NASA-CR-189134

IN-03

72159

P-121

(NASA-CR-189134) PARTICLE TRAJECTORY COMPUTER PROGRAM FOR ICING ANALYSIS OF AXISYMMETRIC BODIES Final Report (FWG Associates) 121 p

N92-19276

CSCL 01C

Unclas

G3/03 0072159

FWG

FWG ASSOCIATES, INC.

"Continuity with the Future"

Prepared by:

Walter Frost
Ho-Pen Chang
Kenneth R. Kimble

Prepared for:

NASA
Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

FINAL REPORT

Contract NAS3-22448

PARTICLE TRAJECTORY COMPUTER PROGRAM FOR ICING ANALYSIS OF AXISYMMETRIC BODIES

April 21, 1982

Approved:

Walter Frost, President
FWG Associates, Inc.

ACKNOWLEDGMENTS

The work reported herein was supported by the National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio, under Contract NAS3-22448. The authors give their special thanks to Dr. Robert Joseph Shaw of Lewis Research Center who was the scientific monitor of the program and who provided considerable technical advice and input to the project.

TABLE OF CONTENTS

[redacted]

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Symbol	Definition
A	Reference area of particle
Be	Best number ($Be = C_d Re^2$)
C_d	Drag coefficient
C_{do}	Steady-state drag coefficient
$C_{d\alpha}$	Drag coefficient due to angle of attack
C_ℓ	Lift coefficient
$C_{\ell o}$	Steady-state lift coefficient
$C_{\ell\alpha}$	Lift coefficient due to angle of attack
C_m	Moment coefficient
C_{mo}	Steady-state moment coefficient
$C_{m\alpha}$	Moment coefficient due to angle of attack
D	Particle diameter
$\tilde{D}$	Aerodynamic drag force
E	Total collection efficiency
FRL	Flight reference line
g	Accleration of gravity (9.8 m/s^2)
g_c	Conversion factor (1 $kg\text{-}m/newton\text{-}sec^2$)
h	Projected height of the body along the vertical coordinate line
I_{zz}	Moment of inertia of mass relative to the z axis
$\tilde{L}$	Aerodynamic lift force
LWC	Liquid water content
ℓ_c	Reference length of body

Symbol	Definition
M	Moment of aerodynamic forces acting on the particle
m	Mass of water droplet (kg)
$\dot{m}$	Mass flow rate
n_i	Number of particles of size i per unit volume
Re	Reynolds number based on the diameter of the particle
s	Surface distance measured from the leading edge of each body; positive along the lower surface and negative along the upper surface
S_L	Lower surface impingement limit
S_U	Upper surface impingement limit
t	Time
$\vec{V}$	Velocity of particle
$\vec{V}_a$	Velocity of particle relative to flow field
$\vec{W}$	Velocity of flow field
W_x	x-component velocity of flow field
W_y	y-component velocity of flow field
W_∞	Free-stream velocity
x	x-coordinate of particle
$\dot{x}$	x-component velocity of particle
$\ddot{x}$	x-component acceleration of particle
x_o	Initial value of the horizontal coordinate of particle
y	y-coordinate of particle
$\dot{y}$	y-component velocity of particle
$\ddot{y}$	y-component acceleration of particle
y_o	Initial value of the vertical coordinate of particle
y_{ou}	Upper tangent trajectory of the particle corresponding to S_U

Symbol	Definition
$y_{o\ell}$	Lower tangent trajectory of the particle corresponding to S_L

Greek Symbols

α	Angle of attack
β	Local collection efficiency
γ	Particle path angle $\left(\gamma = \tan^{-1} \frac{\dot{y} - W_y}{\dot{x} - W_x}\right)$
δ	The angle between $\vec{V}$ and $\vec{V}_a$
η_i	Percentage liquid water contained in particles of size D_i
θ	Pitch angle of particle
$\dot{\theta}$	Angular velocity of particle
$\ddot{\theta}$	Angular acceleration of particle

1.0 INTRODUCTION

General aviation aircraft and helicopters exposed to an icing environment can accumulate ice resulting in a sharp increase in drag and reduction of maximum lift causing hazardous flight conditions. NASA Lewis Research Center (LeRC) is conducting a program to examine , with the aid of high-speed computer facilities, how the trajectories of particles contribute to the ice accumulation on airfoils and engine inlets. This study, as part of the NASA/LeRC research program, develops a computer program for the calculation of icing particle trajectories and impingement limits relative to axisymmetric bodies in the leeward-windward symmetry plane.

The methodology employed in the current particle trajectory calculation is to integrate the governing equations of particle motion in a flow field computed by the Douglas axisymmetric potential flow program [1]. The three-degrees-of-freedom (horizontal, vertical, and pitch) motion of the particle is considered. The particle is assumed to be acted upon by aerodynamic lift and drag forces, gravitational forces, and, for nonspherical particles, aerodynamic moments. The particle momentum equation is integrated to determine the particle trajectory. Derivation of the governing equations and the method of their solution are described in Section 2.0.

General features, as well as input/output instructions for the particle trajectory computer program, are described in Section 3.0. The details of the computer program are described in Section 4.0. Examples of the calculation of particle trajectories demonstrating application of the trajectory program to given axisymmetric inlet test cases are presented in Section 5.0. For the examples presented, the particles are treated as spherical water droplets. In Section 6.0, limitations of the program relative to excessive computer time and recommendations in this regard are discussed.

2.0 METHODOLOGY

The procedure for computing the particle trajectory around an axisymmetric body has been divided into the following steps:

1. Compute the potential field around an axisymmetric body assuming the particles do not influence the flow field.
2. Generate grids around the body to satisfy the refinement level and velocity error criteria which are input by the user.
3. Determine the velocity flow field on the grids around the body with or without angle of attack for a given free-stream airspeed.
4. Calculate the trajectories of droplets in the flow field determined in Step 3 using the Adams-Moulton predictor-corrector method to integrate the equations of particle motion.
5. Calculate the local collection efficiency for the body.

The computational procedures used in Step 1 are described in Section 2.1, those used in Steps 2 and 3 are discussed in Section 2.2, and those used in Steps 4 and 5 are described in Section 2.3.

2.1 Potential Flow About an Axisymmetric Body

The Douglas axisymmetric potential flow program developed by Hess and Smith [2] is used in the present study for calculating the flow field about the body. This computer program uses a distribution of sources, sinks, and/or vortices along the body surface to calculate the potential flow field. The body surface is represented by an arbitary number of straight line (or curve) segments. In calculating the flow field, contributions from all the sources, sinks, and/or vortices are summed. The accuracy of this method was tested by comparing its predicted velocities and surface pressure coefficients with both analytic solutions and experimental data [2]. Excellent agreement has been found.

The Douglas axisymmetric flow program consists of essentially three parts: a geometry-generating program called SCIRCL and an axisymmetric flow field computer program called EOD. The SCIRCL program generates the geometry input to the potential flow program (EOD) for a given specified analytical shape. Using the input from SCIRCL the EOD program calculates the flow field at any position in space. The EOD program is used to generate an input data tape containing the sources, sinks, and vortices which are then used to compute the velocity at each time step along the particle trajectory.

Only limited details of the potential flow program are provided in this report since the major thrust of this study was to integrate the axisymmetric potential flow program into the program for computing particle trajectories and local collection efficiencies documented in [3]. Details of the particle trajectory program is reproduced in this section for completeness. The already-developed Douglas potential flow program is simply used to provide flow field input [1]. For complete details of the potential flow program, the user should consult References 1 and 2.

2.2 Particle Equations of Motion

In the present study the motion of a particle has been analyzed as a point mass particle which is acted on by the potential flow field but which itself does not influence the flow. The forces acting on the particle are considered to be those of lift, drag, and gravity. Pitch moments acting on the particle are also considered. Figure 2.1 shows the forces acting on the particle and the velocity vectors relative to the motion of the particle. The flight reference line (FRL) shown in Figure 2.1 is not significant for a spherical particle; however, for more arbitrarily shaped particles, e.g., a snow flake, the FRL must be defined relative to the lift, drag, and moment coefficient data available for the given particle shape. The governing equations of the particle motion are derived for the more general case, i.e., arbitrary particle shapes. In turn, the computer program developed in this study and described in Sections 3.0 and 4.0 provides the option for general-shaped particles. However, the test cases given in Section 5.0 are for spherical particles only. A valid drag law for spherical particles is

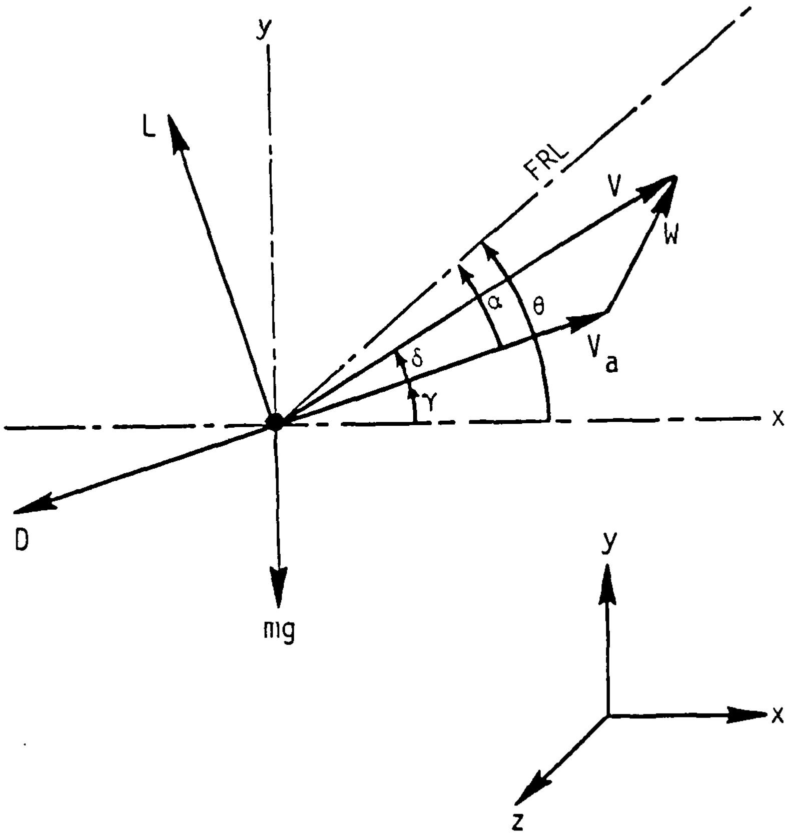


Figure 2.1 Diagram of the velocity vectors and forces acting on a point mass particle.

built into the computer program. The user must input lift, drag, and moment coefficient data for general-shaped particles.

The equations of motion of the particle derived from a force balance on a point mass particle as shown in Figure 2.1 are:

$$m\ddot{x} = -\tilde{D} \cos \gamma - \tilde{L} \sin \gamma \tag{2.1}$$

$$m\ddot{y} = -\tilde{D} \sin \gamma + \tilde{L} \cos \gamma - mg \tag{2.2}$$

where

$$\gamma = \tan^{-1} \frac{\dot{y} - W_y}{\dot{x} - W_x} \tag{2.3}$$

The flow field velocity components in the longitudinal and radial directions, i.e., W_x and W_y, respectively, are obtained from the potential flow program described in Section 2.1. The aerodynamic drag and lift forces are defined as:

$$\tilde{D} = C_d \frac{\rho_a V_a^2}{2g_c} A$$

$$\tilde{L} = C_\ell \frac{\rho_a V_a^2}{2g_c} A$$

where A is a characteristic area of the particle, ρ_a is the density of air at the position of the particle, and V_a is the particle velocity relative to the flow field and is defined as:

$$V_a = \sqrt{(\dot{x} - W_x)^2 + (\dot{y} - W_y)^2} \tag{2.4}$$

For arbitarily shaped particles, expressions for the drag coefficient, C_d, and lift coefficient, C_ℓ, must be provided by the user. They are often approximated with:

$$C_d = C_{do} + C_{d\alpha}\alpha \tag{2.5}$$

$$C_\ell = C_{\ell o} + C_{\ell\alpha}\alpha \tag{2.6}$$

where α is the angle between the FRL and the velocity vector $\vec{V}_a$ (Figure 2.1).

Equations 2.1 and 2.2 can be solved given values of the coefficients of Equations 2.5 and 2.6, i.e., C_{do}, $C_{\ell o}$, $C_{d\alpha}$, and $C_{\ell\alpha}$. The angle α is computed from the expression:

$$\alpha = \theta - \gamma \tag{2.7}$$

where the angle θ, generally called pitch angle in aerodynamics, is governed by:

$$\ddot{\theta} = \frac{M}{I_{zz}} \tag{2.8}$$

where I_{zz} is the moment of inertia of mass relative to the z axis. The moment of aerodynamic forces acting on the particle is:

$$M = C_m \frac{\rho_a V_a^2}{2g_c} A\ell_c \qquad (2.9)$$

where C_m is approximately:

$$C_m = C_{mo} + C_{m\dot{\alpha}}\dot{\alpha}$$

C_{mo} and $C_{m\dot{\alpha}}$ are constants to be provided by the user depending upon the shape of the particle, and ℓ_c is a reference length.

For a spherical particle with zero angular velocity, the lift force is always zero for potential flow. The governing equations are significantly simplified in this case:

$$m\ddot{x} = -\tilde{D} \cos \gamma \qquad (2.10)$$

$$m\ddot{y} = -\tilde{D} \sin \gamma - mg \qquad (2.11)$$

where

$$\tilde{D} = C_{do} \frac{\rho_a V_a^2}{2g_c} A \qquad (2.12)$$

In the present study C_{do} is the steady-state drag coefficient of a sphere as a function of Reynolds number based on particle diameter, Re = $V_a D/\nu$, as shown in Figure 2.2. The diameter of the spherical particle, D, and the kinematic viscosity of the air, ν, are assumed constant along the particle's trajectory in the present study. The drag law utilized was provided by NASA/LeRC (Figure 2.2 and Table 2.1) [5].

The governing Equations 2.1, 2.2, and 2.8 together with the following definitions form a complete set of equations to describe the motion of particles in the flow fields.

$$\frac{dx}{dt} = \dot{x} \qquad (2.13)$$

$$\frac{dy}{dt} = \dot{y} \qquad (2.14)$$

$$\frac{d\theta}{dt} = \dot{\theta} \qquad (2.15)$$

Integration of the Equations 2.1, 2.2, 2.8, 2.13, 2.14, and 2.15 results

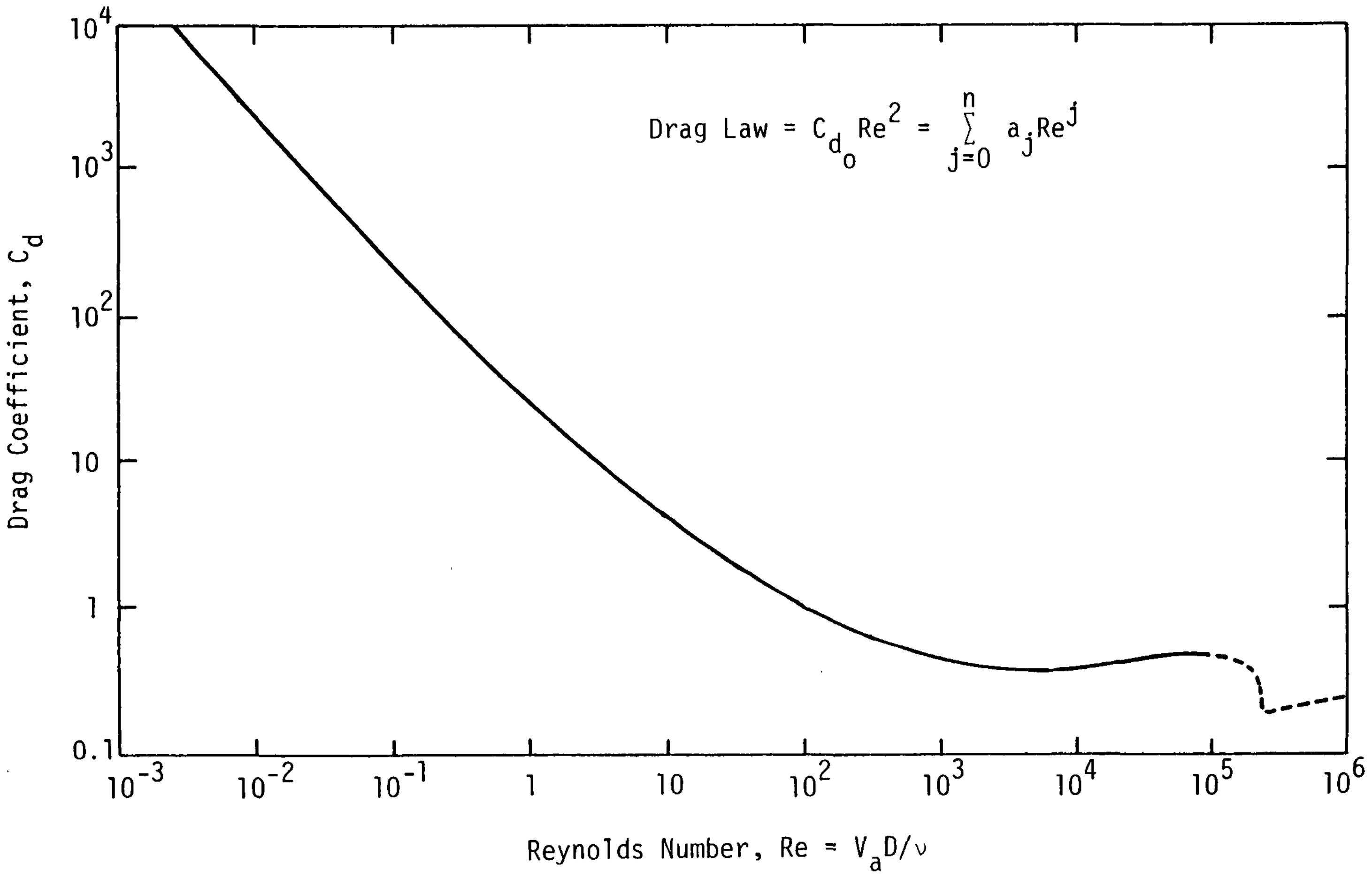


Figure 2.2 Friction (or drag coefficient) for spheres moving relative to a fluid with a velocity, V_∞ [5].

TABLE 2.1 Polynomial Coefficients Relating Best Number (Be) to Reynolds Number for Spherical Particles [5].

$$Be = \sum_{j=0}^{n} a_j Re^j$$

Reynolds Number Range	j	a_j
$0.05 < Re \leq 3$	0	0.0
	1	24.167
	2	3.254
	3	-0.23564
$3 < Re \leq 330$	0	-28.339
	1	38.969
	2	0.73204
	3	-0.00056084
$330 < Re$	0	0.0
	1	93.462
	2	0.37576

where

$$Re = \frac{|\vec{W} - \vec{V}|D}{\nu}$$

Be = Best number ($C_d Re^2$)

D = particle diameter

$\vec{W}$ = inertial velocity of the air ($\vec{W} = W_x\vec{i} + W_y\vec{j}$)

$\vec{V}$ = inertial velocity of the particle ($\vec{V} = \dot{x}\vec{i} + \dot{y}\vec{j}$)

in solutions for $\dot{x}$, $\dot{y}$, $\dot{\theta}$, x, y, θ of the particle at time t. For the sphere particles for which the lifting force is omitted, only Equations 2.10, 2.11, 2.13, and 2.14 are integrated. The integration method is described briefly in the next section.

2.3 Particle Trajectory Solution Algorithm

The method utilized for integrating the governing equations of particle motion is the Adams-Moulton predictor-corrector method [6]. The solution is obtained if the summation of the difference between the particle velocities in the x and y directions obtained by the predictor and corrector, respectively, and divided by the value of the solution at the last time step is less than a specified number ε:

$$\left|\frac{(\dot{X}_{n,predictor} - \dot{X}_{n,corrector})}{\dot{X}_{n-1}}\right| + \left|\frac{(\dot{Y}_{n,predictor} - \dot{Y}_{n,corrector})}{\dot{Y}_{n-1}}\right| \leq \varepsilon$$

The value of ε is specified by the user.

2.4 Computation of Collection Efficiency

Particle trajectories calculated as described in the previous sections are used to establish the relations between the particle's initial position $(-\infty, y_0)$ and the position where it impinges on the body surface, s. s is the length along the body surface measured from the leading edge on the body to the point of particle impingement. The value of s is defined as positive on the lower surface and negative on the upper surface. y_0 is the initial value of the vertical coordinate from which the particle is released (see Figure 2.3). The local collection efficiency, β, is calculated as a function of the distance along the body surface by differentiating $y_0 = y_0(s)$ with respect to s [7]:

$$\beta = - \frac{dy_0}{ds} \tag{2.16}$$

The minus sign is introduced so that β is positive, which is consistent with the definition given in Reference 4.

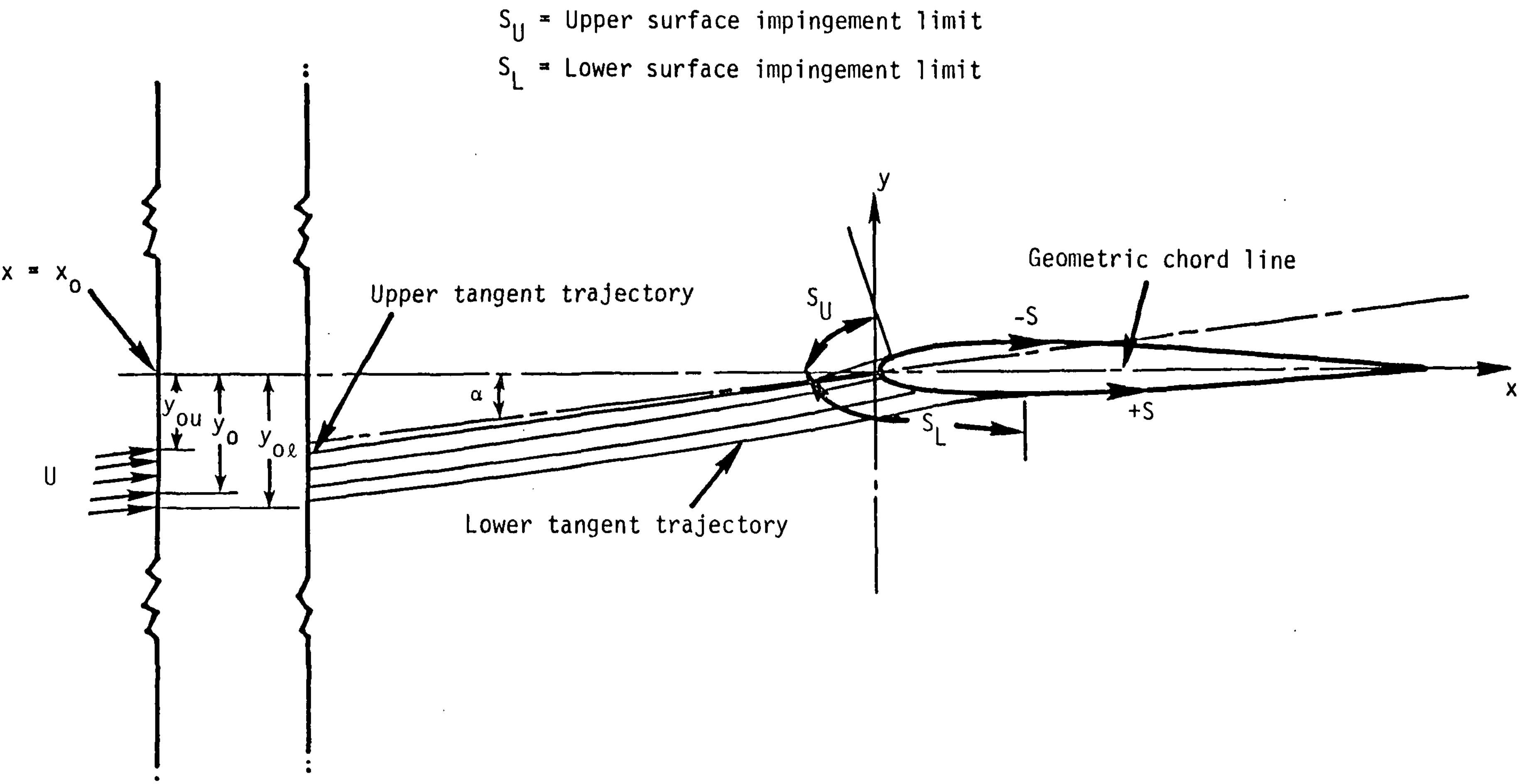


Figure 2.3 Illustration of impingement terminology and water droplet trajectory in an airfoil flow field [7].

The overall collection efficiency, E, is defined as:

$$E = \frac{y_{ou} - y_{o\ell}}{h} \tag{2.17}$$

where y_{ou} and $y_{o\ell}$ are the upper and lower tangent trajectories of the particle relative to the body surface and h is the projected height of the body along the vertical coordinate line.

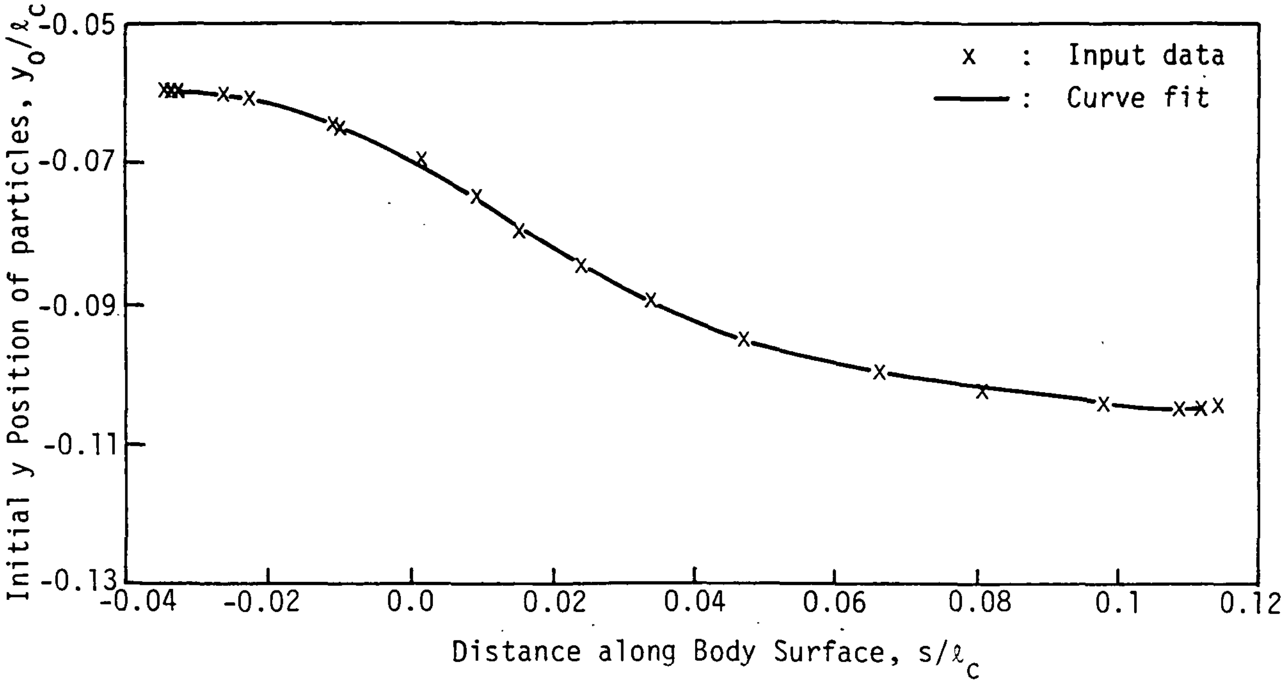
2.5 General Computational Procedure

The general computational procedure to determine the local collection coefficient of a body is carried out as follows. First, the initial conditions for the differential equations governing the particle motion are determined either automatically by the computer program or input manually by the user. These conditions call for specification of an initial particle position x_0, y_0 and an initial particle velocity $\dot{x}_0, \dot{y}_0$. The computer program described in the following sections automatically determines the initial conditions if desired. The initial upstream x-coordinate, x_0, is assigned the value of x at which the difference in the free-stream velocity W_∞ and the local velocity W is less than some small value ε. The value of ε is specified by the user.

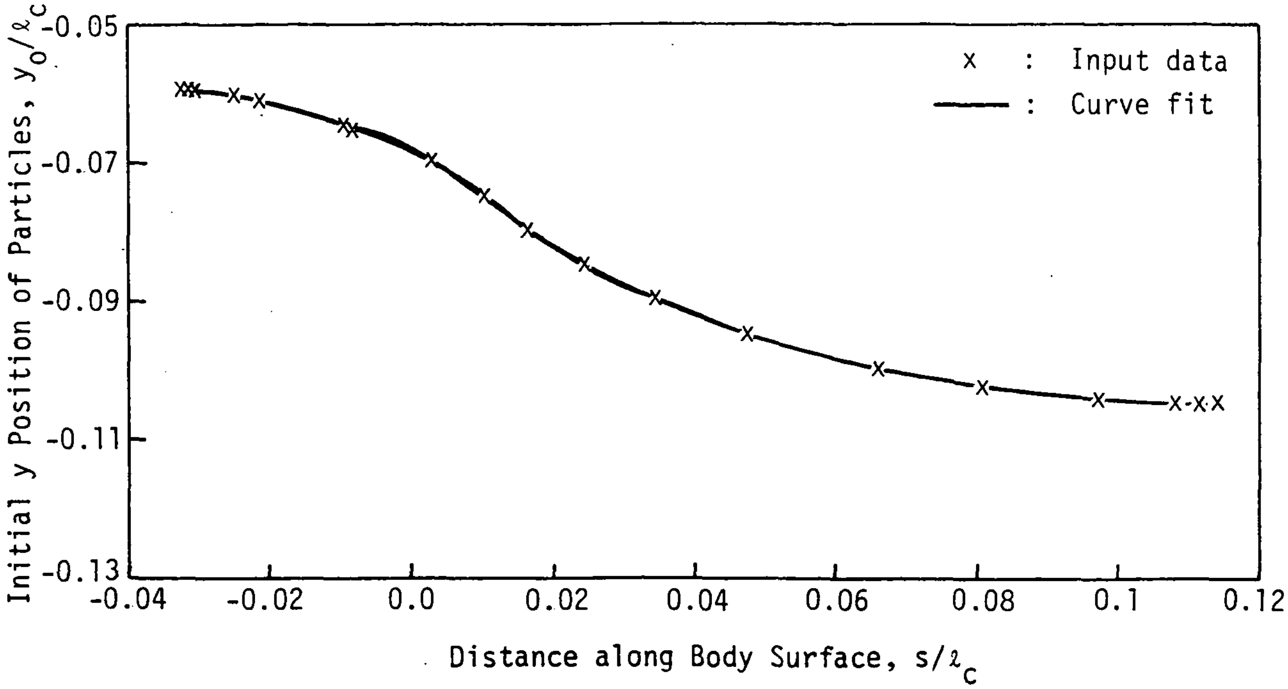
To determine the initial vertical coordinate y_0, the computer automatically searches for the upper and lower limits of the y-coordinate, y_{ou} and $y_{o\ell}$, respectively (see Figure 2.3). Any particles released within this region will strike the body. Any particles released outside this region will miss the body and are of no interest to the computation of collection efficiency. The range of vertical position y_{ou} to $y_{o\ell}$ is then divided into a number of increments prescribed by the user. The trajectory of particles leaving each of these vertical positions is calculated and the impingement position of the particle on the body surface, s, is recorded. This collection of $\{y_0,s\}$ values plus those generated during the computer search for y_{ou} and $y_{o\ell}$ are used to express s as a function of the particle's initial vertical coordinate y_0. The value of $\beta = -dy_0/ds$ is then computed by a linear approximation or by curve fitting the total collection of data points $\{y_0,s\}$ to a polynomial curve fit. The degree of the polynomial is specified by the user. The current program allows the user to curve fit the entire set of datum points to one curve or to fit the curve in segments using a prescribed number of points on either side of the specified position (see Figure 2.4). This latter procedure is similar to segment-averaging or segment-curve fitting of the entire curve.

The initial velocity of the particle is prescribed to be equal to the value of the flow field at x_0,y_0, i.e., $\dot{x} = W_x(x_0,y_0)$ and $\dot{y} = W_y(x_0,y_0)$ if not otherwise specified by the user.

The following sections describe in detail the computer program and necessary user's information to compute particle trajectories and local collection efficiency for two-dimensional airfoils and inlets.



(a) Total data set curve fitting; 6 degree polynomial with all input data. (Joukowski airfoil 0015, $\alpha = 4°$, with walls).



(b) Segment curve fitting; 3 degree polynomial with sequential sets of input data. (Joukowski airfoil 0015, $\alpha = 4°$, with walls).

Figure 2.4 Illustrates total data set and segment curve fitting techniques.

3.0 GENERAL DESCRIPTION OF COMPUTER PROGRAMS

The purpose of this section is to describe the computer program in sufficient detail so that it can be run successfully by the user. Section 3.1 describes some general features of the program which will better enable the user to follow the data input instructions given in Section 3.2. Instructions for the geometry generation program and for the axisymmetric potential flow program are also given in Section 3.2. These, however, are simply reproduced from Reference 1 without appreciable discussion. Input instructions for the grid generation computer code are also given in Section 3.2. The original references should be consulted if additional information is required.

3.1 General Features of the Program

3.1.1 Types of Flow

Axisymmetric potential flow over arbitrarily shaped bodies is considered. The water droplet in the flow field is treated as a solid sphere although the option for a nonspherical particle is provided in the computer program. For nonspherical particles the user must provide expressions for the coefficients of aerodynamic lift, drag, and moment.

3.1.2 Surface Distance Computation

The surface distance along the body is computed by summing elements, Δs, determined from a linear approximation, $\Delta s = \sqrt{(\Delta x)^2 + (\Delta y)^2}$. Surface distance is measured from the leading edge of the body with positive values defined on the lower surface and negative value defined on the upper surface, Figure 2.3.

3.1.3 Initial Longitudinal Coordinate, x_0, of the Particle Trajectory

The initial longidinal coordinate position x_0 at which the particle trajectory calculations begin (Figure 2.3) is automatically determined by the computer program. The value of x_o is selected by testing the maximum difference between the locally computed value of W and the free-stream velocity W_∞ at successively farther distance upstream. The value of x for which the inequality

$$\left| 1 - \frac{W}{(W_\infty)} \right|_{max} \leq 0.001 \tag{3.1}$$

is satisfied is designated as x_o. Equation 3.1 is tested over a specified range YYLO $\leq y/\ell_c \leq$ YYUP, where YYLO and YYUP are initial values input by the user. They represent the expected maximum and minimum range of the upstream y-coordinate. ℓ_c is the reference length which is normally the chord length for an airfoil or the mouth diameter for an inlet. This procedure for selecting the initial position of the particle trajectory has been developed so that computer time may be conserved by starting the particle as near the leading edge as variations in flow field velocity will allow.

3.1.4 Impingement Position of the Particle on the Body

A coordinate transform technique is utilized in determining the position at which a particle strikes the body. The transform technique is illustrated in Figure 3.1. The equations governing the coordinate transform are:

$$X_t = X$$

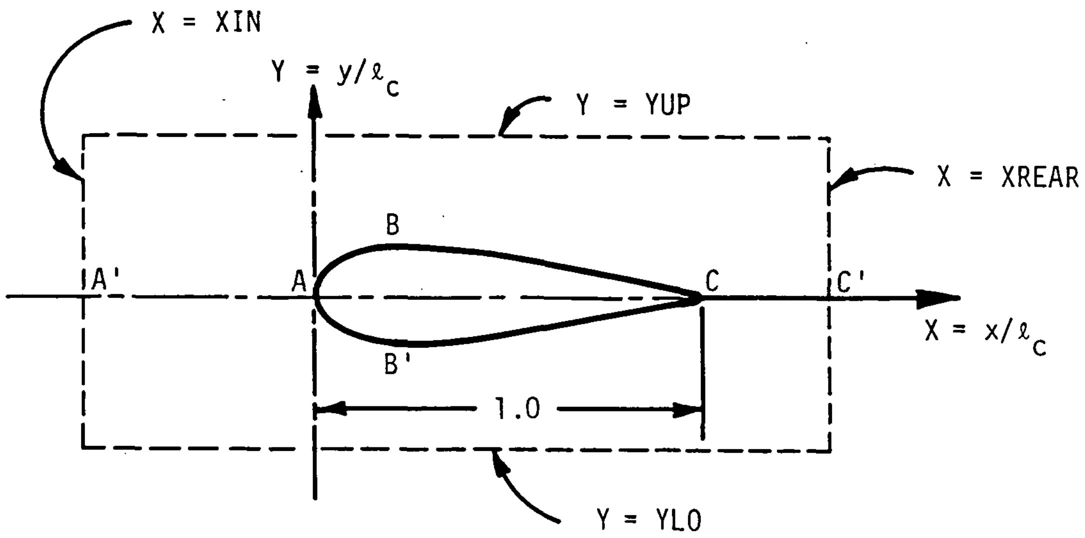
$$Y_t = \frac{Y - YREF}{YREF - YLO} \tag{3.2}$$

where (see Figure 3.1a)

$$YREF = Y_{A'ABCC'} \text{ if } Y \geq 0.0$$

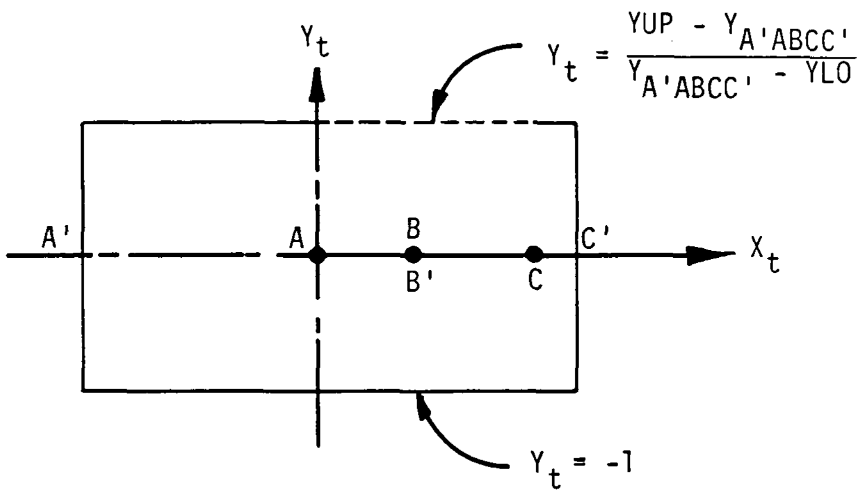
$$YREF = Y_{A'AB'CC'} \text{ if } Y \leq 0.0$$

where X and Y are the x- and y-coordinates normalized by the characteristic length, ℓ_c, respectively. If the particle moves across the coordinate line $Y_t = 0$ in the transformed plane and the X_t position of the particle is greater than zero, the particle is recorded as having crossed the surface of the body. An iteration procedure, described in the following



Note: XREAR is initial input data, YUP and YLO are automatically calculated in the program.

(a) Physical plane



(b) Transformed plane

Figure 3.1 Coordinate transform.

subsections, is then carried out to determine the exact surface location of impingement.

3.1.5 Computation of Surface Impingement Location

The method of computing the surface location of particle impingement is described in this section. First a general case and then a special case (see Figure 3.2) are considered.

3.1.5.1 General Case. During a time step Δt, consider the particle to cross the body along the line $\overline{ON}$ (see Figure 3.2a). Let the coordinates of the particle at time t be (XO,YPO) and the coordinates at $t = t + \Delta t$ be (XN,YPN). The reference coordinates on the body surface A'ABCC' or A'AB'CC' are XO,YRO and XN,YNR, respectively. The analytical function of the line $\overline{ON}$ is:

$$\frac{Y - YPO}{X - XO} = \frac{YPN - YPO}{XN - XO} \tag{3.4}$$

The function of the line $\overline{O'N'}$, which joins the two nodal points describing the body surface, is:

$$\frac{Y - YRO}{X - XO} = \frac{YRN - YRO}{XN - XO} \tag{3.5}$$

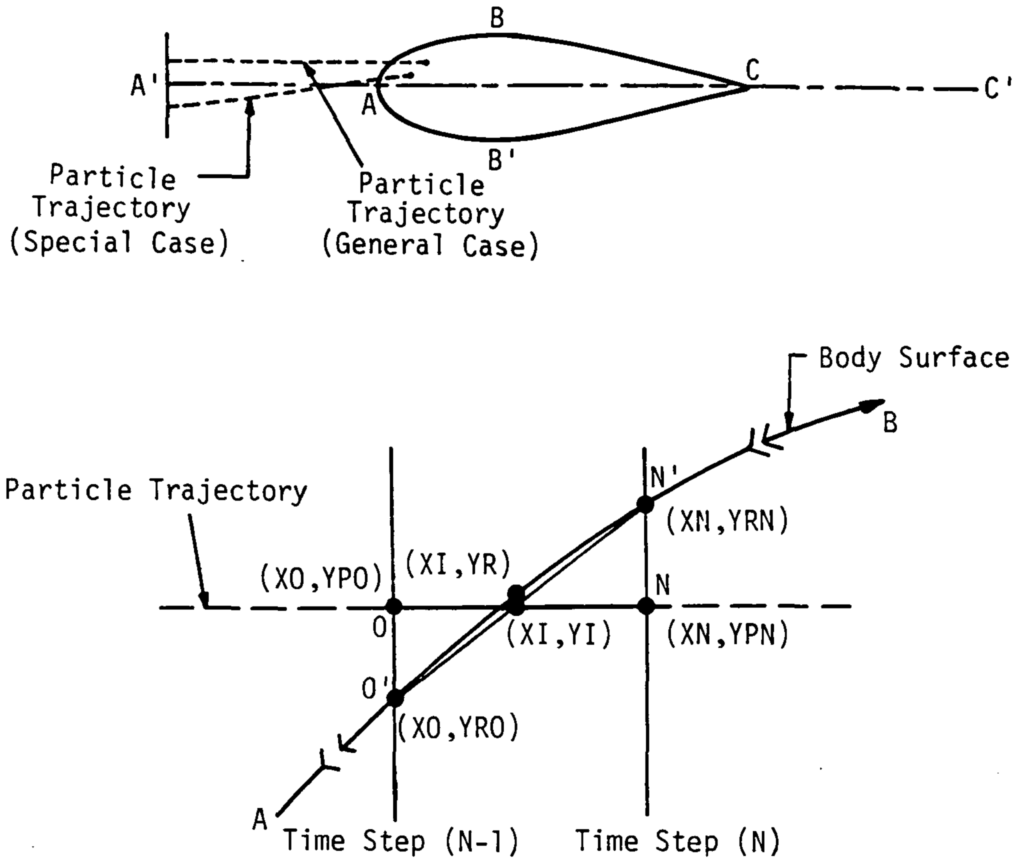
The coordinates (XI,YI) of the intersection of line $\overline{ON}$ and $\overline{O'N'}$ found by simultaneous solution of Equations 3.4 and 3.5 are:

$$XI = \frac{(XN - XO)(YPO - YRO)}{(YRN - YRO - YPN + YPO)} + XO$$

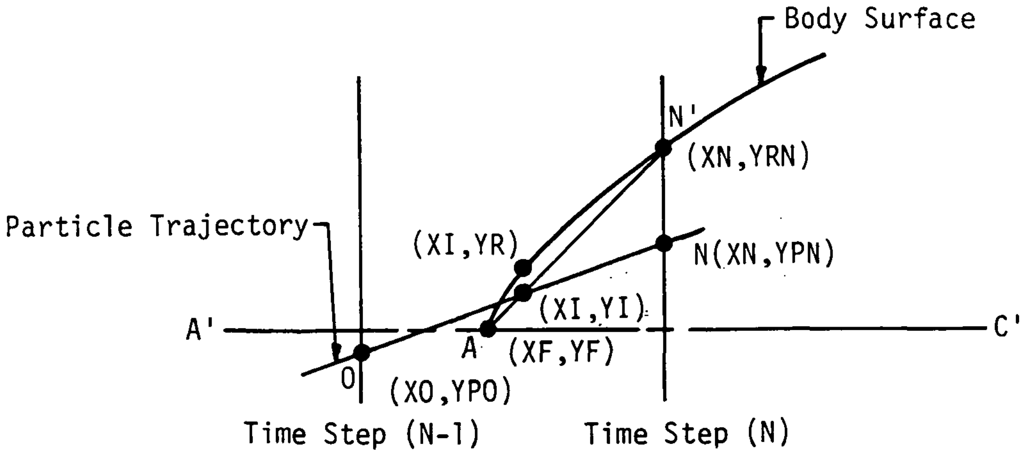
$$YI = \frac{(YPN - YPO)(YPO - YRO)}{(YRN - YRO - YPN + YPO)} + YPO \tag{3.6}$$

The reference coordinates on the body surface (XI,YR) corresponding to the position (XI,YI) is then found. If $|YI - YR| \leq 10^{-5}$, the point (XI,YR) is regarded as the particle impingement position on the body surface. If $|YI - YR| > 10^{-5}$ the point (XI,YI) is redefined as (XN,YPN) and (XI,YR) as (XN,YRN), respectively. The procedure is repeated until the requirement $|YI - YR| \leq 10^{-5}$ is satisfied.

3.1.5.2 Special Case. Consider the particle crossing the body surface for the special trajectory shown in Figure 3.2b. The function of line $\overline{ON}$ is:



(a) General case



(b) Special case

Figure 3.2 Illustration of the method of determining the position of particle impingement on the body surface.

$$\frac{Y - YPO}{X - XO} = \frac{YPN - YPO}{XN - XO} \tag{3.7}$$

and of line $\overline{AN'}$ is

$$\frac{Y - YF}{X - XF} = \frac{YRN - YF}{XN - XF} \tag{3.8}$$

The coordinates (XI,YI) of the point of intersection of line $\overline{ON}$ and line $\overline{AN'}$ are found as follows:

Let

$A = (XN - XF)(YPN - YPO) - (YRN - YF)(XN - XO)$

$B = YPN - YPO$

$C = YRN - YF$

$D = XN - XF$

$E = XN - XO$

then

$$XI = [B\cdot D\cdot XO - E\cdot C\cdot XF + (YF - YPO)\cdot E\cdot D]/A$$

$$YI = [D\cdot B\cdot YF - E\cdot C\cdot YPO + (XO - XF)\cdot B\cdot C]/A \tag{3.9}$$

The same procedure and criteria as for the general case is used to determine the position of particle impact on the body surface.

3.1.6 Upstream y-Coordinate Limits y_{ou},$y_{o\ell}$

The method by which the computer selects the upper and lower trajectory limits y_{ou} and $y_{o\ell}$ is described in this section. A number of options are available to the user depending on the setting of the flags LRANG and LIM. If LRANG=1 and LIM=1, the program searches automatically for the upper and lower limits of the radial (vertical) coordinates of the initial particle position y_{ou} and $y_{o\ell}$ (see Figure 2.3).

The search procedure consists of the computer program initially seeking the range within which y_{ou} and $y_{o\ell}$ lie. This is achieved by computing the particle trajectory from the initial position (XO,YMAX) where YMAX and YMIN are the user's initial guessed values of y_{ou} and $y_{o\ell}$, respectively, and $XO = x_o/\ell_c$. If the particle passes under or hits the body, YMAX and YMIN are redefined as $YMAX_{NEW}$ = YMAX + ΔY and

$YMIN_{NEW}$ = YMAX where ΔY = YMAX - YMIN. The procedure is then repeated until the particles pass over the body. The current values of YMAX and YMIN then specify the range within which more precise values of y_{ou} and $y_{o\ell}$ are sought. If on the first trajectory calculation the particle had passed over the body then the trajectory from (XO,YMIN) is computed. If the particle again passes over the body then the above procedure is reversed (i.e., $YMAX_{NEW}$ = YMIN and $YMIN_{NEW}$ = YMIN - ΔY).

Once this order of magnitude range of y_{ou} and $y_{o\ell}$ is determined, more precise values of these limits are computed as follows. A particle trajectory from the position Y' = (YMAX + YMIN)/2 is computed. If the particle passes under or hits the body then the next trajectory is computed from Y'_{NEW} = (Y' + YMAX)/2. Alternatively, if it passes under the body then Y'_{NEW} = (Y' + YMIN)/2. Successive halving of the range YMAX to YMIN in this manner continues until convergence is achieved. Convergence is assumed when the difference of the y_o coordinate between two trajectories, for which one impinges on the body and one misses the body, is less than a small value specified by the user. In the program the small number is designated as YLIM. For the test cases given in Section 5.0, YLIM = 10^{-6} m was used. Determination of the values of y_{ou} and $y_{o\ell}$ also provides values of the upper surface impingement limit, s_U, and the lower limit, s_L, respectively.

If the control flags are set such that LRANG=0 and LIM=1, the computer program will search for y_{ou} and $y_{o\ell}$ within the range YMAX and YMIN input by the user. If a poor guess was made and y_{ou} and $y_{o\ell}$ are not in this range, then the program is terminated. If the user desires to compute only one particle trajectory, the flags should be set to LRANG=0 and LIM=0.

3.1.7 Calculation of the Local Collection Efficiency

With the values of y_{ou} and $y_{o\ell}$ determined, the computer divides the range into NPL segments (i.e., $(y_{ou} - y_{o\ell})$/NPL) which represents the number of particles to be computed. Particles are then released from each of the NPL locations. The surface locations of impingement of each particle is then recorded and a collection of coordinates $\{y_o, s\}$ are stored in a file. These values are used to construct the functional relationship between y_0 and s from which the local collection efficiency is calculated.

Two methods are used to calculate the local collection efficiency, β. The first method is linear approximation:

$$\beta = - \frac{\Delta y_0}{\Delta s} \tag{3.10}$$

The second method utilizes a polynomial curve fit. The computed values of surface impingement, s, are determined as a function of y_0, i.e., $s = s(y_0)$, by a polynomial curve fit. The value of β is computed by taking the derivative of the polynomial function. The number of coefficent of the polynomial function is input by the user as NCOEF (the order of the polynomial is then NCOEF-1). The total number of points $\{y_0,s\}$ can be curve fit or the function can be curve fit in segments similar to a running average. For segment curve fitting the variable NS=0 and the number of sequential coordinate points used in a segment is specified by the user with the variable NPTS. If NS=1, all points are used. For multi-size distribution cases (NSI>1), only the all-points polynomial curve fit option is available.

3.2 Program Input Instructions

A general flowchart for the computer program is shown in Figure 3.3. The general procedure for running the program consists of first manually constructing a geometry input data tape, Tape 05. The geometry is specified in terms of a number of coordinate positions on the surface. The origin for the Cartesian coordinate system used to specify the geometry can be selected arbitrarily. The geometry generation computer program, SCIRCL, is then run to create Tape 17 which is the input file for the axisymmetric potential flow program. Section 3.2.1 describes the input tape or card deck structure for the geometry generation program. The format of the output tape, Tape 17, created by SCIRCL is the input to the axisymmetric potential flow program and is described in Section 3.2.2.

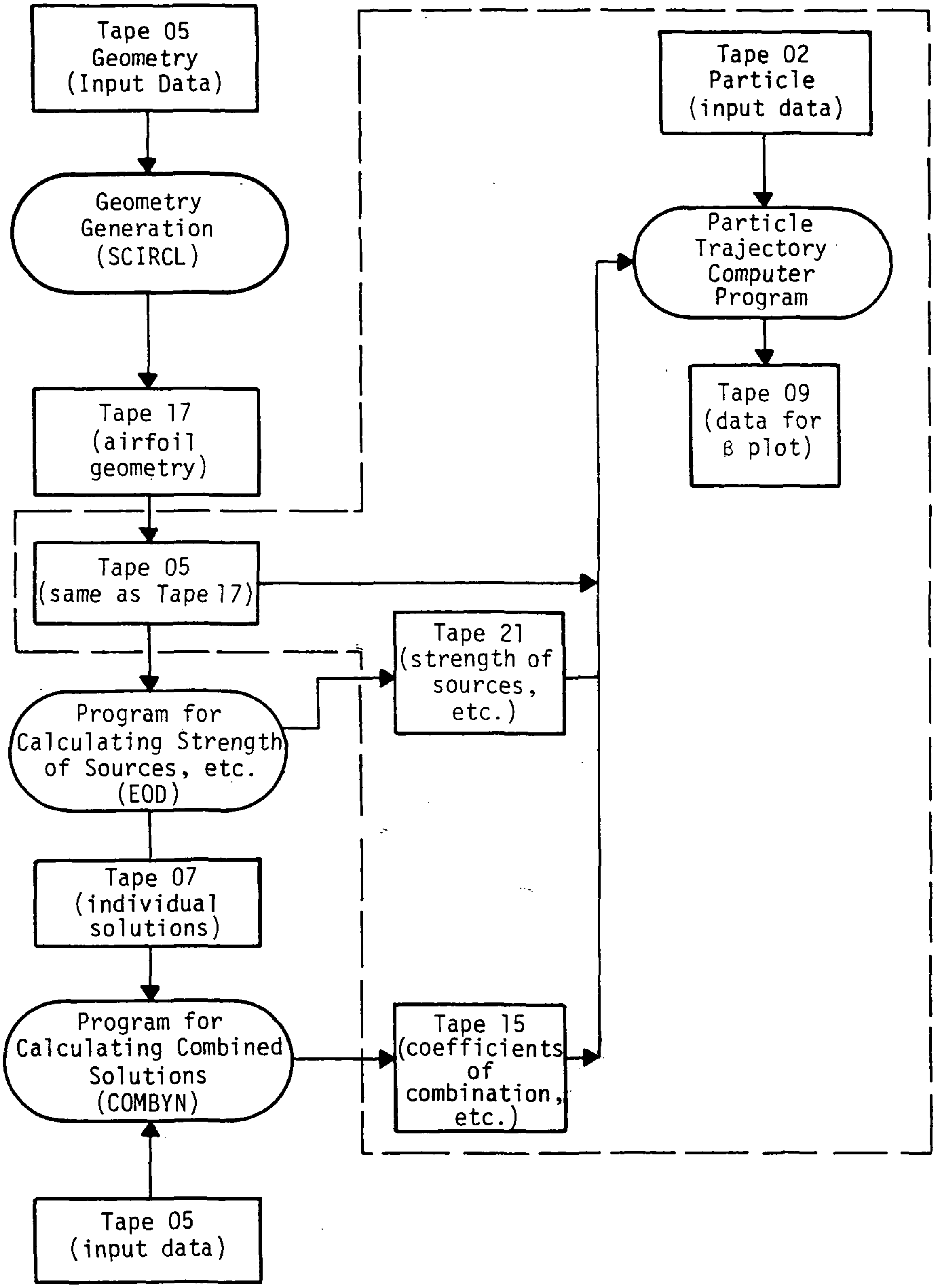


Figure 3.3 General flowchart.

Now using Tape 17 as input, which must be renamed Tape 05, the axisymmetric flow code is run and creates Tape 21. Tape 21 is directly provided to the particle trajectory computer program. Additionally, Tape 02, which must be manually created as described in Section 3.2.3, is required to run the particle trajectory program.

Running the particle trajectory program creates the output tapes listed below:

Tape 01: Data stored on Tape 01 is later written to Tape 03. Tape 01 = Tape 03 for unsymmetric flow cases.

Tape 03: Data stored on Tape 03 is used for the calculation of collection efficiency.

Tape 04: Tape 04 is created if LOPT$\neq$0 and contains coordinates of the particle trajectories as well as values of the wind speed component as a function of time.

Tape 08: Data sotred on Tape 08 is used for trajectory plots. The data sotred are the x/ℓ_c,y/ℓ_c coordinates of the particle XP,YP for each time step and the surface impingement point SW. SW is initially set at 88.8888. When SW$\neq$88.8888, the particle trajectory is terminated.

Tape 09: Tape 09 contains the local collection efficiency, β, as a function of surface position, s/ℓ_c. The program automatically plots β versus s/ℓ_c at the NASA/LeRC facility.

The input instructions for running the computer programs are presented in the following sections. The instructions are given in terms of an input card deck structure which compares directly with data tapes or files.

3.2.1 Geometry Generation Program Input Instructions

The input instructions for the geometry generation program are taken directly from Reference 1. These instructions are given for continuity of this report. It is assumed, however, that the user is familiar with the details of the geometry generation program. The input deck for the geometry generation program has the following card structure:

Card	Description
1	
2	
3	
4	
5	(only if flag J > 0)
6	
7	
8	Number of '8' cards = NRAKES
.	
.	
.	
8	
9	Number of '9' cards = ANBDYS
10	Number of '10-11-12' groups for each '9' card = ANSEG *If ENREED = 99 on card 10, use 11a and 12a instead of 11 and 12 *If ANSEG = 0 and TYPBDY ≠ 0 on card 9, skip 10 and substitute 11a and 12a for 11 and 12
11	
12	
.	
.	
10	
11	
12	

Card No.	Column	Code	Routine/ Format	Explanation
1	1-36	TITLE	Main/ 9A4	Description of case
2	1-10	XX	Main/ F10.2	Length, in plot inches, of x-axis required
	11-20	XMIN	"	Value, in data inches, of far left x point
	21-30	EXEP	"	Data in per plot inch along x-axis
	31-40	YY	"	Length, in plot inches, of y-axis
	41-50	YMIN	"	Value, in data inches, of bottom y point
	51-60	ORD	"	Data inch per plot inch along y-axis (usually equal to EXEP)
	61-70	ELREF	"	The x values in area output data are nondimensionalized by ELREF. Default value is 1
	61-80	AREF	"	The areas in area output data are nondimensionalized by AREF. Default value is 1

For information to be passed on to EOD:

Card No.	Column	Code	Routine/ Format	Explanation
3	1	IGEOMF	Main/I1	=1, Use flat elements =0, Use curved elements
	2	ISIGF	"	=2, Use constant source densities =1, Use linear source densities =0, Use parabolic source densities
	3	ICURVN	"	=1, Read in curvature values =0, EOD will compute curvatures
	4	NONEWF	"	=1, Use old velocity formula =0, New formula
	5-14	ALPHER	Main/ F10.2	If only one body is input, ALPHER is the angle of attack (used by EOD)

Card No.	Column	Code	Routine/ Format	Explanation
3	15	IVORT	Main/I1	=1, Perform axisymmetric, closed-duct solution =0, Perform strip vortex solution
	16	IPAR	"	Element geometry flag used by EOD =1, Parabolic elements =0, Linear elements
	17	IFST	"	First-order terms flag =3, Both first-order terms =2, Curvature terms =1, First derivative terms =0, No first-order terms
	18	ISND	"	Second-order terms flag =3, Both second-order terms =2, Curvature squared terms =1, Second derivative terms
	19-20	IFLLL	Main/I2	=1, A combination solution will be calculated by EOD =0, No combination solution will be calculated by EOD
4	1-8	IDENT	Main/A8	Eight-character tag for case I.D.
	9-12	PRCG	Main/A4	Title "EOD"
	17-20	N06	Main/I4	=0
	21	LPNCHO	Main/I1	Flag A =1, Do not save output for EOD on Unit 17
	22	IPLOTA	Main/I1	Flag B. Plot area against x position (see Reference 1)
	23-24	IPLOTC	Main/I2	Flag C =+1, Plot curvature versus S =-1, Plot curvature versus X
	25-26	IREAD	Main/I2	Flag D =0 (obselete)

Card No.	Column	Code	Routine/ Format	Explanation
4	37	IAB	Main/I1	Flag J. Redo geometry from point (see Reference 1)
	47	IREDON(1)	"	Flag E. Redo entire geometry via direct interpolation (see Reference 1)
	48	IREDON(2)	"	Flag F. LPNCHO for any redo
	49	IREDON(3)	"	Flag G. IPLOTA for any redo
	50-51	IREDON(4)	Main/I2	Flag H. IPLOTC for any redo
	52-53	IREDON(5)	"	Flag I. IREAD for any redo

All flag are on when equal to 1 unless otherwise noted. (Either E or J or neither can be on but not both.) Skip card 5 if J = 0.

Card No.	Column	Code	Routine/ Format	Explanation
5	1-12	XAA	Main/ F12.5	x position of starting point for partial redo
	13-24	YAA	"	y position of starting point for partial redo
	25-36	XBB	"	x position of ending point for partial redo
	37-48	YBB	"	y position of ending point for partial redo
6	1-10	ANBDYS	Main/ F10.2	Number of bodies
	11-20	DELS	"	Spacing between points in region of interest
	21-30	DELSMX	"	Maximum spacing far from region of interest
	31-40	XRI	"	Axial distance at which surface distance equals zero
7	1-4	NRAKE	Main/I4	Number of axial rake locations
8	1-8	XRAK	Main/ F8.5	Axial location of rake
	9-16	YLO	"	y value of first point (lowest point) on rake at XRAK

Card No.	Column	Code	Routine/ Format	Explanation
8	17-24	YHI	Main/ F8.5	y value of last point (highest point) on rake at XRAK
	25-27	NY	Main/I3	Number of points in rake at XRAK: Restriction $\Sigma NY<200$. Rake points are equally spaced, ΔY, between YHI and YLO where $\Delta Y = \frac{YHI - YLO}{(NY - 1)}$
	28-35	XTRAN	Main/ F8.0	Value of axial translation of rake
	36-43	YTRAN	"	Value of vertical translation of rake
	44-51	XSCALE	"	Value of axial scaling of rake
	52-59	YSCALE	"	Value of vertical scaling of rake
9	1-10	TYPBDY	Main/ F10.0	Body number. However, if there is symmetry, then any body can be input as a mirror image of any other body. That can be accomplished by setting TYPBDY= -M.N where M is the number of the body to be created and N is the number of the body to be copied. ANSEG is set to the Y value of the line about which body N is to be mirrored. No other input is required for this body except for ANLF.
	11-20	ASNEG	"	=Number of segments for the particular body, except as stated in TYPBDY

Card No.	Column	Code	Routine/ Format	Explanation
9	21-30	DELNEW	Main/ F10.0	=-1, Delta S spacing is set to original value of DELS =0, Delta S is set to value of DELS from previous body =+number, Delta S is set to value of input DELNEW
	31-40	ANLF	"	=0, Body is a lifting body, i.e., in EOD a vorticity solution about this body will be calculated =1, Body is a nonlifting body, i.e., no vorticity solution will be calculated

<u>Note</u>: All lifting bodies must be input prior to any nonlifting bodies.

Card No.	Column	Code	Routine/ Format	Explanation
	41-48	XTRAN	Main/ F8.0	Value of axial translation of this body
	49-56	YTRAN	"	Value of vertical translation of this body
	57-64	XSCALE	"	Axial scaling factor
	65-72	YSCALE	"	Vertical scaling factor
	73-80	XTMAX	"	Maximum value of x for which scaling is to be applied

(Code indicating type of curve to be fitted through given points.)

Card No.	Column	Code	Routine/ Format	Explanation
10	1-10	ENREED	Main/ F10.2	=0., for bisuperellipses [1]. =1,000. Same as =0 but with finer point spacing near one end of segment (two such segments required). Usually used to give finer spacing at the highlight. The superellipse going into the highlight and the one coming out should have this flag. For bisuperellipses where the '1,000' option is to be

Card No.	Column	Code	Routine/ Format	Explanation
10	1-10	ENREED	Main/ F10.2	used, the rate at which the point spacing, ds, changes near one end $dS_i = dS_{i-1} - (Rate)(dS_{i-1})$ can be specified on input. The rate (program name = PACE) is entered as the fractional part of ENREED for each segment. For example, if ENREED were input as 1,000.06, the spacing for consecutive points would be evaluated as follows: $DS_i = DS_{i-1} - (0.06)DS_{i-1}$ if segment is to go from large to small spacing or: $DS_i = DS_{i-1} + 1.5(0.06)DS_{i-1}$ if segment is to go from small to large spacing. If PACE is entered as zero (i.e., ENREED = 1,000.), the default value, 0.05, is used. (PACE $\leq$ 0.133) *The first '1,000' superellipse ON A BODY reduces the point spacing as far as possible, down to a limit of 2 percent of the ds value at the beginning of the segment. *All subsequent '1,000' superellipses input will increase ds as far as possible up to the input value of DELS. **Any number or types of* segments may be input between the first and subsequent '1,000' bisuperellipses, with the exception *of a normal bisuperellipse* (ENREED=0.).

Card No.	Column	Code	Routine/ Format	Explanation
10	1-10	ENREED	Main/ F10.2	=1., Is a straight line, input 2 coordinates (XIN(1), YIN(1), XIN(2), YIN(2)). The first and last straight lines on bodies 2 and 3 and the last straight line on body 1 will automatically have their spacing increased from approximately DELS near the region of interest to approximately DELSMX away from the region of interest. To get this type of spacing in the first straight line of body 1, ENREED must be specified as 10. =10., Special straight line used for initial straight line on lower shroud. The straight line starts with large spacing (DEMSMX) and ends with small spacing (DELS). =-1., Fits a lemniscate between a straight line and a point. Input is three coordinates. =-3., Fits a cubic between two straight lines. Input 4 coordinates. =-4.0, Generates a segment which is a mirrored image of all the points from (XIN(1), YIN(1)) to (XIN(2), YIN(2)) about the line Y = YIN(3). See cards 11 and 12 for XIN and YIN formats. =99., For direct interpolation option over one segment (see input instructions for card 12).

Card No.	Column	Code	Routine/ Format	Explanation
10	11-20	REEDEN(1)	Main/ F10.2	Input exponent of x-term for bisuperellipse equation. Blank for all other segment types [1].
	21-30	REEDEN(2)	"	Input exponent of y-term for bisuperellipse [1].
11	1-12	XIN(1)	Main/ 6F12.5	x-coordinate for specified points
	13-24	XIN(2)		
	25-36	XIN(3)		
	37-48	XIN(4)		
	49-60	XIN(5)		
	61-72	XIN(6)		
12	1-12	YIN(1)	Main/ 6F12.5	y-coordinate for specified points
	13-24	YIN(2)	"	
	25-36	YIN(3)	"	
	37-48	YIN(4)	"	
	49-60	YIN(5)	"	
	61-72	YIN(6)	"	

Note: If ENREED=99, input the following cards instead of cards 11 and 12.

Card No.	Column	Code	Routine/ Format	Explanation
11a		Z(1)	Name list/ $BODYIN/	z is a complex array containing the x value (in the real part) and y value (imaginary part) of each given point along the segment. The name list will normally be longer than one card. The program will use the input points with finer point spacing near regions of high curvature.

Card No.	Column	Code	Routine/ Format	Explanation
12a		DONE	Name list/ $AUXIN/	A logical variable which should be input as: =.TRUE.
		BYPASS		=.TRUE. if no refinement of points is required

Note: If ANSEG=0 and TYPBDY≠0, skip card 10 and substitute 11a for 11 and 12a for 12.

3.2.2 Axisymmetric Potential Flow Program Input Instructions

The input instructions for the axisymmetric potential flow program are taken from References 1 and 2 with some modifications. These instructions are given for continuity. It is assumed, however, that the user is familiar with the details of the axisymmetric potential flow program. The input format given is the format of Tape 17 (renamed Tape 05 when input to EOD). The card or tape structure is as follows:

Card Type	Description	Subroutine
1	Body title and case number	PART1
2	Control flag card	PART1
3	Chord/Mach number card	PART1
4 5	Body control card	BASIC1
6,6' 6a 6b	Input body element coordinates. IFORMT=0 input card 6 IFORMT=1 input card 6a IFORMT=2 input card 6b	BASIC1
7	Input curvature values (needed only if ICURVN≠0)	BASIC1
	Repeat cards 4-7 (NB+FLG05) times.	
8	Rake number card (needed only if IRAKE≠0)	BASIC1
9	Rake definition card (needed only if IRAKE≠0)	IRAKES
	Repeat card 9 NN times.	
10	Nonuniform flow control flag (needed only if NNU≠0)	BASIC2
11	Nonuniform flow velocities (needed only if NNU≠0)	BASIC2
	Repeat cards 10 and 11 NNU times.	

Card No.	Column	Code	Routine/ Format	Explanation
1	1-60	HEDR	PART1/ 15A4	Body description.
	66-69	CASE	PART1/ A4	Case number.
2	1	NB	PART1/I1	Number of bodies (I≤NB≤9).
	2	NNU	"	Number of nonuniform flows (0≤NNU≤5).
	3	FLG03	"	Axisymmetric flow flag.
	4	FLG04	"	Crossflow flag.
	5	FLG05	"	Off-body point input flag.
	6	FLG06	"	Basic data only flag.
	7	FLG06	"	Ellipse generator flag (consistent with card 5).
	8	FLG08	"	Blank
	9	FLG09	"	
	10	FLG10	"	
	11	FLG11	"	Perturbation velocities only.
	12	FLG12*	"	Solve potential matrix.
	13	FLG13	"	Blank.
	14	FLG14	"	Prescribed tangential velocity.
	15	FLG15	"	Strip-ring vorticity flag.
	16	FLG16	"	Omit axisymmetric uniform flow solution.
	17	FLG17	"	Omit crossflow uniform flow solution.
	18	FLG18	"	Surface vorticity (instead of sources) for the final bodies.
	19	FLG19	"	Prescribed values of the surface vortex strengths for the final bodies will be input.

---

*Available if and only if NONEWF=1, ISIGF=1, and IGEOMF=1 (see card 4).

Card No.	Column	Code	Routine/ Format	Explanation
2	20	FLG20	PART1/I1	All bodies are surface vorticity bodies.
	21	FLG21*	"	Extra crossflow.
	22	FLG22	"	Generated boundary conditions.
	23	FLG23	"	Ring wing option.
	24	FLG24	"	Blank.
	28	IPUVEL	"	Punched output.
	29-30	NIN	I2	Unit number for input coordinates (default=05).
	31	IPRIN1	I1	Matrix print flag.
	32	IPRIN2	"	Matrix-assembly coefficient print flag.
	33	IPRIN3	"	Very detailed matrix construction print flag.
	34	IRAKF	"	Automatic rake generation flag (see also cards 8 and 9).
	35	ISAVE	"	Blank.
3	1-10	CHORD	PART1/ F10.0	Reference chord length (default=1.0).
	11-20	MN	"	Mach number for Goethert correction (0.0 implies incompressible).
	21-30	TCNST	"	Blank.
	31-40	EPSLON	"	
4	1	IGEOMF	BASIC1/ I1	=0, curved elements =1, flat elements.
	2	ISIGF	"	=0, parabolic σ =1, linear σ =2, constant σ (on each element).
	3	ICURVN	"	=0, internally calculated element curvatures =1, input curvature (see card 7).

---

*Ibid.

Card No.	Column	Code	Routine/ Format	Explanation
4	4	NONEWF	BASIC1/ I1	=0, use the newest formulae =1, use the old formulae (implies flat elements and constant σ).
	5	IFORMT	"	Input format flag (see card 6).
	6-10	NN	BASIC1/ I5	Number of defining end points for the body.
	11-20	MX	BASIC1/ F10.0	x-multiplier value (default=1.0).
	21-30	MY	"	y-multiplier value (default=1.0).
	31-40	THETA	"	Coordinate rotation value (degrees, measured about -z axis).
	41-50	ADDX	"	x-increment (to be applied to all input x-coordinates for this body).
	51-60	ADDY	"	y-increment (to be applied to all input y-coordinates for this body).
5	6-10	BDN	BASIC1/ I5	Body number (sequential for bodies, zero for off-body points).
	16-20	SUBKS	"	Subcase flag.
	26-30	NLF	"	Nonlifting flag (for combination cases only).
	31-40	XE	BASIC1/ F10.0	Semi-major axis for ellipse cases (input if FLG07≠0).
	41-50	YE	"	Semi-minor axis for ellipse cases (input if FLG07≠0).
6	1-60	UTX1(I)	BASIC1/ 6E13.4	x-coordinates.
6'	1-60	UTY1(I)	"	y-coordinates.
6a	1-10	UTX1(I)	BASIC1/ F10.5	x-coordinates.
	11-20	UTY1(I)	"	y-coordinates.
6b	1-10	UTX1(I)	BASIC1/ F10.0	x-coordinates.

Card No.	Column	Code	Routine/ Format	Explanation
6b	21-30	UTY1(I)	BASIC1/ F10.0	y-coordinates.
7	1-60	HCURV(II)	BASIC1/ 6E13.4	Curvature values for the NN-1 elements which constitute this body.
8	6-10	NN	BASIC1/ I5	Number of automatically generated mass flow rakes.
9	1-10	X1	IRAKES/ F10.5	x-coordinate of start of the rake.
	11-20	Y1	"	y-coordinate of start of the rake.
	21-30	X2	"	x-coordinate of end of the rake.
	31-40	Y2	"	y-coordinate of end of the rake.
	41-45	N	IRAKES/ I5	Number of intervals to be used in the rake (note $4<N\leq200$ and N must be an even integer).
10	6-10	NUN	BASIC2/ I5	Flow identification number.
	16-20	MSF	"	=0, axisymmetric onset flow =1, crossflow onset flow =2, both 0 and 1.
	21-30	TYPE	"	= +1.0, velocity will be input in x,y component form = 0., velocity will be input in normal tangential form =-1.0, automatic generation of flow due to rotation about the z-axis (for crossflow only).
	31-40	FG	"	Flow generator constant.
11	1-60	NG(I)	BASIC2/ 6F10.0	x or normal component velocity.
11'	1-60	TG(I)	"	y or tangential component velocity.

3.2.3 COMBYN Program Input Instructions

The input instructions for the COMBYN program are taken from Reference 1. It is assumed that the user is familiar with the details of the COMBYN program. Data file Tape 07, which is one of the output data files of the potential flow program EOD, is an input data file for COMBYN. Data stored on Tape 07 are the coordinates and the corresponding individual velocity solutions of the flow field for on-body and off-body points with format (4E13.6). Additionally, Tape 05 generated manually by the user is also an input data file for the COMBYN program. The card or tape structure of Tape 05 is as follows.

Card Type	Description	Subroutine
1	Title card.	READS
2	Number of on-body and off-body points.	"
3 4	Input initial conditions of the flow.	"
5	Control flag.	"
6	Circumferential coordinates.	"
7	Location of control station rake.	"
8	Input for defining zero surface distance.	"

English engineering units are used throughout the program.

Length, in.
Velocities, ft/sec
Angles, deg
Pressures, lb/ft^2
Temperature, °R
Density, $slug/ft^3$
Force, lb
Weight flow, lb/sec

Card No.	Column	Code	Routine/ Format	Explanation
1	1-72	TITLE	READS/ 18A4	Title card.
2	1-4	NT(1)	READS/I4	Number of on-body points for the closed-end solution.
	5-8	NP(1)	"	Total number of off-body points.
	9-12	NT(2)	"	Number of on-body points for the open-end solution (eliminate the last body).
	13-16	NP(2)	"	Total number of off-body points.
	17-20	NID	"	Number of EOD I.D. cards.
	21-24	KSKIP	"	=0, for first case of COMBYN =1, for successive cases using the same EOE output.
	25-28	IOVHUB	"	=0, hub vorticity solution from EOD is not read. =1, hub vorticity solution from EOD is read.
3	1-8	VC	READS/ F8.3	Average axial velocity at the control station. Based on live flow area, i.e., the flow area minus the area associated with the boundary layer displacement thickness. If WDOT$\neq$0, the program will interpret this as a code to ignore the input VC and will calculate VC from WDOT. (To run a case with VC actually equal to zero set WDOT=0.0 and VC=0.0). If ICTLPT (card 5) is not zero, VC is interpreted as the specified pressure ratio (PS/P_T) at a "control point" rather than a velocity. Note: All three inputs, WDOT, ICTLPT, and VC, must be nonzero when the "control point" calculation is desired.
	9-16	VINF	"	Free-stream velocity.
	17-24	ALFAF	"	Angle of attack, 0.0 for free-stream perpendicular to inlet axis. Note that $\alpha_F = \alpha - 90°$. For "control point" cases only, ALFAF will be calculated when ALFAF is input as =999.0.

Card No.	Column	Code	Routine/ Format	Explanation
3	25-32	TTOTAL	READS/ F8.3	Total temperature, if PSTAT and TSTAT are read in (to be explained later), the program will calculate TTOTAL. If TTOTAL=0 and PSTAT and TSTAT=0, then TTOTAL=518.67 will be used.
	33-40	ELND	"	ELND is the arbitrary length used for scaling or normalizing. Refer to KND input, card 5. See also CUTOFF input below.
	41-48	YWING	"	Upper limit of integration for surface forces (used in subroutine INFRCE).
	49-56	UTIP	"	Rotor tip speed. Need not be input unless relative rotor inlet quantities are desired. (See COMBYN OUTPUT.)
	57-64	VA	"	Bulk velocity at control station, i.e., average inlet axial velocity based on geometric area. If VA=0.0, the program will interpret this as a code and set VA=VC.
	65-72	PT	"	Total pressure. If PT=0.0 and PSTAT=0.0, the program will set PT=2116. If PT=0.0 and PSTAT≠0.0, PT is calculated.
	73-80	CUTOFF	"	If CUTOFF≠0, the pressure ratio P_s/P_t on the shroud will be plotted (on Calcomp) against dimensionless surface distance S/ELND starting at X=XRI and proceeding in both directions along the surface for a distance of S=CUTOFF. Length of plot in paper inches is 10 (CUTOFF/ELND). There is one plot for each circumferential angle THETA.
4	1-8	PSTAT	"	Static pressure.
	9-16	TSTAT	"	Static temperature. (If PSTAT and TSTAT are not 0.0, total pressure (PT) and total temperature (TTOTAL) will be calculated using PSTAT and TSTAT.
	17-24	WDOT	"	Weight flow - required unless VC≠0 and concurrently ICTLPT=0.

Card No.	Column	Code	Routine/ Format	Explanation
	25-32	DELQ	READS/ F8.3	Increment in mass flow fraction for spacing for calculated streamlines. NOTE: Default Value = 0.1, if DELQ is input as 0.
	33-40	VPERIN	"	Blank.
	41-48	MC	"	
	49-56	MINF	"	
5	1-4	NTHETA	READS/I4	Number of THETA's where THETA is the circumferential coordinate. If NTHETA=0, one THETA (THETA<70°) will be read in and used as the initial angle for the start of three-dimensional, on-body streamlines. For this option, THETA will vary as the streamline is followed up the shroud instead of remaining a constant on one meridian. NOTE: No INFRCE (force) calculations or pressure plots can be requested when NTHETA=0.
	5-8	NCLO	"	One rake must be chosen as the control station. NCLO is the number of the first point on this rake.
	9-12	NCHI	"	The number of the last point on the control station rake.
	13-16	NX	"	If NX=-1, inlet total force calculations are obtained (subroutine INFRCE). If NX=+1, a supersonic velocity correction is activated. At those on-body points where local supersonic flow is detected, velocities and pressure ratios are readjusted based on local Mach numbers and the rate of change of the local velocities. (Since off-body points having supersonic velocity are not corrected, there will be an inconsistency between the corrected on-body points and adjacent off-body points.)

Card No.	Column	Code	Routine/ Format	Explanation
5	17-20	KND	READS/I4	Flag for scaling variables before velocity and pressure calculations and also for nondimensionalizing after calculations and just before printout: Scaling: All input lengths and coordinates are divided by ELND immediately after being input, and WDOT is set to WDOT/ELND^2. If KND = -1, ELND = YTESTS = 0, ELND = 1.0 (no scaling) = +1, ELND = YTESTS - YTESTH = +2, ELND = the read-in value from card 3. Nondimensionalizing: If KND=8, the surface distance, S, will be divided by the read-in ELND just prior to printout. If KND=9, the on-body X and Y coordinates will be divided by the read-in ELND just prior to printout. NOTE: If CUTOFF is nonzero, surface distance will automatically be normalized by ELND before printout.
	21-24	NOTHET	"	If = 1, WDOT and VC will be left constant, as input, for all values of THETA (neglecting crossflow term). If = 0/blank, WDOT and VC will be corrected for crossflow and will vary with THETA.
	25-28	ICTLPT	"	Index number (from EOD output) of the desired "control point" where a known pressure ratio is to be input in lieu of a control station velocity. See VC.
	29-32	ISWIRL	"	(Required when NTHETA=0). Index number of point on shroud where three-dimensional, on-body streamline calculation will begin; preferably near the fan face.
6	1-80	THETA(I)	READS/ 10F8.3	Circumferential coordinate in degrees (number of THETA's read in depends on NTHETA, $NTHETA \leq 10$).
7	1-10	XTEST	READS/ F10.2	Axial location of control station rake. Must be compatible with NCLO and NCHI.

Card No.	Column	Code	Routine/ Format	Explanation
7	11-20	YTESTH	READS/ F10.2	Y on the hub at XTEST (control station).
	21-30	YTESTS	"	Y on the shroud at XTEST (control station).
8	1-10	XRI	"	Value of X at which the surface distance is zero.
	11-20	YRIHUB	"	Y on the hub at XRI.
	21-30	YRISHR	"	Y on the shroud at XRI.
	31-34	NHUBMX	READS/I4	The number of the last point on the hub (this can be found in the printed output of SCIRCL).

3.2.4 Particle Trajectory Program Input Instructions

There are five input data files used in the trajectory program. They are described in the following:

1. Tape 05 stores the airfoil geometry data. This is the same file as the input file Tape 05 of the potential flow program, Section 3.2.2.

2. Tape 21 stores the data for calculating the flow velocities. This is the same file as the output file Tape 21 of the potential flow program.

3. Tape 15, which is one of the output data files of COMBYN program, stores the coefficients for calculating the combination solution.

4. Tape 02 stores the initial input data for the trajectory program. Details are described in the following. The data must be in MKS units with the exception of particle diameter, which is input as microns.

The input deck structure for the particle trajectory program is as follows.

Card Type	Description
1	Number of bodies; degree of governing equations
2	Flow field control flags
3	Trajectory control flags
4	Initial conditions of flow field
5	Initial conditions of particles
6	
7	
8	
9	Number of size increments
10	This card is input only when NSI>1; number of 10 cards = NSI
.	
.	
10	

Card Type	Description
11	
12	Input for polynomial least square fit
13	TITLE of experimental data
14	Number of experimental data
15 . . 15	Number of 13 cards = NASA
16 17 18 19	Input for plotting

For more than one body, cards 11 through 19 are cycled NBDY times.

Card No.	Column	Code	Routine/ Format	Explanation
1	6-10	NBDY	Main/I5	Number of bodies. =1, One-body case (airfoil without walls) =2, Two-body case (inlet without hub) =3, Three-body case i) Airfoil with walls (LNTL=1) ii) Inlet with hub (LNTL≠1) Axisymmetric inlet case, NBDY=3.
	16-20	NPL	"	Number of particle trajectories to be computed.
	26-30	NSEAR	"	Maximum number of loops allowed in the search for the upper and lower impingement limits for the case LIM=1 (card 3).
	36-40	NEQ	"	Number of equations to be solved. NEQ=6 for a lifting, rotating particle; NEQ=4 for a spherical particle undergoing drag only (Section 2.0).

Card No.	Column	Code	Routine/ Format	Explanation
2	6-10	LEQM	Main/I5	=1, The initial particle velocity is equal to the flow at the initial particle location. ≠1, The initial velocity conditions input on cards 4 and 6 are = used as the initial particle velocity.
	16-20	LSYM	"	Symmetric flow field flag =0, Unsymmetric flow field (general case). =1, Symmetric flow field (only half plane is computed). Axisymmetric case LSYM=0.
	26-30	LRANG	"	≠0, Locates approximate values of y_{ou} and $y_{o\ell}$ (see Section 3.1.6).
3	6-10	LIM	"	≠0, Calculates surface impingement limits y_{ou} and $y_{o\ell}$. =0, Calculates single particle trajectories; LRANG=0 (see Section 3.1.6).
	16-20	LOPT	"	≠0, Stores details of particle trajectories on data Tape 04.
	26-30	LPLOT	"	≠0, Executes subroutine for plotting local collection efficiency, β, versus surface distance, s/ℓ_c.
	36-40	LTNL	"	=1, Computes flow over airfoil with walls (i.e., wind tunnel simulation). ≠1, Computes flow over airfoil without walls and inlet case. Axisymmetric case, LTNL≠1.
	46-50	LXOR	"	Control flag for adjusting the upstream x-coordinates, XORC, for upstream position of particle release (see Section 3.1.3). =0, The upstream x-coordinates of particle release is not adjusted. Particles are released from XORC input on card 6 by user. =1, Particles are released from the x-coordinate position adjusted by the criteria: $\|1 - (W/W_\infty)\| \leq$ EPSX.

Card No.	Column	Code	Routine/ Format	Explanation
4	6-15	G	Main/ F10.0	Acceleration of gravity, m/s^2.
	21-30	YINF	"	Velocity of free-stream, m/s.
5	6-15	DP	"	Particle volume median diameter in microns (10^{-6} m).
	21-30	RL	"	Reference length, e.g., chord length of the airfoil, m.
	36-45	TIMSTP	"	Initial value of the time step used in the Adams-Moulton predictor-corrector method (Section 2.3).
	51-60	YLIM	"	Accuracy criteria for computing the surface impingement limits (Section 3.1.6) YLIM=10^{-5} m is recommended.
	66-75	CF	"	Cunningham correction factor. Use if volume median diameter DP<10μ; otherwise, CF=1.
6	6-15	ATK	"	Initial value of the angle, α, Figure 2.1 (deg).
	21-30	PIT	"	Initial value of the angle, θ, Figure 2.1 (deg).
	36-45	PITDOT	"	Initial value of $\dot{\theta}$ (deg/sec).
	51-60	XORC	"	Upstream x-coordinate position of particle release, x/ℓ_c (LXOR=0).
	66-75	YORC	"	Upstream y-coordinate position of particle release, y/ℓ_c (NPL=1, LRANG=0, and LIM=0).
7	6-15	XREAR	"	Maximum downstream value of x/ℓ_c for which the flow field is computed.

Card No.	Column	Code	Routine/ Format	Explanation
7	21-30	YYLO	Main/ F10.0	Minimum value of y/ℓ_c for which the flow field is computed.
	36-45	YYUP	"	Maximum value of y/ℓ_c for which the flow field is computed.
	51-60	YMAX	"	Initial guess of the upper limit y_{ou}/ℓ_c (LIM=1) (Section 3.1.6).
	66-75	YMIN	"	Initial guess of the lower limit $y_{o\ell}/\ell_c$ (LIM=1) (Section 3.1.6).
8	6-15	ADEPS	"	Convergence criteria of the Adams-Moulton predictor-corrector method. ADEPS=0.001 is recommended.
	21-30	ANLFLO	"	Angle of flow direction relative to airfoil chord line measured from positive direction of x-axis.
	51-60	EPSX	"	Accuracy criteria for the case LXOR=1 (see card 3); EPSX=0.001 is recommended.
9	6-10	NSI	Main/I5	Number of droplet size increments.
10	This card is input only when NSI>1.			
	1-10	PLWC	Main/ F10.0	Percentage of liquid water content for each drop size.
	11-20	DPD	"	Distribution of the particle diameter ratio to volume median diameter.
	21-30	CFP	"	Cunningham correction factor corresponding to DPD (see Table 6.2). If DP*DPD>10μ, CFP=1.

Number of 10 cards = NSI.

Card No.	Column	Code	Routine/ Format	Explanation
11	1-80	TITLE	Main/ 20A4	Initial input data for local collection efficiency calculations.
12	6-10	NCOEF	Main/I5	Number of coefficients of polynomial curve fit for calculating β. The order of the polynomial is NCOEF-1 (Section 3.1.7).
	21-25	NPTS	"	Number of data points used in curve fitting for computing local collection efficiency (Section 3.1.7).
	36-40	NS	"	=0, Segment curve fit data =1, Total curve fit (see Section 3.1.7).
13	1-80	TITLE	Main/ 20A4	Experimental values of local collection efficiency.
14	1-5	NASA	Main/I5	Number of experimental data points input for comparison with computed results.
15	1-10	SFOIL	Main/ F10.5	Surface distance, s/ℓ_c, along the body at the location of the measured value of local collection efficiency.
	11-20	EFC	"	Experimental values of local collection efficiency at positions SFOIL.
16	1-80	XTITL	APLOT/ 20A4	Title of the x-axis.
17	1-80	YTITL	"	Title of the y-axis.
18	6-15	YHEIT	APLOT/ F10.0	Length of y-axis for graph to be plotted (inches).
	21-30	YMAX	"	Highest value of datum point on y-axis (inches).
	36-45	XMIN	"	Far right value of datum point on x-axis (inches).

Card No.	Column	Code	Routine/ Format	Explanation
19	6-15	XLENG	APLOT/ F10.0	Length of x-axis for graph to be plotted (inches).
	21-30	XMAX	"	Far left value of datum point on x-axis (inches).
	36-45	XMIN	APLOT/ F10.0	Far right value of datum point on x-axis (inches).

Cards 11 through 19 are cycled NBDY times.

3.3 Description of Program Output

The output files of the programs are divided into three categories: (1) output of the geometry generation program, (2) output of the axisymmetric potential flow program, and (3) output of the particle trajectory program. For the first two programs only the output files related to the present particle trajectory program are described.

3.3.1 Geometry Generation Program Output Description

The geometry generation program is named SCIRCL. The output data file Tape 17 of SCIRCL is used as the input data file Tape 05 of the axisymmetric potential flow program and of the particle trajectory program. Tape 17 stores the airfoil geometry data. In general, Tape 17 is the same for both the EOD and the particle trajectory program; however, in some cases it is modified when used in the trajectory program as described in Section 3.2.2 and Section 5.3. The format of Tape 05 is the same as the input data described in Section 3.2.2.

3.3.2 Axisymmetric Potential Flow Program Output Description

The output data file Tape 21 of the potential flow program is used as the input data file Tape 21 for the particle trajectory program. Tape 21 records all the necessary information including the source, sink, and/or vorticity distributions along the body surfaces for calculating the flow field about the bodies. Note that Tape 21 will not be generated by the axisymmetric potential flow program unless at least one value has been assigned to NRAKE during input to the SCIRCL program.

Thus, file Tape 05 must contain one or more off-body points (Section 3.2.2).

The output data file Tape 07 of the EOD program is used as the input data file Tape 07 for the COMBYN program. Tape 07 records the coordinates and the corresponding individual solutions for on-body and off-body points.

3.3.3 COMBYN Program Output

The output data file Tape 15 of the COMBYN program is used as the input data file Tape 15 for the particle trajectory program. Data stored on Tape 15 are the coefficients for evaluating the combination solution.

3.3.4 Trajectory Program Output

A data file Tape 04 is generated if LOPT≠0 and contains the values of x, y, $\dot{x}$, $\dot{y}$, θ, and $\dot{\theta}$ for each time step of the particle trajectories. Values of the flow field velocities at the particle locations along the particle trajectory are also contained on this file.

Data file Tape 08 is also generated if LOPT≠0 and contains the particle locations along the computed trajectories for plotting purposes. The data stored on Tape 08 and Tape 04 are nondimensional. Because Tape 04 and Tape 08 are used to record the information at each time step, the program will need much more storage than that generally needed with the option LOPT=0. LOPT≠0 should be used only if a few particle trajectories are to be calculated.

Additionally, information on particle initial position, impingement limits, surface impingement point coordinates, and surface distance are recorded on data file Tape 06 for printout. The computed local collection efficiency, β, is also recorded on Tape 06. The value of β versus surface distance is recorded on Tape 09 for plotting purposes.

Tape 09 is unformatted. The format for Tape 06 will be explained in Section 5.0 where test cases are documented. The format for Tape 04 and Tape 08 is illustrated as follows.

Format for Tape 04 (FORMAT(10E10.3))

Column	Code	Explanation
1-10	X	Time
11-20	E1	x-coordinate of the particle, x/ℓ_c
21-30	E2	y-coordinate of the particle, y/ℓ_c
31-40	YLAST(5)	Pitch angle of the particle in radians
41-50	E3	x-component velocity of the particle
51-60	E4	y-component velocity of the particle
61-70	YLAST(6)	z-component angular velocity of the particle
71-80	E5	Velocity of the particle relative to the flow field
81-90	W1	x-component velocity of the flow field
91-100	W2	y-component velocity of the flow field

Format for Tape 08 (FORMAT(3F10.5))

Column	Code	Explanation
1-10	XP	x-coordinate of the particle, x/ℓ_c
11-20	YPL	y-coordinate of the particle, x/ℓ_c
21-30	SW(IW)	The parameter is used to indicate the end of each trajectory, i.e., if SW≠88.8888 the trajectory is terminated.

4.0 PARTICLE TRAJECTORY COMPUTER PROGRAM CAPABILITIES AND FUNCTION DESCRIPTIONS

Details of the particle trajectory computer program are presented in this section. Each of the subroutines is described individually in Section 4.1. A list of FORTRAN variable names is given for each subroutine, and flowcharts are provided to illustrate the order of calculation. Error messages and typical corrective measures are given in Section 4.2.

4.1 Main Program and Subroutines

4.1.1 Main Program

Objective: The main program serves as an executive program for initialization and program control.

Variables: All input initial values of variables and control flags are described in Section 3.2.3.

Y(1)	Value of x
Y(2)	Value of y
Y(3)	Value of $\dot{x}$
Y(4)	Value of $\dot{y}$
Y(5)	Value of θ
Y(6)	Value of $\dot{\theta}$
VINF	Free-stream velocity, W_∞
YP	y/ℓ_c coordinate of current particle location
YHIT	Initial y/ℓ_c coordinate of a particle which impacts body
YMIS	Initial y/ℓ_c coordinate of a particle which moves away from the body
SL	Lower surface impingement limit, s_L/ℓ_c

SU	Upper surface impingement limit, s_U/ℓ_c
ZL	Lower limit, $y_{o\ell}/\ell_c$, corresponding to SL
ZU	Upper limit, y_{ou}/ℓ_c, corresponding to SU

Flowchart: See Figure 4.1

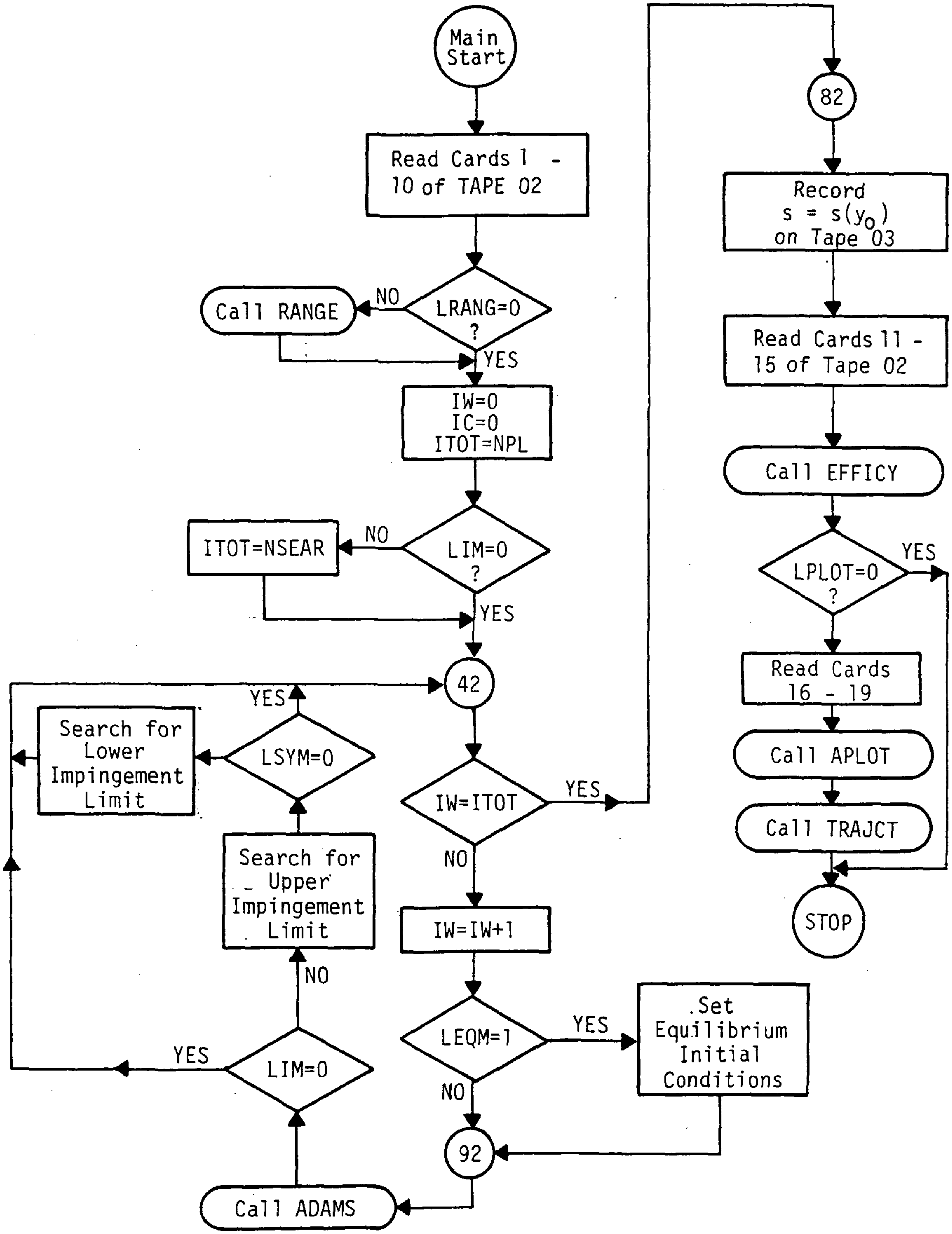


Figure 4.1 Flowchart of main program.

4.1.2 Subroutine ADAMS

Objective: Use Adams-Moulton, variable time step, predictor-corrector method to solve the equations of particle motion.

Variables:

YPRED	Adams-Moulton predictor
YCORR	Adams-Moulton corrector
YLAST	Value of function y (see Section 4.1) at previous time step
TSP	Initial time step increment
X	Time

Flowchart: See Figure 4.2

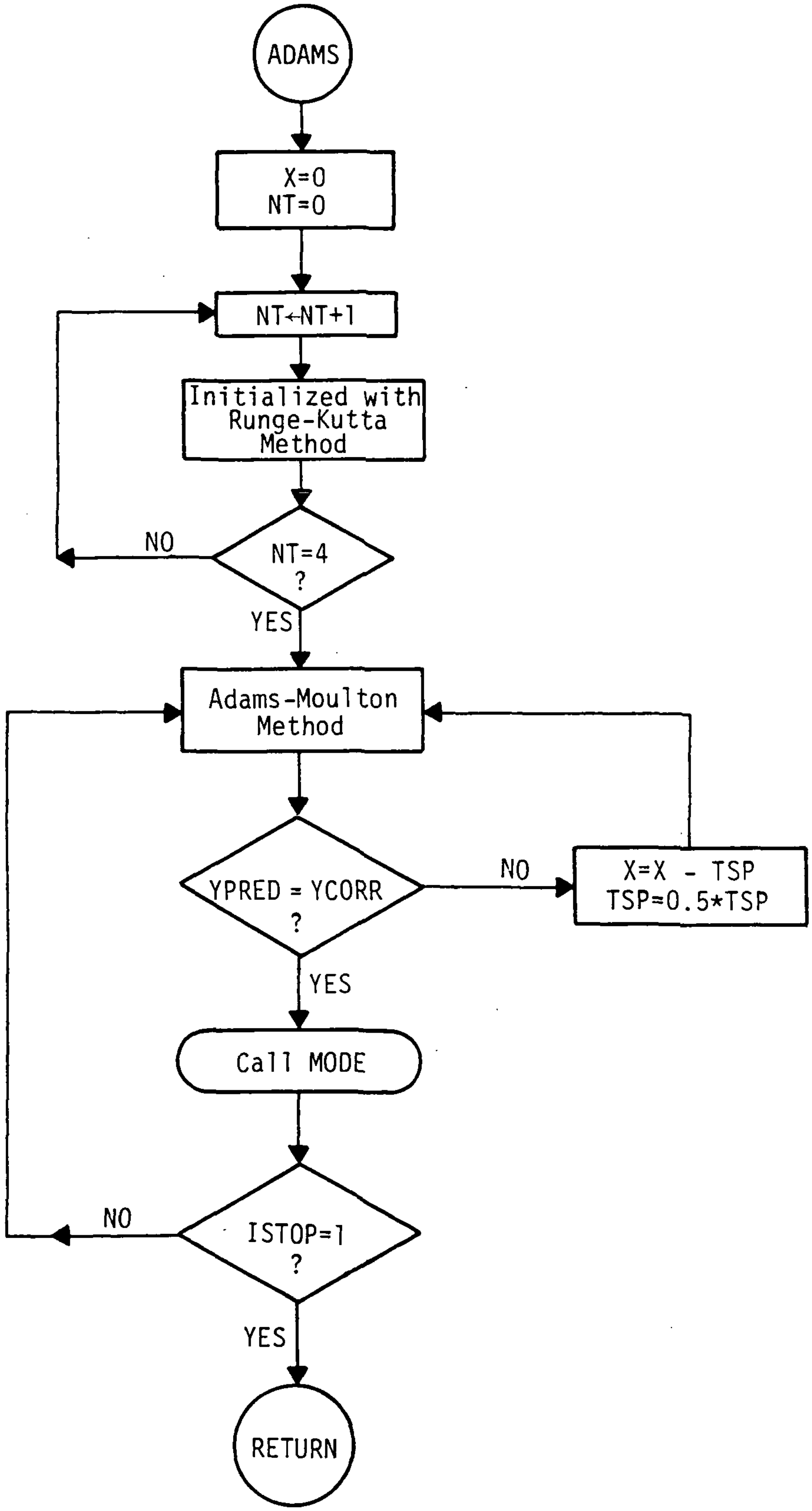


Figure 4.2 Flowchart of subroutine ADAMS.

4.1.3 Subroutine RANGE

Objective: Locate an approximate range of y_0 which contains the limits y_{ou} and $y_{o\ell}$.

Flowchart: See Figure 4.3

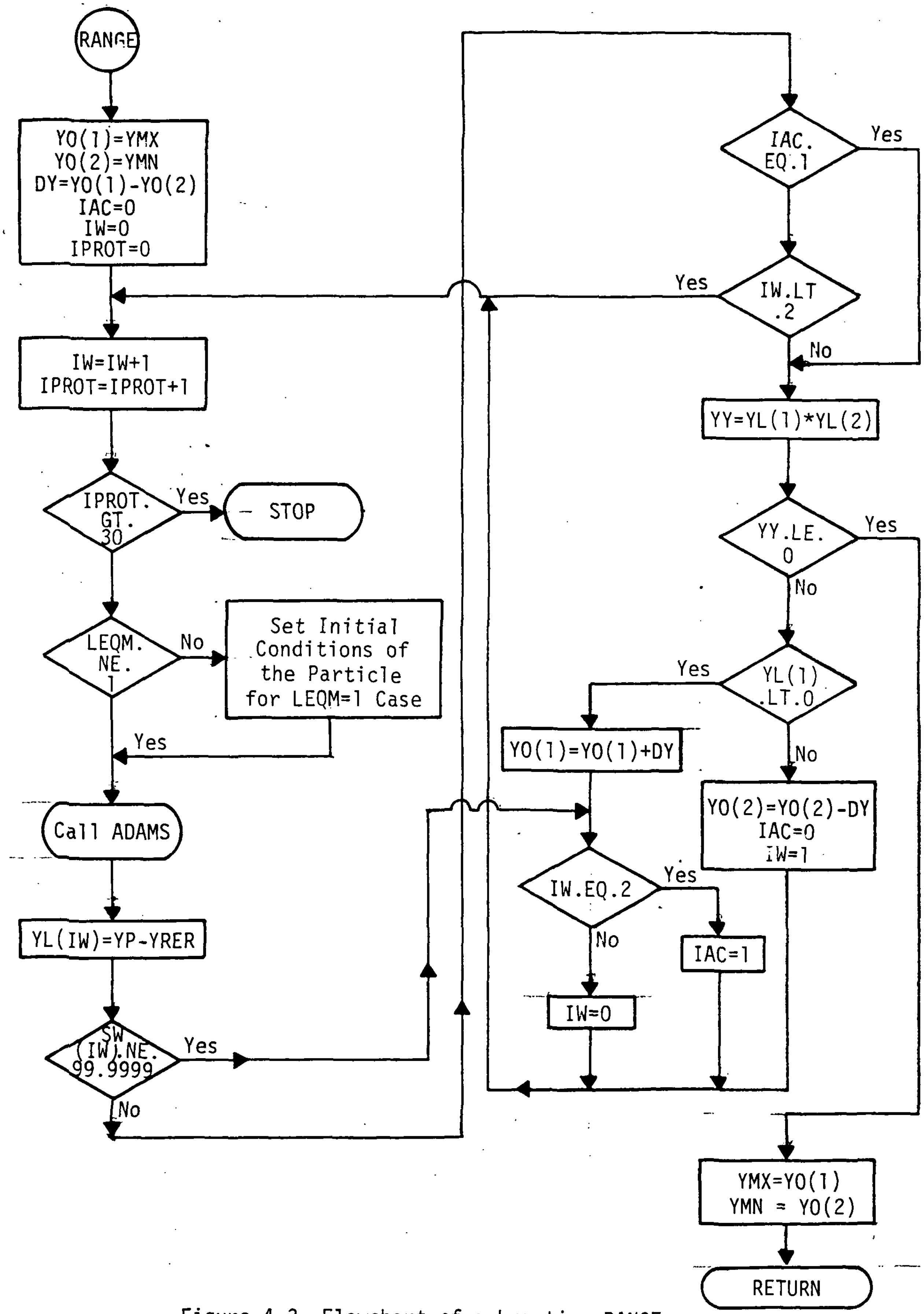


Figure 4.3 Flowchart of subroutine RANGE.

4.1.4 Subroutine F

Objective: Supply functional form of equations governing the particle motion.

Variables:

T	Time
X	Values of the functions (correspond to Y in Section 4.1)
XDOT	Derivatives of x with respect to time (i.e., $\dot{x}$, $\dot{y}$, $\dot{\theta}$, $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}$, see Section 2.2)
CL	Coefficient of lift force
CD	Coefficient of drag force
CM	Coefficient of moment

4.1.5 Subroutine COEFF

Objective: Supply aerodynamic coefficients CL, CD, and CM.

4.1.6 Subroutine MODE

Objective: Determine whether the particle impacts the body. Also, terminate the trajectory calculation for particles which move away from the bodies.

Flowchart: See Figure 4.4

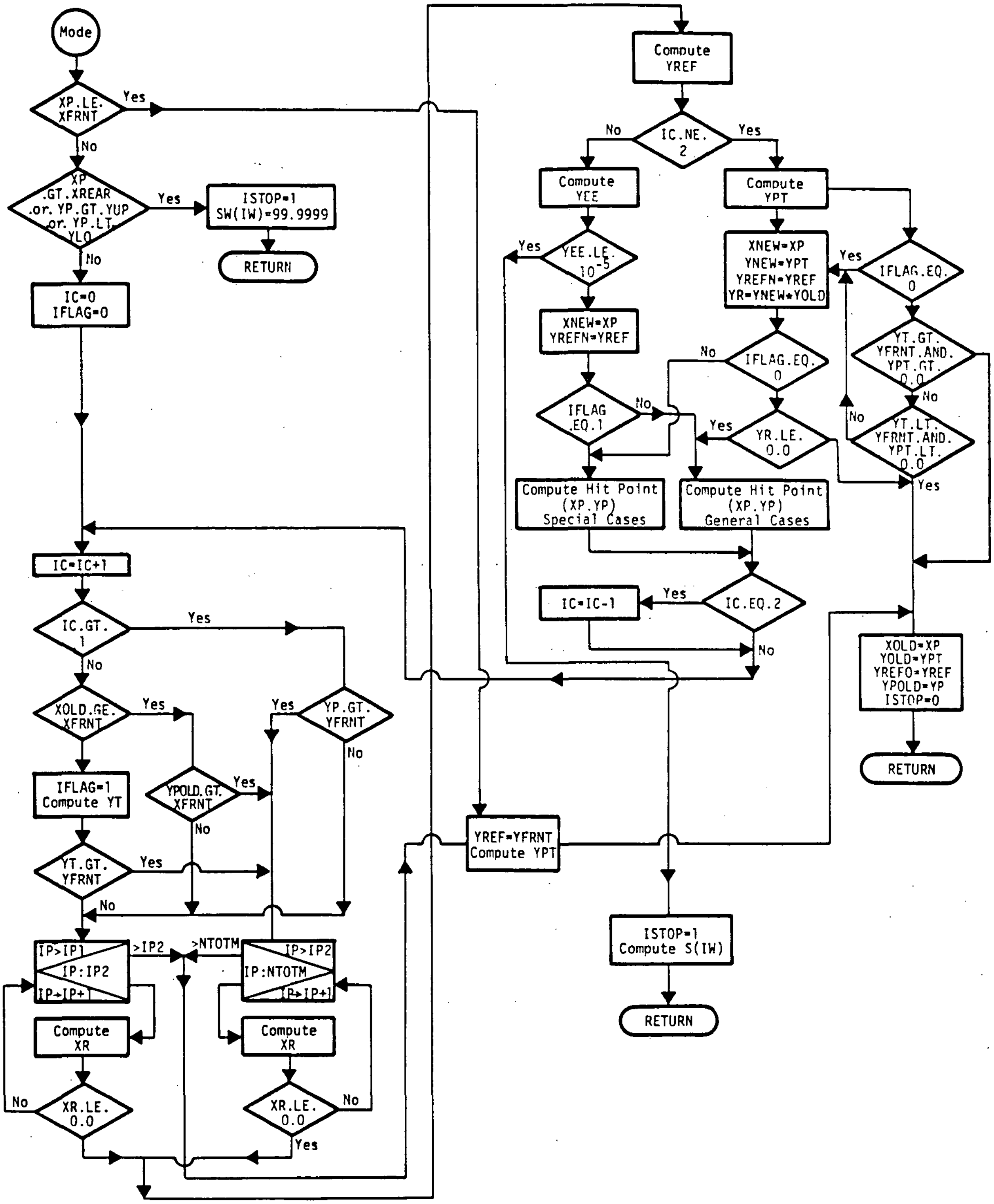


Figure 4.4 Flowchart of subroutine MODE.

4.1.7 Subroutine READIN

Objective: Read Tape 21 and Tape 05.

Flowchart: See Figure 4.5

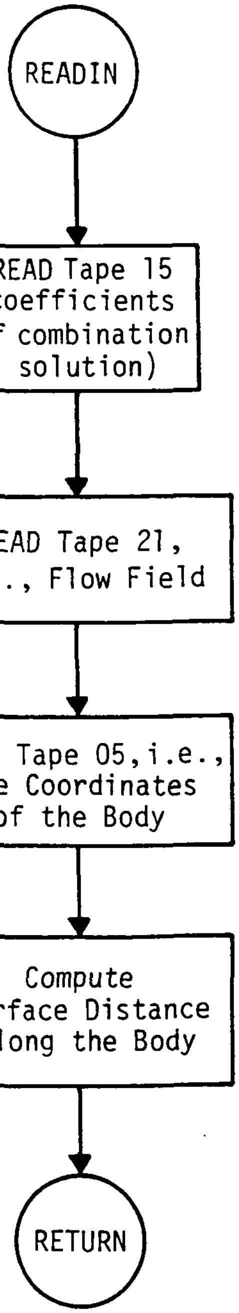


Figure 4.5 Flowchart of subroutine READIN.

4.1.8 Subroutine VELCTY

Objective: Compute flow velocities for a given position (x,y). Subroutine VELCTY calls subroutines MATRIX, AXIS, CROSS, SINTP, and VBARIT directly for each velocity calculation. Subroutines SORTXY, ARSIN, ELINT3, ELIP, ELLC, HLAMB, INEL, PARAB, PARAB2, QC, and SIMSON are called from EOD and COMBYN only once.

Variables:

X	Dimensionless x coordinate, x/ℓ_c, of particle position
Y	Dimensionless y coordinate, y/ℓ_c, of particle position
VXC	Dimensionless x component velocity, W_x/W_∞, of airflow
VYC	Dimensionless y component velocity, W_y/W_∞, of airflow

Flowchart: See Figure 4.6

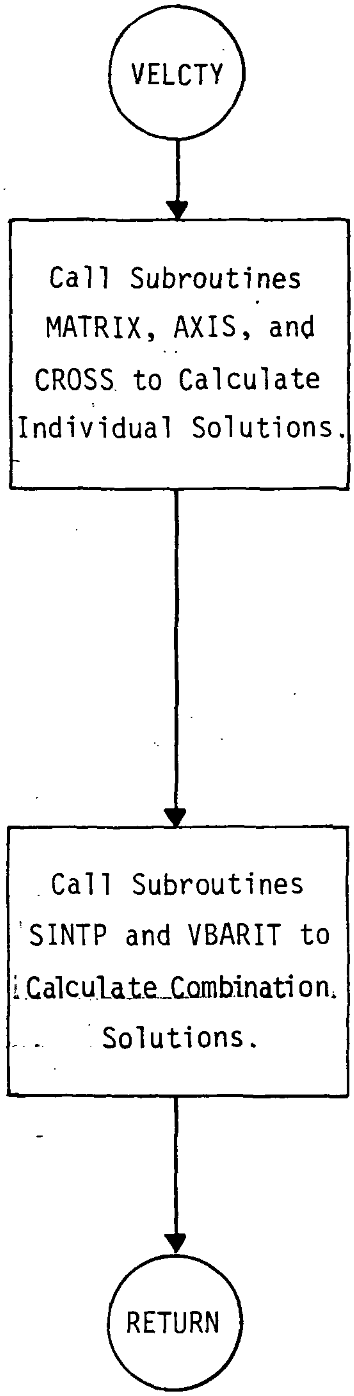


Figure 4.6 Flowchart of subroutine VELCTY.

4.1.9 Subroutine EFFICY

Objective: Calculate the local collection efficiency, β, using both a *linear approximation and a polynomial curve fit interpolation* technique.

Variables:

NCOEF	Number of coefficients for polynomial curve fit
NPTS	Number of data points used for each curve fitting procedure
NS	Do-loop index (see flowchart, Figure 4.7)
YP	Collection efficiency, $\beta = -dy_0/ds$
XIN	Value of $s(y_0)$

Flowchart: See Figure 4.7

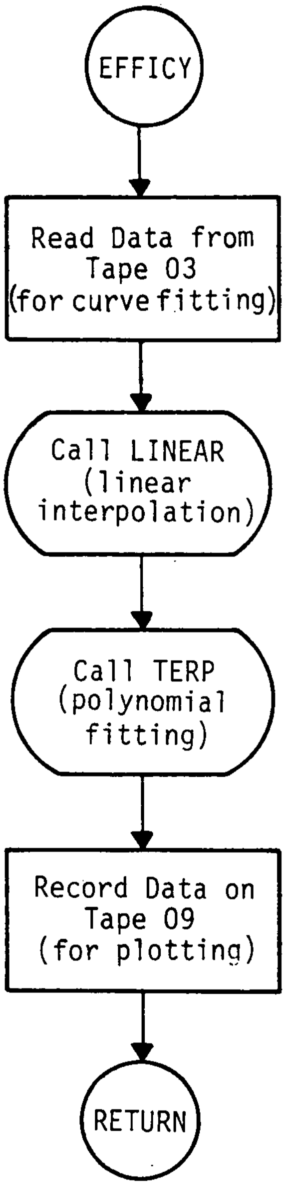


Figure 4.7 Flowchart of subroutine EFFICY.

4.1.10 Subroutine EF

Objective: Calculate the local collection efficiency, β, for multi-size particle distribution NSI>1 case using a polynomial curve fit interpolation technique.

4.1.11 Subroutine LINEAR

Objective: Compute the local collection efficiency by a linear approximation.

Variables:

X	Value of s
Y	Value of y_0
YP	Local collection efficiency, $-dy_0/ds$

4.1.12 Subroutine TERP

Objective: Curve fit the $\{s,y_0\}$ data to a polynomial function.

Variables:

XIN	x(s) coordinate at position (x,y)
YOUT	y(s) coordinate at (x,y) obtained by the polynomial data fit
YPOUT	Derivative of y with respect to x, dy/dx (dy_0/ds)
XA	Data set of x-coordinate
YA	Data set of y-coordinate
N1	Index of the first data point used for data fit
N2	Index of the last data point used for data fit
NCOEF	Number of coefficients in the polynomial function
NPTS	Number of data points used in each curve fit procedure

4.1.13 Subroutine CHOLES

Objective: Matrix solver called by subroutine TERP.

4.1.14 Subroutines APLOT and TRAJCT

Objective: Plot routines for β and for particle trajectories to be furnished by intended users.

4.2 Error Messages

Table 4.1 lists the error messages and recommended corrective actions to be taken for the computer program. The particular subroutine in which the error message occurs is also listed in the table. The table is essentially self descriptive and needs no further discussion.

TABLE 4.1 Error Messages and Corrective Actions.

Error Message Generated by the Program	Problem and Corrective Action	Error Location
The number of y-coordinates read is not equal to the number of x-coordinates read.	Corrective Action: Check the input data file, Tape 05, for equal number of x- and y-coordinates.	Subroutine READIN
Time-halving loops exceed 100.	Problem: The Adams-Moulton method of solution is not converging to the desired accuracy at the current time step. The problem itself sets NLOOP=100. The value is large enough for all cases investigated in this report where TIMSTP is set equal to 10^{-4}. Corrective Action: 1) Reduce the magnitude of TIMSTP. 2) Increase the size of ADEPS. 3) Check for severe gradients or error in the flow field.	Subroutine ADAMS
IPORT.GE.31 in SUB.RANGE--program stopped.	Problem: Initially guessed values of YMAX and YMIN are too far from the body or ΔY = YMAX - YMIN is too small. Corrective Action: 1) If particles pass below the body, increase the input values of YMAX and YMIN. 2) If particles pass above the body, decrease the input values of YMAX and YMIN. 3) Increase magnitude of ΔY.	Subroutine RANGE

TABLE 4.1 (continued).

Error Message Generated by the Program	Problem and Corrective Action	Error Location
FORMAT 172: Number of particle trajectories exceeds NSEAR, program stopped.	Problem: Number of particle trajectories which hit body exceeds the input value of NSEAR (Card 1 on Tape 02). Corrective Action: 1) Increase the input value of NSEAR. NSEAR=50 is recommended. 2) Increase the input value of YLIM (Card 5 on Tape 02). 3) Decrease the input value of NPL (Card 1 on Tape 02).	MAIN
FORMAT 747: Number of trajectories is more than 100, program stopped.	Problem: Number of trajectories calculated exceeds dimension state. Either the program is having difficulty finding limits or specified value of NPL is too large. Corrective Action: 1) Refer to the actions in error message #4. 2) Adjust input values of YMAX and YMIN (Card 7 on Tape 02--refer to error message #3).	MAIN
Particle released between the region ZL-ZU crosses upper or lower impingement limit trajectory, program stopped.	Corrective Action: Check the flow field around the body and increase absolute value of pseudo-surface DX (Card 2 on Tape 05).	MAIN

5.0 TEST CASES FOR PARTICLE TRAJECTORY CALCULATIONS

Two test cases for water droplet trajectory calculations are given in this section. Excessive CPU time is required with the EOD program, even with the significant improvements to accelerate the computation as described in Section 6.0. For this reason calculation of local collection coefficients is impractical. The calculation of sufficient trajectories, however, has been carried out for the test cases to fully verify the trajectory program. The test cases illustrated are:

1. Axisymmetric inlet at 0° angle of attack with M = 0.4.
2. Axisymmetric inlet at 30° angle of attack with M = 0.4.

An interpolation scheme was tested but found to have no advantage in terms of computational time for the specified test cases and, therefore, is not recommended (see Section 6.0 for details).

The input/output step-by-step procedures for applying the program to each test case is described. Details of test cases 1 and 2 are described in Sections 5.1 and 5.2, respectively. To fully understand the input/output printouts presented in these sections, the user should coordinate the results with the card structure and input instructions given in Section 3.2.

A source listing of the particle trajectory computer program is given in the appendix.

5.1 Axisymmetric Inlet at 0° Angle of Attack with Mach Number = 0.4

The geometry of the axisymmetric inlet is given in Figure 5.1. The input data for generating the geometry using the program SCIRCL is listed in Table 5.1. The output data file Tape 17 of SCIRCL (Table 5.2) is used directly as the input data file for the axisymmetric potential flow program (Section 3.2.2). The output data file Tape 07 of the potential flow program and Tape 05 (Table 5.3) created manually by the user (Section 3.2.3) are used as the input data files for the COMBYN

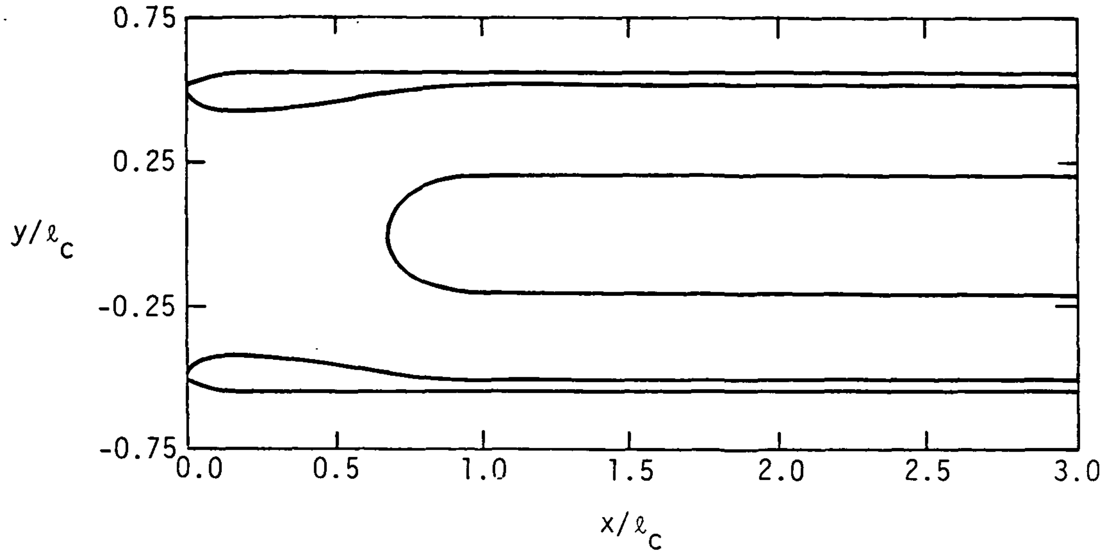


Figure 5.1 Geometry of axisymmetric inlet normalized by inlet diameter (11.68 units).

TABLE 5.1 Test Cases Computed with Particle Trajectory Program.

```
TEST CASE RELEASE 2.0

RFQC1     EOD
2.00         0.02          0.08          1.00          0.00
-0.0428  0.0       0.5565    10
0.0      0.0       0.4709    10
0.0428   0.0       0.0       10
0.0856   0.0       0.0       10
0.1284   0.0       0.0       10
0.1454   0.0       0.0       10
0.231    0.0       0.0       10
0.3166   0.0       0.0       10
0.9932   0.0       0.0       10
1.0          2.0
0.0          2.0          2.0
0.68587        0.68587                                         0.99409        3.08219
-1.00          0.00                                            0.20548        0.20548
1.00
0.99409        3.08219
0.20548        0.20548
2.0            6.0
1.0
3.08219        0.99409
0.51370        0.51370
-3.00
1.07971        0.99409        0.14538                          0.04281
0.51370        0.51370        0.42731                          0.42731
1000.0     2.0           2.0
1.25685        0.14538                                         0.00           0.00
0.42731        0.42731                                         0.5            0.6
1000.
0.00           0.00           0.02209        0.06627        0.22098        0.34247
0.34247        0.500          0.52389        0.53741        0.55248        0.55248
1.0
0.22098        0.99409
0.55248        0.55248
1.00
0.99409        3.08219
0.55248        0.55248
```

TABLE 5.2 Input Data File for Geometry Generation Program SCIRCL.

```
TEST CASE RELEASE 2.0                                                          2-BODY      BFQC.
2 1110              1                                                                    BFQC1
                                                                                         BFQC1
5000      56                                                                             BFQC1
          1            0            0                                                    BFQC1
   0.65587E+00  0.68685E+00  0.68978E+00  0.69460E+00  0.70101E+00  0.70690E+00
   0.71209E+00  0.72840E+00  0.73928E+00  0.75141E+00  0.76387E+00  0.77698E+00
   0.79054E+00  0.80477E+00  0.81932E+00  0.83421E+00  0.84941E+00  0.86467E+00
   0.88055E+00  0.89642E+00  0.91242E+00  0.92861E+00  0.94487E+00  0.96122E+00
   0.97762E+00  0.99409E+00  0.10122E+01  0.10339E+01  0.10599E+01  0.10912E+01
   0.11287E+01  0.11737E+01  0.12277E+01  0.12925E+01  0.13703E+01  0.14512E+01
   0.15334E+01  0.16149E+01  0.16964E+01  0.17779E+01  0.18594E+01  0.19409E+01
   0.20225E+01  0.21040E+01  0.21655E+01  0.22670E+01  0.23485E+01  0.24300E+01
   0.25115E+01  0.25931E+01  0.26746E+01  0.27561E+01  0.28376E+01  0.29192E+01
   0.30007E+01  0.30822E+01
   0.00000E+00  0.16372E-01  0.32630E-01  0.48046E-01  0.63603E-01  0.77935E-01
   0.91465E-01  0.10415E+00  0.11600E+00  0.12667E+00  0.13663E+00  0.14585E+00
   0.15436E+00  0.16213E+00  0.16925E+00  0.17557E+00  0.18144E+00  0.18655E+00
   0.19103E+00  0.19489E+00  0.19814E+00  0.20079E+00  0.20284E+00  0.20431E+00
   0.20519E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00
   0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00
   0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00
   0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00
   0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00  0.20548E+00
   0.20548E+00  0.20548E+00
5000     188                                                                     BFQC1
          2            0            0                                            BFQC1
   0.30822E+01  0.30007E+01  0.29192E+01  0.28376E+01  0.27561E+01  0.26746E+01
   0.25931E+01  0.25116E+01  0.24300E+01  0.23485E+01  0.22670E+01  0.21655E+01
   0.21040E+01  0.20225E+01  0.19409E+01  0.18594E+01  0.17779E+01  0.16964E+01
   0.16149E+01  0.15334E+01  0.14512E+01  0.13703E+01  0.12925E+01  0.12277E+01
   0.11737E+01  0.11287E+01  0.10912E+01  0.10599E+01  0.10339E+01  0.10122E+01
   0.99409E+00  0.97785E+00  0.96166E+00  0.94550E+00  0.92937E+00  0.91326E+00
   0.89716E+00  0.88105E+00  0.86500E+00  0.84891E+00  0.83282E+00  0.81671E+00
   0.80059E+00  0.78444E+00  0.76827E+00  0.75206E+00  0.73581E+00  0.71952E+00
   0.70319E+00  0.68680E+00  0.67035E+00  0.65384E+00  0.63726E+00  0.62061E+00
   0.60388E+00  0.58708E+00  0.57018E+00  0.55320E+00  0.53630E+00  0.51949E+00
   0.50276E+00  0.48611E+00  0.46973E+00  0.45301E+00  0.43656E+00  0.42017E+00
   0.40382E+00  0.38754E+00  0.37130E+00  0.35509E+00  0.33891E+00  0.32277E+00
   0.30664E+00  0.29054E+00  0.27445E+00  0.25836E+00  0.24228E+00  0.22619E+00
   0.21010E+00  0.19399E+00  0.17785E+00  0.16170E+00  0.14532E+00  0.12800E+00
   0.11162E+00  0.96094E-01  0.81174E-01  0.67765E-01  0.54979E-01  0.43141E-01
   0.32592E-01  0.23108E-01  0.14956E-01  0.86669E-02  0.39056E-02  0.94453E-03
   0.00000E+00  0.18558E-02  0.65746E-02  0.11913E-01  0.17632E-01  0.24281E-01
   0.31724E-01  0.40271E-01  0.49658E-01  0.60043E-01  0.71496E-01  0.84102E-01
   0.97757E-01  0.11317E+00  0.12986E+00  0.14799E+00  0.16616E+00  0.18436E+00
   0.20258E+00  0.22098E+00  0.23939E+00  0.25779E+00  0.27620E+00  0.29461E+00
   0.31302E+00  0.33142E+00  0.34983E+00  0.36824E+00  0.38665E+00  0.40505E+00
   0.42346E+00  0.44187E+00  0.46028E+00  0.47868E+00  0.49709E+00  0.51550E+00
   0.53391E+00  0.55231E+00  0.57072E+00  0.58913E+00  0.60754E+00  0.62594E+00
   0.64435E+00  0.66276E+00  0.68116E+00  0.69957E+00  0.71798E+00  0.73639E+00
   0.75479E+00  0.77320E+00  0.79161E+00  0.81002E+00  0.82842E+00  0.84683E+00
   0.86524E+00  0.88365E+00  0.90205E+00  0.92046E+00  0.93887E+00  0.95728E+00
   0.97568E+00  0.99409E+00  0.10125E+01  0.10346E+01  0.10611E+01  0.10929E+01
   0.11311E+01  0.11769E+01  0.12318E+01  0.12978E+01  0.13769E+01  0.14581E+01
   0.15393E+01  0.16206E+01  0.17018E+01  0.17830E+01  0.18642E+01  0.19454E+01
   0.20266E+01  0.21078E+01  0.21890E+01  0.22702E+01  0.23514E+01  0.24326E+01
   0.25138E+01  0.25950E+01  0.26762E+01  0.27574E+01  0.28386E+01  0.29198E+01
   0.30010E+01  0.30822E+01
   0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00
   0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00
   0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00
   0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00
   0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00  0.51370E+00
   0.51370E+00  0.51361E+00  0.51333E+00  0.51288E+00  0.51227E+00  0.51150E+00
   0.51058E+00  0.50951E+00  0.50831E+00  0.50698E+00  0.50553E+00  0.50396E+00
   0.50228E+00  0.50049E+00  0.49861E+00  0.49663E+00  0.49457E+00  0.49243E+00
   0.49021E+00  0.48792E+00  0.48558E+00  0.48318E+00  0.48073E+00  0.47824E+00
   0.47571E+00  0.47315E+00  0.47057E+00  0.46798E+00  0.46541E+00  0.46287E+00
   0.46036E+00  0.45790E+00  0.45549E+00  0.45313E+00  0.45084E+00  0.44861E+00
   0.44647E+00  0.44440E+00  0.44241E+00  0.44053E+00  0.43874E+00  0.43705E+00
   0.43548E+00  0.43403E+00  0.43270E+00  0.43150E+00  0.43043E+00  0.42951E+00
   0.42874E+00  0.42813E+00  0.42768E+00  0.42740E+00  0.42731E+00  0.42783E+00
   0.42930E+00  0.43161E+00  0.43471E+00  0.43854E+00  0.44307E+00  0.44832E+00
   0.45416E+00  0.46068E+00  0.46789E+00  0.47528E+00  0.48326E+00  0.49173E+00
   0.50009E+00  0.50811E+00  0.51415E+00  0.51833E+00  0.52170E+00  0.52487E+00
   0.52786E+00  0.53070E+00  0.53341E+00  0.53601E+00  0.53850E+00  0.54089E+00
   0.54315E+00  0.54528E+00  0.54725E+00  0.54902E+00  0.55042E+00  0.55148E+00
   0.55218E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55216E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00  0.55248E+00
   0.55248E+00  0.55248E+00
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5.2 (continued).

```
0000     90                                                                    8F7C1
           0              0              0                                     8FQC1
 -0.42800E-01   -0.42800E-01   -0.42800E-01   -0.42800E-01   -0.42800E-01   -0.42800E-01
 -0.42300E-01   -0.42300E-01   -0.42800E-01   -0.42800E-01    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    0.42800E-01    0.42800E-01    0.42600E-01    0.42800E-01
  0.42800E-01    0.42800E-01    0.42800E-01    0.42800E-01    0.42800E-01    0.42800E-01
  0.85600E-01    0.85600E-01    0.85600E-01    0.85600E-01    0.85600E-01    0.85600E-01
  0.85600E-01    0.85600E-01    0.85600E-01    0.85600E-01    0.12840F+00    0.12840E+00
  0.12840E+00    0.12840E+00    0.12840E+00    0.12840E+00    0.12840E+00    0.12840E+00
  0.12840E+00    0.12940E+00    0.14540E+00    0.14540E+00    0.14540E+00    0.14540E+00
  0.14540E+00    0.14540E+00    0.14540E+00    0.14540E+00    0.14540E+00    0.14540E+00
  0.23100E+00    0.23100E+00    0.23100E+00    0.23100E+00    0.23100E+00    0.23100E+00
  0.23100E+00    0.23100E+00    0.23100E+00    0.23100E+00    0.31660E+00    0.31660E+00
  0.31660E+00    0.31660E+00    0.31660E+00    0.31660E+00    0.31660E+00    0.31660E+00
  0.31660E+00    0.31660E+00    0.99320E+00    0.99320E+00    0.99320E+00    0.99320E+00
  0.99320E+00    0.99320E+00    0.99320E+00    0.99320E+00    0.99320E+00    0.99320E+00
  0.00000E+00    0.61833E-01    0.12367E+00    0.18550E+00    0.24733E+00    0.30917E+00
  0.37100E+00    0.43283E+00    0.49467E+00    0.55650E+00    0.00000E+00    0.52322E-01
  0.10464E+00    0.15697E+00    0.20929E+00    0.26161E+00    0.31393E+00    0.36626E+00
  0.41858E+00    0.47090E+00    0.00000E+00    0.48429E-01    0.96979E-01    0.14547E+00
  0.19396E+00    0.24244E+00    0.29093E+00    0.33942E+00    0.38791E+00    0.42640E+00
  0.00000E+00    0.46534E-01    0.93067E-01    0.13960E+00    0.18613E+00    0.23267E+00
  0.27920E+00    0.32373E+00    0.37227E+00    0.41880E+00    0.00000E+00    [illegible]
  0.91217E-01    0.13683E+00    0.18243E+00    0.22804E+00    0.27365E+00    [illegible]
  0.36487E+00    0.41048E+00    0.00000E+00    0.45666E-01    0.91331E-01    0.13700E+00
  0.18266E+00    0.22833E+00    0.27399E+00    0.31966E+00    0.36533E+00    0.41099E+00
  0.00000E+00    0.45962E-01    0.91924E-01    0.13789E+00    0.18385E+00    [illegible]
  0.27577E+00    0.32173E+00    0.36770E+00    0.41366E+00    0.00000E+00    0.44673E-01
  0.93337E-01    0.14008E+00    0.18677E+00    0.23347E+00    0.28016E+00    0.32663E+00
  0.37355E+00    0.42024E+00    0.22195E+00    0.25256E+00    0.28317E+00    0.31177E+00
  0.34440E+00    0.37501E+00    0.40562E+00    0.43623E+00    0.46683E+00    0.49746E+00
```

TABLE 5.3 Input Data File Tape 05 for the COMBYN Program.

```
              TEST CASE RELEASE 2.0
242   90 242   90    1    0    1    0
434.00  145.50  -90.000 458.180  0.00       0.000    .000    .000  714.410  0.00
   .000      .000  0.0000     .100
  2  81   90   -1    0    1    0    0
   .000 180.000
0.9932      0.2055      0.5137
    .00        .00       0.50    55
```

computer program. The output data file Tape 17 (renamed as Tape 05) of program SCIRCL, the output data file Tape 15 of program COMBYN, and the output data file Tape 21 generated by the potential flow program are used as the input data files for the trajectory program (Section 3.2.3).

In addition to the data files described above, the user must manually create data file Tape 02 (Table 5.4) for the trajectory program.

In data file Tape 02 (Table 5.4), the input value of NPL is 1 and the particle initial position is assigned at x = -0.5 and y = -0.7. In this case, we only consider one particle trajectory which is released from the prescribed position as data file Tape 02 (Card 6). The particle hits body 1 at x = 0.01325, y = -0.51912 with surface distance s = 0.02432 measured from the leading edge point. The particle released from x = -0.5 and y = -0.71 will miss the body from the lower side. Four particle trajectories are shown in Figure 5.2.

TABLE 5.4 Input Data File Tape 02 for the Particle Trajectory Program (0° angle of attack).

```
$NBDY     3      NPL$$     1      NSEAR   60      NEQ$$     4
$LEQM     1      LSYM$     0      LRANG    0
$LIM$     0      LOPT$     0      LPLOT    0      LTNL$     0      LXOR$     0
$G   $   0.0     VINF$   44.35
$DP  $  16.7     RL   $   1.00    TIMSP  0.0010   YLIM$   0.000001CF   $ 1.0
PRATK    0.0     PIT  $   0.0     PITDT  0.0      XORC$   -0.5    YORC$ -0.70
XREAR    0.40    YLO  $  -0.60    YUP  $ -0.4     YMAX$ 0.02      YMIN$-0.02
ADEPS    0.0010 ANLPL     0.0     ANLCR  0.0      EPSX$  0.001
NSI$$     1
INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
NCOEF     4      NPTS$    10      $ NS$     0
EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY :
    1
-0.08       0.045
INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
NCOEF     4      NPTS$    10      $ NS$     0
EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY :
    1
-0.08       0.045
INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
NCOEF     4      NPTS$    10      $ NS$     0
EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY :
    1
-0.08       0.045
```

---

*The names of the variables including those with $'s are printed to fill the appropriate spaces. Normally, these alphabetic symbols would not be input by the user but are given here to clarify the illustration.

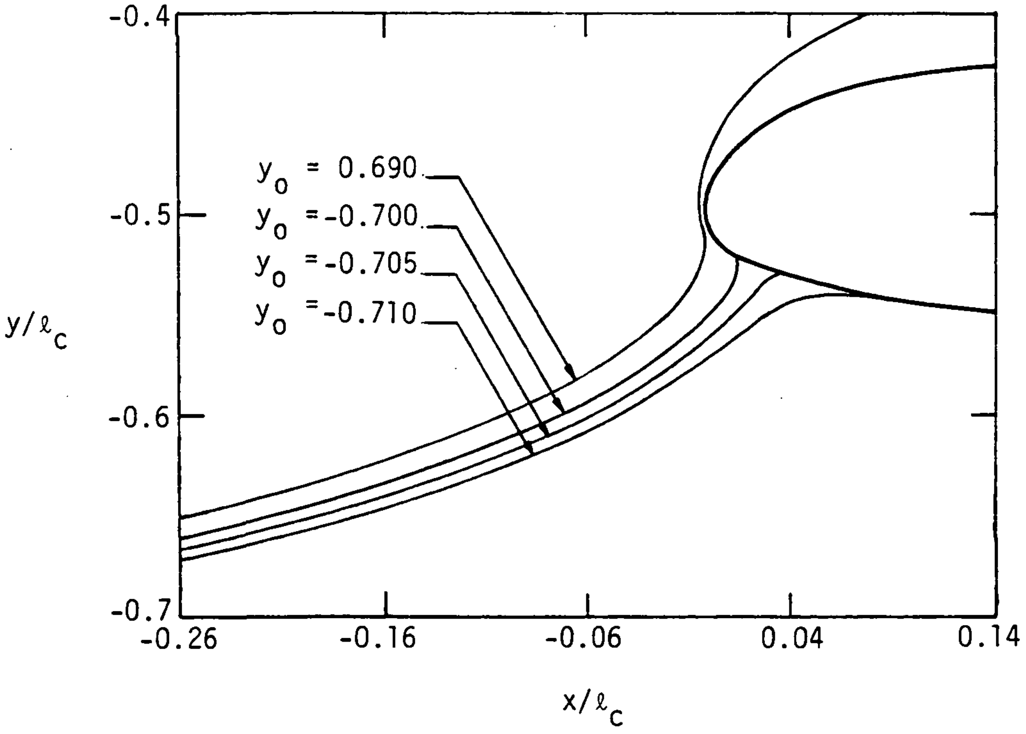


Figure 5.2 Particle trajectories released from x_o = -0.5, α = 0° (lower body).

ORIGINAL PAGE IS
OF POOR QUALITY

5.2 Axisymmetric Inlet at 30° Angle of Attack with Mach Number = 0.4

The geometry for this case is the same as that shown in Figure 5.1. The input data files for generating the geometry with the program SCIRCL and axisymmetric potential flow program are the same as those given in Tables 5.1 and 5.2. The output data file Tape 07 of the potential flow program and Tape 05 (Table 5.5) created manually by the user are used as the input data files for the COMBYN computer program. The output data file Tape 17 (renamed as Tape 05) of the SCIRCL program, the output data file Tape 21 of the potential flow program, the output data file Tape 15 of the COMBYN program, and the data file Tape 02 created manually by the user (Section 3.2.3) are used as input data files for the trajectory program. As in Section 5.1, only four particle trajectories are computed. The trajectories are shown in Figure 5.3.

TABLE 5.5 Input Data File Tape 05 for the Particle Trajectory Program (30° angle of attack).

```
               TEST CASE RELEASE 2.0
242   90 242   90    1    0    1    0
434.00  145.50  -60.000 438.180  0.00      0.000    .000    .000  914.410  0.01
   .000     .000  0.0000     .100
  2   81   90   -1    0    1    0    0
   .000 180.000
0.9932      0.2055      0.3137
    .00        .00       0.50     55
```

---

*The names of the variables including those with $'s are printed to fill the appropriate spaces. Normally, these alphabetic symbols would not be input by the user but are given here to clarify the illustration.

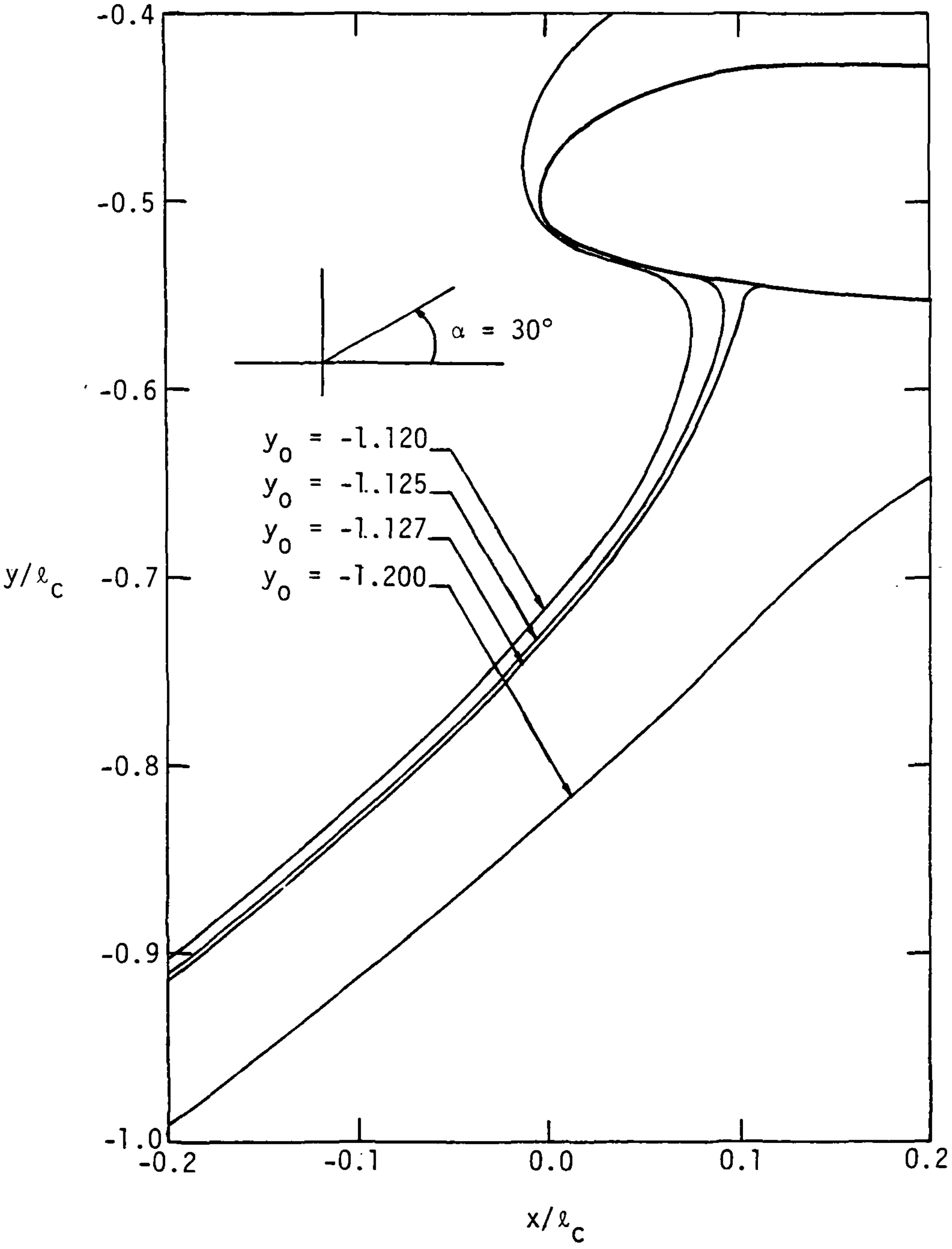


Figure 5.3 Particle trajectories released from x_0 = -0.5, α = 30°.

6.0 DISCUSSION OF RESULTS

This section discusses some of the limitations of the axisymmetric computer program in carrying out water droplet trajectory analyses. Section 6.1 describes modifications made to the original EOD program to accelerate the calculations of velocity at given points. Section 6.2 describes a mesh generator program which was developed to investigate interpolation methods. Although the mesh generator when used in two-dimensional analytical solutions gave very good agreement, it was found to have no inherent advantages in terms of computational time for either the two-dimensional or axisymmetrical case. It is, therefore, not recommended and has not been incorporated into the trajectory program. This and other recommendations are described in Section 6.3.

6.1 Modification to EOD Computer Program

The original EOD program computed velocities at off-body points in parallel with on-body point calculations. All element contributions were calculated simultaneously for both on- and off-body points under control of subroutine MATRIX. The on-body point contributions were used to compute the system matrix which is then inverted while the off-body point calculations were carried forward for later computations. In subroutine PARTY each of the densities is read back from disk and all on- and off-body velocities are calculated in terms for the axisymmetric and crossflows.

A straightforward application of EOD to calculate off-body velocities for the trajectories requires:

1. Prespecifying all off-body points.
2. Inversion of the system matrix.
3. Multiple disk file reads to evaluate the off-body velocities.

Repetition of Steps 1 to 3 is necessary to accommodate the trajectory solver which has no way to prespecify more than one point per trajectory before using that information to compute the next point.

EOD was supplemented with a program which:

1. Calculates the element contributions for one off-body point.

2. Uses previous values obtained for densities thus eliminating matrix inversion.

3. Retains all information required in memory during computation thus eliminating a substantial portion of execution time needed for disk reads.

EOD was modified to output the supplementary information needed by the trajectory solver. The supplement to the EOD code was then incorporated in the trajectory solver.

The calculation of velocity at a point is thus reduced to a straightforward integration. This integration is carried out by a Simpson rule algorithm which requires approximately one second* per point. Thus a typical particle trajectory calculation requiring approximately 500 iterations to achieve the accuracy needed to compute local collection coefficients uses approximately 8 to 9 minutes of CPU time just to compute the velocities. Computing the particle trajectory between time steps requires at most 0.15 second (1.25 minutes per trajectory). Considering that roughly 10 to 15 trajectories are necessary to fully define the local collection efficiency, β, 2 to 3 hours of CPU time are used to determine one plot of β. Since the axisymmetric program consumes 85 percent of the CPU time per calculation, a mesh generator with an interpolation routine was developed and tested to determine if the economics of the computational time could be improved.

6.2 Interpolation Method

A sophisticated mesh generation program was developed in order to investigate the feasibility of using interpolation methods to find velocity components after computing their values at the nodal points. It was believed that greater efficiency could be achieved in the

---

*All CPU times are based on a VAX 11/780 computer.

calculation of trajectories if it were not necessary to calculate all velocities on a trajectory from the potential code (the direct method). The results, while promising, were mixed.

For the two-dimensional flow case, the time required to generate the mesh was comparable to that required to run the specified test cases by the direct method. Once the mesh was generated, trajectory calculations were approximately ten times faster than by the direct method. However, adding the time required to generate the mesh, the total computation time of the local collection efficiency was essentially equal to and in some cases more than that of the direct method.

Accuracy was thoroughly tested on a Joukowski airfoil using both the analytic solution and the potential flow solver (24Y). Relative and absolute accuracies in the velocity components of 0.1 percent were maintained throughout the field.

For the axisymmetric case the mesh generation was considerably more time consuming. Even after eliminating matrix inversion, the potential code for this case (EOD) is a much slower procedure than 24Y consuming approximately one second per mesh point. Experience with the mesh generator indicated that approximately 17,000 points (17,000 seconds) would be required on the mesh to achieve 0.1 percent accuracy for the axisymmetric inlet case using the mesh generator. This gives more than five hours CPU time to generate the mesh. For the calculation of a local collection coefficient requiring a limited number of trajectories there is little question that the direct method is the faster procedure.

6.3 Recommendations

Based on the results of this study, the direct method of computing trajectories is recommended over the interpolation method. Both methods are extremely time consuming due to structure of the axisymmetric EOD computer program, despite the excessive modification to this program accelerates the velocity calculation. The EOD program is, therefore, not recommended as a viable computational tool for collection efficiency studies where numerous geometry configurations (whether due to design changes or due to ice build-up) are to be investigated. The physical principles of the axisymmetric program are, of course, sound and fully tested. Therefore, if the program is to be used for icing droplet trajectory studies, it is recommended that the computational logic be completely restructured using modern developments in computer science.

REFERENCES

1. Stockman, N. O., C. A. Farrell, Jr. "Improved Computer Programs for Calculating Potential Flow in Propulsion System Inlets," NASA TM 73728, July 1977.

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

3. Frost, W., H. P. Chang, C. F. Shieh, and K. R. Kimble. "Two-Dimensional Particle Trajectory Computer Program," Interim report prepared for NASA Lewis Research Center, Cleveland, Ohio, under Contract NAS3-22448, March 1982.

4. Gray, V. H. "Correction Among Ice Measurements, Impingement Rates, Icing Conditions, and Drag Coefficients for Upswept NACA 65_A-004 Airfoil," NACA TN 4151, February 1958.

5. Data provided by Dr. R. Joseph Shaw of NASA Lewis Research Center, Cleveland, Ohio 44135.

6. Conte, S. D., and C. de Boor. Elementary Numerical Analysis: An Algorithmic Approach, 2nd edition. New York: McGraw-Hill Book Company, 1972.

7. Bowden, D. T., A. E. Gensemer, and C. A. Skeen. FAA Technical Report ADS-4, pp. 2-1 - 2-76, December 1963.

APPENDIX

PARTICLE TRAJECTORY COMPUTER PROGRAM LISTING

```
      DIMENSION Y(6),XDOT(6),TITLE(20),XL(1000),YL(1000),N(10),
     1          CFP(10),EP(10)
C------------------------------------------------------------------------
C--      PROGRAM FOR CALCULATING PARTICLE TRAJECTORIES.
C--      MKS UNITS ARE USED.
C--      THE INITIAL INPUT OF PARTICLE DIAMETER IS IN MICRONS WHICH
C--      IS THEN CONVERTED TO METER IN THE PROGRAM.
C--- INPUT DATA FILES :
C--     TAPE02  :  INITIAL INPUT OF THIS TRAJECTORY CODE
C--     TAPE05  :  AEROFOIL GEOMETRY.   THE SAME FILE TAPE05
C-                 USED IN THE NASA POTENTIAL FLOW CODE
C--     TAPE21  :  THE SAME AS THE OUTPUT FILE TAPE21 OF NASA POTENTIAL
C-                 FLOW CODE.   USED FOR AIR VELOCITY CALCULATIONS.
C--- OUTPUT DATA FILES :
C--     TAPE01  :  DATA STORED IN TPAE01 IS WRITTEN TO TAPE03 LATTER.
C-                 TAPE01=TAPE03 FOR UNSYMMETRIC FLOW CASES
C--     TAPE03  :  DATA USED FOR THE CALCULATION OF COLLECTION EFFICIENCY
C--     TAPE04  :  CREATED IF (LOPT.NE.0) .  CONTAINS DETAILS OF THE
C-                 TRAJECTORIES AS WELL AS INFORMATION OF WIND FIELD.
C--     TAPE08  :  CREATED IF(LOPT .NE. 0)
C-                 DATA STORED IN TAPE08 IS FOR TRAJECTORY PLOTS.
C-                 THE DATA STORED ARE   XP,YP AND SW WITH THE
C-                 FORMAT(2X,3F10.5).     IF (SW.NE.88.8888) IMPLIES
C-                 THE END OF EACH TRAJECTORY.
C-                 XP, YP HAS BEEN NORMALIZED BY CHORD LENGTH.
C--     TAPE09  :  DATA FOR PLOTTING THE COLLECTION EFFICIENCY
C--------------------------------------------------------------------------
      COMMON/COMBYN/VVREF
      COMMON/OBY/XA(1000),YA(1000)
      COMMON/XTN/XFRNT,YFRNT,XRER,YRER,NTOTAL(5),NBDY,ID
      COMMON/XFR/XF(5),YF(5),XR(5),YR(5),THIK(5),YYMX(5),YYMN(5)
      COMMON/SDS/S(1000),SW(150),ZO(150),IS
      COMMON/IN2/XIN,YIN,DPM,RL,VXREF,VYREF,PITDOT,ATK,PIT,VREF,VXP,VYP
      COMMON/F1/Q,CBAR,AMASS,G,YYI,ALFAB,VISCOS
      COMMON/MOD/LOPT,LEQM,XREAR,YLO,YUP
      COMMON/IMM/IP1,IP2,NTOTM
      COMMON/FRI/IFRNT(5)
      COMMON/DPD1/NSI,PLWC(10),IRS(10),DPD(10),EPT
      COMMON/CDD/CF
C---- Y(1)=X, Y(2)=Z, Y(3)=X1DOT, Y(4)=Z1DOT, Y(5)=THET
C---- Y(6)=THETD
      REWIND 4
      REWIND 7
      REWIND 8
      DATA PI/3.141592654/
      READ(2,252) NBDY,NPL,NSEAR,NEQ
C---- CONTROL FLAGS
      READ(2,252) LEQM,LSYM,LRANG
      CALL READIN
      READ(2,252) LIM,LOPT,LPLOT,LTNL,LXOR
C---- INITIAL CONDITIONS
      READ(2,254) G,VINF
      READ(2,254) DP,RL,TIMSTP,YLIM,CF
      READ(2,254) PRATK,PIT,PITDOT,XORC,YORC
      READ(2,254) XREAR,YYLO,YYUP,YMAX,YMIN
      READ(2,254) ADEPS,ANLFLO,ANLCRD,EPSX
      READ(2,252) NSI
      IF(NSI.EQ.1)GO TO 2
      DO 3 I=1,NSI
      READ(2,221) PLWC(I),DPD(I),CFP(I)
    3 CONTINUE
    2 BDYATK=ANLFLO
      ANK=BDYATK+ANLCRD
      PEI=PI/180.
      BDYATK=BDYATK*PEI
      ANLCRD=ANLCRD*PEI
      ANLFLO=ANLFLO*PEI
      ADE=ADEPS
      ADEPS=VINF*ADEPS
      VXREF=VINF*COS(ANLFLO)
      VYREF=VINF*SIN(ANLFLO)
      LIMA=LIM
      VREF=SQRT(VXREF*VXREF+VYREF*VYREF)
      VVREF=VREF
      VXP=VXREF
      VYP=VYREF
      IF(NBDY.EQ.0)NBDY=1
      IF(ID.EQ.0)ID=1
      YMAX=YMAX*RL
      YMIN=YMIN*RL
      ATK=PRATK
      NSEA=NSEAR
C---- SEARCH THE PROPER X-LOCATION TO RELEASE PARTICALS
      IF(LXOR.EQ.0) GO TO 14
      NMAX=50
      XFRNT=XF(1)
      XIN=XFRNT-0.2
      EPS=EPSX
      DY=(YYUP-YYLO)/10.
      NC=0
  134 CONTINUE
      NC=NC+1
      DWX=0.
      XIN=XIN-0.2
      DO 131 I=1,11
      YP=YYLO+FLOAT(I-1)*DY
```

```
      CALL VELCTY(XIN,YP,W1,W2)
      VT=SQRT(W1*W1+W2*W2)
      DW=ABS(1.0-VT)
      IF(DW.GT.DWX) DWX=DW
  131 CONTINUE
      IF(NC.GE.NMAX) GO TO 135
      IF(DWX.GT.EPS) GO TO 134
  135 WRITE(6,133) XIN,NC,NMAX
      XIN=XIN*RL
      GO TO 15
   14 XIN=XORC*RL
   15 CONTINUE
C---- FINISH SEARCHING
      PEI=PI/180.
      DPM=DF*1.0E-06
      AREA=0.25*PI*DPM*DPM
      RHOP=1.0E03
      AMASS=PI*DPM**3*RHOP/6.0
      YYI=0.4*AMASS*(DPM/2.)**2
      CBAR=DPM
      RHO=1.23
      Q=0.5*RHO*AREA
      VISCOS=1.5E-05
      WRITE(6,439)DPM,RL,TIMSTP,NPL,VXREF,VYREF,AKK,PIT,PITDOT,G,
     1YYI,ADE
      NBY=NBDY
      IF(LTNL.EQ.0) GO TO 954
      NBY=1
      ANLCRD=0.0
  954 CONTINUE
      DO 94 IDK=1,NBY
      REWIND 3
      REWIND 9
      ID=IDK
      IF(LTNL .EQ. 1) ID=1
      H=TIMSTP
      XFRNT=XF(ID)
      YFRNT=YF(ID)
      XRER=XR(ID)
      IF(XREAR .GT. XRER) XREAR=XRER
      IP1=1
      NTOTM=NTOTAL(IDK)
      IF(IDK .NE. 1) IP1=NTOTAL(IDK-1)+1
      IP2=IFRNT(IDK)
      YRER=YR(ID)
      THICK=THIK(ID)
      YUP=YYMX(ID)+0.1*THIK(ID)
      YLO=YYMN(ID)-0.1*THIK(ID)
      AO=ABS(ANLFLO)
      YPDW=ABS(YR(ID)-YF(ID))
      IF(AO.GT.0.000001) THICK=YPDW+(XR(ID)-XF(ID))*TAN(BDYATK)
      WRITE(6,271)
      WRITE(6,272)NBDY,ID,XFRNT,YFRNT,XRER,YRER,THICK,YUP,YLO
      IF(ID.EQ.1) GO TO 97
      YMIN=ZU*RL+0.002
      YMAX=YMIN+0.5*ABS(YMIN)
   97 CONTINUE
      DO 211 IJK=1,NSI
      LIM=LIMA
      IS=0
      REWIND 1
      IF(NSI.EQ.1) GO TO 4
      WRITE(6,437) IJK
      DPM=DPD(IJK)*DF
      CF=CFP(IJK)
      DPM=DPM*1.0E-06
      AREA=0.25*PI*DPM*DPM
      AMASS=PI*DPM**3*RHOP/6.0
      YYI=0.4*AMASS*(DPM/2.)**2
      CBAR=DPM
      Q=0.5*RHO*AREA
      WRITE(6,438)DPD(IJK),PLWC(IJK),DPM,YYI
    4 CONTINUE
      IF(LEQM .EQ.1)WRITE(6,186)
      WRITE(6,433)
C---- COMPUTATION OF DROPLET TRAJECTORIES
C---- IF LIM=1 SEARCH THE SURFACE RANGE WHERE THE PARTICLES IMPACT AT
      IF(LRANG.NE.0) CALL RANGE(YMAX,YMIN,TIMSTP,ADEPS,NEQ)
      IW=0
      IUP=2
      YHIT=YMIN
      YMIS=YMAX
      IC=0
      ITO1=100
      NSEAR=NSEA
      ITOT=NSEAR
      IPROT=0
      IF(LIM.EQ.1) GO TO 41
      ITOT=NPL*2
      ITO1=NPL
      IW=NPL
      ZU=YYUP
      ZL=YYLO
      YUP=YYUP
      YLO=YYLO
   41 IUP=IUP-1
```

```
C---- IUP=1 SEARCHING UPPER LIMIT
C---- IUP=0 SEARCHING LOWER LIMIT
   42 CONTINUE
      TSP=TIMSTP
      IF(LIM .EQ. 0) GO TO 47
      IF(SW(IW).EQ.99.9999)IW=IW-1
      IF(ABS(YMIS-YHIT) .LE. YLIM) GO TO 75
      YIN=(YHIT+YMIS)/2.
      GO TO 48
   47 IF(NPL .NE. 1) GO TO 85
      YIN=YORC*RL
      GO TO 46
   85 AC=1.0
      DY=ZU-ZL
      IF(LSYM.EQ.1) DY=ZU
      DY=AC*DY/FLOAT(NPL-1)
      YIN=AC*ZU-DY*(IW-ITO1)
      YIN=YIN*RL
   46 IF(IW .EQ. ITOT) GO TO 82
   48 IW=IW+1
      IPROT=IPROT+1
      IF(IPROT.GT.100) GO TO 743
      IF(IW.EQ.1) YAD=YIN
      IF(IW.LE.NSEAR) GO TO 80
      WRITE(6,172)
      GO TO 236
C---- INITIALLY SW(I)=88.8888
C---- SW(I)=SURFACE DISTANCE IF PARTICLE HIT THE BODY
C---- SW(I)=99.9999 IF PARTICLE MOVES OUT OF RANGE
   80 SW(IW)=88.8888
      Y(1)=XIN
      Y(2)=YIN
      Y(3)=VXP
      Y(4)=VYP
      Y(5)=PIT*PEI
      Y(6)=PITDOT*PEI
      IF(LEQM.NE.1) GO TO 92
C---- SET INITIAL CONDITIONS OF THE CASE FOR WHICH THE PARTICLE
C----    IS EQUILIBRIUM IN THE AIR STREAM
      X=XIN/RL
      YP=YIN/RL
      CALL VELCTY(X,YP,WX,WY)
      Y(5)=ATAN(WY/WX)
      Y(3)=WX*VREF
      Y(4)=WY*VREF
      Y(6)=0.
   92 CONTINUE
      ALFAB=ATK*PEI
      CALL ADAMS(Y,XDOT,TIMSTP,ADEPS,IW,YP,NEQ)
      IF (IW.GT.ITO1.AND.SW(IW).EQ.99.9999) GO TO 798
      IF(LIM .EQ. 0) GO TO 42
C---- SEARCHING SURFACE LIMITS
      IF(SW(IW) .EQ. 99.9999) GO TO 72
      YHIT=YIN
      GO TO 42
   72 IF(IW .EQ. 1) GO TO 73
      YMIS=YIN
      GO TO 42
   73 IW=0
      IUP=2
      IF(YP.GT.YFRNT.OR.YP.GT.YRER) GO TO 787
      YHIT=YIN
      GO TO 41
  787 CONTINUE
      YMIS=YIN
      GO TO 41
  798 WRITE(6,793)
      GO TO 236
C---- UPPER LIMIT OR LOWER LIMIT IS FOUND
   75 IC=IC+1
      IF(IC .EQ. 1) GO TO 76
      SL=SW(IW)
      ZL=ZO(IW)
      GO TO 78
   76 CONTINUE
      SU=SW(IW)
      ZU=ZO(IW)
      YMIS=YMIN
      YHIT=YAD
      IF(LSYM.EQ.1) GO TO 79
      GO TO 41
   79 WRITE(6,101)
      SL=-SU
      ZL=-ZU
   78 WRITE(6,321) ZU,SU,ZL,SL
      ITO1=IW
C---- END OF LIMIT SEARCH
C---- TOTAL COLLECTION EFFICIENCY
      IF(THICK.EQ.0) GO TO 99
      E=(ZU-ZL)/THICK
      IF(NSI.GT. 1) EF(IJK)=E
      IF(NSI .GT. 1) WRITE(6,310)DPM,E
      IF(NSI.GT.1) GO TO 214
      WRITE(6,309) E
  214 CONTINUE
      WRITE(6,433)
```

```
      IF(NPL.EQ.0) GO TO 236
      LIM=0
      ITOT=IW+NPL
      NSEAR=ITOT
      ZU=ZU-0.00001
      ZL=ZL+0.00001
      GO TO 42
   82 CONTINUE
      IF(NPL.EQ.1) GO TO 19
      IA=0
      IST=1
  586 ZMAX=ZO(IST)
      IMX=IST
      I1=IST+1
      DO 575 I=I1,ITOT
      ABZ=ABS(ZO(I))
      IF(ABZ.LE.0.0000001) ZO(I)=0.0
      IF(ZO(I).LE.ZMAX) GO TO 575
      IMX=I
      ZMAX=ZO(I)
  575 CONTINUE
      IF(IST .EQ. 1) GO TO 551
      IF(ZO(IMX) .EQ. ZO(IST-1)) GO TO 552
  551 IF(SW(IMX) .GE. 99.9999) GO TO 552
      IF(LSYM.NE.1.AND.ZO(IMX).EQ.0.0) GO TO 552
      IA=IA+1
      WRITE(1) SW(IMX),ZO(IMX)
  552 ZDUM=ZO(IMX)
      SDUM=SW(IMX)
      ZO(IMX)=ZO(IST)
      SW(IMX)=SW(IST)
      ZO(IST)=ZDUM
      SW(IST)=SDUM
      IST=IST+1
      IF(IST .LE. ITOT) GO TO 586
      REWIND 1
      DO 232 I=1,IA
      READ(1) SW(I),ZO(I)
  232 CONTINUE
      IS=IA
      IF(LSYM.EQ.1) IS=IA*2-1
      WRITE(3) IS
      DO 234 I=1,IA
      WRITE(3) SW(I),ZO(I)
  234 CONTINUE
      IRS(IJK)=IS
      IF(IA.EQ.IS) GO TO 211
      IA=IA-1
      DO 233 I=1,IA
      J=IA-I+1
      SWA=-SW(J)
      ZOA=-ZO(J)
      WRITE(3) SWA,ZOA
  233 CONTINUE
  211 CONTINUE
      EPT=0.0
      IF(NSI .EQ. 1) GO TO 212
      DO 213 IK=1,NSI
      EPT=EPT+EP(IK)*PLWC(IK)
  213 CONTINUE
C---- CALCULATION OF LOCAL CATCH EFFICIENCY
  212 READ(2,120) (TITLE(I),I=1,20)
      READ(2,252) NCOEF,NPTS,NS
C---- READ NASA/LEWIS DATA WHICH IS THEN STORED IN TAPE09
      READ(2,120) (TITLE(I),I=1,20)
      READ(2,220) NASA
      WRITE(9) NASA
      DO 17 I=1,NASA
      READ(2,221)SFOIL,EFC
      WRITE(9) SFOIL,EFC
   17 CONTINUE
      CALL EFFICY(NCOEF,NPTS,NS)
   19 IF(LPLOT.EQ.0) GO TO 94
      IF(NPL .EQ. 1) GO TO 18
C---- DATA PLOTING
C---- LOCAL CATCH EFFICIENCY  -
      LDP=0
      REWIND 9
      NA=0
      NSET=2
      DO 11 K=1,NSET
      READ(9) NX
      N(K)=NX
      IN=NA+1
      NA=NA+NX
      DO 1 I=IN,NA
    1 READ(9) XL(I),YL(I)
   11 CONTINUE
      CALL APLOT(XL,YL,N,NSET,LDP,FC,CZ)
C---- AIRFOIL GEOMETRY
   18 CONTINUE
      LDP=0
      IF(IDK.GT.1) LDP=1
      N(1)=NTOT
      CALL APLOT(XA,YA,N,1,LDP,FC,CZ)
C---- TRAJECTORIES AND AIRFOIL GEOMETRY
```

```
      CALL TRAJCT
   94 CONTINUE
      GO TO 236
   99 WRITE(6,861)
      GO TO 236
  743 WRITE(6,747)
C---- FORMATS
  101 FORMAT(/,2X,'SYMMETRIC FLOW CASE IS COMPUTED',/)
  120 FORMAT(20A4)
  133 FORMAT(/,2X,'THE PARTICLES ARE RELEASED FROM X/RL=',1PE12.5,/,
     1     2X,'WHICH IS OBTAINED AT THE',I5,' LOOP OF',I5,' LOOPS')
  172 FORMAT(/,2X,'FORMAT 172:NO. OF PARTICLE TRAJECTORIES EXCEED
     1 NSEAR, PROGRAM STOPPED ')
  186 FORMAT(/,2X,'INITIAL CONDITIONS FOR THE CASE LEOM=1',/)
  220 FORMAT(I5)
  221 FORMAT(3F10.5)
  252 FORMAT(5(5X,I5,5X),I1)
  254 FORMAT(5(5X,F10.0))
  271 FORMAT(/,2X,'NBDY','  ID',12X,'XFRNT',10X,'YFRNT',10X,'XREAR',10X,
     1    'YREAR',10X,'THICK',10X,'YUPPP',10X,'YLOOO')
  272 FORMAT(2X,2I4,6X,7(3X,1PE12.5),/)
  309 FORMAT(//,2X,'TOTAL COLLECTION EFFICIENCY',3X,1PE12.5,/)
  310 FORMAT(/,2X,'TOTAL COLLECTION EFFICIENCY FOR THE DROPLET SIZE='
     1          ,E10.3,'  IS  ',E12.5,/)
  321 FORMAT(/,6X,'UPPER SURFACE LIMIT',9X,'LOWER SURFACE LIMIT',/,
     19X,'ZU',12X,'SU',12X,'ZL',12X,'SL',/,2X,4(E12.4,2X),/,2X)
  433 FORMAT(8X,'XO',8X,'YO',23X,'XP',8X,'YP',8X,'SD',8X,'DT',
     1    10X,'NSTP')
  437 FORMAT(///,1X,'***FOLLOWING OUTPUT IS FOR DROPLET SIZE
     1 DISTRIBUTION-- ',I3,/)
  438 FORMAT(/,2X,'DIAMETER',4X,'PERCENTAGE',4X,'PARTICLE',4X,'INERTIA',
     1          /,2X,'DISTRIBUT.',2X,'OF W. CONT.',3X,'DIAMETER',4X,
     2          'MOMENT',//,2X,4(1PE10.3,3X),/)
  439 FORMAT(/,2X,'PARTICLE',4X,'CHORD',5X,'TIME',5X,'NO. OF',5X,'VX',
     1         8X,'VY',8X,'ANGLE',5X,'PITCH',5X,'PIT',7X,'GRAVT ',5X,
     2         'INERT.',4X,'ADAMS',/,2X,'DIAMETER',4X,'LENGTH',4X,
     3         'STEP',5X,'DROPLET',4X,'REF.',6X,'REF.',5X,'OF ATK.',4X,
     4         'ANGLE',5X,'DOT',5X,'CONSTANT',4X,'MOMENT',2X,'CRITERIA',
     5          //,3(1PE9.2,1X),I9,1X,8(1PE9.2,1X),/)
  747 FORMAT(/,'FORMAT 747: NO. OF TRAJECTORIES IS MORE THAN
     1 100, PROGRAM STOPPED')
  764 FORMAT(/,2X,'DATA STORED IN TAPE',I3,2X,'NO. OF DATA',I5,/
     1      ,6X,'S/RL',5X,'YO/RL')
  793 FORMAT(/,2X,'PARTICLE RELEASED BETWEEN THE REGION ZL--ZU
     1 CROSSES UPPER OR LOWER IMPINGEMENT LIMIT TRAJECTORY,
     2 PROGRAM STOPPED')
  361 FORMAT(/,2X,'THICK=0.0--PROGRAM STOPED')
  236 STOP
      END
      SUBROUTINE ADAMS(YLAST,YD1,TSP,EPS,IW,YPL,NEQ)
C-----------------------------------------------------------------------
C--     EQUARTION SOLVER
C--     THE ADAMS-MOULTON PREDICTOR-CORRECTOR METHOD IS USED
C------------------------------------------------------------------------
      DIMENSION YD1(6),YLAST(6),YPRED(6),YCORR(6),YP(4,6),XPA(6),YPA(6)
      DIMENSION A1(6),A2(6),A3(6),A4(6)
      COMMON/XTN/XFRNT,YFRNT,XRER,YRER,NTOTAL(5),NBDY,ID
      COMMON/IN2/XIN,YIN,DP,RL,VXREF,VYREF,PITDOT,ATK,PIT,VREF,VXP,VYP
      COMMON/SDS/S(1000),SW(150),Z0(150),IS
      COMMON/MOD/LOPT,LEQM,XREAR,YLO,YUP
C---- IW,YPL, AND ALL THE COMMONS ARE USED EITHER
C---- FOR PROGRAM STOP CONTROL OR FOR OPTIONAL PRINTS   .
C---- LOOP FORM TIMESTEP TO START USING SIMPLE RUNGE-KUTTA METHOD
      H=TSP
      X=0.
      ISTOP=0
      NLOOP=100
      DX=TSP
      HO24=DX/24.0
      DO 11 NT=1,4
      X1=X
      DO 1 J=1,NEQ
      YPRED(J)=YLAST(J)
    1 YCORR(J)=YPRED(J)
      CALL F(X1,YCORR,YD1)
      DO 2 J=1,NEQ
      A1(J)=H*YD1(J)
    2 YCORR(J)=YPRED(J)+A1(J)/2.
      X1=X1+H/2.
      CALL F(X1,YCORR,YD1)
      DO 3 J=1,NEQ
      A2(J)=H*YD1(J)
    3 YCORR(J)=YPRED(J)+A2(J)/2.
      CALL F(X1,YCORR,YD1)
      DO 4 J=1,NEQ
      A3(J)=H*YD1(J)
    4 YCORR(J)=YPRED(J)+A3(J)
      X1=X1+H/2.
      CALL F(X1,YCORR,YD1)
      DO 5 J=1,NEQ
      A4(J)=H*YD1(J)
    5 YLAST(J)=YLAST(J)+(A1(J)+2.*(A2(J)+A3(J))+A4(J))/6.
      X=X1
      DO 11 NV=1,NEQ
   11 YP(NT,NV)=YD1(NV)
      NSTP=0
   20 CONTINUE
```

```
      DO 50 ILOOP=1,NLOOP
      X=X+DX
      NSTP=NSTP+1
C---- SIMPLE ADAMS PREDICTION
      DO 30 NV=1,NEQ
   30 YPRED(NV)=YLAST(NV)+HO24*(55.0*YP(4,NV)-59.0*YP(3,NV)
     1+37.0*YP(2,NV)-9.0*YP(1,NV))
      XPR=YPRED(1)/RL
      YPL=YPRED(2)/RL
      IF(XPR .LT. XFRNT) GO TO 111
      CALL MODE(XPR,YPL,ISTOP,IW,DX,NSTP)
      IF(ISTOP .NE. 0) GO TO 52
  111 CALL F(X,YPRED,YD1)
C---- BASFORD CORRECTION
      DLYMAX=0.0
      DO 31 NV=1,NEQ
      YCORR(NV)=YLAST(NV)+HO24*(9.0*YD1(NV)+19.0*YP(4,NV)
     1-5.0*YP(3,NV)+YP(2,NV))
      IF(NV.NE.3.AND.NV.NE.4) GO TO 31
      DLYMAX=DLYMAX+ABS((YPRED(NV)-YCORR(NV))/YLAST(NV))
   31 CONTINUE
      IF(ABS(DLYMAX) .LE. EPS) GO TO 32
C---- HALVE DX
      DO 42 NV=1,NEQ
      DO 41 I=1,4
      XPA(I)=FLOAT(I)
   41 YPA(I)=YP(I,NV)
      YP(2,NV)=YP(3,NV)
      CALL TERP(2.5,YP1,YPPD,XPA,YPA,1,4,3,4)
      YP(1,NV)=YP1
      CALL TERP(3.5,YP3,YPPD,XPA,YPA,1,4,3,4)
      YP(3,NV)=YP3
   42 CONTINUE
      NSTP=NSTP-1
      X=X-DX
      DX=0.50*DX
      HO24=DX/24.0
      GO TO 50
C---- SHIFT VARAIABLE FOR NEXT STEP
   32 CONTINUE
      DO 35 NV=1,NEQ
      YP(1,NV)=YP(2,NV)
      YP(2,NV)=YP(3,NV)
      YP(3,NV)=YP(4,NV)
      YP(4,NV)=YD1(NV)
   35 YLAST(NV)=YCORR(NV)
C---- PROGRAM STOP CONTROL AND OPTIONAL PRINT CONTROL
      XP=YLAST(1)/RL
      YPL=YLAST(2)/RL
      CALL MODE(XP,YPL,ISTOP,IW,DX,NSTP)
      IF(ISTOP.NE.0) GO TO 52
      IF(LOFT.EQ.0) GO TO 94
      WRITE(8,117) XP,YPL,SW(IW)
  117 FORMAT(3F10.5)
      CALL VELCTY (XP,YPL,W1,W2)
      WX=-W1*VREF
      WZ=-W2*VREF
      V=((YLAST(3)+WX)**2.0+(YLAST(4)+WZ)**2.0)**0.5
      ALFA=ATAN((-WZ-YLAST(4))/(WX+YLAST(3)))+YLAST(5)
      E1=YLAST(1)/RL
      E2=YLAST(2)/RL
      E3=YLAST(3)/VREF
      E4=YLAST(4)/VREF
      E5=V/VREF
      WRITE(4,105)X,E1,E2,YLAST(5),E3,E4,YLAST(6),E5,W1,W2
  105 FORMAT(10E10.3)
   94 CONTINUE
C---- *******   END OF STOP CONTROL     *********
      GO TO 20
   50 CONTINUE
      WRITE(6,100)
      STOP
  100 FORMAT(6X,'TIME-HALVING LOOPS EXCEED 100')
   52 RETURN
      END
      SUBROUTINE RANGE(YMX,YMN,TIMSTP,ADEPS,NEQ)
C-----------------------------------------------------------------------
C--   SEARCH THE UPSTREAM Y-RANGE SUCH THAT NONE OF THE PARTICLES
C--    OUTSIDE THE RANGE WILL HIT THE AIRFOIL.
C-----------------------------------------------------------------------
      DIMENSION YO(2),YL(2),Y(6),XDOT(6)
      COMMON/IN2/XIN,YIN,DP,RL,VXREF,VYREF,PITDOT,ATK,PIT,VREF,VXP,VYP
      COMMON/XTN/XFRNT,YFRNT,XRER,YRER,NTOTAL(5),NBDY,ID
      COMMON/SDS/S(1000),SW(150),ZO(150),IS
      COMMON/MOD/LOFT,LEQM,XREAR,YLO,YUP
      PI=3.141592654/180.
      YO(1)=YMX
      YO(2)=YMN
      IAC=0
      DY=1.0*(YO(1)-YO(2))
      IW=0
      IPROT=0
    1 ISTOP=0
      IW=IW+1
      IPROT=IPROT+1
      IF(IPROT.GT.30) GO TO 33
      SW(IW)=88.8888
```

```
      YIN=YO(IW)
      Y(1)=XIN
      Y(2)=YIN
      Y(3)=VXP
      Y(4)=VYP
      Y(5)=PIT*PI
      Y(6)=PITDOT*PI
      IF(LEQM.NE.1) GO TO 92
C---- SET INITIAL CONDITIONS OF THE CASE FOR WHICH THE PARTICLE
C---- IS EQUILIBRIUM IN THE AIR STREAM
      XINN=XIN/RL
      YINN=YIN/RL
      CALL VELCTY(XINN,YINN,WX,WY)
      Y(5)=TAN(WY/WX)
      Y(3)=WX*VREF
      Y(4)=WY*VREF
      Y(6)=0.
   92 CONTINUE
      YKL=YO(2)/RL
      YKU=YO(1)/RL
      IF(YKU.GT.YUP) YUP=YKU
      IF(YLO.GT.YKL) YLO=YKL
      CALL ADAMS(Y,XDOT,TIMSTP,ADEPS,IW,YP,NEQ)
      YL(IW)=YP-YRER
      IF(SW(IW).NE.99.9999) GO TO 10
      IF(IAC.EQ.1) GO TO 29
      IF(IW.LT.2) GO TO 1
   29 YY=YL(1)*YL(2)
      IF(YY.LE.0.0) GO TO 13
      IF(YL(1).LT.0.0) GO TO 11
      GO TO 12
   10 IF(IW.EQ.2) GO TO 12
   11 YO(1)=YO(1)+DY
      IF(IW.EQ.2) IAC=1
      IW=0
      GO TO 1
   12 YO(2)=YO(2)-DY
      IAC=0
      IW=1
      GO TO 1
   13 YMX=YO(1)
      YMN=YO(2)
      YO(1)=YO(1)/RL
      YO(2)=YO(2)/RL
      WRITE(6,21) YO(1),YO(2)
      GO TO 34
   33 WRITE(6,24)
      STOP
   21 FORMAT(/,2X,'YMAX=',1PE12.4,3X,'YMIN=',1PE12.4)
   24 FORMAT(//,'IPROT .GE. 31 IN SUB.RANGE - PROGRAM STOPPED')
   34 RETURN
      END
      SUBROUTINE F(T,X,XDOT)
C-----------------------------------------------------------------------
C--   GOVERNING EQUATIONS OF THE PARTICLE MOTION
C-----------------------------------------------------------------------
      DIMENSION X(6),XDOT(6)
      COMMON/IN2/XIN,YIN,DP,RL,VXREF,VYREF,PITDOT,ATK,PIT,VREF,VXP,VYP
      COMMON/F1/Q,CBAR,AMASS,G,YYI,ALFAB,VISCOS
      COMMON/CDD/CF
C---- XDOT(1)=X1DOT, XDOT(2)=Z1DOT, XDOT(3)=X2DOT
C---- XDOT(4)=Z2DOT, XDOT(5)=THETD, XDOT(6)=TH2DOT
      PI=3.141592654
      XP=X(1)/RL
      YP=X(2)/RL
      CALL VELCTY (XP,YP,W1,W2)
      WX=-VREF*W1
      WZ=-VREF*W2
      VX=X(3)+WX
      VY=X(4)+WZ
      V=SQRT(VX*VX+VY*VY)
      RE=V*DP/VISCOS
      CALL COEFF(RE,CD,CL,CM)
      CD=CD/CF
      IF(CL .EQ. 0.0 .AND. CM.EQ.0.0) GO TO 20
      IF(VX.EQ.0.) GO TO 17
      GAMMA=ATAN(VY/VX)
      IF(VX.LT.0.) GAMMA=GAMMA+PI
      IF(VX.GT.0.0.AND.VY.LT.0.0) GAMMA=GAMMA+2.*PI
      GO TO 19
   17 GAMMA=0.5*PI
      IF(VY.LT.0.) GAMMA=1.5*PI
   19 CONTINUE
      ALFA=X(5)-GAMMA
      ALFAD=ALFA-ALFAB
      ALFAB=ALFA
      AL=Q*V*V*CL
      AD=Q*V*V*CD
      AM=Q*V*V*CM*CBAR
      XDOT(1)=X(3)
      XDOT(2)=X(4)
      XDOT(5)=X(6)
      CGAMMA=COS(GAMMA)
      SGAMMA=SIN(GAMMA)
      XDOT(3)=-(AL/AMASS)*SGAMMA-(AD/AMASS)*CGAMMA
      XDOT(4)=(AL/AMASS)*CGAMMA-(AD/AMASS)*SGAMMA-G
      XDOT(6)=AM/YYI
```

```
      GO TO 21
   20 CONTINUE
      XDOT(1)=X(3)
      XDOT(2)=X(4)
      XDOT(5)=X(6)
      XDOT(3)=-CD*Q*V*VX/AMASS
      XDOT(4)=-CD*Q*V*VY/AMASS
      XDOT(6)=0.
   21 RETURN
      END
      SUBROUTINE CUEFF(RE,CD,CL,CM)
C-----------------------------------------------------------------------
C--   COEFFICIENTS OF AERODYNAMIC FORCES AND OF MOMENTS
C-----------------------------------------------------------------------
      CL=0.
      CM=0.
      IF(RE .EQ. 0.0) GO TO 22
      CD=24.0/RE
      IF(RE.LE.0.05) GO TO 14
      IF(RE.LE.3.0) GO TO 2
      IF(RE.GT.330) GO TO 5
      A0=-28.339
      A1=38.969
      A2=0.73204
      A3=-0.00056084
      GO TO 10
    2 A0=0.0
      A1=24.167
      A2=3.2540
      A3=-0.23564
      GO TO 10
    5 A0=0.0
      A1=93.462
      A2=0.37576
      A3=0.0
   10 R2=RE*RE
      CD=(A0+A1*RE+A2*R2+A3*R2*RE)/R2
      GO TO 14
   22 CD=0.0
   14 RETURN
      END
      SUBROUTINE MODE(XP,YP,ISTOP,IW,T,NSTP)
C-----------------------------------------------------------------------
C--   DECIDES WETHER THE PARTICLE HIT THE AIRFOIL OR NOT
C-----------------------------------------------------------------------
      COMMON/SDS/S(1000),SW(150),Z0(150),IS
      COMMON/XFR/XX(5),YY(5),XB(5),YB(5),THIK(5),YYMX(5),YYMN(5)
      COMMON/IN2/XIN,YIN,DP,RL,VXREF,VYREF,FITDOT,ATK,PIT,VREF,VXP,VYP
      COMMON/OBY/X(1000),Y(1000)
      COMMON/MOD/LOFT,LEQM,XREAR,YLO,YUP
      COMMON/XTN/XFRNT,YFRNT,XRER,YRER,NTOTAL(5),NBDY,ID
      COMMON/IMM/IP1,IP2,NTOTM
      WRITE(8,*),'XP,YP,NSTP',XP,YP,NSTP
      XF=(XP-XFRNT)*RL
      IF(ABS(XF).LE.0.00001) XP=XFRNT
      IC=0
      IFLAG=0
      II=0
      IF(XP.LE.XFRNT) GO TO 27
      IF(XP .GT. XREAR .OR. YP .GT. YUP .OR. YP .LT. YLO) GO TO 32
C---- FIND X-POSITION OF PARTICLE RELATIVE TO X(I) AND X(IL)
   13 IC=IC+1
      IF(IC .GT. 1) GO TO 7
      IF(XOLD .GE. XFRNT) GO TO 5
      IFLAG=1
      YT=(YP-YPOLD)/(XP-XOLD)*(XFRNT-XOLD)+YPOLD
      IF(YT .GT. YFRNT) GO TO 3
      GO TO 6
    5 IF(YPOLD .GT. YFRNT) GO TO 3
      GO TO 6
    7 IF(YP .GT. YFRNT) GO TO 3
    6 CONTINUE
      DO 2 IP=IP1,IP2
      I=IP2-IP+IP1
      IL=I-1
      IF(IL .EQ. 0) GO TO 27
      XR=(XP-X(I))*(XP-X(IL))
      IF(XR .LE. 0.) GO TO 29
    2 CONTINUE
      GO TO 27
    3 CONTINUE
      DO 4 IP=IP2,NTOTM
      IL=IP
      I=IP+1
      IF(I .EQ. NTOTM+1) GO TO 27
      XR=(XP-X(I))*(XP-X(IL))
      IF( XR .LE. 0.) GO TO 29
    4 CONTINUE
   27 YREF=YFRNT
      IF(YP .EQ. YFRNT.AND.XP.EQ.XFRNT) GO TO 51
      YPT=(YP-YREF)/(YREF-YLO)
      GO TO 33
   29 YREF=((Y(I)-Y(IL))/(X(I)-X(IL)))*(XP-X(IL))+Y(IL)
      IF(IC .NE. 2) GO TO 31
      II=II+1
      YEE=ABS(YREF-YP)
      IF(YEE .LE. 0.00001) GO TO 51
```

```
      IF(II .GT. 20) GO TO 51
      XNEW=XP
      YREFN=YREF
      IF(IFLAG .EQ. 1) GO TO 9
      GO TO 34
C---- FIND THE PARTICULAR TRAJ. ACROSS THE BODY SURFACE OR NOT
   31 YPT=(YP-YREF)/(YREF-YLO)
      IF(IFLAG .EQ. 0) GO TO 8
      IF(YT .GT. YFRNT .AND. YPT.GT. 0.0) GO TO 33
      IF(YT .LT. YFRNT .AND. YPT.LT. 0.0) GO TO 33
    8 XNEW=XP
      YNEW=YPT
      YREFN=YREF
      YR=YNEW*YOLD
C---- FIND INTERSECTION POINT OF SPECIAL CASE
      IF(IFLAG .EQ. 0) GO TO 86
    9 A=(XNEW-XFRNT)*(YP-YPOLD)-(YREFN-YFRNT)*(XNEW-XOLD)
      B=YP-YPOLD
      C=YREFN-YFRNT
      D=XNEW-XFRNT
      E=XNEW-XOLD
      XP=(B*D*XOLD-C*E*XFRNT+(YFRNT-YPOLD)*E*D)/A
      YP=(D*B*YFRNT-E*C*YPOLD+(XOLD-XFRNT)*B*C)/A
      GO TO 1
   86 IF(YR .LE. 0.0) GO TO 34
   33 XOLD=XP
      YOLD=YPT
      YREFO=YREF
      YPOLD=YP
      ISTOP=0
      RETURN
C---- FIND INTERSECTION POINT OF GENERAL CASE
   34 XP=(XNEW-XOLD)*(YPOLD-YREFO)/(YREFN-YREFO-YP+YPOLD)+XOLD
      YP=(YP-YPOLD)*(YPOLD-YREFO)/(YREFN-YREFO-YP+YPOLD)+YPOLD
    1 CONTINUE
      IF(IC.EQ.2) IC=IC-1
      GO TO 13
   51 YP=YREF
      GO TO 55
C---- PRINTS DATA IF PARTICLE MOVES OUT OF RANGE
   52 XN=XIN/RL
      YN=YIN/RL
      WRITE(6,53) XN,YN,XP,YP,T,NSTP
   53 FORMAT(2F10.5,'  OUT OF RANGE  ',2F10.5,10X,1PE12.4,I10)
      ISTOP=1
      SW(IW)=99.9999
      ZO(IW)=YN
      GO TO 57
C---- END OF DATA PRINT IF PARTICLE MOVES OUT OF RANGE
C---- INTERPOLATION OF SURFACE DISTANCE
   55 CONTINUE
      DO 24 IQ=IP1,NTOTM
      I=IQ
      IM=I+1
      IF(YP .GT. YFRNT) GO TO 25
      GO TO 21
   25 I=NTOTM+IP1-IQ
      IM=I-1
   21 XR1=(XP-X(I))*(XP-X(IM))
      IF(XR1.GT.0.) GO TO 24
      P=(S(I)-S(IM))/(X(I)-X(IM))*(XP-X(IM))+S(IM)
      GO TO 59
   24 CONTINUE
   59 CONTINUE
      YN=YIN/RL
      XN=XIN/RL
      SW(IW)=P
      IS=IS+1
      ZO(IW)=YN
      WRITE(6,56) XN,YN,XP,YP,P,T,NSTP
   56 FORMAT(2F10.5,'  HIT BODY AT   ',3F10.5,1PE12.4,I10)
      ISTOP=1
   57 CONTINUE
      RETURN
      END
      SUBROUTINE READIN
      DIMENSION A(500),KT(2)
      COMMON/COMBYN/VVREF
      COMMON/COMBYN1/JJ,NHUBP1,NHUBMX,ALIL,BLIL,RHOTOT,ATOTAL,
     1                XON(500),FACTO(2),CCOST(2),CSINT(2),
     2                A1(500,2)
      COMMON/XFR/XF(5),YF(5),XR(5),YR(5),THIK(5),YYMX(5),YYMN(5)
      COMMON/FRI/IFRNT(5)
      COMMON/OBY/X(1000),Y(1000)
      DIMENSION XX(500),YY(500)
      COMMON/SDS/S(1000),SW(150),ZO(150),IS
      COMMON/XTN/XFRNT,YFRNT,XRER,YRER,NTOTAL(5),NBDY,IBD
      COMMON /BLOCK1/ IGEOMF(9),ISIGF(9)
      COMMON /BLOCK2/ HCURV(500),HARC(500)
      COMMON /BLOCK3/NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,LSSE,SO,
     1NSMALL
      COMMON /BLOCK4/A11(500),A12(500),A13(500),A21(500),A22(500),
     1A23(500)
      COMMON/FLG/   HEDR(15)  ,CASE      ,NB        ,NNU
     1,FLG03     ,FLG04     ,FLG05     ,FLG06     ,FLG07
     1,FLG08     ,FLG09     ,FLG10     ,FLG11     ,FLG12
     1,FLG13     ,FLG14     ,FLG15     ,FLG16     ,FLG17
```

```
     1,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     1,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON/NNN/ NT,         ND(11),   MN,        NUNA(5),  TYPEA(5),
     1NER1,      NER2,      NMA,        NSIGA,     NSIGC,
     1NUNC(5),  TYPEC(5), NLF(11),  IEC,        NSIGEC,
     1TYPEEC(5),NUNEC(5)
      COMMON /KITZEL/ VXA(5),VYA(5),       SIGA(500,5),
     1                VXC(5),VYC(5),VZC(5),SIGC(500,5),
     2                VN(5),VT(5)
      DOUBLE PRECISION  HEDR, CASE
      INTEGER      FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07
     1,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     1,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     1,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     1,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      REAL         MN,LSSE,MX,MY
      INTEGER      BDN,SUBKS
      COMMON /CL/    X1(500),   Y1(500),   X2(500),   Y2(500),   DELS(500),
     1SINA(500),COSA(500)
      LOGICAL NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE
C---- READ INFORMATION FOR CALCULATING WIND VELOCITIES
      EPS=0.0000001
      DR=0.0174532925
      REWIND 15
      READ(15)NTHETA
      READ(15)JJ,NHUBP1,NHUBMX,ALIL,BLIL,RHOTOT,ATOTAL,THEA,XON
      DO 12 I=1,NTHETA
      READ(15)FACTOR,CCOSTH,CSINTH,A
      FACTO(I)=FACTOR
      CCOST(I)=CCOSTH
      CSINT(I)=CSINTH
      DO 11 J=1,500
   11 A1(J,I)=A(J)
   12 CONTINUE
      REWIND 21
      READ(21)IGEOMF,ISIGF
      READ(21)X1,Y1,X2,Y2,DELS,SINA,COSA
      READ(21)HEDR,CASE,NB,NNU,
     1        FLG03,FLG04,FLG05,FLG06,FLG07,FLG08,FLG09,FLG10,FLG11,
     2        FLG12,FLG13,FLG14,FLG15,FLG16,FLG17,FLG18.FLG19,FLG20,
     3        FLG21,FLG22,FLG23,FLG24,FLG25,FLG26,FLG27
      READ(21)NT,ND,MN,NUNA,TYPEA,NER1,NER2,NMA,NSIGA,NSIGC,NUNC,
     1        TYPEC,NLF,IEC,NSIGEC,TYPEEC,NUNEC
      READ(21)HCURV,HARC
      READ(21)NOAXI,NOCROS,NOVORT,NOV1,NOV2.FIRSTE,LASTE,
     1        LSSE,SO,NSMALL
      READ(21)A11,A12,A13,A21,A22,A23
      READ(21)SIGA,SIGC
      TYPE*,'FINISH READING'
      REWIND 5
C---- READ X-, Y-, COORDINATES
C---- READ TITLE
      NTOT=0
      READ(5,50)HEDR,CASE,PSF,NB
   50 FORMAT(15A4,5X.A4,7X,A4,/,28I1)
      READ(5,60)CHORD,MN,TCNST,EPSLON
   60 FORMAT(4F10.0)
      N11=1
      N12=0
      DO 30 K-1,NB
      READ(5,10) IGEOMF(K),ISIGF(K),ICURVN,NONEWF,IFORMT,NN.MX,
     1           MY,THETA,ADDX,ADDY
   10 FORMAT(5I1,I5,5F10.0)
      READ(5,20)BDN,SUBKS,NLF(K),XE,YE
   20 FORMAT(3(5X,I5),2F10.0)
      N22=N12+NN
      IF(IFORMT.EQ.2) GO TO 120
      IF(IFORMT.EQ.1) GO TO 110
C---- IFORMT=0
      READ(5,710)(XX(I),I=N11,N22)
      READ(5,710)(YY(I),I=N11,N22)
      GO TO 130
  110 READ(5,790)(XX(I),YY(I),I=N11,N22)
      GO TO 130
  120 READ(5,800)(XX(I),YY(I),I=N11,N22)
  710 FORMAT(6E13.4)
  790 FORMAT(2F10.5)
  800 FORMAT(F10.5,10X,F10.5)
  130 CONTINUE
      N11=N11+NN
      N12=N12+NN
      KT(K)=NN
   30 CONTINUE
      NTOTAL(1)=KT(2)
      NTOTAL(2)=KT(2)+KT(1)*2-1
      NTOTAL(3)=KT(2)+NTOTAL(2)
      DO 35 K=1,N22
      X(K)=XX(N22-K+1)
      Y(K)=-YY(N22-K+1)
      IK=N22+K
      IF(K.EQ.N22) GO TO 35
      X(IK)=XX(K+1)
      Y(IK)=YY(K+1)
   35 CONTINUE
      DO 41 K=1,NBDY
      IIB=NTOT+1
      IIA=NTOT+2
```

```
      NTOT=NTOTAL(K)
      S(IIB)=0.0
      XXO=X(IIB)
      NNO=1
      DO 2 I=IIA,NTOT
      DDX=X(I)-X(I-1)
      DDY=Y(I)-Y(I-1)
      S(I)=S(I-1)+SQRT(DDX*DDX+DDY*DDY)
      IF(X(I).GT.XXO) GO TO 2
      XXO=X(I)
      NNO=I
    2 CONTINUE
      SO=S(NNO)
      DO 4 I=IIB,NTOT
      S(I)=SO-S(I)
    4 CONTINUE
      IF(ABS(MX).LE.EPS)GO TO 200
      DO 210 I=IIB,NTOT
  210 X(I)=X(I)*MX
  200 IF(ABS(MY).LE. EPS) GO TO 220
      DO 230 I=IIB,NTOT
  230 Y(I)=Y(I)*MY
  220 IF(ABS(ADDX).LE.EPS) GO TO 41
      DO 270 I=IIB,NTOT
  270 X(I)=X(I)+ADDX
   41 CONTINUE
C---- CALCULATION OF SURFACE DISTANCE
      IA=1
      NTOT=NTOTAL(1)
      XMAX=X(1)
      XOO=X(1)
      YMIN=Y(1)
      YMAX=Y(1)
      DO 91 K=1,NBDY
      IF(K.EQ.1) GO TO 51
      IA=NTOTAL(K-1)+1
      NTOT=NTOTAL(K)
      XMAX=X(IA)
      XOO=X(IA)
      YMAX=Y(IA)
      YMIN=Y(IA)
   51 IB=IA+1
      NO=1
      DO 21 I=IB,NTOT
      IF(X(I).GE.XOO) GO TO 81
      XOO=X(I)
      NO=I
      GO TO 21
   81 IF(X(I).LT.XMAX) GO TO 21
      XMAX=X(I)
      NR=I
   21 CONTINUE
      IFRNT(K)=NO
      XF(K)=X(NO)
      YF(K)=Y(NO)
      XR(K)=X(NR)
      YR(K)=Y(NR)
      DO 61 I=IB,NTOT
      IF(Y(I).GT.YMIN) GO TO 62
      YMIN=Y(I)
      GO TO 61
   62 IF(Y(I).LT.YMAX) GO TO 61
      YMAX=Y(I)
   61 CONTINUE
      THIK(K)=YMAX-YMIN
      YYMX(K)=YMAX
      YYMN(K)=YMIN
   91 CONTINUE
      RETURN
      END
      SUBROUTINE VELCTY(XP,YP,VXCC,VYCC)
      COMMON/COMBYN/VVREF
      COMMON/COMBYN1/JJ,NHUBP1,NHUBMX,ALIL,BLIL,RHOTOT,ATOTAL,
     1                 XON(500),FACTOR(2).CCOSTH(2).CSINTH(2).
     2                 A(500,2)
      COMMON /KITZEL/ VXA(5),VYA(5),       SIGA(500.5).
     1                VXC(5),VYC(5),VZC(5),SIGC(500,5),
     2                VN(5),VT(5)
      VREF=VVREF*100./30.48
      XPKT=XP
      YPKT=ABS(YP)
C---- CALCULATE THE INDIVIDUAL SOLUTIONS
      CALL MATRIX(XPKT,YPKT)
      CALL AXIS
      CALL CROSS
C---- TO CALCULATE THE COMBINATION SOLUTION
      IT=1
      IF(YP.LT.0.0) IT=2
      EXCON=1.0954*ATOTAL*125./216.
      VSONCC=ATOTAL/SQRT(1.2)
      IF(XPKT.LE.XON(JJ))GO TO 115
      CALL SINTP(XON(NHUBP1),A(NHUBP1.IT),JJ-NHUBMX,XPKT,AR)
      GO TO 120
  115 AR=A(JJ,IT)
  120 VB=FACTOR(IT)/AR
      VXCC=ALIL*VXA(1)+BLIL*VXA(3)+CCOSTH(IT)*VXC(1)
      VYCC=ALIL*VYA(1)+BLIL*VYA(3)+CCOSTH(IT)*VYC(1)
```

```
      VZCC=CSINTH(IT)*VZC(1)
      CALL VBARIT(VB,ATOTAL,RHOTOT,RHB)
      VRESOF=SQRT(VXCC**2+VYCC**2+VZCC**2)
      RESVV=VRESOF
      RBFORT=RHB/RHOTOT
      VRESOF=VRESOF*(1./RBFORT)**(VRESOF/VB)
      VXCC=VRESOF*VXCC/RESVV
      VYCC=VRESOF*VYCC/RESVV
      VZCC=VRESOF*VZCC/RESVV
      IF(VRESOF.LT.VSONCC)GO TO 200
      VSAVE=VRESOF
      VCONA=.2*(VSAVE/ATOTAL)**2
      PSOPTC=(1.-VCONA)**3.5
      RHORTC=VSAVE*PSOPTC**.715/EXCON
      IF(RHORTC.EQ.0.) RHORTC=1.
      VRESOF=VSONCC*(1.+(VSAVE/VSONCC-1.)**(1./RHORTC))
      CHANGE=VRESOF/VSAVE
      VXCC=VXCC*CHANGE
      VYCC=VYCC*CHANGE
      VZCC=VZCC*CHANGE
  200 CONTINUE
      VXCC=VXCC/VREF
      VYCC=VYCC/VREF
      IF(IT.EQ.2)VYCC=-VYCC
      RETURN
      END
      SUBROUTINE SINTP(Z,W,N,X1,Y1)
      DIMENSION X(400),Y(400),Z(1),W(1)
      DO 10 I=1,N
      X(I)=Z(I)
   10 Y(I)=W(I)
      CALL SORTXY (X,Y,N)
      DO 15 I=1,N
      K=I
      IF (X1.GT.X(I)) GO TO 15
      IF (X1.EQ.X(I)) GO TO 20
      IF (X1.LT.X(I)) GO TO 25
   15 CONTINUE
   20 Y1=Y(K)
      GO TO 30
   25 IF (K.EQ.1) GO TO 35
      IF (K.EQ.N) K=N-1
      IF (X(K).EQ.X(K+1)) K=K-1
      W1=(X1-X(K))*(X1-X(K+1))/(X(K-1)-X(K))/(X(K-1)-X(K+1))
      W2=(X1-X(K-1))*(X1-X(K+1))/(X(K)-X(K-1))/(X(K)-X(K+1))
      W3=(X1-X(K-1))*(X1-X(K))/(X(K+1)-X(K-1))/(X(K+1)-X(K))
      Y1=Y(K-1)*W1+Y(K)*W2+Y(K+1)*W3
   30 RETURN
   35 Y1=0.0
      RETURN
      END
      SUBROUTINE SORTXY(X,Y,NPTS)
      DIMENSION X(100),Y(100)
   10 N=NPTS
   15 NN=N-1
   20 DO 55 KT=1,NN
      XMIN=X(KT)
      JAD=KT
      JKL=KT+1
   25 DO 45 JK=JKL,N
   30 IF (XMIN-X(JK)) 45,45,35
   35 XMIN=X(JK)
   40 JAD=JK
   45 CONTINUE
   50 YMIN=Y(JAD)
      X(JAD)=X(KT)
      Y(JAD)=Y(KT)
      X(KT)=XMIN
      Y(KT)=YMIN
   55 CONTINUE
      RETURN
      END
      SUBROUTINE VBARIT(VBAR,ATOTAL,RHOTOT,RHOBAR)
C
C     APPROACH 5
C     TO SOLVE VBAR COMP ITERATIVELY
C
      VCRIT=ATOTAL/SQRT(1.2)
      I=0
      VGUES=VBAR
   10 VGUESA=(VGUES/ATOTAL)**2
      A=1.0-.2*VGUESA
      B=A-VGUESA
      VCOMP=(VBAR-A**2.5*VGUES)/(A**1.5*B)+VGUES
      IF (ABS((VCOMP-VGUES)/VCOMP).LT..0001) GO TO 15
      I=I+1
      IF (VCOMP.GE.VCRIT) VCOMP=.5*(VGUES+VCRIT)
      VGUES=VCOMP
      IF (I.GT.20) GO TO 15
      GO TO 10
   15 RHOBAR=(1.0-.2*(VCOMP/ATOTAL)**2)**2.5*RHOTOT
      IF (I.GT.20) WRITE (6,20)VBAR,VCOMP,RHOBAR
      IF(I.GT.20) VBAR=VCOMP*RHOBAR/RHOTOT
   20 FORMAT (1H0,34HI EXCEEDS 20 ITERATIONS FOR RHOBAR,5X,7HVBAR = ,1PE
     110.3,2X,8HVCOMP = ,1PE10.3,2X,9HRHOBAR = ,1PE10.3/
     270H VBAR HAS BEEN REDUCED TO VCOMP*RHOBAR/RHOTOT,WHERE VCOMP=VCRIT
     3ICAL          )
```

```
      RETURN
      END
C**$$**$$**BEGINNING OF ELEMENT ASINE/            **$$**$$**C
      FUNCTION ARSIN(X)
      ARSIN=ASIN(X)
      RETURN
      END
C**$$**$$**BEGINNING OF ELEMENT AXIS/            **$$**$$**C
      SUBROUTINE AXIS
C              * COMPUTE AXISYMMETRIC VELOCITY COMPONENTS AND PRINT
      COMMON/NNN/ NT,          ND(11),    MN,        NUNA(5),  TYPEA(5),
     1NER1,      NER2,      NMA,        NSIGA,     NSIGC,
     1NUNC(5),  TYPEC(5), NLF(11),  IEC,         NSIGEC,
     1TYPEEC(5),NUNEC(5)
      REAL        MN
      COMMON /KITZEL/ VXA(5),VYA(5),        SIGA(500,5),
     1                VXC(5),VYC(5),VZC(5),SIGC(500,5),
     2                VN(5),VT(5)
      COMMON /TL/   A(500),    B(500),    AX(500),   AY(500),  AZ(500),
     1              CX(500),   CY(500),   CZ(500),   AXV(500),AYV(500),
     3              I,         J,         J1,        SJ,       DS,
     4              DX,        DY,        NI,        XJ,       YJ,
     5              XK,        EEK,       EKK,       K,        PF
      L=0
      DO 550 N=1,NSIGA
      SX=0.0
      SY=0.0
C              * NO. OF ELEMENTS LOOP
      DO 510 J=1,NT
      SX=SX+AX(J)*SIGA(J,N)
  510 SY=SY+AY(J)*SIGA(J,N)
      IF (N.NE.1) GO TO 520
      VXA(N) = SX+1.
      GO TO 540
  520 IF (NUNA(N-1).NE.123456) GO TO 530
      L=L+1
      VXA(N)=SX+VN(L)
      VYA(N)=SY+VT(L)
      GO TO 550
  530 VXA(N) = SX
  540 VYA(N) = SY
  550 CONTINUE
      IF (MN.EQ.0.0) GO TO 580
C              * MACH NO. ADJUSTMENT
      DO 560 N=1,NSICA
      VYA(N)=VYA(N)/SQRT(1.0-MN*MN)
  560 VXA(N)=(VXA(N)-1.)/(1.0-MN*MN)+1.
  580 CONTINUE
      RETURN
      END
C**$$**$$**BEGINNING OF ELEMENT CROS/            **$$**$$**C
      SUBROUTINE CROSS
C              * COMPUTE CROSS FLOW VELOCITY COMPONENTS AND PRINT
      COMMON/NNN/ NT,          ND(11),    MN,        NUNA(5).  TYPEA(5),
     1NER1,      NER2,      NMA,        NSIGA,     NSIGC,
     1NUNC(5),  TYPEC(5), NLF(11),  IEC,         NSIGEC,
     1TYPEEC(5),NUNEC(5)
      REAL        MN
      COMMON /KITZEL/ VXA(5),VYA(5),        SIGA(500,5),
     1                VXC(5),VYC(5),VZC(5),SIGC(500,5),
     2                VN(5),VT(5)
      COMMON /TL/   A(500),    B(500),    AX(500),   AY(500),  AZ(500),
     1              CX(500),   CY(500),   CZ(500),   AXV(500).AYV(500),
     3              I,         J,         J1,        SJ,       DS,
     4              DX,        DY,        NI,        XJ,       YJ,
     5              XK,        EEK,       EKK,       K,        PF
      LOGICAL PF
      DO 370 N=1,NSIGC
      SX=0.0
      SY=0.0
      SP=0.0
C              * NO. OF ELEMENTS LOOP
      DO 350 J=1,NT
      SX=SX+CX(J)*SIGC(J,N)
      SY=SY+CY(J)*SIGC(J,N)
  350 SP=SP+CZ(J)*SIGC(J,N)
      VXC(N)=SX
      VYC(N)=SY+1.
      VZC(N)=SP+1.
  370 CONTINUE
      RETURN
      END
C**$$**$$**BEGINNING OF ELEMENT ELINT/            **$$**$$**C
      SUBROUTINE ELINT3(XKSQ,XN,PHI,PIE)
C   THIS SUBROUTINE CALCULATES THE INCOMPLETE ELLIPTIC INTEGRAL OF THE
C   THIRD KIND. THE ARGUMENTS ARE'
C        XKSQ    VALUE OF K SQUARED
C        XN      VALUE OF MINUS ALPHA SQUARED
C        PHI     VALUE OF FUI
C        PIE     VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THIRD KIND
      DATA HP /1.570796/
      DATA ROUND /.0000050/
      SK=XKSQ
      FN=XN
      P=PHI
      IF (FN.EQ.-1.0.AND.SK.EQ.1.0) GO TO 480
      IF(SK.GT.1.) GO TO 470
```

**ORIGINAL PAGE IS
OF POOR QUALITY**

```
      IF(FN.LT.(-1.)) GO TO 470
      IF(P) 10 , 470 , 20
C     NORMALIZE PHI
   10 A=-1.
      P=-P
      GOTO 30
   20 A=1.
   30 B=1.
      SB=1.
      IF (ABS(P-1.570796    ).LE.10.0**(-7)) GO TO 100
      IF(P-HP) 110 , 100 , 40
   40 J=P/(2.*HP)
      XX=2*J
      P1=P-XX*HP
      P=HP
      B=-1.
      GOTO 100
   50 D=SUM
      B=0.
      IF(P1-HP) 60 , 70 , 80
   60 P=P1
      XXX=1.
      GOTO 110
   70 PIE=(XX+1.)*A*D
      GOTO 460
   80 XXX=-1.
      XX=XX+2.
      P=2.*HP-P1
      GOTO 110
   90 PIE=A*(XX*D+XXX*SUM)
      GOTO 460
  100 IF(SK.EQ.1.) GOTO 470
      IF(FN.EQ.(-1.)) GO TO 470
  110 IF(P.GT.10.E-4)GOTO 130
      IF(FN.GT.0.)GOTO 120
      SUM=P
      GOTO 440
  120 RRT=SQRT(FN)
      SUM=ATAN(P*RRT)/RRT
      GOTO 440
  130 S=SIN(P)
      S2=S**2
      C=COS(P)
      IF(SK.GT.0.64)GOTO 210
      IF(ABS(FN).GE.0.6)GOTO 160
C     POWER SERIES IN N AND K SQUARED
      SA=1.
      SB=SK/2.
      CB=S*C
      CA=P
      FM=0.
      SUM=P
      X=SUM*1.E-8
  140 SA=SB-SA*FN
      CA=(-CB/(2.*(FM+1.)))+(1.-.5/(FM+1.))*CA
      Y=SA*CA
      SUM=SUM+Y
      IF((SB*CA).GT.X) GO TO 150
      IF(ABS(Y) .LT.X) GO TO 440
  150 FM=FM+1.
      CB=CB*S2
      SB=(1.-.5/(FM+1.))*SK*SB
      GOTO 140
C     POWER SERIES IN K SQUARED
  160 FK=SK
      RT=SQRT(1.+FN)
      IF(RT.NE.0.) GO TO 170
      G=S/C
      GOTO 190
  170 IF(C.GT.4.E-3)GOTO 180
      G=(HP-(C/(RT*S)))/RT
      GOTO 190
  180 G=ATAN(RT*S/C)/RT
  190 C1=G*1.E-8
      E=P
      F=S*C
      H=1.
      SUM=G
      FM=0.
  200 G=(E-G)/FN
      H=H*(1.-0.5/(FM+1.))
      G2=H*G*FK
      SUM=SUM+G2
      IF(G2.LE.G1)GOTO 440
      FM=FM+1.
      E=-F/(2.*FM)+(1.-0.5/FM)*E
      F=F*S2
      FK=FK*SK
      GOTO 200
  210 SKP=1.-SK
      IF(S.LT.C) GO TO 320
C     ADDITION FORMULA
      ZP=SQRT(1.-SK*S2)
      RT1=SQRT(ABS(FN*(FN+1.)*(FN+SK)))
      SST=(1.-ZP)/(SK*(C+1.))
      KP=(SST*S*RT1)/(1.+FN*S2-FN*SST*C*ZP)
      IF(FN) 220 , 290 , 250
```

```
  220 IF(RT1.NE.0.) GO TO 240
      R=S/(C+1.)
      IF(FN.NE.(-1.)) GO TO 230
      CF=(2.*R-SK*SST*S-(S/C)*ZP)/SKP
      GOTO 300
  230 CF=(SST*(S-(2./R))+S*C/ZP)*(SK/SKP)
      GOTO 300
  240 IF(FN*(FN+SK) .LT.0.) GO TO 260
  250 CF=(FN/RT1)*ATAN(XP)
      GOTO 300
  260 IF(ABS(XP).GE.0.1)GOTO 270
      YX=XP**2
      YX=2.*XP*(1.+YX*(1./3.+YX*(.2+YX/7.)))
      GOTO 280
  270 YX=ALOG((1.+XP)/(1.-XP))
  280 CF=(FN*YX)/(2.*RT1)
      GOTO 300
  290 CF=0.
  300 BB=-1.
      S=SQRT(SST)
      C=SQRT(1.-SST)
      GOTO 320
  310 SUM=2.*SUM-CF
      GOTO 440
  320 U=S/C
      V=1./C
      T=U*V
      W=U**2
      IF(S.GT.0.1)GOTO 330
      R=S*(1.+S2*(1./3.+S2*(.2+S2/7.)))
      GOTO 340
  330 R=ALOG(U+V)
  340 D=1.+FN
      IF(D.GT.SKP) GOTO 360
C     POWER SERIES IN 1+N AND 1 - (K SQUARED)
      CA=1.
      CB=-0.5*SKP
      AL=(T+R)/2.0
      BE=U*V**3
      FM=0.
      SUM=AL
      T1=SUM*1.E-8
  350 CA=-D*CA+CB
      AL=(BE-(2.*FM+1.)*AL)/(2.*(FM+2.))
      X=CA*AL
      SUM=SUM+X
      IF(ABS(X).LT.T1)GOTO 430
      FM=FM+1.0
      CB=-((2.*FM+1.)/(2.*(FM+1.)))*CB*SKP
      IF(ABS(BE).LT.10.E-30) BE=0.
      BE=BE*W
      GOTO 350
C     POWER SERIES IN 1 - (K SQUARED)
  360 RT=SQRT(ABS(FN))
      IF(FN) 370 , 380 , 390
  370 Q=ALOG((1.+RT*S)/(1.-RT*S))/(2.*RT)
      GOTO 400
  380 Q=S
      GOTO 400
  390 Q=ATAN(RT*S)/RT
  400 SUM=FN*Q+R
      PKP=SKP
      AP=-0.5
      FM=1.
      T1=SUM*1.E-8
  410 Q=(R-Q)/D
      R=T/(2.*FM)-(1.-.5/FM)*R
      X=AP*(FN*Q+R)*PKP
      SUM=SUM+X
      IF(ABS(X).LT.T1)GOTO 420
      T=T*W
      PKP=PKP*SKP
      FM=FM+1.
      AP=-AP*(1.-.5/FM)
      GOTO 410
  420 SUM=SUM/D
  430 IF(BB.LT.0.)GOTO 310
  440 IF(B) 50 , 90 , 450
  450 PIE=A*SUM
  460 PIE=PIE+PIE*ROUND
      RETURN
C     ERROR RETURN
  470 PIE=0.
      GOTO 460
C   CASE OF PI(1,1,PHI)
  480 PIE = 0.5*(TAN(P )/ COS(P )+ALOG(TAN((HP+P )/2.0)))
      GO TO 460
      END
C**$$**$$**BEGINNING OF ELEMENT ELIP/                **$$**$$**C
      SUBROUTINE ELIP(XK,EEK,EKK)
C                  * START
      ETA = 1. - XK
      IF (ETA.LE.0.0)ETA=.000005
   40 ELN=ALOG(ETA)
      EKK = 1.386294E0      + ETA * (.9666344E-1    + ETA *
     1      (.3590092E-1    + ETA * (.3742564E-1    + ETA *
     2      .1451196E-1   ))) - ELN * (.5 + ETA * (.1249859E0
```

```
     3       ETA * (.6880249E-1     + ETA * (.3328355E-1     + ETA *
     4       .4417870E-2  ))))
      EEK = 1. + ETA * (.4432514E0      + ETA * (.6260601E-1    + ETA *
     1       (.4757384E-1     + ETA * .1736506E-1    ))) - ELN * (ETA *
     2       (.2499837E0     + ETA * (.9200180E-1     + ETA *
     3       (.4069698E-1     + ETA * .5264496E-2  ))))
      RETURN
      END
C***$**$$**BEGINNING OF ELEMENT ELLLC/                    **$$***$$**C
      SUBROUTINE ELLC (A,K,E,I)
C   THIS SUBROUTINE CALCULATES THE ASSOCIATED COMPLETE ELLIPTIC INTEGRALS
C   OF THE FIRST OR SECOND KIND
C   THE ARGUMENTS ARE'
C         A    ARGUMENT (K SQUARED) FOR WHICH E' OR K' WILL BE FOUND
C         K    VALUE OF ASSOCIATED COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND
C         E    VALUE OF ASSOCIATED COMPLETE ELLIPTIC INTEGRAL OF SECOND KIN
C         I    IF EQ 1, COMPUTE K ; IF EQ 2, COMPUTE E
      DOUBLE PRECISION K,E,CON(32),A,LN4,CF(29),CL(3),DLOG
      EQUIVALENCE (CON(1),CF(1)), (CON(30),CL(1))
      DATA CF  /9.6573590797589018D-2,3.0885573486752694D-2,1.4978988178
     1704629D-2,9.6587579861753113D-3,1.1208913554644092D-2,1.3855601247
     215656D-2,6.6905509906897936D-3,6.4998443329390180D-4,1.2499999999411
     37923D-1,7.0312426464627361D-2,4.8818058565403952D-2,3.7068398993415
     45422D-2,2.7189861116788250D-2,1.4105380776153048D-2,3.1331309927862
     5886D-3,1.5049181783601883D-4,4.4314718112155306D-1,5.6805657874695
     6358D-2,2.1876220647186198D-2,1.2510592410844644D-2,1.3034146073731
     7432D-2,1.5377102528552019D-2,7.3356164974290365D-3,7.0980964089997
     8229D-4,2.4999999993617622D-1,9.3749920249680113D-2,5.6582839536559
     9024D-2,4.23828074569479D-2,3.0302747729412848D-2 /
      DATA CL  / 1.5525129948040721D-2,3.4838679435896492D-3,1.642721079
     17048025D-4 /
      LN4 = 1.38629436111989D0
      IF (A.EQ.0.0) GO TO 40
      GO TO ( 10 , 20 ),I
   10 K = LN4 + ((((((((CON(8)*A+CON(7))*A+CON(6))*A+CON(5))*A+CON(4))*A
     1+ CON(3))*A+CON(2))*A+CON(1))*A - DLOG(A)*(0.5+(((((((CON(16)*A+
     2CON(15))*A+CON(14))*A+CON(13))*A+CON(12))*A+CON(11))*A+CON(10))*A
     3+ CON(9))*A)
      GO TO 30
   20 E = 1.0D0+(((((((CON(24)*A+CON(23))*A+CON(22))*A+CON(21))*A+CON(20
     1))*A+CON(19))*A+CON(18))*A+CON(17))*A - DLOG(A)*((((((((CON(32)*A
     2+ CON(31))*A+CON(30))*A+CON(29))*A+CON(28))*A+CON(27))*A+CON(26))*
     3A+CON(25))*A)
   30 RETURN
   40 K = 0.99999999D30
      E = 1.0D0
      RETURN
      END
C***$**$$**BEGINNING OF ELEMENT HLAMB/                    **$***$$**C
      FUNCTION HLAMB (BETA,K)
C   THIS SUBROUTINE CALCULATES THE HEUMAN'S LAMBDA FUNCTION OF BETA AND
      DOUBLE PRECISION A,F,E
      REAL K
      DATA TWOP/0.6366197724/
      CALL INEL (FI,EI,PI,BETA,BETA,1.0-K**2          ,0,1,1)
      A = 1.0 - K  **2
      CALL ELLC (A              ,F,E,1)
      CALL ELLC (A              ,F,E,2)
      HLAMB = TWOP*(F*EI +(E-F)*FI)
      RETURN
      END
C***$***$**BEGINNING OF ELEMENT INNEL/                    **$$***$$**C
      SUBROUTINE INEL (F,E,PI,A,PHI,SKI,K3,K2,K1)
C   THIS SUBROUTINE CALCULATES THE INCOMPLETE ELLIPTIC INTEGRALS OF THE
C   FIRST, SECOND AND THIRD KINDS. THE ARGUMENTS ARE'
C         F       VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE FIRST KIND
C         E       VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE SECOND KIND
C         PI      VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THE THIRD KIND
C         A       VALUE OF ALPHA SQUARED
C         PHI     VALUE OF PHI
C         SKI     VALUE OF K SQUAREDO
C         K3      IF EQ 0, DO NOT COMPUTE PI ; IF NE 0, COMPUTE PI
C         K2      IF EQ 0, DO NOT COMPUTE E  ; IF NE 0, COMPUTE E
C         K1      IF EQ 0, DO NOT COMPUTE F  ; IF NE 0, COMPUTE F
      DOUBLE PRECISION ARG,FD,ED
      DATA PIT/1.57079633/
      E=0.0
      F=0.0
      IF (K3.EQ.0) GO TO 10
      CALL ELINT3 (SKI,-A,PHI,PI)
   10 IF (K1.EQ.0) GO TO 30
      IF (ABS(PHI-PIT).GT.10.0**(-7)) GO TO 20
      ARG=1.0-SKI
      CALL ELLC (ARG,FD,ED,1)
      F=FD
      GO TO 30
   20 CALL ELINT3 (SKI,0.0,PHI,F)
   30 IF (K2.EQ.0) GO TO 50
      IF (ABS(PHI-PIT).GT.10.0**(-7)) GO TO 40
      ARG=1.0-SKI
      CALL ELLC (ARG,FD,ED,2)
      E=ED
      GO TO 50
   40 CALL ELINT3 (SKI,-SKI,PHI,E)
      E=(1.0-SKI)*E+0.5*SKI*SIN(2.0*PHI)/SQRT(1.0-SKI*SIN(PHI)**2)
   50 RETURN
      END
```

```
C**********BEGINNING OF ELEMENT MATX/            **********C
      SUBROUTINE MATRIX(XPKT,YPKT)
C
C                 * COMPUTE MATRIX A,B,Z OR X,Y,Z
C
      COMMON /BLOCK1/ IGEOMF(9),ISIGF(9)
      COMMON /BLOCK3/NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,LSSE,SO,
     1 NSMALL
      COMMON/FLG/    HEDR(15)   ,CASE        ,NB         ,NNU
     1,FLG03      ,FLG04      ,FLG05      ,FLG06      ,FLG07
     1,FLG08      ,FLG09      ,FLG10      ,FLG11      ,FLG12
     1,FLG13      ,FLG14      ,FLG15      ,FLG16      ,FLG17
     1,FLG18      ,FLG19      ,FLG20      ,FLG21      ,FLG22
     1,FLG23      ,FLG24      ,FLG25      ,FLG26      ,FLG27
      COMMON/NNN/ NT,         ND(11),     MN,        NUNA(5),   TYPEA(5),
     1NER1,      NER2,      NMA,        NSIGA,     NSIGC,
     1NUNC(5),   TYPEC(5), NLF(11),   IEC,       NSIGEC,
     1TYPEEC(5),NUNEC(5)
      DOUBLE PRECISION  HEDR, CASE
      INTEGER       FLG03       ,FLG04      ,FLG05      ,FLG06      ,FLG07      MTRX 240
     1             ,FLG08       ,FLG09      ,FLG10      ,FLG11      ,FLG12      MTRX 250
     2             ,FLG13       ,FLG14      ,FLG15      ,FLG16      ,FLG17      MTRX 260
     3             ,FLG18       ,FLG19      ,FLG20      ,FLG21      ,FLG22      MTRX 270
     4             ,FLG23       ,FLG24      ,FLG25      ,FLG26      ,FLG27
      REAL          MN
      LOGICAL NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,PF
      REAL LSSE
C
      COMMON /TL/     A(500),     B(500),     AX(500),    AY(500),   AZ(500),
     1                CX(500),    CY(500),    CZ(500),    AXV(500),AYV(500),
     3                I,          J,          J1,         SJ,        DS,
     4                DX,         DY,         NI,         XJ,        YJ,
     5                XK,         EEK,        EKK,        K,         PF
      COMMON /KITZEL/ VXA(5),VYA(5),        SIGA(500,5),
     1                 VXC(5),VYC(5),VZC(5),SIGC(500,5),
     2                 VN(5),VT(5)
C
C                 * START
C                 * INITIALIZE
      DO 120 J = 1,5
      VN(J) = 0.0
  120 VT(J) = 0.
      J1=0
      N1 = 0
C INITIALIZE THE ENTIRE ROW ...
      DO 150 J=1,NT
      AX(J) = 0.0
      AY(J) = 0.0
      AZ(J) = 0.0
      CX(J) = 0.0
      CY(J) = 0.0
      CZ(J) = 0.0
      AXV(J) = 0.0
  150 AYV(J) = 0.0
      DO 220 K=1,NB
      NOV1 = .FALSE.
      NOV2 = .FALSE.
      IF (ISIGF(K) .EQ. 1) NOV2 = .TRUE.
      IF (ISIGF(K) .EQ. 2) NOV1 = .TRUE.
C BEGIN MATRIX FORMULATION OF THE COLUMNS OF THIS ITH ROW AS INFLUENCED
C BY THIS KTH BODY ...
      M1 = N1 + 1
      N1 = N1 + ND(K) - 1
      FIRSTE = .TRUE.
      LASTE = .FALSE.
      DO 170 J=M1,N1
      J1 = J1 + 1
      IF (J .EQ. N1) LASTE = .TRUE.
      CALL PARAB(XPKT,YPKT)
  170 FIRSTE = .FALSE.
      J1 = J1 + 1
C
C MATRIX FORMULATION OF COLUMNS FOR THIS BODY HAVE BEEN COMPLETED.
      IF(FLG15.LE.0)GO TO 220
      IF (NLF(K).GT.0) GO TO 220
      DO 190 J = M1, N1
      VN(K) = VN(K)+AXV(J)
  190 VT(K) = VT(K)+AYV(J)
  220 CONTINUE
      RETURN
      END
C**********BEGINNING OF ELEMENT PRAB/            **********C
      SUBROUTINE PARAB(XPKT,YPKT)
C     OVERLAY (PAN,1,2)
C     PROGRAM PARAB(XPKT,YPKT)
C CONTROL FOR PARABOLIC X,Y,Z MATRICES COMPUTATION.
      COMMON /CL/     X1(500),    Y1(500),    X2(500),    Y2(500),   DELL(500),
     1                SINA(500),COSA(500)
      COMMON /TL/     A(500),     B(500),     AX(500),    AY(500),   AZ(500),
     1                CX(500),    CY(500),    CZ(500),    AXV(500),AYV(500),
     3                I,          J,          J1,         SJ,        DS,
     4                DX,         DY,         NI,         XJ,        YJ,
     5                XK,         EEK,        EKK,        K,         PF
      COMMON /BLOCK2/ HCURV(500),HARC(500)
      COMMON /BLOCK3/NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,LSSE,SO,
     1NSMALL
      LOGICAL NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE
```

```
      REAL  LSSE
C OFF-BODY ...
      J1P1 = J1 + 1
      D1 = (XPKT-X1(J1))**2 + (YPKT-Y1(J1))**2
      D2 = (XPKT-X2(J))**2 + (YPKT-Y2(J))**2
      D3 = (XPKT-X1(J1P1))**2 + (YPKT-Y1(J1P1))**2
      DMINSQ = D1
      IF (D2 .LT. DMINSQ) DMINSQ = D2
      IF (D3 .LT. DMINSQ) DMINSQ = D3
      DMIN = SQRT(DMINSQ)
      IF (DMIN .EQ. 0.0) GO TO 60
      NI = 16.*HARC(J)/DMIN + 0.9
      IF (NI .GT. 0) GO TO 50
      NI = 3
      DS = HARC(J)
      GO TO 80
   50 NI = 2*NI
      IF (NI .LT. 128) GO TO 70
   60 NI = 129
      DS = HARC(J)/64.
      GO TO 80
   70 XNI = NI
      DS = 2.*HARC(J)/XNI
      NI = NI + 1
   80 XJ = X2(J)
      YJ = Y2(J)
      SO = -HARC(J)
      CALL PARAB2(XPKT,YPKT)
      RETURN
      END
C*****$***BEGINNING OF ELEMENT PRAB2/               **$$***$$**C
      SUBROUTINE PARAB2(XPKT,YPKT)
C     THIS ROUTINE CALCULATES THE VELOCITY CONTRIBUTIONS OF
C     OFF-DIAGONAL (AND ENDS OF THE ON-DIAGONAL) ELEMENTS BY A
C     SIMPSON-RULE.  ELEMENTS MAY BE FLAT OR CURVED, WITH PIECEWISE
C     CONSTANT-, LINEAR-, OR QUADRATIC-SOURCE DENSITY.
      COMMON /BLOCK2/HCURV(500), HARC(500)
      COMMON /BLOCK3/NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,LSSE,SO,
     1 NSMALL
      COMMON /BLOCK4/A11(500),A12(500),A13(500),A21(500),A22(500),
     1A23(500)
      COMMON /CL/   X1(500),   Y1(500),   X2(500),   Y2(500),   DELL(500),
     1              SINA(500),COSA(500)
      COMMON /TL/   A(500),    B(500),    AX(500),   AY(500),   AZ(500),
     2              CX(500),   CY(500),   CZ(500),   AXV(500),AYV(500),
     3              I,         J,         J1,        SJ,        DS,
     4              DX,        DY,        NI,        XJ,        YJ,
     5              XK,        EEK,       EKK,       K,         FF
      LOGICAL NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE
      LOGICAL FF,YSMALL
      REAL LSSE
      DIMENSION VOAXIX(129),VOAXIY(129),VOCRSX(129),VOCRSY(129),
     1 VOCRSZ(129), VOVORX(129),VOVORY(129),XRING(129),YRING(129)
      DATA PI/3.1415927/
C OFF-BODY ...
      X = XPKT
      Y = YPKT
C
      SINAL=SINA(J)
      COSAL=COSA(J)
      HC = HCURV(J)
      M = J
      IF (FIRSTE) M=J+1
      IF (LASTE) M=J-1
      MM1=M-1
      MP1=M+1
C     BEGIN BY GENERATING THE INDUCED VELOCITIES AT THE ENDS OF NI
C     INTERVALS (OF UNIFORM LENGTH)....
      S = SO
      DO 110 L = 1,NI
      SSQ=S**2
C     CALCULATE LOCAL SOURCE RADIUS, QA, AND AXIAL LOCATION, QB.
      QA = YJ + SINAL*S+HC*COSAL*SSQ
      QB = XJ + COSAL*S-HC*SINAL*SSQ
      XRING(L) = QB
      YRING(L) = QA
      IF (QA .GT. 0.0) GO TO 39
      VOAXIX(L) = 0.0
      VOAXIY(L) = 0.0
      VOCRSX(L) = 0.0
      VOCRSY(L) = 0.0
      VOCRSZ(L) = 0.0
      VOVORX(L) = 0.0
      VOVORY(L) = 0.0
      GO TO 110
C
C     CALCULATE THE COMPLETE ELLIPTIC INTEGRALS, K(K) AND E(K),
C             OF THE FIRST AND SECOND KINDS....
   39 XMB=X-QB
      XMBSQ= XMB**2
      YMASQ = (Y-QA)**2
      YPASQ=(Y+QA)**2
      T1 = YPASQ + XMBSQ
      XK= 4.*QA*Y/T1
      CALL ELIP(XK,EEK,EKK)
C     SUBROUTINE ELIP RETURNS THE VALUES AS EKK AND EEK.
      YSQ = Y**2
```

```
      ASQ = QA**2
      ROOT1 = SQRT(T1)
      T2 = YMASQ + XMBSQ
      T3 = XMBSQ + ASQ
      Q1 = (-4.*EEK)/(ROOT1*T2)
      Q2 = (2./ROOT1)*(EKK+(YSQ-ASQ-XMBSQ)*EEK/T2)
      YSMALL = (YSQ/T3) .LT. .0001
      IF (YSMALL) GO TO 40
      GO TO 50
   40 ROOT32 = (SQRT(T3))**3
      T4 = PI*QA/ROOT32
      T3SQ = T3**2
C
   50 IF (NOAXI) GO TO 70
      IF (YSMALL) GO TO 60
      VOAXIX(L) = Q1*QA*XMB
      VOAXIY(L) = -Q2*QA/Y
      GO TO 70
   60 VOAXIX(L) = (-2.*XMB*T4)*(1.0+0.75*YSQ*(3.0*ASQ-2.*XMBSQ)/T3SQ)
      VOAXIY(L) = -Y*T4*(2.*XMBSQ-ASQ)/T3
   70 IF (NOCROS) GO TO 90
      IF (YSMALL) GO TO 80
      VOCRSX(L) = 2.*(XMB   /(Y*ROOT1))*(EKK-(YSQ+T3)*EEK/T2)
      VOCRSY(L)=2.*(1./(YSQ*ROOT1))*(-(T3)*EKK+(T3**2+YSQ*(XMBSQ-ASQ))
     1 *EEK/T2)
      VOCRSZ(L)=2.*    (ROOT1/YSQ)*((YSQ+T3)*EKK/T1-EEK)
      GO TO 90
   80 VOCRSX(L) = -3.*QA*XMB*Y*T4/T3
      VOCRSY(L) = QA*T4*(1.0-0.125*YSQ*(4.*XMBSQ-ASQ)/T3SQ)
      VOCRSZ(L) = QA*T4*(1.0+0.375*YSQ*(ASQ-4.*XMBSQ)/T3SQ)
   90 IF (NOVORT) GO TO 110
      VOVORX(L) = (ASQ-YSQ)*Q1 - Q2
      IF (YSMALL) GO TO 100
      VOVORY(L) = Y*XMB*Q1 + XMB*Q2/Y
      GO TO 110
  100 VOVORY(L) = 0.0
  110 S = S + DS
C     ASSEMBLE THE ROW OF EACH DESIRED MATRIX...
      IF (NOAXI) GO TO 120
      CALL SIMSON(VOAXIX,NI,DS,VOAX)
      CALL SIMSON(VOAXIY,NI,DS,VOAY)
      AX(J) = AX(J) + VOAX
      AY(J) = AY(J) + VOAY
  120 IF (NOCROS ) GO TO 130
      CALL SIMSON(VOCRSX,NI,DS,VOCX)
      CALL SIMSON(VOCRSY,NI,DS,VOCY)
      CALL SIMSON(VOCRSZ,NI,DS,VOCZ)
      CX(J) = CX(J) + VOCX
      CY(J) = CY(J) + VOCY
      CZ(J) = CZ(J) + VOCZ
  130 IF (NOVORT) GO TO 140
      CALL SIMSON(VOVORX,NI,DS,VOVX)
      CALL SIMSON(VOVORY,NI,DS,VOVY)
      AXV(J) = AXV(J) + VOVX
      AYV(J) = AYV(J) + VOVY
  140 CONTINUE
      IF (NOV1) GO TO 250
      IF (NOAXI) GO TO 170
      S = S0
      DO 160 L = 1,NI
      VOAXIX(L) = VOAXIX(L)*S
      VOAXIY(L) = VOAXIY(L)*S
  160 S = S+DS
      CALL SIMSON (VOAXIX,NI,DS,V1AX)
      CALL SIMSON (VOAXIY,NI,DS,V1AY)
      AX(MM1) = AX(MM1) + A11(J)*V1AX
      AY(MM1) = AY(MM1) + A11(J)*V1AY
      AX(M) = AX(M) + A12(J)*V1AX
      AY(M) = AY(M) + A12(J)*V1AY
      AX(MP1) = AX(MP1) + A13(J)*V1AX
      AY(MP1) = AY(MP1) + A13(J)*V1AY
  170 IF (NOCROS) GO TO 200
      S = S0
      DO 180 L = 1,NI
      VOCRSX(L) = VOCRSX(L)*S
      VOCRSY(L) = VOCRSY(L)*S
      VOCRSZ(L) = VOCRSZ(L)*S
  180 S = S + DS
      CALL SIMSON(VOCRSX,NI,DS,V1CX)
      CALL SIMSON(VOCRSY,NI,DS,V1CY)
      CALL SIMSON(VOCRSZ,NI,DS,V1CZ)
      CX(MM1) = CX(MM1) + A11(J)*V1CX
      CY(MM1) = CY(MM1) + A11(J)*V1CY
      CZ(MM1) = CZ(MM1) + A11(J)*V1CZ
      CX(M) = CX(M) + A12(J)*V1CX
      CY(M) = CY(M) + A12(J)*V1CY
      CZ(M) = CZ(M) + A12(J)*V1CZ
      CX(MP1) = CX(MP1) + A13(J)*V1CX
      CY(MP1) = CY(MP1) + A13(J)*V1CY
      CZ(MP1) = CZ(MP1) + A13(J)*V1CZ
  200 IF (NOV2) GO TO 250
      IF (NOAXI) GO TO 220
      S = S0
      DO 210 L = 1,NI
      VOAXIX(L) = VOAXIX(L)*S
      VOAXIY(L) = VOAXIY(L)*S00
  210 S = S + DS
      CALL SIMSON (VOAXIX,NI,DS,V2AX)
      CALL SIMSON (VOAXIY,NI,DS,V2AY)
```

```
      AX(MM1) = AX(MM1) + A21(J)*V2AX
      AY(MM1) = AY(MM1) + A21(J)*V2AY
      AX(M) = AX(M) + A22(J)*V2AX
      AY(M) = AY(M) + A22(J)*V2AY
      AX(MP1) = AX(MP1) + A23(J)*V2AX
      AY(MP1) = AY(MP1)+ A23(J)*V2AY
  220 IF (NOCROS) GO TO 250
      S = SO
      DO 230 L = 1,NI
      VOCRSX(L) = VOCRSX(L)*S
      VOCRSY(L) = VOCRSY(L)*S
      VOCRSZ(L) = VOCRSZ(L)*S
  230 S = S + DS
      CALL SIMSON(VOCRSX,NI,DS,V2CX)
      CALL SIMSON(VOCRSY,NI,DS,V2CY)
      CALL SIMSON(VOCRSZ,NI,DS,V2CZ)
      CX(MM1) = CX(MM1) + A21(J)*V2CX
      CY(MM1) = CY(MM1) + A21(J)*V2CY
      CZ(MM1) = CZ(MM1) + A21(J)*V2CZ
      CX(M) = CX(M) + A22(J)*V2CX
      CY(M) = CY(M) + A22(J)*V2CY
      CZ(M) = CZ(M) + A22(J)*V2CZ
      CX(MP1) = CX(MP1) + A23(J)*V2CX
      CY(MP1) = CY(MP1) + A23(J)*V2CY
      CZ(MP1) = CZ(MP1) + A23(J)*V2CZ
C
250   RETURN
      END
C***$***$$**BEGINNING OF ELEMENT QCC/                *******$**C
      SUBROUTINE QC(OMEG,QM,Q)
C
C  THIS SUBROUTINE CALCULATES THE LEGENDRE FUNCTIONS OF THE SECOND KIND
C  AND HALF ORDER. THE ARGUMENTS ARE'
C        OMEG    ARGUMENT FOR WHICH LEGENDRE FUNCTIONS WILL BE FOUND
C        QM      VALUE OF LEGENDRE FUNCTION OF MINUS ONE HALF ORDER
C        Q       VALUE OF LEGENDRE FUNCTION OF PLUS ONE HALF ORDER
      DOUBLE PRECISION OMEGD,ARG,A,F,E,QMD,QD
      OMEGD=OMEG
      ARG=2.0/(OMEGD+1.0)
      A=1.0-ARG
      CALL ELLC (A,F,E,1)
      CALL ELLC (A,F,E,2)
      QMD=F*ARG**0.5
      QD=-E*(2.0*(OMEGD+1.0))**0.5+OMEGD*QMD
      QM=QMD
      Q=QD
      RETURN
      END
C***$***$$**BEGINNING OF ELEMENT SIMSO/              **$***$$**C
      SUBROUTINE SIMSON(Y,N,DX,AREA)
C THIS ROUTINE INTEGRATES Y OVER (N-1) INTERVALS OF EQUAL LENGTH, DX,
C YIELDING THE ENCLOSED AREA. (N MUST BE AN ODD INTEGER.)
C
      DIMENSION Y(1)
C
      SUM = Y(1) + 4.0*Y(N-1) + Y(N)
      IF (N .EQ. 3) GO TO 20
      NM2 = N - 2
      DO 10 I=2,NM2,2
   10 SUM = SUM + 4.0*Y(I) + 2.0*Y(I+1)
C
   20 AREA = (ABS(DX)/3.0) * SUM
      RETURN
      END
      SUBROUTINE LINEAR(X,Y,IMAX,IMIN)
      DIMENSION X(1),Y(1)
      IM=IMAX-1
      NTOT=IMAX-IMIN
      WRITE(6,199) NTOT
  199 FORMAT(I5)
      DO 10 I=IMIN,IM
      XMN=0.5*(X(I+1)-X(I))
      YMN=0.5*(Y(I+1)-Y(I))
      AX=ABS(XMN)
      IF(AX.LE.0.0000001) GO TO 10
      YP=-YMN/XMN
      XMN=X(I)+XMN
      YMN=Y(I)+YMN
      WRITE(6,93) I,YMN,XMN,YP
   93 FORMAT(2X,I5,3(1PE12.4))
   10 CONTINUE
      RETURN
      END
      SUBROUTINE EFFICY(NCOEF,NPTS,NS)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XIN,YYP,YYD,YOUT,YPOUT,XMAX,XMIN,XMA,XMI,DPD,PLWC,EPT,
     1         XA(1000),YA(1000)
      REAL*4 XBTA(100),YBTA(100)
      COMMON/COEF/A(20,21)
      COMMON/COEF1/B(20,30)
      COMMON/COEF2/XMA(10),XMI(10)
      COMMON/DPD1/NSI,PLWC(10),IRS(10),DPD(10),EPT
      REWIND 3
      N1=1
      N2=0
      NN=NCOEF+1
      DO 1 IJK=1,NSI
      NN1=NCOEF+IJK
      IF(IJK .EQ. NSI/2) NDA=IRS(IJK)
```

```
      READ(3) NDATA
      IF(NS.EQ.1 .OR. NSI .GT. 1) NPTS=NDATA
      N2=NDATA+N2
      DO 543 I=N1,N2
      READ(3) XA(I),YA(I)
      IF(ABS(XA(I)).LE.0.0000001) XA(I)=0.0
  543 CONTINUE
      IF(NSI.GT.1) WRITE(6,72)IJK
      IF(NSI .GT. 1) WRITE(6,71)DPD(IJK),PLWC(IJK)
      WRITE(6,73) NDATA
      WRITE(6,90)
      DO 97 I=N1,N2
   97 WRITE(6,93) I,YA(I),XA(I)
      IF(NSI .EQ. 1) GO TO 2
      XMIN=XA(N1)
      XMI(IJK)=XA(N1)
      XMAX=XA(N2)
      XMA(IJK)=XA(N2)
      IF(XA(N1).LE.XMIN) XMIN=XA(N1)
      IF(XA(N2).GE.XMAX) XMAX=XA(N2)
      XIN=XA(N1)
      CALL TERP(XIN,YOUT,YPOUT,XA,YA,N1,N2,NCOEF,NPTS)
      DO 3 II=1,NCOEF
      B(II,NN1)=A(II,NN)
    3 CONTINUE
      N1=NDATA+N1
    1 CONTINUE
      WRITE(6,198) XMIN,XMAX,EPT
      CALL EF(XMAX,XMIN,NDA,NSI,NCOEF,PLWC)
      RETURN
    2 WRITE(6,99)
      WRITE(6,91)
      NTOT=N2-N1+1
      CALL LINEAR (XA,YA,N2,N1)
      WRITE(6,70) NCOEF,NPTS
      IAC=0
      NTOA=NTOT
      NN=NCOEF+1
      WRITE(6,91)
      DO 20 I=N1,N2
      IAC=IAC+1
      XIN=XA(I)
      IF(NS.EQ.1.AND.I.GT.1) GO TO 21
      CALL TERP(XIN,YOUT,YPOUT,XA,YA,N1,N2,NCOEF,NPTS)
   21 IF(NS.EQ.1) GO TO 22
      YYP=YOUT
      YYD=-YPOUT
      GO TO 23
   22 YYD=0.0
      YYP=A(1,NN)
      DO 9 J=2,NCOEF
      IF(XA(I) .EQ. 0.0) XA(I)=0.0000001
      YYP=YYP+A(J,NN)*XA(I)**(J-1)
      YYD=YYD+(J-1)*A(J,NN)*XA(I)**(J-2)
    9 CONTINUE
      YYD=-1.*YYD
   23 IF(YYD.LT.0.0) GO TO 27
      WRITE(6,93) I,YYP,XA(I),YYD
      GO TO 28
   27 NTOA=NTOA-1
      IAC=IAC-1
      IF(I.EQ.N1.OR.I.EQ.N2) GO TO 29
      WRITE(6,93) I,YYP,XA(I),YYD
      GO TO 20
   29 YYD=0.0
      WRITE(6,93)I,YYP,XA(I),YYD
      NTOA=NTOA+1
      IAC=IAC+1
   28 YBTA(IAC)=YYD
      XBTA(IAC)=XA(I)
   20 CONTINUE
      WRITE(6,92)
      WRITE(6,199)NTOA
      WRITE(9)NTOA
      DO 24 I=1,NTOA
      WRITE(6,95)I,XBTA(I),YBTA(I)
      WRITE(9) XBTA(I),YBTA(I)
   24 CONTINUE
C---- FORMATS
   70 FORMAT(/,2X,'POLYNOMINAL LEAST SQUARE FIT',//,2X,
     1        'NO. OF COEFFICIENCTS IS ',I3,/,
     2       2X,'NO. OF POINTS FOR DATA FITTING IS',I3,/)
   71 FORMAT(/,2X,'DIAMETER',4X,'PERCENTAGE',
     1        /,2X,'DISTRIBUT.',2X,'OF W. CONT.',
     2        //,2X,2(1PE10.3,3X),/)
   72 FORMAT(///,1X,'***FOLLOWING OUTPUT IS FOR DROPLET SIZE
     1 DISTRIBUTION -- ',I3,/,
     2 3X,'(THE DATA ARE USED TO DO CURVE FITTING)',/)
   73 FORMAT(/,2X,'NO. OF DATA POINTS STORED IN TAPE03 = ',I6,/)
   90 FORMAT(/,5X,'I',6X,' ZO ',7X,' SD',/)
   91 FORMAT(/,5X,'I',6X,' ZO ',7X,' SD',8X,' BETA',/)
   92 FORMAT(/,5X,'I',6X,'SD',8X,'BETA',/)
   93 FORMAT(2X,I5,3(1PE12.4))
   95 FORMAT(2X,I5,2(1PE12.4))
   99 FORMAT(/,2X,'LINEAR APPROACH',/)
  198 FORMAT(///,1X,'FOLLOWING OUTPUT IS FOR THE OVERALL
     1 MULTI-SIZE CASE',//,4X,'UPPER SURFACE LIMIT',9X,
     2'LOWER SURFACE LIMIT',/,12X,'SU',26X,'SL',/,6X,
     32(1PE12.4,17X),//,2X,'TOTAL COLLECTION EFFICIENCY',
```

```
     43X,1PE12.5,/)
  199 FORMAT(I5)
      RETURN
      END
      SUBROUTINE EF(XMAX,XMIN,NDA,NSI,NCOEF,PLWC)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XA(100),PLWC(1),XBTA(100),YBTA(100)
      REAL*4 XMAX,XMIN,PLWC,YYD,DES,XMA,XMI,XA
      COMMON/COEF1/B(20,30)
      COMMON/COEF2/XMA(10),XMI(10)
      DES=(XMAX-XMIN)/FLOAT(NDA-1)
      IAC=0
      NTOA=NDA
      WRITE(6,70)NCOEF
      WRITE(6,92)
      DO 1 IJ=1,NDA
      IAC=IAC+1
      YYD=0.0
      XA(IJ)=XMAX-DES*(NDA-IJ)
      DO 2 I=1,NSI
      IF(XA(IJ) .GT. XMA(I).OR. XA(IJ).LT. XMI(I)) GO TO 2
      NN=NCOEF+I
      DO 2 J=2,NCOEF
      IF(XA(IJ) .EQ. 0.0) XA(IJ)=0.0000001
      YYD=YYD+(J-1)*B(J,NN)*XA(IJ)**(J-2)*PLWC(I)
    2 CONTINUE
      YYD=-1.*YYD
      IF(YYD .LT. 0.0) GO TO 27
      WRITE(6,93) IJ,XA(IJ),YYD
      GO TO 28
   27 NTOA=NTOA-1
      IAC=IAC-1
      IF(IJ.EQ.1.OR.IJ.EQ.NDA) GO TO 29
      WRITE(6,93)IJ,XA(IJ),YYD
      GO TO 1
   29 YYD=0.0
      WRITE(6,93) IJ,XA(IJ),YYD
      NTOA=NTOA+1
      IAC=IAC+1
   28 YBTA(IAC)=YYD
      XBTA(IAC)=XA(IJ)
    1 CONTINUE
      WRITE(6,92)
      WRITE(6,199)NTOA
      WRITE(9)NTOA
      DO 24 I=1,NTOA
      WRITE(6,95)I,XBTA(I),YBTA(I)
      WRITE(9) XBTA(I),YBTA(I)
   24 CONTINUE
C---- FORMATS
   70 FORMAT(/,2X,'POLYNOMIAL LEAST SQUARE FIT',//,2X,
     1        'NO. OF COEFFICIENCTS IS ',I3,/,
     2        2X,'WITH WHOLE DATA SETS',/)
   92 FORMAT(/,5X,'I',6X,'SD',8X,'BETA',/)
   93 FORMAT(2X,I5,3(1PE12.4))
   95 FORMAT(2X,I5,2(1PE12.4))
  199 FORMAT(I5)
      RETURN
      END
      SUBROUTINE TERP(XIN,YOUT,YPOUT,XA,YA,N1,N2,NCOEF,NPTS)
C---- DOES SIMPLE NCOEF POLYNOMIAL LEAST SQUARES FIT TO NPTS OF XA, YA
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 XIN,YOUT,YPOUT,XA(1),YA(1)
      DIMENSION F(1000,20),Y(1000)
      COMMON/COEF/A(20,21)
      NCM1 = NCOEF - 1
      NCOL = NCOEF + 1
C---- SEARCH XA FOR XIN
      DO 1 IF=N1,N2
      I=IF
      IF(XIN.LE.XA(IF))GO TO 5
    1 CONTINUE
    5 CONTINUE
      IMIN=I-NPTS/2
      IF(IMIN .LT. N1) IMIN=N1
      IMAX=IMIN+NPTS-1
      IF(IMAX.GT.N2) IMAX=N2
      IMIN=IMAX-NPTS+1
      III = 0
      DO 2 II = IMIN,IMAX
      III = III + 1
      F(III,1) = 1.0
      DO 3 JJ = 2,NCOEF
    3 F(III,JJ) = XA(II)**(JJ-1)
    2 Y(III) = YA(II)
      A(1,1) = NPTS
      DO 10 J = 1,NCOEF
      DO 20 I = J,NCOEF
      IF(I.EQ.1.AND.J.EQ.1)GO TO 20
      SUM = 0.0
      DO 11 ISUM = 1,NPTS
   11 SUM - SUM + F(ISUM,I)*F(ISUM,J)
      A(J,I) = SUM
      A(I,J) = SUM
   20 CONTINUE
      SUM = 0.0
      DO 50 I = 1,NPTS
   50 SUM = SUM + Y(I)*F(I,J)
      A(J,NCOL) = SUM
```

```
   10 CONTINUE
      CALL CHOLES(A,NCOEF,NCOL,1)
      YOUT=A(1,NCOL)
      YPOUT=0.0
      DO 4 I=2,NCOEF
      IF(XIN .EQ.0.0) XIN=0.0000001
      YOUT=YOUT+A(I,NCOL)*XIN**(I-1)
    4 YPOUT=YPOUT+(I-1)*A(I,NCOL)*XIN**(I-2)
      RETURN
      END
      SUBROUTINE CHOLES(A,N,M,MATSYM)
C---- CDPD MATRIX SOLUTION ROUTINE FROM DON TODD ON 4-24-80
      REAL*8 A(20,21),SUM,TEMP
      IF(A(1,1).NE.0.0) GO TO 47
      DO 37 J=2,N
      IF(A(J,1).EQ.0.0) GO TO 37
      IFLIP=J
      GO TO 27
   37 CONTINUE
      GO TO 54321
   27 DO 57 K=1,M
      TEMP=A(IFLIP,K)
      A(IFLIP,K)=A(1,K)
      A(1,K)=TEMP
   57 CONTINUE
   47 DO 2 J=2,M
      A(1,J)=A(1,J)/A(1,1)
    2 CONTINUE
      DO 6 I=2,N
      DO 7 J=2,M
      IF(MATSYM.EQ.0)GO TO 49
      IF(I-J)69,68,67
   49 IF(J.GT.I)GO TO 69
   68 K=J-1
      SUM=0.0
      DO 3 IR=1,K
      SUM=SUM+A(I,IR)*A(IR,J)
    3 CONTINUE
      A(I,J)=A(I,J)-SUM
      GO TO 7
   69 K=I-1
      SUM=0.0
      DO 4 IR=1,K
      SUM=SUM+A(I,IR)*A(IR,J)
    4 CONTINUE
      IF(A(I,I).EQ.0.0) GO TO 54321
      A(I,J)=(A(I,J)-SUM)/A(I,I)
      GO TO 7
   67 A(I,J)=A(J,I)*A(J,J)
    7 CONTINUE
    6 CONTINUE
      DO 52 K=2,N
      I=N+1-K
      SUM=0.0
      LL=I+1
      DO 51 IR=LL,N
      SUM=SUM+A(I,IR)*A(IR,M)
   51 CONTINUE
      A(I,M)=A(I,M)-SUM
   52 CONTINUE
      GO TO 12345
54321 N=-1
12345 RETURN
      END
      SUBROUTINE APLOT (U,YL,N,K,LDP,FC,CZ)
      DIMENSION U(1),YL(1),N(1)
      DIMENSION XTITL(20),YTITL(20)
      DIMENSION DSNAME(9)
      DATA IDF/1/,DSNAME/'SAVE','PLOT',7*'    '/
C---- K:    NO. OF DATA SETS
C---- N(K): NO. OF DATA POINTS IN KTH DATA SET
C---- U:    X-COORDINATE
C---- YL:   Y-COORDINATE
      IF(LDP .EQ. 1) GO TO 19
      READ(2,100) (XTITL(I),I=1,20)
      READ(2,100) (YTITL(I),I=1,20)
      READ(2,101)YHEIT,YMAX,YMIN
      READ(2,101)XLENG,XMAX,XMIN
  100 FORMAT(20A4)
  101 FORMAT(5(5X,F10.0))
      CZ=(YMAX-YMIN)/YHEIT
      FC=(XMAX-XMIN)/XLENG
      CALL CCBEGN(10.0,1.0)
      CALL BEGID(IDP)
      CALL PLOT(1.,2.5,-3)
      CALL AXIS(0.,0.,XTITL,-80,XLENG,0.,XMIN,FC,0.0)
      CALL PLOT(0.,0.,3)
      CALL AXIS(0.,0.,YTITL,80,YHEIT,90.,YMIN,CZ,0.0)
      CALL PLOT(0.,YHEIT,3)
      CALL PLOT(XLENG,YHEIT,2)
      CALL PLOT(XLENG,0.,2)
      CALL PLOT(0.0,0.0,3)
   19 CONTINUE
      NK=-1
      JN=0
      DO 14 I=1,K
      J1=JN+1
      JN=JN+N(I)
      NK=NK+2
      DO 13 J=J1,JN
      Z=(YL(J)-YMIN)/CZ
      X=(U(J)-XMIN)/FC
      IF(J .EQ. J1) CALL PLOT(X,Z,3)
      IF(Z.GT.YHEIT) GO TO 13
      IF(LDP.EQ.1) GO TO 17
      IF(I.NE.K) GO TO 67
   17 CALL PLOT(X,Z,2)
      GO TO 13
   67 CONTINUE
      CALL SYMBOL(X,Z,0.14,NK,0.,-1)
   13 CONTINUE
      CALL PLOT(0.0,0.0,3)
   14 CONTINUE
      CALL ENDID(IDP,-1,DSNAME)
      CALL CCBEGN(-1.0,1.0)
      RETURN
      END
      SUBROUTINE TRAJCT
      RETURN
      END
```

The following computer programs and data files are recorded on this magtape.

The order in magtape	Description	Name suggested in IBM system
1	Particle trajectory computer program for axisymmetric inlet case.	SOURCE.TRAJ3
2	Axisymmetric potential flow computer program (EOD)	SOURCE.EOD
3	Axisymmetric COMBYN computer program	SOURCE.COMBYN
4	Input data file (Tape 05) for geometry generation program SCIRCL (Axisymmetric inlet)	DAT.AI005S
5	Input data file Tape 17 (renamed on Tape 05) for EOD	DAT.AI005Y
6	Input data file (Tape 05) for COMBYN. (Axisymmetric inlet with $\alpha = 0°$, M=0.4)	DAT.AI005C
7	Input data file (Tape 05) for COMBYN (Axisymmetric inlet with $\alpha = 30°$, $M = 0.4$)	DAT.AI305C
8	Input data file (Tape 02) for Trajectory program, SOURCE.TRAJ3. (Axisymmetric inlet with $\alpha = 0°$, $M = 0.4$)	DAT.AI002T

9	Input data file (Tape 02) for Trajectory program, SOURCE.TRAJ3. (Axisymmetric inlet with α = 30°, M=0.4)	DAT.AI302T
10	Two-Dimensional potential flow computer program, Y24. (Original program).	SOURCE.Y240
11	Two-dimensional COMBIN computer program. (Original program)	SOURCE.COMBINO
12	Two-Dimensional COMBIN computer program. (Original program). This program is almost the same as 11. We didn't use this one.	Up to you.
13	Axisymmetric potential flow computer program, EOD. (Original program).	SOURCE.EODO
14	Axisymmetric COMBYN computer program. (Original program).	SOURCE.COMBYNO

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
February 1992

3. REPORT TYPE AND DATES COVERED
Final Contractor Report

4. TITLE AND SUBTITLE
Particle Trajectory Computer Program for Icing Analysis of Axisymmetric Bodies

5. FUNDING NUMBERS
WU-505-68
C-NAS3-22448

6. AUTHOR(S)
Walter Frost, Ho-Pen Chang, and Kenneth R. Kimble

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
FWG Associates, Inc.
Rt. 2, Box 271-A Lakewood Drive
Tullahoma, Tennessee 37388

8. PERFORMING ORGANIZATION REPORT NUMBER
None

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-189134

11. SUPPLEMENTARY NOTES
Project Manager, Robert J. Shaw, Propulsion Systems Division, NASA Lewis Research Center.

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

12b. DISTRIBUTION CODE

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

General aviation aircraft and helicopters exposed to an icing environment can accumulate ice resulting in a sharp increase in drag and reduction of maximum lift causing hazardous flight conditions. NASA Lewis Research Center (LeRC) is conducting a program to examine, with the aid of high-speed computer facilities, how the trajectories of particles contribute to the ice accumulation on airfoils and engine inlets. This study, as part of the NASA/LeRC research program, develops a computer program for the calculation of icing particle trajectories and impingement limits relative to axisymmetric bodies in the leeward-windward symmetry plane. The methodology employed in the current particle trajectory calculation is to integrate the governing equations of particle motion in a flow field computed by the Douglas axisymmetric potential flow program [1]. The three-degrees-of-freedom (horizontal, vertical, and pitch) motion of the particle is considered. The particle is assumed to be acted upon by aerodynamic lift and drag forces, gravitational forces, and, for nonspherical particles, aerodynamic moments. The particle momentum equation is integrated to determine the particle trajectory. Derivation of the governing equations and the method of their solution are described in Section 2.0. General features, as well as input/output instructions for the particle trajectory computer program, are described in Section 3.0. The details of the computer program are described in Section 4.0. Examples of the calculation of particle trajectories demonstrating application of the trajectory program to given axisymmetric inlet test cases are presented in Section 5.0. For the examples presented, the particles are treated as spherical water droplets. In Section 6.0, limitations of the program relative to excessive computer time and recommendations in this regard are discussed.

14. SUBJECT TERMS
Aircraft icing; Particle trajectories; Inlet flow fields; Potential flow

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT

FWG

FWG ASSOCIATES, INC.

CORPORATE HEADQUARTERS • 217 LAKEWOOD DRIVE TULLAHOMA, TN 37388 • TELEPHONE (615) 455-1982 FAX (615) 455-1805

HUNTSVILLE OPERATION • P.O. BOX 12056 HUNTSVILLE, AL 35815—2056 • TELEPHONE (205) 880-2553 FAX (205) 880-2554