible]-coordinate, which is then incre[illegible]ted
      C by a very s[illegible]ll amount (DELTAY) for the second trajectory.
      C If both trajectories intercept the cylinder, the normal
      C impact velocity, impact angle, impact position, and local
      C collision efficiency are calculated.
      C
      C    The grazing trajectory is determined by successively halving
      C the distance between a pair of initial y-coordinates which
      C bracket the initial y-coordinate of the grazing trajectory.
      C The initial values of y-min and y-max are input by the user;
      C a judicious choice will decrease the number of trajectories
      C needed to determine the grazing one.
      C    If a Stokes flow calculation is requested (i.e. Langmuir phi
      C parameter = 0), the drag coefficient is set to 24 / Reynolds
      C number. Otherwise, the formulation of Beard and Pruppacher is
      C used for the drag coefficient.
      C
      C    All operations [illegible] carried out in double precision.
      C Output is written [illegible]*MSOURCE.
      C
      C    To compile on [illegible] enter:
      C    $RUN *FORTRANVS [illegible]ARDS=GRAZE SPUNCH=GRAZERUN
      C
      C    To run, enter $RUN GRAZERUN. You will be prompted for
      C the following input parameters:
      C
```

```
        *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C       Droplet DIAMETER in microns
C
C       Cylinder RADIUS in metres
C
C       Free stream air speed in metres per second
C
C       Initial droplet y position in metres
C
C       Initial value for y-min
C
C       Initial value for y-max
C
C       y(es) or n(o) for Stokes flow option
C
C  ******************** Subprograms *********************
C
C
C       SUBROUTINE TRAJEC - calculates droplet trajectories
C         by integrating the droplet equations of motion by
C         the trapezoid rule with one iteration step.
C
C       SUBROUTINE ITERAT- calculates the air speeds (from
C         the analytical flow solution) and droplet
C         accelerations (from the equations of motion) for
C         the current iteration and time step.
C
C       SUBROUTINE COLLID -calculates final impact position
C         if the droplet collides with the surface.
C
C
C ***************** Variable Dictionary ******************
C
C       ARC1,      - arc lengths along cylinder surface
C       ARC2         from stagnation line to the impact points
C                    of the two trajectories
C
C       BETA       - local collision efficiency
C
C       CYLRAD     - cylinder radius in metres (INPUT)
C
C       DELTAT     - time step in seconds
C
C       DELTAY     - difference in initial y coordinate of
C                    the two trajectories, in metres
C
C       DROPDI     - droplet diameter, input in microns, but
C                    converted to metres (INPUT)
C
C       FINVX,     - average values of the final velocities
C       FINVY
C
C       FINVX1,    - x-components of droplet velocity at
C       FINVX2       impact
C
C       FINVY1,    - as above, for y-components
C       FINVY2
```

```
                 ....*....1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                C
                C      HALFPI     - pi / 2 (FORTRAN77 constant parameter
                C
                C      HIT1,      - logical flags for each trajectory
                C      HIT2         which indicate whether a collision
                C                   with the cylinder has occurred
                C
                C      NOTRAJ     - no. of trajectories used to find grazing traj
                C
                C      NSTEPS     - maximum number of time steps to be
                C                   integrated
                C
                C      PHI        - angle of trajectory to the surface
                C                   normal at impact
                C
                C      STK        - input character variable to set Stokes
                C                   flow flag
                C
                C      STOKES     - logical flag for Stokes flow option
                C
                C      THETA      - average of THETA1 and THETA2
                C
                C      THETA1,    - position angles of the impact points
                C      THETA2       of the two trajectories
                C
                C      VEL        - free stream air speed in metres/sec (INPUT)
                C
                C      VNORM      - droplet velocity normal to the cylinder
                C                   surface at impact
                C
                C      XINIT      - initial droplet x-coordinate in metres
                C
                C      YINIT      - initial droplet y-coordinate in metres (INPUT)
                C
                C      YMIN,      - minimum and maximum initial y-coordinates
                C      YMAX         which bracket the graze coordinate
                C
                C *******************************************************************
                C
ISN    1              DOUBLE PRECISION DROPDI,VEL,CYLRAD,XINIT,YINIT,FINVX,
                     +                 FINVY,VNORM,PHI,THETA,THETA1,THETA2,
                     +                 BETA,FINVX1,FINVX2,FINVY1,FINVY2,
                     +                 ARC1,ARC2,DELTAT,DELTAY,YMIN,YMAX
                C
ISN    2              INTEGER          NSTEPS,NOTRAJ
                C
ISN    3              LOGICAL          HIT1,HIT2,STOKES
                C
ISN    4              CHARACTER*1      RESPNS,STK
                C
ISN    5              PARAMETER (HALFPI = 3.1415926535897932/2.0)
                C
                C Prompt user for input parameters
                C
ISN    6              WRITE (6,10)
ISN    7           10 FORMAT (///,' ********* Droplet Trajectories about a',
```

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                      + ' Cylinder **********',//)
                 C
ISN    8               WRITE (6,*) 'Droplet DIAMETER in microns?'
ISN    9               READ  (5,*) DROPDI
ISN   10               DROPDI = DROPDI * 1.0E-6
                 C
ISN   11               WRITE (6,*)
ISN   12               WRITE (6,*) 'Cylinder RADIUS in metres?'
ISN   13               READ  (5,*) CYLRAD
                 C
ISN   14               WRITE (6,*)
ISN   15               WRITE (6,*) 'Free Stream air speed in m/sec?'
ISN   16               READ  (5,*) VEL
                 C
ISN   17               WRITE (6,*)
ISN   18               WRITE (6,*) 'Initial droplet y-coordinate in metres?'
ISN   19               READ  (5,*) YINIT
                 C
ISN   20               WRITE (6,*)
ISN   21               WRITE (6,*) 'Initial value of ymin?'
ISN   22               READ(5,*) YMIN
                 C
ISN   23               WRITE (6,*)
ISN   24               WRITE (6,*) 'Initial value of ymax?'
ISN   25               READ (5,*) YMAX
                 C
ISN   26               WRITE (6,*)
ISN   27               WRITE (6,*) 'Do you wish a Stokes flow calculation? y/n'
ISN   28               READ  (5,15) STK
ISN   29            15 FORMAT (A1)
                 C
                 C  Set flag for Stokes flow case
                 C
ISN   30               IF ((STK .EQ. 'y') .OR. (STK .EQ. 'Y')) THEN
ISN   31                  STOKES = .TRUE.
ISN   32               ELSE
ISN   33                  STOKES = .FALSE.
ISN   34               ENDIF
                 C
                 C  Initialize the total number of trajectory pairs.
                 C
ISN   35               NOTRAJ = 0
ISN   36           100 CONTINUE
                 C
                 C  Set maximum number of time steps , initial x-coordinate, and
                 C  the difference in initial y-coordinates  Increment number
                 C  of trajectories.
                 C
ISN   37               NOTRAJ = NOTRAJ + 1
ISN   38               NSTEPS = 7000
ISN   39               DELTAY = 1.0D-8
ISN   40               XINIT  = 10 * CYLRAD
                 C
                 C  Calculate time step
                 C
ISN   41               DELTAT = CYLRAD / (VEL * 200)
```

```
                  *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                  C
                  C  Integrate first trajectory
                  C
ISN   42                CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,NSTEPS,
                     +                  STOKES,THETA1,ARC1,HIT1,FINVX1,FINVY1)
                  C
                  C  Correct initial y position and integrate second trajectory
                  C
ISN   43                YINIT = YINIT + DELTAY
ISN   44                CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,NSTEPS,
                     +                  STOKES,THETA2,ARC2,HIT2,FINVX2,FINVY2)
                  C
                  C  Test for result of trajectory (miss, graze or collision)
                  C
ISN   45                IF ((HIT1) .AND. (HIT2)) THEN
                  C
                  C        Calculate impact parameters and local collision efficiency
                  C
ISN   46                   IF (YINIT-DELTAY .NE. 0.0) YMIN = YINIT - DELTAY
ISN   48                   BETA  = DELTAY / (ARC2 - ARC1)
ISN   49                   THETA = (THETA1 + THETA2) / 2.0
ISN   50                   FINVX = (FINVX1 + FINVX2) / 2.0
ISN   51                   FINVY = (FINVY1 + FINVY2) / 2.0
ISN   52                   PHI   = HALFPI - DATAN(FINVX/FINVY) + THETA
ISN   53                   VNORM = DSQRT((FINVX * FINVX) + (FINVY * FINVY))
                     +               * DCOS(PHI)
                  C
                  C  Write results to screen
                  C
ISN   54                   WRITE (6,20)THETA,VNORM,PHI,BETA
ISN   55          20       FORMAT (//,' Both trajectories collided with cylinder.',
                     +              //,' Position angle at impact point: ',F7.3
                     +              //,' Velocity normal to surface at impact: ',F7.3,
                     +              //,' Angle of trajectory to surface normal: ',F7.3
                     +              //,' Local collision efficiency: ',F5.3,//)
ISN   56                ELSE IF (HIT1) THEN
ISN   57                   WRITE (6,30) THETA1
ISN   58          30       FORMAT (//,' First trajectory grazed cylinder.',
                     +              //,' Surface angle of max accretion point: ',F7.3,//)
ISN   59                WRITE (6,*) ' overall coll. eff.:', (YINIT - DELTAY)/CYLRAD
ISN   60                WRITE (6,*) ' no of trajectories: ',NOTRAJ
ISN   61                   GO TO 200
ISN   62                ELSE
ISN   63                   YMAX = YINIT -DELTAY
ISN   64                   WRITE (6,40)
ISN   65          40       FORMAT (//,' Both trajectories missed the cylinder.')
ISN   66                ENDIF
                  C
                  C  Determine new initial position and start next trajectory pair
                  C
ISN   67                YINIT = (YMAX + YMIN) / 2.0
ISN   68                GO TO 100
                  C
ISN   69          200   STOP
ISN   70                END
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986   TI[illegible]3:13:40      NAME: MAIN     PAGE:    6
*STATISTICS*   SOURCE STATEMENTS = 69, PROGRAM SIZE = 2732 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED        NOTEST   SEQ
                    OPTIMIZE(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                   *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                   C
                   C *************************************************************
                   C *    SUBROUTINE TRAJEC                                         *
                   C *                                                              *
                   C *       Integrates trajectory of droplet whose initial         *
                   C *    position is specified in the parameter list. Calls        *
                   C *    ITERAT for iterative calculations, and COLLID to          *
                   C *    determine impact parameters for colliding trajectories.   *
                   C *                                                              *
                   C *    INPUT VARIABLES:                                          *
                   C *                 XINIT  - initial x position                  *
                   C *                 YINIT  - initial Y position                  *
                   C *                 DROPDI - droplet diameter                    *
                   C *                 CYLRAD - cylinder radius                     *
                   C *                 VEL    - free stream air speed               *
                   C *                 DELTAT - time step                           *
                   C *                 NSTEPS - maximum number of time steps        *
                   C *                 STOKES - logical flag for Stokes flow        *
                   C *                                                              *
                   C *    OUTPUT VARIABLES:                                         *
                   C *                 THETA  - surface angle of impact             *
                   C *                 ARC    - arc length along cylinder surface   *
                   C *                          from stagnation line to impact      *
                   C *                          point                               *
                   C *                 HIT    - logical flag set to .TRUE. if       *
                   C *                          a collision occurs                  *
                   C *                 FINVX  - droplet x velocity on impact        *
                   C *                 FINVY  - droplet y velocity on impact        *
                   C *                                                              *
                   C *    OTHER VARIABLES:                                          *
                   C *                 X,Y    - arrays containing x,y positions     *
                   C *                          for each time step                  *
                   C *                 VX,VY  - arrays containimg x,y components    *
                   C *                          of droplet speed for each step      *
                   C *                 DVX,   - arrays containing x,y components    *
                   C *                 DVY      of droplet accelerations            *
                   C *                                                              *
                   C*************************************************************
                   C
ISN   1                  SUBROUTINE TRAJEC(XINIT,YINIT,DROPDI,VEL,CYLRAD,DELTAT,
                        +                  NSTEPS,STOKES,THETA,ARC,HIT,FINVX,FINVY)
                   C
ISN   2                  DOUBLE PRECISION X(7000),Y(7000),
                        +                 VX(7000),VY(7000),DVX(7000),DVY(7000)
                   C
ISN   3                  DOUBLE PRECISION XINIT,YINIT,DROPDI,CYLRAD,VEL,DELTAT,
                        +                 THETA,ARC,FINVX,FINVY
                   C
ISN   4                  INTEGER          NSTEPS
                   C
ISN   5                  LOGICAL          HIT,STOKES
                   C
                   C  Initialize values for integration
```

138

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C
ISN    6               X(1)  = -XINIT
ISN    7               Y(1)  = YINIT
ISN    8               VY(1) = (DABS(2 * VEL * CYLRAD * CYLRAD * X(1) * Y(1)) /
                     +         (((X(1) * X(1)) + (Y(1) * Y(1)))**2)) / 2.0
ISN    9               VX(1) = VEL
ISN   10               I     = 1
ISN   11               HIT   = .FALSE.
                 C
                 C  Begin integration
                 C
ISN   12          100  CONTINUE
                 C
                 C  Calculate current step air speeds and accelerations
                 C
ISN   13                  CALL ITERAT(X(I),Y(I),VX(I),VY(I),VEL,CYLRAD,DROPDI,
                     +                 STOKES,DVX(I),DVY(I))
                 C
                 C  Calculate next step droplet speeds and positions
                 C
ISN   14                  VX(I+1) = VX(I) - (DELTAT * DVX(I))
ISN   15                  VY(I+1) = VY(I) + (DELTAT * DVY(I))
ISN   16                  X(I+1)  = X(I) + (VX(I) * DELTAT) - (0.5 * DVX(I)
                     +                * (DELTAT * DELTAT))
ISN   17                  Y(I+1)  = Y(I) + (VY(I) * DELTAT) + (0.5 * DVY(I)
                     +                * (DELTAT * DELTAT))
                 C
                 C  Calculate next step air speeds and accelerations
                 C
ISN   18                  CALL ITERAT(X(I+1),Y(I+1),VX(I+1),VY(I+1),VEL,
                     +                 CYLRAD,DROPDI,STOKES,DVX(I+1),DVY(I+1))
                 C
                 C  Iterate calculations of next step droplet speed and
                 C  position using averaged values of accelerations
                 C
ISN   19                  VX(I+1) = VX(I) - ((DELTAT / 2.0) * (DVX(I) +
                     +               DVX(I+1)))
ISN   20                  VY(I+1) = VY(I) + ((DELTAT / 2.0) * (DVY(I) +
                     +               DVY(I+1)))
ISN   21                  X(I+1) =  X(I) + (VX(I) * DELTAT) - (0.25 *
                     +               (DVX(I) + DVX(I+1)) * (DELTAT * DELTAT))
ISN   22                  Y(I+1) =  Y(I) + (VY(I) * DELTAT) + (0.25 *
                     +               (DVY(I) + DVY(I+1)) * (DELTAT * DELTAT))
                 C
                 C  Test whether trajectory has intercepted the cylinder
                 C
ISN   23                  IF (DSQRT((X(I+1)**2) + (Y(I+1)**2)) .LE. CYLRAD) THEN
ISN   24                     HIT   = .TRUE.
ISN   25                     FINVX = (VX(I) + VX(I+1)) / 2.0
ISN   26                     FINVY = (VY(I) + VY(I+1)) / 2.0
ISN   27                     CALL COLLID(X(I),X(I+1),Y(I),Y(I+1),CYLRAD,
                     +                   THETA,ARC)
ISN   28                  ELSE IF (I .LT. (NSTEPS - 1)) THEN
ISN   29                     I = I + 1
ISN   30                     GO TO 100
ISN   31                  ENDIF
```

```
                       *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 386.    C
ISN      32      387.          RETURN
ISN      33      388.          END
```

STATISTICS   SOURCE STATEMENTS = 33, PROGRAM SIZE = 338354 BYTES, PROGRAM NAME = TRAJEC   PAGE: 7.

STATISTICS   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 2 ******

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED   NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C ***************************************************************
                 C *                                                             *
                 C *  SUBROUTINE ITERAT  -  calculates for the current step:     *
                 C *      air speed components from analytical solution to       *
                 C *      flow around a cylinder, the droplet Reynolds           *
                 C *      number, drag coefficient according to the formula-     *
                 C *      tion of Beard and Pruppacher, and the droplet          *
                 C *      acceleration components from the equations of motion.*
                 C *                                                             *
                 C *  INPUT VARIABLES :                                          *
                 C *                 X,Y     - current droplet positions         *
                 C *                 VX,VY,  - current droplet speeds            *
                 C *                 VEL     - free stream air speed             *
                 C *                 CYLRAD  - cylinder radius                   *
                 C *                 DROPDI  - droplet diameter                  *
                 C *                 STOKES  - logical flag for Stokes flow      *
                 C *                                                             *
                 C *  OUTPUT VARIABLES :                                         *
                 C *                 DVX,DVY - current droplet accelerations     *
                 C *                                                             *
                 C *  OTHER VARIABLES :                                          *
                 C *                 UX,UY    - current air speed components     *
                 C *                 DROPRE   - droplet Reynolds number          *
                 C *                 DRAGCO   - drag coefficient                 *
                 C *                 COEFF    - combined coefficients of the     *
                 C *                            equations of motion              *
                 C *                 DENOM    - denominator of air speed         *
                 C *                            equations                        *
                 C *                                                             *
                 C ***************************************************************
                 C
ISN    1                SUBROUTINE ITERAT(X,Y,VX,VY,VEL,CYLRAD,DROPDI,STOKES,DVX,DVY)
                 C
ISN    2                DOUBLE PRECISION X,Y,VX,VY,DVX,DVY,VEL,CYLRAD,DROPDI,
                      +                  DROPRE,DRAGCO,COEFF,UX,UY,DENOM
                 C
ISN    3                LOGICAL          STOKES
                 C
                 C  Calculate air speeds
                 C
ISN    4                DENOM = ((X * X) + (Y * Y)) ** 2
ISN    5                UX    = VEL * (1 + ((CYLRAD**2) * (((Y * Y) - (X * X))/DENOM)))
ISN    6                UY    = DABS(2 * VEL * (CYLRAD**2) * ((X * Y) / DENOM))
                 C
                 C Calculate Reynolds number and drag coefficient
                 C
ISN    7                DROPRE    = DROPDI * DSQRT(((VX - UX) ** 2) + ((VY - UY)
                      +             ** 2)) / 1.262E-5
ISN    8                IF (STOKES) THEN
ISN    9                   DRAGCO = 24.0 / DROPRE
ISN   10                ELSE IF (DROPRE .LT. 2.0) THEN
ISN   11                   DRAGCO = (24.0 / DROPRE) + (2.45 / (DROPRE ** 0.045))
ISN   12                ELSE IF (DROPRE .LT. 21.0) THEN
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: AUG 07, 1986     TIME: 10:44:54      NAME: ITERAT  PAGE:   11

                                *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN      13      442.                 DRAGCO = (24.0 / DROPRE) + (2.76 / (DROPRE ** 0.198))
ISN      14      443.              ELSE IF (DROPRE .GT. 200.0) THEN
ISN      15      444.                 DRAGCO = (24.0 / DROPRE) + (4.73 / (DROPRE ** 0.37))
                 445.          +                + (6.24E-3 * (DROPRE ** 0.38))
ISN      16      446.              ELSE
ISN      17      447.                 DRAGCO = (24.0 / DROPRE) + (4.536 / (DROPRE ** 0.368))
ISN      18      448.              ENDIF
                 449.     C
                 450.     C  Calculate accelerations
                 453.     C
ISN      19      454.              COEFF = 9.92E-4 * (DRAGCO / DROPDI)
ISN      20      455.              DVX    = COEFF * DSQRT(((VX - UX) **2 ) + ((VY - UY) ** 2))
                 456.          +            * (VX - UX)
ISN      21      457.              DVY    = -COEFF * DSQRT(((VX - UX) ** 2) + ((VY - UY) ** 2))
                 458.          +            * (VY - UY)
                 459.     C
ISN      22      460.              RETURN
ISN      23      461.              END

*STATISTICS*    SOURCE STATEMENTS = 23, PROGRAM SIZE = 1380 BYTES, PROGRAM NAME = ITERAT    PAGE:   10.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                C ***************************************************************
                C *                                                             *
                C *  SUBROUTINE COLLID   -  Calculates the x,y position         *
                C *    of impact on the cylinder surface, position angle        *
                C *    of the impact point, and arc length measured from       *
                C *    the stagnation line.                                     *
                C *                                                             *
                C *  INPUT VARIABLES:                                           *
                C *                XOUT,  - last droplet position outside      *
                C *                YOUT     the cylinder                        *
                C *                XIN,   - first droplet position inside       *
                C *                YIN      the cylinder                        *
                C *                CYLRAD - cylinder radius                     *
                C *                                                             *
                C *  OUTPUT VARIABLES :                                         *
                C *                THETA  - position angle of impact point      *
                C *                         on the cylinder measured from       *
                C *                         the stagnation line                 *
                C *                ARC    - arc length from stagnation line     *
                C *                         to the impact point                 *
                C *                                                             *
                C *  OTHER VARIABLES :                                          *
                C *                SLOPE  - slope of the trajectory segment     *
                C *                         which crosses the cylinder surface  *
                C *                         i.e., between XOUT,YOUT and         *
                C *                         XIN,YIN                             *
                C *                A,B,C  - coefficients of the quadratic       *
                C *                         equation to solve for the intercept *
                C *                         of the trajectory and cylinder      *
                C *                         surface                             *
                C *                IMPX,  - impact position on the surface      *
                C *                IMPY                                         *
                C *                                                             *
                C ***************************************************************
                C
ISN    1              SUBROUTINE COLLID (XOUT,XIN,YOUT,YIN,CYLRAD,THETA,ARC)
                C
ISN    2              DOUBLE PRECISION XOUT,XIN,YOUT,YIN,CYLRAD,THETA,ARC,
                    +                  SLOPE,A,B,C,IMPX,IMPY
                C
                C  Calculate slope of trajectory at impact point
                C
ISN    3              SLOPE = (YIN - YOUT) / (XIN - XOUT)
ISN    4              XOUT  = XOUT / CYLRAD
ISN    5              XIN   = XIN / CYLRAD
ISN    6              YOUT  = YOUT / CYLRAD
ISN    7              YIN   = YIN / CYLRAD
                C
                C  Find impact point by solving quadratic equation with
                C  coefficients as follows
                C
ISN    8              A     = (SLOPE * SLOPE) + 1.0
ISN    9              B     = (2.0 * SLOPE * YOUT) - (2.0 * (SLOPE * SLOPE) * XOUT)
```

```
                          *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.......*
ISN     10      515.            C     = -(2.0 * SLOPE * XOUT * YOUT) + ((SLOPE * SLOPE) *
                516.          +         (XOUT * XOUT)) + (YOUT * YOUT) - 1.0
ISN     11      517             IMPX = (-B - DSQRT((B * B) - (4 * A * C))) / (2 * A)
ISN     12      518.            IMPY = (SLOPE * (IMPX - XOUT)) + YOUT
                519.      C
                520.      C  Calculate impact point angle and arc length
                521.      C
ISN     13      522.            THETA = DATAN (IMPY / DABS(IMPX))
ISN     14      523.            ARC   = CYLRAD * THETA
                524.      C
ISN     15      525.            RETURN
ISN     16      526.            END
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 840 BYTES, PROGRAM NAME = COLLID    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
```

```
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS

*STATISTICS*   SOURCE STATEMENTS = 69, PROGRAM SIZE = 2732 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 33, PROGRAM SIZE = 338354 BYTES, PROGRAM NAME = TRAJEC    PAGE:    7.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 23, PROGRAM SIZE = 1380 BYTES, PROGRAM NAME = ITERAT    PAGE:   10.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 840 BYTES, PROGRAM NAME = COLLID    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY CODE IS 0.
```

145

Appendix II

Program Listing: OMNICYL

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C        *************** Cylinder Rime Icing Model **************
C
C        Karen J. Finstad      June, 1986
C
C        This program will calculate and plot rime ice accretion
C     layers on a cylinder, as well as calculate the mass and
C     mean density of each layer. Local collision efficiencies
C     for each point on the surface are calculated as a cosine
C     function of the current local surface angle
C     and scaled to the value of stagnation line collision
C     efficiency (Beta zero), and maximum accretion angle
C     (Alpha max).
C
C        Two forms of this function are used for droplet diameters
C     above and below 20 microns. Beta zero and Alpha max are
C     estimated from a non-linear regression fit to data computed
C     by droplet trajectory program GRAZE.
C
C       Two optional procedures are available for treating
C     the effects of droplet size spectra, when the assumption
C     of a monodisperse population at the median-volume
C     diameter (mvd) becomes inaccurate (roughly, when the
C     mvd is less than 50 micron, for ground level wind speeds):
C          1. the user may enter weighting factors for each
C     of 18 size bins of width 10 microns, or
C          2. use a "simulated spectrum" curve for the local
C     collision efficiency (l.c.e.), which differs principally
C     from the cosine approximation by the addition of a "tail"
C     onto the maximum accretion limit. The length of this
C     tail is a function of the mvd. Again, there is a
C     slightly different form of the curve for mvd above and
C     below 20 microns.
C       The first of these options is recommended only when
C     three or less layers are to be accreted, since the weighted
C     averaging procedure eventually produces first-order
C     discontinuities in the l.c.e. curve, and thus on the
C     modelled ice surface.
C
C        Local density is calculated from an empirically-
C     derived function of local surface slope.
C
C       The cylinder may be rotated, clockwise, after each
C     layer is accreted. The rotation increment is included
C     in the input parameter list. When this option is
C     chosen, (i.e., if the input rotation increment is greater
C     than zero), the l.c.e. and local density are calculated
C     as a function of position angle, instead of surface angle.
C     This somewhat less accurate method is necessary to avoid
C     problems when accreting over top of the previous layers'
C     accretion limit.
C        Also please note that rotation of more than 90 degrees
C     in total is not possible in this version.
C
```

```
C          The output consists of a plot description file written into
C     -pdf attached to unit 8, and a mass and density table written
C     into -mass attached to unit 9. (These are temporary files.)
C
C          The plot description file is produced using the system
C     subroutine library *PLOTLIB. All of the plotting is done
C     in a single program subroutine, so that other available
C     plotting routines may be substituted if desired.
C          Please note: in order to run *PLOTLIB subroutines (or any
C     subroutines which have not been compiled in Fortran77 and
C     which contain character strings or variables in their
C     argument list), in conjunction with this program, (or any
C     Fortran77 program) the command @PROCESS(list of subroutines)
C     MUST appear in ROW 1, COLUMN 1 if the subroutines are
C     called directly from MAIN. If they are called from
C     another subroutine, the @PROCESS command must be placed
C     in COLUMN 1 OF THE LINE IMMEDIATELY FOLLOWING THE END
C     STATEMENT OF THE PREVIOUS ROUTINE.
C
C         This code will also run on the IBM Professional Fortran
C     compiler for PC's and PC compatibles. A math co-processor
C     and graphics capability is required. (Graphics routines
C     must be supplied). A 6-layer accretion with rotation
C     takes about 90 seconds on a standard PC. Storage needed
C     is about 370,000 bytes.
C
C          To compile (on MTS), first empty -mass and -pdf. Then
C     $RUN *FORTRANVS SCARDS=OMNIC SPUNCH=CYLRUN.
C
C         To run, enter $RUN CYLRUN+*PLOTLIB. You will be prompted
C     for the following input parameters:
C
C          median volume droplet DIAMETER in microns
C
C          y or n to enter full spectrum, followed by weights,
C            entered individually
C
C          y or n to simulated spectrum option
C
C          cylinder RADIUS in metres
C
C          free stream air speed in metres per second
C
C          liquid water content in kg per cubic metre
C
C          time step per layer in seconds
C
C          total number of layers
C
C          rotation increment in degrees
C
C
C     *********************** Subprograms ***********************
C
C
C     INTERNAL:
```

```
     *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C
C        SUBROUTINE DRAW - Plots cylinder profile and all
C      accreted ice layers with appropriate legend.
C
C        DOUBLE PRECISION FUNCTION MONBET - Estimates local
C      collision efficiency as a function of surface slope for
C      a monodisperse droplet population at the mvd. The
C      function has two forms for use with mvd smaller or
C      larger than than 20 microns.
C
C        SUBROUTINE PARAM - Estimates the stagnation line
C      collision-efficiency and maximum accretion angle
C      for bare cylinders at atmospheric pressure of 100 kPa
C      from a non-linear regression fit to data derived from
C      droplet trajectory calculations.
C
C        SUBROUTINE ROTAT1 - When the rotation increment
C      is non-zero, this subroutine calculates the array
C      of position angles for the current surface, to replace
C      surface angles in the array ALPHA.
C
C        SUBROUTINE ROTAT2 - shifts the array indices of
C      the current LAYRX and LAYRY arrays to face the new
C      free stream direction, which has been rotated
C      according to the rotation increment.
C
C        DOUBLE PRECISION FUNCTION SPCBET - Estimates local
C      collision efficiency as a function of surface slope
C      which simulates the effect of a typical droplet size
C      spectrum centred on the mvd. The function has two forms
C      for use with mvd smaller or larger than 20 microns.
C
C        SUBROUTINE WTBET - Calculates weighted average of
C      local collision efficiency estimates from FUNCTION
C      MONBET, for droplet spectrum weights in 18 10-micron
C      size bins.
C
C
C     EXTERNAL:
C
C       *PLOTLIB  -  The subroutines PLOTS, METRIC, PLOT, AXIS2,
C     LINE, SYMBOL and NUMBER (called from SUBROUTINE DRAW) are
C     found in the public system subroutine library *PLOTLIB.
C     (See note above regarding external subroutines in Fortran77).
C
C     ****************** Variable Dictionary ********************
C
C       ALPHA        - array of angles between surface normal
C                      and the free stream direction
C
C       ALPHM        - Alpha max, the maximum impingement angle
C
C       AM           - array of maximum impingement angles for
C                      each size category of the droplet size
C                      spectrum
C
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C       AREA          - cross sectional area between
C                       adjacent points in the current layer
C                       (dimensioned in metres)
C
C       BETA          - local collision effiency
C
C       BZERO         - stagnation line value of the local collision
C                       efficiency
C
C       BZ            - array of Bzero values for each size
C                       category of the droplet size spectrum
C
C       CYLRAD        - radius of cylinder in metres (INPUT)
C
C       CYLX,CYLY     - arrays containing non - dimensional x and
C                       y coordinates of the cylinder surface
C
C       DI            - array of 18 10-micron wide droplet
C                       diameter size bins
C
C       DIAG          - cross diagonal of the quadrilateral
C                       area used in mass calculations
C
C       DROPDI        - droplet diameter in microns (INPUT)
C
C       HALFPI        - pi/2 (Fortran77 constant PARAMETER)
C
C       LAREA         - cross sectional area of the current layer
C                       in sq. metres
C
C       LAYERS        - integer number of layers
C
C       LAYRX,LAYRY   - array containing non - dimensional x and y
C                       coordinates of the accretion surfaces for
C                       up to 30 accretion layers
C
C       LDENS         - mean density of the current layer in kg/cubic m
C
C       LMASS         - mass of the current layer in kg per unit length
C
C       LWC           - liquid water content in kg per cubic metre
C                       (INPUT)
C
C       L1,L2,L3      - dimensioned sides of the quadrilateral area
C       L4              defined by two adjacent points on the
C                       current surface and two on the surface below
C
C       MASS          - array of masses per unit length for areas
C                       defined by adjacent points, in kg
C
C       PI            - pi (Fortran77 constant PARAMETER)
C
C       R             - temporary variable for storing real result
C                       during conversion of the rotation increment
C                       an integer number of surface points
C
```

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                C     RHO          - array of local densities in kg/cubic metre
                C
                C     ROTN         - no. of surface points in a rotation increment
                C
                C     RSPEED       - rotation increment per layer in radians (INPUT)
                C
                C     SIMSPC       - logical variable, true if simulated spectrum
                C                    option is chosen
                C
                C     SIMOPT       - character variable for reading in response to
                C                    simulated spectrum option query (INPUT)
                C
                C     SLOPE        - array of local surface slopes calculated
                C                    at each surface point
                C
                C     SMDROP       - logical flag for droplet size, which determines
                C                    function used to calculate local c.e.
                C
                C     SPCOPT       - character variable for reading in response
                C                    to spectrum weighting option query (INPUT)
                C
                C     SPEC         - logical variable, true if spectrum weighting
                C                    option is chosen
                C
                C     S1,S2        - quantities used to determine the quadrilaterals
                C                    for the layer area calculation
                C
                C     TAILEN       - filter length for optional smoothing filter
                C                    applied to c. e. curve
                C
                C     TAU          - accretion time step in seconds (INPUT)
                C
                C     THICK        - array of ice thicknesses normal to surface
                C                    (non - dimensional)
                C
                C     TMASS        - total mass of the accretion
                C
                C     VEL          - free stream air speed in m per sec (INPUT)
                C
                C     WGHT         - array of droplet spectrum volume percentages
                C                    for each size bin (INPUT)
                C
                C
                C ****************************************************************
                C
ISN      1            DOUBLE PRECISION ALPHA(600),CYLX(1202),CYLY(1202),
                     +                 LAYRX(30,600),LAYRY(30,600),
                     +                 RHO(600),SLOPE(600),THICK(600),
                     +                 DI(18),WGHT(18),BZ(18),AM(18)
                C
ISN      2            DOUBLE PRECISION ALPHM,AREA,BETA,BZERO,CYLRAD,DIAG,
                     +                 DROPDI,LAREA,LDENS,LMASS,LWC,L1,L2,
                     +                 L3,L4,MASS,RSPEED,S1,S2,TAILEN,TAU,
                     +                 TMASS,VEL
                C
ISN      3            REAL             R
```

```
                 *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C
ISN    4               INTEGER             LAYERS,ROTN
                 C
ISN    5               CHARACTER*1         SIMOPT,SPCOPT
                 C
ISN    6               LOGICAL             SIMSPC,SMDROP,SPEC
                 C
ISN    7               PARAMETER (PI = 3.1415926535897932,HALFPI = PI/2)
                 C
ISN    8               COMMON CYLX,CYLY,LAYRX,LAYRY
                 C
                 C  Fill array of droplet diameter bins
                 C
ISN    9                  DATA (DI(I),I=1,18)/5.E-6,15.E-6,25.E-6,35.E-6,45.E-6,
                      +             55.E-6,65.E-6,75.E-6,85.E-6,95.E-6,105.E-6,
                      +             115.E-6,125.E-6,135.E-6,145.E-6,155.E-6,
                      +             165.E-6,175.E-6/
                 C
                 C  Prepare mass table and plot description files.
                 C
ISN   10               OPEN (UNIT = 8, FILE = '-MASS', STATUS = 'OLD')
ISN   11               OPEN (UNIT = 9, FILE = '-PDF', STATUS = 'OLD')
                 C
                 C  Set up headings for mass and density table in -MASS.
                 C
ISN   12               WRITE (8,110)
ISN   13           110 FORMAT (///,15X,' Mass of Ice Accreted',//,5X,' Layer no.',
                      +         7X,' Mass',7X,' Mean Density',//,
                      +         20X,' (kg/m)',9X,' (kg/cu.m)',//,
                      +         ' ***********************************************')
                 C
                 C  Write program title and intro to screen.
                 C
ISN   14               WRITE (6,120)
ISN   15           120 FORMAT (/////,' *************  CYLINDER RIME ICING MODEL  *****',
                      +         ' ***********',//,20X,' Version Feb. 17, 1986',//,
                      +         22X,' K. J. Finstad',/,19X,
                      +         ' Meteorology Division',/,15X,' The University of Alberta'
                      +         ,//,' Please enter parameters to begin accretion: ',//)
                 C
                 C  Prompt user for input parameters.
                 C
ISN   16               WRITE (6,*) 'Median Volume Droplet DIAMETER in microns? '
ISN   17               READ  (5,*) DROPDI
                 C
                 C  Set droplet size flag
                 C
ISN   18               SMDROP = .FALSE.
ISN   19               IF (DROPDI .LT. 20) SMDROP = .TRUE.
ISN   21               DROPDI = DROPDI*1.0E-6
                 C
                 C  Set flags for requested treatment of droplet spectrum.
                 C  If full spectrum weighting is requested, input volume
                 C  weights for each size bin.
                 C
ISN   22               WRITE (6,*)
```

```
                   *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN      23             WRITE (6,*) 'Do you wish to enter full droplet spectrum? '
ISN      24             READ  (5,9) SPCOPT
ISN      25           9 FORMAT (1A1)
ISN      26             SPEC = .FALSE.
ISN      27             IF ((SPCOPT .EQ. 'y') .OR. (SPCOPT .EQ. 'Y')) THEN
ISN      28                SPEC = .TRUE.
ISN      29                WRITE (6,*)
ISN      30                WRITE (6,*) 'Enter weight for the following size ranges:'
ISN      31                DO 13 I=1,18
ISN      32                   WRITE (6,*)
ISN      33                   WRITE (6,*) DI(I)/1.0E-6, 'microns'
ISN      34                   READ (5,*) WGHT(I)
ISN      35          13    CONTINUE
ISN      36             ELSE
ISN      37                WRITE (6,*)
ISN      38                WRITE (6,*) 'Do you wish simulated spectrum weighting? '
ISN      39                READ  (5,8) SIMOPT
ISN      40           8    FORMAT (1A1)
ISN      41             ENDIF
                  C
                  C For simulated spectrum, calculate 'tail' length from
                  C median volume diameter.
                  C
ISN      42             IF ((SIMOPT .EQ. 'Y') .OR. (SIMOPT .EQ. 'y')) THEN
ISN      43               SIMSPC = .TRUE.
ISN      44               IF (SMDROP) THEN
ISN      45                  TAILEN = 0.52 * DDEXP(-DROPDI / 7.0E-5)
ISN      46               ELSE
ISN      47                  TAILEN = 0.37 * DDEXP(-DROPDI / 7.0E-5)
ISN      48               ENDIF
ISN      49             ELSE
ISN      50               SIMSPC = .FALSE.
ISN      51               TAILEN = 0.0
ISN      52             ENDIF
                  C
ISN      53             WRITE (6,*)
ISN      54             WRITE (6,*) 'Cylinder RADIUS in metres? '
ISN      55             READ  (5,*) CYLRAD
                  C
ISN      56             WRITE (6,*)
ISN      57             WRITE (6,*) 'Free stream speed in m / sec? '
ISN      58             READ  (5,*) VEL
                  C
ISN      59             WRITE (6,*)
ISN      60             WRITE (6,*) 'Liquid water content in kg / cubic metre? '
ISN      61             READ  (5,*) LWC
                  C
ISN      62             WRITE (6,*)
ISN      63             WRITE (6,*) 'Time step per layer in seconds? '
ISN      64             READ  (5,*)  TAU
                  C
ISN      65             WRITE (6,*)
ISN      66             WRITE (6,*) 'Total number of layers? '
ISN      67             READ  (5,*) LAYERS
                  C
                  C  Bare cylinder will be layer 1, so add 1 to LAYERS
```

```
                  *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                  C
ISN    68               LAYERS = LAYERS + 1
                  C
ISN    69               WRITE (6,*)
ISN    70               WRITE (6,*) 'Rotation speed in degrees per layer? '
ISN    71               READ  (5,*) RSPEED
                  C
                  C  Convert rotation increment to radians, and calculate
                  C  approx. number of surface points in rotation increment
                  C
ISN    72               RSPEED = (RSPEED * 2.0 * PI) / 360.0
ISN    73               R      = (RSPEED / 5.235987E-3)
ISN    74               ROTN   = NINT(R)
                  C
                  C
                  C  Call Subroutine Param to calculate Beta zero and alpha max
                  C  for the mvd. If spectrum weighting option has been chosen,
                  C  call Param for each size bin.
                  C
ISN    75               CALL PARAM (DROPDI,VEL,CYLRAD,BZERO,ALPHM)
ISN    76               IF (SPEC) THEN
ISN    77                  DO 12 I=1,18
ISN    78                     IF (WGHT(I) .NE. 0.0) THEN
ISN    79                        CALL PARAM(DI(I),VEL,CYLRAD,BZ(I),AM(I))
ISN    80                     ELSE
ISN    81                        BZ(I) = 0.0
ISN    82                        AM(I) = 0.0
ISN    83                     ENDIF
                  C
                  C  Make sure accretion limit is revised to reflect the
                  C  largest droplets present.
                  C
ISN    84                     IF (AM(I) .GT. AM(I-1)) ALPHM = AM(I)
ISN    86          12      CONTINUE
ISN    87               ENDIF
                  C
                  C  Fill array of points defining the initial  surface of the
                  C  cylinder in non-dimensional coordinates. The complete
                  C  surface contains 1200 points spaced equally in x.
                  C
                  C
ISN    88               DO 100 I=601,1200
ISN    89                  CYLX(I) = (DBLE(I-601) * 3.33333E-3)
ISN    90                  CYLY(I) = DSQRT(1.0 - ((1.0 - CYLX(I))*(1.0 - CYLX(I))))
ISN    91         100   CONTINUE
ISN    92               DO 150 I = 1,600
ISN    93                  CYLX(I) = 2.0 - (DBLE(I) * 3.33333E-3)
ISN    94                  CYLY(I) = -DSQRT(1. - ((1. - CYLX(I)) * (1. - CYLX(I))))
ISN    95         150   CONTINUE
                  C
                  C  Copy front half of  surface to array which will contain
                  C  all of the layers.
                  C
ISN    96               DO 200 I = 1,600
ISN    97                 LAYRX(1,I) = CYLX(299 + I)
ISN    98                 LAYRY(1,I) = CYLY(299 + I)
```

```
                    *....*...1.........2.........3.........4.........5.........6.........7.*........8.........9.........*
ISN   99            200  CONTINUE
                    C
                    C  Initialize total mass and begin loop to accrete ice layers.
                    C
ISN  100                 TMASS = 0.0
ISN  101                 DO 300 J = 2,LAYERS
                    C
                    C  Calculate slope and surface angle from finite differences for
                    C  each point on current surface. If rotation increment is
                    C  non-zero, call subroutine to calculate slopes and
                    C  position angles instead.
                    C
ISN  102                 IF (ROTN .GT. 0) THEN
ISN  103                   CALL ROTAT1(J,RSPEED,SLOPE,ALPHA)
ISN  104                 ELSE
ISN  105                   DO 400 I = 2,599
ISN  106                     SLOPE(I) = (LAYRX(J-1,I+1) - LAYRX(J-1,I-1)) /
                         +               DABS(LAYRY(J-1,I+1) - LAYRY(J-1,I-1))
ISN  107                     IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN  108                       ALPHA(I) = DABS(PI - DATAN(SLOPE(I)))
ISN  109                     ELSE
ISN  110                       ALPHA(I) = DABS(DATAN(SLOPE(I)))
ISN  111                     ENDIF
ISN  112            400    CONTINUE
ISN  113                 ENDIF
                    C
                    C  Calculate for each point local density, collision efficiency and
                    C  ice thickness. The method of calculation for Beta depends on
                    C  the values of SMDROP, SPEC and SIMSPC.
                    C
ISN  114                   DO 500 I = 2,599
ISN  115                     RHO(I) = 890. *
                         +            (1.0 - .143 * (ALPHA(I)/(ALPHM+TAILEN)) -
                         +             .246 * ((ALPHA(I)/(ALPHM+TAILEN)) ** 2) -
                         +             .309 * ((ALPHA(I)/(ALPHM+TAILEN)) ** 3))
                    C
                    C  Make sure density does not become too small
                    C
ISN  116                     IF (RHO(I) .LT. 175.) RHO(I) = 175.
                    C
                    C  Calculate local collision efficiency
                    C
ISN  118                     IF (SPEC) THEN
ISN  119                       CALL WTBET(ALPHA(I),SMDROP,BZ,AM,WGHT,BETA)
ISN  120                     ELSE IF (SIMSPC) THEN
ISN  121                       BETA = SPCBET(ALPHA(I),SMDROP,BZERO,ALPHM,TAILEN)
ISN  122                     ELSE
ISN  123                       BETA = MONBET(ALPHA(I),SMDROP,BZERO,ALPHM)
ISN  124                     ENDIF
                    C
                    C  Calculate local ice thickness perpendicular to surface
                    C  (in non-dimensional units)
                    C
ISN  125                     THICK(I)   = BETA * VEL * TAU * LWC / (RHO(I) * CYLRAD)
                    C
                    C  Define new layer surface coordinates.
```

```
                  *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                  C
ISN   126                     IF (LAYRY(J-1,I) .LT. LAYRY(J-1,I-1)) THEN
ISN   127                        LAYRX(J,I) = LAYRX(J-1,I) + (THICK(I) *
                       +                      DCOS(DATAN(SLOPE(I))))
ISN   128                     ELSE
ISN   129                        LAYRX(J,I) = LAYRX(J-1,I) - (THICK(I) *
                       +                      DCOS(DATAN(SLOPE(I))))
ISN   130                     ENDIF
ISN   131                     LAYRY(J,I) = LAYRY(J-1,I) + (THICK(I) *
                       +                   DSIN(DATAN(SLOPE(I))))
ISN   132              500    CONTINUE
ISN   133                  LAYRX(J,1)   = LAYRX(J-1,1)
ISN   134                  LAYRX(J,600) = LAYRX(J-1,600)
ISN   135                  LAYRY(J,1)   = LAYRY(J-1,1)
ISN   136                  LAYRY(J,600) = LAYRY(J-1,600)
                  C
                  C  Calculate mass and mean density of the layer from
                  C  the sum of areas between surface points on successive
                  C  layers. First, initialize layer mass and area.
                  C
ISN   137                   LMASS = 0.0
ISN   138                   LAREA = 0.0
ISN   139                   DO 560 I = 2,598
ISN   140                      IF ((THICK(I) .GT. 0.0).AND.(THICK(I+1) .GT. 0.0)) THEN
ISN   141                         L4 = THICK(I) * CYLRAD
ISN   142                         L2 = THICK(I+1) * CYLRAD
ISN   143                         L1 = DSQRT((LAYRX(J-1,I+1) - LAYRX(J-1,I))**2 + (LAYRY
                       +                 (J-1,I+1) - LAYRY(J-1,I))**2) * CYLRAD
ISN   144                         L3 = DSQRT((LAYRX(J,I+1) - LAYRX(J,I))**2 + (LAYRY
                       +                 (J,I+1) - LAYRY(J,I))**2) * CYLRAD
ISN   145                         DIAG = DSQRT((LAYRX(J,I+1)-LAYRX(J-1,I))**2 + (LAYRY
                       +                   (J,I+1) - LAYRY(J-1,I))**2) * CYLRAD
ISN   146                         S1   = 0.5 * (DIAG+L1+L2)
ISN   147                         S2   = 0.5 * (DIAG+L3+L4)
ISN   148                         AREA = DSQRT(S1 * (S1-DIAG) * (S1-L1) * (S1-L2)) +
                       +                     DSQRT(S2 * (S2-DIAG) * (S2-L3) * (S2-L4))
ISN   149                         MASS = ((RHO(I) + RHO(I+1)) / 2.0) * AREA
ISN   150                         LMASS   = LMASS + MASS
ISN   151                         LAREA   = LAREA + AREA
ISN   152                      ENDIF
ISN   153              560    CONTINUE
                  C
                  C  Calculate mean density.
                  C
ISN   154                   LDENS = LMASS/LAREA
                  C
                  C  Write results to mass table.
                  C
ISN   155                   WRITE (8,570) J-1,LMASS,LDENS
ISN   156              570  FORMAT (//,8X,I2,7X,F9.5,7X,F9.5)
                  C
                  C  Add mass of current layer to total mass.
                  C
ISN   157                   TMASS = TMASS + LMASS
                  C
                  C  Call subroutine to rotate array indices if required.
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:27       NAME: MAIN     PAGE:    11

                                  *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                       556.     C
ISN      158           557.           IF (ROTN .GT. O) CALL ROTAT2(J,ROTN)
                       558.     C
ISN      160           559.       300  CONTINUE
                       560.     C
                       561.     C  Write total mass accreted to table
                       562.     C
ISN      161           563.            WRITE (8,590) TMASS
ISN      162           564.       590  FORMAT (8X,//,' Total mass accreted per metre:',F9.5)
                       565.     C
                       566.     C  Call subroutine to plot cylinder and accreted layers
                       567.     C
ISN      163           568.            CALL DRAW(ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,LWC,RSPEED,
                       569.          +           SIMSPC,SPEC,TAILEN,TAU,VEL)
ISN      164           570.       999  STOP
ISN      165           571.            END

*STATISTICS*   SOURCE STATEMENTS = 161, PROGRAM SIZE = 27174 BYTES, PROGRAM NAME = MAIN    PAGE:     1.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 1 ******
```

REQUESTED OPTIONS (PROCESS): SC(AXIS2,SYMBOL)

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED   NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C
                 *****************************************************************
                 *   SUBROUTINE DRAW - to plot cylinder and all accretion          *
                 *                     layers, with approriate information         *
                 *                     in the legend. This subroutine uses         *
                 *                     the plotting library *PLOTLIB               *
                 *                                                                 *
                 *   Input Variables : as defined for Main program ;               *
                 *                     ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,LWC,       *
                 *                     RSPEED,SIMSPC,SPEC,TAILEN,TAU,VEL           *
                 *                                                                 *
                 *   Common Variables : CYLX,CYLY,LAYRX,LAYRY                      *
                 *                                                                 *
                 *   Other Variables : PLOTX,PLOTY - arrays containing x,y         *
                 *                     plotting data and scaling information       *
                 *                     for accretion layer currently being         *
                 *                     plotted                                     *
                 *                                                                 *
                 *****************************************************************
                 C
ISN    1               SUBROUTINE DRAW(ALPHM,BZERO,CYLRAD,DROPDI,LAYERS,
                     +                 LWC,RSPEED,SIMSPC,SPEC,TAILEN,TAU,VEL)
                 C
ISN    2               DOUBLE PRECISION CYLX(1202),CYLY(1202),LAYRX(30,600),
                     +                  LAYRY(30,600),PLOTX(602),PLOTY(602)
                 C
ISN    3               DOUBLE PRECISION ALPHM,BZERO,CYLRAD,DROPDI,LWC,RSPEED,
                     +                  TAILEN,TAU,VEL
                 C
ISN    4               INTEGER          LAYERS
                 C
ISN    5               LOGICAL          SIMSPC,SPEC
                 C
ISN    6               COMMON CYLX,CYLY,LAYRX,LAYRY
                 C
                 C  To plot cylinder surface, form lower surface array,
                 C  and place scaling data at the end of coordinate arrays.
                 C
ISN    7               CYLX(1201) = -6.0
ISN    8               CYLX(1202) =  1.0
ISN    9               CYLY(1201) = -3.0
ISN   10               CYLY(1202) =  1.0
ISN   11               PLOTX(601) = -6.0
ISN   12               PLOTX(602) =  1.0
ISN   13               PLOTY(601) = -3.0
ISN   14               PLOTY(602) =  1.0
                 C
                 C  Open plotting library
                 C
ISN   15               CALL PLOTS
ISN   16               CALL METRIC(0)
```

```
                 *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C
                 C  Draw 6 by 9 inch box
                 C
ISN     17             CALL PLOT(2.0,2.0,-3)
ISN     18             CALL PLOT(0.0,6.0,3)
ISN     19             CALL PLOT(9.0,6.0,2)
ISN     20             CALL PLOT(9.0,0.0,2)
                 C
                 C  Label axes and draw tics
                 C
ISN     21             CALL AXIS2(0.0,0.0,'X/CYLRAD',-8,9.0,0.0,-6.0,1.0,1.0)
ISN     22             CALL AXIS2(0.0,0.0,'Y/CYLRAD',8,6.0,90.0,-3.0,1.0,0.5)
                 C
                 C  Write legends to top left corner
                 C
ISN     23             CALL SYMBOL(1.1,5.4,0.10,'DROPLET MVD  = ',0.0,15)
ISN     24             CALL NUMBER(2.6,5.4,0.10,DROPDI,0.0,07)
ISN     25             CALL SYMBOL(3.5,5.4,0.10,'M',0.0,01)
ISN     26             CALL SYMBOL(1.1,5.2,0.10,'CYL RADIUS   = ',0.0,15)
ISN     27             CALL NUMBER(2.6,5.2,0.10,CYLRAD,0.0,04)
ISN     28             CALL SYMBOL(3.5,5.2,0.10,'M',0.0,01)
ISN     29             CALL SYMBOL(1.1,5.0,0.10,'AIR SPEED    = ',0.0,15)
ISN     30             CALL NUMBER(2.6,5.0,0.10,VEL,0.0,02)
ISN     31             CALL SYMBOL(3.5,5.0,0.10,'M / SEC',0.0,07)
ISN     32             CALL SYMBOL(1.1,4.8,0.10,'LWC          = ',0.0,15)
ISN     33             CALL NUMBER(2.6,4.8,0.10,LWC,0.0,05)
ISN     34             CALL SYMBOL(3.5,4.8,0.10,'KG / M**3',0.0,09)
ISN     35             CALL SYMBOL(1.1,4.6,0.10,'TIME STEP    = ',0.0,15)
ISN     36             CALL NUMBER(2.6,4.6,0.10,TAU,0.0,02)
ISN     37             CALL SYMBOL(3.5,4.6,0.10,'SEC',0.0,03)
ISN     38             CALL SYMBOL(1.1,4.4,0.10,'BETA ZERO    = ',0.0,15)
ISN     39             CALL NUMBER(2.6,4.4,0.10,BZERO,0.0,03)
ISN     40             CALL SYMBOL(1.1,4.2,0.10,'ALPHA MAX    = ',0.0,15)
ISN     41             CALL NUMBER(2.6,4.2,0.10,(ALPHM+TAILEN)*360./6.283,0.0,01)
ISN     42             CALL SYMBOL(3.5,4.2,0.10,'DEGREES',0.0,07)
ISN     43             CALL SYMBOL(1.1,4.0,0.10,'ROTN INCR    = ',0.0,15)
ISN     44             CALL NUMBER(2.6,4.0,0.10,RSPEED*360./6.283,0.0,01)
ISN     45             CALL SYMBOL(3.5,4.0,0.10,'DEGREES',0.0,07)
ISN     46             CALL SYMBOL(1.1,3.8,0.10,'SPECTRUM:',0.0,09)
ISN     47             IF (SPEC) THEN
ISN     48                CALL SYMBOL(2.6,3.8,0.10,'YES',0.0,03)
ISN     49             ELSE IF (SIMSPC) THEN
ISN     50                CALL SYMBOL(2.6,3.8,0.10,'SIMULATED',0.0,09)
ISN     51             ELSE
ISN     52                CALL SYMBOL(2.6,3.8,0.10,'NO',0.0,02)
ISN     53             ENDIF
                 C
                 C  Draw initial cylinder surface
                 C
ISN     54             CALL LINE(CYLX,CYLY,4200,2,0)
                 C
                 C  Transfer each layer in turn to the array holding
                 C  the scaling parameters
                 C
ISN     55             DO 610 J = 2,LAYERS
ISN     56                DO 700 L = 1,600
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:29      NAME: DRAW    PAGE:   14
                                 *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN     57      680.                    PLOTX(L) =  LAYRX(J,L)
ISN     58      681.                    PLOTY(L) =  LAYRY(J,L)
ISN     59      682.          700     CONTINUE
                683.       C
                684.       C  Draw accretion layer
                685.       C
ISN     60      686.                  CALL LINE(PLOTX,PLOTY,600,2,0)
ISN     61      687.          610  CONTINUE
                688.       C
                689.       C  Close plotting library
                690.       C
ISN     62      691.               CALL PLOT(1.0,1.0,999)
ISN     63      692.               RETURN
ISN     64      693.               END

*STATISTICS*   SOURCE STATEMENTS = 64, PROGRAM SIZE = 12132 BYTES, PROGRAM NAME = DRAW    PAGE:   12.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** [illegible] OF COMPILATION 2 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK  SOURCE  TERM  OBJECT FIXED      NOTEST  SEQ
                      OPTIMIZE(Q)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                           *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                           C *************************************************************
                           C *  FUNCTION MONBET - Calculates local c.e.                  *
                           C *   estimate for monodisperse droplet populations          *
                           C *   at mvd either above or below 20 microns                *
                           C *   using a cosine curve scaled to the stagnation          *
                           C *   line c. e. and the maximum accretion angle.            *
                           C *                                                          *
                           C *  Input Variables.: as defined for main program:          *
                           C *                    ALPHM,BZERO,ALPHA,SMDROP               *
                           C *                                                          *
                           C *************************************************************
                           C
ISN       1                      DOUBLE PRECISION FUNCTION MONBET(ALPHA,SMDROP,BZERO,ALPHM)
                           C
ISN       2                      DOUBLE PRECISION BZERO,ALPHM,ALPHA
                           C
ISN       3                      LOGICAL          SMDROP
                           C
ISN       4                      PARAMETER        (HALFPI = 3.1415926535897932/2.)
                           C
ISN       5                      IF (ALPHA .GE. ALPHM) THEN
ISN       6                         MONBET = 0.0
ISN       7                      ELSE IF (SMDROP) THEN
ISN       8                         MONBET = BZERO + (BZERO * (DSQRT(DCOS(ALPHA)) - 1.0)) /
                                +              (1.0 - DSQRT(DCOS(ALPHM)))
ISN       9                      ELSE
ISN      10                         MONBET = BZERO * (DCOS((ALPHA/ALPHM) * HALFPI))
ISN      11                      ENDIF
ISN      12                      RETURN
ISN      13                      END

*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 586 BYTES, PROGRAM NAME = MONBET    PAGE:   15

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

```
                 *....*..1.......2.......3.......4.......5.......6.......7.*.......8.......9.......*
                 C
                 C****************************************************************
                 C *  Subroutine PARAM - Estimates Beta zero and alpha max       *
                 C *            from the input parameters according to           *
                 C *            an empirical fit to droplet trajectories         *
                 C *            calculated with program [illegible]ISION.        *
                 C *                                                             *
                 C *  Input variables ; as defined for Main program:            *
                 C *                      CYLRAD,DROPDI,VEL                       *
                 C *                                                             *
                 C *  Output variables : as defined for Main program:           *
                 C *                      ALPHM,BZERO                             *
                 C *                                                             *
                 C *  Other variables :                                         *
                 C *          RE        - free stream Reynolds number           *
                 C *          K         - Langmuir inertia parameter            *
                 C *          PHI       - Langmuir parameter                    *
                 C *          TERM1,    - terms used in the fitting function    *
                 C *          TERM2,                                            *
                 C *          TERM3                                             *
                 C *          A(2,12)   - array containing constants for        *
                 C *                        the fitting functions               *
                 C *                                                             *
                 C****************************************************************
                 C
ISN    1               SUBROUTINE PARAM (DROPDI,VEL,CYLRAD,BZERO,ALPHM)
                 C
ISN    2               DOUBLE PRECISION ALPHM,BZERO,CYLRAD,DROPDI,K,
                      +                 PHI,RE,TERM1,TERM2,TERM3,VEL
                 C
ISN    3               DOUBLE PRECISION A(2,12)
                 C
ISN    4               DATA (A(1,I), I=1,12)/1.215,-7.42D-3,-0.561,-0.644,-0.164,
                      +           3.05D-3,0.430,2.22,-0.45,-0.767,-0.806,-0.068/
ISN    5               DATA (A(2,I), I=1,12)/2.327,-4.09D-3,-0.388,-0.593,-0.689,
                      +           8.5D-3,0.383,1.757,-0.298,-0.42,-0.96,-0.179/
                 C
                 C  Calculate free stream Reynolds number, phi and K assuming
                 C  water density is 1000, air density 1.323 and air
                 C  viscosity 1.669 in MKS units. These correspond to
                 C  an air temperature of -10 C and pressure 100 kPa.
                 C
                 C
ISN    6               RE = DROPDI * VEL / 1.262E-5
ISN    7               K  = (VEL * DROPDI * DROPDI * 3.329E6) / CYLRAD
ISN    8               PHI = (RE * RE) / K
                 C
                 C  Calculate Beta zero and alpha max
                 C
ISN    9               DO 10 I = 1,2
ISN   10                  TERM1 = (A(I,1) * (K ** A(I,2))*DEXP(A(I,3)*K**A(I,4)))
                      +              + A(I,5)
ISN   11                  TERM2 = A(I,6) *(PHI ** A(I,7))
```

```
LEVEL 1.1.1 (DEC  81)           VS FORTRAN        DATE: JUL 24, 1986    TIME: 13:15:30      NAME: PARAM   PAGE:    17
                                      *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN      12        777.                  TERM3 = (A(I,8) * (K ** A(I,9))*DEXP(A(I,10)*K**A(I,11)))
                   778.              +        + A(I,12)
ISN      13        779.                  IF (I .EQ. 1) THEN
ISN      14        780.                     BZERO = TERM1 - (TERM2 * TERM3)
ISN      15        781.                  ELSE
ISN      16        782.                     ALPHM = TERM1 - (TERM2 * TERM3)
ISN      17        783.                  ENDIF
ISN      18        784.            10  CONTINUE
                   785.       C
                   786.       C  Check for extreme values of K
                   787.       C
ISN      19        788.               IF (K .LT. 0.17) THEN
ISN      20        789.                  ALPHM = 0.0
ISN      21        790.                  BZERO = 0.0
ISN      22        791.               ELSE IF (K .GT. 1000.) THEN
ISN      23        792.                  ALPHM = 1.56
ISN      24        793.                  BZERO = 0.99
ISN      25        794.               ENDIF
ISN      26        795.               RETURN
ISN      27        796.               END

*STATISTICS*   SOURCE STATEMENTS = 27, PROGRAM SIZE = 1362 BYTES, PROGRAM NAME = PARAM    PAGE:   16.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 4 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                *....1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C
                 C ***********************************************************************
                 C *  SUBROUTINE ROTAT1 - calculates position angles and                 *
                 C *                      slopes for each point on the                   *
                 C *                      current surface. Position angles               *
                 C *                      are with respect to the current                *
                 C *                      free stream direction.                         *
                 C *                                                                     *
                 C *  Input Variables : as defined for Main program;                     *
                 C *                    J,RSPEED                                         *
                 C *                                                                     *
                 C *  Output Variables :                                                 *
                 C *                    ALPHA - array of position angles                 *
                 C *                            measures w.r.t. center of                *
                 C *                            cylinder and free stream                 *
                 C *                    as defined for Main program;                     *
                 C *                    SLOPE                                            *
                 C *                                                                     *
                 C *  Common Variables : CYLX,CYLY,LAYRX,LAYRY                           *
                 C *                                                                     *
                 C *                                                                     *
                 C ***********************************************************************
                 C
ISN    1                SUBROUTINE ROTAT1(J,RSPEED,SLOPE,ALPHA)
                 C
ISN    2                DOUBLE PRECISION ALPHA(600),CYLX(1202),CYLY(1202),
                     +                   LAYRX(30,600),LAYRY(30,600),SLOPE(600)
                 C
ISN    3                DOUBLE PRECISION RSPEED
                 C
ISN    4                INTEGER          J
                 C
ISN    5                PARAMETER        (PI = 3.14159265358979332,HALFPI=PI/2.)
                 C
ISN    6                COMMON CYLX,CYLY,LAYRX,LAYRY
                 C
                 C  Calculate slopes
                 C
ISN    7                DO 590 I=2,599
ISN    8                   IF (LAYRY(J-1,I+1) .EQ. LAYRY(J-1,I-1)) THEN
ISN    9                      SLOPE(I) = 1.E10
ISN   10                   ELSE
ISN   11                      SLOPE(I) = (LAYRX(J-1,I+1)-LAYRX(J-1,I-1)) / DABS
                     +                   (LAYRY(J-1,I+1)-LAYRY(J-1,I-1))
ISN   12                   ENDIF
                 C
                 C  Smooth over any discontinuities in slope caused
                 C  by accreting over the previous layer's accretion
                 C  limit.
                 C
ISN   13                   IF (LAYRY(J-1,I-1) .GT. 0.0) THEN
ISN   14                      IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN   15                         IF (SLOPE(I) .LT. SLOPE(I-1)) SLOPE(I) = SLOPE(I-1)
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:31      NAME: ROTAT1  PAGE:   19
                              *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN     17      850.                   ENDIF
ISN     18      851.                ENDIF
                852.      C
                853.      C  Calculate position angles.
                854.      C
ISN     19      855.                IF (LAYRX(J-1,I) .EQ. 1.0) THEN
ISN     20      856.                   ALPHA(I) = DABS(HALFPI - ((J-2) * RSPEED))
ISN     21      857.                ELSE IF (LAYRX(J-1,I) .GT. 1.0) THEN
ISN     22      858.                   ALPHA(I) = HALFPI - ((J-2)*RSPEED) +
                859.          +                  DATAN((LAYRX(J-1,I) - 1.) / LAYRY(J-1,I))
ISN     23      860.                ELSE
ISN     24      861.                   IF (DABS(LAYRY(J-1,I)) .LT. 1.0D-10) LAYRY(J-1,I) = 0.0
ISN     26      862.                   IF (DABS(LAYRX(J-1,I)) .LT. 1.0D-10) LAYRX(J-1,I) = 0.0
ISN     28      863.                   ALPHA(I) = DABS(DATAN(LAYRY(J-1,I) /
                864.          +                  (1.0 - LAYRX(J-1,I))) - ((J-2)*RSPEED))
ISN     29      865.                   ENDIF
ISN     30      866.        590  CONTINUE
                867.      C
ISN     31      868.              RETURN
ISN     32      869.              END

*STATISTICS*   SOURCE STATEMENTS = 29, PROGRAM SIZE = 1998 BYTES, PROGRAM NAME = ROTAT1    PAGE:   18.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 5 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:31      NAME: MAIN     PAGE:   20

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                       OPTIMIZE(0)  LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN  ),  LINECOUNT(60)

                             *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                             C ************************************************************
                             C * SUBROUTINE ROTAT2 - shifts array indices of current layer*
                             C *                     by an amount corresponding to the     *
                             C *                     rotation increment. Points from the   *
                             C *                     cylinder array must be added to one   *
                             C *                     end.                                  *
                             C *                                                           *
                             C * Input variables  :  as defined in main program:           *
                             C *                     J,ROTN                                *
                             C *                                                           *
                             C * Common variables : CYLX,CYLY,LAYRX,LAYRY                  *
                             C *                                                           *
                             C * Other variables  :  K,M - index limits for shifting       *
                             C *                                                           *
                             C ************************************************************
                             C
ISN      1                         SUBROUTINE ROTAT2(J,ROTN)
                             C
ISN      2                         DOUBLE PRECISION CYLX(1202),CYLY(1202),LAYRX(30,600),
                                  .                LAYRY(30,600)
                             C
ISN      3                         INTEGER J,ROTN
                             C
ISN      4                         COMMON CYLX,CYLY,LAYRX,LAYRY
                             C
                             C  Renumber layer array to face new direction of air stream,
                             C  dropping increment number of points at the 'front' end.
                             C
ISN      5                         K = 600-ROTN
ISN      6                         M = 600-ROTN+1
ISN      7                         DO 580 I = 1,K
ISN      8                            LAYRX(J,I) = LAYRX(J,I+ROTN)
ISN      9                            LAYRY(J,I) = LAYRY(J,I+ROTN)
ISN     10                     580 CONTINUE
                             C
                             C  Add increment number of points from the bare cylinder to
                             C  the 'back' end.
                             C
ISN     11                         DO 585 I = M,600
ISN     12                            LAYRX(J,I) = CYLX(299 + (J-1)*ROTN + I)
ISN     13                            LAYRY(J,I) = CYLY(299 + (J-1)*ROTN + I)
ISN     14                     585 CONTINUE
ISN     15                         RETURN
ISN     16                         END

*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 780 BYTES, PROGRAM NAME = ROTAT2     PAGE:   20.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 6 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED       NOTEST   SEQ
                    OPTIMIZE(O)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN)  LINECOUNT(60)

                    *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                    C
                    C *************************************************************
                    C *  Function SPCBET - Estimates Beta(alpha) for a                *
                    C *           simulated droplet size distribution, for an         *
                    C *           mvd either smaller or larger than 20 microns.       *
                    C *           The polynomial function is scaled to (alpha         *
                    C *           max + tail length), where the tail length           *
                    C *           is a function of the mvd.                           *
                    C *                                                               *
                    C * Input Variables : as defined for Main program;               *
                    C *                   ALPHA,ALPHM,BZERO,SMDROP,TAILEN             *
                    C *                                                               *
                    C *  Other Variables : X - scaled angle = Alpha /                 *
                    C *                        (Alphm + Tailen)                       *
                    C *                                                               *
                    C *************************************************************
                    C
ISN      1                DOUBLE PRECISION FUNCTION SPCBET(ALPHA,SMDROP,BZERO,
                         +                                 ALPHM,TAILEN)
                    C
ISN      2                DOUBLE PRECISION ALPHA,ALPHM,BZERO,TAILEN,X
                    C
ISN      3                LOGICAL      SMDROP
                    C
ISN      4                IF (ALPHA .GE. (ALPHM+TAILEN)) THEN
ISN      5                  SPCBET = 0.0
ISN      6                ELSE
ISN      7                  X = ALPHA / (ALPHM+TAILEN)
ISN      8                  IF (SMDROP) THEN
ISN      9                    SPCBET = BZERO * (1.0 + 0.2[illegible]5 * X - 3.084 *
                         +             (X ** 2) + 1.866 * (X ** 3))
ISN     10                  ELSE
ISN     11                    SPCBET = BZERO * (1.0 + 0.0287 * X - 1.936 *
                         +             (X ** 2) + 2.484 * (X ** 3) - 4.112 *
                         +             (X ** 4) + 2.538 * (X ** 5))
ISN     12                  ENDIF
ISN     13                  IF (SPCBET .LT. 0.0) SPCBET = 0.0
ISN     15                ENDIF
ISN     16                RETURN
ISN     17                END
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 708 BYTES, PROGRAM NAME = SPCBET   PAGE:   21
*STATISTICS*   NO DIAGNOSTICS GENERATED
****** END OF COMPILATION 7 ******
```

```
LEVEL 1.1.1 (DEC    81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:32      NAME: MAIN     PAGE:   22

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED          NOTEST   SEQ
                    OPTIMIZE(O)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                    *....1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                954.   C
                955.   C ************************************************************
                956.   C * SUBROUTINE WTBET - Calculates the average local           *
                957.   C *                    collision efficency according to      *
                958.   C *                    the input volume weighted droplet     *
                959.   C *                    size spectrum.                         *
                960.   C *                                                           *
                961.   C * Input Variables : as defined for Main program;            *
                962.   C *                   ALPHA,AM,BZ,SMDROP,WGHT                 *
                963.   C *                                                           *
                964.   C * Output Varibles : as defined for Main program;            *
                965.   C *                   BETA                                    *
                966.   C *                                                           *
                967.   C ************************************************************
                968.   C
ISN      1      969.         SUBROUTINE WTBET(ALPHA,SMDROP,BZ,AM,WGHT,BETA)
                970.   C
ISN      2      971.         DOUBLE PRECISION AM(18),BZ(18),WGHT(18)
                972.   C
ISN      3      973.         DOUBLE PRECISION ALPHA,BETA
                974.   C
ISN      4      975.         LOGICAL       SMDROP
                976.   C
ISN      5      977.               BETA = 0.0
ISN      6      978.               DO 22 L=1,18
ISN      7      979.                  IF (WGHT(L) .NE. 0.0) THEN
ISN      8      980.                     BETA = BETA + WGHT(L) * MONBET(
                981.     +                      ALPHA(I),BZ(L),AM(L),SMDROP)
ISN      9      982.                  ENDIF
ISN     10      983.          22   CONTINUE
ISN     11      984.         RETURN
ISN     12      985.         END

*STATISTICS*   SOURCE STATEMENTS = 12, PROGRAM SIZE = 654 BYTES, PROGRAM NAME = WTBET    PAGE:   22.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 8 ******
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:32     NAME: MAIN    PAGE:   23
SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS
*STATISTICS*   SOURCE STATEMENTS = 161, PROGRAM SIZE = 27174 BYTES, PROGRAM NAME = MAIN    PAGE:    1.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 64, PROGRAM SIZE = 12132 BYTES, PROGRAM NAME = DRAW    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 586 BYTES, PROGRAM NAME = MONBET    PAGE:   15.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 27, PROGRAM SIZE = 1362 BYTES, PROGRAM NAME = PARAM    PAGE:   16.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
*STATISTICS*   SOURCE STATEMENTS = 29, PROGRAM SIZE = 1998 BYTES, PROGRAM NAME = ROTAT1    PAGE:   18.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 5 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 780 BYTES, PROGRAM NAME = ROTAT2    PAGE:   20.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 6 ******
*STATISTICS*   SOURCE STATEMENTS = 16, PROGRAM SIZE = 708 BYTES, PROGRAM NAME = SPCBET    PAGE:   21.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 7 ******
*STATISTICS*   SOURCE STATEMENTS = 12, PROGRAM SIZE = 654 BYTES, PROGRAM NAME = WTBET    PAGE:   22.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 8 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED  HIGHEST SEVERITY CODE IS 0
```

Appendix III

Approximation of the mvd from Stagnation Line Thickness

The measurement of water droplet sizes and size distributions is one of the major experimental problems in icing studies. The very common measurement technique, known as the oil slide method, has been shown by Makkonen and Stallabrass (1984) to overestimate the median volume diameters of observed droplet spectra in comparison to measurements made with an optical sizing probe, the Forward Scattering Spectrometer Probe, or FSSP, manufactured by Particle Measuring Systems and described by Knollenberg (1981). Makkonen and Stallabrass provide an empirical correction formula for oil slide measured mvds in the range 10 to 40 μm, as follows:

$$mvd_{FSSP} = 8.8 + 0.27\ mvd_{oil\ slide} \qquad A.1$$

All of the wind tunnel experiments performed in the FROST tunnel and described in this thesis have had droplet size distributions determined by the oil slide method. The mvds derived from this data appear to be too large, not only in light of Makkonen and Stallabrass' results, but also because the theoretical collection efficiencies calculated for these mvds imply a minimum stagnation line thickness much larger than that observed.

In order to supply the most accurate input data possible to the model, these oil slide measurement should be corrected. However, some of the circular cylinder cases lie outside of the range given for Equation A.1. For these cases an approximate mvd has been derived from the observed stagnation line thicknesses of accreted ice, using the method outlined below, similar to the rotating cylinder methods which have been used for the same purpose.

For cases which lie inside the range of validity, these approximate mvds agree well (within about 10 per cent) with the results of Equation A.1. The correlation of Makkonen and Stallabrass has therefore been used for all cases within the appropriate range.

Outside of that range, however, it is important to remember that the approximation given here is derived independently of the actual size spectrum, and represents a droplet size corresponding to the average value of β_0 for the time period of the accretion layer. It has been used here only because of the absence of a reliable calibration for oil slide measurements of droplets larger than 40 μm.

The method is as follows:

1. The stagnation line thickness (Th) of the chosen layer is measured from a photograph of the accretion in cross-section.
2. A first estimate of β_0 is calculated from applying

the dry growth equation to the stagnation line:

$$\beta_0 = (\rho_0 \, Th) / (U \, lwc \, \tau) \qquad A.2$$

A best estimate of ρ_0 may be made as in Chapter IV. For lwc, the first guess is made using the oil slide mvd, and the measured rate of mass collected on a small rotating cylinder, and calculated by the algorithm given by Stallabrass (1978).

3. Calculate the Stokes parameter:

$$\phi = 9 \, \rho_a^{\,2} \, D_c \, U / (\mu \, \rho_d); \qquad A.3$$

then read from Figure II.3, a value of K corresponding to ϕ and β_0.

4. Derive a new value of mvd from the definition of K:

$$K = (\rho_d / 18 \, \mu) \, (mvd^2 \, U / D_c) \qquad A.4$$

5. Recalculate lwc using the new mvd.

6. Recalculate β_0 from new lwc. Repeat steps 2 through 6 until the values of β_0, mvd and lwc appear to have converged.

For the examples given below, the rotating cylinder diameter was 2.46 mm.

Example 1:

oil slide mvd = 24.2 μm

Th = 1.49 mm

U = 10.7 m/sec

τ = 600 sec

ρ_0 = 750 kg/m^3

rotating cylinder mass = 0.029 g/min

revised mvd = 13.8 μm

mvd from Equation A.1 = 15.3 μm

Example 2:

oil slide mvd = 136 μm

Th = 13.5 mm

U = 10.24 m/sec

τ = 1800 sec

ρ_0 = 890 kg/m^3

rotating cylinder mass = 0.122 g/min

revised mvd = 62 μm

References

Knollenberg, R.G. 1981: Techniques for Probing Cloud Microstructure. Clouds, Their Formation, Optical Properties, and Effects, ed.s P.V. Hobbs and A. Deepak. Academic Press, 15-92.

Makkonen, L. and Stallabrass, J.R. 1984: Ice Accretion on Cylinders and Wires. National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-005, 50 pp.

Stallabrass, J.R. 1978: National Research Council, Division of Mechanical Engineering, Technical Report TR-LT-92, 26 pp., 7 fig.s, + 2 App.s.

Appendix IV

A Geometrical Derivation of Equation V.3

Figure A.1 illustrates two droplet trajectories impinging on an airfoil, at zero angle of attack. If the initial separation of the trajectories is Δy, then the distance separating them (in the direction normal to the lower trajectory) will have increased at the point of impact to, say, $(\Delta y + a)$. Referring to Figure A.1, if the angle γ is the departure from horizontal of the tangent to the upper trajectory at the point X, it follows that for small $\Delta \ell$:

$$\cos(\alpha + \gamma) \approx (\Delta y + a) \,/\, \Delta \ell = \beta + (a \,/\, \Delta \ell), \qquad \text{A.5}$$

using the definition of β given in Chapter II.

For airfoils, γ is always $<< \alpha$, so:

$$\cos(\alpha + \gamma) \approx \cos \alpha \qquad \text{A.6}$$

Now if it is assumed that $(a \,/\, \Delta \ell)$ is approximately a constant, and applying the boundary condition at $\alpha = 0$, this gives:

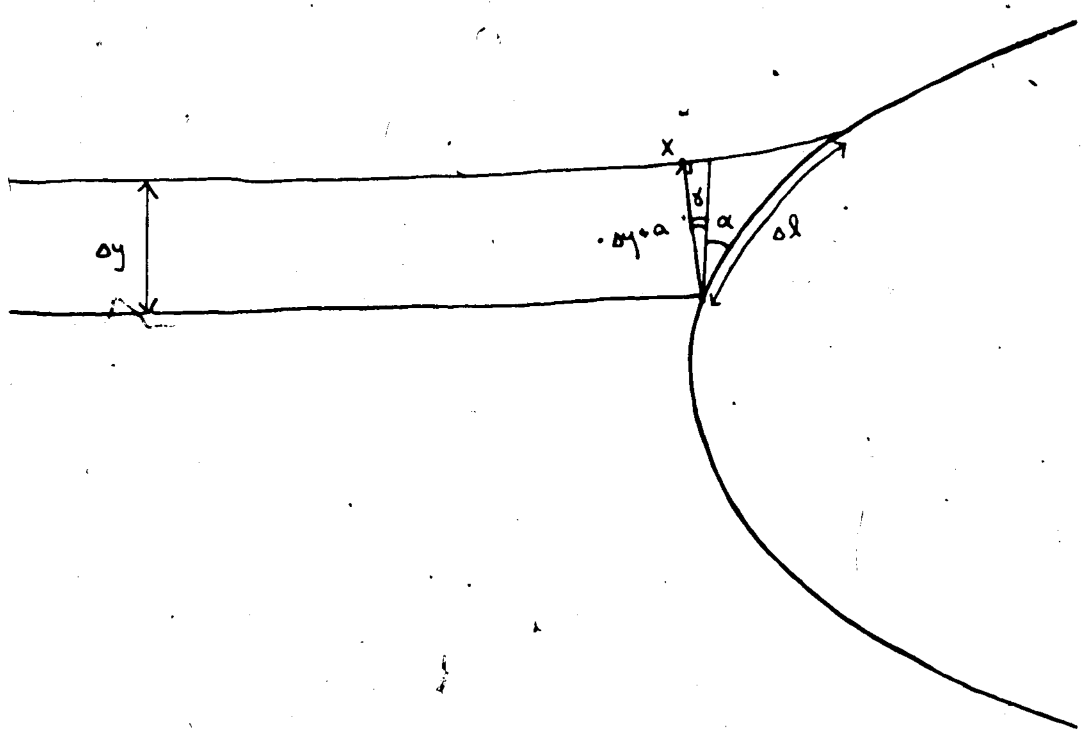


Figure A.1 - Droplet trajectories impinging on an airfoil.

$$\beta \approx \cos \alpha - (1 - \beta_0), \qquad \text{A.7}$$

which is Equation V.3. The above approximation should then hold when the impinging trajectories have small curvature.

Appendix V

Program Listing: OMNIFOIL

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(O) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C       ***************  Airfoil Rime Icing Model **************
C
C       Karen J. Finstad       June, 1986
C
C         This program will calculate and plot rime ice accretion
C       layers on an airfoil, as well as calculate the mass and
C       mean density of each layer. Local collision efficiencies
C       for each point on the surface are calculated as a cosine
C       function of the current local surface angle
C       and scaled to the value of stagnation line collision
C       efficiency (Beta zero), which is determined from droplet
C       trajectory calculations.
C
C         Beta zero is updated on every fourth layer to account
C       for the changing flow around the accretion. Layer time
C       steps should be chosen so that Beta zero is updated at
C       reasonable intervals, say after a maximum thickness of no
C       more than 1/10 of the chord length has been accreted.
C
C         The maximum accretion angle, alpha max, is determined
C       from Beta zero.
C
C         When the angle of attack is zero, a "simulated spectrum"
C       version of local collision efficiency distribution may be
C       used to approximate the effects of droplet size spectra
C       with median volume diameters less than about 20 microns.
C       Larger drops are well represented by the monodisperse
C       version of collision efficiency parameterisation at the mvd.
C
C         For other angles of attack, an approximate "skewed"
C       distribution is used, which should apply to all droplet
C       sizes.
C
C         Local density is approximated by a polynomial
C       function of the local surface slope.
C
C         The output consists of a plot description file written into
C       -pdf attached to unit 8, and a mass and density table written
C       into -mass attached to unit 9. (A negative sign identifies
C       temporary files on MTS.)
C
C         The plot description file is produced using the system
C       subroutine library *PLOTLIB. All of the plotting is done
C       in a single program subroutine, so that other available
C       plotting routines may be substituted if desired.
C
C         An additional external subroutine , from the *IMSLDPLIB
C       library, is used in the solution of the potential flow.
C       Other matrix inversion routines may be substituted.
C
C         Please note: in order to run *PLOTLIB subroutines (or any
C       subroutines which have not been compiled in Fortran77 and
C       which contain character strings or variables in their
```

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C     argument list), in conjunction with this program, (or any
C     Fortran77 program) the command @PROCESS(list of subroutines)
C     MUST appear in ROW 1, COLUMN 1 if the subroutines are
C     called directly from MN. If they are called from
C     another subroutine, the @PROCESS command must be placed
C     in COLUMN 1 OF THE LINE IMMEDIATELY FOLLOWING THE END
C     STATEMENT OF THE PREVIOUS ROUTINE.
C
C         To compile (on MTS), first empty -mass and -pdf. Then
C     $RUN *FORTRANVS SCARDS=OMNIFOIL SPUNCH=FOILRUN
C
C         To run, enter $RUN FOILRUN+*IMSLDPLIB+*PLOTLIB.
C     You will be prompted for the following input parameters:
C
C         median volume droplet DIAMETER in microns.
C
C         y or n to simulated spectrum option
C
C         airfoil chord length in metres
C
C         NACA airfoil height as fraction of chord
C
C         free stream air speed in metres per second
C
C         liquid water content in kg per cubic metre
C
C         time step per layer in seconds
C
C         total number of layers
C
C         angle of attack of the free stream in degrees
C
C         OR, input data may be entered in a file. You must
C      then specify "5 = filename" in the $RUN command.
C
C      ********************** Subprograms **********************
C
C
C      INTERNAL:
C
C          SUBROUTINE DRAW - Plots airfoil profile and all
C       accreted ice layers with appropriate legend.
C
C          SUBROUTINE DROPS - Integrates droplet trajectories
C       and calculates stagnation line collision efficiency.
C       Subroutines called only from within DROPS (COLLID,
C       FLOWXY,KCALC,POFLO,TRAJEC) are described in the program
C       summary to DROPS.
C
C          DOUBLE PRECISION FUNCTION ONEBET - Estimates local
C       collision efficiency as a function of surface slope for
C       a monodisperse droplet population at the mvd.
C
C          DOUBLE PRECISION FUNCTION PHIBET - calculates local
C       collision efficiency as a function of slope w.r.t to
C       free stream direction, and distance from the maximum
```

```
       ...1.........2.........3.........4.........5.........6.........7.........8.........9.........
C     local value of collision efficiency (set to 0.95 times
C     the value for zero angle of attack). The location of
C     the maximum is estimated from a correlation derived
C     from previous numerical results .
C
C
C        DOUBLE PRECISION FUNCTION SPCBET - Estimates local
C     collision efficiency as a function of surface slope
C     for an average distribution shape of droplet sizes,
C     scaled to the mvd. The function assumes that small
C     droplets dominate in the spectrum, with mvd less than
C     about.20 microns.
C
C
C
C    EXTERNAL:
C
C       *IMSLDPLIB - The subroutine LEQT1F is used to solve
C    the matrix of potential vorticites for the potential
C    flow calculation in subroutine DROPS.
C
C       *PLOTLIB  -  The subroutines PLOTS, METRIC, PLOT, AXIS2,
C    LINE, SYMBOL and NUMBER (called from SUBROUTINE DRAW) are
C    found in the public system subroutine library *PLOTLIB.
C    (See note above regarding external subroutines in Fortran77).
C
C    ****************** Variable Dictionary *******************
C
C      ALPHA         - array of angles between surface normal
C                      and the free stream direction
C
C      ALPHM         - Alpha max, the maximum impingement surface
C                      angle
C
C      APHI          - parameter for determining the distribution
C                      of collision efficiency and of density
C                      about the position of maximum c.e. when
C                      phi is non-zero.
C
C      AREA          - cross sectional area between
C                      adjacent points in the current layer
C                      (dimensioned in metres)
C
C      AZERO         - the surface angle of maximum collision
C                      efficiency when phi is non-zero,
C                      determined from an approximating
C                      expression.
C
C      BETA          - local collision efficiency.
C
C      BZERO         - stagnation line value of the local collision
C                      efficiency
C
C      CHORD         - length of airfoil chord plus accreted ice
C
C      CHORD1        - length of bare airfoil chord in metres
```

```
 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C                   (INPUT)
C
C     DIAG        - cross diagonal of the quadrilateral
C                   of AREA
C
C     DROPDI      - droplet diameter in microns (INPUT)
C
C     H           - airfoil thickness as a fraction of the
C                   chord length (INPUT)
C
C     HALFPI      - pi/2 (Fortran77 constant PARAMETER)
C
C     LAREA       - cross sectional area of the current layer
C                   in sq. metres
C
C     LAYERS      - integer number of layers
C
C     LAYRX,LAYRY - array containing non - dimensional x and y
C                   coordinates of the accretion surfaces for
C                   up to 30 accretion layers
C
C     LDENS       - mean density of the current layer in kg/cubic m
C
C     LMASS       - mass of the current layer in kg per unit length
C
C     LWC         - liquid water content in kg per cubic metre
C                   (INPUT)
C
C     L1,L2,L3    - dimensioned sides of the quadrilateral area
C     L4            defined by two adjacent points on the
C                   current surface and two on the surface below
C
C     MASS        - array of masses per unit length for areas
C                   defined by adjacent points, in kg
C
C     NPOINT      - number of points defining the initial
C                   airfoil surface (note not the same as
C                   for potential flow calculation)
C
C     PFOILX,     - arrays of non-dimensional x,y points
C     PFOILY        of the front of the initial surface,
C                   for plotting
C
C     PHI         - angle of attack of the free stream
C                   (INPUT)
C
C     PI          - pi (Fortran77 constant PARAMETER)
C
C     RHO         - array of local densities in kg/cubic metre
C
C     SIMSPC      - logical variable, true if simulated spectrum
C                   option is chosen
C
C     SIMOPT      - character variable for reading in response to
C                   simulated spectrum option query (INPUT)
C
```

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9..........*
                 C       SLOPE        - array of local surface slopes calculated
                 C                      at each surface point
                 C
                 C       S1,S2        - - quantities used to determine the quadrilaterals
                 C                      for the layer area calculation
                 C
                 C       TAILEN       - filter length for optional smoothing filter
                 C                      applied to c. e. curve
                 C
                 C       TAU          - accretion time step in seconds (INPUT)
                 C
                 C       TH           - accumulated stagnation line thickness
                 C
                 C       THICK        - array of ice thicknesses normal to surface
                 C
                 C       TMASS        - total mass of the accretion
                 C
                 C       VEL          - free stream air speed in m per sec (INPUT)
                 C
                 C       XFOIL,       - arrays containing  dimensional x and
                 C       YFOIL          y coordinates of the airfoil surface
                 C
                 C
                 C ***************************************************************
                 C
ISN     1                DOUBLE PRECISION ALPHA(600),GAMMA(41),LAYRX(20,300),
                     +                    LAYRY(20,300),PFOILX(302),PFOILY(302),
                     +                    RHO(300),SLOPE(300),THICK(300),
                     +                    XFOIL(600),YFOIL(600)
                 C
ISN     2                DOUBLE PRECISION AZERO,ALPHM,APHI,AREA,BETA,BZERO,
                     +                    CHORD,CHORD1,DIAG,DROPDI,H,LAREA,LDENS,
                     +                    LMASS,LWC,L1,L2,L3,L4,MASS,PHI,S1,
                     +                    S2,TAILEN,TAU,TH,TMASS,VEL
                 C
ISN     3                INTEGER          LAYERS,NPOINT
                 C
ISN     4                CHARACTER*1      SIMOPT
                 C
ISN     5                LOGICAL          SIMSPC
                 C
ISN     6                PARAMETER (PI = 3.1415926535897932,HALFPI = PI/2)
                 C
ISN     7                COMMON /MN/XFOIL,YFOIL,GAMMA/DR/LAYRX,LAYRY,PFOILX,PFOILY
                 C
                 C  Prepare mass table and plot description files.
                 C
ISN     8                OPEN (UNIT = 8, FILE = '-MASS', STATUS = 'UNKNOWN')
ISN     9                OPEN (UNIT = 9, FILE = '-PDF', STATUS = 'UNKNOWN')
                 C
                 C  Set up headings for mass and density table in -MASS.
                 C
ISN    10                WRITE (8,110)
ISN    11          110   FORMAT (///,15X,'Mass of Ice Accreted',//,5X,'Layer no.',
                     +           7X,'Mass',7X,'Mean Density',//,
                     +           20X,'(kg/m)',9X,'(kg/cu.m)',//,
```

```
                   *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                        +          ' ***************************************************')
                   C
                   C   Write program title and intro to screen.
                   C
ISN    12                 WRITE (6,120)
ISN    13            120  FORMAT (/////,' *************  AIRFOIL RIME ICING MODEL  *****',
                        +          '************',//,17X,' Version Mar. 12, 1986',//,
                        +          20X,' K. J. Finstad',/,17X,
                        +          ' Meteorology Division',/,12X,' The University of Alberta'
                        +          ,//,' Please enter parameters to begin accretion: ',//)
                   C
                   C   Prompt user for input parameters.
                   C
ISN    14                 WRITE (6,*) 'Median Volume Droplet DIAMETER in microns? '
ISN    15                 READ  (5,*) DROPDI
                   C
                   C   Convert to metres
                   C
ISN    16                 DROPDI = DROPDI*1.0E-6
                   C
                   C   Set flags for requested treatment of droplet spectrum.
                   C
ISN    17                    WRITE (6,*)
ISN    18                    WRITE (6,*) 'Do you wish simulated spectrum weighting? '
ISN    19                    READ  (5,8) SIMOPT
ISN    20              8     FORMAT (1A1)
                   C
                   C   For simulated spectrum, calculate 'tail' length from
                   C   median volume diameter.
                   C
ISN    21                 IF ((SIMOPT .EQ. 'Y') .OR. (SIMOPT .EQ. 'y')) THEN
ISN    22                   SIMSPC = .TRUE.
ISN    23                    TAILEN = 0.13 * DEXP(-DROPDI / 7.0E-5)
ISN    24                 ELSE
ISN    25                   SIMSPC = .FALSE.
ISN    26                   TAILEN = 0.0
ISN    27                 ENDIF
                   C
ISN    28                 WRITE (6,*)
ISN    29                 WRITE (6,*) 'Airfoil chord length in metres? '
ISN    30                 READ  (5,*) CHORD1
                   C
                   C   Set current chord length equal to bare airfoil
                   C
ISN    31                 CHORD = CHORD1
                   C
ISN    32                 WRITE (6,*)
ISN    33                 WRITE (6,*) 'Airfoil height as fraction of chord? '
ISN    34                 READ  (5,*) H
                   C
ISN    35                 WRITE (6,*)
ISN    36                 WRITE (6,*) 'Free stream speed in m / sec? '
ISN    37                 READ  (5,*) VEL
                   C
ISN    38                 WRITE (6,*)
ISN    39                 WRITE (6,*) 'Liquid water content in kg / cubic metre? '
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN        DATE: JUL 24, 1986    TIME: 13:15:48     NAME: MAIN    PAGE:    7

                              *....*..1.......2.......3.......4.......5.......6.......7.*.....8.......9.......*

ISN    40                          READ  (5,*) LWC
                              C
ISN    41                          WRITE (6,*)
ISN    42                          WRITE (6,*) 'Time step per layer in seconds? '
ISN    43                          READ  (5,*)  TAU
                              C
ISN    44                          WRITE (6,*)
ISN    45                          WRITE (6,*) 'Total number of layers? '
ISN    46                          READ  (5,*) LAYERS
                              C
                              C  Bare surface will be layer 1, so add 1 to LAYERS
                              C
ISN    47                          LAYERS = LAYERS + 1
                              C
ISN    48                          WRITE (6,*)
ISN    49                          WRITE (6,*) 'Angle of attack in degrees? '
ISN    50                          READ  (5,*) PHI
                              C
                              C Convert angle of attack to radians, and calculate
                              C surface angle of the max collision efficiency point
                              C for non-zero attack angles (AZERO).
                              C
ISN    51                          AZERO = DLOG10(PHI + 1.) / (0.001 * DEXP(30.5 * H))
ISN    52                          PHI   = (PHI * 2.0 * PI) / 360.0
ISN    53                          AZERO = (AZERO * 2.0 * PI) / 360.0
                              C
                              C Fill array of points defining the initial  surface of the
                              C airfoil in dimensional coordinates. The complete
                              C surface contains NPOINT points.
                              C
ISN    54                           NPOINT = 600
                              C
ISN    55                          DO 100 I = 1,NPOINT
ISN    56                             XFOIL(I) = 0.5 * (1.0 - DCOS(2.0 * PI * DBLE(I-1)/
                                  +                DBLE(NPOINT)))
ISN    57                             YFOIL(I) = 5.0 * H * (0.2969 * DSQRT(XFOIL(I)) -
                                  +            0.126 * XFOIL(I) - 0.3516 * XFOIL(I) ** 2
                                  +                + 0.2843 * XFOIL(I) ** 3 -
                                  +                0.1015 * XFOIL(I) ** 4) * CHORD1
ISN    58                             IF (I .GT. 301) YFOIL(I) = -YFOIL(I)
ISN    60                             XFOIL(I) = XFOIL(I) * CHORD1
ISN    61                      100  CONTINUE
                              C
                              C  Fill arrays of non-dimensional coordinates for later plotting.
                              C
ISN    62                          DO 101 I=1,150
ISN    63                             PFOILX(150+I) = XFOIL(I) / CHORD1
ISN    64                             PFOILY(150+I) = YFOIL(I) / CHORD1
ISN    65                             PFOILX(I)     = XFOIL(450+I) / CHORD1
ISN    66                             PFOILY(I)     = YFOIL(450+I) / CHORD1
ISN    67                      101  CONTINUE
                              C
                              C  Copy front part of  surface to array which will contain
                              C  all of the layers.
                              C
ISN    68                          DO 200 I = 1,NPOINT / 4
```

```
                    *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN  69                    LAYRX(1,I) = XFOIL((3 * NPOINT / 4) + I)
ISN  70                    LAYRY(1,I) = YFOIL((3 * NPOINT / 4) + I)
ISN  71              200   CONTINUE
ISN  72                    DO 210 I = (NPOINT / 4) + 1, NPOINT / 2
ISN  73                      LAYRX(1,I) = XFOIL(I - (NPOINT / 4))
ISN  74                      LAYRY(1,I) = YFOIL(I - (NPOINT / 4))
ISN  75              210   CONTINUE
                     C
                     C  Call Subroutine DROPS to calculate Beta zero. Alpha max is
                     C  then determined from Beta zero.
                     C
ISN  76                    CALL DROPS(DROPDI,VEL,CHORD1,NPOINT,BZERO)
ISN  77                    ALPHM = DACOS(1.0 - BZERO)
                     C
                     C
                     C  Initialize total mass and begin loop to accrete ice layers.
                     C
ISN  78                    TMASS = 0.0
ISN  79                    DO 300 J = 2,LAYERS
                     C
                     C  Check for recalculation of BZERO every fifth layer
                     C
ISN  80                    IF ((J .EQ. 6) .OR. (J .EQ. 11) .OR. (J .EQ. 16)) THEN
                     C
                     C  Copy new shape to arrays for potential flow calculation in DROPS
                     C
ISN  81                      DO 220 I=1,NPOINT / 4
ISN  82                         XFOIL(I + (3*NPOINT/4)) = LAYRX(J-1,I)
ISN  83                         YFOIL(I + (3*NPOINT/4)) = LAYRY(J-1,I)
ISN  84              220     CONTINUE
ISN  85                      DO 230 I = (NPOINT / 4) + 1,NPOINT / 2
ISN  86                         XFOIL(I - NPOINT / 4) = LAYRX(J-1,I)
ISN  87                         YFOIL(I - NPOINT / 4) = LAYRY(J-1,I)
ISN  88              230     CONTINUE
                     C
                     C  Update length of airfoil chord plus accreted ice
                     C
ISN  89                    CHORD = CHORD1 + (((J-2) * BZERO * VEL * LWC * TAU)/890.0)
                     C
                     C  Update BZERO, and ALPHM if necessary
                     C
ISN  90                      CALL DROPS(DROPDI,VEL,CHORD,NPOINT,BZERO)
ISN  91                      IF (DACOS(1.0 - BZERO) .LT. ALPHM) ALPHM =
                          +      DACOS(1.0 - BZERO)
ISN  93                    ENDIF
                     C
                     C  Calculate slope and surface angle from finite differences for
                     C  each point on current surface
                     C
ISN  94                    DO 400 I = 2,299
ISN  95                      SLOPE(I) = (LAYRX(J-1,I+1) - LAYRX(J-1,I-1)) /
                          +             DABS(LAYRY(J-1,I+1) - LAYRY(J-1,I-1))
ISN  96                      IF (LAYRY(J-1,I+1) .EQ. LAYRY(J-1,I-1))
                          +      SLOPE(I) = 1.0E10
ISN  98                      IF (LAYRY(J-1,I+1) .LT. LAYRY(J-1,I-1)) THEN
ISN  99                         ALPHA(I) = PI - DATAN(SLOPE(I))
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:48        NAME: MAIN     PAGE:

                               *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.......*
ISN     100      444.                   ELSE
ISN     101      445.                      ALPHA(I) = DATAN(SLOPE(I))
ISN     102      446.                   ENDIF
ISN     103      447.            400  CONTINUE
                 448.          C
                 449.          C  Calculate for each point local density, collision efficiency and
                 450.          C  ice thickness. The method of calculation for Beta and rho
                 451.          C  depend on the values of SIMSPC and PHI.
                 452.          C
ISN     104      453.                 DO 500 I = 2,299
ISN     105      454.                   IF (PHI .EQ. 0.0) THEN
ISN     106      455.                       RHO(I) =890. *
                 456.              +               (1.0 - 0.163 * DABS(ALPHA(I)/(ALPHM+TAILEN)) +
                 457.              +               0.235 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 2) -
                 458.              +               2.357 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 3) +
                 459.              +               1.374 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 4) +
                 460.              +               0.0188 * (DABS(ALPHA(I)/(ALPHM+TAILEN)) ** 5))
ISN     107      461.                   ELSE
ISN     108      462.                       APHI   = ((-ALPHA(I) - AZERO) / (HALFPI - AZERO)
                 463.              +                  * HALFPI)
ISN     109      464.                       RHO(I) = 890.0 *
                 465.              +                  (1.0 - (0.163 * DABS(APHI/HALFPI)) +
                 466.              +                  (0.235 * DABS(APHI/HALFPI) ** 2) -
                 467.              +                  (2.357 * DABS(APHI/HALFPI) ** 3) +
                 468.              +                  (1.374 * DABS(APHI/HALFPI) ** 4) +
                 469.              +                  (0.0188 * DABS(APHI/HALFPI) ** 5))
ISN     110      470.                   ENDIF
                 471.          C        IF (ALPHA(I) .GT. HALFPI) RHO(I) =170.
                 472.          C
                 473.          C  Make sure density does not become too small
                 474.          C
ISN     111      475.                   IF (RHO(I) .LT. 170.) RHO(I) =170.
                 476.          C
                 477.          C  Calculate local collision efficiency
                 478.          C
ISN     113      479.                   IF (PHI .NE. 0.0) THEN
ISN     114      480.                      BETA = 0.95 * SPCBET(APHI,BZERO,HALFPI,0.0)
ISN     115      481.                   ELSE IF (SIMSPC) THEN
ISN     116      482.                      BETA = SPCBET(ALPHA(I),BZERO,ALPHM,TAILEN)
ISN     117      483.                   ELSE
ISN     118      484.                      BETA = ONEBET(ALPHA(I),BZERO,ALPHM)
ISN     119      485.                   ENDIF
                 486.          C
                 487.          C  Calculate local ice thickness perpendicular to surface
                 488.          C
ISN     120      489.                   THICK(I) = BETA * VEL * TAU * LWC / RHO(I)
                 490.          C
                 491.          C  Define new layer surface coordinates.
                 492.          C
ISN     121      493.                   LAYRX(J,I) = LAYRX(J-1,I) - (THICK(I) *
                 494.              +                           DCOS(DATAN(SLOPE(I))))
ISN     122      495.                   LAYRY(J,I) = LAYRY(J-1,I) + (THICK(I) *
                 496.              +                           DSIN(DATAN(SLOPE(I))))
ISN     123      497.             500  CONTINUE
ISN     124      498.                  LAYRX(J,1)   = LAYRX(J-1,1)
ISN     125      499.                  LAYRX(J,300) = LAYRX(J-1,300)
```

```
       . . . . . . .1. . . . . . . .2. . . . . . . .3. . . . . . . .4. . . . . . . .5. . . . . . . .6. . . . . . . .7. . . . . . . .8. . . . . . . .9. . . . . . . . .
ISN 126          LAYRY(J,1)   = LAYRY(J-1,1)
ISN 127          LAYRY(J,300) = LAYRY(J-1,300)
        C
        C  Calculate mass and mean density of the layer from
        C  the sum of areas between surface points on successive
        C  layers. First, initialize layer mass and area
        C
ISN 128            LMASS = 0.0
ISN 129            LAREA = 0.0
ISN 130            DO 560 I = 2,298
ISN 131               IF ((THICK(I) .GT. 0.0) .AND. (THICK(I+1) .GT. 0.0)) THEN
ISN 132                  L4 = THICK(I)
ISN 133                  L2 = THICK(I+1)
ISN 134                  L1 = DSQRT((LAYRX(J-1,I+1) - LAYRX(J-1,I))**2 + (LAYRY
             +                (J-1,I+1) - LAYRY(J-1,I))**2)
ISN 135                  L3 = DSQRT((LAYRX(J,I+1) - LAYRX(J,I))**2 + (LAYRY
             +                (J,I+1) - LAYRY(J,I))**2)
ISN 136                  DIAG = DSQRT((LAYRX(J,I+1)-LAYRX(J-1,I))**2 + (LAYRY
             +                  (J,I+1) - LAYRY(J-1,I))**2)
ISN 137                  S1   = 0.5 * (DIAG+L1+L2)
ISN 138                  S2   = 0.5 * (DIAG+L3+L4)
ISN 139                  AREA = DSQRT(S1 * (S1-DIAG) * (S1-L1) * (S1-L2)) +
             +                     DSQRT(S2 * (S2-DIAG) * (S2-L3) * (S2-L4))
ISN 140                  MASS = ((RHO(I) + RHO(I+1)) / 2.0) * AREA
ISN 141                  LMASS   = LMASS + MASS
ISN 142                  LAREA   = LAREA + AREA
ISN 143               ENDIF
ISN 144   560      CONTINUE
        C
        C  Calculate mean density.
        C
ISN 145            LDENS = LMASS/LAREA
        C
        C  Write results to mass table
        C
ISN 146            WRITE (8,570) J-1,LMASS,LDENS
ISN 147   570      FORMAT (//,8X,I2,7X,F9.5,7X,F9.5)
        C
        C  Add mass of current layer to total mass.
        C
ISN 148            TMASS = TMASS + LMASS
        C
        C  Shift surface points closer to nose by a decreasing
        C  interval between points
        C
ISN 149          IF ((J .GT. 2) .AND. (PHI .EQ. 0.0)) THEN
ISN 150             DO 580 I=1,149
ISN 151                IF (THICK(151-I) .GT. 0.0) THEN
ISN 152                   LAYRX(J,151-I) = LAYRX(J,151-I) -
             +                             ((0.4 * (20./DBLE(I+19))) *
             +                             DABS(LAYRX(J,151-I)-LAYRX(J,152-I)))
ISN 153                   LAYRY(J,151-I) = LAYRY(J,151-I) +
             +                             ((0.4 * (30./DBLE(I+19))) *
             +                              DABS(LAYRY(J,151-I)-LAYRY(J,152-I)))
ISN 154                ENDIF
ISN 155                IF (THICK(I+151) .GT. 0.0) THEN
```

```
                      *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN   156                       LAYRX(J,I+151) = LAYRX(J,I+151) -
                      +                          ((0.4 * (20./DBLE(I+19))) *
                      +                          DABS(LAYRX(J,I+150)-LAYRX(J,I+151)))
ISN   157                       LAYRY(J,I+151) = LAYRY(J,I+151) -
                      +                          ((0.4 * (20./DBLE(I+19))) *
                      +                          DABS(LAYRY(J,I+150)-LAYRY(J,I+151)))
ISN   158                      ENDIF
ISN   159           580    CONTINUE
ISN   160                ENDIF
                  C
ISN   161           800  CONTINUE
                  C
                  C  Write total mass accreted to table
                  C
ISN   162                WRITE (8,590) TMASS
ISN   163           590  FORMAT (8X,//,' Total mass accreted per metre: ',F9.5)
                  C
                  C  Call subroutine to plot airfoil and accreted layers
                  C
ISN   164                CALL DRAW(ALPHM,BZERO,CHORD1,DROPDI,LAYERS,LWC,PHI,
                      +            SIMSPC,TAILEN,TAU,VEL)
                  C
ISN   165           999  STOP
ISN   166                END

*STATISTICS*   SOURCE STATEMENTS = 162, PROGRAM SIZE = 21362 BYTES, PROGRAM NAME = MAIN   PAGE:   1.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 1 ******
```

REQUESTED OPTIONS (PROCESS): SC(AXIS2,SYMBOL)

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                C
                ******************************************************************
                *  SUBROUTINE DRAW - to plot airfoil  and all accretion          *
                *                    layers, with appropriate information        *
                *                    in the legend. This subroutine uses         *
                *                    the plotting library *PLOTLIB               *
                *                                                                *
                *  Input Variables : as defined for Main program :               *
                *                    ALPHM,BZERO,CHORD1,DROPDI,LAYERS,LWC,       *
                *                    PHI,SIMSPC,TAILEN,TAU,VEL                   *
                *                                                                *
                *  Common Variables :LAYRX,LAYRY,PFOILX,PFOILY                   *
                *                                                                *
                *  Other Variables : PLOTX,PLOTY - arrays containing x,y         *
                *                    plotting data and scaling information       *
                *                    for accretion layer currently being         *
                *                    plotted                                     *
                *                                                                *
                ******************************************************************
                C
ISN    1              SUBROUTINE DRAW(ALPHM,BZERO,CHORD1,DROPDI,LAYERS,
                     +           LWC,PHI,SIMSPC,TAILEN,TAU,VEL)
                C
ISN    2              DOUBLE PRECISION PFOILX(302),PFOILY(302),LAYRX(20,300),
                     +                 LAYRY(20,300),PLOTX(302),PLOTY(302)
                C
ISN    3              DOUBLE PRECISION ALPHM,BZERO,CHORD,DROPDI,LWC,PHI,
                     +                 TAILEN,TAU,VEL
                C
ISN    4              INTEGER          LAYERS
                C
ISN    5              LOGICAL          SIMSPC
                C
ISN    6              COMMON /DR/LAYRX,LAYRY,PFOILX,PFOILY
                C
                C  To plot airfoil surface,
                C  place scaling data at the end of coordinate arrays
                C
ISN    7              PFOILX(301) = -0.4
ISN    8              PFOILX(302) =  0.1
ISN    9              PFOILY(301) = -0.25
ISN   10              PFOILY(302) =  0.1
ISN   11              PLOTX(301)  = -0.4
ISN   12              PLOTX(302)  =  0.1
ISN   13              PLOTY(301)  = -0.25
ISN   14              PLOTY(302)  =  0.1
                C
                C  Open plotting library
                C
ISN   15              CALL PLOTS
ISN   16              CALL METRIC(0)
```

```
        *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
               C
               C  Draw 4 by 6 inch box
               C
ISN  17               CALL PLOT(2.0,2.0,-3)
ISN  18               CALL PLOT(0.0,6.0,3)
ISN  19               CALL PLOT(9.0,6.0,2)
ISN  20               CALL PLOT(9.0,0.0,2)
               C
               C  Label axes and draw tics
               C
ISN  21               CALL AXIS2(0.0,0.0,'X/CHORD',-7,9.0,0.0,-0.4,0.1,1.0)
ISN  22               CALL AXIS2(0.0,0.0,'Y/CHORD',7,6.0,90.0,-0.25,0.1,0.5)
               C
               C  Write legends to top left corner
               C
ISN  23               CALL SYMBOL(1.1,5.4,0.10,'DROPLET MVD  = ',0.0,15)
ISN  24               CALL NUMBER(2.6,5.4,0.10,DROPDI,0.0,07)
ISN  25               CALL SYMBOL(3.5,5.4,0.10,'M',0.0,01)
ISN  26               CALL SYMBOL(1.1,5.2,0.10,'CHORD        = ',0.0,15)
ISN  27               CALL NUMBER(2.6,5.2,0.10,CHORD1,0.0,04)
ISN  28               CALL SYMBOL(3.5,5.2,0.10,'M',0.0,01)
ISN  29               CALL SYMBOL(1.1,5.0,0.10,'AIR SPEED    = ',0.0,15)
ISN  30               CALL NUMBER(2.6,5.0,0.10,VEL,0.0,02)
ISN  31               CALL SYMBOL(3.5,5.0,0.10,'M / SEC',0.0,07)
ISN  32               CALL SYMBOL(1.1,4.8,0.10,'LWC          = ',0.0,15)
ISN  33               CALL NUMBER(2.6,4.8,0.10,LWC,0.0,05)
ISN  34               CALL SYMBOL(3.5,4.8,0.10,'KG / M**3',0.0,09)
ISN  35               CALL SYMBOL(1.1,4.6,0.10,'TIME STEP    = ',0.0,15)
ISN  36               CALL NUMBER(2.6,4.6,0.10,TAU,0.0,02)
ISN  37               CALL SYMBOL(3.5,4.6,0.10,'SEC',0.0,03)
ISN  38               CALL SYMBOL(1.1,4.4,0.10,'BETA ZERO    = ',0.0,15)
ISN  39               CALL NUMBER(2.6,4.4,0.10,BZERO,0.0,03)
ISN  40               CALL SYMBOL(1.1,4.2,0.10,'ALPHA MAX    = ',0.0,15)
ISN  41               CALL NUMBER(2.6,4.2,0.10,(ALPHM+TAILEN)*360./6.283,0.0,01)
ISN  42               CALL SYMBOL(3.5,4.2,0.10,'DEGREES',0.0,07)
ISN  43               CALL SYMBOL(1.1,4.0,0.10,'ATTACK ANGLE = ',0.0,15)
ISN  44               CALL NUMBER(2.6,4.0,0.10,PHI*360./6.283,0.0,01)
ISN  45               CALL SYMBOL(3.5,4.0,0.10,'DEGREES',0.0,07)
ISN  46               CALL SYMBOL(1.1,3.8,0.10,'SPECTRUM:',0.0,09)
ISN  47               IF ((SIMSPC) .OR. (PHI .NE. 0.0)) THEN
ISN  48                  CALL SYMBOL(2.6,3.8,0.10,'SIMULATED',0.0,09)
ISN  49               ELSE
ISN  50                  CALL SYMBOL(2.6,3.8,0.10,'NO',0.0,02)
ISN  51               ENDIF
               C
               C  Draw initial airfoil surface
               C
ISN  52               CALL LINE(PFOILX,PFOILY,300,2,0)
               C
               C  Transfer each layer in turn to the array holding
               C  the scaling parameters
               C
ISN  53               DO 610 J = 2,LAYERS
ISN  54                  DO 700 L = 1,300
ISN  55                     PLOTX(L) =  LAYRX(J,L) / CHORD1
ISN  56                     PLOTY(L) =  LAYRY(J,L) / CHORD1
```

```
                          *....*...1.........2.........3.........4.........5.........6.........7.*........8.........9.........*
ISN    57     688.      700     CONTINUE
              689.    C
              690.    C  Draw accretion layer
              691.    C
ISN    58     692.              CALL LINE(PLOTX,PLOTY,300,2,0)
ISN    59     693.      610  CONTINUE
              694.    C
              695.    C  Close plotting library
              696.    C
ISN    60     697.           CALL PLOT(1.0,1.0,999)
ISN    61     698.           RETURN
ISN    62     699.           END
*STATISTICS*   SOURCE STATEMENTS = 62, PROGRAM SIZE = 7298 BYTES, PROGRAM NAME = DRAW    PAGE:   12.
*STATISTICS*    NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
```

```
LEVEL 1.1.1 (DEC   81)        VS FORTRAN        DATE: JUL 24, 1986    TIME: 13:15:50     NAME: MAIN    PAGE:   15

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED        NOTEST   SEQ
                      OPTIMIZE(0)  LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)

                     *...*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

                     C *************************************************************
                     C *  FUNCTION ONEBET - Calculates local collision efficiency*
                     C *   estimate for a monodisperse droplet populations         *
                     C *                                                           *
                     C *                                                           *
                     C *  Input Variables : as defined for main program;           *
                     C *                    ALPHM,BZERO,ALPHA,SMDROP               *
                     C *                                                           *
                     C *************************************************************
                     C
ISN      1                 DOUBLE PRECISION FUNCTION ONEBET(ALPHA,BZERO,ALPHM)
                     C
ISN      2                 DOUBLE PRECISION BZERO,ALPHM,ALPHA
                     C
                     C
ISN      3                 IF (DABS(ALPHA) .GE. ALPHM) THEN
ISN      4                    ONEBET = 0.0
ISN      5                 ELSE
ISN      6                    ONEBET = DCOS(ALPHA) - (1.0 - BZERO)
ISN      7                 ENDIF
ISN      8                 RETURN
ISN      9                 END

*STATISTICS*   SOURCE STATEMENTS = 9, PROGRAM SIZE = 870 BYTES, PROGRAM NAME = ONEBET    PAGE:   15.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 3 ******
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN        DATE: JUL 24, 1986    TIME: 13:15:50      NAME: MAIN    PAGE:   16

OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED         NOTEST   SEQ
                     OPTIMIZE(0)  LANGLVL(77) NOFIPS    FLAG(I)  NAME(MAIN   )  LINECOUNT(60)

                        *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                        C
                        C ***********************************************************
                        C *  Function SPCBET - Calculates the local c.e. for a      *
                        C *           simulated droplet size distribution          *
                        C *           scaled to the median volume droplet,         *
                        C *           and to (alpha max + tail). The tail          *
                        C *           length was determined in the main program.   *
                        C *                                                        *
                        C *  Input Variables : as defined for Main program;        *
                        C *                    ALPHA,ALPHM,BZERO,SMDROP,TAILEN      *
                        C *                                                        *
                        C *  Other Variables : X - scaled angle = Alpha /          *
                        C *                         (Alphm + Tailen)               *
                        C *                                                        *
                        C ***********************************************************
                        C
ISN       1                   DOUBLE PRECISION FUNCTION SPCBET(ALPHA,BZERO,
                             +                                  ALPHM,TAILEN)
                        C
ISN       2                   DOUBLE PRECISION ALPHA,ALPHM,BZERO,TAILEN,X
                        C
ISN       3                   ALPHA = DABS(ALPHA)
ISN       4                   IF (ALPHA .GE. (ALPHM+TAILEN)) THEN
ISN       5                     SPCBET = 0.0
ISN       6                   ELSE
ISN       7                     X = ALPHA / (ALPHM+TAILEN)
ISN       8                       SPCBET = BZERO * (1.000 + 0.039 * X - 1.842 *
                             +             (X ** 2) - 0.543 * (X ** 3)
                             +             + 1.792 * (X **4) - 0.444 * (X ** 5))
ISN       9                       IF (SPCBET .LT. 0.0) SPCBET = 0.0
ISN      11                       IF (SPCBET .GT. BZERO) SPCBET = BZERO
ISN      13                   ENDIF
ISN      14                   RETURN
ISN      15                   END

*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 608 BYTES, PROGRAM NAME = SPCBET    PAGE:   16.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 4 ******
```

```
      *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C
C *************************************************************
C  SUBROUTINE DROPS
C
C     This routine calculates a trajectory pair for water
C  droplets in a potential flow field around a NACA airfoil or
C  similar accretion shape. The stagnation line collision
C  efficiency is then determined.
C
C     The equation of motion of the droplet is integrated
C  by the Heun method (Mesinger and Arakawa, 1976: Numerical
C  Methods used in Atmospheric Models, Volume I. W.M.O. GARP
C  Publications Series No. 17). The time step is set to
C  1/400 of the time to travel one chord length at the free
C  stream speed. A maximum of 3000 steps are allowed,
C  by which time the droplet is assumed to have collided
C  with the airfoil or to have passed it.
C
C     The droplet's initial x-coordinate is set to an
C  upstream distance of 5.0 chord lengths.  At this
C  distance the x-speed is assumed to be equal to the air
C  speed in the x-direction, and the droplet y-speed is
C  set to zero.
C
C     A first trajectory is integrated beginning at y = zero,
C  which is then incremented by a small amount (DELTAY)
C  for the second trajectory. If both trajectories
C  intercept the airfoil, the local collision efficiency, and
C  the angle between the local surface normal and free stream
C  direction are calculated.
C
C     The formulation of Beard and Pruppacher (J.A.S., Vol. 26,
C  p. 1066, 1969) is used for the drag coefficient.  Potential
C  flow calculations are made according to the method of Kennedy
C  and Marsden (Can. Aero. and Space J., Vol. 22, p. 243, 1976)
C
C     One external subprogram is required, from the IMSL library;
C  the subroutine LEQTIF, to solve a system of equations
C  in the potential flow calculation.
C
C     The following internal subprograms are called from DROPS.
C
C  ******************** Subprograms *********************
C
C
C     SUBROUTINE COLLID -calculates final impact position
C       if the droplet collides with the surface.
C
C     SUBROUTINE FLOWXY- calculates the air speeds at a
C       given x,y location using the results of POFLO,
C       and droplet accelerations (from the equations of
C       motion ) for the current iteration and time step.
C
```

```
      *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C     SUBROUTINE KCALC - calculates the matrix elements
C       K used in POFLO and FLOWXY.
C
C     SUBROUTINE POFLO - calculates the potential vorticites
C       at the surface of the airfoil according to the
C       method of Kennedy and Marsden.
C
C     SUBROUTINE TRAJEC - calculates droplet trajectories
C       by integrating the droplet equations of motion by
C       the trapezoid rule with one iteration step.
C
C
C     Input Variables:
C
C     CHORD      - airfoil chord length in metres
C
C     DROPDI     - droplet diameter in microns
C
C
C     NPOINT     - number of points in the complete airfoil
C                  surface (must be divisible by 15)
C     VEL        - free stream air speed in metres per second
C
C     Output Variables :
C
C     BZERO      - stagnation line collision efficiency
C
C     Common Variables: (with Main)
C
C     GAMMA      - vector of potential vorticities for
C                  potential flow calculation
C
C     XFOIL,     - vectors containing points which define
C     YFOIL        the complete dimensioned airfoil surface
C
C
C     Other Variables:
C
C     ALPHA      - angle between the local surface normal
C                  and the free stream direction at
C                  the impact point
C
C     ARC        - arc length along the airfoil surface
C                  between the impact points of the two
C                  trajectories (estimated by a straight
C                  line segment)
C     DELTAT     - time step in seconds
C
C     DELTAY     - difference in initial y position of
C                  the two trajectories
C
C     FINX1,     - x-positions of the droplets at impact
C     FINX2        for each trajectory
C
C     FINY1,     - y-positions of the droplets at impact
C     FINY2        for each trajectory
```

```
        *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
        C
        C     HALFPI     - pi / 2 (FORTRAN77 constant parameter)
        C
        C     HIT1,      - logical flags for each trajectory
        C     HIT2         which indicate whether a collision
        C                  with the airfoil has occurred
        C
        C     NSTEPS     - maximum number of time steps to be
        C                  integrated
        C
        C     PI         - pi (FORTRAN77 constant parameter)
        C
        C     SLOPE1,    - slopes of the airfoil or accretion
        C     SLOPE2       surface at the impact points
        C
        C     XINIT,     - initial droplet x,y position in metres
        C     YINIT
        C
        C
        C *********************************************************
        C
ISN   1         SUBROUTINE DROPS(DROPDI,VEL,CHORD,NPOINT,BZERO)
        C
ISN   2         DOUBLE PRECISION XFOIL(600),YFOIL(600),GAMMA(41)
        C
ISN   3         DOUBLE PRECISION ARC,BZERO,CHORD,DELTAT,DELTAY,
             +                   DROPDI,FINX1,FINX2,FINY1,FINY2,
             +                   SLOPE1,SLOPE2,VEL,XINIT,YINIT
        C
ISN   4         INTEGER          NSTEPS,NPOINT
        C
ISN   5         LOGICAL          HIT1,HIT2
        C
ISN   6         PARAMETER (PI = 3.1415926535897932,HALFPI = PI / 2.0)
        C
ISN   7         COMMON /MN/ XFOIL,YFOIL,GAMMA
        C
        C
        C Set maximum number of time steps , initial x-coordinate, and
        C the difference in initial y-coordinates
        C
ISN   8         NSTEPS = 3000
ISN   9         DELTAY = 1.0E-8
ISN  10         XINIT  = 5.0 * CHORD
ISN  11         YINIT  = 0.0
        C
        C Calculate time step
        C
ISN  12         DELTAT = CHORD / (VEL * 400)
        C
        C Calculate potential vorticity matrix for air speed calculations
        C
ISN  13         CALL POFLO(NPOINT,CHORD)
        C
        C Integrate first trajectory
        C
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:51      NAME: DROPS    PAGE:   20

                    *...|*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN    14                  CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,NSTEPS,
                         +              NPOINT,HIT1,FINX1,FINY1,SLOPE1)
                    C
                    C  Correct initial y position and integrate second trajectory
                    C
ISN    15                  YINIT = YINIT + DELTAY
ISN    16                  CALL TRAJEC (XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,NSTEPS,
                         +              NPOINT,HIT2,FINX2,FINY2,SLOPE2)
                    C
                    C  Test for result of trajectory (miss, graze or collision)
                    C
ISN    17                  IF ((HIT1) .AND. (HIT2)) THEN
                    C
                    C       Calculate impact parameters and local collision efficiency
                    C
ISN    18                     ARC    = DSQRT(((FINX2 - FINX1) ** 2) + ((FINY1 - FINY2)
                         +             ** 2))
ISN    19                     BZERO = DELTAY / ARC
                    C
                    C  If droplet didn't hit the foil, write message to screen
                    C
ISN    20                  ELSE IF ((.NOT. HIT1) .OR. (.NOT. HIT2)) THEN
ISN    21                     WRITE (6,40)
ISN    22            40       FORMAT (//,'Droplet did not reach the airfoil.')
ISN    23                  ENDIF
                    C
ISN    24                  RETURN
ISN    25                  END

*STATISTICS*   SOURCE STATEMENTS = 25, PROGRAM SIZE = 926 BYTES, PROGRAM NAME = DROPS    PAGE   17.

*STATISTICS*   NO DIAGNOSTICS GENERATED

****** END OF COMPILATION 5 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
*....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
C
C ***************************************************************
C *    SUBROUTINE TRAJEC                                         *
C *                                                              *
C *       Integrates trajectory of droplet whose initial         *
C *    position is specified in the parameter list. Calls        *
C *    POFLO to set up matrix for potential flow calculations,   *
C *    FLOWXY for iterative calculations of air speed and        *
C *    droplet acceleration, and COLLID to determine             *
C *    impact points.                                            *
C *                                                              *
C *    INPUT VARIABLES:                                          *
C *                   CHORD  - airfoil chord length              *
C *                   DELTAT - time step                         *
C *                   DROPDI - droplet diameter                  *
C *                   NPOINT - number of surface points          *
C *                   NSTEPS - maximum number of time steps      *
C *                   VEL    - free stream air speed             *
C *                   XINIT  - initial x position                *
C *                   YINIT  - initial y position                *
C *                                                              *
C *    OUTPUT VARIABLES:                                         *
C *                   FINALX - droplet x velocity on impact      *
C *                   FINALY - droplet y velocity on impact      *
C *                   HIT    - logical flag set to .TRUE. if     *
C *                            a collision occurs                *
C *                   SLOPE  - slope of the surface at impact    *
C *                                                              *
C *    COMMON VARIBLES:                                          *
C *                   CO     - vector of cosines of angles of the *
C *                            control elements                  *
C *                   DELTA  - vector ot half-lengths of the     *
C *                            control elements                  *
C *                   GAMMA  - output array of POFLO, passed to  *
C *                            FLOWXY                            *
C *                   SI     - vector of sines of angles of the  *
C *                            control elements                  *
C *                   XCON,  - x,y coordinates of the control    *
C *                   YCON     points                            *
C *                   XFOIL, - vectors of initial surface        *
C *                   YFOIL    coordinates                       *
C *                                                              *
C *    OTHER VARIABLES:                                          *

C *                   DVX,   - arrays containing x,y components  *
C *                   DVY      of droplet accelerations          *
C *                   L      - counter for searching surface array*
C *                   LINIT  - initial value of counter L        *
C *                   VX,VY  - arrays containing x,y components  *
C *                            of droplet speed for each step    *
C *                   X,Y    - arrays containing x,y positions   *
C *                            for each time step                *
C *                                                              *
```

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                 C**************************************************************
                 C
ISN    1               SUBROUTINE TRAJEC(XINIT,YINIT,DROPDI,VEL,CHORD,DELTAT,
                      +          NSTEPS,NPOINT,HIT,FINALX,FINALY,SLOPE)
                 C
ISN    2               DOUBLE PRECISION X(2500),Y(2500),GAMMA(41),XFOIL(600),
                      +                 YFOIL(600),VX(2500),VY(2500),DVX(2500),
                      +                 DVY(2500),DELTA(40),CO(40),SI(40),
                      +                 XCON(41),YCON(41)
                 C
ISN    3               DOUBLE PRECISION XINIT,YINIT,DROPDI,CHORD,VEL,DELTAT,
                      +                 FINALX,FINALY,SLOPE
                 C
ISN    4               INTEGER          NSTEPS,NPOINT,LINIT
                 C
ISN    5               LOGICAL          HIT
                 C
ISN    6               COMMON /MN/ XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
                 C
                 C  Initialize values for integration
                 C
ISN    7               X(1)  = -XINIT
ISN    8               Y(1)  = YINIT
ISN    9               VY(1) = 0.0
ISN   10               VX(1) = VEL
ISN   11               I     = 0
ISN   12               HIT   = .FALSE.
                 C
                 C  Begin integration
                 C
ISN   13           100 I = I+1
ISN   14               IF (I .GE. NSTEPS) GOTO 250
                 C
                 C  Calculate current step air speeds and accelerations
                 C
ISN   15                  CALL FLOWXY(X(I),Y(I),VX(I),VY(I),VEL,CHORD,DROPDI,
                      +               NPOINT,DVX(I),DVY(I))
                 C
                 C  Calculate next step droplet speeds and positions
                 C
ISN   16                  VX(I+1) = VX(I) - (DELTAT * DVX(I))
ISN   17                  VY(I+1) = VY(I) + (DELTAT * DVY(I))
ISN   18                  X(I+1)  = X(I) + (VX(I) * DELTAT) - (0.5 * DVX(I)
                      +               * (DELTAT * DELTAT))
ISN   19                  Y(I+1)  = Y(I) + (VY(I) * DELTAT) + (0.5 * DVY(I)
                      +               * (DELTAT * DELTAT))
                 C
                 C  Calculate next step air speeds and accelerations
                 C
ISN   20                  CALL FLOWXY(X(I+1),Y(I+1),VX(I+1),VY(I+1),VEL,
                      +               CHORD,DROPDI,NPOINT,DVX(I+1),DVY(I+1))
                 C
                 C  Iterate calculations of next step droplet speed and
                 C  position using averaged values of accelerations
                 C
ISN   21                  VX(I+1) = VX(I)   ((DELTAT / 2.0) * (DVX(I) +
```

```
                    +              DVX(I+1)))
ISN     22                VV(I+1) = VV(I) + ((DELTAT / 2.0) * (DVV(I) +
                    +              DVV(I+1)))
ISN     23                X(I+1) =  X(I) + (VX(I) * DELTAT) - (0.25 *
                    +              (DVX(I) + DVX(I+1)) * (DELTAT * DELTAT))
ISN     24                Y(I+1) =  Y(I) + (VY(I) * DELTAT) + (0.25 *
                    +              (DVY(I) + DVY(I+1)) * (DELTAT * DELTAT))
                    C
                    C  If trajectory has reached the nose
                    C  test whether trajectory has intercepted the airfoil surface
                    C
                    C  Set LINIT to begin search on upper or lower surface
                    C  if the droplet has reached to within one current chord
                    C  length of the nose
                    C
ISN     25                LINIT = 0
ISN     26                L = LINIT
ISN     27          200      CONTINUE
ISN     28                      L = L + 1
ISN     29                      IF (L .GE. (LINIT + 125)) GOTO 100
                    C
                    C  Search for adjacent pair of points which bracket the
                    C  y-coordinate of current trajectory position (if pair
                    C  cannot be found, continue with next integration step)
                    C
ISN     30                      IF ((YFOIL(L+1) .GT. Y(I+1)) .AND. (YFOIL(L) .LT.
                    +              Y(I+1))) THEN
                    C
                    C  Check if current position has passed the airfoil surface
                    C  between the two points
                    C
ISN     31                         IF ((XFOIL(L+1) .LT. X(I+1)) .OR. (XFOIL(L) .LT.
                    +                 X(I+1))) THEN
ISN     32                            HIT   = .TRUE.
                    C
                    C  Determine the impact point on the surface, and local
                    C  surface slope at that point , then exit from subroutine
                    C
ISN     33                            CALL COLLID(X(I),X(I+1),Y(I),Y(I+1),XFOIL(L),
                    +                              YFOIL(L),XFOIL(L+1),YFOIL(L+1),
                    +                              FINALX,FINALY)
ISN     34                            SLOPE = (YFOIL(L+1) - YFOIL(L)) / DABS(XFOIL
                    +                         (L+1) - XFOIL(L))
ISN     35                            GOTO 250
ISN     36                         ELSE
                    C
                    C  If not inside surface, continue integration
                    C
ISN     37                            GOTO 100
ISN     38                         ENDIF
ISN     39                      ELSE
                    C
                    C  Else continue search for bracketing pair
                    C
ISN     40                         GOTO 200
ISN     41                      ENDIF
```

```
                                *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN       42      1114.       250       CONTINUE
                  1115.     C
ISN       43      1116.               RETURN
ISN       44      1117.               END
```

STATISTICS   SOURCE STATEMENTS = 44, PROGRAM SIZE = 122570 BYTES, PROGRAM NAME = TRAJEC   PAGE: 21.

STATISTICS   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 6 ******

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED        NOTEST  SEQ
                    OPTIMIZE(O)  LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN   )  LINECOUNT(60)

               *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.....

               C ***********************************************************
               C *                                                         *
               C *  SUBROUTINE POFLO - Uses the method of Kennedy and      *
               C *    Marsden (ref) to solve for the matrix of potential   *
               C *    vorticites about the airfoil surface.  A reduced     *
               C *    number of points is used to define the surface for   *
               C *    this subroutine than for the accretion section.      *
               C *    Subroutine KCALC is called to calculate the matrix   *
               C *    of K values.                                         *
               C *                                                         *
               C *  INPUT VARIABLES  :                                     *
               C *                     CHORD  - airfoil chord length       *
               C *                     NPOINT - number of points defining  *
               C *                              airfoil surface            *
               C *  COMMON VARIABLES :                                     *
               C *                     CO,SI  - vectors of cosines and sines *
               C *                              of the control element angles*
               C *                     DELTA  - vector containing control  *
               C *                              element half lengths       *
               C *                     GAMMA  - vector containing the      *
               C *                              potential vorticities on   *
               C *                              the airfoil surface        *
               C *                     XCON,  - vectors containing control *
               C *                     YCON     point coordinates          *
               C *                     XFOIL, - vectors of surface         *
               C *                     YFOIL    point coordinates          *
               C *  OTHER VARIABLES  :                                     *
               C *                     IDGT,    routine                    *
               C *                     IER                                 *
               C *                     K      - vector input to matrix     *
               C *                              solver subroutine LEQT1F   *
               C *                     N      - NPOINT/15, the no. of surface*
               C *                              points used to calculate   *
               C *                              the stream function        *
               C *                     WKAREA,- input required for LEQT1F  *
               C *                     XF,YF  - reduced vectors of surface *
               C *                              points (length NPOINT/15)  *
               C *                                                         *
               C ***********************************************************
               C
ISN    1              SUBROUTINE POFLO(NPOINT,CHORD)
               C
ISN    2              DOUBLE PRECISION GAMMA(41),XFOIL(600),YFOIL(600),
                    +                  XCON(41),YCON(41),XF(41),YF(41),
                    +                  K(41,41),WKAREA(41),DELTA(40),
                    +                  CO(40),SI(40),CHORD
               C
ISN    3              INTEGER NPOINT,N,IDGT,IER
               C
ISN    4              COMMON /MN/ XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
               C
               C  Construct new surface from every 15th point of old surface
               C  non-dimensionalised to the current chord length
```

```
                   *....*...1.........2.........3.........4.........5.........6.........7.*........8.........9.........*
                   C
ISN     5                N = NPOINT / 15
ISN     6                XF(1)    = XFOIL(1) / CHORD
ISN     7                YF(1)    = YFOIL(1) / CHORD
ISN     8                XF(N+1) = XF(1)
ISN     9                YF(N+1) = YF(1)
ISN    10                DO 10 I=2, N
ISN    11                   XF(I) = XFOIL((15 * (I-1)) + 1) / CHORD
ISN    12                   YF(I) = YFOIL((15 * (I-1)) + 1) / CHORD
ISN    13             10 CONTINUE
                   C
                   C  Form arrays of control points
                   C
ISN    14                DO 20 I = 1,N
ISN    15                   XCON(I) = (XF(I) + XF(I+1)) / 2.0
ISN    16                   YCON(I) = (YF(I) + YF(I+1)) / 2.0
ISN    17             20 CONTINUE
                   C
                   C  Define the trailing point
                   C
ISN    18                XCON(N+1) = ((XF((N/2)+1) - XCON(N/2)) * 0.01) + XF((N/2)+1)
ISN    19                YCON(N+1) = 0.0
                   C
                   C  Calculate R (as defined in Kennedy and Marsden) for each
                   C  control point, store in GAMMA
                   C
ISN    20                DO 30 J = 1,N+1
ISN    21                   GAMMA(J) = (YCON(J))
                   C
                   C  Set last column of K(I,J) to 1.0
                   C
ISN    22                   IF (J .EQ. N+1) THEN
ISN    23                      DO 35 I=1,N+1
ISN    24                         K(I,N+1) = 1.0
ISN    25             35          CONTINUE
                   C
                   C  For all other J, calculate the control element half lengths
                   C  and functions of the angle to the x-axis.
                   C
ISN    26                   ELSE
ISN    27                      DELTA(J) = DSQRT(((XF(J+1) - XF(J)) ** 2) +
                        +                  ((YF(J+1) - YF(J)) ** 2)) / 2
ISN    28                      CO(J)    = (XF(J+1) - XF(J)) / (2.0 * DEL[illegible]
ISN    29                      SI(J)    = (YF(J+1) - YF(J)) / (2.0 * DELT[illegible]
                   C
                   C  Calculate K for all remaining I,J
                   C
ISN    30                      DO 40 I=1,N
ISN    31                         CALL KCALC(XCON(J),YCON(J),XCON(I),YCON(I),
                        +                    DELTA(J),CO(J),SI(J),K(I,J))
ISN    32             40       CONTINUE
ISN    33                   ENDIF
ISN    34             30 CONTINUE
                   C
                   C  Call IMSLDPLIB routine to solve matrix equation;
                   C  output matrix of potential vorticites is in GAMMA.
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:52       NAME: POFLO    PAGE:    27
                                 *....*...1.........2.........3.........4.........5.........6.........7:*.......8.........9.........*
                  1227.    C
ISN       35      1228.            IDGT = 8
ISN       36      1229.            CALL LEQT1F(K,1,N+1,N+1,GAMMA,IDGT,WKAREA,IER)
ISN       37      1230.            WRITE (6,*) GAMMA
                  1231.    C
ISN       38      1232.            RETURN
ISN       39      1233.            END
*STATISTICS*   SOURCE STATEMENTS = 39, PROGRAM SIZE = 16358 BYTES, PROGRAM NAME = POFLO    PAGE:    25.
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 7 ******
```

```
                    *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
             C ***************************************************************
             C *                                                             *
             C *  SUBROUTINE FLOWXY  -  calculates for the current step:     *
             C *      air speed components from matrix equation for          *
             C *      potential flow streamline , using the output           *
             C *      matrix GAMMA from POFLO, and KCALC to generate the     *
             C *      matrix elements K. Droplet accelerations are then      *
             C *      calculated according to the equations of motion        *
             C *      from the air speed, droplet Reynolds number, and       *
             C *      drag coefficient according to the formulation of      *
             C *      Beard and Pruppacher.                                  *
             C *                                                             *
             C *  INPUT VARIABLES :                                          *
             C *               CHORD  - chord length                         *
             C *               DROPDI - droplet diameter                     *
             C *               NPOINT - no of airfoil surface points         *
             C *               VEL    - free stream air speed                *
             C *               VX,VY, - current droplet speeds               *
             C *               X,Y    - current droplet positions            *
             C *                                                             *
             C *  OUTPUT VARIABLES :                                         *
             C *               DVX,DVY - current droplet accelerations       *
             C *                                                             *
             C *  COMMON VARIABLES :                                         *
             C *               CO,SI,GAMMA,DELTA,XCON,YCON,XFOIL,YFOIL       *
             C *                                                             *
             C *  OTHER VARIABLES :                                          *
             C *               COEFF   - combined cefficients of the         *
             C *                         equations of motion                 *
             C *               DRAGCO  - drag coefficient                    *
             C *               DROPRE  - droplet Reynolds number             *
             C *               GRIDX,  - gridpoints used in the finite       *
             C *               GRIDY     difference calculation of air       *
             C *                         speed from the stream function      *
             C *               N       - NPOINT/15, the no. of surface       *
             C *                         points used to calculate the        *
             C *                         stream function                     *
             C *               K, R    - variables  for the calculation      *
             C *                         of stream function                  *
             C *               PSI     - stream function of the air flow     *
             C *                         field                               *
             C *               SUMK    - array containing the sum of K       *
             C *                         times GAMMA(J) for I gridpoints     *
             C *               UX,UY   - current air speed components        *
             C *                                                             *
             C ***************************************************************
             C
ISN    1            SUBROUTINE FLOWXY(X,Y,VX,VY,VEL,CHORD,DROPDI,NPOINT,
                 +                    DVX,DVY)
             C
ISN    2            DOUBLE PRECISION CO(40),DELTA(40),GAMMA(41),GRIDX(4),
                 +                   GRIDY(4),PSI(4),SI(40),SUMK(4),
                 +                   XCON(41),YCON(41),XFOIL(600),
```

```
                   *....*..1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
                         +                  YFOIL(600)
                   C
ISN    3                 DOUBLE PRECISION COEFF,CHORD,DRAGCO,DROPDI,DROPRE,DVX,
                         +                  DVY,K,R,UX,UY,VEL,VX,VY,X,Y
                   C
ISN    4                 INTEGER          N,NPOINT
                   C
ISN    5                 COMMON /MN/XFOIL,YFOIL,GAMMA/DRP/DELTA,CO,SI,XCON,YCON
                   C
                   C  Set up non-dimensional grid for finite difference calculation
                   C  of the streamfunction derivatives
                   C
ISN    6                 GRIDX(1) = X / CHORD
ISN    7                 GRIDX(2) = X / CHORD
ISN    8                 GRIDX(3) = (X - (DROPDI / 2.0)) / CHORD
ISN    9                 GRIDX(4) = (X + (DROPDI / 2.0)) / CHORD
ISN   10                 GRIDY(1) = (Y + (DROPDI / 2.0)) / CHORD
ISN   11                 GRIDY(2) = (Y - (DROPDI / 2.0)) / CHORD
ISN   12                 GRIDY(3) = Y / CHORD
ISN   13                 GRIDY(4) = Y / CHORD
                   C
                   C  Define N as in POFLO
                   C
ISN   14                 N = NPOINT / 15
                   C
                   C  Calculate for each grid point, R and the sum of products
                   C  K times GAMMA(J), for J =1, N
                   C
ISN   15                 DO 10 I =1,4
ISN   16                    R = (GRIDY(I))
ISN   17                    SUMK(I) = 0.0
ISN   18                    DO 20 J = 1,N
ISN   19                       CALL KCALC(XCON(J),YCON(J),GRIDX(I),GRIDY(I),
                         +                  DELTA(J),CO(J),SI(J),K)
ISN   20                       SUMK(I) = SUMK(I) + (GAMMA(J) * K)
ISN   21             20     CONTINUE
                   C
                   C  Calculate stream functions
                   C
ISN   22                    PSI(I) = -SUMK(I) + R - GAMMA(N+1)
ISN   23             10  CONTINUE
                   C
                   C  Calculate dimensional air speeds at point x,y from finite
                   C  difference derivatives of the stream functions
                   C
ISN   24                 UX = VEL * CHORD * (PSI(1) - PSI(2)) / DROPDI
ISN   25                 UY = VEL * CHORD * (PSI(3) - PSI(4)) / DROPDI
                   C
                   C Calculate Reynolds number and drag coefficient
                   C
ISN   26                 DROPRE     = DROPDI * DSQRT(((VX - UX) ** 2) + ((VY - UY)
                         +                ** 2)) / 1.262E-5
ISN   27                 IF (DROPRE .LT. 2.0) THEN
ISN   28                    DRAGCO = (24.0 / DROPRE) + (2.45 / (DROPRE ** 0.045))
ISN   29                 ELSE IF (DROPRE .LT. 21.0) THEN
ISN   30                    DRAGCO = (24.0 / DROPRE) + (2.76 / (DROPRE ** 0.198))
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986    TIME: 13:15:53      NAME: FLOWXY  PAGE:   30

                             *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN     31     1343.               ELSE IF (DROPRE .GT. 200.0) THEN
ISN     32     1344.                  DRAGCO = (24.0 / DROPRE) + (4.73 / (DROPRE ** 0.37))
               1345.           +               + (6.24E-3 * (DROPRE ** 0.38))
ISN     33     1346.               ELSE
ISN     34     1347.                  DRAGCO = (24.0 / DROPRE) + (4.536 / (DROPRE ** 0.368))
ISN     35     1348.               ENDIF
               1349.       C
               1350.       C  Calculate accelerations according to the equations of motion
               1351.       C
ISN     36     1352.               COEFF = 9.9225E-4 * (DRAGCO / DROPDI)
ISN     37     1353.               DVX   = COEFF * DSQRT(((VX - UX) **2 ) + ((VY - UY) ** 2))
               1354.           +           * (VX - UX)
ISN     38     1355.               DVY   = -COEFF * DSQRT(((VX - UX) ** 2) + ((VY - UY) ** 2))
               1356.           +           * (VY - UY)
               1357.       C
ISN     39     1358.               RETURN
ISN     40     1359.               END

*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2150 BYTES, PROGRAM NAME = FLOWXY    PAGE:   28.

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 8 ******
```

```
OPTIONS IN EFFECT:  NOLIST NOMAP NOXREF NOGOSTMT NODECK   SOURCE   TERM   OBJECT FIXED        NOTEST   SEQ
                      OPTIMIZE(0) LANGLVL(77) NOFIPS   FLAG(I)  NAME(MAIN  )  LINECOUNT(60)

                       *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

                        C *************************************************************
                        C *                                                           *
                        C *  SUBROUTINE KCALC - calculates a single element of the    *
                        C *                     matrix K for the control points I     *
                        C *                     and J                                 *
                        C *                                                           *
                        C *  INPUT DATA :                                             *
                        C *                   CO,SI    - direction angles of the segment *
                        C *                   DELTA    - half length of the control   *
                        C *                              element J                    *
                        C *                   XCI,XCJ  - x-coordinates of control points *
                        C *                              I and J , or of grid point I *
                        C *                              and control point J          *
                        C *                   YCI,YCJ  - y-coordinates of above       *
                        C *                                                           *
                        C *  OUTPUT VARIBLES :                                        *
                        C *                   K        - matrix element               *
                        C *                                                           *
                        C *  OTHER VARIABLES :                                        *
                        C *                   A,B      - geometrical quantities defined *
                        C *                              in Figure 2, Kennedy and Marsden *
                        C *                   R1SQ,    - squares of R1 and R2, also   *
                        C *                   R2SQ       defined in Figure 2 of K&M   *
                        C *                   TERM1,   - individual terms of the      *
                        C *                   TERM2,     expression for K             *
                        C *                   TERM3,                                  *
                        C *                   DENOM                                   *
                        C *                                                           *
                        C *************************************************************
                        C
ISN     1                     SUBROUTINE KCALC(XCJ,YCJ,XCI,YCI,DELTA,CO,SI,K)
                        C
ISN     2                     DOUBLE PRECISION A,B,CO,DELTA,DENOM,K,R1SQ,R2SQ,
                             +                 SI,TERM1,TERM2,TERM3,XCI,XCJ,YCI,YCJ
                        C
ISN     3                     PARAMETER (PI = 3.1415926535897932)
                        C
                        C  Calculate K
                        C
ISN     4                     B    = ((XCI-XCJ) * CO) + ((YCI-YCJ) * SI)
ISN     5                     A    = ((YCI-YCJ) * CO) - ((XCI-XCJ) * SI)
ISN     6                     R1SQ = (A*A) + ((B+DELTA)*(B+DELTA))
ISN     7                     R2SQ = (A*A) + ((B-DELTA)*(B-DELTA))
ISN     8                     TERM1 = (B + DELTA) * DLOG(R1SQ)
ISN     9                     TERM2 = (B - DELTA) * DLOG(R2SQ)
ISN    10                     DENOM = (A*A) + (B*B) - (DELTA*DELTA)
ISN    11                     IF (DENOM .LE. 1.0D-30) THEN
ISN    12                        IF (DABS(A) .LT. 1.0D-30) THEN
ISN    13                           TERM3 = 2.0 * A * PI
ISN    14                        ELSE
ISN    15                           TERM3 = 2.0 * A * (DATAN((B + DELTA) / A) -
                             +                 DATAN((B - DELTA) / A))
ISN    16                        ENDIF
```

```
                             *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*
ISN      17      1413.             ELSE
ISN      18      1414.                TERM3 = 2.0 * A * DATAN(2.0 * A * DELTA / DENOM)
ISN      19      1415.             ENDIF
                 1416.      C
ISN      20      1417.             K = (TERM1 - TERM2 + TERM3 - (4.0 * DELTA)) / (4.0 * PI)
                 1418.      C
ISN      21      1419.             RETURN
ISN      22      1420.             END

*STATISTICS*   SOURCE STATEMENTS = 22, PROGRAM SIZE = 1060 BYTES, PROGRAM NAME = KCALC    PAGE:   31.

*STATISTICS*   NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 9 ******
```

OPTIONS IN EFFECT: NOLIST NOMAP NOXREF NOGOSTMT NODECK SOURCE TERM OBJECT FIXED NOTEST SEQ
OPTIMIZE(0) LANGLVL(77) NOFIPS FLAG(I) NAME(MAIN) LINECOUNT(60)

```
                 *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

                 C **************************************************************
                 C *                                                            *
                 C *  SUBROUTINE COLLID   -  Calculates the x,y position        *
                 C *    of impact on the airfoil surface                        *
                 C *                                                            *
                 C *  INPUT VARIABLES:                                          *
                 C *                 XOUT,    - last droplet position outside   *
                 C *                 YOUT       the airfoil                     *
                 C *                 XIN,     - first droplet position inside   *
                 C *                 YIN        the airfoil                     *
                 C *                 XF1,XF2 - x and y coordinates of the end-  *
                 C *                 YF1,YF2   points of the surface segment    *
                 C *                           intercepted by the trajectory    *
                 C *  OUTPUT VARIABLES :                                        *
                 C *                 FINALX, - estimated coordinates of the     *
                 C *                 FINALY    impact point on the surface      *
                 C *                                                            *
                 C *  OTHER VARIABLES :                                         *
                 C *                 SLOPTR   - slope of the trajectory segment *
                 C *                            which crosses the airfoil surface *
                 C *                 SLOPSF   - slope of the surface segment    *
                 C *                            intercepted by the trajectory   *
                 C *                 INT1,    - intercepts for the linear       *
                 C *                 INT2       equations of the two segments   *
                 C *                                                            *
                 C *                                                            *
                 C **************************************************************
                 C
ISN      1              SUBROUTINE COLLID (XOUT,XIN,YOUT,YIN,XF1,YF1,XF2,YF2,
                      +                    FINALX,FINALY)
                 C
ISN      2              DOUBLE PRECISION XOUT,XIN,YOUT,YIN,XF1,XF2,YF1,YF2,
                      +                  SLOPTR,SLOPSF,INT1,INT2,FINALX,FINALY
                 C
                 C  Calculate slope of trajectory at impact point
                 C
ISN      3              SLOPTR = (YIN - YOUT) / (XIN - XOUT)
                 C
                 C  Calculate slope of surface segment at impact point
                 C
ISN      4              SLOPSF = (YF2 - YF1) / (XF2 - XF1)
                 C
                 C  Find intercepts of the two linear equations
                 C
ISN      5              INT1 = YIN - (SLOPTR * XIN)
ISN      6              INT2 = YF2 - (SLOPSF * XF2)
                 C
                 C  Find the impact point as the intersection point
                 C  of the two segments
                 C
ISN      7              FINALX = (INT2 - INT1) / (SLOPTR - SLOPSF)
ISN      8              FINALY = (SLOPTR * FINALX) + INT1
                 C
```

```
LEVEL 1.1.1 (DEC  81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:54      NAME: COLLID  PAGE:   34
                              *....*...1.........2.........3.........4.........5.........6.........7.*.......8.........9.........*

ISN        9      1474.            RETURN
ISN       10      1475.            END

*STATISTICS*   SOURCE STATEMENTS = 10, PROGRAM SIZE = 512 BYTES, PROGRAM NAME = COLLID    PAGE:   33

*STATISTICS*    NO DIAGNOSTICS GENERATED.

****** END OF COMPILATION 10 ******
```

```
LEVEL 1.1.1 (DEC  81)         VS FORTRAN        DATE: JUL 24, 1986    TIME: 13:15:54      NAME: MAIN    PAGE: 35

SUMMARY OF MESSAGES AND STATISTICS FOR ALL COMPILATIONS

*STATISTICS*   SOURCE STATEMENTS = 162, PROGRAM SIZE = 21362 BYTES, PROGRAM NAME = MAIN    PAGE:   1
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 1 ******
*STATISTICS*   SOURCE STATEMENTS = 62, PROGRAM SIZE = 7298 BYTES, PROGRAM NAME = DRAW    PAGE:   12
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 2 ******
*STATISTICS*   SOURCE STATEMENTS = 9, PROGRAM SIZE = 370 BYTES, PROGRAM NAME = ONEBET    PAGE:   15
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 3 ******
*STATISTICS*   SOURCE STATEMENTS = 13, PROGRAM SIZE = 608 BYTES, PROGRAM NAME = SPCBET    PAGE:   16
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 4 ******
*STATISTICS*   SOURCE STATEMENTS = 25, PROGRAM SIZE = 926 BYTES, PROGRAM NAME = DROPS    PAGE:   17
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 5 ******
*STATISTICS*   SOURCE STATEMENTS = 44, PROGRAM SIZE = 122570 BYTES, PROGRAM NAME = TRAJEC    PAGE:   21
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 6 ******
*STATISTICS*   SOURCE STATEMENTS = 39, PROGRAM SIZE = 16358 BYTES, PROGRAM NAME = POFLO    PAGE:   25
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 7 ******
*STATISTICS*   SOURCE STATEMENTS = 40, PROGRAM SIZE = 2150 BYTES, PROGRAM NAME = FLOWXY    PAGE:   28
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 8 ******
*STATISTICS*   SOURCE STATEMENTS = 22, PROGRAM SIZE = 1060 BYTES, PROGRAM NAME = KCALC    PAGE:   31
*STATISTICS*   NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 9 ******
*STATISTICS*   SOURCE STATEMENTS = 10, PROGRAM SIZE = 512 BYTES, PROGRAM NAME = COLLID    PAGE:   33
```

```
LEVEL 1.1.1 (DEC   81)          VS FORTRAN          DATE: JUL 24, 1986     TIME: 13:15:54      NAME: MAIN     PAGE:    36
*STATISTICS*     NO DIAGNOSTICS GENERATED.
****** END OF COMPILATION 10 ******
******* SUMMARY STATISTICS ******* 0 DIAGNOSTICS GENERATED. HIGHEST SEVERITY CODE IS 0.
```

Appendix VI

The MVD Approximation for Collision Efficiency

The task of computing volume-weighted averages of collision efficiencies for a droplet size spectrum is often simplified by substituting a monodisperse size spectrum at the median volume diameter of the distribution. As discussed in Chapter III, this can lead, in some cases, to a significant underestimation of the maximum accretion angle. However, it has often been assumed that the weighted overall and stagnation line collision efficiencies are well approximated by the mvd values. In this Appendix the validity of that assumption is examined.

Table A.1 lists a variety of atmospheric and marine droplet size spectra, as relative numbers of droplets in 10 or 3μm wide size bins. These spectra were either measured in the field, or in icing wind tunnels, or estimated from empirical parameterized distributions, and cover a range of types and mvds.

They are derived from the following sources: Bain and Gayet (1984), Battan and Reitan (1957), Best (1951), Khrgian-Mazin, as reported by Pruppacher and Klett (1980), McComber and Touzot (1981), Preobrazhenskii (1973), and Squires (1958). Those marked FROST were measured by the oil

mid-band droplet diameter	relative number of drops per unit volume						
	Khrgian-Mazin 1	Khrgian-Mazin 2	Khrgian- azin 3	Khrgian-Mazin 4	Squires Hawaiian	Squires dark stratus	Squires Trade-wind Cu
5µm	0.724	0.069	0.014	0.002	0.25	6.5	9
15	2.397	0.340	0.082	0.012	0.65	6.0	16
25	2.449	0.518	0.148	0.024	1.0	5.4	29
35	1.766	0.557	0.189	0.035	0.8	5.1	21
45	1.074	0.505	0.203	0.043	0.8	2.5	3.5
55	0.590	0.414	0.198	0.047	0.7	0.5	
65	0.303	0.317	0.180	0.049	0.35	0.2	
75	0.149	0.232	0.156	0.048	0.15	0.1	
85	0.070	0.164	0.131	0.046	0.10	0.05	
95	0.032	0.112	0.106	0.042	0.06		
105	0.014	0.075	0.085	0.038	0.05		
115	0.006	0.049	0.066	0.034	0.04		
125	0.003	0.032	0.051	0.030	0.03		
135	0.001	0.021	0.039	0.026	0.04		
145		0.013	0.029	0.022	0.01		
155		0.008	0.022	0.019			
165		0.005	0.016	0.016			
175		0.003	0.012	0.013			
185		0.002	0.009	0.011			
195		0.001	0.006	0.009			
205			0.004	0.007			
215			0.003	0.006			
225				0.005			
235				0.004			
245				0.003			

Table A.1 - Droplet size spectra from many different sources (see text for references).

relative number of drops per unit volume

mid-band droplet diameter	Battan and Reitan fairweather cumulus	Battan and Reitan stratus	FROST 1	FROST 2	FROST 3	FROST 4	FROST 5	Preobrazhenskii
5µm	600	100	127	94	97	3	17	5180
15	70	15	327	167	213	37	68	1620
25	13	1.1	104	33	86	63	64	480
35	0.26	0.2	13	6	33	58	31	200
45			1	4	14	51	12	110
55					2	35	11	56
65					1	26	9	32
75						17	3	21
85						14	2	14
95						16	4	10
105						12	9	7
115						5	2	5
125						4	3	4
135						8	2	3
145						7	1	2
155						2	0	1
165						1	2	
175						2		
185						2		

Table A.1 - continued.

relative number of drops per unit volume

mid-band droplet diameter	McComber and Touzot	mid-band droplet diameter	Bain and Gayet 1	Bain and Gayet 2	Bain and Gayet 3
5.1µm	945	1.5	20	7	13.0
15.3	1093	4.5	41	12	15.5
25.5	481	7.5	30	23	15.5
35.7	211	10.5	7.5	33	18.0
45.9	74	13.5	2.5	18	15.5
56.1	38	16.5		5	12.5
66.3	30	19.5		2	6.0
76.5	13	22.5			2
86.7	19	25.5			2
96.9	11				
107.1	10				
117.3	3				
127.5	1				
137.7	1				

percent volume

mid-band droplet diameter	Best 1	Best 2	Best 3	Best 4	Best 5
5µm	63.2	22	4	3	36
15	36.8	41	24	31	47
25		26	39	53	15
35		9	26	13	2
45		2	6		
55			1		

Table A.1 - continued.

slide technique in the FROST icing wind-tunnel facility at The University of Alberta.

For each spectrum, the mvd was calculated following the method derived by Lozowski (1978). Assuming a uniform distribution of droplets within each bin, the partial volume of each is given by:

$$V_i = (n_i/24)\ \pi\ w^3\ (i^4 - (i-1)^4), \qquad \text{A.8}$$

where n_i is the number of drops in the ith bin, and w is the bin width, say 10 μm. Then, if V is the total volume of all droplets, let

$$u_i = (\Sigma V_i)\ /\ V. \qquad \text{A.9}$$

Now find u_k such that $u_{k-1} < 0.5$, and $u_k > 0.5$. Finally,

$$\text{mvd} = w\ (((0.5 - u_{k-1})\ /\ (u_k - u_{k-1}))\ (k^4 - (k-1)^4) + (k-1)^4)^{0.25}, \qquad \text{A.10}$$

Overall and stagnation line collision efficiencies have been calculated for the mvd of each spectrum. For the complete spectrum, weighted average collision efficiencies are calculated using the fractional volume of each bin as the weighting factor for the collision efficiency of the mvd of that bin.

All of the collision efficiencies are calculated according to Table II.6, and for the same conditions, i.e. U = 10 m/sec and D_c = 0.0254 m. The results are shown in Table A.2 and Figures A.2 and A.3. For a different type of comparison, the conditions have been varied for a single spectrum (labelled as Bain and Gayet 2 in Table A.1) to give a complete range of collision efficiencies. These results are compared in Table A.3 and in Figures A.4 and A.5.

Also calculated are collision efficiencies for the droplet at the 42nd volume percentile (d42), on a suggestion made by Lozowski and Makkonen (private communication, 1986), based on a theoretical analysis using Gaussian quadrature to estimate the overall collision efficiency for a droplet spectrum. These data are included in [illegible] figures and tables listed above.

Comparison of Figure A.[illegible] with Figure 1 of Makkonen (1984), which shows the same type of calculations for the same spectrum used here as well as two other spectra given by Bain and Gayet, 1982), reveals a major disagreement in the sign of $E_{spec} - E_{mvd}$. The disagreement is due to a misinterpretation of the original droplet spectrum diagram in Bain and Gayet (1982) on the part of Makkonen (private communication, 1986). Figure A.4 reflects the correct interpretation.

Spectrum	β_{spec}	E_{spec}	mvd	β_{mvd}	E_{mvd}	d42	β_{d42}	E_{d42}
Khrgian-Mazin 1	0.824	0.710	56.9μm	0.852	0.745	52.1μm	0.835	0.718
Khrgian-Mazin 2	0.906	0.836	93.6	0.924	0.862	86.1	0.914	0.846
Khrgian-Mazin 3	0.936	0.886	126.2	0.949	0.906	116.5	0.943	0.895
Khrgian-Mazin 4	0.955	0.919	165.9	0.964	0.933	154.5	0.960	0.927
Squires Hawaiian	0.886	0.804	81.4	0.908	0.835	70.9	0.889	0.804
Squires dark stratus	0.767	0.626	43.0	0.790	0.650	39.7	0.769	0.619
Squires Trade-wind Cu	0.702	0.532	34.9	0.730	0.565	33.2	0.714	0.543
Battan and Reitan Cu	0.393	0.224	18.0	0.459	0.249	16.6	0.419	0.211
Battan and Reitan stratus	0.364	0.199	17.2	0.434	0.225	15.9	0.396	0.191
FROST 1	0.555	0.362	24.7	0.603	0.402	22.6	0.565	0.357
FROST 2	0.583	0.402	26.2	0.628	0.432	23.4	0.580	0.375
FROST 3	0.680	0.516	35.0	0.731	0.566	31.2	0.694	0.515
FROST 4	0.928	0.873	127.7	0.949	0.907	111.3	0.939	0.889
FROST 5	0.919	0.859	112.3	0.940	0.890	107.1	0.936	0.883
Preobrazhenskii	0.838	0.745	78.5	0.903	0.827	67.9	0.883	0.793
McComber and Touzot	0.836	0.736	75.9	0.899	0.820	65.8	0.878	0.786
Bain and Gayet 1	0.108	0.036	8.8	0.088	0.010	8.4	0.065	0.003
Bain and Gayet 2	0.284	0.123	13.3	0.301	0.118	12.5	0.267	0.096
Bain and Gayet 3	0.421	0.232	17.5	0.444	0.233	16.6	0.418	0.210
Best 1	0.415	0.247	18.3	0.467	0.255	17.0	0.429	0.220
Best 2	0.582	0.401	26.9	0.639	0.445	25.0	0.610	0.409
Best 3	0.542	0.350	24.4	0.599	0.397	22.5	0.564	0.357
Best 4	0.313	0.160	15.3	0.375	0.173	13.1	0.291	0.111
Best 5	0.186	0.070	9.4	0.120	0.021	9.0	0.098	0.014
standard error				0.031	0.029		0.027	0.026

Table A.2 Overall and stagnation line collision efficiencies as plotted in Figures A.2 and A.3.

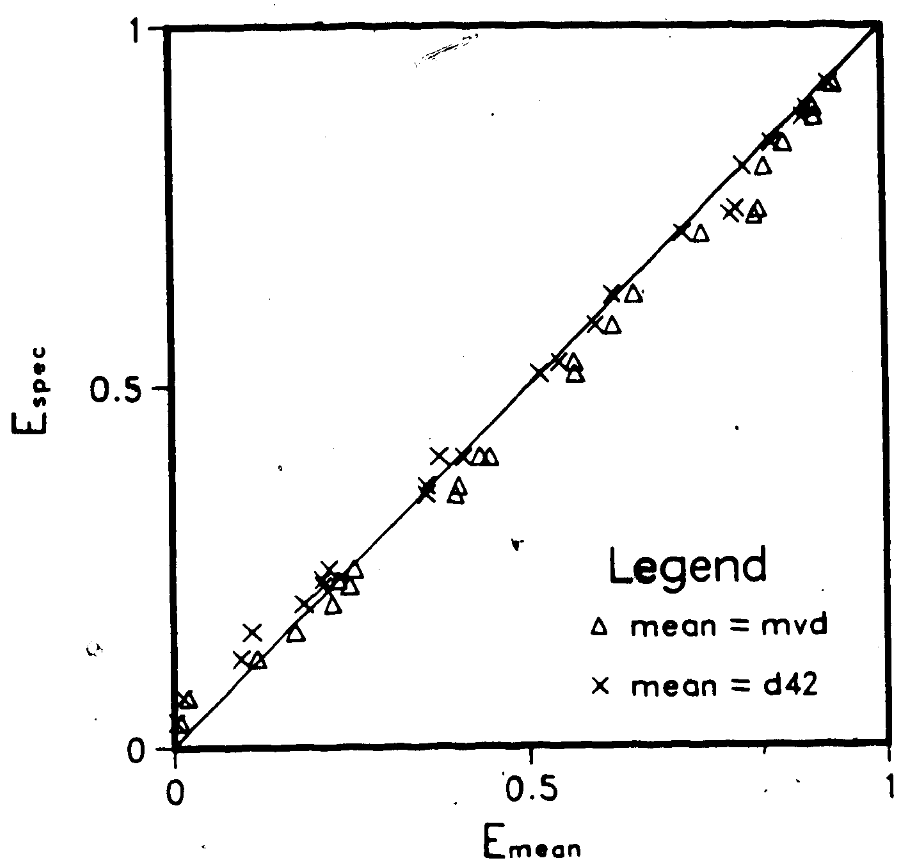


Figure A.2 - Overall collision efficiencies calculated for many different drop size spectra vs. E's calculated for the mvd and d42 of each spectrum. In each case U=10 m/sec, D_c=0.0254 m.

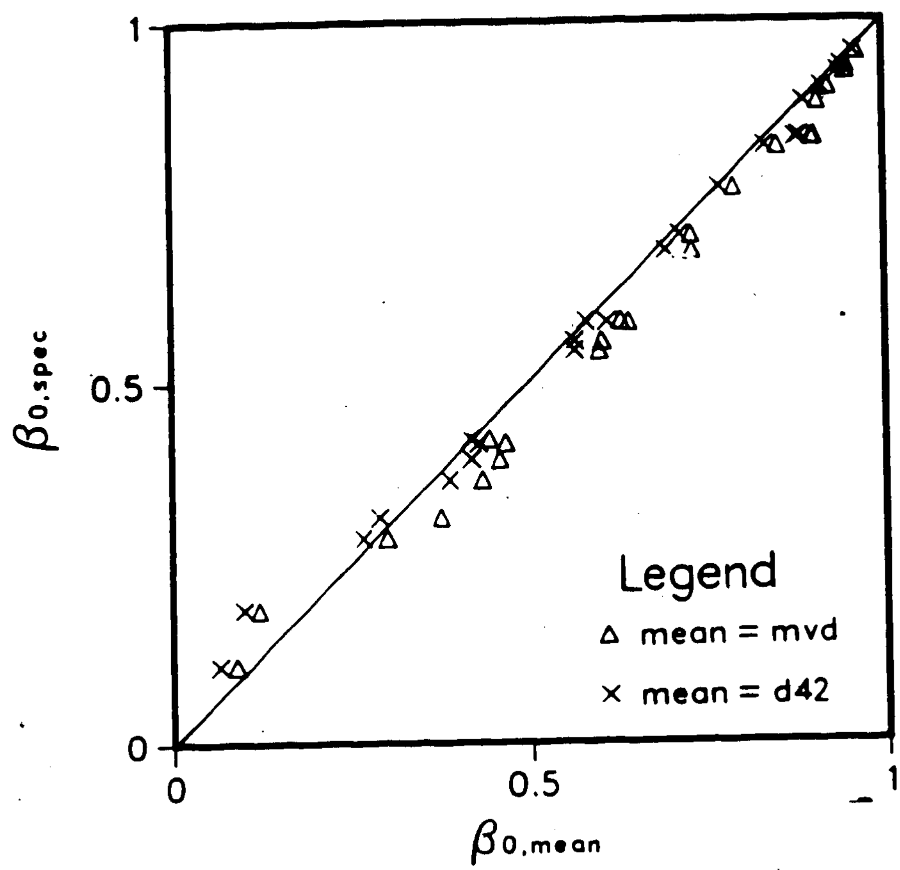


Figure A.3 - Stagnation line collision efficiencies calculated for many different drop size spectra vs. β_0's calculated for the mvd and d42 of each spectrum. In each case U=10 m/sec, D_c=0.0254 m.

U	$\beta_{o\,spec}$	E_{spec}	$\beta_{o\,mvd}$	E_{mvd}	$\beta_{o\,d42}$	E_{d42}
5m/sec	0.121	0.037	0.121	0.022	0.089	0.010
7	0.202	0.072	0.208	0.062	0.175	0.044
10	0.284	0.123	0.301	0.118	0.267	0.096
15	0.383	0.194	0.403	0.196	0.370	0.169
20	0.448	0.250	0.470	0.257	0.439	0.228
25	0.496	0.295	0.518	0.305	0.489	0.275
30	0.532	0.331	0.555	0.344	0.527	0.314
40	0.586	0.389	0.608	0.405	0.582	0.374
50	0.624	0.432	0.646	0.450	0.622	0.420
80	0.693	0.516	0.713	0.536	0.692	0.508
200	0.790	0.648	0.806	0.668	0.791	0.646
350	0.830	0.708	0.844	0.726	0.832	0.708
700	0.865	0.765	0.877	0.781	0.867	0.766
1500	0.892	0.811	0.901	0.825	0.894	0.812
standard error			0.012	0.013	0.011	0.011

Table A.3 - Overall and stagnation line collision efficiencies as plotted in Figures A.4 and A.5.

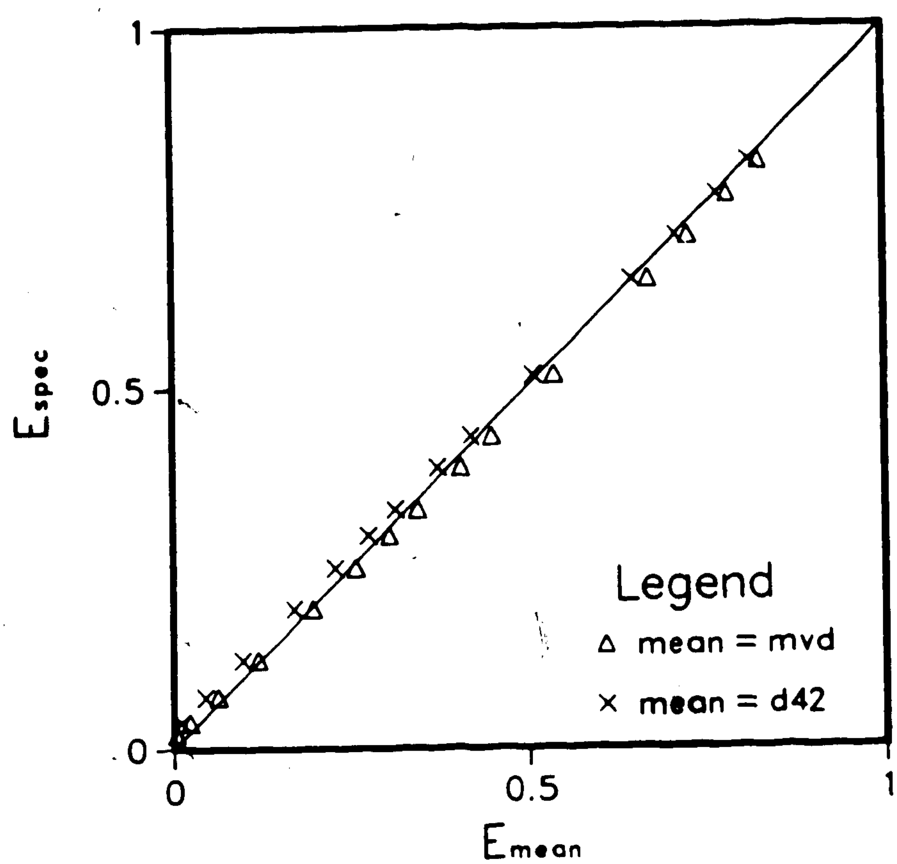


Figure A.4 - Overall collison efficiencies calculated for a single drop size spectrum (Bain and Gayet 2) at many different wind speeds, vs. E's calculated for the mvd and d42 of the same spectrum for the same wind speeds. D_c = 0.0254 m.

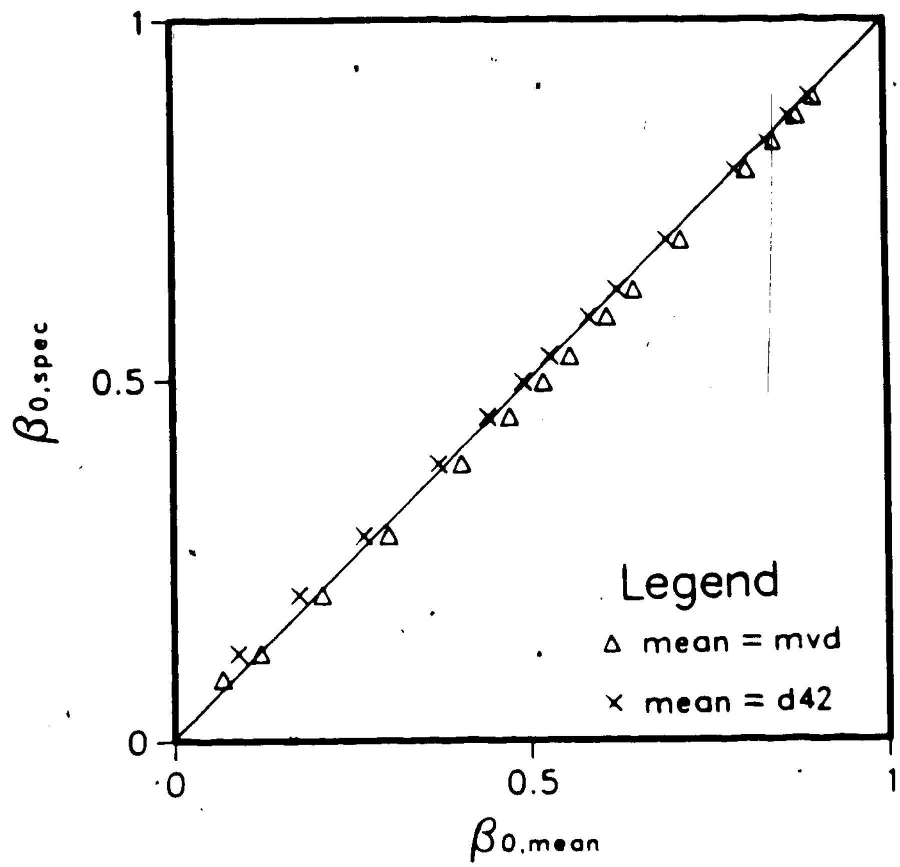


Figure A.5 - Stagnation line collision efficiencies calculated for a single drop size spectrum (Bain and Gayet 2) at many different wind speeds, vs. β_0's calculated for the mvd and d42 of the same spectrum for the same wind speeds. D_c = 0.0254 m.

The conclusion which may be made from this exercise is that, although the standard errors are not significantly different over the entire range of collision efficiencies (see Tables A.2 and A.3), in the range of E or β_0 less than about 0.2 the mvd approximation is preferable. Above that range, for most types of spectra, the d42 approximation gives better agreement with the spectrum-weighted collision efficiencies. However, for certain types of spectra which depart significantly from a Gaussian distribution (e.g. the double peaked spectrum from McComber and Touzot) even the d42 value may be in error by more than 10 percent.

References

Bain, M. and Gayet, J-F. 1982: Contribution to the Modelling of the Ice Accretion Process: Ice Density Variation with the Impacted Surface Angle. Proceedings, First International Workshop on Atmospheric Icing of Structures, Hanover, 13-20.

Battan, L.J. and Reitan, C.H. 1957: Droplet Size Measurements in Convective Clouds. Artificial Stimulation of Rain, Weickmann and Smith, ed.s. London, Pergamon Press, p.184.

Best, A.C. 1951: Drop-size Distribution in Cloud and Fog. *Quart. Jour. Roy. Meteor. Soc.* 77, 418-426.

Lozowski, E.P. 1978: Stochastic Effects in Spray Droplet Sampling with Oiled Slides. National Research Council, Division of Mechanical Engineering, Laboratory Memorandum LT-172, 17 pp.

Makkonen, L. 1984: Modeling of Ice Accretion on Wires. *J. Climate and App. Meteor.*, 23, 929-939.

McComber, P. and Touzot, G. 1981: Calculation of the Impingement of Cloud Droplets in a Circular Cylinder by the Finite Element Method. *J. Atmos.*

Sci. **38**, 1027-1036.

Preobrazhenskii, L.Yu. 1973: Estimate of the Content of Spray Drops in the Near-Water Layer of the Atmosphere. *Fluid Mechanics - Soviet Research*, **2**, 95-100.

Pruppacher, H.R. and Klett, J.D. 1980: Microphysics of Clouds and Precipitation. D. Reidel , p.11.

Squires, P. 1958: The Microstructure and Colloidal Stability of Warm Clouds: I. The Relation between Structure and Stability. *Tellus*, **10**, 256-261.