PARTICLE TRAJECTORY COMPUTER PROGRAM FOR Icing ANALYSIS OF AXISYMMETRIC BODIES Final Report (FWG Associates) 121 p
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

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 METHODOLOGY</td>
<td>2</td>
</tr>
<tr>
<td>2.1 Potential Flow About an Axisymmetric Body</td>
<td>2</td>
</tr>
<tr>
<td>2.2 Particle Equations of Motion</td>
<td>3</td>
</tr>
<tr>
<td>2.3 Particle Trajectory Solution Algorithm</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Computation of Collection Efficiency</td>
<td>9</td>
</tr>
<tr>
<td>2.5 General Computational Procedure</td>
<td>11</td>
</tr>
<tr>
<td>3.0 GENERAL DESCRIPTION OF COMPUTER PROGRAMS</td>
<td>14</td>
</tr>
<tr>
<td>3.1 General Features of the Program</td>
<td>14</td>
</tr>
<tr>
<td>3.1.1 Types of Flow</td>
<td>14</td>
</tr>
<tr>
<td>3.1.2 Surface Distance Computation</td>
<td>14</td>
</tr>
<tr>
<td>3.1.3 Initial Longitudinal Coordinate, ( x_0 ), of the Particle Trajectory</td>
<td>14</td>
</tr>
<tr>
<td>3.1.4 Impingement Position of the Particle on the Body</td>
<td>15</td>
</tr>
<tr>
<td>3.1.5 Computation of Surface Impingement Location</td>
<td>17</td>
</tr>
<tr>
<td>3.1.5.1 General Case</td>
<td>17</td>
</tr>
<tr>
<td>3.1.5.2 Special Case</td>
<td>17</td>
</tr>
<tr>
<td>3.1.6 Upstream ( y )-Coordinate Limits ( y_{ou}, y_{ol} )</td>
<td>19</td>
</tr>
<tr>
<td>3.1.7 Calculation of the Local Collection Efficiency</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Program Input Instructions</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1 Geometry Generation Program Input Instructions</td>
<td>24</td>
</tr>
<tr>
<td>3.2.2 Axisymmetric Potential Flow Program Input Instructions</td>
<td>34</td>
</tr>
<tr>
<td>SECTION</td>
<td>PAGE</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.2.3 COMBYN Program Input Instructions</td>
<td>39</td>
</tr>
<tr>
<td>3.2.4 Particle Trajectory Program Input Instructions</td>
<td>45</td>
</tr>
<tr>
<td>3.3 Description of Program Output</td>
<td>51</td>
</tr>
<tr>
<td>3.3.1 Geometry Generation Program Output Description</td>
<td>51</td>
</tr>
<tr>
<td>3.3.2 Axisymmetric Potential Flow Program Output Description</td>
<td>51</td>
</tr>
<tr>
<td>3.3.3 COMBYN Program Output</td>
<td>52</td>
</tr>
<tr>
<td>3.3.4 Trajectory Program Output</td>
<td>52</td>
</tr>
<tr>
<td>4.0 PARTICLE TRAJECTORY COMPUTER PROGRAM CAPABILITIES AND FUNCTION DESCRIPTIONS</td>
<td>54</td>
</tr>
<tr>
<td>4.1 Main Program and Subroutines</td>
<td>54</td>
</tr>
<tr>
<td>4.1.1 Main Program</td>
<td>54</td>
</tr>
<tr>
<td>4.1.2 Subroutine ADAMS</td>
<td>57</td>
</tr>
<tr>
<td>4.1.3 Subroutine RANGE</td>
<td>59</td>
</tr>
<tr>
<td>4.1.4 Subroutine F</td>
<td>61</td>
</tr>
<tr>
<td>4.1.5 Subroutine COEFF</td>
<td>61</td>
</tr>
<tr>
<td>4.1.6 Subroutine MODE</td>
<td>61</td>
</tr>
<tr>
<td>4.1.7 Subroutine READIN</td>
<td>63</td>
</tr>
<tr>
<td>4.1.8 Subroutine VELCTY</td>
<td>64</td>
</tr>
<tr>
<td>4.1.9 Subroutine EFFICY</td>
<td>66</td>
</tr>
<tr>
<td>4.1.10 Subroutine EF</td>
<td>68</td>
</tr>
<tr>
<td>4.1.11 Subroutine LINEAR</td>
<td>68</td>
</tr>
<tr>
<td>4.1.12 Subroutine TERP</td>
<td>68</td>
</tr>
<tr>
<td>4.1.13 Subroutine CHOLESS</td>
<td>68</td>
</tr>
<tr>
<td>4.1.14 Subroutines APLLOT and TRAJCT</td>
<td>69</td>
</tr>
<tr>
<td>4.2 Error Messages</td>
<td>69</td>
</tr>
</tbody>
</table>

iv
<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0 TEST CASES FOR PARTICLE TRAJECTORY CALCULATIONS</td>
<td>72</td>
</tr>
<tr>
<td>5.1 Axisymmetric Inlet at 0° Angle of Attack with Mach Number = 0.4</td>
<td>72</td>
</tr>
<tr>
<td>5.2 Axisymmetric Inlet at 30° Angle of Attack with Mach Number = 0.4</td>
<td>77</td>
</tr>
<tr>
<td>6.0 DISCUSSION OF RESULTS</td>
<td>79</td>
</tr>
<tr>
<td>6.1 Modification to EOD Computer Program</td>
<td>79</td>
</tr>
<tr>
<td>6.2 Interpolation Method</td>
<td>80</td>
</tr>
<tr>
<td>6.3 Recommendations</td>
<td>81</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>83</td>
</tr>
<tr>
<td>APPENDIX: PARTICLE TRAJECTORY COMPUTER PROGRAM LISTING</td>
<td>84</td>
</tr>
<tr>
<td>FIGURE</td>
<td>TITLE</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.1</td>
<td>Diagram of the Velocity Vectors and Forces Acting on a Point Mass Particle</td>
</tr>
<tr>
<td>2.2</td>
<td>Friction (or drag coefficient) for Spheres Moving Relative to a Fluid with a Velocity, $V_\infty$ [5]</td>
</tr>
<tr>
<td>2.3</td>
<td>Illustration of Impingement Terminology and Water Droplet Trajectory in an Airfoil Flow Field [7]</td>
</tr>
<tr>
<td>2.4</td>
<td>Illustrates Total Data Set and Segment Curve Fitting Techniques</td>
</tr>
<tr>
<td>3.1</td>
<td>Coordinate Transform</td>
</tr>
<tr>
<td>3.2</td>
<td>Illustration of the Method of Determining the Position of Particle Impingement on the Body Surface</td>
</tr>
<tr>
<td>3.3</td>
<td>General Flowchart</td>
</tr>
<tr>
<td>4.1</td>
<td>Flowchart of Main Program</td>
</tr>
<tr>
<td>4.2</td>
<td>Flowchart of Subroutine ADAMS</td>
</tr>
<tr>
<td>4.3</td>
<td>Flowchart of Subroutine RANGE</td>
</tr>
<tr>
<td>4.4</td>
<td>Flowchart of Subroutine MODE</td>
</tr>
<tr>
<td>4.5</td>
<td>Flowchart of Subroutine READIN</td>
</tr>
<tr>
<td>4.6</td>
<td>Flowchart of Subroutine VELCTY</td>
</tr>
<tr>
<td>4.7</td>
<td>Flowchart of Subroutine EFFICY</td>
</tr>
<tr>
<td>5.1</td>
<td>Geometry of Axisymmetric Inlet Normalized by Inlet Diameter (11.68 units)</td>
</tr>
<tr>
<td>5.2</td>
<td>Particle Trajectories Released from $x_0 = -0.5$, $\alpha = 0^\circ$ (lower body)</td>
</tr>
<tr>
<td>5.3</td>
<td>Particle Trajectories Released from $x_0 = -0.5$, $\alpha = 30^\circ$</td>
</tr>
</tbody>
</table>

vi
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Polynomial Coefficients Relating Best Number (Be) to Reynolds Number for Spherical Particles [5]</td>
<td>8</td>
</tr>
<tr>
<td>4.1</td>
<td>Error Messages and Corrective Actions</td>
<td>70</td>
</tr>
<tr>
<td>5.1</td>
<td>Test Cases Computed with Particle Trajectory Program</td>
<td>73</td>
</tr>
<tr>
<td>5.2</td>
<td>Input Data File for Geometry Generation Program SCIRCL</td>
<td>74</td>
</tr>
<tr>
<td>5.3</td>
<td>Input Data File Tape 05 for the COMBYN Program</td>
<td>75</td>
</tr>
<tr>
<td>5.4</td>
<td>Input Data File Tape 02 for the Particle Trajectory Program (0° angle of attack)</td>
<td>76</td>
</tr>
<tr>
<td>5.5</td>
<td>Input Data File Tape 05 for the Particle Trajectory Program (30° angle of attack)</td>
<td>77</td>
</tr>
</tbody>
</table>
NOMENCLATURE

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Reference area of particle</td>
</tr>
<tr>
<td>Be</td>
<td>Best number (Be = CdRe^2)</td>
</tr>
<tr>
<td>Cd</td>
<td>Drag coefficient</td>
</tr>
<tr>
<td>Cd0</td>
<td>Steady-state drag coefficient</td>
</tr>
<tr>
<td>Cdα</td>
<td>Drag coefficient due to angle of attack</td>
</tr>
<tr>
<td>C</td>
<td>Lift coefficient</td>
</tr>
<tr>
<td>Cα</td>
<td>Lift coefficient due to angle of attack</td>
</tr>
<tr>
<td>Cm</td>
<td>Moment coefficient</td>
</tr>
<tr>
<td>Cm0</td>
<td>Steady-state moment coefficient</td>
</tr>
<tr>
<td>Cmα</td>
<td>Moment coefficient due to angle of attack</td>
</tr>
<tr>
<td>D</td>
<td>Particle diameter</td>
</tr>
<tr>
<td>Ḋ</td>
<td>Aerodynamic drag force</td>
</tr>
<tr>
<td>E</td>
<td>Total collection efficiency</td>
</tr>
<tr>
<td>FRL</td>
<td>Flight reference line</td>
</tr>
<tr>
<td>g</td>
<td>Acceleration of gravity (9.8 m/s^2)</td>
</tr>
<tr>
<td>g_c</td>
<td>Conversion factor (1 kg-m/newton-sec^2)</td>
</tr>
<tr>
<td>h</td>
<td>Projected height of the body along the vertical coordinate line</td>
</tr>
<tr>
<td>I_{zz}</td>
<td>Moment of inertia of mass relative to the z axis</td>
</tr>
<tr>
<td>L̇</td>
<td>Aerodynamic lift force</td>
</tr>
<tr>
<td>LWC</td>
<td>Liquid water content</td>
</tr>
<tr>
<td>l_c</td>
<td>Reference length of body</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>$M$</td>
<td>Moment of aerodynamic forces acting on the particle</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass of water droplet (kg)</td>
</tr>
<tr>
<td>$\dot{m}$</td>
<td>Mass flow rate</td>
</tr>
<tr>
<td>$n_i$</td>
<td>Number of particles of size $i$ per unit volume</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number based on the diameter of the particle</td>
</tr>
<tr>
<td>$s$</td>
<td>Surface distance measured from the leading edge of each body; positive along the lower surface and negative along the upper surface</td>
</tr>
<tr>
<td>$S_L$</td>
<td>Lower surface impingement limit</td>
</tr>
<tr>
<td>$S_U$</td>
<td>Upper surface impingement limit</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$\vec{V}$</td>
<td>Velocity of particle</td>
</tr>
<tr>
<td>$\vec{V}_a$</td>
<td>Velocity of particle relative to flow field</td>
</tr>
<tr>
<td>$\vec{W}$</td>
<td>Velocity of flow field</td>
</tr>
<tr>
<td>$W_x$</td>
<td>$x$-component velocity of flow field</td>
</tr>
<tr>
<td>$W_y$</td>
<td>$y$-component velocity of flow field</td>
</tr>
<tr>
<td>$W_\infty$</td>
<td>Free-stream velocity</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$-coordinate of particle</td>
</tr>
<tr>
<td>$\dot{x}$</td>
<td>$x$-component velocity of particle</td>
</tr>
<tr>
<td>$\ddot{x}$</td>
<td>$x$-component acceleration of particle</td>
</tr>
<tr>
<td>$x_0$</td>
<td>Initial value of the horizontal coordinate of particle</td>
</tr>
<tr>
<td>$y$</td>
<td>$y$-coordinate of particle</td>
</tr>
<tr>
<td>$\dot{y}$</td>
<td>$y$-component velocity of particle</td>
</tr>
<tr>
<td>$\ddot{y}$</td>
<td>$y$-component acceleration of particle</td>
</tr>
<tr>
<td>$y_0$</td>
<td>Initial value of the vertical coordinate of particle</td>
</tr>
<tr>
<td>$y_{ou}$</td>
<td>Upper tangent trajectory of the particle corresponding to $S_U$</td>
</tr>
</tbody>
</table>
Symbol Definition

\( y_{02} \) Lower tangent trajectory of the particle corresponding to \( S_L \)

Greek Symbols

\( \alpha \) Angle of attack

\( \beta \) Local collection efficiency

\( \gamma \) Particle path angle \( \left( \gamma = \tan^{-1} \frac{\dot{y} - W_y}{\dot{x} - W_x} \right) \)

\( \delta \) The angle between \( \vec{V} \) and \( \vec{V}_a \)

\( \eta_i \) Percentage liquid water contained in particles of size \( D_i \)

\( \theta \) 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 arbitrary 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 snowflake, 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:

\[ m\ddot{x} = -D \cos \gamma - L \sin \gamma \]  \hspace{1cm} (2.1)
\[ m\ddot{y} = -D \sin \gamma + L \cos \gamma - mg \]  \hspace{1cm} (2.2)

where

\[ \gamma = \tan^{-1} \left( \frac{\dot{y} - W_y}{\dot{x} - W_x} \right) \]  \hspace{1cm} (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:

\[ D = C_d \frac{\rho_a V_a^2}{2g_c} A \]

\[ L = C_\alpha \frac{\rho_a V_a^2}{2g_c} A \]

where \( A \) is a characteristic area of the particle, \( \rho_a \) 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:

\[ V_a = \sqrt{(x - W_x)^2 + (y - W_y)^2} \] (2.4)

For arbitrarily shaped particles, expressions for the drag coefficient, \( C_d \), and lift coefficient, \( C_\alpha \), must be provided by the user. They are often approximated with:

\[ C_d = C_{d0} + C_{d\alpha} \alpha \] (2.5)

\[ C_\alpha = C_{\alpha0} + C_{\alpha\alpha} \alpha \] (2.6)

where \( \alpha \) 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., \( C_{d0} \), \( C_{\alpha0} \), \( C_{d\alpha} \), and \( C_{\alpha\alpha} \). The angle \( \alpha \) is computed from the expression:

\[ \alpha = \theta - \gamma \] (2.7)

where the angle \( \theta \), generally called pitch angle in aerodynamics, is governed by:

\[ \theta = \frac{M}{I_{zz}} \] (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:
\[ M = C_m \frac{\rho a V_a^2}{2g_c} A \approx C \]  

(2.9)

where \( C_m \) is approximately:

\[ C_m = C_{m0} + C_{m\alpha} \dot{a} \]

\( C_{m0} \) and \( C_{m\alpha} \) are constants to be provided by the user depending upon the shape of the particle, and \( \varepsilon_c \) 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:

\[ m \ddot{x} = -\ddot{D} \cos \gamma \]  

(2.10)

\[ m \ddot{y} = -\ddot{D} \sin \gamma - mg \]  

(2.11)

where

\[ \ddot{D} = C_{d0} \frac{\rho a V_a^2}{2g_c} A \]  

(2.12)

In the present study \( C_{d0} \) is the steady-state drag coefficient of a sphere as a function of Reynolds number based on particle diameter, \( Re = \frac{V_a D}{\nu} \), as shown in Figure 2.2. The diameter of the spherical particle, \( D \), and the kinematic viscosity of the air, \( \nu \), 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.

\[ \frac{dx}{dt} = \dot{x} \]  

(2.13)

\[ \frac{dy}{dt} = \dot{y} \]  

(2.14)

\[ \frac{d\dot{e}}{dt} = \ddot{e} \]  

(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, $V_\infty$ [5].
TABLE 2.1 Polynomial Coefficients Relating Best Number (Be) to Reynolds Number for Spherical Particles [5].

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

<table>
<thead>
<tr>
<th>Reynolds Number Range</th>
<th>j</th>
<th>(a_j)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.05 &lt; Re \leq 3)</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>24.167</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.254</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.23564</td>
</tr>
<tr>
<td>(3 &lt; Re \leq 330)</td>
<td>0</td>
<td>-28.339</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>38.969</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.73204</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>-0.00056084</td>
</tr>
<tr>
<td>(330 &lt; Re)</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>93.462</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.37576</td>
</tr>
</tbody>
</table>

where

\[
Re = \frac{|\hat{W} - \hat{V}|D}{\nu}
\]

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

D = particle diameter

\(\hat{W}\) = inertial velocity of the air

\((\hat{W} = W_x \hat{i} + W_y \hat{j})\)

\(\hat{V}\) = inertial velocity of the particle

\((\hat{V} = x \hat{i} + y \hat{j})\)
in solutions for $\dot{x}$, $\dot{y}$, $\dot{\theta}$, $x$, $y$, $\theta$ 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 $\varepsilon$:

$$\left| (\dot{x}_{n, \text{predictor}} - \dot{x}_{n, \text{corrector}}) \right| + \left| (\dot{y}_{n, \text{predictor}} - \dot{y}_{n, \text{corrector}}) \right| \leq \varepsilon$$

The value of $\varepsilon$ 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, $\beta$, is calculated as a function of the distance along the body surface by differentiating $y_0 = y_0(s)$ with respect to $s$ [7]:

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

(2.16)

The minus sign is introduced so that $\beta$ is positive, which is consistent with the definition given in Reference 4.
$S_U = \text{Upper surface impingement limit}$

$S_L = \text{Lower surface impingement limit}$

**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:

$$E = \frac{y_{ou} - y_{ol}}{h}$$

(2.17)

where $y_{ou}$ and $y_{ol}$ 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 $W_M$ and the local velocity $W$ is less than some small value $\varepsilon$. The value of $\varepsilon$ 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 $y_{ol}$, 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 $y_{ol}$ 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 $y_{ol}$ 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.
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, $\Delta s$, 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 longitudinal coordinate position $x_0$ at which the particle trajectory calculations begin (Figure 2.3) is automatically determined
by the computer program. The value of \( x_0 \) is selected by testing the maximum difference between the locally computed value of \( W \) and the free-stream velocity \( W_\infty \) at successively farther distance upstream. The value of \( x \) for which the inequality

\[
1 - \frac{W}{(W_\infty)} \leq 0.001
\]

is satisfied is designated as \( x_0 \). Equation 3.1 is tested over a specified range \( YYLO \leq y/x_c \leq YYUP \), where \( YYLO \) and \( YYUP \) are initial values input by the user. They represent the expected maximum and minimum range of the upstream \( y \)-coordinate. \( x_c \) 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:

\[
X_t = X
\]

\[
Y_t = \frac{Y - YREF}{YREF - YYLO}
\]

where (see Figure 3.1a)

\[
YREF = Y'_{A'BCC}, \text{ if } Y \geq 0.0
\]

\[
YREF = Y'_{A'AB'CC}, \text{ if } Y \leq 0.0
\]

where \( X \) and \( Y \) are the \( x \)- and \( y \)-coordinates normalized by the characteristic length, \( x_c \), 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 $\Delta t$, 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 $(X_O, Y_P)O$ and the coordinates at $t = t + \Delta t$ be $(X_N, Y_P N)$. The reference coordinates on the body surface $A'ABCC'$ or $A'AB'CC'$ are $X_O, Y_R O$ and $X_N, Y_N R$, respectively. The analytical function of the line $\overline{ON}$ is:

$$\frac{Y - Y_P O}{X - X_O} = \frac{Y_P N - Y_P O}{X_N - X_O}$$

(3.4)

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

$$\frac{Y - Y_R O}{X - X_O} = \frac{Y_R N - Y_R O}{X_N - X_O}$$

(3.5)

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

$$X_I = \frac{(X_N - X_O)(Y_P O - Y_R O)}{(Y_R N - Y_R O - Y_P N + Y_P O)} + X_O$$


(3.6)

The reference coordinates on the body surface $(X_I, Y_R)$ corresponding to the position $(X_I, Y_I)$ is then found. If $|Y_I - Y_R| < 10^{-5}$, the point $(X_I, Y_I)$ is regarded as the particle impingement position on the body surface. If $|Y_I - Y_R| > 10^{-5}$ the point $(X_I, Y_I)$ is redefined as $(X_N, Y_P N)$ and $(X_I, Y_R)$ as $(X_N, Y_N R)$, respectively. The procedure is repeated until the requirement $|Y_I - Y_R| < 10^{-5}$ 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:
Figure 3.2 Illustration of the method of determining the position of particle impingement on the body surface.
The coordinates \((X_1, Y_1)\) of the point of intersection of line \(ON\) and line \(AN^T\) are found as follows:

Let

\[
\begin{align*}
A &= (X_N - X_F)(Y_{PN} - Y_{PO}) - (Y_{RN} - Y_F)(X_N - X_O) \\
B &= Y_{PN} - Y_{PO} \\
C &= Y_{RN} - Y_F \\
D &= X_N - X_F \\
E &= X_N - X_O
\end{align*}
\]

then

\[
\begin{align*}
X_1 &= \frac{[B \cdot D \cdot X_O - E \cdot C \cdot X_F + (Y_F - Y_{PO}) \cdot E \cdot D]}{A} \\
Y_1 &= \frac{[D \cdot B \cdot Y_F - E \cdot C \cdot Y_{PO} + (X_O - X_F) \cdot B \cdot C]}{A}
\end{align*}
\]

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 \(y_{0u}, y_{0l}\)

The method by which the computer selects the upper and lower trajectory limits \(y_{0u}\) and \(y_{0l}\) 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 \(y_{0u}\) and \(y_{0l}\) (see Figure 2.3).

The search procedure consists of the computer program initially seeking the range within which \(y_{0u}\) and \(y_{0l}\) lie. This is achieved by computing the particle trajectory from the initial position \((X_0, Y_{MAX})\) where \(Y_{MAX}\) and \(Y_{MIN}\) are the user's initial guessed values of \(y_{0u}\) and \(y_{0l}\), respectively, and \(X_0 = x_0/\lambda_c\). If the particle passes under or hits the body, \(Y_{MAX}\) and \(Y_{MIN}\) are redefined as \(Y_{MAX}^{NEW} = Y_{MAX} + \Delta Y\) and
YMINNEW = YMAX where \( \Delta Y = YMAX - YMIN \). 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 \( y_{OL} \) 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., YMAXNEW = YMIN and YMINNEW = YMIN - \( \Delta Y \)).

Once this order of magnitude range of \( y_{OU} \) and \( y_{OL} \) 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 \( y_0 \) 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 \( y_{OL} \) 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 \( y_{OL} \) within the range YMAX and YMIN input by the user. If a poor guess was made and \( y_{OU} \) and \( y_{OL} \) 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 \( y_{OL} \) determined, the computer divides the range into NPL segments (i.e., \((y_{OU} - y_{OL})/NPL\)) 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 \( \{y_0,s\} \) 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, \( \beta \). The first method is linear approximation:

\[
\beta = -\frac{\Delta y_0}{\Delta s}
\]

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 \( \beta \) is computed by taking the derivative of the polynomial function. The number of coefficient 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 (NS1>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 $\text{LOPT} \neq 0$ and contains coordinates of the particle trajectories as well as values of the wind speed component as a function of time.

Tape 08: Data stored on Tape 08 is used for trajectory plots. The data stored are the $x/\lambda_{C}, y/\lambda_{C}$ coordinates of the particle $X_P, Y_P$ for each time step and the surface impingement point SW. SW is initially set at 88.8888. When $\text{SW} \neq 88.8888$, the particle trajectory is terminated.

Tape 09: Tape 09 contains the local collection efficiency, $\beta$, as a function of surface position, $s/\lambda_{C}$. The program automatically plots $\beta$ versus $s/\lambda_{C}$ 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:

<table>
<thead>
<tr>
<th>Card</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>(only if flag J &gt; 0)</td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Number of '9' cards = ANBDYS</td>
</tr>
<tr>
<td>10</td>
<td>Number of '10-11-12' groups for each '9' card = ANSEG</td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

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

For information to be passed on to EOD:

<table>
<thead>
<tr>
<th>3</th>
<th>1</th>
<th>IGEOMF</th>
<th>Main/11</th>
</tr>
</thead>
</table>
|   |   | =1, Use flat elements  
|   |   | =0, Use curved elements  |

<table>
<thead>
<tr>
<th>2</th>
<th>ISIGF</th>
<th>&quot;</th>
</tr>
</thead>
</table>
|   | =2, Use constant source densities  
|   | =1, Use linear source densities  
|   | =0, Use parabolic source densities  |

<table>
<thead>
<tr>
<th>3</th>
<th>ICURVN</th>
<th>&quot;</th>
</tr>
</thead>
</table>
|   | =1, Read in curvature values  
|   | =0, EOD will compute curvatures  |

<table>
<thead>
<tr>
<th>4</th>
<th>NONEWF</th>
<th>&quot;</th>
</tr>
</thead>
</table>
|   | =1, Use old velocity formula  
|   | =0, New formula  |

<table>
<thead>
<tr>
<th>5-14</th>
<th>ALPHER</th>
<th>Main/F10.2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If only one body is input, ALPHER is the angle of attack (used by EOD)</td>
<td></td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>------</td>
</tr>
</tbody>
</table>
| 3       | 15     | IVORT| Main/I1       | =1, Perform axisymmetric, closed-duct solution  
<p>|         |        |      |               | =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    | PRC3   | Main/A4|               | Title &quot;EOD&quot;                                |
| 17-20   | NO6    | Main/I4|               | =0                                         |
| 21      | LPNCHO | Main/I2|               | Flag A                                     |
|         |        |      |               | =1, Do not save output for EOD on Unit 17   |
| 22      | IPIOTA | Main/I2|               | Flag B. Plot area against x position (see Reference 1) |
| 23-24   | IPOTC  | Main/I2|               | Flag C                                     |
|         |        |      |               | =+1, Plot curvature versus S               |
|         |        |      |               | =-1, Plot curvature versus X               |
| 22-24   | IREAD  | Main/I2|               | Flag D                                     |
|         |        |      |               | =0 (obselete)                              |</p>
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>37</td>
<td>IAB</td>
<td>Main/I1</td>
<td>Flag J. Redo geometry from point (see Reference 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Flag E. Redo entire geometry via direct interpolation (see Reference 1)</td>
</tr>
<tr>
<td>47</td>
<td></td>
<td>IREDON(1)</td>
<td>&quot;</td>
<td>Flag F. LPNCHO for any redo</td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>IREDON(2)</td>
<td>&quot;</td>
<td>Flag G. IPLOTA for any redo</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td>IREDON(3)</td>
<td>&quot;</td>
<td>Flag H. IPOLOT for any redo</td>
</tr>
<tr>
<td>50-51</td>
<td></td>
<td>IREDON(4)</td>
<td>Main/I2</td>
<td>Flag I. IREAD for any redo</td>
</tr>
<tr>
<td>52-53</td>
<td></td>
<td>IREDON(5)</td>
<td>&quot;</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>1-12</td>
<td>XAA</td>
<td>Main/F12.5</td>
<td>x position of starting point for partial redo</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
<td>YAA</td>
<td>&quot;</td>
<td>y position of starting point for partial redo</td>
</tr>
<tr>
<td></td>
<td>25-36</td>
<td>XBB</td>
<td>&quot;</td>
<td>x position of ending point for partial redo</td>
</tr>
<tr>
<td></td>
<td>37-48</td>
<td>YBB</td>
<td>&quot;</td>
<td>y position of ending point for partial redo</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>ANBDYS</td>
<td>Main/F10.2</td>
<td>Number of bodies</td>
</tr>
<tr>
<td>11-20</td>
<td>DELS</td>
<td>&quot;</td>
<td>Spacing between points in region of interest</td>
<td></td>
</tr>
<tr>
<td>21-30</td>
<td>DELSMX</td>
<td>&quot;</td>
<td>Maximum spacing far from region of interest</td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td>XRI</td>
<td>&quot;</td>
<td>Axial distance at which surface distance equals zero</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1-4</td>
<td>NRAKE</td>
<td>Main/I4</td>
<td>Number of axial rake locations</td>
</tr>
<tr>
<td>8</td>
<td>1-8</td>
<td>XRAK</td>
<td>Main/F8.5</td>
<td>Axial location of rake</td>
</tr>
<tr>
<td>9-16</td>
<td>YLO</td>
<td>&quot;</td>
<td>y value of first point (lowest point) on rake at XRAK</td>
<td></td>
</tr>
</tbody>
</table>

27
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>17-24</td>
<td>YHI</td>
<td>Main/F8.5</td>
<td>y value of last point (highest point) on rake at XRAK</td>
</tr>
<tr>
<td></td>
<td>25-27</td>
<td>NY</td>
<td>Main/I3</td>
<td>Number of points in rake at XRAK: Restriction ΣNY&lt;200. Rake points are equally spaced, ΔY, between YHI and YLO where ΔY = ( \frac{YHI - YLO}{NY - 1} )</td>
</tr>
<tr>
<td>28-35</td>
<td>XTRAN</td>
<td>Main/F8.0</td>
<td>Value of axial translation of rake</td>
<td></td>
</tr>
<tr>
<td>36-43</td>
<td>YTRAN</td>
<td>&quot;</td>
<td>Value of vertical translation of rake</td>
<td></td>
</tr>
<tr>
<td>44-51</td>
<td>XSCALE</td>
<td>&quot;</td>
<td>Value of axial scaling of rake</td>
<td></td>
</tr>
<tr>
<td>52-59</td>
<td>YSCALE</td>
<td>&quot;</td>
<td>Value of vertical scaling of rake</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1-10</td>
<td>TYPBDY</td>
<td>Main/F10.0</td>
<td>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.</td>
</tr>
<tr>
<td>11-20</td>
<td>ASNEG</td>
<td>&quot;</td>
<td>=Number of segments for the particular body, except as stated in TYPBDY</td>
<td></td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 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 |

Note: All lifting bodies must be input prior to any nonlifting bodies.

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

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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| 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
used, the rate at which the point spacing, \( ds \), changes near one end \( ds_i = ds_{i-1} - (\text{Rate})(ds_{i-1}) \) 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 < 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.).
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1-10</td>
<td>ENREED</td>
<td>Main/F10.2</td>
<td><strong>=1.</strong>, 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=10.</strong>, 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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=-1.</strong>, Fits a lemniscate between a straight line and a point. Input is three coordinates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=-3.</strong>, Fits a cubic between two straight lines. Input 4 coordinates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=-4.0</strong>, 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=99.</strong>, For direct interpolation option over one segment (see input instructions for card 12).</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>11-20</td>
<td>REEDEN(1)</td>
<td>Main/F10.2</td>
<td>Input exponent of x-term for bisuperellipse equation. Blank for all other segment types [1].</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>REEDEN(2)</td>
<td>&quot;</td>
<td>Input exponent of y-term for bisuperellipse [1].</td>
</tr>
<tr>
<td>11</td>
<td>1-12</td>
<td>XIN(1)</td>
<td>Main/6F12.5</td>
<td>x-coordinate for specified points</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
<td>XIN(2)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-36</td>
<td>XIN(3)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>37-48</td>
<td>XIN(4)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>49-60</td>
<td>XIN(5)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>61-72</td>
<td>XIN(6)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1-12</td>
<td>YIN(1)</td>
<td>Main/6F12.5</td>
<td>y-coordinate for specified points</td>
</tr>
<tr>
<td></td>
<td>13-24</td>
<td>YIN(2)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25-36</td>
<td>YIN(3)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>37-48</td>
<td>YIN(4)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>49-60</td>
<td>YIN(5)</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>61-72</td>
<td>YIN(6)</td>
<td>&quot;</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>11a</td>
<td></td>
<td>Z(1)</td>
<td>Name list/ $BODYIN/</td>
<td>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.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>12a</td>
<td>DONE</td>
<td>Name list/ $AUXIN/$</td>
<td>A logical variable which should be input as: =.TRUE.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BYPASS</td>
<td></td>
<td></td>
<td>=.TRUE. if no refinement of points is required</td>
</tr>
</tbody>
</table>

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:

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

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

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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>NONEWF</td>
<td>BASIC1/I1</td>
<td>=0, use the newest formulae</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>=1, use the old formulae (implies flat elements and constant $c$).</td>
</tr>
<tr>
<td>5</td>
<td>6-10</td>
<td>IFORMT</td>
<td>&quot;</td>
<td>Input format flag (see card 6).</td>
</tr>
<tr>
<td>6-10</td>
<td>NN</td>
<td>BASIC1/I5</td>
<td></td>
<td>Number of defining end points for the body.</td>
</tr>
<tr>
<td>11-20</td>
<td>MX</td>
<td>BASIC1/F10.0</td>
<td></td>
<td>x-multiplier value (default=1.0).</td>
</tr>
<tr>
<td>21-30</td>
<td>MY</td>
<td>&quot;</td>
<td></td>
<td>y-multiplier value (default=1.0).</td>
</tr>
<tr>
<td>31-40</td>
<td>THETA</td>
<td>&quot;</td>
<td></td>
<td>Coordinate rotation value (degrees, measured about -z axis).</td>
</tr>
<tr>
<td>41-50</td>
<td>ADDX</td>
<td>&quot;</td>
<td></td>
<td>x-increment (to be applied to all input x-coordinates for this body).</td>
</tr>
<tr>
<td>51-60</td>
<td>ADDY</td>
<td>&quot;</td>
<td></td>
<td>y-increment (to be applied to all input y-coordinates for this body).</td>
</tr>
<tr>
<td>5</td>
<td>6-10</td>
<td>BDN</td>
<td>BASIC1/I5</td>
<td>Body number (sequential for bodies, zero for off-body points).</td>
</tr>
<tr>
<td>16-20</td>
<td>SUBKS</td>
<td>&quot;</td>
<td></td>
<td>Subcase flag.</td>
</tr>
<tr>
<td>26-30</td>
<td>NLF</td>
<td>&quot;</td>
<td></td>
<td>Nonlifting flag (for combination cases only).</td>
</tr>
<tr>
<td>31-40</td>
<td>XE</td>
<td>BASIC1/F10.0</td>
<td></td>
<td>Semi-major axis for ellipse cases (input if FLG07#0).</td>
</tr>
<tr>
<td>41-50</td>
<td>YE</td>
<td>&quot;</td>
<td></td>
<td>Semi-minor axis for ellipse cases (input if FLG07#0).</td>
</tr>
<tr>
<td>6</td>
<td>1-60</td>
<td>UTX1(I)</td>
<td>BASIC1/6E13.4</td>
<td>x-coordinates.</td>
</tr>
<tr>
<td>6'</td>
<td>1-60</td>
<td>UTY1(I)</td>
<td>&quot;</td>
<td>y-coordinates.</td>
</tr>
<tr>
<td>6a</td>
<td>1-10</td>
<td>UTX1(I)</td>
<td>BASIC1/F10.5</td>
<td>x-coordinates.</td>
</tr>
<tr>
<td>11-20</td>
<td>UTY1(I)</td>
<td>&quot;</td>
<td></td>
<td>y-coordinates.</td>
</tr>
<tr>
<td>6b</td>
<td>1-10</td>
<td>UTX1(I)</td>
<td>BASIC1/F10.0</td>
<td>x-coordinates.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>----------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6b</td>
<td>21-30</td>
<td>UTY1(I)</td>
<td>BASIC1/</td>
<td>y-coordinates.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F10.0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1-60</td>
<td>HCURV(II)</td>
<td>BASIC1/</td>
<td>Curvature values for the NN-1 elements which constitute this body.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6E13.4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6-10</td>
<td>NN</td>
<td>BASIC1/</td>
<td>Number of automatically generated mass flow rakes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>I5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1-10</td>
<td>X1</td>
<td>IRAKES/</td>
<td>x-coordinate of start of the rake.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>F10.5</td>
<td></td>
</tr>
<tr>
<td>11-20</td>
<td>Y1</td>
<td>&quot;</td>
<td>&quot;</td>
<td>y-coordinate of start of the rake.</td>
</tr>
<tr>
<td>21-30</td>
<td>X2</td>
<td>&quot;</td>
<td>&quot;</td>
<td>x-coordinate of end of the rake.</td>
</tr>
<tr>
<td>31-40</td>
<td>Y2</td>
<td>&quot;</td>
<td>&quot;</td>
<td>y-coordinate of end of the rake.</td>
</tr>
<tr>
<td>41-45</td>
<td>N</td>
<td>IRAKES/</td>
<td>Number of intervals to be used in the rake (note 4&lt;N&lt;200 and N must be an even integer).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>I5</td>
<td>I5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6-10</td>
<td>NUN</td>
<td>BASIC2/</td>
<td>Flow identification number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>I5</td>
<td></td>
</tr>
<tr>
<td>16-20</td>
<td>MSF</td>
<td>&quot;</td>
<td>=0, axisymmetric onset flow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>=1, crossflow onset flow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>=2, both 0 and 1.</td>
<td></td>
</tr>
<tr>
<td>21-30</td>
<td>TYPE</td>
<td>&quot;</td>
<td>= +1.0, velocity will be input in x,y component form</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>= 0., velocity will be input in normal tangential form</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>=-1.0, automatic generation of flow due to rotation about the z-axis (for crossflow only).</td>
<td></td>
</tr>
<tr>
<td>31-40</td>
<td>FG</td>
<td>&quot;</td>
<td>Flow generator constant.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1-60</td>
<td>NG(I)</td>
<td>BASIC2/</td>
<td>x or normal component velocity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6F10.0</td>
<td></td>
</tr>
<tr>
<td>11'</td>
<td>1-60</td>
<td>TG(I)</td>
<td>&quot;</td>
<td>y or tangential component velocity.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Description</th>
<th>Subroutine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title card.</td>
<td>READS</td>
</tr>
<tr>
<td>2</td>
<td>Number of on-body and off-body points.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Input initial conditions of the flow.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Control flag.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Circumferential coordinates.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Location of control station rake.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Input for defining zero surface distance.</td>
<td></td>
</tr>
</tbody>
</table>

English engineering units are used throughout the program.

- Length, in.
- Velocities, ft/sec
- Angles, deg
- Pressures, lb/ft²
- Temperature, °R
- Density, slug/ft³
- Force, lb
- Weight flow, lb/sec
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-72</td>
<td>TITLE</td>
<td>READS/18A4</td>
<td>Title card.</td>
</tr>
<tr>
<td>2</td>
<td>1-4</td>
<td>NT(1)</td>
<td>READS/I4</td>
<td>Number of on-body points for the closed-end solution.</td>
</tr>
<tr>
<td></td>
<td>5-8</td>
<td>NP(1)</td>
<td>&quot;</td>
<td>Total number of off-body points.</td>
</tr>
<tr>
<td></td>
<td>9-12</td>
<td>NT(2)</td>
<td>&quot;</td>
<td>Number of on-body points for the open-end solution (eliminate the last body).</td>
</tr>
<tr>
<td></td>
<td>13-16</td>
<td>NP(2)</td>
<td>&quot;</td>
<td>Total number of off-body points.</td>
</tr>
<tr>
<td></td>
<td>17-20</td>
<td>NID</td>
<td>&quot;</td>
<td>Number of EOD I.D. cards.</td>
</tr>
<tr>
<td></td>
<td>21-24</td>
<td>KSKIP</td>
<td>&quot;</td>
<td>=0, for first case of COMBYN =1, for successive cases using the same EOE output.</td>
</tr>
<tr>
<td></td>
<td>25-28</td>
<td>IOVHUB</td>
<td>&quot;</td>
<td>=0, hub vorticity solution from EOD is not read. =1, hub vorticity solution from EOD is read.</td>
</tr>
<tr>
<td>3</td>
<td>1-8</td>
<td>VC</td>
<td>READS/F8.3</td>
<td>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#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 &quot;control point&quot; rather than a velocity. Note: All three inputs, WDOT, ICTLPT, and VC, must be nonzero when the &quot;control point&quot; calculation is desired.</td>
</tr>
<tr>
<td></td>
<td>9-16</td>
<td>VINF</td>
<td>&quot;</td>
<td>Free-stream velocity.</td>
</tr>
<tr>
<td></td>
<td>17-24</td>
<td>ALFAF</td>
<td>&quot;</td>
<td>Angle of attack, 0.0 for free-stream perpendicular to inlet axis. Note that $\alpha_| = \alpha - 90^\circ$. For &quot;control point&quot; cases only, ALFAF will be calculated when ALFAF is input as $=999.0$.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>25-32</td>
<td>TTOTAL</td>
<td>READS/F8.3</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>33-40</td>
<td>ELND</td>
<td></td>
<td>ELND is the arbitrary length used for scaling or normalizing. Refer to KND input, card 5. See also CUTOFF input below.</td>
</tr>
<tr>
<td></td>
<td>41-48</td>
<td>YWING</td>
<td></td>
<td>Upper limit of integration for surface forces (used in subroutine INFRCE).</td>
</tr>
<tr>
<td></td>
<td>49-56</td>
<td>UTIP</td>
<td></td>
<td>Rotor tip speed. Need not be input unless relative rotor inlet quantities are desired. (See COMBLYN OUTPUT.)</td>
</tr>
<tr>
<td></td>
<td>57-64</td>
<td>VA</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>65-72</td>
<td>PT</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>73-80</td>
<td>CUTOFF</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>1-8</td>
<td>PSTAT</td>
<td></td>
<td>Static pressure.</td>
</tr>
<tr>
<td></td>
<td>9-16</td>
<td>TSTAT</td>
<td></td>
<td>Static temperature. (If PSTAT and TSTAT are not 0.0, total pressure (PT) and total temperature (TTOTAL) will be calculated using PSTAT and TSTAT.</td>
</tr>
<tr>
<td></td>
<td>17-24</td>
<td>WDOT</td>
<td></td>
<td>Weight flow - required unless VC≠0 and concurrently ICTLPT=0.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>25-32</td>
<td>DELQ</td>
<td>READS/F8.3</td>
<td></td>
<td>Increment in mass flow fraction for spacing for calculated streamlines. NOTE: Default Value = 0.1, if DELQ is input as 0.</td>
</tr>
<tr>
<td>33-40</td>
<td>VPERIN</td>
<td>&quot;</td>
<td></td>
<td>Blank.</td>
</tr>
<tr>
<td>41-48</td>
<td>MC</td>
<td>&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49-56</td>
<td>MINF</td>
<td>&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1-4</td>
<td>NTHETA</td>
<td>READS/I4</td>
<td>Number of THETA's where THETA is the circumferential coordinate. If NTHETA=0, one THETA (THETA&lt;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.</td>
</tr>
<tr>
<td>5-8</td>
<td>NCLO</td>
<td>&quot;</td>
<td></td>
<td>One rake must be chosen as the control station. NCLO is the number of the first point on this rake.</td>
</tr>
<tr>
<td>9-12</td>
<td>NCHI</td>
<td>&quot;</td>
<td></td>
<td>The number of the last point on the control station rake.</td>
</tr>
<tr>
<td>13-16</td>
<td>NX</td>
<td>&quot;</td>
<td></td>
<td>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.)</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>5</td>
<td>17-20</td>
<td>KND</td>
<td>READS/I4</td>
<td>Flag for scaling variables before velocity and pressure calculations and also for nondimensionalizing after calculations and just before printout:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Scaling</strong>: 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Nondimensionalizing</strong>: 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. <strong>NOTE</strong>: If CUTOFF is nonzero, surface distance will automatically be normalized by ELND before printout.</td>
</tr>
<tr>
<td>21-24</td>
<td>NOTHET</td>
<td>&quot;</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>25-28</td>
<td>ICTLPT</td>
<td>&quot;</td>
<td></td>
<td>Index number (from EOD output) of the desired &quot;control point&quot; where a known pressure ratio is to be input in lieu of a control station velocity. See VC.</td>
</tr>
<tr>
<td>29-32</td>
<td>ISWIRL</td>
<td>&quot;</td>
<td></td>
<td>(Required when NTHETA=0). Index number of point on shroud where three-dimensional, on-body stream-line calculation will begin; preferably near the fan face.</td>
</tr>
<tr>
<td>6</td>
<td>1-80</td>
<td>THETA(I)</td>
<td>READS/10F8.3</td>
<td>Circumferential coordinate in degrees (number of THETA's read in depends on NTHETA, NTHETA&lt;10).</td>
</tr>
<tr>
<td>7</td>
<td>1-10</td>
<td>XTEST</td>
<td>READS/F10.2</td>
<td>Axial location of control station rake. Must be compatible with NCLO and NCHI.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>11-20</td>
<td>YTESTH</td>
<td>READS/F10.2</td>
<td>Y on the hub at XTEST (control station).</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>YTESTS</td>
<td>&quot;</td>
<td>Y on the shroud at XTEST (control station).</td>
</tr>
<tr>
<td>8</td>
<td>1-10</td>
<td>XRI</td>
<td>&quot;</td>
<td>Value of X at which the surface distance is zero.</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>YRIHUB</td>
<td>&quot;</td>
<td>Y on the hub at XRI.</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>YRISHR</td>
<td>&quot;</td>
<td>Y on the shroud at XRI.</td>
</tr>
<tr>
<td></td>
<td>31-34</td>
<td>NHUBMX</td>
<td>READS/I4</td>
<td>The number of the last point on the hub (this can be found in the printed output of SCIRCL).</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Card Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of bodies; degree of governing equations</td>
</tr>
<tr>
<td>2</td>
<td>Flow field control flags</td>
</tr>
<tr>
<td>3</td>
<td>Trajectory control flags</td>
</tr>
<tr>
<td>4</td>
<td>Initial conditions of flow field</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Initial conditions of particles</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Number of size increments</td>
</tr>
<tr>
<td>10</td>
<td>This card is input only when NSI&gt;1; number of 10 cards = NSI</td>
</tr>
</tbody>
</table>

45
<table>
<thead>
<tr>
<th>Card Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Input for polynomial least square fit</td>
</tr>
<tr>
<td>12</td>
<td>TITLE of experimental data</td>
</tr>
<tr>
<td>13</td>
<td>Number of experimental data</td>
</tr>
<tr>
<td>14</td>
<td>Number of 13 cards = NASA</td>
</tr>
<tr>
<td>15</td>
<td>Input for plotting</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| 1        | 6-10   | NBDY | Main/15        | 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). |
<table>
<thead>
<tr>
<th>Card No.</th>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>6-10</td>
<td>LEQM</td>
<td>Main/I5</td>
<td>=1, The initial particle velocity is equal to the flow at the initial particle location.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#1, The initial velocity conditions input on cards 4 and 6 are used as the initial particle velocity.</td>
</tr>
<tr>
<td>16-20</td>
<td>LSYM</td>
<td></td>
<td>Explanation</td>
<td>Symmetric flow field flag =0, Unsymmetric flow field (general case).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>=1, Symmetric flow field (only half plane is computed). Axisymmetric case LSYM=0.</td>
</tr>
<tr>
<td>26-30</td>
<td>LRANG</td>
<td></td>
<td>Explanation</td>
<td>#0, Locates approximate values of ( y_{0} ) and ( y_{05} ) (see Section 3.1.6).</td>
</tr>
<tr>
<td>3</td>
<td>6-10</td>
<td>LIM</td>
<td>Explanation</td>
<td>#0, Calculates surface impingement limits ( y_{0} ) and ( y_{05} ).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>=0, Calculates single particle trajectories; LRANG=0 (see Section 3.1.6).</td>
</tr>
<tr>
<td>16-20</td>
<td>LOPT</td>
<td></td>
<td>Explanation</td>
<td>#0, Stores details of particle trajectories on data Tape 04.</td>
</tr>
<tr>
<td>26-30</td>
<td>LPL   O</td>
<td></td>
<td>Explanation</td>
<td>#0, Executes subroutine for plotting local collection efficiency, 8, versus surface distance, ( s/\gamma_{C} ).</td>
</tr>
<tr>
<td>36-40</td>
<td>LTNL</td>
<td></td>
<td>Explanation</td>
<td>=1, Computes flow over airfoil with walls (i.e., wind tunnel simulation).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#1, Computes flow over airfoil without walls and inlet case. Axisymmetric case, LTNL#1.</td>
</tr>
<tr>
<td>46-50</td>
<td>LXOR</td>
<td></td>
<td>Explanation</td>
<td>Control flag for adjusting the upstream x-coordinates, XORC, for upstream position of particle release (see Section 3.1.3).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>=0, The upstream x-coordinates of particle release is not adjusted. Particles are released from XORC input on card 6 by user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>=1, Particles are released from the x-coordinate position adjusted by the criteria: (</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>4</td>
<td>6-15</td>
<td>G</td>
<td>Main/F10.0</td>
<td>Acceleration of gravity, m/s².</td>
</tr>
<tr>
<td>21-30</td>
<td>YINF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Velocity of free-stream, m/s.</td>
</tr>
<tr>
<td>5</td>
<td>6-15</td>
<td>DP</td>
<td>&quot;</td>
<td>Particle volume median diameter in microns (10⁻⁶ m).</td>
</tr>
<tr>
<td>21-30</td>
<td>RL</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Reference length, e.g., chord length of the airfoil, m.</td>
</tr>
<tr>
<td>36-45</td>
<td>TIMSTP</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Initial value of the time step used in the Adams-Moulton predictor-corrector method (Section 2.3).</td>
</tr>
<tr>
<td>51-60</td>
<td>YLIM</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Accuracy criteria for computing the surface impingement limits (Section 3.1.6) YLIM=10⁻⁵ m is recommended.</td>
</tr>
<tr>
<td>66-75</td>
<td>CF</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Cunningham correction factor. Use if volume median diameter DP&lt;10μ; otherwise, CF=1.</td>
</tr>
<tr>
<td>6</td>
<td>6-15</td>
<td>ATK</td>
<td>&quot;</td>
<td>Initial value of the angle, α, Figure 2.1 (deg).</td>
</tr>
<tr>
<td>21-30</td>
<td>PIT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Initial value of the angle, θ, Figure 2.1 (deg).</td>
</tr>
<tr>
<td>36-45</td>
<td>PITDOT</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Initial value of ¨θ (deg/sec).</td>
</tr>
<tr>
<td>51-60</td>
<td>XORC</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Upstream x-coordinate position of particle release, x/ℓc (LXOR=0).</td>
</tr>
<tr>
<td>66-75</td>
<td>YORC</td>
<td>&quot;</td>
<td>&quot;</td>
<td>Upstream y-coordinate position of particle release, y/ℓc (NPL=1, LRANG=0, and LIM=0).</td>
</tr>
<tr>
<td>7</td>
<td>6-15</td>
<td>XREAR</td>
<td>&quot;</td>
<td>Maximum downstream value of x/ℓc for which the flow field is computed.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>7</td>
<td>21-30</td>
<td>YYLO</td>
<td>Main/F10.0</td>
<td>Minimum value of $y/l_c$ for which the flow field is computed.</td>
</tr>
<tr>
<td></td>
<td>36-45</td>
<td>YYUP</td>
<td>&quot;</td>
<td>Maximum value of $y/l_c$ for which the flow field is computed.</td>
</tr>
<tr>
<td></td>
<td>51-60</td>
<td>YMAX</td>
<td>&quot;</td>
<td>Initial guess of the upper limit $y_{ou}/l_c$ (LIM=1) (Section 3.1.6).</td>
</tr>
<tr>
<td></td>
<td>66-75</td>
<td>YMIN</td>
<td>&quot;</td>
<td>Initial guess of the lower limit $y_{ol}/l_c$ (LIM=1) (Section 3.1.6).</td>
</tr>
<tr>
<td>8</td>
<td>6-15</td>
<td>ADEPS</td>
<td>&quot;</td>
<td>Convergence criteria of the Adams-Moulton predictor-corrector method. ADEPS=0.001 is recommended.</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>ANLFLO</td>
<td>&quot;</td>
<td>Angle of flow direction relative to airfoil chord line measured from positive direction of x-axis.</td>
</tr>
<tr>
<td></td>
<td>51-60</td>
<td>EPSX</td>
<td>&quot;</td>
<td>Accuracy criteria for the case LXOR=1 (see card 3); EPSX=0.001 is recommended.</td>
</tr>
<tr>
<td>9</td>
<td>6-10</td>
<td>NSI</td>
<td>Main/I5</td>
<td>Number of droplet size increments.</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>Number of 10 cards = NSI.</td>
</tr>
</tbody>
</table>

This card is input only when NSI>1.

<table>
<thead>
<tr>
<th>Column</th>
<th>Code</th>
<th>Routine/Format</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>PLWC</td>
<td>Main/F10.0</td>
<td>Percentage of liquid water content for each drop size.</td>
</tr>
<tr>
<td>11-20</td>
<td>DPD</td>
<td>&quot;</td>
<td>Distribution of the particle diameter ratio to volume median diameter.</td>
</tr>
<tr>
<td>21-30</td>
<td>CFP</td>
<td>&quot;</td>
<td>Cunningham correction factor corresponding to DPD (see Table 6.2). If DP*DPD&gt;10u, CFP=1.</td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------</td>
<td>----------------</td>
</tr>
<tr>
<td>11</td>
<td>1-80</td>
<td>TITLE</td>
<td>Main/20A4</td>
</tr>
<tr>
<td>12</td>
<td>6-10</td>
<td>NCOEF</td>
<td>Main/I5</td>
</tr>
<tr>
<td>21-25</td>
<td>NPTS</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>36-40</td>
<td>NS</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1-80</td>
<td>TITLE</td>
<td>Main/20A4</td>
</tr>
<tr>
<td>14</td>
<td>1-5</td>
<td>NASA</td>
<td>Main/I5</td>
</tr>
<tr>
<td>15</td>
<td>1-10</td>
<td>SFOIL</td>
<td>Main/F10.5</td>
</tr>
<tr>
<td>11-20</td>
<td>EFC</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1-80</td>
<td>XTITL</td>
<td>APLLOT/20A4</td>
</tr>
<tr>
<td>17</td>
<td>1-80</td>
<td>YTITL</td>
<td>&quot;</td>
</tr>
<tr>
<td>18</td>
<td>6-15</td>
<td>YHEIT</td>
<td>APLLOT/F10.0</td>
</tr>
<tr>
<td>21-30</td>
<td>YMAX</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>36-45</td>
<td>XMIN</td>
<td>&quot;</td>
<td></td>
</tr>
<tr>
<td>Card No.</td>
<td>Column</td>
<td>Code</td>
<td>Routine/Format</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------</td>
<td>----------------</td>
</tr>
<tr>
<td>19</td>
<td>6-15</td>
<td>XLENG</td>
<td>APLOT/ F10.0</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>XMAX</td>
<td>&quot;</td>
</tr>
<tr>
<td></td>
<td>36-45</td>
<td>XMIN</td>
<td>APLOT/ F10.0</td>
</tr>
</tbody>
</table>

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, ı, ı, ȳ, and ě 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, ı, is also recorded on Tape 06. The value of ı 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))

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

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>XP</td>
<td>x-coordinate of the particle, (x/\lambda_c)</td>
</tr>
<tr>
<td>11-20</td>
<td>YPL</td>
<td>y-coordinate of the particle, (y/\lambda_c)</td>
</tr>
<tr>
<td>21-30</td>
<td>SW(IW)</td>
<td>The parameter is used to indicate the end of each trajectory, i.e., if SW#88.8888 the trajectory is terminated.</td>
</tr>
</tbody>
</table>
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 \( \theta \)
Y(6) Value of \( \dot{\theta} \)
VINF Free-stream velocity, \( W_\infty \)
YP \( y/\ell_c \) coordinate of current particle location
YHIT Initial \( y/\ell_c \) coordinate of a particle which impacts body
YMIS Initial \( y/\ell_c \) coordinate of a particle which moves away from the body
SL Lower surface impingement limit, \( s_L/\ell_c \)
SU  Upper surface impingement limit, \( s_U/\xi_c \)

ZL  Lower limit, \( y_{O\alpha}/\xi_c \), corresponding to SL

ZU  Upper limit, \( y_{OU}/\xi_c \), corresponding to SU

Flowchart:  See Figure 4.1
Figure 4.1 Flowchart of main program.

56
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 $y_{0u}$ and $y_{0l}$.

Flowchart: See Figure 4.3
YO(1) = YMX
YO(2) = YMN
DY = YO(1) - YO(2)
IAC = 0
IW = 0
IPROT = 0

IW = IW + 1
IPROT = IPROT + 1

IPROT G30

YL(IW) = YP - YRER

LEQM NE.

Set Initial Conditions of the Particle for LEM=1 Case

Call ADAMS

YN(IW) = 09,9999

YW(IW) = YP - YRER

Figure 4.3 Flowchart of subroutine RANGE.

60
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., \( \dot{x}, \dot{y}, \dot{\theta}, x, y, \dot{\theta} \), 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, \(x/\lambda_c\), of particle position
- \(Y\) Dimensionless y coordinate, \(y/\lambda_c\), of particle position
- \(VXC\) Dimensionless x component velocity, \(W_x/W_\infty\), of airflow
- \(VYC\) Dimensionless y component velocity, \(W_y/W_\infty\), 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, $\beta$, 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, $\eta$, 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 APLLOT and TRAJCT

Objective: Plot routines for $\beta$ 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>
<thead>
<tr>
<th>Error Message Generated by the Program</th>
<th>Problem and Corrective Action</th>
<th>Error Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of y-coordinates read is not equal to the number of x-coordinates read.</td>
<td>Corrective Action: Check the input data file, Tape 05, for equal number of x- and y-coordinates.</td>
<td>Subroutine READIN</td>
</tr>
<tr>
<td>Time-halving loops exceed 100.</td>
<td>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.</td>
<td>Subroutine ADAMS</td>
</tr>
<tr>
<td>IPORT.GE.31 in SUB.RANGE--program stopped.</td>
<td>Problem: Initially guessed values of YMAX and YMIN are too far from the body or ( \Delta Y = 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 ( \Delta Y ).</td>
<td>Subroutine RANGE</td>
</tr>
<tr>
<td>Error Message Generated by the Program</td>
<td>Problem and Corrective Action</td>
<td>Error Location</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>FORMAT 172: Number of particle trajectories exceeds NSEAR, program stopped.</td>
<td>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).</td>
<td>MAIN</td>
</tr>
<tr>
<td>FORMAT 747: Number of trajectories is more than 100, program stopped.</td>
<td>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).</td>
<td>MAIN</td>
</tr>
<tr>
<td>Particle released between the region ZL-ZU crosses upper or lower impingement limit trajectory, program stopped.</td>
<td>Corrective Action: Check the flow field around the body and increase absolute value of pseudo-surface DX (Card 2 on Tape 05).</td>
<td>MAIN</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>TEST CASE RELEASE 2.0</th>
<th>2.00</th>
<th>0.02</th>
<th>0.06</th>
<th>1.00</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>y/l_c</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.10</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.14</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.18</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.22</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.26</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.30</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.34</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.38</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.42</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.46</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.54</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.58</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.62</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.66</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.74</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.78</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.82</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.86</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.94</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0.98</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.04</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.08</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.12</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.16</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.20</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.24</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.28</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.32</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.36</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.40</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.44</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.48</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.52</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.56</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.60</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.64</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.68</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.72</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.76</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.80</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.84</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.88</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.92</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1.96</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.04</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.08</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.12</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.16</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.20</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.24</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.28</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.32</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.36</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.40</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.44</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.48</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.52</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.56</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.60</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.64</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.68</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.72</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.76</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.80</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.84</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.88</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.92</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.96</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td></td>
<td>0.0428</td>
<td>0.5565</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

ORIGINAL PAGE IS OF POOR QUALITY
TABLE 5.2 Input Data File for Geometry Generation Program SCIRCL.

<table>
<thead>
<tr>
<th>CASE RELEASE</th>
<th>7,0</th>
<th>2-BOV1</th>
<th>FGCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE 1</td>
<td>110</td>
<td>0</td>
<td>0.9587E+00 0.8935E+00 0.6873E+00 0.7303E+00 0.5940E+00 0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
<tr>
<td>CASE 2</td>
<td>120</td>
<td>0</td>
<td>0.8935E+00 0.6873E+00 0.7303E+00 0.5940E+00 0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
<tr>
<td>CASE 3</td>
<td>130</td>
<td>0</td>
<td>0.6873E+00 0.7303E+00 0.5940E+00 0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
<tr>
<td>CASE 4</td>
<td>140</td>
<td>0</td>
<td>0.7303E+00 0.5940E+00 0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
<tr>
<td>CASE 5</td>
<td>150</td>
<td>0</td>
<td>0.5940E+00 0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
<tr>
<td>CASE 6</td>
<td>160</td>
<td>0</td>
<td>0.7201E+00 0.4202E+00 0.5420E+00 0.3201E+00 0.3000E+00</td>
</tr>
</tbody>
</table>

*Note: The table contains input data for the geometry generation program SCIRCL.*
TABLE 5.2 (continued).

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>90</td>
<td>0</td>
</tr>
<tr>
<td>0.4280E+01</td>
<td>-0.4280E+01</td>
<td>-0.4280E+01</td>
</tr>
<tr>
<td>0.4280E+01</td>
<td>-0.4280E+01</td>
<td>-0.4280E+01</td>
</tr>
<tr>
<td>0.4280E+01</td>
<td>-0.4280E+01</td>
<td>-0.4280E+01</td>
</tr>
<tr>
<td>0.4280E+01</td>
<td>-0.4280E+01</td>
<td>-0.4280E+01</td>
</tr>
</tbody>
</table>

In data file Tape 05 (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 \( y = -0.7 \) 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).

$NBODY$ 3 $NLPS** 1 $NSEAR 60 $NED** 4
$LEOM 1 $LSYM 0 $LRANG 0
$NLS** 0 $LPOL 0 $LTHN 0
$G $ 0.0 $VINP 44.35
$SPAN 3.0 $RL 1.0 $TIMSP 0.0010 $VIL * 0.000001 $CF $ 1.0
$PRATK 0.0 $PIT 0.0 $FINT 0.0 $YORP $-0.9 $YORPI $-0.5
$PIT 0.40 $YLO $-0.60 $UP $-0.4 $YPRD 0.52 $MIN $-0.02
$EDFL 0.0010 $ANLF 0.0 $ANLCR 0.9 $EPFX $0.00:
$TI $1 $INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
$NCDEF 4 $NFTS* 10 $ND* 0
$EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY:
$-0.08 0.045
$INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
$NCDEF 4 $NFTS* 10 $ND* 0
$EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY:
$-0.03 0.045
$INITIAL INPUT DATA FOR LOCAL CATCH EFFICIENCY CALCULATIONS
$NCDEF 4 $NFTS* 10 $ND* 0
$EXPERIMENTAL DATA OF LOCAL CATCH EFFICIENCY:
$-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 $x_0 = -0.5$, $\alpha = 0°$ (lower body).

$y_0 = 0.690$
$y_0 = -0.700$
$y_0 = -0.705$
$y_0 = -0.710$

$76$
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).

<table>
<thead>
<tr>
<th>TEST CASE RELEASE 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>443.00 60.00 -60.000 43.180 0.00 0.000 0.000 0.000 0.000 0.000 0.000</td>
</tr>
<tr>
<td>2.000 90.000 0.000 0.000 0.100</td>
</tr>
<tr>
<td>2.000 100.000</td>
</tr>
<tr>
<td>0.000 0.2055 0.00 0.30 0.85</td>
</tr>
</tbody>
</table>

*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 $x_o = -0.5$, $\alpha = 30^\circ$. 
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, \( \beta \), 2 to 3 hours of CPU time are used to determine one plot of \( \beta \). 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


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


PARTICLE TRAJECTORY COMPUTER PROGRAM LISTING

DIMENSION Y(6), XDOT(6), TITLE(20), XL(1000), YL(1000), N(10), CFP(10), EP(10)

PROGRAM FOR CALCULATING PARTICLE TRAJECTORIES.

Title units are used.

Input data files:

Tape05: Aerofoil geometry. The same file Tape03.

Tape02: The same as the output file Tape05 of NASA potential flow code, used for air velocity calculations.

Output data files:

Tape01: Data stored in Tape01 is written to Tape03 latter.

Tape03: Same as the output file Tape05 of NASA potential flow code as well as information of wind field.

Tape08: Data stored in Tape08 is for trajectory plots.

The data stored are X, Y, and Z with the format(2F10.5). The end of each trajectory (YF, YF) has been normalized by chord length.

INPUT DATA FILES:

Tape03
Tape05
Tape21

INITIAL INPUT OF THIS TRAJECTORY CODE AEROFOIL GEOMETRY. THE SAME FILE TAPE03

INPUT FILES TAPE05, TAPE03, TAPE21

INPUT DATA TAPE01 = TAPE03 FOR UNSYMMETRIC FLOW CASES

DATA USED FOR THE CALCULATION OF COLLECTION EFFICIENCY CREATED IF (LOPT .NE. 0).

CREATED IF (LOPT .NE. 0)

DATA STORED ARE X, YF, AND Z WITH THE FORMAT(2F10.5). THE END OF EACH TRAJECTORY

YF, YF HAS BEEN NORMALIZED BY CHORD LENGTH.

COMMON / COMB YND/V VREF
COMMON/OS X/ XA(1000), YA(1000)
COMMON/XTN/XFRONT, YFRONT, XREAR, YREAR, NTOTAL(5), NBDY, ID
COMMON/SBP/SU(1000), SV(150), DU(150), WM(5)
COMMON/IN1/X1, IN2, IN3, IN4, IN5, IN6, IN7, IN8, IN9
COMMON/CONT/IF(5), IDP(5), IC(5), IDP, IC, IC, IC, IC
COMMON/XFRONT/XFRONT(5), YFRONT(5), XRC, YRC(5), YRC
COMMON/SDS/S(1000), SU(ISO), ZO(150), I =
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/FSL/FLS, XFRONT, YFRONT, XREAR, YREAR, NFRONT(5), YFRONT(5)
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMMON/CONT/IF(5), IDP(5), IC(5), IC, IC, IC
COMMON/IN2/XIN, YIN, DPMP, RL, XREF, YREF, VRFX
COMM...
CALL VELOCITY(XIN, YIN, W1, W2)
YK=SOBT(W1, W2, T)
DUX=ABS(1.0-T)
DUX=DUX

131 CONTINUE
IF(INC.GE.XMAX) GO TO 135
IF(INC.GE.2) GO TO 135
WRITE(*,133) XIN, NC, MAX
IN8=IN8+1
GO TO 19
14 XIN=XCR, + RL
5 CONTINUE

C------ FINISH SEARCHING
75 IF(15.GE.0) DP=D1, 0.6-0.6*DPM*DFM
AMASS=PIDPMM+IDPMPF/5.0
TF=0.4+AMASS*(DFM/2.)**2
CBAR=DFM
SM=1.23
GO TO 35
WRITE(*,135) DFRM, RL, TIMSTP, MPL, YKREF, YKREF, AKKEF, PIT, PIT, DQT, 0.
77: IF (LMAX.EQ.0) GO TO 954
954 CONTINUE
DO 94 ID=1, NBY
REWIND 3
REWIND 9
IF(15.GE.0) DP=D1, 0.6-0.6*DFM
AMASS=PIIDPMM+IDPMPF/5.0
TF=0.4+AMASS*(DFM/2.)**2
CBAR=DFM
IF-functions
I=1
YDIM=TOTAL(IDK)
SP=SPM
R=(1 + YDIM(IDK))
YREAR=XRER
YFRONT=XRFR
WRITE(*,121) NBY, ID, XFRONT, YFRONT, XRFR, XRER, THICK, YUP, YLO
IF(LIM.EQ.0) GO TO 97
97 CONTINUE
DO 77 IJK=1, NPL
REWIND 1
IF(IJK.EQ.1) GO TO 4
WRITE(*,136)
C COMPUTATION OF DROPLET TRAJECTORIES
C IF LIM=1 SEARCH THE SURFACE RANGE WHERE THE PARTICLES IMPACT AT
IF(LIM.NE.0) CALL RANGE(YMAX, YMIN, TIMSTP, A0EPS, NEQ)
70
IUF=0
WRITE(*,123) 
C------ COMPUTATION OF DROPLET TRAJECTORIES
C IF LIM=1 SEARCH THE SURFACE RANGE WHERE THE PARTICLES IMPACT AT
IF(LIM.NE.0) CALL RANGE(YMAX, YMIN, TIMSTP, A0EPS, NEQ)
---- IUP=1 SEARCHING UPPER LIMIT
42 CONTINUE
   FIP=1NSTP
   IF(SW(EQ.99.9999))GO TO 75
   IF(ABS YIN-TIN)LE. YLIN) GO TO 75
   YIN=YHT+YMS/2.
   GO TO 48
47 IF(NPL .NE. 1) GO TO 85
   YIN=SRC#AL
   GO TO 46
65 AC=1.0
   D=ZU-ZL
   IF(AC*EQ.1) D=ZU
   IF(MOD(IF,100)) GO TO 101
   IF(110. EQ.101) YHIT=YIN
   IF(111. EQ.101) YHIT=YIN
   GO TO 42
48 IF(IW .EQ. 1) GO TO 49
   YHIS=YIN
   GO TO 42
49 IU=0
   IUP=2
   IF( YP.GT. YFRNT .OR.TP.GT. fRER) GO TO 73
   YHIT=YIN
   GO TO 42
72 CONTINUE
   FIP=1AMEPE
   CALL VELCTY(X,YP,UX,UY)
   IF(IJ. GT. ITOT.AND.SU(IU) .EQ.99.9999) GO TO 79
   IF(115 .GT. 1) GO TO 42
C---- SEARCHING SURFACE LIMITS
75 IF(SW(EQ.99.9999))GO TO 75
   YHIT=YIN
   GO TO 42
72 IF(IW .EQ. 1) GO TO 73
   YHIS=YIN
   GO TO 42
73 IF(IU .EQ. IPROT) GO TO 73
   if(IW .EQ. 1)GO TO 42
76 IF(IU .EQ. IPROT) GO TO 76
   ZO(1)=ZU
   ZL=-ZU
   GO TO 78
79 CONTINUE
   WRITE(6,793)
   GO TO 250
C---- UPPER LIMIT OR LOWER LIMIT IS FOUND
75 FIP=1AMEPE
   SL=SU(1)
   ZL=ZO(1)
   GO TO 78
76 CONTINUE
   SU=SU(IU)
   ZL=ZO(IU)
   YHT=YIN
   FIP=1AMEPE
   IF(SW(EQ.99.9999))GO TO 75
   IF(115 .GT. 1) GO TO 42
   WRITE(6,101)
   SL=SU
   ZL=ZO
   GO TO 78
C---- END OF LIMIT SEARCH
C---- TOTAL COLLECTION EFFICIENCY
   IF(SW(EQ.99.9999))GO TO 79
   E=(ZU-ZL)/THICK
   IF(SW(EQ.99.9999))GO TO 79
   WRITE(6,E,3101)
   IF(SW(EQ.99.9999))GO TO 79
   WRITE(6,E,3091)
   IF(SW(EQ.99.9999))GO TO 79
   WRITE(6,E,433)
IF(NPL.EQ.0) GO TO 236

IF(NPL.EQ.1) GO TO 19

IA=0

IS=1

IA=IA+1

CALL APLT(X(1),Y(1),NSET,LDF,FC,C2)

C---- TRAJECTORIES AND AIRFOIL GEOMETRY

C---- AIRFOIL GEOMETRY

CONTINUE

C---- DATA Plotting

C---- LOCAL CATCH EFFICIENCY

C---- CALCULATION OF LOCAL CATCH EFFICIENCY

C---- READ NASA/LEWIS DATA WHICH IS THEN STORED IN TAPE09

C---- NASA/LEWIS DATA WHICH IS THEN STORED IN TAPE09

C---- LOCAL CATCH EFFICIENCY

C---- DATA Plotting

C---- LOCAL CATCH EFFICIENCY

C---- LOCAL CATCH EFFICIENCY

C---- LOCAL CATCH EFFICIENCY

C---- LOCAL CATCH EFFICIENCY

C---- LOCAL CATCH EFFICIENCY

C---- LOCAL CATCH EFFICIENCY
CALL TRAJCT
94 CONTINUE
GO TO 236
99 WRITE(*,881)
WRITE(*,747)
C- - - - - FORMATS
101 FORMAT(94 CONTINUE
GO TO 236
102 WRITE(*,881)
GO TO 236
743 WRITE(*,747)
C- FORMATS
101 FORMAT(/,2X,'SYMMETRIC FLOW CASE IS CONFUSED',/)
120 FORMAT(/,2X,'THE PARTICLES ARE RELEASED FROM X/Y=',1PE12.5,/,120 FORMAT(/,2X,'NO OF PARTICLE TRAJECTORIES EXCEED
1 NSEE, PROGRAM STOPPED')
236 STOP
END
SUBROUTINE ADAMS!LAST-YD1.TSP.EPS.IJ.YPL.NEC!>
T-- EQUATION SOLVER
C-- THE ADAMS-MULTON PREDICTOR-CORRECTOR METHOD IS USED
DIMENSION YD1(!6),YLAST(!4),YPRED(!4),YC0RR(6),YP(!4,6).XPA(!4),tPA(!4)
DIMENSION A1(!6),A2(!6).A3(!4),A4(!4)
COMMON/XFRNT,YFRNT,XRER,YRER,NTOTAL(!5),NEiDY.ID
COMMON/XIN,YIN,DP,RL,VX,RUP,YREF,PITDOT,ATK,PIT,UREF.WXP.VYP
COMMON/S(1000).Z(150).IS
COMMON/LOPT, LECM, XREAR, YLO, YUP
IU,YPL, AND ALL THE COMMONS ARE USED EITHER
FOR PROGRAM STOP CONTROL OR FOR OPTIONAL PRINTS
LOOP FOR TIME STEP TO START USING SIMPLE RUNGE-KUTTA METHOD
H=TSP
ISTP=0
END
SUBROUTINE ADAMS!LAST-YD1.TSP.EPS.IJ.YPL.NEC!>
T-- EQUATION SOLVER
C-- THE ADAMS-MULTON PREDICTOR-CORRECTOR METHOD IS USED
DIMENSION YD1(!6),YLAST(!4),YPRED(!4),YC0RR(6),YP(!4,6).XPA(!4),tPA(!4)
DIMENSION A1(!6),A2(!6).A3(!4),A4(!4)
COMMON/XFRNT,YFRNT,XRER,YRER,NTOTAL(!5),NEiDY.ID
COMMON/XIN,YIN,DP,RL,VX,RUP,YREF,PITDOT,ATK,PIT,UREF.WXP.VYP
COMMON/S(1000).Z(150).IS
COMMON/LOPT, LECM, XREAR, YLO, YUP
IU,YPL, AND ALL THE COMMONS ARE USED EITHER
FOR PROGRAM STOP CONTROL OR FOR OPTIONAL PRINTS
LOOP FOR TIME STEP TO START USING SIMPLE RUNGE-KUTTA METHOD
H=TSP
ISTP=0
END
DO 30 NV=1,NEQ
NSTP=NSTP+1

C ---- SIMPLE ADAMS PREDICTION
DO 30 NV=1,NEQ
30 YPRED(NV)=YLAST(NV)+H*YF1(NV)+H*YF2(NV)+H*YF3(NV)
XPR=YF1(NV)
YP=YF2(NV)
IF(XPR.gt.XFRT) GO TO 111
CALL MODE(XPR,YP,NSTOP,IMAX,NSTP)
IF(NSTOP.NE.0) GO TO 52

111 CALL F(X,LAST,YD1)

C ---- BASFORB CORRECTION
DLYMAX=0.0
30 31 NV=1,NEQ
31 YCORR(NV)=YLAST(NV)+H*YD1(NV)+H*YF1(NV)+H*YF2(NV)
IF(NV.NE.3.AND.NV.NE.4) GO TO 31
31 CONTINUE
IF(ABS(DLYMAX).le.EPS) GO TO 32

C ---- HALVE DX
DO 42 NV=1,NEQ
DO 41 I=1,4
41 YPA(I)=YP(I,NV)
YP(I,NV)=YP(I,NV)+H*YF(I)
42 CONTINUE
NSTP=NSTP-1
X=X-DX
DX=0.50*DX
H024=DX/24.0
DO 30

C ---- Shift variable for next step
32 CONTINUE
DO 32 NV=1,NEQ
FP(NV)=YP3(NV)
FP(NV)=YP3(NV)
YF(1,NV)=YP(1,NV)
YF(2,NV)=YP(2,NV)
YF(3,NV)=YP(3,NV)
YF(4,NV)=YP(4,NV)
YF(5,NV)=YP(5,NV)
YF(6,NV)=YP(6,NV)

C ---- PROGRAM STOP CONTROL AND OPTIONAL PRINT CONTROL
XF=YF1(LAST)/RL
YF=YF1(LAST)/RL
CALL MODE(XF,YP,NSTOP,IMAX,NSTP)
IF(NSTOP.NE.0) GO TO 52
WRITE(11,111) XF,YP,NSTOP

111 CONTINUE
DO 170 I=1,5
WRITE(11,117) XF,YP,NSTOP
170 CONTINUE
END

SUBROUTINE RANG(YMX,YMN,TIMSTP,ADEPS,NEQ)
C-- SEARCH THE UPSTREAM Y-RANGE SUCH THAT NONE OF THE PARTICLES
C-- OUTSIDE THE RANGE WILL HIT THE AIRFOIL.
C--
DO 100 I=1,NEQ
YMX(I)=YMX(I)+H*YF(I)

100 CONTINUE
END
YIN=Y0(IW)
Y(1)=XIN
Y(2)=YIN
Y(3)=UX
Y(4)=UY
Y(5)=PIT*PI
Y(6)=PITDOT*PI

IF(LEQM.NE.1) GO TO 93

C ------ SET INITIAL CONDITIONS OF THE CASE FOR WHICH THE PARTICLE IS EQUILIBRIUM IN THE AIR STREAM
XINN=XIN/RL
YINN=YIN/RL
CALL VELCTY(XINN,YINN,UXfUY)
Y(5)=TAN(UY/UX)
Y(3)=UX*YREF
Y(4)=UY*VREF
Y(6)=0.

92 CONTINUE
YNL=YO(2)/RL
YKL=YO(1)/RL
IF(YKL.GT.YUP) YUP=YKL
IF(YLO.GT.YKL) YLO=YKL
CALL ADAMS(Y • XDO T . T I M E . ADEPS • IUr YP.NEQ)
YL< IU)=YRER
IF(SU< IU> .NE. 99.9999) 00 TO 10
IF(IAC.EQ.l) GO TO 29
IF(IW.LT.2) GO TO 1

29 YY=YL< 1 >*YL<2>
IF(YY.LE.O.O) GO TO 13
IF(YL( 1) .LT.0.0) GO TO 11
GO TO 12

10 IF(IU.EQ.2) GO TO 12
11 Y0( 1 )=Y0( D+HY
IF(IU.EQ.2) IAC=1
IU=0
30 TO 1

12 YO(2)=YO(2)-DY
IAC=0
:u=i

13 YM=Y0(3)
YM=Y0(2)
OF(1)=YO(1)/RL
OF(2)=YO(2)/RL
WRITE (6 .21) YO(1).YO(2)
GO TO 34

14 RETURN

C ------ GOVERNING EQUATIONS OF THE PARTICLE MOTION

DIMENSION X(6) tXIiOT(6)
COMMON/Fl/QfCBARf AMASS. G r Y Y I , ALFAB » U ISC03

C ---- XDOTd )=X1DOT. xriOT(2)=Z1DOT. XDOT < 3 ) =X2DOT
C ---- XDOT(4)=Z2DOT. XDOT ( 5 ) =THETD. XDOT ( 6 ) =TH2DOT

PI=3. 141592654
XP=X< 1)/RL
YP=Y(1)/RL
CALL VELCTY (XP, YP,UX,UY)
RE=V*DP/VISCOS
CALL COEFF(RE,CO,CL,CM)
CD=CF/DF
IF(FL.EQ. 0.0 .AND. CM.EQ.0.0) GO TO 20
IF(VX.EQ.O.) GO TO 17

17 GAMMA=0.5*PI
IF(VY.GT.0.0.AND.VX.GT.0.0) GAMMA=GAMMA+PI
GAMMA=GAMMA+2.*PI
GO TO 18

18 GAMMA=0.5*PI
IF(VY.LT.0.0.AND.VX.LT.0.0) GAMMA=GAMMA+2.*PI
GO TO 19

19 CONTINUE
ALFA=X(5)
ALFA=X(5)
ALFAD=ALFA-ALFA
ALFAD=ALFA
ALFAD=ALFA
ALFAD=ALFA

91
GO TO 21
20 CONTINUE
XDOT(1)=X(3)
XDOT(2)=X(4)
XDOT(3)=X(4)
XDOT(4)=-C*G*VX/AMASS
XDOT(5)=X(4)
XDOT(6)=0.
21 RETURN
END
SUBROUTINE COEFF(RE,CD,CL,CM)
C-- COEFFICIENTS OF AERODYNAMIC FORCES AND OF MOMENTS
C
C= PLR= 0.
IF(RE .EQ. 0.0) GO TO 22
CD=24.0/RE
IF(RE.LE.0.05) GO TO 1
IF(RE.LE.3.0) GO TO 2
IF(RE.GT.330) GO TO 5
AO=-28.339
Al=38.969
A2=0.73204
A3=-0.00056084
GO TO 10
2 A0=0.0
Al=24.167
A2=3.2540
A3=-0.23544
GO TO 10
3 AO=0.0
A1=93.462
A2=0.37576
A3=0.0
GO TO 10
5 CD=1.0
GO TO 10
10 R2=RE*RE
GO TO 14
22 CD=0.0
14 RETURN
END
SUBROUTINE NODE(XF,YF,ISTOP,IW,JNSTP)
C-- DECIDES WHETHER THE PARTICLE HIT THE AIRFOIL OR NOT
C-- aih
C
COMMON/SDS/3(1000),SD(500),SO(150),60(150),l3
COMMON/XRF/X(5),Y(5),B(5),R(5),YMX(5),YMY(5)
COMMON/IN/(XIN,N(INP),VXREF,YREF,RTOT1,ATK,TVIT,VREF+VXP+VYP
COMMON/IDR/XFRNT,YFRNT,XREF,YREF,NTOTAL(NBDY+1)
COMMON/IMF/1P1,1P2,JNTOTM
COMMON/IF/IFL
C
IF(ABS(XF).LE.0.00001) XP=XFRNT
IF(ABS(YF).LE.0.00001) YF=YFRNT
IF(IFL .EQ. 2) GO TO 27
IF(IFL .EQ. 1) GO TO 22
IF(IFL .EQ. 0) GO TO 21
GO TO 20
2 IF(IFL .EQ. 1) GO TO 22
IF(IFL .EQ. 0) GO TO 21
GO TO 20
20 CONTINUE
GO TO 27
21 CONTINUE
GO TO 27
22 CONTINUE
GO TO 27
23 CONTINUE
GO TO 27
24 CONTINUE
GO TO 27
25 CONTINUE
GO TO 27
26 CONTINUE
GO TO 27
27 CONTINUE
GO TO 27
28 CONTINUE
GO TO 27
29 CONTINUE
GO TO 27
92
IF (II .GT. 20) GO TO 51
XNEW=XP
YPEN=XP
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 .LT. 1) GO TO 9
GO TO 34
C ------- FIND THE PARTICULAR TRAJ. ACROSS THE BODY SURFACE OR NOT
90 XNEW=XP
YPEN=YP
YREFN=YREF
IF (IFLAG .EQ. 1) GO TO 8
IF (YPT .GT. YFRNT .AND. YPT .GT. 0.0) GO TO 33
IF (YPT .LT. YFRNT .AND. YPT .LT. 0.0) GO TO 33
3 XNEW=XP
YNEN=YPT
YREFN=YREF
YR=YNEW*YOLD
C ------- FIND INTERSECTION POINT OF SPECIAL CASE
31 YPT=(YP-YREF)/(YREF-YFRNT)
IF (IFLAG .EQ. 0) GO TO 8
IF (YPT .GT. YFRNT .AND. YPT .GT. 0.0) GO TO 33
IF (YPT .LT. YFRNT .AND. YPT .LT. 0.0) GO TO 33
3 XNEW=XP
YNEN=YPT
YREFN=YREF
YR=YNEW*YOLD
C ------- FIND INTERSECTION POINT OF GENERAL CASE
34 XP=(XNEW-XFRNT)*YP-YPOLD)/(YREFN-YREF-YP+YPOLD)
YP=(YP-YPOLD)*YPOLD/(YREFN-YREF-YP+YPOLD)
CONTINUE
C ------- PRINTS DATA IF PARTICLE MOVES OUT OF RANGE
52 XN=XIN/RL
YN=YIN/RL
WRITE(6,53) XN,YN,XP,YP,NSTP
53 FORMAT(2F10.5,' OUT OF RANGE ',2F10.5,1PE12.4,10)
ISTOP=1
SUX(1)=99.9999
IST=1
30 TO 57
C ------- END OF DATA PRINT IF PARTICLE MOVES OUT OF RANGE
C ------- INTERPOLATION OF SURFACE DISTANCE
55 CONTINUE
DO 24 IQ=1,NTOTM
1=10
IM=I+1
IF (YP .GT. YFRNT) GO TO 25
GO TO 21
25 I=NTOTM+IP1-IQ
1(1)=1-1
21 XR1=(XP-X(I))*(XP-X(IM))
TF(XR1 .GT. 0. ) GO TO 24
GO TO 59
CONTINUE
24 CONTINUE
YN=YIN/RL
XN=XIN/RL
SU(1)=N
IS=IS-H
ZO(1)=YN
WRITE(6,56) XN,YN,XP,YP,NSTP
56 FORMAT(2F10.5,' HIT BODY AT ',3F10.5,1PE12.4,10)
ISTOP=1
57 CONTINUE
RETURN
END
SUBROUTINE READIN
DIMENSION A(500),KT(2)
COMMON/COMBYN/VYREF
COMMON/COMBYN1/JJ,NHUBP1,NHUBPI,AL1,EL1,RLTOT,ATOTAL,
1 XOM(500),FACTO(2),COST(2),CSINT(2),
2 YP=150.0,2.3
COMMON/FEI/IFRNT(5)
COMMON/OBTX(X(1000),Y(1000))
DIMENSION XX(500),YY(500)
COMMON/S(S1000),SUS(150),Z0(150),IS
COMMON/XT/XFRNT(1000),YREF,RESTOT(5),NB_DY,IBD
COMMON /BLOCK1/ IDOMF(9),ISDF(9)
COMMON /BLOCK2/ MCUS(500),MARC(500)
COMMON /BLOCK3/ NOAXI,NOOAS,NOOVR,NOV1,NOV2,FIRSTE,LASTE,LASSE,S0,
1 NISMALL
COMMON /BLOCK4/ A11(500),A12(500),A13(500),A21(500),A22(500),A23(500),
1 A24(500)
COMMON/FLG/ HEDR(15),FLG00,FINISH(15)
CASE
1,FLG03,FLG04,FLG05,FLG06,FLG07
1,FLG08,FLG09,FLG10,FLG11
1,FLG12,FLG13,FLG14,FLG15,FLG16,FLG17
COMMON /NNN/ NT, NN, NUNA(N), TYPEA(N), NER1, NER2, NMA, NSIGA, NSIGC, NUNC(N), TYPEC(N), NUNEC(N)

COMMON /KITZEL/ XA(500), YA(500), SIGA(500), SIGC(500)

DOUBBLE PRECISION INTEGER FLG03
REAL INTEGER
COMMON /CL/
X2(500), Y2(500), LS(500)
LOGICAL NOAX1, NOCROS, NOUORT, NOV1, NOV2, FIRSTE, LASTE

READ INFORMATION FOR CALCULATING WIND VELOCITIES

EPS=0.000001
DR=0.0174532925
REUIND 15
READ (15) NTHETA
READ (15) JJ, NHUBP1, NHUBMX, ALIL, BLIL, RHOTOT, ATOTAL, THEA, XQN
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 Al(J, I)=A(J)
12 CONTINUE
REUIND 21
REM 21)IGEOMF.ISIGF
READ (21) X1, Y1, X2, Y2, DELS, SINA, COSA
READ (21) HEDR, CASE, PSF, NB, NNU, TYPEC
READ (21) NT, ND, ilN, NUNA, TYPEA, NER1, NER2, NMA, NSIGA, NSIGC, NUNC, TYPEC, NUNEC, NENS, NSIGEC, TYPEEC, NUNEC
READ (21) HS, HI, NUNA, NMA
READ (21) All, A12, A13, A21, A22, A23
READ (21) SIGA, SIGC
TYPE* 'FINISH READING'
REWIND 5

C——— READ X-, Y- COORDINATES

READ (5, SO) HEDR, CASE, PSF, NB
READ (5, 60) CHORD, MN, TCNS, EPSLON
60 FORMAT (4F10.0)
DO 30 K=1, NB
READ (5, 10) IGEOMF(K), ISIGF(K), ICURVN, NONEUF, IFORMT, NN, MX, MY, THETA, ADDX, ADDY
10 FORMAT (5I1, I5, 5F10.0)
READ (5, 20) BDN, SUBKS, NLF(K), XE, YE
20 FORMAT (3(5X, 5F10.0), 2(10X, F10.0))
94
SUBROUTINE FCEXY(XP,YP,WXCC,YYCC)

COMMON/CBBY/YYREF
COMMON/COH/YNHUBMX,YNHUBP,STIL,STIL*125./216.,
STIL,STIL*125./216.,
COMMON/ioxid/YIXCC(YCC),(SYCSCC(YCC)),
COMMON/XYCC(VXCC(5),CVCC(5)),
VREF=VREF*100./30.48

XP=XP
YP=YP

C---- CALCULATE THE INDIVIDUAL SOLUTIONS
CALL MATRIX(XP,YP)
CALC AXES

C---- TO CALCULATE THE COMBINATION SOLUTION
IF(IT.GT.0.0)
EXCON=1.0954*TOTAL*125./216.

95
SUBROUTINE SINTP(Z,U,N,X1,Y1)
  DIMENSION X(400),Y(400),Z(1),U(1)
  DO 10
  1=1,N
  X(I)=Z(I)
  10 Y(I)=U(I)
  CALL SORTXY(X,Y,N)
  DO 30
  K=I
  IF (X(I) .GT.X(K)) GO TO 15
  IF (X(I) .EQ.X(K)) GO TO 20
  IF (X(I) .LT.X(K)) GO TO 25
  15 CONTINUE
  20 Y1=Y(K)
  30 CONTINUE
  IF (K.EQ.1) GO TO 35
  IF (K.EQ.N) K=N-1
  IF (X(K) .NE.X(K+1)) K=K-1
  Y2=(X(I)-X(K-1))*(X(I)-X(K+1))/(X(K)-X(K-1))/(X(K)-X(K+1))
  R1 = Y(K-1)*U1-Y(K)*U2+Y(K+1)*U3
  RETURN
  35 Y1=0.0
  RETURN
END
SUBROUTINE SORTXY(X,Y,NPTS)
  DIMENSION X(100),Y(100)
  N=NPTS
  JN=N-1
  DO 55 KT=1,JN
    JK=KT+1
    DO 45 JK=JK,N
      IF (XMIN-X(JK)) 45,45,35
      XMIN=X(JK)
      JMIN=JK
    45 CONTINUE
    YMIN=Y(JMIN)
    X(JMIN)=XMIN
    Y(JMIN)=YMIN
  55 CONTINUE
RETURN
END
SUBROUTINE VBARIT(YBAR,ATOTAL,RHOTOT,RHOBAR)
  C APPROACH 5
  TO SOLVE YBAR COMP ITERATIVELY
  VCRIT=ATOTAL/SQRT(1.2)
  IF (VCRIT .LT.2) WRITE (6,30) YBAR,YCOMP,RHOBAR
  RETURN
  10 VUESA+YBAR
  15 VUESA=(VUESA(ATOTAL)**2)*1.0-2.5*VUESA
  VUESA=VUESA-(VUESA)**2/(1.5*VUESA)+VUESA
  IF (ABS(VUESA-VCOMP).LT.0.0001) GO TO 15
  IF (VCOMP.GE.VCRIT) VCOMP=5*(VUESA/VCRIT)
  VUESA=VCOMP
  IF (VCRIT .GT.20) GO TO 15
  GO TO 10
  15 RHOBAR=(1.0-.2*(VCOMP-ATOTAL)**2)*.5*RHOTOT
  IF (VCRIT .GT.20) WRITE (6,30) YBAR,YCOMP,RHOBAR
  GO TO 10
  30 FORMAT (1H0.34HI EXCEEDS 20 ITERATIONS FOR RHOBAR , 5X , 7HyB AR =• • -1PE10.3,2X,8HyCOMP =• , 1PE10.3,2X,9HRHOBAR = -1PE10.3/70H V6AR HAS BEEN REDUCED TO yCOMP»RHOBAR/RHOTOT , UHERE yCOKP = \cR I T
cICAL )

96
C BEGINNING OF ELEMENT ASINE/

FUNCTION ARSIN(X)

ARSIN = ASIN<X>

RETURN

END

C BEGINNING OF ELEMENT AXIS/

SUBROUTINE AXIS

COMMON/NNN/ NT, N0(11), NGO(11), TYP2G(5)

THETA(5), NER1, NER2, NMA, NTA0, NTS1G, NSIGC

NMA(5), TYPEC(5), NLF(11), TEC, NSIGEC

RETURN

END

C BEGINNING OF ELEMENT CROSS/

SUBROUTINE CROSS

COMMON/NN/ NT, NGO(11), NMA, NTA0, NTS1G, NSIGC

NMA(5), TYPEC(5), NLF(11), TEC, NSIGEC

RETURN

END

C BEGINNING OF ELEMENT ELINT/

SUBROUTINE ELINT(XKSQ,XN,PHI,PIE)

THIS SUBROUTINE CALCULATES THE INCOMPLETE ELLIPTIC INTEGRAL OF THE

THIRD KIND. THE ARGUMENTS ARE:

X KSQ VALUE OF IS SQUARED

X N VALUE OF MINUS ALPHA SQUARED

PHI VALUE OF PSI

PIE VALUE OF INCOMPLETE ELLIPTIC INTEGRAL OF THIRD KIND

DATA HP /1.570794/

DATA ROUND / .0000005 /

RETURN
IF (FNL. LT. (-1.1)) GO TO 470

IF (P) 10, 470, 20

NORMALIZE FMI

A = -1.
P = P

GOTO 30

40 IF (ABS(P-1.570796) .LE. 0.001) GO TO 100

IF (P-HP) 110, 100, 40

J=F/(2.*HP)

X=J

P1=P*X*HP

F=HP

GOTO 100

50 D=SUM

X=0.

IF (P-HP) 60, 70, 80

60 P=P+P

XXX=1.

GOTO 110

70 PIE=(XXX+1.)*A*0

GOTO 460

80 XXX=-1.

X=XX*2.

P=2.*HP-P1

I8OTQ 110

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.0.64) GOTO 120

120 SUM=SUM+P*PI/RT

GOTO 440

130 SUM=SUM+P*PI/RT

GOTO 440

C POWER SERIES IN N AND K SQUARED

SA=1.

SB=SN/2.

CA=C

FB=F

SUM=SUM+SA*FB

SA=SA+SB

CA=CA*CA

SUM=SUM+CA

IF (SUM.LT.0.001) GO TO 460

140 F=FM+1.

CB=C*S

GOTO 460

150 FM=FM+1.

CB=C*B

GOTO 460

160 PM=SK

ST=SQR(1.+FN)

IF (ST.NE.0.) GO TO 170

170 IF (C.GT.4.E-3) GOTO 180

180 G=ATAN(S/ST)/RT

GOTO 190

190 G=ATAN(S/ST)/RT

C ADDITION FORMULAS

ZP=SQR((1.-SK*(S2+1.))*FM)

EP=(ZP/S+P)/E

GOTO 460

200 IF (SN.LT.(-1.)) GO TO 320

C ADDITION FORMULA

ZP=SQR((1.-SK+Z2))

EP=(Z2/ZP)+(-.5*FM*E

GOTO 460

GOTO 300

J=10

IF (S.LT.C) GO TO 320
220 IF(RT1.NE.0.) GO TO 240
   RT1=RT1/11.1)
   IF(FN.NE.(-1.)) GO TO 330
   CF=(.99999SK200*(3+C*Z)/Z) /SKP
   GOTO 300
230 CF=+ST*(S(-2./R)+S/S*Z) /SKP
   GOTO 300
240 IF(FM=FM+SK), LT.0.) GO TO 260
250 CF=FM/RT1*ATAN(XP)
   GOTO 300
260 IF(ABS(XP).GE.0.1)GOTO 270
   XP=XP**2
   Y=2.4*XP*(1.1*Y/1.3+Y*(2+Y/7.))
   GOTO 390
270 Y=ALOG(1+XP)/(-1.-XP))
   CF=FM**2)/(2.*RT1)
   GOTO 300
300 BB=-1.
310 S=SORT(SST)
   C=SQRT(1.-SST)
   GOTO 320
320 U=1. /C
   V=1. /C
   T=V
330 CF=(U+V) /GOTO 370
340 IF(D.F.10.5)**GOTO 360
   POWER SERIES IN 1+F AND 1-(K SQUARED)
   T=1.
   B=-0.5*S
   A=1.14/2.0
   E=U/V*4.5
   F=0.
   SUM=1.
   X=SINT(1.E-8)
350 CA=+SINT(1.E-8)
   AL=(1.+SINT(1.E-8)/(1.+SINT(1.E-8))
   X=CA
   SUM=SUM+X
   IF(ABS(X).LT.T)GOTO 430
   FM=FM+1.0
   B=(1.+SINT(1.E-8)/(1.+SINT(1.E-8)) 
   C=CB/0.
   B=B+0.
   A=B+0.
   E=BEAU
   GOTO 350
360 RT=SINT(1.RT+1.E-3)
   POWER SERIES IN 1-(K SQUARED)
   if (RT.GT.6.E-3)
   370 GOTO(1.RT+1.E-3)/1.RT)
   GOTO 340
380 G=0.
390 S=0.
400 SUM=SUM+X
   PX=SP
   SX=0.5
   FM=0.
410 B=(R-D/R)
   D=1.14/2.0
   Y=1.14/2.0
   X=1.14/2.0
   SUM=SUM+X
   IF(ABS(X).LT.T)GOTO 420
   FM=FM+1.0
   AP=AP+1.
   SF=AP+1.
   SUM=SINT(1.E-8)
   GOTO 410
420 SUM=SINT(1.E-8)
430 IF(RB.LT.0.)GOTO 310
440 IF(B) 50, 90, 450
450 PIE=ABSUM
460 PIE=PIE+PIE*ROUND
   RETURN
470 50 40.
480 S=0.
490 PIE=PIE+PIE*ROUND
   RETURN
   CALL ELIPCXK,EKX,EKX)
   C START
   ETA = 1. - XE
   E = ETA**1.0 ETA = 0.000005
   40 ELM=ALOG(ETA)
   EKX = 1.386274800 + ETA # (.964974E-1 + ETA #
   1.0-.964974E-1 + ETA # (.1290095E-1 + ETA #
   1.145116E-1)) - ELM # 1.5 + ETA #(.1249859E0
   EXIT
**BEGINNING OF ELEMENT ELLLC/**

**SUBROUTINE ELLLC (AfK.E.I)**

This subroutine calculates the associated complete elliptic integrals of the first or second kind.

The arguments are:
- A (Argument, $k^2$) for which $E'$ or $E$ will be found.
- K (Value of associated complete elliptic integral of first kind).
- E (Value of associated complete elliptic integral of second kind).
- I (If EQ 0, compute $E$; if EQ 2, compute $E'$).

**DOUBLE PRECISION A,E,CON(32),ALN4,CF(29),CL(3),DLOG**

**EQUIVALENCE (CON(I)>CF(I)) (CON(30),CL(1))**


43580-2

7432D-2

32290-4

9024D-2

1.5049181733401883 D-4. 4. 4314713'. 12155304D-1.5. 43054573744952.1874220447184198D-2,1.2510592410344444D-2,1.5377102523552019D-2,7.3354144974290345D-3,7.0980944039932


**DATA CL / 1.55251299480407210-2,3.48334794358944920-3,1.442721079117048025D-4 /**

**LN4 = 1.3842943411198900**

**IF (A.EQ.0.0) GO TO 40**

**GO TO (10,20),I**

10 K = LN4 + (((((CON(a)*A+CON(7) )>A + CQN,i) >*A + CQN<5> )-«A+CON(4> )*A)

+f CQN<3) )*AtCON<2) )*A-fCON( 1) )*A - DLOG < A > * < 0 . 5+« C ( « ( CON < 14 > * At

2CON'. 15))*AtCON(14))*A + CON(13))*AtCON(12:)«ATCON( 11))*A + CON(9))»A)

10 RETURN

20 E = 1.ODOt(((<(<(CON(24)*A+CON<23)>*A+CGN<22>)*AtCON<21))*A+CON(20

l))*AtCCN<19))*A + CON(18))«A-tCON(17))*A - BLOGCA>*<(((<<«CON<32>*A

2t CON(31)XA+CONC30))*A+CQN(29))*A+CON^23))*A+CQN<27))«A+CON<26))*

3A+CON(2S))*A)

30 RETURN

40 K = 0.99999999030

E = 1.000

RETURN

**END**

**BEGINNING OF ELEMENT HLAMB/**

**SUBROUTINE HLAMB (BETA,K)**

This subroutine calculates the Heuman's lambda function of beta and K.

The arguments are:
- BETA (Value of beta).
- K (Value of K).

**DOUBLE PRECISION A,F,£**

**CALL INEL (FI,E,PI,A,PHI,SKI,K3,K2,K1)**

H = $1.0 - K^2$

**CALL ELLC (A ,F,E,1)**

= $F - E$

**RETURN**

**BEGINNING OF ELEMENT INNEL/**

**SUBROUTINE INEL (F,E,PI,A.PHI,SKI,K3,K2,K1)**

This subroutine calculates the incomplete elliptic integrals of the first, second and third kinds. The arguments are:
- F (Value of incomplete elliptic integral of the first kind).
- E (Value of incomplete elliptic integral of the second kind).
- PI (Value of incomplete elliptic integral of the third kind).
- A (Value of alpha squared).
- PHI (Value of phi squared).
- SKI (Value of K squared).
- K3 (If EQ 0, do not compute PI; if NE 0, compute PI).
- K2 (If EQ 0, do not compute E; if NE 0, compute E).
- K1 (If EQ 0, do not compute F; if NE 0, compute F).

**DOUBLE PRECISION ARG,FD,ED**

**RA = 0.4344197724/**

**CALL INEL (FI,E,PI,A,PHI,SKI,K3,K2,K1)**

H = $1.000+((((((CON(24)*A+CON(23))*A+CON(22))*A+CON(21))*A+CON(20

l))*A+CON(19))*A+CON(18))*A+CON(17))*A+CON(16))*A+CON(15))*A+CON(14))*A+CON(13))*A+CON(12))*A+CON(11))*A+CON(10))*A+CON(9))»A)

**RETURN**

**END**

**FUNCTION HLAMB (BETA,K)**

This subroutine calculates the Heuman's lambda function of beta and K.

**CALL INEL (F,E,PI,A,PHI,SKI,K3,K2,K1)**

H = $1.000$

**RETURN**

**FUNCTION INEL (F,E,PI,A.PHI,SKI,K3,K2,K1)**

This subroutine calculates the incomplete elliptic integrals of the first, second and third kinds. The arguments are:
- F (Value of incomplete elliptic integral of the first kind).
- E (Value of incomplete elliptic integral of the second kind).
- PI (Value of incomplete elliptic integral of the third kind).
- A (Value of alpha squared).
- PHI (Value of phi squared).
- SKI (Value of K squared).
- K3 (If EQ 0, do not compute PI; if NE 0, compute PI).
- K2 (If EQ 0, do not compute E; if NE 0, compute E).
- K1 (If EQ 0, do not compute F; if NE 0, compute F).

**DOUBLE PRECISION ARG,FD,ED**

**RA = 0.4344197724/**

**CALL INEL (F,E,PI,A,PHI,SKI,K3,K2,K1)**

H = $1.000$

**RETURN**
BEGINNING OF ELEMENT MATRIX

SUBROUTINE MATRIX(XPKT, YPKT)

COMMON / BLOCK1/ IDEOMF(9), ISIOF(9)
COMMON / BLOCK3/ NOAXI, NOCROS, NOU1, NOU2, FIRSTE, LASTE, LSSE, SO

COMMON / BLOCK4/ HCURU(500), HARC(500)

DO 150 K = 1, MB

IF (ISIOF(K)) .EQ. 1 NOU2 = .TRUE.

DO 170 J = M1 + 1

IF (J .EQ. N1) LASTE = .FALSE.

CALL FARAEUXPNT, YPKT)

170 FIRSTE = .FALSE.

JL = JL + 1

C MATRIX FORMULATION OF COLUMNS FOR THIS BODY HAVE BEEN COMPLETED.

IF (NLF(K) .GT. 0) GO TO 220

DO 190 J = N1 + 1

UN(A) = UN(A) + AVJ

190 VTK = VTK + AVJ

220 CONTINUE

RETURN

***************BEGINNING OF ELEMENT PARAB

SUBROUTINE PARAB(XPKT, YPKT)

COMMON / BLOCK1/ IDEOMF(9), ISIOF(9)

COMMON / BLOCK3/ NOAXI, NOCROS, NOU1, NOU2, FIRSTE, LASTE, LSSE, SO

COMMON / BLOCK4/ HCURU(500), HARC(500)

DO 150 K = 1, MB

IF (ISIOF(K)) .EQ. 1 NOU2 = .TRUE.

DO 170 J = M1 + 1

IF (J .EQ. N1) LASTE = .TRUE.

CALL PARAB(XPKT, YPKT)

170 FIRSTE = .FALSE.

JL = JL + 1

C MATRIX FORMULATION OF COLUMNS FOR THIS BODY HAVE BEEN COMPLETED.

IF (NLF(K) .GT. 0) GO TO 220

DO 190 J = N1 + 1

UN(A) = UN(A) + AVJ

190 VTK = VTK + AVJ

220 CONTINUE

RETURN

***************BEGINNING OF ELEMENT MATRICES COMPUTATION

C CONTROL FOR PARABOLIC X+Y+Z MATRICES COMPUTATION

COMMON / CL/ SI(500), TI(500), AI(500), RI(500), SI(500),

COMMON /CL/ S0(500), T0(500), A0(500), R0(500),

COMMON / CL/ S(500), T(500), A(500)
REAL LSSE

C OFF-BODY...
J1P = J1 + 1
J1 = (XPK-T1(J1))**2 + (YPK-T1(J1))**2
J2 = (XPK-T2(J2))**2 + (YPK-T2(J2))**2
J3 = (XPK-T3(J3))**2 + (YPK-T3(J3))**2
DMSQ = 0
IF (D1 .LT. DMSQ) DMSQ = D1
IF (D2 .LT. DMSQ) DMSQ = D2
MIN = MIN(DMSQ)
IF (MIN .EQ. 0.0) GO TO 60
HI = 10.*MARC(J)/MIN + 0.9
IF (HI .LT. 0.0) GO TO 50
NS = MARC(J)
GO TO 80
50
NI = 2*NI
GO TO 60
60
HI = MARC(J)/64.
GO TO 80
70
XI = XI
NS = 2.*MARC(J)/XI
NI = NI + 1
80
XJ = XJ
GO TO 80
RETURN
END

C***BEGINNING OF ELEMENT PRAB2/ ***C

SUBROUTINE PARAB2(XPKT,YPKT)

THIS ROUTINE CALCULATES THE VELOCITY CONTRIBUTIONS OF
OFF-DIAGONAL AND ENDS OF THE ON-DIAGONAL) ELEMENTS BY A
SIMPSON-RULE. ELEMENTS MAY BE FLAT OR CURVED, WITH PIECEWISE
CONSTANT-, LINEAR-, OR QUADRATIC-SOURCE DENSITY.

COMMON /BLOCK2/HCURY(500), HACRY(500)
JOMMQN /BLOCK3/NOAXI,NOCROS,NOVORT,NOV1,NOV2,FIRSTE,LASTE,LSSE,30.
NSMALL
..JMMO.M /BLOCN4/A11(500) ,A12(500) ,A13(500), A21(500) ,A22(500)
,;A23(500)
COMMON /CL/ X11500), YK500), X21500), Y21500), DELH500).
SINA(SOO),COSA500)
COMMON /TL/ A(SOO), B(500), AX(SOO), AY(500), A2(500).
1 CX(500), CY(500), CZ(500), AXY(500) .AY'(500) .!
J. J1. SJ. DS.
J.>
DX. OYf NI, XJ, YJ,
3
AK, EKB, EKK, K, PF
LOGICAL NOAXI , NOCROS , NOVORT , NOV1 , NOV2 , FIRSTE , LASTE
LOGICAL DF.YSMALL

SEAL LSSE
DIMENSION NOAXIX(129),NOAXIY(129),VOCRSX(129),VOCR5Y(129).
1 .VOCR5Z(129), VOCRSZ(129). VOCR5X(129),
0 .'VOCR5Y(129), ARING(129), YRING(129)
DATA PI/3.1415927/

C OFF-BODY

X = XPK
Y = YPK

C SIGNAL=SINAJ)
COSAI=COSAJ)
HC = HCURVI)

IF (FIRSTE) M=J+1
IF (LASTE) M=J-1
MM1=M + 1

BEGIN BY GENERATING THE INDUCED VELOCITIES AT THE ENDS OF NI
INTERVALS (OF UNIFORM LENGTH)....

3 = 50
GO TO 110

110 L = 1

C CALCULATE LOCAL SOURCE SADUS, QA, AND AXIAL LOCATION, SB.

QA = YJ + SINAL*S+HC*COSAL*SSQ
SB = XJ + COSAL*S-HC*SINAL*SSQ
XRING(L) = OB
YRING(L) = QA
IF (QA .GT. 0.0)
GO TO 39
V0AXIX(L) = 0.0
VOAXIY(L) = 0.0
VOCR5X(L) = 0.0
VOCR5Y(L) = 0.0
VOCR5Z(L) = 0.0
VOCR5(L) = 0.0
GO TO 110

39
39 XM5=6-09
XM5=0 XM5=1
YP5 = (T-0A)*2
YP5 = TYP5 + XM55Q
X = 4.30807/T1
CALL ELIP(KA,KEK)

SUBROUTINE ELIP RETURNS THE VALUES AS EKX AND EEX.

!SO = TYP5

102
\[
\begin{align*}
A_{SO} &= \text{Gass}^2 \\
R_{O} &= \text{SORT(T1)} \\
T2 &= (A_{SO} + \text{XMBq}) \\
T3 &= T_{MBq} + A_{SO} \\
T4 &= (\text{ROOT1}(\text{YSG} + A_{SO} - \text{XMBq}))^{2} \\
T3_{SQ} &= T_{3}^{2} \\
R_{SHALL} &= \frac{(\text{YSq}/T_{3}) - 0.0001}{T2} \\
& \text{IF (YSMALL) GO TO 40} \\
& \text{GO TO 50} \\
40 &= \text{ROOT32} = (\text{SORT}(T_{3}))^{2} \\
T4 &= \frac{\text{PI}^{\text{QA}/\text{ROOT32}}}{T3_{SQ}} \\
& \text{IF (NOAXI) GO TO 70} \\
& \text{IF (YSMALL) GO TO 60} \\
& \text{VOAXIX(L)} = Q1 * \text{OA}^{XMB} \\
& \text{VOAXIY(L)} = -Q2 * \text{QA}/Y \\
& \text{GO TO 70} \\
60 &= \text{VOAXIX(L)} = \frac{-2 * XhEi * T4}{1.0 + 0.75 * \text{SCi}} \cdot T3_{SQ} \\
& \text{VOAXIY(L)} = \frac{-Y * T4}{2.0 + 0.125 * \text{YSq} * 2 * XhBSq} \\
& \text{GO TO 90} \\
70 &= \text{IF (NOAXI) GO TO 120} \\
& \text{CALL SIMSON(VOAXIX,NI,DS,VOAX)} \\
& \text{CALL SIMSON(VOAXIY,NI,DS,VOAY)} \\
& \text{AX(J)} = AX(J) + VOAX \\
& \text{AY(J)} = AY(J) + VOAY \\
& \text{IF (NOAXI) GO TO 170} \\
& \text{DO 180 L = 1,NI} \\
& \text{VOCRX(L)} = VOCRX(L)^{2} \\
& \text{VOCRY(L)} = VOCRY(L)^{2} \\
& \text{VOCRZ(L)} = VOCRZ(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRX,NI,DS,VOCX)} \\
& \text{CALL SIMSON(VOCRY,NI,DS,VOCY)} \\
& \text{CALL SIMSON(VOCRZ,NI,DS,VOCR)} \\
& \text{CX(J)} = CX(J) + VOCX \\
& \text{CY(J)} = CY(J) + VOCY \\
& \text{CZ(J)} = CZ(J) + VOCZ \\
& \text{IF (NOAXI) GO TO 200} \\
& \text{S = S + DS} \\
& \text{DO 210 L = 1,NI} \\
& \text{VOCRX(L)} = VOCRX(L)^{2} \\
& \text{VOCRY(L)} = VOCRY(L)^{2} \\
& \text{VOCRZ(L)} = VOCRZ(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRNX,NI,DS,VCRX)} \\
& \text{CALL SIMSON(VOCRNY,NI,DS,VCRY)} \\
& \text{CALL SIMSON(VOCRNZ,NI,DS,VCRZ)} \\
& \text{EXR(MJ)} = EXR(MJ) + A11(J) * V1CRX \\
& \text{EYR(MJ)} = EYR(MJ) + A11(J) * V1CRY \\
& \text{EZR(MJ)} = EZR(MJ) + A11(J) * V1CRZ \\
& \text{SXR(MJ)} = SXR(MJ) + A12(J) * V1CRX \\
& \text{SYR(MJ)} = SYR(MJ) + A12(J) * V1CRY \\
& \text{SZR(MJ)} = SZR(MJ) + A12(J) * V1CRZ \\
& \text{IF (NOAXI) GO TO 250} \\
& \text{S = S + DS} \\
& \text{DO 260 L = 1,NI} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAX)} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAY)} \\
& \text{AX(J)} = AX(J) + VOVX \\
& \text{AY(J)} = AY(J) + VOVY \\
& \text{GO TO 260} \\
& \text{S = S + DS} \\
& \text{DO 290 L = 1,NI} \\
& \text{VOCRQ(L)} = VOCRX(L) + VOCRQ(L) \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CX(J)} = CX(J) + VORX \\
& \text{CY(J)} = CY(J) + VORY \\
& \text{CZ(J)} = CZ(J) + VORZ \\
& \text{GO TO 320} \\
& \text{S = S + DS} \\
& \text{DO 330 L = 1,NI} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAX)} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAY)} \\
& \text{AX(UJ)} = AX(UJ) + VOVX \\
& \text{AY(UJ)} = AY(UJ) + VOVY \\
& \text{GO TO 330} \\
& \text{S = S + DS} \\
& \text{DO 360 L = 1,NI} \\
& \text{VOCRQ(L)} = VOCRX(L) + VOCRQ(L) \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CX(J)} = CX(J) + VORX \\
& \text{CY(J)} = CY(J) + VORY \\
& \text{CZ(J)} = CZ(J) + VORZ \\
& \text{GO TO 390} \\
& \text{S = S + DS} \\
& \text{DO 400 L = 1,NI} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAX)} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAY)} \\
& \text{AX(UJ)} = AX(UJ) + VOVX \\
& \text{AY(UJ)} = AY(UJ) + VOVY \\
& \text{GO TO 400} \\
& \text{S = S + DS} \\
& \text{DO 430 L = 1,NI} \\
& \text{VOCRQ(L)} = VOCRX(L) + VOCRQ(L) \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CX(J)} = CX(J) + VORX \\
& \text{CY(J)} = CY(J) + VORY \\
& \text{CZ(J)} = CZ(J) + VORZ \\
& \text{GO TO 460} \\
& \text{S = S + DS} \\
& \text{DO 470 L = 1,NI} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAX)} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAY)} \\
& \text{AX(UJ)} = AX(UJ) + VOVX \\
& \text{AY(UJ)} = AY(UJ) + VOVY \\
& \text{GO TO 470} \\
& \text{S = S + DS} \\
& \text{DO 500 L = 1,NI} \\
& \text{VOCRQ(L)} = VOCRX(L) + VOCRQ(L) \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CALL SIMSON(VOCRQ,NI,DS,VOCRQ)} \\
& \text{CX(J)} = CX(J) + VORX \\
& \text{CY(J)} = CY(J) + VORY \\
& \text{CZ(J)} = CZ(J) + VORZ \\
& \text{GO TO 530} \\
& \text{S = S + DS} \\
& \text{DO 540 L = 1,NI} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{VORAIX(L)} = VORAIX(L)^{2} \\
& \text{S = S + DS} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAX)} \\
& \text{CALL SIMSON(VORAIX,NI,DS,VORAY)} \\
& \text{AX(UJ)} = AX(UJ) + VOVX \\
& \text{AY(UJ)} = AY(UJ) + VOVY 
\end{align*}
\]
AX(MM1) = AX(MM1) + A21(J)*V2AX
AX(MM1) = AX(MM1) + A21(J)*V2AX
AX(MM1) = AX(MM1) + A21(J)*V2AX
AX(MM1) = AX(MM1) + A21(J)*V2AX
AX(MM1) = AX(MM1) + A21(J)*V2AX
AX(MM1) = AX(MM1) + A21(J)*V2AX

220 IF (NOCROS) GO TO 250
B = 50
DO 230 L = 1, NI
VCREX(L) = VCREX(L)*S
VCREY(L) = VCREY(L)*S
VCREZ(L) = VCREZ(L)*S

230 V = 5 + DS
CALL SIMSON(VCREX,NI , CPUX,2CPUX)
CALL SIMSON(VCREY,NI , CPUY,2CPUY)
CALL SIMSON(VCREZ,NI , CPUZ,2CPUZ)

CX(MM1) = CX(MM1) + A21(J)*V2CX
CY(MM1) = CY(MM1) + A21(J)*V2CY
CZ(MM1) = CZ(MM1) + A21(J)*V2CZ

250 RETURN
END

SUBROUTINE QC ( OMEG, QM, Q )
C
C THIS SUBROUTINE CALCULATES THE LEGENDRE FUNCTIONS OF THE SECOND KIND
AND HALF ORDER. THE ARGUMENTS ARE:
OMEG ARGUMENT FOR WHICH LEGENDRE FUNCTIONS WILL BE FOUND
QM VALUE OF LEGENDRE FUNCTION OF MINUS ONE HALF ORDER
Q VALUE OF LEGENDRE FUNCTION OF PLUS ONE HALF ORDER

DOUBLE PRECISION OMEGD, ARG, A, F, E, QMD, QD
IRG = 2.0/(OMEGD-0.0)
H = 0.0-ARG
CALL ELLC (A,F,E,H)
CALL ELLC (A,F,E,2>H)
C)MD = F*ARG**0.5
1
1D = -E*(2.0*(OMEGD+1.0))**0.
QM=QMD
0 = QD
RETURN
END

SUBROUTINE SIMSON( Y,N>DXf AREA)
C
C THIS ROUTINE INTEGRATES Y OVER (N-1) INTERVALS OF EQUAL LENGTH, DX,
YIELDING THE ENCLOSED AREA. (N MUST BE AN ODD INTEGER.)

DIMENSION Y(1>
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(H-1)
AREA = ( ABS(DX)/3.0) * SUM
RETURN
END

SUBROUTINE LINEAR(X,Y,IMAX,IMIN)
DIMENSION X(1), Y(1)
N=IMAX-IMIN
N=IMAX-IMIN
IF1=IMAX-IMIN
WRITE(6,199) NTOT
199 FORMAT(6,I93) I,YMN,XMN,YMN
DO 10 I=IMIN,IMIN
XMN=0.5*(X(I-H>-X(H>-X(I)>>
YMN=0.5*(Y(I+1)-Y(I) )
AX=ABS(XMN)
IF (AX .LE. 0.0000001) GO TO 10
XMN=0.5*(X(I+1)-X(I) )
YMN=0.5*(Y(I+1)-Y(I) )
WRITE(6,93) I,YMN, XMN,YP
93 FORMAT(6, I93) I,YMN, XMN,YP
10 CONTINUE
RETURN
END

SUBROUTINE EFFICY(NCOEF,NPTS,NS)
IMPLICIT REAL*8(A-H,0-Z)
REAL* 4 XIN,YIN,YN,YOUT,YF-OUT, XN,XMAY,XMIN, XMAX,DFD,PLCC, EPT.
XIN(1000) = X(1000)
XMAX(100) = X(100)
XMIN(100) = X(100)
DFD(10) = DFD(10)
PLCC(10) = PLCC(10)
EPT(10) = EPT(10)
N=1
NS=0
N=NCOEF+1
DO 10 1=NS+1, NS+1
1=NS+1, NS+1
NS=NS+1
ENDIF .EO. NS/2) NDA-IRS(IJK)
104
READ(3) NDATA
IF(NSI .EQ. 1 .OR. MSI .GT. 1) NPTS=NDATA
N2=NDATA-N1
DO 43 I=N1,N2
READ(3,9) XA(I),YA(I)
43 CONTINUE

DO 97 I=N1,N2
IF(ABS(XA(I)) .LE. 0.0000001) XA(I)=0.0
543 CONTINUE

UWRITE(6,72) IJK J
IF(NSI .GT. 1) WRITE(6,71)DPD(IJK),PLWC(IJK)
WRITE(6,73) NDATA
UWRITE(6,74)
DO 97 I=N1,N2
97 UWRITE(6,75) I,YA(I),XAU(I)

IF(NSI .GT. 1) WRITE(6,72) IJK J
IF(NSI .GT. 1) WRITE(6,71)DPD(IJK),PLWC(IJK)
WRITE(6,73) NDATA
UWRITE(6,74)
DO 97 I=N1,N2
97 UWRITE(6,75) I,YA(I),XAU(I)

XMIN=XA(N1)
XMAX=XA(N2)
XINC=XA(N1)-XA(N2)
XMAX=XA(N2)

CALL TREP(XMIN,YOUT,YPOUT,XA,YA,N1,N2,NCDEF,NPTS)
DO 3 II=1,NCQEF
B(II,NN)=A(II,NN)
3 CONTINUE
N1=NDATA+N1
5 CONTINUE

UWRITE(6,98) XMIN,XMAX,EPT
CALL EF(XMAX,XMIN,NDATA,MSI,NCDEF,PLUC)
RETURN

2 UWRITE(6,99) I
DO 6 I=1,NCQEF
C(II)=C(II)-1
6 CONTINUE

UWRITE(6,93) I,YYF+YAI(I),YYD
GO TO 22

DO 7 I=1,N1
IF(ISI .EQ. 1) GO TO 21
CALL TREP(XMIN,YOUT,YPOUT,XA,YA,N1,N2,NCDEF,NPTS)
21 IF(ISI .EQ. 1) GO TO 22
YYD=-YPOUT
GO TO 23

IF(ISI .EQ. 1) GO TO 22
YYD=0.0
YYP=A(1,NN)
DO 9 J=2,NCQEF
IF(XA(J) .LE. 0.0) XA(J)=0.0000001
YYP=YYP+A(J,NN)*XAI(J)**(J-1)
YYD=YYD+(J-1)*A(J,NN)*XAI(J)**(J-2)
9 CONTINUE

YYD=-YYD
IF(YYD .LT. 0.0) GO TO 27
UWRITE(6,93) I,YYF+YAI(I),YYD
GO TO 28

27 NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
GO TO 29

29 YTOP=0.0
UWRITE(6,93) I,YYF+YAI(I),YYD
NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
GO TO 20

20 CONTINUE

UWRITE(6,92)
UWRITE(6,99)NTOA
UWRITE(6,99)
WRITE(6,95)
WRITE(6,99)
UWRITE(6,93) I,YYF+YAI(I),YYD
NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
NTOTA=NTOTA+1
IAC=IAC+1
UWRITE(6,93) I,YYF+YAI(I),YYD
GO TO 20

24 CONTINUE

C---- FORMATS
70 FORMAT(/2X,'POLYNOMIAL LEAST SQUARE FIT',/2X)
1 'NO. OF COEFFICIENTS IS ',I3/2 'NO. OF POINTS FOR DATA FITTING IS',I3/2 'DIAMETER','PERCENTAGE',/2 'DISTRIBUTION',/2 'OF W. CONT./2 'X',/2 'DIST',/2 'OF W. CONT./2 'TOTAL COLLECTION EFFICIENCY',/2 'LOWER SURFACE LIMIT',/2 'UPPER SURFACE LIMIT',/2 'TOTAL COLLECTION EFFICIENCY',/2
UWRITE(9)NTOA
DO 24 I=1,NTOA
UWRITE(9) XBTA(I),YBTA(I)
24 CONTINUE

C---- FORMATS
70 FORMAT(/2X,'POLYNOMIAL LEAST SQUARE FIT',/2X)
1 'NO. OF COEFFICIENTS IS ',I3/2 'NO. OF POINTS FOR DATA FITTING IS',I3/2 'DIAMETER','PERCENTAGE',/2 'DISTRIBUTION',/2 'OF W. CONT./2 'X',/2 'DIST',/2 'OF W. CONT./2 'TOTAL COLLECTION EFFICIENCY',/2 'LOWER SURFACE LIMIT',/2 'UPPER SURFACE LIMIT',/2 'TOTAL COLLECTION EFFICIENCY',/2
UWRITE(9)NTOA
DO 24 I=1,NTOA
UWRITE(9) XBTA(I),YBTA(I)
24 CONTINUE

C---- FORMATS
70 FORMAT(/2X,'POLYNOMIAL LEAST SQUARE FIT',/2X)
1 'NO. OF COEFFICIENTS IS ',I3/2 'NO. OF POINTS FOR DATA FITTING IS',I3/2 'DIAMETER','PERCENTAGE',/2 'DISTRIBUTION',/2 'OF W. CONT./2 'X',/2 'DIST',/2 'OF W. CONT./2 'TOTAL COLLECTION EFFICIENCY',/2 'LOWER SURFACE LIMIT',/2 'UPPER SURFACE LIMIT',/2 'TOTAL COLLECTION EFFICIENCY',/2
UWRITE(9)NTOA
DO 24 I=1,NTOA
UWRITE(9) XBTA(I),YBTA(I)
24 CONTINUE
SUBROUTINE EFCXMAX,XMIN,NDA,NCOEF,PLUC
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XADOO) TPLUC( ) > >XBTAdOO) .YBTAdOO)
REAL* 4 XMAX,XMIN.PLUC,YYD.DES.XMAdO.XMAd>..XA
COMMON/CORE1/B<20.30>
COMMON/CORE2/XMAdO>.XMM,0.30>
DES=MXMAX-XMIN)/FLOAT (NDA-1)
IAC=0
NDA=NDA
WRITE(*)NCOEF
WRITE(*,2) DO 1 IJ=1,NDA
IAC=IAC-1
YYD=0.0
XA( IJ)=XMAX-DES(NDA-IJ)
DO 2 I=1,NSI
IF(XA(I) .GT. XMAd).OR. XA(I).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)*XAdJ)**(J-2)*PLUCd>
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(XA(IJ) .GE. XA(IJ) )GO TO 5
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 YBTAdIAC)=YYD
XBTA(IAC)=XA(IJ)
2 CONTINUE
WRITE(6,94) NCOEF
WRITE(6,199)NTOA
WRITE(6,199)NTOA
DO 24 I=1,NTOA
WRITE(6,95)I,XBTA(I),YBTA(I)
24 CONTINUE
C ---- FORMATS
199 FORMAT(IS)
RETURN
END
SUBROUTINE TERF<XIN,YOUT,YPOUT,XA,YA,N1,N2,NCOEF,NPTS)
C ---- DOES SIMPLE NCOEF POLYNOMIAL LEAST SQUARES FIT TO NPTS OF
C DATA.
IMPLICIT REAL*8(A-H,O-Z)
REAL*4 XIN,YOUT»YPOUTfXA(l).YA(l)
DIMENSION FC1000.20) .Y< 1000)
COMMON/COEF/A< 20.21)
1VCM1 = NCOEF - 1
C ---- SEARCH XA FOR XIN
DO 1 IF = N1.r(2
I*IF
IF(XIN.LE.XA(IF) )GO TO 5
1 CONTINUE
1 CONTINUE
IMIN=.NS/2
IF(XIN.LT.XA(1)) IMIN=1
MAX=IMIN+NPTS-1
IF(MAX.GT.MN) MAX=MN
II=1
II = IMIN+IMAX
III = II + 1
DO 3 JJ = 2,NCOEF
IF(II(IJ) .LE. XA(JJ)) III=III+1
3 II = II
A(JJ) = NPTS
DO 10 J = 1,NCOEF
[10 II = I+J
A(J) = NPTS
DO 30 I = 1,NCOEF
[30 I = I+J
A(J) = NPTS
SUM = 0.0
DO 1 ISUM = 1,NPTS
[1 SUM = SUM + F(ISUM,1)*F(ISUM,J)
[1 SUM = SUM + F(ISUM,1)*F(ISUM,J)
A(JMCOL) = SUM
1 CONTINUE
SUM = 0.0
DO 30 I = 1,NPTS
[30 SUM = SUM + Y(I)*F(I,J)
A(JMCOL) = SUM
 106
SUBROUTINE CHOLES(N,M,SYN,DA)

CDPD MATRIX SOLUTION ROUTINE FOR DOM TOOH ON 4-24-80

REAL A(MxN),SUM, TEMP

IF(A(i,i) .NE.0.0) GO TO 47

DO 37 J=1,N

IF(A(j,j) .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)=TEMP

57 CONTINUE

47 DO 2 J=2,M

A(J,J)=A(J,J)/A(J,1)

2 CONTINUE

DO 41 I=2,N

DO 7 J=2,M

IF(A(i,j) .EQ.0.0) GO TO 49

IF(J.GT.I) GO TO 69

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

A(I,J)=A(I,J)/SUM

51 CONTINUE

A(I,J)=A(I,J)-SUM

GO TO 12345

49 IF(J.GT.I) GO TO 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

54321 N=-1

12345 RETURN

END

SUBROUTINE APLLOT(U,YL,N,K,LDP,FC,CZ)

DIMENSION U(N),YL(N),N(K)

DIMENSION XTITL(20),YTITL(20)

DIMENSION DSNAME(9)

CALL CCBEGN(10.0,1.0)

CALL BEGID(IOP)

CALL PLOT(0.0,0.0,3)

CALL AXIS(0.0,0.0,XTITL,-90,XLENG,0.,FC,0.0)

CALL PLOT(0.0,0.0,3)

CALL PLOT(YTITL,0.0,0.0,90,YMIN+FC,0.0)

100 FORMAT(20A4)

101 FORMAT(5F10.0)

CALL PLOT(XMAX,YMAX,YHEIT)

CALL PLOT(XMIN,YMIN,YHEIT)

CALL PLOT(XMAX,YMIN,YHEIT)

CALL PLOT(XMIN,YMAX,YHEIT)

CALL PLOT(YMAX,YMIN,YHEIT)

CALL PLOT(YMIN,YMAX,YHEIT)

19 CONTINUE

JH=1

DO 14 I=1,N

JH=JH+1

14 CONTINUE

CALL BEGID(LDP,-1,OSNAME)

CALL CCBEGN(-1.0,1.0)

RETURN

END

SUBROUTINE TRAJCT RETURN END
The following computer programs and data files are recorded on this magtape.

<table>
<thead>
<tr>
<th>The order in magtape</th>
<th>Description</th>
<th>Name suggested in IBM system</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Particle trajectory computer program for axisymmetric inlet case.</td>
<td>SOURCE.TRAJ3</td>
</tr>
<tr>
<td>2</td>
<td>Axisymmetric potential flow computer program (EOD)</td>
<td>SOURCE.EOD</td>
</tr>
<tr>
<td>3</td>
<td>Axisymmetric COMBYN computer program</td>
<td>SOURCE.COMBYN</td>
</tr>
<tr>
<td>4</td>
<td>Input data file (Tape 05) for geometry generation program SCIRCL (Axisymmetric inlet)</td>
<td>DAT.AI005S</td>
</tr>
<tr>
<td>5</td>
<td>Input data file Tape 17 (renamed on Tape 05) for EOD</td>
<td>DAT.AI005Y</td>
</tr>
<tr>
<td>6</td>
<td>Input data file (Tape 05) for COMBYN. (Axisymmetric inlet with $\alpha = 0^\circ$, $M=0.4$)</td>
<td>DAT.AI005C</td>
</tr>
<tr>
<td>7</td>
<td>Input data file (Tape 05) for COMBYN (Axisymmetric inlet with $\alpha = 30^\circ$, $M = 0.4$)</td>
<td>DAT.AI305C</td>
</tr>
<tr>
<td>8</td>
<td>Input data file (Tape 02) for Trajectory program, SOURCE.TRAJ3. (Axisymmetric inlet with $\alpha = 0^\circ$, $M = 0.4$)</td>
<td>DAT.AI002T</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Source</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>9</td>
<td>Input data file (Tape 02) for Trajectory program, SOURCE.TRAJ3. (Axisymmetric inlet with $\alpha = 30^\circ$, $M=0.4$)</td>
<td>DAT.AI302T</td>
</tr>
<tr>
<td>10</td>
<td>Two-Dimensional potential flow computer program, Y24. (Original program)</td>
<td>SOURCE.Y240</td>
</tr>
<tr>
<td>11</td>
<td>Two-dimensional COMBIN computer program. (Original program)</td>
<td>SOURCE.COMBINO</td>
</tr>
<tr>
<td>12</td>
<td>Two-Dimensional COMBIN computer program. (Original program). This program is almost the same as 11. We didn't use this one.</td>
<td>Up to you.</td>
</tr>
<tr>
<td>13</td>
<td>Axisymmetric potential flow computer program, EOD. (Original program)</td>
<td>SOURCE.EODO</td>
</tr>
<tr>
<td>14</td>
<td>Axisymmetric COMBYN computer program. (Original program)</td>
<td>SOURCE.COMBYNO</td>
</tr>
</tbody>
</table>
### Title and Subtitle

Particle Trajectory Computer Program for Icing Analysis of Axisymmetric Bodies

### Authors

Walter Frost, Ho-Pen Chang, and Kenneth R. Kimble

### Performing Organization Name(s) and Address(es)

FWG Associates, Inc.
Rt. 2, Box 271-A Lakewood Drive
Tullahoma, Tennessee 37388

### Sponsoring/Monitoring Agency Name(s) and Address(es)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

### Distribution/Availability Statement

Unclassified - Unlimited
Subject Category 03

### 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.